Title
Adaptive Techniques for Mitigating Circuit Imperfections in High Performance A/D Converters

Author
Ting, Shang Kee

Publication Date
2014

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Adaptive Techniques for Mitigating Circuit Imperfections in High Performance A/D Converters

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Electrical Engineering

by

Shang Kee Ting

2014

Abstract of the Dissertation

Adaptive Techniques for Mitigating Circuit Imperfections in High Performance A/D Converters

by

Shang Kee Ting

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2014

Professor Ali H. Sayed, Chair

In this dissertation, we examine the effect of four sources of circuit imperfections on the performance of analog-to-digital converters (ADCs), including sampling clock jitters, spurious sidebands, timing mismatches, and gain mismatches. These imperfections distort the sampled data and degrade the signal-to-noise ratio (SNR) of the ADCs. We develop signal models for the distortions and propose effective adaptive signal processing techniques to filter the sampled data and mitigate the spurious effects. Rather than remove the distortions by perfecting the circuitry, the proposed techniques focus on processing the sampled data by using adaptive DSP algorithms.

Analog circuit impairments create many distortions including I/Q imbalances, phase noise, frequency offsets, and sampling clock jitter. Timing jitters generally arise from noise in the clock generating crystal and phase-locked-loop (PLL). The jitters cause the ADCs to sample the input signals at non-uniform sampling times and introduce distortion that limits the signal fidelity and degrades the SNR. While the effects of jitter noise can be neglected at low frequencies, applications

requiring enhanced performance at higher frequencies demand higher SNR from the sampling circuit. We first examine the effect of the clock jitter on the SNR of the sampled signal and subsequently propose compensation methods based on a signal injection structure for direct down-conversion architectures.

We also address the effect of non-ideal PLL circuitry on the quality of the sampled data. In a non-ideal PLL circuit, leakage of the reference signal into the control line produces spurious tones. When the distorted PLL signal is used to generate the sampling clock, it injects the spurious tones into the sampled data. These distortions are harmful for wideband applications, such as spectrum sensing, since they affect the detection of vacant frequency bands. We again examine the distortion effect in some detail and propose techniques in the digital domain to clean the data and remove the PLL leakage effects. We study the performance of the proposed algorithms and compare it against the corresponding Cramer-Rao bound (CRB).

We further propose an adaptive frequency-domain structure to compensate the effect of timing and gain mismatches in time-interleaved ADCs. An M-channel time-interleaved ADC uses M ADCs to sample an input signal to obtain a larger effective sampling rate. However, in practice, combining ADCs introduces mismatches among the various ADC channels. In the proposed solution, the signal is split into multiple frequency bins and adaptation across the frequency channels is combined by means of an adaptive strategy. The construction is able to assign more or less weight to the various frequency channels depending on whether their estimates are more or less reliable in comparison to other channels.

The dissertation of Shang Kee Ting is approved.

Robert M'Closkey

Danijela Cabric

Dejan Markovic

Ali H. Sayed, Committee Chair

University of California, Los Angeles

2014

Table of Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

I would like to thank my advisor, Professor Ali H. Sayed, for his guidance during my Ph.D. program. His careful reading and critique of my work helped me dig deeper and improve the results on a continual basis.

Next, I would like to acknowledge the members and visitors of the Adaptive Systems Laboratory. I learned a lot from them, and they have provided lots of good advice and inspiration. I will remember the times when we – Zaid Towfic, Sheng-yuan Tu, Jianshu Chen, Xiaochuan Zhao, and Chung-Kai Yu – went for group outings, house-warming, and Turkey dinners. I would also like to thank some past members from the lab – Federico Cattivelli, Qiyue Zou and Zhi Quan, who helped me during my first year. It was also great to meet the visiting scholars who passed by our lab - Jesus F. Bes, Mohammad Reza Gholami, Sergio Valcarcel-Macua, Milad A. Toutounchian, Reza Abdolee, Jae-Woo Lee, Victor Lora, Paolo Di Lorenzo, Alexander Bertrand, Oyvind L. Rortveit, Hongyang Chen, Jingon Joung, Eva Hamou, Henri Pesonen, and Noriyuki Takahashi.

I would also like to thank my family for everything they have done over the years. Firstly, my parents who brought me up and supported me. My Dad who gave me a head-start in Mathematics, and my Mom who gave me lots of love and encouragement. I am sad that Mom departed during my studies, but I will always remember her. Also, my brothers who took care of my parents while I was not around. Finally, I am very grateful to have a loving and supportive wife, Lee Ngee. Although the journey towards our PhDs has been challenging, I am lucky that we are accompanying each other through it. Moreover, it is really amazing that our daughter, Clare, was born during this journey. Let us continue our journey together hand-in-hand.

Vita

1976	Born, Singapore.
2001	B.S. (Electrical Engineering), National University of Singapore (NUS), Singapore.
2002	M.S. (High Performance Computation for Engineered Systems), NUS.
2002–2008	Senior Member of Technical Staff, DSO, Singapore.
2008–present	PhD Candidate, Department of Electrical Engineering, University of California, Los Angeles, USA.

CHAPTER 1

Introduction

1.1 Imperfections in analog-to-digital conversion

One of the key components in radio and communication devices is the analog-to-digital converter (ADC). In modern communication systems, there is a trend to miniaturize radio devices and, yet, increase their flexibility to handle higher carrier frequencies and larger bandwidths. For example, certain applications of modern radios, such as cognitive radios and UWB radios [3], may require ADCs operating at high sampling rates due to the use of wide frequency bands. However, these circuit requirements are generally hard to meet in current practice and have cost implications on hardware design [4, 5], especially since variations in the fabrication processes make it difficult to control RF/analog circuit impairments. Furthermore, the desire to reduce costs by simplifying circuit design can only accentuate the problem. Circuit impairments create many distortions, some of which manifest themselves in the form of I/Q imbalances, phase noise, frequency offsets, and sampling jitter. Advances in digital processing and VLSI techniques enable designers to use elaborate digital signal processing (DSP) methods to reduce the effects of these impairments in the digital domain at more affordable costs than trying to perfect the circuits and the fabrication processes [6–13]. In this dissertation, we will be applying DSP techniques to mitigate several sources of circuit imperfections in ADC design.

In the ideal case, an ADC should sample the input signal at uniform intervals. However, due to circuit imperfections, this is not the case. For example, the ADC requires a sampling clock that triggers it at the correct instants. One way to generate the sampling clock is to use a phase-locked loop (PLL) frequency synthesizer. The PLL uses a reference signal to control the voltage-controlled ocillator (VCO) that produces the clock signal. However, the reference signal can leak into the control line of the VCO, and this leakage signal creates spurious tones in the clock signal. As a result, spurious sidebands are introduced into the sampled data. In applications such as spectrum sensing in cognitive radios, spurious tones from primary signals might give a false positive detection on actual free channels. Another source of imperfections in ADCs is due to the phase noise of the sampling clock. The phase noise creates random perturbations in the sampling instants of the ADC. This random jitter reduces the signal-to-noise ratio (SNR) of the sampled data.

An alternative way to sample the input signal without requiring faster ADCs is to interleave multiple ADCs in order to produce an effective higher sampling rate [14–16]. This ADC architecture is called time-interleaved ADC (TI-ADC). Since each ADC operates at a slower rate, the clock will have less distortions thereby reducing the distortion effects due to the sampling clock. This technique, however, introduces other problems such as mismatch in the delay of the clock fed into each ADC, the gain of each ADC, and DC offset between ADCs.

This dissertation focuses on problems related to spurious sidebands and random jitter in single-channel ADCs, and timing and gain mismatches in TI-ADCs. We will examine how these imperfections affect the sampled data, and propose adaptive signal processing solutions to mitigate their effects. In Chapter 2, we study the effect of spurious sidebands in the sampling clock of the ADC, and

propose a solution to estimate and remove the sideband distortions. In Chapter 3, we extend the results to spectrum sensing applications where we propose a low-complexity solution that reuses the existing spectrum sensing modules, and analyze the impact of spurious sidebands on spectrum sensing. In Chapter 4, we modify the proposed structure of Chapter 2 to mitigate sampling errors caused by random jitter in the clock signal. Next, in Chapter 5, we study and propose solutions for the timing and gain mismatches in TI-ADC. More details are provided in the next two sections where we discuss some of our contributions and summarize the work in each chapter.

1.2 Contributions

In this dissertation, we study the effect of circuit imperfections on both ADCs and TI-ADCs, and propose DSP techniques to mitigate the problems due to the imperfections. Specifically, we examine the distortions that arise due to the imperfect sampling clock, which generates spurious sidebands and random jitter in the sampled data of ADCs. We also examine the distortions that are due to the gain and timing mismatches in the TI-ADCs. For each type of distortion, we approach the problem in the following way. First, we examine the effect of the imperfection on the sampled data, and provide system models that describe the distortions. Next, using the system models, we propose algorithms that estimate and remove the distortions from the sampled data using DSP techniques. We also carry out performance analysis to predict the theoretical limits of performance and compare against simulated results. The results show that the proposed solutions are effective in reducing the distortions.

1.3 Organization

The organization of the dissertation is as follows.

- **Chapter 2**: This chapter first examines the PLL and how the leakage of the reference signal into the control line of the VCO creates spurious tones in the sampling clock and the sampled data [17]. We show that the sideband distortions in the sampled data can be expressed as a function of some parameters. Using a training signal, we propose an estimation scheme that estimates the distortion parameters online, and a compensation scheme that corrects the distorted signal. The Cramer-Rao bound for estimating the distortion parameters is derived and compared against the simulation results. The simulations also examine the effect of bit resolution, amplitude of the training signal, additional noise in the system, e.g., random jitter in the sampling clock or in the training signal, on the performance of the proposed solution.

- **Chapter 3**: In this chapter, we extend our Chapter 2 and examine the impact of spurious tones in spectrum sensing applications [18]. In these applications, the presence of spurious sidebands can lead to false detection of signals in otherwise empty channels. Here, we assume that the PLL is in tracking mode (when the loop is in lock) and the distortion parameters are estimated using a training signal before spectral sensing. In spectrum sensing applications, a commonly used module is one that performs the discrete Fourier transform (DFT) or the fast Fourier transform (FFT). To reduce hardware complexity and computation cost, we propose an algorithm that uses the FFT block to estimate the sampling errors from the spurious sidebands. We also analyze the effects of the spurious sidebands

on spectrum sensing. Theoretical analysis of the detection performance in the presence of the distortions is derived and it shows that the detection performance is degraded. Computer simulations are included to show that the proposed solution can remove the spurious sidebands and improve the detection performance.

- **Chapter 4**: In this chapter, we extend the work from Chapter 2 to handle distortions due to random jitter in the sampling clock. We further extend the work to the case where the signal is down-converted into in-phase and quadrature-phase components before they are sampled [19]. We analyze the performance of the proposed techniques in some detail and provide supporting simulations.

- **Chapter 5**: In this chapter, we develop and analyze an adaptive frequency-domain structure to compensate the effects of timing and gain mismatches in TI-ADCs [20, 21]. The solution eliminates some of the conditions and limitations of prior approaches and is able to deliver enhanced performance. The signal is split into multiple frequency bins and adaptation across the frequency channels is combined by means of an adaptive strategy. The construction is able to assign more or less weight to the various frequency channels depending on whether their estimates are more or less reliable in comparison to other channels. Analysis and simulations are used to illustrate the superior performance of the proposed technique.

- **Chapter 6**: The last chapter conclude the dissertation and discuss future research directions.

CHAPTER 2

Digital Suppression of Spurious PLL Tones in ADCs

This chapter focuses on the distortions caused by the spurious sidebands that are induced by the imperfections in the sampling clock of an ADC [17]. The sampling clock is usually generated by a phase-locked loop (PLL) frequency synthesizer. Spurious tones in the ADC clock result from leakage of the reference signal in the PLL into the control line of the voltage-controlled oscillator (VCO). As a result, spurious sidebands are introduced into the sampled data. In applications such as spectrum sensing in cognitive radios, spurious tones from primary signals might give a false positive detection on actual free channels [18]. Conventional ways to mitigate the problem include reducing mismatch in the charge pump (CP) and using large capacitors in the loop filter of the PLL [22, 23].

Other approaches [24, 25] include increasing the complexity of the circuit design. For example, [24] proposed using multiple phase-frequency detectors (PFD) and CPs that operate in delay with respect to one another. This approach reduces the magnitude of the spurs, and shifts the frequency of the sidebands away from the frequency of the PLL clock. Reference [25] proposed adding another PFD, integrators and voltage-controlled current sources. The additional components are used to reduce the distortions after the PLL is locked.

Other ways to mitigate the problem is to change the architecture of the ADC.

Some approaches [14–16] interleave several ADCs in order to produce an effective higher sampling rate. Since each ADC operates at a slower rate, the clock will have lower distortions thereby reducing the distortion effects. This technique, however, introduces other problems such as mismatch in the delay of the clock fed into each ADC, the gain of each ADC, and DC offset between ADCs. In [1, 14, 15], methods to estimate and remove these mismatches are proposed. Later in Chapter 5, we will present a new compensation technique, which gives better performance than existing solutions. Another form of ADC that differs from conventional impulse sampling is the weighted integration sampler. References [26–29] describe and analyze integration sampling circuits with internal antialiasing filtering. The integration sampler creates an internal filter, which can be used to reduce distortions. However, the integration sampler is more complex than a conventional impulse sampler. The basic component in the integration sampler is a charge sampling circuit. The circuit contains a capacitor that is first charged, then sampled and finally discharged. This process is repeated continuously. The integration sampler is designed either using multiple charge sampling circuits that are time-interleaved, or using a charge sampling circuit that has a larger sampling frequency in comparison to the desired channel bandwidth.

These compesation techniques are largely in the analog domain. We pursue a different approach to the problem by using digital signal processing (DSP) techniques. DSP techniques rely on processing the data algorithmically in the digital domain, which is a more affordable approach than trying to perfect the circuitry. There already exist works that handle various types of distortions in the ADC via digital signal processing methods. For example, in [30], a technique was proposed to remove jitter in narrowband signals with the help of a reference signal. This method was improved in [31] and used to handle jitter errors in OFDM signals. References [32,33] extended the method to bandpass signals with

an input reference signal. Reference [34] analyzed the effects of finite aperture time and sampling jitter in wideband data acquisition systems. Reference [35] addressed a problem in front-end ADC circuitry involving nonlinear frequency-dependent errors using calibration signals. These works were proposed to solve the distortions caused by the random jitter in the ADC. In this chapter, we consider the distortions due to the spurious tones in the sampling clock of the ADC. We will show that the spurious tones in the sampling clock give rise to a deterministic (as opposed to random) distortion. We further show that the effect of these distortions can be modeled by a few parameters. Moreover, the estimation of these parameters can be improved by using a longer integration time. Consequently, the distortions in the sampled data can be removed more effectively.

Rather than reduce the PLL sidebands, we propose a method to estimate the sidebands imparted on a sinusoid training signal and then use this information to compensate for the distortions caused by spurious sidebands on the actual sampled data. This is done by estimating the distortion errors caused by the sidebands and using an interpolation scheme to remove their effect in the *digital domain*. The work here is based on [17], which expands on the earlier and shorter work [36] and provides detailed derivations, derives the Cramer-Rao bound for the estimation error, compares the estimation performance against the CRB, and simulates the proposed algorithm under various types of noise and parameters.

The chapter is organized as follows. Section 2.1 discusses a mathematical model for the VCO clock and what happens when the reference signal is leaked into the control line of the VCO. Section 2.2 derives a model for the non-ideal sampling instants of the ADC. Section 2.3 shows the effect of the non-ideal sampling model on a sinusoidal tone. Section 2.4 proposes an architecture to remove

the distortion effect caused by the PLL sidebands. Section 2.5 develops a method to estimate the distortion errors from the sampled data; the errors are used in the proposed architecture in section 2.4. We also derive a Cramer-Rao bound for the estimates to illustrate the performance of the estimation algorithm. Section 2.6 provides computer simulations and section 2.7 summarizes the paper.

2.1 Effect of leakage on the clock signal

In this section and the next one, we develop an analytical model that captures the effect of spurious PLL sidebands on the sampling time instants. Our aim is to arrive at an expression that describes the resulting fluctuations in the sampling times of the ADC.

2.1.1 Source of leakage

To begin with, in [22] and [37], a VCO is defined as a circuit that generates a periodic clock signal, $s(t)$, whose frequency is a linear function of a control voltage, V_{cont}. Let the gain of the VCO and its "free running" frequency be denoted by K_{vco} and f_s, respectively. The generated clock signal is described by

$$s(t) = A_s \sin\left(2\pi f_s t + K_{\text{vco}} \int_{-\infty}^{t} V_{\text{cont}} dt\right) \tag{2.1}$$

To attain some desired oscillation frequency, the quantity V_{cont} is set to an appropriate constant value. However, the generated signal, $s(t)$, may not be an accurate tone due to imperfections. To attain good frequency synthesis, the clock signal is divided and fed back into a control block that consists of a phase-detector (PD) and a low-pass filter (LPF) as shown in Figure 2.1 [22, 37]. The PD/LPF block compares the divided frequency clock signal with a low-frequency reference signal at f_{ref} and makes adjustments to V_{cont}.

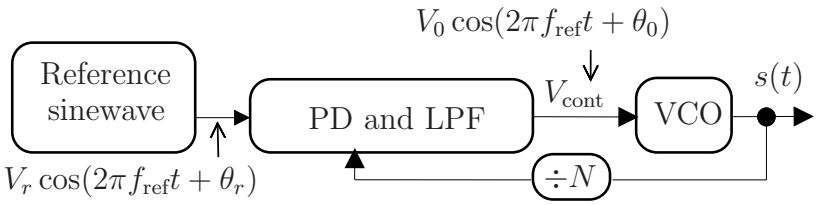


Figure 2.1: Block diagram of a PLL.

The reference leakage into the control line of the VCO is typically due to the imperfections in the PD and LPF before the VCO. A non-ideal PD leaks the reference signal and a non-ideal LPF fails to remove the leakage fully. Interested users can refer to [22,38–40] for detailed explanations. This leakage feed-through causes the PLLs to have spurious tones. The periodic leakage signal has a fundamental frequency at f_{ref} and may have higher harmonic components. However, the most dominant component in the leakage signal is the one at f_{ref}. This most dominant component also relates directly to the most dominant spurious tones in the distorted clock signal. Here, we assume that the leakage signal is a sinusoidal signal at f_{ref}. We will show that the sinusoidal leakage creates sampling offsets that are defined by a sinusoidal expression at the same frequency. Furthermore, once the parameters in the sinusoidal expression are estimated, we can compensate for the distorted sampling offsets. Similarly, any periodic signal can be represented by a summation of sinusoidal signals at multiples of f_{ref}. The sampling offsets that are created by the summation of sinusoidal signals will also result at some summation of sinusoidal expression. More interestingly, there is a unique relationship between each sinusoidal signal in the leakage signal and a pair of sinusoidal expression in the sampling offsets. Thus, the work in this chapter (which examines sinusoidal leakage) can be be extended for periodic signals as well.

2.1.2 Effect of leakage

Due to imperfections in the circuitry, the reference signal leaks into the control line of the VCO. We assume that the presence of the LPF before the control line of the VCO attenuates the leakage to some extent (but is not able to remove it completely) so that it is reasonable to assume the variable C_0 further ahead in (2.4) satisfies $C_0 \ll 1$. For simplicity, we assume that the desired clock signal at f_s is obtained when V_{cont} is 0. Now, suppose there is leakage from the reference signal so that V_{cont} becomes

$$V_{\text{cont}} = V_0 \cos(2\pi f_{\text{ref}} t + \theta_0) \tag{2.2}$$

for some $\{V_0, \theta_0\}$. Then, the output of the VCO becomes

$$s(t) = A_s \sin(2\pi f_s t + C_0 \sin(2\pi f_{\text{ref}} t + \theta_0) + \phi_s) \tag{2.3}$$

where ϕ_s is some unknown phase offset and

$$C_0 = \frac{K_{\text{vco}}}{2\pi f_{\text{ref}}} V_0 \tag{2.4}$$

We will be analyzing the signal model with respect to an arbitrary reference of time. Using a change of variables, let $t = t' - \frac{\phi_s}{2\pi f_s}$ and substitute t into (2.3). The new equation is similar to (2.3) except that ϕ_s is 0. Therefore, we can set $\phi_s = 0$ without loss of generality. Using a trigonometric identity, (2.3) can be expressed as

$$\begin{aligned} s(t) = {} & A_s \sin(2\pi f_s t) \cos(C_0 \sin(2\pi f_{\text{ref}} t + \theta_0)) \\ & + A_s \cos(2\pi f_s t) \sin(C_0 \sin(2\pi f_{\text{ref}} t + \theta_0)) \end{aligned} \tag{2.5}$$

Using $\sin(x) \approx x$ and $\cos(x) \approx 1$ when x is small, and the assumption $C_0 \ll 1$, then (2.5) can be approximated as:

$$\begin{aligned} s(t) &\approx A_s \sin(2\pi f_s t) + A_s C_0 \cos(2\pi f_s t)\ \sin(2\pi f_{\text{ref}}\ t + \theta_0) \\ &= A_s \sin(2\pi f_s t) + \frac{A_s C_0}{2} \sin(2\pi (f_s + f_{\text{ref}})t + \theta_0) \\ &\quad - \frac{A_s C_0}{2} \sin(2\pi (f_s - f_{\text{ref}})t - \theta_0) \end{aligned} \tag{2.6}$$

The value of C_0 determines the relative power. For example, if the ratio of the sideband power $(A_s C_0)^2/8$ to the desired clock power $(A_s^2/2)$ is -50 dBc to -70 dBc, then C_0 would be in the range 6.32×10^{-3} to 6.32×10^{-4}.

2.2 Non-ideal sampling and distortion model

The distorted signal, $s(t)$, in (2.3) is often used as the sampling clock for an ADC. The leakage in $s(t)$ results in some deterministic distortions on the sampling instants. To analyze the effect of these distortions, we derive an approximate model for the sampling offsets first, and then examine the accuracy of the model.

2.2.1 Sampling instants

We start by determining the sampling instants of the ADC that would result from using (2.3) as a clock signal. For ease of notation, define $\epsilon_s(t)$ and $\epsilon_s[n]$ as

$$\epsilon_s(t) = \frac{C_0}{2\pi f_s} \sin(2\pi f_{\text{ref}} t + \theta_0) \tag{2.7a}$$

$$\epsilon_s[n] \triangleq \epsilon_s(t)|_{t=nT_s} \tag{2.7b}$$

The sampling instants of the ADC are the zero-crossings of (2.3). Using (2.3) and (2.7) and defining $T_s = 1/f_s$, the sampling instants, t_n, of the ADC must

satisfy the condition:

$$2\pi f_s \left(t_n + \epsilon_s(t_n)\right) = 2\pi n \tag{2.8}$$

or, equivalently,

$$t_n + \epsilon_s(t_n) = nT_s \tag{2.9a}$$

$$t_n = nT_s - \epsilon_s(t_n) \tag{2.9b}$$

This is a nonlinear equation in t_n. We solve it as follows. Let

$$t_n \triangleq nT_s + e[n] \tag{2.10}$$

for some perturbation terms $e[n]$ that we wish to determine. From (2.9) we have that

$$\begin{aligned} e[n] &= -\epsilon_s(t_n) \\ &= -\epsilon_s(nT_s - \epsilon_s(t_n)) \end{aligned} \tag{2.11}$$

Since $C_0 \ll 1$, we know that $\epsilon_s(t_n)$ is bounded by

$$|\epsilon_s(t_n)| \leq \frac{C_0}{2\pi f_s} \ll T_s \tag{2.12}$$

Therefore, the discrete sequence of offsets $e[n]$ is approximated as

$$e[n] \approx -\epsilon_s[n] \tag{2.13}$$

The next section provides a bound for the approximation.

2.2.2 Accuracy of model

Let $\hat{x}_n$ refer to the approximate value (i.e., $-\epsilon_s[n]$ for the true value x_n (i.e., $e[n] = -\epsilon_s(t_n)$). For brevity's sake, the relative error bound is stated here and the derivations are shown in Appendix 2.A. The relative error bound is found to be:

$$\left|\frac{\hat{x}_n - x_n}{x_n}\right| \leq \frac{\gamma(1+\gamma)}{1-\gamma} \tag{2.14}$$

where

$$\gamma \triangleq C_0 \frac{f_{\text{ref}}}{f_s}, \quad 0 < \gamma < 1 \tag{2.15}$$

For example, if the ratio of the sideband power ($A_s^2 C_0^2/8$) to the desired signal power ($A_s^2/2$) in (2.6) is -50 dBc, then $C_0 = 6.32 \times 10^{-3}$. Suppose the frequency of the clock signal, f_s, and the frequency of the reference signal, f_{ref}, are 1 GHz and 20 MHz, respectively. Then $\gamma = 1.26 \times 10^{-4}$. In this case, we conclude from (2.14) that the relative error is upper bounded by 1.26×10^{-4}. The error bound shows that the model (2.13) approximates well the perturbations.

2.3 Effect of sampling distortions on ADC performance

Using the sampling model (2.10) and (2.13) derived in the previous section, we can examine the effect of the spurious PLL tones on the performance of the ADC. Let the input signal to the ADC be

$$w(t) = A_w \cos(2\pi f_w t + \phi_w) \tag{2.16}$$

Using (2.10), the distorted sampled signal, $\breve{w}[n]$, is given by

$$\begin{aligned} \breve{w}[n] &\triangleq w(t)|_{t=t_n} \\ &= w(nT_s + e(n)) \quad \text{(distorted sample)} \end{aligned} \tag{2.17}$$

Let

$$w[n] \triangleq w(t)|_{t=nT_s} \quad \text{(desired sample)} \tag{2.18a}$$

$$\dot{w}[n] \triangleq \dot{w}(t)|_{t=nT_s} \tag{2.18b}$$

From Taylor series approximations, we know that when $|y - a|$ is small, a differentiable function $f(y)$ can be approximated to first-order by

$$f(y) \approx f(a) + (y - a)\dot{f}(a) \tag{2.19}$$

in terms of the derivative of f at a. If we set $a = nT_s$ and $y = nT_s + e[n]$, and apply (2.19) to $w(t)$ we find that $\breve{w}[n]$ and $w[n]$ are related via

$$\breve{w}[n] \approx w[n] + e[n]\dot{w}[n] \tag{2.20}$$

Let

$$w_c[n] = \cos(2\pi f_w n T_s + \phi_w) \tag{2.21a}$$

$$w_s[n] = \sin(2\pi f_w n T_s + \phi_w) \tag{2.21b}$$

$$\dot{w}[n] = -2\pi f_w A_w w_s[n] \tag{2.21c}$$

The term $e[n]\dot{w}[n]$ in (2.20) can be expressed using (2.13) as

$$e[n]\dot{w}[n] = 2\pi f_w A_w w_s[n]\epsilon_s[n] \tag{2.22}$$

Using (2.7) and (2.21b), the above equation can be expressed as

$$\frac{f_w A_w C_0}{2f_s}\left[\cos(2\pi(f_w - f_{\text{ref}})nT_s + \phi_w - \theta_0) - \cos(2\pi(f_w + f_{\text{ref}})nT_s + \phi_w + \theta_0)\right] \tag{2.23}$$

This represents two sideband frequencies at $f_w \pm f_{\text{ref}}$. These results show that when the input $w(t)$ is a tone at frequency f_w, then the sampled data, $\breve{w}[n]$, will consist of three sinusoids at f_w and $f_w \pm f_{\text{ref}}$.

An interesting observation in the sampled data is that the ratio of the power of the sidebands (2.23) to the carrier signal (2.16) is smaller compared to the case in the spurious clock signal in (2.6). To observe this, recall from the last paragraph of Section 2.1 that if the sideband of the clock, $s(t)$, is at -50 dBc, then $C_0 = 6.32 \times 10^{-3}$. Suppose f_w and f_s are chosen to be 40 MHz and 1 GHz, respectively. From $w[n]$ and (2.23), the ratio of the power of the sideband to the power of the carrier signal is approximately:

$$20\log_{10}\left(\frac{f_w}{f_s}\frac{C_0}{2}\right) = -78 \text{ dBc} \tag{2.24}$$

Figure 2.2 shows a realization of the distorted PLL clock at 1GHz and the sampled sinusoidal tone at 40 MHz. The distorted PLL clock and sampled signal are simulated using the expressions in (2.3) and (2.17), respectively. The power ratio of the sideband to carrier signal in the PLL and sampled signal are -50.8 dBc and -78.2 dBc, respectively.

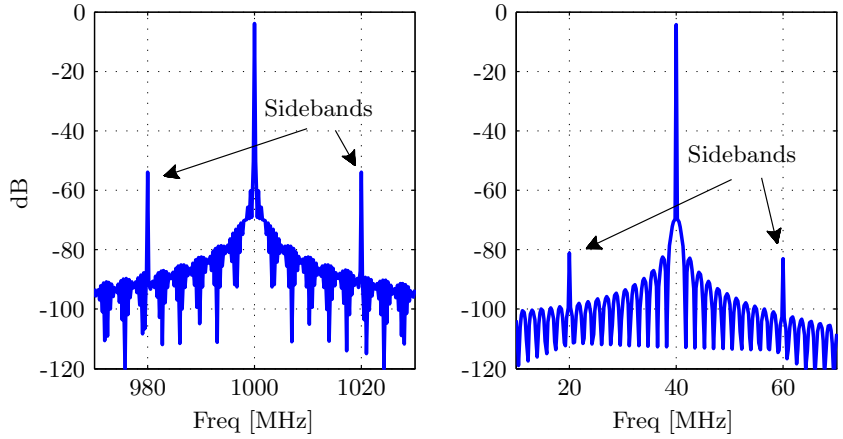


Figure 2.2: The left figure shows the power spectral density (PSD) of the distorted PLL clock at 1 GHz and the right figure shows the PSD of the sampled sinusoidal tone at 40 MHz.

2.4 Sideband suppression

The previous section showed how the input tone is distorted by the offsets $e[n]$ (see (2.20)). If $e[n]$ were known, then we could remove its effects. From (2.7) and (2.13), $e[n]$ is dependent on the value of the parameters C_0 and θ_0. Therefore, our first step towards compensating for the effect of $e[n]$ is to estimate $\{C_0, \theta_0\}$.

One initial approach is to inject a training sinusoidal signal into the ADC and sample it before acquiring any signal of interest. Then, the parameters $\{C_0, \theta_0\}$ and the offsets can be estimated from the training signal. Subsequently, it be-

comes possible to compensate signals of interest to obtain the desired signals. This approach assumes that the parameters $\{\hat{C}_0, \hat{\theta}_0\}$ of the PLL sideband distortions do not change over the duration of signal acquisition.

However, if the parameters $\{\hat{C}_0, \hat{\theta}_0\}$ change during signal acquisition, then it is desirable to have a mechanism to measure $\{\hat{C}_0, \hat{\theta}_0\}$ either continuously or intermittently during the acquisition. This is the approach we shall adopt and it will be based on extending the technique proposed in [41]. Figure 2.3 shows the proposed design and is motivated as follows.

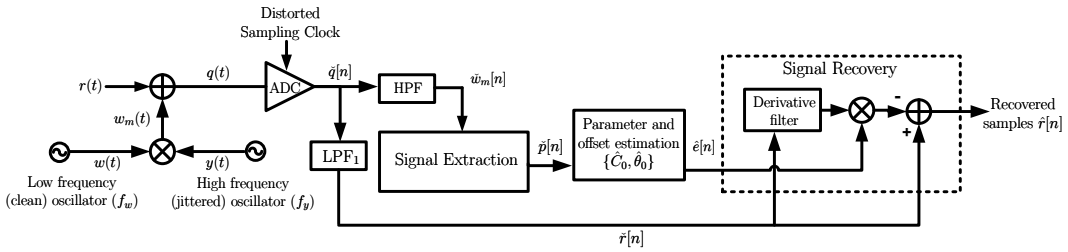


Figure 2.3: Proposed architecture for reducing the effect of PLL sidebands on A/D converters.

Two tone signals are used; one at low frequency and another at high frequency. A low-frequency tone, $w(t)$, is multiplied by a high-frequency tone, $y(t)$, to obtain a modulated signal, $w_m(t)$. It is possible that the signal $y(t)$ has some jitter. The signal $w_m(t)$ is then injected into the ADC along with the desired input signal, $r(t)$. We assume that $r(t)$ is in a lower frequency band and does not overlap with $w_m(t)$ in the frequency domain; the purpose of the high-frequency tone $y(t)$ is to modulate $w(t)$ to higher bands where this overlap is minimal. The jittered sampled signal $\breve{q}[n]$ contain contributions from the desired signal $r(t)$ and the control signal $s(t)$. By examining the effect of the ADC conversion on $w_m(t)$, we can infer the distortion caused on $r(t)$ and use this information to recover the samples $r[n]$. We now explain the operation of the proposed structure in

greater detail. We will split the structure into four main components. They are called training signal injection, training signal extraction, parameter and offset estimation, and signal recovery. We will discuss the training signal injection, training signal extraction and signal recovery in this section. The parameter and offset estimation is covered in the next section.

2.4.1 Training signal injection

Let us write

$$y(t) = \cos(2\pi f_y(t + \tau(t)) + \theta_y) \tag{2.25}$$

where $\tau(t)$ models the jitter in $y(t)$ and is assumed to be small. We assume that the jitter arises from a second-order PLL phase noise model as described in Appendix 2.B. Multiplying $w(t)$ by $y(t)$ yields a modulated signal $w_m(t)$. The signal $w_m(t)$ is non-uniformly sampled by the ADC using the sampling instants $nT_s + e[n]$, i.e,

$$\check{w}_m[n] \triangleq \check{w}[n]\check{y}[n] \tag{2.26}$$

where $\check{w}[n]$ is defined in (2.17) and $\check{y}[n]$ is

$$\begin{aligned}
\check{y}[n] &\triangleq \cos(2\pi f_y(nT_s + e[n] + \tau(nT_s + e[n]) + \theta_y) && \text{(2.27a)}\\
&\approx \cos(2\pi f_y(nT_s + e[n] + \tau[n]) + \theta_y) \\
&\approx \cos(2\pi f_y nT_s + \theta_y) - 2\pi f_y(e[n] + \tau[n]) \sin(2\pi f_y nT_s + \theta_y) && \text{(2.27b)}\\
\tau[n] &\triangleq \tau(t)|_{t=nT_s} && \text{(2.27c)}
\end{aligned}$$

The spectrum of the jitter (see Appendix 2.B) is a skirt where the spectrum decays with frequency. Thus, we assume that the jitter is relatively slowly-varying. Therefore, when the slow-varying jitter is sampled with offset $e[n]$, the term

$\tau(nT_s + e[n])$ in (2.27a) is approximated by $\tau[n]$. (2.27b) is derived using a first-order Taylor series approximation. Observe that $\check{w}_m[n]$ contains the distortion from both the deterministic offset $e[n]$ and the jitter from $y(t)$.

2.4.2 Training signal extraction

We are interested in estimating the offset $e[n]$ that is in $\check{w}[n]$. Thus, we would like to remove $\check{y}[n]$, which contains both $e[n]$ and $\tau[n]$. This can be done by creating an in-phase cosine sequence digitally, and multiplying the sequence with $\check{w}_m[n]$ to yield

$$\begin{aligned}\check{w}_m[n]\cos(2\pi f_y nT_s + \theta_y) &= \check{w}[n]\check{y}[n]\cos(2\pi f_y nT_s + \theta_y) \\ &\approx \frac{1}{2}\left(\check{w}[n] + \check{w}[n]\cos\left(4\pi f_y nT_s + 2\theta_y\right) - \right. \\ &\quad \left. 2\pi f_y(\tau[n] + e[n])\check{w}[n]\sin(4\pi f_y nT_s + 2\theta_y)\right)\end{aligned} \tag{2.28}$$

The above equation shows $\check{w}[n]$ multiplied by a DC term, a noiseless cosine sequence and a noisy sine sequence. The noisy sine sequence contains $\tau[n]$ and $e[n]$. In the frequency domain, the spectrum of the noisy sine sequence is the spectrum of $\tau[n]$ and $e[n]$ centered around $\pm(2f_y + f_w)$ and $\pm(2f_y - f_w)$, and repeated at multiples of f_s. The dominant frequency content is concentrated around $\pm(2f_y + f_w)$ and $\pm(2f_y - f_w)$ and its replica are spaced at multiples of f_s. However, there is some noisy frequency content from $\tau[n]$ in the low frequency region where $\check{w}[n]$ occurs. An illustration of the spectrum in (2.28) is shown in Fig. 2.4. The parameters used are $f_w = 40$ MHz, $f_y = 420$ MHz.

If the dominant noisy frequency content is far from $\check{w}[n]$, then its effect on $\check{w}[n]$ is reduced. Therefore, a low-pass filter is used to retain the sequence $\check{w}[n]$, and remove the effects of the sine and cosine sequences in (2.28). Under the simulation parameters used in the paper, we assume that the noise from $\tau[n]$ is not significant and the output after the low-pass filter contains only $\check{w}[n]$. This

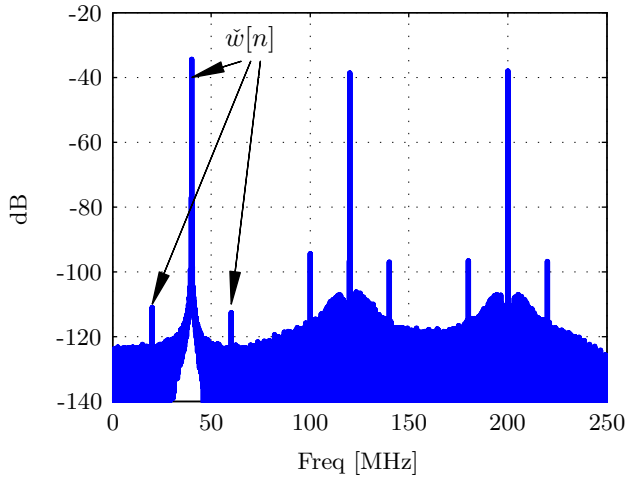


Figure 2.4: PSD of $\check{w}_m[n]\cos(2\pi f_y n T_s + \theta_y)$ in (2.28). Note that $\check{w}[n]$ is in the lower frequency range.

is verified in the next two sections. First, in section 2.5, we analyze the mean-square error in estimating $e[n]$ from $\check{w}[n]$. Next, in section 2.6, we simulate the estimation and compensation process and verify that the noise from $\tau[n]$ does not affect performance. If it is required, we can use the phase-noise model of Appendix 2.B to characterize the effect of $\tau[n]$ on $\check{w}[n]$. For completeness, we examine the effect of $\tau[n]$ in Appendix 2.C. Thus, we assume in the paper that the output after the low-pass filter is $\check{p}[n]$:

$$\check{p}[n] = \frac{A_w}{2}\cos(2\pi f_w(nT_s + e[n]) + \phi_w) \tag{2.29}$$

The training signal extraction block diagram is shown in Figure 2.5. This stage is represented by the "Parameter and offset estimation" block in Figure 2.3. The signal $\check{p}[n]$ will be used in the next section to estimate $\{C_0, \theta_0\}$ and the sampling offset $e[n]$. The phase recovery in Figure 2.5 estimates θ_y from $\check{w}_m[n]$ and generates $\cos(2\pi f_y n T_s + \hat{\theta}_y)$. The phase θ_y can be estimated by approximating

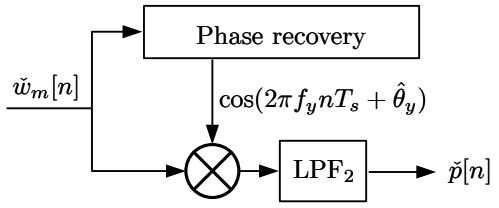


Figure 2.5: Diagram of signal extraction block.

$\check{w}_m[n]$ as the summation of two tones:

$$\begin{aligned}
\check{w}_m[n] &= \check{w}[n]\check{y}[n] \\
&= A_w \cos(2\pi f_w(nT_s + e[n]) + \phi_w) \cos(2\pi f_y(nT_s + e[n] + \tau[n]) + \theta_y) \\
&\approx A_w \cos(2\pi f_w nT_s + \phi_w) \cos(2\pi f_y nT_s + \theta_y) \\
&= \frac{A_w}{2} [\cos(2\pi(f_w - f_y)nT_s + \phi_w - \theta_y) \\
&\qquad + \cos(2\pi(f_w + f_y)nT_s + \phi_w + \theta_y)]
\end{aligned} \tag{2.30}$$

Thus, $\hat{\theta}_y$ can be found after estimating the phases in the two tones at $(f_w - f_y)$ and $(f_w + f_y)$. One way to estimate the phases of the tones is shown in Appendix 2.D.

2.4.3 Signal recovery

Let us assume for now that $e[n]$ has been estimated. We can then recover the desired signal, $r[n]$, from $\check{r}[n]$ as follows:

$$\begin{aligned}
r[n] &\triangleq r(nT_s) \\
&= r\left(nT_s + e[n] - e[n]\right) \\
&\approx r(nT_s + e[n]) - e[n]\dot{r}(nT_s + e[n]) \\
&= \check{r}[n] - e[n]\dot{r}(nT_s + e[n])
\end{aligned} \tag{2.31}$$

where $\check{r}[n]$ are the distorted samples, $e[n]$ are the sampling offsets (2.13) and $\dot{r}(nT_s + e[n])$ are the derivatives of $r(t)$ at $t = nT_s + e[n]$. These derivatives can be approximated using a discrete filter applied to $\check{r}[n]$. Equation (2.31) shows that $r(n)$ can be recovered by subtracting from $\check{r}[n]$ the product of the offsets, $e[n]$, and the output of the differentiator filter. A block diagram showing the signal recovery process using the estimated offsets, $\hat{e}[n]$, is illustrated in Figure 2.6. An

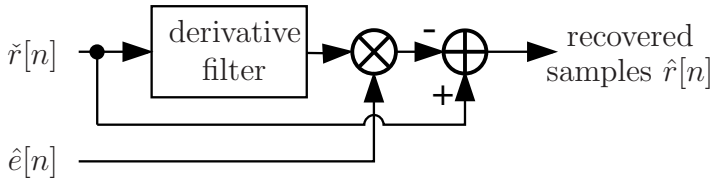


Figure 2.6: Block diagram of the signal recovery.

example of a differentiator filter is a finite impulse response (FIR) filter with the weights shown in Table 2.1 below. In summary, the steps for the proposed algorithm are summarized in Algorithm 2.1.

Table 2.1: Filter coefficients of a FIR differentiator filter.

Tap	1	2	3	4	5	6
Weights	0.004	-0.028	0.106	-0.307	0.888	0.000
Tap	7	8	9	10	11	-
Weights	-0.888	0.307	-0.106	0.028	-0.004	-

Algorithm 2.1 Summary of sideband suppression algorithm

Require: The sampled data is filtered by the LPF_1 and HPF in Fig. 2.3 to obtain $\check{r}[n]$ and $\check{w}_m[n]$, respectively.

% Training signal extraction (See Fig. 2.5)

$\check{p}[n] = \mathrm{LPF}_2\left\{\check{w}_m[n]\cos(2\pi f_y n T_s + \hat{\theta}_y)\right\}$

% $\hat{C}_0$ and $\hat{\theta}_0$ are estimated using Algorithm 2.2 or 2.3 in Section 2.5.

$[\hat{C}_0, \hat{\theta}_0] = \textsc{ParameterEstimate}(\check{p}[0, \cdots, L-1])$

repeat

% Offeset estimation (2.13)

$\hat{e}[n] = -\dfrac{\hat{C}_0}{2\pi f_s}\sin(2\pi n f_{\mathrm{ref}} T_s + \hat{\theta}_0)$

% Signal recovery (see Fig. 2.6)

$\hat{r}[n] = \check{r}[n] - e[n]\dot{r}(nT_s + e[n])$

until end of data

2.5 Parameter and offset estimation

2.5.1 Estimation algorithm

We estimate $\{C_0, \theta_0\}$ from the sidebands appearing in $\check{p}[n]$ given by (2.29). We formulate a least-squares estimation problem. We first express $\check{p}[n]$ using (2.10) and (2.13) as

$$\begin{aligned}
\check{p}[n] &= \frac{A_w}{2}\cos(2\pi f_w(nT_s + e[n]) + \phi_w) \\
&\approx \frac{A_w}{2}[\cos(2\pi f_w n T_s + \phi_w) - 2\pi f_w e[n]\sin(2\pi f_w n T_s + \phi_w)] \\
&= \frac{A_w}{2}\cos(2\pi f_w n T_s + \phi_w) + \\
&\quad \frac{f_p A_w C_0}{4 f_s}[\cos(2\pi(f_w - f_{\mathrm{ref}})nT_s + \phi_w - \theta_0) \\
&\qquad - \cos(2\pi(f_w + f_{\mathrm{ref}})nT_s + \phi_w + \theta_0)]
\end{aligned} \tag{2.32}$$

The signal $\check{p}[n]$ depends on the parameters in the following vector (see (2.36) further ahead):

$$\boldsymbol{\lambda} \triangleq \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} \frac{A_w}{2}\cos(\phi_w) \\ \frac{A_w}{2}\sin(\phi_w) \\ \frac{f_w C_0}{2f_s}\cos(\theta_0) \\ \frac{f_w C_0}{2f_s}\sin(\theta_0) \end{bmatrix} \tag{2.33}$$

Here, we assume that A_w and ϕ_w are unknown parameters to be estimated along with $\{C_0, \theta_0\}$. However, their values may be known or controlled. Later, we will show how to simplify the proposed algorithm if their values are known. For now, let's assume that we need to estimate all the parameters in $\boldsymbol{\lambda}$. Observe that if $\boldsymbol{\lambda}$ is estimated, then the parameters C_0 and θ_0 can be recovered as:

$$\begin{aligned} \hat{C}_0 &= \frac{2f_s}{f_w}\sqrt{\lambda_3^2 + \lambda_4^2} \\ \hat{\theta}_0 &= \tan^{-1}\left(\frac{\lambda_4}{\lambda_3}\right) \end{aligned} \tag{2.34}$$

For ease of notation, the following sequences are defined:

$$\begin{aligned} c_1[n] &= \cos\left(2\pi f_w n T_s\right), & s_1[n] &= \sin\left(2\pi f_w n T_s\right) \\ c_2[n] &= \cos\left(2\pi\left(f_w - f_{\text{ref}}\right) n T_s\right), & s_2[n] &= \sin\left(2\pi\left(f_w - f_{\text{ref}}\right) n T_s\right) \\ c_3[n] &= \cos\left(2\pi\left(f_w + f_{\text{ref}}\right) n T_s\right), & s_3[n] &= \sin\left(2\pi\left(f_w + f_{\text{ref}}\right) n T_s\right) \end{aligned} \tag{2.35}$$

Using trigonometric identities and some algebra, we can write (2.32) as

$$\begin{aligned} \check{p}(n, \boldsymbol{\lambda}) = {} & g_1[n]\lambda_1 + g_2[n]\lambda_2 + g_3[n]\lambda_1\lambda_3 \\ & + g_4[n]\lambda_2\lambda_3 + g_5[n]\lambda_2\lambda_4 + g_6[n]\lambda_1\lambda_4 \end{aligned} \tag{2.36}$$

where

$$\begin{aligned} g_1[n] &= c_1[n], & g_2[n] &= -s_1[n] \\ g_3[n] &= c_2[n] - c_3[n], & g_4[n] &= -s_2[n] + s_3[n] \\ g_5[n] &= c_2[n] + c_3[n], & g_6[n] &= s_2[n] + s_3[n] \end{aligned} \tag{2.37}$$

Suppose we collect a segment of data of length L perturbed by noise $v[n]$, say, $y_p[n] = \check{p}[n] + v[n]$. Then we can pose the problem:

$$\min_{\boldsymbol{\lambda}} \sum_{k=0}^{L-1} [y_p[k] - \check{p}(k, \boldsymbol{\lambda})]^2 \tag{2.38}$$

It is noted that $\check{p}(n, \boldsymbol{\lambda})$ is not a linear function over $\boldsymbol{\lambda}$; it is linear if either $\{\lambda_1, \lambda_2\}$ or $\{\lambda_3, \lambda_4\}$ are fixed. Therefore, a sub-optimal approach is used by iteratively fixing a pair of variables while solving for the other pair. When $\{\lambda_3, \lambda_4\}$ are fixed, we solve for $\{\lambda_1, \lambda_2\}$ using

$$\min_{\boldsymbol{p_\alpha}} \|\boldsymbol{y_\alpha} - G_\alpha \boldsymbol{p_\alpha}\|^2 \tag{2.39}$$

where

$$G_\alpha = \begin{bmatrix} \boldsymbol{g_1} + \lambda_3 \boldsymbol{g_3} + \lambda_4 \boldsymbol{g_6} & \boldsymbol{g_2} + \lambda_3 \boldsymbol{g_4} + \lambda_4 \boldsymbol{g_5} \end{bmatrix} \tag{2.40a}$$

$$\boldsymbol{g_i} = \begin{bmatrix} g_i[0] \\ \vdots \\ g_i[L-1] \end{bmatrix} \tag{2.40b}$$

$$\boldsymbol{y_\alpha} = \begin{bmatrix} y_p[0] \\ \vdots \\ y_p[L-1] \end{bmatrix} \tag{2.40c}$$

$$\boldsymbol{p_\alpha} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \tag{2.40d}$$

From [42], the closed-form solution is

$$\boldsymbol{p_\alpha} = (G_\alpha^T G_\alpha)^{-1} G_\alpha^T \boldsymbol{y_\alpha} \tag{2.41}$$

Similarly, when $\{\lambda_1, \lambda_2\}$ is fixed, we could solve for $\{\lambda_3, \lambda_4\}$ using

$$\min_{\boldsymbol{p_\beta}} \|\boldsymbol{y_\beta} - G_\beta \boldsymbol{p_\beta}\|^2 \tag{2.42}$$

where

$$G_\beta = \begin{bmatrix} \lambda_1 \boldsymbol{g_3} + \lambda_2 \boldsymbol{g_4} & \lambda_2 \boldsymbol{g_5} + \lambda_1 \boldsymbol{g_6} \end{bmatrix} \tag{2.43a}$$

$$\boldsymbol{p_\beta} = \begin{bmatrix} \lambda_3 \\ \lambda_4 \end{bmatrix} \tag{2.43b}$$

$$\boldsymbol{y_\beta} = \boldsymbol{y_\alpha} - \lambda_1 \boldsymbol{g_1} - \lambda_2 \boldsymbol{g_2} \tag{2.43c}$$

The closed-form solution is

$$\boldsymbol{p_\beta} = (G_\beta^T G_\beta)^{-1} G_\beta^T \boldsymbol{y_\beta} \tag{2.44}$$

The closed-form solutions in (2.41) and (2.44) involve a matrix-matrix multiplication and an inverse matrix operation. The matrices $G_\alpha^T G_\alpha$ and $G_\beta^T G_\beta$ are 2×2 matrices. Thus, their inverses can be computed easily. The computation of the matrix-matrix multiplication can be reduced by exploiting the structure in the matrices. For example, for the matrix $G_\alpha^T G_\alpha$, its elements are linear combinations of $\boldsymbol{g_k^T g_l}, \{k, l\} \in \{1, 2, ..., 6\}$ and $\boldsymbol{g_k^T g_l}$ can be pre-computed and reused in the iterative algorithm. Alternatively, if we assume that the length of the data (L) is large, $\boldsymbol{g_k^T g_k}$ can be approximated as:

$$\begin{aligned} \boldsymbol{g_1^T g_1} \approx \tfrac{L}{2}, \quad \boldsymbol{g_3^T g_3} \approx L, \quad \boldsymbol{g_5^T g_5} \approx L \\ \boldsymbol{g_2^T g_2} \approx \tfrac{L}{2}, \quad \boldsymbol{g_4^T g_4} \approx L, \quad \boldsymbol{g_6^T g_6} \approx L \end{aligned} \tag{2.45}$$

and $\boldsymbol{g_k^T g_l} \approx 0,\ k \neq l$. Thus, $G_\alpha^T G_\alpha$ and $G_\beta^T G_\beta$ can be approximated as

$$G_\alpha^T G_\alpha \approx \frac{L}{2}(1 + 2\lambda_3^2 + 2\lambda_4^2) I \tag{2.46a}$$

$$G_\beta^T G_\beta \approx L(\lambda_1^2 + \lambda_2^2) I \tag{2.46b}$$

From simulations, it was observed that the mean-square error (MSE) for A_w and θ_0 deviates from the Cramer Rao Bound (CRB) for small sample length when (2.46) is used. It was found that the estimated values are biased and it is caused

by a poor approximation of the matrix $G_\alpha^T G_\alpha$. A better approximation is found to be:

$$G_\alpha^T G_\alpha \approx \begin{bmatrix} \frac{L}{2}(1+2\lambda_3^2+2\lambda_4^2) & \boldsymbol{g_1^T g_2} \\ \boldsymbol{g_1^T g_2} & \frac{L}{2}(1+2\lambda_3^2+2\lambda_4^2) \end{bmatrix} \tag{2.47}$$

where $\boldsymbol{g_1^T g_2}$ can be pre-computed and reused. In summary, the proposed algorithm to estimate the sampling offset parameters C_0 and θ_0 is shown in Algorithm 2.2, where it solves for all 4 parameters in $\boldsymbol{\lambda}$. If A_p and ϕ_p are known, then the problem is simplified into solving the minimization problem (2.42) only. The simplified algorithm is shown in Algorithm 2.3.

2.5.2 Cramer-Rao bound

The previous section estimates C_0 and θ_0 using (2.34). These two parameters are used to estimate the sampling offsets via (2.13) and (2.7). We will derive the Cramer-Rao Bound (CRB) [43] for the parameters and the sampling offsets in white Gaussian noise (WGN). Let

$$\begin{aligned} \boldsymbol{\kappa} &= [\kappa_1 \ \kappa_2 \ \kappa_3 \ \kappa_4]^T \\ &\triangleq \left[\frac{A_w}{2} \ \frac{f_w C_0}{2f_s} \ \phi_w \ \theta_0\right]^T \end{aligned} \tag{2.48}$$

The vector $\boldsymbol{\kappa}$ is now used instead of $\boldsymbol{\lambda}$ from the previous section since the Fisher Information Matrix (FIM) (see (2.52) further ahead) involving $\boldsymbol{\kappa}$ can be easily inverted to yield the CRB for the parameters, κ_i, and the sampling offsets. We rewrite (2.32) in terms of $\boldsymbol{\kappa}$ as follows:

$$\begin{aligned} \check{p}(n,\boldsymbol{\kappa}) = \kappa_1 \cos(2\pi f_w n T_s + \kappa_3) + \\ \kappa_1\kappa_2 \left[\cos(2\pi(f_w - f_{\text{ref}})nT_s + \kappa_3 - \kappa_4)\right. \\ \left. - \cos(2\pi(f_w + f_{\text{ref}})nT_s + \kappa_3 + \kappa_4)\right] \end{aligned} \tag{2.49}$$

Algorithm 2.2 Parameter estimation algorithm

Require: Let the number of iterations be N and $\boldsymbol{\lambda} = [\ 0 \quad 0 \quad 0 \quad 0\ ]^T$. Precompute $\boldsymbol{g_i}$ from (2.40b) and (2.37).

procedure PARAMETERESTIMATE($y_p[0, \cdots, L-1]$)

for $k = 1, \cdots, N$ **do**

$$\begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \end{bmatrix}^T = \boldsymbol{\lambda}$$

% Estimate $\{\lambda_1, \lambda_2\}$ *using (2.41)*

$$\boldsymbol{y_\alpha} = \begin{bmatrix} y_p[0] & \cdots & y_p[L-1] \end{bmatrix}^T$$

$$G_\alpha = \begin{bmatrix} \boldsymbol{g_1} + \lambda_3 \boldsymbol{g_3} + \lambda_4 \boldsymbol{g_6} & \boldsymbol{g_2} + \lambda_3 \boldsymbol{g_4} + \lambda_4 \boldsymbol{g_5} \end{bmatrix}$$

$$G_\alpha^T G_\alpha = \begin{bmatrix} \frac{L}{2}(1 + 2\lambda_3^2 + 2\lambda_4^2) & \boldsymbol{g_1^T g_2} \\ \boldsymbol{g_1^T g_2} & \frac{L}{2}(1 + 2\lambda_3^2 + 2\lambda_4^2) \end{bmatrix}$$

$$\boldsymbol{p_\alpha} = (G_\alpha^T G_\alpha)^{-1} G_\alpha^T \boldsymbol{y_\alpha}$$

$$\begin{bmatrix} \lambda_1 & \lambda_2 \end{bmatrix}^T = \boldsymbol{p_\alpha}$$

% Estimate $\{\lambda_3, \lambda_4\}$ *using (2.44)*

$$G_\beta = \begin{bmatrix} \lambda_1 \boldsymbol{g_3} + \lambda_2 \boldsymbol{g_4} & \lambda_2 \boldsymbol{g_5} + \lambda_1 \boldsymbol{g_6} \end{bmatrix}$$

$$G_\beta^T G_\beta = L(\lambda_1^2 + \lambda_2^2) I$$

$$\boldsymbol{y_\beta} = \boldsymbol{y_\alpha} - \lambda_1 \boldsymbol{g_1} - \lambda_2 \boldsymbol{g_2}$$

$$\boldsymbol{p_\beta} = (G_\beta^T G_\beta)^{-1} G_\beta^T \boldsymbol{y_\beta}$$

$$\begin{bmatrix} \lambda_3 & \lambda_4 \end{bmatrix}^T = \boldsymbol{p_\beta}$$

end for

% $\hat{C}_0$ and $\hat{\theta}_0$ are estimated from $\{\lambda_3, \lambda_4\}$ using (2.34)

$$\hat{C}_0 = \frac{2 f_s}{f_w} \sqrt{\lambda_3^2 + \lambda_4^2}$$

$$\hat{\theta}_0 = \tan^{-1}\left(\frac{\lambda_4}{\lambda_3}\right)$$

return $\hat{C}_0, \hat{\theta}_0$

end procedure

Algorithm 2.3 Simplified parameter estimation algorithm

Require: Let the number of iterations be N and $\boldsymbol{\lambda} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$. Precompute $\boldsymbol{g_i}$ from (2.40b) and (2.37). Let $\lambda_1 = A_p \cos(\phi_p)$ and $\lambda_2 = A_p \sin(\phi_p)$.

procedure ParameterEstimate($y_p[0, \cdots, L-1]$)

$\begin{bmatrix} - & - & \lambda_3 & \lambda_4 \end{bmatrix}^T = \boldsymbol{\lambda}$

*% **Estimate** $\{\lambda_3, \lambda_4\}$ **using (2.44)***

$G_\beta = \begin{bmatrix} \lambda_1 \boldsymbol{g_3} + \lambda_2 \boldsymbol{g_4} & \lambda_2 \boldsymbol{g_5} + \lambda_1 \boldsymbol{g_6} \end{bmatrix}$

$G_\beta^T G_\beta = L(\lambda_1^2 + \lambda_2^2) I$

$\boldsymbol{y_\alpha} = \begin{bmatrix} y_p[0] & \cdots & y_p[L-1] \end{bmatrix}^T$

$\boldsymbol{y_\beta} = \boldsymbol{y_\alpha} - \lambda_1 \boldsymbol{g_1} - \lambda_2 \boldsymbol{g_2}$

$\boldsymbol{p_\beta} = (G_\beta^T G_\beta)^{-1} G_\beta^T \boldsymbol{y_\beta}$

$\begin{bmatrix} \lambda_3 & \lambda_4 \end{bmatrix}^T = \boldsymbol{p_\beta}$

*% $\hat{C}_0$ **and** $\hat{\theta}_0$ **are estimated from** $\{\lambda_3, \lambda_4\}$ **using (2.34)***

$\hat{C}_0 = \dfrac{2 f_s}{f_w} \sqrt{\lambda_3^2 + \lambda_4^2}$

$\hat{\theta}_0 = \tan^{-1}\left(\dfrac{\lambda_4}{\lambda_3}\right)$

return $\hat{C}_0, \hat{\theta}_0$

end procedure

The FIM of the sampled signal $\check{p}(n, \boldsymbol{\kappa})$ with length L in WGN with variance σ^2 is

$$[I(\boldsymbol{\kappa})]_{ij} = \frac{1}{\sigma^2} \sum_{k=0}^{L-1} \frac{\partial \check{p}(k, \boldsymbol{\kappa})}{\partial \kappa_i} \frac{\partial \check{p}(k, \boldsymbol{\kappa})}{\partial \kappa_j} \tag{2.50}$$

The partial derivatives are given by

$$\begin{aligned}
\frac{\partial \check{p}(n, \boldsymbol{\kappa})}{\partial \kappa_1} &= \cos\left(2\pi f_w n T_s + \kappa_3\right) \\
&\quad + \kappa_2 \cos\left(2\pi \left(f_w - f_{\text{ref}}\right) n T_s + \kappa_3 - \kappa_4\right) \\
&\quad - \kappa_2 \cos\left(2\pi \left(f_w + f_{\text{ref}}\right) n T_s + \kappa_3 + \kappa_4\right) && \text{(2.51a)} \\
\frac{\partial \check{p}(n, \boldsymbol{\kappa})}{\partial \kappa_2} &= \kappa_1 \cos\left(2\pi \left(f_w - f_{\text{ref}}\right) n T_s + \kappa_3 - \kappa_4\right) \\
&\quad - \kappa_1 \cos\left(2\pi \left(f_w + f_{\text{ref}}\right) n T_s + \kappa_3 + \kappa_4\right) && \text{(2.51b)} \\
\frac{\partial \check{p}(n, \boldsymbol{\kappa})}{\partial \kappa_3} &= -\kappa_1 \sin\left(2\pi f_w n T_s + \kappa_3\right) \\
&\quad - \kappa_1 \kappa_2 \sin\left(2\pi \left(f_w - f_{\text{ref}}\right) n T_s + \kappa_3 - \kappa_4\right) \\
&\quad + \kappa_1 \kappa_2 \sin\left(2\pi \left(f_w + f_{\text{ref}}\right) n T_s + \kappa_3 + \kappa_4\right) && \text{(2.51c)} \\
\frac{\partial \check{p}(n, \boldsymbol{\kappa})}{\partial \kappa_4} &= \kappa_1 \kappa_2 \sin\left(2\pi \left(f_w - f_{\text{ref}}\right) n T_s + \kappa_3 - \kappa_4\right) \\
&\quad + \kappa_1 \kappa_2 \sin\left(2\pi \left(f_w + f_{\text{ref}}\right) n T_s + \kappa_3 + \kappa_4\right) && \text{(2.51d)}
\end{aligned}$$

Assuming L is large, the FIM matrix $I(\boldsymbol{\kappa})$ can be approximated to:

$$I(\boldsymbol{\kappa}) \approx \frac{1}{\sigma^2} \begin{bmatrix} \frac{L}{2}\left[1 + 2\kappa_2^2\right] & L\kappa_1\kappa_2 & 0 & 0 \\ L\kappa_1\kappa_2 & L\kappa_1^2 & 0 & 0 \\ 0 & 0 & \frac{L}{2}\left[\kappa_1^2 + 2\kappa_1^2\kappa_2^2\right] & 0 \\ 0 & 0 & 0 & L\left(\kappa_1\kappa_2\right)^2 \end{bmatrix} \tag{2.52}$$

and the inverse FIM, $I^{-1}(\boldsymbol{\kappa})$, becomes

$$I^{-1}(\boldsymbol{\kappa}) \approx \sigma^2 \begin{bmatrix} \frac{2}{L} & -\frac{2\kappa_2}{L\kappa_1} & 0 & 0 \\ -\frac{2\kappa_2}{L\kappa_1} & \frac{1}{L\kappa_1^2}\left[1+2\kappa_2^2\right] & 0 & 0 \\ 0 & 0 & \frac{2}{L\kappa_1^2(1+2\kappa_2^2)} & 0 \\ 0 & 0 & 0 & \frac{1}{L(\kappa_1\kappa_2)^2} \end{bmatrix} \tag{2.53}$$

The CRB for each parameter, κ_i, is the diagonal value of the matrix, $\{I^{-1}(\boldsymbol{\kappa})\}_{i,i}$. Now we derive the CRB for the sampling offset estimates. Introduce the sampling offset function (2.13) that we want to estimate using $\boldsymbol{\kappa}$ as:

$$\begin{aligned} g(\boldsymbol{\kappa}) &= -\epsilon_s[n] \\ &= -\frac{1}{\pi f_w}\left(\frac{f_w C_0}{2f_s}\right)\sin(2\pi f_{\text{ref}}\, nT_s + \theta_0) \\ &= -\frac{\kappa_2}{\pi f_w}\sin(2\pi f_{\text{ref}}\, nT_s + \kappa_4) \end{aligned} \tag{2.54}$$

Then

$$\frac{\partial g(\boldsymbol{\kappa})}{\partial \boldsymbol{\kappa}} = \begin{bmatrix} 0 & \frac{\partial g(\boldsymbol{\kappa})}{\partial \kappa_2} & 0 & \frac{\partial g(\boldsymbol{\kappa})}{\partial \kappa_4} \end{bmatrix}^T \tag{2.55}$$

where

$$\frac{\partial g(\boldsymbol{\kappa})}{\partial \kappa_2} = -\frac{1}{\pi f_w}\sin(2\pi f_{\text{ref}}\, nT_s + \kappa_4) \tag{2.56a}$$

$$\frac{\partial g(\boldsymbol{\kappa})}{\partial \kappa_4} = -\frac{\kappa_2}{\pi f_w}\cos(2\pi f_{\text{ref}}\, nT_s + \kappa_4) \tag{2.56b}$$

The Cramer Rao Bound (CRB) for the sampling offsets $e(n)$ is then

$$\begin{aligned} C_e &= \frac{\partial g(\boldsymbol{\kappa})^T}{\partial \boldsymbol{\kappa}} I^{-1}(\boldsymbol{\kappa}) \frac{\partial g(\boldsymbol{\kappa})}{\partial \boldsymbol{\kappa}} \\ &= \frac{1}{L}\left(\frac{\sigma}{\pi f_w \kappa_1}\right)^2 \left(1 + 2\kappa_2^2 \sin^2(2\pi f_{\text{ref}}\, nT_s + \kappa_4)\right) \\ &\approx \frac{1}{L}\left(\frac{\sigma}{\pi f_w \kappa_1}\right)^2 \\ &= \frac{1}{L}\left(\frac{2\sigma}{\pi f_w A_w}\right)^2 \end{aligned} \tag{2.57}$$

The derived CRB is used to assess the performance of the proposed compensation and estimation algorithm in the next section. The CRB of $e[n]$ is inversely proportional to L and the square of f_w and A_w and is proportional to σ^2. This reveals that $e[n]$ can be estimated more accurately when L, f_w or A_w are increased or when σ^2 is reduced.

2.5.3 Performance analysis

To verify the performance of the parameter estimation algorithm, the following simulation is done. Sampled data $\check{p}[n]$ are created using (2.32) by fixing the parameters to $C_0 = 6.32 \times 10^{-3}$, $A_w = 0.1$ V, $f_w = 40$ MHz, $f_{\text{ref}} = 20$ MHz. The chosen value of C_0 simulates a sideband of -78 dBc in the data. The phases are randomly chosen and WGN is added to the signal. Simulations are repeated using different noise powers. The standard deviations of the noise , σ, are $\{\frac{1}{\sqrt{2}}(1\times 10^{-3}), \frac{1}{\sqrt{2}}(1 \times 10^{-4}), \frac{1}{\sqrt{2}}(1 \times 10^{-5})\}$. The factor $\frac{1}{\sqrt{2}}$ is used to represent the noise power reduction due to the multiplication with the cosine sequence in the signal extraction block (Figure 2.5). This let us compare with the simulation results where we simulate the entire proposed architecture process (Figure 2.3) in section 2.6. The length of the data, L, is varied from 2^8 to 2^{20} and the results are obtained by averaging over 300 simulations.

The parameter estimation algorithm stated in section 2.5.1 is used to estimate C_0 and θ_0. Recall that in the estimation algorithm, the matrices $G_\alpha^T G_\alpha$ and $G_\beta^T G_\beta$ can be approximated as (2.47). In the simulations, the performance using no approximation and the approximated matrices are compared. The methods using no approximation and (2.47) are labeled as Mtd 1 and Mtd 2, respectively.

The sampling offset $e(n)$ can be estimated using $\{\hat{C}_0, \hat{\theta}_0\}$ with (2.13). The mean-square-error (MSE) of the sampling offset is calculated for performance

analysis. The CRB bound for the sampling offset estimates in (2.57) is used to benchmark the performance. The power of $e[n]$ can be calculated and is found to be 5.06×10^{-25} W. Thus, we normalize the MSE and CRB by dividing them by the power of $e[n]$. Recall that the parameter estimation algorithm stated in Section 2.5.1 has a user-defined number of iteration, N. In the simulations, N is fixed at 1.

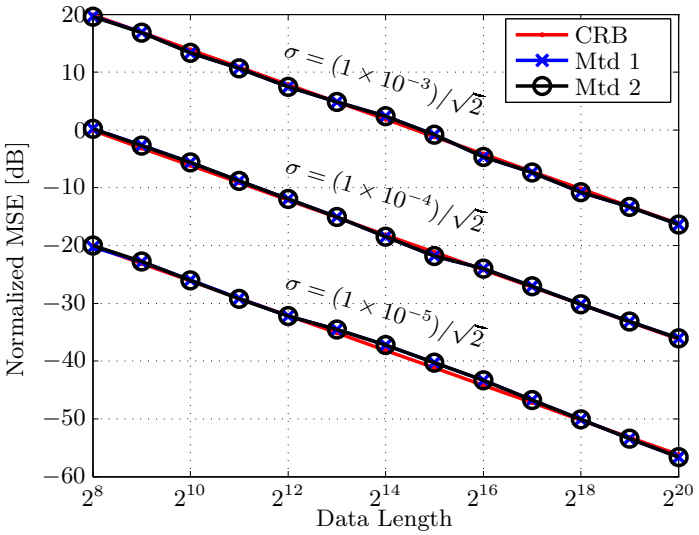


Figure 2.7: The figure shows the normalized MSE of the sampling offset estimates averaged over 300 simulations and the normalized CRB bound.

Figure 2.7 shows the normalized MSE of the estimated sampling offset using various data lengths L and in the presence of noise. The normalized CRB of the sampling offset estimation at the 3 different noise powers are the 3 lines in the plot. As the noise power increases, the CRB increases. From the plot, it can be seen that the estimation algorithm is performing close to the CRB.

2.6 Simulations

The proposed method is tested over a range of input frequencies. The frequency of the sampling clock, the low-frequency sinusoidal signal, and the high-frequency sinusoidal signal are set at $f_s = 1$ GHz, $f_w = 40$ MHz and $f_y = 420$ MHz, respectively. Recall that we assume the high frequency signal $y(t)$ is jittery. We assume that $y(t)$ from (2.25) is generated using a second-order PLL clock and has a phase noise. This phase noise can be translated to random jitter $\tau(t)$ expressed in $y(t)$. The phase noise model is described in Appendix 2.B and the parameter f_n in the model is set to 5 MHz. The standard deviation of the random jitter $\tau(t)$ due to phase noise is set to 1% of the sampling period. The frequency of the reference signal in the PLL feedback loop is $f_{\text{ref}} = 20$ MHz. The input signal used is a sinusoidal tone whose frequency is varied from 25 MHz to 250 MHz in steps of 25 MHz. To reduce the effects of the training signal on the dynamic range of the input data, the amplitude of the input signal $r(t)$ and the low-frequency training signal $w(t)$ are set to 0.8 V and 0.1 V, respectively. In the simulations, the length L is set to $\{2^{18}, 2^{19}, 2^{20}\}$. In all the simulations, WGN with standard deviation $\sigma_v = 1 \times 10^{-3}$ is introduced at the input of the ADC and all the results are averaged over 50 simulation runs. The lowpass filter LPF_1 in Figure 2.3 uses 64 taps with passband to 300 MHz and stopband from 350 MHz. The highpass filter HPF in Figure 2.3 uses 64 taps with stopband up to 300 MHz and passband from 350 MHz. The lowpass filter LPF_2 in the signal extraction block in Figure 2.5 uses 128 taps with a passband up to 70 MHz and stopband from 90 MHz.

2.6.1 Effect of bit resolution

In the first set of simulations, the ADC is assumed to have a 1V peak-to-peak input range and various bit resolutions $\{10, 12, 14, 16\}$ are simulated. Figure 2.8

and 2.9 show the ratio of the power of the spurious sidebands to the power of the input tone before and after using the proposed method at 10 bits and 16 bits ADC resolution, respectively. We denote this ratio as RPSI.

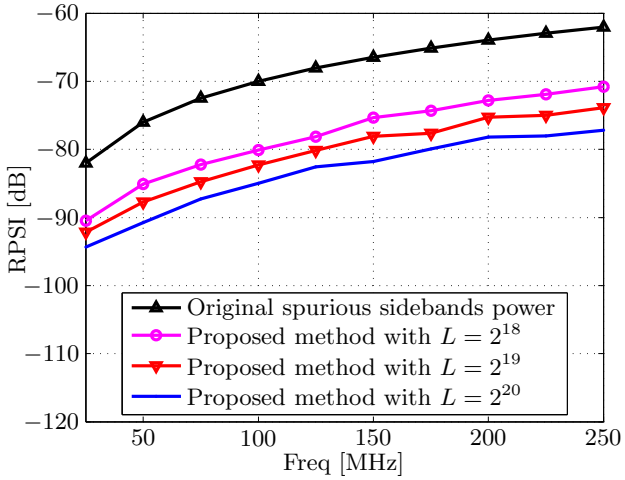


Figure 2.8: The plots show the RPSI, ratio of the power of the spurious sidebands (in the sampled data of a 10 bit ADC) to the power of the input tone, before and after compensation with $\sigma_v = 1 \times 10^{-3}$.

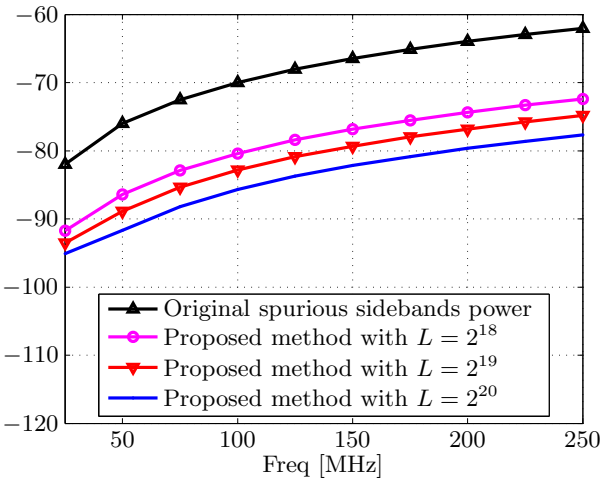


Figure 2.9: The plots show the RPSI in the sampled data of a 16 bit ADC, before and after compensation with $\sigma_v = 1 \times 10^{-3}$.

From the plots, in the original signal, the RPSI varies across frequency. This can be shown using (2.24) where f_w denotes the frequency of the input signal. When more samples are used to estimate the parameters, the accuracy of the sampling offset estimates improves. Hence, the spurious sideband suppression also improves.

To analyze the effect of bit resolution on performance, the next two plots are generated in the following manner. For each simulated ADC bit resolution, the average improvement in suppressing the spurious sidebands and the average MSE of the estimated sampling offset normalized to the power of the sampling offset are calculated and the results are shown in Figures 2.10 and 2.11, respectively. The

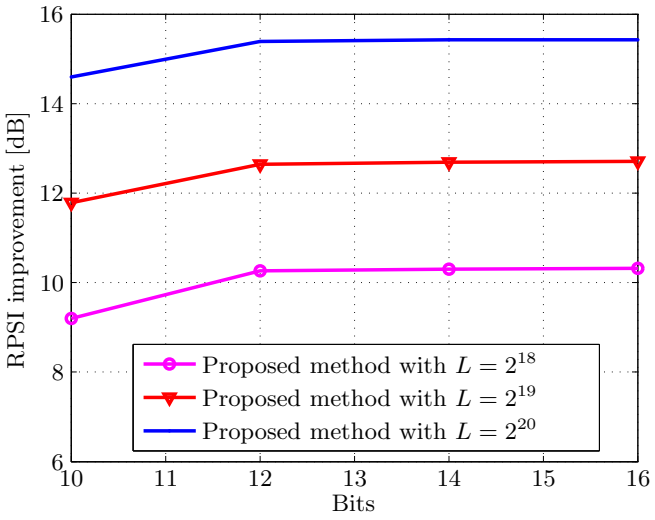


Figure 2.10: The plot shows the trend in suppressing the spurious sidebands using various bit resolution ADCs.

results show that the sideband suppression performance is directly related to the accuracy of the sampling offset estimation. The figures also show that when more data are used, the suppression of the spurious sideband and the sampling offset estimation improves. In the simulations, WGN with $\sigma_v = 1 \times 10^{-3}$ is fixed while

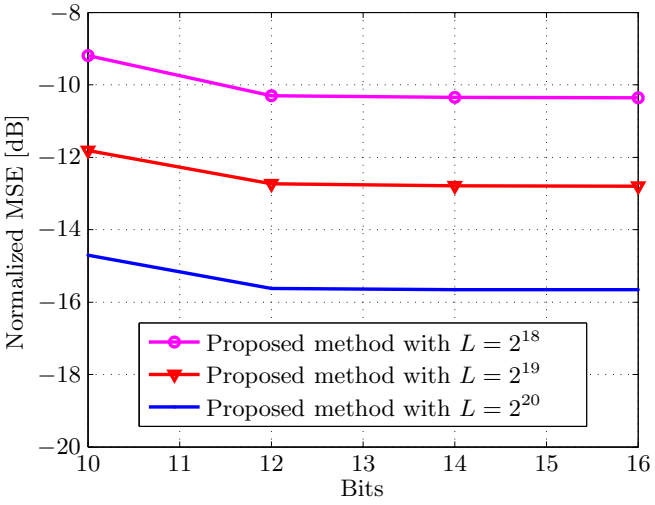


Figure 2.11: The plot shows the trend in MSE of the estimated sampling offset using various bit resolution ADCs.

the bit resolution is increased. Increasing bit resolution reduces quantization noise and there is an improvement from 10 bits to 12 bits. However, when WGN dominates over quantization noise, the performance is limited when more bits are used.

We can also compare the MSE performance with the CRB in the previous section. Recall that in section 2.4.2, we assume the effects of noise from the noisy high frequency tone is negligible and the signal extraction block output $\check{p}[n]$ only. Subsequently, the performance analysis in the previous section shows that the estimation error based on $\check{p}[n]$ is close to the CRB bound. Here, we simulate the entire process and evaluate the performance of the estimation error. When $\sigma_v = 1 \times 10^{-3}$ and $L = \{2^{18}, 2^{19}, 2^{20}\}$, we show in Figure 2.11 that the MSE of the sampling offsets is -10, -13 and -16 dB, respectively. These operating conditions correspond to $\sigma_{v'} = (1 \times 10^{-3})/\sqrt{2}$ and $L = \{2^{18}, 2^{19}, 2^{20}\}$ in Figure 2.7. From the two figures, the MSE values are similar. Therefore, we verifies that the assumption holds for these simulation parameters.

2.6.2 Effect of increasing the amplitude of the training signal

If we increase the amplitude of the training signal, at the expense of reducing the dynamic range of the input signal, it is possible to improve the parameter estimation accuracy and, hence, the spur suppression performance. The following results are generated by fixing $L = 2^{18}$ and using a 10-bit ADC. The amplitude of the training signal is increased from 0.05 V to 0.25 V, while the amplitude of the input signal is decreased from 0.85 V to 0.65 V in tandem. The sum of the two amplitudes remains at 0.9 V. The other simulation parameters remain the same. Figure 2.12 shows the RPSI as the amplitude of the training signal increases.

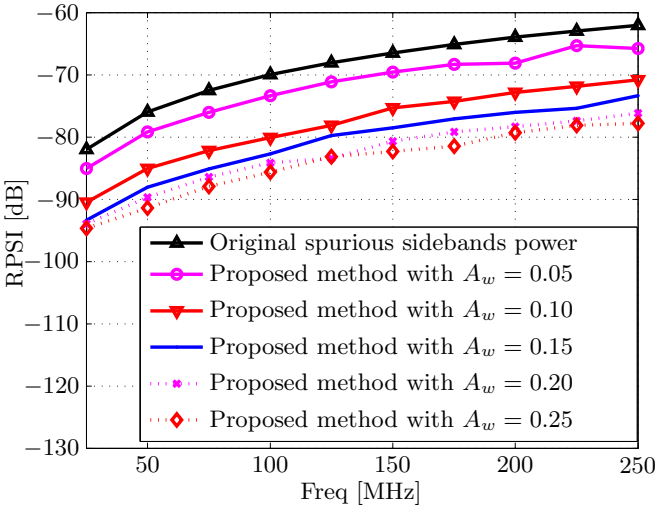


Figure 2.12: The same simulation parameters as in Fig. 2.8 are used, except that L is fixed at 2^{18} and the amplitudes of the training and input signals are varied.

To examine the improvement in spur suppression against the amplitude of the training signal, the next plot is shown. Figure 2.13 shows the average spur suppression improvement as the amplitude of the training signal is increased from 0.05 V to 0.25 V. We see that doubling the amplitude of the training signal from 0.1 V to 0.2 V reduces the spurious tones from 9 dB to 14 dB. If we compare this

against Figure 2.10, the same amount of improvement is obtained when the data length L is quadrupled (with a 10 bit ADC). This can also be inferred from the CRB bound (2.57) where it is inversely proportional to the data length L and the square of the amplitude of the training signal κ_1.

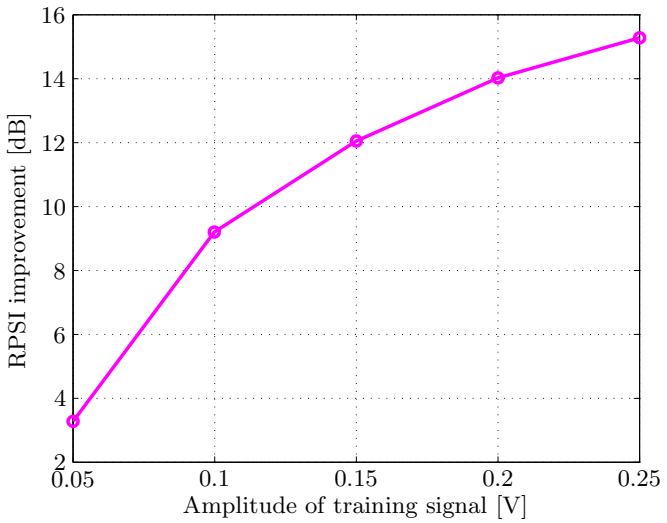


Figure 2.13: The plot shows the trend in suppressing the spurious sidebands as the amplitude of the training signal increases.

2.6.3 Effect of additional random jitter in ADC

In the proposed architecture (Figure 2.3), other sources of imperfection may be present. One source of imperfection is that the sampling clock itself may have random jitter in addition to the deterministic sideband distortion caused by the PLL and which we examined in this paper. Another possible error is that the low-frequency training signal $w(t)$ may have some amount of random jitter as well. As such, we included these imperfections into the simulations with a 10 bit ADC. In the next set of simulations, random time jitter based on a second-order PLL model is generated. The phase noise model is described in Appendix 2.B.

In the phase noise model, f_n is set to 5MHz and the standard deviation of the random jitter is normalized to the sampling interval. We denote the standard deviation of the jitter in the ADC and $w(t)$ by σ_{ADC} and σ_{LF}, respectively. We choose σ_{ADC} and σ_{LF}, expressed as percentages of the sampling period, from the set $\{0.1, 0.5, 1, 5, 10\}\%$. We first investigate the effects when random jitter is present in the ADC.

The left-sided plots in Figure 2.14 and 2.15 show the PSD where a tone at 125 MHz is perturbed by both the spurious sideband and random jitter (σ_{ADC} at 0.5% and 1%) in the ADC. Notice that as σ_{ADC} is increased, the random jitter

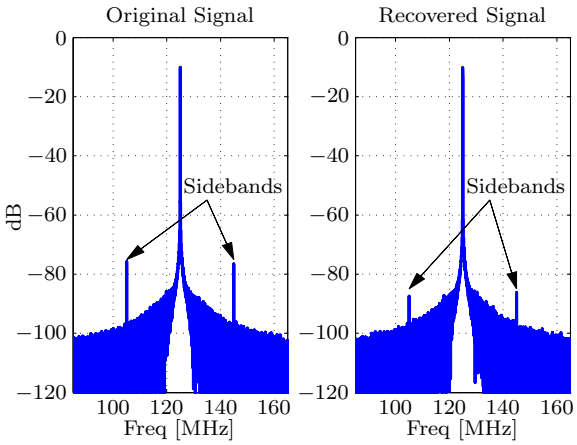


Figure 2.14: The plots show the PSD of a tone at 125 MHz before and after signal recovery, when the random jitter σ_{ADC} in the ADC is 0.5%.

creates a noisy spectrum around the tone. The right-sided plots in the figures show the PSD of the recovered signal.

Figure 2.16 shows the ratio of the power of the spurious sidebands in the sampled data to the power of the input signal in the presence of the random jitter in the ADC. The parameters used are the same as those used in Figure 2.8 except that random jitter (σ_{ADC} at 1% of the sampling period) in the ADC is added.

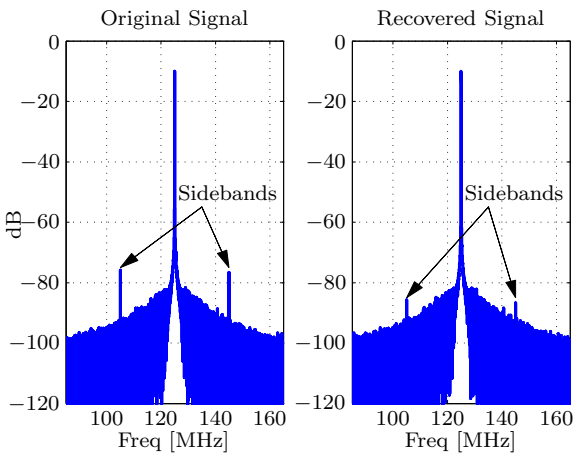


Figure 2.15: The plots show the PSD of a tone at 125 MHz before and after signal recovery, when the random jitter $\sigma_{\mathrm{A}DC}$ in the ADC is 1%.

Figure 2.17 is used to analyze the performance when the power of the random jitter in the ADC is changed. The plot shows the average improvement in the sideband suppression when $\sigma_{\mathrm{A}DC}$ is varied. From the plot, the performance degrades when $\sigma_{\mathrm{A}DC}$ increases above 1%.

2.6.4 Effect of noise in training signal

The simulations are repeated again except that the training signal has some random jitter due to imperfections. Figure 2.18 shows the average improvement in the sideband suppression when $\sigma_{\mathrm{L}F}$ is varied. From the plot, the algorithm performs well when $\sigma_{\mathrm{L}F}$ is below 1% and degrades when it is increased beyond 1%.

2.7 Conclusion

In this chapeter we modeled the effect of PLL imperfections on sampled data and proposed a technique to compensate for the presence of the spurious tones.

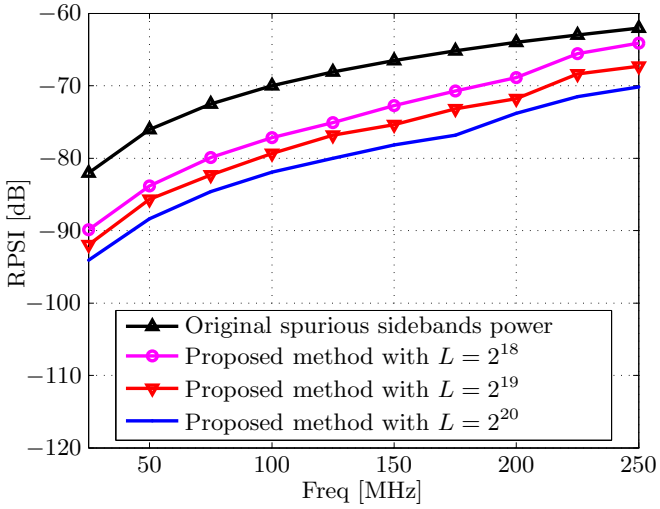


Figure 2.16: The plot is generated using the same parameters as in Figure 2.8 except that random jitter ($\sigma_{ADC} = 1\%$) in the 10-bit ADC is included.

A training signal is used to estimate the distortion and a filter implementation is used to remove the distortions from the sampled data by using discrete-time processing techniques. Simulation results verify that the proposed method is effective under Gaussian noise, quantization noise, random jitter in the ADC, and random jitter in the training signal itself.

2.A Derivation of relative error bounds

In this appendix, we derive the relative error bound between the actual $e[n]$ and its approximation in (2.13). Let $\hat{x}_n$ refer to the approximate value (i.e., $-\epsilon_s[n]$) for the true value x_n (i.e., $e[n] = -\epsilon_s(t_n)$). To find how close $\hat{x}_n$ is to x_n, we call upon Taylor's theorem [44].

Suppose h is a real function on $[a, b]$, m is a positive integer, the $(m-1)$-th derivative $h^{(m-1)}(t)$ is continuous on $[a, b]$ and $h^{(m)}(t)$ exists for every $t \in (a, b)$.

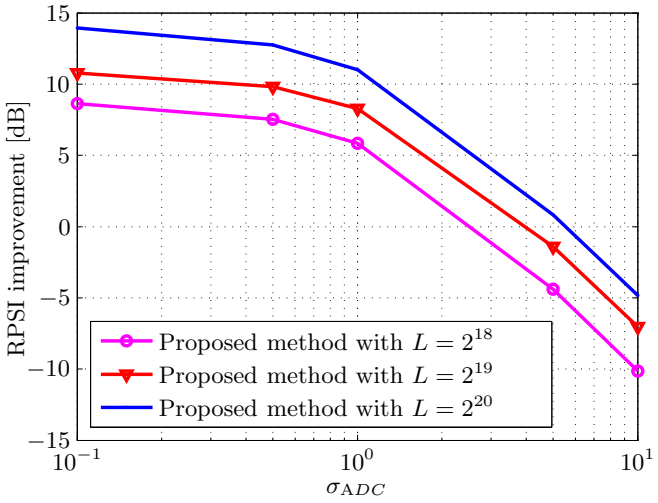


Figure 2.17: The plot shows the trend in suppressing the spurious sidebands using the same parameters in Fig. 2.10 with a fixed 10 bit ADC and varying the ADC jitter σ_{ADC} (as a percentage to the sampling period).

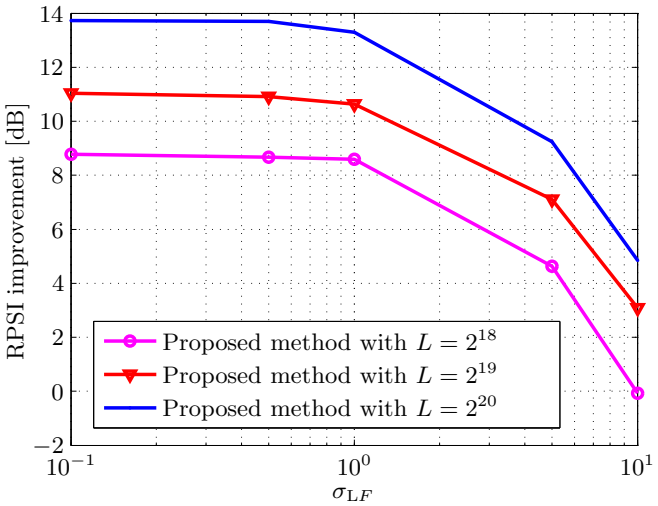


Figure 2.18: The plot uses the same parameters as in Fig. 2.17, except that $\sigma_{ADC} = 0$ and σ_{LF} is varied.

Let α, β be distinct points in $[a, b]$, and define

$$P(t) \triangleq h(\alpha) + \sum_{k=1}^{m-1} \frac{h^{(k)}(\alpha)}{k!}(t-\alpha)^k \tag{2.58}$$

Then Taylor's theorem states that there exists a point ξ between α and β such that

$$h(\beta) = P(\beta) + \frac{h^{(m)}(\xi)}{m!}(\beta - \alpha)^m \tag{2.59}$$

with exact equality in (2.59). The theorem shows that $h(t)$ can be approximated by $P(t)$ (a polynomial of degree $m-1$) and (2.59) allows us to estimate the error if we know bounds on $|h^{(m)}(\xi)|$.

First, we define a function $g_n(x)$ as

$$g_n(x) \triangleq x + \epsilon_s(nT_s + x) \tag{2.60}$$

where the root of $g_n(x)$ is at $x_n = e[n]$. Also note that $\hat{x}_n = -g_n(0)$. Using Taylor's theorem and setting $m = 1$, $\alpha = 0$, $\beta = x$ and $0 \leq \xi \leq x$, we use (2.59) to express $g_n(x)$ in (2.60) *exactly* as:

$$g_n(x) = g_n(0) + \dot{g}_n(\xi)\ x \tag{2.61}$$

where

$$\dot{g}_n(x) = 1 + \gamma \cos(2\pi f_{\text{ref}}(nT_s + x) + \theta_0) \tag{2.62a}$$

$$\gamma \triangleq C_0 \frac{f_{\text{ref}}}{f_s}, \quad 0 < \gamma < 1 \tag{2.62b}$$

The condition $\gamma = C_0 f_{\text{ref}}/f_s < 1$ is assumed to hold since the frequency, f_{ref}, of the reference signal and C_0 in a typical PLL system are small compared to the frequency, f_s, of the clock signal. Let $\{x_n, \xi_n\}$ be the solution to $g_n(x_n) = 0$. Then (2.61) gives

$$x_n = -\frac{g_n(0)}{\dot{g}_n(\xi_n)} = e[n] \tag{2.63}$$

From (2.62), the term $\dot{g}_n(\xi_n)$ can be bounded as

$$0 \leq 1 - \gamma \leq \dot{g}_n(\xi_n) \leq 1 + \gamma \tag{2.64}$$

Therefore, if $g_n(0) < 0$ then x_n is bounded by

$$\frac{-g_n(0)}{1+\gamma} \leq x_n \leq \frac{-g_n(0)}{1-\gamma} \tag{2.65}$$

If instead $g_n(0) > 0$, the lower and upper bounds on x_n are reversed. Now observe from (2.60) that $\hat{x}_n$ is

$$\begin{aligned} \hat{x}_n &= -\epsilon_s[n] \\ &= -g_n(0) \end{aligned} \tag{2.66}$$

Therefore,

$$\begin{cases} g_n(0)\left(\frac{-\gamma}{1-\gamma}\right) \leq x_n - \hat{x}_n \leq g_n(0)\left(\frac{\gamma}{1+\gamma}\right), & \text{when } g_n(0) > 0 \\ g_n(0)\left(\frac{\gamma}{1+\gamma}\right) \leq x_n - \hat{x}_n \leq g_n(0)\left(\frac{-\gamma}{1-\gamma}\right), & \text{when } g_n(0) < 0 \end{cases} \tag{2.67}$$

The absolute error is bounded by

$$|\hat{x}_n - x_n| \leq |g_n(0)| \frac{\gamma}{1-\gamma} \tag{2.68}$$

From (2.63), if $g_n(0) \neq 0$ then $x_n \neq 0$. Therefore, we could use (2.63) and (2.68) to express the relative error as

$$\left|\frac{\hat{x}_n - x_n}{x_n}\right| \leq |\dot{g}_n(\xi_n)| \frac{\gamma}{1-\gamma} \tag{2.69}$$

Using (2.62) we note that $|\dot{g}_n(\xi_n)| \leq 1+\gamma$. We conclude that the relative error is bounded by

$$\left|\frac{\hat{x}_n - x_n}{x_n}\right| \leq \frac{\gamma(1+\gamma)}{1-\gamma} \tag{2.70}$$

2.B Modeling of phase noise in second-order PLL

This appendix describes the phase noise model used in the simulations for the random jitter in the high frequency jittery tone $f_y(t)$, and when additional random jitter is introduced into the ADC and training signal. From [9, 22], the

closed-loop transfer function for a second-order PLL is

$$H(s) = \frac{2\eta\omega_n s + \omega_n^2}{s^2 + 2\eta\omega_n s + \omega_n^2} \tag{2.71}$$

where ω_n is the loop natural frequency and η is the damping factor. If we let $\eta = \frac{1}{\sqrt{2}}$, then the single-sided PSD of the phase noise model for the second-order PLL can be shown to be

$$\begin{aligned} S_\phi(f) &= |1 - H(j2\pi f)|^2 \frac{\nu}{\pi f^2} \\ &= \frac{\nu f^2}{\pi(f^4 + f_n^4)} \end{aligned} \tag{2.72}$$

where $f_n = \frac{\omega_n}{2\pi}$ is a measure of the loop bandwidth and ν is called the oscillator linewidth. The variance is $\frac{\nu}{2\sqrt{2}f_n}$ and the autocorrelation function of the phase noise $\phi(t)$ is

$$R_\phi(\lambda) = \int_{-\infty}^{\infty} \frac{\nu f^2}{2\pi(f^4 + f_n^4)} e^{j2\pi f\lambda} df \tag{2.73}$$

Using the PSD and autocorrelation model, it is possible to simulate the phase noise in a second-order PLL. We can relate the phase noise to the time jitter in a clock signal with frequency f_s as $\phi(t) = 2\pi f_s \tau(t)$. The standard deviation of the random jitter is $\frac{1}{2\pi f_s}\sigma_\phi$. Normalizing the standard deviation to the sampling interval yields $\sigma_\tau = \frac{1}{2\pi}\sigma_\phi$.

2.C Effect of jitter in $\check{y}(n)$ on training signal extraction

Based on the simulation parameters in section 2.6, we fix the standard deviation of the jitter in $y(t)$ to be 1% of the sampling period. Equivalently, this means that the standard deviation of the phase noise is 1% of 2π. From Appendix 2.B, the variance of the phase noise is $\frac{\nu}{2\sqrt{2}f_n}$. In this work, f_n is set to 5 MHz. Hence, ν is 5.58×10^4 Hz.

In section 2.4.2, we like to retain the low-frequency signal $\check{w}(n)$ (as shown in Figure 2.4). The highest frequency component of $\check{w}(n)$ is at 60 MHz. Also, we we see that the closest jitter's spectrum is the one that is centered at 120 MHz. The offset between the two frequencies is 60 MHz. Using (2.28) in section 2.4.2 and (2.72) in Appendix 2.B, the noise power from the jitter onto the highest frequency component of $\check{w}(n)$ is

$$\frac{A_w^2 f_y^2 B}{16 f_s^2} S_\phi(f_d) \tag{2.74}$$

where f_d is the offset frequency, B is the frequency resolution of the PSD. Using the parameters in the simulations, $A_w = 0.1$ V, $f_y = 420$ MHz, $f_s = 1$ GHz and $B \approx 1$ KHz, the noise power at 60 MHz is -122 dB.

2.D Phase estimation

This appendix describes a way to estimate the phases in the tones in (2.30), which are used in the phase recovery of the signal extraction block diagram in Figure 2.5. To estimate the phase of a sinusoid at frequency f_k in a signal of the form

$$s[n] = \sum_{k=0}^{K} A_k \cos(2\pi f_k n T_s + \theta_k) \tag{2.75}$$

we collect N samples and calculate the following expressions:

$$w_{ma} = \frac{1}{N} \sum_{n=0}^{N-1} s[n] \cos(2\pi f_k n T_s) \tag{2.76a}$$

$$w_{mb} = \frac{1}{N} \sum_{n=0}^{N-1} s[n] \sin(2\pi f_k n T_s) \tag{2.76b}$$

Substituting (2.75) into (2.76) gives:

$$\begin{aligned}
w_{ma} &= \frac{1}{N}\sum_{n=0}^{N-1} s[n]\ \cos(2\pi f_k n T_s) \\
&= \frac{1}{N}\sum_{n=0}^{N-1}\sum_{\ell=0}^{K}\ A_\ell \cos(2\pi f_\ell n T_s + \theta_k)\cos(2\pi f_k n T_s) \\
&= \frac{1}{N}\sum_{n=0}^{N-1}\sum_{\ell=0}^{K}\ \frac{A_\ell}{2}\left[\cos(2\pi(f_\ell + f_k)nT_s + \theta_k) + \cos(2\pi(f_\ell - f_k)nT_s + \theta_k)\right] \\
&= \frac{1}{N}\sum_{n=0}^{N-1}\frac{A_k}{2}\cos(\theta_k) + \sum_{\ell=0}^{K}\frac{1}{N}\sum_{n=0}^{N-1}\ \frac{A_\ell}{2}\cos(2\pi(f_\ell + f_k)nT_s + \theta_k) \\
&\quad + \sum_{\ell=0,\ell\neq k}^{K}\frac{1}{N}\sum_{n=0}^{N-1}\ \frac{A_\ell}{2}\cos(2\pi(f_\ell - f_k)nT_s + \theta_k) \\
&\approx \frac{A_k}{2}\cos(\theta_k)
\end{aligned} \tag{2.77}$$

and

$$\begin{aligned}
w_{mb} &= \frac{1}{N}\sum_{n=0}^{N-1} s[n]\ \sin(2\pi f_k n T_s) \\
&= \frac{1}{N}\sum_{n=0}^{N-1}\sum_{\ell=0}^{K}\ A_\ell \cos(2\pi f_\ell n T_s + \theta_k)\sin(2\pi f_k n T_s) \\
&= \frac{1}{N}\sum_{n=0}^{N-1}\sum_{\ell=0}^{K}\ \frac{A_\ell}{2}\left[\sin(2\pi(f_\ell + f_k)nT_s + \theta_k) - \sin(2\pi(f_\ell - f_k)nT_s + \theta_k)\right] \\
&= -\frac{1}{N}\sum_{n=0}^{N-1}\frac{A_k}{2}\sin(\theta_k) + \sum_{\ell=0}^{K}\frac{1}{N}\sum_{n=0}^{N-1}\ \frac{A_\ell}{2}\sin(2\pi(f_\ell + f_k)nT_s + \theta_k) \\
&\quad - \sum_{\ell=0,\ell\neq k}^{K}\frac{1}{N}\sum_{n=0}^{N-1}\ \frac{A_\ell}{2}\sin(2\pi(f_\ell - f_k)nT_s + \theta_k) \\
&\approx -\frac{A_k}{2}\sin(\theta_k)
\end{aligned} \tag{2.78}$$

Then we estimate the phases θ_k using

$$\hat{\theta}_k = -\tan^{-1}\left(\frac{w_{mb}}{w_{ma}}\right) \tag{2.79}$$

CHAPTER 3

Compensating Spurious PLL Tones in Spectrum Sensing Architectures

In the previous chapter, we showed that spurious tones in the sampling clock creates spurious tones in the sampled data of the ADCs. In applications like spectrum sensing in cognitive radios, spurious tones might give a false positive detection on actual free channels. To share the frequency spectrum effectively (such as unused TV bands) there is a need to design better receivers that can reliably sense free spectrum holes even in the presence of spurious tones [45–47], especially since cognitive radios are expected to be able to detect very weak signals [46, 48]. There are various methods that have been used in the literature to detect such signals, including, energy detection methods, matched filtering methods and feature detection methods [46, 48, 49]. Moreover, for wideband applications, it is common to split the spectrum into smaller channels for detection. For example, references [50, 51] use fast Fourier transform (FFT) to channelize the spectrum before energy detection. Alternative ways to channelize the spectrum also exist [52–54]. Reference [52] proposed a method to sense the channels serially using a reconfigurable downconverter and filter. Reference [53] proposed a low-power multiresolution spectrum sensing IC which uses energy detection. Reference [54] proposed and analyzed a two-stage sensing technique that performs a coarse resolution sensing (CRS) followed by a fine resolution sensing (FRS).

The CRS and FRS are performed using a random search and a sequential search, respectively. In these techniques, it is assumed that an ideal clock is used in the ADC [50, 51], or in the DDC reference frequency generator [52], or in the mixer [53]. Here, we show that when the spurious tones in the non-ideal clock of the ADC are not considered, the false alarm rate will increase. Even if the difficulty caused by the spurious tones is considered and the thresholds are raised to reduce the false alarm, the resulting detection rates are likely to decrease. For these reasons, it is desirable to seek an alternative approach to remove the spurious sidebands from the sampled data and improve the sensing performance.

In this chapter, we extend the work in the previous chapter from using a sinusoidal reference signal to using a general periodic reference signal, and propose a new approach [18, 55] that relies specifically on the use of a Fourier transform block (since it is a common building block in wideband applications such as spectrum sensing). By reusing existing components, we aim to reduce the hardware complexity and computation cost when estimating the distortions. The work here proposes a modification to a spectrum sensing architecture by first performing offset estimation on a training sinusoidal signal and then switching to compensating the distorted samples to obtain the dejittered samples for spectrum sensing. The offset estimation algorithm (using the training signal and the Fourier transform), and the compensation algorithm (using first-order Taylor series) are called the TFT and FOT, respectively. We provide detailed derivations of the distortion sampling offsets, show that replicas of the modulated signals are created when the modulated signals are sampled non-uniformly with the derived offsets, evaluate the impact of this distortion during spectrum sensing, derive the theoretical detection performance for some signals, and simulate to verify the detection performance before and after using the proposed algorithm. To evaluate the effects on detection performance, we consider the case when the signal of interest is a

weak signal. We also assume that there is a strong signal at some frequency offset from the signal of interest. The distortions to the sampled data create spurious sidebands from the strong signal and we assume that the spurious sidebands overlap with the signal of interest. Hence, the spurious sidebands become an interference to the detection of the weak signals.

The chapter is organized as follows. Section 3.1 discusses a mathematical model for a VCO clock and what happens when the reference signal is leaked into the control line of the VCO. It also shows the effect on sampling instants and the sampled data when the sampling clock is used with an ADC. Section 3.2 proposes the TFT that estimates the sidebands of the sampled data using a Fourier transform block, and the FOT that compensates for the distortions. Section 3.3 considers the detection performance using an energy detector in the presence of the spurious sidebands distortion. Section 3.4 presents the simulation results and Section 3.5 summarizes the chapter.

3.1 Effects of leakage from reference signal

3.1.1 Reference leakage in PLL

In [22, 37], a voltage-controlled oscillator (VCO) is described as a circuit that generates a periodic clock signal, $s(t)$, whose frequency is a linear function of a control voltage, V_{cont}. Let the gain of the VCO and its "free running" frequency be denoted by K_{vco} and f_s, respectively. The generated clock signal is described by

$$s(t) = A_s \sin\left(2\pi f_s t + K_{\text{vco}} \int_{-\infty}^{t} V_{\text{cont}} dt\right) \tag{3.1}$$

To attain some desired oscillation frequency, V_{cont} is set to some constant value. However, the generated signal, $s(t)$, may not be an accurate tone. To attain good

frequency synthesis, a frequency-divided version of the clock signal is fed into a block that consists of a PFD, a CP and a low-pass filter (LPF) as shown in Fig. 3.1. The PFD/CP/LPF block compares the frequency-divided clock signal with

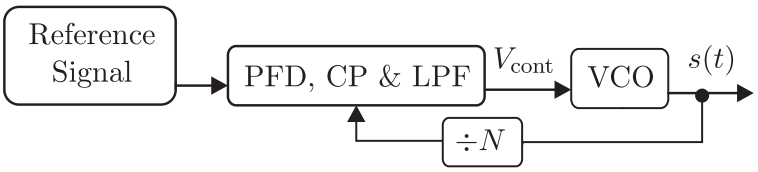


Figure 3.1: Block diagram of a PLL.

a low-frequency reference signal at f_{ref} and makes adjustments to V_{cont}. The low-frequency reference signal can be generated using a low-frequency oscillator (e.g., a crystal oscillator). Due to imperfections in the circuitry, the reference signal leaks into the control line of the VCO. Also note that the presence of the LPF attenuates the leakage signal but fails to remove it completely. For simplicity, we assume that the desired clock signal at f_s is obtained when V_{cont} is 0. The reference leakage is assumed to be some periodic signal with fundamental frequency f_{ref} [40]. From [40], one source of reference leakage in the PFD/CP/LPF block is due to the current mismatch in the CP. The current mismatch creates rectangular pulses which are passed through the LPF. Hence, the higher frequencies components are attenuated and the leakage waveform changes (depending on the LPF's frequency response). Another source of leakage is from the reference signal and the PFD. Similarly, the leakage waveform is also changed due to the LPF. In this paper, we do not need to know the exact waveform except that it is a periodic signal. For illustration purposes, we assume here that the periodic signal is a triangular waveform and can be described by its Fourier series representation. First, note that the Fourier series representation of a triangular waveform with

peak amplitude of 1 and a fundamental frequency of f_{ref} is given by:

$$\begin{aligned} V_r(t) &= \frac{8}{\pi^2}\sum_{k=0}^{\infty}\frac{(-1)^k}{(2k+1)^2}\sin((2k+1)2\pi f_{\text{ref}}t) \\ &= \frac{8}{\pi^2}\sum_{k=0}^{\infty}\frac{(-1)^k}{(2k+1)^2}\cos((2k+1)2\pi f_{\text{ref}}t - \frac{\pi}{2}) \end{aligned} \tag{3.2}$$

Now, suppose there is leakage into the control line so that V_{cont} becomes a delayed and scaled version of $V_r(t)$, say,

$$V_{\text{cont}} = \sum_{k=0}^{\infty} V_k \cos\left(2\pi f_k t + \theta_k\right) \tag{3.3}$$

where

$$\theta_k = 2\pi f_{\text{ref}}(2k+1)\tau - \frac{\pi}{2} \tag{3.4a}$$

$$V_k = V_0\frac{(-1)^k}{(2k+1)^2} \tag{3.4b}$$

$$f_k = (2k+1)f_{\text{ref}} \tag{3.4c}$$

for some $\{V_0, \tau\}$.

3.1.2 Effect of leakage on the sampling clock of the ADC

Then, using (3.1), the output of the VCO becomes

$$s(t) = A_s \sin\left(2\pi f_s t + \epsilon_s(t) + \phi_s\right) \tag{3.5}$$

where ϕ_s is some unknown phase offset and

$$\epsilon_s(t) = \sum_{k=0}^{\infty} C_k \sin\left(2\pi f_k t + \theta_k\right) \tag{3.6}$$

and

$$C_k = \frac{K_{\text{vco}}}{2\pi f_k}V_k \tag{3.7}$$

We assume that the presence of the LPF before the control line of the VCO attenuates the reference leakage to some extent (but is not able to remove it completely) (3.3) so that it is reasonable to assume that

$$|\epsilon_s(t)| \ll 1 \tag{3.8}$$

We will be analyzing the signal model with respect to an arbitrary reference time. Using a change of variables, let $t = t' - \frac{\phi_s}{2\pi f_s}$, and substitute t into equations (3.3) and (3.5). The new equations are similar to the original equations except that ϕ_s is 0. Therefore, we can let $\phi_s = 0$ without loss of generality. Applying a first order approximation to (3.5) we get:

$$\begin{aligned} s(t) \approx A_s \sin(2\pi f_s t) + \sum_{k=0}^{\infty} \Bigg[& \frac{A_s C_k}{2} \sin(2\pi(f_s + f_k)t + \theta_k) \\ & - \frac{A_s C_k}{2} \sin(2\pi(f_s - f_k)t - \theta_k) \Bigg] \end{aligned} \tag{3.9}$$

This expansion shows that the distorted sampling clock signal contains multiple sidebands at $f_s \pm f_k$. Now the actual sampling instants of an ADC that uses (3.5) as the clock signal are the zero-crossings of $s(t)$. Using (3.5) and defining $T_s = 1/f_s$, the sampling instants, t_n, of the ADC must satisfy the condition:

$$\begin{aligned} t_n + \frac{\epsilon_s(t_n)}{2\pi f_s} &= nT_s \\ t_n &= nT_s - \frac{\epsilon_s(t_n)}{2\pi f_s} \end{aligned} \tag{3.10}$$

This is a nonlinear equation in t_n. We solve it as follows. Let

$$t_n \triangleq nT_s + e[n] \tag{3.11}$$

for some perturbation terms $e[n]$ that we wish to determine. From (3.10) we have that

$$\begin{aligned} e[n] &= -\frac{\epsilon_s(t_n)}{2\pi f_s} \\ &= -\frac{1}{2\pi f_s} \epsilon_s \left(nT_s - \frac{\epsilon_s(t_n)}{2\pi f_s} \right) \end{aligned} \tag{3.12}$$

Since $|\epsilon_s(t)| \ll 1$,

$$\left|\frac{\epsilon_s(t_n)}{2\pi f_s}\right| \ll T_s \tag{3.13}$$

Therefore, the discrete sequence of offsets $e[n]$ is approximated as

$$e[n] \approx -\frac{\epsilon_s[n]}{2\pi f_s} \tag{3.14}$$

3.1.3 Effect of distorted sampling offsets on training signal

Let us now analyze the effect of this distorted sampling on a pure sinusoidal training tone at the input of the ADC. Since the frequency of the training signal is much lower than the frequency of the sampling clock generated by the PLL, it is reasonable to assume that a low frequency oscillator (e.g., a crystal oscillator) can be used to accurately generate the signal. As such, we assume that the training signal does not have the same distortion problems as the sampling clock. Let the input signal to the ADC be

$$w(t) = A_w \cos(2\pi f_w t + \phi_w) \tag{3.15}$$

Then the sampled signal, $\check{w}[n]$, is approximated as

$$\begin{aligned} \check{w}[n] &\approx w\left(nT_s + e[n]\right) \\ &\approx w[n] + e[n]\,\dot{w}[n] \end{aligned} \tag{3.16}$$

where

$$\begin{aligned} w[n] &= w(t)|_{t=nt_s} \\ \dot{w}[n] &= \dot{w}(t)|_{t=nt_s} \end{aligned} \tag{3.17}$$

Using trigonometry expansions, the term $e[n]\,\dot{w}[n]$ in (3.16) is

$$\begin{aligned} e[n]\dot{w}[n] = \sum_{k=0}^{\infty} \frac{f_w A_w C_k}{2f_s} \big[&\cos(2\pi(f_w - f_k)nT_s + \phi_w - \theta_k) \\ &- \cos(2\pi(f_w + f_k)nT_s + \phi_w + \theta_k)\big] \end{aligned} \tag{3.18}$$

The above expression shows that the sampled data consists of the input signal and multiple frequency components at $f_w \pm f_k$. If the magnitude of the Fourier series coefficients of the reference signal in the PLL decreases rapidly, then the higher frequencies components in (3.14) and (3.18) can be ignored. Observe that the amplitude term in (3.18) is directly proportional to the f_w. This means that the power of the spurious sidebands increases with the frequency of the input signal.

It is possible to relate the power of the sidebands in the sampled data (3.18) to the sidebands in the sampling clock (3.9). For example, suppose the power ratio of the sideband at $f_s + f_0$ of the clock, $s(t)$, to the tone at f_s is -50 dBc, then C_0 is 6.32×10^{-3}. Thus, the power of the sideband at $f_w + f_0$ of the sampled data can be derived. As an example, the reference leakage in the PLL is simulated as a triangular wave with a fundamental frequency of 20 MHz and is approximated using the first 4 Fourier series coefficients. A sinusoidal training signal at 45 MHz is distorted by the jittered sampling and its power spectral density (PSD) is shown in Fig. 3.2. From the plot, only the sidebands at 25 MHz, 65 MHz and 105 MHz are detected (i.e., the effects from the first 2 Fourier series coefficients).

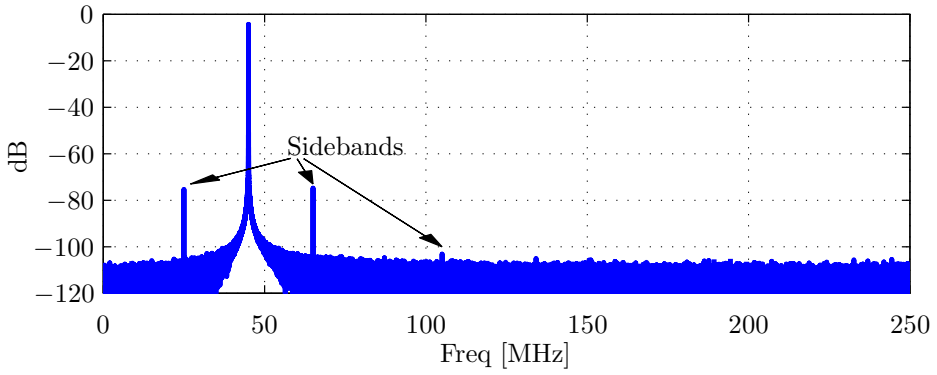


Figure 3.2: The plot shows the PSD of the distorted training signal with two sidebands.

3.1.4 Effects of the spurious sidebands on spectrum sensing

The previous section shows that the ADC creates spurious tones in the sampled data. These tones are not supposed to exist and can lead to false alarm in wideband spectrum sensing applications. Moreover, we can further show that when the input signal is a modulated signal, then the distorted samples create replicas of the modulated signal at some offset frequencies. Specifically, suppose now that $w(t)$ is a modulated signal of the form:

$$w(t) = m(t)\cos(2\pi f_w t + \phi_w) \tag{3.19}$$

where $m(t)$ is the message signal and f_w is the carrier frequency. Differentiating $w(t)$ gives

$$\begin{aligned} \dot{w}(t) &= \dot{m}(t)\cos(2\pi f_w t + \phi_w) - 2\pi f_w m(t)\sin(2\pi f_w t + \phi_w) \\ &\approx -2\pi f_w m(t)\sin(2\pi f_w t + \phi_w) \end{aligned} \tag{3.20}$$

The first term in the first line of (3.20) is removed because we assume that $m(t)$ is a bandlimited baseband signal, whose frequency components are much smaller than the carrier frequency f_w. As such, we assume that $\dot{m}(t)$ is small relative to $2\pi f_w m(t)$. Consequently, the distorted samples become

$$\begin{aligned} \breve{w}[n] &\approx w[n] + e[n]\,\dot{w}[n] \\ &= m[n]\cos(2\pi f_w n T_s + \phi_w) + e[n]\,\dot{w}[n] \end{aligned} \tag{3.21}$$

where the sidebands are in the second term:

$$\begin{aligned} e[n]\dot{w}[n] = \sum_{k=0}^{\infty} \frac{f_w C_k}{2f_s} m[n][&\cos(2\pi(f_w - f_k)nT_s + \phi_w - \theta_k) \\ &- \cos(2\pi(f_w + f_k)nT_s + \phi_w + \theta_k)] \end{aligned} \tag{3.22}$$

The above expressions, as indicated earlier, show that the spurious sidebands are replicas of the original modulated signal at a lower amplitude and frequency-shifted by f_k.

3.2 Proposed solution

In a typical spectrum sensing application, there is usually a module that performs Short-Time Fourier Transform (STFT) with windowing functions. To save computation and hardware complexity, we will use this module as a building block in our proposed solution. Figure 3.3 shows the proposed architecture. We assume

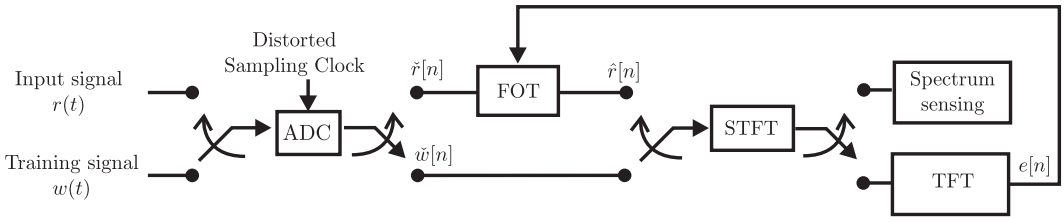


Figure 3.3: Proposed architecture for reducing the effects of PLL sidebands in spectrum sensing applications.

that the PLL is in tracking mode (when the loop is in lock) and the distortions to the sampled data due to the PLL sidebands can be estimated from a sinusoidal training tone $w(t)$. The distorted sampled data $\check{w}[n]$ are used with the STFT module to estimate the sampling distortions (using the TFT algorithm). Once the distortions (3.14) are estimated, the circuit switches and starts sampling the desired input signal and the sampled data is corrected in the digital domain before the spectrum sensing application (using the FOT algorithm).

3.2.1 TFT algorithm

We can use the results in (3.18) to evaluate the sampling offsets' parameters $\{C_k, \theta_k\}$ in (3.14) from the sidebands present in $\check{w}[n]$. First, we express (3.15) as

$$w[n] = A_w \cos(2\pi f_w n T_s + \phi_w) \tag{3.23}$$

Let us assume we have estimated the amplitude and phase of the tones in (3.18). We will show how to estimate them further ahead (see (3.31)). Let $Z(f) = Ae^{j\theta}$ denote the complex representation of the estimated amplitude A and phase θ at frequency f. Let * denote complex conjugation. Then, using (3.23) and (3.18), $\{C_k, \theta_k\}$ can be estimated from the relation

$$\begin{aligned} C_k e^{j\theta_k} &= \frac{2f_s e^{-j\pi}}{f_w}\left(\frac{Z(f_w + f_k)}{Z(f_w)}\right) \\ &= \frac{2f_s e^{-j\pi}}{f_w}\left(\frac{Z(f_w + f_k)Z^*(f_w)}{|Z(f_w)|^2}\right) \end{aligned} \tag{3.24}$$

or,

$$\begin{aligned} C_k e^{j\theta_k} &= \frac{2f_s}{f_w}\left(\frac{Z^*(f_w - f_k)}{Z^*(f_w)}\right) \\ &= \frac{2f_s}{f_w}\left(\frac{Z^*(f_w - f_k)Z(f_w)}{|Z(f_w)|^2}\right) \end{aligned} \tag{3.25}$$

The question now is how to estimate the sinusoidal sidebands to enable evaluation of $\{\hat{C}_k, \hat{\theta}_k\}$ through (3.24) or (3.25). As mentioned before, in spectrum sensing applications, there is a module that performs STFT with windowing. Essentially, this module splits the data into different frequency bins for further processing. The operation of the STFT is as follows [56].

In the m-th iteration of STFT, an N-point Fast Fourier Transform (N-FFT) is applied on an N-point data sequence with a windowing function $w_x[n]$. Let us assume that data sequences do not overlap and let us denote the data sequences by $x[n + Nm]$. Thus, the STFT output is

$$X[m, k] = \sum_{n=0}^{N-1} w_x[n]x[n + Nm]e^{-j2\pi\frac{k}{N}n} \tag{3.26}$$

where k is a particular frequency bin in the N-FFT. Suppose we want to estimate

the amplitude and phase of a sinusoid at frequency f_p in a signal of the form:

$$\begin{aligned} x[n] &= \sum_{i=0}^{P} A_i \cos(2\pi f_i n T_s + \theta_i) \\ &= \sum_{i=0}^{P} \frac{A_i}{2} \left[e^{j(2\pi f_i T_s n + \theta_p)} + e^{-j(2\pi f_i T_s n + \theta_p)} \right] \end{aligned} \tag{3.27}$$

and only the sinusoid lies in the p-th frequency bin, then the STFT output of the bin is

$$X[m,p] = X_p[m,p] + X_n[m,p] \tag{3.28}$$

where

$$X_p[m,p] = A_p \left[a_p e^{j(2\pi f_p T_s N m + \theta_p)} + b_p e^{-j(2\pi f_p T_s N m + \theta_p)} \right] \tag{3.29a}$$

$$a_p = \frac{1}{2} \sum_{n=0}^{N-1} w_x[n] e^{j2\pi(f_p T_s - \frac{p}{N})n} \tag{3.29b}$$

$$b_p = \frac{1}{2} \sum_{n=0}^{N-1} w_x[n] e^{-j2\pi(f_p T_s + \frac{p}{N})n} \tag{3.29c}$$

The $X_n[m,p]$ are nuisance terms involving the rest of the frequency components in $x[n]$ that are out of the p-th frequency band of the FFT. Using proper windowing functions $w_x[n]$, we can attenuate the effect of $X_n[m,p]$. As an example, a STFT using 1024-pt FFT is applied on the training signal shown in Fig. 3.2. The STFT output in the frequency bin that contains the sideband tone at 65MHz is extracted and its frequency spectrum is plotted in Fig. 3.4. The left and right plots show the result when no windowing is used, i.e., $w_x[n] = 1$ and when a Blackman-Harris window is used, respectively. As shown, the window function reduces the spectral leakage of out-of-band signals into the frequency channel.

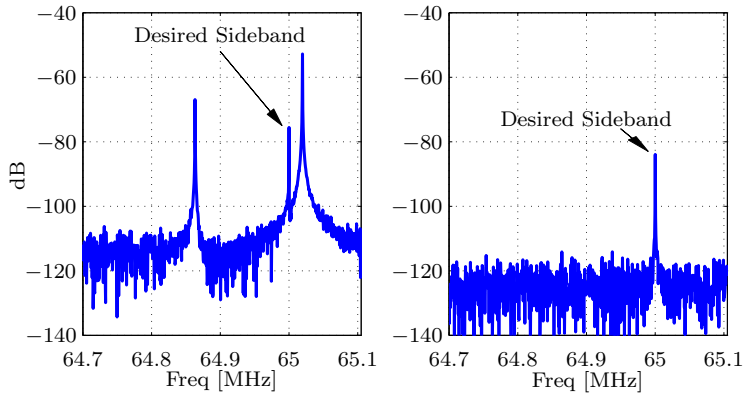


Figure 3.4: The plots show the frequency domain output of a frequency channel in the STFT when no windowing is used (left) and windowing is used (right).

Thus, manipulating $X(m,p)$ yields

$$\begin{aligned} d_p[m] &\triangleq \frac{1}{|a_p|^2 - |b_p|^2} \begin{bmatrix} a_p^* & -b_p \end{bmatrix} \begin{bmatrix} X[m,p] \\ X[m,p]^* \end{bmatrix} \\ &= A_p e^{j\theta_p} e^{j2\pi f_p T_s N m} + \nu[m] \\ &= Z(f_p) e^{j2\pi f_p T_s N m} + \nu[m] \end{aligned} \tag{3.30}$$

where $\nu[m]$ is some noise residual in terms of $X_n[m,p]$. Thus, we can estimate $Z(f_p)$ from the data $d_p[m]$ using M samples.

$$\hat{Z}(f_p) = \frac{1}{M} \sum_{m=0}^{M-1} d_p[m] e^{-j2\pi f_p T_s N m} \tag{3.31}$$

The TFT algorithm is summarized in Algorithm 3.1.

3.2.2 Block diagram of TFT

The TFT algorithm described in the previous section can be converted to a block diagram as shown in Fig. 3.5. First, let us assume that we are interested in suppressing the spurious sidebands at f_0 and f_1 away from the training signal's frequency (f_w). The data $\check{w}[n]$ is segmented into blocks of size N before it is

Algorithm 3.1 Summary of TFT algorithm

Require: Denote the frequency of the training signal as $f_{p0} = f_w$ and the frequency of the sidebands as $f_{pk} = f_w + f_{k-1}$, $\{k = 1, \cdots, K\}$. Also, denote the STFT channels that the training signal and sidebands lie in as p_0 and p_k, $\{k = 1, \cdots, K\}$, respectively. Finally, precompute a_{pk} and b_{pk}, $\{k = 0, \cdots, K\}$, from (3.29).

for $m = 0, \cdots, M-1$ **do**

for $k = 0, \cdots, K$ **do**

$$d_{pk}[m] = \frac{1}{|a_{pk}|^2 - |b_{pk}|^2} \begin{bmatrix} a_{pk}^* & -b_{pk} \end{bmatrix} \begin{bmatrix} X[m, p_k] \\ X[m, p_k]^* \end{bmatrix}$$

end for

end for

for $k = 0, \cdots, K$ **do**

$$\hat{Z}(f_{pk}) = \frac{1}{M} \sum_{m=0}^{M-1} d_{pk}[m] e^{-j2\pi f_{pk} T_s N m}$$

end for

for $k = 1, \cdots, K$ **do**

$$\hat{C}_{k-1} e^{j\hat{\theta}_{k-1}} = \frac{2 f_s e^{-j\pi}}{f_w} \left(\frac{\hat{Z}(f_{pk})}{\hat{Z}(f_w)} \right)$$

end for

$$\epsilon_s[n] = \sum_{k=0}^{K-1} \hat{C}_k \sin\left(2\pi n f_k T_s + \hat{\theta}_k\right)$$

$$e[n] = -\frac{\epsilon_s[n]}{2\pi f_s}$$

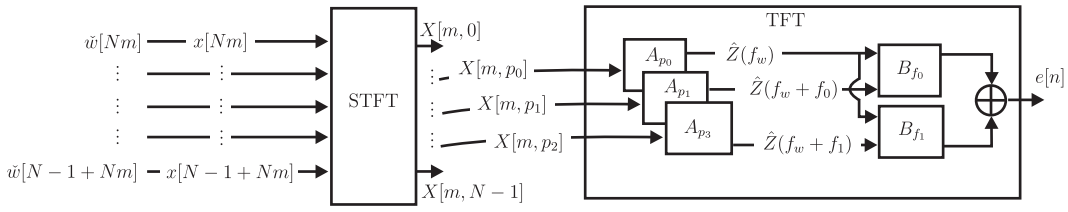


Figure 3.5: Block diagram of the TFT algorithm. The STFT, A_{pk} and B_{fk} blocks represent the different stages of the proposed algorithm and their block diagrams are shown in Fig. 3.6, Fig. 3.7 and Fig. 3.8, respectively. Firstly, the STFT block channelizes the training signal $\breve{w}[n]$. Then, the parameters of the sidebands in selected channels (p_k) are estimated using the A_{pk} blocks. Finally, the B_{fk} blocks generate some data streams which are added together to obtain the estimated sampling offsets $e[n]$.

processed by the existing STFT block. The input to the STFT is $x[n] = \breve{w}[n]$. The STFT block performs the windowed FFT and outputs $X[m,k]$ for the mth channel (step 1 of TFT algorithm). The detailed block diagram of STFT is shown in Fig. 3.6. We denote multiplication with fixed constant by a triangular symbol. We also assume that $X[m,p_0]$, $X[m,p_1]$ and $X[m,p_2]$, contain the training signal at f_w and its spurious sidebands at f_w+f_0 and f_w+f_1, respectively.

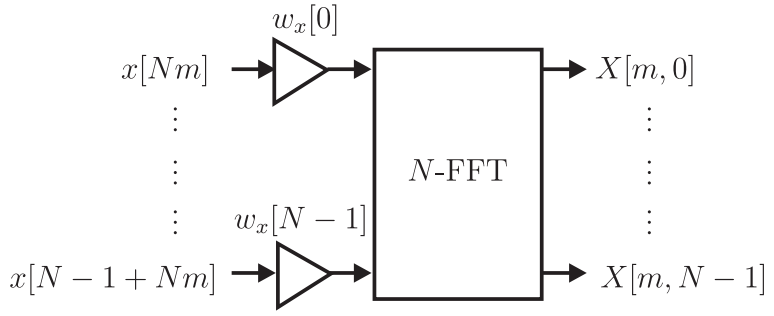


Figure 3.6: Block diagram of STFT.

In the next stage (A_{pk} blocks) of Fig. 3.5, the phase and amplitude of the tone in each of the STFT channels $X[m,p_0]$, $X[m,p_1]$ and $X[m,p_2]$ are estimated as

shown in Fig. 3.7. As shown in the figure, $X[m, p_k]$ and its conjugate $X^*[m, p_k]$ is multiplied by some fixed constants in terms of a_{pk} and b_{pk} and they sum together to $d_{pk}[m]$ (step 2 of TFT algorithm). Then, $d_{pk}[m]$ is multiplied with a complex exponential signal to estimate $\hat{Z}(f_{pk})$ (step 3 of TFT algorithm).

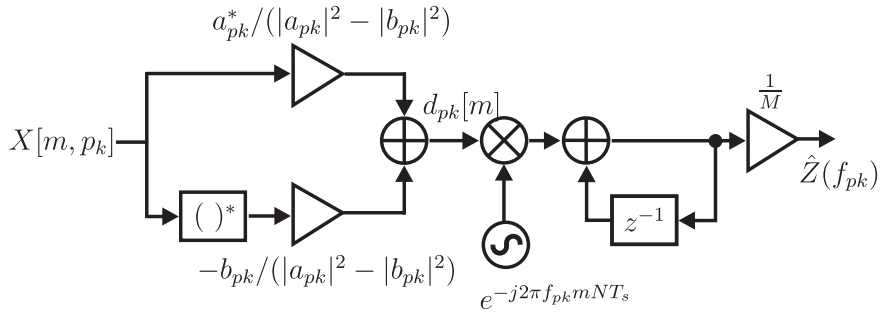


Figure 3.7: Block diagram of A_{pk} block.

In the last stage (B_{fk} blocks), $\{C_k, \theta_k\}$ is estimated as shown in Fig. 3.8 (step 4 of TFT algorithm). After $\{C_k, \theta_k\}$ is estimated, the waveform $\frac{-C_k}{2\pi f_s}\sin(2\pi f_k nT_s + \theta_k)$ is generated at the output of B_{fk}. Finally, the offset $e[n]$ is obtained by summing all the outputs of the B_{fk} blocks as shown in Fig. 3.5 (step 5 of TFT algorithm).

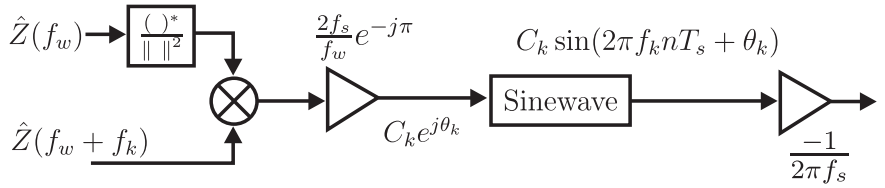


Figure 3.8: Block diagram of B_{fk} block.

3.2.3 FOT algorithm

Once the sampling offsets are known, the next step is to compensate the sampled data to obtain the desired samples. There exist various useful techniques in

the literature [57–59] that reconstruct signals from non-uniformly sampled data. For example, [57] proposed a technique based on Taylor series and derived a differentiator-multiplier cascade (DMC) system to reconstruct signals using up to the third order of the Taylor series. Reference [58] proposed a method based on Lagrange interpolation and modified it for band-limited signals. This modified Lagrange interpolator is called the functionally weighted (FW) interpolator. Reference [59] described the barycentric interpolator and showed that the FW interpolator in [58] can be converted to a barycentric interpolator. Reference [59] states that the reconstruction performance of the algorithms in [58] and [59] is similar. The algorithms in [57–59] are able to handle large sampling offsets. For example, [58, 59] can handle sampling offsets up to half of the sampling period.

In this work, we use a similar method as our previous works [17, 36, 41, 55] to dejitter the sampled data. The method is motivated using a first-order Taylor series and it is denoted as FOT. As such, it is the same as stage 1 of the DMC in [57]. We have also evaluated the performance if we use the techniques from [57, 58]. We do not compare with the barycentric interpolator from [59] since the performance is similar to the FW interpolator from [58]. We found that under the simulation settings used in this work, when the distortion sampling offsets are small (0.1% of the sampling period), no significant improvement is provided by these other methods. However, when the size of the distortion sampling offsets is increased, the techniques in [57–59] can be useful albeit at higher computational complexity. The details of the comparison are described in Appendix 3.B. Therefore, our proposed technique is able to compensate for the effect of small sampling distortions using an efficient architecture that exploits the convenience of the FFT operation.

The desired data, $r[n]$, can be expressed as:

$$\begin{aligned} r[n] &\triangleq r(nT_s) \\ &= r\left(nT_s + e[n] - e[n]\right) \\ &\approx r(nT_s + e[n]) - e[n]\dot{r}(nT_s + e[n]) \\ &= \check{r}[n] - e[n]\dot{r}(nT_s + e[n]) \end{aligned} \tag{3.32}$$

where $\check{r}[n]$ are the distorted samples, $e[n]$ are the estimated sampling errors (see (3.14)), and $\dot{r}(nT_s + e[n])$ are the derivatives of $r(t)$ at $t = nT_s + e[n]$. A block diagram showing the FOT algorithm is illustrated in Fig. 3.9. The derivatives can be approximated using a discrete filter applied to $\check{r}[n]$. The ideal frequency response of a derivative is a slope (see Fig. 3.10). Here, we like to compensate the input signal up to 200 MHz (or 0.8π rad/sample). Therefore, we designed a 15-tap filter where we minimized the frequency response error (from 0 to 0.8π) using a norm-1 criterion. The frequency response of the 15-tap filter is shown in Fig. 3.10.

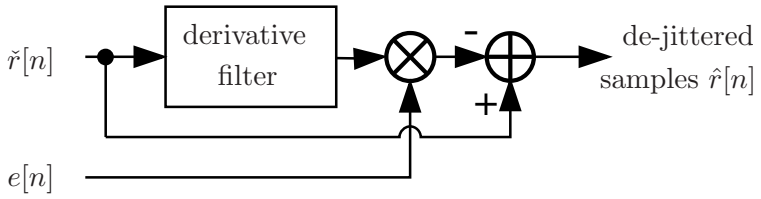


Figure 3.9: Block diagram of the FOT algorithm.

3.3 Detection of signals using an energy detector

In spectrum sensing, one way to detect signals at different frequencies is to use a STFT to channelize the wideband spectrum and to detect the presence of signals in each of the channels [49, 50].

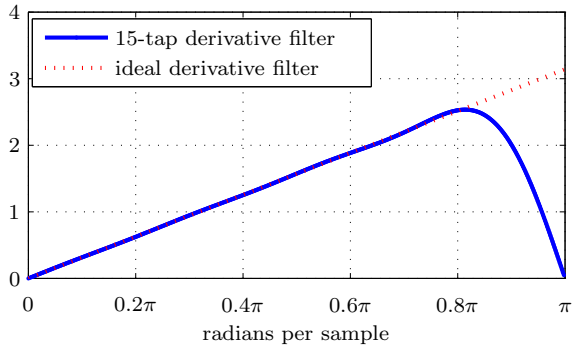


Figure 3.10: The plot shows the frequency response of the derivative filter.

In this work, we employ an energy detector to detect signals in frequency bins of the STFT. Recall from (3.26) that the STFT output at the kth frequency bin is $X[m,k]$. In the analysis, we assume that k is fixed and the energy detector is used on the kth frequency band. To simplify the notation, we drop k from the variable $X[m,k]$ and denote it as $X[m]$. The energy detector using L samples of $X[\ell]$ is defined as

$$T_L[X] \triangleq \frac{1}{L}\sum_{\ell=0}^{L-1}|X[\ell]|^2 \tag{3.33}$$

where L is the length of FFT snapshots used for averaging. We will consider different scenarios such as when the signal is a tone or an unknown white signal. We also consider the effect of spurious sidebands in the sampled data. In these scenarios, we assume that the signal of interest is a weak signal in the presence of white Gaussian noise. To consider the effect of spurious sidebands, we assume that there is a strong signal at some frequency offset from the signal of interest. In the ideal case, the strong signal does not affect the detection of the weak signal as they do not overlap in frequency. However, the distortions create spurious sidebands from the strong signal. Furthermore, we assume that the spurious sidebands from the strong signals overlap with the signal of interest. Therefore, the spurious sidebands interfere with the detection of the weak signals. Hence,

we will regard these spurious sidebands as interference.

3.3.1 No signal of interest - $\mathcal{H}_0$

Let suppose the signal of interest is absent and define this scenario as $\mathcal{H}_0$. In this case, $X[n]$ only contains white noise, say,

$$X[n; \mathcal{H}_0] = V[n] \tag{3.34}$$

where $V[n]$ is assumed to be zero-mean, white circular complex Gaussian noise with variance σ_v^2. From [60], the variable $\frac{L}{\sigma_v^2} T_L[X]$ has a chi-square distribution with L degrees of freedom. Therefore, the mean and variance of $T_L[X]$ are

$$E[T_L(X); \mathcal{H}_0] = \sigma_v^2 \tag{3.35a}$$

$$\text{var}[T_L(X); \mathcal{H}_0] = \frac{1}{L}\sigma_v^4 \tag{3.35b}$$

Using the Central Limit Theorem [61], when L is large, we can approximate the distribution of $T_L[X]$ under $\mathcal{H}_0$ as a normal distribution with the above mean and variance parameters.

3.3.2 Sinusoidal tone- $\mathcal{H}_1$

Let us now suppose that the signal of interest is a sinusoidal tone in the presence of white Gaussian noise and define this scenario as $\mathcal{H}_1$. This scenario occurs for some classes of signals. For example, in TV channels, two common transmission schemes are the national television system committee (NTSC) scheme and the advanced television standard committee (ATSC) scheme. The spectrum of the two schemes are shown in reference [48]. Both schemes have a bandwidth of 6 MHz. NTSC's spectrum contains three peaks in a bandwidth of 6 MHz, which represent the video, color and audio carriers. ATSC's spectrum is flat but has

a pilot tone located at the lower end of the channel. Hence, during spectrum sensing, some channels in the STFT block has a sinusoidal tone and an energy detector can be used to detect the presence of the tone (and, consequently, the presence of TV signals). Then, $X[n;\mathcal{H}_1]$ can be expressed as

$$X[n;\mathcal{H}_1] = Ae^{j(2\pi f'_w nT+\theta_a)} + V[n] \tag{3.36}$$

We can interpret $X[n;\mathcal{H}_1]$ as a sinusoidal signal with frequency f_w that is *down-converted* to a complex baseband signal with frequency f'_w, amplitude A and phase θ_a. When A is fixed, the sinusoidal signal in $X[n;\mathcal{H}_1]$ has a fixed power for any phase. Hence, we can use Appendix 3.A and conclude that the energy detector on $X[n;\mathcal{H}_1]$ has a normal distribution with mean and variance parameters given by:

$$E[T_L[X];\mathcal{H}_1] = \sigma_v^2 + A^2 \tag{3.37a}$$

$$\text{var}[T_L[X];\mathcal{H}_1] = \frac{\sigma_v^2}{L}(\sigma_v^2 + 2A^2) \tag{3.37b}$$

The absence or presence of the sinusoidal tone is determined by performing the hypothesis test:

$$T_L[X] \lessgtr \gamma \tag{3.38}$$

where γ is a pre-defined threshold value. The resulting probabilities of false alarm and detection are given by:

$$P_{FA,\mathcal{H}_1} = Q\left(\frac{\gamma - \sigma_v^2}{\sqrt{\frac{1}{L}\sigma_v^4}}\right) \tag{3.39a}$$

$$P_{D,\mathcal{H}_1} = Q\left(\frac{\gamma - \sigma_v^2 - A^2}{\sqrt{\frac{1}{L}\sigma_v^2\left(\sigma_v^2 + 2A^2\right)}}\right) \tag{3.39b}$$

where the Q-function is defined as:

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp\left(-\frac{u^2}{2}\right) du \tag{3.40}$$

The parameter γ can be selected to enforce a desired probability of false alarm.

3.3.3 Sinusoidal tone in the presence of strong interfering tone from a neighboring band - $\mathcal{H}_1'$

Let us suppose that in addition to the scenario $\mathcal{H}_1$, there is a strong tone at $f_w + f_{\text{ref}}$. Hence, the spurious sidebands from the strong tone will appear as an interfering tone to the tone at f_w. We define this scenario as $\mathcal{H}_1'$. Then, $X[n;\mathcal{H}_1']$ is expressed as

$$\begin{aligned} X[n;\mathcal{H}_1'] &= Ae^{j(2\pi f_w' nT+\theta_a)} + Be^{j(2\pi f_w' nT+\theta_b)} + V[n] \\ &= (Ae^{j\theta_a} + Be^{j\theta_b})e^{j2\pi f_w' nT} + V[n] \end{aligned} \tag{3.41}$$

where

$$\|Ae^{j\theta_a} + Be^{j\theta_b}\|^2 = A^2 + B^2 + 2AB\cos(\theta_a - \theta_b) \tag{3.42}$$

We can interpret $X[n;\mathcal{H}_1']$ as the sum of two complex sinusoidal signals in Gaussian noise. The amplitude and phase of the interfering tone is B and θ_b, respectively. We observe that the magnitude of the resultant signal depends on A, B and the phase-differences $\theta_a - \theta_b$. Hence, we cannot use the same fixed power arguments we used before for $\mathcal{H}_1$ to conclude that the energy detector on $X[n;\mathcal{H}_1']$ has a normal distribution.

In general, the energy detector on $X[n;\mathcal{H}_1']$ does not have a normal distribution. Assuming that θ_a and θ_b are independent, we can model their phase-difference as uniformly distributed between 0 to 2π. Let the phase-difference be denoted by $\alpha = \theta_a - \theta_b$ and denote (3.42) by

$$P_1(\alpha) = A^2 + B^2 + 2AB\cos(\alpha) \tag{3.43}$$

When α is fixed, $P_1(\alpha)$ is fixed and the energy detector will have a normal dis-

tribution with mean and variance parameters given by:

$$E[T_L[X]; \mathcal{H}_1', \alpha] = \sigma_v^2 + P_1(\alpha)$$
$$= \mu_1(\alpha) \tag{3.44a}$$
$$\text{var}[T_L[X]; \mathcal{H}_1', \alpha] = \frac{\sigma_v^2}{L}(\sigma_v^2 + 2P_1(\alpha))$$
$$= (\sigma_1(\alpha))^2 \tag{3.44b}$$

Note that the mean and variance values are functions of α. Hence, the distribution of the energy detector can be expressed as a mixture of normal distribution:

$$f_{T_L[X]}(y) = \frac{1}{2\pi} \int_0^{2\pi} f_N\left(y; \mu_1(\alpha), (\sigma_1(\alpha))^2\right) \, d\alpha \tag{3.45}$$

where $f_N(.)$ is the normal distribution function:

$$f_N\left(y; \mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \tag{3.46}$$

Hence, we can derive the probabilities of false alarm (in the presence of noise and interference) and detection using the Q-function as follows:

$$P_{FA,\mathcal{H}_1'} = Q\left(\frac{\gamma - \sigma_v^2 - B^2}{\sqrt{\frac{1}{L}\sigma_v^2\left(\sigma_v^2 + 2B^2\right)}}\right) \tag{3.47a}$$
$$P_{D,\mathcal{H}_1'} = \int_\gamma^\infty f_{T_L[X]}(y) dy$$
$$= \int_\gamma^\infty \frac{1}{2\pi} \int_0^{2\pi} f_N\left(y; \mu_1(\alpha), (\sigma_1(\alpha))^2\right) \, d\alpha \, dy$$
$$= \frac{1}{2\pi} \int_0^{2\pi} \int_\gamma^\infty f_N\left(y; \mu_1(\alpha), (\sigma_1(\alpha))^2\right) \, dy \, d\alpha$$
$$= \frac{1}{2\pi} \int_0^{2\pi} Q\left(\frac{\gamma - \sigma_v^2 - P_1(\alpha)}{\sqrt{\frac{1}{L}\sigma_v^2\left(\sigma_v^2 + 2P_1(\alpha)\right)}}\right) \, d\alpha \tag{3.47b}$$

where γ is a threshold value. $P_{FA,\mathcal{H}_1'}$ occurs when only the interfering tone and noise is present. Hence, the distribution of the detector in the absence of the

desired signal is normal. $P_{D,\mathcal{H}_1'}$ follows from (3.45) and is obtained by integrating (3.45) from γ to ∞. The $P_{D,\mathcal{H}_1'}$ can be approximated by discretization:

$$P_{D,\mathcal{H}_1'} \approx \frac{1}{N}\sum_{k=0}^{N-1} Q\left(\frac{\gamma - \sigma_v^2 - P_1(\frac{2\pi k}{N})}{\sqrt{\frac{1}{L}\sigma_v^2\left(\sigma_v^2 + 2P_1(\frac{2\pi k}{N})\right)}}\right) \tag{3.48}$$

3.3.4 Unknown white signals - $\mathcal{H}_2$ and $\mathcal{H}_2'$

Let us suppose the signal of interest is an unknown zero-mean signal and only the signal power is known. Also, consider the case when the primary user of the spectrum has complete freedom to choose its waveforms (as long as it satisfies its power and bandwidth constraints) and the spectrum sensing detector does not know what waveforms the primary user is going to transmit. A challenging signal to be detected is a zero-mean white signal in the frequency band of interest [47]. We define this scenario as $\mathcal{H}_2$. Then, $X[n]$ can be expressed as

$$X[n;\mathcal{H}_2] = W_a[n] + V[n] \tag{3.49}$$

where the signal is $W_a[n]$; it is white with average power σ_{wa}^2. The distribution of the energy detector in $\mathcal{H}_2$ can be approximated as a normal distribution with mean and variance:

$$E[T_L[X];\mathcal{H}_2] = \sigma_v^2 + \sigma_{wa}^2 \tag{3.50a}$$

$$\text{var}[T_L[X];\mathcal{H}_2] = \frac{1}{L}(\sigma_v^2 + \sigma_{wa}^2)^2 \tag{3.50b}$$

Hence, we can derive the probability of false alarm and the probability of detection using the Q-function:

$$P_{FA,\mathcal{H}_2} = Q\left(\frac{\gamma - \sigma_v^2}{\sqrt{\frac{1}{L}\sigma_v^4}}\right) \tag{3.51a}$$

$$P_{D,\mathcal{H}_2} = Q\left(\frac{\gamma - \sigma_v^2 - \sigma_{wa}^2}{\sqrt{\frac{1}{L}(\sigma_v^2 + \sigma_{wa}^2)^2}}\right) \tag{3.51b}$$

where γ is a threshold value.

In the case of an interfering white signal (denoted as $\mathcal{H}_2'$), if the signals are independent, then the energy detector will also have a normal distribution and its statistical properties will be similar to (3.50) and (3.51) except that σ_v^2 is replaced by the power of the noise and interference. Define the power of the interference as $\sigma_{wa'}^2$.

$$E[T_L[X]; \mathcal{H}_2'] = \sigma_v^2 + \sigma_{wa'}^2 + \sigma_{wa}^2 \tag{3.52a}$$

$$\text{var}[T_L[X]; \mathcal{H}_2'] = \frac{1}{L}(\sigma_v^2 + \sigma_{wa'}^2 + \sigma_{wa}^2)^2 \tag{3.52b}$$

Hence, we can derive the probability of false alarm and the probability of detection using the Q-function:

$$P_{FA,\mathcal{H}_2'} = Q\left(\frac{\gamma - \sigma_v^2 - \sigma_{wa'}^2}{\sqrt{\frac{1}{L}(\sigma_v^2 + \sigma_{wa'}^2)^2}}\right) \tag{3.53a}$$

$$P_{D,\mathcal{H}_2'} = Q\left(\frac{\gamma - \sigma_v^2 - \sigma_{wa'}^2 - \sigma_{wa}^2}{\sqrt{\frac{1}{L}(\sigma_v^2 + \sigma_{wa'}^2 + \sigma_{wa}^2)^2}}\right) \tag{3.53b}$$

where γ is a threshold value.

3.4 Simulations

The previous sections discuss how the distorted ADC samples affect spectrum sensing and propose a solution to estimate and correct the samples before spectrum sensing. In the next subsection, we first simulate the proposed solution to verify that spurious sidebands can be removed. Next, we setup a simulation where we simulate the different scenarios ($\mathcal{H}_1$, $\mathcal{H}_1'$, $\mathcal{H}_2$ and $\mathcal{H}_2'$) as described in Section 3.3. We verify that the simulation results match well with the analytical expressions. The simulations also show that the proposed solution is able to remove the spurious sidebands and improve the detection performance.

3.4.1 Sideband suppression

The simulation parameters are as follows. The sampling frequency f_s is 500 MHz and the reference leakage in the PLL is assumed to be a triangular wave with fundamental frequency f_{ref} of 20 MHz. C_0 is set to 6.32×10^{-3} so that the power ratio of the spurious sideband at $f_s \pm f_{\text{ref}}$ to the signal at f_s is -50 dB. The Fourier series representation of the triangular wave is truncated to the first 4 coefficients. Since the coefficients decrease rapidly, the effects of the 3rd and above coefficients are not observable in both the training signal in Fig. 3.2 and even when the input signal frequency is high (see Fig. 3.11 ahead). The frequency of the training tone, f_w, is 45 MHz. We first estimate the spurious sidebands that are 20 MHz and 60 MHz from the distorted training signal. Then, we switch to sample the input signal and compensate the sampled data to obtain the dejittered samples. The input signal is simulated as a sinusoidal tone ranging from 25 MHz to 200 MHz in steps of 25 MHz and the distortion suppression performance are averaged over 50 runs. The amplitude of the training signal and the input signal is set to 0.9 and white Gaussian noise with standard deviation of 1×10^{-3} is added to the input of the ADC. Finally, the STFT uses a 1024-pt FFT with a Blackman-Harris windowing function and the length of the data $d_p[m]$, M, is chosen from the set $\{64, 128, 256, 512, 1024\}$.

Figure 3.11 shows a realization where the desired input signal's frequency is 200 MHz. The left plot shows the PSD of the signal before compensation and the right plot shows the result using the proposed method. It can be seen that the spurious sidebands are reduced by 11 to 34 dB.

Figure 3.12 shows the spurious sideband suppression performance using the proposed solution. From Fig. 3.11, the PLL sideband induces sidebands that are 20 MHz and 60 Mhz away from the input tone. The left plot in Fig. 3.12

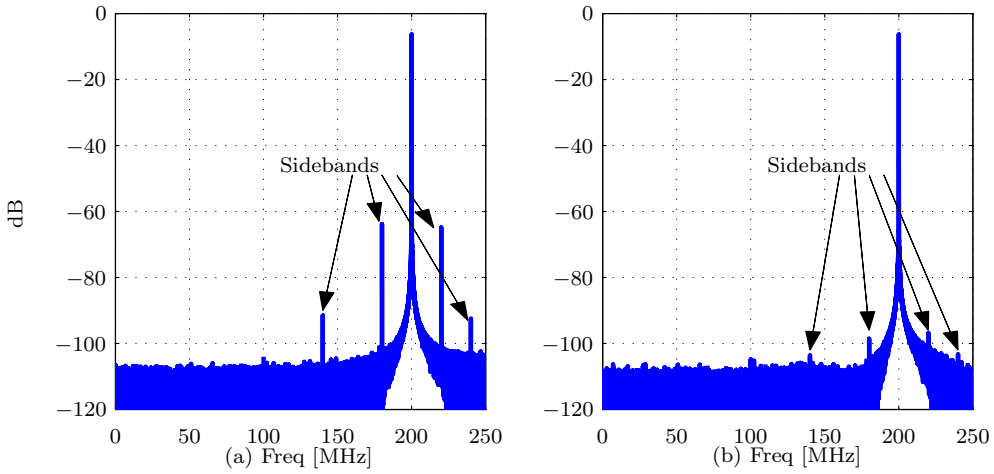


Figure 3.11: The plot shows the PSD of a 200 MHz input signal (a) before and (b) after compensation.

shows the the reduction of the sideband power at 20 MHz and 60 MHz away from the input signal using M=1024 samples of $d_p[m]$. The right plot shows the average sideband performance when M is varied. The simulations show that when M=1024, the algorithm reduces the sideband distortions at 20 MHz and 60 MHz from the input tone by an average of 35 dB and 8 dB respectively.

3.4.2 Detection performance - $\mathcal{H}_1$ and $\mathcal{H}_1'$

To analyze the detection performance, the following simulation is done. Recall from Fig. 3.3 that the training signal is fed directly into the ADC. Hence, the noise during the training phase is small. In the simulation, we used the same noise and training signal parameters as the previous simulation. Also, recall that the TFT algorithm uses M samples of $d_p[m]$ to estimate the parameters. In the simulation, we fix M to 448, which corresponds to 0.91 ms of integration time. After estimating the distortion parameters, the input signal is compensated

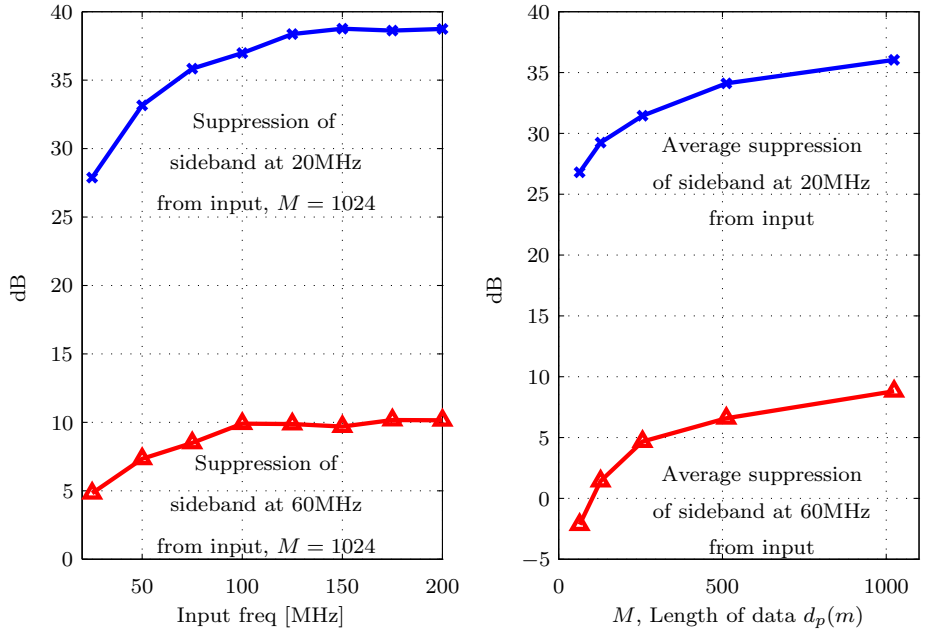


Figure 3.12: The plots show the sideband suppression performance at 20 MHz and 60 MHz away from the input frequency signal and when M is varied.

before spectrum sensing.

During spectrum sensing, we assumed there are strong and weak tones in the presence of additional noise (e.g., receiver noise). We assume the signals lie in some frequency bins of the STFT. We set the signal-to-noise ratio (SNR) of the strong signal and weak signals (in the frequency bin) to 40 dB and -20 dB, respectively. We also set the strong and weak signals at $f_{\text{ref}} = 20$ MHz apart in the frequency domain. Let us denote the frequency of the strong and weak signal by f_a and f_b, respectively. Then, we can relate the frequencies as $f_b = f_a + f_{\text{ref}}$. For example, suppose f_a is 100 MHz, then f_b is 120 MHZ. Due to the distortions in the ADC, spurious sidebands from the strong signal are created in the sampled data. Although the strong and weak signals do not overlap in the frequency domain, the most dominant spurious sideband does overlap with the weak signal. From (3.18), we know that the amplitude of the spurious sidebands

increases proportionally to frequency. Using the same distortion parameters (C_0 and f_s) as in the previous simulation, and supposing f_a is 100 MHZ, 150 MHz or 200 MHz, the SNR of the dominant spurious sideband is -24 dB, -20 dB or -18 dB, respectively. An energy detector (3.33) is used to detect the weak signal and the performance before and after using the proposed solution is analyzed. In the simulation, we fix the length L in the energy detector to 22400. This corresponds to 45.9 ms of sensing time. The simulation results are averaged over 300 runs.

Figure 3.13 shows the detection performance in terms of P_D and P_{FA} when $f_a = \{100, 150, 200\}$ MHz. The legend is the same for all the plots. We can make 2 observations from the receiver operating characteristics (ROC) curve. First, the plots show that the theoretical and experimental performance before compensation ($\mathcal{H}_1'$) match well. Also, the ROC curve before compensation changes for different f_a. This is due to the fact that the SNR of the dominant spurious sideband increases with f_a and, hence, affects the performance. Note that in the ideal case ($\mathcal{H}_1$), the curve is independent of f_a. Second, the proposed solution improves the detection performance and approaches the ideal case.

We note that there is a crossover point at the lower P_{FA}, which is dependent on the sensing time. When we increase the sensing time, the ROC curve of the undistorted case ($\mathcal{H}_1$) will shift upwards and the crossover point will move further to the left. For example, see Fig. 3.19 of Section 3.C where we double the sensing time and plot the ROC curve.

3.4.3 Detection performance - $\mathcal{H}_2$ and $\mathcal{H}_2'$

The simulation setup is similar to the previous section using sinusoidal tones. The main difference is that the tones are replaced by QPSK signals. In this case, the frequencies f_a and f_b represent the carrier frequency of the signals. Similarly,

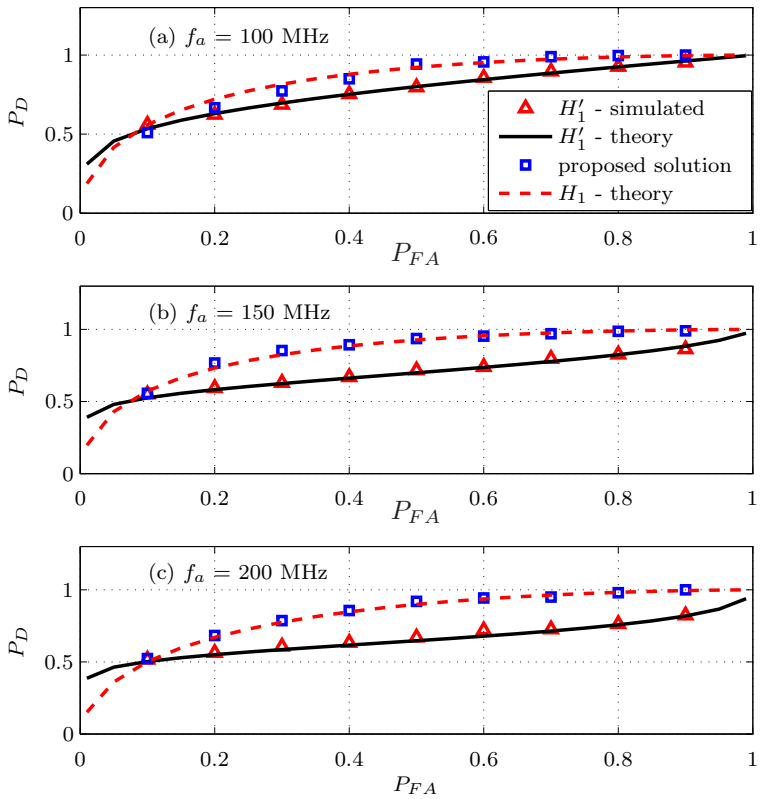


Figure 3.13: The plots show the P_D vs P_{FA} of the weak signal when f_a is (a) 100 MHz, (b) 150 MHz or (c) 200 MHz. The plots show the theoretical and simulated performance before ($\mathcal{H}_1'$) and after using the proposed solution. The ideal solution is $\mathcal{H}_1$. The legend is the same for all plots.

we first compare detection performance in terms of P_D and P_{FA}. Figure 3.14 shows the ROC curve before and after compensation. In the simulation settings, the noise power dominates over both the desired signal and the interfering signal. As such, the ROC curve is almost the same for all cases. The benefits of the proposed solution is revealed when we examine the threshold γ used in the P_D and P_{FA} for the different cases.

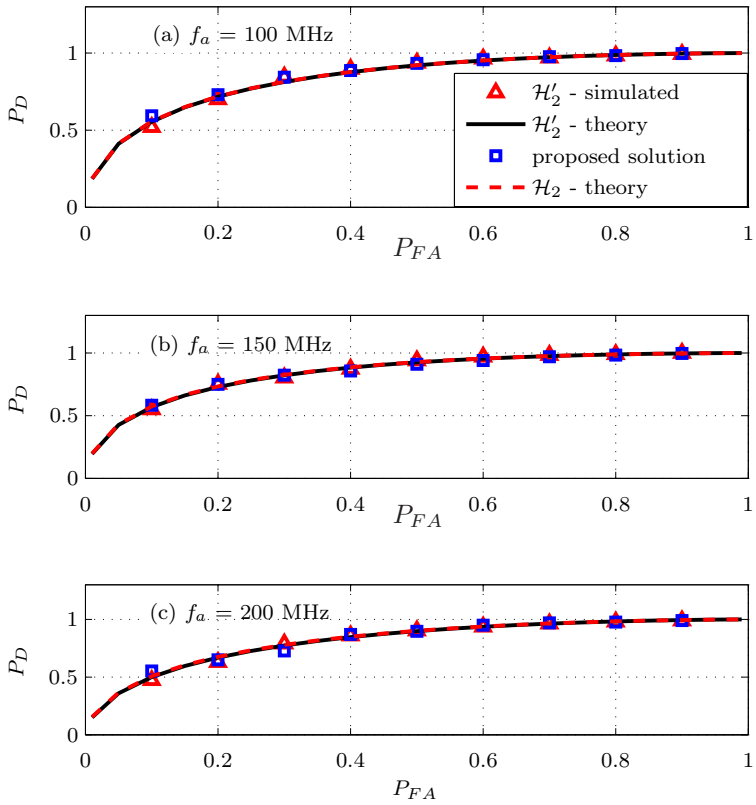


Figure 3.14: The plots show the P_D vs P_{FA} of the weak signal when the carrier frequency f_a is (a) 100 MHz, (b) 150 MHz or (c) 200 MHz. The legend is the same for all plots.

3.4.4 Impact of threshold during sensing

We can derive the threshold γ that corresponds to different P_D and P_{FA} through analysis (using the P_{FA} and P_D expressions) or through simulations. Here, we generate the plots using the simulated results from the previous sections. The threshold is normalized to the noise power.

Recall that the frequency of the strong signal is selected to be 100 MHz, 150 MHz or 200 MHz. Before using the proposed solution, for a desired constant P_D or P_{FA}, the threshold must vary with the frequency. This can be seen in Fig. 3.15 and Fig. 3.16, which represent the case for sinusoidal tone $\mathcal{H}_1$ and QPSK

signal $\mathcal{H}_2$, respectively.

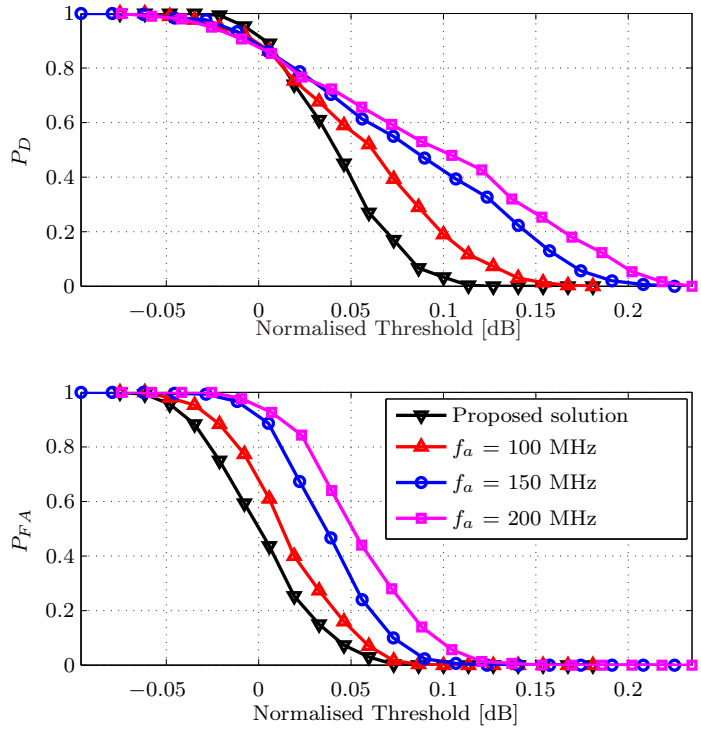


Figure 3.15: The plots show the normalized threshold used to obtain the various pairs of P_D amd P_{FA} for sinusoidal signals $\mathcal{H}_1$.

The curves denoted by f_a show the required threshold when the strong signal's frequency is f_a. The black line with inverted triangle markers shows the threshold required after using the proposed solution. Note that after compensation, the required threshold for a fixed detection performance is constant and independent of f_a. Hence, the proposed solution reduces the complexity of finding an adaptive threshold. Moreover, even when an adaptive threshold is used to obtain a fixed P_{FA}, the resultant P_D may degrade significantly (for example, in $\mathcal{H}_1$).

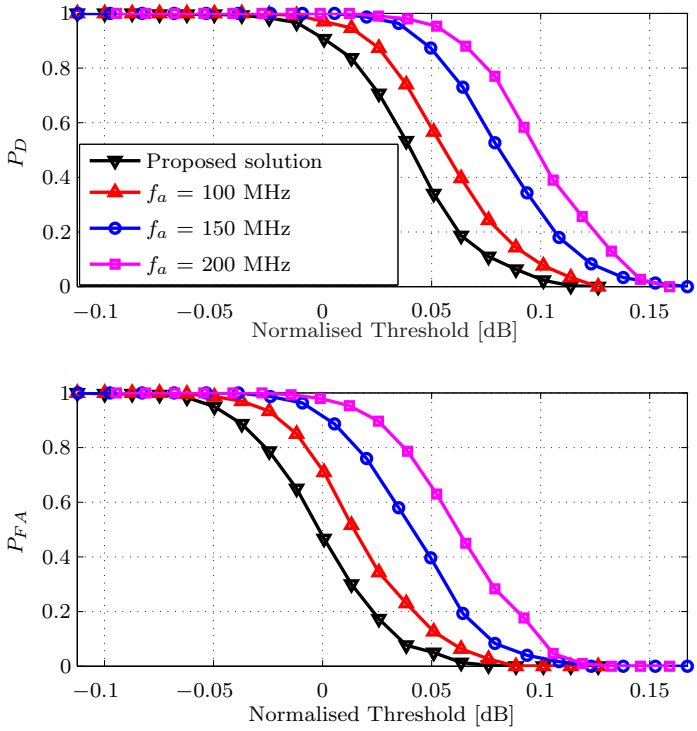


Figure 3.16: The plots show the normalized threshold used to obtain the various pairs of P_D amd P_{FA} for QPSK signals $\mathcal{H}_2$.

3.5 Conclusion

In this chapter, we showed how spurious tones are created in the distorted ADC samples. We also described a way to use the STFT block to estimate the distortions and then compensate the samples. Simulations show that the sideband tones can be estimated and suppressed using the proposed algorithm. The simulation results show that by increasing the data length, the suppression can be as much as 35dB. The chapter also investigated the effects of spurious sidebands on the detection performance in spectrum sensing. Simulations were performed to analyze the detection performance and theoretical results are derived for verifications. The results show that compensating for the presence of the spurious sidebands improves the detection performance in spectrum sensing in a tangible

way.

3.A Energy detector for signals with fixed power

This section shows that when the energy detector is used on signals with a fixed power, the statistics of the detector can be modeled with a normal distribution. The first part defines a sequence where each element is a circular complex normal random variables; all elements are independent and have the same variance.

3.A.1 Noncentral chi-square for circular complex variables

Let $z[n] = z_R[n] + jz_I[n]$ be independent and circular complex normal random variables. We assume that the variance of $z[n]$ is the same for all n. That is,

$$z[n] \sim CN(\mu_z[n], \sigma_z^2) \tag{3.54}$$

with $\mu_z[n] = \mu_{zR}[n] + j\mu_{zI}[n]$ and $\sigma_z^2 = \sigma_{zR}^2 + \sigma_{zI}^2$, where $\sigma_{zR}^2 = \sigma_{zI}^2 = \frac{1}{2}\sigma_z^2$. Define

$$\begin{aligned} G_L(z) &\triangleq \frac{2}{\sigma^2} \sum_{i=0}^{L-1} |z[i]|^2 \\ &= \sum_{i=0}^{L-1} \left(\frac{\sqrt{2}}{\sigma} z_R[i] \right)^2 + \sum_{i=0}^{L-1} \left(\frac{\sqrt{2}}{\sigma} z_I[i] \right)^2 \end{aligned} \tag{3.55}$$

and

$$\begin{aligned} \lambda &\triangleq \frac{2}{\sigma_z^2} \sum_{i=0}^{L-1} |\mu_z[i]|^2 \\ &= \sum_{i=0}^{L-1} \left(\frac{\sqrt{2}}{\sigma_z} \mu_{zR}[i] \right)^2 + \sum_{i=0}^{L-1} \left(\frac{\sqrt{2}}{\sigma_z} \mu_{zI}[i] \right)^2 \end{aligned} \tag{3.56}$$

where $\frac{\sqrt{2}}{\sigma_z} z_R[i] \sim N\left(\frac{\sqrt{2}}{\sigma_z}\mu_{zR}[i], 1\right)$ and $\frac{\sqrt{2}}{\sigma_z} z_I[i] \sim N\left(\frac{\sqrt{2}}{\sigma_z}\mu_{zI}[i], 1\right)$. From [60], the mean and variance of $G_L(z)$ are

$$E[G_L(z)] = 2L + \lambda \tag{3.57a}$$

$$\text{var}[G_L(z)] = 4(L + \lambda) \tag{3.57b}$$

When L is large, $G_L(z)$ can be approximated as a normal distribution.

3.A.2 Energy detector for signals with a fixed power

We define a class of deterministic signals where each member has the same power, i.e.

$$P_{zs} = \frac{1}{L}\sum_{i=0}^{L-1} |z_s[i]|^2 \tag{3.58}$$

Suppose an energy detector is used to detect this class of signals in the presence of circular complex white Gaussian noise:

$$z[n] = z_s[n] + z_v[n] \tag{3.59}$$

where $z_s[n]$ is a signal in the above class and $z_v[n] \sim CN(0, \sigma_{zv}^2)$ is noise. Then, we see that

$$z[n] \sim CN(z_s[n], \sigma_{zv}^2) \tag{3.60}$$

Also, note that $z[n]$ has the same form as (3.54) when $z_s[n] = \mu_z[n]$ and $\sigma_{zv} = \sigma_z$. Using an energy detector on $z[n]$ and (3.55) yields

$$\begin{aligned} T_L(z) &= \frac{1}{L}\sum_{i=0}^{L-1} |z[i]|^2 \\ &= \frac{\sigma_{zv}^2}{2L} G_L(z) \end{aligned} \tag{3.61}$$

When L is large, $T_L(z)$ has a normal distribution. Using (3.56) to (3.58), the mean and variance are:

$$E[T_L(z)] = \sigma_{zv}^2 + P_{zs} \tag{3.62a}$$

$$\begin{aligned} \text{var}[T_L(z)] &= \frac{\sigma_{zv}^2}{L}(\sigma_{zv}^2 + 2P_{zs}) \\ &= \frac{1}{L}[(\sigma_{zv}^2 + P_{zs})^2 - P_{zs}^2] \end{aligned} \tag{3.62b}$$

3.B Comparison of various compensation schemes

In this work, the proposed signal compensation algorithm, FOT, is based on a first-order Taylor series approximation (3.32). Other useful compensation techniques have been proposed [57–59]. This section compares the performance of the proposed algorithm with these other algorithms. The evaluation is done by comparing the suppression performance of the spurious sidebands as described in Section 3.4.1.

Reference [57] proposed the DMC structure based on a Taylor series expression. Using Taylor series, it shows that the distorted signal can be represented as the true signal and higher order error terms. The proposed DMC has 3 stages and removes up to the third-order errors. It is noted that when the derivative filter is applied on the distorted signal, additional error components are created and the DMC solution removes them as well. The first stage of the DMC has the same form as our proposed FOT solution in Fig. 3.9. The second and third stages, however, use more derivative filters, multipliers and adders. It should be noted that all the derivative filters have the same filter coefficients. In this simulation, we implement the DMC using the same filter coefficients we used in our proposed solution.

Reference [58] modifies the conventional Lagrange interpolator to handle bandlimited signals. The interpolator is called the functionally weighted (FW) interpolator and it is of the form

$$s_B(t) \approx \frac{1}{\gamma(t)} \sum_{k=1}^{N} s_B(t'_k)\gamma(t'_k)\frac{L(t)}{L'(t'_k)(t-t'_k)} \tag{3.63}$$

where $s_B(t'_k)$ and t'_k are given. The FW interpolator assumes that the signal $s_B(t)$ is bandlimited and its frequency spectrum lies in $[-B/2, B/2]$. The algorithm also assumes that $t'_k = (p+\delta_p)T_s$ and $|\delta_p| < 1/2$, where $p = -P + k - 1$ and $N = 2P+1$. Therefore, the true signal value at $t = 0$ is estimated as $s_B(0)$. The rest of the functions in (3.63) are defined as

$$\gamma(t) = \frac{w_{ap}(t)L_o(t)}{\sin(\pi t/T)} \tag{3.64a}$$

$$L(t) = \prod_{k=1}^{N} t - t'_k \tag{3.64b}$$

$$L'(t'_k) = \left.\frac{dL(t)}{dt}\right|_{t=t'_k} \tag{3.64c}$$

where

$$w_{ap}(t) = \frac{sinc\left(B_w\sqrt{t^2 - T_w^2}\right)}{sinc(jB_wT_w)} \tag{3.65a}$$

$$L_o(t) = \prod_{p=-P}^{P} t - pT_s \tag{3.65b}$$

and

$$B_w = \frac{1}{T_s} - B \tag{3.66a}$$

$$T_w = T_s(P+1) \tag{3.66b}$$

Hence, given a set of N distorted sampled data $\{\breve{w}[n], \cdots, \breve{w}[n+N-1]\}$ and time instants $\{t_n, \cdots, t_{n+N-1}\}$, we can estimate $w((n-P)T_s)$ (denoted as $\hat{w}[n-P]$)

by defining

$$t'_k = t_{n+k-1} - (n+P)T_s \tag{3.67a}$$

$$s_B(t'_k) = \check{w}[n+k-1] \tag{3.67b}$$

and, therefore,

$$\hat{w}[n-P] = s_B(0) \tag{3.68}$$

In the simulation, we set $B = 0.8/T_s$ and $P = 12$. It is noted that the functions in the FW interpolator includes evaluations of $\gamma(t)$ and $L'(t_k)$. In [58], there are details on how to reduce the complexity of evaluating these functions.

In our first simulation, we use the same settings as described in Section 3.4.1, with two differences. Firstly, we assume that the distorted sampling offsets are known. Secondly, we fix the frequency of the input sinusoidal signal to 200 MHz. We assume that the distorted sampling offsets are known because we are only interested in comparing the various signal compensation algorithm. Also, we fix the frequency of the input signal to 200 MHz because we know that the magnitude of the spurious sidebands increases with the input frequency, and hence it is easier to compare the suppression performance of the various techniques. In this experiment, we use the proposed FOT algorithm, the DMC algorithm and FW algorithm.

Figure 3.17 shows a realization of the distorted signal's spectrum before and after using the 3 compensation algorithms. From the figure, we see that the distorted signal contains some spurious sidebands around its input frequency (200 MHz). The figure also shows that the 3 algorithms are able to remove the spurious sidebands. The average suppression of the sidebands at 20 MHz away from the input frequency for FOT, DMC and FW are 40 dB, 40 dB and 42 dB, respectively. This shows that under the current simulation settings, our proposed

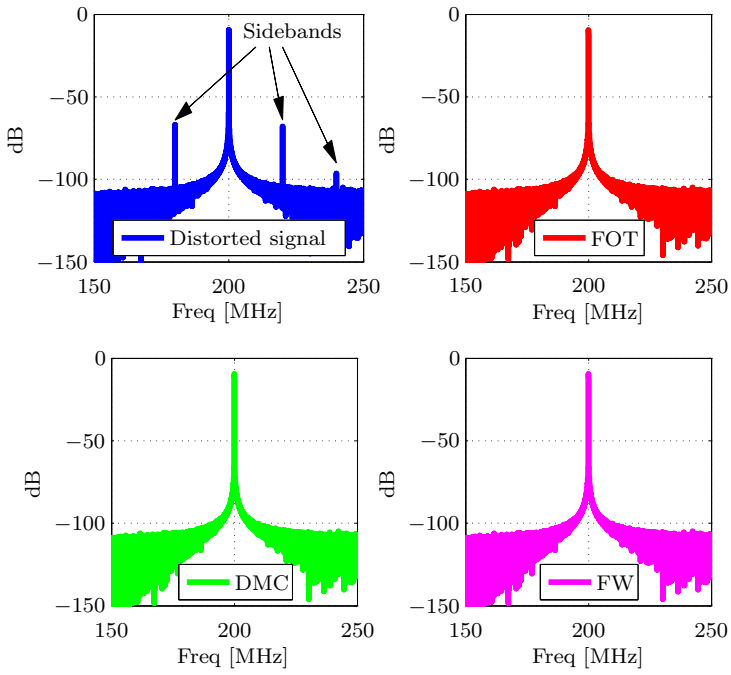


Figure 3.17: The plots show the spectrum of the distorted signal and the compensated signal using the various methods. The distorted signal is generated based on the same settings in Section 3.4.1. FOT, DMC and FW denote the resulting spectrum after using the proposed compensation method, [57] and [58], respectively. In these simulations, the distortion sampling offsets are at most 0.1% of T_s.

compensation algorithm works as well as the techniques in [57,58]; albeit at much lower computational complexity.

The reason why the advanced techniques do not have a more significant impact on performance than the simpler proposed architecture, is because of the magnitude of the jitter/sampling offsets. Under the current simulation settings, the distorted sampling offsets are at most 0.1% of the sampling period (0.1% of T_s). In the next simulations, we increase the distorted sampling offsets to 1% of T_s and repeat the experiment. A realization of the new simulation is shown in Fig. 3.18. The top-left plot shows the frequency spectrum of the distorted signal

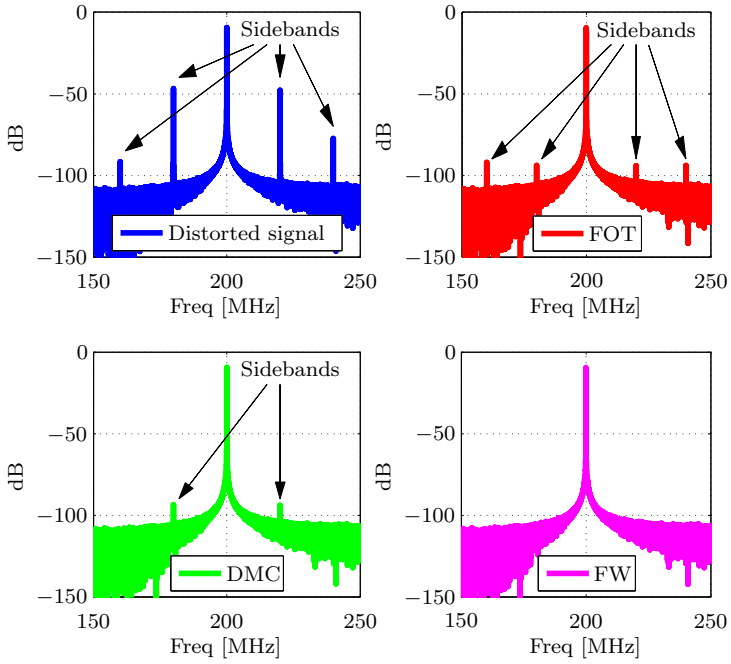


Figure 3.18: The plots are created using the same settings as the settings in Fig. 3.17, except that the sampling offsets is 1 order of magnitude larger (i.e. up to 1% of T_s).

before compensation. It can be seen that increasing the sampling offsets creates additional spurious sidebands. These additional spurious sidebands come from the higher order error terms in the distorted signal. In this case, we find that the techniques in [57, 58] can be helpful. The top-right plot of Fig. 3.18 shows the spectrum of the signal after using our proposed FOT solution. At shown, the spurious sidebands are not removed completely. The bottom plots in Fig. 3.18 show that the techniques in [57, 58] remove more spurious sidebands. Therefore, the simulations show that if the sampling offsets increase, then one can use more advanced techniques to remove the spurious sidebands.

3.C Increasing sensing time

The below ROC curve is obtained by doubling the sensing time (L) used in Fig. 3.13. It illustrates that increasing L shift the crossover point further to the left.

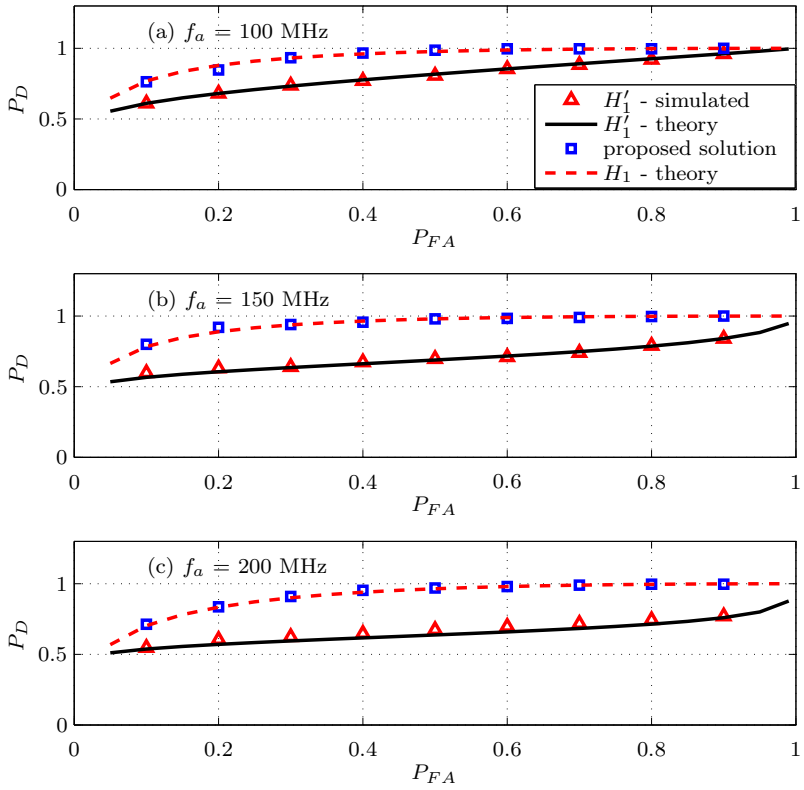


Figure 3.19: The plots show the P_D vs P_{FA} of the weak signal when f_a is (a) 100 MHz, (b) 150 MHz or (c) 200 MHz. The plots show the theoretical and simulated performance before ($\mathcal{H}_1'$) and after using the proposed solution.

CHAPTER 4

Clock Jitter Compensation in High-Rate ADC Circuits

Certain applications of modern radios, such as cognitive radios and UWB radios, require a high signal-to-noise ratio (SNR) in order to detect the presence of the desired signal from noisy measurements. Unfortunately, signals with high-frequency content, either due to high bandwidth or high carrier frequencies, are very sensitive to jitter at the ADC. Clock jitters arise from noise in the clock generating crystal and phase-locked-loop (PLL). The jitters cause the ADC to sample the input signal at non-uniform sampling times and introduce distortion that limits the signal fidelity and degrades the SNR [62]. While the effects of jitter noise can be neglected at low frequencies, applications requiring enhanced performance at higher frequencies [32, 33, 41, 63, 64] demand higher SNR from the sampling circuit.

There has already been considerable work in the literature on modeling clock jitter and understanding its effects [9, 62, 65, 66]. Most models assume the jitter is Gaussian-distributed with zero mean and with a standard-deviation that is represented as a percentage of the sampling interval [67]. The models also assume that clock-jitter (as opposed to aperture jitter) is correlated. This is because the PLL incorporates a low-pass filter that correlates the jitter. Expressions for the power-spectral-density (PSD) of the phase-noise output by the PLL appear

in [9, 62].

Various approaches have been proposed to treat sampling-clock jitters. Some approaches interleave several ADCs in order to produce an effective higher sampling rate. This technique, however, introduces other challenges such as mismatches in the delays of the clocks fed into the ADCs, mismatches in the gain of each ADC, and the DC offset between the ADCs (we discuss some of these problems later in Chapter 5). The works [14–16, 68–70] propose techniques to address such mismatches in order to allow time-interleaved ADCs to serve reliably as high-speed ADCs. In [57, 71], alternative approaches for signal compensation are presented; these approaches, however, assume full knowledge of the jitter itself. We do not make this assumption in this work and assume the system must estimate and compensate for the jitter and its effects. Other approaches [72] transform the signal into the wavelet domain and use linear least-mean-squares estimation (LLMSE) techniques to recover the original signal. This approach assumes that the signal has small support in the wavelet domain, which simplifies the estimation of the covariance matrix used in the LLMSE step to a diagonal representation. This approach, however, is computationally intensive to implement—especially at high sampling rates.

Some recent work has focused on the recovery of the jitter through the injection of training tones and on the subsequent compensation of the distorted data [32, 41, 64, 73]. The schemes proposed in [32, 73] assumed that the input signal is narrow-band, which is an impractical assumption for ultra-wide-band and cognitive radio applications. Here, in section 4.1, we first examine the effect of the clock time jitter on the SNR of the sampled signal [19]. Subsequently, in sections 4.2 and 4.3, we propose compensation methods based on a signal injection architecture for direct downconversion architectures. Next, in sections 4.4

and 4.5, we analyze the performance of estimation architectures such as those studied in [32, 41, 64, 73] and of the proposed schemes in some detail. Finally, section 4.6 shows our simulations results.

In this chapter, we will be using capital letters to denote matrices, small letters to denote vectors and scalars. We will also use boldface notation to denote random quantities and normal font to denote deterministic (non-random) quantities. We will use the notation $x(t)$ and $x[n] \triangleq x(nT_s)$ to denote, respectively, a continuous-time signal and a discrete-time signal sampled with period T_s. A complex signal $x(t)$ with in-phase and quadrature-phase components $x_I(t)$ and $x_Q(t)$, respectively, is modeled with the equivalent notation $x(t) \triangleq x_I(t) + jx_Q(t)$ where $j \triangleq \sqrt{-1}$ is the imaginary unit.

4.1 Problem formulation

The key challenge in this work is to compensate for clock jitters in ADCs with already small jitters. As explained in the introduction, jitter errors arise when the input signal is not sampled exactly at multiples of the sampling period, T_s, but rather at perturbed versions, say, at time instants $nT_s + \boldsymbol{e}(nT_s)$, where $\boldsymbol{e}(nT_s)$ denotes the amount of random jitter occurring at nT_s. An ADC with a clock jitter of 1% means that the standard-deviation of the jitter $\boldsymbol{e}[n]$ is equal to $0.01T_s$ (1% of T_s). Although apparently small, this amount of jitter is problematic when the sampled signal contains high frequency components due to either having wide bandwidth or a high carrier frequency; in this work, we concentrate on the case where the ADC operates at 1GHz in order to sample a wide-bandwidth signal that resides at baseband. Our objective is to propose filtering and estimation techniques that would compensate for the effect of the jitter in the sampling clock and reduce it to the equivalent of operation at about 0.5% jitter for a 10-

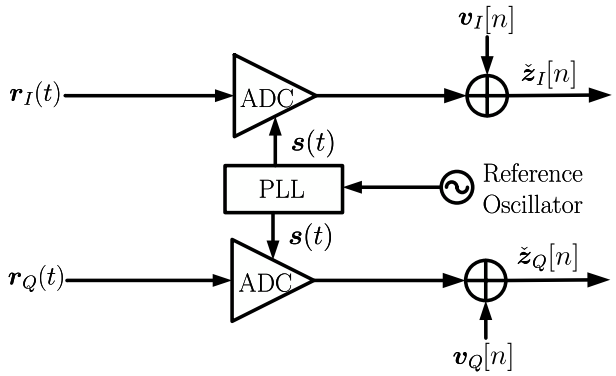


Figure 4.1: The in-phase and quadrature phase components of a random complex-valued signal $\boldsymbol{r}(t)$ are sampled by a pair of ADCs triggered by a clock signal generated by a phase-locked-loop. The complex-valued random noise $\boldsymbol{v}[n]$ represents additive noise sources such as quantization noise.

bit ADC—see Fig. 4.3 further ahead, which illustrates the benefit of such jitter reduction on the output SNR. We accomplish this task by devising processing algorithms in the discrete-time domain, rather than focusing on perfecting the hardware. The latter option is usually costly and effective DSP techniques are desirable alternatives. We take this route in this chapter. We consider the general radio architecture described in Fig. 4.1, where the in-phase and quadrature-phase components of the down-converted input signal $\boldsymbol{r}(t)$ are denoted by $\boldsymbol{r}_I(t)$ and $\boldsymbol{r}_Q(t)$, respectively. These signals are sampled by two ADCs triggered at the rising edge of the same jittered sampling clock, $\boldsymbol{s}(t)$, generated by an analog phase-locked-loop. Before discussing compensation methods, we first discuss the cause of the jitter noise as well as the jitter's stochastic properties. Following this presentation, we will examine the effects of the jitter noise on the signal-to-noise-ratio of the sampled signals and then describe our proposed estimation methods.

4.1.1 Stochastic properties of the clock jitter

We start by describing the statistical properties of the clock jitter. To do so, we comment on the connection between the PLL phase-noise and the clock jitter. Once this connection is defined, and since the statistical properties of PLL phase noise are well understood and documented in the literature, we can then arrive at useful expressions for the power spectral density (PSD) of the clock jitter. This PSD will be helpful in designing estimation and compensation methods for the jitter.

We model the sampling-clock signal that is generated by the PLL as a sinusoidal signal with frequency f_s, say,

$$\boldsymbol{s}(t) = \sin(2\pi f_s t + \boldsymbol{\phi}_s(t) + \theta_s) \tag{4.1}$$

In (4.1), the term $\boldsymbol{\phi}_s(t)$ denotes the phase-noise that corrupts the sampling clock, and θ_s is some deterministic phase-offset. We model $\boldsymbol{\phi}_s(t)$ as a zero-mean random process, If the phase-noise is not zero-mean, it is possible to rewrite (4.1) such that the new phase-noise is zero-mean for a different θ_s value. Modeling the phase-noise in this manner is consistent with phase-noise models used in conjunction with second-order PLL models [74]. We are able to simplify the expression for $\boldsymbol{s}(t)$ by changing the time reference. To this end, consider setting the time variable in (4.1) to $t = t' - \frac{\theta_s}{2\pi f_s}$. In this case, the equations are similar except that $\theta_s = 0$. For this reason, and without loss of generality, we set $\theta_s = 0$.

We now let the discrete-time random process $\boldsymbol{e}[n]$ denote the jitter that perturbs the sampling instants. We denote the actual sampling time instants by $\boldsymbol{t}_n$ and, therefore, write

$$\boldsymbol{t}_n = nT_s + \boldsymbol{e}[n] \tag{4.2}$$

When the phase-noise process $\boldsymbol{\phi}_s(t)$ is small and varies slowly relative to the sampling frequency, it can be verified that the jitter random process $\boldsymbol{e}[n]$ is related to the phase-noise process $\boldsymbol{\phi}_s(t)$ as follows (see App. 4.A):

$$\boldsymbol{e}[n] \approx -\frac{\boldsymbol{\phi}_s(nT_s)}{2\pi f_s} \tag{4.3}$$

Assume the phase-noise has an effective bandwidth of $2\pi f_e$ radians/second. Then, it can be shown that the power spectral density (PSD) of the jitter can be approximated by (see App. 4.A and [75])

$$S_e(e^{j\omega}) \approx \frac{1}{(2\pi f_s)^2 T_s} \sum_{k=-\infty}^{\infty} S_\phi\left(\frac{\omega - 2\pi k}{2\pi T_s}\right) \tag{4.4}$$

where

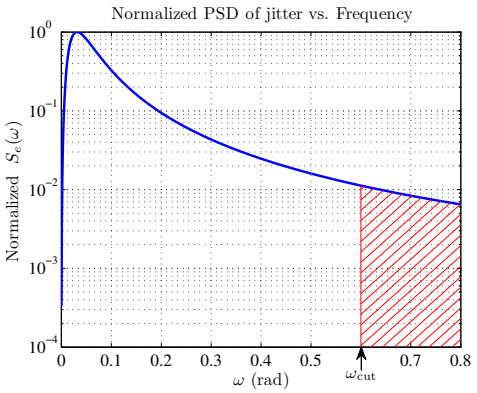
$$S_\phi(f) = \frac{\beta_\phi f^2}{(f^2 + f_e^2)^2} \tag{4.5}$$

over $0 \leq \omega \leq 2\pi$ and for some constants β_e and $\sigma_e^2 = \pi\beta_e/2f_e$; the variable σ_e^2 denotes the variance of the clock jitter and j is the imaginary unit $j \triangleq \sqrt{-1}$. The PSD given by (4.4) is low-pass and decays as ω approaches π. The corresponding autocorrelation function of the jitter process is given by the sampled inverse discrete-time Fourier transform of (4.4) (See App. 4.A)

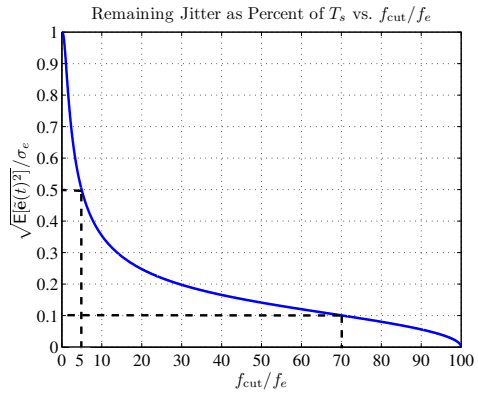
$$R_e(m) \approx \sigma_e^2(1 - 2\pi f_e\,|m|\,T_s)e^{-2\pi f_e|m|T_s} \tag{4.6}$$

Moving forward, we express the root-mean-square (RMS) jitter value, σ_e, as a percentage of the sampling interval T_s by writing $\sigma_e^2 = (\alpha T_s)^2$. For example, if $\alpha = \frac{1}{100}$, then the RMS value of the jitter would be 1% of T_s. The jitter will be modeled as a Gaussian random process with zero-mean and autocorrelation function $R_e(m)$. Figure 4.2(a) plots the PSD of the jitter as given by (4.4).

At this point, we explore up to what frequency the jitter must be recovered in order to reduce the jitter standard-deviation to a certain level. This computation



(a) The shape of the PSD of the jitter for $f_e = 5$MHz and $f_s = 1$GHz. The PSD is normalized such that its peak value is equal to one.



(b) An illustration of the expected amount of jitter reduction as a function of the perfect recovery frequency $\omega_{\text{cut}}/(2\pi T_s)$ when $f_e = 5$MHz and $f_s = 1$GHz.

Figure 4.2: The PSD of the jitter using $f_e = 5$MHz, $f_s = 1$GHz as well as the expected amount of jitter reduction as a function of ω_{cut}.

allows us to evaluate which jitter recovery goals are feasible with digital recovery. Assume our recovery methods are able to recover well the frequency content of the jitter up to some frequency ω_{cut} – see Figure 4.2(a). Denote the estimated jitter by $\hat{\boldsymbol{e}}[n]$ and the remaining jitter by $\tilde{\boldsymbol{e}}[n] = \boldsymbol{e}[n] - \hat{\boldsymbol{e}}[n]$. This means that the data samples will still contain contributions from the jitter components at frequencies above ω_{cut}. The power of the remaining jitter can be found by integrating the PSD of the original jitter from ω_{cut} up to $2\pi - \omega_{\text{cut}}$ and normalizing the result by 2π. We find the RMS value by taking the square root of the result and divide by the original RMS value in order to get the jitter recovery ratio:

$$\frac{\sqrt{\mathsf{E}\left[\tilde{e}(t)^2\right]}}{\sigma_e} = \frac{1}{\sigma_e}\sqrt{\frac{1}{2\pi}\int_{\omega_{\text{cut}}}^{2\pi-\omega_{\text{cut}}} S_e(e^{j\omega})d\omega} \tag{4.7}$$

The difficulty for digital recovery methods can be seen from a plot of (4.7). Assume the numerical values $f_s = 1$GHz and $f_e = 5$MHz. As seen in Figure 4.2(b), in order to reduce the jitter by a factor of 2 (i.e., for the ratio in (4.7)

to evaluate to 0.5), the jitter must be perfectly estimated up to a frequency of 25MHz; the reduction of jitter by a factor of 10 (ratio of 0.1) requires the jitter to be perfectly estimated up to a frequency of $\approx$ 350MHz. Although halving the jitter standard-deviation may be possible in the digital domain, reducing it by a factor of 10 is generally not feasible due to the presence of interfering signals. We will see in the next section (specifically Fig. 4.3) how halving the jitter standard-deviation will improve the SNR of the incoming signal significantly.

Throughout the remainder of the article, we use the following assumptions:

Assumption 4.1. *The jitter $\boldsymbol{e}[n]$ is zero-mean Gaussian with standard-deviation $\sigma_e = \alpha T_s$.*

Assumption 4.2. *The jitter $\boldsymbol{e}[n]$ is small in comparison to the sampling interval T_s (i.e, $\alpha \ll 1$).*

Assumption 4.3. *The jitter $\boldsymbol{e}[n]$ is slowly varying (i.e, $f_e \ll f_s$).*

Assumption 4.4. *The jitter $\boldsymbol{e}[n]$ is independent of the incoming signal and any noise in the system.*

As noted in App. 4.A, Assumption 4.1 is reasonable when the PLL is locked with small error [76]. When the PLL is not locked, or when the PLL model is not well-approximated as a linear system, then Assumption 4.1 will not hold. Assumption 4.2 also reflects the fact that the jitter is relatively small in practice in comparison to the sampling period; in fact, it is usually less than 1% of the sampling period using current technologies for high-speed sampling systems [77]. Assumption 4.3 assumes that the jitter is slowly varying. This assumption can be controlled by the design of the loop filter of the PLL, which dictates the bandwidth of the phase-noise at the output of the PLL—see App. 4.B. Finally, Assumption 4.4 states that the jitter noise is independent of the incoming signal

and noise sources in the system, including quantization noise. The jitter may be weakly dependent with the thermal noise present in the system (since the phase-noise of the free-running oscillator in the PLL is also affected by thermal noise). However, it is usually assumed that such dependence is sufficiently weak and that thermal noise at the input of the ADC is small in comparison to the incoming signal and other noise in the system.

4.1.2 Effects of the clock jitter

In this section, we investigate the effect of the clock jitter on the sampled data. We quantify by how much the random shifts in the sampling instants reduce the signal-to-noise ratio (SNR) at the output of the ADC. We consider two models for the incoming signal: (1) a complex-sinusoidal tone and (2) a complex band-limited random signal. We will use the latter case for analysis of signal degradation in UWB and cognitive radio scenarios. The case of a sinusoidal tone allows us to analyze the effect of the jitter on training tones that we inject into the ADC. When other noise sources are ignored, this analysis was previously conducted in, for example, [78] and [79, p.68]. The analysis of the SNR due to only jitter for band-limited random signals was conducted in [78]. Here we include other noise sources such as front-end noise and quantization noise, as well as jitter. The purpose of the discussion below is to highlight the benefit of reducing the jitter standard deviation in terms of the SNR of the incoming desired signal. It may not be clear, otherwise, how reducing the jitter by a factor of two would improve performance.

4.1.2.1 Complex Exponential Input Signal

We consider first the case of a deterministic complex-sinusoidal incoming signal $r(t)$ with amplitude A_r and frequency f_r:

$$r(t) = A_r e^{j(2\pi f_r t + \theta_r)} \tag{4.8}$$

The samples of $r(t)$ corrupted by the jitter and additive noise source $\boldsymbol{v}[n]$ can be written as (see Fig. 4.1 and (4.66) in App. 4.A):

$$\check{\boldsymbol{z}}[n] \approx r\left(t - \frac{\boldsymbol{\phi}_s(t)}{2\pi f_s}\right)\Bigg|_{t=nT_s} + \boldsymbol{v}[n] \tag{4.9}$$

where $\boldsymbol{v}[n]$ denotes complex-valued noise, such as quantization and front-end noise. Note that we are denoting $\check{\boldsymbol{z}}[n]$ with boldface notation as this signal is random due to the presence of the jitter $\boldsymbol{e}[n]$ and the random noise $\boldsymbol{v}[n]$. The signal $r[n]$ with no distortion is denoted by normal font. By using the first order Taylor series, we have

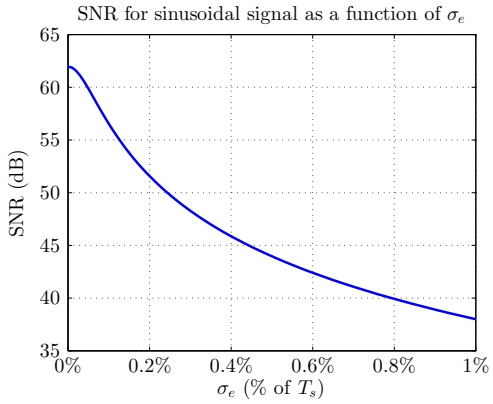
$$\begin{aligned} r\left(t - \frac{\boldsymbol{\phi}_s(t)}{2\pi f_s}\right)\Bigg|_{t=nT_s} &= r\left(nT_s - \frac{\boldsymbol{\phi}_s(nT_s)}{2\pi f_s}\right) \\ &\approx r(nT_s) - \frac{\boldsymbol{\phi}_s(nT_s)}{2\pi f_s}\dot{r}(nT_s) \\ &\approx r[n] + \boldsymbol{e}[n]\dot{r}[n] \end{aligned} \tag{4.10}$$

where $\dot{r}(t)$ denotes the derivative of $r(t)$ evaluated at time t, and $\boldsymbol{e}[n] \approx -\frac{\boldsymbol{\phi}_s(nT_s)}{2\pi f_s}$ by (4.3). Substituting (4.10) into (4.9), we have:

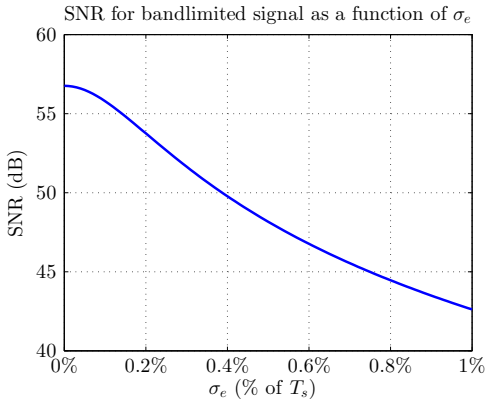
$$\check{\boldsymbol{z}}[n] \approx r[n] + \boldsymbol{e}[n]\dot{r}[n] + \boldsymbol{v}[n] \tag{4.11}$$

where the notation $\dot{r}[n]$ denotes

$$\dot{r}[n] \triangleq \frac{d}{dt}r(t)\Bigg|_{t=nT_s}$$



(a) SNR vs. σ_e for complex-sinusoidal signals.



(b) SNR vs. σ_e for bandlimited signals.

Figure 4.3: In (a), (b), we show the relationship between SNR and σ_e for complex-sinusoidal signals and bandlimited signals, respectively. The parameters include: $B_r = f_r = 200\text{MHz}$, $A_r = 1$, $\sigma_r = 0.548$, $T_s = 1\text{ns}$, and $\sigma_v^2 = \frac{2}{3 \cdot 4^{10}}$. This noise variance corresponds to the quantization noise power for a 10-bit ADC.

Expression (4.11) reveals how jitter and and other noise sources distort the sample value. Using the fact that the power of the complex tone $r(t)$ is A_r^2, the resulting SNR at the output of the sampler is:

$$\text{SNR} \approx \frac{A_r^2}{\left(2\pi f_r A_r \sigma_e\right)^2 + \sigma_v^2} \tag{4.12}$$

Expression (4.12) was derived in [78] and [79, p.68] when $\sigma_v^2 = 0$. Figure 4.3(a) plots the SNR degradation vs. σ_e. Observe that the SNR degrades quickly with σ_e and the jitter becomes the dominant noise source as $\left(2\pi f_r A_r \sigma_e\right)^2$ becomes larger than σ_v^2. Observe that a reduction of the jitter standard deviation from 1% of the sampling period to 0.5% of the sampling period yields approximately 6dB in SNR improvement for a complex sinusoid with the listed parameters.

4.1.2.2 Band-limited Random Signal

We now examine the effect of jitter on a zero-mean wide-sense-stationary (WSS) random process, $\boldsymbol{r}(t)$, with auto-correlation function $R_r(\tau)$ where $R_r(0) = \sigma_r^2$. Again, similar to (4.11), we get

$$\check{\boldsymbol{z}}[n] \approx \boldsymbol{r}[n] + \boldsymbol{e}[n]\dot{\boldsymbol{r}}[n] + \boldsymbol{v}[n] \tag{4.13}$$

The SNR after sampling is given by

$$\text{SNR} \approx \frac{\sigma_r^2}{\sigma_e^2 \cdot \mathsf{E}\left|\dot{\boldsymbol{r}}[n]\right|^2 + \sigma_v^2} \tag{4.14}$$

In order to evaluate the power of the derivative we assume that $\boldsymbol{r}(t)$ is mean-square continuous, i.e.,

$$\mathsf{E}\left|\boldsymbol{r}(t)\right|^2 < \infty, \quad \lim_{s\to t} \mathsf{E}\left|\boldsymbol{r}(s) - \boldsymbol{r}(t)\right|^2 = 0.$$

For such a process, it holds that [80]:

$$R_{\dot{\boldsymbol{r}}}(\tau) = -\frac{d^2}{d\tau^2} R_r(\tau) \tag{4.15}$$

in terms of the auto-correlation function of the process $\boldsymbol{r}(t)$. From this result, it follows that

$$\mathsf{E}\left|\dot{\boldsymbol{r}}[n]\right|^2 = R_{\dot{\boldsymbol{r}}}(0) = -\frac{d^2}{d\tau^2} R_r(\tau)\bigg|_{\tau\to 0} \tag{4.16}$$

so that (4.14) becomes

$$\text{SNR} \approx \frac{\sigma_r^2}{-\sigma_e^2 \cdot \frac{d^2}{d\tau^2} R_r(\tau)\big|_{\tau\to 0} + \sigma_v^2} \tag{4.17}$$

For illustration purposes, consider a bandlimited signal $\boldsymbol{r}(t)$ with box-car power spectral density (PSD) and sinc autocorrelation function:

$$S_r(f) = \frac{\sigma_r^2}{2B_r} \text{rect}\left(\frac{f}{2B_r}\right) \tag{4.18}$$

where

$$\text{rect}(a) = \begin{cases} 1, & |a| < \frac{1}{2} \\ \frac{1}{2}, & |a| = \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.19}$$

and $2B_r$ is the passband bandwidth of the signal $\boldsymbol{r}(t)$ in Hz. Then

$$R_r(\tau) = \begin{cases} \sigma_r^2 \frac{\sin(2B_r\pi\tau)}{2B_r\pi\tau}, & \tau \neq 0 \\ \sigma_r^2, & \tau = 0 \end{cases} \tag{4.20}$$

which is derived by taking the inverse Fourier transform of the PSD $S_r(f)$. It follows that

$$-\frac{d^2}{d\tau^2} R_r(\tau)\Big|_{\tau\to 0} = \frac{4}{3}\pi^2 B_r^2 \sigma_r^2 \tag{4.21}$$

and the SNR expression (4.17) gives

$$\text{SNR} \approx \frac{\sigma_r^2}{\frac{1}{3}\left(2\pi B_r \sigma_r \sigma_e\right)^2 + \sigma_v^2} \tag{4.22}$$

Observe that when the jitter noise dominates the noise $\boldsymbol{v}[n]$, the SNR is determined solely by the variance of the jitter σ_e^2 and the bandwidth of the incoming signal B_r (σ_r^2 is eliminated). Expression (4.22) was derived in [78] when $\sigma_v^2 = 0$.

The effect of jitter on the SNR is illustrated in Figure 4.3(b). We choose the per-channel standard-deviation of the signal $\boldsymbol{r}(t)$ by assuming the signal is Gaussian and the probability of exceeding an absolute-value amplitude of each channel of one is less than 1% since it is assumed that the ADC has a range of ± 1. That is, we select the single-channel standard-deviation σ_r^{SC} such that:

$$Q\left(\frac{1}{\sigma_r^{\text{SC}}}\right) \leq \frac{1}{200} \Longrightarrow \sigma_r^{\text{SC}} \leq \frac{1}{Q^{-1}\left(\frac{1}{200}\right)} \approx 0.388 \tag{4.23}$$

where

$$p = Q(x) \triangleq \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du$$

is the standard Q-function and $x = Q^{-1}(p)$ is the inverse Q-function. Assuming the data on each channel is independent, the total signal standard-deviation σ_r becomes $\sigma_r = \sqrt{2}\sigma_r^{\mathrm{SC}}$. Figure 4.3(b) again shows that the clock jitter reduces the SNR considerably if left uncompensated. Also notice that a reduction of the jitter standard deviation from 1% of the sampling period to 0.5% of the sampling period yields approximately 5.5dB in SNR improvement for a bandlimited signal with the listed parameters. In the next section, we derive methods for estimating the jitter $\boldsymbol{e}[n]$ that will be used for the compensation of the jittered samples.

4.2 Estimation of clock jitter

As illustrated in the previous section, the sampling jitter has an adverse effect on the SNR of the ADC. In this section, we propose methods to estimate the jitter sequence, $\boldsymbol{e}[n]$. We propose a signal injection algorithm for wideband receivers similar to the technique we proposed earlier in [41, 64] and independently by [32, 33, 73]. There is however a fundamental difference between our techniques and other related works such as [32, 33]. While our approach and the approach taken in [32, 33] rely on the use of a reference signal injection outside the received signal band (we explain why it is useful to inject such training tones in App. 4.C.), the problem studied in this paper is *not* the same as the one discussed in [32, 33]. In these references, the receiver sub-samples the incoming narrow-bandwidth RF signal and a reference tone using a low sampling rate. The incoming signal then folds down to an intermediate frequency and is subsequently digitally processed. We *do not* assume that the incoming signal has a narrow bandwidth; in fact, we assume that the incoming signal has a wide bandwidth. This is a key difference between the two approaches and this fact presents new challenges and difficulties in the signal injection step. First, it is not possible to inject a low-frequency

signal anymore along with the down-converted signal since the down-converted signal now has a wide bandwidth and will interfere with the reference tone (and vice versa). It is also challenging to generate a clean sinusoid (with negligible phase-noise) at high-frequencies. For this reason, we must propose a signal injection architecture that is able to produce a relatively clean sinusoid outside the desired signal bandwidth. Second, the jitter estimation algorithm from [32, 33] requires the use of $\tan^{-1}(\cdot)$ blocks. Such blocks generally consume considerable power and introduce delay in the system. We avoid this problem by performing some processing off-line (at the system start-up) and proceed with low-complexity estimation in real-time. Finally, the compensation method proposed in [32, 33] effectively only compensates the carriers of the bandpass signals, and not the signals themselves, since it is assumed that the signals are narrowband and thus the dominant jitter effects act on the carriers and not the signals. Since we assume that the incoming signal is wide-band, we must compensate the signal itself for the jitter effects.

For all these reasons, we replace the radio architecture of Fig. 4.1 with Fig. 4.4 where a training signal $\boldsymbol{y}(t)$ is generated inside the radio and injected along with the downconverted received signal $\boldsymbol{r}(t)$. The samples at the output of the ADCs constitute the in-phase and quadrature-phase components of the perturbed signal $\check{\boldsymbol{z}}[n]$. We focus our discussion on "Direct Downconversion Receivers[1]". We will describe a direct recovery method and an optimal jitter recovery scheme based on the minimization of a mean-square-error cost.

[1] A Direct-Downconversion Receiver performs an analog quadrature down-conversion to base-band of the signal of interest. The base-band signal is then sampled at a rate at least twice the bandwidth of the signal per channel. [79, p.43]

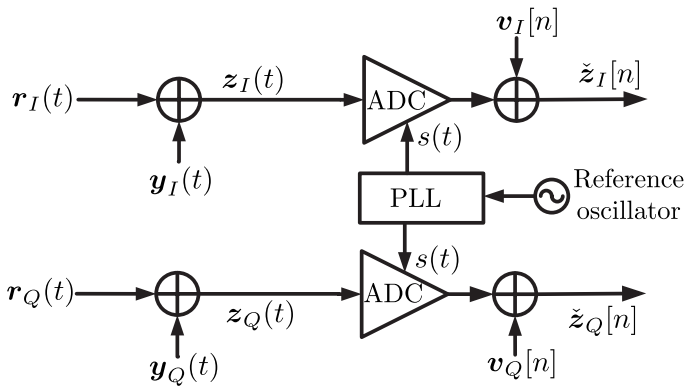


Figure 4.4: Proposed signal injection architecture used for jitter estimation.

4.2.1 Direct downconversion receiver

When the received radio signal $\boldsymbol{r}(t)$ is a wide-band radio signal at baseband (downconverted in the circuit), it is not possible to inject the training signal $\boldsymbol{y}(t)$ in the low-frequency region as it will interfere with $\boldsymbol{r}(t)$. For this reason, we must inject $\boldsymbol{y}(t)$ at high-frequency. Unfortunately, it is generally not practical to generate a noise-free training signal $\boldsymbol{y}(t)$ at high-frequency. It is, however, feasible to up-convert a low-frequency training signal with a high-frequency carrier. To choose the carrier signal, we rely on a frequency-halved sampling-clock. Thus, let $\boldsymbol{s}(t)$ denote the sampling clock signal that is generated by the PLL–recall (4.1):

$$\boldsymbol{s}(t) = \sin\left(2\pi f_s t + \boldsymbol{\phi}_s(t)\right) \tag{4.24}$$

where f_s is the sampling frequency and $\boldsymbol{\phi}_s(t)$ is the phase-noise. Now a scaled, frequency-halved clock signal, which we shall denote by $\boldsymbol{p}(t)$, will have the form

$$\boldsymbol{p}(t) = A_p \sin\left(2\pi\frac{f_s}{2}(t-\tau_p) + \frac{\boldsymbol{\phi}_s\left(t-\tau_p\right)}{2}\right) \tag{4.25}$$

where τ_p models some constant delay implemented in the circuit and A_p is a scaling factor. The frequency division is a low-noise, simple integer frequency divider that preserves the time jitter $\boldsymbol{e}[n]$ and halves the phase-noise $\boldsymbol{\phi}_s(t)$ [40, p.159].

When τ_p is relatively small, we can say that the delayed signal $\boldsymbol{\phi}_s(t-\tau_p)$ is approximately the same as $\boldsymbol{\phi}_s(t)$ since we have assumed earlier in Assumption 4.3 that the phase-noise is relatively slow compared to the sampling time. Sampling $\boldsymbol{p}(t)$ with $\boldsymbol{s}(t)$ gives $\check{\boldsymbol{p}}[n] = \boldsymbol{p}(nT_s + \boldsymbol{e}[n])$ where $\boldsymbol{e}[n]$ is given by (4.3). Thus, we can write the samples $\boldsymbol{p}[n]$ as

$$\begin{aligned}\check{\boldsymbol{p}}[n] &= A_p \sin\left(2\pi\frac{f_s}{2}(nT_s+\boldsymbol{e}[n]-\tau_p)+\frac{\boldsymbol{\phi}_s(nT_s+\boldsymbol{e}[n]-\tau_p)}{2}\right)\\ &\approx A_p \sin\left(2\pi\frac{f_s}{2}(nT_s+\boldsymbol{e}[n]-\tau_p)+\frac{\boldsymbol{\phi}_s[n]}{2}\right)\\ &= e^{j2\pi\left(\frac{f_s}{2}\right)nT_s}\end{aligned} \tag{4.26}$$

where the first equality is due to (4.67), the second approximation is a consequence of (4.3) and the fact that the jitter is small (Assumption 4.2) and slow (Assumption 4.3) and the last equality is a consequence of choosing $A_p = -\csc\left(2\pi\frac{f_s}{2}\tau_p\right)$ where $\csc(x) \triangleq 1/\sin(x)$. Thus, we observe that the samples of the signal derived by frequency-halving the sampling-clock and with negligible delay will have no jitter; it is approximately deterministic. Notice that $\check{\boldsymbol{p}}[n]$ is a real signal that we represented as a complex signal, which will be used for convenience in (4.28)-(4.30).

We now modulate $\boldsymbol{p}(t)$ by a low-frequency, low-noise training tone $w(t)$ in order to up-convert $w(t)$ to become a high-frequency training signal without the difficulty of generating it as a high-frequency signal directly. Thus, we choose the training signal for the direct-downconversion receiver to be:

$$\boldsymbol{y}(t) = w(t)\boldsymbol{p}(t) \tag{4.27}$$

and the corresponding samples of the training signal become

$$\begin{aligned}\check{\boldsymbol{y}}[n] &\triangleq \boldsymbol{y}(nT_s + e[n]) \\ &= w\left(nT_s + \boldsymbol{e}[n]\right)\check{\boldsymbol{p}}[n] \\ &\approx w\left(nT_s + \boldsymbol{e}[n]\right)e^{j2\pi\left(\frac{f_s}{2}\right)nT_s}\end{aligned} \tag{4.28}$$

We chose the tone $w(t)$ to be a low-frequency complex-sinusoid:

$$w(t) \triangleq A_w e^{j(2\pi f_w t + \theta_w)} \tag{4.29}$$

Then, by substituting (4.29) into (4.28), we can show that the sampled injected signal $\check{\boldsymbol{y}}[n]$ will be a complex-sinusoid located at $\frac{f_s}{2} + f_w$:

$$\begin{aligned}\check{\boldsymbol{y}}[n] &\approx w\left(nT_s + \boldsymbol{e}[n]\right)e^{j2\pi\left(\frac{f_s}{2}\right)nT_s} \\ &= A_w e^{j\left(2\pi\left(\frac{f_s}{2}+f_w\right)nT_s + 2\pi f_w \boldsymbol{e}[n] + \theta_w\right)}\end{aligned} \tag{4.30}$$

Figure 4.5 illustrates the tone injection scheme. The training tone $\boldsymbol{y}(t)$ is injected along with the desired signal $\boldsymbol{r}(t)$ to form the complex-valued signal $\boldsymbol{z}(t)$:

$$\boldsymbol{z}(t) \triangleq \boldsymbol{r}(t) + \boldsymbol{y}(t) \tag{4.31}$$

The composite signal $\boldsymbol{z}(t)$ is then sampled by the ADC to yield the complex-valued discrete-time signal $\check{\boldsymbol{z}}[n]$:

$$\check{\boldsymbol{z}}[n] \triangleq \boldsymbol{z}(nT_s + \boldsymbol{e}[n]) + \boldsymbol{v}[n] \tag{4.32}$$

where $\boldsymbol{v}[n]$ is noise introduced at the ADC, which includes quantization noise.

4.2.2 Direct jitter estimation

We assume that we are able to estimate θ_w as $\hat{\boldsymbol{\theta}}_w$ say, by using a digital PLL technique [81] up to an ambiguity term that is a multiple of 2π (see App. 4.D). A

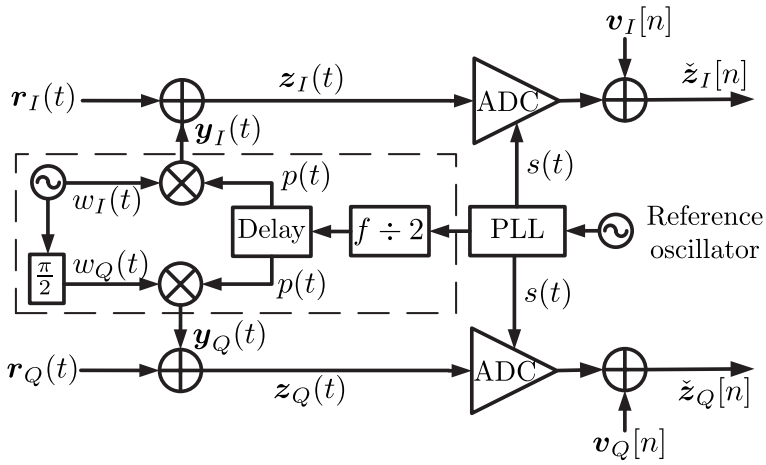


Figure 4.5: High-frequency injection model. The received signal $\boldsymbol{r}(t)$ is sampled along with a training signal $\boldsymbol{y}(t)$ generated by multiplying by a frequency divided and delayed sampling clock by a low-frequency complex-oscillator. The complex-valued noise $\boldsymbol{v}[n]$ represents other noise sources such as quantization noise.

good estimate for A_w is also assumed to be available through a calibration routine since A_w is a design parameter and is expected to remain relatively constant. Then, we can introduce the following filtered signal, obtained by down-converting $\check{\boldsymbol{y}}[n]$ and low-pass filtering the result:

$$\begin{aligned}
\boldsymbol{x}[n] &\triangleq \text{LPF}\left\{\frac{1}{A_w}\check{\boldsymbol{z}}[n]e^{-j\left(2\pi\left(\frac{f_s}{2}+f_w\right)nT_s+\hat{\boldsymbol{\theta}}_w\right)}\right\} \\
&= \text{LPF}\left\{e^{j\left(2\pi f_w \boldsymbol{e}[n]+\theta_w-\hat{\boldsymbol{\theta}}_w\right)}\right. \\
&\quad \left. + \frac{1}{A_w}\left(\boldsymbol{r}\left(nT_s+\boldsymbol{e}[n]\right)+\boldsymbol{v}(nT_s+\boldsymbol{e}[n])\right)e^{-j\left(2\pi\left(\frac{f_s}{2}+f_w\right)nT_s+\hat{\boldsymbol{\theta}}_w\right)}\right\} \\
&= \text{LPF}\left\{e^{j\left(2\pi f_w \boldsymbol{e}[n]+\theta_w-\hat{\boldsymbol{\theta}}_w\right)} + \boldsymbol{u}_d[n]\right\}
\end{aligned} \tag{4.33}$$

where we combined all the noise and interfering signals into $\boldsymbol{u}_d[n]$. By utilizing Assumption 4.2, it is possible to recover the jitter as:

$$\hat{\boldsymbol{e}}_1[n] \triangleq \frac{\sin^{-1}\left(\mathfrak{Im}\left\{\boldsymbol{x}[n]\right\}\right)}{2\pi f_w} \tag{4.34}$$

where $\sin^{-1}(\cdot)$ returns the angle in either the first or fourth quadrant and $\mathfrak{Im}\{x\}$ returns the imaginary part of the complex number x. When $\mathfrak{Im}\{\boldsymbol{x}[n]\}$ is small, we can use a small-angle approximation for $\sin(\cdot)$ to have:

$$\hat{\boldsymbol{e}}_2[n] \approx \frac{\mathfrak{Im}\{\boldsymbol{x}[n]\}}{2\pi f_w} \tag{4.35}$$

A CORDIC algorithm may be used to evaluate the arcsin in (4.34). Alternatively, it is possible to examine the effect of using a series expansion for $\sin^{-1}(\cdot)$ to an arbitrary number of terms. We perform this analysis later for a single term in the series expansion. It can be shown that when the jitter is small, the single term analysis is sufficiently accurate due to the linearity of $\sin(\cdot)$ for small angles. Fig. 4.6 illustrates the algorithm when (4.35) is used as the estimator for the jitter.

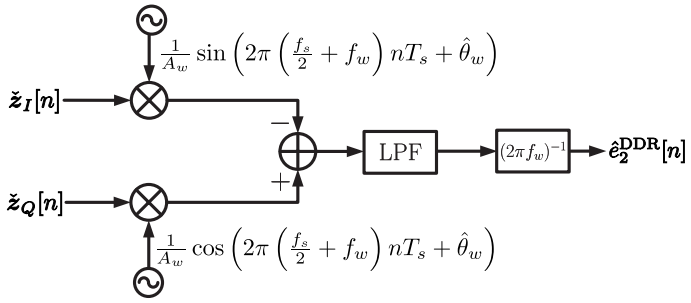


Figure 4.6: Jitter recovery structure based on (4.35).

4.2.3 Adaptive jitter estimation

We wish to recover the jitter $\boldsymbol{e}[n]$ from the samples $\check{\boldsymbol{z}}[n]$. In order to accomplish this, we introduce a processed complex-valued signal $\boldsymbol{d}[n]$ that is the result of shifting the perturbed training tone $\check{\boldsymbol{y}}[n]$ in (4.30) down to baseband and nor-

malizing its amplitude by dividing by the factor A_w:

$$\begin{aligned}\boldsymbol{d}[n] &\triangleq \frac{1}{A_w}\check{\boldsymbol{z}}[n]e^{-j2\pi\left(\frac{f_s}{2}+f_w\right)nT_s} \\ &= e^{j(2\pi f_w \boldsymbol{e}[n]+\theta_w)} + \boldsymbol{u}_a[n] \end{aligned} \tag{4.36}$$

where the complex-valued noise $\boldsymbol{u}_a[n]$ is defined as

$$\boldsymbol{u}_a[n] \triangleq \frac{1}{A_w}(\check{\boldsymbol{r}}[n] + \boldsymbol{v}(nT_s + \boldsymbol{e}[n]))e^{-j2\pi\left(\frac{f_s}{2}+f_w\right)nT_s}$$

We now write down a stochastic optimization problem that allows us to recover the jitter $\boldsymbol{e}[n]$. We note that the data signal $\boldsymbol{d}[n]$ in (4.36) is a complex-sinusoid at baseband and we assume that the jitter $\boldsymbol{e}[n]$ is slowly varying and that most of its power lies at low-frequencies (as assumed in Assumption 4.3). We let the phase of the complex sinusoid in (4.36) be denoted by $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} \triangleq 2\pi f_w \boldsymbol{e}[n] + \theta_w \tag{4.37}$$

We estimate $\boldsymbol{\theta}$ by solving the following minimum mean-square-error problem: [42, p.163].

$$J_{\text{MSE}}(\boldsymbol{\theta}) = \mathsf{E}\left[\text{LPF}\left\{\left|\boldsymbol{d}[n] - e^{j\boldsymbol{\theta}}\right|^2\right\}\right] \tag{4.38}$$

The low-pass filter removes the out-of-band noise caused by the incoming signal $\check{\boldsymbol{r}}[n] \triangleq \boldsymbol{r}(nT_s + \boldsymbol{e}[n])$ and the broadband noise $\boldsymbol{v}(nTs + \boldsymbol{e}[n])$ while retaining most of the power of the jitter process. In order to minimize (4.38) iteratively, we implement a stochastic gradient-descent algorithm that moves along the opposite direction of the gradient of (4.38) at each iteration. Thus, we compute the gradient as:

$$\begin{aligned}\nabla J_{\text{MSE}}(\boldsymbol{\theta}) &= \mathsf{E}\left[\text{LPF}\left\{j\boldsymbol{d}[n]e^{-j\boldsymbol{\theta}}\right\}\right] + \mathsf{E}\left[\text{LPF}\left\{-j\boldsymbol{d}^*[n]e^{j\boldsymbol{\theta}}\right\}\right] \\ &= -2\mathsf{E}\left[\text{LPF}\left\{\mathfrak{Im}\left\{\boldsymbol{d}[n]e^{-j\boldsymbol{\theta}}\right\}\right]\right\} \end{aligned} \tag{4.39}$$

and the algorithm is given by:

$$\hat{\theta}[n+1] = \hat{\theta}[n] + \mu \mathsf{E}\left[\mathrm{LPF}\left\{\mathfrak{Im}\left\{\boldsymbol{d}[n]e^{-j\hat{\theta}[n]}\right\}\right\}\right] \tag{4.40}$$

where μ is a small positive step-size and $\hat{\theta}[n]$ is the estimate for θ at iteration n. It is possible to use an instantaneous approximation for the gradient by eliminating the expectation operation in (4.40) [42, p.165]:

$$\mathsf{E}\left[\mathrm{LPF}\left\{\mathfrak{Im}\left\{\boldsymbol{d}[n]e^{-j\hat{\theta}[n]}\right\}\right\}\right] \approx \mathrm{LPF}\left\{\mathfrak{Im}\left\{\frac{1}{A_w}\check{z}[n]e^{-j(2\pi(\frac{f_s}{2}+f_w)nT_s+\hat{\theta}[n])}\right\}\right\} \tag{4.41}$$

where $\check{z}[n]$ denotes the actual realization of the random process $\check{\boldsymbol{z}}[n]$. Also, since we are only interested in the imaginary part, we compute the expression for it directly as:

$$\begin{aligned}
&\mathsf{E}\left[\mathrm{LPF}\left\{\mathfrak{Im}\left\{\boldsymbol{d}[n]e^{-j\hat{\theta}[n]}\right\}\right\}\right] \\
&\approx \frac{1}{A_w}\mathrm{LPF}\Bigg\{\check{z}_Q[n]\cos\left(2\pi\left(\frac{f_s}{2}+f_w\right)nT_s+\hat{\theta}[n]\right) - \\
&\qquad\qquad \check{z}_I[n]\sin\left(2\pi\left(\frac{f_s}{2}+f_w\right)nT_s+\hat{\theta}[n]\right)\Bigg\}
\end{aligned} \tag{4.42}$$

where $\check{z}_I[n]$ and $\check{z}_Q[n]$ are the in-phase and quadrature phase components of $\check{z}[n]$. Since the training signal will have some constant phase shift due to θ_w, we recover the instantaneous jitter from $\hat{\theta}[n]$ (4.37) as:

$$\hat{e}_3[n] = \frac{\hat{\theta}[n] - \bar{\theta}[n]}{2\pi f_w} \tag{4.43}$$

The running average $\bar{\theta}[n]$ can be estimated using a forgetting factor filter such as:

$$\bar{\theta}[n+1] = \lambda\bar{\theta}[n] + (1-\lambda)\hat{\theta}[n] \tag{4.44}$$

where λ is a forgetting factor close to one. The block diagram for the adaptive jitter recovery method is presented in Fig. 4.7.

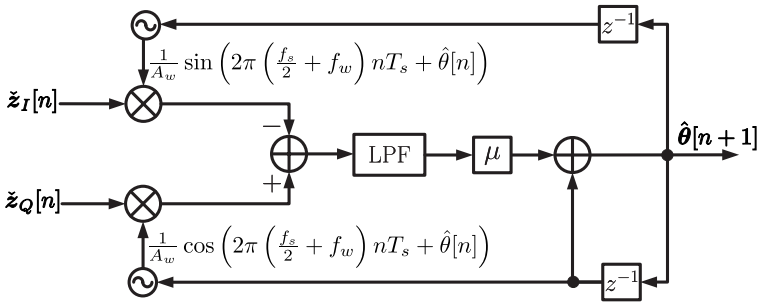


Figure 4.7: Adaptive jitter recovery structure based on (4.40)–(4.43)

4.3 Compensation of clock jitter

Once the jitter, $\boldsymbol{e}[n]$, has been estimated through (4.34), (4.35), or (4.43), it is still necessary to process $\check{\boldsymbol{r}}[n]$ to recover a cleaner signal $\hat{\boldsymbol{r}}[n]$. Many different methods have been proposed to accomplish this task, such as in [57, 71]. In this section we use a simple compensator based on a first-order approximation like the one proposed in [57] due to its simplicity in implementation. As we will see through simulations, the use of a single derivative compensator is sufficient to provide near-theoretical SNR performance as predicted in Sec. 4.1.2. More sophisticated differentiators, however, can have benefits when the noise variance is lower. We motivate the compensation step via a Taylor expansion of the realization of the process $\check{\boldsymbol{r}}[n]$ by noting that

$$\begin{aligned} r[n] &= r(nT_s + e[n] - e[n]) \\ &\approx r(nT_s + e[n]) - e[n]\ \dot{r}(t)\big|_{t=nT_s+e[n]} \\ &= \check{r}[n] - e[n]\ \dot{r}(t)\big|_{t=nT_s+e[n]} \end{aligned} \tag{4.45}$$

The above approximation can be extended to include higher-order derivatives to improve the accuracy of the compensation step, as was done in [57]. In general, however, the noise in the estimation of the jitter reduces the benefit of using

higher derivatives (the work in [57] assumes that the jitter is perfectly estimated). The derivative of the realization $r(t)$ is not available after sampling. However, consider the derivative of $r\left(t - \frac{\boldsymbol{\phi}_s(t)}{2\pi f_s}\right)$ evaluated at $t = nT_s$:

$$\left.\frac{dr(t - \frac{\boldsymbol{\phi}_s(t)}{2\pi f_s})}{dt}\right|_{t=nT_s} = \dot{r}(t)|_{t=nT_s+e[n]} \left(1 - \left.\frac{\dot{\boldsymbol{\phi}}_s(t)}{2\pi f_s}\right|_{t=nT_s}\right)$$

where $-\left.\frac{\boldsymbol{\phi}_s(t)}{2\pi f_s}\right|_{t=nT_s} \approx \boldsymbol{e}[n]$ due to (4.3). If $\boldsymbol{\phi}_s(t)$ is assumed to be small and slowly varying (in comparison to the sampling frequency), as effectively assumed in Assumptions 4.1, 4.2 due to (4.3), it is possible to ignore the term $\left.\frac{\dot{\boldsymbol{\phi}}_s(t)}{2\pi f_s}\right|_{t=nT_s}$ and write

$$\left.\frac{dr(t - \frac{\boldsymbol{\phi}_s(t)}{2\pi f_s})}{dt}\right|_{t=nT_s} \approx \dot{r}(t)|_{t=nT_s+e[n]} \tag{4.46}$$

It is then possible to create a discrete differentiator [56] in order to approximate the derivative. A simple example that was used in the simulations is the central difference differentiator [82]:

$$\begin{aligned}\left.\frac{dr\left(t - \frac{\boldsymbol{\phi}_s(t)}{2\pi f_s}\right)}{dt}\right|_{t=nT_s} \approx & -\frac{\check{r}(n-5)}{1260T_s} + \frac{5\check{r}(n-4)}{504T_s} - \frac{5\check{r}(n-3)}{84T_s} + \\ & \frac{5\check{r}(n-2)}{21T_s} - \frac{5\check{r}(n-1)}{6T_s} + \frac{5\check{r}(n+1)}{6T_s} - \\ & \frac{5\check{r}(n+2)}{21T_s} + \frac{5\check{r}(n+3)}{84T_s} - \frac{5\check{r}(n+4)}{504T_s} + \\ & \frac{\check{r}(n+5)}{1260T_s}\end{aligned} \tag{4.47}$$

A more sophisticated differentiator design may yield the same results as a long impulse response filter with fewer taps [56, 83]. In practice, a low-pass filter is placed prior to the derivative filter to suppress noise at high-frequencies. We first recover the signal $\check{r}[n]$ as

$$\check{r}[n] \approx \text{LPF}\left\{\check{z}[n]\right\} \tag{4.48}$$

We then apply (4.45) by replacing $e[n]$ by its estimate $\hat{e}[n]$ found using any of the methods in the previous section and the derivative approximation (4.46) to recover the estimate $\hat{r}[n]$ of the signal $r[n]$:

$$\hat{r}[n] \triangleq \check{r}[n] - \hat{e}[n] \left. \frac{dr\left(t - \frac{\phi_s(t)}{2\pi f_s}\right)}{dt} \right|_{t=nT_s} \tag{4.49}$$

The architecture for a single derivative reconstruction is illustrated in Fig. 4.8.

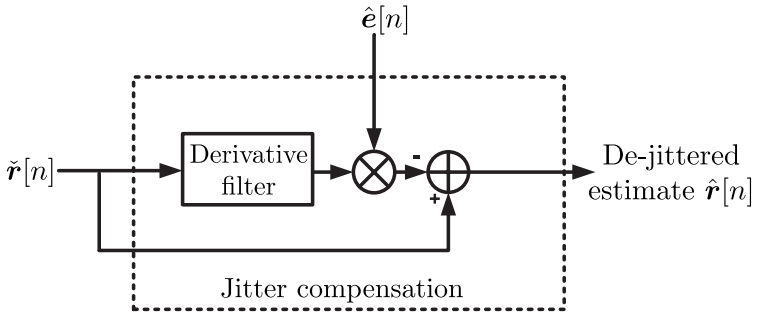


Figure 4.8: Proposed compensation algorithm (4.49).

4.4 Mean-square-error analysis

In this section, we analyze the mean-square-error (MSE) performance of the sinusoidal tone injection methods described in Sec. 4.2. The analysis will be performed in the following subsection for the direct jitter estimation method described in section 4.2.2 and in the next subsection for the adaptive jitter estimation method described in section 4.2.3.

4.4.1 Direct estimation method

Although the analysis in this section is performed for the estimator in (4.35), the derivation can be extended to a higher number of terms used in the approximation

of the $\sin^{-1}(\cdot)$ function in order to extend the analysis to the estimator in (4.34). To begin with, we substitute (4.33) into (4.35) to obtain

$$\hat{\boldsymbol{e}}_2[n] = \frac{1}{2\pi f_w}\text{LPF}\left\{\sin\left(2\pi f_w \boldsymbol{e}[n] + \tilde{\boldsymbol{\theta}}_w\right) + \boldsymbol{u}_d[n]\right\} \tag{4.50}$$

We wish to compute the error:

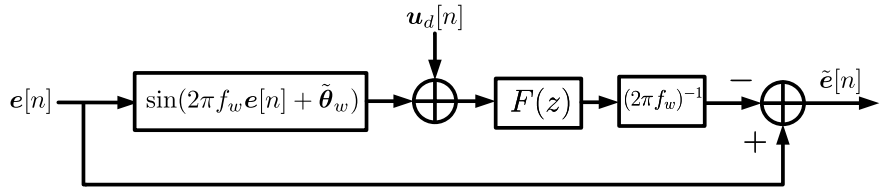
$$\begin{aligned}\tilde{\boldsymbol{e}}[n] &\triangleq \boldsymbol{e}[n] - \hat{\boldsymbol{e}}_2[n] \\ &= \boldsymbol{e}[n] - \frac{1}{2\pi f_w}\text{LPF}\left\{\sin\left(2\pi f_w \boldsymbol{e}[n] + \tilde{\boldsymbol{\theta}}_w\right) + \boldsymbol{u}_d[n]\right\}\end{aligned} \tag{4.51}$$

We illustrate the above system in Fig. 4.9(a), where $F(z)$ represents the response of the low-pass filter. We note that this system is not a linear system. For this reason, we use a common approximation method frequently used in the analysis of PLLs where we assume that the argument of the $\sin(\cdot)$ function is small, and thus we can linearize the model by using a small angle approximation, as illustrated in Fig. 4.9(b).

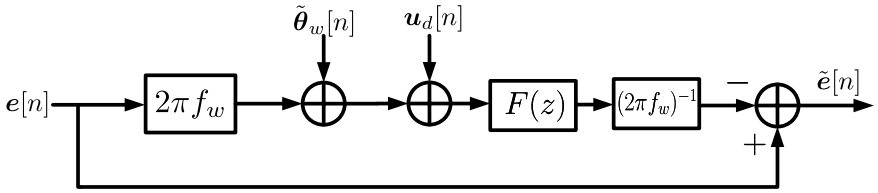
Assuming all input signals are independent of each other, it is possible to use superposition to evaluate the average power of the output $\tilde{\boldsymbol{e}}[n]$. We take advantage of Parseval's theorem for discrete-time Fourier transforms to write the mean-square-error as shown in (4.52).

$$\begin{aligned}\sigma^2_{\tilde{e}} \approx \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|1 - F(e^{j\omega})\right|^2 S_e(e^{j\omega}) d\omega \\ + \frac{1}{(2\pi f_w)^2}\frac{1}{2\pi}\int_{-\pi}^{\pi} \left|F(e^{j\omega})\right|^2 \left(S_u(e^{j\omega}) + S_{\tilde{\theta}}(e^{j\omega})\right) d\omega\end{aligned} \tag{4.52}$$

In (4.52), $S_u(e^{j\omega})$ and $S_{\tilde{\theta}}(e^{j\omega})$ are the PSDs of $\boldsymbol{u}_d[n]$ and $\tilde{\boldsymbol{\theta}}_w[n]$, respecively. The MSE is dominated by terms that are dependent on the design of the phase-recovery method and on the low-pass filter $F(z)$ used in the final step of the recovery method, respectively. Thus, if the noise terms dominate the rest, then it is unnecessary to invest the computational power in computing the $\sin^{-1}(\cdot)$



(a) Nonlinear system to model jitter error $\tilde{\boldsymbol{e}}[n]$ as function of input jitter $\boldsymbol{e}[n]$ and noise $\boldsymbol{u}_d[n]$.



(b) Approximate linear system to model jitter error $\tilde{\boldsymbol{e}}[n]$ as function of input jitter $\boldsymbol{e}[n]$ and noise $\boldsymbol{u}_d[n]$.

Figure 4.9: (a) shows the nonlinear system model while (b) shows the linearized time-invariant system.

operation as suggested by (4.34) and expression (4.35) would be sufficient. Care must be taken during the design of the PLL as discussed in Sec. 4.1.1 since the loop-bandwidth of the PLL can determine upper limits on the estimation capability of any recovery method. Thus, some investment in the PLL can be beneficial as it allows the bandwidth of the low-pass filter to be reduced and relaxes the demands on the analog circuit that generates the training tone.

4.4.2 Adaptive estimation methods

In this section, we analyze the performance of the adaptive estimation method in Fig. 4.7 presented in section 4.2.3. The recovery algorithm closely mirrors the equations of a digital PLL [84]. Thus, we can analyze the tracking performance of the loop by considering the equivalent phase-domain model presented in Fig.

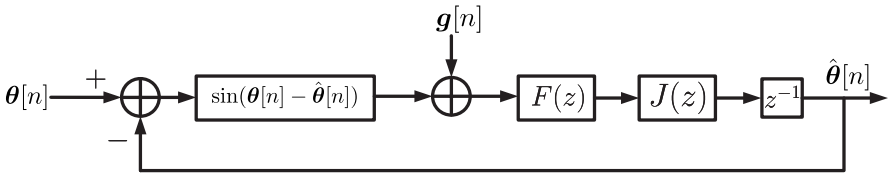


Figure 4.10: Phase domain model of adaptive jitter recovery algorithm

4.10. In the figure, $F(z)$ represents the transfer function of the low-pass filter while $J(z)$ represents the transfer function of the accumulator:

$$J(z) = \frac{\mu}{1 - z^{-1}} \tag{4.53}$$

The noise $\boldsymbol{g}[n]$ in this context is taken to be all interfering noise:

$$\boldsymbol{g}[n] \triangleq \mathfrak{Im}\{\boldsymbol{u}_a[n]e^{-j\hat{\boldsymbol{\theta}}[n]}\} \tag{4.54}$$

It is important to note that the model in Fig. 4.10 is not an LTI system due to the presence of the $\sin(\cdot)$ operator. During steady-state operation, however, it is possible to assume that the estimated angle $\hat{\boldsymbol{\theta}}[n]$ is close to the true angle $\boldsymbol{\theta}[n]$ listed above. The condition that $\hat{\boldsymbol{\theta}}[n]$ is close to $\boldsymbol{\theta}[n]$ roughly implies that the average value of the process $\hat{\boldsymbol{\theta}}[n]$ is θ_w. When this condition is satisfied, we can approximate the $\sin(\cdot)$ operator using the small angle approximation:

$$\sin(\boldsymbol{\theta}[n] - \hat{\boldsymbol{\theta}}[n]) \approx \boldsymbol{\theta}[n] - \hat{\boldsymbol{\theta}}[n] \tag{4.55}$$

This is a common approximation made to study the steady-state performance of phase-locked-loops [9, 84, 85]. We let $\theta_w = 0$ for simplicity and we can model the loop under such conditions as the linear system illustrated in Fig. 4.11 with inputs: noise $\boldsymbol{g}[n]$ and phase-noise $\boldsymbol{\theta}[n]$. The closed-loop transfer function of the system from the point of view of either input can be written as:

$$H(z) = \frac{F(z)J(z)z^{-1}}{1 + F(z)J(z)z^{-1}} \tag{4.56}$$

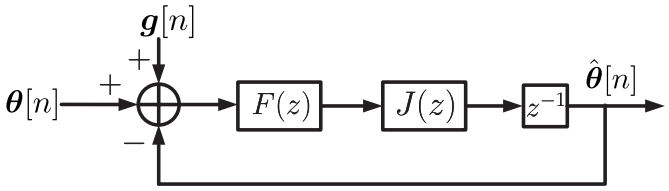


Figure 4.11: Linear system model for adaptive jitter recovery during jitter-tracking operation.

Since the noise and the phase-noise are independent, we use the superposition property of linear time-invariant systems to evaluate the error PSD as [9, 85]

$$S_{\tilde{\theta}[n]}(e^{j\omega}) = |1 - H(e^{j\omega})|^2 S_\theta(e^{j\omega}) + |H(e^{j\omega})|^2 S_g(e^{j\omega}) \tag{4.57}$$

where $S_\theta(e^{j\omega})$ is the PSD of the phase-noise input $\boldsymbol{\theta}[n]$ determined by the PSD of the jitter in (4.4) and $S_g(e^{j\omega})$ is the PSD of the noise. $\tilde{\theta}[n]$ is the estimation error $\tilde{\theta}[n] \triangleq \theta[n] - \hat{\theta}[n]$. The mean-square-error is found using the integral:

$$\sigma_{\tilde{e}}^2 = \frac{1}{(2\pi f_w)^2} \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{\tilde{\theta}[n]}(e^{j\omega}) d\omega \tag{4.58}$$

4.5 SNR analysis

In this section we analyze the performance of the compensation with jitter recovery as described in Sec. 4.2. We once again assume that the signal $\boldsymbol{r}\,(nT_s + \boldsymbol{e}[n]) + \boldsymbol{v}(nTs + \boldsymbol{e}[n])$ is obtained by filtering the incoming samples $\check{\boldsymbol{z}}[n]$ and removing $\boldsymbol{y}(nT_s + \boldsymbol{e}[n])$ as described in (4.48). We further assume that the remaining noise $\boldsymbol{v}(nT_s + \boldsymbol{e}[n])$ is independent of $\boldsymbol{r}[n]$ and that the signal $\boldsymbol{y}(t)$ is sufficiently far

away from $\boldsymbol{r}(t)$. In this case, we describe the recovered samples as

$$\begin{aligned}\hat{\boldsymbol{r}}[n] &= \boldsymbol{r}\left(nT_s + \boldsymbol{e}[n]\right) + \boldsymbol{v}(nT_s + \boldsymbol{e}[n]) - \sum_{i=1}^{N} \frac{\left(\hat{\boldsymbol{e}}[n]\right)^i}{i!} \left.\frac{d^i}{dt^i}\boldsymbol{r}(t+\boldsymbol{e}[n])\right|_{t=nT_s} \\ &= \boldsymbol{r}[n] + \sum_{i=1}^{\infty} \frac{\left(\boldsymbol{e}[n]\right)^i}{i!} \left.\frac{d^i}{dt^i}\boldsymbol{r}(t)\right|_{t=nT_s} + \boldsymbol{v}(nT_s + \boldsymbol{e}[n]) \\ &\quad - \sum_{i=1}^{N} \frac{\left(\hat{\boldsymbol{e}}[n]\right)^i}{i!} \left.\frac{d^i}{dt^i}\boldsymbol{r}(t+\boldsymbol{e}[n])\right|_{t=nT_s} \end{aligned} \tag{4.59}$$

We subtract $\boldsymbol{r}[n]$ from the above expression to find:

$$\begin{aligned}\hat{\boldsymbol{r}}[n] - \boldsymbol{r}[n] &\approx \boldsymbol{v}(nT_s + \boldsymbol{e}[n]) + \sum_{i=1}^{N} \frac{\left(\boldsymbol{e}[n]\right)^i - \left(\hat{\boldsymbol{e}}[n]\right)^i}{i!} \left.\frac{d^i}{dt^i}\boldsymbol{r}(t)\right|_{t=nT_s} \\ &\quad + \sum_{i=N+1}^{\infty} \frac{\left(\boldsymbol{e}[n]\right)^i}{i!} \left.\frac{d^i}{dt^i}\boldsymbol{r}(t)\right|_{t=nT_s} \\ &\approx \boldsymbol{v}(nT_s + \boldsymbol{e}[n]) + \left(\boldsymbol{e}[n] - \hat{\boldsymbol{e}}[n]\right) \left.\frac{d}{dt}\boldsymbol{r}(t)\right|_{t=nT_s} \end{aligned} \tag{4.60}$$

where the first approximation follows our argument in Sec. 4.3 regarding the derivative of $\boldsymbol{r}\left(t - \frac{\phi_s(t)}{2\pi f_s}\right)$ and the second approximation is due to the fact that the first term in the summation has the highest magnitude in comparison to the rest. From this, we can compute the MSE in the recovery of $\boldsymbol{r}[n]$ as

$$\mathsf{E}\left[\left(\hat{\boldsymbol{r}}[n] - \boldsymbol{r}[n]\right)^2\right] \approx \sigma_v^2 + \sigma_{\tilde{e}}^2 \cdot \mathsf{E}\left[\left(\left.\frac{d}{dt}\boldsymbol{r}(t)\right|_{t=nT_s}\right)^2\right] \tag{4.61}$$

where $\sigma_{\tilde{e}}^2$ can be computed using the analysis in Sec. 4.4. Thus, the SNR can be written as

$$\text{SNR} \approx \frac{\sigma_r^2}{\sigma_v^2 + \sigma_{\tilde{e}}^2 \cdot \mathsf{E}\left[\left(\left.\frac{d}{dt}\boldsymbol{r}(t)\right|_{t=nT_s}\right)^2\right]} \tag{4.62}$$

If we assume that $\boldsymbol{r}[n]$ has the auto-correlation function defined by (4.15), we can use the result of App. 4.E to find that

$$\mathsf{E}\left[\left|\left.\frac{d^N}{dt^N}\boldsymbol{r}(t)\right|_{t=nT_s}\right|^2\right] = \frac{(\pi 2B_r)^{2N}}{2N+1}\sigma_r^2, \tag{4.63}$$

which allows us to conclude that the SNR in the compensation can be written as

$$\text{SNR} \approx \frac{\sigma_r^2}{\sigma_v^2 + \frac{1}{3}\left(2\pi B_r \sigma_r \sigma_{\tilde{e}}\right)^2} \tag{4.64}$$

4.6 Simulation results

We assume a sampling period of $T_s = 1$ns. The jitter $\boldsymbol{e}[n]$ is modeled as described in Sec. 4.1.1 and is Gaussian with mean zero and standard deviation $\sigma_e = \alpha T_s$. The jitter is correlated and has bandwidth of f_e. The default parameters are listed in Table 4.1. We simulate the recovery methods for sinusoidal signal injection

Table 4.1: Table listing the default values for parameters in the simulation

Parameter	α	f_e	σ_v^2	A_t	f_t	f_s	λ	B_r
Default Value	$\frac{1}{100}$	5MHz	$\frac{1}{3\cdot 4^{10}}$	2^{-4}	100MHz	1GHz	$\frac{255}{256}$	200MHz

described under Sec. 4.2.1. Central-derivative filters are used throughout the simulation (specifically, the 11-tap filter listed in (4.47)). For the simulation of the band-limited signal, a cut-off bandwidth of $B_r = 200$MHz is chosen and the signal is generated by filtering white Gaussian noise with the use of a 256-tap FIR low-pass filter designed using the FIRLS function in MATLAB. The band-limited signal was jittered by including ten terms of the Taylor series expansion by adding $\frac{\boldsymbol{e}[n]^i}{i!}\boldsymbol{r}^{(i)}[n]$ for the i-th term and the derivative signal $\boldsymbol{r}^{(i)}[n]$ is generated by applying a central-difference discrete derivative filter to the signal $\boldsymbol{r}[n]$.

In this section, we evaluate the performance of the sinusoidal tone injection methods discussed in Sec. 4.2. The normalized PSD of the input signal $\boldsymbol{r}[n]$ is illustrated in Fig. 4.12. Figure 4.13(a) shows the performance of the estimators. Both algorithms utilize a 4-th order Butterworth IIR low-pass filter designed with

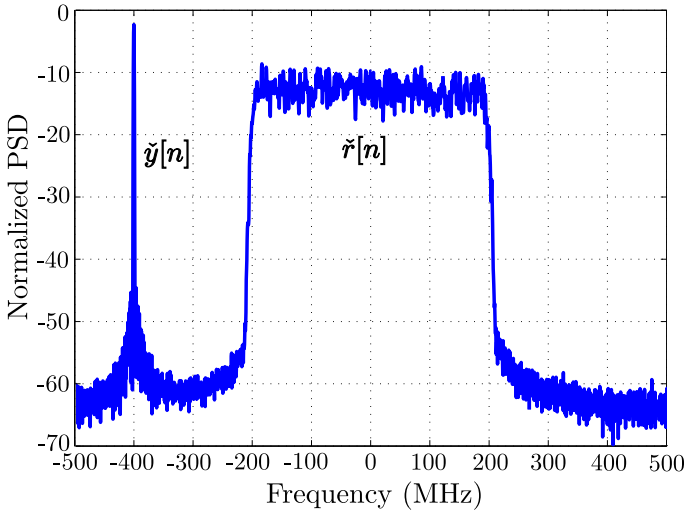
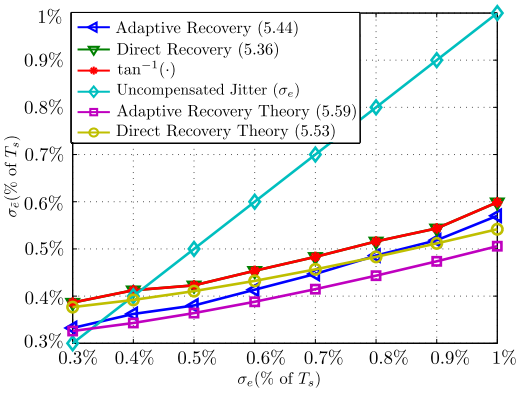


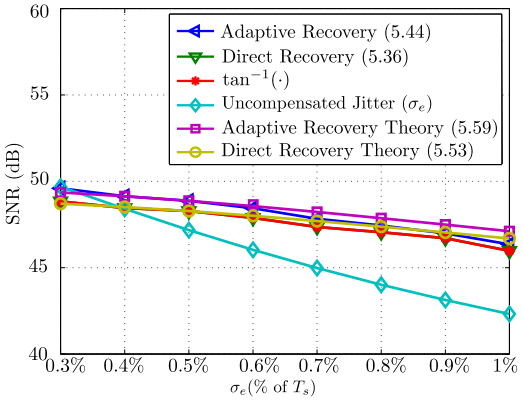
Figure 4.12: Normalized PSD of input signal $\boldsymbol{r}[n]$ used in the simulation.

appropriately chosen bandwidths and for the case of the adaptive methods, an appropriately chosen gain μ to minimize the MSE. The bandwidth of the low-pass-filter must be reduced as the initial jitter standard-deviation σ_e is decreased in general. We fix the low-pass filter in this work to provide smooth theoretical and simulation curves. The cut-off of this low-pass filter will inherently limit the performance of the filter to the performance illustrated in Fig. 4.2(b) even as the low-pass filter becomes ideal. It can be seen from Fig. 4.13(a) that there is a clear improvement in the MSE by using either of the proposed compensation methods for moderate values of σ_e. We also note that the adaptive algorithm in (4.43) outperforms the direct method in (4.35). We also compare to the estimation algorithm used in [32,33] that used a $\tan^{-1}(\cdot)$ operation to recover the jitter from (4.33). We notice that the simple estimator in (4.35) performs at the same level as the estimation algorithm that utilizes the $\tan^{-1}(\cdot)$ operation.

We use (4.64) in order to plot the theoretical SNR curves that correspond to the compensation of a box-car PSD random signal with simulation parameters listed in Table 4.1 and for a 10-bit ADC. Fig. 4.13(b) illustrates the relative



(a) Figure illustrating the MSE in estimating the jitter $\boldsymbol{e}[n]$ vs. σ_e tradeoff for the direct estimation method in (4.35) and the adaptive method in (4.43). The uncompensated jitter $\boldsymbol{e}[n]$ is also displayed.



(b) Figure illustrating the SNR after compensation vs. σ_e tradeoff for the direct estimation method in (4.35) and the adaptive method in (4.43). The SNR for the uncompensated signal is also displayed.

Figure 4.13: In (a), we illustrate the simulated and theoretical MSE in the estimation of the jitter using the direct and adaptive techniques. In (b), we illustrate the expected SNR as a result of jitter compensation.

performance of the estimators and the compensation algorithm described in Sec. 4.3 for varying values of σ_e. It can be seen from the figure that there is a clear improvement in the SNR by using either of the proposed compensation methods. We also note that the adaptive algorithm in (4.43) outperforms the direct method in (4.35). In general, the shape of the PSD of the signal $r(t)$ dictates the slope of the SNR curve as seen by the difference in the SNR curves for complex-sinusoidal signals and box-car signals in Sec. 4.1.2. The discrepancy between the theory and simulation can be explained by the fact that we use a small jitter assumption in the analysis, which is violated as σ_e increases and by leakage of jitter from $r(t)$ onto the training signal band. The latter effect is also amplified as the initial RMS jitter increases.

4.7 Concluding remarks

We illustrated the effect of sampling jitter on the spectrum of the sampled signal. We showed that the jitter can be estimated in the digital domain up to some limits and we derived bounds for the estimation accuracy. We also proposed algorithms that can estimate the jitter in direct-downconversion receivers at high-sampling rates. In addition, we proposed a compensation algorithm that was shown to be sufficient when the jitter is small. Simulations illustrate the performance of the digital jitter recovery schemes and that they can reduce the RMS jitter by half from 1% to approximately 0.53% of the sampling period and improve the SNR by 5dB over the uncompensated samples under given conditions for a box-car signal.

4.A Stochastic properties of jitter

In an ADC circuit, the sampling time instants are the random times $\boldsymbol{t}_n$ at which the clock signal $\boldsymbol{s}(t)$ crosses zero on either the positive or negative edge. In this work, we will consider sampling on the positive edge without loss of generality. This happens at the values of $\boldsymbol{t}_n$ satisfying:

$$2\pi f_s \boldsymbol{t}_n + \boldsymbol{\phi}_s(\boldsymbol{t}_n) = 2\pi n \tag{4.65}$$

for all integers n; or equivalently:

$$\boldsymbol{t}_n = nT_s - \frac{\boldsymbol{\phi}_s(\boldsymbol{t}_n)}{2\pi f_s} \tag{4.66}$$

where $T_s = \frac{1}{f_s}$ is the sampling period. Notice that $\boldsymbol{t}_n$ is random as it is affected by the random process $\boldsymbol{\phi}_s(t)$. When the PLL is treated as approximately a linear system during normal tracking behavior, it is reasonable to assume that the distribution of the phase-noise $\boldsymbol{\phi}_s(\boldsymbol{t}_n)$ is Gaussian [40, pp.135-136], [76]. For

this reason, we assume that the sampling offsets $-\frac{\boldsymbol{\phi}_s(\boldsymbol{t}_n)}{2\pi f_s}$ are Gaussian distributed with zero-mean and standard-deviation αT_s where α parameterizes the value of the RMS jitter as a percent of the sampling period T_s. In general, small values are chosen for α such as 1/100, which implies that the sampling offset has an RMS value of approximately 1% of the sampling period. In addition, since the phase-noise is shaped by the PLL, it is generally low-pass in nature [9] and, in high-speed sampling systems, the bandwidth of the phase-noise is generally much smaller than the sampling frequency of the ADC itself. We model the actual sampling times $\boldsymbol{t}_n$ as random perturbations away from the sampling instants nT_s:

$$\boldsymbol{t}_n = nT_s + \boldsymbol{e}[n] \tag{4.67}$$

where the random process $\boldsymbol{e}[n]$ represents these random perturbations. We substitute (4.67) into (4.66) to find that $\boldsymbol{e}[n]$ can be written as:

$$\boldsymbol{e}[n] = -\frac{\boldsymbol{\phi}_s\left(nT_s - \frac{\boldsymbol{\phi}_s(\boldsymbol{t}_n)}{2\pi f_s}\right)}{2\pi f_s} \tag{4.68}$$

From our Gaussian assumption regarding the distribution of the sampling time offsets $-\frac{\boldsymbol{\phi}_s(\boldsymbol{t}_n)}{2\pi f_s}$ above, it is possible to verify that the probability that the perturbations exceed a magnitude of βT_s are bounded by:

$$\Pr\left(\left|\frac{\boldsymbol{\phi}_s(\boldsymbol{t}_n)}{2\pi f_s}\right| \geq \beta T_s\right) = 2Q\left(\frac{\beta}{\alpha}\right) \tag{4.69}$$

where $Q(z) \triangleq \frac{1}{\sqrt{2\pi}} \int_z^\infty e^{-\frac{t^2}{2}} dt$ denotes the Q-function. It is easy to see that when $\alpha = \frac{1}{100}$ and $\beta = \frac{1}{10}$, the probability that the perturbation exceeds $T_s/10$ is smaller than 1.524×10^{-23}. This implies that in practice, $-\frac{\boldsymbol{\phi}_s(\boldsymbol{t}_n)}{2\pi f_s}$ is small in comparison to the sampling time and since we also assume that the PSD of phase-noise decays for large frequencies (as we will see later in this section), we may conclude that

the phase-noise is also slowly-varying in comparison to the sampling rate. These facts allow us to approximate the jitter random process $\boldsymbol{e}[n]$ as

$$\boldsymbol{e}[n] = -\frac{\boldsymbol{\phi}_s\left(nT_s - \frac{\boldsymbol{\phi}_s(\boldsymbol{t}_n)}{2\pi f_s}\right)}{2\pi f_s} \approx -\frac{\boldsymbol{\phi}_s(nT_s)}{2\pi f_s} \tag{4.70}$$

We therefore note that the relation between the PLL phase-noise and the clock jitter is linear (one is a scaled multiple of the other) [86]. The PSD of the continuous-time phase-noise process $\boldsymbol{\phi}_s(t)$ is known to be given by [9,62,87]—see App. 4.B :

$$S_\phi(f) = \frac{f^4}{f^4 + f_e^4 + 2f^2 f_e^2\,(2\zeta^2 - 1)} \cdot \frac{\beta_\phi}{f^2} \tag{4.71}$$

for some constant β_ϕ that scales the power of the process $\boldsymbol{\phi}_s(t)$ and damping factor $0 \leq \zeta \leq 1$ (for our purposes, it is sufficient to assume $\zeta = 1$ and this will simplify the expression for the jitter PSD). When $\zeta = 1$, we can simplify $S_\phi(f)$:

$$S_\phi(f) = \frac{\beta_\phi f^2}{(f^2 + f_e^2)^2} \tag{4.72}$$

Notice that although $S_\phi(0) = 0$ for the above PSD, this fact does not limit our analysis or our recovery algorithms as long as the PSD decays for large frequencies. We then conclude from (4.3) that the PSD of the clock jitter (which is approximated by the sampled process $-\boldsymbol{\phi}_s(nT_s)/(2\pi f_s)$ by (4.3)) has the form:

$$S_e(e^{j\omega}) \approx \frac{1}{(2\pi f_s)^2 T_s} \sum_{k=-\infty}^{\infty} S_\phi\left(\frac{\omega - 2\pi k}{2\pi T_s}\right) \tag{4.73}$$

over $0 \leq \omega \leq 2\pi$ and where $\beta_e = \beta_\phi/(2\pi f_s)^2$ and $\sigma_e^2 = \pi\beta_e/2f_e$. Note that the above PSD is low-pass and will decay when ω is large.

Now consider a continuous-time WSS process $\boldsymbol{x}(t)$ and its sampled process $\boldsymbol{x}[n]$. It can be shown that the autocorrelation function of $\boldsymbol{x}[n]$, denoted by $R_s(m)$ is related to the autocorrelation function of $\boldsymbol{x}(t)$, denoted by $R(\tau)$, by:

$$R_s(m) = R(\tau)\big|_{\tau = mT_s} \tag{4.74}$$

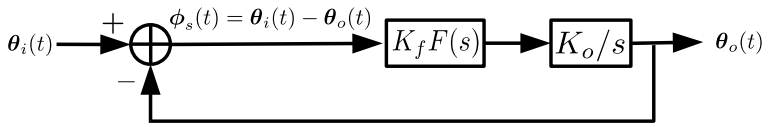


Figure 4.14: PLL phase model used to derive PSD of PLL output phase-noise

Therefore, the autocorrelation function of the jitter random process is given by the sampled inverse Fourier transform of (4.72) [75, p.1119]:

$$\begin{aligned} R_e(m) &\triangleq \int_{-\infty}^{\infty} \frac{1}{(2\pi f_s)^2} S_\phi(f) e^{j2\pi f\tau} df \Big|_{\tau = mT_s} \\ &= \sigma_e^2 (1 - 2\pi f_e \left|m\right| T_s) e^{-2\pi f_e |m| T_s} \end{aligned} \tag{4.75}$$

where any proportionality constants are absorbed into σ_e^2.

4.B Derivation of phase-noise PSD

In this appendix, we motivate the phase-noise PSD listed in (4.71) and derived in [62]. We start with the block diagram of a PLL in Fig. 4.14 where $\boldsymbol{\theta}_i(t)$ indicates the input phase, $\boldsymbol{\phi}_s(t)$ indicates the phase-noise, and $\boldsymbol{\theta}_o(t)$ indicates the output phase by the PLL. In this appendix, we set the DC component of the input phase θ_s in (4.1) to zero, and consider $\boldsymbol{\theta}_i(t)$ to be the phase-noise due to the input free-running oscillator. In this way, the phase-noise $\boldsymbol{\phi}_s(t)$ will represent the shaped phase-noise of the free-running oscillator due to the PLL. Notice that the closed-loop transfer function can be written as:

$$H(s) = \frac{KF(s)}{s + KF(s)} \tag{4.76}$$

where $K = K_f K_o$ is the total gain through the loop and $F(s)$ is the frequency response of the low-pass filter. The phase error transfer function from $\boldsymbol{\theta}_i(t)$ to $\boldsymbol{\phi}_s(t)$, denoted by $H_e(s)$, can be found as:

$$\begin{aligned} H_e(s) &= 1 - H(s) \\ &= \frac{s}{s + KF(s)} \end{aligned} \tag{4.77}$$

The phase error PSD can be found by squaring the magnitude response of $H_e(j2\pi f)$, where j is the imaginary unit $j \triangleq \sqrt{-1}$, and multiplying the result by the PSD of the input generated by the free running oscillator. A free running oscillator's phase-noise is a Wiener process with PSD proportional to $1/f^2$. Therefore, the shaped PSD of the output phase-noise can be written as:

$$S_\phi(f) = |H_e(j2\pi f)|^2 \frac{\beta_\phi}{f^2} \tag{4.78}$$

It is now possible to substitute the transfer function of a low-pass filter into (4.77) in order to find the PSD of the phase-noise output by the PLL in (4.78). We choose a low-pass filter characterized by

$$F(s) \triangleq \frac{1 + \tau_1 s}{\tau_2 s} \tag{4.79}$$

Note that $F(s)$ listed in (4.79) is the equivalent transfer function of the phase/frequency detector/charge-pump/filter cascade presented in Fig. 9.30 in [62, p.618]. There is nothing particularly special about choosing a low-pass filter of this form, but is only used here as an example. Substituting (4.79) into (4.77), we have:

$$H_e(s) = \frac{s}{s + K\frac{1+\tau_1 s}{\tau_2 s}} = \frac{s^2}{s^2 + \frac{\tau_1}{\tau_2} K s + \frac{K}{\tau_2}} \tag{4.80}$$

We now make the identifications:

$$\omega_e \triangleq 2\pi f_e = \sqrt{\frac{K}{\tau_2}}, \quad \zeta \triangleq \sqrt{\frac{K}{\tau_2}} \frac{\tau_1}{2} \tag{4.81}$$

and then we have that

$$H_e(s) = \frac{s^2}{s^2 + 2\zeta\omega_e s + \omega_e^2} \tag{4.82}$$

Notice that (4.82) is consistent with (9.51) from [62, p.639] for the analysis of phase-noise in PLLs. Finally, substituting (4.82) into (4.78) yields (4.71).

4.C Effective signal-to-noise ratio (ESNR)

To illustrate why a training signal is useful, consider the estimation of the jitter $\boldsymbol{e}[n]$ from the model

$$\check{\boldsymbol{r}}[n] \approx \boldsymbol{r}[n] + \boldsymbol{e}[n]\dot{\boldsymbol{r}}[n] + \boldsymbol{v}[n] \tag{4.83}$$

using linear least-mean-squares estimation techniques. Assume we collect data samples of $\check{\boldsymbol{r}}[n]$ into a vector $\check{\boldsymbol{r}}$ and use the data to estimate the corresponding vector of jitter samples, $\boldsymbol{e}$. The linear least-mean-squares estimator of $\boldsymbol{e}$ given $\check{\boldsymbol{r}}$ is given by the expression below in terms of the covariance matrix of $\check{\boldsymbol{r}}$ and the cross-covariance matrix between $\boldsymbol{e}$ and $\check{\boldsymbol{r}}$ [42]:

$$\hat{\boldsymbol{e}} = R_{e\check{r}} R_{\check{r}}^{-1} \check{\boldsymbol{r}} \approx R_e \mathrm{diag}\left(D\check{\boldsymbol{r}}\right) R_{\check{r}}^{-1} \check{\boldsymbol{r}} \tag{4.84}$$

where D is a differentiation matrix such as $\frac{1}{2T_s}\mathrm{toeplitz}\{[-1, 0, 1]\}$.

Note, however, from (4.83) that the jitter signal that we are interested in recovering appears in the term $\boldsymbol{e}[n]\dot{\boldsymbol{r}}[n]$ and $\boldsymbol{r}[n]$ represents a strong interferer. We can assess the power in $\boldsymbol{e}[n]\dot{r}[n]$ relative to the other terms (which can be treated as added disturbances in the problem of recovering $\boldsymbol{e}[n]$ from $\check{\boldsymbol{r}}[n]$) by defining the Effective-Signal-To-Noise-Ratio (ESNR) as

$$\mathrm{ESNR} \triangleq \frac{\mathsf{E}\left[\left|\boldsymbol{e}[n]\dot{\boldsymbol{r}}[n]\right|^2\right]}{\mathsf{E}\left|\boldsymbol{r}[n]\right|^2 + \mathsf{E}\left|\boldsymbol{v}[n]\right|^2} = \frac{\sigma_e^2\sigma_{\dot{r}}^2}{\sigma_r^2 + \sigma_v^2} \tag{4.85}$$

The above expression can be bounded for a band-limited signal $\boldsymbol{r}[n]$ (having frequencies up to B_r) through the use of Parseval's relation – see App. 4.F:

$$\mathrm{ESNR} \leq \frac{\sigma_e^2(2\pi B_r)^2\sigma_r^2}{\sigma_r^2 + \sigma_v^2} \tag{4.86}$$

Observe that even when $\sigma_v^2 \ll \sigma_r^2$, the ESNR is still bounded from above by $(2\pi\gamma_r\alpha)^2$ where $\gamma_r < \frac{1}{2}$ is a factor that parameterizes the bandwidth of the signal

$\boldsymbol{r}[n]$ as $B_r = \gamma_r f_s$. For small jitter, such as 1% of T_s, the ESNR is bounded from above by -30dB and is too small for effective estimation of $\boldsymbol{e}[n]$. For this reason, we pursue an alternative approach to recovering the jitter from the distorted data $\check{\boldsymbol{r}}[n]$. Rather than pursue a direct least-mean-squares estimation problem, we instead inject auxiliary tones along with the input signal. By examining the effect of the jitter on these embedded tones, we can then devise techniques to estimate the jitter and compensate for it.

4.D Clock recovery

The jitter estimation algorithms of Sec. 4.2 require the recovery of the phase of a sinusoid. This recovery can be done before the start of operations since the phase of the injected tones does not depend on the incoming signal $\boldsymbol{r}[n]$. Phase-locked-loops can be used for this purpose [81, 88]. Consider a sequence of the form $x[n] = \cos(2\pi f_c n T_s + \theta)$, and introduce the objective function:

$$\begin{aligned} J\left(\hat{\theta}\right) &= 2 \cdot \text{LPF}\left\{x[n]\cos(2\pi f_c n T_s + \hat{\theta})\right\} \\ &= \text{LPF}\left\{\cos(\theta - \hat{\theta}) + \cos(4\pi f_c n T_s + 2\theta)\right\} \\ &\approx \cos(\theta - \hat{\theta}) \end{aligned}$$

Thus, a steepest-ascent algorithm that maximizes $J(\hat{\theta})$ will force $\hat{\theta} \to \theta$:

$$\begin{aligned} \left.\frac{dJ\left(\hat{\theta}\right)}{d\hat{\theta}}\right|_{\hat{\theta}=\hat{\theta}(k)} &= 2 \cdot \text{LPF}\left\{x[n]\left.\frac{d}{d\hat{\theta}}\cos\left(2\pi f_c n T_s + \hat{\theta}\right)\right|_{\hat{\theta}=\hat{\theta}[n]}\right\} \\ &= 2 \cdot \text{LPF}\left\{-x[n]\sin\left(2\pi f_c n T_s + \hat{\theta}[n]\right)\right\} \end{aligned}$$

An adaptive update of the following form can be used to estimate θ recursively:

$$\hat{\theta}(n+1) = \hat{\theta}[n] - \mu\text{LPF}\left\{x[n]\sin\left(2\pi f_c n T_s + \hat{\theta}[n]\right)\right\} \tag{4.87}$$

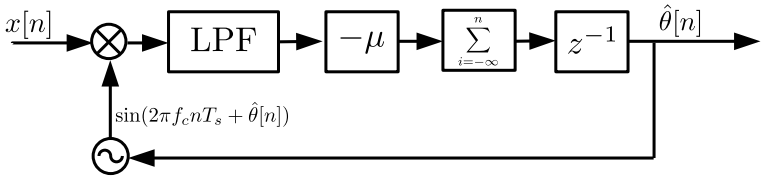


Figure 4.15: Block diagram of the phase-locked-loop

Fig. 4.15 illustrates the algorithm. Other adaptive schemes can be constructed in order to estimate θ [42].

4.E Power of N-th derivative of box-car random signal

In this appendix, we give an expression for the power of the N-th derivative of a random signal $\boldsymbol{r}(t)$ with auto-correlation function given by (4.20). We assume that the random process is mean-square integrable and that the auto-correlation function of the derivative process can be computed using (4.15), which can be extended to

$$R_{r^{(N)}(t)}(\tau) = (-1)^N \frac{d^{2N}}{d\tau^{2N}} R_r(\tau) \tag{4.88}$$

where we use the notation $\boldsymbol{r}^{(N)}(t) \triangleq \frac{d^N}{dt^N}\boldsymbol{r}(t)$. It is possible to find $\mathsf{E}\left|\frac{d^N}{dt^N}\boldsymbol{r}(t)|_{t=nT_s}\right|^2$ by finding the autocorrelation function of the discrete-time process and evaluating it at lag $m = 0$. It is known that the autocorrelation function of the discrete-time process $\frac{d^N}{dt^N}\boldsymbol{r}(t)\Big|_{t=nT_s}$ is the sampled autocorrelation function of the continuous-time process $\frac{d^N}{dt^N}\boldsymbol{r}(t)$ given by $R_{r^{(N)}}(\tau)$ [80]. This can be shown by considering a continuous-time random process $\boldsymbol{x}(t) \triangleq \frac{d^N}{dt^N}\boldsymbol{r}(t)$ with auto-correlation $R_x(\tau)$ and the discrete-time process $\boldsymbol{y}[n]$ defined by $\boldsymbol{y}[n] \triangleq \boldsymbol{x}(nT_s)$. The auto-correlation of the discrete-time process $\boldsymbol{y}[n]$ can be found as $R_y(m) \triangleq \mathsf{E}\left[\boldsymbol{y}[n]\boldsymbol{y}[n+m]\right] = \mathsf{E}\left[\boldsymbol{x}(nT_s)\boldsymbol{x}(nT_s + mT_s)\right] = R_x(mT_s)$. Thus, the two auto-correlation functions coincide at $m = 0$ and it is sufficient to find the power of the continuous-time

process $\frac{d^N}{dt^N}\boldsymbol{r}(t)$. Moreover, we can compute the power in the derivative process using:

$$\begin{aligned}\sigma^2_{r^{(N)}} = R_{r^{(N)}}(0) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{r^{(N)}}(j\Omega)d\Omega \\ &\overset{(a)}{=} \frac{(-1)^N}{2\pi}\int_{-\infty}^{\infty}(j\Omega)^{2N}S_r(j\Omega)d\Omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\Omega^{2N}S_r(j\Omega)d\Omega\end{aligned}$$

where (a) is a consequence of (4.88). We now substitute in the PSD associated with the signal with auto-correlation (4.15) – thus we substitute $S_r(j\Omega) = \frac{\sigma_r^2}{2B_r}\text{rect}\left(\frac{\Omega}{2\pi 2B_r}\right)$ where $\text{rect}(a)$ is defined in (4.19):

$$\begin{aligned}\sigma^2_{r^{(N)}} &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\Omega^{2N}S_r(j\Omega)d\Omega \\ &= \frac{\sigma_r^2}{2\pi 2B_r}\int_{-2\pi B_r}^{2\pi B_r}\Omega^{2N}d\Omega \\ &= \frac{\sigma_r^2}{2\pi B_r}\frac{(2\pi B_r)^{2N+1}}{2N+1} \\ &= \frac{(2\pi B_r)^{2N}}{2N+1}\sigma_r^2\end{aligned}$$

Thus, we conclude that

$$\mathsf{E}\left|\frac{d^N}{dt^N}\boldsymbol{r}(t)\Big|_{t=nT_s}\right|^2 = \frac{(2\pi B_r)^{2N}}{2N+1}\sigma_r^2 \tag{4.89}$$

4.F Bounding the power of the derivative of a random signal

We bound the power of the derivative of a band-limited, zero mean, mean-square differentiable WSS Gaussian random process $\boldsymbol{r}(t)$ with variance σ_r^2, autocorrelation function $R_r(\tau)$, and PSD function $S_r(j\Omega)$. We define the continuous time signal $\boldsymbol{x}(t) = \frac{d^N}{dt^N}\boldsymbol{r}(t)$ with autocorrelation function $R_x(\tau)$. The auto-correlation

$R_x(\tau)$ can be found as an extension of (4.15) to be:

$$R_x(\tau) = (-1)^N \frac{d^{2N}}{d\tau^{2N}} R_r(\tau) \tag{4.90}$$

and PSD function $S_x(j\Omega)$:

$$\begin{aligned}\sigma_x^2 = R_x(0) &\overset{(a)}{=} \frac{1}{2\pi} \int_{-\infty}^{\infty} (-1)^N (j\Omega)^{2N} S_r(j\Omega)\, d\Omega \\ &\overset{(b)}{\leq} (2\pi f_{\max})^{2N} \frac{1}{2\pi} \int_{-\infty}^{\infty} S_r(j\Omega)\, d\Omega = (2\pi f_{\max})^{2N} \sigma_r^2\end{aligned}$$

where (a) is a consequence of (4.15) and (b) is a consequence of $\boldsymbol{r}(t)$ being bandlimited to $f_{\max}$. Thus, we have shown $\sigma_x^2 \leq (2\pi f_{\max})^{2N} \sigma_r^2$.

We note that this power bound is still valid even for a sampled process $\boldsymbol{y}[n] \triangleq \boldsymbol{x}(nT_s)$ since the auto-correlation function of $\boldsymbol{y}[n]$ can be found as $R_y(m) \triangleq \mathsf{E}[\boldsymbol{y}[n]\boldsymbol{y}[n+m]] = \mathsf{E}[\boldsymbol{x}(nT_s)\boldsymbol{x}(nT_s + mT_s)] = R_x(mT_s)$. Thus the two auto-correlation functions coincide at $m = 0$ and it is sufficient to find the bound on the power of the continuous-time process $\boldsymbol{x}(t)$ as we have done above.

CHAPTER 5

Compensating Mismatches in Time-Interleaved A/D Converters

An M-channel time-interleaved analog-to-digital converter (ADC) uses M ADCs to sample an input signal to obtain a larger effective sampling rate. However, in practice, combining ADCs introduces mismatches between the various ADC channels [68]. There is also the possibility of frequency response mismatches [70,89–91]. Several techniques have been advanced in the literature to ameliorate the effect of mismatches on the performance of time-interleaved ADC (TI-ADC) implementations. For example, in [92, 93] a reference ADC is used alongside the TI-ADC structure. In [94], the input signal is assumed to have some empty frequency band and compensation for the mismatches is formulated in terms of a nonlinear least-squares problem. In [95, 96], the timing and gain mismatches are estimated by enforcing some empty frequency band constraints. In [1,15,97,98], it is assumed that the distortions appear in an out-of-band region where no signal components are present. In [99], the input signal is assumed to be sparse to enhance estimation of the mismatch parameters.

While most of these prior works rely on exploiting the existence of some empty frequency bands where no signal components occur, there are other works that pursue alternative routes. In [14], the offset, gain, and timing errors are estimated by optimizing certain loss functions. In [100, 101], an auto-correlation method is

used to detect and estimate timing errors. In [102–104], a two-step approach is implemented. First, the timing mismatch is compensated for and the residual error is computed by means of an approximate Hilbert filter. Subsequently, the residual error is used to reduce the timing error. In [2], a pseudo aliasing signal is generated using the Hadamard transform to compensate for the timing and gain mismatches in the distorted signal. The distortion parameters are estimated by re-generating the pseudo aliasing signal from the compensated signal and correlating it with the compensated signal.

In this work, we propose two solutions for the compensation of time and gain mismatches that eliminate some of the limitations of the existing approaches [20, 21]. One solution technique is implemented in the time-domain and the other solution technique is implemented in the frequency domain. The latter is shown to have significantly superior performance. The time-domain approach has some similarities with the solution methods of [1, 2] and therefore suffers from similar limitations. We explain these difficulties in the body of the chapter. In comparison, the frequency-domain approach addresses the limitations and is able to deliver enhanced performance over existing methods, even under more relaxed conditions. For example, the structure of reference [1] is based on one key assumption: the input signal needs to be oversampled and an out-of-band frequency region should be known beforehand where no signal components are present. In this band, only contributions from the distortions caused by the mistmatches are assumed to be present. By focusing on this region, adaptation can be performed to reduce the effect of the distortions. However, it is not uncommon for the conditions assumed by [1] to be violated. For instance, the input signal may be wideband and the spectrum of the error signals may not be limited to separate frequency regions that are free of signal contribution. When these conditions occur, the solution of [1] fails. The time-domain approach of the

current chapter and that of [2] are able to address some of these limitations in that they do not assume the pre-existence of an out-of-band region. While this is an advantage, it nevertheless degrades the performance of the solution relative to [1] when an out-of-band region happens to be present.

The frequency-domain approach removes this drawback altogether because it employs a structure that is able to automatically locate and exploit out-of-band regions when they are present. In this approach, we transform the data into the frequency domain by means of an $N-$point FFT operation. The data across each FFT channel then amounts to data around a certain frequency bin. In this way, the frequency space is sliced into N adjacent frequency bins. Adaptation is performed over each channel to estimate the mismatch parameters. By tracking the signal power over the various channels, the structure is able to learn which channels have more or less reliable estimates of the mismatch parameters. These estimates are then fused using an adaptive strategy [42] to obtain an enhanced final estimate for the parameters. The frequency-domain approach can therefore adapt to the input signal; it has accelerated convergence and comparable performance to [1] when the input signal contains *known* empty frequency bands, and is still able to mitigate the timing and gain mismatches in cases where [1] cannot, including situations when the input signal contains *unknown* empty frequency bands or does not contain any empty frequency band.

There are additional advantages for the frequency-domain structure. Adaptation is performed across each channel to estimate the parameter mistmatches. Interference cancelation by means of such adaptive constructions is aided by an implicit assumption of uncorrelatedness between the signal and the interference. There are a couple of steps in the frequency-domain implementation that help reduce the effect of correlation between the signal and the interference; (a) first,

data are processed in the frequency domain and not in the time domain. By means of the FFT, the signal is split into small frequency bands. Over these smaller bands, the signal component and the interference component are generally more likely to be uncorrelated with each other [105, 106]; (b) second, the solution combines the estimates of the mismatch parameters from across all channels and the combination weights are chosen in a manner that gives more relevance to estimates arising from bands where there is less frequency overlap between signal and interference (and, hence, less correlation); (c) third, the frequency structure employs a succession of two adaptation stages; the second stage exploits any correlation that is left in the data after processing by the first stage in order to further enhance the estimation accuracy of the mismatch parameters. The analysis and simulations illustrate the superior performance of this construction.

5.1 Problem formulation

The structure of the TI-ADC model in the presence of imperfections across the multiple ADC branches is already well-presented in the literature (see, e.g., [1, 68]). We briefly review the model here. Figure 5.1 shows the block diagram representation of an M-channel TI-ADC with gain and timing mismatches. Let us denote the gain and timing mismatches in the m-th channel by g_m and $r_m T_s$, where r_m represents the timing mismatch relative to the overall sampling period T_s. The input signal $x(t)$ is spilt into M channels and the $m-$th branch is multiplied by g_m and sampled at $(\ell M+m+r_m)T_s$; the resulting sequence is $z_m[\ell]$. A multiplexer (MUX) is used to combine the sampled data from all channels into the output sequence $y[n]$. The effective sampling period for $y[n]$ is seen to be T_s. In the ideal scenario, where the ADCs are perfect data converters, g_m and r_m will be 1 and 0, respectively.

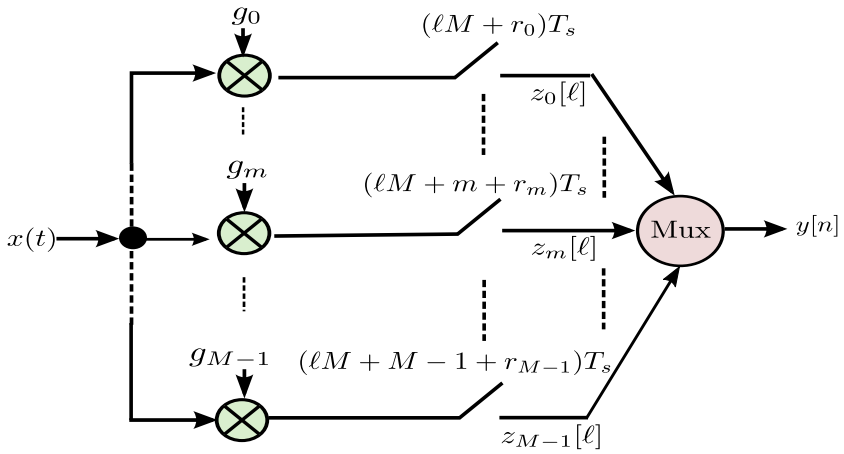


Figure 5.1: The figure shows an M-Channel TI-ADC with linear mismatches (gain and timing).

From [1], it is already known that the discrete-time Fourier transform (DTFT) of the sequence $y[n]$ can be expressed as

$$Y(e^{j\omega}) = \sum_{k=0}^{M-1} X\left(e^{j(\omega - \frac{2\pi k}{M})}\right) \check{H}_k\left(e^{j(\omega - \frac{2\pi k}{M})}\right) \tag{5.1}$$

where $X(e^{j\omega})$ is the DTFT of the sampled sequence $x[n] = x(t)|_{nT_s}$ and

$$\check{H}_k\left(e^{j\omega}\right) = \frac{1}{M} \sum_{m=0}^{M-1} g_m e^{r_m H_d(e^{j\omega})} e^{-jk\frac{2\pi}{M}m} \tag{5.2a}$$

$$H_d\left(e^{j\omega}\right) = j\omega, \text{ for } -\pi < \omega \leq \pi \tag{5.2b}$$

The derivation is shown in Section 5.B. It is clear from (5.1) that when the TI-ADC has mismatches, the output spectrum contains the original spectrum $X(e^{j\omega})$ multiplied by $\check{H}_0(e^{j\omega})$, and frequency-shifted versions of the product $X(e^{j\omega})\check{H}_k(e^{j\omega})$. Figure 5.2 illustrates the spectrum of the distorted output sequence of the TI-ADC for $M = 2, 4$. The figure shows the magnitude DTFT of the input sequence (represented by the black triangular curve centered at 0) and the magnitude DTFTs of the interfering error signals (represented by the remaining triangular curves).

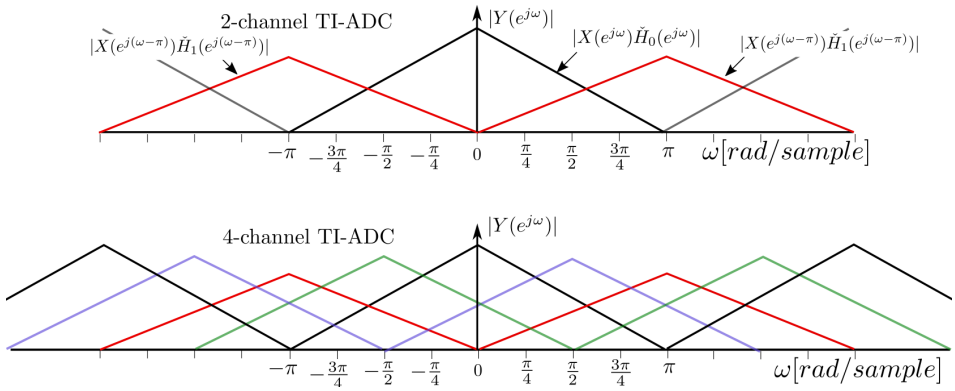


Figure 5.2: The figure shows the spectrum $Y(e^{j\omega})$ of the TI-ADC for $M = 2, 4$. The black triangle centered at 0 is the spectrum $X(e^{j\omega})\check{H}_0(e^{j\omega})$ and the remaining triangles represent the interfering error spectra generated by the mismatches. Each frequency-shifted spectrum $X(e^{j\omega})\check{H}_k(e^{j\omega})$ has a different color.

5.2 Existing techniques and limitations

As already remarked in the introduction, there have been several efforts in the literature to address the TI-ADC compensation problem with varied degrees of success and often under varying assumptions on the nature of the data. One useful approach is proposed in [1] and appears to lead to the best performance among existing techniques (when its assumptions hold). To facilitate comparison with [1], we review briefly its main contribution (using our notation) and comment on some of the limitations of the approach before moving on to develop the solution method of this work.

5.2.1 Linear approximation

Reference [1] assumes the timing mismatches r_m are small and assumes a first-order Taylor's series approximation can be applied to the term $e^{r_m H_d(e^{j\omega})}$ in (5.2a)

as

$$e^{r_m H_d(e^{j\omega})} \approx 1 + r_m H_d(e^{j\omega}) \tag{5.3}$$

Substituting (5.3) into (5.2a) gives

$$\check{H}_k\left(e^{j\omega}\right) \approx G_k + R_k H_d(e^{j\omega}) \tag{5.4}$$

where

$$G_k = \frac{1}{M}\sum_{m=0}^{M-1} g_m e^{-j\frac{2\pi}{M}mk} \tag{5.5a}$$

$$R_k = \frac{1}{M}\sum_{m=0}^{M-1} g_m r_m e^{-j\frac{2\pi}{M}mk} \tag{5.5b}$$

The variables $\{G_k, R_k\}$ contain information about the gain and delay mismatches across the branches. Observe that $G_o = \frac{1}{M}\sum_{m=0}^{M-1} g_m$ represents the average gain mismatch across all channels. It can be assumed, without loss of generality, that the average value of the timing mismatches is zero, i.e., $\frac{1}{M}\sum_{m=0}^{M-1} r_m \approx 0$. Reference [1] further assumes that R_0 is small and can be neglected. Under these conditions, the inverse discrete-time Fourier transform (IDTFT) of (5.1) is derived

$$y[n] = G_0 x[n] + e[n] \tag{5.6}$$

where all interfering terms are collected into the error signal, $e[n]$. It is shown in [1] that, when M is even, the error $e[n]$ can be expressed as the sum of two inner-product components:

$$e[n] = c_g^T x_{g,n} + c_r^T x_{r,n} \tag{5.7}$$

where the vectors $x_{g,n}$ and $x_{r,n}$ contain modulated samples of the signal $x[n]$ and its "derivative" version $h_d[n] \star x[n]$. Specifically, the notation $h_d[n]$ denotes the

impulse response sequence of a derivative filter. Therefore,

$$x_{g,n} = m_n \; x[n] \tag{5.8a}$$

$$x_{r,n} = m_n \left(h_d[n] \star x[n] \right) \tag{5.8b}$$

where m_n denotes the modulation vector of size $M-1$.

$$m_n = \begin{bmatrix} 2\cos\left(\frac{2\pi n}{M}\right) \\ -2\sin\left(\frac{2\pi n}{M}\right) \\ \vdots \\ 2\cos\left(\left(\frac{M}{2}-1\right)\frac{2\pi n}{M}\right) \\ -2\sin\left(\left(\frac{M}{2}-1\right)\frac{2\pi n}{M}\right) \\ (-1)^n \end{bmatrix} \tag{5.9}$$

The vectors $\{c_g, c_r\}$ of size $M-1$ in (5.7) are defined in terms of the real and imaginary parts of the gain and delay mismatch parameters $\{G_k, R_k\}$ as follows:

$$c_g = \begin{bmatrix} \mathrm{R}e\{G_1\} \\ \mathrm{I}m\{G_1\} \\ \vdots \\ \mathrm{R}e\left\{G_{\frac{M}{2}-1}\right\} \\ \mathrm{I}m\left\{G_{\frac{M}{2}-1}\right\} \\ G_{\frac{M}{2}} \end{bmatrix}, \quad c_r = \begin{bmatrix} \mathrm{R}e\{R_1\} \\ \mathrm{I}m\{R_1\} \\ \vdots \\ \mathrm{R}e\left\{R_{\frac{M}{2}-1}\right\} \\ \mathrm{I}m\left\{R_{\frac{M}{2}-1}\right\} \\ R_{\frac{M}{2}} \end{bmatrix} \tag{5.10}$$

In reference [1], when G_0 in (5.6) is not equal to one, the recovered signal will be $G_0 x[n]$ instead of $x[n]$; and the estimated c_g and c_r will be $\hat{c}_g = \frac{1}{G_0} c_g$ and $\hat{c}_r = \frac{1}{G_0} c_r$. Interested readers can refer to reference [1] for the derivations. For ease of notation, we assume $G_0 = 1$ in this paper, if $G_0 \neq 1$, then our recovered signal will be $G_0 x[n]$.

5.2.2 Compensation

The challenge is to recover the samples $x[n]$ in (5.6) from knowledge of $y[n]$. The error component $e[n]$ in (5.6) is not known because it depends on unknown gain and delay mismatch parameters. Reference [1] replaces $x[n]$ by $y[n]$ on the right-hand side of (5.8) and uses the following expression to estimate $e[n]$:

$$\hat{e}[n] = \hat{c}_{g,n-1}^T m_n y[n] + \hat{c}_{r,n-1}^T m_n \left(h_d[n] * y[n]\right) \tag{5.11}$$

where $\hat{c}_g$ and $\hat{c}_r$ are estimates for the vectors c_g and c_r, respectively; these estimates are computed as explained further ahead. Using (5.11), then

$$y[n] - \hat{e}[n] \approx x[n] \tag{5.12}$$

This construction leads to the compensation structure shown in Fig. 5.3.

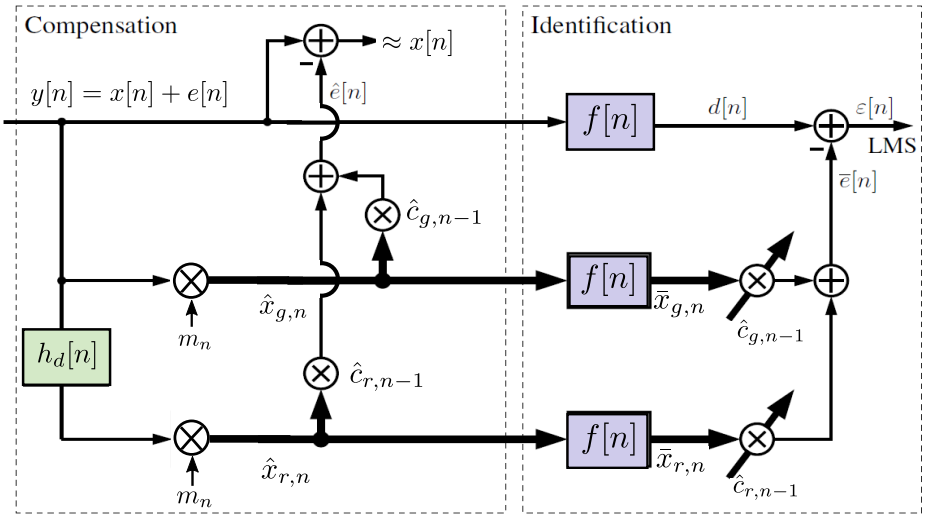


Figure 5.3: The figure shows the compensation and identification structure used in reference [1].

In order to estimate the vectors $\{c_g, c_r\}$, reference [1] assumes that the input signal is a low-pass signal and that the TI-ADC oversamples the signal. In this

way, an empty frequency band in higher frequencies will become available that is free of signal content. This frequency band will only contain contributions that arise from the error spectra due to the gain and delay mismatches. This situation is illustrated in Fig. 5.4 for $M = 2, 4$. As such, reference [1] uses a high-pass

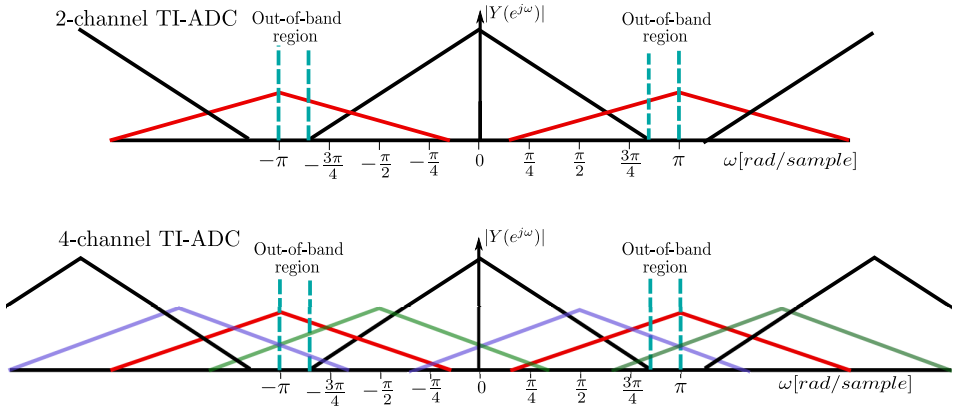


Figure 5.4: The figure shows the spectrum of the TI-ADC for $M = 2, 4$ when the input signal is oversampled. The black triangle centered at 0 is the spectrum of the original signal $x[n]$, and the remaining triangles represent the error spectra generated by the mismatches. Note that the original spectrum does not cover the entire band from $-\pi$ to π. Hence, there is an out-of-band region that contains only error spectra.

filter $f[n]$ to remove the input signal and concentrates on estimating the vectors $\{c_r, c_g\}$ in order to reduce the distortion that is present in the out-of-band region. The LMS algorithm [42] is used to estimate the parameters c_g and c_r as follows. Referring to Fig. 5.3, the estimated error signal, $\hat{e}[n]$ is high-pass filtered by $f[n]$ and used to generate an error component $\epsilon[n]$ to drive the adaptation process. The estimated vectors $\hat{x}_{g,n}$ and $\hat{x}_{r,n}$ are also filtered through $f[n]$ to generate $\{\bar{x}_{g,n}, \bar{x}_{r,n}\}$. These quantities are then used in adapting $\{\hat{c}_{r,n}, \hat{c}_{r,n}\}$:

$$\hat{c}_{g,n} = \hat{c}_{g,n-1} + \mu \bar{x}_{g,n}^T \epsilon[n] \tag{5.13a}$$

$$\hat{c}_{r,n} = \hat{c}_{r,n-1} + \mu \bar{x}_{r,n}^T \epsilon[n] \tag{5.13b}$$

5.2.3 Limitations

Three of the main limitations of the approach described so far is that it assumes that (a) the input signal is oversampled, (b) there exists an out-of-band-region that is influenced solely by the error spectra, and (c) the location of the out-of-band region is known beforehand. When this happens, adaptation can run over this region alone to estimate the gain and delay mismatches. However, it is not difficult to see that even when the input signal has a limited bandwidth, it does not necessarily follow that all components of the error spectra will fall into an out-of-band-region. Cases (a) and (c) in Fig. 5.5 show that it is possible that some or all distortion components (the red small triangles) lie completely within the original signal bandwidth (the black large triangle). In case (b), while the

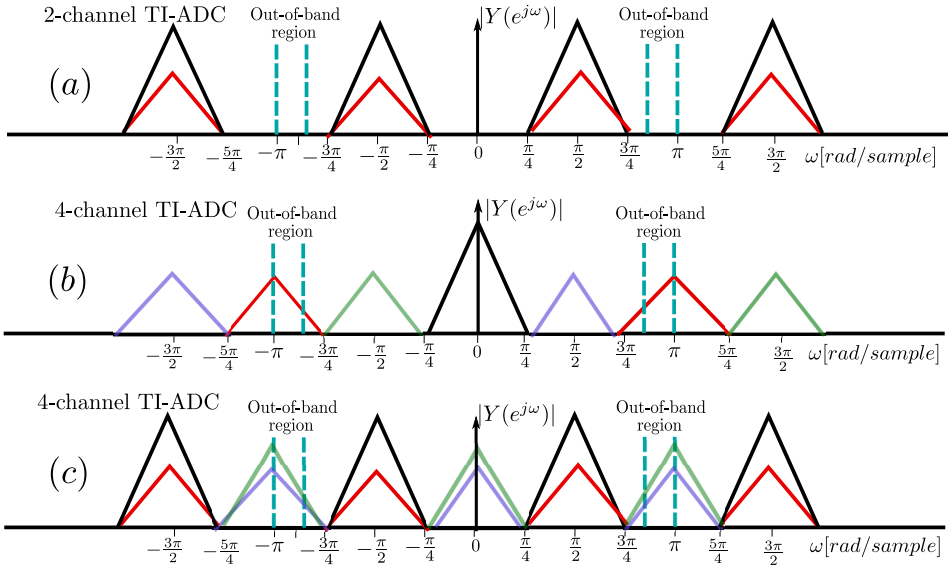


Figure 5.5: The figure shows the spectrum of the TI-ADC for $M = 2, 4$. For (a) and (c), the black large triangles centered at $\pm\frac{\pi}{2}$ are the original spectrum; whereas, for (b), the original spectrum is centered at 0. The remaining smaller triangles are components of the error spectra due to mismatches.

error components do not interfere with the signal bandwidth, only part of them lie within the out-of-band region. In these scenarios, the algorithm of [1] will not be able to remove the distortions and recover the desired signal. The techniques proposed in the current work do not have these limitations. In particular, they do not rely on the use of any out-of-band region for adaptation. This is achieved by processing data in the frequency domain and by using adaptation strategies to combine information from across frequency bins to carry out the desired compensation. When an out-of-band region exists, we will see that our proposed method performs similarly to [1]. When an out-of-band region does not exist, our proposed method will continue to perform well while the method of [1] will not be suitable for such situations.

5.3 Proposed solutions

Two solutions are proposed in this section. One solution is in the time-domain and is able to address more scenarios than described so far. The second solution is in the frequency domain and leads to superior performance.

5.3.1 Time-domain solution

Using (5.6) and (5.7), we can write

$$y[n] = \underbrace{\begin{bmatrix} x_{g,n}^T & x_{r,n}^T \end{bmatrix}}_{\triangleq u_n^T} \underbrace{\begin{bmatrix} c_g \\ c_r \end{bmatrix}}_{\triangleq w^o} + x[n] \tag{5.14a}$$

$$= u_n^T w^o + x[n] \tag{5.14b}$$

which expresses the output signal $y[n]$ as the sum of two components: the desired clean signal, $x[n]$, and the linear regression term, $u_n^T w^o$. We now refer to the basic

structure for adaptive interference cancelation shown in Fig. 5.6. In traditional

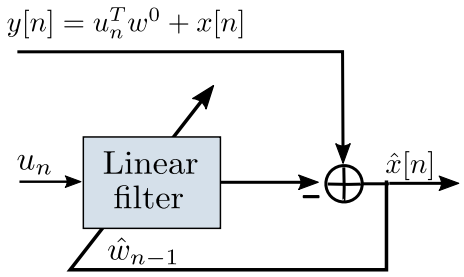


Figure 5.6: An adaptive structure for interference cancelation.

adaptive filtering, it is customary for the term $x[n]$ to represent the noise component when we estimate w^o. In the current setting, though, $x[n]$ represents the signal component that we wish to estimate and retain. This can be accomplished by using LMS adaptation to estimate w^o from knowledge of $\{y[n], u_n\}$ (5.14):

$$w_n = w_{n-1} + \mu u_n^T (y[n] - u_n^T w_{n-1}) \tag{5.15}$$

The main challenge in running this algorithm is the need to know the regression data $\{u_n\}$; this data depends on the vectors $\{x_{g,n}, x_{r,n}\}$, which in turn depend on the unknown signal $x[n]$. In the sequel, we present two methods for estimating these quantities and replacing u_n by

$$u_n = \begin{bmatrix} \hat{x}_{g,n}^T & \hat{x}_{r,n}^T \end{bmatrix}^T \tag{5.16}$$

We first describe a time-domain solution. We again substitute $x[n]$ by $y[n]$ in (5.8) to approximate the vectors $\{x_{g,n}, x_{r,n}\}$ by

$$\hat{x}_{A,g,n} = m_n y[n], \quad \hat{x}_{A,r,n} = m_n (h_d[n] * y[n]) \tag{5.17}$$

This is the same approximation that was employed earlier in (5.11). Due to the approximation, the estimate for $x(n)$ that results from this approximation in Fig.

5.6 can be refined further. We denote the initial estimate by $\hat{x}_A[n]$ in the left-part of Fig. 5.7. We then use it to feed another similar interference cancelation structure to refine it into $\hat{x}_B[n]$. The net effect is the cascade structure shown in Fig. 5.7. We will illustrate the performance of this mechanism in a later section. We move on to describe the second solution method, which is in the frequency domain.

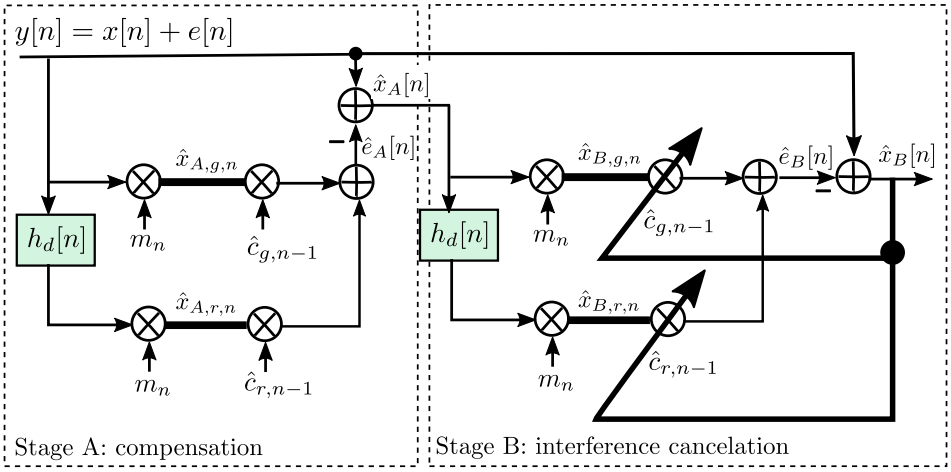


Figure 5.7: Block diagram representation of the proposed time-domain solution.

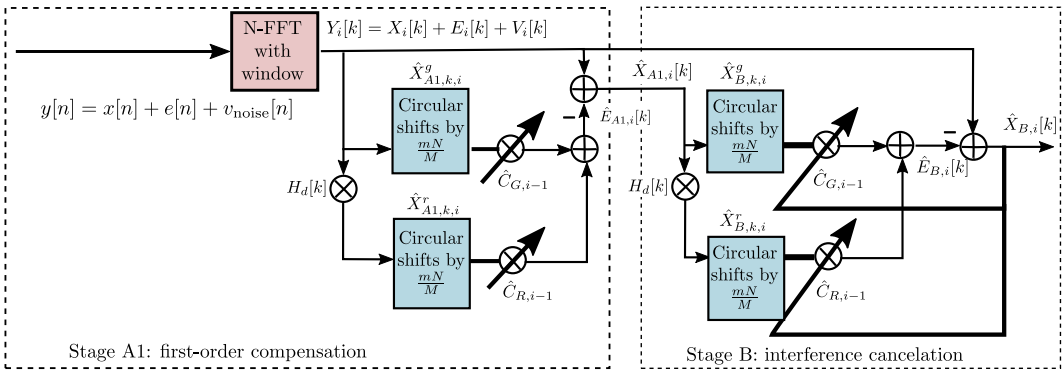


Figure 5.8: Block diagram representation of the frequency-domain solution based on a first-order compensation stage, where $H_d[k]$ is defined in (5.19).

5.3.2 Frequency-domain solution

The time-domain solution removes the limitations of the previous work. However, it ignores the possibility that the input signal may contain out-of-band regions with only distortion components and that these regions could be exploited for additional enhancements. This observation motivates an alternative approach where data are processed in the frequency domain. By partitioning the data spectrum into smaller frequency bands, it becomes possible to search the frequency content of these bands to detect the existence of out-of-band regions. The frequency-domain approach assesses the relevance of the various frequency bands to the estimation task and this information is shared across the frequency domain filters. Through an adaptive process, the information is aggregated and exploited to assign more or less relevance to bands that carry more or less information about signal content and error content. Bands that help reduce the error components are given more weight to drive the adaptation process more steadily towards its aim. Since signals and their frequency contents can change over time, the adaptive process is able to evolve and track these variations continuously.

The block diagram of the frequency-domain approach is shown in Fig. 5.8. A block of data of size N is collected and windowed before undergoing an FFT transformation of size N. This step results in N bins denoted by $Y_i[k] \approx X_i[k] + E_i[k]$, for $k = 0, 1, \ldots, N-1$. The subscript i is used to denote successive blocks of data. For each frequency bin, we employ a structure similar to Fig. 5.7 consisting of a compensation block followed by a cancellation block. The compensation block is used to estimate the first order error $E_i[k]$ and to generate an initial estimate for $X_i[k]$ (denoted by $\hat{X}_{A1,i}[k]$). This information is then fed into a subsequent cancellation block to generate $\hat{X}_{B,i}[k]$. In the model shown in Fig. 5.8 we are including an additional noise variable, denoted by $v_{noise}[n]$ in

the time-domain and by $V_i[k]$ in the frequency domain. Recall from (5.3) and (5.5) that a first-order Taylor series approximation is applied to the model (5.2a). Therefore, this noise term is useful to model the effect of unmodeled dynamics. We also use $V_i[k]$ to include the effects of spectral leakage. $v_{noise}[n]$ and $V_i[k]$ can also be used to model other sources of noise, for example, thermal noise and quantisation noise. Observe from Fig. 5.8 that the adaptive compensation and cancellation tasks across each bin are now performed at a processing rate that is N times smaller than the rate of the time-domain solution in Fig. 5.7. In this way, the amount of computations involved per unit time in both implementations remains essentially invariant.

We now describe the frequency-domain solution in greater detail. First, the distorted signal $y[n]$ undergoes an $N-$point FFT after windowing by a Blackman-Harris function. This step results in N frequency bins or channels, and each bin is represented by $Y_i[k]$, where $k = 0, 1, \ldots, N = 1$ denotes the $k-$th bin and i denotes the i-th FFT block or slice. Expanding (5.1) using (5.4), and assuming $R_0 = 0$ and $G_0 = 1$, the discrete Fourier Transform (DFT) can be obtained by sampling the DTFT at $\omega = \frac{2\pi k}{N}$:

$$\begin{aligned} Y[k] &\triangleq Y(e^{j\omega})|_{\omega=\frac{2\pi k}{N}} \\ &\approx X[k] + \sum_{m=1}^{M-1} \Bigg\{ G_m X\left[\mathrm{mod}\left(k - \frac{mN}{M}, N\right)\right] \\ &\qquad + R_m H_d\left[\mathrm{mod}\left(k - \frac{mN}{M}, N\right)\right] \\ &\qquad X\left[\mathrm{mod}\left(k - \frac{mN}{M}, N\right)\right] \Bigg\} \end{aligned} \tag{5.18}$$

where $X[k] = X(e^{j\omega})|_{\omega=\frac{2\pi k}{N}}$, $H_d[k] = H_d(e^{j\omega})|_{\omega=\frac{2\pi k}{N}}$. The function $r = \mathrm{mod}(k, N)$ returns integers in the range $0 \leq r \leq N-1$, and these correspond to the remainder of dividing k by N. From (5.18), the term $\frac{N}{M}$ in the modulus function must be an integer, i.e., N must be some multiple of M. We see that the first term in

(5.18) is the original signal and the terms in the summation are the distortions. In (5.18), multiplying $H_d[k]$ with $X[k]$ can be interpreted as finding the discrete-time derivative of the time-domain signal by modeling the time-domain signal as a trigonometry polynomial and applying a spectral differentiation approach in [107, p. 23]. Using (5.2b), $H_d[k]$ is given by:

$$H_d[k] = \begin{cases} j\frac{2\pi k}{N} & \text{if } 0 \leq k \leq \frac{N}{2} - 1 \\ 0 & \text{if } k = \frac{N}{2} \\ j\frac{2\pi(k-N)}{N} & \text{if } \frac{N}{2} + 1 \leq k \leq N - 1. \end{cases} \tag{5.19}$$

Note that $H_d[k]$ has the conjugate symmetry property $H_d[k] = (H_d[N-k])^*$. Using (5.18), we can write $Y_i[k]$ as

$$Y_i[k] = X_i[k] + E_i[k] + V_i[k] \tag{5.20}$$

where $X_i[k]$ represents the output when the original signal $x[n]$ goes through the N-FFT block with the same windowing function, $E_i[k]$ represents first-order distortion terms, and $V_i[k]$ contains the spectral leakage due to windowing, unmodeled higher-order dynamics and other sources of system noise like thermal noise and quantization noise. We can express $E_i[k]$ compactly as follows: Firstly, we define the differentiation operator $\dot{X}_i[\ell]$ on the block of data (for example, on a block of $X_i[\ell], \ell = \{0, ..., N-1\}$) as

$$\dot{X}_i[k] \triangleq H_d[k] X_i[k] \tag{5.21}$$

where $H_d[k]$ is defined in (5.19). Secondly, we let

$$\alpha_{k,m} \triangleq \text{mod}\left(k - \frac{(m+1)N}{M}, N\right) \tag{5.22}$$

Next, we define the following vectors:

$$X^g_{k,i} \triangleq \begin{bmatrix} X_i[\alpha_{k,0}] & ... & X_i[\alpha_{k,M-2}] \end{bmatrix}^T \tag{5.23a}$$

$$X^r_{k,i} \triangleq \begin{bmatrix} \dot{X}_i[\alpha_{k,0}] & ... & \dot{X}_i[\alpha_{k,M-2}] \end{bmatrix}^T \tag{5.23b}$$

We can interpret each element in the vectors $X^g_{k,i}$ and $X^r_{k,i}$ as a circular shift of $X_i[k]$ and $\dot{X}_i[k]$ by multiples of $\frac{N}{M}$. Hence, we can write $E_i[k]$ in (5.20) as

$$E_i[k] = \left(X^g_{k,i}\right)^T C_G + \left(X^r_{k,i}\right)^T C_R \tag{5.24a}$$

$$C_G = \left[\begin{array}{cccc} G_1 & G_2 & ... & G_{M-1} \end{array} \right]^T \tag{5.24b}$$

$$C_R = \left[\begin{array}{cccc} R_1 & R_2 & ... & R_{M-1} \end{array} \right]^T \tag{5.24c}$$

Using (5.24), the compensation block (labeled as Stage A1) in Fig. 5.8 estimates the first-order distortion $\hat{E}_{A1,i}[k]$ and subtracts it from $Y_i[k]$ to obtain an estimate for the original signal, $\hat{X}_{A1,i}[k]$:

$$\hat{X}_{A1,i}[k] = Y_i[k] - \hat{E}_{A1,i}[k] \tag{5.25}$$

Similar to the time-domain approach, the distortion $\hat{E}_{A1,i}[k]$ is approximated using $Y[k]$ in place of $X[k]$ in (5.24) as

$$\hat{E}_{A1,i}[k] = \left(\hat{X}^g_{A1,k,i}\right)^T \hat{C}_{G,i-1} + \left(\hat{X}^r_{A1,k,i}\right)^T \hat{C}_{R,i-1} \tag{5.26}$$

where $\hat{X}^g_{A1,k,i}$ and $\hat{X}^r_{A1,k,i}$ are obtained from (5.21) and (5.23) by replacing $X_i[\cdot]$ by $Y_i[\cdot]$.

$$\hat{X}^g_{A1,k,i} \triangleq \left[\begin{array}{cccc} Y_i\left[\alpha_{k,0}\right] & ... & Y_i\left[\alpha_{k,M-2}\right] \end{array} \right]^T \tag{5.27a}$$

$$\hat{X}^r_{A1,k,i} \triangleq \left[\begin{array}{cccc} \dot{Y}_i\left[\alpha_{k,0}\right] & ... & \dot{Y}_i\left[\alpha_{k,M-2}\right] \end{array} \right]^T \tag{5.27b}$$

The terms $\hat{C}_{G,i-1}$ and $\hat{C}_{R,i-1}$ are estimates for the vectors C_g and C_r computed at the $(i-1)-$th iteration in a manner described further ahead. In the next stage (labeled as stage B: interference cancelation) in Fig. 5.8, the estimated regressors $\hat{X}^g_{B,k,i}$ and $\hat{X}^r_{B,k,i}$ are generated in a similar manner as $\hat{X}^g_{A1,k,i}$ and $\hat{X}^r_{A1,k,i}$ in the first stage. They are obtained from (5.21) and (5.23) by replacing $X_i[\cdot]$ by $\hat{X}_{A1,i}[\cdot]$. The output from stage B is the recovered signal $\hat{X}_{B,i}[k]$:

$$\hat{X}_{B,i}[k] = Y_i[k] - \hat{E}_{B,i}[k] \tag{5.28}$$

where $\hat{E}_{B,i}[k]$ is obtained in a similar manner as (5.26), where we replace $\hat{X}^g_{A1,k,i}$ and $\hat{X}^r_{A1,k,i}$ by $\hat{X}^g_{B,k,i}$ and $\hat{X}^r_{B,k,i}$, respectively.

$$\hat{E}_{B,i}[k] = \left(\hat{X}^g_{B,k,i}\right)^T \hat{C}_{G,i-1} + \left(\hat{X}^r_{B,k,i}\right)^T \hat{C}_{R,i-1} \tag{5.29}$$

We explain further ahead how the regressors $\hat{X}^g_{B,k,i}$ and $\hat{X}^r_{B,k,i}$ are used to estimate $\hat{C}_{G,i}$ and $\hat{C}_{R,i}$ from $Y_i[k]$ — see (5.33) and (5.34).

We now describe the details of the adaptation process and how estimates from across different frequency bins are aggregated. We do so by explaining the analogy with adaptive networks where nodes cooperate with their neighbors to improve their estimates [108–110]. Adaptive networks deal generally with arbitrary topologies linking their nodes. In the context of the current problem, each bin plays the role of an agent and its neighbors are the remaining bins, i.e., in this case, it is sufficient to assume that we have a fully connected mesh topology as shown in Fig. 5.9. The adaptation process is described as follows. We adjust

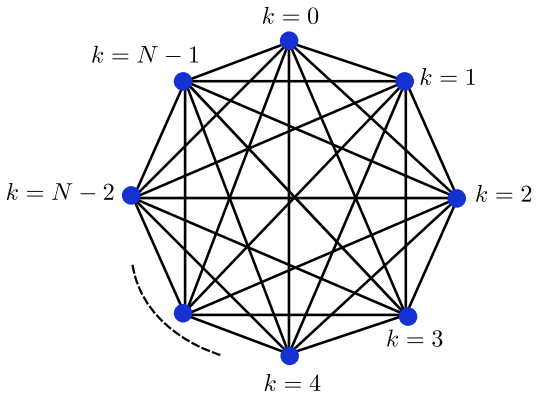


Figure 5.9: Mesh network representing the interactions among all N frequency bins for $k = 0, 1, \ldots, N-1$.

the notation to emphasize the role that is played by the different frequency bins. We now denote $Y_i[k]$ in (5.20) by $\boldsymbol{D}_k[i]$, with the subscript k emphasizing that

we are now examining data collected over time at bin k. We then observe that we can rewrite the data model for the $k-$th node (or bin) in the $i-$th FFT block as:

$$\begin{aligned}\boldsymbol{D}_k[i] \triangleq Y_i[k] &= E_i[k] + \underbrace{X_i[k] + V_i[k]}_{\boldsymbol{Z}_k[i]} \\ &= \boldsymbol{U}_{k,i} W^o + \boldsymbol{Z}_k[i] \end{aligned} \tag{5.30}$$

The linear regression, $\boldsymbol{U}_{k,i}W^o$, is the mismatch distortion, $E_i[k]$. Moreover, $X_i[k]$ and $V_i[k]$ are combined together as $\boldsymbol{Z}_k[i]$. We assume that $V_i[k]$ is small relative to $X_i[k]$, therefore $Z_k[i] \approx X_i[k]$. Hence, if the distortion $E_i[k] = \boldsymbol{U}_{k,i}W^o$ is estimated correctly and removed from $Y_i[k]$, we can recover our desired signal $X_i[k]$ at the output of Stage B. From (5.24), we collect the entries of $\{C_R, C_G\}$ into the column vector W^o, and $\{X^g_{k,i}, X^r_{k,i}\}$ into the regression row vector $\boldsymbol{U}_{k,i}$, namely,

$$\boldsymbol{U}_{k,i} = \begin{bmatrix} \boldsymbol{U}^a_{k,i}[0] & ... & \boldsymbol{U}^a_{k,i}[M-2] \end{bmatrix} \tag{5.31a}$$

$$W^o = \begin{bmatrix} W^{a,o}[0] & ... & W^{a,o}[M-2] \end{bmatrix}^T \tag{5.31b}$$

where the row sub-vector $\boldsymbol{U}^a_{k,i}[m]$ and column sub-vectors $W^{a,o}[m]$ are constructed as:

$$\boldsymbol{U}^a_{k,i}[m] = \begin{bmatrix} X^g_{k,i}[m] & X^r_{k,i}[m] \end{bmatrix} \tag{5.32a}$$

$$W^{a,o}[m] = \begin{bmatrix} C_G[m] & C_R[m] \end{bmatrix}^T \tag{5.32b}$$

The sub-vectors $W^{a,o}[m]$ are expressed in this manner since we will linearly combine the intermediate sub-vectors $\psi^a_{\ell,i}[m]$ to estimate $W^{a,o}[m]$ in the next step (see (5.33b) and (5.33c)). Each bin k now uses its own data $\{\boldsymbol{D}_k[i], \boldsymbol{U}_{k,i}\}$, arising from the k-th frequency bin, and estimates from the other nodes arising from data

in their frequency bins, to estimate W^o in the following cooperative manner:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{W}_{k,i-1} + \mu_{k,i}\boldsymbol{U}_{k,i}^{*}\left(\boldsymbol{D}_k[i] - \boldsymbol{U}_{k,i}\boldsymbol{W}_{k,i-1}\right) \tag{5.33a}$$

$$\boldsymbol{\psi}_{k,i}^{a}[m] = \left[\begin{array}{cc} \boldsymbol{\psi}_{k,i}[2m] & \boldsymbol{\psi}_{k,i}[2m+1] \end{array}\right]^T \tag{5.33b}$$

$$\boldsymbol{W}_{k,i}^{a}[m] = \sum_{\ell=0}^{N-1} a_{\ell,k,i}[m]\boldsymbol{\psi}_{\ell,i}^{a}[m] \tag{5.33c}$$

$$\boldsymbol{W}_{k,i} = \left[\begin{array}{ccc} \left(\boldsymbol{W}_{k,i}^{a}[0]\right)^T & \ldots & \left(\boldsymbol{W}_{k,i}^{a}[M-2]\right)^T \end{array}\right]^T \tag{5.33d}$$

where $m = \{0, ..., M-2\}$. Similar to the time-domain approach, and since we do not have the actual $\boldsymbol{U}_{k,i}$, we use the estimated regressors $\hat{X}_{B,k,i}^{g}$ and $\hat{X}_{B,k,i}^{r}$:

$$\boldsymbol{U}_{k,i} \approx \left[\begin{array}{ccc} \hat{\boldsymbol{U}}_{k,i}^{a}[0] & \ldots & \hat{\boldsymbol{U}}_{k,i}^{a}[M-2] \end{array}\right] \tag{5.34a}$$

$$\hat{\boldsymbol{U}}_{k,i}^{a}[m] = \left[\begin{array}{cc} \hat{X}_{B,k,i}^{g}[m] & \hat{X}_{B,k,i}^{r}[m] \end{array}\right] \tag{5.34b}$$

During the first step of the implementation in (5.33a), the k-th bin (or node) starts from its existing vector estimate $\boldsymbol{W}_{k,i-1}$ and updates it to an intermediate value $\boldsymbol{\psi}_{k,i}$ using solely data from its frequency bin. All other bins are performing a similar operation and updating their estimates, $\boldsymbol{W}_{\ell,i-1}$, to the intermediate values $\boldsymbol{\psi}_{\ell,i}$. In the remaining steps of (5.33), the $k-$th bin combines in a convex manner the intermediate estimates of all other bins to obtain the sub-vector, $\boldsymbol{W}_{k,i}^{a}[m]$, of its updated weight vector $\boldsymbol{W}_{k,i}$. The scalar coefficients $\{a_{\ell,k,i}[m]\}$ are nonnegative values that add up to one; observe that they are allowed to vary with the iteration index i since we are going to select these weights adaptively as well. Each weight $a_{\ell,k,i}[m]$ can be interpreted as the amount of trust that bin k places on the weight estimate from bin ℓ at iteration i. Since some bins may correspond to frequency bands that are better suited for the compensation task (such as bands that only contain error spectra), then by allowing the algorithm to identify these bands on the fly and to adjust the combination coefficients $\{a_{\ell,k,i}[m]\}$ in real-time, more or less weight can be assigned to the data from bin ℓ depending on whether the

data is deemed to be more or less relevant to the estimation task. We explain in the sequel how these combination weights are adapted.

Recall that the adaptive algorithm (5.33) is estimating the distortion parameter W^o in (5.30). Moreover, from (5.30), we see that we are estimating W^o from $\boldsymbol{U}_{k,i}W^o$ under the presence of $\boldsymbol{Z}_k[i]$. In this context, we see that if the magnitude of $\boldsymbol{U}_{k,i}$ and $\boldsymbol{Z}_k[i]$ is large and small, respectively, the intermediate estimate $\boldsymbol{\psi}_{k,i}$ should be estimated more accurately. Hence, we would like to find a measure that emphasizes the estimates that originate from bins that have better accuracy. For this purpose, we first define the average power of the signals $\boldsymbol{Z}_k[i]$ and $\dot{\boldsymbol{Z}}_k[i]$ as:

$$\text{average power of } \boldsymbol{Z}_k[i] \triangleq P_Z[k] \tag{5.35a}$$

$$\text{average power of } \dot{\boldsymbol{Z}}_k[i] \triangleq \dot{P}_Z[k] = |H_d[k]|^2 P_Z[k] \tag{5.35b}$$

Next, we quantify the average power of each element in $\boldsymbol{U}_{k,i}$. From (5.31a) and (5.32a), we see that $\boldsymbol{U}_{k,i}$ contains $X_i[k]$ and $\dot{X}_i[k]$, which are circularly shifted by $\alpha_{k.m}$. We approximate the average power of $X_i[k]$ and $\dot{X}_i[k]$ by $P_Z[k]$ and $\dot{P}_Z[k]$, respectively. Then, we can express the average power of each element in $\boldsymbol{U}_{k,i}$ as the vector $P_{U,k}$:

$$P_{U,k} \triangleq \Big[P_k^g[0] P_k^r[0] ... P_k^g[M-2] P_k^r[M-2] \Big] \tag{5.36a}$$

$$P_k^g \approx \begin{bmatrix} P_Z\left[\alpha_{k,0}\right] & ... & P_Z\left[\alpha_{k,M-2}\right] \end{bmatrix}^T \tag{5.36b}$$

$$P_k^r \approx \begin{bmatrix} \dot{P}_Z\left[\alpha_{k,0}\right] & ... & \dot{P}_Z\left[\alpha_{k,M-2}\right] \end{bmatrix}^T \tag{5.36c}$$

We define our initial measure of the power of the regressor components, $P_k^g[m]$ and $P_k^r[m]$, versus the power of $\boldsymbol{Z}_k[i]$, $P_Z[k]$, as the ratio:

$$P_{R,k}[m] = \frac{P_k^g[m] + P_k^r[m]}{P_Z[k]} \tag{5.37}$$

Using a low-pass input signal as an illustration, we now explain the general form of $P_{R,k}[m]$ looks like, and why it is problematic to rely on $P_{R,k}[m]$ alone as a

weighting function. The average power in each bin, $P_Z[k]$, is shown in the left plot of Fig. 5.10. The taller purple blocks represent the input signal. We assume that due to the presence of noise (e.g., spectral leakage, higher order dynamics, thermal noise or quantization noise), the average noise power in the bins is non-zero and is represented by the shorter green block. The right plot shows the sum $P_Z[k] + \dot{P}_Z[k]$.

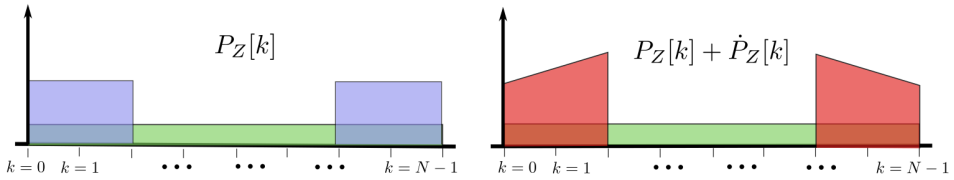


Figure 5.10: Example of $P_Z[k]$ and $P_Z[k] + \dot{P}_Z[k]$ for a low-pass signal in noise.

Continuing with the example from Fig. 5.10, $P_Z[k]$ and $P_k^g[m] + P_k^r[m]$ are shown in the top-left and middle-left plots in Fig. 5.11, respectively. The term $P_k^g[m] + P_k^r[m]$ can be obtained from the right plot in Fig. 5.10 by frequency-shifting the spectrum. The two plots show that the lower bins (closer to $k = 0$) contain large regressor components $P_k^g[m]+P_k^r[m]$, and a part of these lower bins (checkerboard region) has less contribution from the input signal and unmodeled dynamics, $P_Z[k]$. We expect the checkboard region to provide better intermediate estimates than the other regions. The ratio $P_{R,k}[m]$, in the bottom-left plot, shows that more weights are assigned to this particular region.

The main problem with using $P_{R,k}[m]$ can be seen in the same plot. We see that the higher bins contain smaller $P_k^g[m] + P_k^r[m]$, and it is less desirable to use the estimates from these bins. However, using $P_{R,k}[m]$ alone, we note that it will also emphasize some of the higher bins (i.e., polka-dot region), where the contributions of both $P_k^g[m]+P_k^r[m]$ and $P_Z[k]$ are small. To resolve this issue, we need to emphasize regions where the regressor components exists. One indicator

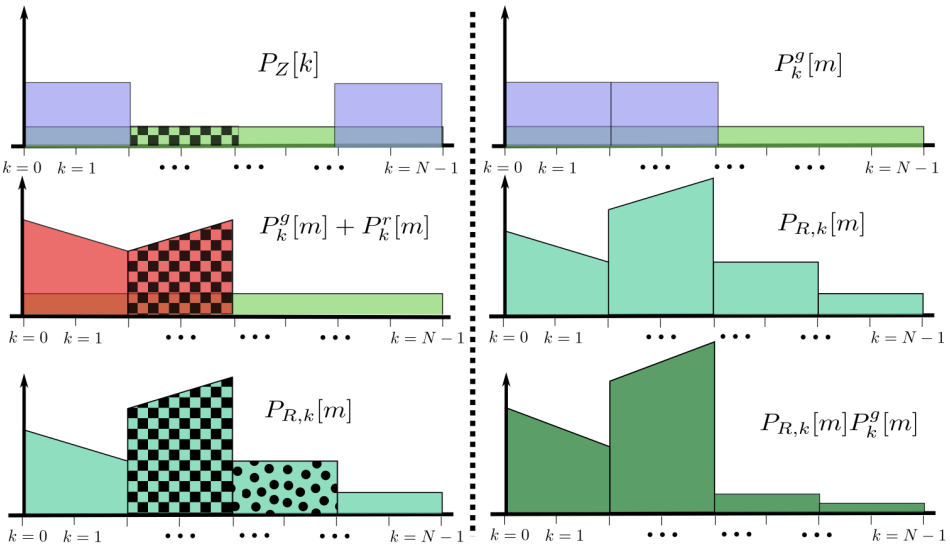


Figure 5.11: Illustration for the motivation behind the the proposed combination weights.

is $P_k^g[m]$, which is obtained by frequency-shifting $P_Z[k]$. Therefore, the new measure is the product $P_{R,k}[m]P_k^g[m]$. The top, middle and bottom plots in the right column of Fig. 5.11 show $P_k^g[m]$, $P_{R,k}[m]$ and $P_{R,k}[m]P_k^g[m]$, respectively. Using (5.36b) and (5.37), the combination weight is set as

$$a_{\ell,k}[m] = \frac{P_{R,\ell}[m]P_\ell^g[m]}{\sum_{n=0}^{N-1}\left(P_{R,n}[m]P_n^g[m]\right)} \tag{5.38}$$

where $m = \{0, ..., M-2\}$. The division operations can be implemented using look-up-table (LUT) or CORDIC algorithms [111]. Observe that $a_{\ell,k}[m]$ is in terms of $P_Z[k]$, which we do not know. One way to estimate it is [112]:

$$P_{Z,i}[k] = \begin{cases} \left|\hat{X}_{B,i}[k]\right|^2 & \text{if } i = 0 \\ \lambda P_{Z,i-1}[k] + (1-\lambda)\left|\hat{X}_{B,i}[k]\right|^2 & \text{if } i > 0 \end{cases} \tag{5.39}$$

where λ is a forgetting factor close to one (say, $\lambda = 0.95$), and $\hat{X}_{B,i}[k]$ is the recovered signal at the output of stage B in Fig. 5.8. Since we are using (5.39) in

(5.38) in each iteration, we denote the adaptive combination weight as $a_{\ell,k,i}[m]$, where we include the index i.

Observe that the combination weights in (5.38) computed in this manner are independent of k. With reference to (5.33c), this property means that after combining the intermediate estimates $\boldsymbol{\psi}_{k,i}^{a}[m]$, the estimated sub-vector $\boldsymbol{W}_{k,i}^{a}[m]$ for the k-th bin is the same for the sub-vector $\boldsymbol{W}_{\ell,i}^{a}[m]$ for all other bins, i.e., $\boldsymbol{W}_{k,i}^{a}[m] = \boldsymbol{W}_{\ell,i}^{a}[m], \ell = \{0, ..., N-1\}$. Therefore, instead of updating $\boldsymbol{W}_{k,i}$ at every node, we can update it once as $\boldsymbol{W}_{i}$. As such, we will drop the index k from $a_{\ell,k,i}$, $\boldsymbol{W}_{k,i}^{a}[m]$, and $\boldsymbol{W}_{k,i}[m]$ in (5.33). This configuration is analogous to the intermediate estimates being fused centrally to determine the aggregate estimate. In some situations, it may be known that some bins or nodes do not have any signal content, in that case it is possible to set their weights to zero and remove them from the aggregation step.

Now, we are left with the stepsize $\mu_{k,i}$ in (5.33a). The step-size needs to be sufficiently small to ensure the mean stability of the adaptation process. This can be assessed as follows (the justification is delayed to Appendix 5.A where we establish condition (5.95) further ahead). For now, we remark the following. We denote the covariance matrix of the regression vector $\boldsymbol{U}_{k,i}$ (assumed to be wide-sense stationary and zero-mean) by:

$$R_{U,k} \triangleq \mathbb{E}\left[\boldsymbol{U}_{k,i}^{*}\boldsymbol{U}_{k,i}\right] \tag{5.40}$$

We also denote the m-th non-zero eigenvalue of $R_{U,k}$ by $\lambda_m\left(R_{U,k}\right)$. Further ahead in (5.62), we will show that $R_{U,k}$ has a block diagonal matrix structure, and each block is a rank-1 matrix. Hence, $R_{U,k}$ is positive semi-definite. The non-zero eigenvalue of each such rank-1 matrix is equal to its trace. Therefore, from

(5.36), each eigenvalue is

$$\lambda_m\left(R_{U,k}\right) = P_k^g[m] + P_k^r[m] \tag{5.41}$$

The stepsize $\mu_{k,i}$ has to be bounded using the reciprocal of the maximum eigenvalue. Specifically, let

$$\lambda_{\max}\left(R_{U,k}\right) = \max\left\{\lambda_0\left(R_{U,k}\right), ..., \lambda_{M-2}\left(R_{U,k}\right)\right\} \tag{5.42}$$

Then, for mean stability, it must hold that

$$\mu_{k,i} \leq \frac{2}{\lambda_{\max}\left(R_{U,k}\right)} \tag{5.43}$$

Furthermore, recall from (5.30) that each bin is estimating W^o from $\boldsymbol{U}_{k,i}W^o$ in the presence of $\boldsymbol{Z}_k[i]$. In this sense, $\boldsymbol{Z}_k[i]$ is a noisy component in the estimation. The average power of $\boldsymbol{Z}_k[i]$ is defined in (5.35) as $P_Z[k]$. Since, $P_Z[k]$ generally differs across the bins, we are motivated to use a larger or smaller stepsize for bins that have less or more noise power, respectively. One way to do so is to set the stepsizes to be inversely proportional to $P_Z[k]$. Together with (5.43), the stepsize $\mu_{k,i}$ for the k-th node is then selected to be sufficiently small and to satisfy:

$$\mu_{k,i} = \beta \cdot \min\left\{\frac{\mu_{\text{init}}}{P_Z[k]}, \frac{2}{\lambda_{\max}\left(R_{U,k}\right)}\right\} \tag{5.44a}$$

$$\beta \leq 1 \tag{5.44b}$$

where μ_{init} is some fixed constant. Again, note that $\mu_{k,i}$ and $\lambda_{\max}\left(R_{U,k}\right)$ are in terms of $P_Z[k]$. Therefore, similar to the combination weights, we use (5.39) in (5.44) to set $\mu_{k,i}$. A summary of the steps is shown in Algorithm 5.1.

5.3.3 Enhanced frequency-domain solution

Before proceeding to analyzing the performance of the frequency-domain solution, we first highlight two sources of higher-order approximation errors that have been

Algorithm 5.1 Summary of frequency-domain solution

Require: Apply N-FFT with windowing (e.g. Blackman-Harris window) on the $i-$th block of samples to obtain $Y_i[k]$.

Let $W_{-1} = 0$, $\hat{C}_{G,-1} = 0$ and $\hat{C}_{R,-1} = 0$.

repeat

% Stage A1: Compensation

STAGE_A1$(Y_i[k], \hat{C}_{G,i-1}, \hat{C}_{R,i-1})$

% Stage B: Interference cancelation

STAGE_B$(Y_i[k], \hat{X}_{A1,i}[k], \hat{C}_{G,i-1}, \hat{C}_{R,i-1}, W_{i-1})$

until end of data blocks

Algorithm 5.1A Stage A1: first-order compensation algorithm

procedure STAGE_A1$(Y_i[k], \hat{C}_{G,i-1}, \hat{C}_{R,i-1})$

for $k = 0$ to $N - 1$ **do**

$\dot{Y}_i[k] = H_d[k] Y_i[k]$

end for

for $k = 0$ to $N - 1$ **do**

$$\hat{X}^g_{A1,k,i} = \left[\begin{array}{cccc} Y_i[\alpha_{k,0}] & ... & Y_i[\alpha_{k,M-2}] \end{array} \right]^T$$

$$\hat{X}^r_{A1,k,i} = \left[\begin{array}{cccc} \dot{Y}_i[\alpha_{k,0}] & ... & \dot{Y}_i[\alpha_{k,M-2}] \end{array} \right]^T$$

$$\hat{E}_{A1,i}[k] = \left(\hat{X}^g_{A1,k,i}\right)^T \hat{C}_{G,i-1} + \left(\hat{X}^r_{A1,k,i}\right)^T \hat{C}_{R,i-1}$$

$$\hat{X}_{A1,i}[k] = Y_i[k] - \hat{E}_{A1,i}[k]$$

end for

return $\hat{X}_{A1,i}[k]$

end procedure

Algorithm 5.1B Stage B: interference cancelation algorithm

procedure STAGE_B($Y_i[k], \hat{X}_{A1,i}[k], \hat{C}_{G,i-1}, \hat{C}_{R,i-1}, W_{i-1}$)

for $k = 0$ to $N-1$ **do**

$\hat{\hat{X}}_{A1,i}[k] = H_d[k]\hat{X}_{A1,i}[k]$

end for

for $k = 0$ to $N-1$ **do**

$\hat{X}^g_{B,k,i} = \left[\begin{array}{cccc} \hat{X}_{A1,i}\left[\alpha_{k,0}\right] & ... & \hat{X}_{A1,i}\left[\alpha_{k,M-2}\right] \end{array} \right]^T$

$\hat{X}^r_{B,k,i} = \left[\begin{array}{cccc} \hat{\hat{X}}_{A1,i}\left[\alpha_{k,0}\right] & ... & \hat{\hat{X}}_{A1,i}\left[\alpha_{k,M-2}\right] \end{array} \right]^T$

$\hat{E}_{B,i}[k] = \left(\hat{X}^g_{B,k,i}\right)^T \hat{C}_{G,i-1} + \left(\hat{X}^r_{B,k,i}\right)^T \hat{C}_{R,i-1}$

$\hat{X}_{B,i}[k] = Y_i[k] - \hat{E}_{B,i}[k]$

$$P_{Z,i}[k] = \begin{cases} \left|\hat{X}_{B,i}[k]\right|^2 & \text{if } i = 0 \\ \lambda P_{Z,i-1}[k] + (1-\lambda)\left|\hat{X}_{B,i}[k]\right|^2 & \text{if } i > 0 \end{cases}$$

$\dot{P}_{Z,i}[k] = |H_d[k]|^2 P_i[k]$

end for

for $k = 0$ to $N-1$ **do**

$P^g_{k,i} = \left[\begin{array}{cccc} P_{Z,i}\left[\alpha_{k,0}\right] & ... & P_{Z,i}\left[\alpha_{k,M-2}\right] \end{array} \right]^T$

$P^r_{k,i} = \left[\begin{array}{cccc} \dot{P}_{Z,i}\left[\alpha_{k,0}\right] & ... & \dot{P}_{Z,i}\left[\alpha_{k,M-2}\right] \end{array} \right]^T$

end for

for $k = 0$ to $N-1$ **do**

for $m = 0$ to $M-2$ **do**

$$P_{R,k,i}[m] = \frac{P^g_{k,i}[m] + P^r_{k,i}[m]}{P_{Z,i}[k]}$$

$$a_{k,i}[m] = \frac{P_{R,k,i}[m]P^g_{k,i}[m]}{\sum_{n=0}^{N-1}\left(P_{R,n,i}[m]P^g_{n,i}[m]\right)}$$

$\lambda_{m,i}\left(R_{U,k}\right) = P^g_{k,i}[m] + P^r_{k,i}[m]$

$\hat{\boldsymbol{U}}^a_{k,i}[m] = \left[\begin{array}{cc} \hat{X}^g_{B,k,i}[m] & \hat{X}^r_{B,k,i}[m] \end{array} \right]$

end for

$\lambda_{\max,i}\left(R_{U,k}\right) = \max\left\{\lambda_{0,i}\left(R_{U,k}\right), ..., \lambda_{M-2,i}\left(R_{U,k}\right)\right\}$

$$\mu_{k,i} = \beta \min\left\{\frac{\mu_{\text{init}}}{P_{Z,i}[k]}, \frac{2}{\lambda_{\max,i}\left(R_{U,k}\right)}\right\}$$

$\boldsymbol{U}_{k,i} = \left[\begin{array}{ccc} \hat{\boldsymbol{U}}^a_{k,i}[0] & ... & \hat{\boldsymbol{U}}^a_{k,i}[M-2] \end{array} \right]$

$\boldsymbol{\psi}_{k,i} = \boldsymbol{W}_{i-1} + \mu_{k,i}\boldsymbol{U}^*_{k,i}\left(Y_i[k] - \boldsymbol{U}_{k,i}\boldsymbol{W}_{i-1}\right)$

for $m = 0$ to $M-2$ **do**

$\boldsymbol{\psi}^a_{k,i}[m] = \left[\begin{array}{cc} \boldsymbol{\psi}_{k,i}[2m] & \boldsymbol{\psi}_{k,i}[2m+1] \end{array} \right]^T$

end for

end for

for $m = 0$ to $M-2$ **do**

$\boldsymbol{W}^a_i[m] = \sum_{\ell=0}^{N-1} a_{\ell,i}[m]\boldsymbol{\psi}^a_{\ell,i}[m]$

$\left[\begin{array}{cc} \hat{C}_{G,i}[m] & \hat{C}_{R,i}[m] \end{array} \right]^T = W^a_i[m]$

end for

$\boldsymbol{W}_i = \left[\begin{array}{ccc} \left(\boldsymbol{W}^a_i[0]\right)^T & ... & \left(\boldsymbol{W}^a_i[M-2]\right)^T \end{array} \right]^T$

return $\hat{X}_{B,i}[k], \boldsymbol{W}_i, \hat{C}_{G,i}[m], \hat{C}_{R,i}[m]$

end procedure

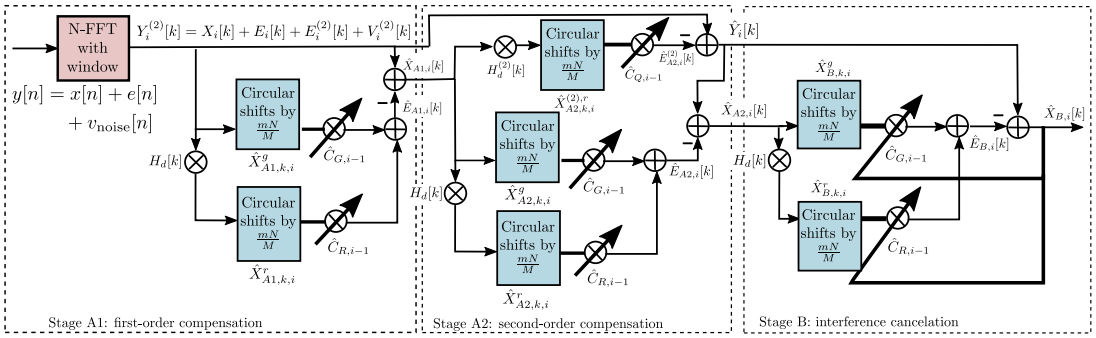


Figure 5.12: Block diagram representation of the frequency-domain solution based on a second-order compensation stage.

ignored so far at the input of the interference cancelation block in Fig. 5.8. To begin with, some higher-order errors exist in $Y_i[k]$ since we employed earlier the first-order approximation (5.3) to arrive at (5.4) and (5.5). The adaptive filtering algorithm assumes this linear model in (5.20). Similarly, there are higher-order errors in the signal $\hat{X}_{A1,i}[k]$; these errors occur when we compensate for the first-order distortion $E_i[k]$ and ignore the higher order distortions. In this section, we examine the effect of the second-order errors and propose a scheme to compensate for them, thus enhancing further the performance of the frequency-domain solution.

5.3.3.1 Second-order modelling error

Suppose we expand $e^{r_m H_d(e^{j\omega})}$ in (5.3) up to second-order as follows:

$$e^{r_m H_d(e^{j\omega})} \approx 1 + r_m H_d(e^{j\omega}) + \frac{1}{2}\left[r_m H_d(e^{j\omega})\right]^2 \tag{5.45}$$

Then, substituting (5.45) into (5.2a) gives

$$\check{H}_k\left(e^{j\omega}\right) \approx G_k + R_k H_d(e^{j\omega}) + \left[H_d(e^{j\omega})\right]^2 Q_k \tag{5.46}$$

where G_k and R_k are defined in (5.5) and

$$Q_k = \frac{1}{M}\sum_{m=0}^{M-1}\frac{g_m r_m^2}{2}e^{-j\frac{2\pi}{M}mk} \tag{5.47}$$

Repeating the derivations from (5.18) to (5.20) with the higher order error yields the following expressions. We can rewrite the output from the FFT block as

$$\begin{aligned} Y_i^{(2)}[k] &= X_i[k] + E_i[k] + E_i^{(2)}[k] + V_i^{(2)}[k] \\ &\approx X_i[k] + E_i[k] + E_i^{(2)}[k] \end{aligned} \tag{5.48}$$

where $Y_i^{(2)}[k]$ includes the input signal $X_i[k]$ and the first-order distortion $E_i[k]$ in (5.20). The second-order distortion term is $E_i^{(2)}[k]$. Recall that $V_i[k]$ in (5.20) represents the spectral leakage and higher order error terms (starting from second-order and up). Here, $V_i^{(2)}[k]$ in (5.48) represents the spectral leakage and higher order error terms (starting from the third-order and up). In a manner similar to the vector notation in (5.23) and (5.24), $E_i^{(2)}[k]$ can be expressed as:

$$E_i^{(2)}[k] = \left(X_{k,i}^{(2),r}\right)^T C_Q \tag{5.49a}$$

$$C_Q = \begin{bmatrix} Q_0 & Q_1 & \ldots & Q_{M-1} \end{bmatrix}^T \tag{5.49b}$$

where

$$X_{k,i}^{(2),r} \triangleq \begin{bmatrix} \ddot{X}_i[k] & \ddot{X}_i[\alpha_{k,0}] & \ldots & \ddot{X}_i[\alpha_{k,M-2}] \end{bmatrix}^T \tag{5.50a}$$

$$\ddot{X}_i[k] \triangleq (H_d[k])^2 X_i[k] \tag{5.50b}$$

Note that the size of the vectors here is M, while the size of the vectors in (5.23) and (5.24) is $M-1$. Therefore, we see that the input to the interference cancelation block includes an additional term $E_i^{(2)}[k]$, and we expect the performance to improve if we remove it. We see from (5.49) that $E_i^{(2)}[k]$ is determined by the $\{Q_k\}$. From (5.47), we see that Q_k is the DFT of the sequence

of $\{g_0 r_0^2,\ ...,\ g_{M-1} r_{M-1}^2\}$ scaled by $\frac{1}{2M}$. Therefore, we can estimate Q_k if $g_m r_m^2$ is known.

Recall from (5.33d) and (5.32b) that $\boldsymbol{W}_{k,i}[m]$ are the estimated $C_G[m]$ and $C_R[m]$ for $m = 0, ..., M-2$ at time i; and from (5.24), $C_G[m]$ and $C_R[m]$ are G_{m+1} and R_{m+1}, respectively. From (5.5), we see that G_k and R_k are the DFT of the sequences of $\{g_0,\ ...,\ g_{M-1}\}$ and $\{g_0 r_0,\ ...,\ g_{M-1} r_{M-1}\}$ scaled by $\frac{1}{M}$. Using the estimated G_{m+1} and R_{m+1} and the assumption that $G_0 = 1$ and $R_0 \approx 0$, we can estimate g_m and $g_m r_m$ at time i using the inverse discrete Fourier transform (IDFT) and scaling it by M:

$$\hat{g}_{m,i} = \sum_{k=0}^{M-1} \hat{G}_{k,i} e^{j\frac{2\pi}{M}mk} \tag{5.51a}$$

$$\hat{g}_{m,i}\hat{r}_{m,i} = \sum_{k=0}^{M-1} \hat{R}_{k,i} e^{j\frac{2\pi}{M}mk} \tag{5.51b}$$

After estimating g_m and $g_m r_m$, we can estimate $g_m r_m^2$ at time i. Subsequently, Q_k at time i is estimated using (5.47). Therefore, we can obtain a better linear model for $Y_i[k]$ in (5.20) from $Y_i^{(2)}[k]$ by removing the second order distortion:

$$\hat{Y}_i[k] = Y_i^{(2)}[k] - \hat{E}_{A2,i}^{(2)}[k] \tag{5.52}$$

where

$$\hat{E}_{A2,i}^{(2)}[k] = \left(\hat{X}_{A2,k,i}^{(2),r}\right)^T \hat{C}_{Q,i-1} \tag{5.53a}$$

$$\hat{C}_{Q,i-1} = \begin{bmatrix} \hat{Q}_{0,i-1} & \hat{Q}_{1,i-1} & ... & \hat{Q}_{M-1,i-1} \end{bmatrix}^T \tag{5.53b}$$

with $\hat{Q}_{m,i-1}$ denoting the estimated Q_m in (5.49b) at time $i-1$ and $\hat{X}_{A2,k,i}^{(2),r}$ is obtained by using the estimated $\hat{X}_{A1,i}[k]$, from the compensation block in Fig. 5.8, in place of $X_i[k]$ in (5.50b) as:

$$\hat{X}_{A2,k,i}^{(2),r} \triangleq \begin{bmatrix} \hat{\dot{X}}_i[k] & \hat{\dot{X}}_i[\alpha_{k,0}] & ... & \hat{\dot{X}}_i[\alpha_{k,M-2}] \end{bmatrix}^T \tag{5.54a}$$

$$\hat{\dot{X}}_i[k] \triangleq (H_d[k])^2 \hat{X}_{A1,i}[k] \tag{5.54b}$$

The resulting structure for removing the higher-order terms from $Y_i^{(2)}[k]$ is shown in Fig. 5.12. Comparing with the original solution in Fig. 5.8, a new block is inserted between the original compensation and interference cancelation blocks. The top-most path shows how the second-order distortion $\hat{E}_{A2,i}^{(2)}[k]$ is generated and removed from $Y_i^{(2)}[k]$. The resultant output $\hat{Y}_i[k]$ is fed into the interference cancelation block.

5.3.3.2 Second-order compensation error

Recall from (5.26) that we are estimating the first-order distortion as $\hat{E}_{A1,i}[k]$ using $Y_i[k]$ in place of $X_i[k]$. After stage A1 of the compensation block in Fig. 5.12, we can find a better estimate for the first-order distortion using $\hat{X}_{A1,i}[k]$:

$$\hat{E}_{A2,i}[k] = \left(\hat{X}_{A2,k,i}^{g}\right)^T \hat{C}_{G,i-1} + \left(\hat{X}_{A2,k,i}^{r}\right)^T \hat{C}_{R,i-1} \tag{5.55}$$

where $\hat{E}_{A2,i}[k]$ is the new estimate for $E_1[k]$ and $\hat{X}_{A2,k,i}^{g}$ and $\hat{X}_{A2,k,i}^{r}$ are obtained by replacing $X_i[\cdot]$ in (5.21) and (5.23) by $\hat{X}_{A1,i}[\cdot]$:

$$\hat{X}_{A2,k,i}^{g} \triangleq \left[\begin{array}{ccc} \hat{X}_{A1,i}\left[\alpha_{k,0}\right] & \ldots & \hat{X}_{A1,i}\left[\alpha_{k,M-2}\right] \end{array} \right]^T \tag{5.56a}$$

$$\hat{X}_{A2,k,i}^{r} \triangleq \left[\begin{array}{ccc} \hat{\hat{X}}_{A1,i}\left[\alpha_{k,0}\right] & \ldots & \hat{\hat{X}}_{A1,i}\left[\alpha_{k,M-2}\right] \end{array} \right]^T \tag{5.56b}$$

$$\hat{\hat{X}}_{A1,i}\left[k\right] = H_d[k]\hat{X}_{A1,i}\left[k\right] \tag{5.56c}$$

Using $\hat{E}_{A2,i}[k]$ and $\hat{E}_{A2,i}^{(2)}[k]$ in (5.55) and (5.53a), respectively, we obtain a better estimate for $X_i[k]$ as

$$\hat{X}_{A2,i}[k] = Y_i^{(2)}[k] - \hat{E}_{A2,i}[k] - \hat{E}_{A2,i}^{(2)}[k] \tag{5.57}$$

The structure for estimating $X_i[k]$ is shown in the middle block of Fig. 5.12. The resulting output $\hat{X}_{A2,i}[k]$ is fed into the interference cancelation block. A summary of the enchanced algorithm and the second-order compensation algorithm are shown in Algorithm 5.2.

Algorithm 5.2 Summary of enhanced frequency-domain solution

Require: Apply N-FFT with windowing (e.g. Blackman-Harris window) on the $i-$th block of samples to obtain $Y_i^{(2)}[k]$.

Let $W_{-1} = 0$, $\hat{C}_{G,-1} = 0$ and $\hat{C}_{R,-1} = 0$.

repeat

% Stage A1: 1st-order compensation

STAGE_A1$(Y_i^{(2)}[k], \hat{C}_{G,i-1}, \hat{C}_{R,i-1})$

% Stage A2: 2nd-order compensation

STAGE_A2$(Y_i^{(2)}[k], \hat{X}_{A1,i}[k], \hat{C}_{G,i-1}, \hat{C}_{R,i-1})$

% Stage B: Interference cancelation

STAGE_B$(\hat{Y}_i[k], \hat{X}_{A2,i}[k], \hat{C}_{G,i-1}, \hat{C}_{R,i-1}, W_{i-1})$

until end of data blocks

5.4 Comparison with prior work

We will compare the proposed solution against [1,2]; both of these works deal with similar scenarios. We already explained earlier the differences of our approach in relation to [1]. With regards to [2], this work relies again on a useful time-domain solution but it does not exploit the various aspects of the frequency-domain transformations that the proposed solution considers. For this reason, the approach of [2] suffers from the same limitation as the time-domain solution: it is not able to exploit fully information from frequency bands that may be free of signal components.

There are also some differences with respect to the proposed time-domain approach as well. The block diagram representation of the solution in [2] is shown in Fig. 5.13. We will give an overview of the algorithm. In [2], the authors found that the distortions can be estimated by generating some pseudo-signal

Algorithm 5.2A Stage A2: second-order compensation algorithm

Require: Generate $\hat{C}_{Q,i-1}$ from $\hat{C}_{G,i-1}, \hat{C}_{R,i-1}$ using (5.51) and (5.47).

procedure STAGE_A2($Y_i^{(2)}[k], \hat{X}_{A1,i}[k], \hat{C}_{G,i-1}, \hat{C}_{R,i-1}$)

for $k = 0$ to $N-1$ **do**

$\hat{\dot{X}}_i[k] = (H_d[k])^2 \hat{X}_{A1,i}[k]$

$\hat{\dot{X}}_{A1,i}[k] = H_d[k]\hat{X}_{A1,i}[k]$

end for

for $k = 0$ to $N-1$ **do**

$\hat{X}^{(2),r}_{A2,k,i} = \left[\begin{array}{cccc} \hat{\dot{X}}_i[k] & \hat{\dot{X}}_i[\alpha_{k,0}] & ... & \hat{\dot{X}}_i[\alpha_{k,M-2}] \end{array} \right]^T$

$\hat{E}^{(2)}_{A2,i}[k] = \left(\hat{X}^{(2),r}_{A2,k,i}\right)^T \hat{C}_{Q,i-1}$

$\hat{Y}_i[k] = Y_i^{(2)}[k] - \hat{E}^{(2)}_{A2,i}[k]$

$\hat{X}^{g}_{A2,k,i} = \left[\begin{array}{ccc} \hat{X}_{A1,i}[\alpha_{k,0}] & ... & \hat{X}_{A1,i}[\alpha_{k,M-2}] \end{array} \right]^T$

$\hat{X}^{r}_{A2,k,i} = \left[\begin{array}{ccc} \hat{\dot{X}}_{A1,i}[\alpha_{k,0}] & ... & \hat{\dot{X}}_{A1,i}[\alpha_{k,M-2}] \end{array} \right]^T$

$\hat{E}_{A2,i}[k] = \left(\hat{X}^{g}_{A2,k,i}\right)^T \hat{C}_{G,i-1} + \left(\hat{X}^{r}_{A2,k,i}\right)^T \hat{C}_{R,i-1}$

$\hat{X}_{A2,i}[k] = \hat{Y}_i[k] - \hat{E}_{A2,i}[k]$

end for

return $\hat{Y}_i[k], \hat{X}_{A2,i}[k]$

end procedure

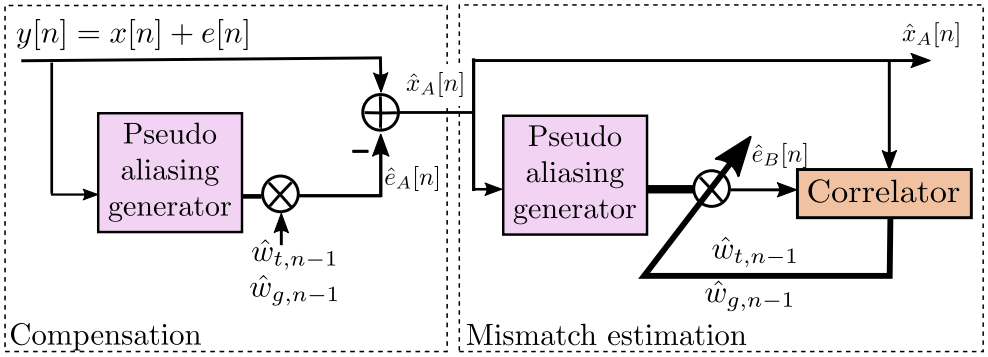


Figure 5.13: Block diagram representation of algorithm in reference [2].

and multiply it by some parameters w_t and w_g. In their solution in Fig. 5.13, the estimated parameters at $n-1$ are denoted by $\hat{w}_{t,n-1}$ and $\hat{w}_{g,n-1}$, and the estimated distortion is $\hat{e}_A[n]$. In reference [2], the distortion parameters w_g and w_t are related to the gain offsets g_m and time offsets r_m as follows:

$$w_g \triangleq \begin{bmatrix} w_{g1} & \dots & w_{g(M-1)} \end{bmatrix}^T \tag{5.58a}$$

$$w_t \triangleq \begin{bmatrix} w_{t1} & \dots & w_{t(M-1)} \end{bmatrix}^T \tag{5.58b}$$

where w_{tk} and w_{gk} are

$$\begin{bmatrix} w_{g0} \\ \vdots \\ w_{g(M-1)} \end{bmatrix} \approx \frac{1}{M} F_H \begin{bmatrix} (1-g_0) \\ \vdots \\ (1-g_{M-1}) \end{bmatrix} \tag{5.59a}$$

$$\begin{bmatrix} w_{t0} \\ \vdots \\ w_{t(M-1)} \end{bmatrix} \approx \frac{1}{M} F_H \begin{bmatrix} r_0 \\ \vdots \\ r_{M-1} \end{bmatrix} \tag{5.59b}$$

and F_H denote the Hadamard matrix [2]. Suppose $M = 4$, then F_H is

$$F_H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{5.60}$$

Note that w_{g0} and w_{t0} in (5.59) are the average of $1 - g_m$ and r_m, respectively. In [2], both w_{g0} and w_{t0} are 0 as the authors assumed that the average of the gain and time mismatches are 1 and 0, respectively.

The recovered signal $\hat{x}_A[n]$ is obtained by subtracting $\hat{e}_A[n]$ from $y[n]$. The pseudo-signal is generated using the Hadamard transform. This is similar to the first-stage of the proposed time-domain solution and frequency-domain solution in Figs. 5.7 and 5.8, respectively, where we create the distortions and remove them from the distorted samples. The difference in our solutions is that our estimated distortions are not from a pseudo-signal as in [2]; the distortions are estimated by frequency-shifting the signal instead.

Another difference is in the second stage where the parameters are estimated. In [2], the authors use the recovered signal $\hat{x}_A[n]$ from the first stage to generate a new pseudo-signal and correlate it with $\hat{x}_A[n]$. Their motivation is that if $\hat{x}_A[n]$ is free from distortions, then the new pseudo-signal should be uncorrelated with $\hat{x}_A[n]$. Adjustments are made to the estimated parameters using the correlator output. For our proposed solutions, our motivation is to use successive cancelation where we use the recovered signal in the first stage as an input to the interference cancelation block to obtain a cleaner signal, $\hat{x}_B[n]$ or $\hat{X}_{B,i}[k]$. Observe also that the recovered signal in the first stage of the proposed time-domain solution uses the same implementation as [1]. In Section 5.6.2, where we compare the previous works using simulations, we calculate the SNR of the recovered

signal in both stages of our proposed solutions. We find that the SNR from the final stage is better.

5.5 Performance analysis

This section carries out a mean-square analysis of the behavior of the frequency-domain solution, and derives expressions that characterize its limits of performance. The analysis is carried out under some simplifying conditions on the data that follow from the fact that the DFT helps decorrelate the frequency bins for sufficiently large N. For this reason, we shall assume whenever necessary that the signal components across different frequency bins are largely uncorrelated. We shall also assume that the input signal is wide-sense stationary and has zero-mean and is uncorrelated of any measurement noise; the latter is assumed to be white. It can also be shown that when the DFT length is large, the distribution of the DFT coefficients become Gaussian, and the DFT coefficients become independent. Under these conditions, we can derive theoretical performance expressions that will be shown later in the simulations to match well the simulated performance of the frequency-domain solution. Two useful measures of performance are the mean-square-deviation (MSD), and the excess-mean-square error (EMSE). The EMSE can be used to quantify the improvement in signal-to-noise ratio (SNR) of the algorithm. These measures are defined in the sequel.

5.5.1 Assumptions on the signal and its distortions

From (5.30) and (5.31), we observe that for the k-th bin, the noise $\boldsymbol{Z}_k[i]$ is defined in terms of the FFT coefficient $X_i[k] + V_i[k]$ of the current bin, while the regressor $\boldsymbol{U}_{k,i}$ is made up of FFT coefficients from other bins. It is well-known that the

coefficients of the DFT of stationary signals are asymptotically uncorrelated [105]. Therefore, we may assume that when the FFT length N is large, both $X_i[k]$ and $V_i[k]$ in (5.20) are uncorrelated with $X_i[\ell]$ and $V_i[\ell]$, $k \neq \ell$, respectively. We also assume that $X_i[k]$ and $V_i[k]$ are zero mean. Recall further from (5.30) and (5.31) that the regressor $\boldsymbol{U}_{k,i}$ contains frequency-shifted components of $X_i[k]$. Therefore, we also assume that the $\boldsymbol{Z}_k[i]$ and $\boldsymbol{U}_{k,i}$ are uncorrelated. We also assume that the $X_i[\ell]$ and $V_i[\ell]$ are independent of each other, and $X_i[\ell]$ and $V_i[\ell]$ are wide-sense stationary and zero-mean.

Moreover, in reference [106], it has been shown that for stationary signal, its DFT coefficients can be assumed to have a Gaussian distribution when the DFT length is large. Reference [106] also showed that covariance matrix of the DFT coefficients is nearly diagonal, hence the DFT coefficients are statistically independent. In our work, we will use this stronger independent assumption (vs uncorrelated assumption) in some of our analysis (see (5.80), (5.81) and (5.82)).

5.5.2 Statistical properties of data model

Under the assumed statistical conditions, it follows that the variance of $\boldsymbol{Z}_k[i]$ in (5.30) is given by

$$\begin{aligned} \sigma_{Z,k}^2 \triangleq \mathbb{E}|\boldsymbol{Z}_k[i]|^2 &= \mathbb{E}|X_i[k]|^2 + \mathbb{E}|V_i[k]|^2 \\ &= \sigma_{X,k}^2 + \sigma_{V,k}^2 \end{aligned} \tag{5.61}$$

where $\sigma_{X,k}^2$ is the power of the input signal, and $\sigma_{V,k}^2$ is the power of the unmodeled dynamics and spectral leakage in the k-th FFT bin. Moreover, examining the vector $\boldsymbol{U}_{k,i}$ in (5.31a), we deduce that its covariance matrix is mainly block

diagonal, namely,

$$\begin{aligned} R_{U,k} &= \mathbb{E}\left[\boldsymbol{U}_{k,i}^{*}\boldsymbol{U}_{k,i}\right] \\ &= \text{diag}\{\ R_{X,\alpha_{k,0}},\ \ \ldots\ ,\ \ R_{X,\alpha_{k,M-2}}\ \} \end{aligned} \tag{5.62}$$

where

$$\begin{aligned} R_{X,\ell} &= \sigma_{X,\ell}^{2}\begin{bmatrix} 1 & H_d[\ell] \\ (H_d[\ell])^{*} & |H_d[\ell]|^2 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ (H_d[\ell])^{*} \end{bmatrix}\sigma_{X,\ell}^{2}\begin{bmatrix} 1 & H_d[\ell] \end{bmatrix} \end{aligned} \tag{5.63}$$

and $R_{X,\alpha_{k,m}}$ and $R_{U,k}$ are square matrices of size 2 by 2 and $2(M-1)$ by $2(M-1)$, respectively. Note that $R_{X,\alpha_{k,m}}$ and $R_{U,k}$ have ranks one and $M-1$, respectively. It also follows from the assumed uncorrelatedness of $X_i[k]$ and $V_i[\ell]$, for all $k \neq \ell$, that $\boldsymbol{Z}_k[i]$ is spatially white:

$$\mathbb{E}\boldsymbol{Z}_k[i]\boldsymbol{Z}_{\ell}^{*}[j] = 0,\ i,j \text{ whenever } k \neq \ell \tag{5.64}$$

We further assume that $\boldsymbol{Z}_k[i]$ is temporally white.

$$\mathbb{E}\boldsymbol{Z}_k[i]\boldsymbol{Z}_{k}^{*}[j] = 0,\ i \neq j \tag{5.65}$$

The temporal whiteness is satisfied when the FFT length N is large so that the input signal bandwidth can be assumed to have been divided into small narrow channels where the signal spectrum is almost flat. Recall from (5.31) and (5.32) that $\boldsymbol{Z}_{\ell}[i]$ contains $X_i[\ell]$, and $\boldsymbol{U}_{k,i}$ contains $X_{k,i}^{g}$ and $X_{k,i}^{r}$, where $X_{k,i}^{g}$ and $X_{k,i}^{r}$ are related to $X_i[\alpha_{k,m}]$ and $\dot{X}_i[\alpha_{k,m}]$ in (5.23). Therefore, there will be correlation between $X_i[\ell]$ and $\boldsymbol{U}_{k,i}$ if the indices of ℓ and $\alpha_{k,m}$ match. Therefore,

$$\mathbb{E}\boldsymbol{Z}_{\ell}[i]\boldsymbol{U}_{k,j}^{*}[2m] = \sigma_{X,\ell}^{2}\ \delta_{i,j}\ \delta_{\ell,\alpha_{k,m}} \tag{5.66a}$$

$$\mathbb{E}\boldsymbol{Z}_{\ell}[i]\boldsymbol{U}_{k,j}^{*}[2m+1] = (H_d[\ell])^{*}\ \sigma_{X,\ell}^{2}\ \delta_{i,j}\ \delta_{\ell,\alpha_{k,m}} \tag{5.66b}$$

where $m = \{0, ..., M-2\}$ and $\alpha_{k,m}$ is defined in (5.22). When the input to the FFT is real, $X_i[k]$ has conjugate symmetry. Therefore, we can also write

$$\mathbb{E}\boldsymbol{Z}_\ell[i]\boldsymbol{U}_{k,j}[2m] = \sigma^2_{X,\ell}\ \delta_{i,j}\ \delta_{\ell,N-\alpha_{k,m}} \tag{5.67a}$$

$$\mathbb{E}\boldsymbol{Z}_\ell[i]\boldsymbol{U}_{k,j}[2m+1] = (H_d[\ell])^*\ \sigma^2_{X,\ell}\ \delta_{i,j}\ \delta_{\ell,N-\alpha_{k,m}} \tag{5.67b}$$

where the index $\alpha_{k,m}$ in (5.66) is replaced by $N - \alpha_{k,m}$.

5.5.3 MSD and SNR measures

The error vector $\tilde{\boldsymbol{W}}_{k,i}$ and the a-priori error $\boldsymbol{e}_{a,k}[i]$ are defined as

$$\tilde{\boldsymbol{W}}_{k,i} \triangleq W^o - \boldsymbol{W}_{k,i} \tag{5.68a}$$

$$\boldsymbol{e}_{a,k}[i] \triangleq \boldsymbol{U}_{k,i}\tilde{\boldsymbol{W}}_{k,i-1} \tag{5.68b}$$

The mean-square-deviation (MSD) and the excess-mean-square-error (EMSE) of each bin are defined as

$$\text{MSD}_k \triangleq \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{W}}_{k,i}\|^2 \tag{5.69a}$$

$$\text{EMSE}_k \triangleq \lim_{i\to\infty} \mathbb{E}|\boldsymbol{e}_{a,k}[i]|^2 \tag{5.69b}$$

The overall MSD and EMSE measures are defined as the average values across all bins:

$$\text{MSD} \triangleq \lim_{i\to\infty} \frac{1}{N}\sum_{k=0}^{N-1} \mathbb{E}\|\tilde{\boldsymbol{W}}_{k,i}\|^2 \tag{5.70a}$$

$$\text{EMSE} \triangleq \lim_{i\to\infty} \frac{1}{N}\sum_{k=0}^{N-1} \mathbb{E}|\boldsymbol{e}_{a,k}[i]|^2 \tag{5.70b}$$

Note that in this work, due to the choice of the combination weights in (5.38), after combining the intermediate estimates across bins, the error vector $\tilde{\boldsymbol{W}}_{k,i}$ is the same for all k. Therefore, the overall MSD is equal to the MSD of each bin:

$$\text{MSD} = \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{W}}_{k,i}\|^2 \tag{5.71}$$

We can relate the EMSE to the signal-to-noise ratio (SNR) measure as follows. First, recall the system model for the k-th bin (5.30):

$$\boldsymbol{D}_k[i] = \underbrace{X_i[k]}_{\text{desired signal}} + \underbrace{\boldsymbol{U}_{k,i}W^o + V_i[k]}_{\text{distortion}} \tag{5.72}$$

The expected power of the desired signal and the distortion for each bin are $\sigma_{X,k}^2$ and $(W^o)^* R_{U,k} W^o + \sigma_{V,k}^2$, respectively. The recovered signal after the interference cancelation in (5.28) (see Fig. 5.8) can also be rewritten as

$$\begin{aligned}
\hat{X}_{B,i}[k] &= Y_i[k] - \hat{E}_{B,i}[k] \\
&\approx \boldsymbol{D}_k[i] - \boldsymbol{U}_{k,i}\hat{\boldsymbol{W}}_{k,i} && \text{(5.73a)} \\
&= \underbrace{X_i[k]}_{\text{desired signal}} + \underbrace{\boldsymbol{U}_{k,i}\tilde{\boldsymbol{W}}_{k,i} + V_i[k]}_{\text{remaining distortion}} && \text{(5.73b)}
\end{aligned}$$

The approximation in (5.73a) arises as $\hat{E}_{B,i}[k]$ from (5.29) is created by estimating the true regressor $\boldsymbol{U}_{k,i}$, whereas in (5.73a), we use true regressor $\boldsymbol{U}_{k,i}$ instead. The expected power of $\boldsymbol{U}_{k,i}\tilde{\boldsymbol{W}}_{k,i}$ as $i \to \infty$, is the EMSE_k in (5.69b). Therefore, across all bins, the overall SNR of the distorted signal, recovered signal, and the SNR improvement are

$$\text{SNR}_{\text{distorted}} = \frac{\sum_{k=0}^{N-1} \sigma_{X,k}^2}{\sum_{k=0}^{N-1} (W^o)^* R_{U,k} W^o + \sigma_{V,k}^2} \tag{5.74a}$$

$$\text{SNR}_{\text{recovered}} = \frac{\sum_{k=0}^{N-1} \sigma_{X,k}^2}{N \text{ EMSE} + \sum_{k=0}^{N-1} \sigma_{V,k}^2} \tag{5.74b}$$

$$\text{SNR}_{\text{improvement}} = \frac{\text{SNR}_{\text{recovered}}}{\text{SNR}_{\text{distorted}}} \tag{5.74c}$$

5.5.4 Error recursions

To evaluate the performance measures, we first need to examine how the errors evolve over time. The error recursions corresponding to the adaptive algorithm

(5.33) are derived by subtracting W^o from them to get

$$\tilde{\boldsymbol{\psi}}_{k,i} = \left(I_{2M-2} - \mu_{k,i}\boldsymbol{U}_{k,i}^*\boldsymbol{U}_{k,i}\right)\tilde{\boldsymbol{W}}_{k,i-1} - \mu_{k,i}\boldsymbol{U}_{k,i}^*\boldsymbol{Z}_k[i] \tag{5.75a}$$

$$\tilde{\boldsymbol{\psi}}_{k,i}^a[m] = \left[\begin{array}{cc} \tilde{\boldsymbol{\psi}}_{k,i}[2m] & \tilde{\boldsymbol{\psi}}_{k,i}[2m+1] \end{array}\right]^T \tag{5.75b}$$

$$\tilde{\boldsymbol{W}}_{k,i}^a[m] = \sum_{\ell=0}^{N-1} a_{\ell,i}[m]\tilde{\boldsymbol{\psi}}_{\ell,i}^a[m] \tag{5.75c}$$

$$\tilde{\boldsymbol{W}}_{k,i} = \left[\begin{array}{ccc} \left(\tilde{\boldsymbol{W}}_{k,i}^a[0]\right)^T & ... & \left(\tilde{\boldsymbol{W}}_{k,i}^a[M-2]\right)^T \end{array}\right]^T \tag{5.75d}$$

where $m = \{0, ..., M-2\}$. Recall from (5.38) that $a_{\ell,i}[m] = a_{\ell,k,i}[m]$, $\forall\ k$. Therefore, we use $a_{\ell,i}[m]$ in place of $a_{\ell,k,i}[m]$ in (5.75c). We are interested in approximating the MSD and EMSE measures in steady-state after the algorithm has had sufficient time to converge, i.e., as $i \to \infty$. For this reason, we shall assume that, at that stage, the step-sizes $\mu_{k,i}$ and the combination coefficients $a_{\ell,i}[m]$ would have converged towards steady-state values and drop their time index i. We now collect the recursions from all channels into a single vector model as follows:

$$\tilde{\boldsymbol{\psi}}_i = \left[\begin{array}{ccc} \tilde{\boldsymbol{\psi}}_{0,i}^T & ... & \tilde{\boldsymbol{\psi}}_{N-1,i}^T \end{array}\right]^T \tag{5.76a}$$

$$\tilde{\boldsymbol{W}}_i = \left[\begin{array}{ccc} \tilde{\boldsymbol{W}}_{0,i}^T & ... & \tilde{\boldsymbol{W}}_{N-1,i}^T \end{array}\right]^T \tag{5.76b}$$

and define the block diagonal matrices $\mathcal{M}$ and $\boldsymbol{\mathcal{R}}_i$ and vector $\boldsymbol{s}_i$:

$$\mathcal{M} \triangleq \text{diag}\{\ \mu_0 I_{2M-2},\ \ ...,\ \ \mu_{N-1} I_{2M-2}\ \} \tag{5.77a}$$

$$\boldsymbol{\mathcal{R}}_i \triangleq \text{diag}\{\ \boldsymbol{U}_{0,i}^*\boldsymbol{U}_{0,i},\ \ ...,\ \ \boldsymbol{U}_{N-1,i}^*\boldsymbol{U}_{N-1,i}\ \} \tag{5.77b}$$

$$\boldsymbol{s}_i \triangleq \text{col}\{\ \boldsymbol{U}_{0,i}^*\boldsymbol{Z}_0[i],\ \ ...,\ \ \boldsymbol{U}_{N-1,i}^*\boldsymbol{Z}_{N-1}[i]\ \} \tag{5.77c}$$

Using (5.62) and (5.66), it can be verified that

$$\mathbb{E}\boldsymbol{\mathcal{R}}_i \triangleq \mathcal{R} = \text{diag}\{\ R_{U,0},\ \ ...,\ \ R_{U,N-1}\ \} \tag{5.78a}$$

$$\mathbb{E}\boldsymbol{s}_i = 0 \tag{5.78b}$$

Let

$$\mathcal{S} \triangleq \mathbb{E}\boldsymbol{s}_i\boldsymbol{s}_i^* \tag{5.79}$$

which is an N by N block matrix with blocks of size $2(M-1)$ by $2(M-1)$. Using (5.61) and (5.62), the k-th diagonal block is

$$\mathcal{S}_{k,k} = \sigma_{Z,k}^2 R_{U,k},\ k = \{0, ..., N-1\} \tag{5.80}$$

For the p-th block row and k-th block column in $\mathcal{S}$ where $p \neq k$,

$$\mathcal{S}_{p,k} \triangleq \mathbb{E}\boldsymbol{Z}_p[i]\boldsymbol{U}_{p,i}^*\boldsymbol{U}_{k,i}\boldsymbol{Z}_k^*[i] \tag{5.81}$$

Using (5.31), (5.32) and (5.23), we note that $\boldsymbol{Z}_p[i]$ contains $X_i[p]$, while $\boldsymbol{Z}_k[i]$ contains $X_i[k]$ and $\boldsymbol{U}_{p,i}$ contains $X_i[\alpha_{p,m}]$ and $\dot{X}_i[\alpha_{p,m}]$. From the assumed spatial independence of $X_i[p]$, we conclude that $\boldsymbol{Z}_p[i]$ is independent of $\boldsymbol{Z}_k[i]$ and $\boldsymbol{U}_{p,i}$. Next, we also see that $\boldsymbol{U}_{k,i}^*$ contains $X_i^*[\alpha_{k,m}]$ and $\dot{X}_i^*[\alpha_{k,m}]$. Using the conjugate symmetry property we have, $X_i^*[\alpha_{k,m}] = X_i[N-\alpha_{k,m}]$ and $\dot{X}_i^*[\alpha_{k,m}] = \dot{X}_i[N-\alpha_{k,m}]$. Therefore, $\boldsymbol{Z}_p[i]$ is correlated with $\boldsymbol{U}_{k,i}^*$ when the indices p and $N-\alpha_{k,m}$ match. Applying the same reasoning to $\boldsymbol{Z}_k[i]$ on $\boldsymbol{U}_{p,i}^*$, we see that $\boldsymbol{Z}_k[i]$ is correlated with $\boldsymbol{U}_{p,i}^*$ when the indices k and $N-\alpha_{p,m}$ match. Therefore, it follows that

$$\begin{aligned}
\mathcal{S}_{p,k} &= \mathbb{E}\left(\boldsymbol{Z}_k[i]\boldsymbol{U}_{p,i}\right)^*\left(\boldsymbol{Z}_p[i]\boldsymbol{U}_{k,i}\right),\ p \neq k \\
&= \left(\begin{bmatrix} 1 & H_d[k] \end{bmatrix} P_{k,p}\right)^* \left(\begin{bmatrix} 1 & H_d[p] \end{bmatrix} P_{p,k}\right) \\
&= P_{k,p}^T \begin{bmatrix} 1 & \\ & (H_d[k])^* \end{bmatrix} \begin{bmatrix} 1 & H_d[p] \end{bmatrix} P_{p,k}
\end{aligned} \tag{5.82}$$

where p and k are in $\{0, ..., N-1\}$ and matrix $P_{p,k}$ is

$$P_{p,k} = \sigma^2_{X,p} \left[\begin{array}{cccc} I_2 \ \delta_{p,N-\alpha_{k,0}} & ... & I_2 \ \delta_{p,N-\alpha_{k,M-2}} \end{array} \right] \tag{5.83}$$

We can rewrite (5.75d) as

$$\tilde{\boldsymbol{W}}_{k,i} = \sum_{\ell=0}^{N-1} A_\ell \tilde{\boldsymbol{\psi}}_{\ell,i} \tag{5.84a}$$

$$A_\ell = \text{diag}\{ \ a_\ell[m] I_2, \ \ ..., \ \ a_\ell[M-2] I_2 \ \} \tag{5.84b}$$

$$\sum_{\ell=0}^{N-1} A_\ell = I_{2M-2} \tag{5.84c}$$

A block matrix representation of (5.84) is shown in (5.85).

$$\begin{aligned}
&\left[\begin{array}{c} \tilde{\boldsymbol{W}}_{0,i} \\ \vdots \\ \tilde{\boldsymbol{W}}_{N-1,i} \end{array} \right] \\
&= \left[\begin{array}{c} \left. \begin{array}{c} \left[\begin{array}{c} \tilde{\boldsymbol{W}}^a_{0,i}[0] \\ \vdots \\ \tilde{\boldsymbol{W}}^a_{0,i}[M-2] \end{array} \right] \end{array} \right\} \tilde{\boldsymbol{W}}_{0,i} \\ \vdots \\ \tilde{\boldsymbol{W}}_{N-1,i} \end{array} \right] \\
&= \left[\begin{array}{ccc} \underbrace{\left[\begin{array}{ccc} a_0[0] I_2 & & \\ & \ddots & \\ & & a_0[M-2] I_2 \end{array} \right]}_{A_0 : (2M-2) \times (2M-2) \text{ diagonal matrix}} & ... & \underbrace{\left[\begin{array}{ccc} a_{N-1}[0] I_2 & & \\ & \ddots & \\ & & a_{N-1}[M-2] I_2 \end{array} \right]}_{A_{N-1}} \\ \vdots & ... & \vdots \\ A_0 & ... & A_{N-1} \end{array} \right] \left[\begin{array}{c} \left[\begin{array}{c} \tilde{\boldsymbol{\psi}}^a_{0,i}[0] \\ \vdots \\ \tilde{\boldsymbol{\psi}}^a_{0,i}[M-2] \end{array} \right] \\ \vdots \\ \left[\begin{array}{c} \tilde{\boldsymbol{\psi}}^a_{N-1,i}[0] \\ \vdots \\ \tilde{\boldsymbol{\psi}}^a_{N-1,i}[M-2] \end{array} \right] \end{array} \right] \\
&= \left[\begin{array}{c} I_{2M-2} \\ \vdots \\ I_{2M-2} \end{array} \right] \left[\begin{array}{ccc} A_0 & ... & A_{N-1} \end{array} \right] \left[\begin{array}{c} \tilde{\boldsymbol{\psi}}_{0,i} \\ \vdots \\ \tilde{\boldsymbol{\psi}}_{N-1,i} \end{array} \right]
\end{aligned} \tag{5.85}$$

From (5.85), we define the matrix $\mathcal{A}$ for all bins as

$$\mathcal{A}^T \triangleq KA^T \tag{5.86a}$$

$$K \triangleq \begin{bmatrix} I_{2M-2} & ... & I_{2M-2} \end{bmatrix}^T \tag{5.86b}$$

$$A \triangleq \begin{bmatrix} A_0 & ... & A_{N-1} \end{bmatrix}^T \tag{5.86c}$$

Therefore, substituting (5.75a) into (5.85), we conclude that, in steady-state, the error vectors across all channels evolve over time according to the following recursion:

$$\tilde{\boldsymbol{W}}_i = \mathcal{A}^T \left(I_{N(2M-2)} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right) \tilde{\boldsymbol{W}}_{i-1} - \mathcal{A}^T \mathcal{M} \boldsymbol{s}_i, \ i \geq 1 \tag{5.87}$$

5.5.5 Combination weights and stepsizes

From (5.30), the interference term is $\boldsymbol{U}_{k,i} W^o$. We assume that when the solution converges, the residual error $\boldsymbol{U}_{k,i}\tilde{\boldsymbol{W}}_{k,i}$ is smaller than the signal component and unmodelled dynamics (together as $\boldsymbol{Z}_k[i]$). Therefore, when the solution converges, the average power in each recovered bin can be approximated by $\sigma^2_{Z,k}$. The combination weights $a_\ell[m]$ in (5.84) are based on (5.38), where we have dropped the indices k from $a_{\ell,k}[m]$. Let us examine $a_\ell[m]$ for $m = 0$. In (5.37), we see that the numerator is $P^g_\ell[m] + P^r_\ell[m]$. From (5.36), we see that $P^g_\ell[0] \approx P_Z[\alpha_{\ell,0}]$ and $P^r_\ell[0] \approx \dot{P}_Z[\alpha_{\ell,0}]$. From (5.35), $P_Z[\alpha_{\ell,0}]$ and $\dot{P}_Z[\alpha_{\ell,0}]$ are the estimated power of the recovered signal and its "derivative" in the $\alpha_{\ell,0}$-th channel, which are $\sigma^2_{Z,\alpha_{\ell,0}}$ and $\sigma^2_{Z,\alpha_{\ell,0}}|H_d[\alpha_{\ell,0}]|^2$, respectively. Similarly, in (5.37), we see that the denominator is $P_Z[\ell] = \sigma^2_{Z,\ell}$. Therefore, the converged combination weights in (5.84b) can be approximated as:

$$a_\ell[m] \approx \bar{\gamma}_m \gamma_{\ell,m} \tag{5.88}$$

where

$$\bar{\gamma}_m = \left(\sum_{k=0}^{N-1} \gamma_{k,m} \right)^{-1} \tag{5.89a}$$

$$\gamma_{\ell,m} = \frac{\sigma^4_{Z,\alpha_{\ell,m}}}{\sigma^2_{Z,\ell}} \left(1 + |H_d[\alpha_{\ell,m}]|^2 \right) \tag{5.89b}$$

We assume that $\sigma^2_{Z,\ell} > 0$ for all ℓ and hence, $\gamma_{\ell,m} > 0$. Therefore, the combination weight $a_\ell[m]$ is always positive.

$$0 < a_\ell[m] < 1 \tag{5.90}$$

Recall from (5.43) and (5.44) that the stepsizes μ_ℓ are bounded by the eigenvalues of $R_{U,\ell}$. From (5.41) and (5.36), we see that the eigenvalues are estimated as the sum of $P_Z[\alpha_{\ell,m}]$ and $\dot{P}_Z[\alpha_{\ell,m}]$. These quantities can be expressed as $\sigma^2_{Z,\alpha_{\ell,m}}$ and $\sigma^2_{Z,\alpha_{\ell,m}}|H_d[\alpha_{\ell,m}]|^2$, respectively. Therefore, the step-sizes satisfy the bound:

$$\mu_\ell \leq \frac{2}{\sigma^2_{Z,\alpha_{\ell,m}} \left(1 + |H_d[\alpha_{\ell,m}]|^2\right)}, \; m = \{0, ..., M-2\} \tag{5.91}$$

From (5.62) and (5.63), we see that the non-zero eigenvalues of $R_{U,\ell}$ are the non-zero eigenvalues of the rank-1 matrix $R_{X,\alpha_{\ell,m}}$. The non-zero eigenvalue of $R_{X,\alpha_{\ell,m}}$ is the trace $\sigma^2_{X,\alpha_{\ell,m}} \left(1 + |H_d[\alpha_{\ell,m}]|^2\right)$. Since $\sigma^2_{X,\ell} < \sigma^2_{Z,\ell}$, the eigenvalues of $\mu_\ell R_{X,\alpha_{\ell,m}}$ is bounded by

$$\lambda_{\max} \left(\mu_\ell R_{X,\alpha_{\ell,m}} \right) \leq \frac{2\sigma^2_{X,\alpha_{\ell,m}}}{\sigma^2_{Z,\alpha_{\ell,m}}} < 2 \tag{5.92}$$

5.5.6 Convergence in mean

Taking expectation of both sides of (5.87) and using (5.78) and (5.86), we obtain

$$\mathbb{E}\tilde{\boldsymbol{W}}_i = (KB)\mathbb{E}\tilde{\boldsymbol{W}}_{i-1} \tag{5.93}$$

where

$$B \triangleq A^T \left(I_{N(2M-2)} - \mathcal{M}\mathcal{R} \right) \tag{5.94}$$

Asymptotic mean stability is guaranteed when

$$\rho\left(KB\right) < 1 \tag{5.95}$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument. In appendix 5.A, we show that conditions (5.90) and (5.92) ensure (5.95) so that stability in the mean is guaranteed.

5.5.7 Mean square stability

In this section, we derive the MSD and EMSE measures defined in (5.69) and (5.70). From [109], it is known that

$$\mathbb{E}|\boldsymbol{e}_{a,k}[i]|^2 = \mathbb{E}\|\tilde{\boldsymbol{W}}_{k,i}\|^2_{R_{U,k}} \tag{5.96}$$

where the notation $\|x\|^2_A$ denotes the squared weighted Euclidean norm x^*Ax. We therefore proceed to evaluating the expression for $\mathbb{E}\|\tilde{\boldsymbol{W}}_{k,i}\|^2_{\Sigma_k}$ for an arbitrary Hermitian non-negative definite matrix Σ_k. Setting $\Sigma_k = R_{U,k}$ or I_{2M-2} yields the MSD or EMSE expressions in (5.69) and (5.70), respectively. First using (5.86), we re-write (5.87) more compactly as

$$\tilde{\boldsymbol{W}}_i = \boldsymbol{\mathcal{B}}_i\tilde{\boldsymbol{W}}_{i-1} - \mathcal{G}\boldsymbol{s}_i,\ i \geq 0 \tag{5.97a}$$

$$\boldsymbol{\mathcal{B}}_i \triangleq \mathcal{A}^T\left(I_{N(2M-2)} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right) \tag{5.97b}$$

$$\mathcal{G} \triangleq \mathcal{A}^T\mathcal{M} = KA^T\mathcal{M} \tag{5.97c}$$

From (5.93) and (5.94), we can also denote the mean of $\boldsymbol{\mathcal{B}}_i$ as

$$\mathcal{B} \triangleq \mathbb{E}\boldsymbol{\mathcal{B}}_i = KB \tag{5.98}$$

Finally, using (5.79), we introduce the matrices $\mathcal{F}$ and $\mathcal{Y}$ as

$$\begin{aligned}\mathcal{F} &\triangleq \mathcal{B}^T \otimes \mathcal{B}^* \\ \mathcal{Y} &\triangleq \mathcal{G}\mathcal{S}\mathcal{G}^T\end{aligned} \tag{5.99}$$

where $\otimes$ denotes the Kronecker product. In [109], for small step sizes, it was argued that if $\mathcal{F}$ is stable, then (5.97) is mean-square stable. Moreover, if $\mathcal{B}$ is stable, then $\mathcal{F}$ is stable.

5.5.8 MSD and EMSE

Following the same arguments from [109], the MSD and EMSE in (5.70) are given by the following expressions:

$$\text{MSD} = \frac{1}{N}\left[\text{vec}\left(\mathcal{Y}^{\mathcal{T}}\right)\right]^T (I-\mathcal{F})^{-1}\,\text{vec}\left(I_{N(2M-2)}\right) \tag{5.100a}$$

$$\text{EMSE} = \frac{1}{N}\left[\text{vec}\left(\mathcal{Y}^{\mathcal{T}}\right)\right]^T (I-\mathcal{F})^{-1}\,\text{vec}\left(\mathcal{R}\right) \tag{5.100b}$$

where the vec operator vectorizes its matrix argument by stacking the columns on top of each other, and $\mathcal{R}$ is defined in (5.78).

Due to the structure of $\mathcal{A}$, where all estimates after the combination step are the same, we can use (5.86) to reduce (5.97) to a single node recursion. Thus note first that

$$\begin{aligned} K\tilde{\boldsymbol{W}}_{k,i} = &\underbrace{KA^T\left(I_{N(2M-2)} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right)}_{\boldsymbol{\mathcal{B}}_i}\underbrace{K\tilde{\boldsymbol{W}}_{k,i}}_{\tilde{\boldsymbol{W}}_{i-1}} \\ &- \underbrace{KA^T\mathcal{M}}_{\mathcal{G}}\boldsymbol{s}_i \end{aligned} \tag{5.101}$$

Multiplying both sides of (5.101) by $\frac{1}{N}K^T$ and noting that $\frac{1}{N}KK^T = I_{2M-2}$, we obtain the single node recursion as:

$$\tilde{\boldsymbol{W}}_{k,i} = \boldsymbol{\mathcal{B}}_{\text{node},i}\tilde{\boldsymbol{W}}_{k,i-1} - \mathcal{G}_{\text{node}}\boldsymbol{s}_i,\ i \geq 0 \tag{5.102a}$$

$$\boldsymbol{\mathcal{B}}_{\text{node},i} \triangleq A^T\left(I_{N(2M-2)} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right)K \tag{5.102b}$$

$$\mathcal{G}_{\text{node}} \triangleq A^T\mathcal{M} \tag{5.102c}$$

Hence, the MSD, EMSE_k and EMSE can also be written as

$$\text{MSD} = h_{\text{node}}^T \text{ vec}\left(I_{2M-2}\right) \tag{5.103a}$$

$$\text{EMSE}_k = h_{\text{node}}^T \text{ vec}\left(R_{U,k}\right) \tag{5.103b}$$

$$\begin{aligned}\text{EMSE} &= \frac{1}{N}\sum_{k=0}^{N-1} h_{\text{node}}^T \text{ vec}\left(R_{U,k}\right) \\ &= \frac{1}{N} h_{\text{node}}^T \text{ vec}\left(K^T \mathcal{R} K\right)\end{aligned} \tag{5.103c}$$

where

$$h_{\text{node}}^T \triangleq \left[\text{vec}\left(\mathcal{Y}_{\text{node}}^{\mathcal{T}}\right)\right]^T \left(I - \mathcal{F}_{\text{node}}\right)^{-1} \tag{5.104a}$$

$$\mathcal{B}_{\text{node}} \triangleq \mathbb{E}\boldsymbol{\mathcal{B}}_{\text{node},i} = BK \tag{5.104b}$$

$$\mathcal{F}_{\text{node}} \triangleq \mathcal{B}_{\text{node}}^T \otimes \mathcal{B}_{\text{node}}^* \tag{5.104c}$$

$$\mathcal{Y}_{\text{node}} \triangleq \mathcal{G}_{\text{node}} \mathcal{S} \mathcal{G}_{\text{node}}^T \tag{5.104d}$$

The advantage of using (5.103) over (5.100) is that the matrices $\mathcal{Y}_{\text{node}}^T$ and $\mathcal{F}_{\text{node}}$ are much smaller compared to $\mathcal{Y}^T$ and $\mathcal{F}$.

5.6 Simulations results

The simulations in this section are carried out by generating a random Gaussian signal with variance equal to one. The signal is either low-pass or band-pass filtered according to the desired simulation settings. This signal is the clean input signal to the ADC. Sinc interpolation is used to create the distorted signal with some time and gain offsets. The gain and time offsets are randomly generated; their standard deviation is set to 0.01 and their means are set to 1 and 0, respectively. The time offsets and gain mismatches when $M = 4$ are {-7.55, 7.60, 9.60, -9.65} $\times 10^{-3}$ and {0.9994, 0.9998, 1.0126, 0.9881}, respectively. The step-sizes used in the time-domain solution (5.15) and frequency-domain solution

(5.44) are set to $\mu = 1.75 \times 10^{-6}$ and $\mu_{\text{init}} = 2 \times 10^{-4}$, respectively. The factor β in (5.44) is set to 1. The FFT length is 2^{10}. The FFT windowing function is the Blackman-Harris window. The number of ADCs $M = 4$. The λ used in the averaging of (5.39) is 0.95.

5.6.1 MSD and SNR measures

Recall from (5.30) and (5.33d) that the true and estimated distortion parameter at the i-th block are W^o and W_i, respectively. The estimation error is defined as $\tilde{W}_i = W^o - W_i$. The mean-square-deviation (MSD) is defined as

$$\text{MSD} = \lim_{i \to \infty} \mathbb{E}\|\tilde{W}_i\|^2 \tag{5.105}$$

In the simulations, the MSD is calculated by averaging $\|\tilde{W}_i\|^2$ over some runs. The signal power to noise power ratio (SNR) of the distorted signal and the recovered signal can also be calculated from the simulations. From (5.30) and (5.28), at the i-th block, we denote the power of the input signal, original distortion and the residual error after the interference cancelation block as:

$$P_{\text{signal}} = \sum_{k=0}^{N-1} |X_i[k]|^2 \tag{5.106a}$$

$$P_{\text{distortion}} = \sum_{k=0}^{N-1} |Y_i[k] - X_i[k]|^2 \tag{5.106b}$$

$$P_{\text{residual}} = \sum_{k=0}^{N-1} |\hat{X}_{B,i}[k] - X_i[k]|^2 \tag{5.106c}$$

P_{signal}, $P_{\text{distortion}}$ and P_{residual} can be averaged over some runs, and the average SNR of the original distorted signal and the recovered signal is calculated as:

$$\text{SNR}_{\text{distorted}} = \frac{\text{“average” of } P_{\text{signal}}}{\text{“average” of } P_{\text{distortion}}} \tag{5.107a}$$

$$\text{SNR}_{\text{recovered}} = \frac{\text{“average” of } P_{\text{signal}}}{\text{“average” of } P_{\text{residual}}} \tag{5.107b}$$

5.6.2 Comparing with prior works

Our first set of simulations compare our proposed algorithms with some prior works [1, 2]. The comparison shows that the frequency-domain approach gives better performance. Recall in Section 5.4 that we discussed that our solutions compensate the distorted signal in the first stage, and then use it in the interference cancelation block (stage B of Fig. 5.8 and 5.12) to recover a better signal. As such, in the simulations, we calculate the SNR of the recovered signal from both stages.

Recall that the distortion parameters in the various solutions are different. For the frequency-domain solution, the distortion parameters are G_k and R_k in (5.5). For our proposed time-domain solution and the work in [1], the distortion parameters are the same c_g and c_r in (5.10). For [2], the distortion parameters are w_g and w_t, defined in (5.58) and (5.59). Therefore, to compare the estimation errors from the various solutions using the MSD measure defined in (5.105), we have to convert all the estimates to the distortion parameters used for the frequency-domain solution. We can convert c_g and c_r to G_k and R_k:

$$G_k = \begin{cases} c_g[2k-2] + jc_g[2k-1] & \text{if } 1 \le k \le \frac{M}{2} - 1 \\ c_g[2k-2] & \text{if } k = \frac{M}{2} \\ G^*_{N-k} & \text{otherwise} \end{cases} \tag{5.108}$$

and

$$R_k = \begin{cases} c_r[2k-2] + jc_r[2k-1] & \text{if } 1 \le k \le \frac{M}{2} - 1 \\ c_r[2k-2] & \text{if } k = \frac{M}{2} \\ R^*_{N-k} & \text{otherwise} \end{cases} \tag{5.109}$$

For the distortion parameters w_g and w_t in [2], we first convert them to g_m and r_m using (5.58) and (5.59):

$$\begin{bmatrix} g_0 \\ \vdots \\ g_{M-1} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} - MF_H^{-1} \begin{bmatrix} w_{g0} \\ \vdots \\ w_{g(M-1)} \end{bmatrix} \tag{5.110a}$$

$$\begin{bmatrix} r_0 \\ \vdots \\ r_{M-1} \end{bmatrix} = MF_H^{-1} \begin{bmatrix} w_{t0} \\ \vdots \\ w_{t(M-1)} \end{bmatrix} \tag{5.110b}$$

Once g_m and r_m are obtained, we can use (5.5) to find G_k and R_k. Therefore, in the simulations, we can calculate and plot the MSD of the various solutions as defined in (5.105) and (5.71) and compare their performance.

Similarly, we also compare the SNR measure defined in (5.107) and (5.74) for the various solutions. For the comparison, we applied the same windowed-FFT used in the proposed frequency-domain solution on the recovered signals from the time-domain solutions, and calculate the SNR for each FFT block. The SNR of the last block is tabulated for comparison purposes.

5.6.2.1 Low-pass input signal

Recall that reference [1] assumes that there exists an out-of-band region, and uses a high-pass filter to estimate the distortion parameters in that region (recall Figs. 5.4 and 5.3). Therefore, in this simulation, we simulate a low-pass signal with a bandwidth of 0.7π, the high-pass filter's cutoff point is set to 0.8π. We denote the algorithm in reference [1] as "Vogel's solution". We also implemented the algorithm in reference [2] and denote it as "Matsuno's solution". The length of the input data is 2^{21}, the FFT length is 2^{10} and the simulation results are averaged over 30 experiments.

Table 5.1: SNR of the distorted and recovered low-pass signal.

Description	**SNR [dB]**
Distorted signal	37.02
Matsuno's solution [2]	51.28
Time-domain solution (Stage A of Fig. 5.7)	52.27
Time-domain solution (Stage B of Fig. 5.7)	52.32
Vogel's solution [1]	68.11
Frequency-domain solution (a) (Stage A1 of Fig. 5.8)	69.98
Frequency-domain solution (a) (Stage B of Fig. 5.8)	77.88
Frequency-domain solution (b) (Stage A2 of Fig. 5.12)	78.30
Frequency-domain solution (b) (Stage B of Fig. 5.12)	81.15

Fig. 5.14 shows the simulation results averaged over 30 runs. The average SNR of the distorted signal and the average SNR of the recovered signal after using the various algorithms are shown in Table 5.1. The SNR of the distorted signal is 37dB. In this scenario, when all the distortion components are present in the out-of-band region, Vogel's solution works very well and the SNR of the recovered signal is high (68dB). Both our time-domain and Matsuno's solution are able to improve the SNR ($\sim$52dB), albeit at a worse performance level. This is because the algorithms use the entire bandwidth of the data to estimate the distortion parameters. In this case, the input signal becomes the "noise". For Vogel's solution, the out-of-band region contains only the distortion signal. Hence, the time-domain and Matsuno's solutions require more samples. The frequency-domain solutions (a) and (b) use the structures of Figs. 5.8 and 5.12, respectively. We recall that the proposed frequency-domain solutions examine the frequency content of the signal and selectively emphasize the estimation

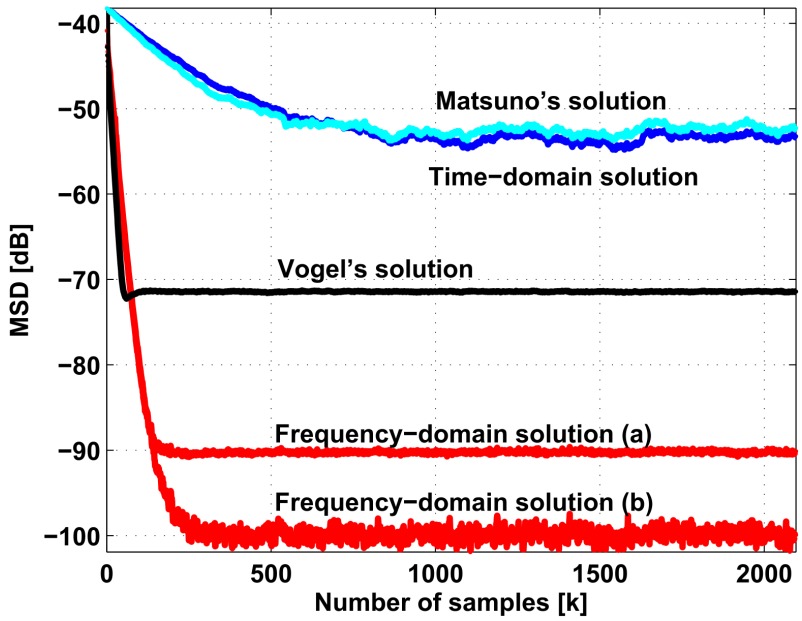


Figure 5.14: Comparing various algorithms on a low-pass signal at 0.7π. The frequency-domain solutions (a) and (b) use the proposed structures in Figs. 5.8 and 5.12, respectively.

in regions where there is less "noise" via the combination weights (5.38) and the stepsizes (5.44). Therefore, the frequency-domain solutions converge faster than the proposed time-domain solution. Moreover, we see that the frequency-domain solutions obtain a significant improvement in MSD over all other solutions. The MSD plot also shows that the frequency-domain solution (b), where the second order distortion is removed, gives a further 10dB improvement over the frequency-domain solution (a). The SNR of the recovered signal using the frequency-domain solutions are also better than the time-domain solutions, i.e., 78dB to 81dB.

When we compare the SNR of the recovered signals in both stages of our algorithms, we note that stage B of the proposed solutions gives a better SNR. The averaged spectrum of the distorted signal and the recovered signal using the

frequency-domain solution is shown in the left plot of Fig. 5.15. Note that the SNR improvement of the recovered signal is over the entire bandwidth of the signal. The spectrum plots show the reduction of the distortion spectrum in the out-of-band region. We noted that, visually, the spectrum of the recovered signal using the frequency-domain solutions (a) and (b) are similar.

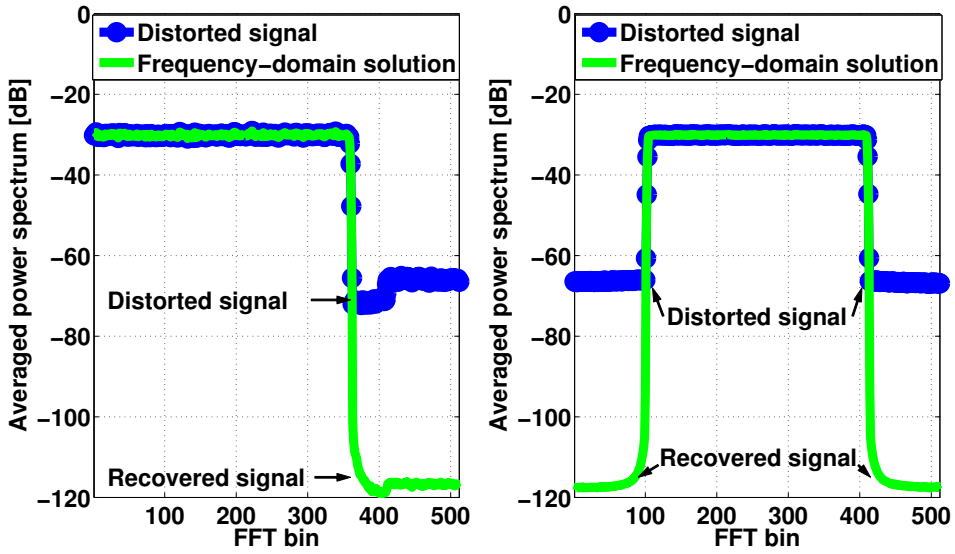


Figure 5.15: The left and right plots show the spectrum of the distorted and recovered signals using the frequency-domain solution for the two scenarios of a low-pass signal (left) and a bandpass signal (right).

5.6.2.2 Band-pass input signal

Now, we change the input signal to a band-pass signal from 0.2π to 0.8π. As discussed in Section 5.2.3, Vogel's solution will not work in this scenario as one of the distortion components is within the input signal's bandwidth (recall Fig. 5.5c). If the high-pass filter's cutoff point is set to 0.8π as in the previous simulation, it will fail dramatically since some of the original signal's bandwidth (from

0.7π to 0.8π) leaks into the high-pass region, and the SNR of the recovered signal will be worse than the distorted signal itself. As such, we change the high-pass filter's cutoff point to 0.85π. This reduces the leakage and allows the algorithm to estimate the distortions that are outside the signal's bandwidth. We will denote the results when the high-pass cutoff is at 0.8π and 0.85π as Vogel's solution (a) and (b), respectively. For the other algorithms, their parameters remain as in the previous set of simulations. However, as one distortion component is completely within the signal bandwidth, the algorithms take more time to converge. Therefore, we increase the input data samples to $2^{21} \times 5$.

Table 5.2: SNR of the distorted and recovered band-pass signal.

Description	**SNR [dB]**
Distorted signal	35.47
Matsuno's solution [2]	50.23
Time-domain solution (Stage A of Fig. 5.7)	51.13
Time-domain solution (Stage B of Fig. 5.7)	51.21
Vogel's solution (a) [1]	27.73
Vogel's solution (b) [1]	44.08
Frequency-domain solution (a) (Stage A1 of Fig. 5.8)	56.93
Frequency-domain solution (a) (Stage B of Fig. 5.8)	57.26
Frequency-domain solution (b) (Stage A2 of Fig. 5.12)	57.27
Frequency-domain solution (b) (Stage B of Fig. 5.12)	57.28

Fig. 5.16 shows the MSD for the various algorithms and Table 5.2 shows the SNR of the distorted and recovered signals. The figure and table show that the frequency-domain solutions perform better than the other algorithms. More importantly, recall that the algorithms' parameters used in the low-pass input

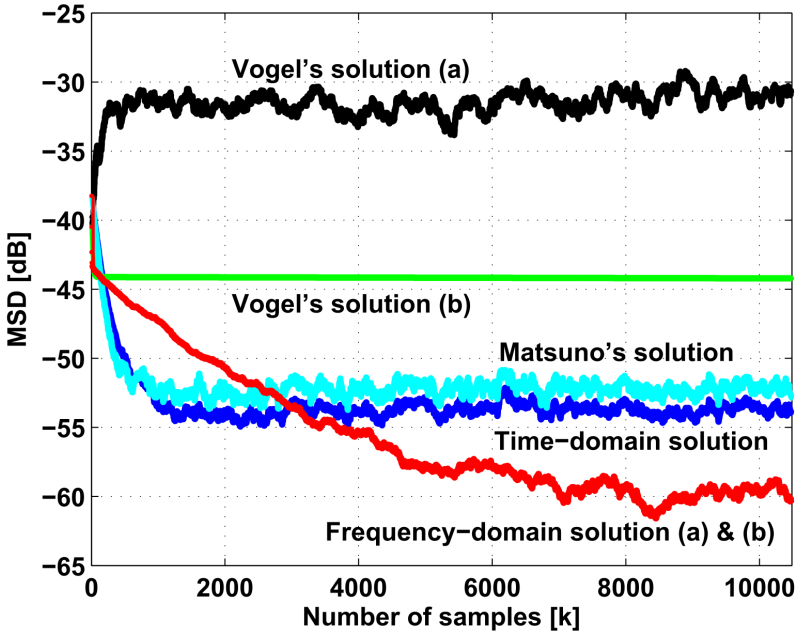


Figure 5.16: Comparing various algorithms on a band-pass signal from 0.2π to 0.8π. The simulation results for Vogel's solution (a) and (b) use a high-pass filter cutoff at 0.8π and 0.85π, respectively.

signal case remains the same in the band-pass signal case; However, for the frequency-domain solution, the stepsizes and combination weights are adaptive to the input signal, and is able to adapt these algorithm parameters accordingly. As a result, the convergence rate is adaptively reduced to obtain MSD and SNR of -60dB and 57dB, respectively. We also noted that in the band-pass signal case, removing the second-order distortion does not improve performance. As Vogel's solution cannot estimate all the distortion components, it gives the worst SNR and MSD performance. Vogel's solution (a) performs worse than Vogel's solution (b) because the actual signal leaks into the out-of-band region; This breaks the assumption that the out-of-band region contains only distortions.

For our proposed time-domain solution and Matsuno's solution, the stepsize

used in their algorithm is the same as the previous simulation. Hence, unlike the frequency-domain solutions, they are not able to adapt to the input signal. Although they converge faster, their MSD and SNR performance is worse. If a designer were to use either of these two solutions, and would like to get similar MSD and SNR performance as the frequency-domain solution, then the designer needs to reduce the stepsize. However, this will further reduce their slow convergence rate for the low-pass input signal scenario in the previous section (see Fig. 5.14). For the sake of completeness, we emprically reduce the stepsizes of the proposed time-domain solution and Matsuno's solution by a factor of 0.2 to obtain MSD and SNR that are similar to the frequency-domain solution. The SNR is in Table 5.3 and the MSD curve is in Fig. 5.17.

Table 5.3: SNR of the distorted and recovered band-pass signal after reducing the stepsizes for the time-domain solution and Matsuno's solution.

Description	**SNR [dB]**
Distorted signal	35.47
Matsuno's solution [2]	56.62
Time-domain solution (Stage A of Fig. 5.7)	57.73
Time-domain solution (Stage B of Fig. 5.7)	58.03
Frequency-domain solution (a) (Stage A1 of Fig. 5.8)	56.93
Frequency-domain solution (a) (Stage B of Fig. 5.8)	57.26

The frequency-domain solution tries to exploit the case when the distortion components leaks into empty frequency bands, and the algorithm detects them and weighs the estimated distortion parameters more in these frequency bands. For the band-pass case where it is centered at $\frac{\pi}{2}$, as shown in Fig. 5.5c, we see that there is at least one distortion component (one of the smaller triangles) lies

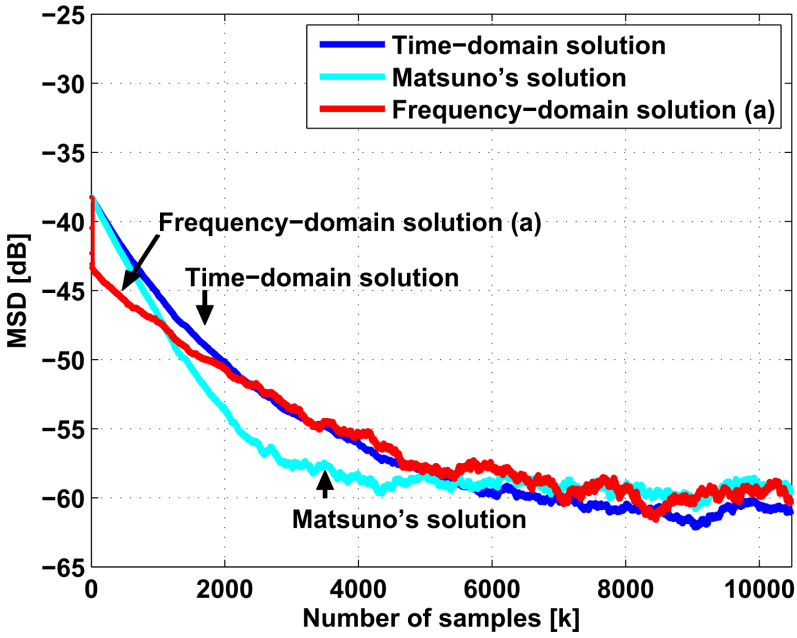


Figure 5.17: Comparing various algorithms on a band-pass signal from 0.2π to 0.8π, after reducing the stepsizes for the time-domain solution and Matsuno's solution.

completely in the signal band and does not leak into any empty frequency bands. This means that at least *one* of the unknown distortion parameters *cannot* be estimated quickly. If the band-pass signal is slightly off-centered from $\frac{\pi}{2}$, then some parts of all the individual distortion components will leak into some empty frequency bands. In this situation, the frequency-domain solution will be able estimate the distortion parameters quickly and more accurately. The averaged spectrum of the distorted signal and the recovered signal using the frequency-domain solution is shown in the right plot of Fig. 5.15.

5.6.3 Performance measures using MSD and SNR

In this section, the performance of the frequency domain method is compared against the approximate theoretical MSD and SNR values in (5.70) and (5.74).

The algorithm's settings are the same as before and the FFT length is varied from 2^9, 2^{10} and 2^{11}. The input signal used is either a low-pass signal or a band-pass signal with some varying bandwidth. Recall that our frequency-domain solution in Fig. 5.8 can also be extended to remove higher order distortion in Fig. 5.12. As in the previous sections, we will label the solutions as solution (a) and (b), respectively. We will show that in some cases, solution (b) improves the MSD and approaches the theoretical performance.

5.6.3.1 Low-pass input signal

When the input signal is a low-pass signal, the bandwidth is varied from 0.1π, 0.3π, 0.5π, 0.7π to 0.9π radians/sample. The input signal data length for the first four bandwidth regions and for the last bandwidth region are $(2^{18}) \times 5$ and $(2^{21}) \times 5$, respectively. The results are averaged over 30 runs. Using the frequency-domain solution (a), the MSD (left-sided plot) and SNR (right-sided plot) results are shown in Fig. 5.18. From the MSD plot, we noted that increas-

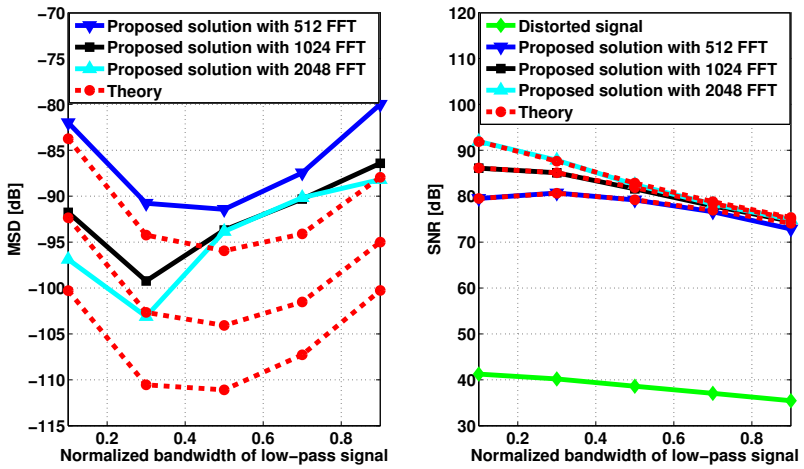


Figure 5.18: Performance of solution (a), which removes the first order distortions, as lowpass signal's bandwidth varies from 0.1π to 0.9π.

ing the FFT length did not improve the MSD performance significantly, and the curves do not match the theoretical performance. We found that the mismatch is due to the higher order distortions that we have ignored; When we remove the second-order distortion in solution (b), we found that the simulation results match with the theory (see Fig. 5.19). However, the SNR plots show that the SNR performance in both solution (a) and (b) match well with the theoretical performance. Depending on the input signal's bandwidth, the distorted signal's and recovered signal's SNR ranged between 35dB to 41dB and 70dB to 90dB, respectively. From the SNR plot, we see that increasing the FFT length improves the SNR; especially when the input signal bandwidth is smaller, but not as significant at larger bandwidth. From the SNR plot, using 1024 FFT, we see that the SNR of the recovered signal (black line with square marker) varies from 86dB to 79dB as the bandwidth increases, and the SNR improvement from the distorted signal (green line with diamond) is about 45dB.

Using the frequency-domain solution (b), the MSD (left-sided plot) and SNR (right-sided plot) results are shown in Fig. 5.19. Now, from the MSD plot, we noted that the simulation results approach the theoretical performance. We also noted that doubling the FFT length improves the MSD result by 8dB to 10dB. The SNR plot shows that the simulation performance match well with the theoretical performance (similar to solution (a)).

5.6.3.2 Band-pass input signal

When the input signal is a band-pass signal (centered at 0.5π), the bandwidth is varied from 0.2π, 0.4π, 0.6π, 0.8π to 0.94π radians/sample. The input signal data length is $(2^{21}) \times 10$. The results are averaged over 30 runs. The performance results using both solutions (a) and (b) are shown in Figs. 5.20 and 5.21,

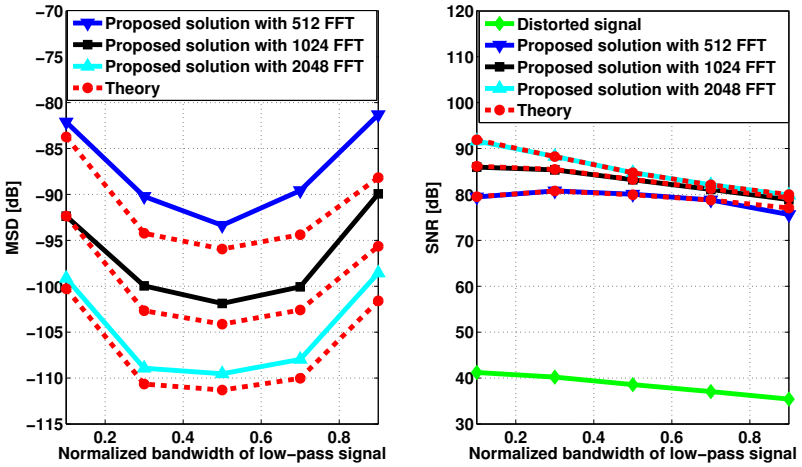


Figure 5.19: Performance of solution (b), which removes the first and second order distortions, as lowpass signal's bandwidth varies from 0.1π to 0.9π.

respectively. First, we noted that for the band-pass signal, both solutions give the same performance. The results also show that increasing the FFT length improves the performance. Using 1024 FFT, the SNR of the recovered signal (black line with square marker) varies from 53dB to 57dB, and the SNR improvement from the distorted signal (green line with diamond) is about 19dB to 22dB.

5.7 Discussion and conclusion

We proposed time-domain and frequency-domain solutions to mitigate the timing and gain mismatches in TI-ADC. One of the main advantages of this work over prior works is that it does not need to assume an out-of-band frequency region. By spliting the input signal into separate FFT bins and modelling each frequency bin as a node in a network, we are able to exploit adaptive diffusion strategies [109] to get better performance over the time-domain approach. Consequently, we showed that the frequency-domain solution is able to adapt to different scenarios;

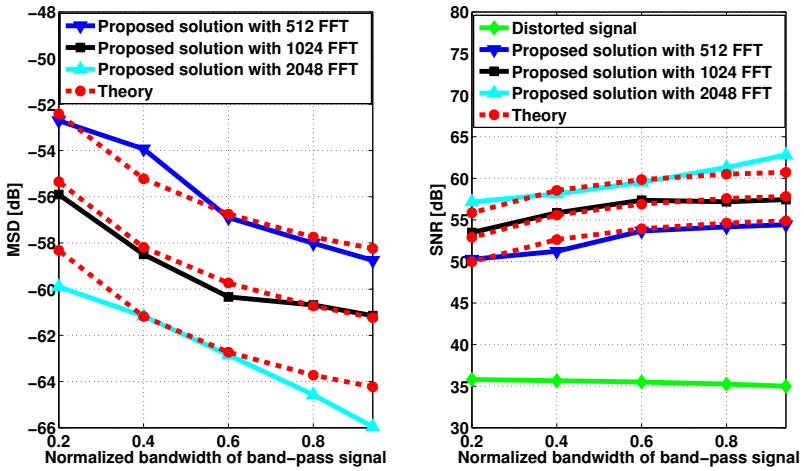


Figure 5.20: Performance of solution (a), which removes the first and second order distortions,as bandpass signal's bandwidth varies from 0.2π to 0.94π.

it performs as well as the prior works (which assume out-of-band region), and also in cases when the prior works fail. Using simulations, we verified that the derived performance measures match well with the experimental results.

5.A Proof for mean convergence

This appendix proves that conditions (5.90) and (5.92) ensure (5.95) and, hence, ensure mean stability in the long term regime as $i \to \infty$. To begin with, it is known that for any two matrices, the products XY and YX have the same eigenvalues, counting multiplicity and eigenvalues equal to 0 [113]. Therefore, $\rho(KB) = \rho(BK)$. Recall that matrix B is defined in (5.94). Using $\mathcal{M}$ from (5.77a), $\mathcal{R}$ from (5.78a), A_ℓ from (5.84b), A^T in (5.86c), and $R_{U,k}$ from (5.62),

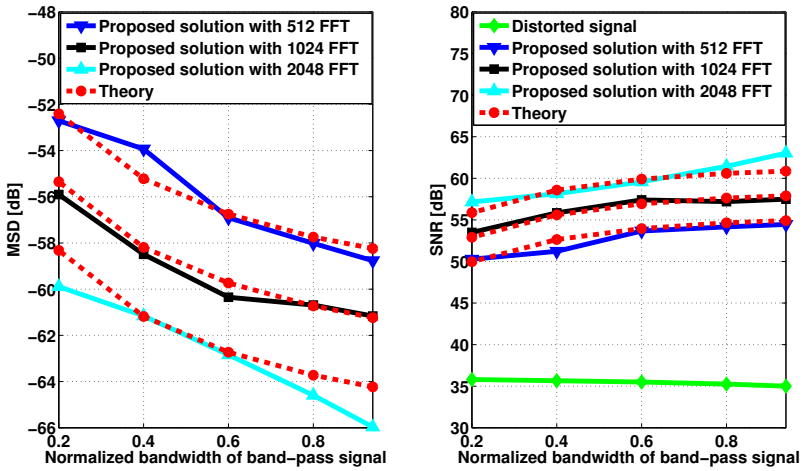


Figure 5.21: Performance of solution (b), which removes the first and second order distortions, as bandpass signal's bandwidth varies from 0.2π to 0.94π.

the block representation for B is shown in (5.111).

$$\begin{aligned} B &\triangleq A^T \left(I_{N(2M-2)} - \mathcal{M}\mathcal{R} \right) \\ &= A^T - \left[\begin{array}{ccc} \mu_0 A_0 R_{U,0} & \ldots & \mu_{N-1} A_{N-1} R_{U,N-1} \end{array} \right] \end{aligned} \tag{5.111}$$

Multiplying (5.111) by K from (5.86b), and using the property in (5.84c), we obtain the expression

$$\begin{aligned} BK &= \sum_{k=0}^{N-1} A_k - \sum_{k=0}^{N-1} \mu_k A_k R_{U,k} \\ &= I_{2M-2} - \sum_{k=0}^{N-1} \mu_k A_k R_{U,k} \\ &= \text{diag} \left\{ I_2 - \sum_{k=0}^{N-1} \mu_k a_k[0] R_{X,\alpha_{k,0}}, \ \ldots, \right. \\ &\qquad \left. I_2 - \sum_{k=0}^{N-1} \mu_k a_k[M-2] R_{X,\alpha_{k,M-2}} \right\} \end{aligned} \tag{5.112}$$

Using (5.112), mean stability holds when

$$\begin{aligned} 1 &> \rho\left(BK\right) \\ &= \max_{m=0,\ldots,M-2} \left\{ \rho\left(I_2 - \sum_{k=0}^{N-1} \mu_k a_k[m] R_{X,\alpha_{k,m}} \right) \right\} \end{aligned} \tag{5.113}$$

We note that the matrix sum $\sum_{k=0}^{N-1} \mu_\ell a_k[m] R_{X,\alpha_{k,m}}$ in (5.113) is Hermitian as $R_{X,\alpha_{k,m}}$ is Hermitian and the coefficients are real. This means that the matrix sum has real eigenvalues. Now, we need to verify that (5.113) holds using conditions (5.90) and (5.92). We establish (5.113) in two steps. The first step is to prove that the matrix sum has positive eigenvalues,

$$\lambda_{\min}\left(\sum_{k=0}^{N-1} \mu_k a_k[m] R_{X,\alpha_{k,m}} \right) > 0 \tag{5.114}$$

where $\lambda_{\min}(\cdot)$ represents the minimum eigenvalue of the matrix argument. The second step is to prove that the maximum eigenvalue of the matrix sum is bounded by 2:

$$\lambda_{\max}\left(\sum_{k=0}^{N-1} \mu_k a_k[m] R_{X,\alpha_{k,m}} \right) < 2 \tag{5.115}$$

The two steps (5.114) and (5.115) imply that the matrix sum contains eigenvalues between 0 and 2 so that

$$\rho\left(I_2 - \sum_{k=0}^{N-1} \mu_k a_k[m] R_{X,\alpha_{k,m}} \right) < 1 \tag{5.116}$$

Let us proceed with the first step. Using $R_{X,\ell}$ from (5.63), and the conjugate symmetry property of $H_d[k]$,we find that

$$\begin{aligned} R_{X,N-\ell} &= \sigma^2_{X,N-\ell} \begin{bmatrix} 1 & H_d[N-\ell] \\ (H_d[N-\ell])^* & |H_d[N-\ell]|^2 \end{bmatrix} \\ &= \sigma^2_{X,\ell} \begin{bmatrix} 1 & (H_d[\ell])^* \\ H_d[\ell] & |H_d[\ell]|^2 \end{bmatrix} \\ &= \mathrm{conj}\left(R_{X,\ell}\right) \end{aligned} \tag{5.117}$$

where the operator conj applies the conjugation operation elementwise. Note that $R_{X,N-\ell}$ and $R_{X,\ell}$ are both rank-1 matrices. Also, note that when ℓ is 0 or $\frac{N}{2}$, $R_{X,\ell}$ is real. We will show in the next paragraph that combining the conjugate pairs will make the sum of matrices in (5.114) contain only positive eigenvalues.

We assume that there exists some $\alpha_{k,m}$-th FFT channel and its conjugate channel that contain some signal content (i.e., $\sigma^2_{X,\alpha_{k,m}} > 0$). Let us verify that

$$\lambda_{\min}\left(\mu_k a_k[m] R_{X,\alpha_{k,m}} + \mu_{k'} a_{k'}[m] R_{X,\alpha_{k',m}}\right) > 0 \tag{5.118}$$

where k' represents the appropriate index that matches the conjugate counterpart of $R_{X,\alpha_{k,m}}$. Using the rank-1 decomposition of the two matrices from (5.63):

$$\begin{aligned}
\mu_k a_k[m] R_{X,\alpha_{k,m}} &= \mu_k a_k[m] \sigma^2_{X,\alpha_{k,m}} b_{\alpha_{k,m}} b^*_{\alpha_{k,m}} \\
\mu_{k'} a_{k'}[m] R_{X,\alpha_{k',m}} &= \mu_{k'} a_{k'}[m] \sigma^2_{X,\alpha_{k',m}} b_{\alpha_{k',m}} b^*_{\alpha_{k',m}} \\
b_{\alpha_{k,m}} &= \begin{bmatrix} 1 & (H_d[\alpha_{k,m}])^* \end{bmatrix}^T
\end{aligned} \tag{5.119}$$

Now, we denote

$$c_{k,m} \triangleq \mu_k a_k[m] \tag{5.120}$$

Using (5.90), we see that $c_{k,m} > 0$. We can write the sum in (5.118) as:

$$\begin{aligned}
& c_{k,m} R_{X,\alpha_{k,m}} + c_{k',m} R_{X,\alpha_{k',m}} \\
&= \sigma^2_{X,\alpha_{k,m}} \begin{bmatrix} b_{\alpha_{k,m}} & b_{\alpha_{k',m}} \end{bmatrix} \begin{bmatrix} c_{k,m} & 0 \\ 0 & c_{k',m} \end{bmatrix} \begin{bmatrix} b^*_{\alpha_{k,m}} \\ b^*_{\alpha_{k',m}} \end{bmatrix} \\
&= \Gamma_{\alpha_{k,m}} \Gamma^*_{\alpha_{k,m}}
\end{aligned} \tag{5.121}$$

where

$$\Gamma_{\alpha_{k,m}} = \sigma_{X,\alpha_{k,m}} \begin{bmatrix} b_{\alpha_{k,m}} & b_{\alpha_{k',m}} \end{bmatrix} \begin{bmatrix} \sqrt{c_{k,m}} & 0 \\ 0 & \sqrt{c_{k',m}} \end{bmatrix} \tag{5.122}$$

It can be verified that $b_{\alpha_{k,m}}$ and $b_{\alpha_{k',m}}$ are linearly independent and, hence, and $\Gamma_{\alpha_{k,m}}$ is non-singular and (5.121) is positive-definite. Hence, (5.118) is true. This

means that at least one pair sum from (5.114) is positive-definite and, therefore, the sum in (5.114) is positive-definite so that (5.114) holds.

Next, we cover the second step (5.115) of the argument. Using (5.92), we can bound (5.115) by:

$$\begin{aligned}\lambda_{\max}\left(\sum_{k=0}^{N-1}\mu_k a_k[m]R_{X,\alpha_{k,m}}\right) &\leq \sum_{k=0}^{N-1} a_k[m]\lambda_{\max}\left(\mu_k R_{X,\alpha_{k,m}}\right)\\ &< 2\sum_{k=0}^{N-1} a_k[m]\\ &< 2 \end{aligned} \tag{5.123}$$

5.B Proof for (5.1)

Consider an ideal M-channel TI-ADC with a sampling interval of T_s and a input signal of $x(t)$. Assume it is band-limited to the Nyquist frequency, i.e. $X(j\Omega) = 0$ for $|\Omega| \geq \frac{\Omega_s}{2}$, $\Omega_s = \frac{2\pi}{T_s}$. The sampled signal $x_s(t)$ is

$$x_s(t) = \sum_{m=0}^{M-1}\sum_{\ell=-\infty}^{\infty}\delta(t - \ell MT_s - mT_s)x(t) \tag{5.124}$$

Now, consider the M-channel TI-ADC has gain and timing mismatches of g_m and $r_m T_s$, respectively. The sampled signal $y_s(t)$ becomes

$$\begin{aligned} y_s(t) &= \sum_{m=0}^{M-1}\sum_{\ell=-\infty}^{\infty} g_m\delta(t - \ell MT_s - mT_s - r_m T_s)x(t)\\ &= \sum_{m=0}^{M-1}\sum_{\ell=-\infty}^{\infty} g_m\delta(t - \ell MT_s - mT_s)x(t + r_m T_s)\\ &= \sum_{m=0}^{M-1}\left\{\left[\sum_{\ell=-\infty}^{\infty}\delta(t - \ell MT_s - mT_s)\right] g_m x(t + r_m T_s)\right\} \end{aligned} \tag{5.125}$$

First, we state the below well-known Fourier transform (FT) properties (impulse train, time-delay and multiplication in time):

$$\mathcal{F}\left\{\sum_{n=-\infty}^{\infty}\delta\left(t-nT_s\right)\right\}=\Omega_s\sum_{k=-\infty}^{\infty}\delta\left(j\Omega-jk\Omega_s\right) \tag{5.126a}$$

$$\mathcal{F}\left\{f(t-\tau)\right\}=F(j\Omega)e^{-j\Omega\tau} \tag{5.126b}$$

$$\mathcal{F}\left\{f(t)g(t)\right\}=\frac{1}{2\pi}F(j\Omega)*G(j\Omega) \tag{5.126c}$$

where $\Omega_s=\frac{2\pi}{T_s}$. Therefore, when an impulse train (with a sampling interval of MT_s) is delayed by mT_s, the FT becomes

$$\mathcal{F}\left\{\sum_{n=-\infty}^{\infty}\delta\left(t-(nM+m)T_s\right)\right\}=\left[\frac{\Omega_s}{M}\sum_{k=-\infty}^{\infty}\delta\left(j\Omega-j\frac{\Omega_s}{M}k\right)\right]e^{-j\Omega mT_s} \tag{5.127a}$$

$$=\frac{\Omega_s}{M}\sum_{k=-\infty}^{\infty}\delta\left(j\Omega-j\frac{\Omega_s}{M}k\right)e^{-j\frac{2\pi}{M}mk} \tag{5.127b}$$

where we use (5.126a) and (5.126b) to obtain (5.127a), and substitute $\Omega=\frac{\Omega_s}{M}k$ into $e^{-j\Omega mT_s}$ in (5.127a) to obtain (5.127b). Also, the FT of $g_m x(t+r_mT_s)$ is

$$\mathcal{F}\left\{g_m x(t+r_mT_s)\right\}=X(j\Omega)H_m(j\Omega) \tag{5.128a}$$

$$H_m(j\Omega)=\begin{cases}g_m e^{j\Omega r_mT_s} & \text{if } |\Omega|<\frac{\Omega_s}{2}\\ 0 & \text{otherwise}\end{cases} \tag{5.128b}$$

where we use the assumption that the signal is bandlimited and hence $H_m(j\Omega)$ is written as a bandlimited transfer function. Now, recall the following convolution property (associativity with scalar multiplication):

$$a[f(t)*g(t)]=[af(t)]*g(t) \tag{5.129}$$

Hence, the FT of $y_s(t)$ in (5.125) is denoted as $Y_s(j\Omega)$:

$$\begin{aligned} Y_s(j\Omega) &= \sum_{m=0}^{M-1} \frac{1}{2\pi} X(j\Omega) H_m(j\Omega) * \left[\frac{\Omega_s}{M} \sum_{k=-\infty}^{\infty} \delta\left(j\Omega - j\frac{\Omega_s}{M} k \right) e^{-j\frac{2\pi}{M}mk} \right] \\ &= \sum_{m=0}^{M-1} \sum_{k=-\infty}^{\infty} \frac{1}{MT_s} X\left(j\Omega - j\frac{\Omega_s}{M} k \right) H_m\left(j\Omega - j\frac{\Omega_s}{M} k \right) e^{-j\frac{2\pi}{M}mk} \end{aligned} \tag{5.130}$$

We define $\bar{H}_k(j\Omega)$ as

$$\bar{H}_k\left(j\Omega\right) \triangleq \begin{cases} \frac{1}{M} \sum_{m=0}^{M-1} H_m\left(j\Omega\right) e^{-j\frac{2\pi}{M}mk} & \text{if } |\Omega| < \frac{\Omega_s}{2} \\ 0 & \text{otherwise} \end{cases} \tag{5.131}$$

Now, we can rewrite (5.130) as

$$\begin{aligned} Y_s(j\Omega) = &\sum_{k=-\infty}^{\infty} \frac{1}{T_s} X\left(j\Omega - j\frac{\Omega_s}{M} k \right) \\ &\left[\frac{1}{M} \sum_{m=0}^{M-1} H_m\left(j\Omega - j\frac{\Omega_s}{M} k \right) e^{-j\frac{2\pi}{M}mk} \right] \end{aligned} \tag{5.132}$$

$$= \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X\left(j\Omega - j\frac{\Omega_s}{M} k \right) \bar{H}_k\left(j\Omega - j\frac{\Omega_s}{M} k \right) \tag{5.133}$$

The relationship between discrete-time Fourier transform (DTFT) of a sampled signal $g[n]$ and the FT of sampled $g_s(t)$ is

$$G(e^{j\omega}) = G_s\left(j\Omega\right)|_{\Omega=\frac{\omega}{T_s}} \tag{5.134a}$$

$$G_s(j\Omega) = \frac{1}{T_s} \left(\sum_{k=-\infty}^{\infty} G(j\Omega - jk\Omega_s) \right) \tag{5.134b}$$

Hence, from (5.133), DTFT of $y[n]$ is related to the FT of $y_s(t)$ is

$$\begin{aligned} Y(e^{j\omega}) &= Y_s\left(j\Omega\right)|_{\Omega=\frac{\omega}{T_s}} \\ &= \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X\left(j\Omega - j\frac{\Omega_s}{M} k \right) \bar{H}_k\left(j\Omega - j\frac{\Omega_s}{M} k \right) \Big|_{\Omega=\frac{\omega}{T_s}} \end{aligned} \tag{5.135}$$

From (5.128), we see that the DTFT counterpart of $X(j\Omega)$ and $H_m(j\Omega)$ are

$$X(e^{j\omega}) = \frac{1}{T_s}\left(\sum_{k=-\infty}^{\infty} X(j\Omega - jk\Omega_s)\right)\Bigg|_{\Omega=\frac{\omega}{T_s}} \tag{5.136a}$$

and

$$\begin{aligned} H_m(e^{j\omega}) &= \frac{1}{T_s}\left(\sum_{k=-\infty}^{\infty} H_m(j\Omega - jk\Omega_s)\right)\Bigg|_{\Omega=\frac{\omega}{T_s}} \\ &= \frac{1}{T_s} g_m e^{j\Omega r_m T_s}|_{\Omega=\frac{\omega}{T_s}} \\ &= \frac{1}{T_s} g_m e^{j\omega r_m} \end{aligned} \tag{5.137}$$

Therefore, the DTFT counterpart of $\bar{H}_\ell(j\Omega)$ in (5.131) is

$$\begin{aligned} \bar{H}_\ell(e^{j\omega}) &= \frac{1}{T_s}\left(\sum_{k=-\infty}^{\infty} \bar{H}_\ell(j\Omega - jk\Omega_s)\right)\Bigg|_{\Omega=\frac{\omega}{T_s}} \\ &= \frac{1}{T_s}\left(\sum_{k=-\infty}^{\infty} \frac{1}{M}\sum_{m=0}^{M-1} H_m\left(j\Omega - jk\Omega_s\right) e^{-j\frac{2\pi}{M}m\ell}\right)\Bigg|_{\Omega=\frac{\omega}{T_s}} \\ &= \frac{1}{M}\sum_{m=0}^{M-1}\left(\frac{1}{T_s}\sum_{k=-\infty}^{\infty} H_m\left(j\Omega - jk\Omega_s\right)|_{\Omega=\frac{\omega}{T_s}}\right) e^{-j\frac{2\pi}{M}m\ell} \\ &= \frac{1}{M}\sum_{m=0}^{M-1} H_m(e^{j\omega}) e^{-j\frac{2\pi}{M}m\ell} \end{aligned} \tag{5.138}$$

The bandlimited property of $X(j\Omega)$ and $\bar{H}(j\Omega)$ in (5.131) means

$$X(j\Omega - jk_1\Omega_s)\bar{H}_\ell(j\Omega - k_2\ell\Omega_s) = 0,\ \forall\ k_1 \neq k_2 \tag{5.139}$$

Therefore, we can derive $X\left(e^{j\omega}\right)\bar{H}_\ell\left(e^{j\omega}\right)$ as

$$\begin{aligned} X\left(e^{j\omega}\right)\bar{H}_\ell\left(e^{j\omega}\right) &= \frac{1}{T_s}\left(\sum_{k_1=-\infty}^{\infty} X(j\Omega - jk_1\Omega_s)\right)\Bigg|_{\Omega=\frac{\omega}{T_s}} \\ &\quad \frac{1}{T_s}\left(\sum_{k_2=-\infty}^{\infty} \bar{H}_\ell(j\Omega - jk_2\Omega_s)\right)\Bigg|_{\Omega=\frac{\omega}{T_s}} \\ &= \frac{1}{T_s^2}\left(\sum_{k=-\infty}^{\infty} X(j\Omega - jk\Omega_s)\bar{H}_\ell(j\Omega - jk\Omega_s)\right)\Bigg|_{\Omega=\frac{\omega}{T_s}} \end{aligned} \tag{5.140}$$

Now in (5.135), we let $k = k'M + \ell$ where $\ell = 0, 1, ..., M-1$, to obtain

$$\begin{aligned} Y(e^{j\omega}) &= \frac{1}{T_s} \sum_{k'M+\ell=-\infty}^{\infty} X\left(j\Omega - j\frac{\Omega_s}{M}(k'M+\ell)\right) \\ &\qquad \bar{H}_{k'M+\ell}\left(j\Omega - j\frac{\Omega_s}{M}(k'M+\ell)\right)\Big|_{\Omega=\frac{\omega}{T_s}} \\ &= \frac{1}{T_s} \sum_{k'=-\infty}^{\infty} \sum_{\ell=0}^{M-1} X\left(j\Omega - j\frac{\Omega_s(k'M+\ell)}{M}\right) \\ &\qquad \bar{H}_{k'M+\ell}\left(j\Omega - j\frac{\Omega_s(k'M+\ell)}{M}\right)\Big|_{\Omega=\frac{\omega}{T_s}} \\ &= \sum_{\ell=0}^{M-1} \frac{1}{T_s} \sum_{k'=-\infty}^{\infty} X\left(j\left(\Omega - \frac{2\pi\ell}{MT_s}\right) - j\Omega_s k'\right) \\ &\qquad \bar{H}_{\ell}\left(j\left(\Omega - \frac{2\pi\ell}{MT_s}\right) - j\Omega_s k'\right)\Big|_{\Omega=\frac{\omega}{T_s}} \end{aligned} \tag{5.141}$$

where we use $\Omega_s = \frac{2\pi}{T_s}$ to obtain the term $\frac{2\pi\ell}{MT_s}$ in the last expression, and the observation from (5.131) that $\bar{H}_{\ell+k'M}(j\Omega) = \bar{H}_{\ell}(j\Omega)$. It can be verified that the above expression (5.141) is the same as the one below (5.142a):

$$Y(e^{j\omega}) = \sum_{\ell=0}^{M-1} \frac{1}{T_s} \sum_{k'=-\infty}^{\infty} X\left(j\Omega - j\Omega_s k'\right) \bar{H}_{\ell}\left(j\Omega - j\Omega_s k'\right)\Big|_{\Omega=\frac{1}{T_s}\left(\omega-\frac{2\pi\ell}{M}\right)} \tag{5.142a}$$

$$= \sum_{\ell=0}^{M-1} T_s X\left(e^{j\left(\omega-\frac{2\pi\ell}{M}\right)}\right) \bar{H}_{\ell}\left(e^{j\left(\omega-\frac{2\pi\ell}{M}\right)}\right) \tag{5.142b}$$

where (5.142b) is obtained from (5.140) by replacing ω by $\omega - \frac{2\pi\ell}{M}$. Now, using (5.138) and (5.137), we can define

$$\begin{aligned} \check{H}_{\ell}\left(e^{j\omega}\right) &\triangleq T_s \bar{H}_{\ell}\left(e^{j\omega}\right) \\ &= \frac{T_s}{M} \sum_{m=0}^{M-1} H_m\left(e^{j\omega}\right) e^{-j\frac{2\pi}{M}m\ell} \\ &= \frac{T_s}{M} \sum_{m=0}^{M-1} \frac{1}{T_s} g_m e^{j\omega r_m} e^{-j\frac{2\pi}{M}m\ell} \\ &= \frac{1}{M} \sum_{m=0}^{M-1} g_m e^{j\omega r_m} e^{-j\frac{2\pi}{M}m\ell} \end{aligned} \tag{5.143}$$

Therefore, from (5.143) and (5.142b), we obtain (5.1) as

$$Y(e^{j\omega}) = \sum_{k=0}^{M-1} X\left(e^{j(\omega-\frac{2\pi k}{M})}\right) \check{H}_k\left(e^{j(\omega-\frac{2\pi k}{M})}\right) \tag{5.144}$$

CHAPTER 6

Conclusion and Future Research

In this dissertation, we examined the effect of spurious sidebands and random jitter in ADCs, as well as, timing and gain mismatches in TI-ADCs. We developed and analyzed adaptive DSP techniques to correct for these circuit imperfections. Our simulations and theoretical results show that the proposed solutions reduce the distortions in the sampled data to great effect. There are a couple of issues that deserve further investigation. We list three of them here.

One future research direction is to estimate the spurious sidebands without the use of a training signal (as compared to the techniques in Chapter 2 and 3). Such a possibility would be attractive because it imposes less requirements on the circuit design process and on the hardware implementation.

Another research direction is to study how random jitter in ADC impacts applications like spectrum sensing. By examining the impact in terms of its influence on the probabilities of detection and false alarm, the results can be used to complement analyses that rely mainly on assessing the SNR of the recovered signals.

A third research direction is to extend the gain and timing mismatches in TI-ADC to frequency response mismatches [69, 89, 93]. The gain and timing mismatches can be modeled as special cases of frequency response mismatches. These extensions are more demanding and would incorporate a wider range of

imperfections into the design models.

References

[1] C. Vogel, S. Saleem, and S. Mendel, "Adaptive blind compensation of gain and timing mismatches in M-channel time-interleaved ADCs," in *Proc. 15th IEEE International Conference on Electronics, Circuits and Systems*, Malta, Aug 2008, pp. 49–52.

[2] J. Matsuno, T. Yamaji, M. Furuta, and T. Itakura, "All-digital background calibration technique for time-interleaved ADC using pseudo aliasing signal," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 5, pp. 1113–1121, 2013.

[3] H. Arslan, *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems*. Springer-Verlag, New York, 2007.

[4] N. Al-Dhahir, A. Dabak, M. Moonen, A. H. Sayed, and Z. Zvonar, "Introduction to the issue on DSP techiques for RF/analog circuit impairments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 3, pp. 345–347, 2009.

[5] M. Valkama, A. Springer, and G. Hueber, "Digital signal processing for reducing the effects of RF imperfections in radio devices — An overview," in *Proc. IEEE ISCAS*, Paris, France, May 2010, pp. 813–816.

[6] Y. Zou, M. Valkama, and M. Renfors, "Analysis and compensation of transmitter and receiver I/Q imbalances in space-time coded multiantenna OFDM systems," *EURASIP J. Wirel. Commun. Netw.*, vol. 2008, pp. 1–10, 2008.

[7] A. Tarighat, R. Bagheri, and A. H. Sayed, "Compensation schemes and performance analysis of IQ imbalances in OFDM receivers," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3257–3268, Aug. 2005.

[8] Q. Zou, A. Tarighat, and A. H. Sayed, "Joint compensation of IQ imbalance and phase noise in OFDM wireless systems," *IEEE Transactions on Communications*, vol. 57, no. 2, pp. 404–414, Feb. 2009.

[9] ——, "Compensation of phase noise in OFDM wireless systems," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5407–5424, 2007.

[10] M. Valkama, M. Renfors, and V. Koivunen, "Blind I/Q signal separation-based solutions for receiver signal processing," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 2708–2718, 2005.

[11] D. Tandur and M. Moonen, "Compensation of RF impairments in MIMO OFDM systems," in *Proc. IEEE ICASSP*, Las Vegas, Neveda, Mar 2008, pp. 3097–3100.

[12] B. Narasimhan, D. Wang, S. Narayanan, H. Minn, and N. Al-Dhahir, "Digital compensation of frequency-dependent joint Tx/Rx I/Q imbalance in OFDM systems under high mobility," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 3, pp. 405–417, 2009.

[13] B. Narasimhan, S. Narayanan, H. Minn, and N. Al-Dhahir, "Reduced-complexity baseband compensation of joint Tx/Rx I/Q imbalance in mobile MIMO-OFDM," *IEEE Transactions on Wireless Communications*, vol. 9, no. 5, pp. 1720–1728, 2010.

[14] J. Elbornsson, F. Gustafsson, and J.-E. Eklund, "Blind adaptive equalization of mismatch errors in a time-interleaved A/D converter system," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 1, pp. 151–158, Jan. 2004.

[15] S. Huang and B. C. Levy, "Blind calibration of timing offsets for four-channel time-interleaved ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 4, pp. 863–876, April 2007.

[16] C. Fu and H. Wang, "Adaptive optimization in A/D converter system," *6th Proc. IEEE International Conference on Industrial Informatics, Daejeon, Korea*, pp. 627–630, July 2008.

[17] S.-K. Ting and A. H. Sayed, "Digital suppression of spurious PLL tones in A/D converters," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5275–5288, 2011.

[18] S.-K. Ting, D. Cabric, and A. H. Sayed, "An algorithm to compensate the effects of spurious PLL tones in spectrum sensing architectures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 5, pp. 1093–1106, 2012.

[19] Z. J. Towfic, S.-K. Ting, and A. H. Sayed, "Clock jitter compensation in high-rate ADC circuits," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5738–5753, 2012.

[20] S.-K. Ting, H. Chen, A. A. Abidi, and A. H. Sayed, "Adaptive compensation of mismatches in time-interleaved A/D converters - Part I: Compenstation structures," *submitted for publication*, 2014.

[21] ——, "Adaptive compensation of mismatches in time-interleaved A/D converters - Part II: Performance analysis," *submitted for publication*, 2014.

[22] R. Best, *Phase-Locked Loops, Design, Simulation, and Applications, 5th Edition.* McGraw-Hill, 2003.

[23] F. Gardner, "Charge-pump phase-lock loops," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1849–1858, 1980.

[24] T.-C. Lee and W.-L. Lee, "A spur suppression technique for phase-locked frequency synthesizers," in *Proc. ISSCC*, San Francisco, USA, Feb. 2006, pp. 2432–2441.

[25] H. Wang, G. Shou, and N. Wu, "An adaptive frequency synthesizer architecture reducing reference sidebands," in *Proc. ISCAS*, Island of Kos, Greece, 2006, pp. 3081–3084.

[26] Y. Poberezhskiy and G. Poberezhskiy, "Sample-and-hold amplifiers performing internal antialiasing filtering and their applications in digital receivers," in *Proc. IEEE ISCAS*, vol. 3, Geneva Switzerland, May 2000, pp. 439–442.

[27] ——, "Flexible analog front ends of reconfigurable radios based on sampling and reconstruction with internal filtering," *EURASIP J. Wirel. Commun. Netw.*, vol. 2005, no. 3, pp. 364–381, 2005.

[28] A. Mirzaei, S. Chehrazi, R. Bagheri, and A. A. Abidi, "Analysis of first-order anti-aliasing integration sampler," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 10, pp. 2994–3005, 2008.

[29] ——, "A second-order antialiasing prefilter for a software-defined radio receiver," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 7, pp. 1513–1524, 2009.

[30] R. Rutten, L. J. Breems, and R. H. M. van Veldhoven, "Digital jitter-cancellation for narrowband signals," in *Proc. IEEE ISCAS*, Seattle, Washington, May 2008, pp. 1444–1447.

[31] V. Syrjala and M. Valkama, "Jitter mitigation in high-frequency bandpass-sampling OFDM radios," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC)*, Budapest, Hungary, Apr 2009, pp. 1–6.

[32] ——, "Sampling jitter estimation and mitigation in direct RF sub-sampling receiver architecture," in *Proc. 6th Int. Symp. Wireless Communication Systems (ISWCS)*, Italy, Sep 2009, pp. 323–327.

[33] ——, "Sampling jitter cancellation in direct-sampling radio," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC)*, Sydney, Australia, Apr 2010, pp. 1–6.

[34] H. Kobayashi, K. Kobayashi, Y. Takahashi, K. Enomoto, H. Kogure, Y. Onaya, and M. Morimura, "Finite aperture time and sampling jitter effects in wideband data acquisition systems," in *Proc. ARFTG Conf. Digest*, vol. 38, Anaheim, CA, May 2000, pp. 1–7.

[35] P. Nikaeen and B. Murmann, "Digital compensation of dynamic acquisition errors at the front-end of high-performance A/D converters," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 3, pp. 499–508, 2009.

[36] S.-K. Ting and A. H. Sayed, "Reduction of the effects of spurious PLL tones on A/D converters," in *Proc. IEEE ISCAS*, Paris, France, May 2010, pp. 3985–3988.

[37] G.-C. Hsieh and J. C. Hung, "Phase-locked loop techniques: A survey," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 6, pp. 609–615, Dec. 1996.

[38] W. Egan, *Phase-Lock Basics*, 2nd ed. Wiley-IEEE Press, 2007.

[39] ——, *Frequency Synthesis by Phase Lock*. Wiley, 2000.

[40] F. Gardner, *Phaselock Techniques*. Wiley-Blackwell, 2005.

[41] Z. J. Towfic, S.-K. Ting, and A. H. Sayed, "Sampling clock jitter estimation and compensation in ADC circuits," in *Proc. IEEE ISCAS*, France, May 2010, pp. 829–832.

[42] A. H. Sayed, *Adaptive Filters*. NJ: Wiley, 2008.

[43] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory, Vol I.* Prentice Hall, 1993.

[44] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. McGraw-Hill, 1976.

[45] D. Cabric, I. D. O'Donnell, M. S.-W. Chen, and R. W. Brodersen, "Spectrum sharing radios," *IEEE Circuits and Systems Magazine*, vol. 6, no. 2, pp. 30–45, 2006.

[46] D. Cabric, "Addressing feasibility of cognitive radios," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 85–93, 2008.

[47] R. Tandra and A. Sahai, "SNR walls for signal detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 4–17, 2008.

[48] Z. Quan, W. Zhang, S. J. Shellhammer, and A. H. Sayed, "Optimal spectral feature detection for spectrum sensing at very low SNR," *IEEE Transactions on Communications*, vol. 59, no. 1, pp. 201–212, 2011.

[49] T. Yu, S. Rodriguez-Parera, D. Markovic, and D. Cabric, "Cognitive radio wideband spectrum sensing using multitap windowing and power detection with threshold adaptation," in *Proc. IEEE Int. Conf. Communications (ICC)*, Cape Town, South Africa, May 2010, pp. 1–6.

[50] T.-H. Yu, O. Sekkat, S. Rodriguez-Parera, D. Markovic, and D. Cabric, "A wideband spectrum-sensing processor with adaptive detection threshold and sensing time," *IEEE Transactions on Circuits and Systems I: Regular Papers*, no. 99, 2011, early Access.

[51] W.-B. Chien, C.-K. Yang, and Y.-H. Huang, "Energy-saving cooperative spectrum sensing processor for cognitive radio system," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 4, pp. 711–723, 2011.

[52] R. Mahesh and A. P. Vinod, "A low-complexity flexible spectrum-sensing scheme for mobile cognitive radio terminals," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 6, pp. 371–375, 2011.

[53] T. Song, J. Park, S. M. Lee, J. Choi, K. Kim, C.-H. Lee, K. Lim, and J. Laskar, "A 122-mW low-power multiresolution spectrum-sensing IC with self-deactivated partial swing techniques," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 3, pp. 188–192, 2010.

[54] L. Luo, N. M. Neihart, S. Roy, and D. J. Allstot, "A two-stage sensing technique for dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, pp. 3028–3037, 2009.

[55] S.-K. Ting and A. H. Sayed, "Reducing spurious PLL tones in spectrum sensing architectures," in *Proc. IEEE ISCAS*, Rio de Janeiro, Brazil, May 2011, pp. 2445–2448.

[56] A. Oppenheim, R. Schafer, and J. Buck, *Discrete-Time Signal Processing*. Prentice-Hall, 1999.

[57] S. Tertinek and C. Vogel, "Reconstruction of nonuniformly sampled bandlimited signals using a differentiator–multiplier cascade," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 8, pp. 2273–2286, 2008.

[58] J. Selva, "Functionally weighted Lagrange interpolation of band-limited signals from nonuniform samples," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 168–181, 2009.

[59] ——, "Design of barycentric interpolators for uniform and nonuniform sampling grids," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1618–1627, 2010.

[60] S. M. Kay, *Fundamentals of Statistical Processing: Detection Theory*. Prentice Hall, 1998.

[61] R. Durrett, *Probability: Theory and Examples*. Cambridge Univ Pr, 2010.

[62] B. Razavi, *RF Microelectronics*. Prentice-Hall, Inc, NJ, 2011.

[63] M. Shinagawa, Y. Akazawa, and T. Wakimoto, "Jitter analysis of high-speed sampling systems," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 1, pp. 220–224, Feb. 1990.

[64] Z. J. Towfic and A. H. Sayed, "Clock jitter estimation in noise," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, Rio de Janeiro, Brazil, May 2011, pp. 1251–1254.

[65] M. Löhning and G. Fettweis, "The effects of aperture jitter and clock jitter in wideband ADCs," *Computer Standards & Interfaces*, vol. 29, no. 1, pp. 11–18, 2007.

[66] V. Syrjälä and M. Valkama, "Analysis and mitigation of phase noise and sampling jitter in OFDM radio receivers," *International Journal of Microwave and Wireless Technologies*, vol. 2, no. 2, pp. 193–202, April 2010.

[67] M. Valkama, J. Pirskanen, and M. Renfors, "Signal processing challenges for applying software radio principles in future wireless terminals: an overview," *International Journal of Communication Systems*, vol. 15, no. 8, pp. 741–769, Oct. 2002.

[68] C. Vogel and H. Johansson, "Time-interleaved analog-to-digital converters: status and future directions," in *Proc. IEEE ISCAS*, Island of Kos, Greece, May 2006, pp. 3386–3389.

[69] H. Johansson and P. Löwenborg, "A least-squares filter design technique for the compensation of frequency response mismatch errors in time-interleaved A/D converters," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 11, pp. 1154–1158, Nov. 2008.

[70] H. Johansson, "A polynomial-based time-varying filter structure for the compensation of frequency-response mismatch errors in time-interleaved ADCs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 3, pp. 384–396, 2009.

[71] H. Johansson and P. Löwenborg, "Reconstruction of nonuniformly sampled bandlimited signals by means of time-varying discrete-time FIR filters," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, no. 1, pp. 1–18, Jan. 2006.

[72] N. Testoni, N. Speciale, A. Ridolfi, and C. Pouzat, "Adaptive wavelet-based signal dejittering," *Research in Microelectronics and Electronics Conference, Talence, France*, pp. 257–260, July 2007.

[73] G. L. Fudge, M. A. Chivers, S. Ravindran, and A. Yeh, "System and method for clock jitter compensation in direct RF receiver architectures," U.S. Patent 2010/0 189 208 A1, Jul. 21, 2010.

[74] B. Razavi, "The role of PLLs in future wireline transmitters," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 8, pp. 1786–1793, Aug. 2009.

[75] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*. Academic Press, New York, 2007.

[76] A. J. Viterbi, "Phase-locked loop dynamics in the presence of noise by Fokker-Planck techniques," *Proc. of the IEEE*, vol. 51, no. 12, pp. 1737–1753, Dec. 1963.

[77] A. Zaziabl, "Low power 1 GHz charge pump phase-locked loop in 0.18 μm CMOS process," in *Proc. of the 17th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES)*, Gdynia, Poland, Jun. 2010, pp. 277–282.

[78] N. Da Dalt, M. Harteneck, C. Sandner, and A. Wiesbauer, "On the jitter requirements of the sampling clock for analog-to-digital converters," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 9, pp. 1354–1360, Sep. 2002.

[79] P. F. Horlin and A. Bourdoux, *Digital Compensation for Analog Front-Ends: A New Approach to Wireless Transceiver Design*. Wiley, NJ, 2008.

[80] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. Mc-Graw Hill, 2002.

[81] C. R. J. Jr and W. A. Sethares, *Telecommunication Breakdown: concepts of communications transmitted via software-defined radio.* Prentice Hall, 2004.

[82] J. Mathews and K. Fink, *Numerical Methods Using MATLAB.* Simon & Schuster, NY, 1998.

[83] R. W. Hamming, *Digital Filters.* Dover Publications, 1998.

[84] J. Barry, E. Lee, and D. Messerschmitt, *Digital Communication*, 3rd ed. Springer, MA, 2004.

[85] J. G. Proakis, *Digital Communications.* McGraw-Hill, NY, 2000.

[86] W. Egan, "Modeling phase noise in frequency dividers," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 37, no. 4, pp. 307–315, Jul. 1990.

[87] A. Demir, A. Mehrotra, and J. Roychowdhury, "Phase noise in oscillators: A unifying theory and numerical methods for characterization," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 5, pp. 655–674, 2000.

[88] R. McDonough and A. Whalen, *Detection of Signals in Noise.* Academic Press, 1995.

[89] S. Saleem and C. Vogel, "Adaptive blind background calibration of polynomial-represented frequency response mismatches in a two-channel time-interleaved ADC," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 6, pp. 1300–1310, 2011.

[90] M. Seo, M. Rodwell, and U. Madhow, "Generalized blind mismatch correction for two-channel time-interleaved A-to-D converters," in *Proc. ICASSP*, vol. 3, Honolulu, Hawaii, USA, Apr 2007, pp. 1505–1508.

[91] M. Seo and M. Rodwell, "Generalized blind mismatch correction for a two-channel time-interleaved ADC: Analytic approach," in *Proc. IEEE ISCAS*, New Orleans, USA, May 2007, pp. 109–112.

[92] W. Liu and Y. Chiu, "Time-interleaved analog-to-digital conversion with online adaptive equalization," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 7, pp. 1384–1395, 2012.

[93] S. Saleem and C. Vogel, "Adaptive compensation of frequency response mismatches in high-resolution time-interleaved ADCs using a low-resolution ADC and a time-varying filter," in *Proc. IEEE ISCAS*, Paris, France, May 2010, pp. 561–564.

[94] T. Strohmer and J. Xu, "Fast algorithms for blind calibration in time-interleaved analog-to-digital converters," in *Proc. IEEE ICASSP*, vol. 3, Honolulu, Hawaii, USA, Apr 2007, pp. III–1225–III–1228.

[95] V. Divi and G. Wornell, "Bandlimited signal reconstruction from noisy periodic nonuniform samples in time-interleaved ADCs," in *Proc. IEEE ICASSP*, Las Vegas, Nevada, USA, March 2008, pp. 3721–3724.

[96] V. Divi and G. W. Wornell, "Blind calibration of timing skew in time-interleaved analog-to-digital converters," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 3, pp. 509–522, 2009.

[97] S. Huang and B. Levy, "Adaptive blind calibration of timing offset and gain mismatch for two-channel time-interleaved ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 6, pp. 1278–1288, 2006.

[98] S. Saleem and C. Vogel, "LMS-based identification and compensation of timing mismatches in a two-channel time-interleaved analog-to-digital converter," in *Proc. of the 25th IEEE Norchip Conference,*, Aalborg, Nov. 2007, pp. 1–4.

[99] Y. X. Zou and X. J. Xu, "Blind timing skew estimation using source spectrum sparsity in time-interleaved ADCs," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 9, pp. 2401–2412, 2012.

[100] Y. Zou, B. Li, and X. Chen, "An efficient blind timing skews estimation for time-interleaved analog-to-digital converters," in *Proc. 17th International Conference on Digital Signal Processing (DSP)*, Corfu, Greece, July 2011, pp. 1–4.

[101] B. Yu, C. Chen, Y. Zhu, P. Zhang, Y. Zhang, X. Zhu, F. Ye, and J. Ren, "A 14-bit 200-MS/s time-interleaved ADC with sample-time error detection and cancelation," in *Proc. Asian Solid State Circuits Conference (A-SSCC)*, Jeju, Korea, Nov 2011, pp. 349–352.

[102] S. Jamal, D. Fu, N.-J. Chang, P. Hurst, and S. Lewis, "A 10-b 120-Msample/s time-interleaved analog-to-digital converter with digital background calibration," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1618–1627, 2002.

[103] S. Jamal, D. Fu, M. Singh, P. Hurst, and S. Lewis, "Calibration of sample-time error in a two-channel time-interleaved analog-to-digital converter," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 1, pp. 130–139, 2004.

[104] C. H. Law, P. Hurst, and S. Lewis, "A four-channel time-interleaved ADC with digital calibration of interchannel timing and memory errors," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 10, pp. 2091–2103, 2010.

[105] J. Pearl, "On coding and filtering stationary signals by discrete Fourier transforms," *IEEE Transactions on Information Theory*, vol. 19, no. 2, pp. 229–232, 1973.

[106] J. Schoukens and J. Renneboog, "Modeling the noise influence on the Fourier coefficients after a discrete Fourier transform," *IEEE Transactions on Instrumentation and Measurement*, vol. IM-35, no. 3, pp. 278–286, 1986.

[107] L. N. Trefethen, *Spectral Methods in Matlab*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000.

[108] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, May 2013.

[109] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Academic Press Elsevier, 2013, vol. 3, pp. 323–454, also available as arXiv:1205.4220 [cs.MA], May 2012.

[110] F. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[111] P. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures, and applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 9, pp. 1893–1907, 2009.

[112] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5107–5124, Oct 2012.

[113] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY, USA: Cambridge University Press, 1985.