

DATA COMPRESSION FOR RADAR SIGNALS:
AN SVD BASED APPROACH

BY

ZHEN ZHOU

B.S.E.E., Tsinghua University, 1998

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

AT

STATE UNIVERSITY OF NEW YORK AT BINGHAMTON
BINGHAMTON

MAY 2001

STATE UNIVERSITY OF NEW YORK AT BINGHAMTON
DEPARTMENT OF
ELECTRICAL ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled “**Data Compression for radar signals: An SVD based approach**” by **Zhen Zhou** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: May 2001

Supervisor:

Mark L. Fowler

Readers:

Edwards X. Li

Eva N. Wu

STATE UNIVERSITY OF NEW YORK AT
BINGHAMTON

Date: **May 2001**

Author: **Zhen Zhou**

Title: **Data Compression for radar signals: An SVD
based approach**

Department: **Electrical Engineering**

Degree: **M.Sc.** Convocation: **May** Year: **2001**

Permission is herewith granted to State University of New York at Binghamton to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

THE AUTHOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION.

THE AUTHOR ATTESTS THAT PERMISSION HAS BEEN OBTAINED FOR THE USE OF ANY COPYRIGHTED MATERIAL APPEARING IN THIS THESIS (OTHER THAN BRIEF EXCERPTS REQUIRING ONLY PROPER ACKNOWLEDGEMENT IN SCHOLARLY WRITING) AND THAT ALL SUCH USE IS CLEARLY ACKNOWLEDGED.

Abstract

Multiple platform coherent location systems operate by computing the time difference of arrival (TDOA) and frequency difference of arrival (FDOA) among signals received at geographically separated platforms. The bandwidth of the data link is the major bottleneck in the processing. Previously developed data compression methods [1, 2] can not satisfy the compression ratio and the accuracy requirements because they were designed for the generic signal case and do not fully exploit the characteristics of the radar signal.

A new compression scheme presented in this thesis is built from the ground up with the characteristics of the radar signal in mind. It is based on the idea that a radar pulse train can be modelled as one prototype pulse and a parameter vector for each pulse to transform the prototype pulse to each specific pulse. A compression ratio of $10 \sim 20 : 1$ has been achieved with minor, if any, FDOA/TDOA accuracies degradation in most cases. The two major techniques involved here are the fractional delay filter and the singular value decomposition (SVD).

This thesis starts with some preliminary technical background used later in this thesis. Then two chapters are dedicated to the fractional delay filter and the SVD method, respectively. In addition to presenting and verifying our compression scheme, a newly developed LMS adaptive FIR fractional delay filter and an alternative to the cross-ambiguity processing based on the parameterization method are developed. Extensive simulation results are presented throughout the thesis. In the last chapter, conclusions and suggestions for future work are given.

To Jingzhou

Acknowledgements

I would like to thank Prof. Mark Fowler, my supervisor, for his guidance, support and many suggestions during this research. I am also thankful to Anupama Shivaprasad, my colleague, for her friendship, patience and the insightful discussion between us.

Prof. Wu and Prof. Li have given me many valuable suggestions during the writing of this thesis.

I should also mention that this research project is supported by Lockheed Martin Federal Systems, Owego, New York.

Of course, I am grateful to my parents for their patience and *love*. Without them this work would never have come into existence.

Hat off to the people who developed the wonderful \LaTeX and *Ghostschrift* software and made them free. They make the writing of professional documents a joyful task.

Finally, I wish to thank the following: Shu, Qian (for their friendship); Ruofei, Hai, Jie, Kin, Jinmin, Donghui, Qing, Sanner, Tao, Mary, Wendy, Li, Edwards, Jia, Jabu ... (for all the good and bad times we had together); Uma and Raman (for their tolerance of the mess I made); and The Scorpions (for their inspiring music); and my sister and her husband (can you extend the loan?); Elaine (she is my angel).

Binghamton, New York
May 2, 2001

Zhen Zhou

Table of Contents

Abstract	iv
Acknowledgements	vi
Table of Contents	vii
List of Figures	ix
Introduction	1
1 Background	3
1.1 Multiple Platform Coherent Emitter Location Systems	3
1.2 Lossy Data Compression	7
1.3 The Radar Signal	9
1.3.1 Review of the Radar Theory	10
1.3.2 The Linear-FM Radar Pulse Train	10
1.3.3 Insights into Compressing the Data	12
2 The Fractional Delay Filters	15
2.1 Frequency Domain Methods	16
2.1.1 The Brute Force Fourier Method	16
2.1.2 The Block Fourier Method	17
2.2 The Filtering Methods	20
2.2.1 The Lagrange FIR Filter	20
2.2.2 The All-pass Thiran Filter	23
2.3 The Adaptive FIR Filter	25
2.3.1 The LMS FIR Adaptive FD Filter	26
2.4 Comparisons	30

3	The SVD Method	32
3.1	Review of the SVD Theory	32
3.1.1	Matrix Manipulation by SVD	33
3.1.2	SVD Denoising	35
3.2	Parameterizing the Pulses	37
3.2.1	Pulse Extraction using SVD	37
3.2.2	Simulation Results	40
3.3	Non-coherent Method	43
3.3.1	Direct Estimation of the FDOA/TDOA	44
3.3.2	Simulation Results	46
3.4	Parameters Encoding	49
4	Conclusions and Suggestions for Future Work	53
4.1	Conclusions	53
4.2	Suggestions for Future Work	54
	Bibliography	56

List of Figures

1.1	Multiple Platform Coherent Emitter Location Systems	4
1.2	Pulse gating and thinning	11
1.3	A pulse from the pulse train	12
1.4	10 pulses after processing	13
2.1	The diagram of the BFFD	17
2.2	Blocky Effect of the BFFD	18
2.3	The BFFD under band-limited white noise input	19
2.4	Weighted sum of N samples	20
2.5	Lagrange FIR FD filters responses	21
2.6	The Lagrange filter under band-limited white noise	22
2.7	The Thiran filters' group-delay responses	23
2.8	The Thiran filter under band-limited white noise	24
2.9	The adaptive filter	25
2.10	The forward-reverse filter	27
2.11	The snapshot of the adaptive filter's response	28
2.12	The adaptive filter under sinusoid input	29
2.13	The adaptive filter under band-limited white noise	30
3.1	Denoising using SVD	36
3.2	Singular values before and after alignment	38
3.3	FDOA/TDOA accuracy (coherent, A noise free)	41
3.4	FDOA/TDOA accuracy (coherent, A 20dB)	42

3.5	Pulse-to-pulse mapping	44
3.6	FDOA/TDOA accuracy (non-coherent, A noise free)	47
3.7	FDOA/TDOA accuracy (non-coherent, A 20dB)	48
3.8	FDOA/TDOA accuracy (coherent, A 20dB)	51
3.9	FDOA/TDOA accuracy (non-coherent, A 20dB)	52

Introduction

This thesis presents an advanced data compression scheme specially developed for compressing the radar signal collected in a coherent emitter location system. In order to compress the radar signal, two major techniques are involved: the fractional delay filter, which is used to align pulses to within a fraction of the sampling interval and the singular value decomposition, which is used to extract the prototype pulse from a matrix of aligned pulses.

Chapter 1 discusses some preliminary technical background used later in this thesis. Section 1.1, “Multiple Platform Coherent Emitter Location Systems,” presents the overall image of such a system and describes the cross-ambiguity processing which is used to estimate the frequency difference of arrival (FDOA) and the time difference of arrival (TDOA). Section 1.2, “Lossy Data Compression,” provides some background on lossy data compression. Section 1.3, “The Radar Signal,” analyzes the characteristics of the signal of interest, specifically the linear-FM radar pulse train, and gives a rough idea on how to compress it.

Chapter 2 discusses various techniques to design a fractional delay filter. Section 2.1, “Frequency Domain Methods,” describes the brute force Fourier method and the block Fourier method. Section 2.2, “The Filtering Methods,” describes some time domain design methods such as the Lagrange filter and the Thiran filter. Section 2.3

“The Adaptive FIR filter,” presents a newly developed variant of the Lagrange filter whose coefficients are adapting to the signal it deals with. All those methods are compared in section 2.4.

Chapter 3 describes the data compression scheme developed for use in the coherent emitter location systems. Section 3.1, “Review of the SVD theory,” briefly reviews the singular value decomposition (SVD) used later in this chapter. Section 3.2, “Parameterizing the Pulses,” describes the use of SVD to compress and parameterize the radar pulse train. Section 3.3, “The Non-coherent Method,” presents a newly developed alternative to the cross-ambiguity processing based on the parameterization method that we used to compress the data. Section 3.4, “Parameters Encoding,” briefly describes the quantization and coding after the parameterization. Simulation results are presented throughout this chapter.

The techniques developed in this thesis are very useful in this application as well as some others. In the final chapter, conclusions and suggestions for future work are presented.

Chapter 1

Background

1.1 Multiple Platform Coherent Emitter Location Systems

Multiple platform coherent location systems operate by computing the time difference of arrival (TDOA) and frequency difference of arrival (FDOA) among signals received at geographically separated platforms. One way to compute the TDOA/FDOA between two signals is to do cross ambiguity processing[3] on them.

Usually, 3 or more platforms are needed in a emitter location system as illustrated in Fig. 1.1. Each of them has identical receiving and processing equipment, and is capable of intercepting the signal of interest and performing the emitter location. Once those platforms have detected an emitter that is desired to be located, they start to collect the signal on this emitter simultaneously. Based upon the quality of the received data [4], some platforms are chosen to transmit the data they collected to other platforms via the inter-platform data-link. Then the other platforms can estimate the FDOA/TDOA between the signal they collected themselves and the signal collected at the other platforms. Since the location and velocity of the platforms

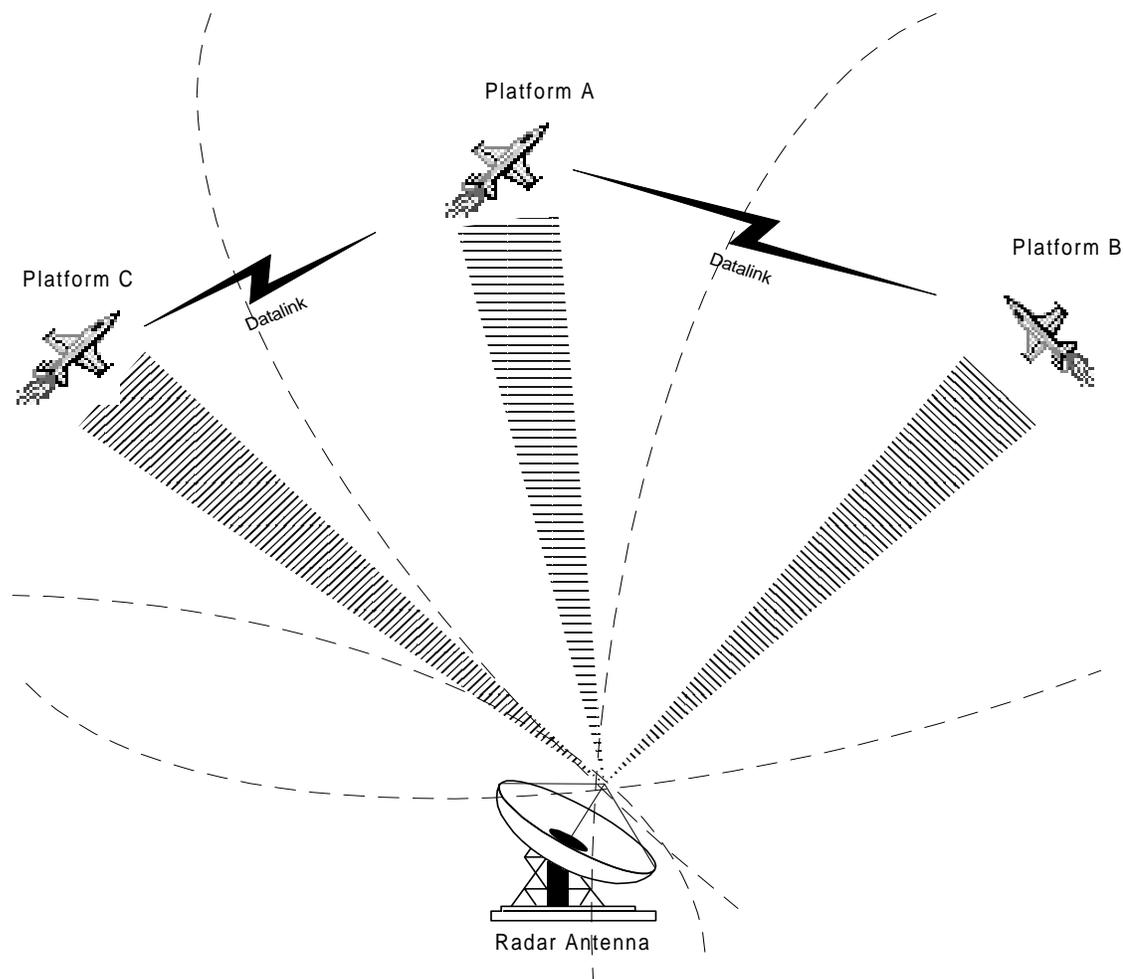


Figure 1.1: Multiple Platform Coherent Emitter Location Systems

are known (possibly by means of GPS,) those sets of FDOA/TDOA can be used to locate the emitter of interest geographically [4].

The accuracy of the location estimate depends on the accuracies of the underlying FDOA/TDOA estimations. In [4] it was shown that proper FDOA/TDOA processing requires at least several tens of radar pulses to be collected, each contains several tens to a few hundreds of samples. The total amount of data puts a heavy burden onto the data links, which usually are not dedicated to such a task.

The two signals received at separated platforms are usually noisy. After being transformed to complex base-band and quantized, they can be modelled by:

$$\begin{aligned}\tilde{x}_A(k) &= x_A(k) + n_1(k) \\ \tilde{x}_B(k) &= x_B(k) + n_2(k)\end{aligned}\tag{1.1.1}$$

where $x_A(k)$ and $x_B(k)$ are the desired complex-valued signal samples and $n_1(k)$ and $n_2(k)$ are the complex-valued noises added in the two measured signals. For simplicity's sake, we regard them as additive, white, Gaussian noise (AWGN). The symbols SNR_A and SNR_B are used to denote the signal-to-noise ratios of the two measured signals $\tilde{x}_A(k)$ and $\tilde{x}_B(k)$, respectively.

Prior to sampling, the signal $x_A(t)$ can be modelled as a time-shifted and Doppler-shifted version of $x_B(t)$:

$$x_A(t) = \alpha x_B(t + D)e^{j(2\pi f_d t + \phi)}\tag{1.1.2}$$

where the real numbers of f_d and D are the FDOA/TDOA to be estimated, the attenuation α and the phase offset ϕ are of less interest. One way to estimate the FDOA (f_d) and the TDOA (D) is to find the peak of the cross-ambiguity function of the two measured signals [3]:

$$A_{sd}(\tau, \nu) = \int_{-\infty}^{\infty} \tilde{x}_A(t) \tilde{x}_B^*(t - \tau) e^{-j2\pi\nu t} dt\tag{1.1.3}$$

where the superscript asterisk denotes complex conjugation. In practice, only the lattice points of the cross-ambiguity surface are computed and interpolation is used to find the location of the peak (τ_m, ν_m) of $|A_{sd}(\tau, \nu)|$. The accuracy with which the FDOA/TDOA estimates can be made are characterized by the standard deviations of the two estimates. It is well known that those standard deviations are bounded by

the so-called *Cramer-Rao* bounds [3] as given below:

$$\begin{aligned}\sigma(\tau_m) &\geq \frac{1}{2\pi B_{rms}\sqrt{SNR_o}} \\ \sigma(\nu_m) &\geq \frac{1}{2\pi D_{rms}\sqrt{SNR_o}}\end{aligned}\tag{1.1.4}$$

where B_{rms} is the RMS measure of the signal's bandwidth in Hz, D_{rms} is the RMS measure of the signal's duration in seconds, and SNR_o is the non-dB value of the SNR at the peak of the cross-ambiguity surface, which is given by:

$$SNR_o = \frac{B_{rms}D_{rms}SNR_A SNR_B}{1 + SNR_A + SNR_B}\tag{1.1.5}$$

It should be noticed that SNR_o is dominated by the smallest of SNR_A and SNR_B . This fact gives us insight to the possibility of data compression because degradation of one signal will not compromise the SNR_o too much while the SNR of the other signal is low.

In practice, there is a serious drawback of the cross ambiguity processing: the signal received at one platform must be totally transmitted to the other platform in order to perform the processing, and usually the rate of the data link is insufficient to accomplish this in a practical amount of time. To mitigate this problem, various data compression approaches [1, 2] have been proposed to reduce the amount of data to be transmitted. However, lossy compression will inevitably introduce distortions to the already noisy signal; higher compression ratio of each compression scheme will yield higher distortions and degrade the performance of the FODOA/TDOA estimator further. Therefore, we must deliberately make the trade-off between the compression ratio and degradation of the performance of the FODOA/TDOA estimator and specify our requirement in both terms:

- The degradation of the FDOA/TDOA estimation accuracies should be marginal

after compression. Specifically, the estimator should remain unbiased; and the standard deviations of the two estimates should not exceed twice as much as those from the uncompressed data.

- The higher the compression ratio the more favorable. To make the system practical, a compression ratio of at least 10:1 [5] should be achieved.

However, those previously developed methods [1, 2] can not match those specifications. One reason is that they were designed for the generic signal case and do not fully exploit the characteristics of the radar signal.

1.2 Lossy Data Compression

Data compression techniques fall into two main categories: lossless compression and lossy compression. Lossless techniques ensure total recovery of the original data by relying solely on the statistical characteristics of the data. However, in the area of compressing signal, lossless techniques usually deliver unsatisfactory compression ratio because of their often too strict nature. Sometimes it is favorable to sacrifice a little bit of the quality to get better compression in return. In contrary to the lossless approaches, lossy techniques attempt to apply an appropriate model upon the underlying signal and reduce the amount of data by simplifying the model in a proper way. The so called *principle of parsimony*[6] applies:

- A model should be complicated enough to reproduce the most important properties of the signal;
- But simple enough to resist the spurious effects that are associated with the use of the model.

Many of the lossy compression techniques widely used in today’s audio/video systems exploit the fact that humans can not perceive all the details in the original signal. Likewise, a good compression scheme for the coherent emitter location system should care less about things that do not affect the FDOA/TDOA measurement accuracies much.

The simplest lossy compression technique is quantization, which is basically a mapping process from a large—possibly infinite—set of values to a much smaller set [7]. For every codeword generated by the coder, a reconstruction value is generated by the decoder to best (in some sense) represent all the possible values corresponding to this codeword. Some of the information contained in the source data is permanently lost; that is where the word “lossy” comes from. The subject of quantization is well-researched; it is widely used in conjunction with other techniques to form more sophisticated lossy compression schemes.

Signals can be represented in various transformation spaces besides their original form. Although they are equivalent to each other, it may be more favorable to use the signal in one of the transformation spaces because:

- In the transformation space the components of the signal could demonstrate some non-uniform statistical characteristics which can be easily exploited by various lossless compression techniques;
- More importantly, human’s perception is not uniformly sensitive to all the components in the transformation space. Therefore, those components can be quantized differently in a way that the introduced distortion—possibly quite large—is nearly imperceptible.

Some of the most widely used transforms are: discrete cosine transform (DCT), discrete Walsh transform (DWT), and the wavelet transform. It should be noticed that the transform itself does not reduce the amount of data; it is the subsequent coding or quantization that does the compression job.

The signals in the real world usually demonstrate some spatial or temporal correlation. In other words, the conditional entropy of the adjacent signal samples are much less than the entropy of one sample. Therefore, if we make use of the past history (either in temporal or spatial meaning) of the data to “predict” the current sample and encode only the prediction error, more efficient compression could be achieved. Usually, a linear model is used to make the prediction; in its simplest form, the past sample can be used as the prediction and the resulting compression scheme is called differential encoding. Like the transformation techniques, this technique is often used in conjunction with various quantization techniques to compress the data.

1.3 The Radar Signal

Radar is an acronym for “RADio Detecting And Ranging.” A radar system’s main purpose is the detection and location of a remote object by transmitting electromagnetic signals from a transmitter powerful enough so that the signals reflected back from the target are detectable by the receiver [8]. However, by doing so, a radar exposes itself to anyone who has a passive receiving device in a far longer range. To actually locate the radar emitter more effort is required and the coherent emitter location system that we described in the first section of this chapter is one method.

1.3.1 Review of the Radar Theory

The received signal power of a radar can be expressed as the following *radar equation* [8]:

$$S = \frac{P_T G^2 \lambda^2 \sigma}{(4\pi)^3 R^4} \quad (1.3.1)$$

where P_T is the transmitting power, λ is the wave length of the radar signal, σ is the scattering cross section of the target, G and R are the gain of the antenna and the range respectively, if similar antennas at the same range are used for both transmitting and receiving. A matched-filter is usually used at the receiver side to detect the receiving signal because it is the optimum linear processor if the receiving signal is only corrupted by AWGN. In the frequency domain, the matched-filter transfer function $H(\omega)$ should be the complex conjugate of the spectrum of the transmitting signal:

$$H(\omega) = cS^*(\omega)e^{-j\omega t_a} \quad (1.3.2)$$

(1.3.1) reveals the fact that the received signal strength is inversely-proportional to R^4 . However, if the target has a receiver tuned to the signal, the received signal strength is only inversely-proportional to R^2 . This fact means that a platform can detect the radar signal using a less sensitive receiver at a further range without the risk of being caught. However, because of the lack of matched-filter at the platform, some other techniques are needed to locate the radar emitter.

1.3.2 The Linear-FM Radar Pulse Train

The signal that a radar transmits can be either a train of pulses or continuous signal. In this thesis, we only consider a special case, the so-called linear-FM pulse train,

which is very commonly used. Nevertheless, the method discussed in this thesis should be applicable to any pulsed radar. In the linear-FM pulsed radar case, the radar signal is a train of pulses with a constant pulse repetition interval (PRI). Usually the PRI is

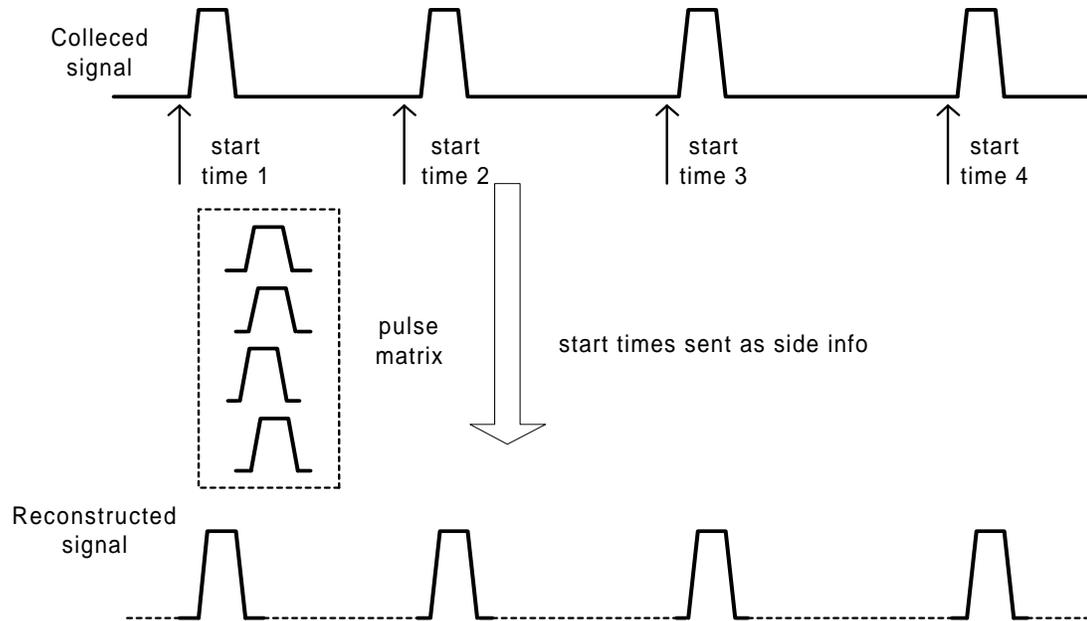


Figure 1.2: Pulse gating and thinning

quite large compared to the width of each pulse. The obvious pre-processing of this kind of signal is the so-called gating and thinning procedure [5] as shown in Fig. 1.2 which extracts the start time of each pulse and sends the pulses as a matrix with the start times sent as side info.

Each pulse in the pulse train is a short piece of linear chirp signal and all the pulses are very similar with the differences mainly in the magnitudes, starting phases, and starting times. Fig. 1.3 shows a typical pulse converted into base band. Its spectrum has been centered at approximately DC and the signal has been sampled at a rate close to the *Nyquist* rate. Usually, it is complex-valued and has a smoothly changing

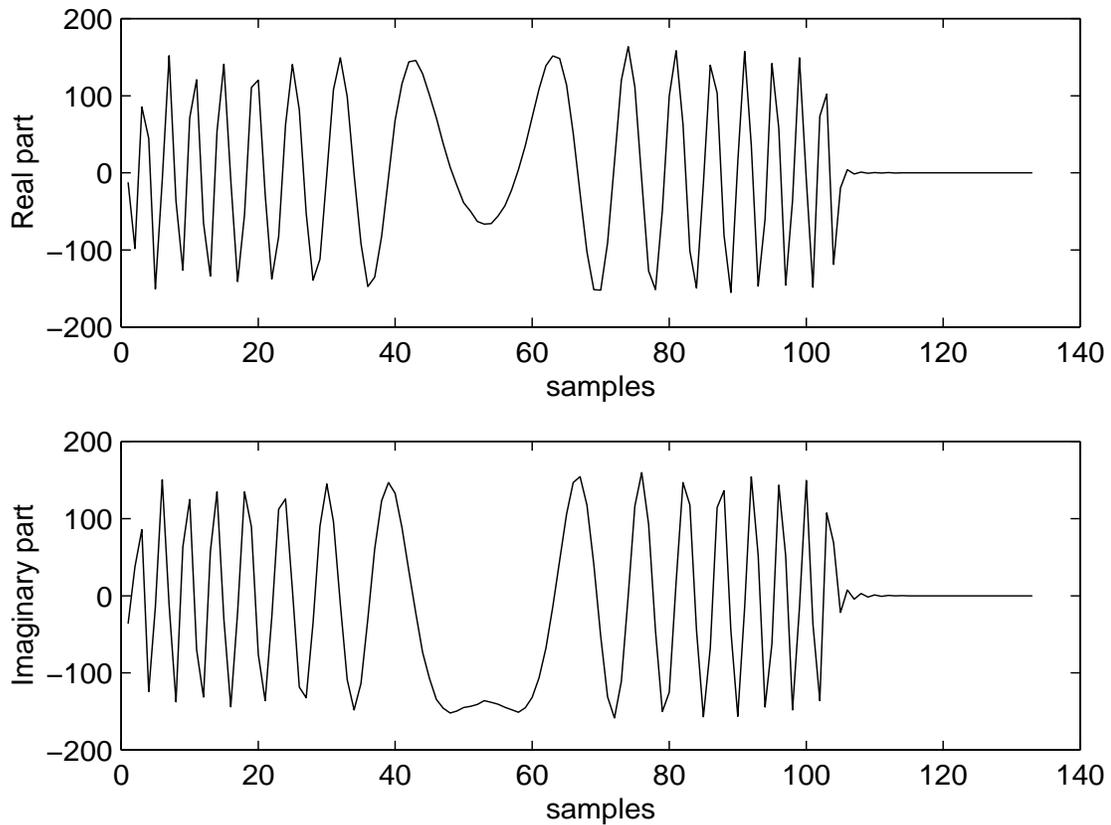


Figure 1.3: A pulse from the pulse train

envelope and its bandwidth is quite wide.

1.3.3 Insights into Compressing the Data

Actually the radar is constantly sending out the same radar pulses. However, because of, but not limited to, channel fading, jitters at both the transmitter and receiver sides and noises, they seem quite different at the receiver side. The differences mainly lie in the magnitudes, starting phases, and starting times. If we can remove those differences by normalizing the magnitudes, aligning starting times and phases, the pulses will return to the similar form as illustrated in Fig. 1.4 which shows 10 pulses

drawn on top of each other. In other words, we want to “undo” the major parts of the disturbances that the channel imposes to the signals. After that, we can use one prototype pulse to represent those similar pulses, and only send this prototype pulse along with side information to “redo” the disturbances. We can reconstruct the signals base on the prototype pulse and the side information fairly well. When the number of pulses and the number of samples within one pulse are larger than $10 \sim 20$, which is usually the case, our goal of compression can be achieved. In chapter 2 we

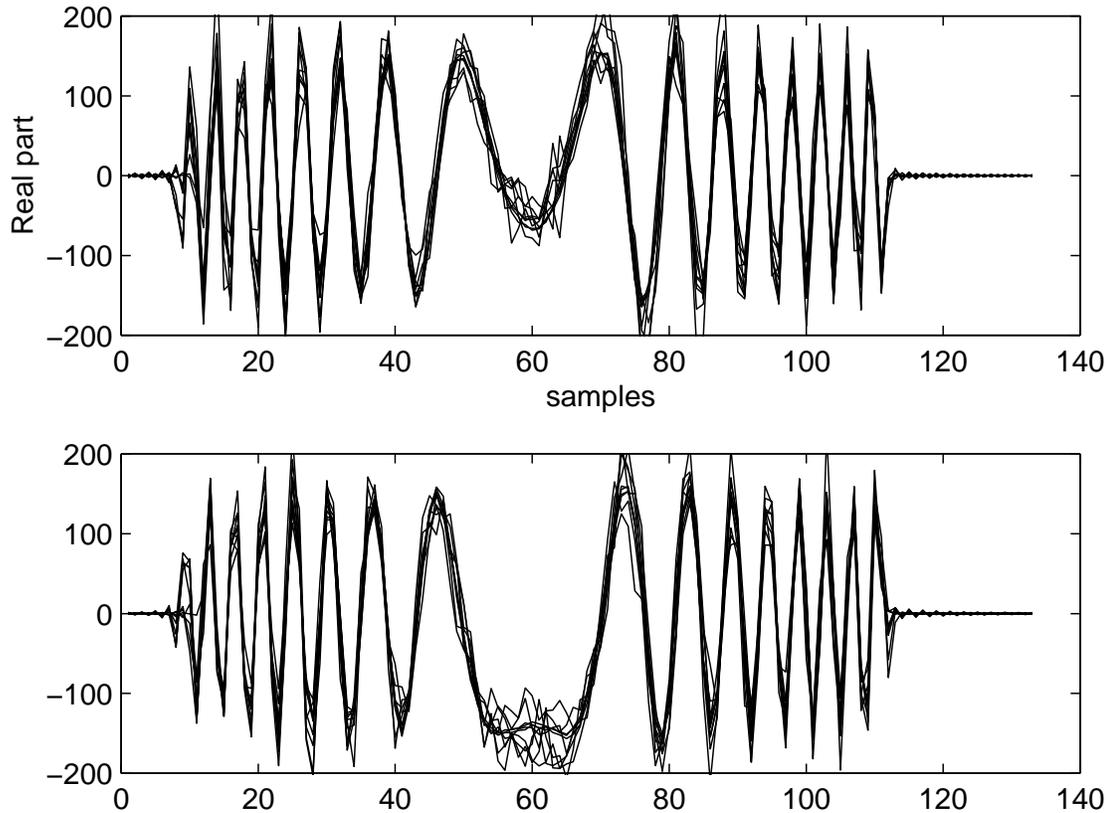


Figure 1.4: 10 pulses after processing

will discuss several ways to shift the pulses in a fraction of the sampling interval which is very important in the time alignment task. In chapter 3 we will discuss how to

extract the prototype pulse and the magnitude and phase parameters of each pulse.

There are some other facts that are worth to be noted. The possible ranges of the FDOA/TDOA are very limited in a typical emitter location system:

- The FDOA reflects the relative velocity of the two platforms to the emitter. In practice, it is normally six orders of magnitude less than the carrier frequency of that emitter, and 2 ~ 3 orders of magnitude less than the sampling rate of the receiver.
- The TDOA reflects the relative displacement of the two platforms to the emitter. In practice, it is normally 2 ~ 3 orders of magnitude less than the PRI.

Those facts are to be exploited in the non-coherent method discussed at the end of chapter 3.

Chapter 2

The Fractional Delay Filters

Delaying a digital signal is easy as long as the desired delay is a multiple of the sampling interval. However, when a delay of a fraction of the sampling interval is needed, as in our radar signal compression problem and various other applications [9, 10, 11], more sophisticated methods must be used. This problem can be viewed as either:

- bandlimited interpolation between samples,
- or a linear time-invariant system with a transfer function roughly equal to z^{-D} ,
 $D \in \mathbb{R}$.

In this chapter, several well-established approaches for fractional delay (FD) filters are discussed, and a new approach is proposed. A comparison is given at the end.

2.1 Frequency Domain Methods

Without loss of generality, we can assume that $D \in (0, 1)$. By substituting z with $e^{-j\omega}$ the ideal frequency response (Fourier transform) of the delaying system is obtained:

$$H_{id}(e^{j\omega}) = e^{-j\omega D} \quad \omega \in [-\pi, \pi] \quad (2.1.1)$$

where $\omega = 2\pi fT$ is the normalized angular frequency, and T is the sample interval.

Thus the desired magnitude and the desired phase response are:

$$|H_{id}(e^{j\omega})| \equiv 1 \quad \omega \in [-\pi, \pi] \quad (2.1.2)$$

$$\arg[H_{id}(e^{j\omega})] = \Theta_{id}(\omega) = -D\omega \quad (2.1.3)$$

2.1.1 The Brute Force Fourier Method

The most intuitive way of doing a fractional delay is to do it in the frequency domain with the following steps:

1. Do an FFT over the signal (proper zero-padding may be required);
2. Multiply the FFT result by $e^{-j\omega D}$;
3. Finally, do an inverse FFT to convert it back to time domain.

We may call it FFD (Fourier Fractional Delay) because it strictly follows (2.1.1). Although it does the job without introducing noise, this method is only valid when the signal is very short, and input-to-output delay is not a concern because it requires gathering all the data before processing. However, a more useful method can be derived from it and will be discussed below.

2.1.2 The Block Fourier Method

The brute force FFD has a fatal pitfall because very long FFT is not affordable. To relieve this barrier, the block Fourier method comes to the rescue. The ideas behind this method are:

1. breaking the signal into overlapping blocks;
2. processing the blocks individually using the FFD method;
3. catenating the results.

Fig. 2.1 illustrates the process of the BFFD (Block Fourier Fractional Delay). A fixed FFT size is used to make it suitable for online processing of very long signals and the maximum processing delay equals the FFT block size minus half of the overlap length.

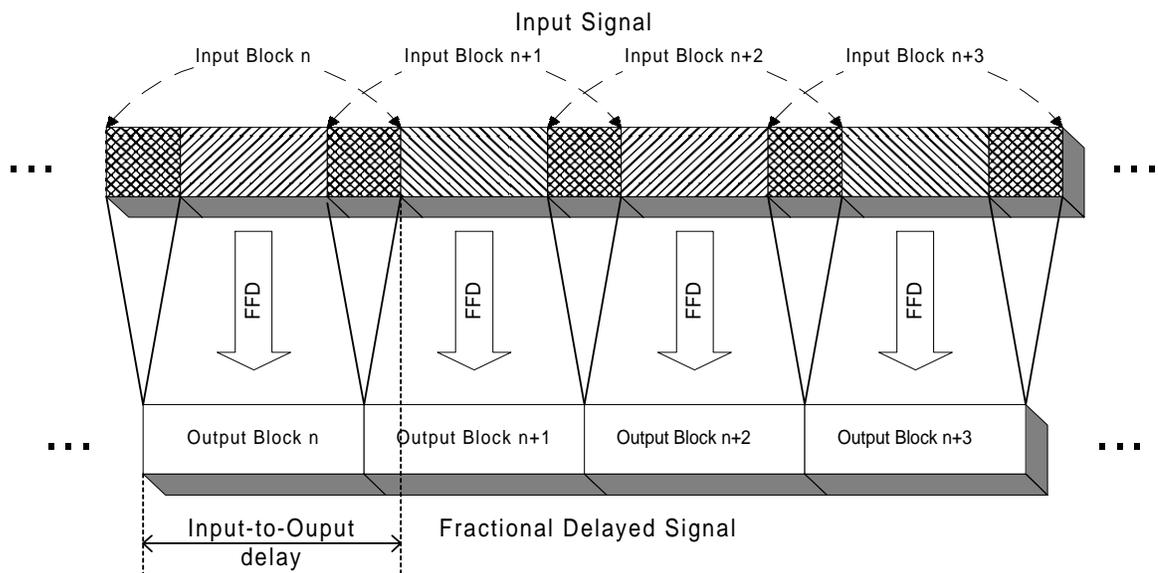


Figure 2.1: The diagram of the BFFD

However, since we truncate the signal prior to doing the FFT, some level of distortion is introduced. Fig. 2.2 demonstrates the distortion while using BFFD to process a sinusoidal signal. A fixed FFT block size (64) is used, while different overlapping length are compared. A blocky pattern of distortion is observed:

- The distortion has a “U” shape pattern; the closer a sample to the border of the processing blocks, the higher the resulting noise.
- Overlapping the processing blocks helps to reduce the blocky effect, the more the overlapping, the better the result.

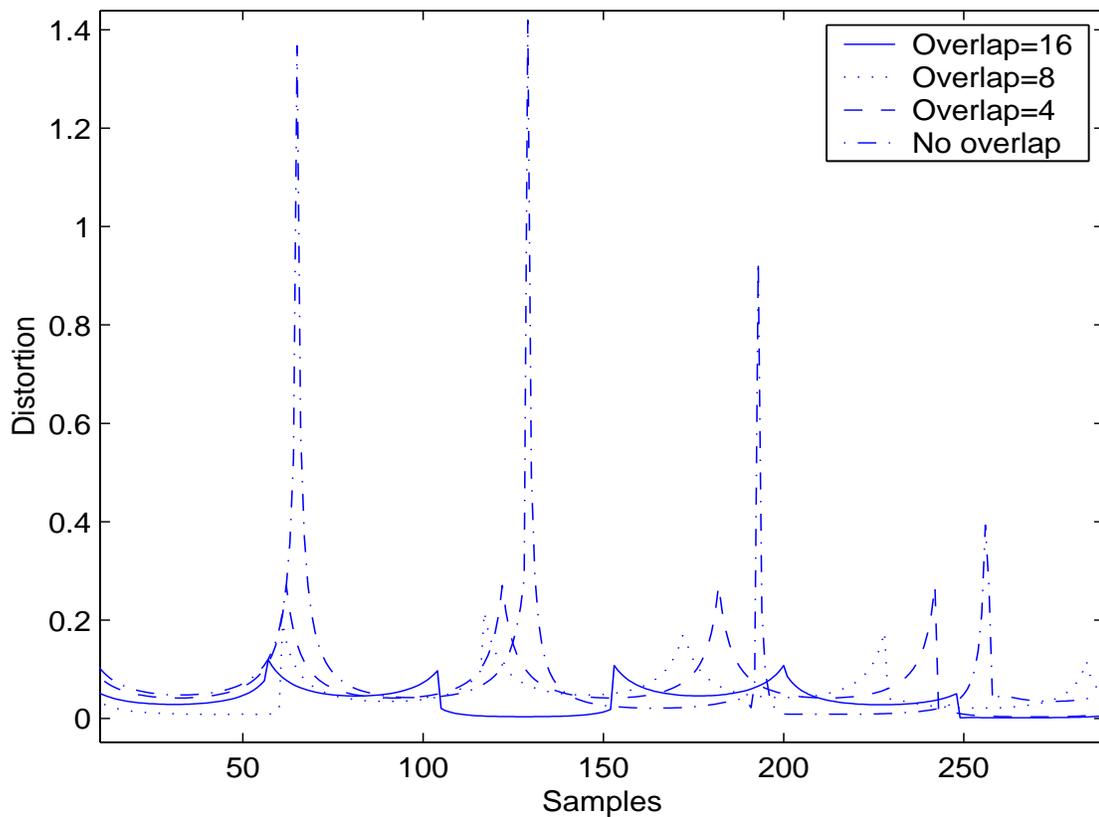


Figure 2.2: Blocky Effect of the BFFD

However, increasing the overlapping has a side-effect to reduce the efficiency of the algorithm. Also, the accuracy gain of using an overlapping length larger than 1/4 of the block size is marginal.

Using a sinusoidal input may not be a very good idea to evaluate the real-world performance of a fractional delay algorithm. In Fig. 2.3, a band-limited white noise sampled at a rate close to the *Nyquist* rate is used as the input signal. BFFDs of different sets of parameters are compared across the fractional delays. It can be noticed that the worst case happens when half sample delay is used.

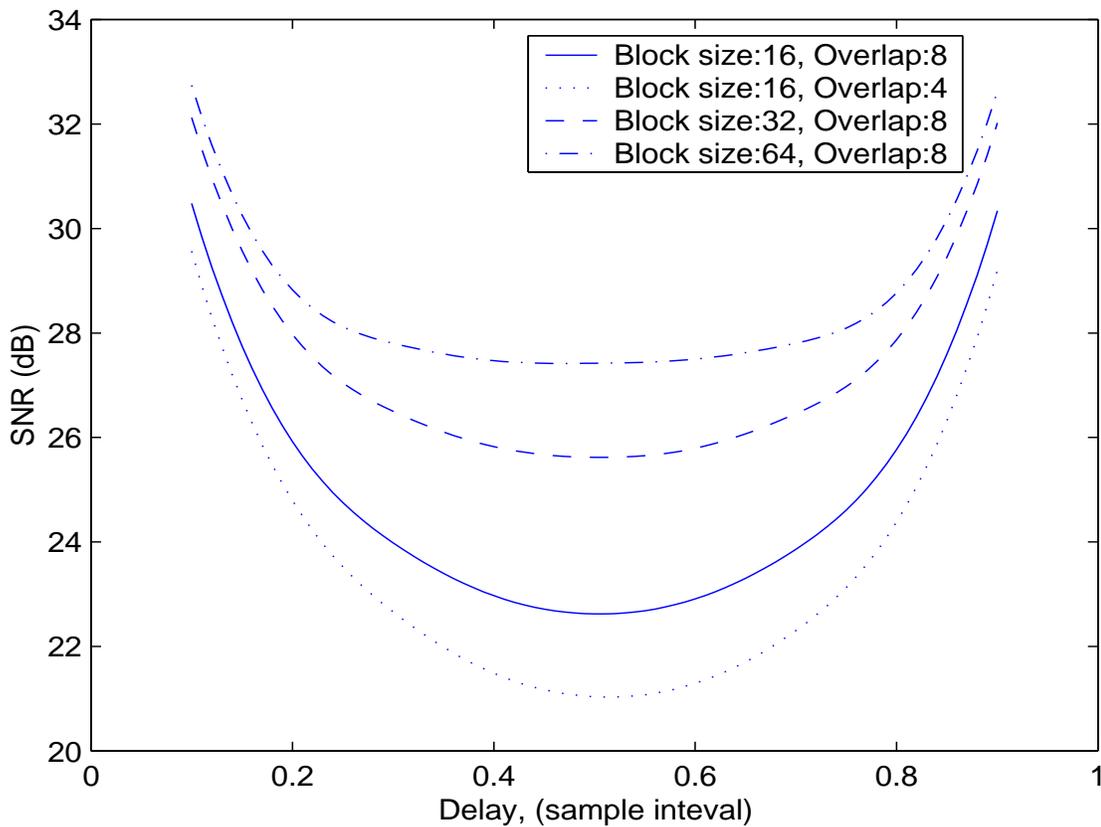


Figure 2.3: The BFFD under band-limited white noise input

2.2 The Filtering Methods

The implementation of a constant delay can be considered as a time domain approximation of the ideal fractional delay filter described in (2.1.1) which has unity magnitude response and a constant group delay of the given delay D . By doing an inverse DTFT over (2.1.1) we get:

$$\begin{aligned}
 h_{id}(n) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H_{id}(e^{j\omega}) e^{j\omega n} d\omega \\
 &= \frac{\sin[\pi(n - D)]}{\pi(n - D)} = \text{sinc}(n - D) \quad \forall n \in \mathbb{I}
 \end{aligned}
 \tag{2.2.1}$$

When D is not an integer value, the impulse response (2.2.1) is a shifted and sampled version of the sinc function, which is infinitely long on both sides. Therefore, a finite-order causal FIR or IIR filter can only approximate the impulse response of the ideal filter.

2.2.1 The Lagrange FIR Filter

FIR filters have been widely used because they are simple, stable, and have short processing delay. To design a FIR filter to approximate ideal fractional delay filter,

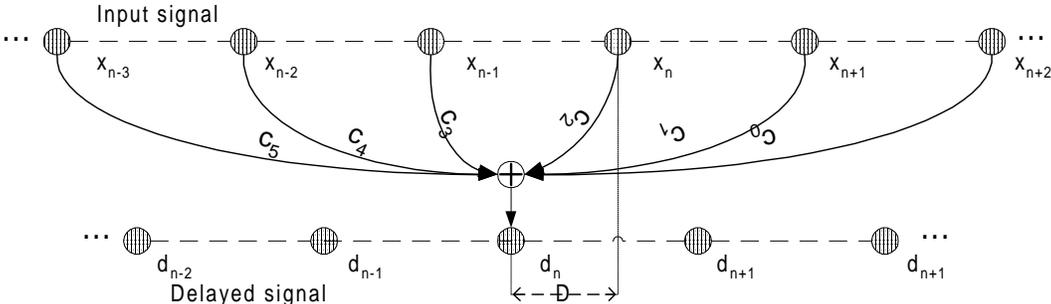


Figure 2.4: Weighted sum of N samples

several methods have been used, such as: least square [12], window-based design [9],

maximally flat design [13] and minimax design [14]. In [12], it was shown that the smallest error for a given filter order N is obtained when the integer part of D equals $N/2$, i.e., each fractionally delayed sample is computed by a weighted sum of $N/2$ samples before this time and $N/2$ samples after this time. Fig. 2.4 shows a 6-tap FIR FD filter.

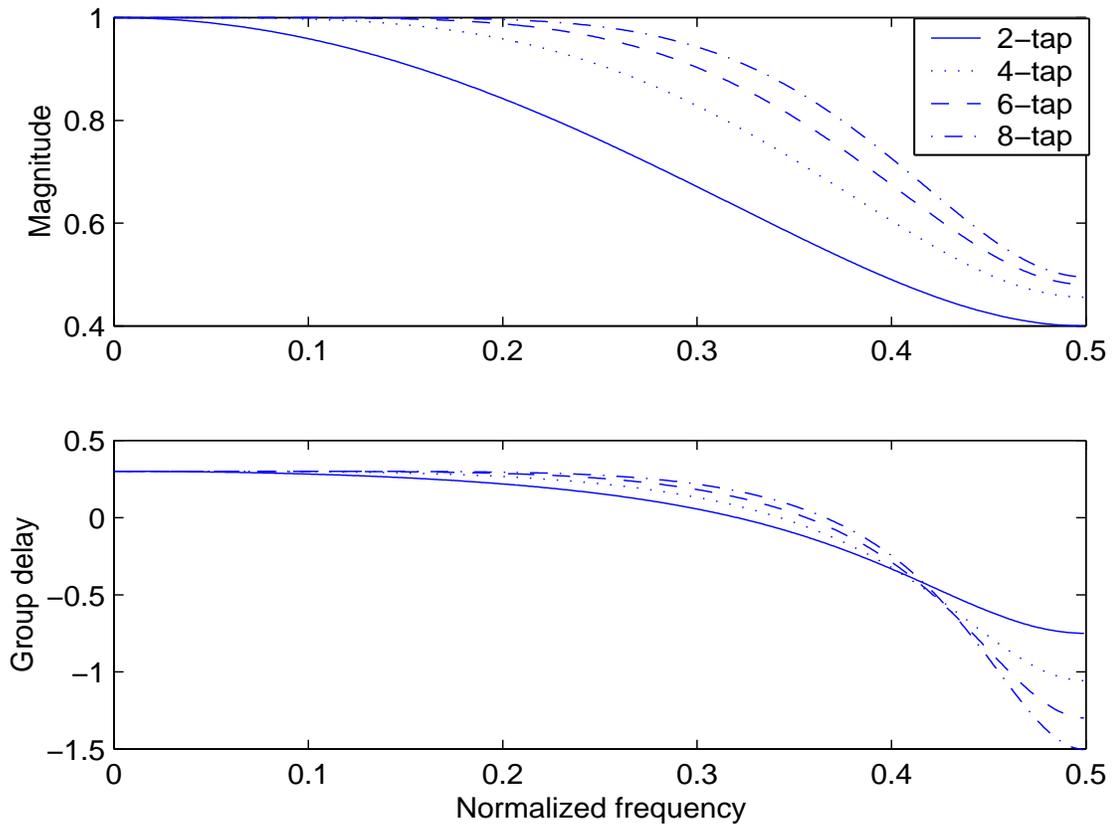


Figure 2.5: Lagrange FIR FD filters responses

Lagrange interpolation is probably the easiest way to design a FIR FD filter. The coefficients are [15]:

$$c(n) = \prod_{\substack{k=0 \\ k \neq n}}^{N-1} \frac{D-k}{n-k} \quad n = 0, 1, \dots, N-1 \quad (2.2.2)$$

Fig. 2.5 shows the magnitude and group delay of several Lagrange FIR FD filters of different length. The expected delay is set to 0.3. The Lagrange filter is a maximally flat filter at $\omega = 0$ [13]. Therefore, it has very good response at low frequencies, and a smooth magnitude response. However, those advantages come at the cost of the performance at higher frequencies. Increasing the filter order will help, but the approximation bandwidth grows very slowly when the order is above 4 or 6. It is also possible to develop a maximally flat filter that favors an arbitrary frequency ω_0 [16], but performance at other frequencies will suffer.

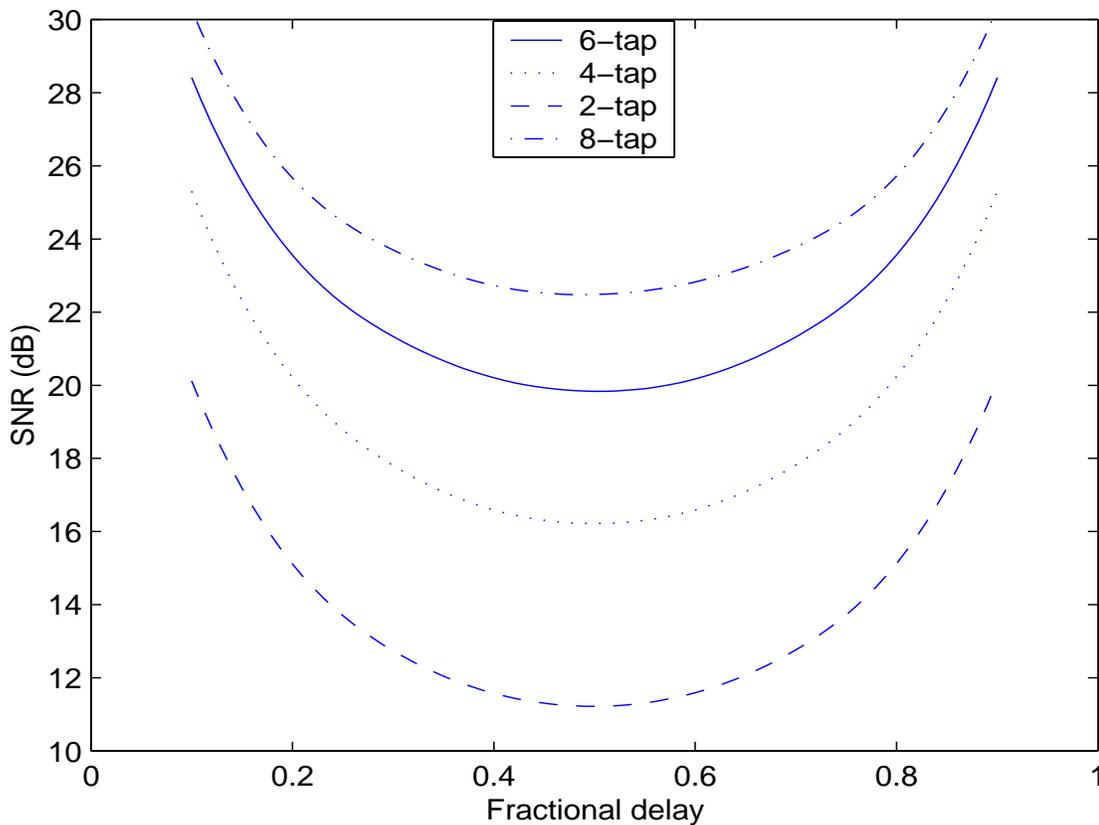


Figure 2.6: The Lagrange filter under band-limited white noise

Fig. 2.6 shows the performance of Lagrange FIR FD filters of different size with

a band-limited white noise as the input signal. It can be noticed that the worst case happens when half sample delay is used. Also, the performance is 9dB lower at a delay of 0.5 compared to when delay is 0.1 independent of the order of the filter.

2.2.2 The All-pass Thiran Filter

Generally, an IIR filter can meet the same frequency-domain specifications with lower order than a FIR filter. However, to design an IIR filter usually is much more complicated. The only solution known to us where the coefficients can be obtained in closed form is the all-pass Thiran filter [13]. Its coefficients are specified by (2.2.3):

$$\begin{aligned}
 A(z) &= \frac{a_{N-1} + a_{N-2}z^{-1} + \dots + a_1z^{-(N-2)} + z^{-(N-1)}}{1 + a_1z^{-1} + \dots + a_{N-2}z^{-(N-2)} + a_{N-1}z^{-(N-1)}} \\
 a_n &= (-1)^n \binom{N-1}{n} \prod_{k=1}^N \frac{D - N + k}{D - N + k + n} \quad n = 0, 1, \dots, N - 1
 \end{aligned}
 \tag{2.2.3}$$

Since it is an all-pass filter, its magnitude response is perfect. Its group-delay response

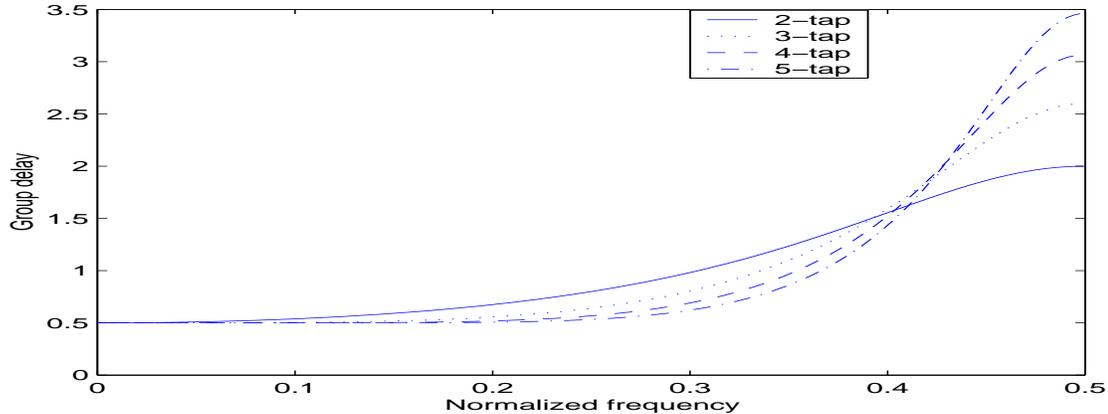


Figure 2.7: The Thiran filters’ group-delay responses

is shown in Fig. 2.7. The expected delay is 0.5. The Thiran method can be viewed as a recursive counterpart of Lagrange interpolation. And it also produces maximally

flat filters at $\omega = 0$ [13]. Although it tends to perform better than a Lagrange FIR FD filter of similar complexity, it inherited the same drawback: small approximation bandwidth except here the limit is only due to the group delay not the magnitude.

Fig. 2.8 shows the performance of All-pass Thiran filters of different size with a band-limited white noise as the input signal. A 6-tap Lagrange filter is also shown as a reference. Keep in mind that an all-pass filter requires about twice as much

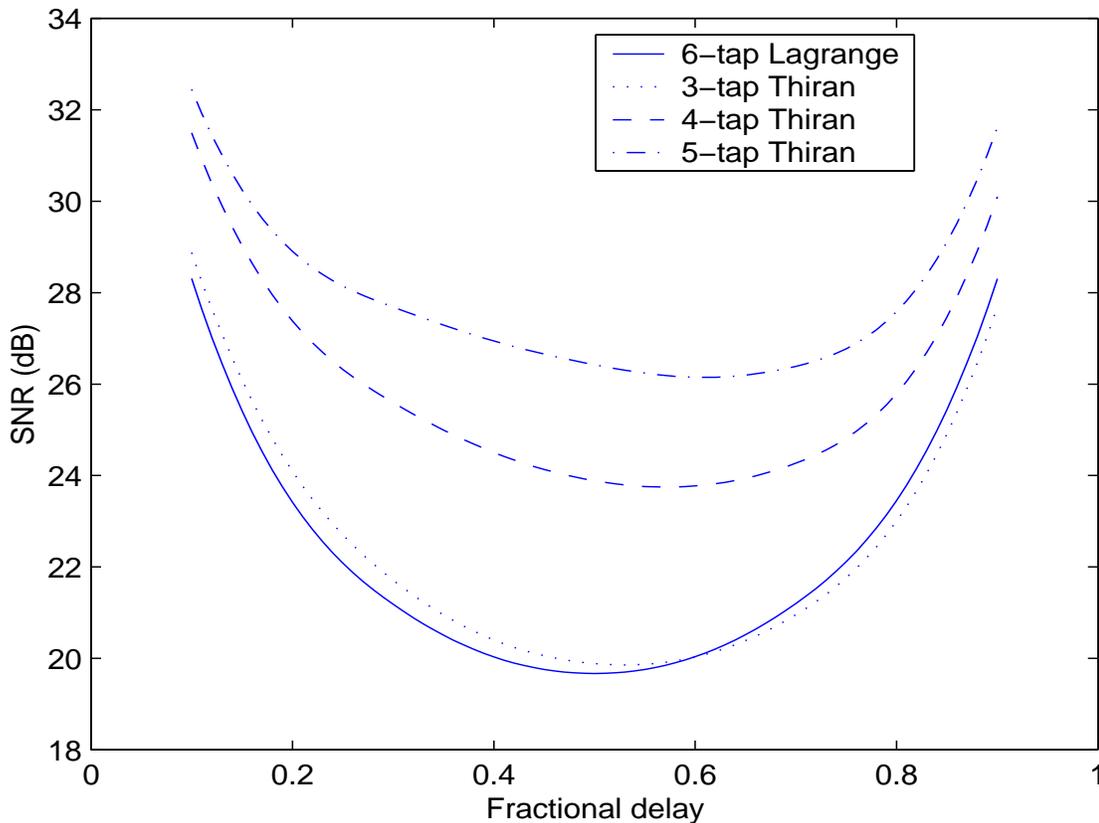


Figure 2.8: The Thiran filter under band-limited white noise

computation power as a FIR filter of the same size. For the Thiran all-pass filters, the worst case no longer happens at half sample delay. However, the worst case is still quite bad.

2.3 The Adaptive FIR Filter

The Lagrange and Thiran filters are very simple to implement because their coefficients can be obtained in closed form. However, they both have two serious problems:

- They are both frequency specific. For signals out of their approximation bandwidth, they do not perform well.
- Their accuracies are both sensitive to the actual fractional delay. In the worst case, the accuracies drop sharply.

The minimax design can address those problems. However, the fractional delay approximation is quite hard since it requires approximating a complex-valued function. Furthermore, the signal may not be stationary at all; then solely relying on approximating the impulse response of the ideal fractional delay filter would not be appropriate.

The adaptive filter shown in Fig. 2.9 is a filter that can evolve its coefficients according to the changing of the input signal. Since the coefficients are not constant,

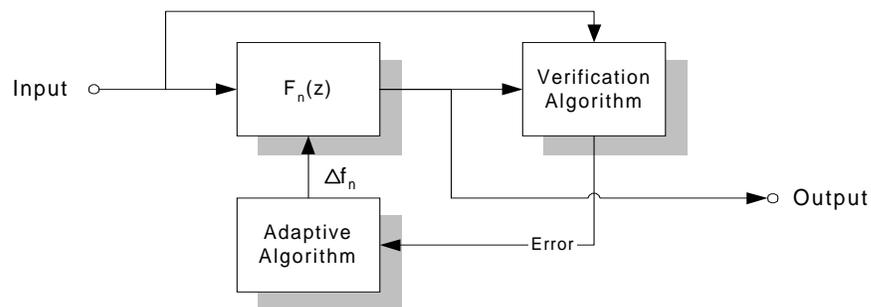


Figure 2.9: The adaptive filter

the adaptive filter is not an LTI system. It has the following properties [17] that are

highly favored in a fractional delay filter design:

- In a stationary environment, the coefficients will converge to steady state values close to the optimal coefficients.
- It does not rely on any *a priori* information of the signal statistics.
- For nonstationary signals, the filter is able to adapt to the changing statistics.

2.3.1 The LMS FIR Adaptive FD Filter

In this section, we begin our study by looking at the simplest form of adaptive filters.

A shift-varying version of the filter shown in Fig. 2.4 is used:

$$\hat{d}_n = \sum_{k=-l}^{k=l-1} w_{k,n} \cdot x_{n+k} \quad (2.3.1)$$

The validity of the coefficients can be verified by a linear model, and an error vector can be computed as:

$$\vec{e}_n = \vec{y}_n - \vec{w}_n * \vec{H}_n \quad (2.3.2)$$

\vec{w}_n should be updated in a fashion that the mean square error $\|\vec{e}_n\|^2$ should be minimized. The definition of \vec{y}_n and \vec{H}_n are application specific. We could use the steepest-descent method, the update equation is:

$$\begin{aligned} \vec{w}_{n+1} &= \vec{w}_n - \mu \nabla \|\vec{e}_n\|^2 \\ &= \vec{w}_n + \mu \vec{e}_n * \vec{H}_n^H \end{aligned} \quad (2.3.3)$$

where the $\mu > 0$ is the step size. This algorithm is known as the *LMS algorithm* [17]. However, unlike the well known Wiener filtering problem, there is no obvious verification method to use.

The difficulty of validating the coefficients of a fractional delay FIR filter rests in the fact that the only information we have available is the signal itself. However, a FIR FD filter of delay D has a close relation with the FIR FD filter of delay $1 - D$, which is just a reversal of the coefficients sequence. If there is a perfect FIR FD filter of delay D , then we can restore the original signal x_n by applying the reverse filter upon the delayed output d_n . This fact gives us the insight of validating the forward filter by reversal filtering of the delay output as in Fig. 2.10. We can rewrite (2.3.2)

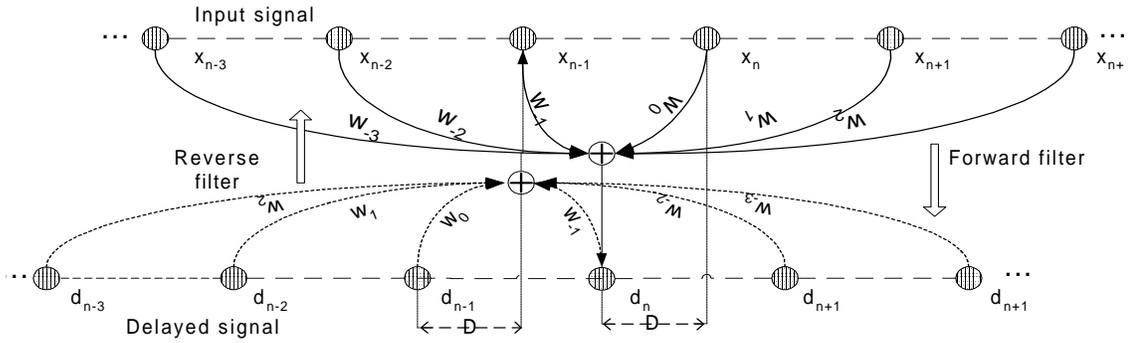


Figure 2.10: The forward-reverse filter

and (2.3.3) as:

$$e_n = x_{n-l} - \vec{w}_n * [d_{n-2l+1} \dots d_n]^T \quad (2.3.4)$$

$$\vec{w}_{n+1} = \vec{w}_n + \mu e_n * [d_{n-2l+1}^* \dots d_n^*] \quad (2.3.5)$$

which produce a $2l$ -tap adaptive FIR filter.

There are two parameters that need to be carefully selected: the initial coefficients \vec{w}_0 and the step size μ . Since this algorithm is based on the static FIR FD filter, using the Lagrange coefficients specified in (2.2.2) as a starting point is natural. The selection of μ is much more difficult: A too large μ can make the LMS algorithm unstable; while a too small μ will make the convergence too slow and limit the performance of

the adaptive algorithm. In [17] an upper bound of μ is given as:

$$\mu_{max} = \frac{2}{2l \cdot E\{|x(n)|^2\}} \quad (2.3.6)$$

However, our algorithm breaks the *independence assumption* because the verification data d_n and the weight vector w_n are not statistically independent, the actual selection of μ should be more conservative. Experiments show a $\mu = 0.1\mu_{max} \sim 0.2\mu_{max}$ is generally acceptable.

Fig. 2.11 shows a snapshot of an adaptive filter's (6-tap, delay is 0.6) response under band-limited white noise input. A 6-tap Lagrange filter is also shown for

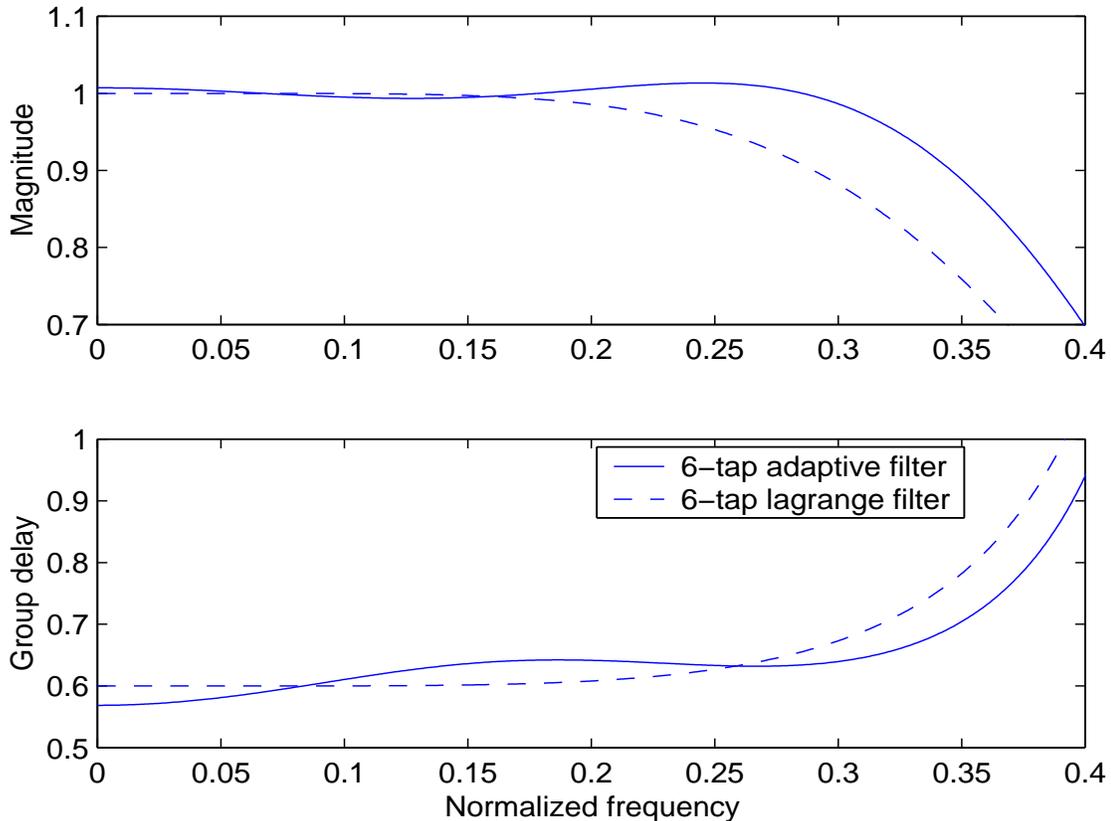


Figure 2.11: The snapshot of the adaptive filter's response

comparison. Note that the adaptive filter is also a low pass filter like the Lagrange

filter but is more like an equiripple filter and has broader approximation bandwidth.

Fig. 2.12 shows the adaptive nature of an adaptive filter's (6-tap, delay is 0.5). The input signal is a sinusoid with unity magnitude and a frequency of 1/4 of the sampling rate. A 6-tap Lagrange filter is also shown for comparison. While the static

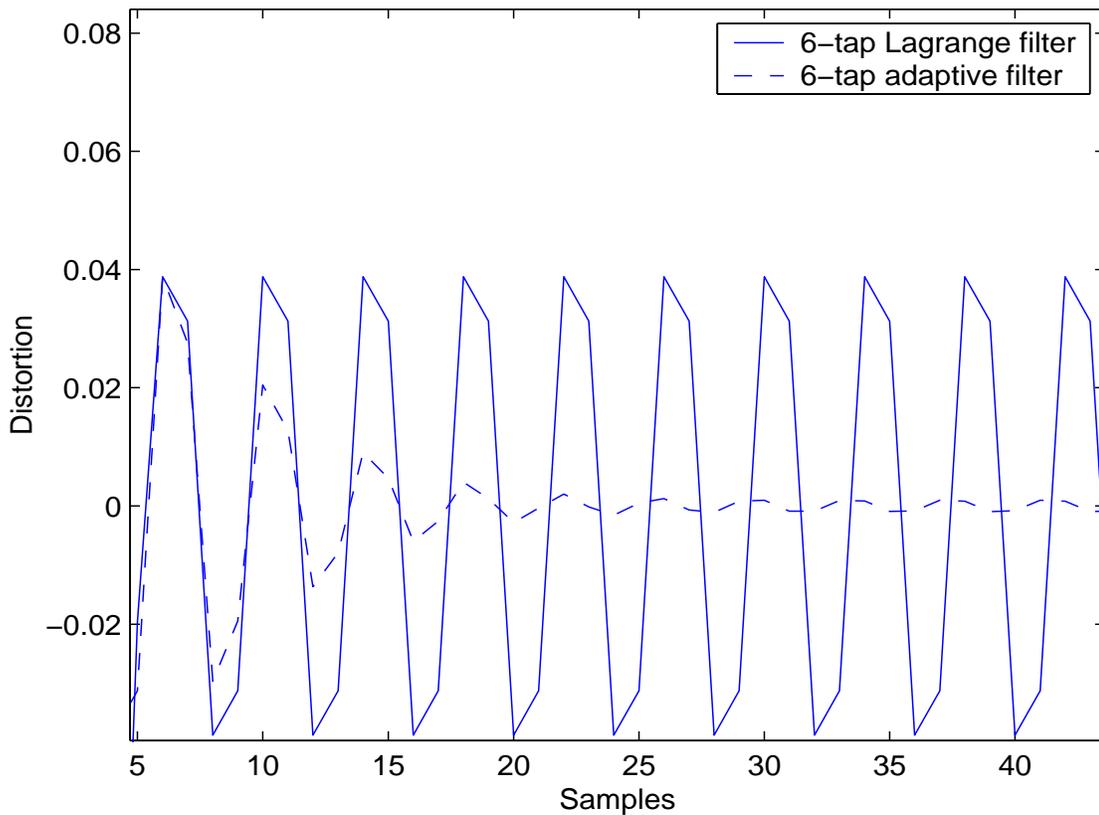


Figure 2.12: The adaptive filter under sinusoid input

filter yields a constant level of distortion, the adaptive filter's distortion drops quickly.

Fig. 2.13 compares an adaptive filter with the Lagrange filter of the same size in band-limited white noise input. It is obvious that the adaptive filter outperforms the Lagrange filter over the entire range of delays and the worst case performance of the adaptive filter is much better than of the Lagrange filter.

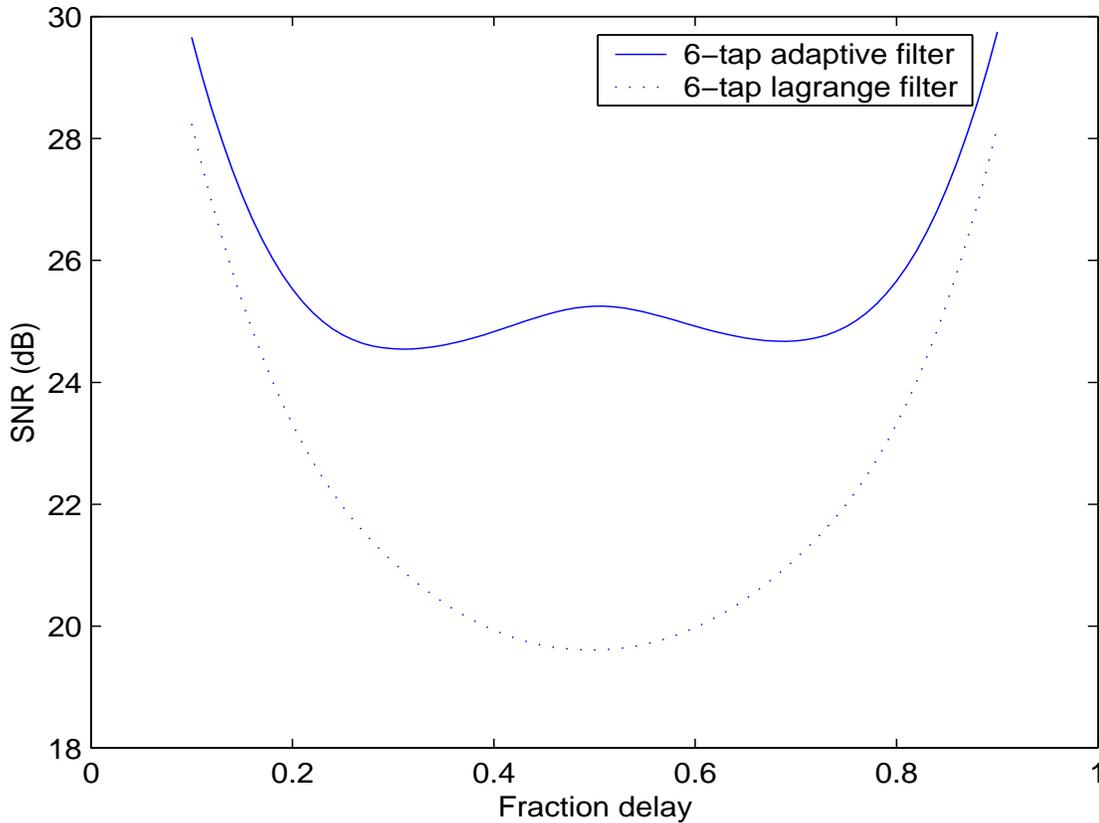


Figure 2.13: The adaptive filter under band-limited white noise

2.4 Comparisons

Having investigated several different methods to do the fractional delay, we can put together a summary in Table. 2.9. Four methods (the Block FFD, the Lagrange filter, the Thiran filter and the LMS filter) are compared on accuracy, latency, and computational complexity. Although the LMS filter seems to be superior, choosing the right method depends on the actual application. In our radar signal compression task, since the processing is done in a batch mode and it is unlikely to have more than a few hundred samples in each pulse, the best method may simply be the brute force FFD. All these methods can find their applications in different situations.

<i>Method</i>	Block FFD	Lagrange	Thiran	LMS
<i>Accuracy</i>	High	Low	Medium	High
<i>Latency</i>	High	Low	Low	Low
<i>Complexity</i>	High	Low	Low	Medium
<i>Other pros</i>	<ul style="list-style-type: none"> • Best for short signals 	<ul style="list-style-type: none"> • Maximally flat at zero frequency • Low-pass filter 	<ul style="list-style-type: none"> • Maximally flat at zero frequency • Perfect magnitude response 	<ul style="list-style-type: none"> • Not frequency specific • Not sensitive to the fractional delay • More suitable for non-stationary input
<i>Other cons</i>	<ul style="list-style-type: none"> • Blocky effect 	<ul style="list-style-type: none"> • Frequency specific • Bad worst case 	<ul style="list-style-type: none"> • Frequency specific • Bad worst case 	<ul style="list-style-type: none"> • Bad selection of step size can lead to either unstable or ineffective design

Table 2.1: Comparisons of four fractional delay methods

Chapter 3

The SVD Method

The Singular Value Decomposition (SVD) is a powerful tool in linear algebra. In particular, we are interested in its reduced-rank approximation property [6] because of the compression potential of this model. In this chapter, we first give a brief review of the SVD theory focusing on subspace estimation and noise suppression. Then we investigate the application of the SVD method in the radar signal scenario, the data compression property and its impact on the performance of the cross ambiguity processing. At the end, a non-coherent method specially developed to take advantage of the SVD method is proposed and studied.

3.1 Review of the SVD Theory

The singular value decomposition (SVD) of data-generated matrices plays an increasingly important role in contemporary signal processing applications. For a generally complex valued $M \times N$ matrix \mathbf{X} of rank p , its associated SVD representation takes

the following form:

$$\begin{aligned}\mathbf{X} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \\ &= \sum_{k=1}^p \sigma_k \vec{u}_k \vec{v}_k^H\end{aligned}\tag{3.1.1}$$

where \mathbf{U} is an orthogonal matrix consisting M *left singular vectors* (\vec{u}_k) as its columns, \mathbf{V} is a orthogonal matrix consisting of N *right singular vectors* (\vec{v}_k) as its columns, and $\mathbf{\Sigma}$ is a pseudo-diagonal $M \times N$ matrix whose main diagonal consists of the positive *singular values* (σ_k) and all other elements are zeros. Normally, the singular values are arranged in a non-decreasing manner, ie., $\sigma_k \geq \sigma_{k+1}$. The SVD is widely used in various signal processing applications because it can:

- approximate a matrix by a low-rank matrix [18];
- split a space into dominant and subdominant subspaces [19];
- aid the computation of the pseudo-inverse, the Grammian, and the projection operator [6].

3.1.1 Matrix Manipulation by SVD

In various signal processing applications such as complex exponentials identification [?] and ARMA impulse responses we may encounter an $M \times N$ signal matrix \mathbf{X} which is rank deficient, ie., $\exists \vec{a} \neq \vec{0}, \mathbf{X}\vec{a} = \mathbf{0}$. However, the signal matrix is often replaced by a noisy measurement matrix $\mathbf{Y} = \mathbf{X} + \mathbf{N}$, the resulting measurement matrix will be full-rank with probability one for any reasonable probability distribution of \mathbf{N} . An approximation ($\tilde{\mathbf{Y}}_r$) of an arbitrary rank $r < \min(M, N)$ can be produced by setting all but the r largest singular values of \mathbf{Y} to 0.

$$\tilde{\mathbf{Y}}_r = \sum_{k=1}^r \sigma_k \vec{u}_k \vec{v}_k^H\tag{3.1.2}$$

Notice the only difference between (3.1.2) and (3.1.1) is the usage of the predetermined r over the actual rank p . The r^{th} approximation is the rank r matrix that best approximates \mathbf{Y} in the Frobenius (least square) sense [20]. In [21] it has also been pointed out that the reduced rank matrix $\tilde{\mathbf{Y}}_r$ approximates the signal matrix \mathbf{X} better than the noisy measurement matrix \mathbf{Y} when the signal-to-noise ratio (SNR) is high. Low-rank approximations with specific patterns (eg. Toeplitz or Hankel) can be obtained by successive approximations using the same technique [19].

The SVD of a matrix \mathbf{X} can be used to split the signal space into subspaces. That is, in decomposition

$$\begin{aligned} \mathbf{X} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \\ &= \begin{pmatrix} \mathbf{U}_r & \mathbf{U}_{p-r} \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_r & 0 \\ 0 & \mathbf{\Sigma}_{p-r} \end{pmatrix} \begin{pmatrix} \mathbf{V}_r^H \\ \mathbf{V}_{p-r}^H \end{pmatrix} \end{aligned} \quad (3.1.3)$$

the orthogonal subspaces $\langle \mathbf{U}_r \rangle$ and $\langle \mathbf{U}_{p-r} \rangle$ can be used to produce the signal space $\langle \mathbf{X} \rangle$:

$$\begin{aligned} \langle \mathbf{X} \rangle &= \langle \mathbf{U}_r \rangle \oplus \langle \mathbf{U}_{p-r} \rangle \\ \langle \mathbf{U}_r \rangle &\perp \langle \mathbf{U}_{p-r} \rangle \end{aligned} \quad (3.1.4)$$

Similarly, the orthogonal subspaces $\langle \mathbf{V}_r \rangle$ and $\langle \mathbf{V}_{p-r} \rangle$ span the signal space $\langle \mathbf{X}^T \rangle$. The orthogonality of the subspaces is a consequence of the orthogonality principle for least squares solutions [6].

There are some useful miscellaneous equations when matrix \mathbf{X} is replaced by its SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$. In this case, the pseudo-inverse \mathbf{X}^\sharp , the Grammian $\mathbf{G}(\mathbf{X})$, and the projection $\mathbf{P}(\mathbf{X})$ can be orthogonally decomposed as follows:

$$\begin{aligned} \mathbf{X}^\sharp &= (\mathbf{X}^H \mathbf{X})^{-1} \mathbf{X}^H = \mathbf{V} \mathbf{\Sigma}^{-1} \mathbf{U}^H \\ \mathbf{G}(\mathbf{X}) &= \mathbf{X}^H \mathbf{X} = \mathbf{V} \mathbf{\Sigma}^2 \mathbf{V}^H \\ \mathbf{P}(\mathbf{X}) &= \mathbf{X} \mathbf{X}^\sharp = \mathbf{U} \mathbf{U}^H \end{aligned} \quad (3.1.5)$$

3.1.2 SVD Denoising

Let the signal vector \vec{x} be a vector of p elements, ie, $\vec{x} = [x_1, x_2, \dots, x_p]^T$. The signal subspace $\langle \mathbf{S} \rangle$ can be defined by the span of $[\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r]$ which are the r eigenvectors corresponding to the r non-zero eigenvalues of the auto-covariance matrix

$$\mathbf{R}_{xx} = E(\vec{x}\vec{x}^H) \quad (3.1.6)$$

However, neither the signal \vec{x} nor the auto-covariance matrix \mathbf{R}_{xx} is accessible. Instead, we only have the noisy measurement $\vec{y} = \vec{x} + \vec{n}$, where \vec{n} is an additive, white, Gaussian noise (AWGN) with an auto-covariance matrix of $\sigma^2\mathbf{I}$. Since the AWGN \vec{n} and the signal \vec{x} are statistically independent, the auto-covariance matrix of \vec{y} is:

$$\mathbf{R}_{yy} = E(\vec{y}\vec{y}^H) = \mathbf{R}_{xx} + \sigma^2\mathbf{I} \quad (3.1.7)$$

It is trivial to prove that \mathbf{R}_{yy} inherited all the eigenvectors of \mathbf{R}_{xx} while the eigenvalues are increased by σ^2 . Therefore, the signal subspace $\langle \mathbf{S} \rangle$ can be constructed by finding the eigenvectors corresponding to the largest eigenvalues of \mathbf{R}_{yy} if:

- the size of the signal subspace (the rank of \mathbf{R}_{xx}) is known, or,
- the signal-to-noise ratio is large so that there is a clear difference between the signal eigenvalues and σ^2 .

Once the signal subspace \mathbf{S} is known, a better measurement \vec{y}' can be constructed by simply projecting \vec{y} on to $\langle \mathbf{S} \rangle$:

$$\vec{y}' = \mathbf{P}(\mathbf{S}) \cdot \vec{y} \quad (3.1.8)$$

where $\mathbf{P}(\mathbf{S})$ is the projection of signal subspace \mathbf{S} as defined in (3.1.5). By doing so, the SNR is improved by p/r because the energy of the AWGN is spread equally over all the p eigenvalues and we throw away $p - r$ of them.

This kind of projection is equivalent to the reduced rank approximation we described in (3.1.2) when we do the SVD over the signal matrix \mathbf{Y} instead of doing the eigenvalue decomposition over the auto-covariance matrix \mathbf{R}_{yy} because \mathbf{R}_{yy} can only be estimated in most applications. In this way, the estimation of the signal subspace is likely to be imperfect because of the limited number of signal realizations. Nevertheless, an SNR gain, though less than the ideal situation (p/r), can still be achieved. Fig. 3.1 shows the noise suppression effect of the rank reduction approximation using

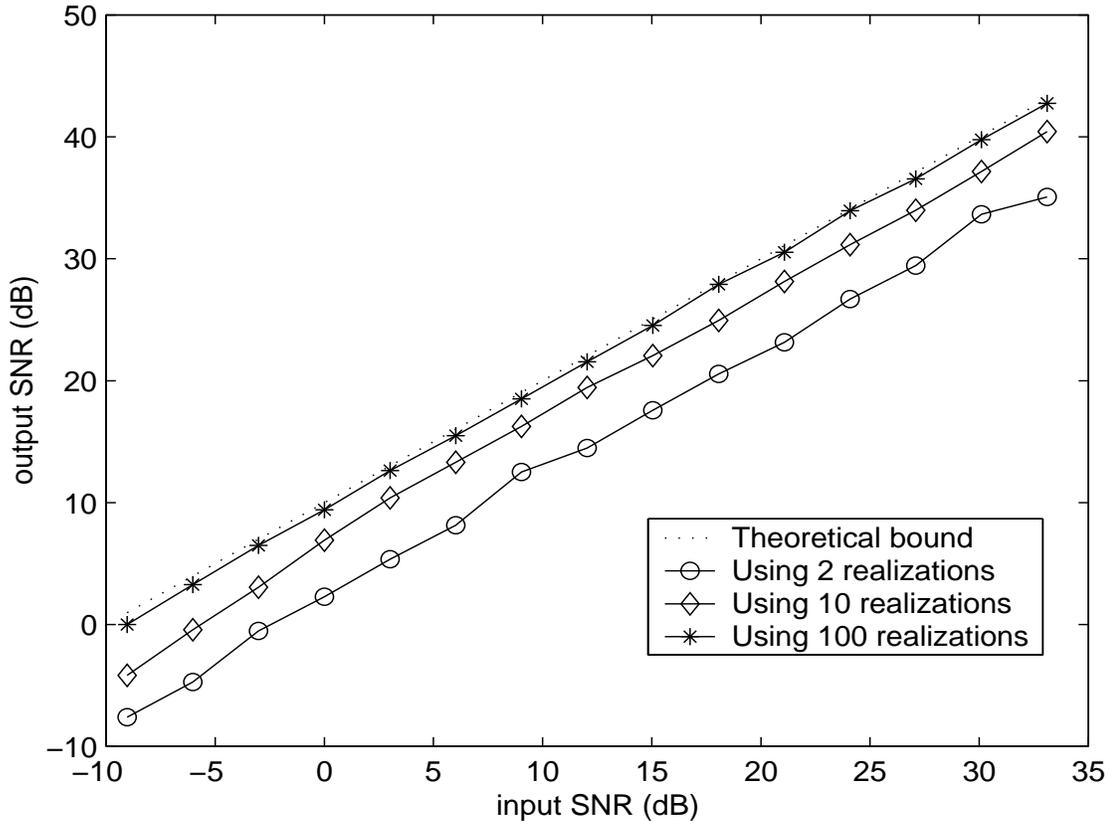


Figure 3.1: Denoising using SVD

SVD. In those cases, the signal subspace is rank one and has 10 elements. Therefore, a theoretical $10dB$ SNR gain could have been achieved if the signal subspace can

be perfectly identified. In reality, some levels of graceful degradation is observed if a limited number of the signal realizations (the cases of 2, 10, 100 realizations are shown) are used to identify the signal subspace.

3.2 Parameterizing the Pulses

In Chapter 1 we demonstrate that after proper thinning and gating, the radar signal can be organized into a matrix whose each row represents a single radar pulse. If we do not consider frequency hopping, those pulses are essentially the same except:

- Their starting time may be different;
- Their magnitude may be different;
- Their starting phase may be different.

Therefore, if we can extract a prototype radar pulse and a parameter vector (starting time d , magnitude A , starting phase θ) for each pulse, the original signal can be restored from those values. When the number of pulses is large, which is usually the case, very good compression ratio can be achieved.

3.2.1 Pulse Extraction using SVD

Since a typical radar is constantly sending out the same radar pulse, the signal subspace should be rank-one. However the signal matrix collected at the receiver side is usually quite far from rank-one because of, but not limited to, the following reasons [5]:

- Incommensurate PRI and sampling interval T ;

- Transmitter jitter;
- Imperfect pulse gating.

These effects can be handled through proper pre-processing to align the pulses in the signal matrix. The correct value of time shift can be found by cross-correlating the

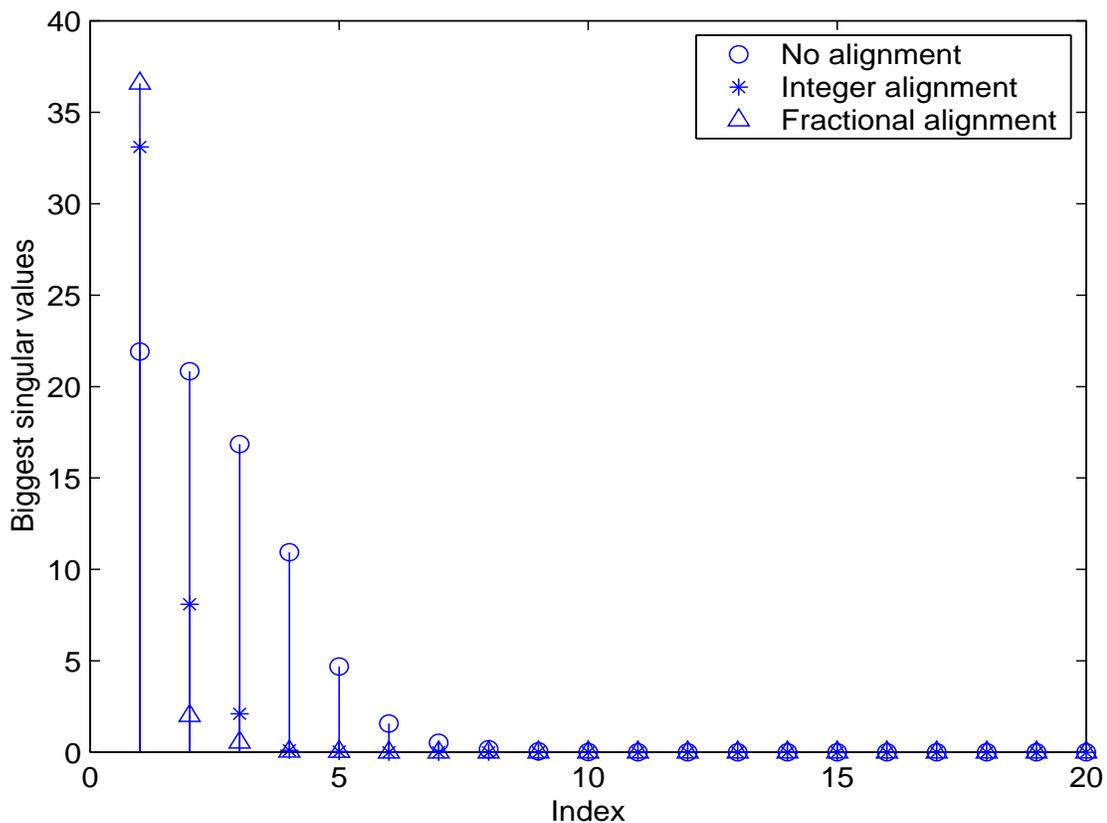


Figure 3.2: Singular values before and after alignment

pulses, interpolating the correlation function and finding the peak. In practice, the pulse that has the biggest magnitude is chosen as a reference to cross-correlate with others because it tends to have the biggest signal-to-noise ratio. Since the amount of time alignment needed is usually not an integer multiple of the sampling interval, one of the methods from the discussion of Chapter 2 can be chosen to do the fractional

delay. For simplicity's sake, the brute force Fourier method is used. Fig. 3.2 compares the 10 biggest singular value of a signal matrix before alignment and after alignment. Alignment not only makes the signal matrix much closer to rank-one but also extracts the starting time of each pulse, which is the first element in the pulse parameter vector we need for each pulse.

Once the alignment is done, we compute the reduced-rank approximation of the signal matrix described in (3.1.2). In particular, we only keep the one biggest singular value, so the equation turns out to be:

$$\tilde{\mathbf{X}} = \sigma_1 \vec{u}_1 \vec{v}_1^H \quad (3.2.1)$$

where σ_1 represents the square root of the total energy in this signal, the magnitude and phase of \vec{u}_1 represents the magnitude and starting phase of each pulse, and \vec{v}_1^H is the better choice of the prototype pulse than any other single pulse due to the noise suppression property of the SVD method that we discuss in 3.1.2. Since the total energy is meaningless in the subsequent ambiguity processing, we can safely discard the σ_1 .

How much compression can we get from this scheme? Suppose the original signal matrix is $m \times n$, ie., it has m pulses and n complex elements in each pulse, so the total matrix requires $2mn$ real values. After the compression, we have the 3 element parameter vector (d, A, θ) for each pulse, which requires $3m$ real values, and the prototype pulse which requires $2n$ real values. Therefore the compression ratio is:

$$CR = \frac{2mn}{3m + 2n} \quad (3.2.2)$$

In a typical setup, m and n range from several tens to a few hundreds, then the compression ratio CR is likely to be larger than 10.

3.2.2 Simulation Results

By keeping only a small amount of the parameters of the signal matrix instead of the whole matrix we have achieved a very good compression ratio. However, this is only part of our goals, which are high compression ratio *and* low degradation. Low degradation does not mean that we want to restore the original signal matrix as accurately as possible; instead, it means good FDOA/TDOA estimation accuracy in the emitter location system. Therefore, simulations are designed to measure the performance of our compression algorithm. The procedure of the simulations are:

1. Generate a set of radar pulses and a time-shifted and Doppler-shifted version of the same pulses to emulate the signals received on two different platforms;
2. Add AWGN to both signals; varying the noise level of one signal (signal B) while keeping the noise level of the other signal (signal A) fixed;
3. Compress and decompress signal B;
4. Compute the FDOA/TDOA pair of the original signal A and the decompressed signal B.

For simplicity's sake, the brute force Fourier method is used in every simulation. The simulation is repeated many times so that we can compute the variances of the FDOA/TDOA estimations. Then we plot the FDOA/TDOA variances versus the SNR of signal B. In the first simulation we use a noise-free signal A. A total of 100 *Monte-carlo* runs are used. The tested signal is a linear-FM pulse train consisting 50 pulses, each has 43 samples. The maximum frequency deviation of FM is set to the sampling rate to make the sampling closed to critical. The time-shift and Doppler-shift are 3.25 sampling intervals and 0.001 sampling rate, respectively. It can be

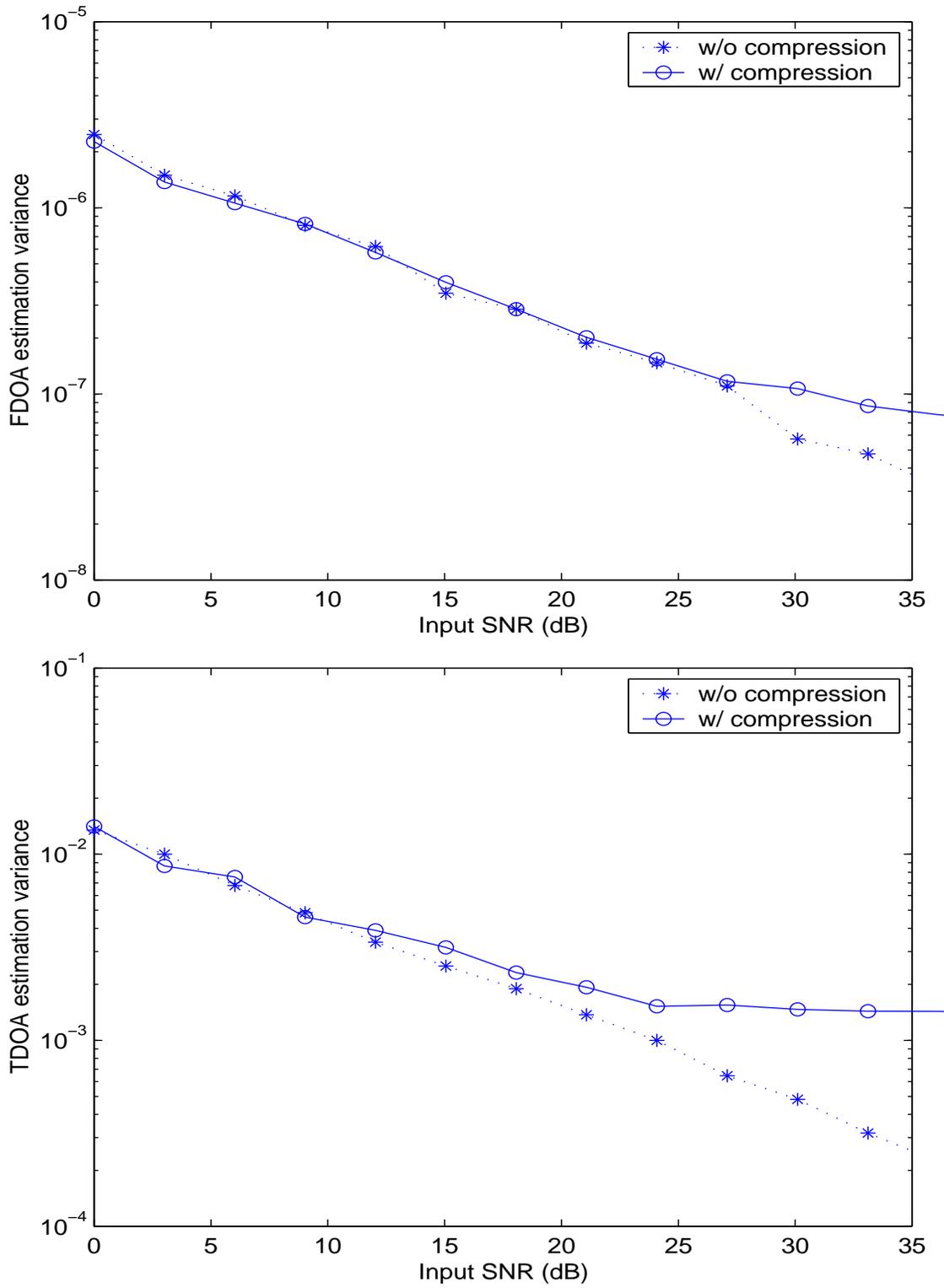


Figure 3.3: FDOA/TDOA accuracy (coherent, A noise free)

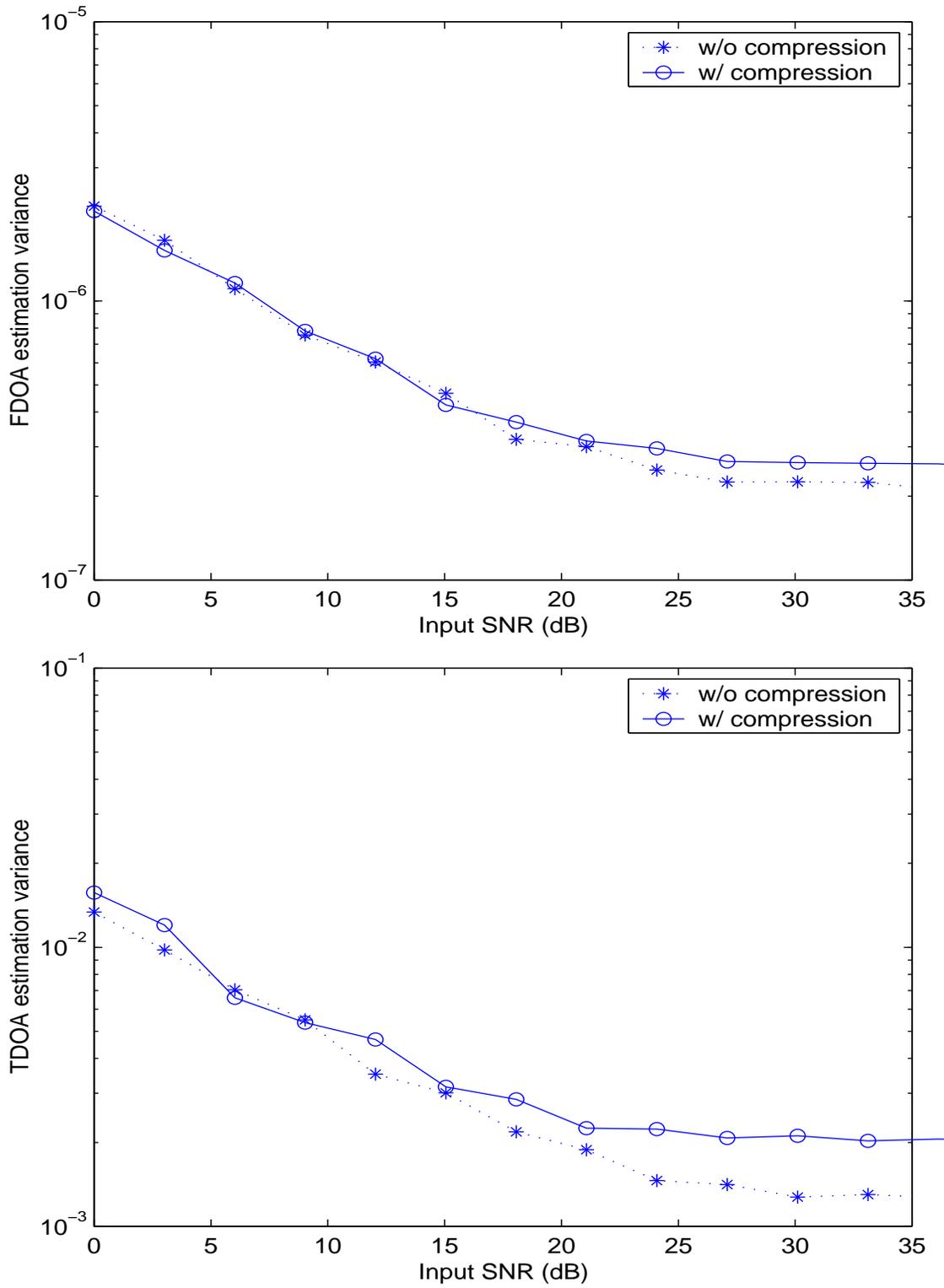


Figure 3.4: FDOA/TDOA accuracy (coherent, A 20dB)

noticed from Fig. 3.3 that:

- The FDOA/TDOA variances of the uncompressed data are inversely-proportional to the square root of the input SNR (a straight line in the figure with a slope of 20dB per decibel). This fact conforms to the *Cramer-Rao Bound* theory mentioned in (1.1.4).
- The performance of the compressed data and the uncompressed data are almost identical when the input SNR is below 25dB. after that, the performance of the compressed data begins to saturate. It makes sense because the compression we used inevitably introduce additional distortion to the signal B.

In reality, the SNR of the received radar signal can not be arbitrarily high. A more realistic simulation that we demonstrate in Fig. 3.4 set the SNR of signal A to 20dB. Other parameters are not changed. In this case, the performance of the compressed data stays close to that of the uncompressed data.

From the above simulations we can conclude that: in all realistic situations, the degradation of the system performance introduced by our data compression is almost negligible.

3.3 Non-coherent Method

In the previous discussion the SVD method is only used as a data compression method; the signal is reconstructed from those pulses' parameters and then the traditional cross ambiguity processing is used to estimate the FDOA/TDOA. This method may have not fully exploited the advantage of the parameterized pulse train. In this section, we develop a non-coherent method based on the idea of parameterizing the signals

received on both platforms and estimating the FDOA/TDOA directly from those parameters without restoration of signals.

3.3.1 Direct Estimation of the FDOA/TDOA

Let's restate the two signals received at two platforms in the parameterized form:

$$\begin{aligned} x_a(t) &= \sum_{n=1}^N p_a(t - d_n) A_n e^{j\theta_n} \\ x_b(t) &= \sum_{n=1}^N p'_b(t - d'_n) A'_n e^{j\theta'_n} \end{aligned} \quad (3.3.1)$$

where N is the total number of pulses; $p(t)$ is the prototype pulse of signal A; (d_n, A_n, θ_n) is the parameter vector describing the individual pulses in signal A; and $p'(t)$, (d'_n, A'_n, θ'_n) are the counterparts in signal B. Since the two platforms are receiving the signal emitted from the same radar, there must be a one-to-one mapping relation between the pulses in the two signals as illustrated in Fig. 3.5. If the following

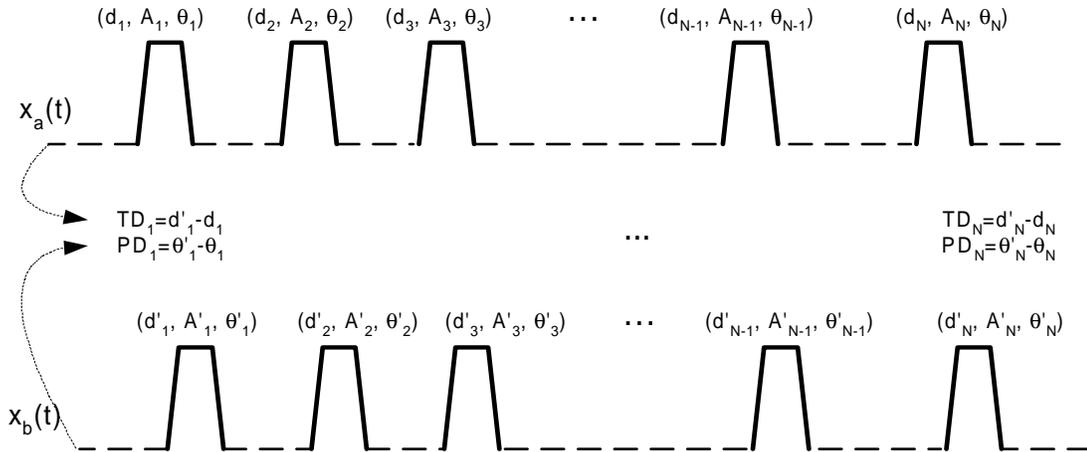


Figure 3.5: Pulse-to-pulse mapping

separation assumptions are valid:

- $TD \ll PRI$, where TD is the TDOA of the two signals, and PRI is the pulse repetition interval;
- $FD \ll F_s$, where FD is the FDOA of the two signals, and F_s is the sampling rate.

we can compare those pulses on a pulse-to-pulse basis. In 1.3.3 we have shown the validity of the two assumption in a typical emitter location system. Therefore, we can define the time difference and phase difference of each pair of pulses as follow:

$$\begin{aligned} TD_n &= d'_n - d_n \\ PD_n &= \theta'_n - \theta_n \end{aligned} \tag{3.3.2}$$

If the *separation assumptions* hold, we claim that:

- TD_n should remain constant for every n and its average can be used as an estimation of the overall TD ;
- PD_n should be linear to d_n and its slope can be used as an estimation of the overall FD .

Therefore, the FDOA/TDOA estimation becomes simple calculation of averaging and linear regression. However, there are still some implementation issues that need attention:

- Because the prototype pulses of the two signals are unlikely to be perfectly aligned, a correcting term should be added to the TD . This correcting term can be produced by cross-correlating the prototype pulses, interpolating the correlation function and finding the peak.

- Because pulses that have bigger magnitudes tend to have higher SNR, the accuracy of the estimation can be improved by using $\sqrt{A_n A'_n}$ as the weights in the averaging and linear regression.

Now we can write out the equations for the estimation of FDOA/TDOA:

$$\begin{aligned} \widehat{TD} &= \frac{\mathbf{W} * \mathbf{TD}}{\text{tr}(\mathbf{W})} + c \\ \begin{pmatrix} \widehat{FD} \\ d \end{pmatrix} &= (\mathbf{M}^H \mathbf{W} \mathbf{M})^{-1} \mathbf{M}^H \mathbf{W} * \mathbf{PD} \end{aligned} \quad (3.3.3)$$

where:

$$\begin{aligned} \mathbf{TD} &= (TD_1, TD_1, \dots, TD_N)^T \\ \mathbf{PD} &= (PD_1, PD_1, \dots, PD_N)^T \\ \mathbf{W} &= \text{diag}(\sqrt{A_1 A'_1}, \sqrt{A_2 A'_2}, \dots, \sqrt{A_N A'_N}) \end{aligned} \quad \mathbf{M} = \begin{pmatrix} d_1 & 1 \\ d_2 & 1 \\ \vdots & \vdots \\ d_N & 1 \end{pmatrix} \quad (3.3.4)$$

c is the correct term produced by cross-correlating $p_a(t)$ and $p_b(t)$ and d can be viewed as an initial phase shift and is of less interest.

3.3.2 Simulation Results

We use the same simulation procedure as in the previous section except that the parameterization is done on both signals and no signal restoration is needed. We investigate two different scenarios: In the first one the signal A is set to be noise-free while the in the second one its SNR is 20dB. Once again we plot the FDOA/TDOA variances versus the SNR of signal B. The results of uncompressed data and compressed data with coherent method (ambiguity processing) are also included as references.

It can be noticed from both Fig. 3.6 and Fig. 3.7 that:

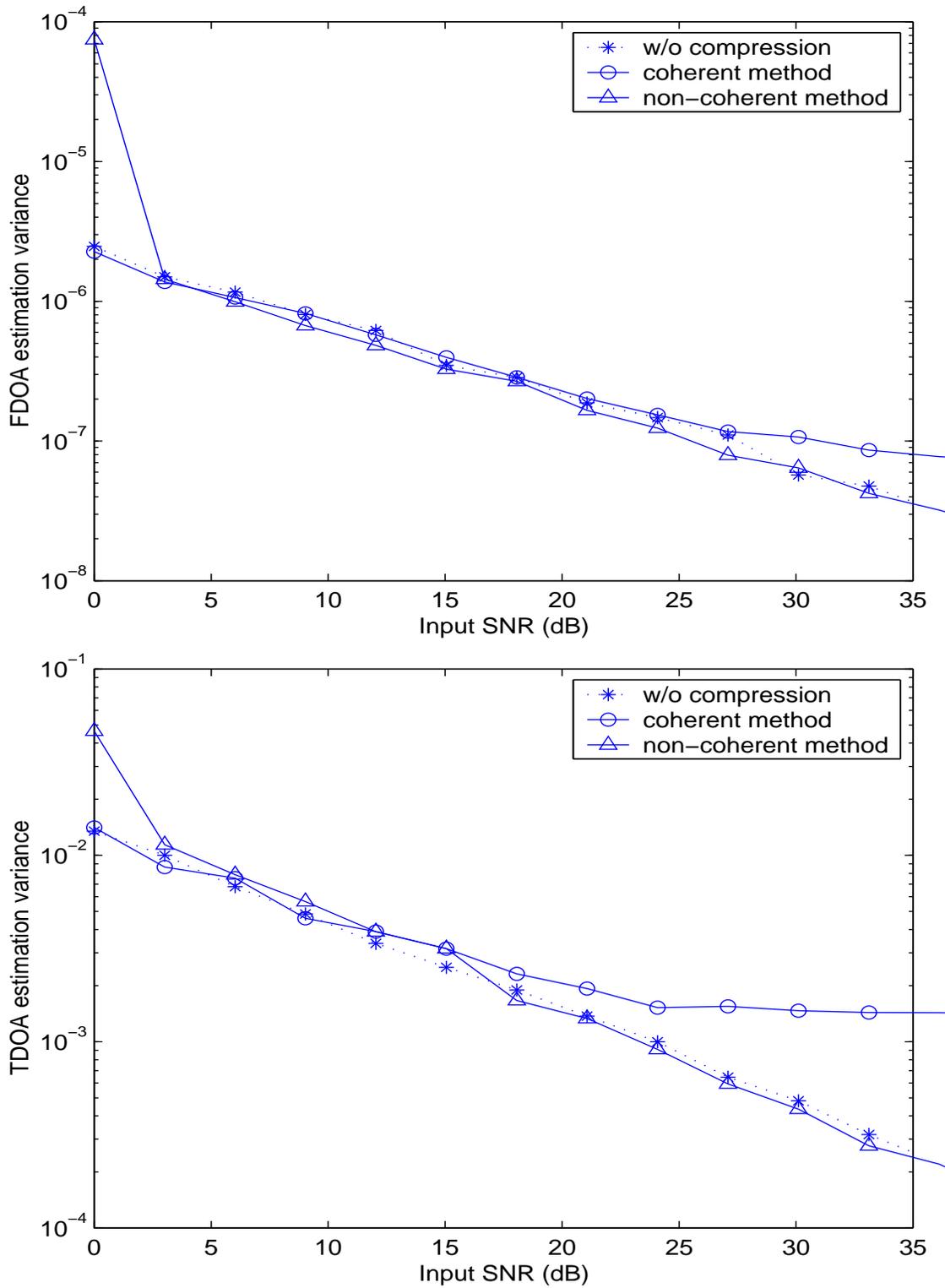


Figure 3.6: FDOA/TDOA accuracy (non-coherent, A noise free)

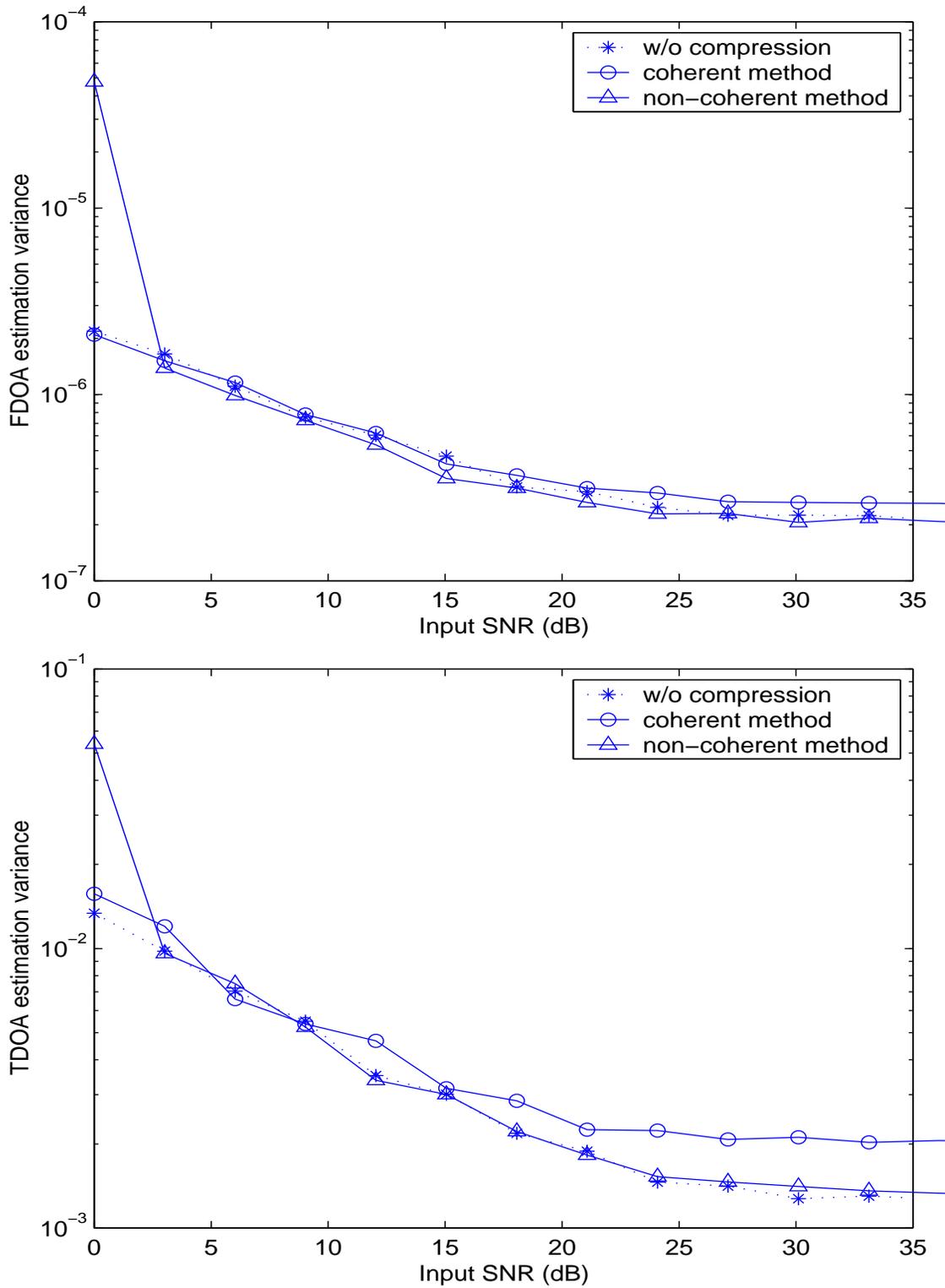


Figure 3.7: FDOA/TDOA accuracy (non-coherent, A 20dB)

- The performance of the non-coherent method matches that of the coherent method for most SNRs.
- A threshold effect is observed at about 3dB; when the input SNR is too low, the performance of the non-coherent method degrades severely.

In terms of accuracy, the non-coherent method is on par with, if not better than, the coherent method above the observed threshold, while the computation is much simpler. However, if the input SNR is below the threshold, we can fall back to the coherent method.

3.4 Parameters Encoding

So far we have not done any aggressive encoding to the parameters. Intelligent bit-allocation can improve the compression ratio because both the coherent method and the non-coherent method are not equally sensitive to the error in all parameters. Specifically, both algorithms are more sensitive to phase error than magnitude error in those parameters. Therefore, we should allocate more bits to phases than magnitudes. In the following quantization scheme we actually use:

- Both θ_n and the phases of the prototype pulses are quantized to 8bits;
- d_n are quantized to 8bits;
- A_n are quantized to 4bits;
- The magnitudes of the prototype pulses are quantized to 1bit ΔM .

In Fig. 3.8 and Fig. 3.9 the performance of quantized parameters and unquantized data (infinite precision) are compared. The SNR of signal A is set to 20dB. It can

be noticed that performance degradation introduced by our quantization scheme is almost negligible.

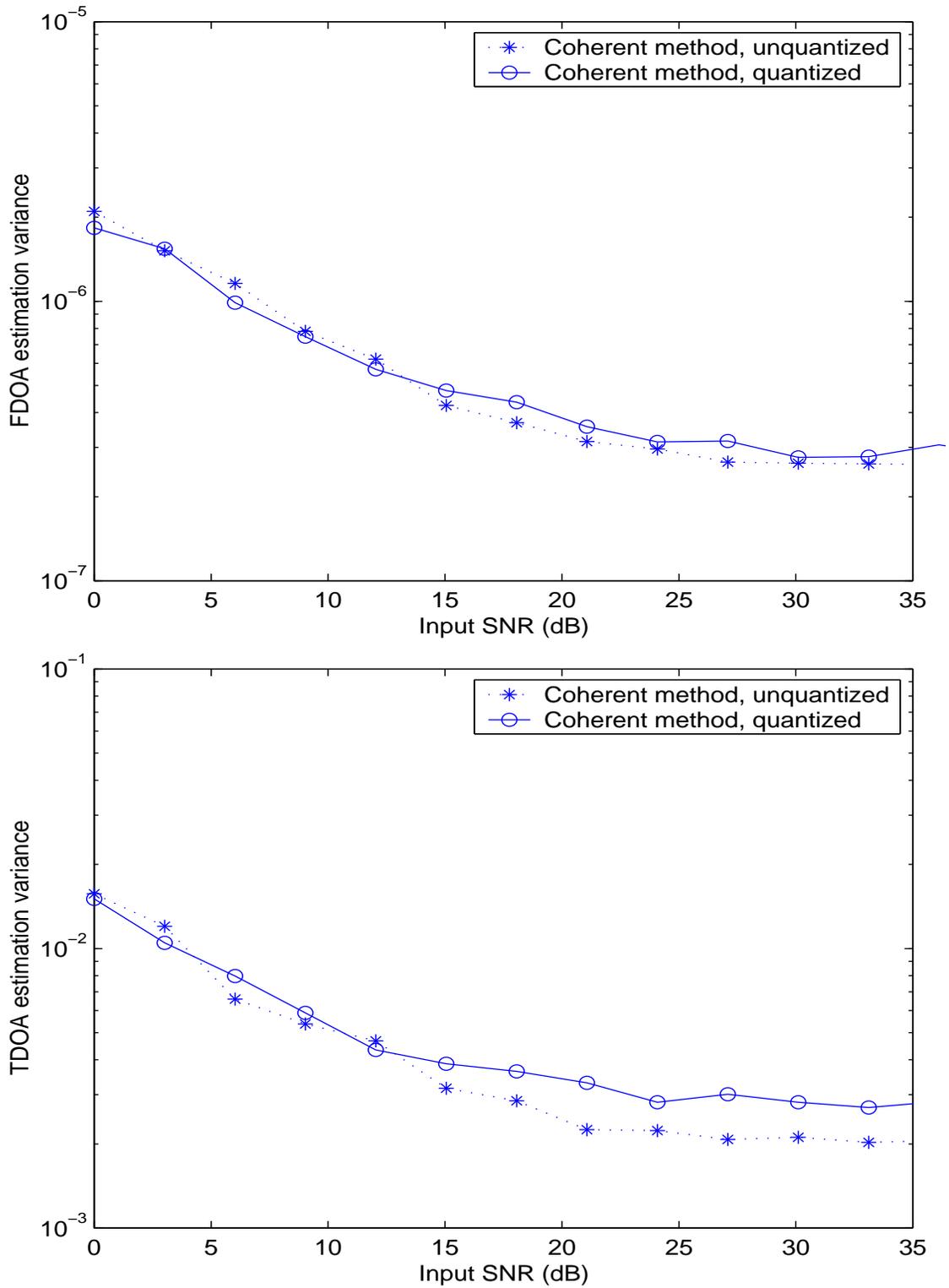


Figure 3.8: FDOA/TDOA accuracy (coherent, A 20dB)

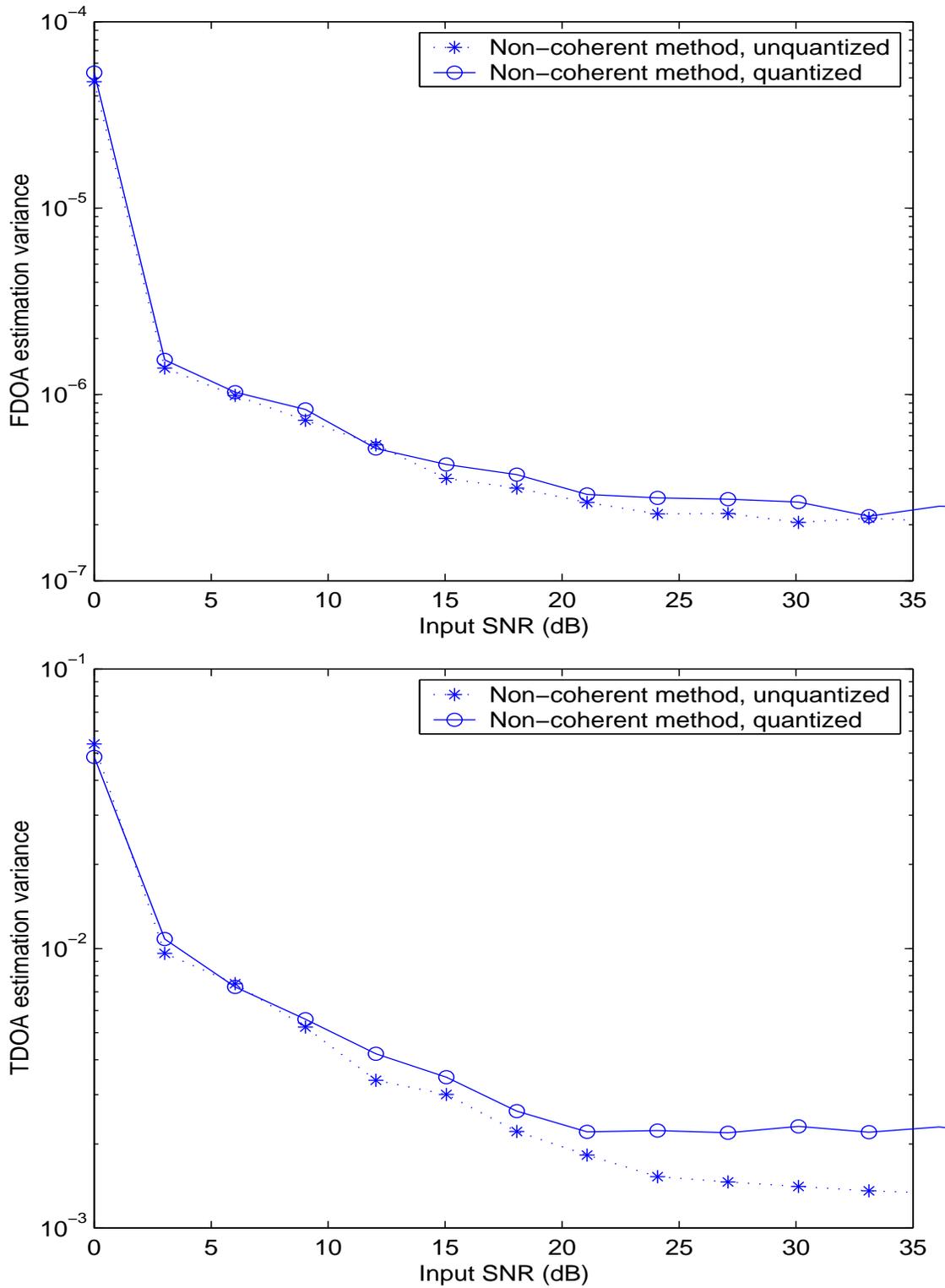


Figure 3.9: FDOA/TDOA accuracy (non-coherent, A 20dB)

Chapter 4

Conclusions and Suggestions for Future Work

4.1 Conclusions

The original goal of this research project was to develop an efficient data compression scheme to compress the signal collected in the coherent emitter location system without compromising the FDOA/TDOA estimation accuracy too much. Throughout the research this goal has been achieved and some other interesting results have been found beyond the original scope. We can summarize our finding as follows:

- The idea of compressing the radar pulse train by parameterizing the pulses has been proven to be viable. Nevertheless, a whole set of algorithms are proposed and tested via computer simulation. High compression ratio ($> 10 : 1$ in most cases) has been achieved with minor, if any, degradation to the FDOA/TDOA estimation accuracy. The added computational complexity to the system is within practical limit.
- The non-coherent method proposed in 3.3 has been proved to be a good alternative to the original cross-ambiguity processing based method. In most cases

it yields similar or better FDOA/TDOA estimation accuracy with much less computational complexity. The limitation of this method is that it can not work below an observed SNR threshold ($\sim 3\text{dB}$).

- Several methods to shift a signal with fractional delay are thoroughly investigated and a new method—the LMS adaptive FD filter—is proposed and tested. While it may not be very useful in this application, it could be very useful in some other applications.

4.2 Suggestions for Future Work

However, this research project is by no means completed. There are some fine-tuning tasks left unfinished as well as some new research opportunities discovered from our research:

- Our algorithm is batch based, which means that the whole data set has to be collected prior to the processing. It would be nice if some on-line processing schemes can be derived from it because on-line processing is more favorable in the emitter location system.
- The quantization and coding of the parameter vector has not been thoroughly investigated. Better quantization and coding can be found and the trade-off between compression ratio and the FDOA/TDOA estimation accuracy is worth further investigation.
- We only studied one specific adaptive fractional delay filter: the LMS FIR adaptive FD filter. The idea of adapting the filter coefficients can be used in

other kinds of FD filter as well.

Bibliography

- [1] M. L. Fowler, “Coarse quantization for data compression in coherent location system,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 36, no. 4, Oct. 2000.
- [2] M. L. Fowler, “Wavelet compression for radar signals,” Unpublished manuscript, 1999.
- [3] S. Stein, “Algorithms for ambiguity function processing,” *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 29, pp. 588–599, June 1981.
- [4] M. L. Fowler, “Data compression for tdoa/dd-based location system,” *US Patent #5,570,099*, vol. Lockheed Martin Federal Systems, Oct. 1999.
- [5] M. L. Fowler, Zhen Zhou, and Anupama Shivaprasad, “Pulse extraction for radar emitter,” *CISS01: 35th Annual Conference on Information Sciences and Systems*, Mar. 2001.
- [6] L. L. Scharf, “The svd and reduced-rank signal processing,” *SVD and Signal Processing II, Algorithms, Analysis and Applications*, pp. 4–25, 1991.
- [7] Khalid Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, San Francisco, 2000.
- [8] D. P. Meyer and H. A. Mayer, *Radar Target Detection: Handbook of Theory and Practice*, Academic Press, New York and London, 1973.
- [9] J. O. Smith and B. Friedlander, “Adaptive interpolated time-delay estimation,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 21, no. 3, pp. 180–199, Mar. 1985.
- [10] G. S. Liu and C. H. Wei, “A new variable fractional sample delay filter with non-linear interpolation,” *IEEE Trans. Circuits Syst.-II: Analog and Digital Signal Processing*, vol. 39, no. 2, pp. 123–126, Feb. 1992.

- [11] P. Knutson, D. McNeely, and K. Horlander, "An optimal approach to digital raster mapper design," *IEEE Trans. Consumer Electronics*, vol. 37, no. 4, pp. 746–752, Nov. 1991.
- [12] S. Kay, "Some results in linear interpolation theory," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 31, pp. 746–749, June 1983.
- [13] G. Oetken, "A new approach for the design of digital interpolating filters," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 27, no. 6, pp. 637–643, Dec. 1979.
- [14] A. G. Deczky, "Equiripple and minimax (chebyshev) approximations for recursive digital filters," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 22, pp. 98–111, Apr. 1974.
- [15] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay," *IEEE Signal Processing Magazine*, vol. 13, pp. 30–60, Jan. 1996.
- [16] S. Sivanand, J. F. Yang, and M. Kaveh, "Focusing filters for wide-band direction finding," *IEEE Trans. Signal Processing*, vol. 39, pp. 437–445, Feb. 1991.
- [17] M. H. Hayes, *Statistical Digital Signal Processing*, John Wiley & Sons, New York, 1996.
- [18] E. Biglieri and K. Yao, "Some properties of singular value decomposition and their applications to digital signal processing," *Signal Processing*, vol. 18, no. 3, pp. 277–89, Nov. 1989.
- [19] J. A. Cadzow, "Signal enhancement: A composite property mapping algorithm," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 36, pp. 49–62, Jan. 1988.
- [20] L. L. Scharf and D. W. Tufts, "Rank reduction for modeling stationary signals," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 35, pp. 350–355, Mar. 1987.
- [21] R. Kumaresan and D. W. Tufts, "Estimating the parameters of exponentially-damped sinusoids and pole-zero modeling in noise," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 30, pp. 833–840, Dec. 1982.