Rapid Frequency Estimation

by

Antti E. Koski

A Thesis
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Master of Science
in
Electrical and Computer Engineering
by

May 2006

APPROVED:

Professor David Cyganski, Major Advisor

Professor Sergey Makarov

Professor James Duckworth

Abstract

Frequency estimation plays an important role in many digital signal processing applications. Many areas have benefited from the discovery of the Fast Fourier Transform (FFT) decades ago and from the relatively recent advances in modern spectral estimation techniques within the last few decades. As processor and programmable logic technologies advance, unconventional methods for rapid frequency estimation in white Gaussian noise should be considered for real time applications. In this thesis, a practical hardware implementation that combines two known frequency estimation techniques is presented, implemented, and characterized. The combined implementation, using the well known FFT and a less well known modern spectral analysis method known as the Direct State Space (DSS) algorithm, is used to demonstrate and promote application of modern spectral methods in various real time applications, including Electronic Counter Measure (ECM) techniques.

Contents

List of Figures

Chapter 1

Introduction

Signal parameter estimation in the presence of noise has long been a focus area of research. Many applications have benefited from advancements made in this field within the last two decades. This thesis presents a novel approach suitable for implementation in a Field Programmable Gate Array (FPGA) to achieve rapid frequency estimation. Several methods will be discussed followed by selection of an approach, implementation details, and results.

1.1 Problem Statement

Electronic Counter Measure (ECM) techniques have long been employed in the battlefield to deceive the enemy for a variety of reasons. As L. Neng-Jing and Z. Yi-Ting point out in [1], some of the AN/ALQ ECM series played an important role in several military conflicts. Generally, the first steps in ECM employment require detection, identification, and classification of the threat system being countered [2]. The faster these steps occur, the more effective ECM techniques can become. One of the first stages of ECM employment, signal parameter estimation, will be the focus of this thesis.

Field Programmable Gate Array (FPGA) architectures allow complex Digital Signal Processing (DSP) algorithms to be implemented efficiently in a flexible programmable chip. Compared to conventional sequential DSP processors, FPGA implementations of parallel designs can generally processes digital data much faster, making FPGAs attractive for real time processing. As FPGA technology has progressed and the gate density has increased, the capabilities to implement rapid frequency estimates in the presence of white Gaussian noise have also progressed. This progression partially prompts the contents of this thesis as it presents the theoretical performance of several common frequency estimators and the results of a selected implementation. Simply stated, as the density and general capability of programmable logic continues to increase, it is important to consider unconventional hardware solutions to rapid frequency estimation that are progressively being enabled.

Several well studied techniques for frequency estimation include the Discrete Fourier Transform (DFT), classically implemented as the Fast Fourier Transform (FFT), some variation of Least Squares (LS), and more recently, the class of Modern Spectral Analysis (MSA). The FFT presents itself as an attractive solution for a FPGA implementation since it can be highly parallel and has acceptable performance for many ECM applications. The Least Squares method shown is not often used for rapid frequency estimation and is only presented for academic value in this context. As you will see, the performance of a MSA technique makes it quite attractive in the field of spectral estimation, but the computational intensity involved makes for a challenging practical real time solution for ECM. In this thesis, we will show that the Direct State Space (DSS) solution, a type of MSA, has exceptional performance and a difficult hardware implementation, but produces superior results.

1.2 Overview

In the second chapter, we present the relevant background for the frequency estimation problem. This includes a few words on the requirements as well as a description of the performance statistics associated with the frequency estimates, which will later be used to evaluate the performance of several selected frequency estimation techniques. As is common, a discussion will be included regarding the performance given by a derived Cramér-Rao Lower Bound (CRB), a maximum theoretical bound on the variance of an unbiased estimator. A review of the Singular Value Decomposition (SVD), QR Decomposition (QRD), and hardware CORDIC algorithm will conclude the second chapter.

The third chapter includes the estimation theory for each estimator: the DFT, LS, DSS, and the FFT with DSS algorithm. Following the theory, simulations to evaluate the Mean Squared Error (MSE) of the frequency estimates for each technique as a function of SNR will be discussed. The MSE will be compared to the best possible variance as described by the Cramér-Rao Lower Bound.

The fourth chapter contains the implementation details for the selected frequency estimator, including a simplified hardware implementation for the complex 2x2 SVD based DSS algorithm. A comparison of the simulated performance, hardware implementation, and the CRB for the Fast Fourier Transform (FFT) with Direct State Space (DSS) frequency estimation algorithm will follow. The conclusion of chapter four discusses the expected performance of expanded FFT with DSS hardware implementations and a few words of other implementations.

The fifth chapter presents additional theoretical work that should be explored to further develop practical rapid frequency estimation techniques. As will be presented, the intense computational requirements of a large SVD for the DSS technique make a practical implementation difficult, but the suggested hardware architecture adapts to the resources available to a certain degree. The conclusion of chapter five presents the results of the simulated and synthesized hardware implementation for the complex SVD.

Chapter 2

Background

This background chapter includes a few words on the requirements as well as a description of the performance statistics associated with spectral estimates. Also, a short treatment is provided to review two common matrix computations, QR Decomposition (QRD) and Singular Value Decomposition (SVD), along with details of a hardware CORDIC algorithm to be referenced later in this thesis.

2.1 Requirements

The requirements of an ECM system are generally vastly different for each case and system of interest. The rapid frequency estimation problem can be categorized into two groups, each with an emphasis on a particular parameter of interest. The first category would include all wideband systems with relatively low achievable dynamic range. For implementations in this category, the fastest Analog to Digital Converters (ADCs) are chosen, with less of an emphasis on the number of bits, that is, the resolution of the ADC. The second category focuses on a narrowband solution where the accuracy of the frequency estimate outweighs the need for a broad bandwidth, assuming better SNR is achievable. Likewise, implementations in the second category require more resolved bits in the digital conversion, with the impact of slower ADC sample rates. In an attempt to generalize this study, we will present the applicable theory for both cases and then proceed to a wideband analysis and a narrowband implementation.

The model for all frequency estimators of parameters in white Gaussian noise is fairly standard. The objective is to extract the frequency components from a sum of complex sinusoids embedded in white Gaussian noise. Each sinusoid must be able to have an arbitrary amplitude and phase offset. Thus, the sampled model is created by taking M samples of a signal in white Gaussian noise over a sampling interval of MT, where T is the sampling period. In the following equation, with s_k defined as the discrete signal model and $\mathbf{n}_k$ being sampled complex white Gaussian noise, the sampled waveform[1] is expressed as

$$\mathbf{y}_k = s_k(\omega_l, A_l, \phi_l) + \mathbf{n}_k \tag{2.1}$$

where $l = 1, 2, \cdots, P$ indicates the number of sinusoids with associated amplitudes and phase offsets, and $k = 1, 2, \cdots, M$ is the sample index. Therefore, ω_l,A_l,and ϕ_l are parameters which could be estimated from the signal model s_k. For the remainder of this thesis, the noise free transmitted radar waveform from which the frequency will be estimated is defined as

$$s_k = \sum_{l=1}^{P} A_l e^{j(kT\omega_l + \phi_l)} = \sum_{l=1}^{P} c_l e^{j2\pi kT f_l} \tag{2.2}$$

where k is the sample index, T is the sample period, $c_l = A_l e^{j\phi_l}$ is the complex amplitude, and f_l are the frequencies of interest. Initially it appears that this model assumes a Continuous Wave (CW) radar, but in this thesis it is used to model a finite number of samples of a single pulse of a pulsed radar.

Unfortunately, Giordano and Schonhoff [3] state, explicit expressions for many frequency estimation techniques are analytically intractable. Primarily, for the purposes of this paper since the only parameter of interest is the frequency component, f_l, we can simply record the impulse response and extract the poles of the transfer function

$$H(z) = \frac{\sum_{k=0}^{L} b_k z^{-k}}{1 + \sum_{k=1}^{R} a_k z^{-k}}$$

of the modeled system close to the unit circle in the Z-plane. This assumes that the system is excited by an impulse response, exciting all frequencies in the system equally. Therefore, it is clear that the angle of the poles can be used to find the frequencies of interest for our estimation problem.

2.2 Cramér-Rao Lower Bound

Cramér-Rao Lower Bound (CRB) defines the best performance (in the form of variance of the estimated parameter) an unbiased estimator can achieve as a function of other parameters. In our case, the CRB will define the variance of the frequency estimate as a function of Signal to Noise Ratio (SNR), number of samples taken and the sample rate.

[1] Without continuous spectrum consideration

To simplify the derivation, we will first consider a one dimensional single real sinusoid and then make the necessary adjustment for a single complex sinusoid. Thus, we have 3 parameters to estimate from the model defined in (2.1). Using (2.1) for this simple case, we have

$$\mathbf{y}_k \quad = \quad s_k(\alpha) + \mathbf{n}_k \tag{2.3}$$

where $\alpha = (A, \phi, f)$ are the parameters to estimate, $k = 1, 2, \cdots, M$ is the sample index, s_k is the noise free one dimensional signal and $\mathbf{n}_k$ is white Gaussian noise with correlation $E[nn^H] = R_n$. Wright [4] defines $R_n = \gamma C_n$ where γ is the noise power and C_n is the normalized noise correlation. This leads to the noise joint probability density function (PDF) [3] [4]

$$p_n(n) \quad = \quad \frac{1}{det(\pi\gamma C_n)} exp\Big(-\frac{1}{\gamma}\mathbf{n}^H R_n^{-1}\mathbf{n}\Big) \tag{2.4}$$

where H denotes a Hermitian. In order to evaluate the CRB for the frequency estimate, we must evaluate

$$var[\hat{f}] \quad \geq \quad \frac{-1}{E\Big[\frac{\partial^2}{\partial f^2}\, ln\, p(y|\alpha)\Big]} \tag{2.5}$$

where $p(y|\alpha)$ is the conditional probability given the parameters. Continuing with the derivation, we begin with

$$p(y|\alpha) \;=\; p_n(\mathbf{y} - s) \quad = \quad \frac{1}{det(\pi\gamma C_n)} exp\Big(-\frac{1}{\gamma}(\mathbf{y} - s)^H C_n^{-1}(\mathbf{y} - s)\Big) \tag{2.6}$$

and the natural log gives

$$ln\, p(y|\alpha) \quad = \quad ln\left(\frac{1}{det(\pi\gamma C_n)}\right) - \frac{1}{\gamma}(\mathbf{y} - s)^H C_n^{-1}(\mathbf{y} - s) \tag{2.7}$$

which can be further simplified realizing that the first term is free of the parameter of interest and will fall out after the first derivative. Therefore, we substitute K for the first term and expand the second term to get [4]

$$ln\, p(y|\alpha) \quad = \quad K - \frac{1}{\gamma}\big(y^H C_n^{-1} y - y^H C_n^{-1} s - s^H C_n^{-1} y + s^H C_n^{-1} s\big) \tag{2.8}$$

where the terms of interest are the cross terms $-y^H C_n^{-1} s - s^H C_n^{-1} = -2\,\mathbb{R}\big[y^H C_n^{-1} s\big]$[2] and the signal term $s^H C_n^{-1} s$, which will influence the result. The other terms are independent of f and will fall out after the first derivative. Wright [4] has shown in detail that continuing this derivation for all parameters it is possible to show that the elements of the Fisher matrix are defined by

$$F_{ij} \quad = \quad \frac{2}{\gamma}\,\mathbb{R}\Big[\frac{\partial s^H}{\partial \alpha_i} C_n^{-1} \frac{\partial s}{\partial \alpha_j}\Big] \tag{2.9}$$

[2]where $\mathbb{R}[\cdot]$ is the real part

which can be used to fill in a 3x3 Fisher matrix for the most simple single sinusoidal one dimensional case [4]

$$F = \frac{2}{\gamma}\begin{bmatrix} N & 0 & 0 \\ 0 & A^2N & A^2\pi N(N-1) \\ 0 & A^2\pi N(N-1) & \frac{2}{3}A^2\pi^2 N(2N-1)(N-1) \end{bmatrix} \tag{2.10}$$

which can be use to solve for the variances on the various parameters in α by taking F^{-1}. Thus, we conclude that

$$Var[\hat{f}] \;\geq\; \frac{\gamma}{A^2(2\pi)^2}\frac{6}{N(N^2-1)} \tag{2.11}$$

defines the variance of any unbiased estimator for the single sinusoidal frequency estimate $\hat{f}$.

We can also expand on the work done by B. Lovell and R. Williamson [5], who have also derived the best performance of a single real frequency estimator with a bandwidth limitation based on the sample rate. Lovell and Williamson [5] assume a real signal exists with the form $x(n) = a_c cos(2\pi f_o n) + \epsilon(n)$, where ϵ is a zero-mean white Gaussian noise sequence with a variance of σ_ϵ^2, a_c is the amplitude, and f_o is the unknown frequency of interest. The Cramér-Rao Lower Bound defines the theoretical Signal to Noise Ratio (SNR) that is required in order to achieve a desired variance on an unbiased estimator of the frequency f_o. Therefore, the best performance we expect of any unbiased estimator of the frequency f_o is defined by [5]:

$$var[\hat{f}_o] \;\geq\; \frac{f_s^2}{(4\pi)^2}\frac{6}{sN_i(N_i^2-1)} \tag{2.12}$$

where f_s is the sample frequency, $\hat{f}_o$ is the frequency estimate, $N_i = \frac{(M+1)}{2}$ and M is the number of samples in our window, and s is the signal to noise (SNR) ratio, given by $s = \frac{a_c^2}{2\sigma^2}$. This implies that in our derivation, the noise power is $\gamma = \frac{1}{2}f_s^2\sigma^2$. Our collection vector can be defined as

$$\vec{x} \;=\; [x_1, x_2, \cdots, x_M] \tag{2.13}$$

which indicates M samples are taken of the form described in (2.2).

Since our frequency estimator was based on a complex sinusoidal model (2.2), we must modify the Lovell and Williamson CRB to accommodate our model. Given our signal is of the form

$$\mathbf{y}_k \;=\; \sum_{l=1}^{P} A_l e^{j(kT\omega_l+\phi_l)} + \mathbf{n}_k \;=\; \sum_{l=1}^{P} c_l e^{jkT\omega_l} + \mathbf{n}_k \tag{2.14}$$

where, as stated before, $\mathbf{y}_k$ contains complex sinusoids with associated complex amplitudes embedded in complex white Gaussian noise. Using $\mathbf{y}_k$ as defined in (2.14) where $P = 1$, we can form a similar discrete model to fit the Lovell and Williamson CRB

$$\begin{aligned} x_k &= c_l e^{j\omega_l kT} + \mathbf{n}_k \\ &= \left[c_l \cos(\omega_l kT) + \mathbf{n}_1\right] + j\left[c_l \sin(\omega_l kT) + \mathbf{n}_2\right] \end{aligned} \tag{2.15}$$

where $c_l = Ae^{j\phi}$ is the complex amplitude, $\mathbf{n}_1$ is the real component of the noise, $\mathbf{n}_2$ is the imaginary component, the sample period is T, and the sample index is k. Thus, by adding the real and imaginary components, we can get a form that fits into the Lovell and Williamson CRB

$$x_k = 2\,A\,\cos(\omega_l kT + \psi) + n_s \tag{2.16}$$

$$n_s = \mathbf{n}_1 + \mathbf{n}_2 \tag{2.17}$$

where the sum of the real and imaginary noise components has a new variance of $2\sigma^2$ and the amplitude is now $2A$. Therefore, the only change required to allow complex sinusoids using the Lovell and Williamson CRB derivation is to re-define the SNR as $s = \frac{4A^2}{2(2\sigma^2)} = \frac{A^2}{\sigma^2} = \frac{a_c^2}{\sigma^2}$. This conclusion is in agreement with our derivation since for the single complex frequency estimation problem we would define the noise power as $\gamma = f_s^2\sigma^2$, since a 3dB noise improvement is obtained by using quadrature sampling [6].

2.3 QR Decomposition (QRD) and Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) and QR Decomposition (QRD) are important matrix operations that will be used in the theory in Chapter 3 to estimate the poles of a system from a digitized impulse response. A brief introduction to SVD and QRD are provided in this section, but we refer the reader to [7] [8] [9] for additional details and implementations.

The objective of the QRD is to find the set of matrices

$$Q = \prod Q_i \tag{2.18}$$

which, when left multiplied, produce an upper triangular matrix from a matrix $A \in \mathbb{R}^{mxn}$ as

$$A = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} = Q \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix} = Q\,R$$

where the diagonal matrix is defined as R. Several algorithms have been well documented to express $A = QR$, but the most relevant to hardware implementations is using a Jacobi rotation matrix iteratively to zero one element at a time. The real Jacobi rotation matrix is defined as [8]

$$J_i = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & & & & & \\ 0 & & \cos(\theta) & & -\sin(\theta) & & \vdots \\ & & & \ddots & & & \\ \vdots & & \sin(\theta) & & \cos(\theta) & & 0 \\ & & & & & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

where θ is chosen[3] so that

$$\cos(\theta) = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}, \quad \sin(\theta) = \frac{-x_j}{\sqrt{x_i^2 + x_j^2}} \tag{2.19}$$

which will zero out x_j. Left multiplying Q_i repeatedly to A will produce a upper triangular matrix R. Thus, Q is produced using (2.18) where $J_i = Q_i$.

Similar to the QRD, the Singular Value Decomposition (SVD) requires Jacobi rotations to produce a modified matrix. In the case of the SVD, the objective is to find the matrices U, S and V^H such that

$$A = U \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \end{bmatrix} V^H = U\,S\,V^H \tag{2.20}$$

where H indicates the Hermitian, $U^H U = I$ and $V^H V = I$ are unitary, and S is a diagonal matrix of singular values, denoted by σ_n. In this case, we are left and right multiplying Jacobi rotation matrices to produce a diagonal matrix. As you may have guessed based on the QRD discussion above, we are determining the matrices

$$U = \prod_{i=1}^{n} J_i, \quad V^H = \prod_{i=1}^{n} J_i^H \tag{2.21}$$

where n is the number of iterations used to solve A. Other well documented methods exist for these computations, but as we will discuss, Jacobi rotation angles can be computed using the CORDIC algorithm, making this iterative method more attractive for hardware implementation.

[3] where $\sin(\theta)$ is the j,i^{th} element

2.4 The CORDIC Algorithm

The COordinate Rotation Digital Computer (CORDIC) architecture has been well documented in literature as a hardware friendly method to compute several complicated functions such as $\sin(\theta)$, $\cos(\theta)$, $\tan^{-1}(\theta)$, and $\sqrt{x^2+y^2}$ to name a few that are of interest in this thesis [10] [11] [12] [13]. The rapid convergence to the correct solution occurs for the selected complex function using only add, subtract, and shift operations. Also, due to the iterative structure, it is straightforward to implement a parallel CORDIC structure that provides a new function solution every clock cycle after an initial propagation delay of a fixed number of cycles. In this section, we will present a brief overview of the CORDIC algorithm, which will be used to solve the matrices of the complex 2x2 SVD to fit the required structure for hardware implementation in a later section.

The iterative CORDIC converges on the correct vector solution in the complex plane by rotating the estimated vector in smaller and smaller increments towards the correct solution. If we wish to rotate the point $[x_n, y_n]^T$ counterclockwise by and angle α_i, we must left multiply the point by a real Jacobi [10]

$$\begin{bmatrix} \tilde{x}_{i+1} \\ \tilde{y}_{i+1} \end{bmatrix} = \begin{bmatrix} \cos(\alpha_i) & \sin(\alpha_i) \\ -\sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \end{bmatrix} \tag{2.22}$$

which can be written as

$$\begin{bmatrix} \tilde{x}_{i+1}\sec(\alpha_i) \\ \tilde{y}_{i+1}\sec(\alpha_i) \end{bmatrix} = \begin{bmatrix} 1 & \tan(\alpha_i) \\ -\tan(\alpha_i) & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \end{bmatrix} \tag{2.23}$$

after multiplying both sides by $\sec(\alpha_i)$. Equation (2.23) can be used to produce an iterative algorithm to solve for what Kota [10] refers to as circular[4] mode functions. The three basis iterative functions are

$$\begin{aligned} x_{i+1} &= x_i + \delta_i\, y_i\, \tan(\alpha_i) \\ y_{i+1} &= y_i - \delta_i\, x_i\, \tan(\alpha_i) \\ z_{i+1} &= z_i + \delta_i\, \alpha_i \end{aligned} \tag{2.24}$$

where $i = 0, 1, \ldots, n$ is the current iteration step, $\delta_i = -1, 1$ indicates the rotation direction, and z_i accumulates the angle being added. Thus, in this notation, after $n+1$ iterations we have reached an approximation for the function of interest by evaluating the current state of x_n, y_n, and z_n. To simplify the hardware arithmetic, we choose angles such that $\tan(\alpha_i) = 2^{-i}$ and $\tan^{-1}(2^{-i}) = \alpha_i$, so at each

[4] The CORDIC algorithm also has linear and hyperbolic modes that will not be discussed in this paper

iteration step, we simply shift the current value to the right by one. The values $\tan^{-1}(2^{-i})$ that are accumulated in z_i can be precomputed and hard coded into a Look Up Table (LUT) for each iteration. The effect of choosing these angles is clear from (2.24), where we now either add or subtract a shifted version of a value stored in an accumulation register.

Using (2.24) we can implement $\tan^{-1}(\frac{y_0}{x_0})$ by forcing the y_n accumulation register to zero. If we initialize the input of the algorithm with [10]

$$x_0 = \tilde{x}, \quad y_0 = \tilde{y}, \quad z_0 = 0$$

and iterate using

$$\begin{aligned} x_{i+1} &= x_i + \delta_i \, y_i \, 2^{-i} \\ y_{i+1} &= y_i - \delta_i \, x_i \, 2^{-i} \\ z_{i+1} &= z_i + \delta_i \, \tan^{-1}(2^{-i}) \end{aligned} \tag{2.25}$$

where $\tan^{-1}(2^{-i})$ are precomputed in a LUT, and

$$\delta_i = \begin{cases} 1 & y_i \geq 0 \\ -1 & y_i < 0 \end{cases}$$

describes the rotation direction, we can converge to a solution after $n+1$ iterations. In hardware, δ_i can quickly be computed by XORing the MSBs of the current x_i and y_i registers. The accumulation registers will converge to

$$\begin{aligned} x_n &= \frac{\sqrt{x_0^2 + y_0^2}}{K_n} \\ y_n &= 0 \\ z_n &= \tan^{-1}\left(\frac{y_0}{x_0}\right) \end{aligned} \tag{2.26}$$

where K_n is a precomputed scalar divider computed by

$$K_{i+1} = K_i \, \cos(\alpha_i) = K_i \, \cos(\tan^{-1}(2^{-i})) \tag{2.27}$$

where $K_0 = 1$. Thus, from (2.26) it is clear that z_i computes the inverse tangent of the two inputs and x_i computes a scaled magnitude. To convert the solution provided by z_i into an angle computation, it is simple enough to detect the quadrant of the input samples [$\tilde{x}$,$\tilde{y}$] and pipeline the quadrant number to apply the correct offset to the solution provided by z_i using either $\pm\pi$ for the second or third quadrant and zero for the first or fourth quadrant.

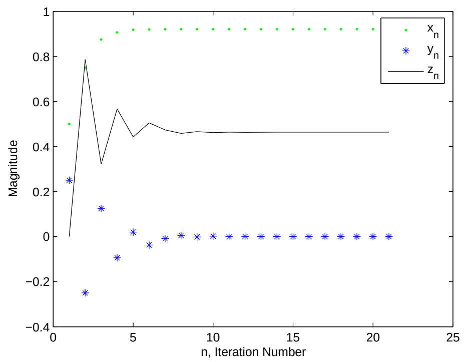

Figure 2.1: CORDIC Inverse Tangent Convergence

Figure 2.1 demonstrates the CORDIC inverse tangent convergence after 21 iterations for the complex number $\frac{1}{2} + j\frac{1}{4}$. The initial conditions are

$$x_0 = \frac{1}{2}, \quad y_0 = \frac{1}{4}, \quad z_0 = 0$$

and after 21 iterations, the registers contain

$$x_{21} = 0.9206, \quad y_{21} = -1.7115e - 006, \quad z_{21} = 0.4636$$

where the precomputed adjustment is $K_{21} \approx 0.6073$. Thus, the final solutions are $\tan^{-1}(\frac{y_0}{x_0}) = 0.4636$ and $\sqrt{x_0^2 + y_0^2} = 0.9206\, K_{21} = 0.5590$. As is expected, the iterations force y_n to zero and x_n and z_n contain the magnitude and phase information.

Similar to the inverse tangent iteration, we can compute a sin() and cos() using the CORDIC algorithm. Instead of iterating y_n to zero, we iterate z_n to zero using [10]

$$\begin{aligned} x_{i+1} &= x_i + \delta_i\, y_i\, 2^{-i} \\ y_{i+1} &= y_i - \delta_i\, x_i\, 2^{-i} \\ z_{i+1} &= z_i - \delta_i\, \tan^{-1}(2^{-i}) \end{aligned} \tag{2.28}$$

with the initial conditions defined as

$$x_0 = x_0, \quad y_0 = y_0, \quad z_0 = \theta$$

where θ is the angle entered into the sin() and cos() functions and δ_i, the rotation direction, is chosen so that

$$\delta_i = \begin{cases} 1 & z_i \geq 0 \\ -1 & z_i < 0 \end{cases}$$

In hardware, it is most simple to compute δ_i by evaluating the MSB of the signed register z_i. As in the inverse tangent iteration, $\tan^{-1}(2^{-i})$ can be precomputed in a small LUT and i defines the current iteration number. Upon the completion of $n+1$ iterations of (2.28), the accumulation registers contain [10]

$$\begin{aligned} x_n &= \frac{x_0\cos(\theta) + y_0\cos(\theta)}{K_n} \\ y_n &= \frac{-x_0\sin(\theta) + y_0\cos(\theta)}{K_n} \\ z_n &= 0 \end{aligned} \tag{2.29}$$

where K_n is computed as in (2.27). To simplify the expression, the initial conditions are often chosen as

$$x_0 = 1, \quad y_0 = 0, \quad z_0 = \theta$$

which allows the accumulation registers to be simplified to

$$\begin{aligned} x_n &= \frac{\cos(\theta)}{K_n} \\ y_n &= \frac{-\sin(\theta)}{K_n} \\ z_n &= 0 \end{aligned} \tag{2.30}$$

completing the algorithm. When an implementation is produced in hardware, care must be taken to determine the quadrant of the input, θ, since this algorithm only correctly computes the solution in the first and fourth quadrants.

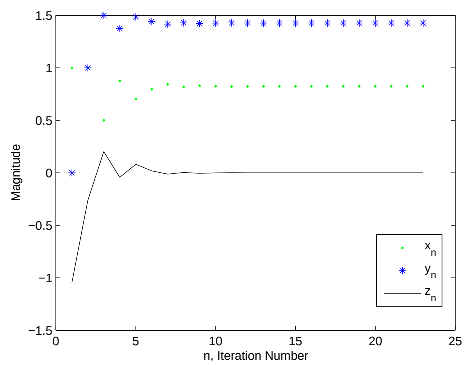

Figure 2.2: CORDIC sin() and cos() Convergence

Figure 2.2 illustrates the convergence of the x_i and y_i as z_i is driven to zero when the initial conditions are

$$x_0 = 1, \quad y_0 = 0, \quad z_0 = -\frac{\pi}{3}$$

and after 22 iterations are

$$x_n = 0.8234, \quad y_n = 1.4261, \quad z_n = 2.1100e - 007$$

and the precomputed scalar adjustment is $K_n \approx 0.6072529$. Therefore, the final solutions are $\cos(-\frac{\pi}{3}) = K_n\, x_n = K_n\, 0.8234 = 0.5000$ and $\sin(-\frac{\pi}{3}) = -K_n\, y_n = -K_n\, 1.4261 = -0.8660$.

Chapter 3

Frequency Estimators

Many frequency estimators have been extensively studied throughout the last several decades. Depending on the requirements, certain estimators have advantages over others. This chapter introduces several estimators of interest and then focuses on those most relevant to the rapid frequency estimation problem.

3.1 Multiple Frequency Estimation

This section describes the theory of three multiple frequency estimators, where $P \geq 1$ in (2.2): Discrete Fourier Transform (DFT), Least Squares (LS), and Direct State Space (DSS). The DSS subsection, which describes the usual method of DSS computation based upon Singular Value Decomposition (SVD), also contains a few words about the implications of substituting a QR Decomposition (QRD) for an SVD. As will be shown, if the QRD is used in place of the SVD there is a loss of accuracy due to the less accurate signal subspace estimation.

3.1.1 Discrete Fourier Transform (DFT)

The Discrete Fourier Transform (DFT), most often implemented as a Fast Fourier Transform (FFT), is a common rapid frequency estimation technique. The FFT examines the spectral content of the signal in the sample window and produces a normalized frequency vs. magnitude and phase spectrum given by sampling the signal on the unit circle in the Z-plane. The Z Transform is given as [14]

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{3.1}$$

for a complex z and discrete time sample x(n). If we sample the complex Z-plane, we can simplify the equation to the Discrete Fourier Transform (DFT)

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \tag{3.2}$$

where N is a fixed number of points conveniently chosen at equidistant points on the unit circle.

The Fast Fourier Transform (FFT) is simply an efficient computation of the DFT. There are several properties of the DFT that allow optimizations to the algorithm that make it attractive for a parallel implementation. A useful property for real valued signals is the symmetry property, where the samples of the DFT in the normalized $[0\ \pi]$ region are related to the samples of the DFT in the normalized $[-\pi\ 0]$ region. Also, since the DFT is periodic with a period of N, the entire spectrum can be represented by repeating the [-π π] region properly.

Proakis and Manolakis derive several computation methods for computing the DFT [14]. For the purposes of this thesis, we will concentrate on a popular one in practice, the Radix-2 FFT algorithm. This algorithm takes advantage of both the periodicity and the symmetry of the DFT by computing just the portion normalized from zero to π. Proakis and Manolakis [14] describe this process. Consider a signal of length $N = 2^v$, x(n), of which we wish to compute the FFT. First, the input is decimated as follows

$$f_1(n) = x(2n) \tag{3.3}$$

$$f_2(n) = x(2n+1), \qquad n = 0, 1, ..., \frac{N}{2} - 1 \tag{3.4}$$

into two sets of samples, the first set taking the even samples, the second taking the odd. Next, if we take the DFT of x(n) in terms of f_1 and f_2 we have

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \tag{3.5}$$

$$= \sum_{m=0}^{(N/2)-1} x(2m)W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1)W_N^{k(2m+1)} \tag{3.6}$$

$$= \sum_{m=0}^{(N/2)-1} f_1(m)W_{N/2}^{mk} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m)W_{N/2}^{km} \tag{3.7}$$

$$= F_1(k) + W_N^k F_2(k) \qquad k = 0, 1, ..., N-1 \tag{3.8}$$

where $W_N = e^{-j2\pi/N}$ is the twiddle factor. Since $F_1(k)$ and $F_2(k)$ are periodic, with a period of $\frac{N}{2}$ we can say $F_1(k + \frac{N}{2}) = F_1(k)$ and $F_2(k + \frac{N}{2}) = F_2(k)$. Using this property, and realizing that

$W_N^{k+N/2} = -W_N^k$, we can express the rest of X(k)

$$X(k) = F_1(k) + W_N^k F_2(k) \qquad k = 0, 1, ..., \frac{N}{2} - 1 \tag{3.9}$$

$$X\left(k + \frac{N}{2}\right) = F_1(k) - W_N^k F_2(k) \qquad k = 0, 1, ..., \frac{N}{2} - 1 \tag{3.10}$$

thus completing the DFT computation using the computation of two sequences of length $\frac{N}{2}$. In general, Proakis and Manolakis [14] give the number of complex multiplications as $\frac{N}{2}log_2N$ for this technique vs N^2 for the direct computation approach. Clearly, as the number of points increases, the computational improvement factor increases dramatically.

The Radix-4 FFT algorithm is similar to the Radix-2 algorithm except the number of points must be $N = 4^v$ and the input data segment is separated into four decimated sets. Due to the parallelizable nature of the resultant hardware architecture and by taking advantage of DFT properties, the Radix-N algorithms are attractive in hardware implementations.

For our frequency estimation application, once the FFT is computed using a finite number of points, the magnitude samples of the DFT that exceed a predetermined magnitude will be considered as the normalized poles of the transfer function.

3.1.2 Least Squares

The Least Squares method is a method that minimizes a sum of squared errors between a given signal and the signal produced by a parameterized transfer function under impulsive excitation. Using the Least Squares method it is possible to estimate the transfer function of interest, which can be used to estimate the transmitted frequencies. As with any standard system analysis problem, it is assumed from a modeling perspective that the transfer function of the victim radar is excited by an impulse, generating an impulse response that includes the victim radar and transfer medium. Our ECM system would then record the impulse response of the unknown system and determine the frequency of the transmitted waveform by examining the poles of the transfer function created using a LS method. Since we are only interested in the poles of the transfer function we will use an all pole model:

$$H(z) = \frac{b_0}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{3.11}$$

and solve for the minimum squared errors with respect to the parameter a_k.

J.G. Proakis and D.G. Manolakis [14] detail a solution starting on page 706. Suppose the unknown system is cascaded with a reciprocal all zero system, named $H_d(z)$ and $H_{LS}(z)$. If the cascaded system is excited by an impulse, the ideal output of the chain would also be an impulse. In reality, the output, y(n), of the system is

$$y(n) = \frac{1}{b_0}\left[h_d(n) + \sum_{k=1}^{N} a_k h_d(n-k)\right] \tag{3.12}$$

where $h_d(n)$ is the impulse response of the unknown system. The required condition y(0) = 1 is satisfied if $b_0 = h_d(0)$ and the remaining terms of y(n), n > 0 are required to be zero. Using the remaining terms to minimize the sum of the squared error between the actual output of the cascaded transfer functions and an impulse gives

$$\epsilon = \sum_{n=1}^{\infty} y^2(n) \tag{3.13}$$

$$= \frac{\sum_{n=1}^{\infty} i\left[h_d(n) + \sum_{k=1}^{N} a_k h_d(n-k)\right]^2}{h_d^2(0)} \tag{3.14}$$

where $y(n)$ is the output sequence of our system, $h_d(n)$ is the recorded impulse response of the radar system, and a_k are the unknown parameters of the new system we wish to minimize. To minimize the equation we set the derivative with respect to a_k equal to zero and solve, giving us the set of linear equations [14]:

$$\sum_{k=1}^{N} a_k r_{hh}(k,l) = -r_{hh}(l,0) \qquad l = 1, 2, ..., N \tag{3.15}$$

where r_{hh} is the correlation sequence defined as [14]:

$$r_{hh}(k,l) = \sum_{n=1}^{\infty} h_d(n-k)h_d(n-l) \tag{3.16}$$

Using the set of linear equations defined by (3.15), we could solve for N unknown values for a_k using N equations and N unknowns. Our problem is to find the dominant frequencies that are being transmitted by the victim radar, so we can instead solve for the gain factor and the P poles that most likely fit the set of equations. Thus, we only have P+1 unknowns and N equations assuming a single noise pole

$$\sum_{k=1}^{P+1} a_k r_{hh}(k,l) = -r_{hh}(l,0) \qquad l = 1, 2, ..., N \tag{3.17}$$

where N is the number of samples in our sample window. With this technique, the number of frequencies to solve for cannot be easily estimated from the derivation at this point. It is assumed that there is *a* *priori* knowledge of the number of signals or that the number of signals is estimated using a different technique, perhaps one similar to that described in Section 3.1.1. Before we can solve for a_k, we must define a few matrices:

$$R_l = \begin{bmatrix} r_{hh}(1,1) & r_{hh}(2,1) & \cdots & r_{hh}(P+1,1) \\ r_{hh}(1,2) & r_{hh}(2,2) & \cdots & r_{hh}(P+1,2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{hh}(1,N) & r_{hh}(2,N) & \cdots & r_{hh}(P+1,N) \end{bmatrix} \tag{3.18}$$

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{P+1} \end{bmatrix} \tag{3.19}$$

$$R_r = \begin{bmatrix} -r_{hh}(1,0) \\ -r_{hh}(2,0) \\ \vdots \\ -r_{hh}(N,0) \end{bmatrix} \tag{3.20}$$

where r_{hh} is the correlation sequence defined in (3.16). If $P+1 \neq N$, then R_l is not a square matrix and we must solve for a_k using the pseudo inverse:

$$A = R_l^{-1} R_r \tag{3.21}$$

Generalizing a single noise pole into M noise poles, using the coefficients in matrix A, and realizing that they are the coefficients of $H_{LS}(z)$, we can build the transfer function of the unknown system

$$H_d(z) = \frac{b_0}{1 - \sum_{k=1}^{P+M} a_k^{-k} z^{-k}} \tag{3.22}$$

where P is the number of signal poles modeled and M is the number of noise poles. Using partial fraction expansion, we always get the unknown gain factor along with the poles that best represents the transmitted frequency

$$H_d(z) = G + \sum_{k=1}^{P} \frac{r_k}{1 + p_k z^{-1}} + \sum_{n=1}^{M} \frac{r_n}{1 + p_n z^{-1}} \tag{3.23}$$

where G is the gain factor, r_k are the residue of the poles, and p_k are the poles of interest. The noise poles, p_n, are disregarded. In practice, the smallest residues and a pole near the unit circle generally imply that the corresponding pole is a signal pole rather than a noise pole. Other techniques for separating the noise and signal partial fraction expansion terms will not be discussed.

As stated earlier, by examining the angle of the pole in the Z-plane, we can produce an estimate for the transmitted frequency. Thus, we can convert this to an actual frequency based on our sample rate

$$f_k = \frac{arg[p_k] f_s}{2\pi} \tag{3.24}$$

where f_k is the transmitted frequency, f_s is the ADC sample rate, and arg[p] is the angle of the pole. Numerical instability arises in this LS algorithm as seen in the next section.

3.1.3 Direct State Space

Modern spectral analysis techniques have provided powerful signal analysis tools that have been heavily studied in recent years. Direct State Space realizations for spectral analysis have emerged as a powerful method for impulse response pole extraction. As before in the Least Squares section, we assume that our system, $H_{unknown}(z)$, is excited by an impulse. The response to the impulse is collected by our ECM system and requires analysis for the most dominant poles. We can model this linear time invariant (LTI) system by assuming the system produces an output y(t) based on an input u(t). State space modeling suggests that this system can be modeled as follows [15]:

$$\dot{x} = Ax + Bu \tag{3.25}$$

$$y = Cx + Du \tag{3.26}$$

where $\dot{x}$ describes the state evolution, A describes state transition, B is a matrix that influences the next state based on the input (state controlling), C influences the output based on the current state (state observing), and D is a feed forward matrix that effects the output directly. For the purpose of this analysis, we can model the signals of interest in a state space model that will allow useful factorization and ultimately parameter extraction. State space theory tells us that a representation of the transfer function using the matrices A,B,C, and D is

$$H_{unknown}(z) = C(zI - A)^{-1}B \tag{3.27}$$

where the feed forward term, D, is assumed to be zero. If we assume that the unknown system transfer function has been broken down by partial fraction expansion, it is clear that the poles of the system fall on the diagonal of matrix A and some form of the residues fall in matrix C [15].

To begin our discussion, we will assume the case where the input, u(t), is an impulse. An impulse in time implies that the controlling matrix excites all states of the system with equal power. Next, since our ECM system collects the impulse response of the system, we have knowledge of the output of the system, y(t). Thus, our problem is to extract the poles of matrix A given the input and output of the system. Mathematically, the matrices are given as [4]:

$$u(t) = \begin{cases} 1 & t = 0 \\ 0 & t > 0 \end{cases} \tag{3.28}$$

$$A = diag(z_1, z_2, ..., z_P) \tag{3.29}$$

$$B = [1, 1, ..., 1]^T \tag{3.30}$$

$$C = [c_1, c_2, ..., c_P] \tag{3.31}$$

$$D = 0 \tag{3.32}$$

where $z_i = e^{j2\pi kTf_i}$ and c_i are as defined in (2.2). Clearly, matrix A contains the parameters of interest, matrix B excites all the states equally when t=0, matrix C contains the complex amplitudes, and matrix D is not included in the model. The primary objective of this analysis is to extract the features of the A matrix in order to identify f_i, given some recorded data from the output of the system, y_k.

To understand the evaluation and feature extraction of this state space model better we will introduce the discrete state space model as follows [15]:

$$x_{k+1} = Ax_k + Bu_k \tag{3.33}$$

$$y_k = Cx_k + Du_k \tag{3.34}$$

where A, B, C, and D matrices can be described the same as in (3.25). Using this discrete model, we can simulate time moving forward from t = 0, the initial state,

$$x_1 = Ax_0 + Bu_0 = B = [1, 1, ..., 1]^T \tag{3.35}$$

$$x_2 = Ax_1 + Bu_1 = Ax_1 = AB \tag{3.36}$$

$$x_3 = Ax_2 = A^2B \tag{3.37}$$

$$x_k = Ax_{k-1} = A^{k-1}B \tag{3.38}$$

and similarly, the output can be described,

$$y_1 = Cx_1 = CB \tag{3.39}$$

$$y_2 = Cx_2 = CAB \tag{3.40}$$

$$y_3 = Cx_3 = CA^2B \tag{3.41}$$

$$y_k = Cx_k = CA^{k-1}B \tag{3.42}$$

allowing the generalization,

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} CB \\ CAB \\ \vdots \\ CA^{k-1}B \end{bmatrix} \tag{3.43}$$

This generalization can aid in the factorization for A. First, generate a KxL Hankel matrix [4] [16].

$$X = \begin{bmatrix} y_1 & y_2 & \cdots & y_{L-1} \\ y_2 & y_3 & \cdots & y_L \\ \vdots & \vdots & \ddots & \vdots \\ y_{K-1} & y_K & \cdots & y_{M-1} \end{bmatrix} = \begin{bmatrix} CB & CAB & \cdots & CA^{L-1}B \\ CAB & CA^2B & \cdots & CA^LB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{K-1}B & CA^KB & \cdots & CA^{M-1}B \end{bmatrix} \tag{3.44}$$

It is noteworthy to state that the finite size of the Hankel matrix, KxL, has an impact on the performance of this frequency estimation method. The size of the Hankel matrix has long been a topic of discussion in a variety of papers, including [4] [16]. Hua suggests [17] that the optimum choices for L would either be $\frac{M}{3}$ or $\frac{2M}{3}$. Wright [4] provides a thorough explanation of the optimizations. To proceed with our derivation, X can be factored into the observability and controllability matrices, O and C respectively

$$X = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{K-1} \end{bmatrix} \begin{bmatrix} B & AB & A^2B & \cdots & A^{L-1}B \end{bmatrix} \triangleq OC \tag{3.45}$$

Using the observability and controlability matrices derived from the state space model, we can solve for A using the equation $O_-A = O_+$, where O_- is O with the last row deleted and O_+ is O with the first row deleted. This gives the state transition matrix solution as

$$A = O_-^{-1}O_+ \tag{3.46}$$

where O_{-}^{-1} is the pseudoinverse of O_{-}. This formulation generally provides a solution to the state transition matrix that is not diagonal as required by equation (3.29). Thus, a similarity transformation to the state space model can be used

$$\begin{aligned} A &= T^{-1}AT = diag(z_1, z_2, \dots, z_P) \\ B &= T^{-1}B \\ C &= CT = [c_1, c_2, \dots, c_P] \end{aligned}$$

to diagonalize matrix A and solve the complex amplitudes in C. Clearly, A now contains the eigenvalues of the observability based solution from (3.46) and T contains the corresponding eigenvectors. Rather than using a similarity transformation, a Schur decomposition can be used to find the eigenvalues of A, from which a Least Squares problem can be implemented using the model to find better amplitude estimates [18].

This method of pole extraction requires that the matrix X be noise free [4] [19], where in practice we must estimate A from a noisy X. To address this issue, we introduce the procedure for DSS pole extraction. First, generate a Hankel matrix with K finite rows and L finite columns as in (3.44). Next, compute the Singular Value Decomposition (SVD) of the noise contaminated Hankel matrix. The largest singular values define the dominant signal subspace, called the principle components [20]. If it is known that there exist P signals in the subspace, it is expected that the noise free estimate of matrix X, $\hat{X}$, will have a rank[1] of P. Thus, we can simply perform a rank truncation to produce $\hat{X}$ by evaluating the largest singular values of the noisy matrix X.

The procedure can be shown as follows: start with a noisy matrix X and it's SVD

$$X = \begin{bmatrix} U_s & U_n \end{bmatrix} \begin{bmatrix} \Sigma_s & 0 \\ 0 & \Sigma_n \end{bmatrix} \begin{bmatrix} V_s^H \\ V_n^H \end{bmatrix} \tag{3.47}$$

where $U_s \Sigma_s V_s^H$ represents the true signal subspace and $U_n \Sigma_n V_n^H$ represents the noise subspace. We define our estimate of the signal subspace as

$$\hat{X} = \hat{U}_s \hat{\Sigma}_s \hat{V}_s^H \tag{3.48}$$

by rank truncation, estimating the rank of the signal subspace by the examining the most dominant singular values. The SVD also performed a factorization required by (3.46) that does not affect the performance of the DSS algorithm [4]. To build the observability matrix, O, in the factorization we

[1] Unless the poles are too close in proximity, which will not be considered in this paper

can multiply a factored form of the rank truncated singular values by the resultant U matrix as

$$\hat{O} \;=\; \hat{U}_s \hat{\Sigma}_s^{1/2} \tag{3.49}$$

where $\hat{U}_s$ and $\hat{\Sigma}_s$ are the rank truncated estimates of the signal space and $\hat{\Sigma}_s^{1/2}$ is defined as

$$\hat{\Sigma}_s^{1/2} \;=\; diag\big(\sigma_{s1}^{1/2}, \sigma_{s2}^{1/2}, \cdots, \sigma_{s\hat{P}}^{1/2}\big) \tag{3.50}$$

where $\hat{P}$ is the estimated number of signals in the signal subspace. After building O_{s+} and O_{s-} by eliminating the first row or last row respectively, the solution for A follows simply as

$$\hat{A} \;=\; \hat{O}_{s-}^{-1} \hat{O}_{s+} \tag{3.51}$$

where $\hat{O}_{s-}^{-1}$ is the pseudo-inverse of $\hat{O}_{s-}$. Once the estimated A has been computed, the eigenvalues, or the poles of the system, contain the desired f_i of interest. As before, once the poles are extracted, we can solve (3.24) for the transmitted frequencies.

Several optimizations are allowed by (3.49) which reduce the computational complexity necessary to resolve the angle of the poles. Wright [4] indicates that if additional parameters are not needed from B or C, the finite word length effects of the factorization are unimportant. Since this is the case for our estimation problem, we can express the estimate of A as

$$\hat{O} \;=\; \hat{U}_s \tag{3.52}$$

$$\hat{A} \;=\; \hat{O}_{s-}^{-1} \hat{O}_{s+} \tag{3.53}$$

and disregard the added weight from the singular values to each column since we are only interested in the angle of the poles, which are computed from the U matrix of the Hankel factorization.

A simple technique can be used to improve performance known as Forward-Backward Averaging [21]. Start by organizing the samples in the initial rank revealing SVD as

$$X \;=\; \begin{bmatrix} H & H_f \end{bmatrix} \tag{3.54}$$

where H is the original Hankel matrix and H_f is the Hankel flipped up-down and conjugate transposed. In this case, the rest of the algorithm remains the same since the row deletes for O_{s+} and O_{s-} are still the first and last rows. If the modified Hankel was stacked vertically, the first and last respective rows would have to be deleted of each Hankel in X.

An alternative method exists when one wishes to avoid the hardware complexity of an SVD. An approximation of the signal subspace can be obtained by finding the collection of matrices that generate an ordered upper right triangular matrix from the Hankel matrix by

$$X = \begin{bmatrix} U_s & U_n \end{bmatrix} \begin{bmatrix} \Sigma_s & \Upsilon \\ 0 & \Sigma_n \end{bmatrix} \tag{3.55}$$

where the elements of Υ are influenced by both the signal and noise subspaces. This operation is known as QR Decomposition. By examination, an estimate of the unitary matrix U_s can then be used to form the estimate of $\hat{A}$ as in (3.53) since this is also an acceptable factorization for the observability and controllability matrices in equation (3.45). Also, since the components of the signal subspace still exist in the elements of Υ, this method does not perform as well as an SVD. The next section describes this performance in detail.

3.2 Frequency Estimator Performance

In this section, the performance of each frequency estimator is compared to the best theoretical performance given by the Cramér-Rao Lower Bound for a single sinusoid in white Gaussian noise with 20dB SNR. Each simple estimator is given the same parameters. The sample rate is 1500 Mega Samples Per Second (MSPS) and the length of the recorded data is 500ns. This gives us a total of 500ns * 1.5GHz = 750 samples. The sample window is then divided into three sections of 250 samples each. For the first segment, the transmitted frequency is 80 MHz, followed by 350 MHz, and then -220 MHz. Figure 3.1 shows the frequency time intensity plot for the transmitted waveform.

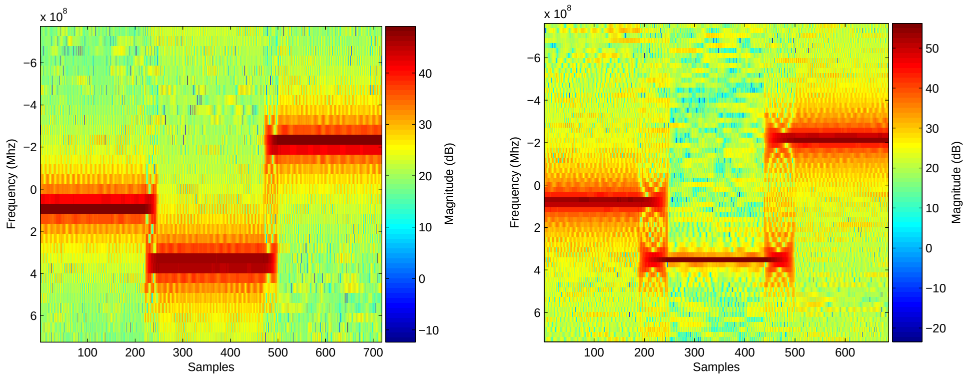

Figure 3.1: Frequency Time Intensity plot with (left) 32 point FFT (right) 64 point FFT windows

The image in the left of Figure 3.1 was obtained using a 32 point FFT window that was slid sample by sample over the 750 sample dataset and the image in the right was obtained by using a 64 point sliding FFT window. As expected, with more points in the FFT, the noise floor is integrated over more bins, so it appears lower. Also, there are clearly transitional phenomena around the frequency changing points since there are influences from both frequencies. The amount of overlap is determined by the size of the sliding FFT window, which is clear by the length of the smear in Figure 3.1.

3.2.1 Discrete Fourier Transform (DFT)

As mentioned in the introduction of this chapter, the DFT computation was performed using a 32 point sliding window FFT over the entire 750 data samples. The frequency estimation algorithm requires the frequency bin with the largest magnitude to be identified as the transmitted frequency at each window of time. Figure 3.2 shows the extracted frequency estimate plotted in white over the frequency time intensity sliding window FFT. As expected, the estimated frequency always takes on a value that is in the center of a frequency bin.

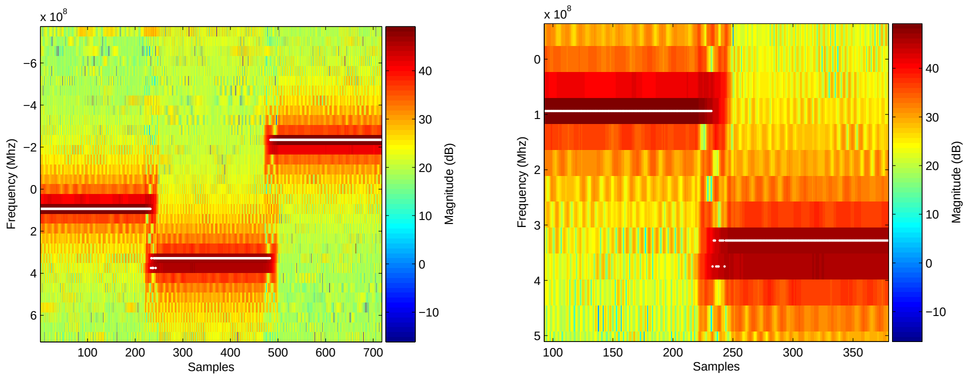

Figure 3.2: FFT extracted frequencies (left) all samples (right) zoomed on transition

The statistics for the extracted FFT frequencies were computed using estimates 20 samples beyond the instantaneous frequency change. This allowed sufficient time for the sliding window FFT algorithm to primarily detect the influence of the current transmitted frequency. Table 3.1 summarizes the frequency estimation statistics for the FFT algorithm.

f_{tx} (MHz)	Mean (MHz)	σ (MHz)	Error (MHz)
80	93.75	0	13.75
350	328.125	0	-21.875
-220	-234.375	0	-14.375

Table 3.1: FFT algorithm statistics at 20 dB SNR

With a bin size of $b_s = \frac{1.5Ghz}{32} = 46.875$MHz, we expect that our error would never exceed $\pm\frac{b_s}{2} = \pm 23.4375$MHz. If the input frequency was on the edge of a bin, it will fall into the closest bin. The peak magnitude of the closest bin would estimate the frequency with a maximum absolute error of $\frac{b_s}{2}$. As Table 3.1 shows, the error is indeed bounded by our expected maximum. Also, as Figure 3.2 shows, the estimated frequency remains constant throughout the duration of the statistical calculation, corresponding to the standard deviation being zero throughout the entries in Table 3.1. This behavior is an artifact of the coarseness introduced by simply reporting the center of the FFT bin as the estimated frequency using just 32 samples.

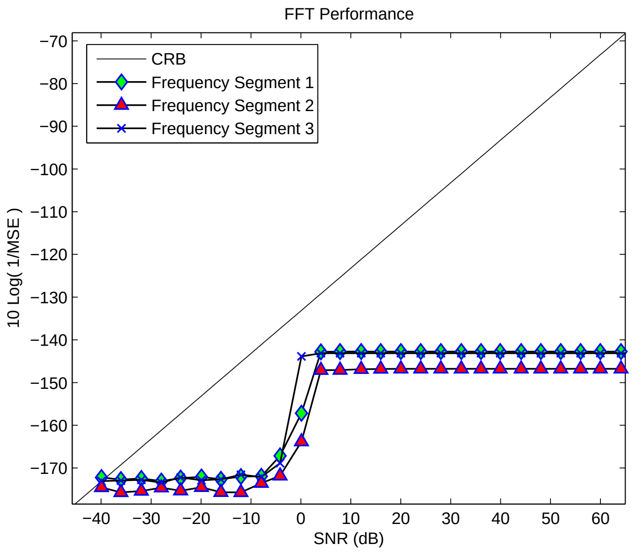

Figure 3.3: FFT performance vs CRB

Figure 3.3 shows the FFT performance of each extracted frequency against the theoretical boundary set by the Cramér-Rao Lower Bound. We can see that the frequency estimation in poor SNR, roughly below 0 dB, produces an inaccurate estimate. Once the SNR is increased, a peak magnitude can be extracted from the sliding window FFT. The performance of the FFT algorithm doesn't even get close to the CRB since there is a discrete bin size and the true frequencies are sufficiently displaced from the center of the frequency bins evaluated by the FFT. Thus, the (Mean Squared Error) MSE becomes bounded as the SNR increases. What must be done to effect a better estimate is to decouple the number of data samples in the window under consideration from the number of frequency samples at which the Z-Transform is being evaluated, thus reducing the coarseness of the estimator.

A common approach for frequency estimation when such a small number of samples is taken is to

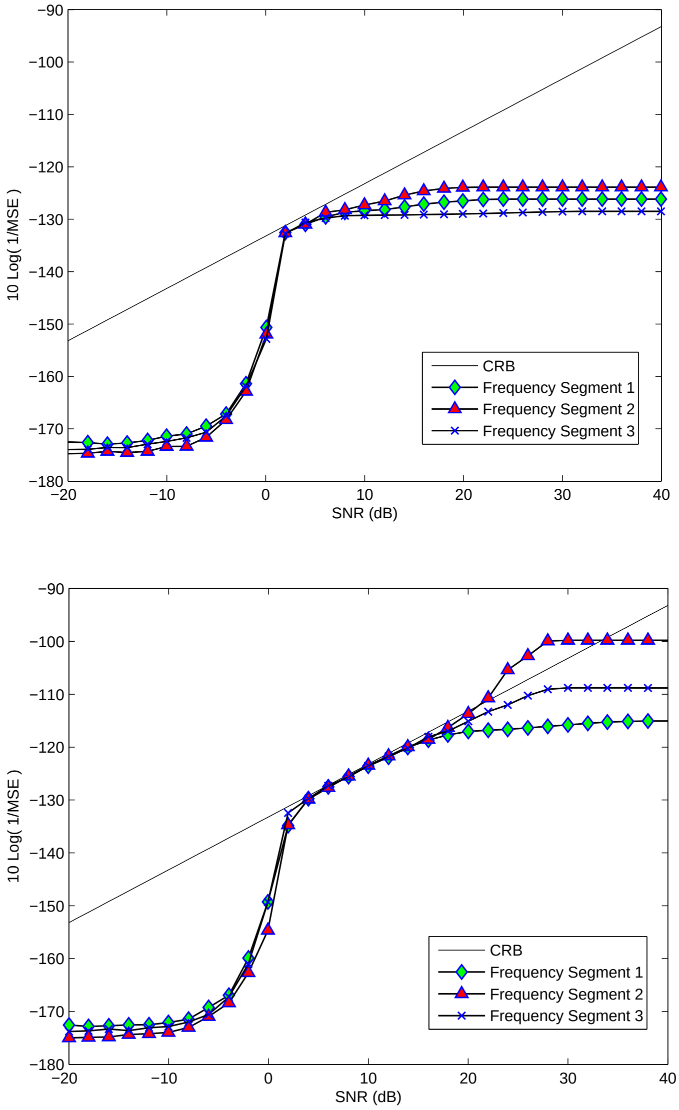

Figure 3.4: Zero padded FFT using (top) 256 points (bottom) 1024 points

zero pad the time domain data and take a larger FFT. The finer granularity in the frequency samples allows a more accurate frequency estimate. Figure 3.4 shows the zero padded FFT performance using a 256 point FFT and a 1024 point FFT using just 32 data samples as before. It is clear that there can be a significant performance improvement by using a larger FFT. In the 256 point case, the MSE becomes bounded much sooner as SNR increases than in the 1024 point case. Interestingly, one frequency in the 1024 point case performs better than the CRB around 25 dB SNR. This is likely due to the discrete nature of the larger FFT where the frequency under analysis happens to fall near the center of a bin.

The performance of this technique can be misleading since the largest error can still be roughly 732 kHz using a 1024 point FFT and the CRB is calculated using $M = 32$ in equation (2.12). If a longer integration period was considered, where the number of samples collected was 1024 or larger, a 32x increase in the number of points in the FFT becomes more difficult to practically implement in an FPGA and more time consuming than perhaps other techniques.

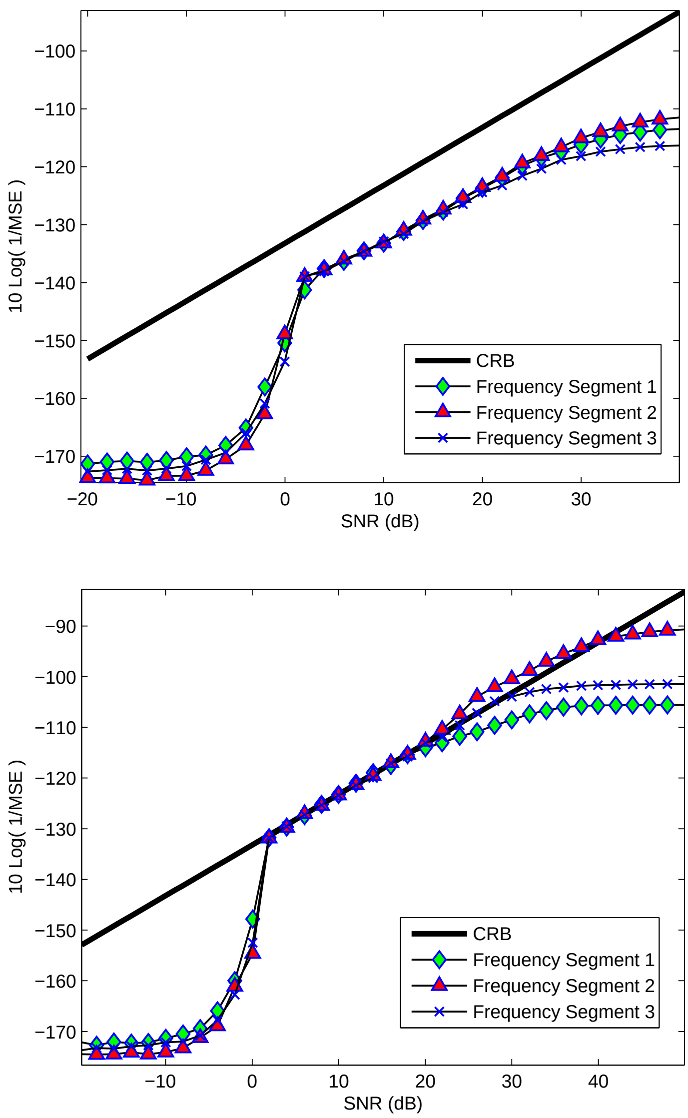

Figure 3.5: Zero padded FFT with center of mass refinement using (top) 256 points (bottom) 1024 points

In addition to performing the zero padding technique, an attempt to refine the frequency estimate by performing a center of mass computation on the adjacent frequency bins was implemented. This seems particularly useful if the frequency happens to fall near the edge of the bin. Figure 3.5 shows the performance of a zero padded FFT with either a 256 or 1024 point FFT followed by the center of mass refinement, using just 32 time domain samples and ±25 frequency bins from the peak frequency bin for the center of mass computation. The MSE appears to suffer a performance loss around 0 dB SNR, particularly in the 256 point FFT case when compared to Figure 3.4. The performance, however, remains unbounded out to a much higher SNR value as would be expected. Not much difference is noticed in the 1024 point FFT case other than the smoothing of the MSE near the previous bounded values in higher SNR.

3.2.2 Least Squares

To evaluate the performance of the Least Squares (LS) algorithm, the same test parameters were employed. Using a 32 point sliding LS window over the 750 samples, frequency estimates were extracted using (3.24). Figure 3.6 shows the estimated frequency points in white over the transmitted frequency time intensity plot in roughly 20 db SNR.

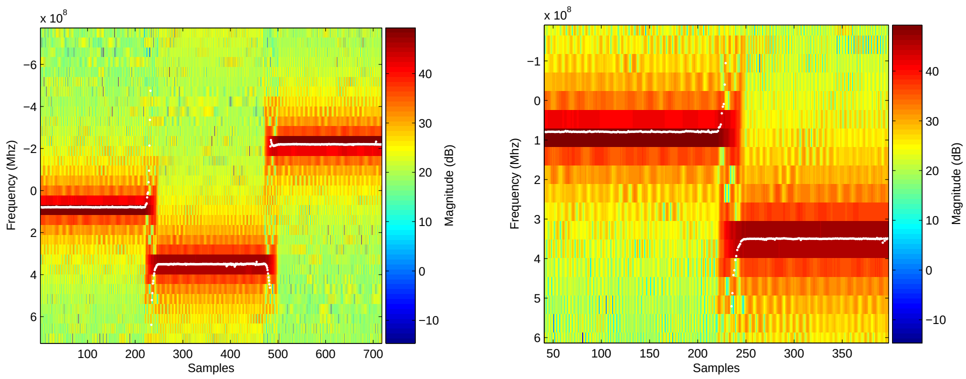

Figure 3.6: LS extracted frequencies (left) all samples (right) zoomed on transition

This method appears to work better than the FFT algorithm using the same number of points since the LS algorithm does not have discrete bin sizes. Looking closely at Figure 3.6, it is clear that the extracted frequency follows quite closely with the actual transmitted frequency rather than being forced into a coarse discrete bin. Table 3.2 summarizes the frequency estimation statistics for the LS algorithm.

As the statistics show, these estimated frequencies are better than the estimated frequencies of the FFT algorithm using the same number of points. When, however, implementing the zero padding or

f_{tx} (MHz)	Mean (MHz)	σ (MHz)	Error (MHz)
80	75.91	2.149	-4.09
350	349.63	1.525	-0.372
-220	-219.91	1.126	-0.0851

Table 3.2: LS algorithm statistics

zero padding with center of mass refinement technique, the FFT algorithm sees to perform better. Two of the three transmitted frequencies have an error of less than ± 1 MHz, indicating reasonable performance over such a large bandwidth. The standard deviation is also small, indicating that most estimates fall relatively close to the mean. The large error on the first frequency estimate was due to a few incorrect pole extractions far away from the real frequency. The majority of the extracted points for the first frequency follow the same statistics as the second and third frequencies.

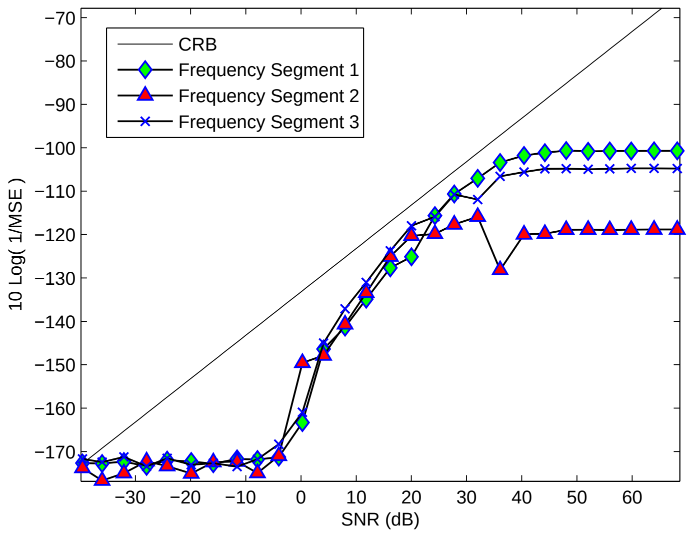

Figure 3.7: Least Squares performance vs CRB

Figure 3.7 shows the LS algorithm performance of each extracted frequency against the CRB. This shows an improvement over the coarse FFT CRB shown in Figure 3.3, but not quite as well as the zero padding techniques seen in Figure 3.4 and Figure 3.5. As with the FFT techniques, the MSE for this implementation of a Least Squares eventually becomes bounded. The MSE of the estimate cannot improve further even as the SNR increases due to numerical inaccuracies of the estimation algorithm.

3.2.3 Direct State Space

The Direct State Space (DSS) algorithm performance was measured in the same way as the FFT and LS algorithms were measured. Using a 32 point sliding DSS window over the 750 samples, frequency estimates were extracted using the angle of the extracted poles of the estimated transfer function.

Using Singular Value Decomposition

Figure 3.8 shows the estimated frequency points in white over the transmitted frequency time intensity plot in roughly 20 db SNR.

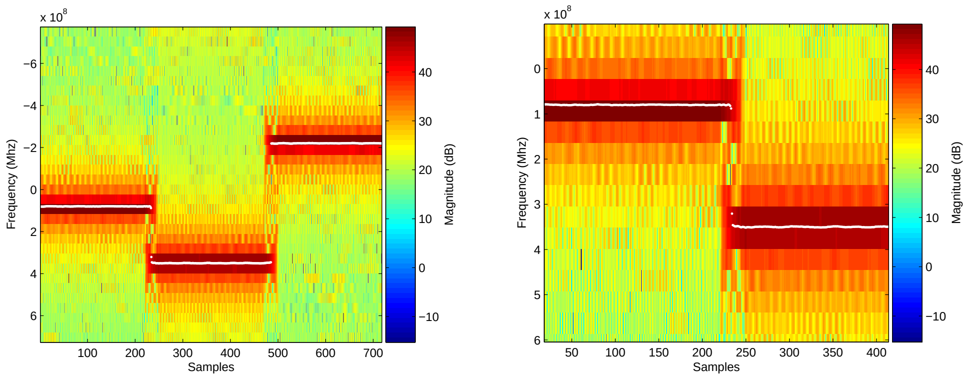

Figure 3.8: DSS extracted frequencies (left) all samples (right) zoomed on transition

It appears that this method is superior to the last two methods based on the rapid transition between frequency estimates and the unwavering estimates during constant transmission. Figure 3.8 clearly shows the extracted frequency following the transmitted frequency quite well, even through the transitions. By analyzing the statistical performance of the DSS algorithm, we will see that it is indeed equivalent or superior to the FFT or LS algorithms in most positive SNR. Table 3.3 shows the frequency estimation statistics for the DSS algorithm.

f_{tx} (MHz)	Mean (MHz)	σ (MHz)	Error (MHz)
80	80.05	0.602	0.047
350	349.98	0.508	-0.022
-220	-219.99	0.525	0.088

Table 3.3: DSS algorithm statistics using SVD

As Table 3.3 indicates, the frequency estimates by the DSS algorithm are much better than the extracted frequencies of the FFT and LS algorithms. Each extracted frequency has errors that are measured in the 10's of kHz rather than in MHz. Also, the confidence factor, indicated by the standard deviation, shows excellent performance with less than a MHz deviation for all cases. The extracted mean shows quite an improvement from both previous algorithms. An analysis of the DSS algorithm plotted against the CRB shows the superior performance of this algorithm.

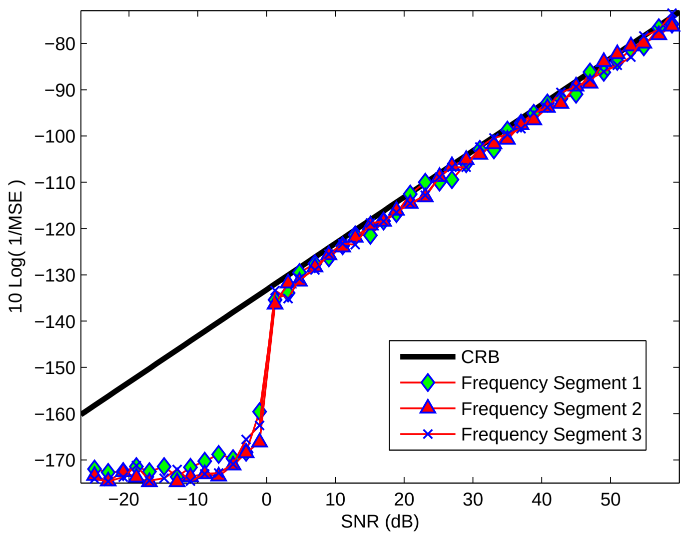

Figure 3.9: DSS Performance vs CRB using SVD

Figure 3.9 shows the DSS algorithm performance of each extracted frequency against the CRB. As soon as sufficient signal energy is detected by the algorithm, it immediately jumps just about onto the CR bound. Even as the SNR increases, the MSE of the estimate becomes more and more accurate, never indicating a limit has been reached like the FFT and LS algorithms.

Using QR Decomposition

As mentioned in Section 3.1.3, an ordered QR Decomposition (QRD) can be used to estimate the signal subspace. As in the SVD case, we can use the signal subspace estimate to directly solve for the angle of the poles of the system, which are the estimated frequencies of interest. Figure 3.10 shows the extracted frequencies using a QRD of a square Hankel matrix filled with 32 samples to estimate the signal subspace.

The left of Figure 3.10 shows the extracted frequencies in white of the entire 500ns pulse over the equivalent FFT generated frequency time intensity. If we zoom on the first transition as seen in the

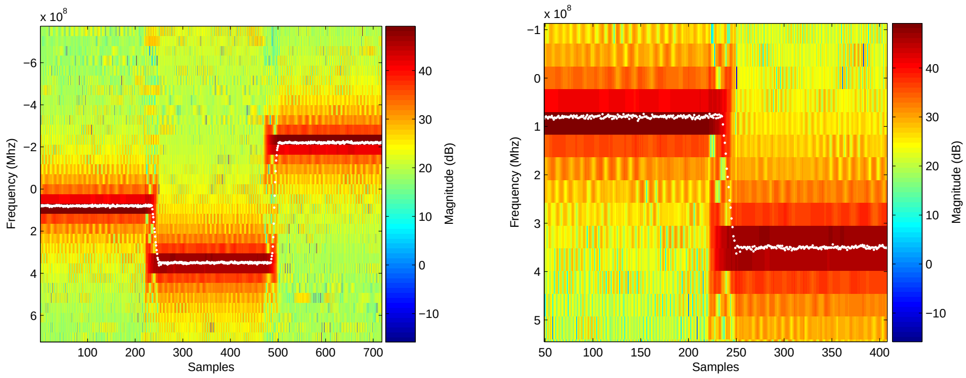

Figure 3.10: DSS extracted frequencies using QRD (left) all samples (right) zoomed on transition

right, we can observe the minor differences between the QRD estimation and the SVD estimation of Figure 3.8. Clearly, there is a larger MSE on the QRD generated estimates and the transition point contains several inaccurate transition estimates. Looking at the statistical elements in Table 3.4 we can see that indeed there is a larger MSE associated with the QRD estimation method.

f_{tx} (MHz)	Mean (MHz)	σ (MHz)	Error (MHz)
80	80.09	2.2917	0.0859
350	349.82	2.3668	-0.1819
-220	-219.62	4.2817	0.3827

Table 3.4: DSS algorithm statistics using QRD

As Table 3.4 indicates, the extracted frequencies by the DSS algorithm are still much better than the extracted frequencies of the FFT algorithm. The extracted mean shows quite a bit of degradation over the SVD method, but still an improvement over the FFT method. This method can be compared to the Least Squares statistical performance as the standard deviation and average error are similar. In fact, several more robust Least Squares algorithms have evolved from QR Decomposition based solutions of the normal equations [22].

Figure 3.11 illustrates the QRD signal subspace estimation method to extract the poles against the CRB in various SNR using a square Hankel matrix filled with 32 samples. In this case, as the signal begins to emerge from the noise around 0dB SNR, the MSE of the estimate appears to slowly rise to some linear offset of the CRB. In this simulation, the offset is around 14 dB.

Comparing the low SNR cases of both the SVD and QRD methods, we can see in Figure 3.12 that the SVD algorithm outperforms the QRD method. On the right, the SVD method quickly jumps to

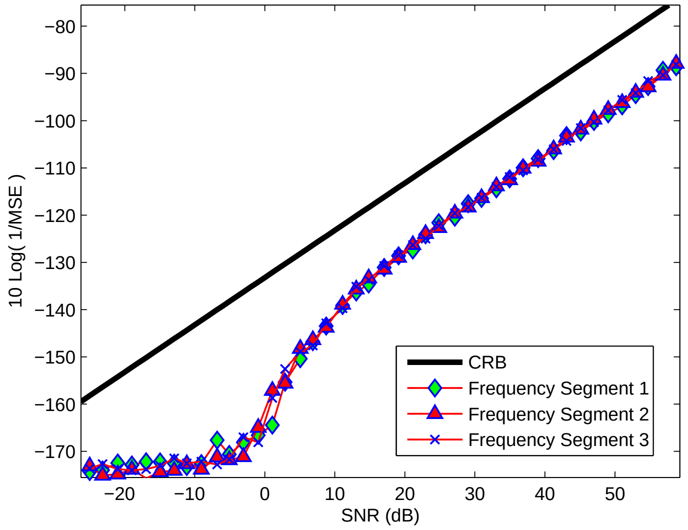

Figure 3.11: DSS Performance vs CRB using QRD

just next to the CRB at 2 dB SNR. On the left, the QRD method slowly converges to the 14 dB linear offset from the CRB as SNR increases. Clearly, if practical, the SVD method is more desirable in an implementation.

If we have a priori knowledge of the number of sinusoids in the band of interest, we can modify the Hankel matrix to be non-square. As in the LS analysis, where we assume a single frequency pole and a single noise pole, we can reduce the size of Υ from (3.55) to reduce the possibility of noise corrupting the signal subspace estimate. Figure 3.13 shows the QRD based DSS algorithm when we solve for a single signal pole using a 1x32 column vector as the Hankel matrix. This performance is similar to the LS performance of Figure 3.7 where we estimated a single noise pole and a single signal pole using autocorrelation, except this implementation uses a QRD that is numerically stable allowing the MSE of the estimate to decrease as SNR increases. Clearly in SNR larger than about 35 dB, the QRD method of Least Squares is superior to the Least Squares method presented in Section 3.2.2.

3.3 Combined Approach for Multiple Frequency Estimation

As seen in the previous sections, it is clear that the DSS performance is far superior to other frequency estimators introduced. The computational burden of estimating the signal subspace using a singular value decomposition (SVD) is on the order of $O(n^3)$ operations on a single processor. The

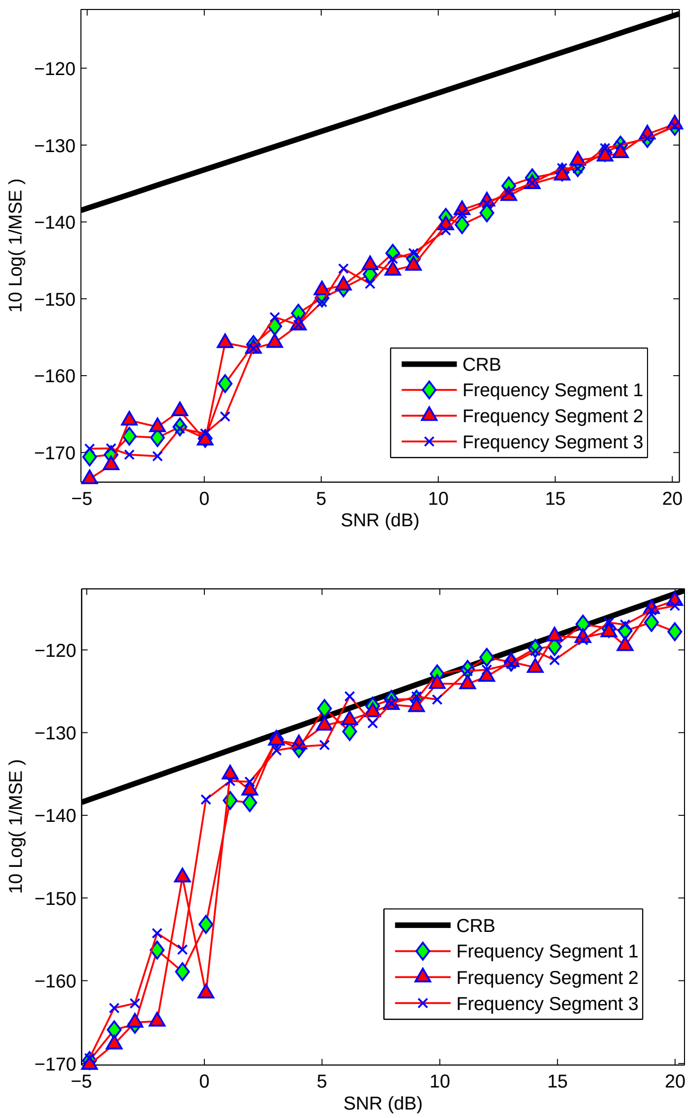

Figure 3.12: DSS Performance Zooming in on low SNR using (top) QRD (bottom) SVD

SVD can be computed using a two step method, first bidiagonalizing the nxn matrix, followed by a fast computation method for the U and S based on the bidiagonalized form. The Intel Math Kernal Library (MLK) [23] for a single processor states that the number of floating point operations for this method is $\frac{32}{3}n^3 + n^2 + 12n^3 = \frac{68}{3}n^3 + n^2$. This is the majority of the computation time of the DSS algorithm, but still does not include the eigenvalue decomposition and pseudoinverse of a smaller matrix when solving for the poles of the system. As the number of samples increases, the DSS algorithm becomes impractical for real time applications. The FFT, however, has only $\frac{n}{2}\log_2 n$ complex multiplications plus several

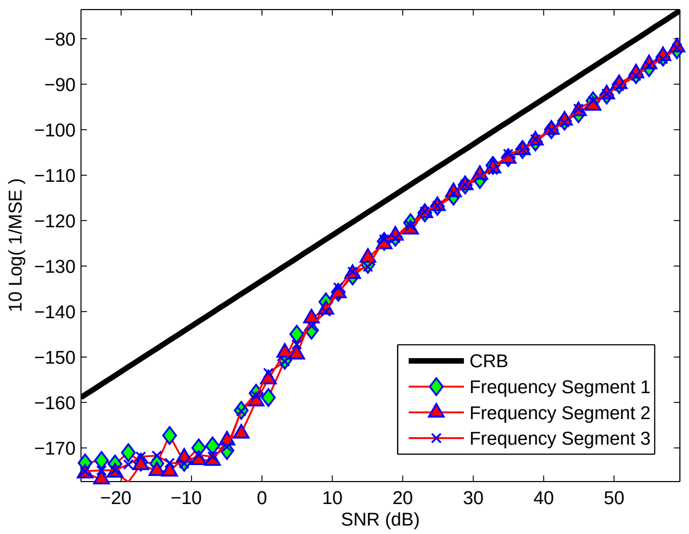

Figure 3.13: DSS Performance vs CRB using a non-square Hankel matrix

addition computations and can be adapted to a parallel computing platform easily. Therefore, the FFT is commonly used in real time applications, suffering the performance loss seen in the previous section. This section proposes a FFT and DSS combined approach designed to allow longer integration periods for the DSS algorithm in real time applications. The proposed implementation will gain many of the estimation advantages of the DSS algorithm while providing a practical hardware solution.

3.3.1 Combined Approach Theory

Longer integration periods are crucial in many applications where thousands to hundreds of thousands of samples are collected but cannot be processed rapidly using DSS due to the large size complex SVDs required to estimate the signal subspace. A practical implementation is desired to reduce the overall hardware complexity and design costs. This implementation, named FFT with DSS (FWD), will be shown to have similar or better statistical performance than the DSS algorithm in practical SNRs while significantly decreasing the computational burden. With few limitations when applied to the ECM problem, this method is the most attractive of all the multiple frequency estimators considered.

Combined Approach using complex DSS

The FWD technique simply applies the FFT to a finite set of samples to produce a coarse estimate of the frequency embedded in noise. Assuming constant (or at least near constant) frequency sinusoid or sinusoids in the set of samples, it is assumed that the magnitude of the bins with the signals will exceed the magnitude of the bins with the noise at sufficient SNR. As described in Section 3.1.1, the magnitude of the bins that exceed a predetermined threshold will be considered detected frequencies. For each bin exceeding the threshold, a tuned, filtered, and decimated subset will be passed to the DSS algorithm for analysis. Effectively, the FFT estimation will be used as *a priori* information for the DSS estimation method.

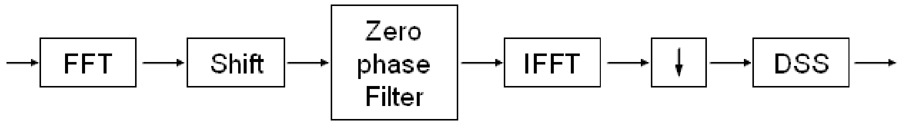

Figure 3.14: FFT with DSS Block Diagram

Figure 3.14 shows the FWD block diagram. Immediately following the FFT, a shift is performed in the frequency domain to shift the bin of interest to the center of the frequency domain at DC. A zero phase low pass filter is then applied in order to preserve phase and to filter unwanted components outside the region of interest. The preservation of the phase is required by the DSS algorithm, which includes the phase relation of the samples when making an estimate. Filtering the unwanted components achieves anti-alias filtering for the future decimation as well as improving the SNR in the region of interest. To achieve the desired zero phase low pass filter in the frequency domain, a Hanning window is used around the region of interest. We refer the reader to page 626 of Proakis and Manolakis [14] for more information on several common window functions, including the Hanning window. The number of points in the Hanning window must be precisely

$$w_h = \left\lceil \frac{\lceil \frac{1.6}{D} M - 1 \rceil}{2} \right\rceil \tag{3.56}$$

where M is the number of points in the FFT[2] and D is the down sample factor to be used after the IFFT. The same implementation may be reproduced in the time domain as long as care is taken to implement a zero phase filter with similar characteristics. Upon completion of the tuning and filtering operation, the signal of interest is then decimated. Decimation is a method of resampling the same data set to decrease the number of samples and the bandwidth. To decimate by an integer factor D, the discrete signal x(m) would produce a subset by y(m) = x(mD). For more information, see page 784 of [14]. If the frequency of interest is exactly in the center of the detection bin, then the frequency component of the decimated sample set would be at DC. The DSS estimation algorithm is used to

[2] also assumed to be the number of samples collected

determine the deviation from DC, which provides an accurate estimate of the error in the frequency estimate generated by the FFT algorithm. As shown in Section 3.1.3, the complex SVD will be used to estimate the signal subspace of the smaller number of samples. It will become clear in the next section that the statistical performance of this technique can be better than the DSS technique alone due to the zero phase filtering on the sinusoid frequency of interest. See the concluding subsection of this Section for a simulated demonstration.

Combined Approach using real DSS

To further simplify the hardware complexity of the FFT with DSS technique discussed in Section 3.3, we could replace the complex SVD in the DSS algorithm with a real SVD. In this modification, the FWD technique first applies the FFT to a finite set of samples to produce a coarse estimate. For each bin that contains a magnitude that exceeds a predetermined threshold, a tuned, filtered, decimated, and tuned real subset will be passed to the DSS algorithm for analysis.

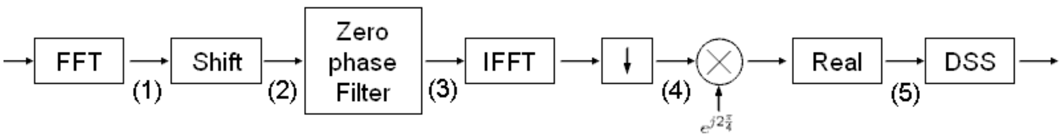

Figure 3.15: Real FFT with DSS Block Diagram

Figure 3.15 shows the real FWD block diagram. Immediately following the FFT, a shift is performed in the frequency domain to shift the bin of interest to the center at DC. A zero phase low pass filter is then applied by using a Hanning window. The number of points in the window remain the same as in equation (3.56). Upon completion of the tuning and filtering operation, the signal of interest is then decimated. Assuming a single sinusoid is present in the decimated sample set, multiplication by a complex exponential is used to tune or separate the tones as far as possible to produce the most accurate estimation results with the real DSS algorithm. Prior to the final stage of execution, the real and imaginary components are added to produce a real set of sinusoids at the decimated sample rate divided by four for furthest separation. If more than one frequency was present in the decimated sample set, a different sinusoid separation algorithm could be developed. See the next subsection for a simulated demonstration.

Since the DSS algorithm is real, there will exist complex conjugate pairs of poles that are used to solve for the frequency content of the system. Thus, the real DSS algorithm may require a larger SVD to estimate the signal subspace. Also, care must be taken when solving for the poles of the system from the estimated signal subspace, which will require larger eigenvalue decompositions of some type since there will always be more than one signal pole. Simulations have suggested that for a single complex conjugate pole, it is sufficient to extract the single column of the U that corresponds to the largest singular value for further processing.

Combined Approach Simulation

Figures 3.16, 3.17, and 3.18 show the frequency domain of a synthetic data set passing through the FFT with DSS algorithm. Matching the numbers in parentheses that indicate the location in the block diagram from Figure 3.15 to the following Figure titles, a two tone sinusoid is used to demonstrate the FFT with DSS algorithm. The SNR for this demonstration is approximately 20 dB using 512 input samples, which are processed and decimated down to 16 samples for the DSS algorithm. The spectrum of the first two stages of the algorithm can be seen in Figure 3.16. The left of Figure 3.16 shows the input frequency spectrum, which is used to identify the FFT bin with the largest magnitude. Using the peak magnitude bin location, the frequency spectrum is shifted to DC as seen on the right of Figure 3.16.

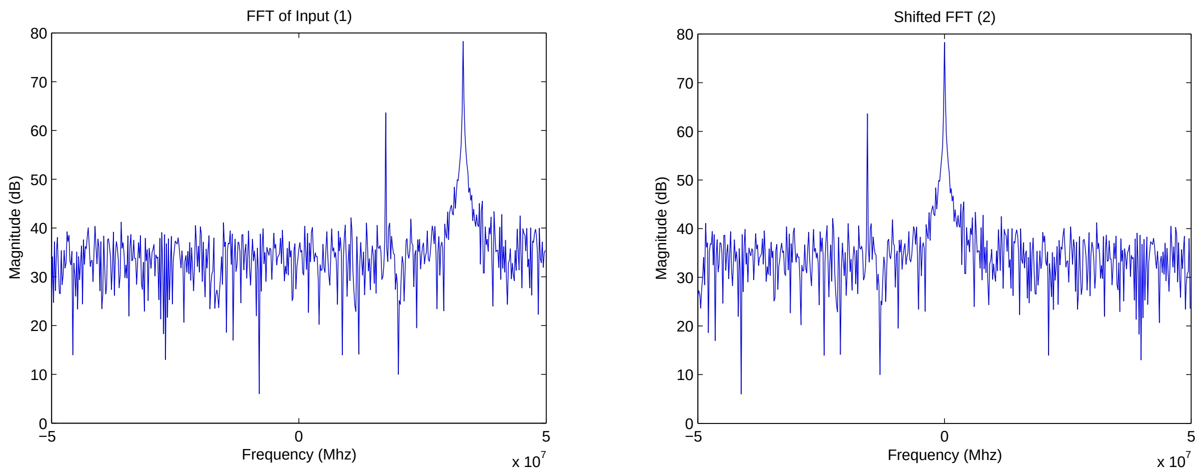

Figure 3.16: (left) FFT of test input signal (right) FFT tuned to DC

Upon tuning the most dominant frequency to DC, the spectrum is filtered to isolate the frequency region of interest using a predetermined Hanning zero phase filter as described above. The frequency spectrum after the zero phase filter can be seen in the left of Figure 3.17. The filtered samples are then converted back into the time domain using an Inverse FFT (IFFT) and decimated to produce the

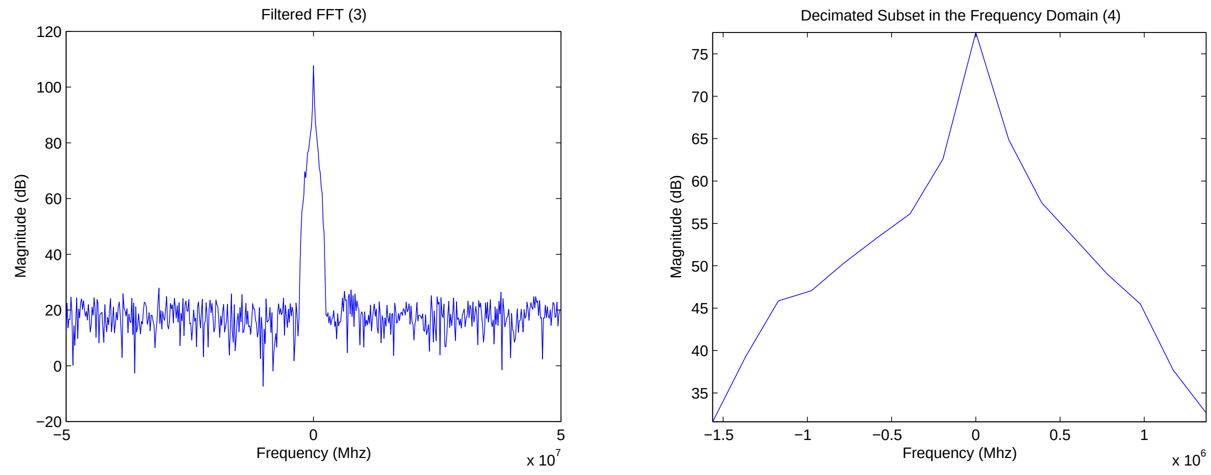

Figure 3.17: (left) Tuned and Filtered input in frequency domain (right) Decimated data subset in frequency domain

spectrum seen in the right of Figure 3.17. No anti-alias filtering is required prior to the decimation since the zero phase low pass filter was used to isolate the frequency location of interest. At this point, it is clear that the deviation from DC in the decimated sample set can be used to determine the error in the FFT estimate. Since the DSS algorithm is reasonable in real time applications using a small number of samples, it is used to estimate the error in the FFT estimate.

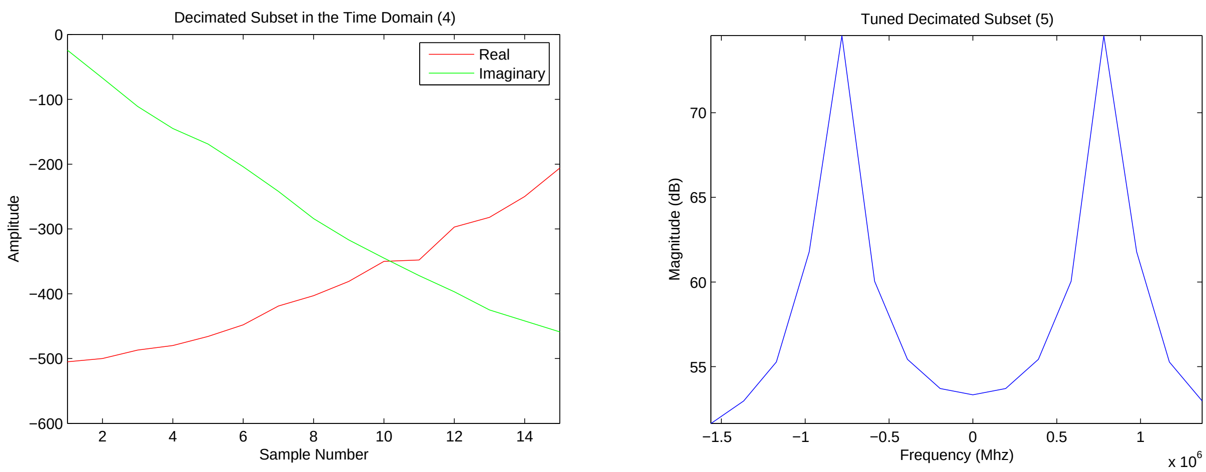

Figure 3.18: (left) Decimated data subset in the time domain (right) Tone separated real decimated data subset

The left of Figure 3.18 shows the decimated sample set in the time domain, indicating a clear error (deviation from DC) exists in the estimate of the FFT. If a complex DSS algorithm is not available to estimate this error, the tones have to be separated as far as possible for optimal real DSS performance. The right of Figure 3.18 shows the separation required to prepare the the samples for a real DSS analysis. The frequency estimate would then be computed using the initial shifted amount, adjusted by the high resolution error estimate provided by the DSS algorithm.

3.3.2 Combined Approach Results

Instead of using 32 samples as in the previous simulations, the FFT with DSS (FWD) algorithm performance was tested using a 1024 point sliding window over 10240 samples at 1.5Ghz. The same frequency segments were chosen at intervals of $\frac{1}{3}$ of the total simulation size. First, a 1024 point FFT is taken to detect the frequencies of interest. After tuning to DC, filtering, and decimating, the real 16x16 SVD based DSS algorithm is applied to observe the deviation from zero of the tuned signal. This error estimate is applied as a correction factor to the initial shifted amount to estimate the transmitted frequency.

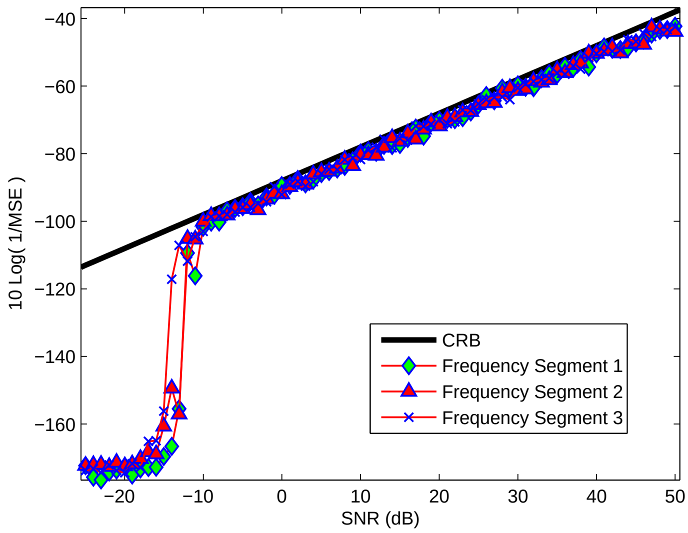

Figure 3.19: FWD using 1024 samples decimated to fit a real 16x16 Hankel matrix

Figure 3.19 shows the FWD performance using a 1024 point sliding window over the 10240 data points at various SNRs. It is initially surprising that the performance is near efficient in a large portion of the negative SNR range. To quantify the maximum improvement gain possible in the FWD algorithm over the DSS algorithm, we solve for the improvement factor using a ratio of the number of FFT samples to the number of DSS samples in the decimated subset. The improvement factor in dB can be expressed as

$$G_{dB} \quad = \quad 10 \log_{10} \left(\frac{P_{FFT}}{P_{DSS}} \right) \tag{3.57}$$

where P_{FFT} is the number of points in the FFT of the FWD and $P_{DSS} = 32$ is fixed at the number of samples in the DSS performance computations from the previous section.

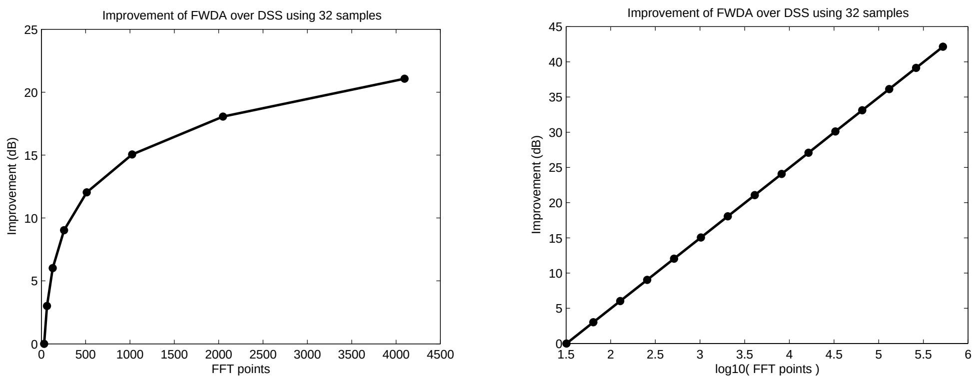

Figure 3.20: FWD performance improvement over DSS using 32 samples on (left) a linear scale and (right) on a log scale

Figure 3.20 illustrates the improvement in dB as a function of the number of points in the FFT both on a linear and log scale. Clearly, there is a linear improvement as the number of samples collected for the FFT detection increases. If sufficient data points are collected, it is theoretically possible to exceed a 40 dB improvement by collecting 512k samples and decimating to 32 prior to the DSS error estimation. For the case in Figure 3.19 where 1024 samples are collected, and from (3.57), we can state $G_{dB} = 10 \log_{10} \frac{1024}{32} \approx 15dB$. Looking back at Figures 3.19 and 3.9, there appears to be up to about 13-14 dB improvement in the threshold SNR at which the performance jumps to the CRB level.

It is also interesting to consider a simplified case wherein the input data stream is real rather than complex. For the real input, as described in Section 3.3.1, a real FFT is computed and the rest of the algorithm is identical. Figure 3.21 shows the performance of the FWD using a real and complex input with a 32 sample DSS algorithm compared to the same CRB based on the SNR being calculated as $\frac{A^2}{\sigma^2}$ (not $\frac{A^2}{2\sigma^2}$ for the real plot). The consistent difference of about 3-4 dB from the real case to the complex case is due to the fact that the complex sampled input has twice as much bandwidth as the real sampled input. Also, additional errors may be introduced for the real case since the complex DSS algorithm does not require tuning to separate the frequency components.

We will now test the claim associated with equation (3.54) that better performance in low SNR can be achieved using a Forward Backward Averaging based upon a horizontal Hankel stack. Figure

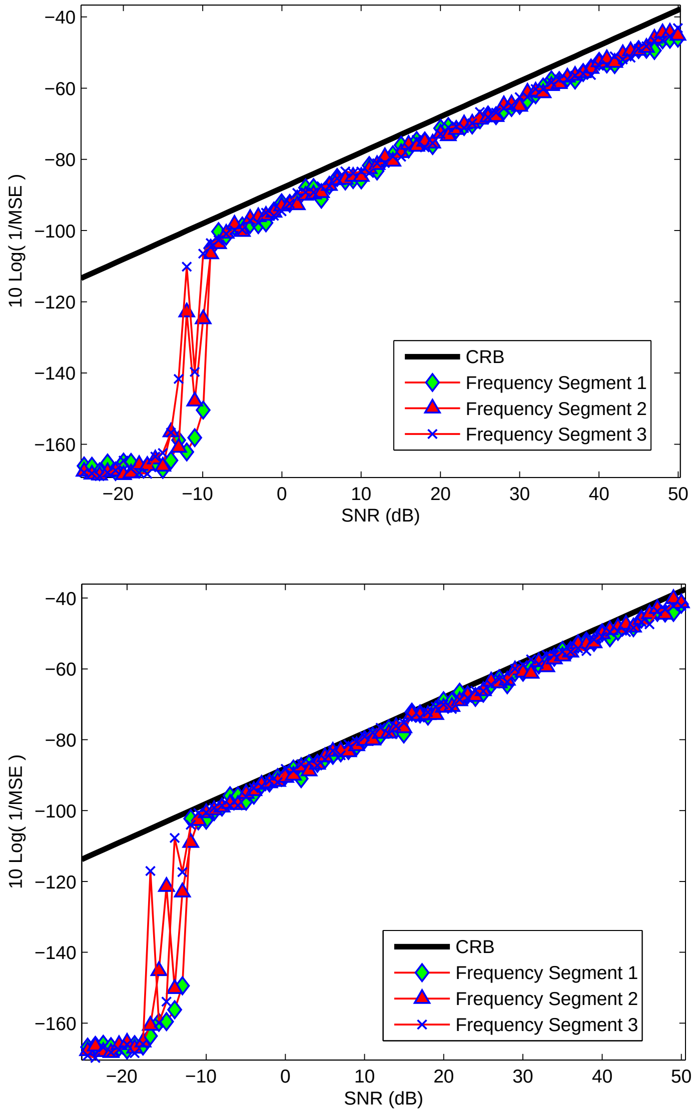

Figure 3.21: FWD using (top) real input and (bottom) complex input compared to complex CRB

3.22 shows the FWD algorithm using the same 1024 point FFT/IFFT pair but modifying the DSS algorithm to use either a complex 8x8 Hankel or a complex 8x16 horizontal Hankel stack. As can be seen, there is a small difference in the two simulation results for most values of SNR. One frequency segment shows a 2-3 dB gain, but other two segments show little change. In a practical system, the computational burden of including the horizontal Hankel stack vs. the expected 3 dB gain would have to be considered on a case by case basis. In our rapid hardware application, this small performance improvement does not warrant using this method.

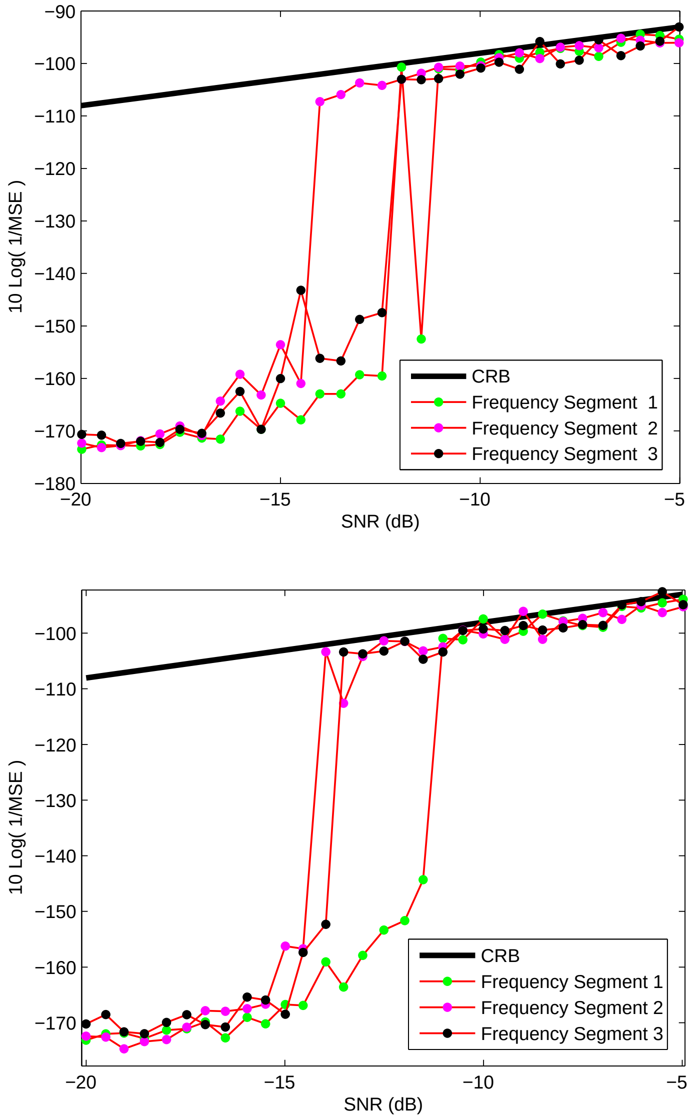

Figure 3.22: FWD using (top) single 8x8 complex Hankel (bottom) horizontal Hankel stack for SNR improvement

Chapter 4

Hardware Implementation

Based on the performance analysis and complexity of several frequency estimators in the previous chapter, it is evident that estimating the frequency of a sinusoid from a single radar pulse is quite difficult, especially in real time. As the Cramér-Rao Bound implies in Section 2.2, with a small number of samples collected, the variance of the frequency estimate will be larger than if many samples are collected. Due to the short duration of a single radar pulse, there is a limited number of samples that can be collected for frequency estimation. In many applications where thousands of samples are collected, a large point FFT is generally an excellent solution due to the parallel nature of the algorithm and the rapid solutions it is able to provide. When a small number of samples are collected, as is the case for frequency estimation from a single radar pulse, a more statistically efficient algorithm is the Direct State Space (DSS) algorithm using a Singular Value Decomposition (SVD). The estimate of the signal subspace is used to solve directly for a frequency estimate. Since the complexity of SVD processing of matrices in the DSS algorithm becomes impractical for a large number of samples, a practical system will need to place restrictions on the number of samples that can be collected. Although QR Decomposition is an easier and more rapid algorithm, such as in [24], we choose to pursue the most efficient algorithm presented using an FFT and a complex SVD. This chapter provides the details of a narrowband implementation of the FFT with DSS (FWD) algorithm using a novel method to solve a 2x2 complex SVD suitable for the DSS algorithm. To follow the work of Hemkumar [11] and others to solve the complex SVD using a hardware CORDIC algorithm, refer to Section 5.1.

4.1 Combined Approach Implementation

The objective of the Combined Approach implementation is to demonstrate the algorithm of Section 3.3.2 in a simple and practical design. Once the hardware is complete, statistics are gathered and compared to the theoretical performance and expected results based on MATLAB simulations. The implementation assumes 1024 samples are collected and processed to produce a modified 3 sample subset that will fill a complex 2x2 matrix for the DSS algorithm.

The hardware selected for this implementation is the Altera Stratix II DSP evaluation kit containing two 12-bit 125 Mega Samples Per Second (MSPS) ADCs, two 14-bit 165 MSPS DACs, 3 8-bit video DACs to drive an RGB monitor, 32 MB of SDRAM, 1MB of SRAM, a high density FPGA (EP2S60F1020C4ES) and other unused extra peripherals.

The most complex and time consuming portion of the FFT with DSS algorithm is the complex SVD. Many authors in open literature discuss Jacobi Rotation based implementations that require angular computations, generally done by the CORDIC algorithm in hardware. We begin our implementation by deriving the Jacobi Rotation based algorithm shown by Hemkumar [11] and then discuss novel 2x2 SVD optimizations for the DSS algorithm. The implementation block diagrams and the hardware performance results compared to MATLAB simulations will conclude this chapter.

4.1.1 Complex 2x2 SVD using Jacobi Rotations

Following a method similar to that of Hemkumar [11], we can describe the two step transformation required to compute a complex 2x2 SVD. The objective is to use complex Jacobi rotations to zero out desired elements in the matrix. After sufficient rotations have been applied, the solution matrix S becomes a real diagonal matrix containing the singular values. The left and right rotation matrices are combined to produce the U and V^H unitary matrices. In the two step method we are searching for a solution to

$$M = USV^H \tag{4.1}$$

where U and V^H are replaced by

$$M = U_2^H U_1^H S V_1^H V_2^H \tag{4.2}$$

so that U_1^H and V_1^H are computed using a complex QR Decomposition (QRD) to produce an upper right 2x2 matrix and then U_2^H and V_2^H are computed such that the solution matrix is a real diagonal matrix.

The complex Jacobi unitary matrix used to perform both transformations is of the form [11]

$$\begin{bmatrix} c_\phi e^{j\theta_\alpha} & -s_\phi e^{j\theta_\beta} \\ s_\phi e^{j\theta_\gamma} & c_\phi e^{j\theta_\delta} \end{bmatrix}$$

where the rotation angles θ_α, θ_β, θ_γ, θ_δ, and θ_ϕ are real numbers with $\theta_\alpha - \theta_\beta - \theta_\gamma + \theta_\delta = k$ where $k \in (0, 2\pi, \cdots, 2n\pi)$. The notation

$$c_\phi = \cos(\theta_\phi), \quad s_\phi = \sin(\theta_\phi)$$

is used to simplify the expressions.

The first transformation can be computed using a complex Jacobi rotation on the left and a transpose of the complex Jacobi on the right. We wish to solve for

$$\begin{bmatrix} c_\phi e^{j\theta_\alpha} & -s_\phi e^{j\theta_\beta} \\ s_\phi e^{j\theta_\alpha} & c_\phi e^{j\theta_\beta} \end{bmatrix} \begin{bmatrix} Ae^{j\theta_a} & Be^{j\theta_b} \\ Ce^{j\theta_c} & De^{j\theta_d} \end{bmatrix} \begin{bmatrix} c_\psi e^{j\theta_\gamma} & s_\psi e^{j\theta_\gamma} \\ -s_\psi e^{j\theta_\delta} & c_\psi e^{j\theta_\delta} \end{bmatrix} = \begin{bmatrix} We^{j\theta_w} & Xe^{j\theta_x} \\ 0 & Z \end{bmatrix}$$

where the lower left element is zero and the lower right element is real valued. By evaluating (4.3) for element [2,1] and [2,2], we can determine that

$$\begin{aligned} 0 &= c_\psi s_\phi A e^{j(\theta_\alpha+\theta_\gamma+\theta_a)} + c_\psi c_\phi C e^{j(\theta_\gamma+\theta_\beta+\theta_c)} \\ &\quad - s_\psi s_\phi B e^{j(\theta_\alpha+\theta_\delta+\theta_b)} - s_\psi c_\phi D e^{j(\theta_\delta+\theta_\beta+\theta_d)} \qquad (4.3) \\ Z &= s_\psi s_\phi A e^{j(\theta_\alpha+\theta_\gamma+\theta_a)} + s_\psi c_\phi C e^{j(\theta_\gamma+\theta_\beta+\theta_c)} \\ &\quad + c_\psi s_\phi B e^{j(\theta_\alpha+\theta_\delta+\theta_b)} - c_\psi c_\phi D e^{j(\theta_\delta+\theta_\beta+\theta_d)} \qquad (4.4) \end{aligned}$$

giving us four equations and four unknowns for the angles since the exponential terms are identical. Now, we need to select left and right rotation angles that will satisfy

$$\begin{aligned} -\theta_a &= \theta_\alpha + \theta_\gamma \\ -\theta_c &= \theta_\gamma + \theta_\beta \\ -\theta_b &= \theta_\alpha + \theta_\delta \\ -\theta_d &= \theta_\delta + \theta_\beta \end{aligned}$$

by requiring the angles from (4.3) and (4.4) to be zero. If we restrict our selection to angles such that $\theta_\alpha = \theta_\beta$ and $\theta_\gamma = -\theta_\delta$, we can solve for the angles using two equations in two unknowns

$$\begin{aligned} -\theta_c &= \theta_\alpha + \theta_\gamma \\ -\theta_d &= \theta_\alpha - \theta_\gamma \qquad (4.5) \end{aligned}$$

and select

$$\begin{aligned}\theta_\alpha &= \theta_\beta = \frac{-\theta_d - \theta_c}{2} \\ \theta_\gamma &= -\theta_\delta = \frac{\theta_d - \theta_c}{2}\end{aligned} \tag{4.6}$$

to zero out the phase angle of both elements [2,1] and [2,2]. Likewise, we need to select a θ_ϕ and θ_ψ that will force the magnitude of the element in [2,1], to zero while allowing other elements to remain non-zero. This can be done by selecting

$$\theta_\phi = 0 \tag{4.7}$$

which simplifies (4.3) to

$$0 = c_\psi c_\phi C - s_\psi c_\phi D \tag{4.8}$$

since the complex exponential angles were forced to zero. Solving for θ_ψ, we get

$$\frac{s_\psi}{c_\psi} = \frac{C}{D} \tag{4.9}$$

and therefore we select

$$\theta_\psi = \tan^{-1}\left(\frac{C}{D}\right) \tag{4.10}$$

to force the magnitude of element [2,1] to zero and thus completing the first transformation.

In a fashion similar to the first transformation, we also apply a different variation of a second complex Jacobi rotation to the left and a transposed Jacobi matrix to the right. We begin the second transformation with

$$\begin{bmatrix} c_\lambda e^{j\theta_\epsilon} & -s_\lambda e^{j\theta_\eta} \\ s_\lambda e^{j\theta_\epsilon} & c_\lambda e^{j\theta_\eta} \end{bmatrix} \begin{bmatrix} We^{j\theta_w} & Xe^{j\theta_x} \\ 0 & Z \end{bmatrix} \begin{bmatrix} c_\rho e^{j\theta_\zeta} & s_\rho e^{j\theta_\zeta} \\ -s_\rho e^{j\theta_\omega} & c_\rho e^{j\theta_\omega} \end{bmatrix} = \begin{bmatrix} P & 0 \\ 0 & Q \end{bmatrix}$$

where we want to zero out the non-diagonal elements and generate real values for the diagonal elements. As before, we begin by evaluating (4.11) for element [1,2] and [2,1], and get

$$\begin{aligned}0 &= c_\lambda s_\rho W e^{j(\theta_\epsilon+\theta_\zeta+\theta_w)} + c_\lambda c_\rho X e^{j(\theta_\omega+\theta_\epsilon+\theta_x)} \\ &\quad - s_\lambda c_\rho Z e^{j(\theta_\omega+\theta_\eta)}\end{aligned} \tag{4.11}$$

$$\begin{aligned}0 &= s_\lambda c_\rho W e^{j(\theta_\epsilon+\theta_\zeta+\theta_w)} - s_\lambda s_\rho X e^{j(\theta_\omega+\theta_\epsilon+\theta_x)} \\ &\quad - c_\lambda s_\rho Z e^{j(\theta_\omega+\theta_\eta)}\end{aligned} \tag{4.12}$$

again giving us a matching set of exponential terms. Also, by restricting our selection to $\theta_\epsilon = \theta_\omega$, we can similarly solve for the left and right rotation angles of (4.11) to force the angles of the non-diagonal elements to zero. We start with the set of equations from (4.11)

$$-\theta_w = \theta_\epsilon + \theta_\zeta \tag{4.13}$$

$$-\theta_x = \theta_\omega + \theta_\epsilon \tag{4.14}$$

$$0 = \theta_\omega + \theta_\eta \tag{4.15}$$

and then, using (4.14) and our assumption $\theta_\epsilon = \theta_\omega$, it is simple to show that

$$\theta_\epsilon = -\frac{\theta_x}{2}$$

and

$$\theta_\omega = -\frac{\theta_x}{2}$$

and with (4.15) and our solution for θ_ω we can show that

$$\theta_\eta = -\theta_\omega = \frac{\theta_x}{2}$$

and finally, using (4.13) and substituting our known solution for θ_ϵ we choose

$$\theta_\zeta = -\theta_w - \theta_\epsilon = -\theta_w + \frac{\theta_x}{2}$$

to zero out the angles of the elements of the real diagonal solution matrix. Similar to our approach for the first transformation, we can also solve for the angles θ_λ and θ_ρ that are required in order to zero out the magnitude of the matrix elements [1,2] and [2,1]. Knowing that the angles are zero, we can simplify (4.11) and (4.12) to

$$0 = c_\lambda s_\rho W + c_\lambda c_\rho X - s_\lambda c_\rho Z \tag{4.16}$$

and

$$0 = s_\lambda c_\rho W - s_\lambda s_\rho X - c_\lambda s_\rho Z \tag{4.17}$$

and solve to isolate the magnitudes on one side and the angles on the other. Performing the computation, a selection of

$$\tan(\theta_\lambda - \theta_\rho) = -\frac{X}{Z + W} \tag{4.18}$$

and

$$\tan(\theta_\lambda + \theta_\rho) \quad = \quad \frac{X}{Z - W} \tag{4.19}$$

are required in order to zero the magnitude of the non-diagonal elements of the 2x2 matrix, and thus completing the SVD.

Therefore, we can define the product of the left rotation matrices as U and right rotation matrices as V^H, and the real diagonal solution matrix as the singular value matrix

$$U \;=\; U_2U_1 \;=\; \begin{bmatrix} c_\lambda e^{j\theta_\epsilon} & -s_\lambda e^{j\theta_\eta} \\ s_\lambda e^{j\theta_\epsilon} & c_\lambda e^{j\theta_\eta} \end{bmatrix} \begin{bmatrix} c_\phi e^{j\theta_\alpha} & s_\phi e^{j\theta_\beta} \\ -s_\phi e^{j\theta_\alpha} & c_\phi e^{j\theta_\beta} \end{bmatrix} \tag{4.20}$$

$$S \;=\; \begin{bmatrix} P & 0 \\ 0 & Q \end{bmatrix} \tag{4.21}$$

$$V \;=\; V_1V_2 \;=\; \begin{bmatrix} c_\psi e^{j\theta_\gamma} & s_\psi e^{j\theta_\gamma} \\ -s_\psi e^{j\theta_\delta} & c_\psi e^{j\theta_\delta} \end{bmatrix} \begin{bmatrix} c_\rho e^{j\theta_\zeta} & -s_\rho e^{j\theta_\zeta} \\ s_\rho e^{j\theta_\omega} & c_\rho e^{j\theta_\omega} \end{bmatrix} \tag{4.22}$$

where M can be formed as in (4.2).

4.1.2 Novel Complex 2x2 SVD for Combined Approach

Rather than using a completely CORDIC based method as shown by Hemkumar [11], we will simplify the mathematical expressions to greatly minimize the angular computations. Additional simplifications can be achieved by introducing assumptions about the input matrix. Given an arbitrary complex matrix

$$M \;=\; \begin{bmatrix} \underline{A} & \underline{B} \\ \underline{C} & \underline{D} \end{bmatrix} = \begin{bmatrix} Ae^{j\theta_a} & Be^{j\theta_b} \\ Ce^{j\theta_c} & De^{j\theta_d} \end{bmatrix} \tag{4.23}$$

we wish to solve for the first transformation (4.3). As shown in Section 4.1.1, the desired angles can be computed as

$$\begin{aligned} \theta_\alpha &= \theta_\beta = -\frac{\theta_d + \theta_c}{2} \\ \theta_\gamma &= -\theta_\delta = \frac{\theta_d - \theta_c}{2} \\ \theta_\psi &= \tan^{-1}\left(\frac{C}{D}\right) \\ \theta_\phi &= 0 \end{aligned}$$

which simplifies the first transformation to (see Appendix A for derivation)

$$\left(CD\sqrt{D^2+C^2}\right)^{-1}\begin{bmatrix} D\underline{A}\sqrt{D\underline{C}}^*\sqrt{D\underline{C}^*}-C\underline{B}\sqrt{D\underline{C}}^*\sqrt{C\underline{D}^*} & C\underline{A}\sqrt{D\underline{C}}^*\sqrt{D\underline{C}^*}+D\underline{B}\sqrt{D\underline{C}}^*\sqrt{C\underline{D}^*} \\ D\underline{C}\sqrt{D\underline{C}}^*\sqrt{D\underline{C}^*}-C\underline{D}\sqrt{D\underline{C}}^*\sqrt{C\underline{D}^*} & C\underline{C}\sqrt{D\underline{C}}^*\sqrt{D\underline{C}^*}+D\underline{D}\sqrt{D\underline{C}}^*\sqrt{C\underline{D}^*} \end{bmatrix}$$

The square root elements are then transformed into the form

$$\begin{aligned} \sqrt{D\underline{C}}^*\sqrt{D\underline{C}^*} &= CDe^{-j\left(\frac{\theta_d+\theta_c}{2}\right)}e^{j\left(\frac{\theta_d-\theta_c}{2}\right)} = CDe^{-j\theta_c} = CD\left(\underline{C}^*/C\right) = D\underline{C}^* \\ \sqrt{D\underline{C}}^*\sqrt{C\underline{D}^*} &= CDe^{-j\left(\frac{\theta_d+\theta_c}{2}\right)}e^{j\left(\frac{-\theta_d+\theta_c}{2}\right)} = CDe^{-j\theta_d} = CD\left(\underline{D}^*/D\right) = C\underline{D}^* \end{aligned}$$

which, when a full computation is required, present an inconvenience due to the quadrant related errors in the wrapping of the exponential terms. For example, suppose $\theta_d+\theta_c>2\pi$, but the complex representation would store the angle as $(\theta_d+\theta_c)\,\%\,2\pi$ where % is the modulo operator. Since we are only interested in the relation of the elements in U, we can disregard this inconvenience. Thus, the first transformation can be written as

$$\left(CD\sqrt{D^2+C^2}\right)^{-1}\begin{bmatrix} D^2\underline{A}\,\underline{C}^*-C^2\underline{B}\,\underline{D}^* & DC\underline{A}\,\underline{C}^*+CD\underline{B}\,\underline{D}^* \\ D^2\underline{C}\,\underline{C}^*-C^2\underline{D}\,\underline{D}^* & DC\underline{C}\,\underline{C}^*+CD\underline{D}\,\underline{D}^* \end{bmatrix} = \begin{bmatrix} We^{j\theta_w} & Xe^{j\theta_x} \\ 0 & Z \end{bmatrix} \tag{4.24}$$

when ignoring the quadrant related errors.

Similarly, the second transformation can be evaluated and combined with the first to get the 2x2 complex solutions of U and S. Redefining the first transformation solution

$$\begin{bmatrix} We^{j\theta_w} & Xe^{j\theta_x} \\ 0 & Z \end{bmatrix} = \begin{bmatrix} \underline{W} & \underline{X} \\ 0 & Z \end{bmatrix} \tag{4.25}$$

and from the equation (4.11), with the given angle requirements

$$\begin{aligned} \theta_\epsilon &= \theta_\omega = -\frac{\theta_x}{2} \\ \theta_\eta &= \frac{\theta_x}{2} \\ \theta_\zeta &= \frac{\theta_x}{2}-\theta_w \\ \tan\left(\theta_\rho+\theta_\lambda\right) &= \frac{X}{Z-W} \\ \tan\left(\theta_\rho-\theta_\lambda\right) &= -\frac{X}{Z+W} = \frac{X}{-Z-W} \end{aligned}$$

we can form the solutions to the left and right rotation angles as

$$\begin{aligned} \theta_\lambda &= \frac{1}{2}\tan^{-1}\left(\frac{X}{Z-W}\right)-\frac{1}{2}\tan^{-1}\left(\frac{X}{-Z-W}\right) \\ \theta_\rho &= \frac{1}{2}\tan^{-1}\left(\frac{X}{Z-W}\right)+\frac{1}{2}\tan^{-1}\left(\frac{X}{-Z-W}\right) \end{aligned}$$

Using the trigonometric half angle formulas and realizing the relation of the sum and difference, the simplification defining c_s = 'cosine of the sum', c_d = 'cosine of the difference', s_s, and s_d of θ_λ and θ_ρ

$$c_d = -s_s = \frac{-Z-W}{\sqrt{(-Z-W)^2+X^2}} \tag{4.26}$$

$$c_s = s_d = \frac{Z-W}{\sqrt{(Z-W)^2+X^2}} \tag{4.27}$$

can be used to provide the complete two transformation solution for

$$U^H = \left(\sqrt{CDX}\right)^{-1}\left(\sqrt{\underline{C}\,\underline{D}}\right)^* \begin{bmatrix} \sqrt{\underline{X}^*}c_\lambda & -\sqrt{\underline{X}}s_\lambda \\ \sqrt{\underline{X}^*}s_\lambda & \sqrt{\underline{X}}c_\lambda \end{bmatrix} \tag{4.28}$$

and, the two singular values

$$S[1,1] = \frac{1}{2}(-Wc_d - Wc_s + Xs_s + Xs_d - Zc_d + Zc_s)$$
$$S[2,2] = \frac{1}{2}(-Wc_d + Wc_s - Xs_s + Xs_d - Zc_d - Zc_s)$$

Since this implementation of the combined approach requires only the computation of a single 2x2 complex SVD to estimate a single signal pole and allow a single noise pole, we can place additional assumptions on the above derivation to simplify the hardware requirements. First, the DSS algorithm only requires the use of the U matrix since we are only interested in the poles and not the residues. Thus, the singular values would only be used to determine the proper column of (4.28) to select for further processing. By assuming that the input matrix is element normalized, we can factor a scalar from the original matrix of equation (4.23), which will not affect the elements of U. Since the original matrix is of the form

$$M = \begin{bmatrix} \underline{A} & \underline{B} \\ \underline{C} & \underline{D} \end{bmatrix} = \begin{bmatrix} e^{j\theta_a} & e^{j\theta_b} \\ e^{j\theta_c} & e^{j\theta_d} \end{bmatrix} \tag{4.29}$$

it becomes clear that $Z \geq W \geq 0$ for all input matrices. Therefore, since $c_s > 0$ and $c_d < 0$, the [2,2] singular value will always be larger than [1,1], indicating that we are only interested in two particular elements of (4.28): [2,1] and [2,2]. By selecting elements [2,1] and [2,2] we are selecting the second column of U[1], which will always correspond to the largest singular value according to the assumptions.

Further implementing the DSS algorithm, we wish to solve (3.53) using the first and last row deleted observability matrix, $\hat{O}_{s-}$ and $\hat{O}_{s+}$. In this case, this solution requires a single element computation

[1]Note the Hermitian

and no eigenvalue decomposition to find the pole of interest. By substituting (4.24) into (4.28) the observability elements can be defined as

$$\hat{O}_{s-} = s_\lambda \sqrt{\underline{AC}^* + \underline{BD}^*}^* \sqrt{\underline{CD}}^* = s_\lambda \sqrt{\underline{AC}^2\underline{D} + \underline{BC}\underline{D}^2}^* \tag{4.30}$$

$$\hat{O}_{s+} = c_\lambda \sqrt{\underline{AC}^* + \underline{BD}^*} \sqrt{\underline{CD}}^* = c_\lambda \sqrt{\underline{A}(\underline{CCD})^* + \underline{B}(\underline{CDD})^*} \tag{4.31}$$

and when solving for $\hat{O}_{s-}^{-1} = 1/|\hat{O}_{s-}|e^{j\arg[\hat{O}_{s-}]} = e^{-j\arg[\hat{O}_{s-}]}/|\hat{O}_{s-}|$, we only require the preservation of the angle. This simplifies the inverse to a simple conjugate since, $\hat{O}_{s-}^* = |\hat{O}_{s-}|e^{-j\arg[\hat{O}_{s-}]}$, leading to

$$\hat{A} = \hat{O}_{s-}^{-1}\hat{O}_{s+} = \hat{O}_{s-}^*\hat{O}_{s+} = s_\lambda c_\lambda \sqrt{\underline{AAC}^*\underline{C}^* + 2\underline{ABC}^*\underline{D}^* + \underline{BBD}^*\underline{D}^*}^* \tag{4.32}$$

where $s_\lambda c_\lambda$ can be considered a scalar that will not effect the angle. If we assume that $\underline{B} = \underline{C}$ due to the Hankel structure, we can write

$$\hat{A} = s_\lambda c_\lambda \sqrt{\underline{AAC}^*\underline{C}^* + 2\underline{AD}^* + \underline{CCD}^*\underline{D}^*}^* \tag{4.33}$$

as the final solution. The frequency estimate of the single sinusoid is then given by

$$\hat{f}_1 = \frac{\arg[\hat{A}]f_{dec}}{2\pi} \tag{4.34}$$

where $f_{dec} = f_s/256 = 390625$ Hz is the decimated sample rate. To avoid the square root and conjugate computation in hardware of equation (4.33), the operations can be moved into (4.34) after the angle computation is complete. Since $\arg[\sqrt{\cdot}^*] = -\frac{1}{2}\arg[\cdot]$, a simple shift right by one and negate operator can be used to compute the square root operator and conjugate. Thus, the final equation

$$\hat{f}_1 = \arg[\underline{AAC}^*\underline{C}^* + 2\underline{AD}^* + \underline{CCD}^*\underline{D}^*]\frac{-f_{dec}}{4\pi} \tag{4.35}$$

describes the complete 2x2 complex SVD based DSS solution.

A hardware implementation of equation (4.34) was synthesized on a Altera DSP evaluation kit with a EP2S60F1020C4ES high density FPGA. The fully parallel design requires three 36 bit complex numbers representing the three matrix elements and provides a solution with a propagation delay of 36 clock cycles at 150 MHz. A 28 clock cycle fully parallel CORDIC is used for the angle computation and 138 out of 288 DSP blocks are used for the complex multiplies and final scale operator. To increase the accuracy of the angle computation, a fixed point autoscale block is used to optimize the bits used for the angle computation. The block uses 138 out of 288 DSP blocks and 3625 out of 48352 (Adaptive Look-Up Tables) ALUTs.

4.1.3 Generating the Decimated Data Subset

As described in Section 3.3, generating the decimated data subset for the DSS algorithm requires implementing the FFT, tuning, filtering, IFFT, and decimation blocks. Altera provides a complex FFT IP Core that allows quick integration into the hardware design. Since 1024 samples are collected, we choose a 1024 point FFT. Rather than implement an I/Q split filter to filter the positive frequencies, the imaginary inputs are set to zero to perform a real streaming FFT on the incoming data samples.

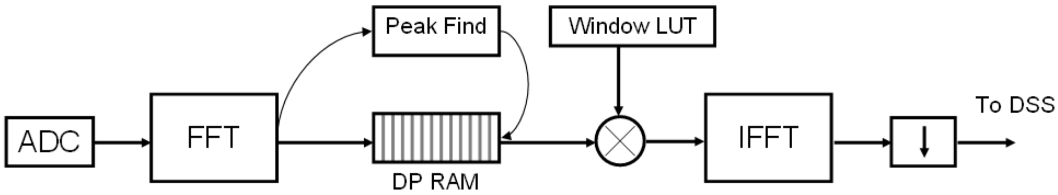

Figure 4.1: Generating the decimated data for the FFT with DSS Algorithm

Figure 4.1 shows the block diagram for generating the decimated data subset for the DSS algorithm. First, the ADC samples are counted and fed into the streaming FFT. The complex output is then stored in a dual port RAM while the peak magnitude of the positive frequency spectra is found. Once the entire FFT is stored in the dual port RAM, the location of the peak magnitude is used as the starting point for the output stream. As the data is read from the output side of the dual port RAM, the appropriate pre-calculated Hanning window weights are applied. The output of the multiplier block now contains the tuned and filtered form of the input waveform in the frequency domain, which is streamed into the IFFT block. While the output of the IFFT starts streaming, the sample number is noted to extract the three complex samples that will be sent to the DSS algorithm.

4.1.4 Complete Implementation

The complete implementation on the Altera DSP evaluation kit contains the FFT with DSS algorithm with several supporting elements of interest. The sinusoid used to test the algorithm is generated by an Altera core based two frequency Numerically Controlled Oscillator (NCO). Each sinusoid is independently generated with an independent amplitude. Also, each independent NCO has the ability to pulse modulate the output by turning on the sinusoid for a given number of clock cycles and then turning off the sinusoid for a longer given number of clock cycles. This design mimics a simple pulsed radar waveform.

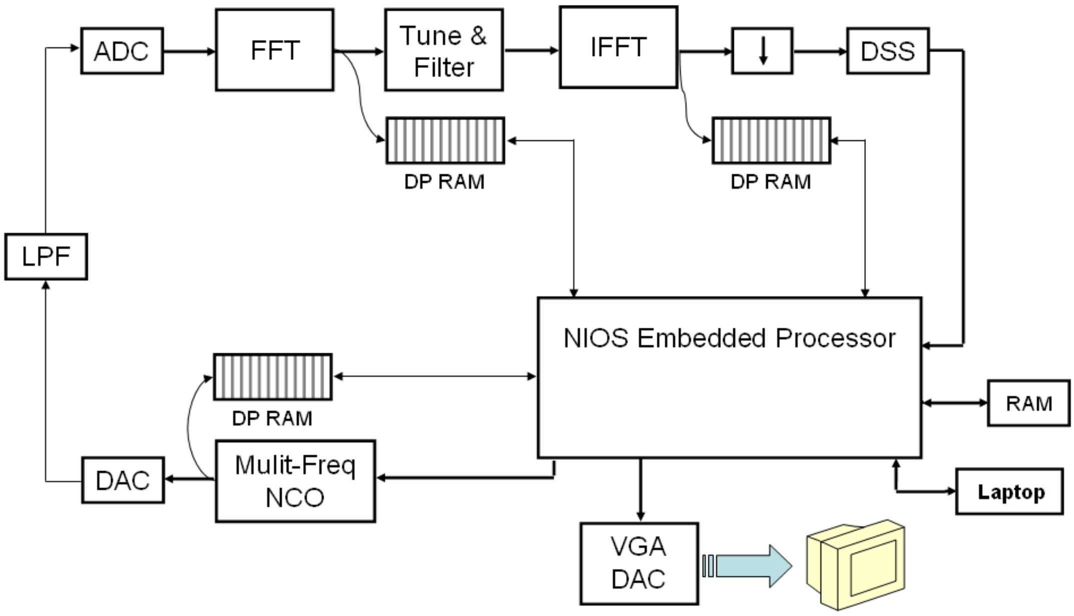

Figure 4.2: FFT with DSS Complete Hardware Implementation

Figure 4.2 shows the block diagram for the complete hardware implementation of the FFT with DSS algorithm. The output of the pulse generating two sinusoid NCO drives the Digital to Analog Converter (DAC), which is then used as the input to the Analog to Digital Converter (ADC). This introduces realistic quantization, thermal, and environmental noise into the data path. The Low Pass Filter (LPF) acts as an anti-aliasing filter and reduces the rise times of the sharp windowing of the pulse generator, adding additional realism. Once digitized by the ADC, the noisy data set is processed by the FFT with DSS algorithm as described in the above two sections and produces a result in approximately 4908 clock cycles at 100 MHz. The ADC and DAC clocks are 100 MHz, the core frequency estimator clock is 100 MHz, and the NIOS embedded processor clock is 50 MHz. The maximum frequency of the core clock in synthesis is 124 MHz.

If time t = 0 specifies the time when the streaming FFT begins, the pipelined implementation generates a series of events at various stages in the algorithm before the frequency estimate is complete. This evaluation allows a simple view into the complex pipelining structure of the implementation. Table 4.1 shows the clock cycle and associated event as the frequency estimate is calculated. Also, this implementation is able to sustain a throughput of 97656.25 solutions per second, or a new solution every 1024 clock cycles at 100 MHz. Optimizations to this implementation are obvious, such as computing a 512 point FFT rather than 1024 and identifying the peak of 512 samples since the input stream is real.

Clock cycle, t	Hardware Event
0	14-bit 1024 point streaming FFT begins
949	FFT output stream starts
965	Tuning block begins
1995	Zero phase filter starts
2000	18-bit 1024 point streaming IFFT starts
4101	IFFT output stream begins
4871	Complex 2x2 SVD based DSS begins
4908	Frequency estimate complete

Table 4.1: FFT with DSS hardware event table

A significant speed improvement to the implementation could be introduced by recognizing that only 3 samples are required for the complex 2x2 SVD based DSS block based on 7 filtered samples. Computing a custom IFFT operator could potentially yield a replacement for the IFFT/decimate components with a single component that requires no multiplies and roughly 15 clock cycles. See Appendix B for more details. For this demonstration, we are only concerned with the statistics and will not discuss further hardware optimizations.

An Altera NIOS embedded processor provides the control of the dual port RAM tap points, NCO, and access by the user through the USB Blaster cable to the NIOS II Embedded Processor Integrated Development Environment (IDE) on a laptop. The external RAM contains both instruction and data memories as well as the video memory used to drive the VGA monitor. Since a simple example using the VGA monitor was available with the evaluation kit, it was copied into this design for a real time verification using the dual port RAM tap points and a custom character set to display the statistics. Upon initialization, the NIOS processor initiates a hardware reset to the external logic, disables the NCO and clears external memory. The next step is to enable the NCO for a small duration, toggle the FFT with DSS algorithm and dual port RAMs, and evaluate the estimated frequency. The remaining statistics, including expected value of the frequency estimate and the MSE, are computed in a similar manner to evaluate the effectiveness of the hardware. The next Section discusses the MSE statistics of the FFT with DSS hardware implementation.

The DSP evaluation kit contains an EP2S60F1020C4ES Stratix II FPGA, which in February 2006 cost roughly $1,100 dollars. Logic Cells (LC) in a Stratix II device contains both a register portion and a combinational portion. An Adaptive Look Up Table (ALUT) consists of a flip flop (lc_ff) and a combinational (lc_comb) section, which have the option of driving various innerconnects. The EP2S60F1020C4ES has a total of 48,352 ALUTs (consisting of 52,506 registers), 2,544,192 memory

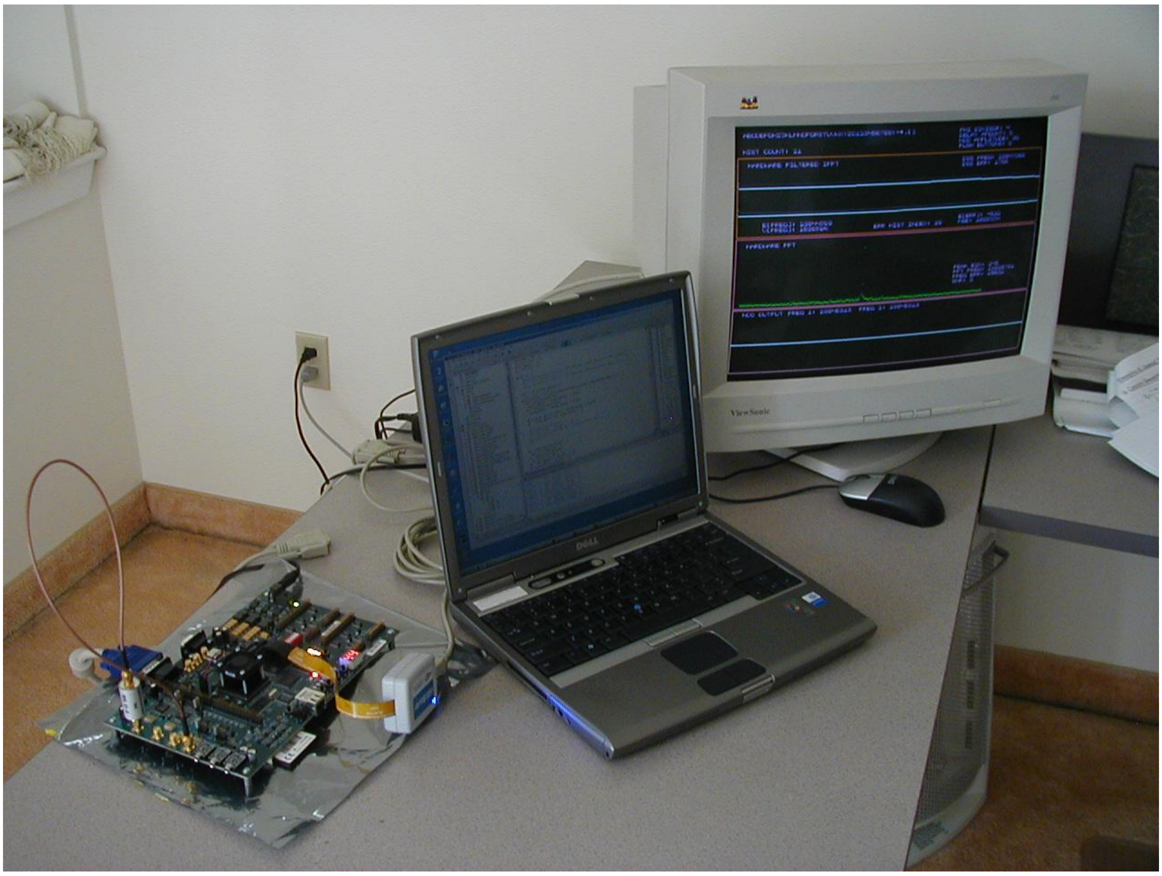

Figure 4.3: Picture of hardware setup

bits, and 288 9-bit DSP multiply blocks.

	LC Registers	LC Combinationals	Memory Bits	DSP Elements
NIOS	3036	4521	63104	8
Multi Freq NCO	3941	2898	14336	4
14-bit Streaming FFT	4135	3020	165016	18
Shift and Filter Block	2901	1264	57456	8
18-bit Streaming IFFT	4985	3456	175256	36
2x2 Complex SVD Based DSS	2798	4705	0	144
	Total resource usage including miscellaneous components			
	Registers	ALUTs	Memory Bits	DSP Elements
Total Usage	22,525 (42%)	29,714 (61%)	503,840 (19%)	222 (77%)

Table 4.2: Stratix II FPGA hardware resource usage

Table 4.2 shows the hardware resource usage of the FFT with DSS implementation. The NIOS embedded processor is quite small and does not require much internal FPGA memory since a large external SRAM and SDRAM are used for code, data, and video memory. The multi-frequency NCO, implemented two independent Altera core CORDIC based oscillators, requires about the same number of registers as the 1024 point streaming FFT and IFFT Altera cores. The main DSP block usage is the fully parallel 2x2 complex SVD based DSS algorithm with a custom CORDIC block. The entire design requires 61 percent ALUT usage, 19 percent internal memory usage, and 77 percent DSP block usage.

4.2 Combined Approach Results

In order to complete a comparison of our hardware implementation with that of the theoretical CRB, we must be able to estimate the current SNR based on the collected hardware FFT data in the NIOS embedded processor. Toner [25] describes a method to approximate the SNR of a single real sinusoid from an FFT as

$$SNR = 10\log_{10}\left(\frac{|X(j)|^2}{\sum_{k=1,k\neq j}^{\frac{N-1}{2}} |X(k)|^2}\right) \tag{4.36}$$

where $X(j)$ is the N-point FFT as defined in (3.2) and $j \in [0, \frac{N-1}{2}]$ defines the single bin that contains majority of the signal power. Clearly, this formula describes the single frequency signal power over the noise power where normalization factors have canceled. Also, if the frequency of interest deviates from the center of the bin significantly, it introduces large unwanted errors in the estimate of the SNR.

The general single quantizer model would consist of a single signal with additive white Gaussian noise that does not change as a function of amplitude. In our setup, for large signals, the non linearity in noise introduced by quantization grows approximately as a $\frac{\delta^2}{12}$ additive noise on top of thermal and environmental influences, where δ is the quantizer step size. The entire model of quantization, thermal, and other noise inducing influences can be verified by measurement, which can provide a meaningful solution in this context.

To reduce the risk of introducing large SNR errors into this evaluation, we estimate the SNR at many different amplitude settings for a single center bin frequency generated by the NCO. Using these data, it is possible to estimate a logarithmic function that best describes this curve. Therefore, when future test sinusoids are not in the center of the bin, a reasonably accurate SNR estimate can be computed. The NCO amplitude setting can then be evaluated using this derived function to compute the estimated SNR. Figure 4.4 shows a 11th order $\log_{10}$ Least Squares fit to the set of data points. The

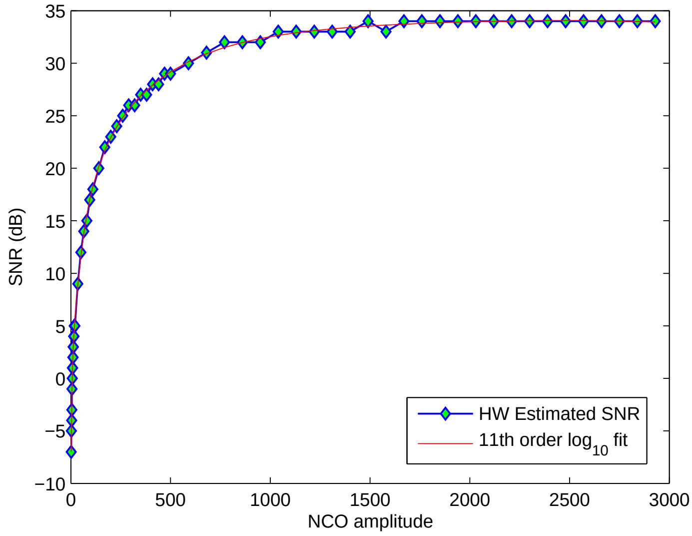

Figure 4.4: NCO linear amplitude to SNR fit with 10th order $\log_{10}$

fit produces results that have errors no larger than about 1 dB, sufficient to proceed with the statistical analysis.

The hardware MSE was computed by evaluating 64 frequency estimates at various NCO amplitude settings. Each 64 point block of estimates along with the truth from the NCO setting were used to compute the $E[(f-\hat{f})^2]$ (MSE), $Var[(f-\hat{f})]$, $E[\hat{f}]$, and $Var[\hat{f}]$. Each estimated SNR was rounded to the nearest integer and each statistical computation was averaged for identical integer SNRs. Figure 4.5 shows the FFT with DSS hardware MSE computed on the NIOS embedded processor using double precision floating point numbers compared to the simulated MSE for the hardware setup described in this chapter. As shown, the results are in excellent agreement for input frequencies in the center and near the center of the FFT bin. Zero point two times b_{sz} off center of bin implies the frequency is $\pm 0.2\, b_{sz}$ displaced from the center of the bin, where $b_{sz} = \frac{f_s}{M}$ is the bin size in Hz, f_s is the sample rate, and M is the number of FFT points. In this implementation, the center of the bin describes a NCO output frequency chosen by $f_{out} = b_{sz}(b_n + \frac{1}{2})$, where b_n is the bin number.

Figure 4.6 compares the hardware computed MSE and the simulated MSE for an input frequency that is $0.4 * b_{sz}$ off the center of the bin. While evaluating the hardware it became evident that the current temperature condition of the ADC plays an important role in estimating the frequency accurately. The top of Figure 4.6 shows the computed hardware MSE under cool conditions, or just when starting the hardware. The bottom of Figure 4.6 shows the condition when the ADC is not fitted

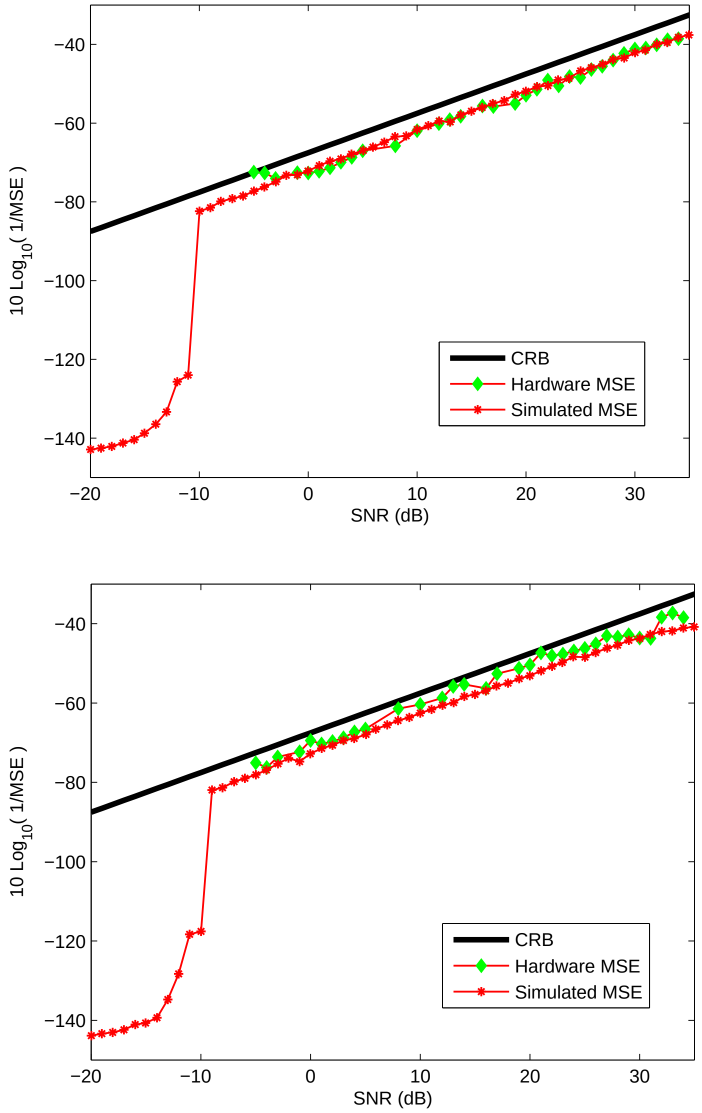

Figure 4.5: Input frequency in (top) center of bin and (bottom) $0.2 * b_{sz}$ off center of bin

properly with heat dissipation devices for this application. Since the calculation method for the SNR remains the same and the two computed MSEs are nearly identical except a constant MSE degradation, it became clear that the noise floor of the ADC changes somewhat significantly as the device heats.

Since $Var[f - \hat{f}] = MSE - E^2[err] = E[(f - \hat{f})^2] - E^2[f - \hat{f}]$ is close to the CRB, it is interesting to notice that $E[f - \hat{f}]$ deviates from zero significantly under different conditions, leading to larger MSE errors. Figure 4.7 shows a 64 sample average of the error in the frequency estimate for different input frequencies and SNR. When the input frequency is close to the center of the bin, the average

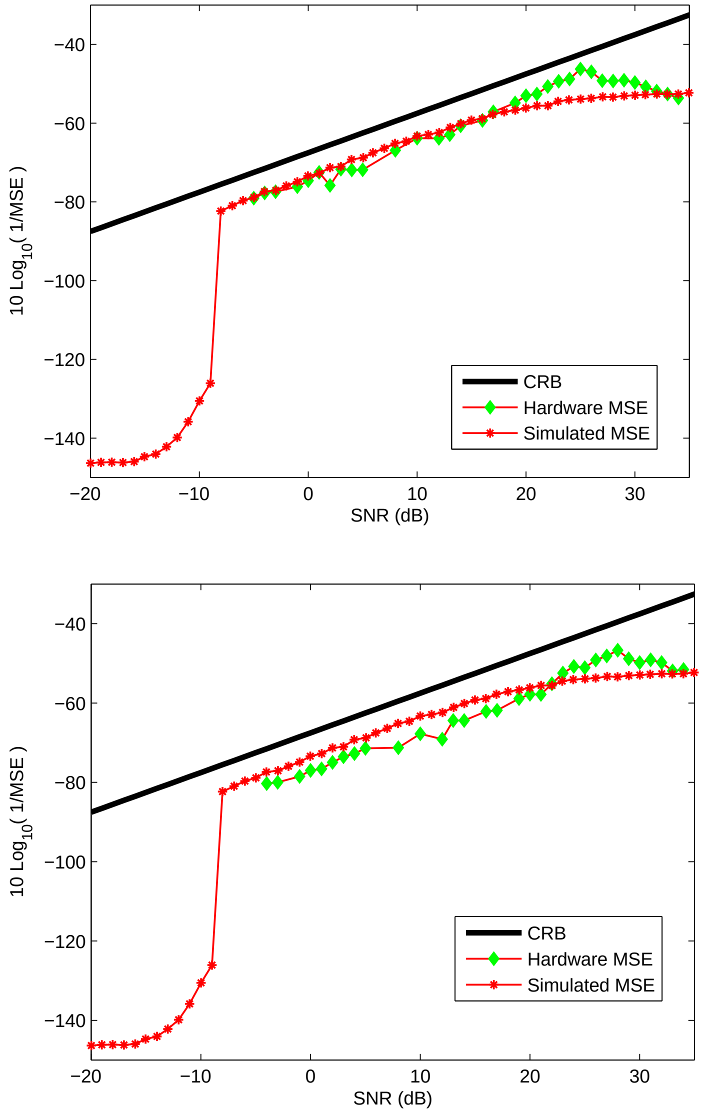

Figure 4.6: Point four times b_{sz} off center of bin with (top) cool ADC (bottom) hot ADC

error is very close to zero. Frequencies that are closer to the edge of the bin result in larger average error. Also, in the case where the input frequency is $0.4 * b_{sz}$ from the center of the bin, there is a bit of an overshoot from zero in the higher SNRs, which causes the rapid separation from the CRB around 25 dB SNR as seen in Figure 4.6. Under hot ADC conditions, the average error is clearly larger and takes slightly longer to correct as SNR increases.

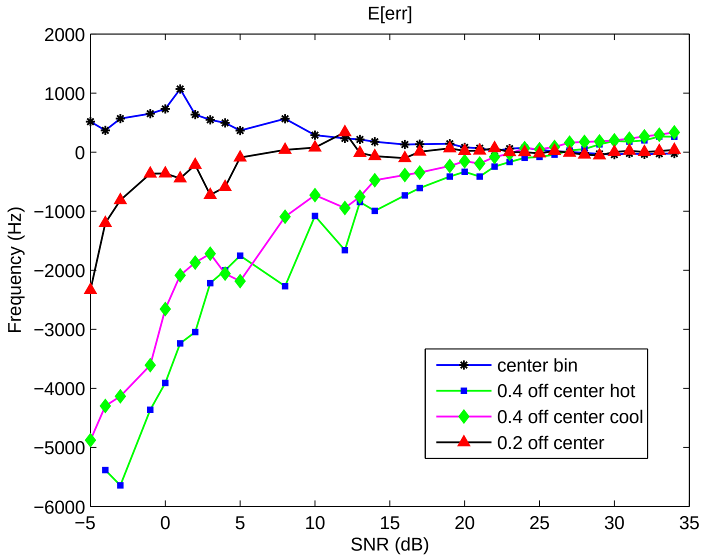

Figure 4.7: $E[f - \hat{f}]$ for different input frequencies vs SNR

4.3 Expanding the Combined Implementation

The hardware implementation described in this chapter requires 1024 real ADC samples, which are filtered and decimated into just 3 samples needed to populate a 2x2 Hankel matrix. Optimizations specific to this scenario were produced to significantly simplify the hardware design. Expansions to the combined implementation using different sample rates and longer time integrations will not allow the same assumptions that allowed the area optimizations for the unique 2x2 SVD case. If, however, the different implementations were produced, the expected performance statistics can be generated.

Table 4.3 shows the statistics for the same implementation using different sample rates. The simulations are similar to the those in Section 3.3.2, where the FFT with DSS (FWD) algorithm performance was tested using a 1024 point sliding window over 10240 samples at various sample rates. The statistics shown for the mean, standard deviation, and MSE were taken as the average of three input frequencies: one in the center of the bin, one at $0.2 * b_{sz}$ from the center of the bin, and one $0.4 * b_{sz}$ from the center of the bin[2]. Similar to the hardware and simulator MSE plots seen in Figure 4.5, there appears to be a constant offset from the CRB of about 5-6 dB near 0 dB SNR for all cases as well as a point in the larger SNRs where the MSE becomes bounded. Appendix C contains the expected performance statistics in the form of tables similar to Table 4.3 for many possible combinations of several available

[2] b_{sz} is the bin size.

	1024 real samples $\rightarrow$ 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.87	-82.52			-155.25	-94.56
-10			-129.54	-77.52			-131.37	-89.56
-5	543	7939	-78.35	-72.52	3216	30038	-89.29	-84.56
0	634	4240	-72.57	-67.52	1241	17395	-84.99	-79.56
10	90	1383	-62.88	-57.52	647	5513	-74.91	-69.56
20	201	409	-54.06	-47.52	546	1552	-64.91	-59.56
30	142	141	-48.05	-37.52	546	547	-59.98	-49.56
40	136	46	-46.57	-27.52	529	164	-58.43	-39.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-167.51	-106.04			-169.45	-108.54
-10			-152.41	-101.04			-142.49	-103.54
-5	5616	116986	-101.50	-96.04	30789	164221	-104.69	-98.54
0	1430	64205	-96.26	-91.04	5657	88287	-98.85	-93.54
10	2569	20349	-86.24	-81.04	2421	28320	-89.08	-83.54
20	2036	6514	-77.12	-71.04	3104	8355	-79.65	-73.54
30	2068	2080	-71.41	-61.04	2635	2839	-73.72	-63.54
40	2049	642	-70.17	-51.04	2610	857	-72.50	-53.54

Table 4.3: 1024 real data samples decimated into a 2x2 complex rank revealing SVD

ADC sample rates, real or complex data, different SNRs, number of samples collected, and the size of the complex SVD required.

4.4 Other Implementations

Based on the DFT derivation in Chapter 3, the Radix-2 FFT would be quite easy to implement in a parallel form using either of the two common Radix-2 or Radix-4 algorithms. Each stage could be pipelined to easily provide a streaming FFT on the order of several thousand points. Xilinx and Altera both provide FFT cores that accomplish this task. The FFT algorithm provides an elegant solution to the wideband frequency detection problem since a fast ADC sample stream could be polyphase filtered and fed into a modified FFT algorithm. This could provide a rapid, coarse frequency solution for a broad bandwith. The Least Squares solution shown in Section 3.1.2 requires autocorrelation, a pseudo inverse using a NxP+1 SVD, then solving a system of equations for the a_k parameters, and finally a partial fraction expansion formulation. The computational burden is quite large for this implementation and as seen from its performance in Section 3.2, it doesn't make sense to implement this method for rapid frequency estimation.

Chapter 5

Future Work

Additional work should be conducted to further explore and develop the area of rapid frequency estimation. The FFT with DSS algorithm as implemented here can provide practical performance improvements to many applications. As the density and complexity of FPGAs continue to increase, it will be worthwhile to extend the simplified 2x2 complex SVD implementation to a larger matrix. This would also allow multiple sinusoidal frequencies to be estimated in or around the single FFT bin of interest. This section will describe a systolic hardware architecture developed at Rice University and then extend it to a Compact Architecture that allows a complex SVD to be implemented using the resources available.

5.1 Compact Architecture for a Complex SVD

The bulk of the computation time for the Direct State Space (DSS) algorithm is the computation of the complex Singular Value Decomposition (SVD). There exists a vast repository of resources dedicated to the computation of both real and complex SVDs. Since the selected method requires a complex implementation, we will focus our attention first on the real systolic processor array structure introduced by Brent, Luk and Van Loan [26] and expanded by Cavallaro and Luk [12], Yang and Bohme [27], and Ahmedsaid, et al. [28]. This structure has been expanded to compute a complex SVD by Hsiao and Delosme [13], Adams et. al [29], Hemkumar [30], and Kota [10] [31], and in some cases, the unitary U and V matrices. A few additional resources which lead to this development include [32] [33] [34] [35] [36] [37] [38] [39] [40] and [30]. Following the complex systolic SVD architecture, a adaptive hardware architecture is introduced. This architecture is adaptive in the sense that several variations exist that utilize the resources available, increasing throughput performance as more resources are used.

5.1.1 Solving a Complex SVD Using a Compact Architecture

Several methods for solving the Singular Value Decomposition (SVD) have been presented in literature. In Section 2.3 an introduction to a hardware friendly method for real matrices was provided using Jacobi rotation matrices using equation (2.19). Brent, Luk, and Van Loan [26] have shown a parallel algorithm to reduce the computation of a real SVD of a $n \times n$ matrix from $O(n^3)$ to $O(n \log n)$ using $\left(\frac{n}{2}\right)^2$ Processing Elements (PEs). Expanding on these implementations, many references cited in the introduction have presented systolic architecture solutions for complex Singular Value Decompositions.

This section extends the derivation of the two step transformation to solve a 2x2 complex SVD in the previous chapter, which is the basic building block of an nxn systolic complex SVD. The linear algebra is then manipulated to fit a convenient form for the well developed CORDIC algorithm that is commonly used for computing rotation based functions in dedicated hardware. Once the single Processing Element (PE) structure is defined for both diagonal and off diagonal elements, a new architecture is presented to compute large complex SVDs in a flexible structure using a Compact Systolic Architecture.

Solving the Complex SVD Rotations using the CORDIC Algorithm

The computation of the rotation angles are trivial using the CORDIC algorithm as described in Section 2.4. It is necessary to compute the angles and magnitudes of selected matrix elements. In this section, several parallel hardware structures are presented for the computation of the rotation angles as well as the matrix elements for the left and right rotation matrices for the first and second transformations. Parallel structures are implemented in order to allow the implementation of the Compact Systolic Architecture for the complex SVD. The number above each block in the following figures indicates the number of pipeline stages required to meet the 150 MHz minimum clock speed in an Altera Stratix II FPGA.

Figure 5.1 illustrates a parallel hardware structure that uses the CORDIC algorithm to compute the rotation angles for the first transformation following equations (4.6) and (4.10). The inputs to the hardware block are the elements $m_{2,1} = Ce^{j\theta_c}$ and $m_{2,2} = De^{j\theta_d}$, from which the magnitude and angle are directly computed. Using the ratio of the magnitudes as derived for (4.10), the inverse tangent is computed using another CORDIC block. The phases of the input elements are shifted, inverted, subtracted, and pipelined as necessary to produce outputs that coincide with the parameter θ_ψ.

Similar to the first transformation, the second transformation requires the computation of the angle and magnitude. Figure 5.2 shows the block diagram for a parallel hardware structure needed to

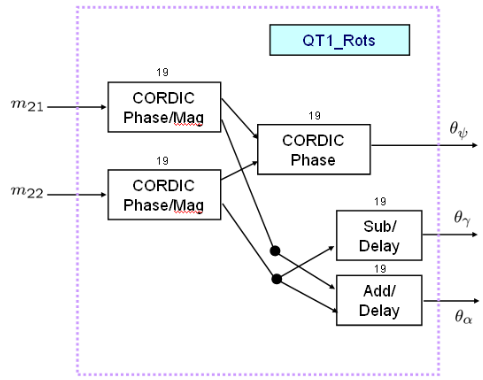

Figure 5.1: First transformation rotation angle computation structure

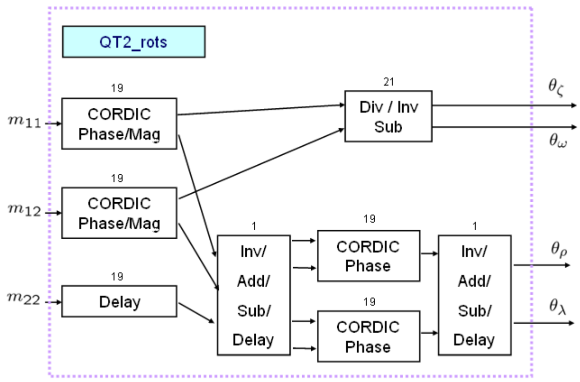

Figure 5.2: Second transformation rotation angle computation structure

compute the rotation angles of the second transformation following the equations just before (4.16), equation (4.18), and (4.19). The angle and magnitude of the inputs, $w_{1,1} = We^{j\theta_w}$, $w_{1,2} = Xe^{j\theta_x}$, and $w_{2,2} = Z$, are first determined. Arriving at the same clock cycle after the computation, the phases are added, inverted, shifted, and pipelined as necessary to compute the exponential angles θ_ζ and θ_ω. The magnitudes are added, subtracted, and processed by another inverse tangent parallel CORDIC block. The solutions of the inverse tangent are used to solve the remaining left and right rotation angles. All the outputs are pipelined accordingly to produce a rotation angle precisely on the same clock cycle for all the inputs.

Once the rotation angles have been computed using the CORDIC algorithm, it is necessary to compute the matrix elements of equations (4.3) and (4.11). In order to make use of this simple iterative structure of the CORDIC, it is necessary to manipulate the complex 2x2 SVD arithmetic into a CORDIC form using sin(), cos(), $\tan^{-1}()$, and $\sqrt{x^2+y^2}$. Starting with the first transformation in equation (4.3), we can redefine the left rotation matrix as

$$R_{1l} \triangleq \begin{bmatrix} c_\phi e^{j\theta_\alpha} & -s_\phi e^{j\theta_\beta} \\ s_\phi e^{j\theta_\alpha} & c_\phi e^{j\theta_\beta} \end{bmatrix} \tag{5.1}$$

where the indices $1l$ define the left rotation matrix of the first transformation. Similarly, the right rotation is redefined as

$$R_{1r} \triangleq \begin{bmatrix} c_\psi e^{j\theta_\gamma} & s_\psi e^{j\theta_\gamma} \\ -s_\psi e^{j\theta_\delta} & c_\psi e^{j\theta_\delta} \end{bmatrix} \tag{5.2}$$

where the indices $1r$ define the right rotation matrix of the first transformation. Using the condition $\theta_\psi = 0$ and $\theta_\alpha = \theta_\beta$ defined in the derivation (4.7), R_{1l} can be simplified to

$$R_{1l} = \begin{bmatrix} e^{j\theta_\alpha} & 0 \\ 0 & e^{j\theta_\beta} \end{bmatrix} = \begin{bmatrix} cos(\theta_\alpha) + j\,sin(\theta_\alpha) & 0 \\ 0 & cos(\theta_\alpha) + j\,sin(\theta_\alpha) \end{bmatrix} \tag{5.3}$$

which can be implemented using a single CORDIC block after the angle α has been computed using (4.6). Similarly, we can solve for R_{1r} using the condition $\theta_\delta = -\theta_\gamma$

$$R_{1r} = \begin{bmatrix} c_\psi e^{j\theta_\gamma} & s_\psi e^{j\theta_\gamma} \\ -s_\psi e^{j\theta_\delta} & c_\psi e^{j\theta_\delta} \end{bmatrix} = \begin{bmatrix} c_\psi c_\gamma + j\,c_\psi s_\gamma & s_\psi c_\gamma + j\,s_\psi s_\gamma \\ -s_\psi c_\gamma + j\,s_\psi s_\gamma & c_\psi c_\gamma - j\,c_\psi s_\gamma \end{bmatrix} \tag{5.4}$$

and using trigonometric identities

$$R_{1r} = \begin{bmatrix} x_{1r} + j\,y_{1r} & z_{1r} + j\,w_{1r} \\ -z_{1r} + j\,w_{1r} & x_{1r} - j\,y_{1r} \end{bmatrix} \tag{5.5}$$

where the duplicated real and imaginary components are defined as

$$\begin{aligned} w_{1r} &= \frac{1}{2}cos(\theta_\gamma - \theta_\psi) - \frac{1}{2}cos(\theta_\gamma + \theta_\psi) \\ x_{1r} &= \frac{1}{2}cos(\theta_\gamma + \theta_\psi) + \frac{1}{2}cos(\theta_\gamma - \theta_\psi) \\ y_{1r} &= \frac{1}{2}sin(\theta_\gamma + \theta_\psi) + \frac{1}{2}sin(\theta_\gamma - \theta_\psi) \\ z_{1r} &= \frac{1}{2}sin(\theta_\gamma + \theta_\psi) - \frac{1}{2}sin(\theta_\gamma - \theta_\psi) \end{aligned} \tag{5.6}$$

which requires two CORDIC blocks to compute the sin() and cos() of $\theta_\gamma \pm \theta_\psi$. Once the sin() and cos() terms are computed, the computations of (5.6) clearly conform to an attractive hardware solution using only a few extra shifts, additions, and subtractions. Thus, the implementation shown resolving R_{1l} and R_{1r} can be completed using hardware shifts and additions, saving the limited dedicated multiplier blocks for complex multiplication of the matrices.

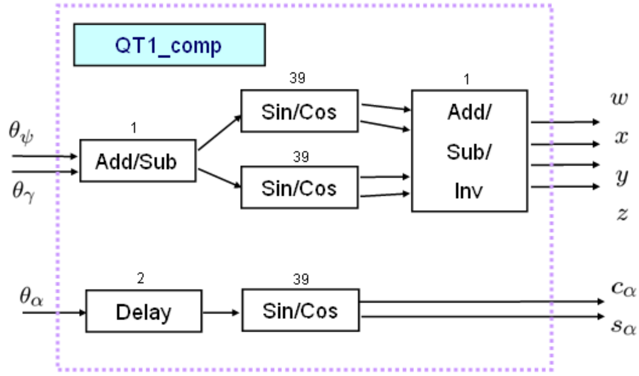

Figure 5.3: First transformation matrix element computation structure

A hardware block diagram for the computation of the rotation matrix elements for the first transformation is seen in Figure 5.3. Each of the previously computed rotation angles using the CORDIC algorithm are passed as inputs. After adding and subtracting the rotation angles as described above, they are passed to the sin() and cos() CORDIC blocks. The outputs of the CORDIC blocks are then shifted and inverted as necessary to produce the matrix elements of (5.3) and (5.6). A hardware simulation of this block was implemented in a high density Altera Stratix II using a 36 bit fixed point complex representation[1]. The fully parallel design synthesized with a maximum clock rate of about 170 MHz and has a propagation delay of 41 clock cycles.

A similar derivation can be used to define the simplified solutions to the second transformation. Redefining the left rotation matrix from (4.11) as

$$R_{2l} \triangleq \begin{bmatrix} c_\lambda e^{j\theta_\epsilon} & -s_\lambda e^{j\theta_\eta} \\ s_\lambda e^{j\theta_\epsilon} & c_\lambda e^{j\theta_\eta} \end{bmatrix} \tag{5.7}$$

and the right rotation matrix as

$$R_{2r} \triangleq \begin{bmatrix} c_\rho e^{j\theta_\zeta} & s_\rho e^{j\theta_\zeta} \\ -s_\rho e^{j\theta_\omega} & c_\rho e^{j\theta_\omega} \end{bmatrix} \tag{5.8}$$

where the indices $2l$ and $2r$ indicate the left and right rotation matrix of the second transformation. Using the same procedure as above for the first transformation, it is simple to show that since $\theta_\eta = -\theta_\epsilon$,

$$R_{2l} = \begin{bmatrix} R_{2lx} + j\,R_{2lz} & -R_{2ly} + j\,R_{2lw} \\ R_{2ly} + j\,R_{2lw} & R_{2lx} - j\,R_{2lz} \end{bmatrix} \tag{5.9}$$

[1] 18 bits for the real and 18 bits for the imaginary

where the real and imaginary components are defined as

$$\begin{aligned}
R_{2lw} &= \frac{1}{2}cos(\theta_\lambda - \theta_\epsilon) - \frac{1}{2}cos(\theta_\lambda + \theta_\epsilon) \\
R_{2lx} &= \frac{1}{2}cos(\theta_\lambda + \theta_\epsilon) + \frac{1}{2}cos(\theta_\lambda - \theta_\epsilon) \\
R_{2ly} &= \frac{1}{2}sin(\theta_\lambda + \theta_\epsilon) + \frac{1}{2}sin(\theta_\lambda - \theta_\epsilon) \\
R_{2lz} &= \frac{1}{2}sin(\theta_\lambda + \theta_\epsilon) - \frac{1}{2}sin(\theta_\lambda - \theta_\epsilon)
\end{aligned} \tag{5.10}$$

which are implemented with two CORDIC sin() and cos() blocks. Since there is no simple symmetry in the right rotation matrix of the second transformation, it can be shown to be

$$R_{2r} = \begin{bmatrix} R_{2rzx} + j\,R_{2rzz} & R_{2rzy} + j\,R_{2rzw} \\ -R_{2rwy} - j\,R_{2rww} & R_{2rwx} + j\,R_{2rwz} \end{bmatrix} \tag{5.11}$$

where the components are

$$\begin{aligned}
R_{2rzw} &= \frac{1}{2}cos(\theta_\rho - \theta_\zeta) - \frac{1}{2}cos(\theta_\rho + \theta_\zeta) \\
R_{2rzx} &= \frac{1}{2}cos(\theta_\rho + \theta_\zeta) + \frac{1}{2}cos(\theta_\rho - \theta_\zeta) \\
R_{2rzy} &= \frac{1}{2}sin(\theta_\rho + \theta_\zeta) + \frac{1}{2}sin(\theta_\rho - \theta_\zeta) \\
R_{2rzz} &= \frac{1}{2}sin(\theta_\rho + \theta_\zeta) - \frac{1}{2}sin(\theta_\rho - \theta_\zeta)
\end{aligned} \tag{5.12}$$

and

$$\begin{aligned}
R_{2rww} &= \frac{1}{2}cos(\theta_\rho - \theta_\omega) - \frac{1}{2}cos(\theta_\rho + \theta_\omega) \\
R_{2rwx} &= \frac{1}{2}cos(\theta_\rho + \theta_\omega) + \frac{1}{2}cos(\theta_\rho - \theta_\omega) \\
R_{2rwy} &= \frac{1}{2}sin(\theta_\rho + \theta_\omega) + \frac{1}{2}sin(\theta_\rho - \theta_\omega) \\
R_{2rwz} &= \frac{1}{2}sin(\theta_\rho + \theta_\omega) - \frac{1}{2}sin(\theta_\rho - \theta_\omega)
\end{aligned} \tag{5.13}$$

which can be expressed using four CORDIC sin() and cos() blocks. The expressions to compute the real and imaginary components for both the left and right rotation matrices for the first and second transformation can be computed using shifts and additions. This makes this implementation attractive for a FPGA hardware implementation.

Figure 5.4 shows a hardware block diagram for the computation of the rotation matrix elements for the second transformation. Each of the previously computed rotation angles for the second transformation are applied as inputs. After adding and subtracting the rotation angles as described above,

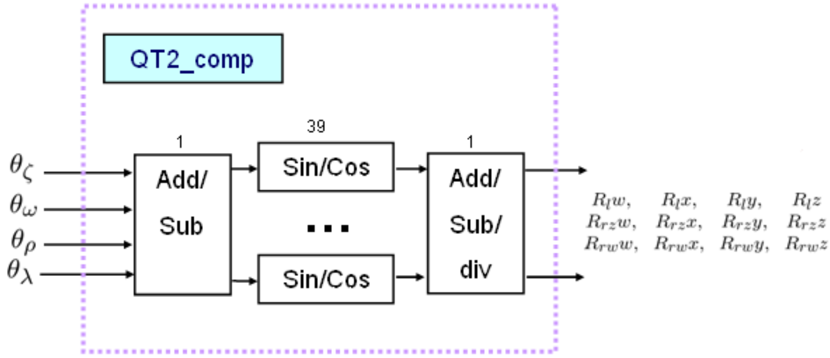

Figure 5.4: Second transformation matrix element computation structure

they are passed to the sin() and cos() CORDIC blocks. The outputs of the CORDIC blocks are then shifted and inverted as necessary to produce the matrix elements of (5.10) to (5.13). It is noteworthy to point out that each hardware structure presented is fully parallel. The hardware structures to compute the rotation angles require nearly the same number of clock cycles while the number of clock cycles to compute the matrix elements are exactly the same. It will become evident that the parallel structure of these hardware components is required in the implementation of the Compact Systolic Architecture for the complex SVD. This portion of the hardware synthesized with a maximum clock rate of 170 MHz and a 41 clock cycle propagation delay.

Completing the Complex 2x2 SVD using the CORDIC Algorithm

In the previous section, we described the necessary modifications to the left and right rotation matrices to conform to the CORDIC Algorithm. To complete the 2x2 SVD, we must solve the matrices of equation (4.3) and (4.11) for the singular values and accumulated left rotation matrices, U. Thus, the solution to the first transformation can be computed using the notation of (5.6) as

$$\begin{aligned}
w_{1,1} &= (c_\alpha + j\, s_\alpha)\Big[m_{1,1}(x_{1r} + j\, y_{1r}) + m_{1,2}(-z_{1r} + j\, w_{1r})\Big] \\
w_{1,2} &= (c_\alpha + j\, s_\alpha)\Big[m_{1,1}(z_{1r} + j\, w_{1r}) + m_{1,2}(x_{1r} - j\, y_{1r})\Big] \\
w_{2,1} &= (c_\alpha + j\, s_\alpha)\Big[m_{2,1}(x_{1r} + j\, y_{1r}) + m_{2,2}(-z_{1r} + j\, w_{1r})\Big] \\
w_{2,2} &= (c_\alpha + j\, s_\alpha)\Big[m_{2,1}(z_{1r} + j\, w_{1r}) + m_{2,2}(x_{1r} - j\, y_{1r})\Big] \qquad (5.14)
\end{aligned}$$

where $m_{1,1}$ to $m_{2,2}$ define the elements of the complex input matrix and $w_{1,1}$ to $w_{2,2}$ define the elements of the solution. This structure can be used to compute both the left rotation matrix as well as aid in the computation of the singular values. Section 5.1.1 describes the systolic architecture that enables the dual use. When computing the singular values using the second transformation, as equation (4.11) suggests, the form of the matrix will be

$$\begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix} = \begin{bmatrix} We^{j\theta_w} & Xe^{j\theta_x} \\ 0 & Z \end{bmatrix}$$

where $w_{2,1} = 0$ and $\Im[w_{2,2}] = 0$[2]. As will be discussed in the following sections, when computing the accumulation of the left rotation matrices, U, the elements $w_{1,1}$ to $w_{2,2}$ will generally be arbitrary and complex.

The second transformation, using the notation of (5.10) to (5.13), can be described as

$$\begin{aligned}
s_{1,1} &= (R_{2rzx} + j\,R_{2rzz})\Big[w_{1,1}(R_{2lx} + j\,R_{2lz}) + w_{2,1}(-R_{2ly} + j\,R_{2lw})\Big] \\
&\quad + (-R_{2rwy} - j\,R_{2rww})\Big[w_{1,2}(R_{2lx} + j\,R_{2lz}) + w_{2,2}(-R_{2ly} + j\,R_{2lw})\Big] \\
s_{1,2} &= (R_{2rzy} + j\,R_{2rzw})\Big[w_{1,1}(R_{2lx} + j\,R_{2lz}) + w_{2,1}(-R_{2ly} + j\,R_{2lw})\Big] \\
&\quad + (R_{2rwx} + j\,R_{2rwz})\Big[w_{1,2}(R_{2lx} + j\,R_{2lz}) + w_{2,2}(-R_{2ly} + j\,R_{2lw})\Big] \\
s_{2,1} &= (R_{2rzx} + j\,R_{2rzz})\Big[w_{1,1}(R_{2ly} + j\,R_{2lw}) + w_{2,1}(R_{2lx} - j\,R_{2lz})\Big] \\
&\quad + (-R_{2rwy} - j\,R_{2rww})\Big[w_{1,2}(R_{2ly} + j\,R_{2lw}) + w_{2,2}(R_{2lx} - j\,R_{2lz})\Big] \\
s_{2,2} &= (R_{2rzy} + j\,R_{2rzw})\Big[w_{1,1}(R_{2ly} + j\,R_{2lw}) + w_{2,1}(R_{2lx} - j\,R_{2lz})\Big] \\
&\quad + (R_{2rwx} + j\,R_{2rwz})\Big[w_{1,2}(R_{2ly} + j\,R_{2lw}) + w_{2,2}(R_{2lx} - j\,R_{2lz})\Big] \qquad (5.15)
\end{aligned}$$

where $w_{1,1}$ to $w_{2,2}$ define the elements of the complex input matrix from the first transformation and $s_{1,1}$ to $s_{2,2}$ define the elements of the solution. This structure is used only to compute the singular values from the solution of the first transformation matrix. As the next section illustrates, the first multiplier structure can be used to compute the left rotation matrix, U. When computing the singular values, as equation (4.11) states, the form of the solution matrix will be

$$\begin{bmatrix} s_{1,1} & s_{1,2} \\ s_{2,1} & s_{2,2} \end{bmatrix} = \begin{bmatrix} P & 0 \\ 0 & Q \end{bmatrix}$$

[2] $\Im[\cdot]$ is the imaginary component

where $s_{1,2}$ and $s_{2,1}$ are zero and the diagonal elements are real. This simplifies the hardware structure of the equations in (5.15) since $s_{1,2}$ and $s_{2,1}$ are zero and need not be computed. Also, since this structure is only required for the singular values, we can assume $w_{2,1}$ is zero and $w_{2,2}$ is real, further simplifying the demanding multiplier structure.

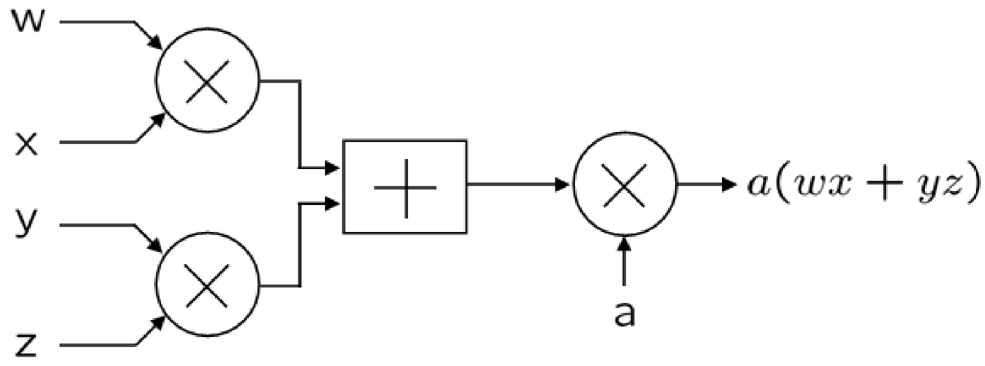

Figure 5.5: Basic hardware multiply-add structure

For the general implementation, we generate a basic hardware multiply-add component seen in Figure 5.5 where w, x, y, z, and a are complex. A single implementation of this structure is required for each of the of the first transformation elements, $w_{1,1}$ to $w_{2,2}$. For the second transformation, we only require portions of the basic multiply-add block. For the diagonal elements, $s_{1,1}$ and $s_{2,2}$, two basic modified multiply-add blocks are required in addition to a complex adder. Some authors prefer to implement the entire structure and allow the compiler to optimize away the unnecessary multipliers, logic elements, and registers.

Systolic Architecture for a Complex SVD

A common approach to computing a large SVD from many small 2x2 SVDs is using a systolic array of processors. Brent, Luk and Van Loan [26] introduced a systolic array method for real matrices, which was extended to the complex case by several authors including Hemkumar [30].

In general, the systolic array architecture requires $\left(\frac{n}{2}\right)^2$ Processing Elements (PEs) to compute an nxn complex SVD. Figure 5.6 shows a systolic architecture for computing a 8x8 complex SVD. Each diagonal processing element computes the two transformation based 2x2 complex SVD described in the previous section. The rotation angles are then passed along the horizontal and vertical connections between PEs. Each non-diagonal processing element applies the rotation angles as they are received,

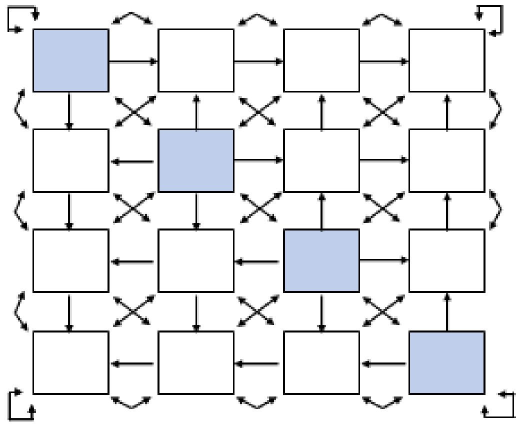

Figure 5.6: Systolic architecture for a complex SVD

and then upon completion of a single sub-sweep, the elements among the processing elements are swapped in a specific manner to make the diagonal element in the processor cluster converge to the singular values. Once the systolic cluster completes $n-1$ sub-sweeps and the same starting element is swapped back to its original starting position, a whole sweep is complete.

Figure 5.7 shows the sweep pattern for the 8x8 complex systolic SVD structure, where Σ_1 indicates computing the first transformation and Σ_2 indicates the processor is computing the second transformation. As mentioned above, the diagonal elements begin the first transformation, after which the rotation parameters are passed to the horizontal and vertical neighbors. Similarly, the second transformation immediately follows, along with the computation of the U matrix. It is left to the reader to refer to the many references cited in Section 5.1 for further information on real and complex systolic arrays for computing SVDs.

Conventional single processor algorithms require $O(n^3)$ operations to compute the SVD. The clear advantage of this systolic architecture is that a real SVD in this type of parallel systolic architecture requires roughly $O(n \log(n))$ operations, where $\log(n)$ is the number of sweeps required. Based on simulations by Hemkumar [30], the complex systolic architecture requires slightly more sweeps than $\log(n)$.

Compact Architecture for a Complex SVD

Generally, in a systolic architecture, a parallel implementation of the rotation computation described in the previous section is undesirable due to the extremely large FPGA area requirements.

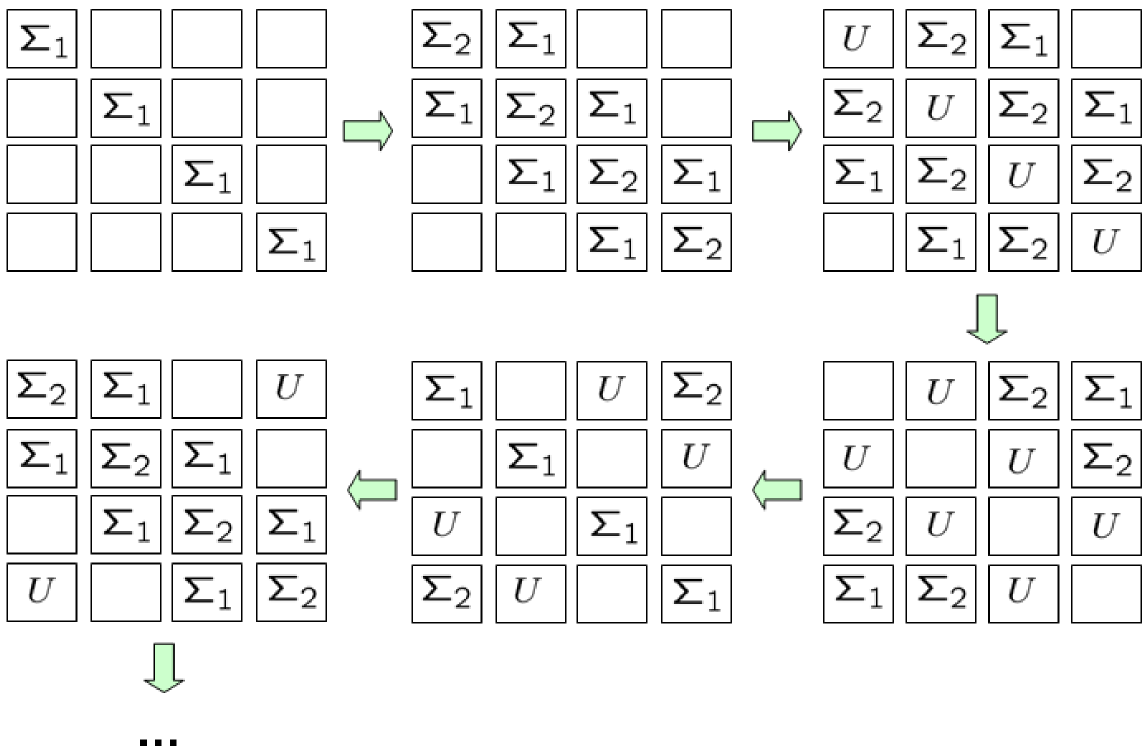

Figure 5.7: Systolic architecture processor activity

Serial architectures for computing CORDIC rotations at each individual Processing Element (PE) are common. The Compact Architecture for a Complex SVD presented in this section takes advantage of the fully parallel nature for computing the first and second transformations, singular values, and U matrix; all desirable for the DSS algorithm implementation.

Figure 5.8 shows the overview of the compact SVD architecture. The complex valued samples resulting from the I/Q split of ADC samples are first stored into a FIFO with independent clocks to allow the clock rate change from 100 MHz to 150 MHz. Once sufficient samples are stored in the FIFO, the load logic builds the matrix format desired into the four dual port RAMs, one RAM per element of a 2x2 PE. This allows the contents of an entire PE to be retrieved or stored in a single clock cycle. Once the master state machine determines that the matrix is full, the elements of the diagonal PEs are streamed through the fully parallel first and second transformations, storing the required rotation parameters into many smaller dual port RAMs. After the rotation parameters are stored, the entire contents of the recently loaded RAM are fed through a massively parallel multiple 2x2 matrix multiplication, applying the correct rotation parameters at the appropriate clock cycle. The results are

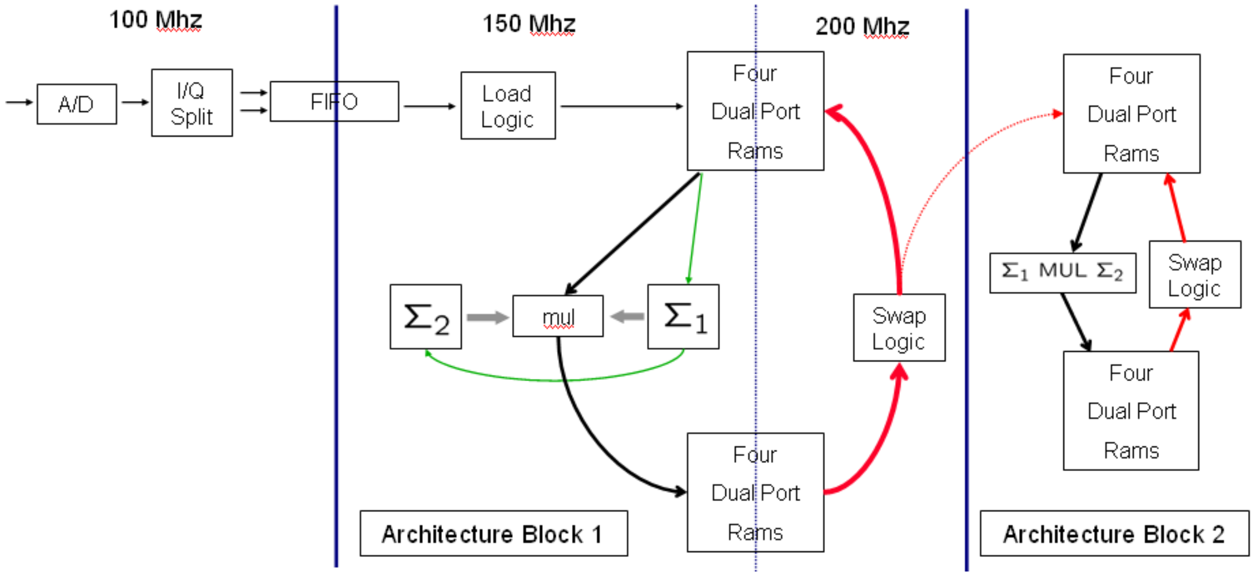

Figure 5.8: Block diagram for Compact Architecture for Complex SVD

streamed into a second temporary dual port RAM, from which the elements can be swapped at a higher clock rate into the first dual port RAM, completing a single sub-sweep of the systolic architecture. If the matrix size is nxn, roughly $n-1$ sub sweep iterations are required in order to compute a full sweep.

Duplicates of the architecture block can be added as area allows to increase the throughput of the complex SVD engine. Rather than swapping the data elements from the second solution dual port RAM into the first dual port RAM initially loaded with complex ADC samples, the solutions can be swapped after a fixed number of iterations into a duplicate architecture. As the next section iterates, a few additional architecture blocks can improve performance significantly.

Figure 5.9 looks more closely at the parallel implementation of the first and second transformations using the hardware blocks detailed in Section 5.1.1. As mentioned above, the first step is to compute the rotation parameters using the elements of the diagonal PEs. The bottom elements of each PE are first streamed into the QT1_rots block to compute the rotation parameters of the first transformation, which are stored in small dual port RAMs for later use. Also, while loading the first rotation parameters, the first transformation elements, complex X, complex Y, and real Z are computed nearly simultaneously. The solutions are immediately used to compute and store the second rotation parameters in a similar manner. This fully parallel implementation has roughly a 180 clock cycle propagation delay and can run faster than 150 MHz, after which each clock cycle produces a new set of rotation parameters from the diagonal PEs. Once the rotation parameters are stored, the entire matrix memory of the four dual

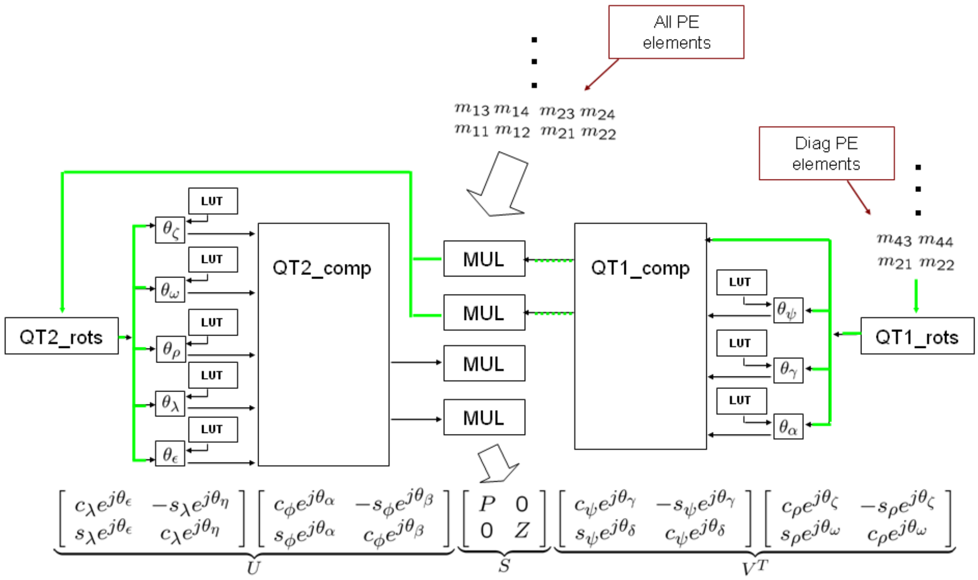

Figure 5.9: Fully parallel first and second transformations

port RAMs are streamed for processing through the massive parallel complex multiply chain. The Look Up Tables (LUTs) are loaded with the correct addresses and timed to allow the transformation matrices to be computed and applied to the matrix elements on the appropriate clock cycle as they stream through the multiply chain.

If the U and V matrices are desired, a duplicate set of four dual port RAMs are required for each matrix and the master control state machine needs to be modified to accommodate the streaming of additional data through the parallel multiply structure.

5.1.2 Compact Complex SVD Architecture Results

The compact complex SVD architecture implementation allows large size matrices to be processed using nearly the same number of Altera Stratix II DSP elements and Logic Elements (LEs). The main cause of area increase is the amount of memory required to store the current working matrix that converges to the singular values, the U matrix, the swap logic tables, and the additional rotation parameters.

The estimated speed of several hardware implementations are compared to a MATLAB complex SVD computation using a 2 Ghz processor with 2G of RAM in Figure 5.10. For matrices smaller than 64x64, the compact complex SVD architecture is expected to be faster than a single high end processor running MATLAB. If d architecture blocks are implemented in a chain as described in the previous

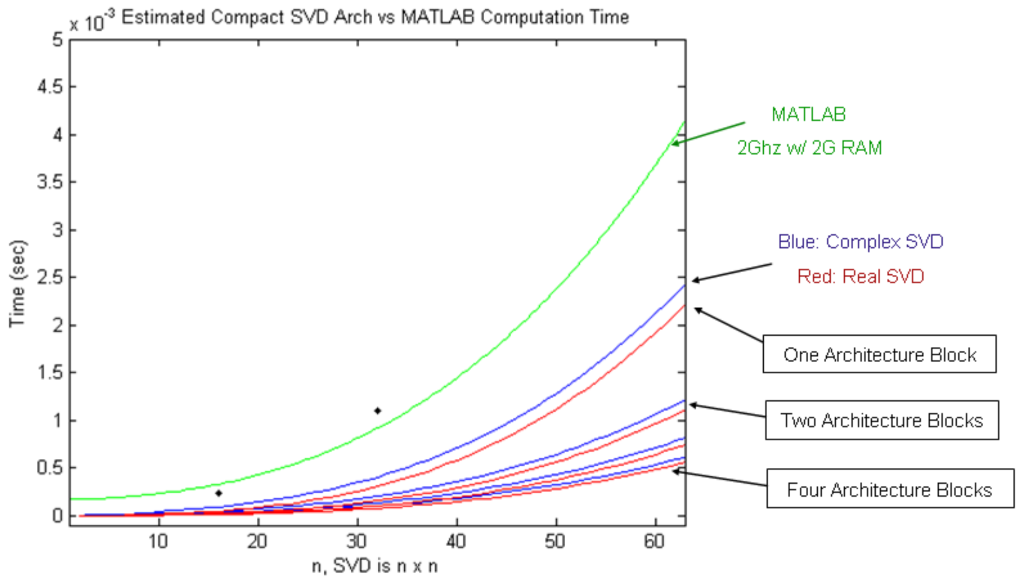

Figure 5.10: Compact complex SVD estimated speed vs MATLAB on 2Ghz with 2G RAM

section and the computed matrix elements are passed down the chain after $\frac{1}{d}$ of the sub sweeps are complete, the expected performance improvement is initially quite dramatic. With just two architecture blocks, a significant improvement can be seen in the computation time (throughput, not propagation delay) of larger matrices. Each subsequent duplicate architecture block speeds the throughput by an increasingly smaller amount. In every case, the real SVD is just slightly faster than the complex SVD, but the area requirements are significantly more expensive for the complex SVD.

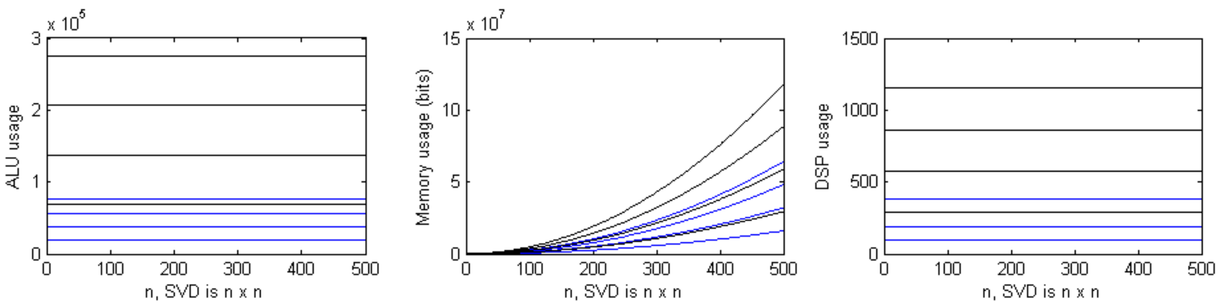

Figure 5.11: Compact complex SVD estimated area usage on Altera Stratix II

Figure 5.11 shows the estimated area required to compute the singular values and the U matrix in a high density Altera Stratix II FPGA. The blue line in the plots indicate the real SVD expected area usage and the black lines indicate the expected complex SVD area usage. The lines closest to zero ALU usage mark the smallest expected area usage in either the complex or real case indicate a single architecture block, the second lowest marks two architecture blocks, and so on. It is clear that the DSP and ALU area usage remain relatively constant given a specific number of architecture blocks and matrix size, while the amount of memory increases very rapidly as the matrix size increases. This makes sense since there are a fixed number of multipliers regardless of the size of the matrix. The elements are just streamed through the parallel transformations and multiplications.

One drawback to this architecture is that the convergence of the singular values in the working matrix make for several increasingly larger values and several increasingly smaller values, especially in high SNR signal subspace estimation, where a single or several dominant singular values are much larger than the smallest few singular values. This makes for a difficult fixed point implementation since the accuracy of the transformations rely on near full precision. For example, assume there are 18 bits used for the real and 18 bit for the imaginary portion of each matrix element. If a 12 bit ADC is used, there are about 6 additional bits the singluar values can grow. As the sub sweeps continue, suppose the largest singular value grows to 17 bits, but the smallest singular value has shrank to 3 bits. When the transformations compute angles based on the large and small magnitudes, the results will become increasingly inaccurate as the sub sweeps continue. Modifications, such as a floating point implementation, would have to be made to make this architecture useful in signal subspace estimation for large matrices. If applications exist where the singular values are known to be similar, a fixed point implementation may be practical.

Chapter 6

Summary and Conclusions

Frequency estimation plays an important role in many digital signal processing applications. Many applications have benefited from discoveries over the last few decades ranging from the Fast Fourier Transform (FFT) decades ago to modern spectral techniques. In this thesis, a technique has been presented and implemented to provide a transition for a modern spectral method to various real time applications, including Electronic Counter Measure (ECM) techniques.

A common method of frequency estimation, the FFT, was shown to have reasonable performance for frequency estimation. It is an attractive option for FPGAs since the algorithm can be parallelized and provides a solution faster than any other algorithm considered in this thesis with the same hardware. In some cases, further accuracy is required, that is, an estimate that is closer to the CRB than the FFT can provide. The Least Squares (LS) technique, which involves matching the input data points to an all pole frequency domain model, requires far too many operations, including a large pseudo-inverse and a partial fraction expansion. The Direct State Space (DSS) solution, wherein the collected data samples are used to build a state space model to extract the poles, is an attractive option due to the exceptional accuracy it provides. The large size Singular Value Decomposition (SVD) required to estimate the signal subspace, however, can become impractical in many real time applications. This led to the development of the Combined Approach, using both an FFT and the DSS algorithms.

The FFT with DSS algorithm first uses the FFT to approximately locate the areas of spectral interest. Once the bins of interest are decided, each bin gets shifted in the frequency domain to DC, where a zero phase Hanning filter is applied. The resultant waveform is converted back into the time domain and decimated to fit a small size SVD for signal subspace approximation and state space pole finding. A dramatic improvement in the Mean Square Error (MSE) of the estimates shown against the

Cramer Rao Bound (CRB) in negative Signal to Noise Ratios (SNRs) is possible due to the filtering operation. Effectively, the error in the FFT based solution is estimated using the high accuracy of the DSS algorithm.

A hardware implementation of the FFT with DSS algorithm was constructed on a Altera Stratix II DSP evaluation kit. The hardware implementation contained a simple Numerically Controlled Oscillator (NCO), which generated a radar type pulsed waveform and realized it with a Digital to Analog Converter (DAC). The low pass filtered version of the analog waveform was digitized and passed to a 1024 point real streaming FFT. The peak magnitude of the FFT in the positive frequency space was determined and used to stream the tuned and filtered waveform into a streaming IFFT. Just three samples of the IFFT output were used in a fully parallel optimized 2x2 DSS realization. The propagation delay through the hardware FWD implementation[1] is 49.08 μs with a throughput of 10.24 μs using a 100 Mhz internal clock. Remarkably, using a fixed point hardware implementation and a double precision floating point embedded processor to compute the statistics, the MSE of the frequency estimates were in excellent agreement with simulated double precision floating point arithmetic. The MSE of the estimate in both the hardware and software simulations were about 5 to 6 dB from the CRB using 1024 samples in the calculation. The Tables are available in Appendix C that estimate the performance of various expanded forms of this hardware implementation using different sample rates, real or imaginary input data, SVD sizes, SNRs and number of samples collected.

The FFT with DSS hardware structure now allows large time integration applications with high sample rates that demand accuracy to be implemented in real time. In areas such as Electronic Counter Measures, where time is of the essence, these techniques can play a crucial role in providing accurate frequency estimates to threat identification and ECM employment systems, increasing overall system effectiveness.

[1]Obvious optimizations could reduce the propagation delay to roughly 30 μs and throughput to roughly 5 μs by adjusting the FFT implementation

Appendix A

Complex 2x2 SVD Derivation

Given an arbitrary complex matrix

$$M = \begin{bmatrix} \underline{A} & \underline{B} \\ \underline{C} & \underline{D} \end{bmatrix} = \begin{bmatrix} Ae^{j\theta_a} & Be^{j\theta_b} \\ Ce^{j\theta_c} & De^{j\theta_d} \end{bmatrix}$$

we wish to solve the first transformation

$$\begin{bmatrix} c_\phi e^{j\theta_\alpha} & -s_\phi e^{j\theta_\beta} \\ s_\phi e^{j\theta_\alpha} & c_\phi e^{j\theta_\beta} \end{bmatrix} \begin{bmatrix} Ae^{j\theta_a} & Be^{j\theta_b} \\ Ce^{j\theta_c} & De^{j\theta_d} \end{bmatrix} \begin{bmatrix} c_\psi e^{j\theta_\gamma} & s_\psi e^{j\theta_\gamma} \\ -s_\psi e^{j\theta_\delta} & c_\psi e^{j\theta_\delta} \end{bmatrix} = \begin{bmatrix} We^{j\theta_w} & Xe^{j\theta_x} \\ 0 & Z \end{bmatrix} \tag{A.1}$$

for the following angles

$$\begin{aligned}
\theta_\alpha &= \theta_\beta = -\frac{\theta_d + \theta_c}{2} \\
\theta_\gamma &= -\theta_\delta = \frac{\theta_d - \theta_c}{2} \\
\theta_\psi &= tan^{-1}\left(\frac{C}{D}\right) \\
\theta_\phi &= 0
\end{aligned}$$

Since

$$\begin{aligned}
e^{j\theta_\beta} &= e^{j\theta_\alpha} = e^{-j\left(\frac{\theta_d+\theta_c}{2}\right)} = \sqrt{e^{j\theta_d}\, e^{j\theta_c}}^{\,*} = \sqrt{(\underline{D}/D)(\underline{C}/C)}^{\,*} = \sqrt{(DC)^{-1}}^{\,*}\sqrt{\underline{DC}}^{\,*} \\
e^{j\theta_\gamma} &= e^{j\left(\frac{\theta_d-\theta_c}{2}\right)} = \sqrt{e^{j\theta_d}\ \left(e^{j\theta_c}\right)^*} = \sqrt{(\underline{D}/D)(\underline{C}/C)^*} = \sqrt{(DC)^{-1}}^{\,*}\sqrt{\underline{DC}^*} \\
e^{j\theta_\delta} &= e^{j\left(\frac{-\theta_d+\theta_c}{2}\right)} = \sqrt{e^{j\theta_c}\ \left(e^{j\theta_d}\right)^*} = \sqrt{(\underline{C}/C)(\underline{D}/D)^*} = \sqrt{(DC)^{-1}}^{\,*}\sqrt{\underline{C}\underline{D}^*} \\
c_\psi &= \frac{D}{\sqrt{D^2 + C^2}} \\
s_\psi &= \frac{C}{\sqrt{D^2 + C^2}}
\end{aligned}$$

we can simplify the first transformation from equation (A.1) as

$$\left(\sqrt{(CD)^{-1}}\right)^2 \begin{bmatrix} \sqrt{\underline{DC}}^* & 0 \\ 0 & \sqrt{\underline{DC}}^* \end{bmatrix} \begin{bmatrix} \underline{A} & \underline{B} \\ \underline{C} & \underline{D} \end{bmatrix} \begin{bmatrix} \frac{D\sqrt{\underline{DC}^*}}{\sqrt{D^2+C^2}} & \frac{C\sqrt{\underline{DC}^*}}{\sqrt{D^2+C^2}} \\ -\frac{C\sqrt{\underline{CD}^*}}{\sqrt{D^2+C^2}} & \frac{D\sqrt{\underline{CD}^*}}{\sqrt{D^2+C^2}} \end{bmatrix} = \begin{bmatrix} We^{j\theta_w} & Xe^{j\theta_x} \\ 0 & Z \end{bmatrix}$$

$$\left(CD\sqrt{D^2+C^2}\right)^{-1} \begin{bmatrix} \sqrt{\underline{DC}}^* & 0 \\ 0 & \sqrt{\underline{DC}}^* \end{bmatrix} \begin{bmatrix} \underline{A} & \underline{B} \\ \underline{C} & \underline{D} \end{bmatrix} \begin{bmatrix} D\sqrt{\underline{DC}^*} & C\sqrt{\underline{DC}^*} \\ -C\sqrt{\underline{CD}^*} & D\sqrt{\underline{CD}^*} \end{bmatrix}$$

$$\left(CD\sqrt{D^2+C^2}\right)^{-1} \begin{bmatrix} \underline{A}\sqrt{\underline{DC}}^* & \underline{B}\sqrt{\underline{DC}}^* \\ \underline{C}\sqrt{\underline{DC}}^* & \underline{D}\sqrt{\underline{DC}}^* \end{bmatrix} \begin{bmatrix} D\sqrt{\underline{DC}^*} & C\sqrt{\underline{DC}^*} \\ -C\sqrt{\underline{CD}^*} & D\sqrt{\underline{CD}^*} \end{bmatrix}$$

$$\left(CD\sqrt{D^2+C^2}\right)^{-1} \begin{bmatrix} D\underline{A}\sqrt{\underline{DC}}^*\sqrt{\underline{DC}^*} - C\underline{B}\sqrt{\underline{DC}}^*\sqrt{\underline{CD}^*} & C\underline{A}\sqrt{\underline{DC}}^*\sqrt{\underline{DC}^*} + D\underline{B}\sqrt{\underline{DC}}^*\sqrt{\underline{CD}^*} \\ D\underline{C}\sqrt{\underline{DC}}^*\sqrt{\underline{DC}^*} - C\underline{D}\sqrt{\underline{DC}}^*\sqrt{\underline{CD}^*} & C\underline{C}\sqrt{\underline{DC}}^*\sqrt{\underline{DC}^*} + D\underline{D}\sqrt{\underline{DC}}^*\sqrt{\underline{CD}^*} \end{bmatrix}$$

where since

$$\begin{aligned} \left(\sqrt{CD}\right)^{-1}\sqrt{\underline{DC}}^* &= e^{-j\left(\frac{\theta_d+\theta_c}{2}\right)} &\implies \sqrt{\underline{DC}}^* &= \left(\sqrt{CD}\right)e^{-j\left(\frac{\theta_d+\theta_c}{2}\right)} \\ \left(\sqrt{CD}\right)^{-1}\sqrt{\underline{DC}^*} &= e^{j\left(\frac{\theta_d-\theta_c}{2}\right)} &\implies \sqrt{\underline{DC}^*} &= \left(\sqrt{CD}\right)e^{j\left(\frac{\theta_d-\theta_c}{2}\right)} \\ \left(\sqrt{CD}\right)^{-1}\sqrt{\underline{CD}^*} &= e^{j\left(\frac{-\theta_d+\theta_c}{2}\right)} &\implies \sqrt{\underline{CD}^*} &= \left(\sqrt{CD}\right)e^{j\left(\frac{-\theta_d+\theta_c}{2}\right)} \end{aligned}$$

the common terms can be computed ignoring the quadrant related errors since we are only interested in the relation of the elements in U

$$\begin{aligned} \sqrt{\underline{DC}}^*\sqrt{\underline{DC}^*} &= CDe^{-j\left(\frac{\theta_d+\theta_c}{2}\right)}e^{j\left(\frac{\theta_d-\theta_c}{2}\right)} = CDe^{-j\theta_c} = CD\left(\underline{C}^*/C\right) = D\underline{C}^* \\ \sqrt{\underline{DC}}^*\sqrt{\underline{CD}^*} &= CDe^{-j\left(\frac{\theta_d+\theta_c}{2}\right)}e^{j\left(\frac{-\theta_d+\theta_c}{2}\right)} = CDe^{-j\theta_d} = CD\left(\underline{D}^*/D\right) = C\underline{D}^* \end{aligned}$$

simplifies the first transformation to

$$\left(CD\sqrt{D^2+C^2}\right)^{-1} \begin{bmatrix} D^2\underline{A}\,\underline{C}^* - C^2\underline{B}\,\underline{D}^* & DC\underline{A}\,\underline{C}^* + CD\underline{B}\,\underline{D}^* \\ D^2\underline{C}\,\underline{C}^* - C^2\underline{D}\,\underline{D}^* & DC\underline{C}\,\underline{C}^* + CD\underline{D}\,\underline{D}^* \end{bmatrix} = \begin{bmatrix} We^{j\theta_w} & Xe^{j\theta_x} \\ 0 & Z \end{bmatrix} = \begin{bmatrix} \underline{W} & \underline{X} \\ 0 & Z \end{bmatrix}$$

Similar to first transformation, we begin the second transformation with the known set of equations

$$\begin{bmatrix} c_\lambda e^{j\theta_\epsilon} & -s_\lambda e^{j\theta_\eta} \\ s_\lambda e^{j\theta_\epsilon} & c_\lambda e^{j\theta_\eta} \end{bmatrix} \begin{bmatrix} We^{j\theta_w} & Xe^{j\theta_x} \\ 0 & Z \end{bmatrix} \begin{bmatrix} c_\rho e^{j\theta_\zeta} & s_\rho e^{j\theta_\zeta} \\ -s_\rho e^{j\theta_\omega} & c_\rho e^{j\theta_\omega} \end{bmatrix} = \begin{bmatrix} P & 0 \\ 0 & Q \end{bmatrix}$$

with

$$\begin{aligned}
\theta_\epsilon &= \theta_\omega = -\frac{\theta_x}{2} \\
\theta_\eta &= \frac{\theta_x}{2} \\
\theta_\zeta &= \frac{\theta_x}{2} - \theta_w \\
\tan\left(\theta_\rho + \theta_\lambda\right) &= \frac{X}{Z-W} \\
\tan\left(\theta_\rho - \theta_\lambda\right) &= -\frac{X}{Z+W} = \frac{X}{-Z-W}
\end{aligned}$$

From the system

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_\lambda \\ \theta_\rho \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{X}{Z-W}\right) \\ \tan^{-1}\left(\frac{X}{-Z-W}\right) \end{bmatrix}$$

it is clear that

$$\begin{aligned}
\theta_\lambda &= \frac{1}{2}\tan^{-1}\left(\frac{X}{Z-W}\right) - \frac{1}{2}\tan^{-1}\left(\frac{X}{-Z-W}\right) \\
\theta_\rho &= \frac{1}{2}\tan^{-1}\left(\frac{X}{Z-W}\right) + \frac{1}{2}\tan^{-1}\left(\frac{X}{-Z-W}\right)
\end{aligned}$$

which will be used later to form $c_\lambda, s_\lambda, c_\rho, s_\rho$. First, using $c_s =$ 'cosine of the sum', $c_d =$ 'cosine of the difference', s_s, and s_d of θ_λ and θ_ρ

$$\begin{aligned}
c_d &= \frac{-Z-W}{\sqrt{(-Z-W)^2 + X^2}} \\
c_s &= \frac{Z-W}{\sqrt{(Z-W)^2 + X^2}} \\
s_d &= \frac{X}{\sqrt{(-Z-W)^2 + X^2}} = c_s \\
s_s &= \frac{X}{\sqrt{(Z-W)^2 + X^2}} = -c_d
\end{aligned}$$

we can simplify to when $Z \leq W$ (we never need to consider the case when $-Z < W$ since $W, Z \geq 0 \,\forall\, W, Z \in \Re$)

$$\begin{aligned}
F_1 &= \sin\left[\frac{1}{2}tan^{-1}\left(\frac{X}{Z-W}\right)\right] = -\sqrt{(1+c_s)/2} \\
F_2 &= \cos\left[\frac{1}{2}tan^{-1}\left(\frac{X}{Z-W}\right)\right] = \sqrt{(1-c_s)/2} \\
F_3 &= \sin\left[\frac{1}{2}tan^{-1}\left(\frac{X}{-Z-W}\right)\right] = -\sqrt{(1+c_d)/2} \\
F_4 &= \cos\left[\frac{1}{2}tan^{-1}\left(\frac{X}{-Z-W}\right)\right] = \sqrt{(1-c_d)/2}
\end{aligned}$$

else $(Z > W)$

$$
\begin{aligned}
F_1 &= \sin\Big[\frac{1}{2}tan^{-1}\Big(\frac{X}{Z-W}\Big)\Big] = \sqrt{(1-c_s)/2} \\
F_2 &= \cos\Big[\frac{1}{2}tan^{-1}\Big(\frac{X}{Z-W}\Big)\Big] = \sqrt{(1+c_s)/2} \\
F_3 &= \sin\Big[\frac{1}{2}tan^{-1}\Big(\frac{X}{-Z-W}\Big)\Big] = -\sqrt{(1+c_d)/2} \\
F_4 &= \cos\Big[\frac{1}{2}tan^{-1}\Big(\frac{X}{-Z-W}\Big)\Big] = \sqrt{(1-c_d)/2}
\end{aligned}
$$

from which we can form

$$
\begin{aligned}
c_\rho &= F_2F_4 - F_1F_3 \\
s_\rho &= F_1F_4 + F_2F_3 \\
c_\lambda &= F_2F_4 + F_1F_3 \\
s_\lambda &= F_1F_4 - F_2F_3
\end{aligned}
$$

providing the complete two transformation solution for

$$
U^H = \left(\sqrt{CDX}\right)^{-1}\left(\sqrt{\underline{C}\,\underline{D}}\right)^{*}\begin{bmatrix} \sqrt{\underline{X}^*}c_\lambda & -\sqrt{\underline{X}}s_\lambda \\ \sqrt{\underline{X}^*}s_\lambda & \sqrt{\underline{X}}c_\lambda \end{bmatrix}
$$

and

$$
S = \begin{bmatrix} Wc_\rho c_\lambda - Xs_\rho c_\lambda + Zs_\rho s_\lambda & Ws_\rho c_\lambda + Xc_\rho c_\lambda - Zc_\rho s_\lambda \\ Wc_\rho s_\lambda - Xs_\rho s_\lambda - Zs_\rho c_\lambda & Ws_\rho s_\lambda + Xc_\rho s_\lambda + Zc_\rho c_\lambda \end{bmatrix}
$$

where the elements [1,1] and [2,2] are the singular values. In hardware, we can simplify as follows if $Z \leq W$

$$
\begin{aligned}
c_\lambda c_\rho &= -\frac{1}{2}c_d - \frac{1}{2}c_s \\
s_\lambda s_\rho &= -\frac{1}{2}c_d + \frac{1}{2}c_s \\
s_\lambda c_\rho &= -\frac{1}{2}s_s + \frac{1}{2}s_d \\
c_\lambda s_\rho &= -\frac{1}{2}s_s - \frac{1}{2}s_d
\end{aligned}
$$

else $(Z > W)$

$$
\begin{aligned}
c_\lambda c_\rho &= -\frac{1}{2}c_d + \frac{1}{2}c_s \\
s_\lambda s_\rho &= -\frac{1}{2}c_d - \frac{1}{2}c_s \\
s_\lambda c_\rho &= \frac{1}{2}s_s + \frac{1}{2}s_d \\
c_\lambda s_\rho &= \frac{1}{2}s_s - \frac{1}{2}s_d
\end{aligned}
$$

which simplifies the singular values to

$$\begin{aligned} S[1,1] &= \frac{1}{2}\left(-Wc_d - Wc_s + Xs_s + Xs_d - Zc_d + Zc_s\right) \\ S[2,2] &= \frac{1}{2}\left(-Wc_d + Wc_s - Xs_s + Xs_d - Zc_d - Zc_s\right) \end{aligned}$$

regardless of the relation of Z to W. This expression could then be modified to fit the hardware as desired.

Appendix B

IFFT and Decimate Hardware Optimizations

Starting with the scaled IDFT formula

$$h_k = \frac{1}{M}\sum_{m=0}^{M-1} H_m e^{j2\pi km/M} \tag{B.1}$$

we wish to solve for samples k = 255, 511, 767 for the DSS algorithm. Selecting k = 256, 512, 768 to minimize the expression, we can write the simplified operation in matrix notation as

$$\begin{bmatrix} h_{256} & h_{512} & h_{768} \end{bmatrix} = \frac{1}{1024}\begin{bmatrix} H_0 & H_1 & H_2 & H_{1020} & H_{1021} & H_{1022} & H_{1023} \end{bmatrix} W \tag{B.2}$$

where W is the IDFT operator

$$W = \begin{bmatrix} 1 & 1 & 1 \\ e^{j\frac{\pi}{2}} & e^{j\pi} & e^{j\frac{3\pi}{2}} \\ e^{j\frac{2\pi}{2}} & e^{j2\pi} & e^{j\frac{6\pi}{2}} \\ e^{j\frac{1020\pi}{2}} & e^{j1020\pi} & e^{j\frac{3(1020)\pi}{2}} \\ e^{j\frac{1021\pi}{2}} & e^{j1021\pi} & e^{j\frac{3(1021)\pi}{2}} \\ e^{j\frac{1022\pi}{2}} & e^{j1022\pi} & e^{j\frac{3(1022)\pi}{2}} \\ e^{j\frac{1023\pi}{2}} & e^{j1023\pi} & e^{j\frac{3(1023)\pi}{2}} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ j & -1 & -j \\ -1 & 1 & -1 \\ 1 & 1 & 1 \\ j & -1 & -j \\ -1 & 1 & -1 \\ -j & -1 & j \end{bmatrix}$$

and the scaler term $\frac{1}{1024}$ is unnecessary since we are only interested in the angle of the samples. Clearly, no multiplies are required in hardware for the IFFT and decimate blocks using this formulation. Simulations suggest that by selecting the three samples offset by one sample, no difference is seen in the MSE performance.

Appendix C

Hardware Expansion Tables

This Appendix includes the MATLAB simulated hardware expansion tables that describe the expected performance under various implementation conditions: real or complex input data samples, decimating into a 2x2, 4x4, 8x8, 16x16, 32x32, or 64x64 complex signal subspace estimating SVD, taking 128, 256, 512, 1024, 2048, 4096, or 8192 data samples, using 100, 400, 1500, or 2000 Mhz ADC sampling clock rate, and in various SNR conditions from -15, -10, -5, 0, 10, 20, 30, and 40 dB. The tables are organized first into increasing rank reviling SVD size from 2x2 all the way to 64x64, then by complex or real input data, and then by number of data samples processed. The input frequency was the same for each statistic produced, which included the average mean, standard deviation, and MSE. The calculation shown in these tables is based the average statistics of three frequencies: one in the center of the bin, one 0.2 b_{sz} offset from the center, and one 0.4 b_{sz} offset from the center of the bin, where b_{sz} is the FFT bin size.

	128 complex samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-147.70	-106.61			-158.73	-118.65
-10			-141.18	-101.61			-151.96	-113.65
-5	25411	142699	-103.11	-96.61	161755	584809	-115.57	-108.65
0	16796	64526	-96.32	-91.61	106717	288073	-109.57	-103.65
10	7095	18720	-85.98	-81.61	30121	80617	-98.57	-93.65
20	3025	6686	-77.34	-71.61	8181	31281	-90.25	-83.65
30	971	2213	-68.14	-61.61	6315	7630	-81.23	-73.65
40	1358	753	-65.88	-51.61	5618	2487	-78.53	-63.65
	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-168.98	-130.13			-172.82	-132.63
-10			-167.18	-125.13			-164.45	-127.63
-5	731707	2235163	-127.40	-120.13	365445	2330735	-127.34	-122.63
0	216676	1056050	-120.52	-115.13	441403	1324221	-122.67	-117.63
10	101293	310109	-110.25	-105.13	198027	349119	-111.93	-107.63
20	25487	92304	-99.67	-95.13	22667	143055	-102.99	-97.63
30	11531	32036	-91.33	-85.13	29456	48207	-95.52	-87.63
40	18300	7069	-88.74	-75.13	20341	14996	-90.38	-77.63

Table C.1: 128 complex data samples decimated into a 2x2 complex rank revealing SVD

	256 complex samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.62	-97.58			-155.99	-109.62
-10			-138.62	-92.58			-146.90	-104.62
-5	11807	42096	-92.90	-87.58	13587	184791	-104.76	-99.62
0	4931	26167	-88.82	-82.58	15663	96698	-99.92	-94.62
10	1452	7988	-78.40	-72.58	4459	32510	-90.01	-84.62
20	728	1843	-65.90	-62.58	2944	9052	-80.13	-74.62
30	522	682	-60.21	-52.58	2327	3096	-73.07	-64.62
40	560	251	-58.81	-42.58	2101	868	-70.68	-54.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.97	-121.10			-171.91	-123.60
-10			-159.47	-116.10			-164.58	-118.60
-5	139132	713893	-117.25	-111.10	176567	878634	-119.19	-113.60
0	26825	360919	-111.70	-106.10	40572	527123	-114.61	-108.60
10	25444	106611	-101.26	-96.10	9303	135181	-102.68	-98.60
20	12094	37159	-91.97	-86.10	14444	48506	-94.77	-88.60
30	8143	12844	-84.65	-76.10	12183	16311	-87.48	-78.60
40	8203	3820	-82.34	-66.10	11524	4803	-85.14	-68.60

Table C.2: 256 complex data samples decimated into a 2x2 complex rank revealing SVD

	512 complex samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-142.24	-88.55			-154.27	-100.59
-10	3127	27759	-88.94	-83.55	5826	120268	-102.48	-95.59
-5	587	14564	-82.50	-78.55	6711	64287	-96.50	-90.59
0	187	8986	-79.32	-73.55	1819	33159	-90.66	-85.59
10	79	2672	-68.59	-63.55	1271	10383	-79.76	-75.59
20	424	826	-60.12	-53.55	1167	3289	-71.60	-65.59
30	258	262	-53.37	-43.55	1147	1120	-66.10	-55.59
40	267	80	-52.50	-33.55	1089	350	-64.64	-45.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-167.00	-112.07			-169.13	-114.57
-10			-150.64	-107.07			-134.65	-109.57
-5	19048	234133	-107.31	-102.07	46973	335266	-110.58	-104.57
0	11962	123025	-101.82	-97.07	18791	159828	-104.47	-99.57
10	5956	42200	-92.51	-87.07	7759	56335	-95.03	-89.57
20	5873	12864	-83.69	-77.07	7281	15952	-85.32	-79.57
30	4630	4039	-77.72	-67.07	5821	5408	-80.33	-69.57
40	3962	1385	-75.90	-57.07	5383	1518	-78.55	-59.57

Table C.3: 512 complex data samples decimated into a 2x2 complex rank revealing SVD

SNR	1024 complex samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-135.93	-79.52			-149.01	-91.56
-10	979	10679	-80.82	-74.52	1096	41195	-92.42	-86.56
-5	298	5429	-74.86	-69.52	955	22631	-87.31	-81.56
0	383	3360	-70.57	-64.52	954	11855	-81.46	-76.56
10	238	1037	-60.91	-54.52	518	3790	-71.80	-66.56
20	143	291	-51.20	-44.52	557	1184	-63.31	-56.56
30	145	98	-47.43	-34.52	512	388	-58.93	-46.56
40	135	29	-46.46	-24.52	537	119	-58.48	-36.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-160.01	-103.04			-161.49	-105.54
-10	22608	153210	-103.71	-98.04	24484	202814	-106.23	-100.54
-5	7326	76631	-97.74	-93.04	2089	115252	-101.12	-95.54
0	4529	46501	-93.27	-88.04	1147	62166	-95.72	-90.54
10	2859	14311	-83.32	-78.04	1821	20250	-86.30	-80.54
20	2407	4666	-75.28	-68.04	3013	5939	-77.55	-70.54
30	2039	1438	-70.54	-58.04	2866	1804	-73.30	-60.54
40	2027	467	-70.06	-48.04	2661	585	-72.58	-50.54

Table C.4: 1024 complex data samples decimated into a 2x2 complex rank revealing SVD

	2048 complex samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-121.44	-70.49	1441	25629	-88.38	-82.53
-10	280	3495	-70.79	-65.49	894	12609	-81.90	-77.53
-5	60	1953	-65.70	-60.49	510	7637	-77.68	-72.53
0	140	1091	-60.75	-55.49	345	4188	-72.45	-67.53
10	91	338	-51.05	-45.49	247	1396	-63.24	-57.53
20	67	108	-43.40	-35.49	265	434	-55.43	-47.53
30	68	34	-40.88	-25.49	269	137	-52.64	-37.53
40	70	10	-40.47	-15.49	279	44	-52.45	-27.53
	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.22	-94.01			-138.88	-96.51
-10	898	51214	-94.27	-89.01	1034	68311	-96.75	-91.51
-5	1984	28316	-88.99	-84.01	3450	39529	-92.11	-86.51
0	2018	15808	-83.95	-79.01	1142	21963	-86.64	-81.51
10	1301	5040	-74.66	-69.01	1313	7041	-77.19	-71.51
20	1110	1622	-67.07	-59.01	1384	2140	-69.57	-61.51
30	1023	492	-64.19	-49.01	1435	699	-67.03	-51.51
40	1007	159	-63.93	-39.01	1563	212	-66.43	-41.51

Table C.5: 2048 complex data samples decimated into a 2x2 complex rank revealing SVD

	4096 complex samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	118	2194	-66.93	-61.46	109	8825	-78.98	-73.50
-10	43	1209	-61.65	-56.46	164	4906	-73.84	-68.50
-5	50	681	-56.67	-51.46	204	2803	-68.96	-63.50
0	36	388	-51.83	-46.46	145	1524	-63.62	-58.50
10	28	123	-42.38	-36.46	155	471	-54.35	-48.50
20	49	39	-36.28	-26.46	183	150	-48.05	-38.50
30	49	12	-34.52	-16.46	200	49	-46.71	-28.50
40	52	4	-34.38	-6.46	208	15	-46.43	-18.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	1514	33750	-90.57	-84.98	1962	44241	-92.97	-87.48
-10	399	18089	-85.25	-79.98	570	24515	-87.92	-82.48
-5	297	9963	-80.04	-74.98	1137	13795	-82.87	-77.48
0	362	5719	-75.26	-69.98	678	7850	-77.92	-72.48
10	479	1846	-66.10	-59.98	610	2465	-68.43	-62.48
20	625	567	-59.49	-49.98	885	760	-62.38	-52.48
30	769	181	-58.02	-39.98	869	234	-60.47	-42.48
40	798	56	-57.90	-29.98	831	76	-60.40	-32.48

Table C.6: 4096 complex data samples decimated into a 2x2 complex rank revealing SVD

	8192 complex samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	19	772	-57.81	-52.43	91	3100	-69.92	-64.47
-10	17	421	-52.50	-47.43	104	1704	-64.66	-59.47
-5	18	242	-47.77	-42.43	74	991	-59.89	-54.47
0	23	134	-42.80	-37.43	78	543	-54.99	-49.47
10	25	42	-33.74	-27.43	118	174	-46.15	-39.47
20	34	14	-29.48	-17.43	137	54	-41.36	-29.47
30	46	4	-28.49	-7.43	178	17	-40.43	-19.47
40	48	1	-28.36	2.57	193	5	-40.40	-9.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	355	11699	-81.39	-75.95	383	15738	-84.00	-78.45
-10	336	6389	-76.12	-70.95	411	8603	-78.74	-73.45
-5	205	3596	-71.19	-65.95	479	4850	-73.76	-68.45
0	427	2036	-66.21	-60.95	666	2662	-68.73	-63.45
10	875	637	-57.42	-50.95	1034	862	-59.92	-53.45
20	1040	201	-52.84	-40.95	1050	259	-55.30	-43.45
30	1184	65	-52.03	-30.95	1093	85	-54.55	-33.45
40	1203	20	-51.90	-20.95	1098	27	-54.37	-23.45

Table C.7: 8192 complex data samples decimated into a 2x2 complex rank revealing SVD

	128 real samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-147.03	-109.61			-157.56	-121.65
-10			-146.21	-104.61			-157.95	-116.65
-5			-137.31	-99.61			-151.93	-111.65
0	28843	95755	-99.91	-94.61	111655	369802	-111.59	-106.65
10	7120	27687	-89.15	-84.61	37294	111930	-101.55	-96.65
20	2101	9198	-79.24	-74.61	5839	34877	-90.75	-86.65
30	1385	3005	-70.76	-64.61	7759	10935	-82.99	-76.65
40	1401	1048	-67.26	-54.61	5694	4475	-79.52	-66.65

	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-168.97	-133.13			-173.06	-135.63
-10			-167.72	-128.13			-171.08	-130.63
-5			-162.61	-123.13			-166.43	-125.63
0	317423	1530590	-123.65	-118.13	731819	1923525	-126.22	-120.63
10	32356	467561	-113.38	-108.13	274405	525609	-115.40	-110.63
20	26035	130931	-102.40	-98.13	40245	150554	-103.69	-100.63
30	20556	47695	-94.29	-88.13	34235	57265	-96.56	-90.63
40	15292	15262	-89.49	-78.13	20779	18823	-91.19	-80.63

Table C.8: 128 real data samples decimated into a 2x2 complex rank revealing SVD

	256 real samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-147.01	-100.58			-157.06	-112.62
-10			-142.58	-95.58			-155.76	-107.62
-5			-128.40	-90.58	66443	259862	-108.47	-102.62
0	9633	33824	-90.90	-85.58	19004	146060	-103.49	-97.62
10	1519	10918	-81.19	-75.58	4951	45269	-93.05	-87.62
20	565	3328	-70.86	-65.58	6279	13851	-84.24	-77.62
30	484	1164	-62.44	-55.58	1842	4799	-74.42	-67.62
40	596	372	-59.81	-45.58	2143	1353	-71.12	-57.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.43	-124.10			-172.26	-126.60
-10			-165.81	-119.10			-170.48	-121.60
-5	154962	1391004	-119.82	-114.10			-159.19	-116.60
0	64884	554777	-114.95	-109.10	103223	722311	-116.95	-111.60
10	54768	145745	-103.91	-99.10	27316	228803	-106.79	-101.60
20	13236	56397	-95.43	-89.10	5849	75059	-97.37	-91.60
30	8579	15440	-85.78	-79.10	12697	21955	-89.13	-81.60
40	8316	5280	-82.92	-69.10	10425	6369	-84.76	-71.60

Table C.9: 256 real data samples decimated into a 2x2 complex rank revealing SVD

	512 real samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.11	-91.55			-157.50	-103.59
-10			-138.77	-86.55			-152.66	-98.59
-5	1709	21697	-86.86	-81.55	7280	90296	-98.86	-93.59
0	3175	13433	-83.00	-76.55	7874	54003	-94.37	-88.59
10	292	3573	-70.93	-66.55	439	14480	-83.28	-78.59
20	332	1101	-61.32	-56.55	940	4807	-73.58	-68.59
30	294	377	-54.99	-46.55	1195	1468	-67.14	-58.59
40	271	126	-52.77	-36.55	1101	487	-64.77	-48.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-168.25	-115.07			-171.21	-117.57
-10			-159.73	-110.07			-165.95	-112.57
-5	84208	363270	-111.57	-105.07	21827	466224	-113.28	-107.57
0	16753	178191	-104.95	-100.07	35024	253364	-108.12	-102.57
10	5418	61441	-95.64	-90.07	6737	73077	-97.58	-92.57
20	4334	18666	-85.63	-80.07	5644	26321	-88.93	-82.57
30	4165	5607	-78.54	-70.07	6284	8355	-81.76	-72.57
40	4004	1659	-76.31	-60.07	5377	2400	-78.71	-62.57

Table C.10: 512 real data samples decimated into a 2x2 complex rank revealing SVD

	1024 real samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.87	-82.52			-155.25	-94.56
-10			-129.54	-77.52			-131.37	-89.56
-5	543	7939	-78.35	-72.52	3216	30038	-89.29	-84.56
0	634	4240	-72.57	-67.52	1241	17395	-84.99	-79.56
10	90	1383	-62.88	-57.52	647	5513	-74.91	-69.56
20	201	409	-54.06	-47.52	546	1552	-64.91	-59.56
30	142	141	-48.05	-37.52	546	547	-59.98	-49.56
40	136	46	-46.57	-27.52	529	164	-58.43	-39.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-167.51	-106.04			-169.45	-108.54
-10			-152.41	-101.04			-142.49	-103.54
-5	5616	116986	-101.50	-96.04	30789	164221	-104.69	-98.54
0	1430	64205	-96.26	-91.04	5657	88287	-98.85	-93.54
10	2569	20349	-86.24	-81.04	2421	28320	-89.08	-83.54
20	2036	6514	-77.12	-71.04	3104	8355	-79.65	-73.54
30	2068	2080	-71.41	-61.04	2635	2839	-73.72	-63.54
40	2049	642	-70.17	-51.04	2610	857	-72.50	-53.54

Table C.11: 1024 real data samples decimated into a 2x2 complex rank revealing SVD

	2048 real samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-136.31	-73.49			-150.33	-85.53
-10	238	5112	-74.40	-68.49	1633	20952	-86.55	-80.53
-5	189	2808	-68.95	-63.49	297	10923	-80.77	-75.53
0	111	1592	-64.12	-58.49	210	6204	-75.81	-70.53
10	58	460	-53.54	-48.49	269	1961	-66.22	-60.53
20	66	145	-45.09	-38.49	269	607	-57.32	-50.53
30	68	49	-41.50	-28.49	283	189	-53.62	-40.53
40	68	16	-40.38	-18.49	277	64	-52.53	-30.53
	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-161.16	-97.01			-164.14	-99.51
-10	5920	77164	-97.99	-92.01	5601	102638	-100.27	-94.51
-5	2264	39544	-91.92	-87.01	4319	53252	-94.59	-89.51
0	1165	22978	-87.23	-82.01	801	31056	-89.93	-84.51
10	918	7448	-77.57	-72.01	1286	9586	-79.72	-74.51
20	1023	2348	-68.98	-62.01	1318	3047	-71.33	-64.51
30	1050	736	-64.81	-52.01	1361	938	-67.17	-54.51
40	1002	214	-63.99	-42.01	1531	291	-66.53	-44.51

Table C.12: 2048 real data samples decimated into a 2x2 complex rank revealing SVD

	4096 real samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-123.50	-64.46			-122.08	-76.50
-10	46	1734	-64.79	-59.46	307	6790	-76.60	-71.50
-5	20	1001	-60.05	-54.46	159	3863	-71.77	-66.50
0	51	543	-54.74	-49.46	210	2238	-67.11	-61.50
10	38	168	-45.01	-39.46	137	683	-57.02	-51.50
20	46	55	-37.90	-29.46	179	212	-49.22	-41.50
30	49	17	-34.72	-19.46	201	68	-46.88	-31.50
40	51	5	-34.39	-9.46	204	22	-46.48	-21.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.69	-87.98			-150.52	-90.48
-10	799	26386	-88.41	-82.98	736	34901	-90.96	-85.48
-5	198	14622	-83.25	-77.98	929	19685	-85.91	-80.48
0	544	7839	-77.99	-72.98	1104	10826	-80.73	-75.48
10	465	2577	-68.55	-62.98	690	3439	-71.16	-65.48
20	604	790	-61.24	-52.98	768	1075	-63.84	-55.48
30	724	251	-58.25	-42.98	879	341	-60.88	-45.48
40	801	82	-57.99	-32.98	842	106	-60.40	-35.48

Table C.13: 4096 real data samples decimated into a 2x2 complex rank revealing SVD

	8192 real samples → 2x2 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	7	1114	-60.91	-55.43	205	4468	-73.05	-67.47
-10	22	601	-55.60	-50.43	75	2390	-67.57	-62.47
-5	20	345	-50.79	-45.43	48	1371	-62.80	-57.47
0	19	192	-45.78	-40.43	73	774	-57.88	-52.47
10	31	60	-36.32	-30.43	112	241	-48.28	-42.47
20	32	19	-30.24	-20.43	130	77	-42.31	-32.47
30	42	6	-28.64	-10.43	176	24	-40.62	-22.47
40	48	2	-28.36	-0.43	193	8	-40.42	-12.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	640	16498	-84.26	-78.95	1216	22408	-87.03	-81.45
-10	203	9232	-79.28	-73.95	314	11977	-81.46	-76.45
-5	474	4979	-74.01	-68.95	447	6646	-76.44	-71.45
0	349	2852	-69.16	-63.95	410	3809	-71.62	-66.45
10	787	902	-59.89	-53.95	879	1213	-62.46	-56.45
20	1003	281	-53.70	-43.95	1045	381	-56.05	-46.45
30	1163	92	-52.07	-33.95	1082	119	-54.59	-36.45
40	1208	28	-51.89	-23.95	1098	38	-54.40	-26.45

Table C.14: 8192 real data samples decimated into a 2x2 complex rank revealing SVD

	128 complex samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-144.15	-106.61			-157.75	-118.65
-10			-142.72	-101.61			-156.08	-113.65
-5			-134.56	-96.61			-127.11	-108.65
0	5019	50790	-93.93	-91.61	49015	184243	-105.33	-103.65
10	4095	15359	-83.86	-81.61	8539	55789	-94.95	-93.65
20	3060	4985	-75.15	-71.61	6594	17864	-86.06	-83.65
30	535	1394	-63.70	-61.61	3309	5956	-77.05	-73.65
40	526	568	-58.12	-51.61	1847	1662	-69.47	-63.65
	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.60	-130.13			-172.36	-132.63
-10			-166.36	-125.13			-169.57	-127.63
-5	381703	1463708	-123.37	-120.13	221728	1776595	-124.92	-122.63
0	183944	664332	-116.48	-115.13	154555	901651	-119.34	-117.63
10	44019	206224	-106.60	-105.13	52757	315297	-109.85	-107.63
20	9035	78700	-97.81	-95.13	35099	88724	-99.37	-97.63
30	4473	23894	-87.53	-85.13	13227	36260	-91.85	-87.63
40	8123	8287	-81.97	-75.13	8389	7410	-81.81	-77.63

Table C.15: 128 complex data samples decimated into a 4x4 complex rank revealing SVD

	256 complex samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.73	-97.58			-156.12	-109.62
-10			-138.78	-92.58			-152.01	-104.62
-5	3987	32197	-90.07	-87.58	45867	130079	-102.83	-99.62
0	4093	16038	-84.79	-82.58	4708	61460	-96.09	-94.62
10	721	4929	-73.40	-72.58	1737	21966	-86.91	-84.62
20	277	1595	-63.89	-62.58	1734	7589	-78.06	-74.62
30	247	515	-55.39	-52.58	865	2313	-68.03	-64.62
40	182	173	-49.27	-42.58	927	717	-62.66	-54.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.80	-121.10			-171.12	-123.60
-10			-163.51	-116.10			-164.51	-118.60
-5	50029	466557	-113.30	-111.10	164433	592664	-115.62	-113.60
0	46390	237376	-106.92	-106.10	54383	362064	-110.84	-108.60
10	8751	75729	-97.43	-96.10	18700	108333	-100.88	-98.60
20	3346	26340	-88.48	-86.10	5919	32589	-90.40	-88.60
30	3806	8341	-78.71	-76.10	4857	11270	-81.48	-78.60
40	3492	2572	-73.92	-66.10	4084	3932	-76.50	-68.60

Table C.16: 256 complex data samples decimated into a 4x4 complex rank revealing SVD

	512 complex samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.44	-88.55			-153.91	-100.59
-10	1825	21848	-87.02	-83.55			-140.15	-95.59
-5	595	10212	-79.80	-78.55	4544	42175	-92.13	-90.59
0	625	5323	-74.87	-73.55	1747	24573	-87.71	-85.59
10	70	1722	-64.63	-63.55	674	7229	-77.27	-75.59
20	170	635	-56.37	-53.55	560	2293	-67.59	-65.59
30	129	191	-47.80	-43.55	488	814	-59.85	-55.59
40	108	60	-43.63	-33.55	423	247	-55.66	-45.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-167.50	-112.07			-169.86	-114.57
-10	10871	279675	-109.02	-107.07			-156.68	-109.57
-5	23018	166076	-104.38	-102.07	11817	229084	-107.21	-104.57
0	11166	101789	-100.58	-97.07	5400	113509	-101.21	-99.57
10	3220	27234	-88.94	-87.07	5104	40293	-92.28	-89.57
20	1871	8471	-78.87	-77.07	3773	11865	-81.89	-79.57
30	1654	2610	-70.33	-67.07	2296	3845	-73.12	-69.57
40	1578	886	-67.04	-57.07	2257	1197	-69.76	-59.57

Table C.17: 512 complex data samples decimated into a 4x4 complex rank revealing SVD

	1024 complex samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-137.22	-79.52			-147.52	-91.56
-10	1313	6879	-76.90	-74.52	2971	27006	-88.71	-86.56
-5	476	3846	-71.77	-69.52	825	15321	-83.79	-81.56
0	169	2110	-66.49	-64.52	1156	8923	-79.10	-76.56
10	63	669	-56.82	-54.52	343	2705	-68.68	-66.56
20	65	225	-47.46	-44.52	299	857	-59.42	-56.56
30	58	68	-40.14	-34.52	229	251	-51.78	-46.56
40	54	22	-37.34	-24.52	224	83	-49.66	-36.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-160.15	-103.04			-161.91	-105.54
-10	3484	103455	-100.35	-98.04	15790	132328	-102.56	-100.54
-5	2491	61941	-95.69	-93.04	8857	78042	-97.90	-95.54
0	2701	32395	-90.10	-88.04	3252	40146	-92.26	-90.54
10	1511	9909	-80.26	-78.04	945	13957	-82.67	-80.54
20	824	3470	-71.11	-68.04	1040	3955	-72.31	-70.54
30	803	1043	-63.26	-58.04	1250	1365	-66.20	-60.54
40	855	315	-61.21	-48.04	1093	418	-63.65	-50.54

Table C.18: 1024 complex data samples decimated into a 4x4 complex rank revealing SVD

	2048 complex samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-123.58	-70.49			-133.74	-82.53
-10	81	2440	-67.81	-65.49	997	9249	-79.22	-77.53
-5	187	1381	-62.80	-60.49	229	5625	-75.00	-72.53
0	27	766	-57.66	-55.49	142	2947	-69.47	-67.53
10	25	240	-47.50	-45.49	67	970	-59.73	-57.53
20	30	79	-38.91	-35.49	135	305	-50.87	-47.53
30	27	24	-32.80	-25.49	97	94	-44.29	-37.53
40	31	8	-31.49	-15.49	117	30	-43.40	-27.53
	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-151.49	-94.01	1253	88570	-98.82	-96.51
-10	2315	33423	-90.33	-89.01	3342	47278	-93.73	-91.51
-5	629	20683	-86.24	-84.01	1329	26471	-88.36	-86.51
0	684	11218	-81.06	-79.01	1283	14491	-83.28	-81.51
10	382	3641	-71.31	-69.01	683	4851	-73.95	-71.51
20	355	1165	-62.05	-59.01	520	1513	-64.49	-61.51
30	410	359	-56.19	-49.01	662	476	-58.93	-51.51
40	362	118	-54.90	-39.01	737	157	-57.51	-41.51

Table C.19: 2048 complex data samples decimated into a 4x4 complex rank revealing SVD

	4096 complex samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	104	1519	-63.64	-61.46	194	6266	-76.01	-73.50
-10	31	880	-58.88	-56.46	81	3489	-70.84	-68.50
-5	23	471	-53.24	-51.46	89	1944	-65.67	-63.50
0	13	263	-48.60	-46.46	85	1070	-60.49	-58.50
10	19	86	-38.91	-36.46	46	340	-50.68	-48.50
20	36	27	-30.36	-26.46	132	108	-42.20	-38.50
30	40	9	-26.15	-16.46	166	33	-37.98	-28.50
40	43	3	-25.31	-6.46	171	11	-37.32	-18.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	1366	23811	-87.47	-84.98	1460	31111	-89.86	-87.48
-10	302	13045	-82.35	-79.98	449	17519	-84.76	-82.48
-5	223	7042	-76.91	-74.98	262	9695	-79.78	-77.48
0	90	4098	-72.25	-69.98	220	5429	-74.65	-72.48
10	210	1264	-62.30	-59.98	262	1769	-65.08	-62.48
20	273	410	-53.63	-49.98	367	553	-56.19	-52.48
30	310	126	-49.60	-39.98	389	173	-52.14	-42.48
40	318	39	-48.80	-29.98	442	55	-51.30	-32.48

Table C.20: 4096 complex data samples decimated into a 4x4 complex rank revealing SVD

	8192 complex samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	32	532	-54.59	-52.43	66	2181	-66.74	-64.47
-10	12	304	-49.59	-47.43	45	1214	-61.68	-59.47
-5	6	170	-44.67	-42.43	38	675	-56.62	-54.47
0	17	97	-39.80	-37.43	67	377	-51.58	-49.47
10	24	31	-30.16	-27.43	101	121	-42.00	-39.47
20	36	10	-22.45	-17.43	137	38	-34.50	-29.47
30	49	3	-19.59	-7.43	197	12	-31.66	-19.47
40	50	1	-19.26	2.57	199	4	-31.29	-9.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	391	8192	-78.23	-75.95	827	10934	-80.86	-78.45
-10	75	4585	-73.31	-70.95	179	6047	-75.61	-73.45
-5	112	2537	-68.13	-65.95	213	3309	-70.49	-68.45
0	491	1430	-63.23	-60.95	427	1877	-65.55	-63.45
10	616	450	-53.46	-50.95	902	597	-55.86	-53.45
20	710	142	-46.03	-40.95	917	192	-48.49	-43.45
30	900	46	-43.17	-30.95	938	59	-45.61	-33.45
40	868	15	-42.76	-20.95	949	20	-45.25	-23.45

Table C.21: 8192 complex data samples decimated into a 4x4 complex rank revealing SVD

	128 real samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-147.30	-109.61			-158.92	-121.65
-10			-144.66	-104.61			-156.70	-116.65
-5			-142.90	-99.61			-152.69	-111.65
0	23294	68141	-96.99	-94.61	81504	267625	-108.83	-106.65
10	4577	18782	-85.62	-84.61	39248	90423	-99.72	-96.65
20	2033	7406	-77.61	-74.61	2661	31777	-89.92	-86.65
30	553	2017	-66.43	-64.61	3666	9390	-80.11	-76.65
40	624	738	-60.93	-54.61	2188	3100	-73.17	-66.65
	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-171.49	-133.13			-171.87	-135.63
-10			-169.11	-128.13			-172.20	-130.63
-5			-163.92	-123.13			-162.63	-125.63
0	93691	1125287	-120.96	-118.13	523835	1288594	-123.14	-120.63
10	73963	310185	-109.98	-108.13	84411	457693	-113.10	-110.63
20	15692	104223	-100.38	-98.13	32139	114692	-101.35	-100.63
30	15700	30223	-90.55	-88.13	8795	37852	-91.59	-90.63
40	8235	10634	-84.26	-78.13	10325	14121	-85.79	-80.63

Table C.22: 128 real data samples decimated into a 4x4 complex rank revealing SVD

	256 real samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-146.82	-100.58			-158.03	-112.62
-10			-142.45	-95.58			-155.92	-107.62
-5	9077	51870	-94.09	-90.58	30869	148812	-103.49	-102.62
0	5397	23798	-87.62	-85.58	31896	93000	-99.63	-97.62
10	908	8822	-78.82	-75.58	7401	31458	-90.23	-87.62
20	567	2537	-68.05	-65.58	1045	9783	-80.06	-77.62
30	255	849	-59.58	-55.58	1201	3130	-70.76	-67.62
40	211	254	-51.59	-45.58	767	917	-62.57	-57.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-168.88	-124.10			-173.28	-126.60
-10			-166.65	-119.10			-169.79	-121.60
-5			-158.56	-114.10			-149.79	-116.60
0	63358	390365	-111.99	-109.10	102188	475606	-112.97	-111.60
10	10153	122054	-101.67	-99.10	41700	155922	-104.31	-101.60
20	2066	36423	-91.22	-89.10	8665	48417	-93.87	-91.60
30	1518	11371	-81.02	-79.10	5663	16848	-85.01	-81.60
40	3570	3522	-75.44	-69.10	4623	4285	-77.94	-71.60

Table C.23: 256 real data samples decimated into a 4x4 complex rank revealing SVD

	512 real samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.54	-91.55			-157.20	-103.59
-10			-139.32	-86.55			-151.17	-98.59
-5	2259	13653	-82.69	-81.55	5543	62986	-95.81	-93.59
0	1045	9309	-79.66	-76.55	3844	38204	-92.14	-88.59
10	348	2601	-68.24	-66.55	880	10447	-80.16	-78.59
20	130	891	-59.16	-56.55	251	3218	-70.43	-68.59
30	108	253	-49.07	-46.55	411	1089	-61.72	-58.59
40	118	85	-44.99	-36.55	465	337	-56.96	-48.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-168.74	-115.07			-172.14	-117.57
-10			-161.58	-110.07			-165.76	-112.57
-5	36809	248299	-107.97	-105.07	54595	310626	-110.03	-107.57
0	11532	126769	-101.91	-100.07	21432	169314	-104.65	-102.57
10	6687	38674	-91.84	-90.07	4117	50322	-93.98	-92.57
20	2468	12380	-82.52	-80.07	4364	18444	-85.76	-82.57
30	1495	3851	-72.73	-70.07	3885	5271	-77.04	-72.57
40	1565	1167	-67.17	-60.07	2220	1701	-70.81	-62.57

Table C.24: 512 real data samples decimated into a 4x4 complex rank revealing SVD

	1024 real samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.83	-82.52			-155.42	-94.56
-10			-126.30	-77.52			-139.65	-89.56
-5	527	5292	-74.16	-72.52	499	20738	-86.09	-84.56
0	137	3077	-69.65	-67.52	263	11994	-81.53	-79.56
10	176	967	-60.06	-57.52	510	3648	-71.53	-69.56
20	75	293	-49.91	-47.52	220	1208	-61.76	-59.56
30	49	91	-40.95	-37.52	200	391	-53.56	-49.56
40	54	31	-37.81	-27.52	227	126	-50.15	-39.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-166.48	-106.04			-169.73	-108.54
-10			-148.99	-101.04			-117.25	-103.54
-5	11476	80031	-98.06	-96.04	7262	104655	-100.45	-98.54
0	4498	46057	-93.40	-91.04	3627	58429	-95.40	-93.54
10	1206	14373	-83.23	-81.04	2939	19440	-85.48	-83.54
20	806	4408	-73.06	-71.04	613	6364	-76.07	-73.54
30	699	1495	-64.71	-61.04	1120	2017	-67.73	-63.54
40	817	483	-61.50	-51.04	1090	597	-63.80	-53.54

Table C.25: 1024 real data samples decimated into a 4x4 complex rank revealing SVD

SNR (dB)	2048 real samples → 4x4 complex SVD							
	Sample Rate							
	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-137.05	-73.49			-149.44	-85.53
-10	272	3698	-71.54	-68.49	1318	14187	-83.05	-80.53
-5	57	2104	-66.30	-63.49	434	7800	-77.78	-75.53
0	20	1087	-60.96	-58.49	168	4163	-72.49	-70.53
10	18	333	-50.42	-48.49	199	1366	-62.63	-60.53
20	30	110	-41.22	-38.49	125	427	-53.16	-50.53
30	29	34	-34.27	-28.49	97	139	-45.75	-40.53
40	29	11	-31.64	-18.49	116	41	-43.57	-30.53

SNR (dB)	Sample Rate							
	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-160.19	-97.01			-163.98	-99.51
-10	4681	51254	-94.15	-92.01	7288	72223	-97.25	-94.51
-5	1184	28673	-89.36	-87.01	2181	38868	-91.77	-89.51
0	244	15941	-83.80	-82.01	491	21917	-86.87	-84.51
10	779	5086	-74.19	-72.01	476	6997	-76.93	-74.51
20	371	1651	-64.73	-62.01	605	2144	-67.17	-64.51
30	434	524	-57.78	-52.01	593	672	-59.54	-54.51
40	381	155	-55.06	-42.01	705	220	-57.42	-44.51

Table C.26: 2048 real data samples decimated into a 4x4 complex rank revealing SVD

	4096 real samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-122.39	-64.46			-120.52	-76.50
-10	55	1220	-61.85	-59.46	114	4931	-73.91	-71.50
-5	15	659	-56.35	-54.46	61	2659	-68.48	-66.50
0	15	379	-51.60	-49.46	32	1518	-63.67	-61.50
10	17	119	-41.67	-39.46	58	493	-54.05	-51.50
20	32	38	-32.47	-29.46	130	152	-44.64	-41.50
30	40	12	-26.60	-19.46	160	49	-39.00	-31.50
40	43	4	-25.37	-9.46	171	16	-37.37	-21.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-147.16	-87.98			-152.09	-90.48
-10	802	18580	-85.38	-82.98	1369	23781	-87.47	-85.48
-5	152	10185	-80.17	-77.98	794	13617	-82.66	-80.48
0	46	5637	-75.08	-72.98	231	7638	-77.74	-75.48
10	218	1783	-65.17	-62.98	283	2454	-67.86	-65.48
20	273	584	-56.21	-52.98	330	746	-58.55	-55.48
30	308	180	-50.42	-42.98	415	239	-52.93	-45.48
40	313	56	-48.87	-32.98	431	80	-51.30	-35.48

Table C.27: 4096 real data samples decimated into a 4x4 complex rank revealing SVD

	8192 real samples → 4x4 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	29	791	-57.99	-55.43	133	3060	-69.73	-67.47
-10	13	431	-52.61	-50.43	58	1728	-64.76	-62.47
-5	5	235	-47.37	-45.43	21	989	-59.84	-57.47
0	13	132	-42.45	-40.43	40	539	-54.61	-52.47
10	20	43	-32.81	-30.43	81	166	-44.69	-42.47
20	31	14	-24.15	-20.43	118	53	-36.10	-32.47
30	46	4	-19.99	-10.43	189	17	-32.03	-22.47
40	50	1	-19.23	-0.43	199	6	-31.32	-12.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	1030	11327	-81.13	-78.95	1339	15579	-83.86	-81.45
-10	88	6325	-76.05	-73.95	346	8557	-78.63	-76.45
-5	70	3657	-71.24	-68.95	198	4797	-73.59	-71.45
0	275	2012	-66.12	-63.95	426	2704	-68.71	-66.45
10	597	616	-56.00	-53.95	854	862	-58.79	-56.45
20	681	205	-47.86	-43.95	906	269	-50.32	-46.45
30	862	65	-43.58	-33.95	933	83	-46.08	-36.45
40	902	20	-42.78	-23.95	943	27	-45.25	-26.45

Table C.28: 8192 real data samples decimated into a 4x4 complex rank revealing SVD

	128 complex samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-147.19	-106.61			-157.75	-118.65
-10			-143.25	-101.61			-154.24	-113.65
-5	20017	78242	-98.15	-96.61	90719	462280	-113.20	-108.65
0	11134	48416	-93.71	-91.61	33897	181185	-105.10	-103.65
10	6018	9567	-81.28	-81.61	14735	63139	-96.17	-93.65
20	826	4683	-73.69	-71.61	7380	20047	-86.38	-83.65
30	548	1875	-65.63	-61.61	737	5275	-74.37	-73.65
40	226	400	-53.14	-51.61	851	1985	-66.57	-63.65
	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-170.84	-130.13			-172.97	-132.63
-10			-166.41	-125.13			-168.49	-127.63
-5			-139.12	-120.13			-163.69	-122.63
0	96588	656935	-116.29	-115.13	221952	783129	-118.51	-117.63
10	17735	204176	-105.99	-105.13	54213	281862	-108.95	-107.63
20	13343	47800	-93.88	-95.13	21099	93499	-99.41	-97.63
30	4707	23248	-87.42	-85.13	11483	29000	-89.96	-87.63
40	3561	6103	-77.22	-75.13	4747	9277	-80.13	-77.63

Table C.29: 128 complex data samples decimated into a 8x8 complex rank revealing SVD

	256 complex samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-146.88	-97.58			-155.23	-109.62
-10			-138.58	-92.58			-147.67	-104.62
-5	2721	31621	-90.03	-87.58	12048	132178	-102.58	-99.62
0	2078	17696	-85.36	-82.58	8093	67807	-96.02	-94.62
10	944	5249	-74.63	-72.58	3367	21536	-86.12	-84.62
20	158	1562	-63.83	-62.58	446	6170	-75.78	-74.62
30	118	438	-53.45	-52.58	270	2141	-66.78	-64.62
40	68	161	-45.11	-42.58	239	726	-57.64	-54.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-168.46	-121.10			-171.18	-123.60
-10			-161.48	-116.10			-164.63	-118.60
-5	155008	436308	-113.10	-111.10	40583	593046	-115.15	-113.60
0	46598	233038	-107.54	-106.10	42935	325391	-110.32	-108.60
10	7287	84190	-98.76	-96.10	20193	101482	-100.23	-98.60
20	4923	22761	-87.67	-86.10	6796	31298	-90.08	-88.60
30	1060	8447	-78.35	-76.10	1183	10153	-79.98	-78.60
40	610	2548	-68.17	-66.10	799	3407	-70.84	-68.60

Table C.30: 256 complex data samples decimated into a 8x8 complex rank revealing SVD

	512 complex samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-142.81	-88.55			-155.36	-100.59
-10			-120.83	-83.55			-136.19	-95.59
-5	2157	9664	-79.84	-78.55	5393	39691	-92.62	-90.59
0	618	5751	-75.24	-73.55	2742	22495	-87.22	-85.59
10	280	1853	-65.40	-63.55	210	7791	-77.91	-75.59
20	65	592	-55.45	-53.55	341	2441	-68.20	-65.59
30	40	181	-45.36	-43.55	116	665	-56.72	-55.59
40	34	60	-37.34	-33.55	131	223	-48.32	-45.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-166.20	-112.07			-167.19	-114.57
-10			-152.29	-107.07			-145.35	-109.57
-5	6220	145164	-103.32	-102.07	5881	219757	-106.48	-104.57
0	9981	77375	-98.03	-97.07	24232	129213	-102.51	-99.57
10	1703	25739	-88.39	-87.07	7199	39029	-92.09	-89.57
20	981	8779	-79.13	-77.07	637	11569	-81.56	-79.57
30	990	2537	-68.82	-67.07	812	3430	-71.07	-69.57
40	481	854	-60.21	-57.07	648	1144	-62.94	-59.57

Table C.31: 512 complex data samples decimated into a 8x8 complex rank revealing SVD

	1024 complex samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-137.41	-79.52			-148.41	-91.56
-10	758	6872	-76.77	-74.52	487	26258	-88.20	-86.56
-5	401	3656	-71.16	-69.52	618	14712	-83.31	-81.56
0	268	2062	-66.18	-64.52	353	7928	-77.82	-76.56
10	67	646	-56.01	-54.52	94	2632	-68.37	-66.56
20	24	213	-46.60	-44.52	44	728	-57.29	-56.56
30	18	65	-36.89	-34.52	74	249	-48.66	-46.56
40	17	21	-29.69	-24.52	71	79	-41.50	-36.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-156.37	-103.04			-163.41	-105.54
-10	13563	96807	-99.79	-98.04	34521	135934	-102.74	-100.54
-5	4316	52747	-94.44	-93.04	7396	76859	-97.51	-95.54
0	3268	29790	-89.32	-88.04	3577	38966	-91.79	-90.54
10	512	10079	-80.19	-78.04	811	12376	-81.64	-80.54
20	457	3170	-70.06	-68.04	619	4169	-72.49	-70.54
30	217	965	-59.80	-58.04	413	1264	-62.57	-60.54
40	272	327	-53.07	-48.04	320	400	-54.98	-50.54

Table C.32: 1024 complex data samples decimated into a 8x8 complex rank revealing SVD

	2048 complex samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	418	4316	-72.72	-70.49			-113.79	-82.53
-10	201	2339	-67.29	-65.49	586	9578	-79.72	-77.53
-5	21	1274	-62.03	-60.49	220	5291	-74.37	-72.53
0	22	726	-57.30	-55.49	207	2757	-68.98	-67.53
10	6	226	-47.11	-45.49	40	867	-58.97	-57.53
20	8	76	-37.78	-35.49	51	289	-49.31	-47.53
30	11	23	-27.94	-25.49	37	95	-40.28	-37.53
40	16	7	-21.85	-15.49	59	30	-33.99	-27.53

	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-142.98	-94.01			-151.64	-96.51
-10	3331	34652	-90.92	-89.01	2929	46018	-93.25	-91.51
-5	538	18975	-85.71	-84.01	1846	24640	-87.87	-86.51
0	504	10536	-80.35	-79.01	261	13757	-82.82	-81.51
10	116	3484	-70.74	-69.01	227	4452	-73.03	-71.51
20	143	1072	-60.54	-59.01	187	1383	-62.82	-61.51
30	131	342	-51.28	-49.01	235	451	-53.99	-51.51
40	222	107	-45.76	-39.01	326	151	-48.31	-41.51

Table C.33: 2048 complex data samples decimated into a 8x8 complex rank revealing SVD

	4096 complex samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	92	1429	-63.17	-61.46	577	5931	-75.60	-73.50
-10	42	816	-58.23	-56.46	98	3284	-70.28	-68.50
-5	13	443	-52.89	-51.46	91	1825	-65.15	-63.50
0	5	253	-48.08	-46.46	48	993	-59.99	-58.50
10	6	79	-37.91	-36.46	52	325	-50.28	-48.50
20	27	26	-28.41	-26.46	104	102	-40.33	-38.50
30	31	8	-19.62	-16.46	127	31	-31.54	-28.50
40	30	2	-15.19	-6.46	126	10	-27.25	-18.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	591	22667	-87.13	-84.98	1343	29686	-89.38	-87.48
-10	431	12212	-81.94	-79.98	290	16011	-84.04	-82.48
-5	332	6582	-76.43	-74.98	123	9049	-79.09	-77.48
0	74	3774	-71.45	-69.98	258	5012	-74.05	-72.48
10	167	1227	-61.86	-59.98	74	1589	-64.06	-62.48
20	254	373	-51.81	-49.98	362	492	-54.13	-52.48
30	276	119	-43.09	-39.98	362	160	-45.51	-42.48
40	289	39	-38.99	-29.98	373	51	-41.31	-32.48

Table C.34: 4096 complex data samples decimated into a 8x8 complex rank revealing SVD

	8192 complex samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	10	515	-54.32	-52.43	46	2043	-66.12	-64.47
-10	15	288	-49.13	-47.43	16	1151	-61.26	-59.47
-5	11	159	-44.02	-42.43	52	632	-56.06	-54.47
0	15	89	-39.04	-37.43	65	363	-51.21	-49.47
10	28	29	-29.13	-27.43	113	112	-41.14	-39.47
20	39	9	-19.19	-17.43	154	36	-31.60	-29.47
30	51	3	-11.77	-7.43	203	11	-23.69	-19.47
40	50	1	-8.69	2.57	201	4	-20.82	-9.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	227	7576	-77.61	-75.95	267	10373	-80.28	-78.45
-10	136	4313	-72.71	-70.95	251	5610	-75.02	-73.45
-5	114	2445	-67.76	-65.95	234	3175	-70.03	-68.45
0	456	1349	-62.60	-60.95	560	1783	-65.02	-63.45
10	670	419	-52.49	-50.95	1003	560	-55.03	-53.45
20	969	133	-42.70	-40.95	907	178	-45.32	-43.45
30	1084	43	-35.31	-30.95	917	56	-37.46	-33.45
40	1132	14	-32.33	-20.95	917	19	-34.83	-23.45

Table C.35: 8192 complex data samples decimated into a 8x8 complex rank revealing SVD

	128 real samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-147.42	-109.61			-158.85	-121.65
-10			-145.89	-104.61			-156.42	-116.65
-5			-140.99	-99.61			-151.86	-111.65
0	29292	54007	-95.62	-94.61	73209	263688	-108.73	-106.65
10	7319	19820	-86.67	-84.61	27297	70806	-97.81	-96.65
20	1277	5544	-75.11	-74.61	10001	21556	-87.90	-86.65
30	387	2010	-65.93	-64.61	2148	7474	-78.12	-76.65
40	187	711	-57.47	-54.61	435	2665	-68.50	-66.65
	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-170.81	-133.13			-172.40	-135.63
-10			-169.69	-128.13			-171.57	-130.63
-5			-162.33	-123.13			-166.21	-125.63
0	214793	846857	-118.56	-118.13	653120	1636056	-124.99	-120.63
10	75949	327421	-110.49	-108.13	66576	312715	-109.90	-110.63
20	32884	106714	-100.95	-98.13	32752	147141	-103.31	-100.63
30	4657	33755	-90.50	-88.13	7600	41407	-92.43	-90.63
40	3525	9794	-80.58	-78.13	4773	15168	-84.15	-80.63

Table C.36: 128 real data samples decimated into a 8x8 complex rank revealing SVD

	256 real samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.96	-100.58			-158.48	-112.62
-10			-142.79	-95.58			-155.96	-107.62
-5	11118	42939	-93.14	-90.58	56131	211439	-106.91	-102.62
0	1202	20572	-86.11	-85.58	17887	97813	-100.20	-97.62
10	1726	7993	-78.12	-75.58	2877	29161	-89.10	-87.62
20	222	2186	-66.99	-65.58	1573	7607	-78.15	-77.62
30	62	786	-57.70	-55.58	275	2642	-68.23	-67.62
40	26	195	-45.93	-45.58	237	897	-59.51	-57.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.71	-124.10			-172.20	-126.60
-10			-167.28	-119.10			-168.36	-121.60
-5			-154.68	-114.10			-155.35	-116.60
0	61468	341171	-110.66	-109.10	139580	436188	-113.04	-111.60
10	16226	100649	-100.22	-99.10	9353	160129	-104.04	-101.60
20	6427	34227	-90.76	-89.10	11444	42064	-93.09	-91.60
30	1044	12644	-82.08	-79.10	3751	16763	-84.95	-81.60
40	1163	3745	-72.47	-69.10	1129	4817	-73.87	-71.60

Table C.37: 256 real data samples decimated into a 8x8 complex rank revealing SVD

	512 real samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.32	-91.55			-157.47	-103.59
-10			-139.41	-86.55			-149.57	-98.59
-5	1438	14424	-83.54	-81.55	12700	59765	-95.49	-93.59
0	240	8097	-78.10	-76.55	2459	33489	-90.44	-88.59
10	382	2480	-67.84	-66.55	1131	10899	-80.83	-78.59
20	78	809	-58.17	-56.55	369	2939	-69.67	-68.59
30	31	267	-48.67	-46.55	103	1043	-60.51	-58.59
40	23	81	-38.40	-36.55	81	323	-50.56	-48.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.39	-115.07			-170.39	-117.57
-10			-161.78	-110.07			-163.34	-112.57
-5	12060	243748	-107.60	-105.07	23039	289174	-109.21	-107.57
0	19599	118138	-101.85	-100.07	5052	174095	-104.91	-102.57
10	2519	40083	-91.82	-90.07	8323	51788	-94.41	-92.57
20	1165	10386	-80.38	-80.07	1763	16068	-84.01	-82.57
30	541	4039	-72.14	-70.07	892	5096	-74.42	-72.57
40	397	1252	-62.41	-60.07	381	1479	-63.76	-62.57

Table C.38: 512 real data samples decimated into a 8x8 complex rank revealing SVD

	1024 real samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.24	-82.52			-154.97	-94.56
-10			-129.60	-77.52			-117.35	-89.56
-5	391	5108	-74.17	-72.52	425	21086	-86.60	-84.56
0	124	3018	-69.53	-67.52	923	11710	-81.30	-79.56
10	67	980	-59.83	-57.52	121	3289	-70.27	-69.56
20	36	292	-49.23	-47.52	109	1197	-61.81	-59.56
30	9	85	-38.56	-37.52	76	369	-51.47	-49.56
40	12	29	-30.09	-27.52	63	113	-42.37	-39.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-167.14	-106.04			-169.59	-108.54
-10			-150.38	-101.04			-153.31	-103.54
-5	7871	74608	-97.40	-96.04	7405	101246	-100.28	-98.54
0	4568	41827	-92.23	-91.04	6441	55098	-95.02	-93.54
10	1027	14220	-83.09	-81.04	1511	18101	-85.15	-83.54
20	304	4405	-72.84	-71.04	272	5401	-74.83	-73.54
30	161	1477	-63.28	-61.04	357	1817	-65.05	-63.54
40	222	422	-53.78	-51.04	301	578	-56.70	-53.54

Table C.39: 1024 real data samples decimated into a 8x8 complex rank revealing SVD

	2048 real samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-138.55	-73.49			-150.04	-85.53
-10	135	3232	-70.20	-68.49	469	13221	-82.28	-80.53
-5	261	1867	-65.66	-63.49	635	7413	-77.50	-75.53
0	53	977	-59.53	-58.49	297	4202	-72.36	-70.53
10	19	316	-50.14	-48.49	85	1257	-62.31	-60.53
20	9	103	-40.35	-38.49	34	407	-52.06	-50.53
30	10	32	-30.31	-28.49	37	125	-42.30	-40.53
40	14	10	-22.97	-18.49	47	41	-34.78	-30.53
	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-161.40	-97.01			-163.70	-99.51
-10	3521	48813	-93.90	-92.01	8248	70664	-97.06	-94.51
-5	1542	27900	-88.90	-87.01	1862	36672	-91.39	-89.51
0	1547	15372	-83.91	-82.01	616	20440	-86.22	-84.51
10	397	4877	-73.59	-72.01	268	6501	-76.18	-74.51
20	143	1568	-64.00	-62.01	172	1946	-65.83	-64.51
30	98	477	-53.72	-52.01	203	648	-56.60	-54.51
40	209	155	-46.60	-42.01	315	198	-48.45	-44.51

Table C.40: 2048 real data samples decimated into a 8x8 complex rank revealing SVD

	4096 real samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-125.90	-64.46			-123.73	-76.50
-10	39	1145	-61.14	-59.46	205	4545	-73.18	-71.50
-5	22	646	-56.15	-54.46	54	2606	-68.30	-66.50
0	18	357	-51.08	-49.46	63	1481	-63.32	-61.50
10	4	113	-41.18	-39.46	24	452	-53.05	-51.50
20	24	36	-31.16	-29.46	90	136	-42.81	-41.50
30	31	11	-21.70	-19.46	122	46	-33.99	-31.50
40	31	4	-15.62	-9.46	123	14	-27.72	-21.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-146.61	-87.98			-149.62	-90.48
-10	1590	17856	-85.18	-82.98	1012	22750	-87.10	-85.48
-5	412	9377	-79.38	-77.98	491	12444	-81.87	-80.48
0	105	5453	-74.80	-72.98	203	7187	-77.15	-75.48
10	66	1691	-64.66	-62.98	71	2253	-67.05	-65.48
20	233	524	-54.44	-52.98	298	720	-57.22	-55.48
30	268	172	-45.28	-42.98	357	216	-47.47	-45.48
40	297	55	-39.37	-32.98	373	71	-41.17	-35.48

Table C.41: 4096 real data samples decimated into a 8x8 complex rank revealing SVD

	8192 real samples → 8x8 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	22	739	-57.44	-55.43	93	2908	-69.28	-67.47
-10	18	414	-52.39	-50.43	34	1619	-64.09	-62.47
-5	6	221	-46.92	-45.43	58	908	-59.14	-57.47
0	12	130	-42.21	-40.43	45	504	-54.02	-52.47
10	26	41	-32.15	-30.43	108	163	-44.23	-42.47
20	35	13	-22.34	-20.43	128	51	-34.15	-32.47
30	49	4	-13.63	-10.43	199	16	-25.66	-22.47
40	50	1	-9.02	-0.43	201	5	-21.10	-12.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	600	11306	-81.07	-78.95	533	14862	-83.46	-81.45
-10	139	6041	-75.63	-73.95	275	8060	-78.05	-76.45
-5	32	3402	-70.73	-68.95	70	4605	-73.28	-71.45
0	267	1865	-65.40	-63.95	347	2546	-68.06	-66.45
10	630	601	-55.53	-53.95	1019	811	-58.18	-56.45
20	849	188	-45.71	-43.95	944	256	-48.28	-46.45
30	1049	60	-37.08	-33.95	911	79	-39.47	-36.45
40	1132	19	-32.43	-23.95	917	25	-35.22	-26.45

Table C.42: 8192 real data samples decimated into a 8x8 complex rank revealing SVD

	128 complex samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-146.04	-106.61			-157.81	-118.65
-10			-142.65	-101.61			-153.67	-113.65
-5	15733	106009	-100.64	-96.61	93053	397274	-111.97	-108.65
0	10135	43663	-92.76	-91.61	27269	192713	-105.50	-103.65
10	4551	16153	-84.40	-81.61	10569	47531	-93.68	-93.65
20	558	4475	-72.92	-71.61	3573	17113	-84.68	-83.65
30	887	1264	-64.23	-61.61	4257	5021	-77.04	-73.65
40	944	386	-62.17	-51.61	3593	1463	-73.50	-63.65
	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.84	-130.13			-171.78	-132.63
-10			-166.80	-125.13			-169.94	-127.63
-5			-158.99	-120.13	289216	1923939	-125.91	-122.63
0	111121	544727	-114.77	-115.13	140155	945528	-119.62	-117.63
10	52379	245858	-107.82	-105.13	66731	310661	-109.98	-107.63
20	14939	72150	-97.44	-95.13	27301	78117	-98.55	-97.63
30	12920	21453	-89.58	-85.13	17595	24551	-90.42	-87.63
40	15067	6379	-85.98	-75.13	20784	10086	-88.99	-77.63

Table C.43: 128 complex data samples decimated into a 16x16 complex rank revealing SVD

	256 complex samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.48	-97.58			-157.72	-109.62
-10			-116.37	-92.58			-149.62	-104.62
-5	3015	30356	-89.08	-87.58	18944	110277	-100.92	-99.62
0	2405	16772	-84.79	-82.58	7768	62511	-95.27	-94.62
10	756	5329	-74.60	-72.58	4493	18658	-85.82	-84.62
20	282	1363	-63.20	-62.58	1125	6172	-76.38	-74.62
30	89	525	-54.64	-52.58	124	2096	-66.29	-64.62
40	31	141	-42.93	-42.58	180	740	-57.70	-54.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.15	-121.10			-171.29	-123.60
-10			-151.89	-116.10			-163.17	-118.60
-5	67430	515331	-114.20	-111.10	71327	571744	-115.31	-113.60
0	37842	253534	-107.69	-106.10	111319	301069	-110.21	-108.60
10	9844	94049	-99.56	-96.10	10783	102699	-99.74	-98.60
20	6383	21765	-87.39	-86.10	9687	31018	-90.28	-88.60
30	676	8121	-78.32	-76.10	2196	9875	-80.02	-78.60
40	532	2496	-67.52	-66.10	252	3018	-69.19	-68.60

Table C.44: 256 complex data samples decimated into a 16x16 complex rank revealing SVD

	512 complex samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.37	-88.55			-155.98	-100.59
-10			-128.66	-83.55	8245	88644	-99.55	-95.59
-5	1411	9834	-79.73	-78.55	5558	37448	-91.56	-90.59
0	395	5901	-75.51	-73.55	3910	21041	-86.70	-85.59
10	128	1842	-65.45	-63.55	341	7064	-76.81	-75.59
20	52	535	-54.45	-53.55	72	2123	-66.51	-65.59
30	23	177	-45.18	-43.55	100	719	-56.79	-55.59
40	8	55	-35.18	-33.55	18	211	-46.78	-45.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-165.76	-112.07			-169.01	-114.57
-10	33340	318956	-110.23	-107.07	24656	382333	-111.64	-109.57
-5	21940	165371	-104.40	-102.07	5840	198534	-106.24	-104.57
0	10450	86762	-99.05	-97.07	17703	116030	-101.51	-99.57
10	2720	23816	-87.58	-87.07	713	34298	-90.72	-89.57
20	609	8039	-77.93	-77.07	1412	11418	-81.11	-79.57
30	444	2342	-67.38	-67.07	277	3477	-70.85	-69.57
40	107	841	-58.32	-57.07	87	1144	-61.26	-59.57

Table C.45: 512 complex data samples decimated into a 16x16 complex rank revealing SVD

	1024 complex samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-133.50	-79.52			-148.47	-91.56
-10	694	6426	-76.20	-74.52	770	26718	-88.52	-86.56
-5	213	3723	-71.30	-69.52	875	14591	-83.12	-81.56
0	127	2076	-66.49	-64.52	679	7682	-77.79	-76.56
10	33	636	-56.09	-54.52	243	2575	-68.26	-66.56
20	23	212	-46.64	-44.52	50	764	-57.63	-56.56
30	11	61	-35.76	-34.52	44	252	-48.23	-46.56
40	3	20	-26.33	-24.52	11	73	-37.22	-36.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-159.51	-103.04			-163.47	-105.54
-10	9484	98591	-99.90	-98.04	12243	133220	-102.47	-100.54
-5	6707	52840	-94.63	-93.04	3372	71568	-97.17	-95.54
0	1252	28964	-88.95	-88.04	3241	38397	-91.82	-90.54
10	260	9117	-79.18	-78.04	1242	12212	-81.55	-80.54
20	379	2879	-69.04	-68.04	187	3723	-71.18	-70.54
30	60	960	-59.52	-58.04	157	1270	-62.16	-60.54
40	84	321	-50.33	-48.04	98	382	-52.09	-50.54

Table C.46: 1024 complex data samples decimated into a 16x16 complex rank revealing SVD

	2048 complex samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-124.00	-70.49			-123.95	-82.53
-10	56	2333	-67.20	-65.49	1242	8697	-78.88	-77.53
-5	40	1209	-61.79	-60.49	226	5129	-74.12	-72.53
0	18	722	-57.09	-55.49	68	2929	-69.15	-67.53
10	9	220	-46.68	-45.49	70	847	-58.64	-57.53
20	3	74	-37.42	-35.49	4	276	-48.84	-47.53
30	6	23	-26.93	-25.49	19	86	-38.87	-37.53
40	13	7	-17.00	-15.49	48	28	-29.14	-27.53
	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	4849	70447	-96.99	-94.01	8446	84712	-98.65	-96.51
-10	2320	33952	-90.59	-89.01	2042	44157	-92.85	-91.51
-5	1600	18269	-85.27	-84.01	940	24998	-88.05	-86.51
0	404	10731	-80.57	-79.01	872	13731	-82.67	-81.51
10	39	3255	-70.23	-69.01	118	4166	-72.34	-71.51
20	57	1063	-60.53	-59.01	67	1417	-63.27	-61.51
30	105	324	-50.19	-49.01	155	423	-52.63	-51.51
40	145	107	-40.92	-39.01	262	141	-43.41	-41.51

Table C.47: 2048 complex data samples decimated into a 16x16 complex rank revealing SVD

	4096 complex samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	13	1428	-63.16	-61.46	373	5773	-75.26	-73.50
-10	11	795	-57.97	-56.46	128	3168	-70.07	-68.50
-5	20	440	-52.87	-51.46	76	1787	-65.04	-63.50
0	12	248	-47.92	-46.46	22	994	-60.06	-58.50
10	4	80	-38.18	-36.46	18	326	-50.38	-48.50
20	23	25	-28.09	-26.46	102	97	-39.60	-38.50
30	29	8	-17.93	-16.46	108	32	-30.28	-28.50
40	25	3	-8.52	-6.46	100	10	-20.46	-18.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	985	21094	-86.51	-84.98	885	29489	-89.32	-87.48
-10	341	11960	-81.50	-79.98	463	15506	-83.93	-82.48
-5	230	6607	-76.38	-74.98	500	8842	-78.92	-77.48
0	55	3649	-71.27	-69.98	118	4900	-73.67	-72.48
10	52	1168	-61.35	-59.98	154	1575	-63.98	-62.48
20	236	372	-51.42	-49.98	341	504	-53.90	-52.48
30	292	122	-41.79	-39.98	372	153	-43.73	-42.48
40	311	37	-32.12	-29.98	377	50	-34.71	-32.48

Table C.48: 4096 complex data samples decimated into a 16x16 complex rank revealing SVD

SNR (dB)	8192 complex samples → 16x16 complex SVD							
	Sample Rate							
	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	18	511	-54.15	-52.43	67	1994	-66.03	-64.47
-10	16	272	-48.66	-47.43	11	1118	-61.04	-59.47
-5	7	157	-43.89	-42.43	25	617	-55.80	-54.47
0	17	87	-38.78	-37.43	61	351	-50.86	-49.47
10	28	27	-28.82	-27.43	128	111	-40.82	-39.47
20	40	9	-18.97	-17.43	161	36	-31.22	-29.47
30	53	3	-8.89	-7.43	209	11	-21.05	-19.47
40	51	1	-0.31	2.57	203	4	-12.30	-9.47
	Sample Rate							
SNR (dB)	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	184	7525	-77.49	-75.95	282	9997	-79.89	-78.45
-10	50	4221	-72.48	-70.95	192	5618	-74.96	-73.45
-5	118	2348	-67.43	-65.95	207	3046	-69.63	-68.45
0	433	1332	-62.50	-60.95	571	1781	-64.92	-63.45
10	660	417	-52.41	-50.95	1067	553	-54.86	-53.45
20	980	133	-42.52	-40.95	928	171	-44.63	-43.45
30	1124	42	-32.51	-30.95	922	56	-35.13	-33.45
40	1137	13	-23.78	-20.95	917	18	-26.20	-23.45

Table C.49: 8192 complex data samples decimated into a 16x16 complex rank revealing SVD

	128 real samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-146.99	-109.61			-160.14	-121.65
-10			-145.86	-104.61			-157.20	-116.65
-5			-136.45	-99.61			-152.93	-111.65
0	19617	61762	-96.32	-94.61	21189	330905	-110.14	-106.65
10	6983	17372	-85.40	-84.61	16465	80872	-98.28	-96.65
20	2070	6911	-77.00	-74.61	10070	20957	-87.30	-86.65
30	1676	1622	-68.57	-64.61	5128	8263	-81.19	-76.65
40	1263	641	-66.40	-54.61	5668	2041	-78.88	-66.65
	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-170.06	-133.13			-172.06	-135.63
-10			-166.92	-128.13			-172.12	-130.63
-5			-162.34	-123.13			-162.25	-125.63
0	233563	961441	-119.67	-118.13	284805	1431054	-123.20	-120.63
10	93169	305002	-109.91	-108.13	63659	399641	-112.03	-110.63
20	31203	79423	-98.77	-98.13	18736	125826	-101.86	-100.63
30	25312	28747	-93.13	-88.13	31275	38873	-94.64	-90.63
40	21701	9888	-90.20	-78.13	28480	13629	-93.00	-80.63

Table C.50: 128 real data samples decimated into a 16x16 complex rank revealing SVD

	256 real samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-146.65	-100.58			-158.96	-112.62
-10			-144.03	-95.58			-156.38	-107.62
-5			-135.17	-90.58			-143.47	-102.62
0	3763	23539	-87.69	-85.58	12701	88474	-99.48	-97.62
10	1648	7870	-77.98	-75.58	4005	30424	-89.63	-87.62
20	493	2499	-67.86	-65.58	432	8318	-78.68	-77.62
30	99	639	-56.33	-55.58	453	2788	-68.76	-67.62
40	88	226	-48.05	-45.58	457	904	-60.17	-57.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-170.37	-124.10			-172.16	-126.60
-10			-169.02	-119.10			-166.68	-121.60
-5	117536	748564	-117.44	-114.10			-152.73	-116.60
0	35561	348388	-110.44	-109.10	103892	459048	-113.63	-111.60
10	29516	89796	-99.78	-99.10	17228	128655	-101.55	-101.60
20	3152	32299	-89.71	-89.10	10791	45119	-93.61	-91.60
30	1973	11734	-81.55	-79.10	2036	12082	-82.03	-81.60
40	1106	3197	-71.30	-69.10	1607	4709	-74.24	-71.60

Table C.51: 256 real data samples decimated into a 16x16 complex rank revealing SVD

	512 real samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.98	-91.55			-157.88	-103.59
-10			-139.20	-86.55			-151.21	-98.59
-5	3665	16061	-83.96	-81.55	10765	65370	-96.81	-93.59
0	681	8535	-79.03	-76.55	4781	31364	-90.25	-88.59
10	417	2454	-68.07	-66.55	1061	9148	-79.43	-78.59
20	71	792	-58.22	-56.55	235	3136	-69.98	-68.59
30	31	241	-47.81	-46.55	165	935	-60.05	-58.59
40	17	78	-38.20	-36.55	52	306	-50.01	-48.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.22	-115.07			-171.69	-117.57
-10			-162.29	-110.07			-165.44	-112.57
-5	23694	232808	-107.40	-105.07	55723	307507	-110.01	-107.57
0	8650	124250	-101.63	-100.07	20196	162959	-104.26	-102.57
10	1914	39037	-91.79	-90.07	6912	53032	-94.59	-92.57
20	1770	11456	-81.03	-80.07	892	14707	-83.26	-82.57
30	796	3700	-71.99	-70.07	704	4700	-73.49	-72.57
40	445	1260	-62.47	-60.07	656	1605	-64.95	-62.57

Table C.52: 512 real data samples decimated into a 16x16 complex rank revealing SVD

	1024 real samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.63	-82.52			-155.28	-94.56
-10			-119.93	-77.52			-137.03	-89.56
-5	457	5350	-74.57	-72.52	1059	20761	-86.22	-84.56
0	175	3027	-69.44	-67.52	1373	10419	-80.35	-79.56
10	57	863	-58.63	-57.52	162	3502	-70.83	-69.56
20	13	262	-48.40	-47.52	107	1127	-61.24	-59.56
30	6	87	-39.01	-37.52	31	343	-50.54	-49.56
40	6	27	-28.73	-27.52	5	123	-41.74	-39.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-166.79	-106.04			-169.79	-108.54
-10			-135.75	-101.04			-153.69	-103.54
-5	8593	78482	-97.81	-96.04	8875	107148	-100.44	-98.54
0	4125	42261	-92.39	-91.04	2831	58483	-95.38	-93.54
10	620	13465	-82.57	-81.04	2601	17774	-85.16	-83.54
20	465	4271	-72.55	-71.04	249	5366	-74.59	-73.54
30	109	1288	-62.22	-61.04	206	1822	-65.21	-63.54
40	82	436	-52.76	-51.04	143	602	-55.66	-53.54

Table C.53: 1024 real data samples decimated into a 16x16 complex rank revealing SVD

	2048 real samples $\rightarrow$ 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-137.69	-73.49			-149.24	-85.53
-10	194	3363	-70.66	-68.49	630	13182	-82.44	-80.53
-5	133	1788	-65.02	-63.49	440	7190	-77.08	-75.53
0	53	1022	-60.22	-58.49	232	3927	-71.83	-70.53
10	17	309	-49.79	-48.49	70	1246	-61.96	-60.53
20	7	96	-39.67	-38.49	38	384	-51.78	-50.53
30	4	30	-29.72	-28.49	10	131	-42.37	-40.53
40	11	10	-19.96	-18.49	47	38	-31.89	-30.53
	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-160.52	-97.01			-164.03	-99.51
-10	4743	49938	-94.03	-92.01	11288	69692	-96.95	-94.51
-5	1561	25934	-88.32	-87.01	1912	35652	-91.12	-89.51
0	915	14632	-83.27	-82.01	2014	20247	-86.18	-84.51
10	40	4904	-73.81	-72.01	577	6112	-75.72	-74.51
20	84	1502	-63.63	-62.01	173	1993	-65.97	-64.51
30	57	485	-53.71	-52.01	107	626	-55.86	-54.51
40	152	145	-43.40	-42.01	241	194	-46.10	-44.51

Table C.54: 2048 real data samples decimated into a 16x16 complex rank revealing SVD

	4096 real samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-126.95	-64.46			-134.31	-76.50
-10	76	1170	-61.44	-59.46	47	4660	-73.33	-71.50
-5	27	621	-55.90	-54.46	67	2455	-67.89	-66.50
0	18	350	-51.02	-49.46	43	1423	-63.02	-61.50
10	4	113	-40.85	-39.46	12	437	-52.86	-51.50
20	18	35	-30.74	-29.46	71	144	-43.15	-41.50
30	27	11	-21.21	-19.46	108	45	-33.04	-31.50
40	26	4	-11.21	-9.46	105	14	-23.14	-21.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.65	-87.98			-150.61	-90.48
-10	389	17092	-84.62	-82.98	1268	22747	-87.07	-85.48
-5	368	9129	-79.23	-77.98	282	12653	-82.03	-80.48
0	207	5324	-74.43	-72.98	192	6892	-76.64	-75.48
10	62	1651	-64.39	-62.98	122	2291	-67.36	-65.48
20	236	534	-54.60	-52.98	277	710	-57.12	-55.48
30	268	166	-44.50	-42.98	361	226	-47.13	-45.48
40	292	52	-34.58	-32.98	372	69	-36.94	-35.48

Table C.55: 4096 real data samples decimated into a 16x16 complex rank revealing SVD

	8192 real samples → 16x16 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	25	719	-57.10	-55.43	77	2953	-69.48	-67.47
-10	9	400	-52.07	-50.43	65	1588	-64.08	-62.47
-5	9	220	-46.77	-45.43	31	870	-58.86	-57.47
0	10	125	-41.95	-40.43	52	495	-53.88	-52.47
10	25	39	-31.75	-30.43	99	160	-44.11	-42.47
20	36	13	-21.91	-20.43	139	51	-34.03	-32.47
30	53	4	-11.92	-10.43	209	16	-23.97	-22.47
40	51	1	-2.31	-0.43	203	5	-14.21	-12.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	303	10777	-80.59	-78.95	754	14598	-83.31	-81.45
-10	268	6053	-75.68	-73.95	384	8112	-78.28	-76.45
-5	62	3303	-70.38	-68.95	166	4468	-73.01	-71.45
0	293	1888	-65.48	-63.95	378	2541	-68.17	-66.45
10	662	595	-55.51	-53.95	1056	776	-57.86	-56.45
20	892	188	-45.45	-43.95	955	248	-47.92	-46.45
30	1105	60	-35.78	-33.95	922	78	-37.86	-36.45
40	1137	19	-25.90	-23.95	917	25	-28.53	-26.45

Table C.56: 8192 real data samples decimated into a 16x16 complex rank revealing SVD

	128 complex samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.63	-106.61			-157.75	-118.65
-10			-143.33	-101.61			-155.97	-113.65
-5			-115.70	-96.61			-145.96	-108.65
0	10196	52196	-94.27	-91.61	27871	173186	-104.65	-103.65
10	4039	11969	-82.00	-81.61	8516	52255	-94.35	-93.65
20	1110	4475	-73.02	-71.61	10286	16706	-86.04	-83.65
30	1802	1546	-68.11	-61.61	6699	5167	-78.85	-73.65
40	1593	544	-66.25	-51.61	6523	2180	-78.41	-63.65
	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.12	-130.13			-172.30	-132.63
-10			-166.75	-125.13			-166.69	-127.63
-5	320427	1259844	-122.23	-120.13	195765	2019379	-125.98	-122.63
0	249683	652951	-116.80	-115.13	317323	906866	-119.83	-117.63
10	45429	185166	-105.47	-105.13	22048	277532	-108.68	-107.63
20	24984	80766	-98.89	-95.13	27552	75752	-98.22	-97.63
30	25004	24639	-91.80	-85.13	36933	20856	-93.52	-87.63
40	25828	6264	-90.07	-75.13	34091	8470	-92.70	-77.63

Table C.57: 128 complex data samples decimated into a 32x32 complex rank revealing SVD

	256 complex samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.72	-97.58			-157.27	-109.62
-10			-133.43	-92.58			-149.21	-104.62
-5	3430	31402	-90.22	-87.58	11283	125392	-101.75	-99.62
0	2644	16021	-84.09	-82.58	15071	57363	-95.36	-94.62
10	1361	5076	-74.83	-72.58	4961	20787	-86.39	-84.62
20	263	1590	-64.27	-62.58	2912	6408	-77.14	-74.62
30	365	482	-56.27	-52.58	1042	2004	-67.53	-64.62
40	335	159	-52.70	-42.58	1397	669	-65.41	-54.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-167.87	-121.10			-172.73	-123.60
-10			-159.93	-116.10			-154.78	-118.60
-5	101442	436286	-113.13	-111.10	110356	658611	-116.34	-113.60
0	41557	207675	-106.79	-106.10	43836	351090	-110.94	-108.60
10	9752	75558	-97.70	-96.10	17252	95334	-100.00	-98.60
20	7129	20336	-86.20	-86.10	4151	31877	-89.92	-88.60
30	5357	6944	-79.25	-76.10	8551	10124	-83.49	-78.60
40	4801	2245	-76.39	-66.10	7079	2972	-79.48	-68.60

Table C.58: 256 complex data samples decimated into a 32x32 complex rank revealing SVD

SNR (dB)	512 complex samples → 32x32 complex SVD: Sample Rate 100 (Mhz), 5.120 μs pulse width: $\mathrm{E}[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	400 (Mhz), 1.280 μs pulse width: $\mathrm{E}[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-144.29	-88.55			-154.87	-100.59
-10	959	21756	-86.22	-83.55	14846	85704	-98.55	-95.59
-5	1354	10154	-80.13	-78.55	4980	43325	-92.66	-90.59
0	895	5636	-75.42	-73.55	1786	22419	-86.90	-85.59
10	147	1661	-64.36	-63.55	681	7125	-76.93	-75.59
20	74	550	-55.10	-53.55	279	2303	-67.17	-65.59
30	56	173	-45.62	-43.55	153	702	-57.20	-55.59
40	47	54	-38.13	-33.55	189	215	-49.99	-45.59
SNR (dB)	**Sample Rate 1500 (Mhz), 0.341 μs pulse width: $\mathrm{E}[\hat{f}]$**	**$\sqrt{V[\hat{f}]}$**	**$10\log_{10}\frac{1}{MSE}$**	**CRB**	**2000 (Mhz), 0.256 μs pulse width: $\mathrm{E}[\hat{f}]$**	**$\sqrt{V[\hat{f}]}$**	**$10\log_{10}\frac{1}{MSE}$**	**CRB**
-15			-165.58	-112.07			-168.89	-114.57
-10			-146.71	-107.07			-157.35	-109.57
-5	7076	157861	-103.83	-102.07	25709	199089	-106.00	-104.57
0	6566	82475	-98.47	-97.07	15028	112373	-101.12	-99.57
10	1142	26450	-88.28	-87.07	3224	37204	-91.17	-89.57
20	1378	8163	-78.41	-77.07	2164	10531	-80.68	-79.57
30	1007	2630	-69.15	-67.07	761	3790	-72.05	-69.57
40	778	876	-62.13	-57.07	1001	1166	-64.56	-59.57

Table C.59: 512 complex data samples decimated into a 32x32 complex rank revealing SVD

	1024 complex samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-136.69	-79.52			-149.86	-91.56
-10	651	6571	-76.52	-74.52	1168	26634	-88.40	-86.56
-5	506	3443	-70.83	-69.52	1205	13925	-82.90	-81.56
0	170	1954	-65.70	-64.52	758	8033	-77.91	-76.56
10	79	598	-55.67	-54.52	46	2456	-67.66	-66.56
20	44	199	-46.05	-44.52	124	749	-57.72	-56.56
30	15	61	-35.89	-34.52	94	247	-48.57	-46.56
40	20	19	-30.09	-24.52	90	81	-42.73	-36.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-158.66	-103.04			-159.59	-105.54
-10	9630	107032	-100.64	-98.04	7882	132016	-102.46	-100.54
-5	5666	51135	-94.18	-93.04	8560	66404	-96.61	-95.54
0	1087	30751	-89.62	-88.04	2623	40835	-92.33	-90.54
10	421	9744	-79.63	-78.04	411	12856	-82.24	-80.54
20	357	2943	-69.32	-68.04	448	3956	-72.10	-70.54
30	313	916	-60.10	-58.04	464	1227	-62.74	-60.54
40	329	297	-53.81	-48.04	419	396	-56.36	-50.54

Table C.60: 1024 complex data samples decimated into a 32x32 complex rank revealing SVD

	2048 complex samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-116.77	-70.49			-129.46	-82.53
-10	183	2260	-67.24	-65.49	167	9184	-79.28	-77.53
-5	35	1287	-62.06	-60.49	171	4801	-73.47	-72.53
0	33	670	-56.46	-55.49	234	2912	-69.30	-67.53
10	24	212	-46.64	-45.49	74	883	-58.82	-57.53
20	7	69	-36.68	-35.49	42	280	-49.13	-47.53
30	8	22	-28.00	-25.49	40	84	-40.06	-37.53
40	13	7	-23.10	-15.49	55	28	-35.01	-27.53
	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	4273	75186	-97.78	-94.01	6826	87837	-98.60	-96.51
-10	3126	34850	-90.86	-89.01	2234	45315	-92.98	-91.51
-5	623	17812	-84.91	-84.01	1974	24923	-87.82	-86.51
0	462	9890	-79.87	-79.01	1227	14069	-82.95	-81.51
10	267	3435	-70.76	-69.01	186	4239	-72.54	-71.51
20	184	1066	-60.68	-59.01	325	1354	-62.81	-61.51
30	245	322	-51.75	-49.01	200	453	-54.64	-51.51
40	320	103	-47.10	-39.01	243	140	-49.28	-41.51

Table C.61: 2048 complex data samples decimated into a 32x32 complex rank revealing SVD

	4096 complex samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	28	1440	-63.05	-61.46	185	5750	-75.26	-73.50
-10	14	784	-57.78	-56.46	57	3076	-69.68	-68.50
-5	10	433	-52.74	-51.46	73	1766	-64.96	-63.50
0	5	253	-48.14	-46.46	13	993	-59.90	-58.50
10	7	80	-38.08	-36.46	17	312	-49.94	-48.50
20	20	25	-28.11	-26.46	73	101	-40.31	-38.50
30	19	8	-20.16	-16.46	80	31	-32.38	-28.50
40	24	2	-16.72	-6.46	93	10	-28.98	-18.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	769	21602	-86.83	-84.98	848	28915	-89.22	-87.48
-10	620	11806	-81.42	-79.98	491	15309	-83.69	-82.48
-5	109	6514	-76.24	-74.98	112	8675	-78.81	-77.48
0	55	3792	-71.67	-69.98	100	4887	-73.79	-72.48
10	116	1168	-61.36	-59.98	244	1557	-63.95	-62.48
20	290	364	-51.47	-49.98	404	499	-54.19	-52.48
30	314	115	-43.80	-39.98	415	157	-46.09	-42.48
40	346	37	-40.31	-29.98	404	51	-42.79	-32.48

Table C.62: 4096 complex data samples decimated into a 32x32 complex rank revealing SVD

	8192 complex samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	8	496	-53.97	-52.43	51	1996	-66.06	-64.47
-10	7	273	-48.67	-47.43	27	1105	-60.82	-59.47
-5	6	149	-43.43	-42.43	66	614	-55.81	-54.47
0	23	87	-38.81	-37.43	72	346	-50.83	-49.47
10	30	28	-29.02	-27.43	125	110	-41.08	-39.47
20	45	9	-19.32	-17.43	176	35	-31.38	-29.47
30	64	3	-12.64	-7.43	256	11	-24.67	-19.47
40	69	1	-10.49	2.57	278	4	-22.60	-9.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	91	7419	-77.37	-75.95	235	10073	-80.11	-78.45
-10	104	4232	-72.54	-70.95	262	5509	-74.88	-73.45
-5	188	2300	-67.17	-65.95	358	3107	-69.82	-68.45
0	521	1295	-62.29	-60.95	726	1698	-64.56	-63.45
10	761	418	-52.44	-50.95	1131	555	-54.95	-53.45
20	1068	129	-42.80	-40.95	992	172	-45.28	-43.45
30	1137	41	-36.01	-30.95	934	55	-38.41	-33.45
40	1140	13	-34.07	-20.95	928	18	-36.47	-23.45

Table C.63: 8192 complex data samples decimated into a 32x32 complex rank revealing SVD

	128 real samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-146.86	-109.61			-158.84	-121.65
-10			-144.81	-104.61			-157.44	-116.65
-5			-132.74	-99.61			-140.68	-111.65
0	12735	57811	-95.30	-94.61	80062	263819	-108.96	-106.65
10	3584	18489	-85.26	-84.61	12681	68889	-96.85	-96.65
20	2724	6614	-77.21	-74.61	12194	27522	-90.20	-86.65
30	2943	2125	-73.16	-64.61	11813	7876	-84.93	-76.65
40	2489	549	-71.26	-54.61	9435	2625	-83.09	-66.65
	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.25	-133.13			-173.15	-135.63
-10			-170.58	-128.13			-172.10	-130.63
-5			-161.37	-123.13			-166.49	-125.63
0	355880	967933	-120.15	-118.13	379728	1370284	-123.06	-120.63
10	64229	295689	-109.81	-108.13	77920	329819	-110.73	-110.63
20	51180	109357	-102.10	-98.13	47061	106255	-101.83	-100.63
30	34023	31227	-95.61	-88.13	49989	39014	-98.17	-90.63
40	35773	8456	-94.36	-78.13	50747	12270	-97.44	-80.63

Table C.64: 128 real data samples decimated into a 32x32 complex rank revealing SVD

SNR (dB)	256 real samples → 32x32 complex SVD — Sample Rate: 100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-146.93	-100.58			-157.46	-112.62
-10			-142.80	-95.58			-156.24	-107.62
-5	221808	1011860	-95.07	-90.58			-136.80	-102.62
0	3949	21571	-86.40	-85.58	9747	92488	-99.07	-97.62
10	433	6503	-76.36	-75.58	5024	32773	-90.59	-87.62
20	834	1885	-66.34	-65.58	3559	8885	-79.99	-77.62
30	507	658	-59.32	-55.58	2067	2944	-72.12	-67.62
40	453	234	-56.54	-45.58	1973	898	-69.61	-57.62

SNR (dB)	Sample Rate: 1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-170.99	-124.10			-173.02	-126.60
-10			-168.24	-119.10			-169.65	-121.60
-5			-157.24	-114.10			-157.93	-116.60
0	56513	337379	-111.24	-109.10	74780	396337	-111.83	-111.60
10	19152	103069	-100.22	-99.10	32089	147134	-103.67	-101.60
20	11969	30673	-90.38	-89.10	10009	46856	-93.60	-91.60
30	7704	10416	-83.63	-79.10	7820	12977	-84.45	-81.60
40	7722	3197	-81.19	-69.10	9345	4005	-83.03	-71.60

Table C.65: 256 real data samples decimated into a 32x32 complex rank revealing SVD

	512 real samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.61	-91.55			-157.52	-103.59
-10			-137.81	-86.55			-149.74	-98.59
-5	1341	17863	-84.79	-81.55	7425	60590	-95.61	-93.59
0	1076	7920	-78.05	-76.55	2877	31757	-89.87	-88.59
10	323	2661	-68.38	-66.55	978	10314	-80.37	-78.59
20	101	823	-58.16	-56.55	245	3147	-70.24	-68.59
30	49	260	-48.50	-46.55	427	955	-60.86	-58.59
40	62	76	-41.19	-36.55	287	326	-53.58	-48.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.26	-115.07			-172.53	-117.57
-10			-160.87	-110.07			-164.80	-112.57
-5	26156	230122	-106.94	-105.07	20196	293309	-109.50	-107.57
0	24737	130725	-102.19	-100.07	18872	156729	-103.68	-102.57
10	3491	37988	-91.56	-90.07	6607	48353	-93.76	-92.57
20	2630	11569	-81.52	-80.07	1533	16660	-84.37	-82.57
30	1057	3684	-71.60	-70.07	1603	4793	-74.14	-72.57
40	1011	1111	-64.87	-60.07	1128	1518	-66.74	-62.57

Table C.66: 512 real data samples decimated into a 32x32 complex rank revealing SVD

	1024 real samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.56	-82.52			-154.93	-94.56
-10			-126.47	-77.52			-144.17	-89.56
-5	406	5347	-74.71	-72.52	2838	21210	-86.97	-84.56
0	75	2847	-69.07	-67.52	1314	11846	-81.51	-79.56
10	79	939	-59.53	-57.52	287	3434	-70.76	-69.56
20	45	276	-49.19	-47.52	128	1096	-60.70	-59.56
30	23	87	-39.30	-37.52	137	331	-51.51	-49.56
40	30	29	-33.53	-27.52	105	110	-44.71	-39.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-167.21	-106.04			-169.02	-108.54
-10	14698	159441	-104.27	-101.04			-149.44	-103.54
-5	4735	75644	-97.80	-96.04	4719	100159	-100.05	-98.54
0	1226	38926	-91.72	-91.04	5590	59018	-95.34	-93.54
10	1825	13131	-82.45	-81.04	1519	17687	-84.89	-83.54
20	412	4434	-72.94	-71.04	798	5665	-75.24	-73.54
30	399	1355	-63.14	-61.04	517	1738	-65.32	-63.54
40	405	411	-56.17	-51.04	507	554	-58.57	-53.54

Table C.67: 1024 real data samples decimated into a 32x32 complex rank revealing SVD

	2048 real samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-137.90	-73.49			-151.15	-85.53
-10	73	3285	-70.50	-68.49	454	13106	-82.40	-80.53
-5	36	1761	-65.02	-63.49	639	6947	-76.80	-75.53
0	66	1008	-60.03	-58.49	94	3846	-71.68	-70.53
10	20	319	-50.03	-48.49	66	1236	-61.93	-60.53
20	15	98	-39.98	-38.49	52	391	-51.84	-50.53
30	10	31	-30.77	-28.49	53	123	-42.89	-40.53
40	13	10	-24.85	-18.49	52	37	-36.56	-30.53
	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-161.48	-97.01			-163.98	-99.51
-10	4202	47574	-93.57	-92.01	4496	66317	-96.48	-94.51
-5	475	25480	-88.12	-87.01	2654	35071	-91.00	-89.51
0	781	15332	-83.86	-82.01	1598	19027	-85.53	-84.51
10	208	4624	-73.29	-72.01	259	6179	-75.86	-74.51
20	173	1536	-63.59	-62.01	309	1972	-66.17	-64.51
30	242	446	-54.30	-52.01	179	603	-56.21	-54.51
40	306	152	-48.49	-42.01	232	195	-50.70	-44.51

Table C.68: 2048 real data samples decimated into a 32x32 complex rank revealing SVD

	4096 real samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-124.14	-64.46			-134.74	-76.50
-10	79	1167	-61.24	-59.46	201	4463	-73.06	-71.50
-5	30	606	-55.68	-54.46	46	2484	-67.99	-66.50
0	16	356	-50.92	-49.46	64	1329	-62.44	-61.50
10	8	108	-40.85	-39.46	25	445	-53.04	-51.50
20	17	35	-31.00	-29.46	63	138	-42.94	-41.50
30	18	11	-22.35	-19.46	76	44	-34.24	-31.50
40	21	3	-17.73	-9.46	84	14	-29.83	-21.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.38	-87.98			-144.04	-90.48
-10	342	17246	-84.78	-82.98	868	22493	-87.14	-85.48
-5	332	9387	-79.53	-77.98	405	12254	-81.87	-80.48
0	109	5467	-74.75	-72.98	308	6870	-76.74	-75.48
10	92	1662	-64.38	-62.98	201	2177	-66.87	-65.48
20	274	518	-54.45	-52.98	367	695	-57.02	-55.48
30	306	162	-45.57	-42.98	420	223	-48.35	-45.48
40	335	53	-41.43	-32.98	409	73	-43.97	-35.48

Table C.69: 4096 real data samples decimated into a 32x32 complex rank revealing SVD

	8192 real samples → 32x32 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	38	709	-57.00	-55.43	161	2882	-69.21	-67.47
-10	12	393	-51.88	-50.43	44	1563	-63.95	-62.47
-5	9	218	-46.60	-45.43	41	855	-58.73	-57.47
0	11	121	-41.65	-40.43	52	484	-53.74	-52.47
10	26	39	-31.70	-30.43	119	154	-43.73	-42.47
20	40	12	-22.00	-20.43	161	48	-33.94	-32.47
30	61	4	-14.24	-10.43	241	16	-26.36	-22.47
40	67	1	-10.98	-0.43	268	5	-23.10	-12.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	471	10904	-80.71	-78.95	261	14383	-83.25	-81.45
-10	151	5896	-75.45	-73.95	171	7943	-78.04	-76.45
-5	110	3400	-70.65	-68.95	198	4308	-72.72	-71.45
0	329	1854	-65.40	-63.95	486	2458	-67.80	-66.45
10	668	582	-55.33	-53.95	1099	762	-57.66	-56.45
20	961	186	-45.67	-43.95	1046	247	-48.06	-46.45
30	1134	58	-37.53	-33.95	939	78	-40.33	-36.45
40	1137	19	-34.54	-23.95	928	25	-37.09	-26.45

Table C.70: 8192 real data samples decimated into a 32x32 complex rank revealing SVD

	128 complex samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.74	-106.61			-158.46	-118.65
-10			-142.63	-101.61			-156.49	-113.65
-5	20877	81396	-98.56	-96.61			-135.38	-108.65
0	13189	40939	-92.64	-91.61	32655	166010	-104.35	-103.65
10	1605	12329	-81.68	-81.61	8927	53490	-94.41	-93.65
20	2100	4937	-74.53	-71.61	7661	14878	-84.81	-83.65
30	1695	1165	-67.44	-61.61	6199	5938	-79.44	-73.65
40	1468	468	-65.30	-51.61	6103	1551	-77.61	-63.65
	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.87	-130.13			-173.12	-132.63
-10			-165.91	-125.13			-167.69	-127.63
-5			-133.38	-120.13	592341	2264367	-127.35	-122.63
0	187225	567728	-115.35	-115.13	266331	989883	-120.12	-117.63
10	50879	193139	-105.99	-105.13	68400	329098	-110.30	-107.63
20	34952	68247	-97.66	-95.13	51739	84937	-100.23	-97.63
30	24163	20168	-90.90	-85.13	36256	34161	-94.66	-87.63
40	22868	6572	-89.01	-75.13	29749	7432	-91.20	-77.63

Table C.71: 128 complex data samples decimated into a 64x64 complex rank revealing SVD

	256 complex samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.35	-97.58			-158.09	-109.62
-10			-135.67	-92.58			-150.77	-104.62
-5	9726	29955	-90.14	-87.58	24329	115967	-101.94	-99.62
0	4462	15953	-84.51	-82.58	8325	53451	-94.39	-94.62
10	869	4525	-73.39	-72.58	4990	20060	-86.43	-84.62
20	438	1548	-64.09	-62.58	1161	6090	-75.69	-74.62
30	430	445	-56.46	-52.58	1395	1676	-67.34	-64.62
40	390	149	-54.15	-42.58	1597	636	-66.28	-54.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-167.29	-121.10			-171.84	-123.60
-10			-154.68	-116.10			-161.97	-118.60
-5	56743	511377	-113.39	-111.10	67724	521037	-114.07	-113.60
0	63610	238785	-107.60	-106.10	25727	293938	-109.34	-108.60
10	12126	72483	-97.71	-96.10	29289	100720	-101.16	-98.60
20	7682	21756	-87.48	-86.10	5313	25939	-88.57	-88.60
30	7710	7102	-80.95	-76.10	8540	8816	-82.19	-78.60
40	6037	2278	-77.94	-66.10	7793	3446	-80.35	-68.60

Table C.72: 256 complex data samples decimated into a 64x64 complex rank revealing SVD

	512 complex samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-142.88	-88.55			-155.32	-100.59
-10	1902	20503	-86.42	-83.55			-114.70	-95.59
-5	996	10807	-80.44	-78.55	2505	36068	-90.83	-90.59
0	453	5069	-73.65	-73.55	3080	23503	-87.34	-85.59
10	153	1542	-63.66	-63.55	287	7218	-77.02	-75.59
20	77	574	-54.96	-53.55	170	2138	-66.73	-65.59
30	22	179	-44.93	-43.55	126	686	-56.79	-55.59
40	58	53	-38.80	-33.55	193	214	-49.55	-45.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-165.74	-112.07			-168.72	-114.57
-10	49895	309416	-110.48	-107.07	52040	469950	-113.67	-109.57
-5	11834	151361	-103.30	-102.07	13505	214781	-106.65	-104.57
0	14262	82071	-98.55	-97.07	11492	121313	-101.63	-99.57
10	3175	25488	-88.29	-87.07	4288	32818	-90.30	-89.57
20	943	7564	-77.24	-77.07	2253	10627	-81.02	-79.57
30	1143	2692	-69.38	-67.07	824	3978	-72.49	-69.57
40	796	795	-61.67	-57.07	963	1084	-64.12	-59.57

Table C.73: 512 complex data samples decimated into a 64x64 complex rank revealing SVD

	1024 complex samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-136.94	-79.52			-147.85	-91.56
-10	413	6118	-75.71	-74.52	732	26256	-88.31	-86.56
-5	254	3699	-71.31	-69.52	1065	13835	-82.88	-81.56
0	225	2084	-66.33	-64.52	840	7935	-78.14	-76.56
10	54	592	-55.68	-54.52	143	2289	-67.32	-66.56
20	5	202	-46.12	-44.52	27	816	-58.35	-56.56
30	4	60	-35.42	-34.52	13	254	-48.08	-46.56
40	3	20	-26.13	-24.52	15	77	-37.94	-36.56
	Sample Rate							
SNR	1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-160.12	-103.04			-161.76	-105.54
-10	3548	100103	-100.16	-98.04	5051	133397	-102.38	-100.54
-5	4320	51540	-94.12	-93.04	7211	70730	-96.96	-95.54
0	3468	28529	-89.36	-88.04	1440	40702	-92.36	-90.54
10	927	8992	-79.19	-78.04	710	12487	-81.94	-80.54
20	144	2987	-69.47	-68.04	201	3926	-71.88	-70.54
30	78	893	-59.05	-58.04	146	1202	-61.60	-60.54
40	53	303	-49.86	-48.04	55	364	-51.74	-50.54

Table C.74: 1024 complex data samples decimated into a 64x64 complex rank revealing SVD

	2048 complex samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-125.62	-70.49			-137.25	-82.53
-10	111	2207	-66.77	-65.49	195	9477	-79.52	-77.53
-5	69	1210	-61.54	-60.49	151	5066	-74.18	-72.53
0	23	700	-57.02	-55.49	98	2842	-68.96	-67.53
10	18	213	-46.70	-45.49	70	934	-59.58	-57.53
20	5	69	-36.89	-35.49	20	273	-48.79	-47.53
30	5	22	-26.86	-25.49	25	90	-39.30	-37.53
40	11	7	-17.60	-15.49	41	28	-29.71	-27.53
	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	$E[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-148.16	-94.01			-138.94	-96.51
-10	2874	32161	-90.00	-89.01	3455	44831	-93.11	-91.51
-5	943	19397	-85.78	-84.01	776	25760	-88.13	-86.51
0	573	10064	-80.09	-79.01	495	14550	-83.22	-81.51
10	146	3511	-70.87	-69.01	379	4413	-73.13	-71.51
20	43	1035	-60.39	-59.01	120	1400	-62.82	-61.51
30	137	312	-49.93	-49.01	118	429	-52.74	-51.51
40	195	104	-41.25	-39.01	262	140	-43.98	-41.51

Table C.75: 2048 complex data samples decimated into a 64x64 complex rank revealing SVD

	4096 complex samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	47	1494	-63.38	-61.46	295	5702	-75.12	-73.50
-10	18	779	-57.88	-56.46	59	3150	-70.03	-68.50
-5	10	431	-52.61	-51.46	35	1719	-64.66	-63.50
0	7	254	-48.10	-46.46	31	1003	-60.15	-58.50
10	4	78	-37.81	-36.46	31	310	-49.85	-48.50
20	25	25	-27.92	-26.46	100	98	-39.88	-38.50
30	28	8	-18.23	-16.46	110	31	-30.15	-28.50
40	25	2	-9.36	-6.46	104	10	-21.23	-18.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	1092	22633	-87.19	-84.98	1695	28321	-89.04	-87.48
-10	205	11941	-81.67	-79.98	530	15705	-83.84	-82.48
-5	348	6428	-76.13	-74.98	336	8524	-78.53	-77.48
0	180	3714	-71.33	-69.98	229	5018	-73.95	-72.48
10	34	1161	-61.12	-59.98	127	1521	-63.62	-62.48
20	242	368	-51.37	-49.98	335	501	-54.17	-52.48
30	274	120	-41.75	-39.98	378	157	-44.13	-42.48
40	319	36	-32.69	-29.98	377	49	-35.29	-32.48

Table C.76: 4096 complex data samples decimated into a 64x64 complex rank revealing SVD

	8192 complex samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	6	500	-53.98	-52.43	66	1964	-65.77	-64.47
-10	9	279	-48.90	-47.43	45	1107	-60.85	-59.47
-5	9	153	-43.65	-42.43	41	612	-55.78	-54.47
0	21	86	-38.73	-37.43	81	340	-50.68	-49.47
10	28	27	-28.66	-27.43	109	109	-40.72	-39.47
20	44	9	-19.02	-17.43	163	35	-30.90	-29.47
30	53	3	-9.03	-7.43	212	11	-21.17	-19.47
40	51	1	-1.45	2.57	203	3	-13.45	-9.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	158	7197	-77.12	-75.95	231	9925	-79.95	-78.45
-10	157	4101	-72.30	-70.95	106	5501	-74.80	-73.45
-5	201	2268	-67.20	-65.95	363	3120	-69.86	-68.45
0	469	1287	-62.25	-60.95	667	1721	-64.67	-63.45
10	668	408	-52.30	-50.95	1067	562	-54.97	-53.45
20	993	130	-42.40	-40.95	934	170	-44.62	-43.45
30	1129	41	-32.65	-30.95	922	55	-35.28	-33.45
40	1137	13	-25.13	-20.95	917	18	-27.67	-23.45

Table C.77: 8192 complex data samples decimated into a 64x64 complex rank revealing SVD

	128 real samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 1.280 μs pulse width				400 (Mhz), 0.320 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-146.48	-109.61			-157.41	-121.65
-10			-145.15	-104.61			-157.53	-116.65
-5			-136.34	-99.61			-121.42	-111.65
0	2223	64944	-95.94	-94.61	95188	313664	-110.51	-106.65
10	6116	17590	-85.41	-84.61	12335	73825	-97.63	-96.65
20	3654	5223	-77.37	-74.61	10967	22201	-88.54	-86.65
30	2801	2032	-72.76	-64.61	9614	6777	-84.75	-76.65
40	2613	514	-72.00	-54.61	10213	2731	-84.17	-66.65

	Sample Rate							
SNR	1500 (Mhz), 0.085 μs pulse width				2000 (Mhz), 0.064 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-170.26	-133.13			-172.13	-135.63
-10			-168.82	-128.13			-171.69	-130.63
-5			-161.30	-123.13			-161.65	-125.63
0	174204	1023392	-120.30	-118.13	332960	1570520	-123.89	-120.63
10	23189	277966	-108.65	-108.13	127499	356374	-111.54	-110.63
20	28111	90114	-99.55	-98.13	61733	124554	-103.36	-100.63
30	37029	32007	-96.08	-88.13	56384	42240	-99.79	-90.63
40	38591	10262	-95.61	-78.13	53957	14048	-98.26	-80.63

Table C.78: 128 real data samples decimated into a 64x64 complex rank revealing SVD

	256 real samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 2.560 μs pulse width				400 (Mhz), 0.640 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-146.82	-100.58			-158.58	-112.62
-10			-142.74	-95.58			-153.82	-107.62
-5			-104.44	-90.58			-145.09	-102.62
0	4709	24042	-86.69	-85.58	14088	94199	-100.09	-97.62
10	1300	6452	-76.12	-75.58	3490	28786	-88.98	-87.62
20	623	2142	-67.13	-65.58	2951	8862	-79.88	-77.62
30	543	710	-60.48	-55.58	3097	3100	-74.43	-67.62
40	647	234	-60.15	-45.58	2527	862	-71.42	-57.62
	Sample Rate							
SNR	1500 (Mhz), 0.171 μs pulse width				2000 (Mhz), 0.128 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-170.04	-124.10			-171.48	-126.60
-10			-168.20	-119.10			-170.46	-121.60
-5	186890	703875	-117.68	-114.10	7756964	46927100	-119.58	-116.60
0	14254	344740	-110.71	-109.10	55055	504241	-114.28	-111.60
10	12175	122105	-101.49	-99.10	38100	143378	-103.45	-101.60
20	12962	30715	-90.60	-89.10	11863	44103	-93.59	-91.60
30	9718	9214	-83.93	-79.10	9964	11469	-85.71	-81.60
40	8974	3066	-82.91	-69.10	12535	4036	-85.51	-71.60

Table C.79: 256 real data samples decimated into a 64x64 complex rank revealing SVD

	512 real samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 5.120 μs pulse width				400 (Mhz), 1.280 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-145.69	-91.55			-157.97	-103.59
-10			-138.03	-86.55			-150.23	-98.59
-5	412	14727	-83.66	-81.55	3354	63109	-95.96	-93.59
0	825	7301	-77.25	-76.55	3240	28715	-89.01	-88.59
10	180	2357	-67.81	-66.55	803	9064	-79.04	-78.59
20	85	754	-57.13	-56.55	813	3000	-69.98	-68.59
30	110	266	-49.81	-46.55	399	905	-60.72	-58.59
40	81	84	-43.31	-36.55	319	302	-54.95	-48.59
	Sample Rate							
SNR	1500 (Mhz), 0.341 μs pulse width				2000 (Mhz), 0.256 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-169.31	-115.07			-171.56	-117.57
-10			-158.16	-110.07			-163.90	-112.57
-5	24113	237183	-106.98	-105.07	50108	330922	-110.80	-107.57
0	10364	122866	-101.60	-100.07	11228	162374	-103.85	-102.57
10	2470	38042	-91.54	-90.07	3273	47740	-93.42	-92.57
20	2523	12769	-82.50	-80.07	3953	16689	-84.26	-82.57
30	894	4338	-72.89	-70.07	1892	5041	-75.06	-72.57
40	1319	1190	-67.17	-60.07	1545	1470	-68.83	-62.57

Table C.80: 512 real data samples decimated into a 64x64 complex rank revealing SVD

SNR (dB)	1024 real samples → 64x64 complex SVD — Sample Rate: 100 (Mhz), 10.240 μs pulse width				400 (Mhz), 2.560 μs pulse width			
	$\mathrm{E}[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	$\mathrm{E}[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-143.77	-82.52			-156.11	-94.56
-10			-126.56	-77.52			-141.49	-89.56
-5	466	5387	-74.51	-72.52	2675	18349	-85.30	-84.56
0	357	2894	-69.26	-67.52	1305	11321	-80.99	-79.56
10	71	849	-58.44	-57.52	235	3424	-70.73	-69.56
20	28	296	-49.39	-47.52	83	1135	-61.05	-59.56
30	3	85	-38.44	-37.52	43	330	-50.34	-49.56
40	4	28	-28.84	-27.52	18	116	-41.45	-39.56

SNR (dB)	Sample Rate: 1500 (Mhz), 0.683 μs pulse width				2000 (Mhz), 0.512 μs pulse width			
	$\mathrm{E}[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	$\mathrm{E}[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-166.98	-106.04			-170.27	-108.54
-10			-148.48	-101.04			-137.64	-103.54
-5	8579	74806	-97.80	-96.04	4285	102865	-100.20	-98.54
0	1499	40491	-92.06	-91.04	3787	55850	-94.90	-93.54
10	609	12813	-81.93	-81.04	1769	16675	-84.41	-83.54
20	340	3957	-71.75	-71.04	343	5445	-74.65	-73.54
30	91	1290	-62.05	-61.04	167	1720	-64.68	-63.54
40	22	405	-52.02	-51.04	101	578	-55.65	-53.54

Table C.81: 1024 real data samples decimated into a 64x64 complex rank revealing SVD

	2048 real samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 20.480 μs pulse width				400 (Mhz), 5.120 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-137.53	-73.49			-150.32	-85.53
-10	64	3417	-70.69	-68.49	1439	13348	-82.64	-80.53
-5	85	1763	-64.82	-63.49	380	7261	-77.27	-75.53
0	24	969	-59.73	-58.49	157	3962	-71.94	-70.53
10	14	292	-49.30	-48.49	75	1266	-62.13	-60.53
20	4	96	-39.75	-38.49	19	368	-51.24	-50.53
30	4	30	-29.43	-28.49	9	120	-41.65	-40.53
40	10	10	-19.95	-18.49	40	38	-31.81	-30.53
	Sample Rate							
SNR	1500 (Mhz), 1.365 μs pulse width				2000 (Mhz), 1.024 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-160.91	-97.01			-163.30	-99.51
-10	2045	50941	-94.13	-92.01	5140	67819	-96.72	-94.51
-5	817	26896	-88.36	-87.01	1384	34281	-90.81	-89.51
0	1150	14639	-83.21	-82.01	907	20109	-86.04	-84.51
10	94	4676	-73.42	-72.01	196	5878	-75.48	-74.51
20	61	1461	-63.25	-62.01	88	1926	-65.81	-64.51
30	78	491	-53.92	-52.01	129	600	-55.59	-54.51
40	144	150	-43.48	-42.01	230	189	-45.85	-44.51

Table C.82: 2048 real data samples decimated into a 64x64 complex rank revealing SVD

	4096 real samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 40.960 μs pulse width				400 (Mhz), 10.240 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-128.41	-64.46			-138.34	-76.50
-10	40	1118	-60.86	-59.46	136	4530	-73.24	-71.50
-5	21	625	-55.91	-54.46	97	2443	-67.70	-66.50
0	8	336	-50.55	-49.46	67	1376	-62.82	-61.50
10	2	108	-40.59	-39.46	13	422	-52.44	-51.50
20	20	34	-30.65	-29.46	78	135	-42.65	-41.50
30	27	11	-21.18	-19.46	109	44	-32.89	-31.50
40	27	3	-11.29	-9.46	104	14	-23.24	-21.50
	Sample Rate							
SNR	1500 (Mhz), 2.731 μs pulse width				2000 (Mhz), 2.048 μs pulse width			
(dB)	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E[$\hat{f}$]	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15			-149.91	-87.98			-149.17	-90.48
-10	569	16871	-84.61	-82.98	875	22229	-87.00	-85.48
-5	288	8887	-78.89	-77.98	241	12271	-81.80	-80.48
0	66	5209	-74.39	-72.98	191	6828	-76.73	-75.48
10	53	1621	-64.21	-62.98	118	2135	-66.58	-65.48
20	222	514	-54.20	-52.98	287	680	-56.65	-55.48
30	276	159	-44.09	-42.98	373	220	-46.93	-45.48
40	311	51	-34.51	-32.98	377	69	-37.09	-35.48

Table C.83: 4096 real data samples decimated into a 64x64 complex rank revealing SVD

	8192 real samples → 64x64 complex SVD							
	Sample Rate							
SNR	100 (Mhz), 81.920 μs pulse width				400 (Mhz), 20.480 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	29	720	-57.17	-55.43	109	2893	-69.20	-67.47
-10	9	395	-51.92	-50.43	21	1585	-64.02	-62.47
-5	7	221	-46.88	-45.43	17	880	-58.85	-57.47
0	16	123	-41.82	-40.43	35	493	-53.87	-52.47
10	27	38	-31.68	-30.43	103	153	-43.73	-42.47
20	32	12	-21.86	-20.43	148	49	-33.71	-32.47
30	53	4	-11.83	-10.43	207	15	-23.83	-22.47
40	51	1	-2.82	-0.43	203	5	-14.87	-12.47
	Sample Rate							
SNR	1500 (Mhz), 5.461 μs pulse width				2000 (Mhz), 4.096 μs pulse width			
(dB)	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB	E$[\hat{f}]$	$\sqrt{V[\hat{f}]}$	$10\log_{10}\frac{1}{MSE}$	CRB
-15	355	10862	-80.81	-78.95	218	14593	-83.20	-81.45
-10	160	5805	-75.29	-73.95	298	7866	-77.90	-76.45
-5	117	3188	-70.06	-68.95	155	4337	-72.75	-71.45
0	273	1834	-65.28	-63.95	415	2470	-67.84	-66.45
10	665	574	-55.15	-53.95	1030	761	-57.60	-56.45
20	924	186	-45.46	-43.95	960	243	-47.71	-46.45
30	1100	57	-35.18	-33.95	917	77	-37.72	-36.45
40	1137	19	-26.55	-23.95	917	26	-29.14	-26.45

Table C.84: 8192 real data samples decimated into a 64x64 complex rank revealing SVD

Bibliography

[1] L. Neng-Jing and Z. Yi-Ting, "A survey of radar ecm and eccm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, July 1995.

[2] S. Johnston, "Cesm - a new category of radar eccm," *IEEE AES Systems Magazine*, Feb. 1995.

[3] A. Giordano and T. Schonhoff, *Detection and Estimation Theory and Its Applications.* 2005.

[4] C. Wright, "Multidimensional direction of arrival performance bounds and optimization for non-stationary noise," 1995.

[5] B. C. Lovell and R. C. Williamson, "The statistical performance of some instantaneous frequency estimators," *IEEE Transactions on Signal Processing*, pp. 1293–1302, July 1992.

[6] L. Pellon, "A double nyquist digital product detector for quadrature sampling," *IEEE Trans. on Signal Processing*, vol. 40, July 1992.

[7] G. Golub and C. V. Loan, *Matrix Computations, 2nd ed.* The Johns Hopkins University Press, 1989.

[8] J. Demmel, *Applied Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, 1997.

[9] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C.* Press Sydicate of the University of Cambridge, 1988.

[10] K. Kota, "Architectural, numerical and implementation issues in the vlsi design of an integrated cordic-svd processor," *Rice University Masters Thesis*, ?

[11] N. Hemkumar, "A vlsi architecture for complex svd," *Rice University MS Thesis*, May 1991.

[12] J. Cavallaro and F. Luk, "Cordic arithmetic for an svd processor," *Journal on Parallel and Distributed Computing*, vol. 5, 1988.

[13] S. Hsiao and J. Delosme, "Parallel singular value decomposition of complex matricies using multidimensional cordic algorithms," *IEEE Transactions on Signal Processing*, vol. 44, Mar. 1996.

[14] J. Proakis and D. Manolakis, *Digital Signal Processing Principles, Algorithms, and Applications, 3rd ed.* Prectice-Hall, Inc., 1996.

[15] C. Chen, *Linear System Theory and Design, 3rd ed.* Oxford University Press, 1999.

[16] K. A. Rao, B.D., "Model based processing of signals: A state space approach," *Proceedings of the IEEE*, vol. 80, Feb. 1992.

[17] T. S. Hua, Y., "On svd for estimating generalized eigenvalues of singular matrix pencil in noise," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-39, Apr. 1991.

[18] N. Hatch, D. Holl, and D. Cyganski, *Power-Law Scattering Models and Nonlinear Parametric Estimation for Super-Resolution Radar.* MIT Lincoln Laboratory, TR-1095, Apr. 2004.

[19] S. Kung and D. Lin, "Optimal hankel-norm model reductions - multivariable systems," vol. AC-26, 1981.

[20] B. Moore, "Principal component analysis in linear systems: Controllability, obervability, and model reduction," vol. AC-26, Feb. 1981.

[21] A. van der Veen, M. Vanderveen, and A. Paulraj, "Joint angle and delay estimation using shift-invariance techniques," *IEEE Trans. on Signal Processing*, vol. 46, Apr. 1998.

[22] L. Ting, "Algorithms and fpga implementations of adaptive lms-based predictors for radar pulse identification," *University of Belfast PhD Dissertation*, July 2001.

[23] Intel, "Intel math kernel library reference manual." http://www.intel.com/, 2005.

[24] R. Wyrzykowski, J. Dongarra, M. Paprzycki, and J. W. (Eds.), "Implementation of givens qr-decomposition in fpga," *Parallel Processing and Applied Mathematics*, Sept. 2001.

[25] M. Toner and G. Roberts, "A bist scheme for an snr test of a sigma-delta adc," *IEEE*, Aug. 1993.

[26] R. Brent, F. Luk, and C. V. Loan, "Computation of the singular value decomposition using mesh-connected processors," *Journal on VLSI Computer Systems*, vol. 1, 1985.

[27] B. Yang and J. Bohme, "Reducing the computations of singular value decomposition array given by brent and luk," *SIAM Journal on Matrix Analysis and Applications*, vol. 12, Oct. 1991.

[28] A. Ahmedsaid, A. Amira, and A. Bouridane, "Improved svd systolic array and implementation on fpga," *The Queen's University of Belfast, UK*, ?

[29] G. Adams, A. Finn, and M. Griffin, "A fast implementation of the complex singular value decomposition on the connection machine," *IEEE Transactions*, 1991.

[30] M. Hestenes, "Inversion of matrices by biorthogonalization and related results," *SIAM Journal*, vol. 6, 1958.

[31] K. Kota, "Parallel algorithms and architectures for near-far resistant cdma acquisition," *Rice University PhD Thesis*, May 1996.

[32] R. Schreiber, "Solving eigenvalue and singular value problems on an undersized systolic array," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, Apr. 1986.

[33] J. Gotze, "On the parallel implementation of jacobi and kogbetliantz algorithms," *SIAM Journal on Scientific and Statistical Computing*, vol. 15, no. 6, 1994.

[34] W. Ferzali and J. Proakis, "Adaptive svd algorithm with applications to narrow band signal tracking," *SVD and signal Processing*, vol. 2, 1991.

[35] M. Moonen, P. V. Dooren, and J. Vandewalle, "A systolic array for svd updating," *SIAM Journal on MMatrix Analysis and Applications*, vol. 14, 1993.

[36] M. Moonen, P. V. Dooren, and J. Vandewalle, "An svd updating algorithm for subspace tracking," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, 1992.

[37] S. Qiao and X. Wang, "Computing the singular values of 2-by-2 complex matrices," *McMaster University*, May 2002.

[38] R. Ulrey and H. S. A. Maciejewski, "Parallel algorithms for singular value decomposition," *IEEE*, vol. 6, 1994.

[39] C. Sengupta, J. Cavallaro, and B. Aazhang, "Solving the svd updating problem for subspace tracking on a fixed sized linear array of processors," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 5, Apr. 1997.

[40] G. Forsythe and P. Henrici, "The cyclic jacobi method for computing the principal values of a complex matrix," *Transaction of the American Mathematical Society*, vol. 94, Jan. 1960.

[41] K. Ashenayi, I. Hoballah, and S. Singh, "Application of normal distribution in modeling global irradiation," in *IEEE System Theory Proceedings of the Twentieth Southeastern Symposium*, Mar. 1998.

[42] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes, 4th ed.* McGraw Hill, 2002.

[43] S. Kay, "A fast and accurate single frequency estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, Dec. 1989.

[44] B. K. Pillai, S.U., "Forward/backward spatial smoothing techniques for coherent signal identification," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-37, Jan. 1989.

[45] B. Dongming, Z. Gengxin, and Y. Xinying, "A maximum likelihood based carrier frequency estimation algorithm," in *ICSP2000*, 2000.

[46] R. Niles, "/ statistics / standard deviation." http://www.robertniles.com/stats/stdev.shtml, 2004.

[47] L. Palafox, "Pisarenko harmonic decomposition." http://www.mathworks.com/matlabcentral/fileexchange/lo June 2001.