Robust GM Wiener Filter in the Complex Domain

Matthew Greco Kayrish

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Electrical Engineering

Lamine Mili, Chair
Amir Zaghloul
Joseph (Yue) Wang

November 2, 2012

Keywords: Adaptive Signal Processing, Robust, Wiener Filter,
GM-estimator, Minimum Covariance Determinant

Robust GM Wiener Filter in the Complex Domain

Matthew Kayrish

ABSTRACT

Space-Time Adaptive Processing is a signal processing technique that uses an adaptive array to help remove nonhomogeneous data points from a dataset. Since the early 1970s, STAP has been used in STAP radar systems for their ability to filter clutter, interference and jamming signals. One major flaw with early STAP radar systems is the reliance on non-robust estimators to estimate the noise condition. When even a single outlier is present, the earliest STAP radar systems would break down, causing the target to be missed. Many algorithms have been developed to successfully estimate the noise condition of the dataset when outliers are present. As recently as 2007, a STAP radar processing system based on Adaptive Complex Projection Statistics has been proposed and successfully filters out the noise condition even when outliers are present. However, this algorithm requires the data to be entirely real. Radar data, which consists of amplitude and phase, is complex valued. Therefore, it must be converted into its rectangular components before processing can commence. This introduces many additional processing steps which significantly increase the computing time. The STAP radar algorithm of this thesis overcomes the problems with early radar systems. It is based on the Complex GM Wiener Filter (CGMWF) with the Minimum Covariance Determinant (MCD) for outlier detection. The robustness of the conventional Wiener filter is enhanced by robust Huber Estimator, and using the MCD enables processing entirely in the complex domain. This results in a STAP radar algorithm with a breakdown point of nearly 35% and that enables processing entirely in the complex domain for fewer computing steps.

To Emily, Mom and Dad

Acknowledgements

I would like to express my gratitude to my graduate committee for providing me with advice and direction on my research. I would like to thank Dr. Lamine Mili for providing unending hours of guidance and support throughout this thesis. Without his direction, surmounting this thesis would not have been possible.

I would also like to thank my coworkers at Modern Technology Solutions, Inc. who have always understood when I needed to leave work early to school work. I thank Dr. Timothy Bole, who provided ideas and incredible knowledge when I needed a second opinion or when I was stuck on a problem. I thank Ken McKenzie for being accommodating throughout my research.

I would like to express my gratitude to my parents, Michael and Bernadette Kayrish for providing years and years of encouragement and support. You have been the driving force for many years through continued support and advice. For this, I am very grateful.

I would like to thank my best friends, Bruno and Ruby, for providing a much needed laugh over a drink or while watching football, especially when school work was beginning to stress me out.

Finally, I would like to thank my wife, Emily. You have supported me through every step of this, sometimes waiting for hours at night while I continued burning the midnight oil. You supported me unconditionally through all of this and I am eternally grateful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"We must make the best of those ills which cannot be avoided."

-Alexander Hamilton

1.1 Introduction

In a world of complex and constantly evolving systems, the application of robust signal adaptation and estimation techniques to solve engineering problems has become increasingly popular. From radar and navigation systems to economics, speech and weather forecasting, this field has proven to enhance the capabilities of existing technologies at a fast pace and the progress will continue in years to come. The study of adaptive estimation theory is here to stay.

Some popular algorithms and mathematical methods have a wide variety of applications and are very commonly used in wireless communications, digital communications, power systems, networking and military and space systems. Research and development of future systems depends largely on the ability to efficiently and reliably recover a transmitted signal even if the original signal has been distorted to the point where part of the information it contains has been modified or destroyed. Robustly estimating the original transmission by applying algorithms in the complex domain has become a hot topic of research today.

1.2 The Wiener Filter

One of the foundations to some of these algorithms and mathematical methods is the Wiener filter. The Wiener filter, originally developed by Norbert Wiener in 1940, is a signal-processing technique commonly used for filtering out noise in an input signal. Ideally, The desired outcome is to recover the deterministic component of the noisy signal.

The Wiener filter is commonly utilized because of its simplicity and its speed. It is deemed simple because it uses a system of linear equations to calculate a set of optimal filter weights that reduce the noise level of a received signal. It estimates cross-correlation and covariance matrices of noisy signals to calculate these weights and provide an accurate estimate of the undistorted deterministic signal under Gaussian noise.

1.2.1 Recent Developments of the Wiener Filter

One of the drawbacks of the Wiener filter is that, as a linear filter that is based on the least squares estimator (L_2-norm estimator), it is highly sensitive to outliers. It can be driven to any value under the action of a single outlier, yielding a breakdown point of zero. Thus, it is non-robust. Therefore, some recent research has involved employing new estimators that will enable the Wiener filter to handle outliers with high statistical efficiency under Gaussian noise.

Following Huber [19], Chambers [8] proposed, in 1997, using a mixed norm M-estimator that combines the L_2-norm estimator and the least absolute deviation estimator (LAD), or simply, the L_1-norm estimator. This estimator compares the standardized residuals to a predetermined threshold. The L_2-norm component of the estimator provides high efficiency under Gaussian noise, while the L_1-norm component provides robustness to outliers. This estimator essentially "switches" between the L_1-norm and the L_2-norm. This Wiener solution is more efficient than the L_1-norm or more robust than the L_2-norm employed alone.

This work spawned a number of papers authored by Zou et al. [53]-[61], that focus on robustifying the Wiener filter even further. However, each of these built upon the

previous work of Chambers, and all rely on M-estimation to estimate the deterministic characteristics of the input signal. In 1999, Zou et al. [56] proposed a mixed-norm M-estimator that built on Chambers' proposal. The main difference with the latter approach is that the authors propose to delete the data point when the standardized residual is larger than a given threshold. The authors further employed adaptive estimation techniques by recursively updating the covariance matrix of the input and the cross-correlation of the input with the desired signal and the weight vector. The simulation resulted in the so-called recursive mixed-norm M-estimator (RMN) that converges faster than the original mixed-norm M-estimator.

In linear regression, a leverage point is a data point whose projection on the factor space spanned by the row vectors of the design matrix is an outlier. If the outlier is consistent with the pattern of the collection of data, it is termed a good leverage point. Otherwise, it is a bad leverage point. It is well-known that M-estimators are not robust to bad leverage points. They will breakdown in presence of a single bad leverage point. This is the reason why GM-estimators have been developed to achieve robustness against bad leverage points [15].

In 2007, Schoenig and Mili proposed, in [27] and [44] , using robust GM-estimation in a Space-Time Adaptive Processing (STAP) radar system. The proposed algorithm employed the projection statistics (PS) algorithm developed by Stahel [47] in 1981 and Donoho [11] in 1982 for robust multivariate location and covariance estimation. In this work, Schoenig and Mili proposed an algorithm called Adaptive Complex Projection Statistics (ACPS), which applied the PS algorithm to complex-valued STAP radar data for detecting targets in noisy data. For the training data set, jammers and clutter are filtered out via robust weights calculated via the projection statistics, allowing us to estimate a robust covariance matrix of the noise. Then, a new robust complex-valued GM-Wiener filter is applied to the reweighted data set to detect the targets in the STAP radar data.

1.3 Shortcomings of Prior Work

The work done by Schoenig and Mili was the first successful implementation of GM-estimation on complex-valued data. While it is groundbreaking in the field of STAP, reliance on the PS algorithm increases processing time because it can only operate on radar data in rectangular form, rather than its native polar form. The extra processing steps to convert polar data into its rectangular form increase processing time. Therefore, improvement is needed. In this work, we propose an outlier diagnostics algorithm that will operate on raw complex-valued data in polar form directly, thereby saving computing time.

1.4 Research Approach and Summary

The method designed in the ensuing chapters combines a robust adaptive estimator with the Wiener filter. Specifically, the Wiener filter is robustified using the Huber GM-estimator based on the minimum covariance determinant (MCD) initiated by Rousseeuw [43] for suppressing outliers. Our work extends the proposal made by Schoenig [44] by using the GM-estimator on complex-valued data. This will build on the state-of-the-art techniques in STAP radar processing by eliminating some of the preprocessing required for target detection. This is advantageous because:

1. The MCD can operate on STAP radar data in either rectangular or polar forms, i.e. no preprocessing is required;

2. The processing time required for target detection is reduced;

3. The Wiener Filter can be implemented entirely in the complex domain;

4. The M-estimator and GM-estimator can be implemented entirely in the complex domain;

5. The MCD algorithm requires less processing time than the PS algorithm when the sample size is significantly larger than the dimension and the sample is less than

15% contaminated;

In this work, the MCD is applied to the STAP training data set to determine a set of robust distances (MCD) for each sample in a dataset. Then, a weight factor is calculated based on the robust distance for each sample and each sample is weighted accordingly. Then, the robust GM-estimator is applied to the dataset to calculate a set of Wiener filter weights. The filter weights are applied to the dataset to calculate the robust signal estimate. The contributions include:

1. Using the MCD as an outlier diagnostics tool in signal processing and STAP;
2. Applying the MCD for processing the input data in the complex domain;
3. Combining the Wiener filter with a complex-valued GM-estimator for processing in the complex domain.

The algorithms developed will be discussed in later sections.

1.5 Report Organization

The report is organized as follows. Chapter 2 of this paper is dedicated to classical and robust estimation theory. The principals of robust estimation theory, the GM Huber estimator as well as the MCD algorithm are discussed in more detail. Chapter 3 is dedicated to the Wiener filter. The derivations of the constrained and unconstrained Wiener filter are also provided. Chapter 4 deals with a robust Wiener Filter in the complex domain. This filter combines the GM Huber estimator and the MCD algorithm with the Wiener filter. Some simulation results are provided. Chapter 5 is dedicated to applying the MCD and the constrained robust Wiener filter to MCRAM radar data. Conclusions and future work are provided in Chapter 6.

Chapter 2

Robust Estimation Theory

This chapter presents a brief overview of robust estimation theory. The principles introduced in this chapter are applied to develop an efficient algorithm, which will be presented in Chapters 4 and 5.

In adaptive signal processing, a data set usually contains two components, a deterministic component and an additive noise component. Each measurement in the data set contains values in a given range, representing its deterministic component. An additive noise component is superimposed to the deterministic component, which may result in outliers in the data set. There are two approaches that may deal with outliers, namely the diagnostic approach and the accomodations approach. The former detects then deletes the outliers from the data set. By contrast, the accomodation approach bounds the influence of the outliers via robust weights. We investigate the accomodation approach in this chapter.

2.1 Basic Estimators used in Statistics

Three basic types of estimation problems are the following:

1. Estimation of Location
2. Estimation of Scale
3. Estimation of Scatter

Each estimation problem will be discussed in more detail in the following sections.

2.1.1 Estimators of Location

The estimators of location include those that estimate the location of a data set. Some estimators of location include:

1. Sample Mean
2. Sample Median

Each estimator of location will now be further discussed.

Sample Mean

Suppose, a set of independent, identically distributed (i.i.d) samples are drawn from an unknown random process. The sample mean, or average, of the samples, provides an estimate of the expected value of the random process. The sample mean of a sample set, $\{x_1, x_2, \ldots, x_M\}$ is calculated by

$$\bar{x} = \frac{1}{M} \sum_{i=1}^{M} x_i. \tag{2.1}$$

The advantages of the sample mean include its simplicity and statistical efficiency under Gaussian noise condition. However, the presence of even a single outlier is detrimental to its overall performance. It is not robust and therefore is unreliable. A more robust estimator of location should be utilized.

Sample Median

The sample median of a sample set is found by taking the centermost point of that set. Suppose the sample set, $\{x_1, x_2, \ldots, x_M\}$. To calculate the sample median, assume the sample set has been ordered in increasing order. Next, with M samples, ν is calculated by $\nu = \left[{}^{M}/_{2}\right] + 1$, where $[\bullet]$ is rounding to the next lower integer. Using this, the sample median is calculated by

$$x_{med} = x_\nu \qquad \text{if M is odd,} \tag{2.2}$$

$$x_{med} = \frac{x_\nu + x_{\nu-1}}{2} \qquad \text{if M is even.} \tag{2.3}$$

The sample median is much more reliable than the sample mean when outliers are present. It can handle up to $^{[M-1]}/_2$ outliers in a set of M data points. Due to this robustness, the sample median is used in adaptive signal processing and robust estimation.

2.1.2 Estimators of Scale

Estimators of scale aim to measure the spread of the data. Popular estimators of scale include

1. The sample standard deviation
2. The Median-Absolute-Deviation (MAD)

Each of these estimators will now be discussed further.

The Sample Standard Deviation

The sample standard deviation estimates the standard deviation from the mean of the dataset. It is expressed as

$$\hat{\sigma} = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(x_i - \bar{x})^2}, \tag{2.4}$$

where $\bar{x}$ is the sample mean calculated in (2.1) and M is the number of data points. The advantage of the sample standard deviation is that it is easy to calculate and the best estimate of spread of the sample drawn from a Gaussian distribution. However, it is not robust to outliers. A single outlier is detrimental to its performance.

Table 2.1: Values for $\tilde{b}_M$ for $M < 10$

M	$\tilde{b}_M$
2	1.196
3	1.495
4	1.363
5	1.206
6	1.200
7	1.140
8	1.129
9	1.107

The Median-Absolute-Deviation from the Median

The MAD is used for a sample with outliers or drawn from a thick tailed distribution. It is calculated by

$$MAD = 1.4826\ \tilde{b}_M\ \underset{i}{median}|x_i - x_{med}| \tag{2.5}$$

where M is the sample size, 1.4826 is a correction factor for Fisher consistency and $\tilde{b}_M$ is a bias correction factor at the Gaussian distribution. Using Monte Carlo simulations, Croux and Rousseeuw [42] provided values for $\tilde{b}_M$. They showed that for $M < 10$, $\tilde{b}_M$ is given in Table 2.1.

For $M \geq 10$, $\tilde{b}_M$ is calculated by

$$\tilde{b}_M = \frac{M}{M - 0.8}. \tag{2.6}$$

Like the sample median, the MAD is robust to outliers. Therefore, it is more commonly used in robust estimation and adaptive signal processing.

2.1.3 Estimator of Scatter

The conventional measure of scatter under Gaussianity is given by the sample covariance matrix, $\hat{\mathbf{\Sigma}}_{\mathbf{x}}$. Assume a sample set, $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$, is drawn from a normally distributed M-dimensional random process, $\mathbf{X}$. The sample covariance matrix, $\hat{\mathbf{\Sigma}}_{\mathbf{x}}$, is given by

$$\hat{\mathbf{\Sigma}}_{\mathbf{x}} = \frac{1}{M-1} \sum_{i=1}^{M} (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T , \tag{2.7}$$

where $\bar{\mathbf{x}}$ is the sample mean. It can be shown that this estimator is Fisher consistent in that it tends to the true covariance matrix of the data set as the sample size increases to infinity.

2.2 Classical Estimation Theory

Classical estimation theory was first introduced by Fisher [10] in the early 1900s. He introduced the maximum likelihood estimation theory, which is described next.

2.2.1 Maximum Likelihood Estimators of Location

The Maximum Likelihood (ML) theory requires the *a priori* knowledge of the probability distribution of the observations. Suppose a set of M realizations, $\{z_1, z_2, \ldots, z_M\}$, of a random variable, z, follow a probabily distribution with probabily density function

$$f(z;\theta) = f(z-\theta) , \tag{2.8}$$

where θ is the parameter of location to be estimated. When viewed as a function of θ given z, $f(z;\theta)$ becomes the likelihood function up to a constant. Defining the residual as

$$r = z - \theta \tag{2.9}$$

the likelihood function can be rewritten as

$$f(z-\theta) = f(r) . \tag{2.10}$$

The ML estimator of location, $\hat{\theta}_{ML}$, is defined as

$$\hat{\theta}_{ML} = \underset{\theta}{argmin} \sum_{i=1}^{M} -\ln f(r_i) \tag{2.11}$$

where $-\ln f(r_i)$ is replaced with $\rho(r_i)$, that is,

$$-\ln f(r) = \rho(r). \tag{2.12}$$

The function

$$J(\theta) = \sum_{i=1}^{M} -\ln f(r_i) = \sum_{i=1}^{M} \rho(r_i) \tag{2.13}$$

is the objective function to be minimized. The ML estimator is obtained by setting the derivative of the objective function with respect to θ equal to zero. That is,

$$\frac{dJ(\theta)}{d\theta} = \frac{d}{d\theta} \sum_{i=1}^{M} \rho(r_i) = \sum_{i=1}^{M} \frac{d\rho(r_i)}{dr_i} \frac{dr_i}{d\theta} = -\sum_{i=1}^{M} \frac{d\rho(r_i)}{dr_i} = 0. \tag{2.14}$$

The above equation can be rewritten with (2.12) substituted back in for $\rho(r_i)$, which yields

$$\frac{dJ(\theta)}{d\theta} = -\sum_{i=1}^{M} \frac{d\rho(r_i)}{dr_i} = -\sum_{i=1}^{M} \frac{-d\ln f(r_i)}{dr_i}. \tag{2.15}$$

The effect a residual has on the estimator is measured by

$$\psi(r) = \frac{d\rho(r)}{dr} = \frac{-d\ln f(r)}{dr} = -\frac{f'(r)}{f(r)}. \tag{2.16}$$

where $-\psi(r)$ is called the *score* function. Substituting (2.16) into (2.14) yields

$$\frac{dJ(\theta)}{d\theta} = -\sum_{i=1}^{M} \frac{d\rho(r_i)}{dr_i} = -\sum_{i=1}^{M} \psi(r_i) = 0. \tag{2.17}$$

When the ψ-function is nonlinear, a solution of (2.17) may be obtained via the Iteratively Reweighted Least Squares (IRLS) algorithm. It is derived by dividing and multiplying the ψ-function by r, yielding

$$\sum_{i=1}^{M} q\left(r_i\right) r_i = 0 \tag{2.18}$$

where

$$q\left(r\right) = \frac{\psi\left(r\right)}{r}. \tag{2.19}$$

is called the weight function. Hence, we get

$$\theta^{(k+1)} = \frac{\sum_{i=1}^{M} q\left(r_i^{(k)}\right) z_i}{\sum_{i=1}^{M} q\left(r_i^{(k)}\right)}. \tag{2.20}$$

2.2.2 Gaussian Likelihood Function

This section assumes that the observation errors follow a Gaussian distribution with $\sigma = 1$. That is, (2.8) is a Gaussian bell curve given by

$$f\left(z-\theta\right) = \frac{1}{\sqrt{2\pi}} e^{\frac{(z-\theta)^2}{2}}. \tag{2.21}$$

By substituting (2.21) into (2.11) for $f\left(r_i\right)$ and minimizing over a sample set, $\{z_1, z_2, \ldots, z_m\}$, the ML estimator becomes

$$\hat{\theta}_{ML} = \underset{\theta}{argmin} \sum_{i=1}^{m} \rho\left(r_i\right). \tag{2.22}$$

where

$$\rho\left(r\right) = -\ln \frac{1}{\sqrt{2\pi}} e^{\frac{r^2}{2}} \tag{2.23}$$

$$= -\left(ln1 - ln\sqrt{2\pi} - \frac{r^2}{2}\right) \tag{2.24}$$

$$= ln\sqrt{2\pi} + \frac{r^2}{2}, \tag{2.25}$$

the ψ-function is

$$\psi(r) = \frac{d\rho(r)}{dr} \tag{2.26}$$

$$= \frac{d}{dr}\left(ln\sqrt{2\pi} + \frac{r^2}{2}\right) \tag{2.27}$$

$$= r, \tag{2.28}$$

and the weight function is

$$q(r) = \frac{\psi(r)}{r} \tag{2.29}$$

$$= \frac{r}{r} \tag{2.30}$$

$$= 1, \tag{2.31}$$

which yields

$$\hat{\theta}_{ML} = \frac{1}{M}\sum_{i=1}^{M} z_i. \tag{2.32}$$

This shows that the sample mean is the ML-estimator at the Gaussian distribution and therefore, it is the most efficient estimator for data following a Gaussian distribution.

2.3 Robust Estimation Theory

This section covers robust estimation theory. In classical ML-estimation theory, some characteristics of the probability distribution of an observed data set are either assumed or known *a priori*. However, the data may be contaminated with samples that follow an unknown probability distribution. In this case, the ML-estimators are no longer optimal and may provide unreliable estimates.

Robust estimation theory strives to overcome these shortcomings. Huber initiated this theory in 1964 [19] and developed a general class of maximum-likelihood-type estimators, called the M-estimators for short.

Following the work of Huber, Hampel proposed several key concepts that are used to measure the robustness of an estimator [14]. These concepts include the *breakdown point* and the *influence function*.

2.3.1 Robustness Concepts

The Breakdown Point

The breakdown point of an estimator measures the percentage of contamination it can handle before breaking down. The percent of contamination is represented by

$$\epsilon = m_c/m \tag{2.33}$$

where m_c is the number of outliers and m is the sample size. The bias of the contaminated data is measured by

$$b_{max} = \sup|\hat{\theta}_m - \hat{\theta}'_m|, \tag{2.34}$$

where $\hat{\theta}_m$ is the estimator applied to a good data set and $\hat{\theta}'_m$ is the estimator applied to an ϵ-contaminated data set. The breakdown point is the maximum fraction of contamination that yields a finite bias. That is

$$\epsilon^* = max\left\{\epsilon = \frac{m_c}{m};\ \ b_{max} \neq \infty\right\}. \tag{2.35}$$

The breakdown point measures the global robustness of an estimator.

The Empirical Influence Function

The empirical influence function measures the influence that a single outlier has on an estimator. It is defined as

$$IF\left(x;\hat{\theta}_m,F\right)=\left[\hat{\theta}_m\left(x_1,x_2,\ldots,x_{m-1},x\right)-\hat{\theta}_{m-1}\left(x_1,x_2,\ldots,x_{m-1}\right)\right], \tag{2.36}$$

where x can take an arbitrary value on the real line. The empirical influence function measures the sensitivity of an estimator to one additional outlier. It also measures its local robustness.

2.3.2 Linear Regression Analysis

This section reviews the basic principles of linear regression analysis. Consider the linear regression model given by

$$\mathbf{z}=\mathbf{X}\boldsymbol{\alpha}+\mathbf{e} \tag{2.37}$$

where $\boldsymbol{\alpha}$ is the true parameter vector of dimension N, $\mathbf{e}$ is the additive error vector of dimension M, $\mathbf{z}$ is the observation vector of dimension M, and $\mathbf{X}$ is an $M\times N$ dimensional matrix. The purpose is to estimate $\boldsymbol{\alpha}$ given $\mathbf{z}$ and $\mathbf{X}$. When $\boldsymbol{\alpha}$ is unknown, we define the residual vector as

$$\mathbf{r}=\mathbf{z}-\mathbf{X}\boldsymbol{\alpha}. \tag{2.38}$$

From this, (2.38) can be written in expanded form as

$$r_1 = z_1 - \alpha_1 x_{1,1} - \alpha_2 x_{1,2} - \cdots - \alpha_N x_{2,N} \tag{2.39}$$

$$r_2 = z_2 - \alpha_1 x_{2,1} - \alpha_2 x_{2,2} - \cdots - \alpha_N x_{2,N} \tag{2.40}$$

$$\vdots \tag{2.41}$$

$$r_M = z_M - \alpha_1 x_{M,1} - \alpha_2 x_{M,2} - \cdots - \alpha_N x_{M,N}. \tag{2.42}$$

In a compact form, we can write

$$r_i = z_i - \mathbf{x}_i^T \boldsymbol{\alpha} \quad \text{for} \quad i = 1, 2, \ldots, M. \tag{2.43}$$

A least-squares estimator is defined as

$$\hat{\boldsymbol{\alpha}}_{LS} = \underset{\boldsymbol{\alpha}}{argmin} \sum_{i=1}^{M} \left(z_i - \mathbf{x}_i^T \boldsymbol{\alpha} \right)^2 \tag{2.44}$$

$$= \underset{\boldsymbol{\alpha}}{argmin} \sum_{i=1}^{M} r_i^2 \tag{2.45}$$

To analyze the statistical properties of this estimator, we rely on estimation theory, which is covered in the following sections.

2.3.3 M-Estimators in Linear Regression

The definition of the M-estimator is given by

$$\hat{\boldsymbol{\alpha}}_M = \underset{\boldsymbol{\alpha}}{argmin}\, J(\boldsymbol{\alpha}) = \underset{\boldsymbol{\alpha}}{argmin} \sum_{i=1}^{m} \rho\left(\frac{r_i}{s} \right) \tag{2.46}$$

where s is a scaling factor. Minimizing (2.46) involves taking the partial derivitives of $J(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ and setting them equal to zero, yielding

$$\frac{\partial J(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = \sum_{i=1}^{m} \frac{1}{s} \frac{\partial \rho(r_{s,i})}{\partial (r_{s,i})} \left(\frac{\partial r_{s,i}}{\partial \boldsymbol{\alpha}} \right)^T = \sum_{i=1}^{m} \psi(r_{s,i}) \frac{1}{s} \mathbf{x}_i = 0, \tag{2.47}$$

where s is a scaling factor and $r_{s,i} = \frac{r_i}{s}$ is the standardized residual. For convenience, we assume $s = 1$, so (2.47) reduces to

$$\frac{\partial J(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = \sum_{i=1}^{m} \frac{\partial \rho(r_i)}{\partial (r_i)} \left(\frac{\partial r_i}{\partial \boldsymbol{\alpha}} \right)^T = \sum_{i=1}^{m} \psi(r_i) \mathbf{x}_i = 0. \tag{2.48}$$

.

(2.48) is similar to (2.14). By this, the ML-estimators are considered special cases of the M-estimators.

The Huber estimator is the estimator of choice for this thesis. The *kernel function* of the Huber estimator is defined as

$$\rho(r_s) = \begin{cases} \frac{1}{2} r_s^2 & \text{if } |r_s| \leq b \\ b|r_s| - \frac{1}{2} b^2 & \text{if } |r_s| > b \end{cases}. \tag{2.49}$$

The *yielding function* is defined as

$$\psi(r_s) = \begin{cases} r_s & \text{if } |r_s| \leq b \\ b \operatorname{sign}(r_s) & \text{if } |r_s| > b \end{cases}, \tag{2.50}$$

and the weight function is defined as

$$q(r_s) = \begin{cases} 1 & \text{if } |r_s| \leq b \\ \frac{b}{r_s} \operatorname{sign}(r_s) & \text{if } |r_s| > b \end{cases}. \tag{2.51}$$

The kernel function, the score function, and the weight function are shown in Figs. 2.1 through 2.3, where $b = 2$.

The Huber estimator is robust to vertical outliers. However, like all M-estimators, the Huber estimator is not robust to *bad leverage points*. For these outliers, we rely on the GM-estimator, which is explained in the following section.

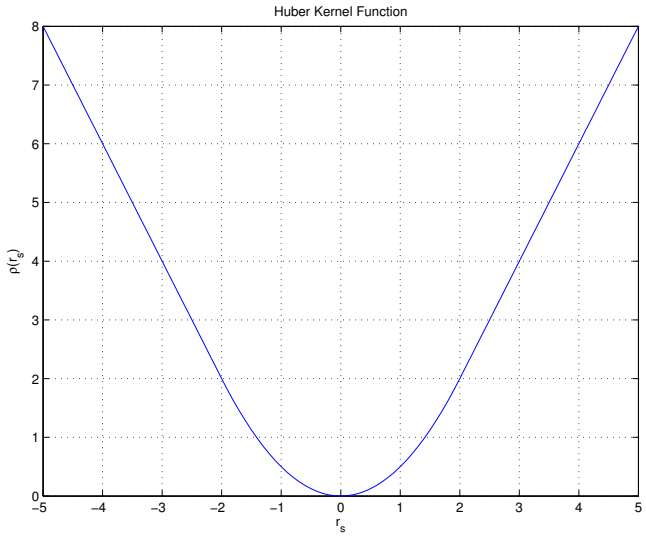

Figure 2.1: Huber ρ-Function

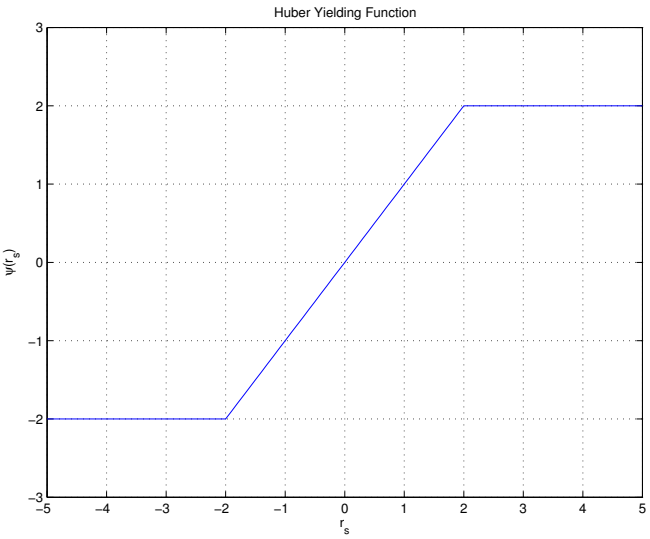

Figure 2.2: Huber ψ-Function

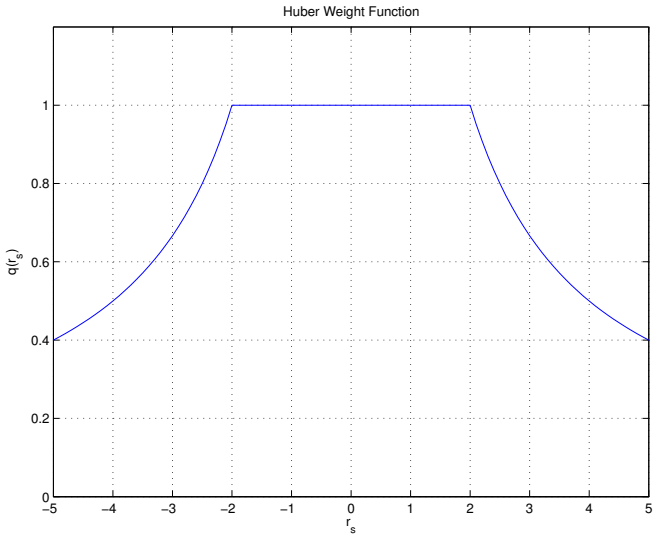

Figure 2.3: Huber q-Function

2.3.4 GM-Estimators

As previously mentioned, there are two types of leverage points, *good* leverage points and *bad* leverage points. A good leverage point is displayed in Fig. 2.4. The good leverage points follow the pattern of the main cluster of the data. They contribute to an improved accuracy of the estimator. A bad leverage point is displayed in Fig. 2.5. Bad leverage points do not follow the pattern of the main cluster. Consequently, they contribute to a reduced accuracy of the estimator.

A bad leverage point has significant influence of position on the M-estimators. GM-estimators bound this influence via a weight factor that depends on the position of the data point in the factor space. The definition of GM-estimators is given by

$$\hat{\boldsymbol{\alpha}}_{GM} = \underset{\boldsymbol{\alpha}}{argmin}\ J(\boldsymbol{\alpha}) = \underset{\boldsymbol{\alpha}}{argmin} \sum_{i=1}^{m} w^2(\mathbf{x}_i)\, \rho\left(\frac{r_i}{s\, w(\mathbf{x}_i)}\right). \tag{2.52}$$

The objective function for GM-estimation is minimized by taking the derivative with respect to $\boldsymbol{\alpha}$ and setting equal to zero. This is given by

$$\frac{\partial J(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = \sum_{i=1}^{m} \frac{w(\mathbf{x}_i)}{s} \frac{\partial \rho\left(\frac{r_i}{s\, w(\mathbf{x}_i)}\right)}{\partial\left(\frac{r_i}{s\, w(\mathbf{x}_i)}\right)} \left(\frac{\partial r_i}{\partial \boldsymbol{\alpha}}\right)^T = \sum_{i=1}^{m} \frac{w(\mathbf{x}_i)}{s} \mathbf{x}_i \psi\left(\frac{r_i}{s\, w(\mathbf{x}_i)}\right) = \mathbf{0}. \tag{2.53}$$

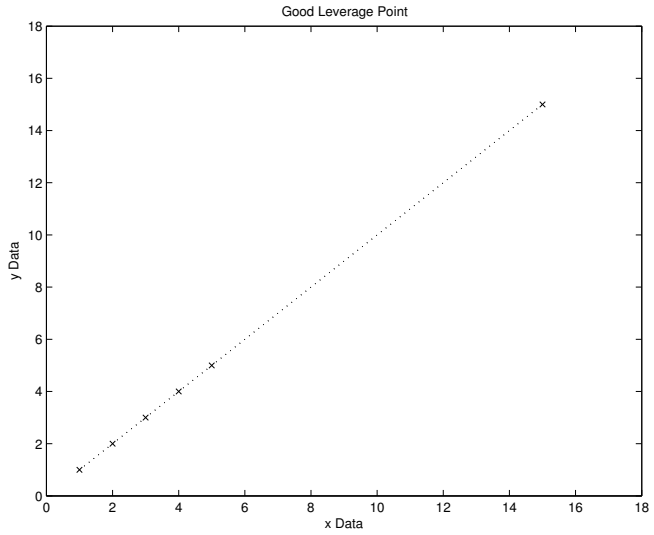

Figure 2.4: In this figure, the good leverage point is the point to the top right corner that is not near the bulk of data.

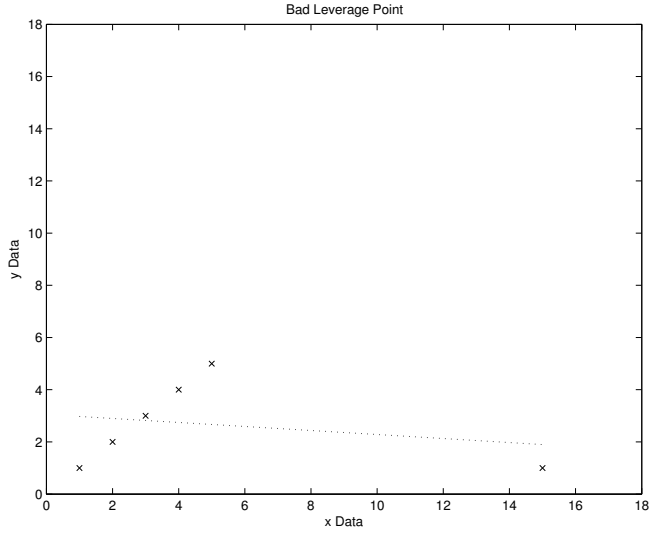

Figure 2.5: In this figure, the bad leverage point is the point to the bottom right corner that is not near the bulk of data.

The weight factor, $w(\mathbf{x}_i)$, of the observation point, $\mathbf{x}_i$ is given by

$$w(\mathbf{x}_i) = \min\left(1, \frac{b^2}{MCD_i}\right) \tag{2.54}$$

where MCD_i^2 is the robust distance of the $i-th$ observation and b is the $97.5th$ percentile of the χ_N^2 distribution with N degrees of freedom (DOF). A weight is calculated by (2.54) and is applied to the outlying observation data points, as shown in 2.53, thereby limiting their influence on the GM-estimate.

2.4 Minimum Covariance Determinant (MCD)

The MCD was first proposed by Rousseeuw and Van Driessen [43] in 1999 as an alternative to the minimum volume ellipsoid (MVD). The MCD is the minimum determinant of the covariance matrix of a subset of data vectors from a dataset, hence, its name. Because its breakdown point approaches 50% and the existance of a fast algorithm to implement it [43], the MCD is the outlier diagnostic tool used in this thesis.

Consider a dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M]$ of N dimensions. The basic steps involved in finding the MCD are:

1. Select a subset, $H_1 \subset \{1, \ldots, M\}$ where H_1 contains $h > N + 1$ data vectors. h must be greater than $N + 1$ to avoid a singular covariance matrix of the subset;
2. Compute the mean and covariance matrix of the subset H_1;
3. Compute and store the Mahalanobis distances, $MD(\mathbf{x}_i)$, of each data vector from the entire data set;
4. Sort the Mahalanobis distances from smallest to largest;
5. Select the vectors having the smallest h distances to create a new subset H_2;
6. Return to step 2.

The details of the algorithm is as follows. Consider a subset $H_1 \subset \{1, \ldots, M\}$, where H_1 contains a random combination of h unique numbers between 1 and M. From this subset, the sample mean is calculated by

$$\mathbf{T}_1 = \frac{1}{h} \sum_{i \in H_1} \mathbf{x}_i \tag{2.55}$$

and the sample covariance matrix of the subset is calculated by

$$\mathbf{\Sigma}_1 = \frac{1}{h} \sum_{i \in H_1} (\mathbf{x}_i - \mathbf{T}_1) (\mathbf{x}_i - \mathbf{T}_1)^T . \tag{2.56}$$

If $\det(\mathbf{\Sigma}_1) \neq 0$, than the Mahalanobis distance for each of the data points from the center of the subset of data is calculated by

$$MD_1(i) = \sqrt{(\mathbf{x}_i - \mathbf{T}_1)^T \mathbf{\Sigma}_1^{-1} (\mathbf{x}_i - \mathbf{T}_1)} \;\; \forall \;\; i = 1, 2, \ldots, M. \tag{2.57}$$

Once all of the Mahalanobis distances are calculated, they must be sorted from least to greatest as

$$MD_1(\mathbf{x}_i)_{1:M} \leq MD_1(\mathbf{x}_i)_{2:M} \leq \ldots \leq MD_1(\mathbf{x}_i)_{M:M} . \tag{2.58}$$

At this point, the new subset, H_2, is generated from the data points having the smallest Mahalanobis distance from the initial cluster, then steps 2 thru 5 are repeated. Rousseeuw refers to this entire process as a C-Step. The C-Steps should be repeated as many times as necessary until the MCD is definitively met. The stopping condition occurs when the determinants are either changing very slightly, or not at all. That is, $\det(\mathbf{\Sigma}_{n+1}) \simeq \det(\mathbf{\Sigma}_n)$. Finding the MCD will occur rather rapidly, usually after 3 iterations and in some cases, after only 2 iterations. The goal is to run the entire process enough times to guarantee that the MCD is reached.

Referring to step 1, smaller subsets are more likely to have all good data points. This is desired because a clean subset guarantees that the MCD is attained. Conversely, setting

h too large increases the probability of obtaining a contaminated subset. Therefore, h should be set near, but not less than $N + 2$ to avoid a singular covariance matrix while maximizing the probability of selecting a clean subset.

Using the MCD algorithm provides numerous advantages. Most notably, the MCD is very robust. Its breakdown point is equal to the breakdown point of the Minimum Volume Ellipsoid [40], which is

$$\epsilon_M^* = \frac{\left[{}^{M}/_{2}\right] - N - 1}{M}, \tag{2.59}$$

where M is the total number of data points and N is the dimension of the data points. As $M \to \infty$, $\epsilon_M^* \to 50\%$.

To use the MCD for outlier detection, enough subsets of data are needed to increase the probability of finding at least one entirely clean subset. The probability of finding a clean subset, as stated in [43], is calculated by

$$P_{CleanSubset}\,(m, N, \epsilon) = 1 - \left(1 - \left(1 - \epsilon\right)^{N+1}\right)^{m} > 0 \tag{2.60}$$

where N is the dimension of the data, m is the number of subsets and ϵ is the fraction of contamination. This inequality breaks down to

$$1 > \left(1 - \left(1 - \epsilon\right)^{N+1}\right)^{m} \tag{2.61}$$

$$ln(1) > ln\Big(\left(1 - \left(1 - \epsilon\right)^{N+1}\right)^{m}\Big) = m\ ln\left(1 - \left(1 - \epsilon\right)^{N+1}\right) \tag{2.62}$$

$$0 > m\ ln\left(1 - \left(1 - \epsilon\right)^{N+1}\right). \tag{2.63}$$

By placing upper and lower bounds on the quantity $\left(1 - (1 - \epsilon)^{N+1}\right)$ and because $ln(0) = -\infty$ and $ln(1) = 0$, this shows that

Table 2.2: Metrics and Statistics for Finding a Clean Subset Using the MCD

m	N	ϵ	$\epsilon\, N$	$(1-\epsilon)^{N+1}$	$P_{CleanSubset}$
20	2	0.1	0.2	0.729	1.0
20	2	0.25	0.5	0.4219	1.0
20	2	0.5	1.0	0.125	0.9308
20	10	0.1	1.0	0.3138	0.9995
20	10	0.25	2.5	0.0422	0.5781
20	10	0.5	5	0.0005	0.0097
20	25	0.1	2.5	0.0646	0.7371
20	25	0.25	6.25	0.0006	0.0112
20	25	0.5	12.5	0.0	0.0
100	2	0.1	0.2	0.729	1.0
100	2	0.25	0.5	0.4219	1.0
100	2	0.5	1.0	0.125	1.0
100	10	0.1	1.0	0.3138	1.0
100	10	0.25	2.5	0.0422	0.9866
100	10	0.5	5	0.0005	0.0477
100	25	0.1	2.5	0.0646	0.9987
100	25	0.25	6.25	0.0006	0.0549
100	25	0.5	12.5	0.0	0.0

$$0 < 1-(1-\epsilon)^{N+1} < 1 \tag{2.64}$$

$$-1 < -(1-\epsilon)^{N+1} < 0 \tag{2.65}$$

$$0 < (1-\epsilon)^{N+1} < 1. \tag{2.66}$$

Maximizing (2.66) will maximize (2.60). This occurs by simultaneously minimizing both N and ϵ. Furthermore, m should also be as large as possible subject to computation time. Table 2.2 shows the probability of finding a clean subset increases as m increases from 20 to 100. Table 2.2 also shows that $P_{CleanSubset}$ significantly increases with decreasing ϵ and N. Fig. 2.6 displays the linear relationship between $P_{CleanSubset}$ and the product of ϵ and N.

Another advantage of the MCD is its computation time. This is for of several reasons. First, the MCD is usually reached after very few C-steps. Therefore, each subset is processed quickly. Second, for a fixed number of dimensions, N, each C-step only takes O (M) calculations. This is because after convergence, sorting the Mahalanobis distances

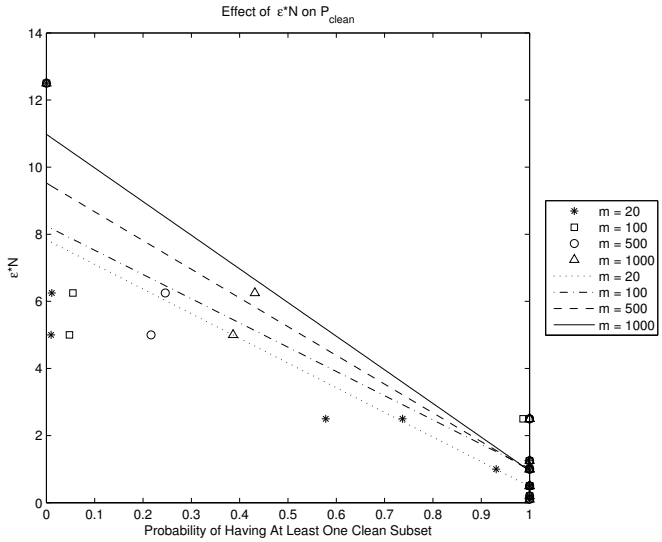

Figure 2.6: Product of ϵN vs. $P_{CleanSubset}$

takes less time. Lastly, the MCD operates on both polar and rectangular forms of complex data. This eliminates extra clock cycles required to convert from polar form to rectangular form.

Another way to decrease the computation time of the MCD is to partition the data set into smaller subsets before processing. Using only a portion of a very large dataset will produce the same MCD as using the entire dataset. In 1999, Rousseeuw proposed the FAST-MCD that was based on this principle [43].

Chapter 3

The Wiener Filter

3.1 The Basics of Stochastic Signals and Systems

This section reviews a few principles of stochastic signals that are used to derive the optimal Wiener solution.

3.1.1 Stationarity and Ergodicity

Stationarity

A complex-valued random process, $\boldsymbol{X}$, with vector $\mathbf{x}$, is said to be stationary in the wide sense if the first and second moments are time invariant. That is, the mean, $\boldsymbol{\mu}_{\mathbf{x}}$ is a constant and the autocorrelation is a function of time lag. This means

$$\gamma(i,j) = \gamma(j-i) \quad \forall \quad i,j = 1,2,\ldots. \tag{3.1}$$

If $\boldsymbol{\mu}_{\mathbf{x}}$ is equal to the zero vector, the covariance matrix, $\boldsymbol{\Sigma}_{\mathbf{x}}$, is a Hermitian matrix. That is

$$\boldsymbol{\Sigma}_{\mathbf{x}} = \boldsymbol{\Sigma}_{\mathbf{x}}^{H}, \tag{3.2}$$

where $(\bullet)^{H}$ is the Hermitian transpose. Wide sense stationary is assumed for this thesis.

Ergodicity

Ergodicity, in a statistical sense, is a property that states the time average of a random process is equal to the ensemble average. In particular, we have

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \mathbf{x}_t = \boldsymbol{\mu}_{\mathbf{x}}. \tag{3.3}$$

Ergodicity is assumed for this thesis.

3.1.2 The Mean of a Random Process

Let $\boldsymbol{X}$ be a random process and $\mathbf{x}$ be a vector of dimension, N, drawn from $\boldsymbol{X}$, given by

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}. \tag{3.4}$$

Assuming that $\boldsymbol{X}$ is an ergodic random process, the mean, or the expected value of the vector $\mathbf{x}$ is calculated by

$$\boldsymbol{\mu}_{\mathbf{x}} = E[\mathbf{x}] = \begin{bmatrix} E[x_1] \\ E[x_2] \\ \vdots \\ E[x_N] \end{bmatrix}. \tag{3.5}$$

Here, E[•] denotes the expectation operator.

3.1.3 The Cross-Correlation and Autocorrelation of a Random Process

The autocorrelation of a random process measures the degree it correlates with itself. The autocorrelation of the vector, $\mathbf{x}$, is defined as

$$\mathbf{\Gamma_x} = E\left[\mathbf{x}\mathbf{x}^H\right]. \tag{3.6}$$

The elements of $\mathbf{\Gamma_x}$ are given by

$$\gamma\left(i,j\right) = E[x_i x_j^*] \quad \forall \quad i,j = 1,2,\ldots,N. \tag{3.7}$$

where $(\bullet)^*$ denotes the complex conjugate operator. $\gamma(i,j)$ is the autocorrelation coefficient of the random variables x_i and x_j, and the autocorrelation matrix is given by

$$\mathbf{\Gamma_x} = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,N} \\ \gamma_{2,1} & \gamma_{2,2} & \cdots & \gamma_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{N,1} & \gamma_{N,2} & \cdots & \gamma_{N,N} \end{bmatrix}. \tag{3.8}$$

When the values of $\mathbf{x}$ represent a time sequence, and $\boldsymbol{X}$ is a stationary process, the values of the autocorrelation matrix can be rewritten as a function of a time lag. That is,

$$\mathbf{\Gamma_x} = \begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{N-2} & \gamma_{N-1} \\ \gamma_{-1} & \gamma_0 & \gamma_1 & \ddots & \gamma_{N-2} \\ \vdots & \gamma_{-1} & \ddots & \ddots & \vdots \\ \gamma_{2-N} & \ddots & \ddots & \gamma_0 & \gamma_1 \\ \gamma_{1-N} & \gamma_{2-N} & \cdots & \gamma_{-1} & \gamma_0 \end{bmatrix}. \tag{3.9}$$

The cross-correlation (correlation hereafter) between two random processes represents the degree of linear dependency between them. Suppose a second random process, $\boldsymbol{Y}$, with a second vector, $\mathbf{y}$. The correlation between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined as

$$\mathbf{\Gamma_{xy}} = E\left[\mathbf{x}\mathbf{y}^H\right]. \tag{3.10}$$

Once again, the correlation coefficients of $\mathbf{\Gamma_{xy}}$ can be calculated by

$$\gamma\left(i,j\right) = E[x_i y_j^*] \quad \forall \quad i,j = 1,2,\ldots,N. \tag{3.11}$$

3.1.4 The Covariance and Autocovariance of a Random Process

The vector $\mathbf{x}$ and the mean $\boldsymbol{\mu}_\mathbf{x}$ are used to calculate the autocovariance matrix by

$$\mathbf{\Sigma_x} = E\left[\left(\mathbf{x}-\boldsymbol{\mu}_\mathbf{x}\right)\left(\mathbf{x}-\boldsymbol{\mu}_\mathbf{x}\right)^H\right] \tag{3.12}$$

$$= E[\mathbf{x}\mathbf{x}^H - \mathbf{x}\boldsymbol{\mu}_\mathbf{x}^H - \boldsymbol{\mu}_\mathbf{x}\mathbf{x}^H + \boldsymbol{\mu}_\mathbf{x}\boldsymbol{\mu}_\mathbf{x}^H] \tag{3.13}$$

$$= \mathbf{\Gamma_x} + \boldsymbol{\mu}_\mathbf{x}\boldsymbol{\mu}_\mathbf{x}^H. \tag{3.14}$$

If $\boldsymbol{\mu}_\mathbf{x}$ is equal to the zero vector, than the autocovariance matrix reduces to the autocorrelation matrix. That is

$$\mathbf{\Sigma_x} = \mathbf{\Gamma_x}. \tag{3.15}$$

The resultant $N \times N$ autocovariance matrix for $\mathbf{x}$ is represented by

$$\mathbf{\Sigma_x} = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,N} \\ \sigma_{2,1} & \sigma_{2,2} & \cdots & \sigma_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N,1} & \sigma_{N,2} & \cdots & \sigma_{N,N} \end{bmatrix}. \tag{3.16}$$

The covariance between random processes $\boldsymbol{X}$ and $\boldsymbol{Y}$ is calculated by

$$\boldsymbol{\Sigma}_{\mathbf{xy}} = E[(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}})^H] \tag{3.17}$$

$$= E[\mathbf{x}\mathbf{y}^H - \mathbf{x}\boldsymbol{\mu}_{\mathbf{y}}^H - \boldsymbol{\mu}_{\mathbf{x}}\mathbf{y}^H + \boldsymbol{\mu}_{\mathbf{x}}\boldsymbol{\mu}_{\mathbf{y}}^H] \tag{3.18}$$

$$= \boldsymbol{\Gamma}_{\mathbf{xy}} + \boldsymbol{\mu}_{\mathbf{x}}\boldsymbol{\mu}_{\mathbf{y}}^H. \tag{3.19}$$

If $\boldsymbol{\mu}_{\mathbf{x}}$ and $\boldsymbol{\mu}_{\mathbf{y}}$ are equal to the zero vector, then the covariance matrix will reduce to the cross-correlation matrix. That is

$$\boldsymbol{\Sigma}_{\mathbf{xy}} = \boldsymbol{\Gamma}_{\mathbf{xy}}. \tag{3.20}$$

3.2 The Wiener Filter

3.2.1 Background of the Wiener Filter

The Wiener filter was developed in 1940, by Norbert Wiener, for filtering the noise in a signal by minimizing the error between the preprocessed input signal and the desired signal. It uses a least squares (LS) estimator, also known as the L_2-norm estimator. The noise statistics are estimated and then used to determine a set of optimal filter weights. By then processing a new input signal, containing similar noise characteristics, with the optimal filter weights, the signal deterministic component is estimated. This method is optimal when the noise distribution is Gaussian. Furthermore, its execution only requires a few computational steps that are very fast to process.

There are two versions of the Wiener filter. Both versions are fully explained in [16]. The first version is the unconstrained Wiener filter. It is called so because it estimates a desired signal without additional constraints. Fig. 3.1 displays the unconstrained Wiener filter.

The objective of the unconstrained Wiener filter is to minimize the mean-square error between the input signal, *y(n)*, and the desired signal, *d(n)*. To do this, the Wiener filter relies on the Principle of Orthogonality, discussed in detail in section 3.3.1, to determine

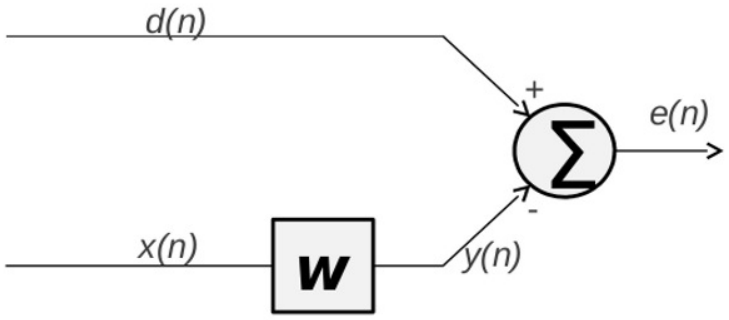

Figure 3.1: Unconstrained Wiener Filter, similar to the unconstrained Wiener Filter given in [16].

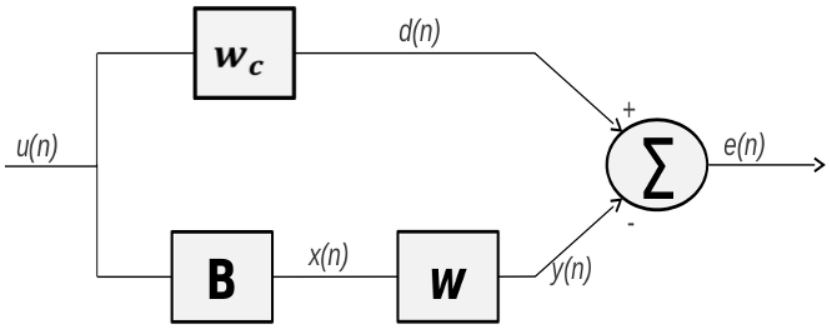

Figure 3.2: Constrained Wiener Filter, similar to the constrained Wiener Filter given in [16].

a set of weights to estimate *d(n)*. The unconstrained Wiener filter is discussed in further detail in sections 3.2.3.

The second version is the constrained Wiener filter. This version is called so because it minimizes the noise condition subject to some predetermined constraint(s). Fig. 3.2 shows the constrained Wiener filter.

In the constrained Wiener filter, the desired response is generated by creating a synthetic response from the input signal. The constrained Wiener filter is discussed in more detail in section 3.2.4.

3.2.2 The Principle of Orthogonality

The Principle of Orthogonality (PO) requires that the dot-product of the response vector, *t(n)*, and error vector, *e(n)*, is equal to 0; hence, they are orthogonal. That is,

$$E[y^*(n-k)e(n)] = 0 \ \ \forall \ \ k, n = 0, 1, 2, \ldots \tag{3.21}$$

Now,suppose an input time series $x(n)$ where $n = 0, 1, 2, \ldots$. Refering to Fig. 3.1, $y(n)$ is calculated by

$$y(n) = \sum_{k=0}^{\infty} w_k^* x(n-k) \ \ \forall \ \ n = 0, 1, 2, \ldots \tag{3.22}$$

or more simply

$$y\left(n\right) = \mathbf{w}^H \mathbf{x}_n \ \ \forall \ \ n = 0, 1, 2, \ldots. \tag{3.23}$$

where $\left(\bullet\right)_n$ is the *n-th* sample from random process $\boldsymbol{X}$. The error is calculated by

$$e(n) = d(n) - y(n) \tag{3.24}$$

$$= d(n) - \mathbf{w}^H \mathbf{x}_n \ \ \forall \ \ n = 0, 1, 2, \ldots. \tag{3.25}$$

Here, $e\left(n\right)$ is minimized when $d\left(n\right) = \mathbf{w}^H \mathbf{x}_n$, which occurs when the filter weights, $\mathbf{w}$, are optimized. Now, the cost function to be minimized is given by

$$J_n\left(\mathbf{w}\right) = E\left[e(n)e^*(n)\right] \tag{3.26}$$

$$= E[|e^2(n)|]. \tag{3.27}$$

In the complex domain, both the real and imaginary parts of the weights must be optimized. The weight vector is defined as

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} a_0 + \jmath b_0 \\ a_1 + \jmath b_1 \\ \vdots \end{bmatrix} = \mathbf{a} + \jmath \mathbf{b}. \tag{3.28}$$

The gradient of $J(\mathbf{w})$ is calculated with respect to a_k and b_k, that is

$$\nabla_{w_k} J = \frac{\partial J(\mathbf{w})}{\partial a_k} + \jmath \frac{\partial J(\mathbf{w})}{\partial b_k} \quad \forall \quad k = 0, 1, 2, \ldots. \tag{3.29}$$

Taking the gradient of (3.26) with respect to $\mathbf{w}$ yields

$$\nabla_{\mathbf{w}} J_n(\mathbf{w}) = E\left[\frac{de^*(n)}{d\mathbf{w}} e(n) + \frac{de(n)}{d\mathbf{w}} e^*(n)\right]. \tag{3.30}$$

Substituting (3.28) into (3.30) for $\mathbf{w}$ yields

$$\nabla_{\mathbf{w}} J_n(\mathbf{a} + \jmath\mathbf{b}) = E\left[\frac{\partial e^*(n)}{\partial \mathbf{a}} e(n) + \frac{\partial e^*(n)}{\partial \mathbf{b}} \jmath e(n) + \frac{\partial e(n)}{\partial \mathbf{a}} e^*(n) + \frac{\partial e(n)}{\partial \mathbf{b}} \jmath e^*(n)\right] \tag{3.31}$$

$$= E\left[\left(\frac{\partial e^*(n)}{\partial \mathbf{a}} + \jmath \frac{\partial e^*(n)}{\partial \mathbf{b}}\right) e(n) + \left(\frac{\partial e(n)}{\partial \mathbf{a}} + \jmath \frac{\partial e(n)}{\partial \mathbf{b}}\right) e^*(n)\right]. \tag{3.32}$$

Now, by substituting (3.28) into (3.25), $e(n)$ and $e^*(n)$ become

$$e(n) = d(n) - y(n) \tag{3.33}$$
$$= d(n) - \mathbf{w}^H \mathbf{x}_n \tag{3.34}$$
$$= d(n) - (\mathbf{a} + \jmath\mathbf{b})^H \mathbf{x}_n \tag{3.35}$$
$$e^*(n) = d^*(n) - y^*(n) \tag{3.36}$$
$$= d^*(n) - \mathbf{x}_n^H \mathbf{w} \tag{3.37}$$
$$= d^*(n) - \mathbf{x}_n^H (\mathbf{a} + \jmath\mathbf{b}). \tag{3.38}$$

Applying the partial derivative to (3.35) results in

$$\frac{de(n)}{d\mathbf{w}} = \frac{\partial e(n)}{\partial \mathbf{a}} + \jmath\frac{\partial e(n)}{\partial \mathbf{b}} \tag{3.39}$$
$$= \frac{\partial}{\partial \mathbf{a}}\left(d(n) - (\mathbf{a} - \jmath\mathbf{b})^H \mathbf{x}_n\right) + \jmath\frac{\partial}{\partial \mathbf{b}}\left(d(n) - (\mathbf{a} - \jmath\mathbf{b})^H \mathbf{x}_n\right) \tag{3.40}$$
$$= -\mathbf{x}_n - \mathbf{x}_n \tag{3.41}$$
$$= -2\mathbf{x}_n \tag{3.42}$$

where

$$\frac{\partial e(n)}{\partial \mathbf{a}} = \frac{\partial}{\partial \mathbf{a}}\left(d(n) - (\mathbf{a} - \jmath\mathbf{b})^H \mathbf{x}_n\right) \tag{3.43}$$
$$= -\mathbf{x}_n, \tag{3.44}$$
$$\frac{\partial e(n)}{\partial \mathbf{b}} = \frac{\partial}{\partial \mathbf{b}}\left(d(n) - (\mathbf{a} - \jmath\mathbf{b})^H \mathbf{x}_n\right) \tag{3.45}$$
$$= \jmath\mathbf{x}_n, \tag{3.46}$$

while applying the partial derivative to (3.38) yields

$$\frac{de^*(n)}{d\mathbf{w}} = \frac{\partial e^*(n)}{\partial \mathbf{a}} + \jmath \frac{\partial e^*(n)}{\partial \mathbf{b}} \tag{3.47}$$

$$= \frac{\partial}{\partial \mathbf{a}} \left(d^*(n) - \mathbf{x}_n^H (\mathbf{a} + \jmath \mathbf{b}) \right) + \jmath \frac{\partial}{\partial \mathbf{b}} \left(d^*(n) - \mathbf{x}_n^H (\mathbf{a} + \jmath \mathbf{b}) \right) \tag{3.48}$$

$$= -\mathbf{x}_n^H + \mathbf{x}_n^H \tag{3.49}$$

$$= 0 \tag{3.50}$$

where

$$\frac{\partial e^*(n)}{\partial \mathbf{a}} = \frac{\partial}{\partial \mathbf{a}} \left(d^*(n) - \mathbf{x}_n^H (\mathbf{a} + \jmath \mathbf{b}) \right) \tag{3.51}$$

$$= -\mathbf{x}_n^H, \tag{3.52}$$

$$\frac{\partial e^*(n)}{\partial \mathbf{b}} = \frac{\partial}{\partial \mathbf{b}} \left(d^*(n) - \mathbf{x}_n^H (\mathbf{a} + \jmath \mathbf{b}) \right) \tag{3.53}$$

$$= -\jmath \mathbf{x}_n^H. \tag{3.54}$$

Substituting (3.44), (3.46), (3.52) and (3.54) into (3.32) and substituting $\mathbf{w}$ back in for $\mathbf{a} + \jmath \mathbf{b}$ yields

$$\nabla_{\mathbf{w}} J_n(\mathbf{w}) = E\left[\; -\mathbf{x}_n^* e(n) - \mathbf{x}_n e^*(n) - \jmath^2 \mathbf{x}_n^* e(n) + \jmath^2 \mathbf{x}_n e^*(n) \; \right] \tag{3.55}$$

$$= E\left[\; -\mathbf{x}_n^* e(n) - \mathbf{x}_n e^*(n) + \mathbf{x}_n^* e(n) - \mathbf{x}_n e^*(n) \; \right] \tag{3.56}$$

$$= E\left[\; -\mathbf{x}_n e^*(n) - \mathbf{x}_n e^*(n) \; \right] \tag{3.57}$$

$$= -2E\left[\mathbf{x}_n e^*(n)\right]. \tag{3.58}$$

$\nabla_{\mathbf{w}} J_n(\mathbf{w})$ is minimized when

$$-2E\begin{bmatrix} x\left(n\right)e_o^*(n) \\ x\left(n-1\right)e_o^*(n) \\ x\left(n-2\right)e_o^*(n) \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} = \mathbf{0} \quad \forall \quad n = 1, 2, \ldots, \tag{3.59}$$

where e_o denotes the optimized error. This shows that $\mathbf{x}_n$ must be orthogonal to $e_o \ \forall \ n = 0, 1, \ldots$.

3.2.3 The Unconstrained Optimal Wiener Solution

The optimal Wiener solution is derived from (3.26). First, we redefine the weight vector of (3.28) to be of filter order N, given by

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix}. \tag{3.60}$$

Then, (3.34) and (3.37) are substituted into (3.26) given by

$$J_n\left(\mathbf{w}\right) = E\left[\left(d\left(n\right) - \mathbf{w}^H\mathbf{x}_n\right)\left(d\left(n\right) - \mathbf{w}^H\mathbf{x}_n\right)^*\right] \ \forall \ n = 0, 1, 2, \ldots, N. \tag{3.61}$$

The optimal filter weights are found by minimizing (3.61). By calculating the gradient of $J\left(\mathbf{w}\right)$ with respect to $\mathbf{w}^*$, (3.61) effectively becomes

$$\nabla_{\mathbf{w}^*} J_n\left(\mathbf{w}\right) = E\left[\frac{\partial}{\partial \mathbf{w}^*}\left(d\left(n\right) - \mathbf{w}^H\mathbf{x}_n\right)\left(d\left(n\right) - \mathbf{w}^H\mathbf{x}_n\right)^*\right] \tag{3.62}$$

$$\nabla_{\mathbf{w}^*} J_n\left(\mathbf{w}\right) = E\left[\frac{\partial}{\partial \mathbf{w}^*}\left(d\left(n\right)d^*\left(n\right) - d\left(n\right)\mathbf{x}_n^H\mathbf{w} - \mathbf{w}^H\mathbf{x}_n d^*\left(n\right) + \mathbf{w}^H\mathbf{x}_n\mathbf{x}_n^H\mathbf{w}\right)\right]. \tag{3.63}$$

By following the rules of complex variable derivatives,

$$\frac{\partial \mathbf{w}}{\partial \mathbf{w}^*} = 0 \text{ and } \frac{\partial \mathbf{w}^*}{\partial \mathbf{w}^*} = 1,$$

(3.63) reduces to

$$\nabla_{\mathbf{w}^*} J_n(\mathbf{w}) = E\left[-\mathbf{x}_n d^*(n) + \mathbf{x}_n \mathbf{x}_n^H \mathbf{w}\right] \quad \forall \quad n = 0, 1, 2, \ldots, N. \tag{3.64}$$

By setting (3.64) equal to 0 and solving for $\mathbf{w}$, we get the optimal Wiener solution, given by

$$E\left[-\mathbf{x}_n d^*(n) + \mathbf{x}_n \mathbf{x}_n^H \mathbf{w}\right] = 0 \tag{3.65}$$

$$E\left[\mathbf{x}_n \mathbf{x}_n^H\right] \mathbf{w} = E\left[\mathbf{x}_n d^*(n)\right] \tag{3.66}$$

$$\mathbf{w} = E\left[\left(\mathbf{x}_n \mathbf{x}_n^H\right)\right]^{-1} E\left[\mathbf{x}_n d^*(n)\right]. \tag{3.67}$$

Optimizing $\mathbf{w}$ is a regression problem that applies $\forall \; n = 0, 1, ..., N$. (3.64) can be condensed into matrix form, given by

$$\nabla_{\mathbf{w}^*} J(\mathbf{w}) = -\mathbf{X}\mathbf{d}^* + \mathbf{X}\mathbf{X}^H \mathbf{w} \tag{3.68}$$

where

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \text{ and } \mathbf{X}^H = \begin{bmatrix} \mathbf{x}_1^H \\ \mathbf{x}_2^H \\ \vdots \\ \mathbf{x}_N^H \end{bmatrix}. \tag{3.69}$$

(3.67) becomes

$$\mathbf{w} = \left(\mathbf{X}\mathbf{X}^H\right)^{-1}\mathbf{X}\mathbf{d}^* \tag{3.70}$$

$$= \mathbf{\Sigma}_{\mathbf{X}}^{-1}\mathbf{\Gamma}_{\mathbf{X}\mathbf{d}^*}. \tag{3.71}$$

where $\mathbf{\Sigma}_{\mathbf{X}}$ is the autocorrelation of random process $\boldsymbol{X}$ and $\mathbf{\Gamma}_{\mathbf{X}\mathbf{d}^*}$ is the cross-correlation between $\mathbf{X}$ and $\mathbf{d}^*$. (3.71) can be expressed in expanded matrix form given by

$$\mathbf{w} = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,N} \\ \sigma_{2,1} & \sigma_{2,2} & \cdots & \sigma_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N,1} & \sigma_{N,2} & \cdots & \sigma_{N,N} \end{bmatrix}^{-1} \begin{bmatrix} \gamma(0) \\ \gamma(-1) \\ \vdots \\ \gamma(-N+1) \end{bmatrix} \tag{3.72}$$

where $\sigma_{i,j}$ is calculated by (3.7) and $\gamma(-j+1)$ is calculated by

$$\gamma(-k) = \sum_{i=N}^{2N-1} x_{i-k}\, d_i^* \quad \forall \quad k = 0, 1, \ldots, N-1. \tag{3.73}$$

By setting the elementwise weight vector of (3.60) equal to (3.72), the elementwise complex filter weights are given by

$$w_1 = \sum_{j=1}^{N} \mathbf{\Sigma}_{1,j}^{-1}\, \gamma(-j+1) \tag{3.74}$$

$$w_2 = \sum_{j=1}^{N} \mathbf{\Sigma}_{2,j}^{-1}\, \gamma(-j+1) \tag{3.75}$$

$$\vdots \tag{3.76}$$

$$w_N = \sum_{j=1}^{N} \mathbf{\Sigma}_{N,j}^{-1}\, \gamma(-j+1). \tag{3.77}$$

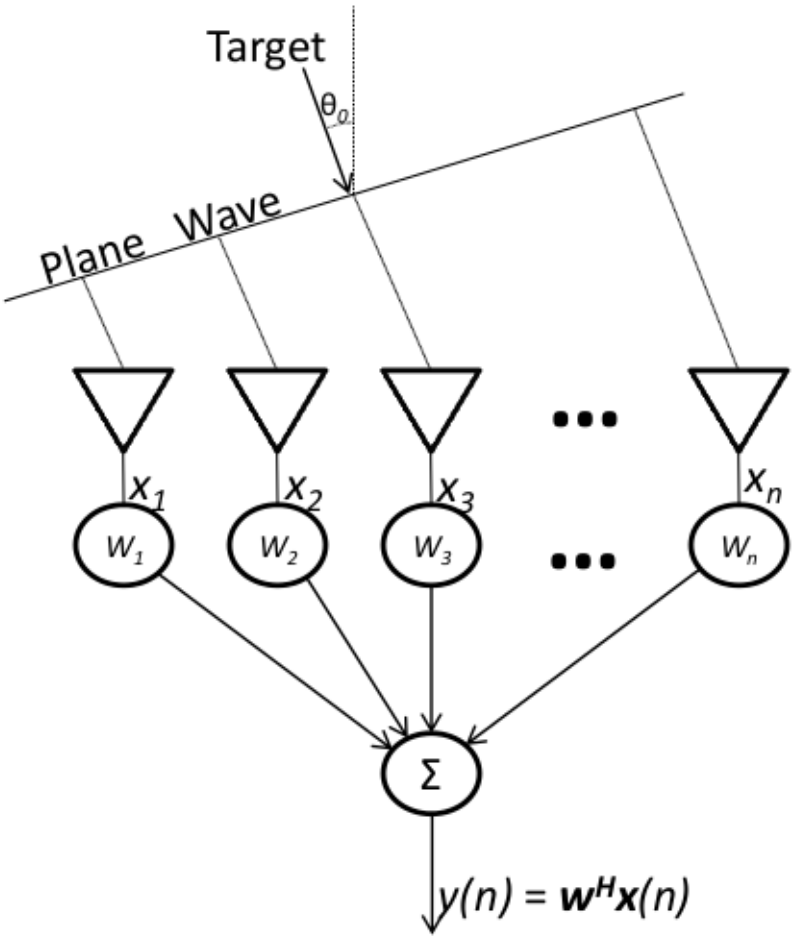

Figure 3.3: Phased Array Antenna, similar to the phased array antenna given in [16].

3.2.4 The Constrained Optimal Wiener Solution

As previously discussed, the desired response may not be available *a priori*. When this is the case, the constrained Wiener filter is used.

Assume the phased array antenna in Fig. 3.3 is the receiving antenna for a STAP radar system. A plane wave impinges upon the antenna array originating from an angle θ_0. The constrained Wiener filter is *constrained* to maximize the amplitude of the signal at θ_0 while cancelling the signals from all other directions. We define a constraint matrix, given by

$$\mathbf{C} = \mathbf{s}\left(\theta_0\right) = \left[1, e^{-\jmath\theta_0}, \ldots, e^{-\jmath(N-1)\theta_0}\right]^T, \tag{3.78}$$

where $\mathbf{s}(\theta_0)$ is known as the steering vector. The latter is an angle dependent vector that accounts for the phase changes in the plane wave observed by subsequent antenna taps. This information is used to generate a *synthetic* desired signal. Furthermore, there may be more than one constraint. For example, it may be necessary to maximize signals from θ_0, while simultaneously minimizing a known interference source at θ_1. In this case, the constraint matrix becomes

$$\mathbf{C} = \begin{bmatrix} \mathbf{s}(\theta_0) \\ \mathbf{s}(\theta_1) \end{bmatrix}^T = \begin{bmatrix} 1, e^{-\jmath\theta_0}, \ldots, e^{-\jmath(N-1)\theta_0} \\ 1, e^{-\jmath\theta_1}, \ldots, e^{-\jmath(N-1)\theta_1} \end{bmatrix}^T. \tag{3.79}$$

The optimal Wiener solution is one that satisfies

$$\mathbf{C}^H \mathbf{w}_0 = \mathbf{g} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.80}$$

where $\mathbf{g}$ is the gain matrix associated with each of the constraints. Now, the constraint matrix is used to generate a set of constrained filter weights, $\mathbf{w}_c$, given by

$$\mathbf{w}_c = \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{g}, \tag{3.81}$$

that are used to generate the synthetic desired signal. From here, $\mathbf{w}_c$ is used to calculate the synthetic desired signal by

$$d(n) = \mathbf{w}_c^H \mathbf{u}(n) \quad \forall \quad n = 0, 1, \ldots, \tag{3.82}$$

where $\mathbf{u}(n)$ is known in STAP radar as the Cell Under Test (CUT). Now, a blocking matrix, $\mathbf{B}$, must be generated such that

$$\mathbf{C}^H\mathbf{B} = \mathbf{0}. \tag{3.83}$$

$\mathbf{B}$ is generated using Gram-Schmidt orthonormalization and is then used to generate a

new input signal, given by

$$\mathbf{x}(n) = \mathbf{B}^H \mathbf{u}(n) \quad \forall \quad n = 0, 1, \ldots. \tag{3.84}$$

At this point, the original input signal was used to generate a synthetic desired signal, $\mathbf{d}$ and a new input signal, $\mathbf{x}$. From here, the process for calculating the optimal Wiener solution for the constrained Wiener filter is identical to that for the unconstrained case. The process initiated in (3.61) can be followed to find the optimal Wiener solution.

3.2.5 Linearly Constrained Minimum Variance Filter

As previously stated, for STAP radar systems, a desired signal is not present. The linearly constrained minimum variance (LCMV) filter is an optimization problem that seeks to maximize the synthetic desired signal while minimizing interference signals. It is solved using the method of Lagrange multipliers. The new cost function becomes

$$J(\mathbf{w}) = \sum_{k=0}^{N-1} \sum_{i=0}^{N-1} w_k^* w_i \gamma(i-k) + Re\left[\lambda^* \sum_{k=0}^{N-1} w_k^* e^{-\jmath\theta_0 k} - g\right]. \tag{3.85}$$

Here, the first term of the left-hand-side of the equality represents the output power of the filter and the second term represents the linear constraint. By minimizing (3.85), the signal is preserved while the interference or noise is minimized. Taking the gradient of (3.85) with respect to w and setting equal to zero yields

$$\nabla_{\mathbf{w}_k} J = 2 \sum_{i=0}^{N-1} w_i \gamma(i-k) + \lambda^* e^{-\jmath\theta_0 k} = 0, \qquad k = 0, 1, 2, \ldots, N-1, \tag{3.86}$$

which represents a system of N simultaneous equations. Notice that $e^{-\jmath\theta_0 k}$ over N simultaneous equations is the steering vector from (3.78). Rewriting (3.86) in matrix form yields

$$2\boldsymbol{\Sigma}\mathbf{w}_0 + \lambda^* \mathbf{s}(\theta_0) = \mathbf{0} \tag{3.87}$$

$$\boldsymbol{\Sigma}\mathbf{w}_0 = -\frac{\lambda^*}{2}\mathbf{s}(\theta_0). \tag{3.88}$$

Solving for $\mathbf{w}_0$ yields

$$\mathbf{w}_0 = -\frac{\lambda^*}{2}\boldsymbol{\Sigma}^{-1}\mathbf{s}(\theta_0). \tag{3.89}$$

Recall from (3.80) that $\mathbf{w}^H\mathbf{s}(\theta_0) = g$. By taking the Hermitian of (3.89) and post-multiplying by $\mathbf{s}(\theta_0)$, λ is found by

$$\mathbf{w}_0^H = -\left(\frac{\lambda^*}{2}\boldsymbol{\Sigma}^{-1}\mathbf{s}(\theta_0)\right)^H, \tag{3.90}$$

$$\mathbf{w}_0^H = -\frac{\lambda}{2}\mathbf{s}^H(\theta_0)\boldsymbol{\Sigma}^{-1}, \tag{3.91}$$

$$\mathbf{w}_0^H\mathbf{s}(\theta_0) = -\frac{\lambda}{2}\mathbf{s}^H(\theta_0)\boldsymbol{\Sigma}^{-1}\mathbf{s}(\theta_0), \tag{3.92}$$

$$g = -\frac{\lambda}{2}\mathbf{s}^H(\theta_0)\boldsymbol{\Sigma}^{-1}\mathbf{s}(\theta_0), \tag{3.93}$$

yielding

$$\lambda = \frac{-2g}{\mathbf{s}^H(\theta_0)\boldsymbol{\Sigma}^{-1}\mathbf{s}(\theta_0)}. \tag{3.94}$$

Now, (3.94) can be substituted into (3.89), yielding

$$\mathbf{w}_0 = \frac{g\,\boldsymbol{\Sigma}^{-1}\mathbf{s}(\theta_0)}{\mathbf{s}^H(\theta_0)\boldsymbol{\Sigma}^{-1}\mathbf{s}(\theta_0)}. \tag{3.95}$$

From (3.80), assume that $g = 1$. Then, the optimal LCMV filter solution simplifies to

$$\mathbf{w}_0 = \frac{\boldsymbol{\Sigma}^{-1}\mathbf{s}(\theta_0)}{\mathbf{s}^H(\theta_0)\boldsymbol{\Sigma}^{-1}\mathbf{s}(\theta_0)}. \tag{3.96}$$

This is the optimal Wiener solution for the LCMV filter used in STAP radar. This is used later in Chapter 5 for the STAP application of this thesis.

3.2.6 Current Shortcomings of the Wiener Filter

Assume a dataset, $\mathbf{X}$, of size $M = 100$ is drawn from a sine wave of unit amplitude with additive Gaussian white noise, given by

$$\mathbf{X} = [x_1, x_2, \ldots, x_M] = \left\{ sin\left(\frac{2\pi i}{M}\right) + 0.1\, N(0,1)\,,\ i = 1, \ldots, M \right\}, \tag{3.97}$$

where $x_i = sin\left(\frac{2\pi i}{M}\right) + 0.1\, N(0,1)\,,\ i = 1, \ldots, M$.

The desired signal, $\mathbf{d}$, is shown in Fig. 3.4. The plot of the time sequence of samples of $\mathbf{X}$ appears in Fig. 3.5. By applying (3.70) to $\mathbf{X}$ and $\mathbf{d}$, the $10th$ order optimal filter weights are calcuated and the estimate of the desired signal in Fig. 3.6 is generated. In this case, the dataset consists entirely of good data points, and the filter can effectively filter the Gaussian white noise from the input signal. However, since the Wiener filter is based on the L_2-norm estimator, even a single outlier is detrimental to its performance.

Referring back to the signal depicted Fig. 3.5, a new input signal is now generated containing an outlier of magnitude 10. The resulting new input signal is displayed in Fig. 3.7. Here, the outlier is implanted at position 49 out of 100. The input signal is processed by the Wiener filter and the resulting output signal is displayed in Fig. 3.8. As can be seen, the L_2-norm cost function broke down. A more robust cost function is needed. Several attempts at robustifying the Wiener filter have been documented and some have succeeded [44], [59], [60], [61]. However, none of the previous attempts have developed a robust Wiener filter entirely in the complex domain. This is the topic of the following chapter.

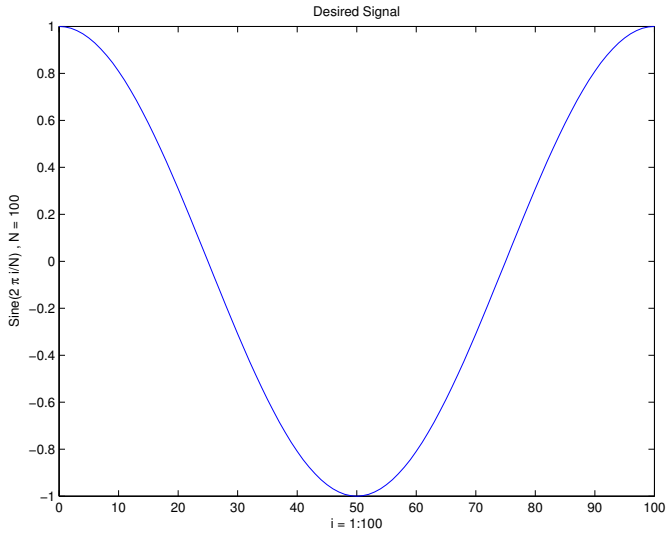

Figure 3.4: The desired signal of the Wiener filter given by $sin\left(\frac{2\pi i}{M}\right),\ i = 1, \ldots, M.$

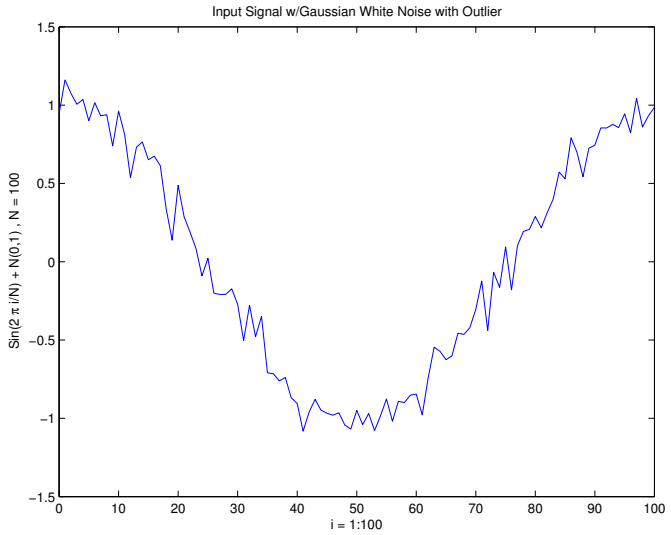

Figure 3.5: The input signal of the Wiener filter given by $sin\left(\frac{2\pi i}{M}\right) + 0.1\, N\left(0, 1\right),\ i = 1, \ldots, M.$

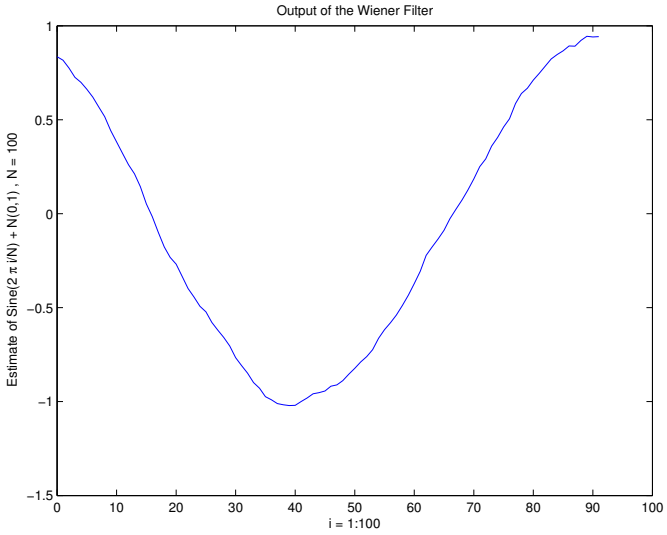

Figure 3.6: The output signal of the Wiener filter, which is an estimate of the input signal $sin\left(\frac{2\pi i}{M}\right),\ i = 1, \ldots, M$.

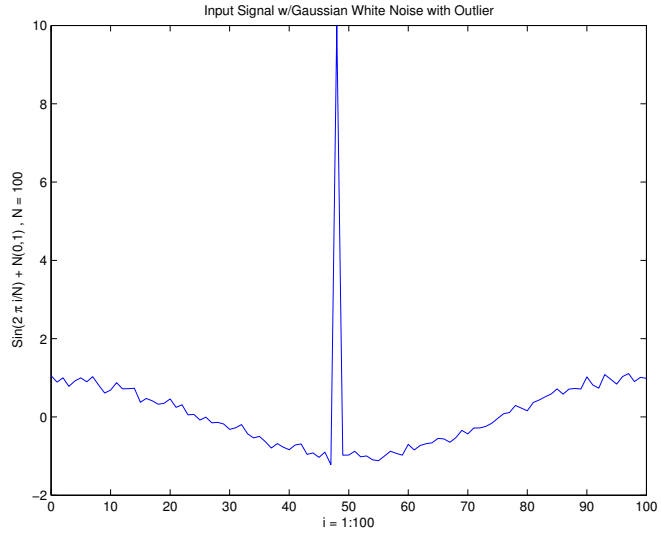

Figure 3.7: The input signal of the Wiener filter given by $sin\left(\frac{2\pi i}{M}\right) + 0.1\,N\left(0,1\right),\ i = 1, \ldots, M$ with an outlier of magnitude 10 embedded at $i = 10$.

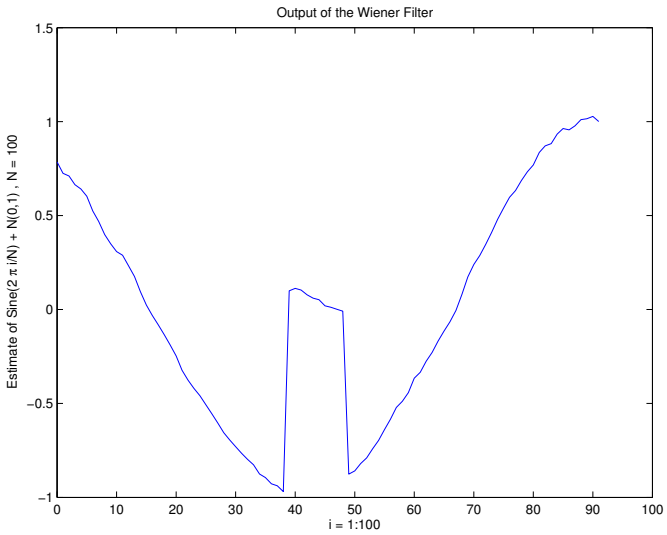

Figure 3.8: The output signal of the Wiener filter, which is an estimate of the input signal $sin\left(\frac{2\pi i}{M}\right),\ i = 1, \ldots, M$ with an outlier of magnitude 10 embedded at $i = 10$.

Chapter 4

Robust Complex Wiener Filter

4.1 The Complex GM Wiener Filter

In chapter 3, the L_2-norm Wiener filter was shown to be non-robust to outliers. In this section, we robustify the Wiener filter by replacing the L_2-norm estimator with the complex GM-Huber estimator, thereby creating the Complex GM-Wiener Filter (CGMWF).

Referring to (3.26), where the errors, $e\,(i)$, are the residuals of the dataset, $r\,(i)$, we have $r(i)r^*(i) = |r^2(i)|$ is the *kernel function*, $\rho\,(r_i)$. That is

$$r(i)r^*(i) = |r^2(i)| = \rho\,(r_i)\,, \tag{4.1}$$

and (3.26) therefore simplifies to its general form

$$J_i\,(\mathbf{w}) = E[\rho\,(r_i)] \;\; \forall \;\; i = 1, 2, \ldots, N, \tag{4.2}$$

where J is the cost function, $\mathbf{w}$ is the weight vector, and r_i is the i-th residual. Here, the Wiener filter is robustified by replacing the L_2-norm cost function with a more robust estimator, i.e. a new $\rho\,(r_i)$ function is used. At this point, we take the derivative of (4.2) with respect to $\mathbf{w}^*$. Using the chain rule for differentiation and setting equal to zero yields

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^*} = E\left[\frac{\partial \rho(r)}{\partial \mathbf{w}^*}\right] \tag{4.3}$$

$$= E\left[\frac{\partial \rho(r)}{\partial r}\frac{\partial r}{\partial \mathbf{w}^*}\right] \tag{4.4}$$

$$= E\left[\frac{\partial \rho(r)}{\partial r}\frac{\partial}{\partial \mathbf{w}^*}\left(d - \mathbf{w}^H\mathbf{x}\right)\right] \tag{4.5}$$

$$= E\left[\frac{\partial \rho(r)}{\partial r} \; -\mathbf{x}\right] \tag{4.6}$$

$$= -E\left[\frac{\partial \rho(r)}{\partial r}\,\mathbf{x}\right] = 0 \tag{4.7}$$

Substituting (2.16) into (4.7) and multiplying by r^*/r^* yields

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^*} = E\left[\frac{\psi(r)}{r^*}\; r^*\,\mathbf{x}\right] = 0. \tag{4.8}$$

At this point, the *weight function*, $q(r)$ is defined as

$$q(r) = \frac{\psi(r)}{r^*} \tag{4.9}$$

and is substituted into (4.8), giving

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^*} = E\left[q(r)\;\; r^*\,\mathbf{x}\right] = 0. \tag{4.10}$$

Rearranging the scalar quantities within (4.10) and substituting (3.38) for r^* in (4.10) yields

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^*} = E\left[q(r)\ \mathbf{x}\ r^*\right] \tag{4.11}$$

$$= E\left[q(r)\ \mathbf{x}\ \left(d - \mathbf{w}^H\mathbf{x}\right)^*\right] \tag{4.12}$$

$$= E\left[q(r)\ \mathbf{x}\ \left(d^* - \mathbf{x}^H\mathbf{w}\right)\right] \tag{4.13}$$

$$= E\left[q(r)\,\mathbf{x}d^* - q(r)\,\mathbf{x}\mathbf{x}^H\mathbf{w}\right] \tag{4.14}$$

$$= E\left[\mathbf{x}q(r)\,d^* - \mathbf{x}q(r)\,\mathbf{x}^H\mathbf{w}\right] \tag{4.15}$$

$$= E\left[\mathbf{x}q(r)\,d^*\right] - E\left[\mathbf{x}q(r)\,\mathbf{x}^H\mathbf{w}\right] = 0, \tag{4.16}$$

which yields

$$E\left[\mathbf{x}q(r)\,\mathbf{x}^H\mathbf{w}\right] = E\left[\mathbf{x}q(r)\,d^*\right]. \tag{4.17}$$

At this point, the weight vector, $\mathbf{w}$, is removed from the expectation operator because it is an unknown vector of constants, i.e. it is not a random variable and does not vary with time. (4.17) becomes

$$E\left[\mathbf{x}q(r)\,\mathbf{x}^H\right]\mathbf{w} = E\left[\mathbf{x}q(r)\,d^*\right], \tag{4.18}$$

and solving for the optimal filter weights yields

$$\mathbf{w} = E\left[\mathbf{x}q(r)\,\mathbf{x}^H\right]^{-1} E\left[\mathbf{x}q(r)\,d^*\right]. \tag{4.19}$$

The expectation operators in (4.19) must be replaced with the corresponding matrix form. This is done by summing over the elements of x_i, $q(r_i)$ and d_i^*, as in

$$\mathbf{w} = \left(\frac{1}{N}\sum_{i=1}^{N} x_i q(r_i)\, x_i^*\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} x_i q(r_i)\, d_i^*\right). \tag{4.20}$$

This reduces to

$$\mathbf{w}^{k+1} = \left(\mathbf{X}\mathbf{Q}^k\mathbf{X}^H\right)^{-1}\mathbf{X}\mathbf{Q}^k\mathbf{d}^*, \tag{4.21}$$

where $\mathbf{w}$ is the weight vector, $\mathbf{X}$ represents the data matrix as presented in the (3.69), and $\mathbf{Q}$ is the diagonal matrix of the residual weights calculated by (4.9). $\mathbf{Q}$ is given by

$$\mathbf{Q} = \begin{bmatrix} q(r_1) & 0 & 0 & \cdots & 0 \\ 0 & q(r_2) & 0 & \cdots & 0 \\ 0 & 0 & q(r_3) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & q(r_N) \end{bmatrix} \tag{4.22}$$

Note the superscripts on Q^k and w^{k+1} in (4.21). The superscripts denote that iteration is required for convergence to the optimal filter weights. The iterative approach will be discussed in further detail in the following section.

Recall from section 2.2.2 that the *weight function* for the L_2-norm estimator is given by (2.31). By assuming a L_2-norm estimator is used, (4.22) reduces to

$$\mathbf{Q} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}. \tag{4.23}$$

Consequently, (4.21) reduces to

$$\mathbf{w} = \left(\mathbf{X}\mathbf{X}^H\right)^{-1}\mathbf{X}\mathbf{d}^*, \tag{4.24}$$

which is identical to the optimal Wiener solution given in (3.70).

4.2 Statistical Test for Outlier Identification Based on the MCD

This section describes the statistical test that is used to flag a data point as an outlier.

Referring back to section 2.4, the MCD is used to determine if a data point is an outlier. The MD is calculated by (2.57), shown here for reference

$$MD_1(i) = \sqrt{(\mathbf{x}_i - \mathbf{T}_1)^T \boldsymbol{\Sigma}_1^{-1} (\mathbf{x}_i - \mathbf{T}_1)} \;\; \forall \;\; i = 1, 2, \ldots, M. \tag{4.25}$$

A weight factor is then calculated for each data point based on the MDs, given by (2.54), shown here for reference

$$w(\mathbf{x}_i) = \min\left(1, \frac{b^2}{MCD_i}\right). \tag{4.26}$$

For N-dimensional data, the MDs follow a χ^2_N distribution with N degrees of freedom (DOF) when the data points are drawn from a Gaussian distribution. Those data points, whose MDs exceed the 97.5-*th* percentile are flagged as outliers. This is displayed in Fig. 4.1. The data points in this case are 2-dimensional following a $\chi^2_{N=2}$. The data points circled in blue are the outliers. The weight factors are calculated by (2.54) and the data points are reweighted according to (2.52), shown here for reference

$$\hat{\boldsymbol{\alpha}}_{GM} = \underset{\boldsymbol{\alpha}}{argmin}\; J(\boldsymbol{\alpha}) = \underset{\boldsymbol{\alpha}}{argmin} \sum_{i=1}^{m} w^2(\mathbf{x}_i)\, \rho\left(\frac{r_i}{s\, w(\mathbf{x}_i)}\right). \tag{4.27}$$

This statistical test is captured in step 2 through step 5 in the algorithm outlined in the following section. The algorithm is applied to calculate an estimate of the optimal weight vector for the Wiener Filter.

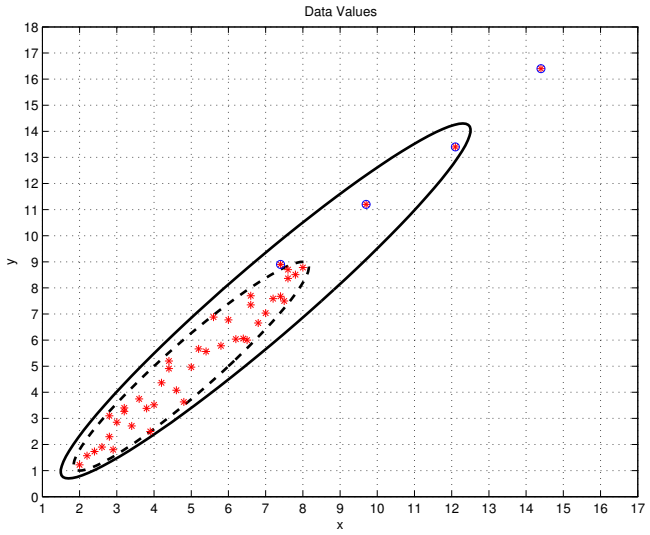

Figure 4.1: In this figure, the red stars are the data points, the solid lined ellipse represents 97.5-*th* percentile of the standard dataset, the dash lined ellipse represents the robust 97.5-*th* percentile of the robust data set based on the MCD, and the red stars that are circled in blue represent the data points that are outliers.

4.3 Robust Complex GM Wiener Filter Algorithm

This section discusses complex GM Wiener filter algorithm and its contributions to adaptive signal processing. The shortcomings discussed in section 3.3.6 are overcome. The modified Wiener filter incorporates the Huber estimator of section 2.4.1, and the MCD algorithm of section 2.6.

Assume a random process, $\mathbf{X}$, with additive Gaussian white noise and random spiky outliers. The steps for filtering the noise are listed below.

1. A random dataset, $\mathbf{X}$ of $M = 100$ samples and N dimensions, is drawn from random process, $\boldsymbol{X}$.

2. The dataset is processed by the MCD algorithm.

3. The robust MCD-based distances, denoted by RD are calculated for each sample in the data set.

4. The robust distances, RD, are compared to the 97.5-*th* percentile of the χ^2_N distribution with N DOF to determine which data points are outliers.

5. The residuals are weighted according to equation (2.54).

6. A best guess weight vector is generated. Usually a unit vector of N dimensions is sufficient.

7. The estimate of the desired signal is calculated using the dataset and the weight vector.

8. The residual vector is calculated.

9. The new residuals are weighted by the Huber estimator to downweight any outliers.

10. The resulting weights are used to calculate $\mathbf{Q}$ given by (4.22).

11. A new weight vector, $\mathbf{w}_{new}$, is calculated according to (4.24).

12. A difference vector, $\mathbf{w}_{diff}$, between the new weight vector and the starting weight vector is calculated as $\mathbf{w}_{diff} = \mathbf{w} - \mathbf{w}_{new}$.

13. The starting weight vector is adjusted by the difference vector, $\mathbf{w} = \mathbf{w} + \mathbf{w}_{diff}$.

14. Repeat steps 8 thru 14 until the weight vector converges.

15. Draw a new dataset, $\mathbf{X}$.

16. Use the RD to detect any outliers in the new dataset.

17. Predict new *good* data points in the dataset.

18. Calculate a new estimate of the desired signal using the new optimized weight vector.

The algorithm outlined above has been applied to the sine wave in section 3.2.6. Several test cases ranging from zero to 50% contamination were generated from the sine wave. The test cases are then processed by the robust CGMWF, by filter orders ranging from 1 to 50.

Table 4.1 displays the performance of the MCD and the CGMWF for outlier detection and sine wave estimation. The parameters include the dimension of the data, the fraction of contamination, number of outliers, the number of outliers detected, number of subsets

Table 4.1: Performance of the Minimum Covariance Determinant for Outlier Detection

Dimension	ε	Outliers	Detected	$m \vert P_{Contaminated} = 1\%$	BD?
1	0.0040	2	2	1	No
1	0.0200	10	10	2	No
1	0.0500	25	25	2	No
1	0.1000	50	50	3	No
1	0.2500	125	125	6	No
1	0.5000	251	250	17	Yes
5	0.0040	2	2	2	No
5	0.0200	10	10	3	No
5	0.0500	25	25	4	No
5	0.1000	50	50	7	No
5	0.2500	125	125	24	No
5	0.5000	251	250	293	Yes
10	0.0040	2	2	2	No
10	0.0200	10	10	3	No
10	0.0500	25	25	6	No
10	0.1000	50	50	13	No
10	0.2500	125	125	107	No
10	0.5000	251	250	9,430	Yes
25	0.0040	2	2	2	No
25	0.0200	10	10	6	No
25	0.0500	25	25	16	No
25	0.1000	50	50	69	No
25	0.2500	125	125	8,157	No
25	0.5000	251	250	309,047,738	Yes
50	0.0040	2	2	3	No
50	0.0200	10	10	11	No
50	0.0500	25	25	61	No
50	0.1000	50	50	991	No
50	0.2500	125	125	10,842,294	No
50	0.5000	251	250	10,369,921,366,795,726	Yes

needed so that the probability of selecting a contaminated subset is at most 1% and if the MCD broke down.

By modifying the L_2-norm Wiener filter with the algorithm above, it is more robust. The signal displayed in Fig. 3.7 is used as a test input. Fig. 4.2 shows the output when the MCD is used for outlier detection and the optimal filter weights are calculated based on a GM Huber estimator. Fig. 4.3 shows the output when the MCD is used for outlier detection, but the optimal filter weights are calculated based on the L_2-norm estimator.

Fig. 4.2 shows a significant improvement over Fig. 3.7. The underlying deterministic sine wave is correctly estimated despite the presence of the outlier. Fig. 4.3 appears

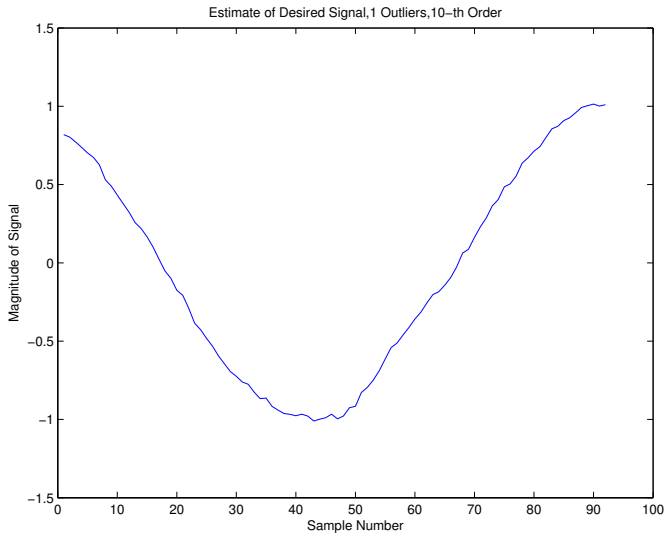

Figure 4.2: Estimate of $sin\left(\frac{2\pi i}{M}\right)+0.1\,N\left(0,1\right)$ with 1 Outlier using CGMWF with MCD.

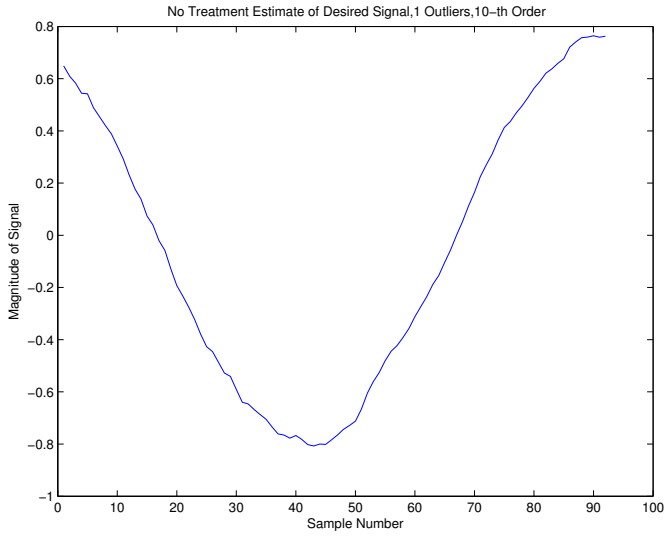

Figure 4.3: Estimate of Sin $\left(\frac{2\pi i}{M}\right)+0.1\,N\left(0,1\right)$ with 1 Outlier using L_2-norm Wiener filter with MCD.

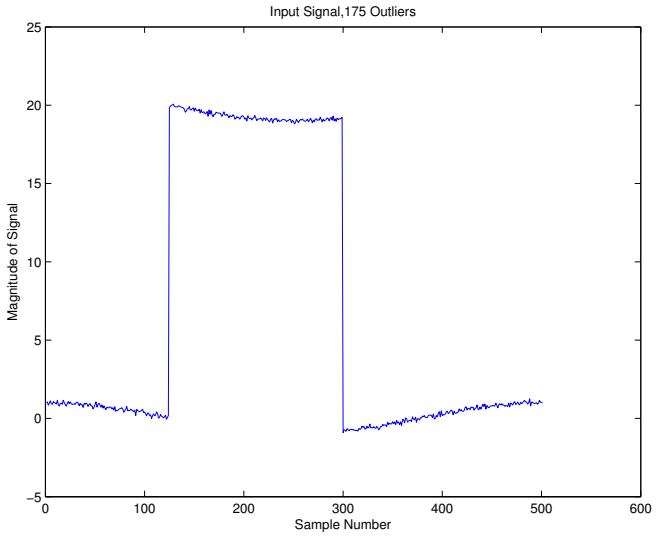

Figure 4.4: Input $sin\left(\frac{2\pi i}{M}\right) + 0.1\,N\left(0,1\right)$ with 35% Outlier Contamination.

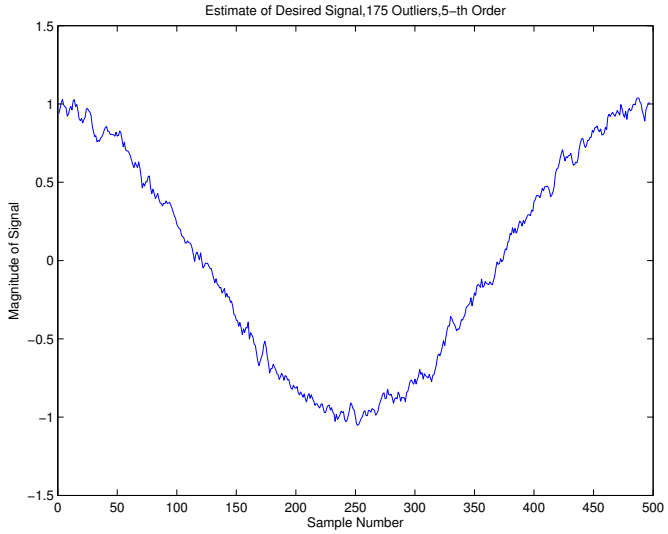

Figure 4.5: Signal Estimate of Fig. 4.4 using a 5-th Order Filter.

to show a significant improvement as well. However, despite the smooth appearance of sine wave estimate, this estimator broke down. The amplitude of the sine wave has been compromised by the outlier. The breakdown is slight because of the modest amplitude of the outlier (magnitude of 10).

Now, the same signal in Fig. 3.7 is contaminated with 35% outliers with a magnitude of 20. In this case, $M = 500$ samples were generated. The new input signal is displayed in Fig. 4.4. The CGMWF was tested with the algorithm above with filters of orders 5 and 10. The resulting signal estimates are displayed in Figs. 4.5 and 4.6.

The 5-th order filter successfully estimated the signal despite a 35% outlier contaimi-

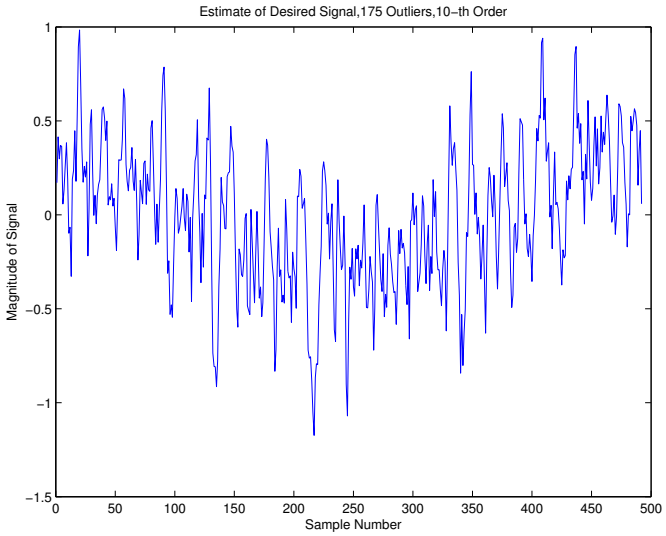

Figure 4.6: Signal Estimate of Fig. 4.4 using a 10-th Order Filter.

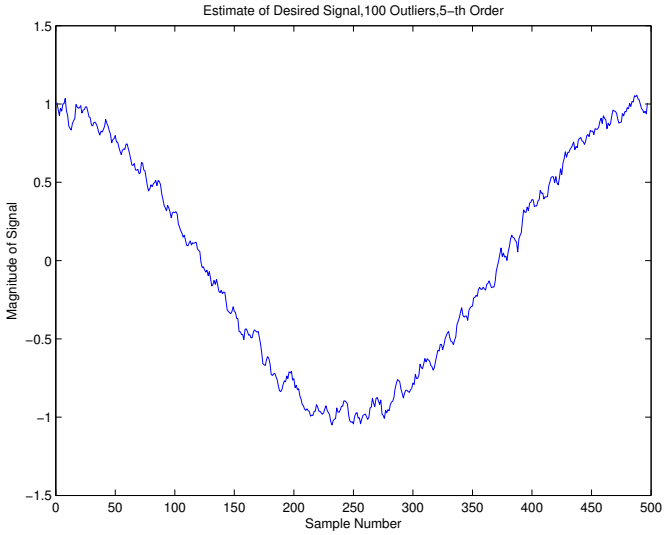

Figure 4.7: Signal Estimate of Fig. 4.4 using a 5-th Order Filter.

nation. The 10-th order filter, however, broke down. This is because the sampling of the data for lower order filters inherently covers smaller ranges of data. The outliers are less likely to contaminate the covariance matrix.

Now, the input signal is contaminated with 20% outliers, and the orders under consideration are the 5-th order filter and the 50-th order filter. The results are displayed in Figs. 4.7 and 4.8. The signal estimate from the 5-th order filter is slightly jagged, while the estimate from the 50-th order filter is smooth. The robustness of the filter is measured by the Mean Deviation from the Desired (MDD). The MDD of the 5-th order filter is 0.74 while the MDD of the 50-th order filter is 0.32.

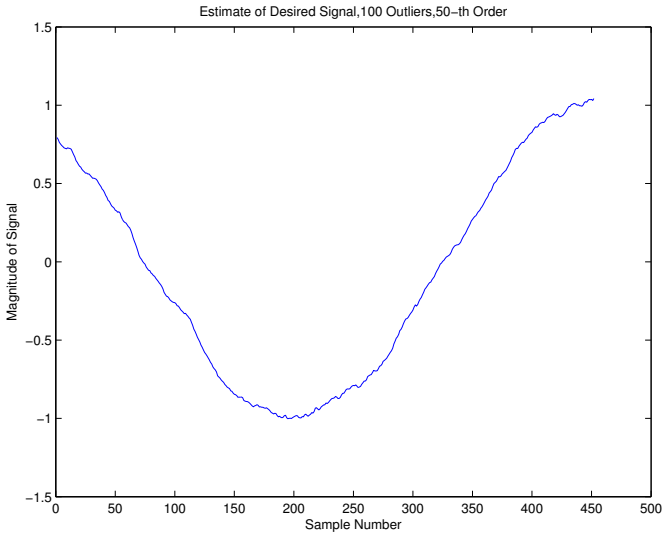

Figure 4.8: Signal Estimate of Fig. 4.4 using a 50-th Order Filter.

Table 4.2: Performance of the RCGMWF for Signal Estimation

Order	Breakdown Point	$\text{MDD}_{CleanSample}$	$\text{MDD}_{ContaminatedSample}$
1	37%	0.112	0.114
5	35%	0.047	0.053
10	26%	0.031	0.034
25	22%	0.020	0.024
50	22%	0.013	0.027

In general, the higher order filters provide a more robust estimate of the underlying deterministic signal, however, they require longer sequences of good data samples. Because of this, the higher order filters have a lower breakdown point. Conversely, lower order filters have a higher breakdown point at the expense of robustness.

The statistics for filters of orders 1, 5, 10, 25 and 50, for the CGMWF, are displayed in Table 4.2. The measured parameters are breakdown point, MDD of a clean sample and MDD of a sample contaminated with 1 outlier. Again, the connection between lower filter order and increasing breakdown point is verified. Furthermore, higher order filters lose more signal quality in the presence of outliers. From Table 4.2, it appears that a 5-*th* order filter is the optimal choice. The breakdown point is significantly higher than the 10-th order filter while only losing 11% accuracy when an outlier is present. Furthermore, the computational time is much better for lower order filters.

4.4 Problems with CGMWF with MCD for Outlier Detection

For the MCD, Table 4.1 shows the probability of obtaining a clean data set falls off quickly with increasing dimension of data and increasing fraction of contamination. Consequently, the computation time begins to increase exponentially. Therefore, the MCD is optimal for use on lower dimensional data with fewer outliers.

Chapter 5

STAP Radar Application

5.1 STAP Radar Overview

Space-Time Adaptive Processing (STAP) is used in many types of radar systems today. Among them are

1. Airborne Radar Systems
2. Automotive Radar Systems
3. Surveillance Radar Systems
4. Ground-Based Sense and Avoid Systems

Of these, airborne radar systems is the most popular use of STAP. The basic airborne STAP scenario is shown in Fig. 5.1. This figure displays the types of clutter that are present in the radar returns. STAP radar systems consider both spatial and temporal domains. For instance, a STAP system may have N antenna taps and processes M temporal snapshots. This gives a total of NM DOF for every range bin. Fig. 5.2 displays a data cube of a typical STAP radar system.

The advantage of considering both the spatial and temporal domains jointly is that the probability of detection is significantly increased. An adaptive filter of NM-order provides better noise filtering than just an N order or an M order filter alone. Furthermore,

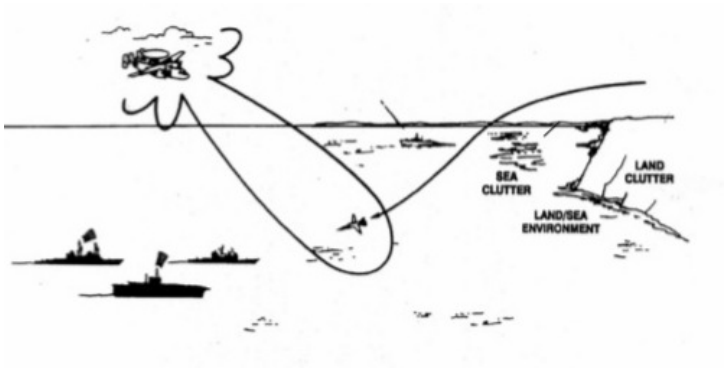

Figure 5.1: Basic Airborne STAP Radar Scenario (used with permission [31])

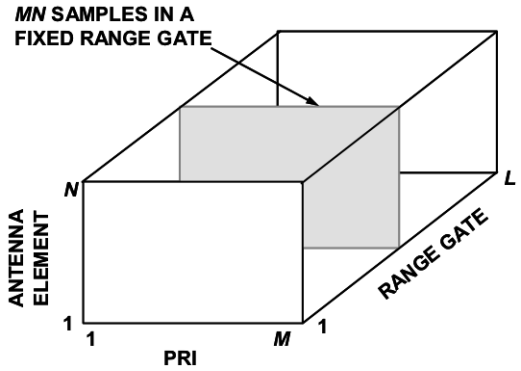

Figure 5.2: Data Cube for STAP Radar Analysis (used with permission [31])

STAP enables the radar system to effectively *listen* for signals originating in a particular direction while cancelling signals originating in other directions. The following section gives a brief history on the adaptive signal processing systems that provide the foundation of STAP radar systems.

5.2 Brief History of Adaptive Radar

Adaptive signal processing is a relatively new field. Prior to the 1970s, high-powered computing systems were nonexistant, therefore making significant advancements in adaptive signal processing research seemed impossible. Due to these limitations, development, testing and analysis of adaptive signal processing occured at a very slow pace.

In the early 1970s, with the advent of higher powered computing, Brennan and Reed, and Brennan, Reed and Mallett published a series of ground breaking works [3]-[7] that introduced the concept of STAP radar systems. These works were the first to propose

the idea that processing not only spatial data, but also temporal data increased the DOF of the system and thereby, provided greater ability to filter the noise and improve target detection. Furthermore, these works also introduced the concept of beamforming, which enabled adaptive radar system to detect targets at a particular angle, as mentioned above. This seminal work resulted in the foundation of adaptive beamforming radar systems that would influence the work in STAP radar for years to come.

One major drawback of this approach is its vulnerability to the presence of targets in the observation field. A reliable outlier detection algorithm is needed here, as shown later.

In 2000, Rangaswamy, Himed and Michaels [36] published a nonhomogeneity detector based on the generalized inner product (GIP). The GIP has been shown to detect nonhomogeneity in a test field and to remove single outliers. It is based on the sample covariance matrix. While it is able to detect a detecting single outlier, it is susceptible to the masking effect when multiple outliers are present in the observation field. It may also produce an elevated noise floor, resulting in a high probability of missing a target. This drawback further necessitates a robust outlier detection algorithm that is robust to multiple outliers in the observation field.

In 2007, Schoenig and Mili [44], [25], [27] proposed using the CGM-estimator with projection statistics for outlier detection and cancellation in STAP radar systems. This algorithm is capable of detecting multiple outliers while filtering out Gaussian noise. However, projection statistics and the CGM-estimator require the data to be entirely in the real domain. Additional processing time is needed to convert STAP radar data from its polar form into rectangular form. This conversion effectively doubled the dimension of the data which increased the computation time fourfold.

The limitations of Schoenig and Mili's work motivated the direction of the research in this thesis. The following sections discuss the use of the MCD and the linearly constrained minimum variance filter (LCMV) in STAP radar systems.

5.3 LCMV with MCD for MCARM Radar Data

In section 3.2.5, the optimal Wiener solution to the LCMV filter was developed. It is shown here for convenience

$$\mathbf{w}_0 = \frac{\mathbf{\Sigma}^{-1}\mathbf{s}\left(\theta_0\right)}{\mathbf{s}^H\left(\theta_0\right)\mathbf{\Sigma}^{-1}\mathbf{s}\left(\theta_0\right)}. \tag{5.1}$$

The multi-channel airborne radar measurement (MCARM) data used in this section consists of 630 range bins across 33 dimensions. This STAP radar systems consists of 11 spatial antenna elements and 3 temporal dimensions for a total of 33 dimensions for each range bin. Targets can be embedded at any location in the dataset. The algorithm for detecting targets involves applying the MCD algorithm from section 2.4 to detect the presence of outliers (targets). The steps involved in the applying the algorithm to the STAP radar problem are

1. Embed targets in the desired position;
2. Select a set of range bins to process;
3. Select a range bin to process as the CUT;
4. Extract 40 range bins leading and trailing the CUT;
5. Apply the MCD to the data to look for outliers in the data field;
6. Downweight any detected outliers and calculate the robust sample covariance matrix;
7. Compute the optimal LCMV filter weights;
8. Compute the reponse to the CUT;
9. If there are more range bins to process, return to step 3 to process the next range bin.

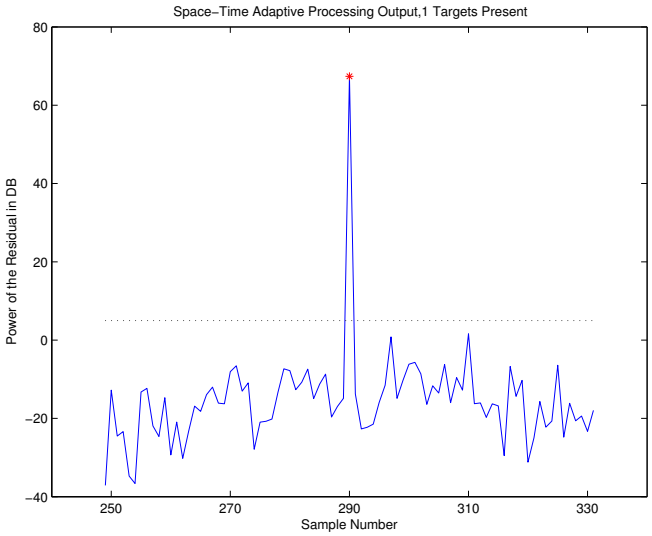

Figure 5.3: This shows the STAP Output power in dB versus Range Bin Number. In this case, a single outlier was used. The output was similar in both cases, when the MCD was and was not used for outlier detection.

Several tests were run with a varying number of targets embedded in the dataset. The output power in dB was collected, as well as the signal to interference plus noise ratio (SINR). Also, the LCMV filter is tested for its breakdown point and sample support requirement. The results are now presented.

In all cases, the dashed line represents the 5 dB threshold. Any power greater than 5 dB is considered a target.

In the first test, a single target was embedded at the 290th range bin. The output power of the LCMV filter is shown in Fig. 5.3. The same test was applied to the STAP data with 2 targets, 11 targets, 26 targets and 31 targets present. The results displayed in Figs. 5.4, 5.6 and 5.8 discuss cases where the MCD was successfully applied for outlier detection and where all targets were detected. Figs. 5.5 and 5.7 display the case where the MCD was not used for outlier detection and consequently, the CGMWF broke down. Fig. 5.9 displays the case where the MCD was used for target detection, but the CGMWF broke down regardless.

With 2, 11 and 26 targets present, the LCMV filter, with the MCD is able to detect the positions of all targets present. With 31 targets present, the LCMV misses some of the targets. This is due to the saturation of the field with many targets in close proximity. This shows the maximum level of resistance of the LCMV filter to outliers for

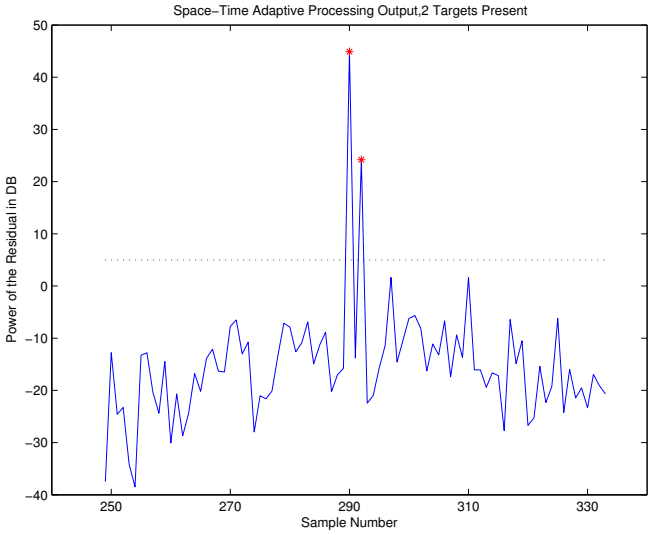

Figure 5.4: This shows the STAP Output power, 2 Targets with MCD Detection. The noise was successfully filtered.

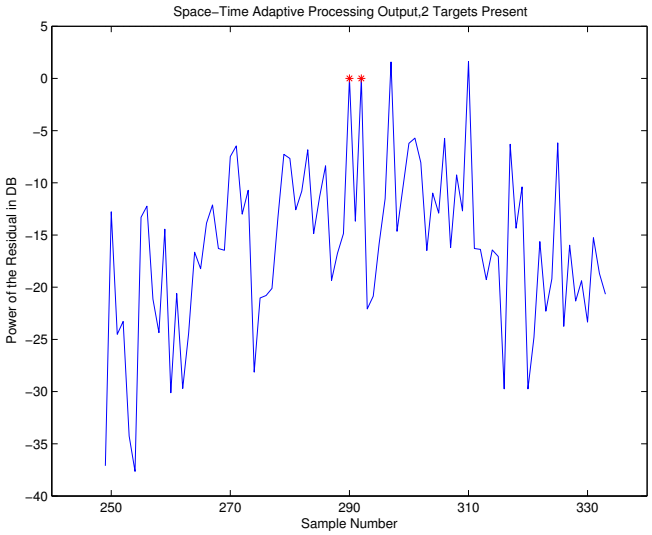

Figure 5.5: This shows the STAP Output power, 2 Targets without MCD Detection. The noise was not successfully filtered.

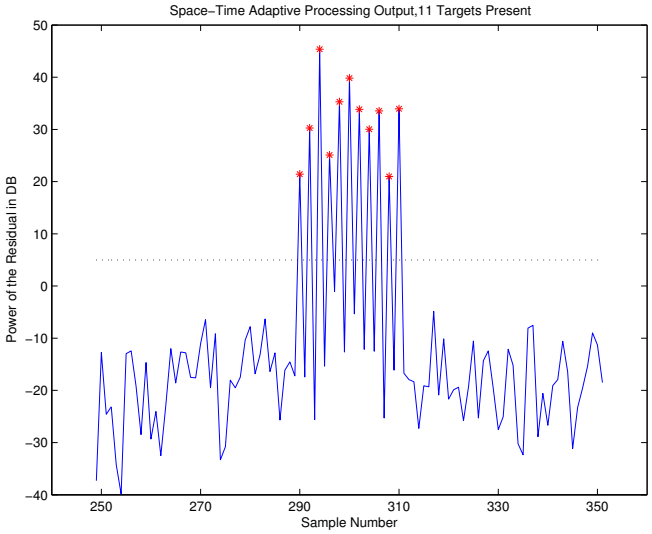

Figure 5.6: This shows the STAP Output power, 11 Targets with MCD Detection. The noise was successfully filtered.

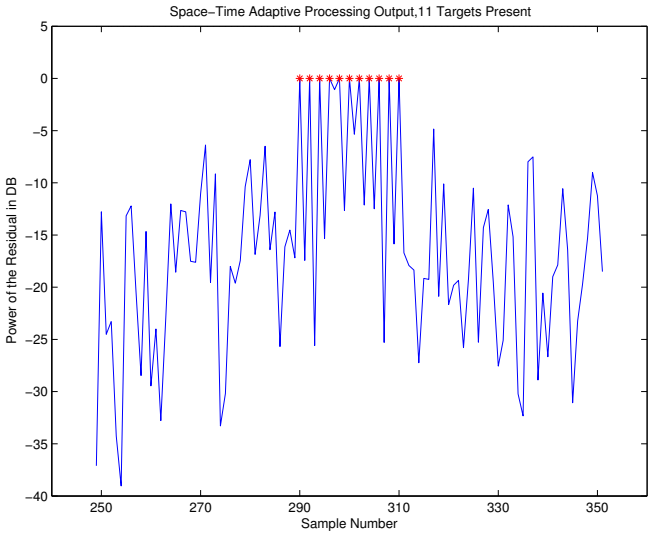

Figure 5.7: This shows the STAP Output power, 11 Targets without MCD Detection. The noise was not successfully filtered.

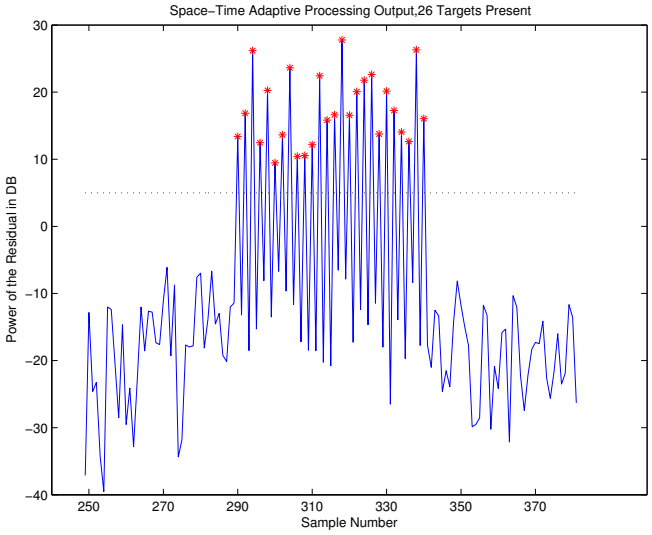

Figure 5.8: This shows the STAP Output power, 26 Targets with MCD Detection. The noise was successfully filtered.

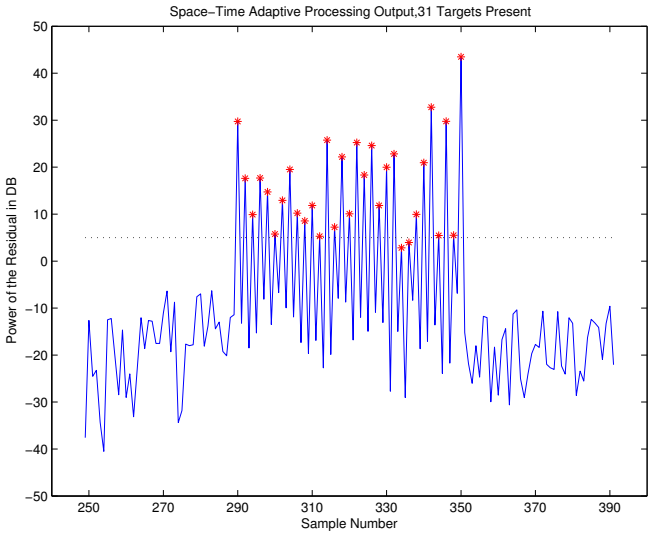

Figure 5.9: This shows the STAP Output power, 31 Targets with MCD Detection. In this case, there were too many targets present and some targets were missed.

Table 5.1: Performance of the CGMWF for STAP radar

Targets	Detected	SINR	$\text{SINR}_{w/outMCD}$	$m \vert P_{Contaminated} = 1\%$	BD	$\text{BD}_{w/outMCD}$
1	1	78.1	78.1	5	No	No
2	2	61.9	-68.6	9	No	Yes
11	11	46.9	-89.8	702	No	Yes
26	26	25.0	-100.8	2930334	No	Yes
31	29	26.2	-101.9	79733075	Yes	Yes

STAP radar. The compiled results are shown in Table 5.1.

Table 5.1 displays the performance of the CGMWF for STAP radar. Notice the CGMWF did not breakdown in both cases when only a single outlier is present. This is because STAP radar systems use guard cells to protect adjacent cells from spillover power.

5.4 Consideration for Sample Support on Signal-to-Interference-plus-Noise-Ratio Convergence

In the STAP radar example presented in the previous section, the stochastic properties of the background clutter were estimated for each range bin. However, the number of samples that were used to calculate the noise characteristics and filter weights was fixed at 80. This section investigates the effects of using a varying number of samples to detect 10 injected targets.

In [4], the sample size for optimizing performance to suppress clutter was found to be $K = 2N - 3$, where N is the dimension of the data. The theoretical optimal sample support for this STAP scenario is at $K = 63$. K is varied from 10 samples up to 200 samples. 10 targets were injected into the data at locations 260, 270, 272, 285, 290, 305, 322, 324, 326 and 340. The target detection threshold was set to 20 dB. Some results are shown in Figs. 5.10, 5.11, and 5.12.

At $K = 36$, some of the targets potentially stand out, however, the noise floor is above the 20 dB threshold. Using a sample size of 36 does not provide enough fidelity to accurately reveal the 10 targets. At this sample size, the detection algorithm would

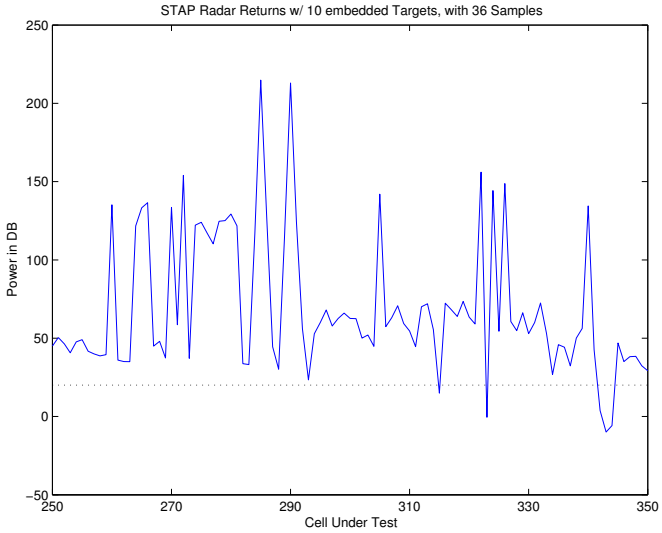

Figure 5.10: N = 33, 10 outliers, K = 36. In this case, the noise floor is too high and false targets are detected.

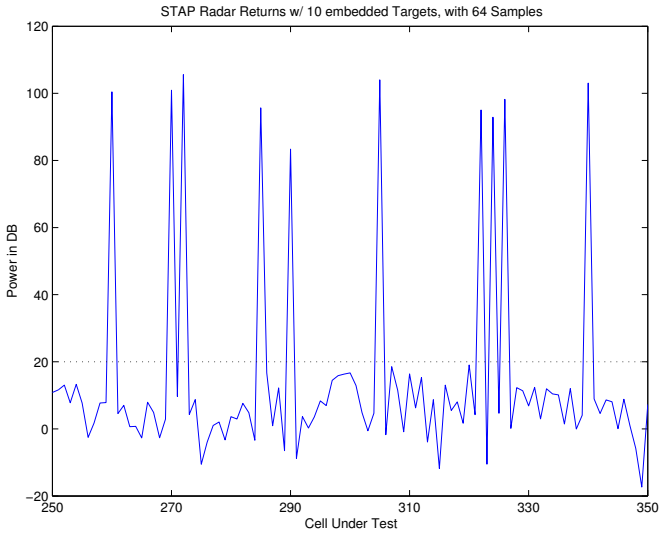

Figure 5.11: N = 33, 10 outliers, K = 64. The noise condition was suppressed and the targets were revealed.

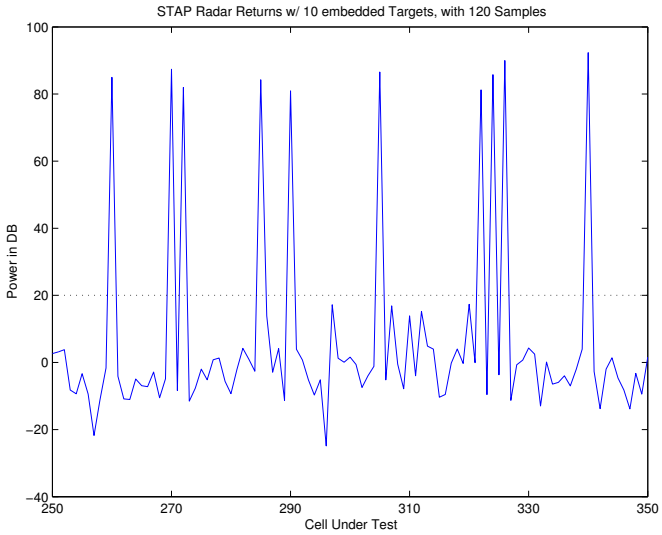

Figure 5.12: N = 33, 10 outliers, K = 120. The noise condition was suppressed and targets were revealed. However, there is not much further suppression beyond the theoretical optimal sample support.

Table 5.2: Performance of RCGMWF with varying degrees of Sample Support

Samples	Targets	Detected	SINR
36	10	0	-2.9
64	10	10	69.8
120	10	10	70.8

determine a target is present at nearly every CUT in the range bins of interest.

At $K = 64$, which is near the theoretical threshold, each of the 10 targets clearly stand out. All 10 targets are present and the noise is suppressed without any false detections present. At $K = 120$, the noise floor is nearly identical to that of $K = 64$. The average SINR for the target CUTs in all three cases is shown in Table 5.2.

5.5 Conclusion

The MCD is an effective algorithm for detecting outliers in STAP radar data. Furthermore, the LCMV filter is effective for filtering the noise once enough samples are used. However, as the number of targets increased, the LCMV filter began breaking down. In the case of 31 outliers, the MCD still detected all of the targets, but the LCMV filter broke down. The output power of some of the targets fell below the 5 dB threshold. The results indicate that the breakdown point of the LCMV filter is approximately 35%.

Furthermore, the the number of random datasets needed to ensure a $P_{CleanSubset} = 99\%$ increased as the fraction of contamination increased. This increases the computation time of the MCD, which may present some limitations to consider when employing the MCD in STAP radar systems.

As for sample support, the algorithm continued to improve up to the theoretical optimum sample size. For this STAP system, near optimal performance was reached at about the theoretical optimal sample support of $K = 2N - 3$. This is because a minimum number of samples is needed before the MCD becomes reliable. The minimum number of samples is tied to the dimension of the data. Added redundancy does not seem to improve the SINR.

Chapter 6

Conclusions

The MCD algorithm was employed to detect outliers in a data set. It was used in place of the projection statistics algorithm that was used in the prior work [44]. The limitation of projection statistics for outlier detection is that it requires splitting the data into its real and imaginary parts, respectively when the data is complex-valued. This introduces several problems. First, the complexity is greater because both the real and imaginary parts must be treated independently. Secondly, the computation time increases because the dimension of the data is doubled. This results in many more computations, thereby increasing computation time.

The MCD algorithm addressed all of these limitations. First, the MCD algorithm processes data entirely in the complex domain. This reduces the number of computations that are necessary for detecting outliers. Secondly, the MCD is very robust. It has a breakdown point that approaches 50% with sufficient sample support.

The performance of the CGMWF and LCMV filters was enhanced when the MCD algorithm was used for outlier detection. In the CGMWF case, the MCD detected every outlier up to the maximum fraction of contamination of 50%. The limitation for the CGMWF was the GM-Huber estimator. It broke down at approximately 22% for filters of 25-th order and higher. However, its breakdown point attains 35% for only a 5-th order filter. Therefore, the 5-th order CGMWF is recommended to maximize the breakdown point while retaining a good accuracy.

In the STAP radar example, the MCD was able to detect every outlier up to the maximum fraction of contamination at 35%. The CGMWF, however, broke down at this point. This is because the abundance of downweighted outliers in the sample used to calculate the covariance matrix effectively decreased the sample support available for calculating the optimal filter weights.

In conclusion, while the MCD is capable of detecting up to 50% outliers and enables processing completely in the complex domain, the MCD did not significantly enhance the Wiener filter beyond what was already accomplished in [44]. When the dimension of the data is much smaller than the size of the subset, the MCD provides exceptional results very quickly. This effectively decreases computation time.

A potential future research topic is to improve the algorithm that implements the Robust Wiener filter to decrease the computating time for high dimensional data. Other research topics include the application of this filter to cognitive radio, pattern recognition, biometrics and internet security systems, to cite a few.

By merging robustness theory with adaptive signal processing, the research in this thesis has been an enjoyable and rewarding experience. The knowledge gained will provide a foundation for our future work.

Bibliography

[1] Bandiera, Francesco, Antonio De Maio and Giuseppe Ricci. Adaptive CFAR Radar Detection With Conic Rejection. IEEE Transactions on Signal Processing. Vol. 55, No. 6, pp. 2533-2541. June 2007.

[2] Bishop, Adrian N., Pubuda N. Pathirana and Audrey V. Savkin. Radar Target Tracking via Robust Linear Filtering. IEEE Signal Processing Letters, Vol. 14, No. 12, pp. 1028-1031. December 2007.

[3] Brennan, Lawrence E. and Irving S. Reed. Theory of Adaptive Radar. IEEE Transactions on Aeorspace and Electronic Systems, Vol. AES-9, No. 2, pp. 237-253. March 1973.

[4] Reed, Irving S., John D. Mallett and Lawrence E. Brennan. Rapid Convergence Rate in Adaptive Arrays. IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-10, No. 6, pp. 853-863. November 1974.

[5] Davis, Roderic C., Lawrence E. Brennan and Irving S. Reed. Angle Estimation With Adaptive Arrays in External Noise Fields. IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-12, No. 2, pp.179-186. March 1976.

[6] Brennan, Lawrence E., John D. Mallett and Irving S. Reed. Adaptive Arrays in Airborne MTI Radar. IEEE Transactions on Antennas and Propagation, Vol. AP-24, No. 5, pp. 607-615. September 1976.

[7] Brennan, Lawrence E. and Irving S. Reed. An Adaptive Array Signal Processing Algorithm for Communications. IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-18, No. 1, pp. 124-130. January 1982.

[8] Chambers, J.A. and Apostolos Avlonitis. A Robust Mixed-Norm Adaptive Filter Algorithm. IEEE Signal Processing Letters, Vol. 4, No. 2, pp. 46-48. February 1997.

[9] Chambers, J.A., O. Tanrikulu and A.G. Constantinides. Least Mean Mixed-Norm Adaptive Filtering. IEE Electronics Letters, Vol. 30, No. 19, pp. 1574-1575. September 1994.

[10] Fisher R.A. Theory of statistical estimation. Proc. Cambridge Philos. Soc., vol. 22, 1925.

[11] Gasko M. and Donoho D. Influential Observation in Data Analysis. American Statistical Association, 1982, Proc. of the Business and Economic Statistics Section, pp. 104-110.

[12] Ghouz, H.H.M., F.I.A. Elghany and M.M. Qutb. Adaptive Space-Time Processing for Interference Suppression in Phased Array Radar Systems (Part 1). Seventeenth National Radio Science Confernce, Feb. 22-24, 2000, Minufiya University, Egypt. 2000.

[13] Griffiths, Lloyd J. and Charles W. Jim. An Alternative Approach to Linearly Constrained Adaptive Beamforming. IEEE Transactions on Antennas and Propagation, Vol. AP-30, No. 1, pp. 27-34. January 1982.

[14] Hampel F.R., Contributions to the theory of robust estimation, Ph.D. thesis, University of California, Berkeley, 1968.

[15] Hampel F.R., Ronchetti E.M., Rousseeuw P.J., and Stahel W.A., Robust Statistics: The Approach based on Influence Functions, Wiley, New York, 1986.

[16] Haykin, Simon. Adaptive Filter Theory, 4th Edition. Pearson Education Inc., 2004.

[17] Himed, Braham, Yassir Salama and James H. Michels. Improved Detection of Close Proximity Targets using Two-Step NHD. IEEE International Radar Conference, pp. 781-786. 2000.

[18] Hoaglin, David C., Frederick Mosteller and John W. Tukey. Understanding Robust and Exploratory Data Analysis. John Wiley and Sons, Inc. Hoboken, NJ. 1983.

[19] Huber P.J. Robust estimation of a location parameter. Ann. Math. Statist., vol. 35, pp. 73-101, 1964.

[20] Kim, Seung-Jim and Ronald A. Iltis. GPS C/A Code Tracking with Adaptive Beamforming and Jammer Nulling. IEEE Transactions, pp. 975-979. 2002.

[21] Kulpa, Krzysztof, Jacek Misiurewicz, Zbigniew Gajo and Mateusz Malanowski. A Simple Robust Detection of Weak Target in Noise Radars. Proceedings of the 37th European Microwave Conference, pp. 1554-1557. October 2007.

[22] Leon-Garcia, Alberto. Probability and Random Processes for Electrical Engineering. Addison-Wesley Publishing Company, Inc., 1994.

[23] Liu, Weifeng, Puskal P. Pokharel and Jose C. Principe. Correntropy: Properties and Applications in Non-Gaussian Signal Processing. IEEE Transactions on Signal Processing, Vol. 55, No. 11, pp. 5286-5298. November 2007.

[24] Melvin, William L. and Michael C. Wicks. Improving Practical Space-Time Adaptive Radar. IEEE National Radar Conference, pp. 48-53. 1997.

[25] Mili, Lamine, M.G. Cheniae, N.S. Vichare and Peter J. Rousseeuw. Robust State Estimation Based on Projection Statistics. IEEE Transactions on Power Systems, Vol. 11, No 2, pp. 1118-1127. May 1996.

[26] Mili, Lamine. ECE 5714, Robust Estimation and Filtering, Spring 2008 Class Notes.

[27] Picciolo, Michael L., Gregory Schoenig, Karl Gerlach and Lamine Mili. Robust Cascade Canceller Using Projection Statistics for Adaptive Radar. IEEEAC Paper No. 1632, Version 3. December 2004.

[28] Moon, Todd K. and Wynn C. Stirling. Mathematical Methods and Algorithms for Signal Processing. Prentice Hall. Upper Saddle River, NJ. 2000.

[29] Muth, Lorant A. C.M. Wang and Timothy Conn. Robust Separation of Background and Target Signals in Radar Cross Section Measurements. IEEE Transactions on Instrumentation and Measurement, Vol. 54, No. 6, pp. 2462-2468. 2005.

[30] Ogle, William C. et al. A Multistage Nonhomogeneity Detector. 2003 IEEE Radar Conference, pp. 121-125. 2003.

[31] Picciolo, M.L., Robust Adaptive Processors, Ph.D. thesis, Virginia Tech, May 2003.

[32] Picciolo, Michael L., Gregory N. Schoenig, Karl Gerlach and Lamine Mili. Robust Cascaded Canceller Using Projection Statistics for Adaptive Radar. IEEEAC paper 1632, Ver. 3. 2005.

[33] Pokharel, Puskal P., Jian-Wu Xu, Deniz Erdogmus and Jose C. Principe. A Closed Form Solution for a Nonlinear Wiener Filter. ICASSP 2006, pp. (III-720)-(III-723). 2006.

[34] Poor, H. Vincent. On Robust Wiener Filtering. IEEE Transactions on Automatic Control, Vol. AC-25, No. 3, pp. 531-536. June 1980.

[35] Poor, H. Vincent. Robust Matched Filters. IEEE Transactions on Information Theory, Vol. IT-29, No. 5, pp. 677-687. September 1983.

[36] Rangaswamy, Muralidhar, Braham Himed and James H. Michels. Performance Analysis of the Nonhomogeneity Detector. IEEE Transactions. PP. 1117-1121. 2000.

[37] Rangaswamy, Muralidhar, Braham Himed and James H. Michels. Performance Analysis of the Nonhomogeneity Detector for STAP Applications. IEEE Transactions. PP. 193-196. 2001.

[38] Rangaswamy, Muralidhar, Freeman C. Lin and Karl R. Gerlach. Robust Adaptive Signal Processing Methods for Heterogeneous Radar Clutter Scenarios. 2003 IEEE Radar Conference, pp. 265-272. 2003.

[39] Richards, Mark A. Ph.D. Fundamentals of Radar Signal Processing. McGraw-Hill Companies, Inc., 2005.

[40] Rousseeuw, Peter J. and Annick M. Leroy. Robust Regression and Outlier Detection. John Wiley and Sons, Inc. Hoboken, NJ. 2003.

[41] Rousseeuw, Peter J. and Bert C. Van Zomeren. Robust Distances: Simulations and Cutoff Values. Directions in Robust Statistics and Diagnostics, Part II. Eds. W.

Stahel and S. Weisberg. The IMA Volumes in Mathematics and its Applications, Vol. 34. Springer-Verlag, New York, 1991.

[42] Croux, Christophe and Peter J. Rousseeuw. Time-Efficient Algorithms for Two Highly Robust Estimators of Scale. Computational Statistics, Volume 1, pp. 411-428. Eds. Y. Dodge and J. Whittaker. Heidelberg: Physika-Verlag, 1992.

[43] Rousseeuw, Peter J. and Katrien van Driessen. A Fast Algorithm for the Minimum Covariance Determinant Estimator. Technometrics, Vol. 41, No. 3, pp. 212-223. August 1999.

[44] Schoenig, Gregory N. Contributions to Robust Adaptive Signal Processing with Application to Space Time Adaptive Radar. Ph.D dissertation, Virginia Tech. 2007.

[45] Schumann, H.K., P.G. Li, W. Szczepanski and J. Graham. Space-Time Adaptive Processing for Space Based Radar. 2004 IEEE Aerospace Conference Proceedings, pp. 1904-1910. 2004.

[46] Smith, Larry. Linear Algebra, 3rd Edition. Springer-Verlag New York, Inc., 1998.

[47] Stahel W. A., Robuste Schatzungen: Infinitesimale Optimalitat und Schatzungen von Kovarianzmatrizen, Ph.D. thesis, E.T.H. Zurich, Switzerland, 1981.

[48] SwindleHurst, A. Lee and Petre Stocia. Maximum LikelihoodMethods in Radar Array Signal Processing. Proceedings of the IEEE, Vol. 86, No. 2, pp. 421-441. February 1998.

[49] Van Veen, Barry D. Soft-Constrained Minimum Variance Beamforming. IEEE Transactions, pp. 2811-2814. 1990.

[50] Van Veen, Barry D. and Kevin M. Buckley. Beamforming: A Versatile Approach to Spatial Filtering. IEEE ASSP Magazine, pp. 4-24. April 1988.

[51] Vastola, Kenneth S. and H. Vincent Poor. Robust Wiener- Kolmogorov Theory. IEEE Transactions on Information Theory, Vol. IT-30, No. 2, pp. 316-327. March 1984.

[52] Verdu, Sergio and H. Vincent Poor. On Minimax Robustness: A General Approach and Applications. IEEE Transactions on Information Theory, Vol. IT-30, No. 2, pp. 328-340. March 1984.

[53] Shing-Chow, Chan and Zou Yuexian. Convergence Analysis of the Recursive Least Squares M-Estimate Adaptive Filtering Algorithm for Impulse Noise Suppression. IEEE Transactions on Digital Signal Processing, pp.663-666. 2002.

[54] Shing-Chow, Chan and Zou Yuexian. A Recursive Least M-Estimate Algorithm for Robust Adaptive Filtering in Impulsive Noise: Fast Algorithm and Convergence Performance Analysis. IEEE Transactions on Signal Processing, Vol. 52, No. 4, pp.975-991. April 2004.

[55] Yuexian, Zou, Chan Shing-Chow and Ng Tung-Sung. A Robust M-Estimate Adaptive Equalizer for Impulse Noise Suppression. IEEE Transactions, pp. 2393-2397. 1999.

[56] Yuexian, Zou, Chan Shing-Chow and Ng Tung-Sung. A Robust M-Estimate Filter for Impulse Noise Suppression. IEEE Transactions, pp. 1765-1768. 1999.

[57] Yuexian, Zou, Chan Shing-Chow and Ng Tung-Sung. A Recursive Least M-Estimate (RLM) Adaptive Filter for Robust Filtering in Impulsive Noise. IEEE Signal Processing Letters, Vol. 7 No. 11, pp. 324-326. 2000.

[58] Yuexian, Zou, Chan Shing-Chow and Ng Tung-Sung. Least Mean M-Estimate Algorithms for Robust Adaptive Filtering in Impulsive Noise. IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 47 No. 12, pp. 1564-1569. December 2000.

[59] Yuexian, Zou and Chan Shing-Chow. A Robust Quasi-Newton Adaptive Filtering Algorithm for Impulse Noise Suppression. IEEE Transactions pp. (II-667)-(II-680). 2001.

[60] Yuexian, Zou, Chan Shing-Chow and Ng Tung-Sung. Robust M-Estimate Adaptive Filtering. IEEE Proceedings on Visual and Image Signal Processing, Vol. 148, No. 4, pp. 289-294. August 2001.

[61] Yuexian, Zou and Chan Shing-Chow. A Huber Recursive Least Squares Adaptive Lattice Filter for Impulse Noise Suppression. IEEE Proceedings, pp. 3769-3772. 2001.