Spectral Estimation Using Nonuniform Sampling

by

James Martin Nohrden

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Master of Science

and

Bachelor of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1995

Author ....
Department of Electrical Engineering and Computer Science
September 8, 1994

Certified by ....
David H. Staelin, Thesis Supervisor (Professor)

Certified by ....
Stephen D. Weiner (Lincoln Laboratory)

Accepted by ....
F.R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Spectral Estimation Using Nonuniform Sampling

by

James Martin Nohrden

Submitted to the Department of Electrical Engineering and Computer Science
on September 8, 1994, in partial fulfillment of the
requirements for the degrees of
Master of Science
and
Bachelor of Science

Abstract

Nonuniform data appears in many applications including phased array radar. If a radar is tracking several targets, it may pulse each target at nonuniformly spaced instants in time. One may characterize a target's motion by its precession and spin frequencies, which can be determined from the radar return. Several methods for spectral estimation using nonuniform sampling are presented and compared. Two effective methods, interpolation and filter banks, are based on periodic nonuniform sampling, or burst sampling. Each burst of samples is taken from a uniform grid and, for the case where only thirty percent of the possible target motion frequencies are of potential interest, they contain approximately five samples. The spacing between bursts is also uniform, approximately fifteen samples long. Both these methods make apriori assumptions that the spectrum is bandlimited and both methods break down if the spectral assumptions are incorrect. In both cases the spectral (bandlimited) assumptions in the frequency domain restrict the sample pattern in the time domain. Methods are compared when approximately thirty percent of the samples are retained from a uniform grid of approximately fifty samples. When the amplitudes of the precession and spin components are approximately equal, the methods which make no spectral assumptions perform adequately. However, in practice the precession component may be greater than the spin component by 10dB or more and assumptions must be made. From simulation results, the two methods that make assumptions do in fact prove to be the only methods considered in this thesis that are useable when the 10dB amplitude ratio exists under the fifty-sample constraint. These two methods are finally compared using real radar data.

Thesis Supervisor: David H. Staelin
Title: Professor

Acknowledgments

First, I wish to thank Professor David H. Staelin for his help and insight throughout the thesis work. I wish to give special thanks to Stephen D. Weiner for both suggesting the problem presented in this thesis, and for giving much advice and encouragement throughout. I also wish to thank Truong Q. Nguyen for his help on the filter banks and continuing encouragement. Especially, I thank Richard F. Mozzicato for checking the math, without which this thesis would not be possible.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Part of the national effort in strategic defense deals with the discrimination of incoming missiles. In a typical battle situation, the aggressor fires multiple missiles carrying warheads together with decoys deployed to overload the resources of the defense. If enough decoys are fired, the defense is unable to intercept all the missiles, and some real warheads will evade the defense. If, however, the defense is able to discriminate between the actual warheads and the decoys, it may use its resources efficiently, intercepting the warheads and allowing the decoys to pass. One way to differentiate between an armed missile and the decoy is through its angular motion. Typically, an armed missile will have a greater mass than the decoy and is deployed more carefully. These differences result in different amplitudes of precession and spin between the two bodies. If the rate of precession and spin can be determined, then it may be possible to differentiate between an actual warhead and its decoy.

The spin and precession cause slowly varying sinusoidal components in the radar and infrared (IR) signals that are returned from the target. With phased array radar, a single radar may be tracking several targets. If this is the case, then it may be difficult to schedule the radar appropriately such that each target is sampled uniformly. If the sample times differ only slightly from uniform sampling, which can happen because of radar scheduling conflicts, techniques such as linear interpolation can be used to compensate this time shift. However, for highly nonuniform samples new techniques are needed. Even if it were possible to perfectly schedule the radar, it may be beyond the resources of the radar to sample all of the targets above the Nyquist rate. In Chapter 4, however, it is shown how nonuniform sampling may actually be made a virtue if the total number of samples is limited by the radar. An IR Track-While-Scan system will sample the targets nonuniformly as long as the target is not directly in the center of the field of view. If there is more than one target, nonuniform sampling is clearly unavoidable.

While these reasons have motivated the current study, the problem does occur in a few other places as well. For example, the equations that govern the analysis of large baseline antenna arrays are closely related to those of sampling theory. Also, in the design of antenna arrays, work has been devoted to achieving high spatial resolution through wide, yet nonuniform, spacing of individual antenna elements [3, 10].

1.2 Background and Previous Work

Throughout this thesis, the true signal is assumed to be a sum of two sinusoids, representing the precession and spin, residing in white noise. The two sinusoids are widely spaced enough so that a lack of resolution is not a problem in telling them apart. Throughout much of the analysis, it is assumed that the two frequencies lie within two non-overlapping frequency bands. Each of these bands typically spans about fifteen percent of the spectrum.

It is certainly possible to determine both of the frequencies and the amplitudes, provided the signal-to-noise ratio (SNR) is high enough. To do this, one can sample uniformly at or above the Nyquist rate, where the Nyquist rate is defined as being twice the highest frequency in the spectrum. This means that, using uniform sampling, if f_h is the highest frequency, then one must take $2f_h$ samples per second. However, we are only looking for two frequencies, and the total bandwidth of the two frequencies together is much smaller than f_h. Uniform sampling above Nyquist may be inefficient because we do not expect to find all the information that this high sample rate is capable of providing.

Therefore, one may ask that, given a finite number of samples, how might they be arranged such that both frequencies can be estimated accurately. One may also ask if the average sample rate may be lowered closer to the total bandwidth of the signal. The question is then, what data can be eliminated, and how much? Also, are some sample patterns more desirable than others? Of course, this depends on the method of sampling and the performance of that method.

Extensive research has been done on spectral estimation using uniform sampling. The Fast Fourier Transform (FFT) is primarily used for uniformly sampled data as are many model based approaches. Some examples of model based estimation methods are the AR (autoregressive), MA(moving average) and ARMA (autoregressive-moving average) models [4]. The most widely used of these is the autoregressive model. These modeling techniques are advantageous in that they improve the resolution of the spectrum over the conventional Fourier transform methods. Unfortunately, almost all the work that has been done in these areas assumes that the data is sampled uniformly in time.

Techniques for frequency estimation using nonuniform sampling and the properties of nonuniform sampling patterns have not been as fully explored. It is known that there are sampling rates below Nyquist [12] that may be used to perfectly reconstruct a bandlimited signal. A sinusoidal waveform composed of two frequencies is essentially a bandlimited signal. Therefore, sampling below the Nyquist rate is possible. However, it is possible to sample at an even lower average rate using nonuniform sampling if some assumptions about the signal are made. Various ways in which such a sample train may be developed will be explored. Comparisons will be based on how accurately each sample train represents the true spectrum of the signal.

Two of the methods, the interpolation method and the filter bank method, make strong assumptions. Both assume that the signal has a spectrum which is zero outside certain frequency bands. While this is not a bad assumption to make for the case of precessing and spinning missiles, the results are completely inaccurate if the assumptions are violated. Methods should therefore be considered that eliminate or alleviate this problem.

1.3 Signal Model

Taking into account that the target of the radar (or IR system) is both spinning and precessing, it is reasonable to assume that the return from the target will be a sum of sinusoids. It is also reasonable to assume that for conical shaped objects, the ratio of the two frequencies is on the order of ten to one, and the amplitude of the lower frequency is on the order of ten times larger than the amplitude of the higher frequency. Finally, errors in the assumption of the uniform, conical mass distribution of the target, atmospheric effects, and measurement error can be approximated by an assumption of zero mean white gaussian noise. Taking all this into account, we have the the following signal model:

$$\begin{aligned} s(t) &= A_1 \sin(2\pi f_1 t + \phi_1) + A_2 \sin(2\pi f_2 t + \phi_2) \\ x(t) &= s(t) + n(t) \end{aligned} \tag{1.1}$$

where $n(t)$ is zero mean white gaussian noise, and

$$\begin{aligned} f_2 &\approx 10 f_1 \\ A_1 &\approx 10 A_2. \end{aligned} \tag{1.2}$$

All of this is, of course, only a first approximation, and a more detailed model may prove to be more accurate. However, the thrust of this thesis is not so much to find the best solution to this entire problem, but more to find efficient sample patterns and sampling techniques based on this simplified model.

1.4 Outline of Thesis

The following chapters explore many different approaches to eliminating data from a uniform sample train such that the major features of the spectrum are preserved. The first method follows some work which has been done on nonuniformly spaced antenna arrays. In [3], P. Jarske et. al. showed that if elements are nonuniformly removed from an above Nyquist [1] array, then it is possible to efficiently remove elements such that the error produced is near some global minimum. However, we find that even with the increase in efficiency that this method may provide in eliminating samples, it proves to be slow and produces results that leave much to be desired.

The rest of the approaches somehow convert the nonuniform data into uniform data. This is convenient because all the standard techniques that have been developed for spectral estimation using uniform data may be directly applied. While extensive modeling techniques may be used, the speed of the FFT will be exploited here, and it will be the vehicle by which the spectrum is created.

The first technique which gives uniform data, presented in Chapter 3, is an interesting idea based on the autocorrelation function. Instead of taking the Fourier Transform of the signal, why not instead take the Fourier Transform of an approximation to the autocorrelation function? If a sample pattern is created such that every possible lag is represented, then the autocorrelation function will be uniformly sampled, even if the original data has not been sampled uniformly. It is difficult, however, to find a sample pattern that has a uniform number of samples at each lag. This nonuniformity in the number of samples leads to errors, but they are much smaller than the errors due to nonuniformity that occurs in the original data.

The last two methods to be considered rely on periodic nonuniform, or burst sampling. This is convenient because it is common for a radar to periodically send a burst of samples toward the target. Both methods start with a set of samples which occur in bursts. Then, based on some assumptions about the spectral content of the

[1] For antenna arrays, there is no Nyquist as such, but there is a minimum element spacing to avoid aliasing.

signal, the points between the bursts are approximated.

In Chapter 4, the Cardinal Series of the Shannon sampling theorem [7] is extended [11] to bandlimited signals which have been sampled in a periodic nonuniform fashion. Analogous to the Shannon sampling theorem, an interpolating function is used to estimate the points between the bursts of samples. The Shannon sampling theorem assumes a lowpass spectrum, and [11] assumes a bandlimited spectrum. The theory is presented first, and simulation follows.

Chapter 5 uses filter banks to approximate the points between the bursts. Some filter bank theory is first presented, as well as the required theory on filter design specific to this method. The ideas for bandlimited signals presented in [6] are implemented, and the derivation is also presented. Simulation results are compared to all the other methods, and the trade offs are considered.

The interpolation method and the filter bank method are tested using real radar data. Both methods perform fairly well, as will be shown. The first two methods do not perform well enough even in the simulations to be considered on the radar data, where the ratio of the two amplitudes is over 20dB.

Chapter 7 addresses the issue of making the wrong assumptions about the spectrum using the interpolation and filter bank methods. A simple method to resolve the problem is considered. Lastly, the summary goes over and considers the most important results and conclusions about the methods presented in the thesis.

1.5 Simulation Setup

For purposes of most easily comparing the many different techniques presented in this thesis, a standard signal which satisfies the model in Section 1.3 is used. Because the methods differ so widely, the actual samples that each of them use will have to vary from one method to the next. Except where otherwise noted, the signal will have the following parameters, and be approximately fifty samples long:

$$s(t) = A_1 \sin(2\pi f_1 t + \phi_1) + A_2 \sin(2\pi f_2 t + \phi_2)$$

where

$$\begin{aligned} A_1 &= 1.0 & A_2 &= 0.1 \\ f_1 &= 0.05 & f_2 &= 0.4 \\ \phi_1 &= \pi/3 & \phi_2 &= 0. \end{aligned} \tag{1.3}$$

The amplitudes are scalars, the frequencies have been normalized with respect to the sample rate, and the phases are in radians. When noise is added to the simulation, a zero mean gaussian random variable is generated with a standard deviation indicated by *noise*. The random variable is generated on Matlab. All simulations are implemented on Matlab as well, and supported on the Athena computer system at MIT.

Chapter 2

Throw Away Method

2.1 Introduction

The idea behind the throw away method is to throw away those samples which cause the minimum loss of useful information and to throw away samples such that the sample pattern is optimal in some way. We start with a complete set of uniform above Nyquist samples, and eliminate some of them. An error measure is necessary to grade the importance of each sample. As an example, the error measure could be simply the difference between the Discrete Fourier Transform (DFT) with the samples removed and the DFT with all the samples. Whatever the error measure, we would like to find those samples which minimize it, and then in a sense, have an optimal sample pattern.

A very similar problem has come up in large baseline antenna arrays. The designer of the array has a set of elements which are spaced on a uniform grid closely enough to avoid spatial ambiguity. However, each antenna element is costly, and eliminating some of them would be desirable if it could be done so without degrading the array performance considerably. This cost consideration has lead to some work on nonuniformly spaced antenna arrays [10, 3]. Mathematically, the relationship between the elements of an antenna array and the gain pattern is essentially the same as that of samples of a signal and its spectrum. Thus, antenna array design should lead to some insight in sample pattern design.

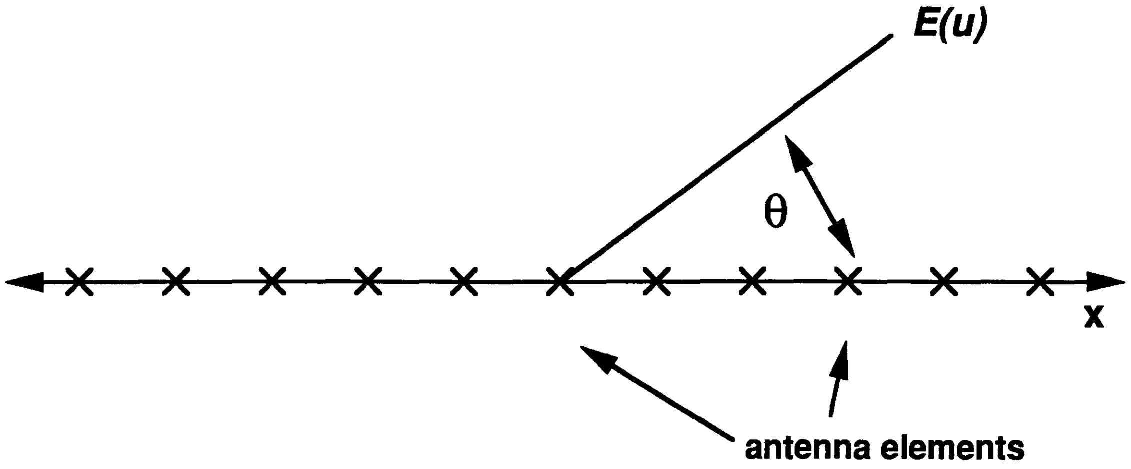

Figure 2-1: An antenna array and the definition of some of the parameters which describe it.

2.2 Antenna Arrays

Significant research has been devoted to direction finding using array antennas. If one is given an array of isotropic antennas that are either transmitting or receiving, the gain pattern as a function of angle can be related to the antenna pattern in the following manner (see Figure 2-1):

$$E(u) = \sum_{n=1}^{N} A(x_n) e^{\frac{j2\pi u x_n}{\lambda}} \tag{2.1}$$

where:

x_n is the position of the Nth element.

$A(x_n)$ is the amplitude of the element at $x(n)$.

$u = \sin(\theta)$, where θ is shown in Figure 2-1.

λ is the wavelength used by the antenna

This is then simply a DFT relationship between the spatial coefficients, $A(x_n)$, and the angle coefficients, $E(u)$. It is because of this relationship in (2.1) that antenna array design is closely related to filter design in signal processing. $A(x_n)$ in array design is $h(n)$ in filter design, and $E(u)$ is $H(\omega)$.

The goal in arrays is to design the antenna array such that a desired gain pattern is closely approximated. Here, a sample pattern is to be designed such that when a signal is present, a specific (known) spectrum is closely approximated. This is how the work found in [3] applies to the problem in this thesis. The method used in [3] was shown to produce a pattern with an error that reaches a local minimum, and that minimum is close to the global minimum. The following is a description of the throw-away method presented in [3] adapted to the problem of nonuniform sampling:

1. Choose an initial duration, D, and fill that time period with enough uniform samples to satisfy the Nyquist criterion.
2. Define an error measure ϵ (i.e. the change in DFT)
3. Remove one of the samples, choosing the sample which causes the minimum error, ϵ.
4. Repeat above until the desired number of samples is reached, or the error is intolerable.

The above method will in a sense provide an optimal sample pattern. Unfortunately, it is optimized for a simulated signal, with a known phase. The sample pattern may alternately be optimized for an ensemble of phases, and this is also shown in the simulations. It is highly unlikely that in the radar applications presented earlier the phase will be known apriori. The question therefore arises as to whether or not this method performs well regardless of the phase. Because the sample pattern is optimized for a signal where the frequency, amplitude, and phase are assumed a priori, we need to know how much the results degrade as these assumptions are violated.

2.3 Spectral Error

In this thesis, it is assumed apriori that the spectral content of the signal is restricted to certain known bands. There is a lower frequency that varies within a known band, depending on the missile, and a higher frequency that varies within another band. It

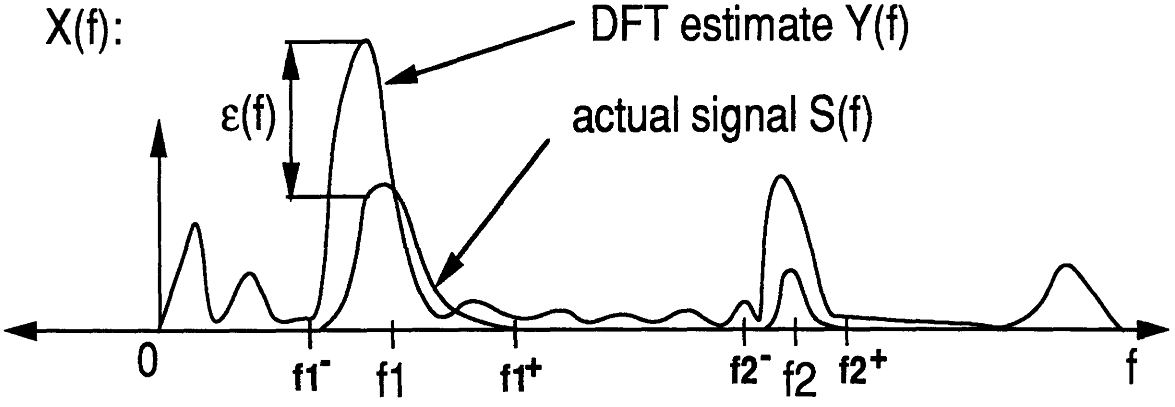

Figure 2-2: Definition of the error measure, ϵ.

would therefore be reasonable to grade the spectral estimate of a particular sample pattern by the accuracy of the DFT in those bands only.

Thus the error measure, ϵ, that will be used in this thesis will be the difference between the true spectrum (above-Nyquist DFT) and the spectrum generated with the DFT in the assumed spectral regions. Specifically, we have (see Figure 2-2):

$$\epsilon = \int_{f_1^-}^{f_1^+} |Y(e^{j\omega}) - S(e^{j\omega})| d\omega + \int_{f_2^-}^{f_2^+} |Y(e^{j\omega}) - S(e^{j\omega})| d\omega \tag{2.2}$$

2.4 Simulation

This section will show the results of the optimal throw-away method for the signal model described in Section 1.5. Before including in the signal model an amplitude ratio of 10dB, we start with equal amplitude sinusoids. Figure 2-3 shows what happens when 15 of the original 50 above-Nyquist samples are saved using the throw-away method and the FFT is performed on the resulting irregular sample train (note that the FFT is the same as the DFT if the nonuniform samples come from a uniform grid). The results are quite good, and both frequency peaks are clear. However, the sample pattern has been optimized for a particular phase. Checking to see if the performance degrades if the phase is changed, we see the results in Figure 2-4. Indeed the phase change does degrade the performance considerably. This is totally incompatible with the model in this thesis.

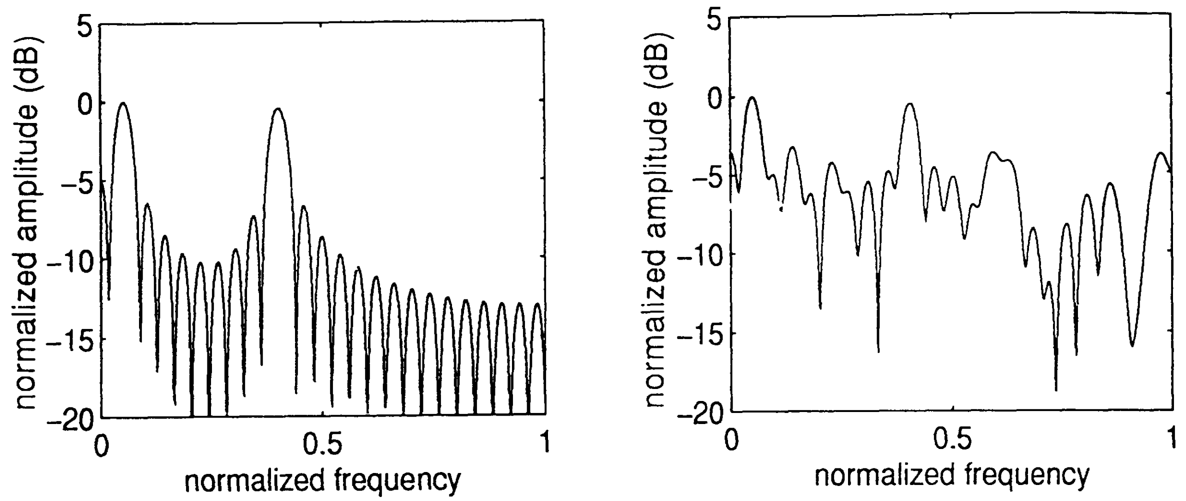

Figure 2-3: The spectrum of the original signal with all 50 samples is shown on the left, and the spectrum using the optimal throw-away method is shown on the right. Note that even though only 15 of 50 samples remain, both frequency peaks are clear.

Figure 2-5 shows an alternative. This shows the spectrum for an arbitrary phase when the sample pattern is optimized for an ensemble of phases instead of for just one phase. The algorithm optimizes the sample pattern taking into account the error produced as each sample is thrown away for nine different test signals, each with a different phase. The nine test signals were chosen by taking phase values of 0 degrees, 60 degrees, and 120 degrees for both the precession and spin components and combining them in the nine possible combinations. Unfortunately, even when optimizing over an ensemble of phases, the performance can still be quite poor for certain phases. The two peaks are not very clear for the case shown in Figure 2-5.

If a radar is tracking a missile, there is no way to know the phase of the incoming signal apriori. Thus, because of the phase sensitivity of this method, we must look for other methods which do not have this problem. As we shall see, the following methods are less sensitive to phase changes. There is no point in testing the case of a 10dB amplitude ratio since this method cannot first pass the test for phase sensitivity.

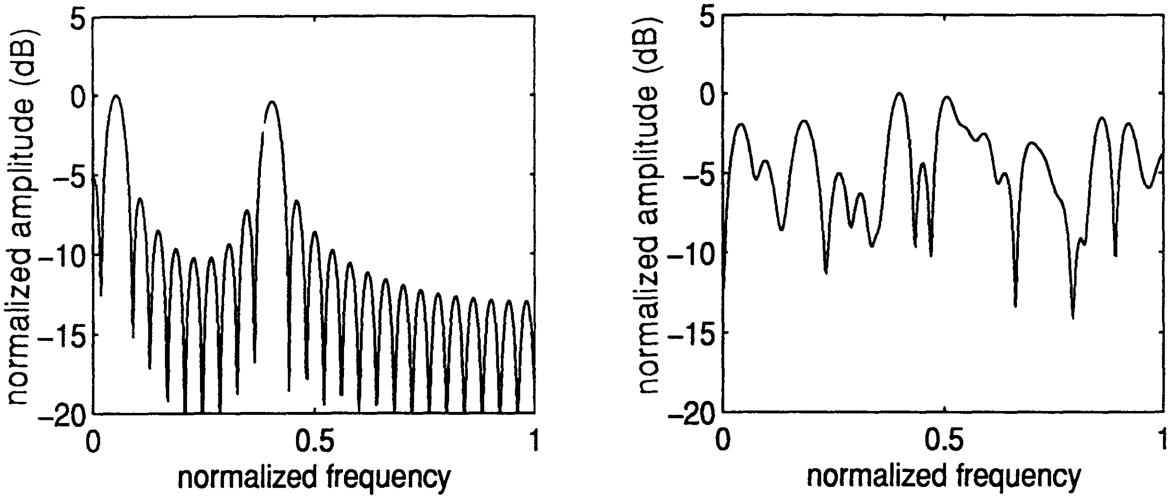

Figure 2-4: The spectrum of the original signal is shown on left (all 50 samples). On the right is the optimal throw-away spectrum when the phase is unknown. The peaks are not clear.

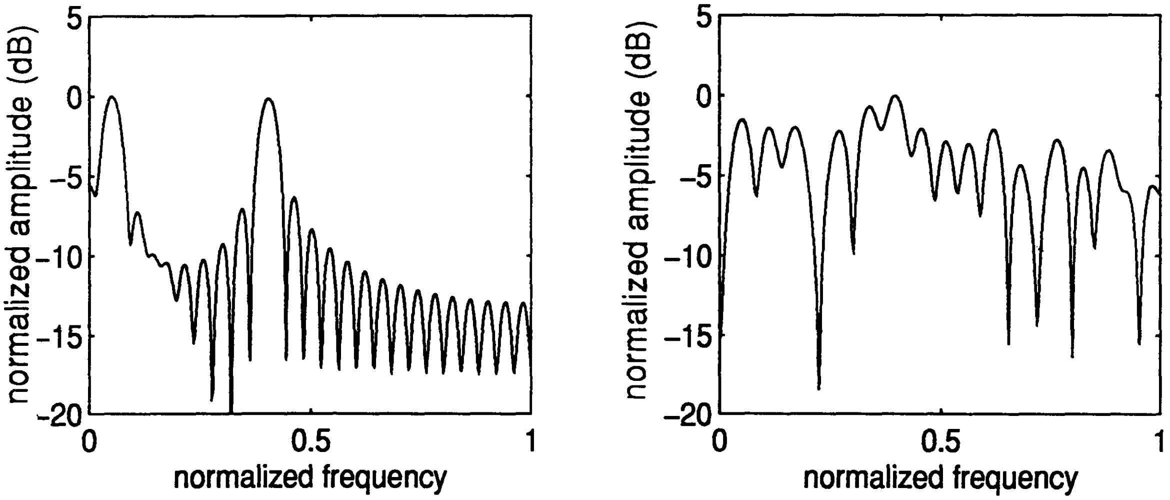

Figure 2-5: The spectrum of the original signal is shown on left (all 50 samples). On the right is the optimal throw-away spectrum optimized for an ensemble of phases when the phase is arbitrary. The peaks are still not very clear, thus the phase sensitivity is a problem.

Chapter 3

Autocorrelation Method

3.1 Introduction

The method presented in this section is based on the autocorrelation function. It is motivated by the idea of lags in sampling; therefore it may most appropriately be called the all-lag method or the all-spacing method. The autocorrelation function refers to the correlation between samples with a given lag. It is essentially the expected value of the correlation between two samples for each lag. Specifically, to find the autocorrelation function of a uniform discrete time data sequence $x(n)$ with N samples, the autocorrelation function $a(n)$, could be defined as:

$$a(n) = \frac{1}{q(n)} \sum_{l=0}^{N-1} x(l)x(l+n). \tag{3.1}$$

where $q(n)$ is the total number of pairs of samples that are separated by a lag of n. Thus, at each n, $a(n)$ is the expected value of the correlation between samples that are separated by n. It is also well known [12] that the Fourier transform of the autocorrelation function, often called the power spectrum, $S(\omega)$, is the magnitude squared of the Fourier transform of the signal, $X(\omega)$.

$$S(\omega) = |X(\omega)|^2 \tag{3.2}$$

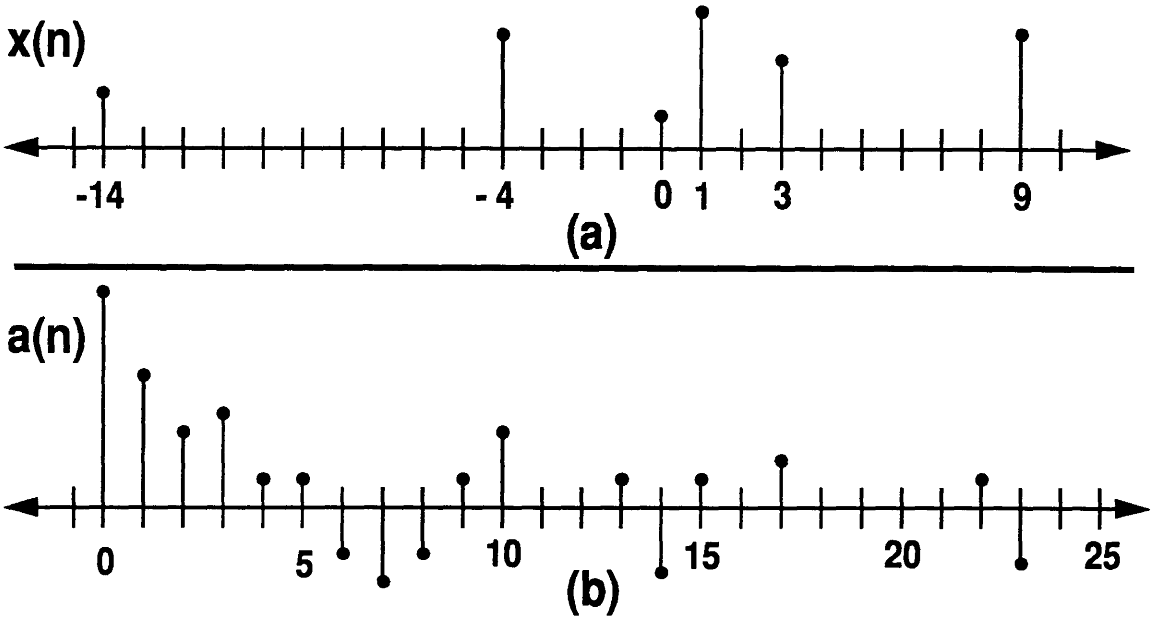

Figure 3-1: (a) Nonuniform samples picked such that the autocorrelation function (b) is as uniformly sampled as possible.

Thus, if the phase of the signal is not needed, and only the amplitudes and frequencies in the spectrum are what we are looking for, it is just as well to find the spectrum with the autocorrelation as with the original $x(n)$. This is indeed the case. Hopefully, this method will not be as phase sensitive as the throw away method since we are actually looking at the power spectrum.

3.2 All Spacings - Golumb's Ruler

In this thesis, the idea is to eliminate samples whenever and wherever possible. So why not use a sample pattern that includes as many lags as possible with as few samples as possible? The benefit of this technique is that there are fewer samples, but there is still uniform sampling of the autocorrelation function, thus the FFT and modeling techniques may be used. This idea of maximum number of lags for a minimum number of sample values leads us to Figure 3-1.

Figure 3-1 shows a nonuniform sample train in (a), and the autocorrelation function in (b). Note that although the data is actually nonuniformly sampled, the

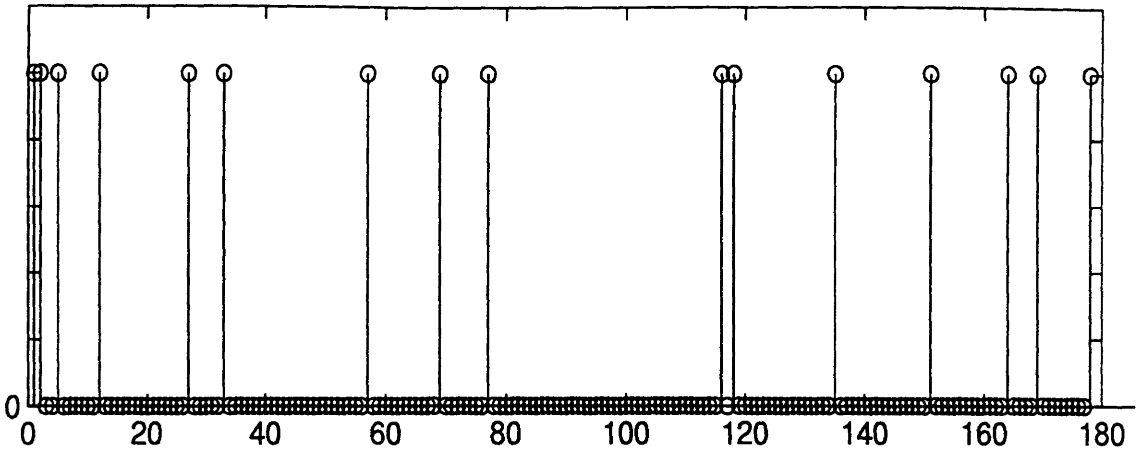

Figure 3-2: An example of a spanning ruler with 15 points.

autocorrelation function is uniformly sampled up to a lag of 10. Thus the goal here is to find sample patterns with a minimum number of samples such that the autocorrelation is as uniformly sampled as possible. This leads to the Golomb and perfect rulers [5].

A *perfect ruler* is one that has one and only one sample for each lag in the autocorrelation function [5]. A perfect ruler has exactly the property we are looking for. It has a minimum number of nonuniform sample points with a maximum number of uniform autocorrelation lags. It is shown in [5] that no perfect rulers exist for rulers with more than 4 samples. There are three possible ways to relax the requirements. The first is a *covering ruler*, which finds a set of samples such that it measures every lag up to a lag L in *at least one way.* The second is a *spanning ruler*, which finds a set of samples such that it measures every lag up to some lag L in *at most one way* [5]. Figure 3-2 shows the shortest spanning ruler for 15 samples. There is no intuitive reason why either the covering ruler or the spanning ruler should provide a better set of samples for spectral estimation, or whether something in between might be better. Therefore, the third way to relax a perfect ruler is to find a set of samples which is a compromise between the two types of rulers. The samples are not distributed in any optimal way, but they are a trade off between the covering ruler and the spanning ruler.

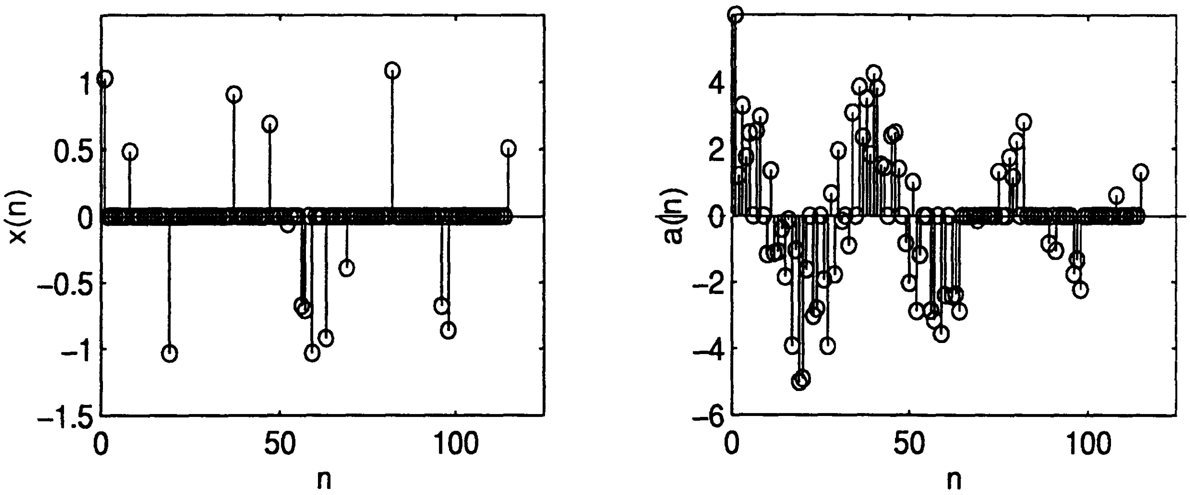

Figure 3-3: 15 sample points chosen such that the autocorrelation function is sampled uniformly. On the left are the samples, and on the right the resulting autocorrelation function. Note that sampling of the autocorrelation function is almost perfectly uniform up to about 60 samples.

3.3 Simulation

As with the optimal throw-away method, the first test is sinusoids of equal amplitude. Also, to form a basis for comparison, a sample pattern with 15 samples is used. The original signal is the same as before (Section 1.5). The sample pattern is of the third type mentioned in the previous section, a combination of the covering ruler and the spanning ruler. Figure 3-3 shows the 15 samples and the autocorrelation function. Note that the autocorrelation function is nearly uniformly sampled up to about 60 samples.

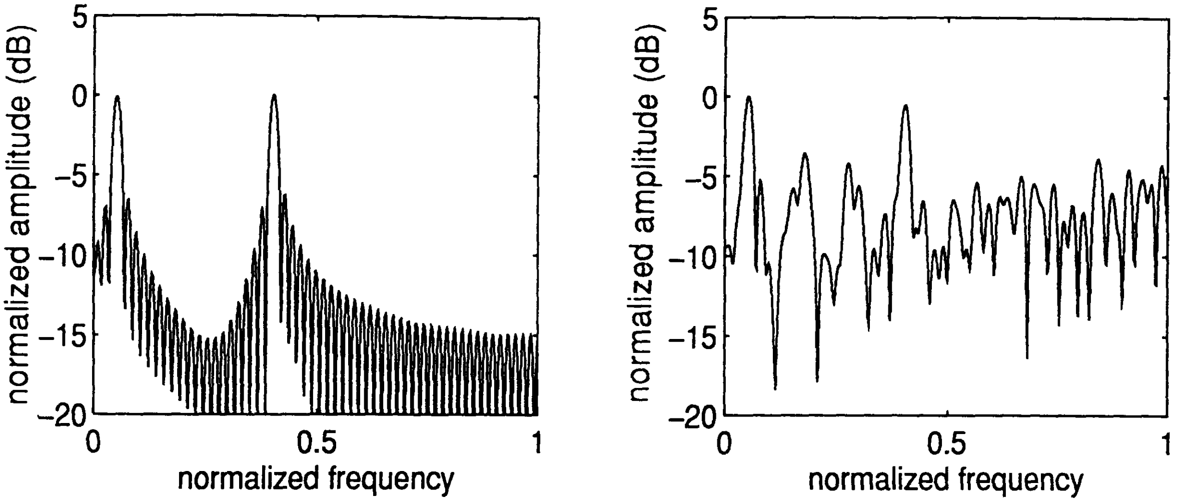

Figure 3-4: All-spacings method. On the left is the original spectrum. On the right is the estimated spectrum with only 15 samples.

Next we move on to the frequency domain. Figure 3-4 shows the original spectrum, and the spectrum when only 15 samples are retained. Again, as we saw for the optimal throw-away method, when the amplitudes are equal, the results are pretty good. However, with this method, if the phase is changed the peaks are still visible. This is shown in Figure 3-5. Thus this method is more useful than the optimal throw-away method in that it is not as phase sensitive.

Now that we have a method which is not so phase sensitive, we can go on to the second test. The signal model assumes an amplitude ratio of about 10dB. If the input signal is thus changed to that on the left in Figure 3-6, the lower pèak is no longer detectable. This is shown on the right side of Figure 3-6. Note that the peak is simply too weak to be detected at all using the all-spacings method. Therefore, we can conclude that the all-spacings method is not useful for this application. The precession and spin ratios will be at least 10dB, and therefore other methods must be examined that can better deal with this situation.

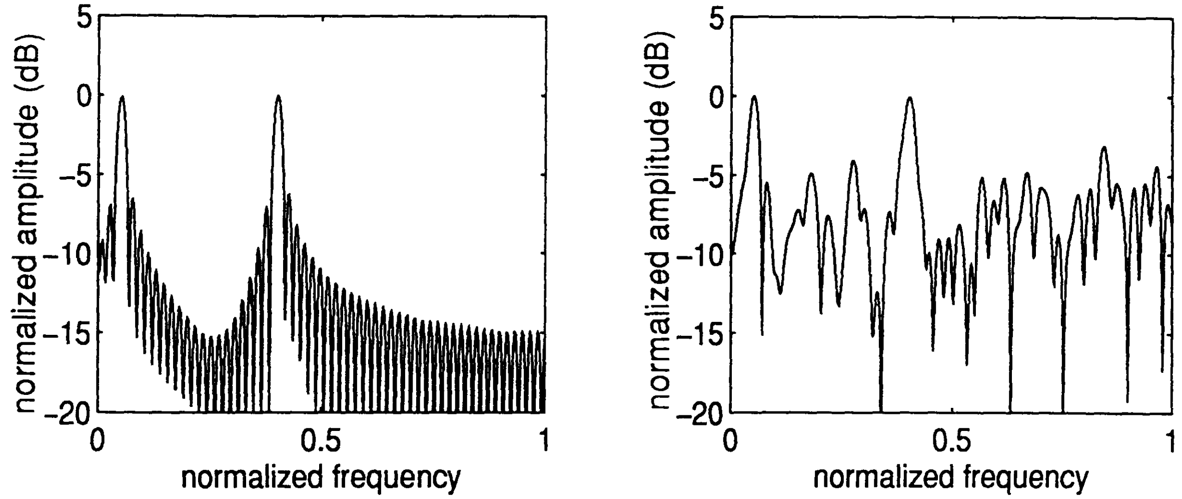

Figure 3-5: Test for phase sensitivity. When the phase is changed arbitrarily, note that both peaks are still detectable.

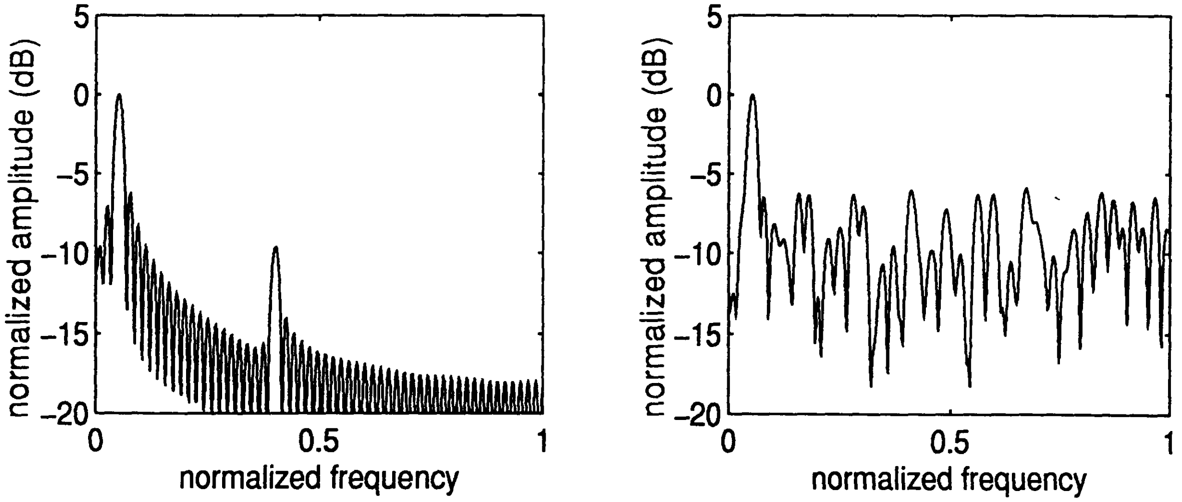

Figure 3-6: Amplitude ratio of 10dB. Note that when this is the case, the higher frequency is too small to be detected using this method.

Finally, it would be interesting to see how simply picking the samples at random compares. This is shown in Figure 3-7. The 15 samples were chosen using a uniform probability distribution. The peaks from the two frequencies are not clear, and this demonstrates that the all-spacings method is better than both the optimal throw-away method and random sampling when the frequencies have equal amplitudes.

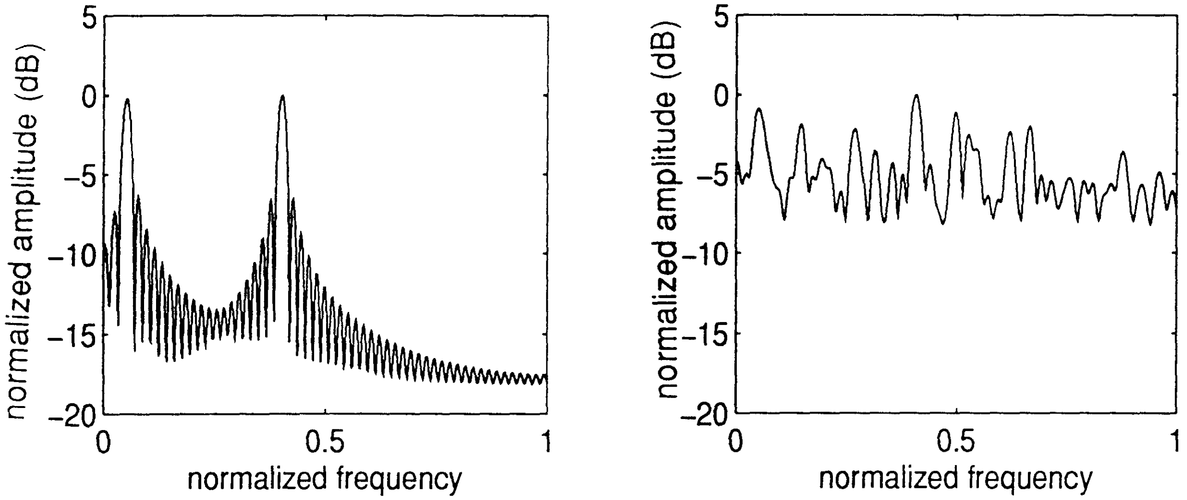

Figure 3-7: Random sampling. On the left is the original spectrum. On the right is the spectrum when the 15 samples are chosen from a random uniform distribution.

Chapter 4

Interpolation Method

4.1 Introduction

The two methods in Chapters 4 and 5 start with a set of periodic nonuniform samples, and from those samples and some apriori assumptions, create the rest. Both methods rely on periodic nonuniform, or burst, sampling and have some constraints due to the spectral assumptions they require. The method described in this chapter is referred to as the interpolation method, and the method of Chapter 5 will be referred to as the filter bank method.

Periodic nonuniform, or burst sampling is shown in Figure 4-1. Note that burst sampling may also be seen as a set of uniform sample trains, each one delayed slightly with respect to the other. Because this type of sampling can be decomposed into delayed uniform sample trains, for which sampling theory is very well developed,

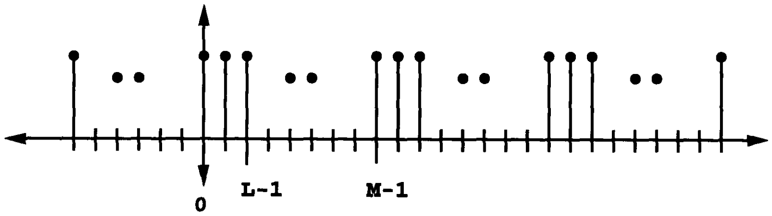

Figure 4-1: Periodic Nonuniform Sampling, and the definition of L and M.

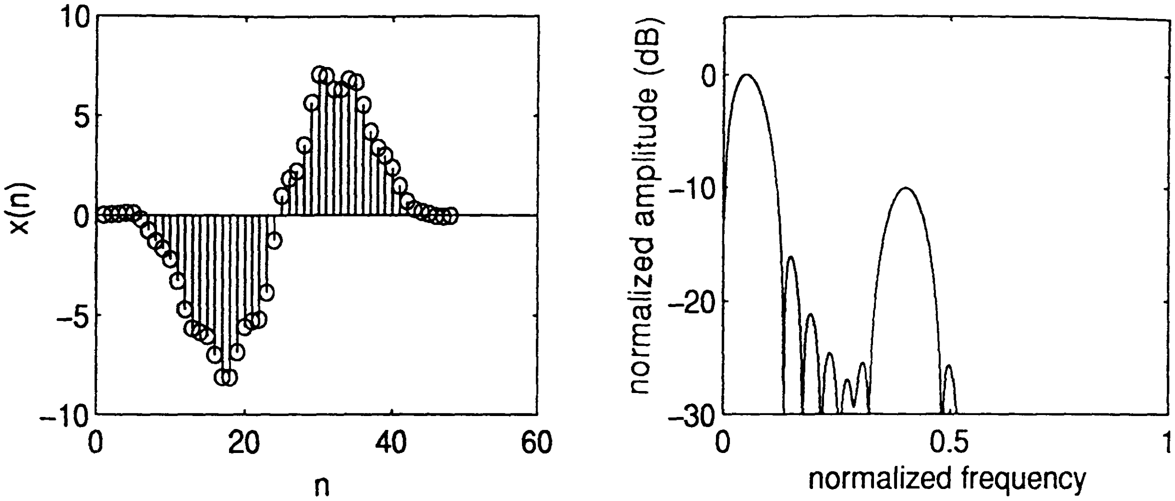

Figure 4-2: Intuitive motivation for burst sampling: This figure shows the oversampled version with all 50 samples.

both the interpolation method and the filter bank method are also well-grounded theoretically and give a more solid approach to the problem that this thesis addresses than did the methods in Chapters 2 and 3.

There is also a more intuitive reason for why periodic nonuniform sampling (or burst sampling) might be applicable to the problem addressed by this thesis. Consider Figures 4-2–4-5. These figures show that if the two sinusoidal components have a frequency ratio of about 10dB, and only 15 of 50 samples are kept, the lower peak cannot be resolved using conventional uniform sampling methods. In Figure 4-3, the duration is too short, and in Figure 4-4 the sample spacing is too long; Figure 4-5 shows a pattern which is a compromise.

4.2 Shannon Sampling - The Cardinal Series

The Shannon sampling theorem, expressed through the Cardinal series, refers to how a signal may be reconstructed from its samples, provided some requirements are met on the samples. Those requirements are essentially the requirements of the Nyquist theorem. In other words, if the samples are uniform and the sample rate is at least twice the highest frequency component of the signal, then the following series, called

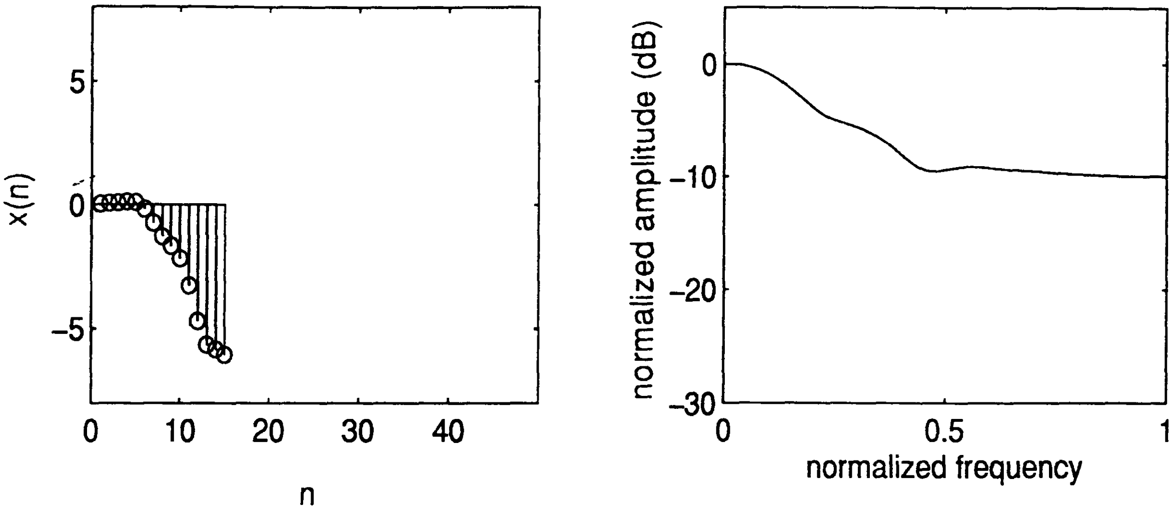

Figure 4-3: Intuitive motivation for burst sampling: This figure shows placing 15 samples all together, and the poor resolution that results.

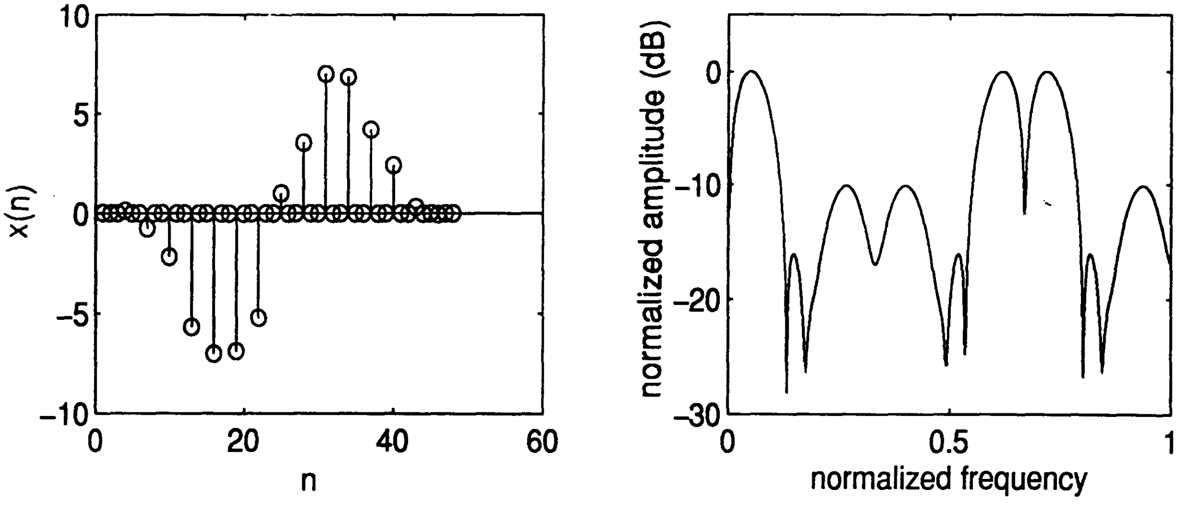

Figure 4-4: Intuitive motivation for burst sampling: This figure shows spreading the 15 samples out, and the aliasing that results.

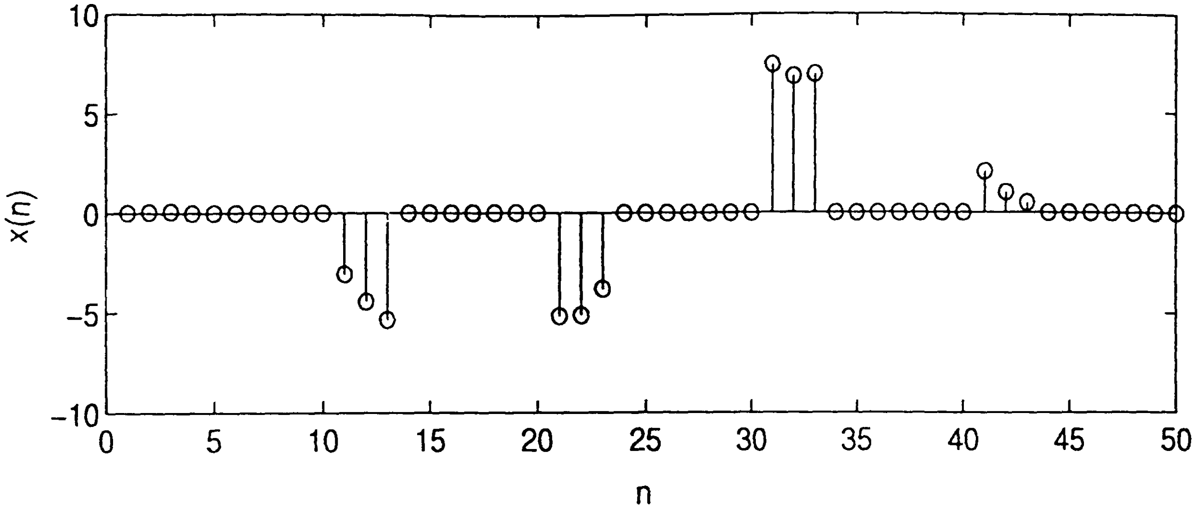

Figure 4-5: Intuitive motivation for burst sampling: Bursts of three. Note that there is both small sample spacing and long duration. However, the FFT is suitable only for uniform sampling and thus cannot be used directly.

the Cardinal series [7], is valid for reconstructing $x(t)$ from the samples $x(n)$:

$$x(t) = \frac{1}{\pi} \sum_{n=-\infty}^{\infty} x(\frac{n}{2f_h}) \frac{\sin[\pi(2f_h t - n)]}{2f_h t - n} \tag{4.1}$$

where f_h is the highest frequency component of the lowpass bandlimited signal. Thus, if one has a set of data points satisfying the Nyquist criterion, then all the rest of the points in between may be interpolated exactly using the Cardinal Series. For example, one might want to upsample a signal by two, finding the value between every pair of samples. Using the Cardinal series, the values in between which are the sum of the sinc functions can be interpolated exactly, as shown in Figure 4-6. This leads right to the extension. If we have bursts of data in time, and we restrict ourselves only to certain frequency bands instead of just lowpass, then it seems possible that another "Cardinal Series" can be found for that case. Indeed, it can, and that is the subject of the following section.

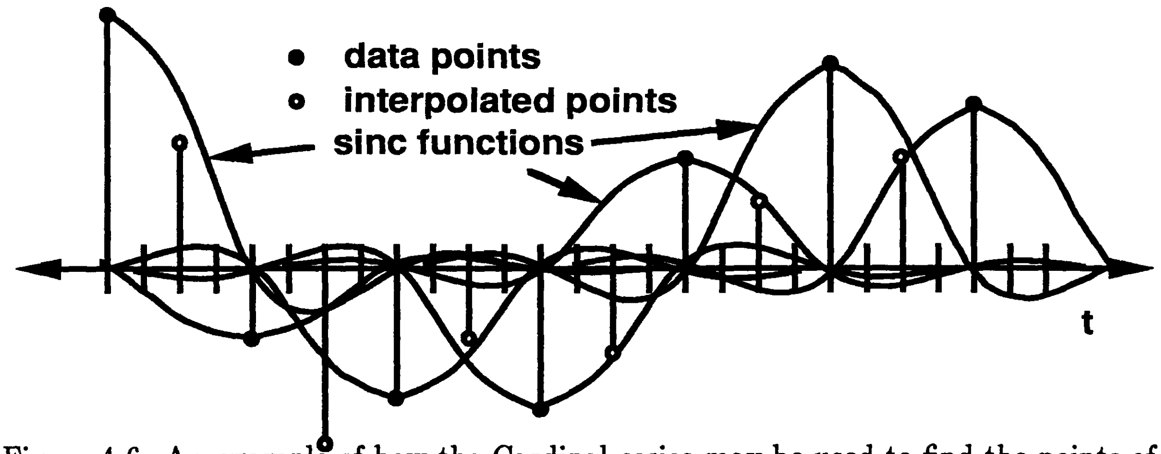

Figure 4-6: An example of how the Cardinal series may be used to find the points of an upsampled signal exactly. The solid dots represent the original data points, with the sinc functions passing through them. The hollow dots represent the sum of the sinc functions at that time.

4.3 Scoular's Extension

The Cardinal series provides a means to estimate a signal for all time from the uniform data samples. The signal is also assumed to be lowpass. However, the motivation for this thesis is a spectrum of two frequencies that are widely spaced. It would then certainly be beneficial if a multiband assumption could be made about the spectrum rather than the lowpass assumption of the Cardinal series. Scoular's extension is exactly that. We assume a multiband spectrum instead of a lowpass spectrum. The other difference is that the data is sampled in bursts rather than uniformly in time.

The results laid out in [11] are summarized as follows: Consider a signal $x(t)$ with a spectrum that is zero outside of L bands (including negative frequency bands) of equal bandwidth $2W = 1/M$. Assume that the bands have center frequencies that are spaced $2f_c$ apart and such that f_c/W is an integer. Define the periodic nonuniform sample points by:

$$\begin{aligned} t_{ni} &= Mn + \tau_i \\ i &= 0, 1, \ldots, L-1 \\ n &= 0, 1, \ldots, N-1. \end{aligned} \tag{4.2}$$

Such a multiband signal may be represented in the following form:

$$x(t) = \sum_{n=0}^{N-1} \sum_{i=0}^{L-1} x(t_{ni}) \Psi_{ni}(t) \tag{4.3}$$

where

$$\Psi_{ni}(t) = \prod_{\substack{k=0 \\ k \neq i}}^{L-1} \left\{ \frac{\sin(2\pi f_c(t - t_{nk}))}{\sin(2\pi f_c(\tau_i - \tau_k))} \right\} \times \text{sinc}(2W(t - t_{ni})) \tag{4.4}$$

and

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. \tag{4.5}$$

All of this is only true provided that one is careful that:

$$\begin{aligned} &\sin(2\pi f_c(\tau_i - \tau_k)) \neq 0, \\ &i, k = 0, 1, 2, \ldots, L-1, \quad i \neq k. \end{aligned} \tag{4.6}$$

Thus, (4.3) is an interpolating function for periodic nonuniform samples much the same way that the Cardinal series is an interpolation function for uniform samples. The difference is in the assumptions that have been made about the spectrum. The Cardinal series assumes a lowpass spectrum and $h(t)$ assumes a multiband spectrum. Figure 4-7 summarizes the comparison between Shannon interpolation and burst sampling interpolation.

Thus the goal has been accomplished. A Cardinal series for multiband signals has been found. Starting with a set of periodic nonuniform data (defined by L and M) the points between are interpolated. The filter bank approach discussed in Chapter 5 turns the periodic nonuniform data into uniform data as well, but is less constrained in the multiband assumptions. However, the question of performance still remains.

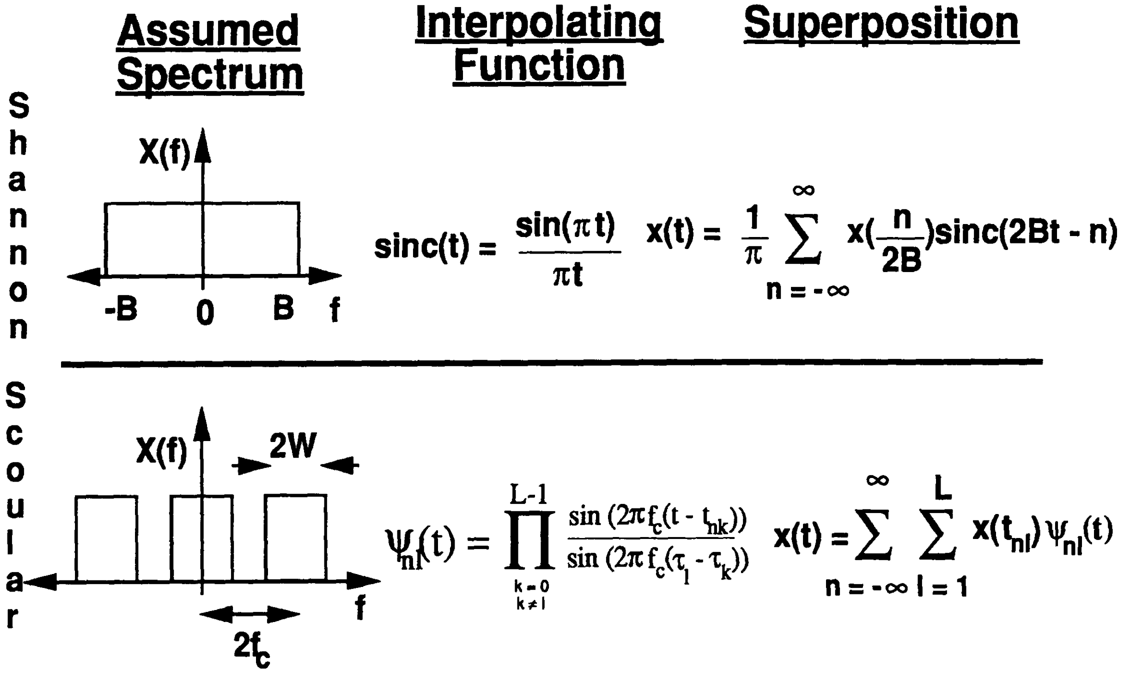
Assumed Spectrum
Interpolating Function
Superposition
Shannon
X(f)
-B
0
B
f
$\mathrm{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$
$x(t) = \frac{1}{\pi} \sum_{n=-\infty}^{\infty} x(\frac{n}{2B})\mathrm{sinc}(2Bt - n)$
Scoular
X(f)
2W
$2f_c$
f
$\psi_{nl}(t) = \prod_{\substack{k=0 \\ k \neq l}}^{L-1} \frac{\sin(2\pi f_c(t - t_{nk}))}{\sin(2\pi f_c(\tau_l - \tau_k))}$
$x(t) = \sum_{n=-\infty}^{\infty} \sum_{l=1}^{L} x(t_{nl})\psi_{nl}(t)$

Figure 4-7: Scoular's extension of the Shannon sampling theorem. The first column shows the spectral assumptions. The second column shows the interpolating function. The third column shows the estimated time domain signal as a weighted sum of interpolating functions.

4.4 Spectral Assumptions

The assumptions are the primary drawback for this method, for as we will see in Section 4.5, when the assumptions fit the true signal, the results of the interpolation method are pretty good. We assume that the true signal lies in two non-overlapping frequency bands, with the precession component in the lower frequency band and the spin component in the higher frequency band. The following are the spectral assumptions which the interpolation method requires:

1. There are L bands (L an integer).

2. Bands have equal bandwidth $2W = 1/M$.

3. Center frequencies of bands are spaced $2f_c$ apart.

4. f_c/W is an integer.

4.5 Simulation

We saw in Chapters 2 and 3 on optimal throw away and autocorrelation that those two methods performed reasonably well when the two sinusoids (precession and spin) had equal amplitudes. While this is fine for simulation, in the real world it is unreasonable. In this chapter and in Chapter 5 on filter banks, we shall see that the methods which use periodic nonuniform sampling are better suited for the 10dB difference in amplitudes.

The simulations begin as in the previous chapters. First, the case where the precession and spin components have equal amplitudes is examined, then the two components have unequal amplitudes (a 10dB difference between precession amplitude and spin amplitude), phase sensitivity is tested, and the method is tested under noisy real world conditions. Finally, it will be necessary to see what happens to the spectral estimate when the spectral assumptions have been violated. All cases have $L = 3$ and $M = 10$. As a clarification, note that L is defined in Figure 4-1 as the number of samples per burst and M is defined as the between-burst spacing. The

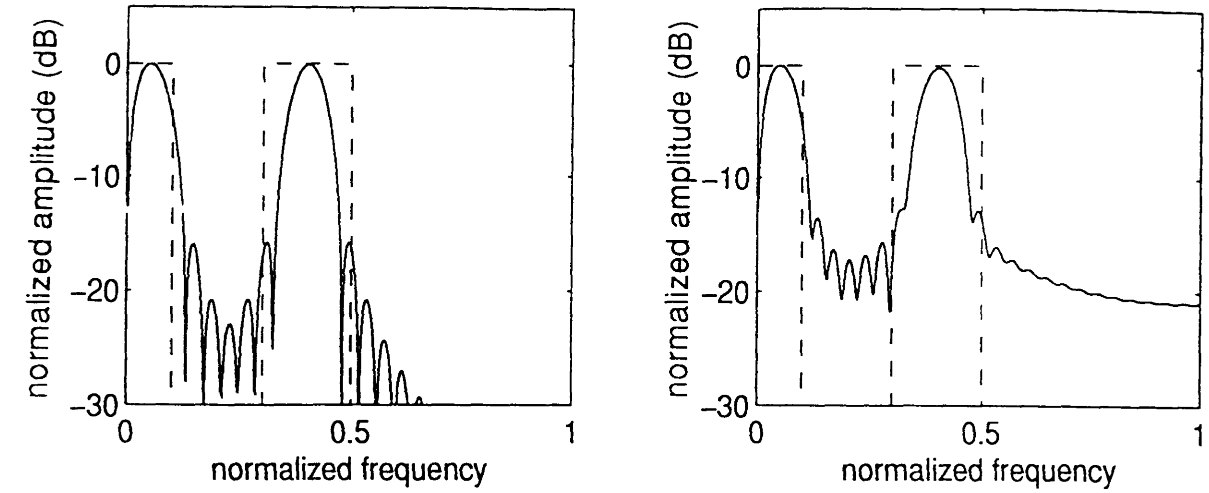

Figure 4-8: Interpolation method. On the left is the fully sampled spectrum and on the right is the spectrum after keeping 15 of 50 samples and using the interpolation method.

spectral assumptions of Sections 4.4 and 5.4 refer to the same L and M and therefore constrain the sample pattern as well.

Figure 4-8 shows a simulation with equal amplitudes. As before, only 15 of 50 samples are retained. On the left is the fully sampled signal, and on the right is the estimate using the interpolation method. The dashed boxes indicate where the spectral energy has been assumed to lie. However, the interpolation method will only be usable if it can resolve the smaller peak 10dB down. This is shown in Figure 4-9. Note that the peak is definitely visible, but it is a little biased in both frequency and amplitude. If the phases are changed arbitrarily, as shown in Figure 4-10, the bias goes away. Thus, there is a small phase-dependent bias characteristic of the interpolation method.

The next step is to test the performance of the interpolation method under noisy conditions. Figure 4-11 shows how the performance degrades as white gaussian noise is added to the signal. Each plot shows 10 runs with the *noise* variable as indicated. The *noise* variable is as defined in Section 1.5. It will be interesting to compare these results with the results using filter banks.

Finally, we would like to test what happens when the spectral assumptions are

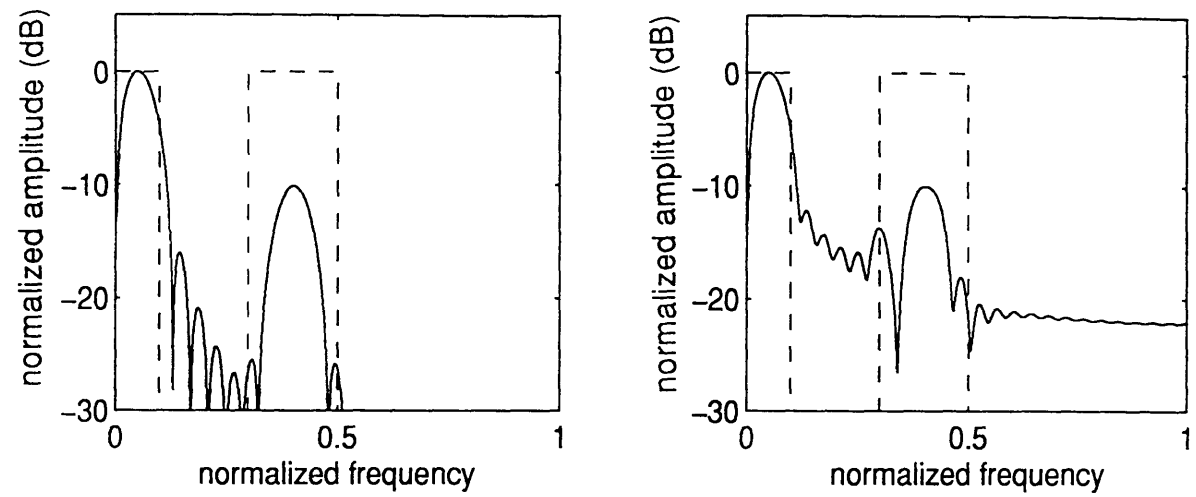

Figure 4-9: $A_1 = 10A_2$. On the left is the fully sampled spectrum and on the right is the spectrum after keeping 15 of 50 samples. Careful examination reveals a slight bias in both frequency and amplitude.

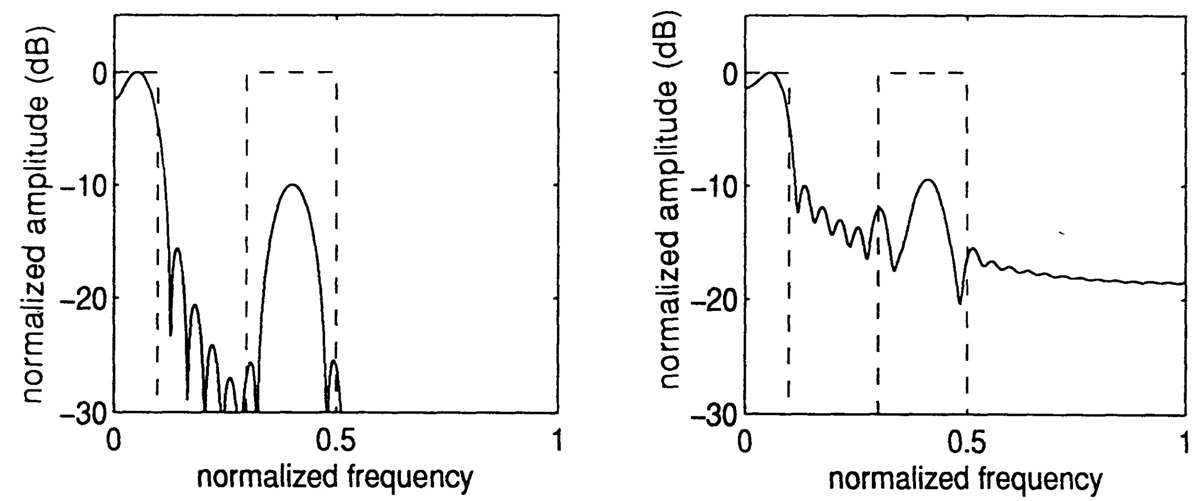

Figure 4-10: Phase sensitivity. The amplitude ratio is still 10 dB, but now there is an arbitrary phase change. Careful inspection reveals that the peak is no longer biased in frequency or amplitude. Thus, the interpolation method is slightly phase sensitive.

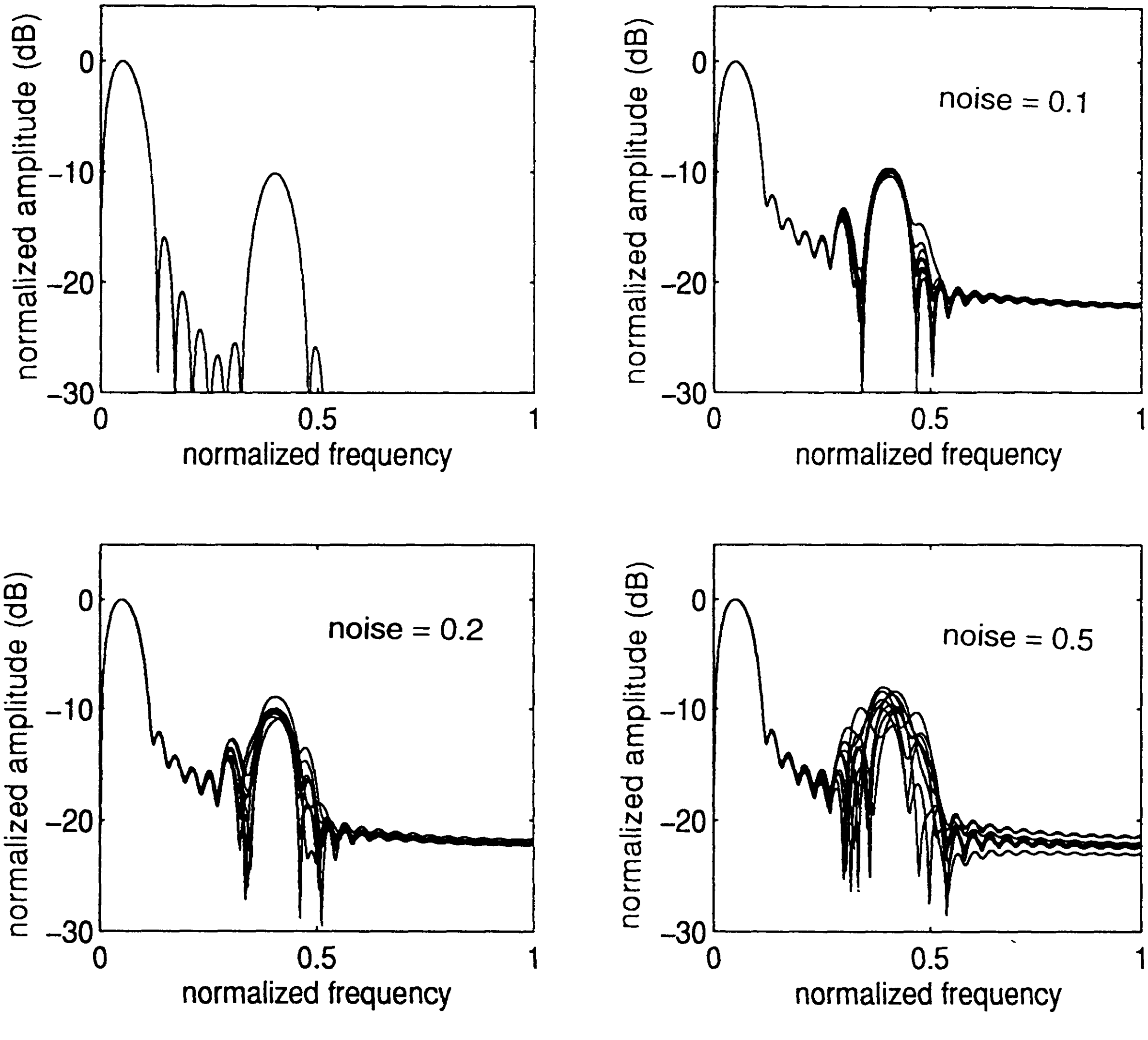

Figure 4-11: Noise sensitivity. This figure shows how the performance degrades as noise is added to the signal. Each plot shows 10 runs for a particular value of *noise*.

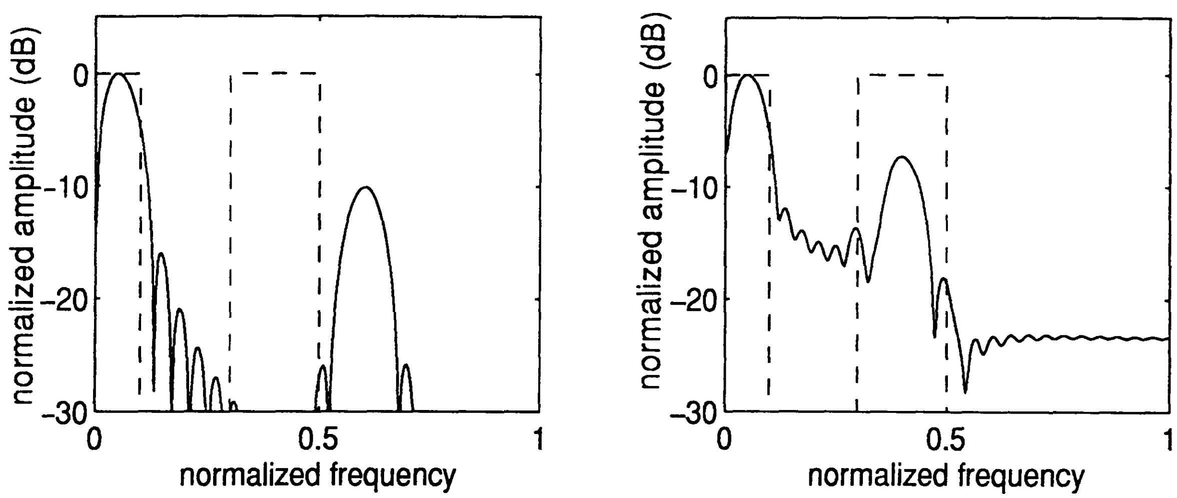

Figure 4-12: Wrong Assumptions. On the left is the true signal and the assumed spectral bands (dashed boxes). On the right is the estimated spectrum using the interpolation method. Note that the signal looks reasonable, but is in the wrong band.

wrong. This is shown in Figure 4-12. The dashed boxes indicate where the spectrum is assumed to lie, but as shown on the left of Figure 4-12 the signal is not actually in that band. When the interpolation method is used to estimate the signal, the spectrum looks reasonable, but is in the wrong place. Figure 4-13 shows how the performance degrades as the true spin component moves near the edge of the band where it has been assumed to lie. Note that the interpolation method performs quite well as long as the true spin component lies within about 5 percent of the edge of the assumption band. However, near and past the band edge, the estimate is wrong. This is very disconcerting because we thus cannot know from the results if we have made the wrong assumptions. This will be discussed more in Chapter 7.

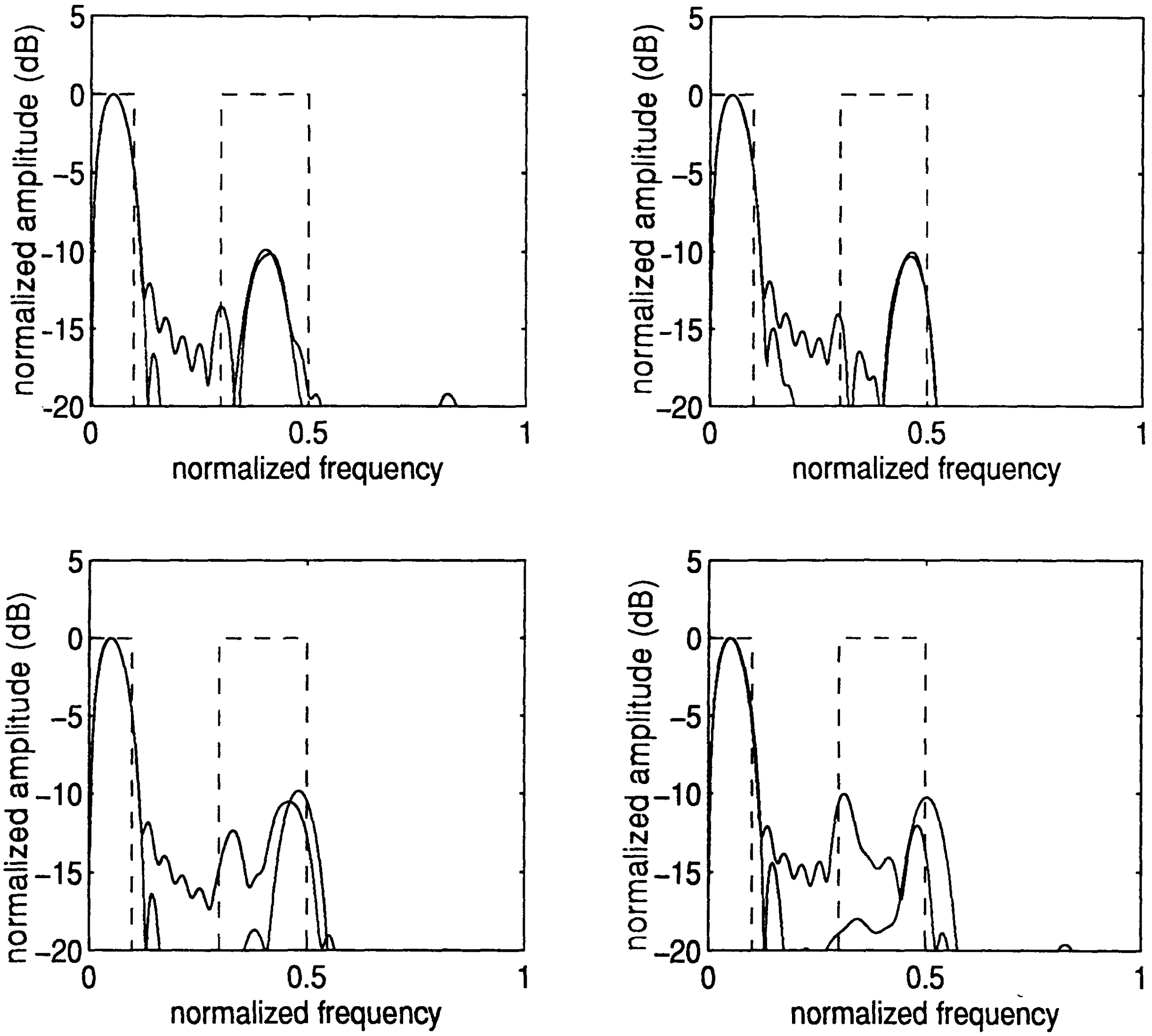

Figure 4-13: Nearing the band edge. This figure shows how the performance degrades as the true spin component begins to move out of the band where it is assumed to lie. The dashed boxes show the bandlimited assumptions about the spectrum. Note that the degredation in performance is not gradual. The spectral estimate is quite accurate as long as the true signal is within about 5 percent of the edge of the assumption band.

Chapter 5

Filter Bank Method

5.1 Introduction

As discussed in Chapter 4, one method for lowering the average sample rate is to use periodic nonuniform sampling (see Figure 4-1). Periodic nonuniform sampling is essentially sampling above the Nyquist rate and then periodically throwing away a series of adjacent samples. Assume a between-burst spacing of M samples, and L samples per burst. Chapter 1 mentioned that if the sampling is not exactly uniform sampling, this can be corrected for using simple linear interpolation. Also, if the sampling within a burst is not quite uniform, or the spacing between bursts is not exactly equal, techniques should be considered to compensate for this error if it becomes a significant factor in the radar resource scheduling. From [6] we find that keeping only L/M samples, and with the Nyquist frequency normalized to 2π, it must then be assumed that the frequency to be estimated is supported on a total bandwidth of $2\pi L/M$. If we assume that the precession frequency of any missile lies in a certain band, and also assume that the wobble frequency lies in another band, then periodic nonuniform sampling can be used as a technique for estimating the frequency (see Figure 5-1).

In the filter bank method, the original samples are left alone, and, as with the interpolation method, the points between bursts are estimated. The spectrum may then be easily estimated using the Fourier Transform. As with the interpolation

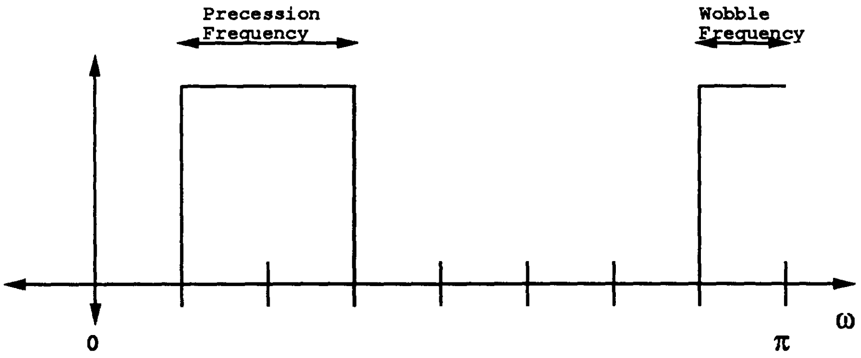

Figure 5-1: Typical frequency regions where wobble and precession may lie.

method, an advantage of using this filter bank approach is that now that we have a set of uniform samples, we can use any of the frequency estimation methods that have already been developed such as the FFT and the AR, MA, and ARMA modeling techniques.

5.2 Classic Filter Banks

5.2.1 Notation

Before explaining the methods that employ periodic nonuniform sampling in filter banks, some preliminary definitions are presented. The last ten or fifteen years have been filled with research into perfect reconstruction filter banks, and with this research has developed the notation and building blocks to be described in this section.

Let $h(n)$ be a discrete time sequence, with n indicating the time index. Then we denote the z-transform of $h(n)$ by:

$$H(z) = \sum_n h(n) z^{-n}, \quad h(n) \text{ real.} \tag{5.1}$$

The Discrete Fourier Transform (DFT) of $h(n)$ is the z-transform, $H(z)$, evaluated at $z = e^{j\omega}$. Mathematically, a delay element is represented by z^{-1}, so if $Y(z) = z^{-1}X(z)$ then using (5.1) in discrete time, we have $y(n) = x(n-1)$.

Matrices are denoted by upper case bold letters, and column vectors by lower case

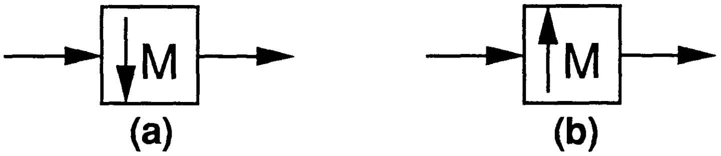

Figure 5-2: Building Blocks of Multirate Signal Processing, (a) the decimator and (b) the interpolator.

bold letters. The (i, j)th entry in a matrix $\mathbf{A}$ is denoted by $a_{i,j}$ and the kth entry in a column vector $\mathbf{a}$ is denoted $[\mathbf{a}]_k$. $\mathbf{A}^T$ is the transposition of the matrix $\mathbf{A}$, $\mathbf{A}^*$ is the conjugate, and $\mathbf{A}^\dagger$ is the conjugate transpose.

5.2.2 Filter Bank Building Blocks

The processing of discrete time signals having different sample rates is known as multirate signal processing [2]. A decimator changes the sample rate by a factor of $1/M$, and is shown in Figure 5-2a. The output $y(n)$ is related to the input $x(n)$ by:

$$y(n) = x(Mn). \tag{5.2}$$

Decimation can cause aliasing and is described in the frequency domain as:

$$Y(e^{j\omega}) = \frac{1}{M} \sum_{m=0}^{M-1} X(e^{j\omega/M} e^{-j2\pi m/M}) \tag{5.3}$$

To change the sample rate by a factor of M, an interpolator is used and shown in Figure 5-2b. The output $y(n)$ is related to the input $x(n)$ by:

$$y(n) = \begin{cases} x(n/M) & \text{when } n \text{ is divisible by } M \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

It is described in the frequency domain by:

$$Y(e^{j\omega}) = X(e^{j\omega M}). \tag{5.5}$$

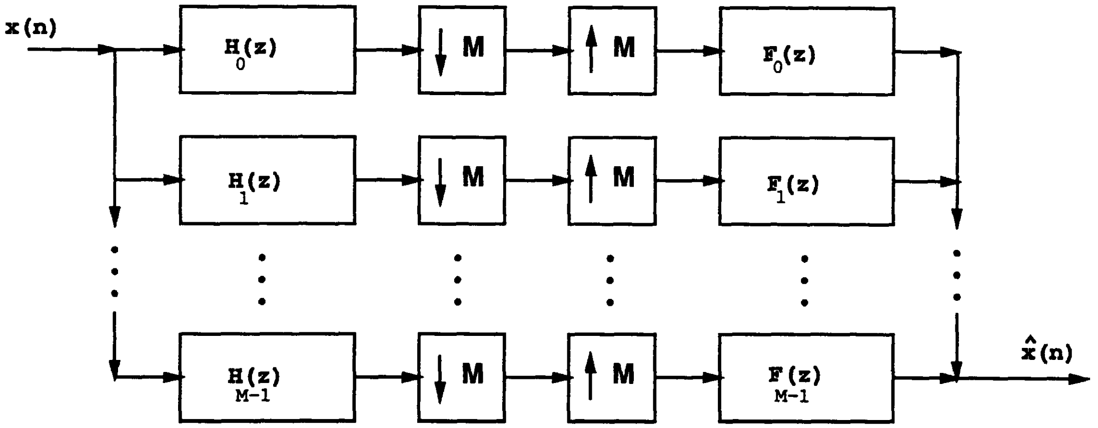

Figure 5-3: General Filter Bank

A typical filter bank setup is shown in Figure 5-3. Traditionally, the filters $H_k(z)$ are bandpass filters which break the spectrum into M adjacent bands. The filters cannot be perfect, and therefore there is crossover between bands. Significant research has been devoted to eliminating the aliasing that is caused in the reconstruction filters $F_k(z)$ due to the crossover. This is the study of perfect reconstruction filter banks. A perfect reconstruction system is defined as a system where the output is related to the input by a constant with a constant delay. In other words, if $y(n)$ is the output and $x(n)$ is the input, they are related by:

$$y(n) = cx(n-l) \tag{5.6}$$

where c and l are constants.

From the effort to design perfect reconstruction filter banks has evolved the polyphase representation. This is shown in Figure 5-4. Given any z-transform, $P(z)$, we may write it in terms of its polyphase components [1] as:

$$P(z) = \sum_{k=0}^{M-1} z^{-k} P_k(z^M). \tag{5.7}$$

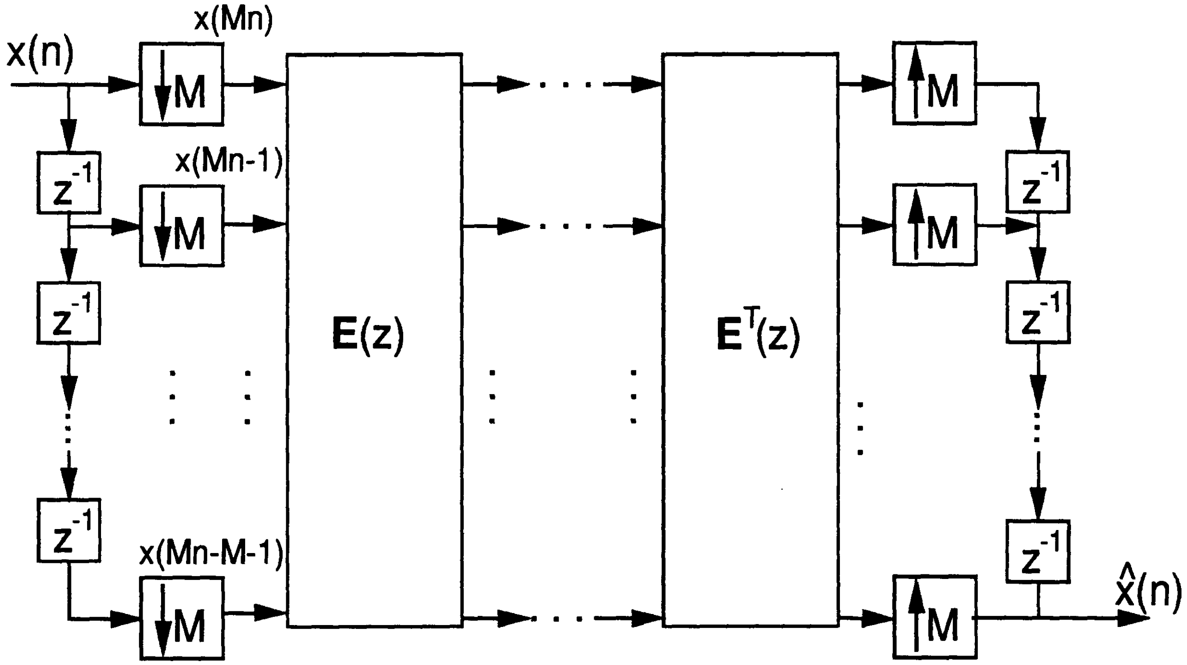

Figure 5-4: The Polyphase Representation of a Filter Bank.

Thus, in matrix form, we have the polyphase representation of the filters $H_k(z)$:

$$\mathbf{h}(z) = \mathbf{E}(z^M) \begin{pmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-M+1} \end{pmatrix} . \tag{5.8}$$

Using this, and moving the decimators to the left of the filters $H_k(z)$ and the interpolators to the right of the filters $F_k(z)$ in Figure 5-3 results in Figure 5-4.

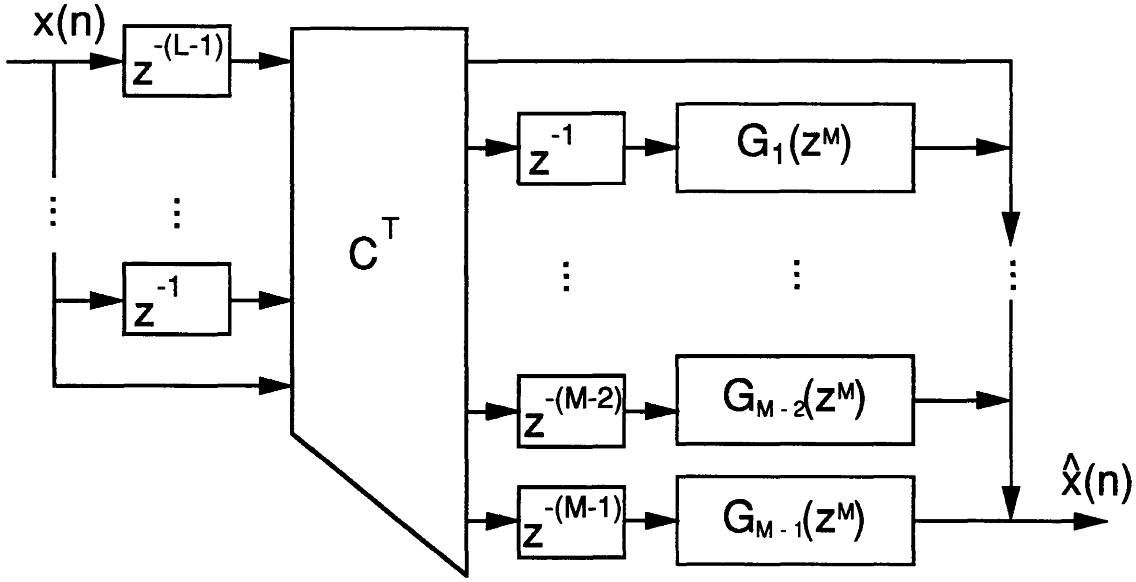

Figure 5-5: Liu's Method: Implementation of a filter bank to convert the nonuniform data to interpolated uniform data

5.3 Liu's Method

5.3.1 Introduction

Before going on, it should be explained why filter banks were even considered as a possible solution to the problem addressed in this thesis. A filter bank is typically a bank of M filters, adjacent to one another in the frequency domain. If we assume that the signal is zero in certain frequency bands, then from a conservation of information point of view, we should think some of the time domain data may be thrown away as well. This is exactly where the polyphase representation comes into play. It is interesting to note that in Figure 5-4 the inputs to the matrix $\mathbf{E}(z)$ are $x(Mn)$, $x(Mn-1)$, $x(Mn-2)$, ... $x(Mn-M+1)$. This is a natural place for periodic nonuniform sampling. If all the inputs except the first L are assumed to be zero, there is no need to sample the last $M-L$ inputs, and we have periodic nonuniform sampling, with L and M as described previously. The goal now is essentially to find an equivalent of the polyphase structure for this situation of periodic nonuniform sampling. This is shown in Figure5-5 [6].

It is first assumed that the data is of the form of a periodic nonuniform sequence. In [15] we see that that for the case of pulse burst sampling, there is guaranteed to be a solution. In pulse-burst (or bunched) sampling, the L samples in each period are consecutive. When the periodic set of samples are not consecutive, the matrices that are involved in the reconstruction may or may not be invertible.

The method proposed in [15], and shown in Figure 5-5, for reconstruction of uniform data from periodic nonuniform data was suggested as a possible way to compress data, with the compression algorithm being very simple. All that is required to compress the data is to throw away some samples. The expense paid, however is that the reconstruction method is proportionately more difficult. Periodic nonuniform sampling can be viewed as uniform sampling while throwing away some of the samples. Therefore, this method will be especially applicable to our problem.

Because some samples are thrown away, we should expect that we are throwing away part of the bandwidth. This assumption is in fact necessary for the method in [15]. If we keep only L of every M samples, then we must assume that the frequencies of our sampled signal lie in a total bandwidth of $2\pi L/M$. In general, this total bandwidth can be made up of multiple bands.

The derivation of the matrix C and the filters $G_k(z)$ is outlined in Section 5.3.2. The methods used to design the filters $G_k(z)$ are outlined in Appendix A. As will be shown in Section 5.3.2, the filters $G_k(z)$ are found by taking the M polyphase components of an M^{th} band prototype filter. This filter, which we will call $P(z)$, is lowpass with a cutoff frequency $\omega_c = 2\pi/M$. As is shown in [9], quality Mth band filters can be designed if they have a total bandwidth of $2\pi k/M$. Thus, we will need a good design technique for M^{th} band filters. This is described in Appendix A. A few techniques have been developed to design M^{th} band filters, and the Eigenfilter approach is used here.

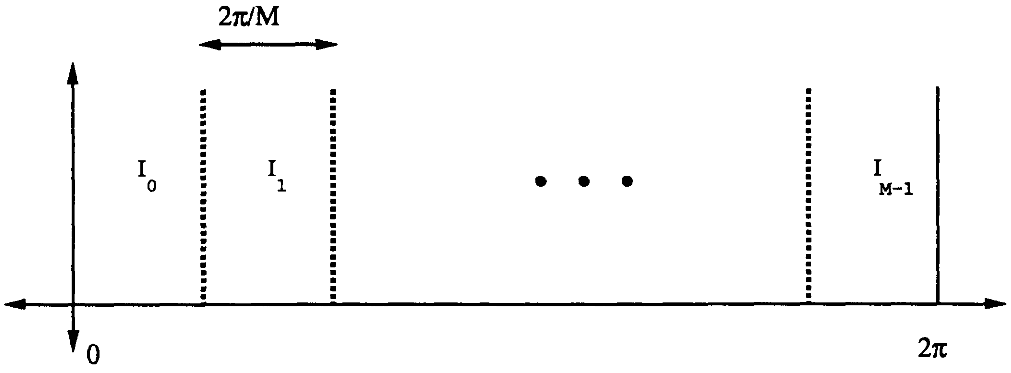

Figure 5-6: Dividing intervals, I_m

5.3.2 Computing the C matrix and filters $G_k(z)$

Introduction

The formulation of C and $G_k(z)$ shown in Figure 5-5 depend on many factors. It is necessary to know both the periodic sample times and the L frequency bands where the signal is assumed to be. The filters $G_k(z)$ are relatively simple to design and are related simply to the polyphase components of an M^{th} band filter. It also turns out that the filters are not dependent on L. Thus, once a set of filters have been designed for any M and L, it can be used for other M-periodic sample patterns.

Background and Motivation for C and $G_k(z)$

It has been stated earlier that since we are only sampling L of every M samples, we can only expect to cancel all aliasing if it is assumed that the signal which we are sampling lies in a total bandwidth of $2\pi L/M$. This total bandwidth need not be connected, but must lie on L frequency intervals of bandwidth $2\pi/M$. Figure 5-6 shows how these M intervals are to be divided from 0 to 2π. Let's define these M frequency intervals consecutively by

$$I_m = \omega \ \text{ for } \ \frac{2\pi}{M}m < \omega < \frac{2\pi}{M}(m+1) \tag{5.9}$$

For the case of the general filter bank as depicted in Figure 5-3, we have the following relationship:

$$\hat{X}(z) = \sum_{m=0}^{M-1} X(ze^{-j2\pi m/M})A_m(z) \tag{5.10}$$

where $A_m(z)$ represents the weighting function for each of the shifted copies of the original $X(z)$. This weighting function can be shown to be [14]

$$A_m(z) = \frac{1}{M}\sum_{k=0}^{L-1} e^{j2\pi mk/M} z^{-k} F_k(z) \tag{5.11}$$

Usually, in filter bank design, we require that aliasing is canceled assuming that all the information we want is in interval I_0. If this is the case, then alias cancellation is satisfied if

$$A_m(z) = 0 \ \text{ for } \ m \neq 0 \tag{5.12}$$

These aliasing cancellation conditions can also be expressed in matrix form as

$$\begin{pmatrix} A_0(z) \\ A_1(z) \\ \vdots \\ A_{M-1}(z) \end{pmatrix} = \frac{1}{M}\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{j2\pi/M} & \cdots & e^{j2\pi(L-1)/M} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{j2\pi(M-1)/M} & \cdots & e^{j2\pi(L-1)(M-1)/M} \end{pmatrix} \mathbf{\Lambda}(z)\mathbf{f}(z) = \begin{pmatrix} T(z) \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{5.13}$$

where $\mathbf{\Lambda}(z)$ is the diagonal matrix

$$\mathbf{\Lambda}(z) = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & z^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & z^{-(L-1)} \end{pmatrix} \tag{5.14}$$

$\mathbf{f}(z)$ is the set of L synthesis filters $F_k(z)$ to be solved for, and $T(z)$ is the overall transfer function of the filter bank.

However, if we make the assumption that the original signal, $X(e^{j\omega})$, is zero on certain intervals, as shown in Figure 5-7, then we can use a different set of constraints.

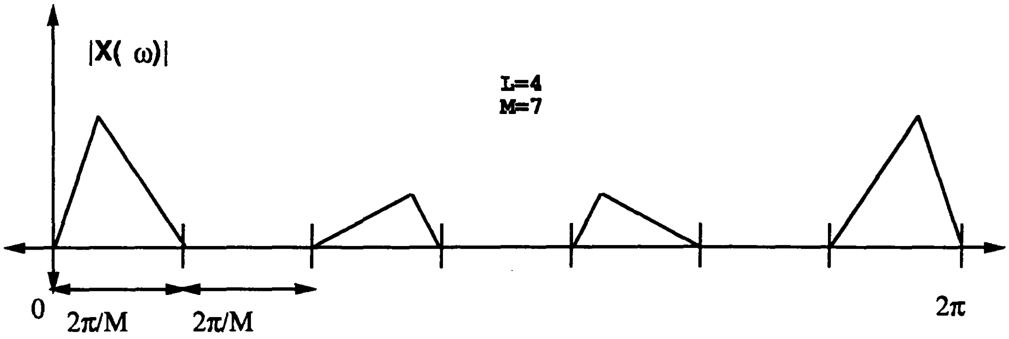

Figure 5-7: Example of the assumed spectrum for $L = 4$ and $M = 7$

We can impose a constraint similar to (5.12) on each of the nonzero frequency bands. Specifically, let's first label each of the M frequency intervals I_m in Figure 5-6 from 0 to $M - 1$ and let there be a set of L integers, l_k, representing those intervals I_m that are nonzero. As an example, say we have $M = 5$ and $L = 2$ and the spectrum is nonzero on only the first interval, 0, and last interval, 4, we would have $l_0 = 0$ and $l_1 = 4$. Using this notation, [6] leads us to the following result

$$\mathbf{\Lambda}(z)\mathbf{f}(z) = MT(e^{j\omega})\mathbf{\Lambda}(e^{j2\pi l_q/M})\mathbf{u}_q \tag{5.15}$$

where the vector $\mathbf{u}_q$ is the q^{th} column of a matrix $\mathbf{U}$ defined by its inverse $\mathbf{V}$ as

$$\mathbf{U}^{-1} = \mathbf{V} = \begin{pmatrix} 1 & e^{-j2\pi l_0/M} & \dots & e^{-j2\pi l_0(L-1)/M} \\ 1 & e^{-j2\pi l_1/M} & \dots & e^{-j2\pi l_1(L-1)/M} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & e^{-j2\pi l_{L-1}/M} & \dots & e^{-j2\pi l_{L-1}(L-1)/M} \end{pmatrix} \tag{5.16}$$

Of course, the whole idea here is to solve for the synthesis filters, $F_k(z)$, or just as well, $\mathbf{\Lambda}(z)\mathbf{f}(z)$, which is each of the filters $F_k(z)$ with a delay of z^{-k}. We can implement the delayed filters as the right hand side of (5.15)

Now, let's define the set of frequency intervals where the signal is assume to lie with the set of integers $\Gamma = \{l_0, l_1, \dots, l_{L-1}\}$. Because we have assumed that the signal is zero for $I_p \notin \Gamma$, we are free to require that the synthesis filters $F_k(z)$ are also zero

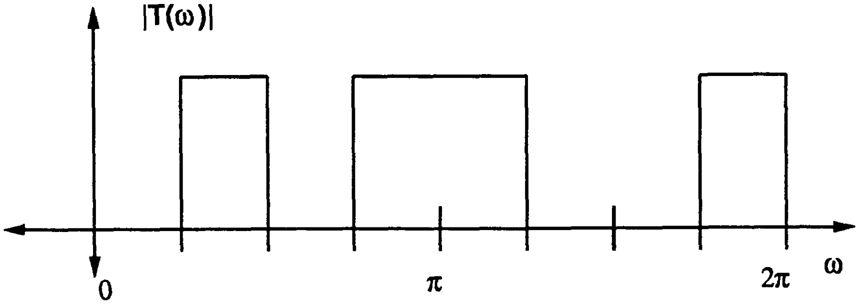

Figure 5-8: Example of $T(e^{j\omega})$ for $L = 4$ and $M = 8$

in these regions. Perfect reconstruction occurs when we have $T(z) = 1$ for ω in Γ. These two constraints can be represented by replacing (5.15) with

$$\mathbf{\Lambda}(z)\mathbf{f}(z) = M\mathbf{b}_p \tag{5.17}$$

where

$$\mathbf{b}_p = \begin{cases} \mathbf{\Lambda}(e^{j2\pi p/M})\mathbf{u}_q & p = l_q \in \Gamma \\ 0 & p \notin \Gamma \end{cases} \tag{5.18}$$

Thus, the overall transfer function has the following response

$$T(e^{j\omega}) = \begin{cases} 1 & \text{for } \omega \in I_p \text{ where } p \in \Gamma \\ 0 & \text{otherwise} \end{cases} \tag{5.19}$$

Figure 5-8 shows an example of $T(e^{j\omega})$ for $L = 4$ and $M = 8$. Note that this would not represent the spectrum of a real filter. To be real, $T(e^{j\omega})$ must be symmetric around π.

Now we have a set of filters, $\mathbf{f}(z)$ that are defined for each of the frequency regions I_p. Thus, each of the filters $z^{-k}F_k(z)$ is a multilevel filter of the form

$$e^{-j\omega k}F_k(e^{j\omega}) = M[\mathbf{b}_p]_k \text{ for } \omega \in I_p \tag{5.20}$$

where $[\mathbf{b}_p]_k$ represents the k^{th} row (or element since $\mathbf{b}_p$ is a vector) of $\mathbf{b}_p$, which is a complex constant.

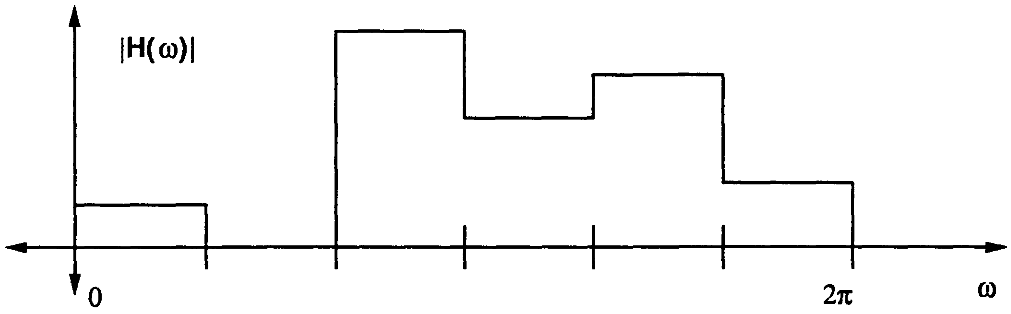

Figure 5-9: Example of a multilevel filter

Design of the filters $G_k(z)$

A general method to design multilevel filters is described in [6]. It is shown that *any* multilevel filter, as shown in Figure 5-9, can be expressed as

$$H(z) = \sum_{k=0}^{M-1} \alpha_k z^{-k} G_k(z^M) \tag{5.21}$$

where, if d_p is the complex amplitude of the multilevel filter for $\omega \in I_p$

$$\alpha_k = \frac{1}{M} e^{j\pi k/M} \sum_{p=0}^{M-1} e^{j2\pi kp/M} d_p \tag{5.22}$$

and $G_k(z^M)$ can be related to the polyphase components of an M^{th} band prototype filter $P(z)$ by

$$G_k(z^M) = \frac{M}{2\cos(k\pi/M)} P_k(z^M) \tag{5.23}$$

where $P_k(z^M)$ is the k^{th} polyphase component of $P(z)$. Finally, $P(z)$ is an M^{th} band real lowpass filter with cutoff at $2\pi/M$, which we can design using the Nyquist-Eigenfilter technique described in Appendix A. Note that if M is even, then we have for $k = M/2$ $\cos\pi/2 = 0$, and an infinite filter $G_k(z)$. We can avoid the problem completely, and use only odd M.

Deriving the matrix C

As was shown in (5.20), each of the filters $z^{-k}F_k(z)$ is a multilevel filter. Thus, from (5.21) each of these filters can be implemented as

$$z^{-i}F_i(z) = \sum_{k=0}^{M-1} c_{i,k} z^{-k} G_k(z^M) \tag{5.24}$$

Using (5.22) and (5.20), we find that the $c_{i,k}$ are given by

$$c_{i,k} = e^{j\pi k/M} \sum_{p=0}^{M-1} e^{j2\pi kp/M} [\mathbf{b}_p]_i \tag{5.25}$$

(5.24) can be represented in matrix form as

$$\begin{pmatrix} F_0(z) \\ z^{-1}F_1(z) \\ \vdots \\ z^{-(L-1)}F_{L-1}(z) \end{pmatrix} = \begin{pmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,M-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,M-1} \\ \vdots & \vdots & \cdots & \vdots \\ c_{L-1,0} & c_{L-1,1} & \cdots & c_{L-1,M-1} \end{pmatrix} \begin{pmatrix} 1 \\ z^{-1}G_1(z^M) \\ \vdots \\ z^{-(M-1)}G_{M-1}(z^M) \end{pmatrix} \tag{5.26}$$

In addition, if we let $\mathbf{B}$ be the matrix whose p^{th} column is $\mathbf{b}_p$ and the matrix of $c_{i,k}$'s in (5.26) be $\mathbf{C}$, then the two are related by

$$\mathbf{C} = \mathbf{B}\mathbf{W}^{\dagger} \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & e^{j\pi/M} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{j\pi(M-1)/M} \end{pmatrix} \tag{5.27}$$

where $\mathbf{W}$ is the DFT matrix having as it's $(i,k)th$ entry $e^{j2\pi ik/M}$. The overall synthesis filter bank $\mathbf{f}(z)$ can now be implemented as

$$\mathbf{f}(z) = \begin{pmatrix} F_0(z) \\ F_1(z) \\ \vdots \\ F_{L-1}(z) \end{pmatrix} = \begin{pmatrix} z^{-(L-1)} & 0 & \cdots & 0 \\ 0 & z^{-(L-2)} & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & 1 \end{pmatrix} \mathbf{C} \begin{pmatrix} 1 \\ z^{-1}G_1(z^M) \\ \vdots \\ z^{-(M-1)}G_{M-1}(z^M) \end{pmatrix} \tag{5.28}$$

The extra delays have been added to make the system causal. This implementation is depicted in Figure 5-5.

5.4 Spectral Assumptions

As is the case for the interpolation technique of Chapter 4 there are a set of spectral assumptions that go along with the method of this Chapter. These assumptions were described in Section 5.3.2 but will be reiterated here so that they may be more easily outlined and compared to those of the interpolation method. Again, these assuptions are made believing that the true signal has a precession component in a lower frequency band and a spin component in a higher frequency band. The assumptions for the filter bank method are as follows:

1. Divide the spectrum from 0 to 2π into M intervals.

2. Choose L intervals.

3. Real signal implies symmetry about π.

These requirements are certainly much simpler than those for the interpolation technique. This is one advantage of the filter bank method, although the most important item is how the performance compares to the other methods.

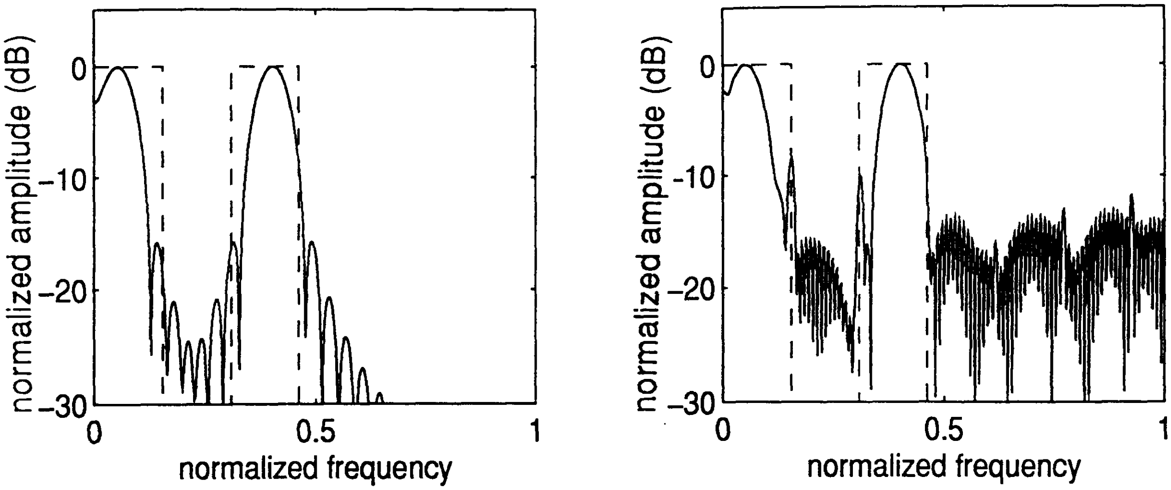

Figure 5-10: Filter Bank method. On the left is the fully sampled spectrum and on the right is the spectrum after keeping 16 of 52 samples and using the filter bank method.

5.5 Simulation

The examples in this section are exactly parallel to those in Section 4.5. First, the case of equal amplitudes is shown (again, keeping only about 15 of 50 samples). Next the case of the 10dB amplitude difference is shown, phase sensitivity is tested, and then noise is added to the simulated signal. Finally, the filter bank method is tested when the assumptions about the spectrum are wrong. All cases have $L = 4$ and $M = 13$. Again, L and M come from the spectral constraints and also define the sample pattern.

Figure 5-10 shows a simulation with equal amplitudes. As before, about 15 of 50 (16 of 52 in this example) samples are retained. On the left is the fully sampled signal spectrum, and on the right is the estimate using the filter bank method. The dashed boxes indicate where the spectral energy has been assumed to lie. Now, this method will only be useful if it can resolve the smaller peak 10dB down. This is shown in Figure 5-11. Note that the peak is definitely visible, and is not biased in either frequency or amplitude as it was with the interpolation method. If the phases are changed arbitrarily, as shown in Figure 5-12, the spectrum doesn't change. Thus, the filter banks method is not phase sensitive.

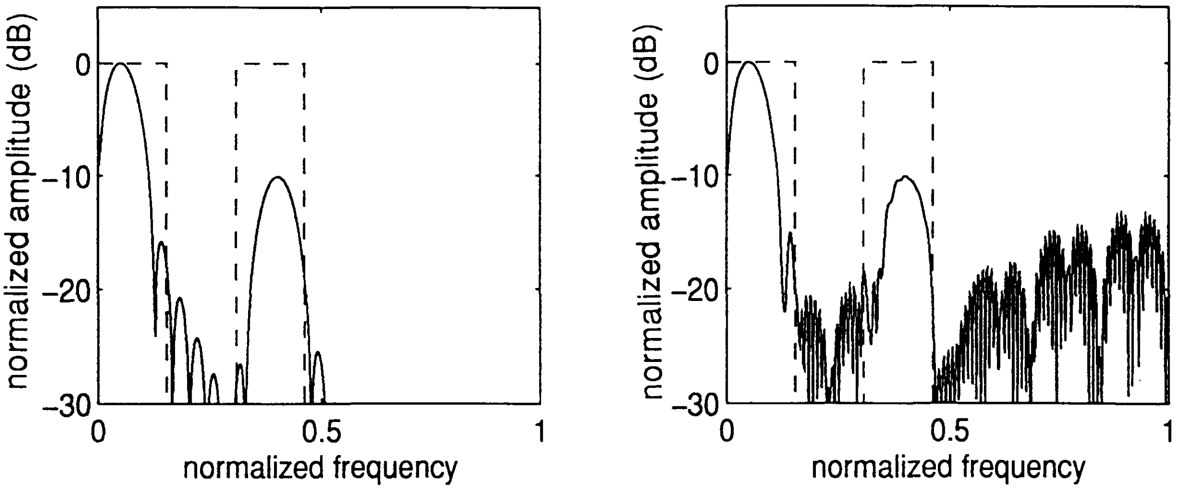

Figure 5-11: $A_1 = 10A_2$. On the left is the fully sampled spectrum and on the right is the spectrum after keeping 16 of 52 samples. Note that there is no bias in frequency or amplitude.

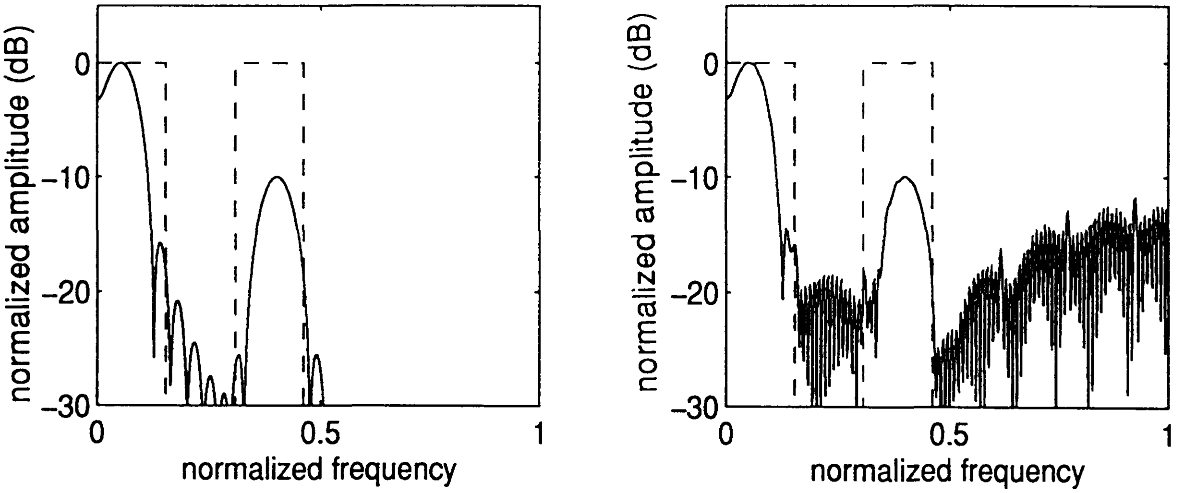

Figure 5-12: Phase sensitivity. The amplitude ratio is still 10 dB, but there is an arbitrary phase change. Note that the peak has not changed, and is therefore not biased. Thus, the filter bank method is not phase sensitive.

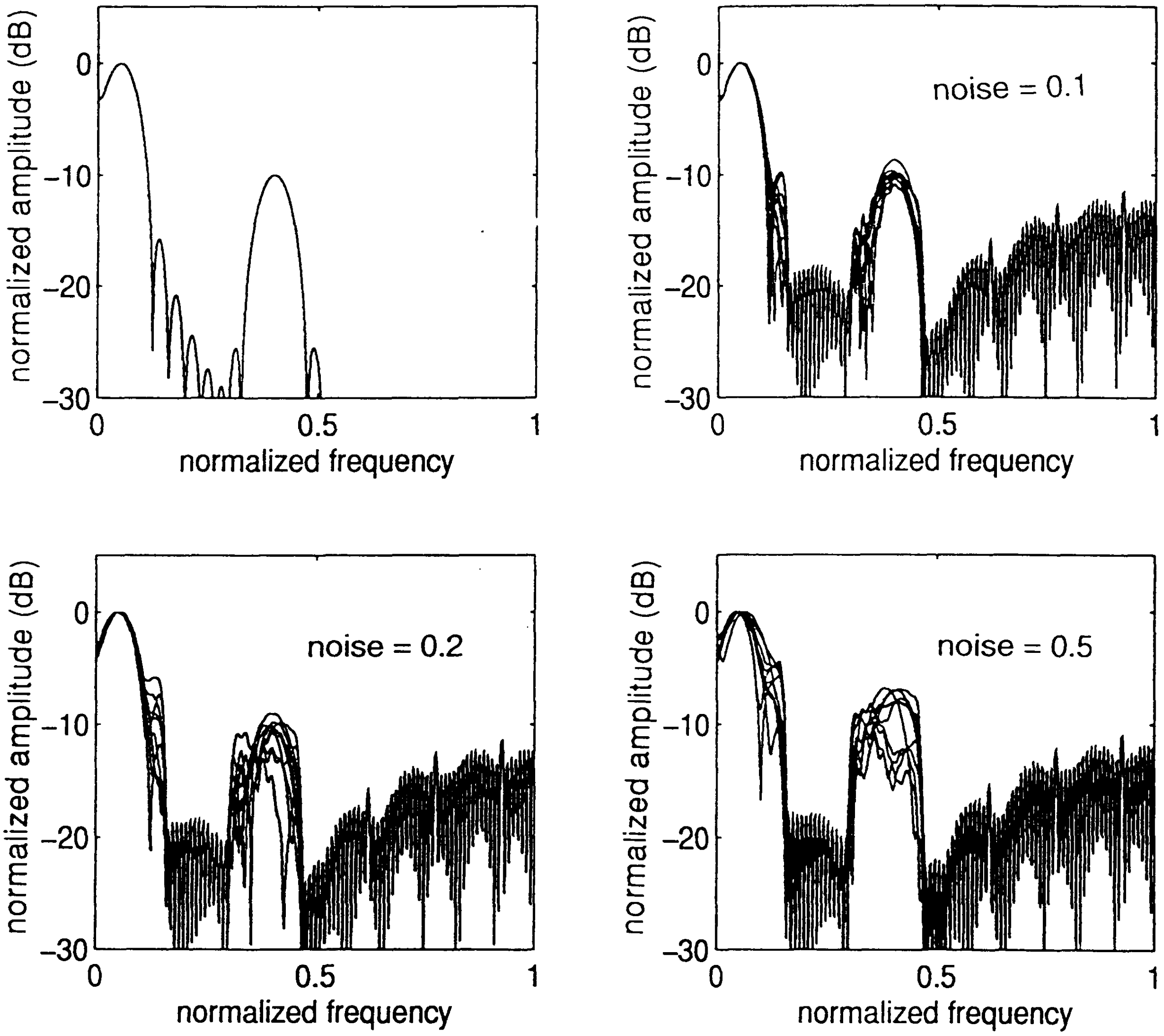

Figure 5-13: Noise sensitivity. This figure shows how the performance degrades as noise is added to the signal. Each plot shows 10 runs for a particular value of *noise*.

The next step is to test the performance of the filter bank method under noisy conditions. Figure 5-13 shows how the performance degrades as white gaussian noise is added to the signal. Each plot shows 10 runs with the *noise* variable as indicated. The *noise* variable is as defined as in Section 1.5. Comparing this plot with the one in Figure 4-11, it is clear that the variance between runs is higher for the filter bank method. Thus, under noisy conditions there is a trade off. The interpolation method can be biased depending on the phase, but with a lower variance between runs, while the filter bank method is not biased but has a higher variance between runs.

Finally, as with the interpolation method, it is necessary to test what happens

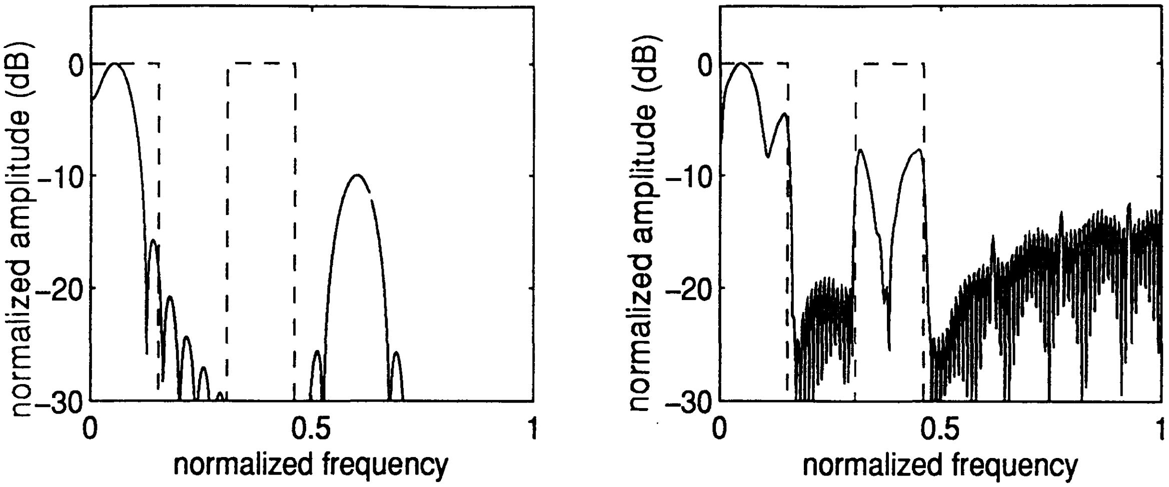

Figure 5-14: Wrong Assumptions. On the left is the true spectrum and the assumed spectral bands (dashed boxes). On the right is the estimated spectrum using the filter bank method. Note that the estimated spectrum lies in the wrong band.

when the spectral assumptions are wrong. This is shown in Figure 5-14. The dashed boxes indicate where the spectrum is assumed to lie, but as shown on the left of Figure 5-14 the signal is not actually in that band. However, when the filter bank method is used to estimate the signal, large errors occur.

Figure 5-15 shows how the performance degrades as the true spin component moves near the edge of the band where it has been assumed to lie. The true spin peak essentially is "wrapped around" into the assumption band. Thus, for this case, as was the case for the interpolation method, the filter banks method is quite accurate as long as the true signal lies within 5 percent of the assumption band edge. Again, this is very disconcerting because we thus cannot know from the results that the wrong assumptions have been made. This will be discussed more in Chapter 7.

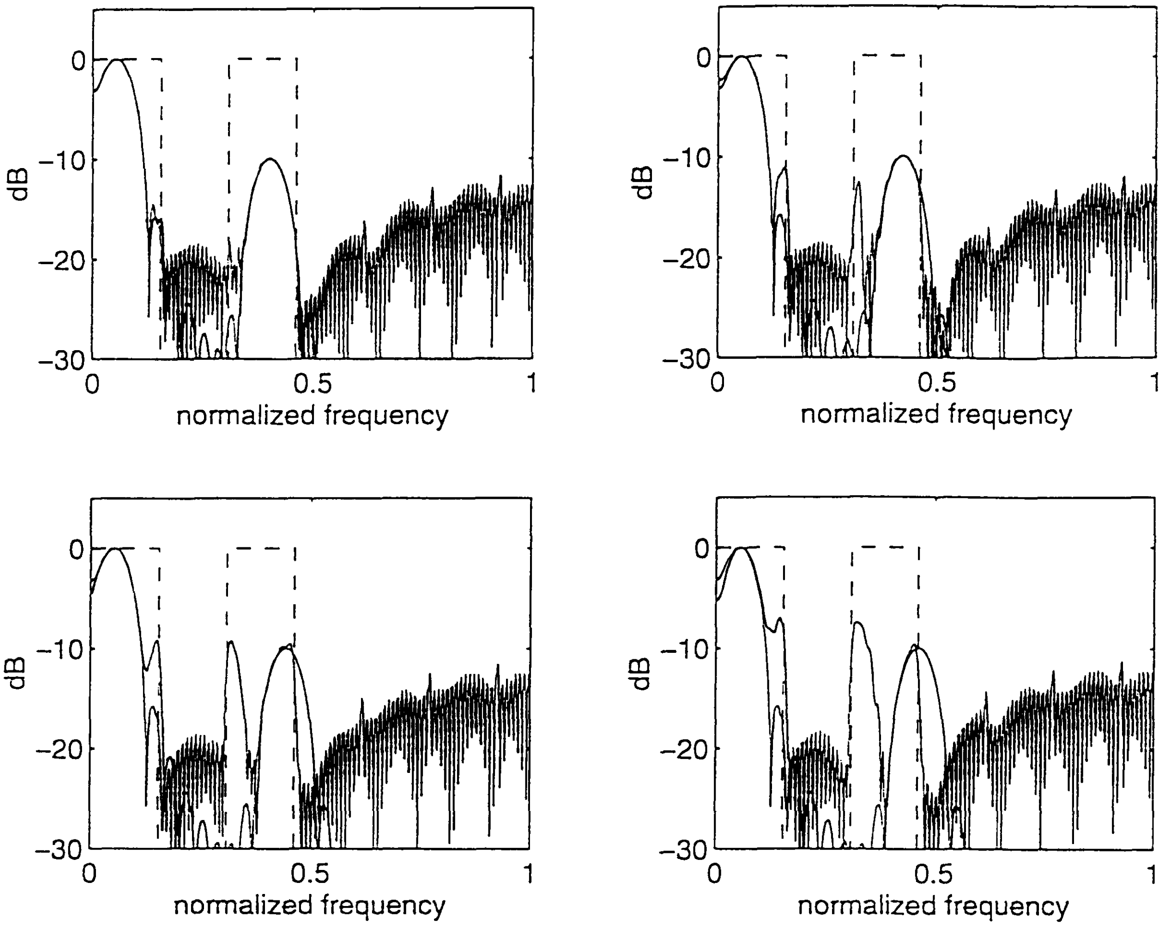

Figure 5-15: Nearing the band edge. This figure shows how the performance degrades as the true spin component begins to move out of the band where is is assumed to lie. The dashed boxes show the bandlimited assumptions about the spectrum. Note that as for the interpolation method, the degredation is not gradual. Note that the part of the true spectrum that falls outside the assumption band gets "wrapped around" back into the assumption band. As with the interpolation method, as long as the true signal is within about 5 percent of the band edge, the spectral estimate is quite accurate.

Chapter 6

Radar Data

The previous chapters dealt entirely with theory and simulation. Assuming that the return from the targets is actually in accordance with the simulation model, the methods which have been presented should perform quite well. However, if interpolation and filter bank methods are to be implemented, they must first be proven repeatedly on real radar data. This chapter includes a test of the two methods on one sequence of real radar data taken from a single target.

Figure 6-1 shows the original full data sequence of 143 samples and its spectrum after it has been preprocessed by passing it through a Hanning window. Note that both the precession frequency (about 0.05) and the spin frequency (about 0.4) are visible. Also note that in this case, the spin frequency is about 23 dB lower in amplitude than the precession. The simulation only assumed about a 10 dB amplitude ratio.

The first of the two methods to be tested was interpolation. Figure 6-2 shows the results when the interpolation method is used. Despite the fact that the spin amplitude was very small, it was detected using only about thirty percent of the original samples ($L = 3$; $M = 10$). The results here can be compared to the filter bank method in Figure 6-3. Using filter banks, and again only about thirty percent of the samples ($L = 4$; $M = 13$), the peak is quite well detected. Therefore both methods seem to do a pretty good job in this case.

Taking a closer look at the results, the interpolation method actually picked out

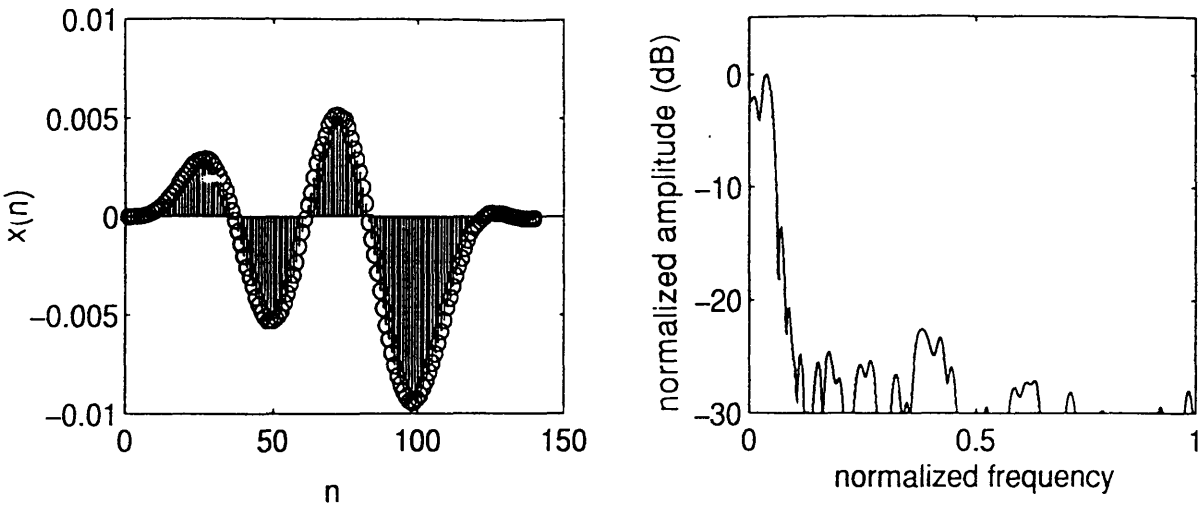

Figure 6-1: Original radar data fully sampled after passing it through a Hanning window (left). On the right is the spectrum. The two main peaks at approximately 0.05 and 0.4 are the precession and spin frequencies, respectively. Note that the spin amplitude is smaller than the precession amplitude by a much bigger factor than in the simulation examples. (approximately 23 dB)

the spin frequency peak better than the filter bank method. Note, however, that the shape of the spectrum near the spin frequency is better preserved using the filter banks method. Making a choice between the two methods may not be clear. It would depend upon what type of performance one is looking for. In any case, it is important to remember that this is only one example, and further testing would be required to make stronger conclusions about the two methods.

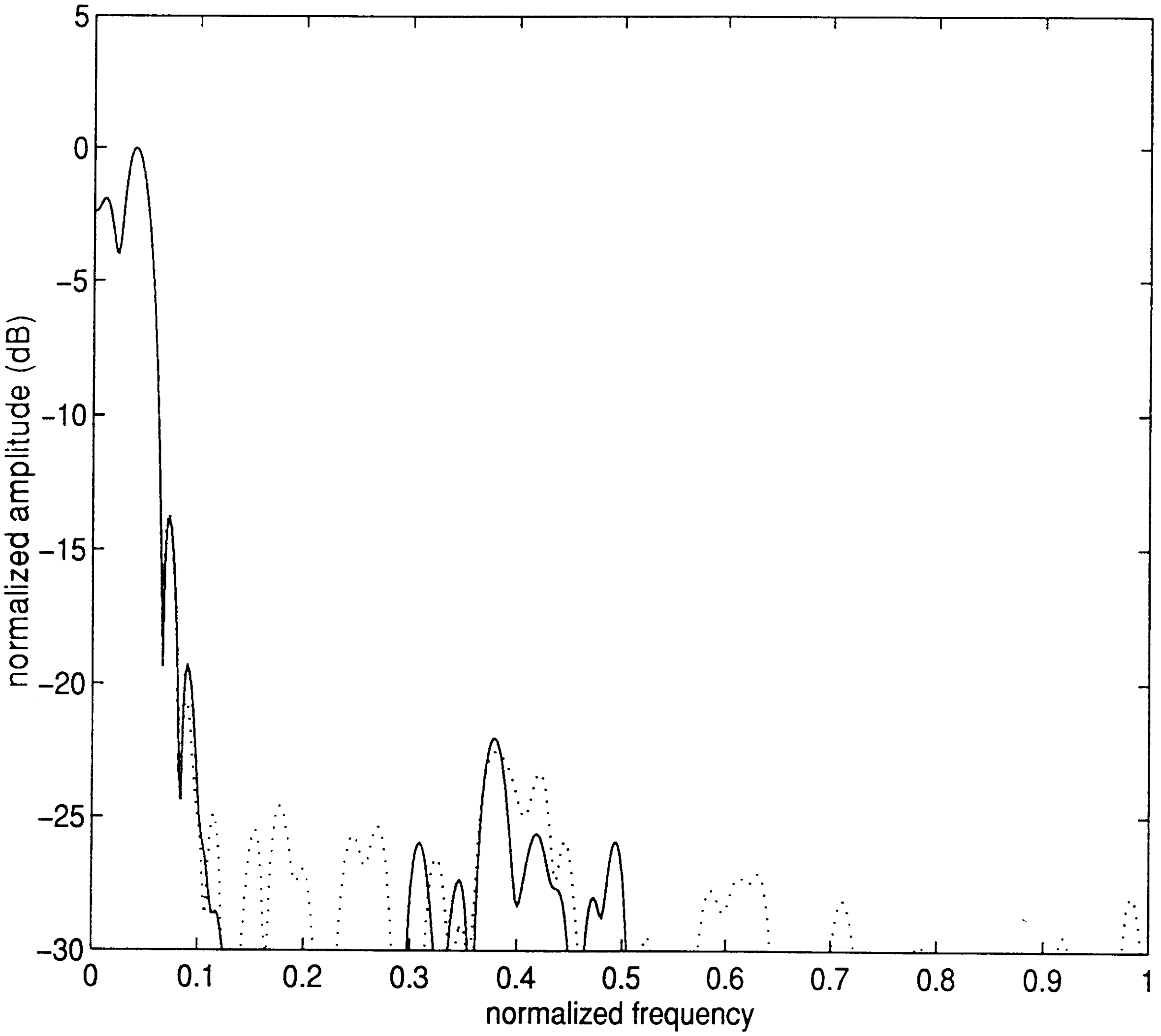

Figure 6-2: Interpolation method: Shown in the dashed line is the fully sampled spectrum, and the solid line is the spectrum using the interpolation method.

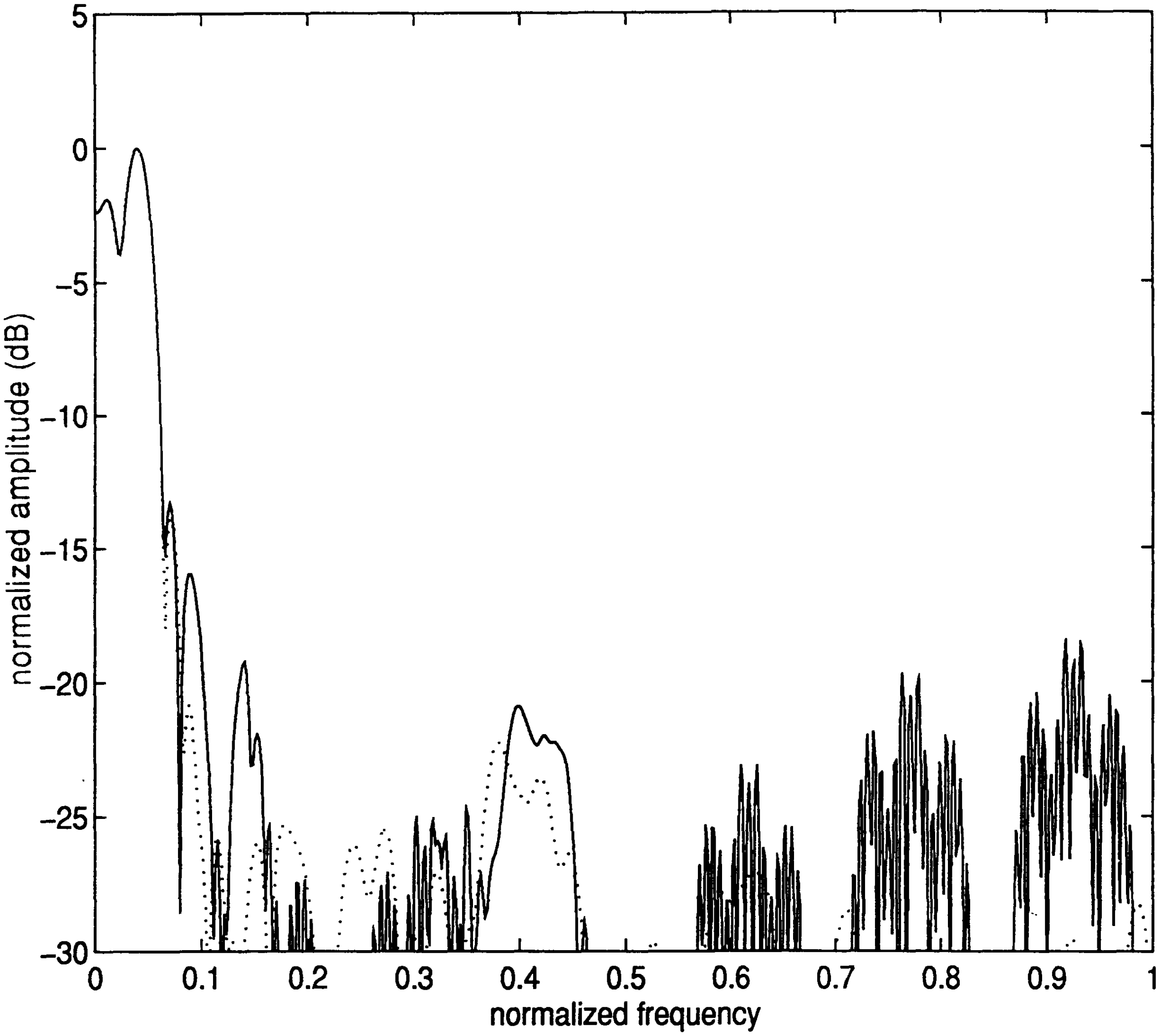

Figure 6-3: Filter bank method: Shown in the dashed line is the fully sampled spectrum, and the solid line is the spectrum using the filter bank method.

Chapter 7

The Problem of Spectral Assumptions

Of the four methods examined, the two that worked best depend on spectral assumptions. If the assumptions are correct, the interpolation and filter bank methods perform remarkably well. However, the spectrum still appears to be reasonable even when the assumptions are wrong. The problem is that there is really no way to know directly from the data that the wrong assumptions have been made. This case was made clear in Figures 4-12 and 5-14.

One method to avoid this problem is to take an extra sample for every data set. This is shown in Figure 7-1. Using the extra sample, some simple hypothesis testing can be performed with an ensemble of assumptions. Another method would be to compare the results of directly using the filter banks and interpolation methods with other estimation techniques. Group 34 at Lincoln Laboratory has used nonlinear processing algorithms to estimate the precession and spin. By comparing the results of multiple techniques, more solid conclusions about the spectrum may be made.

We'll use the "extra sample" technique as an example, and implement it with filter banks. Take the true spectrum shown on the right in Figure 7-2 with the true time sequence on the left. To keep it fairly simple, we'll make an ensemble of three hypotheses, shown by the dashed boxes on the left of Figure 7-3–7-5. On the right are the time sequences estimated using the filter banks method under each hypothesis.

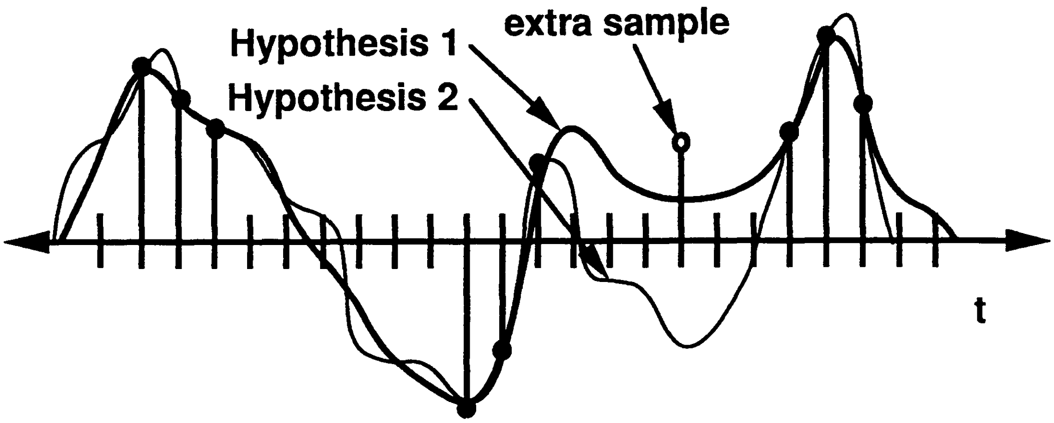

Figure 7-1: Taking extra samples. This figure shows how taking an extra sample may help resolve the errors that result from making the wrong assumptions about the signal.

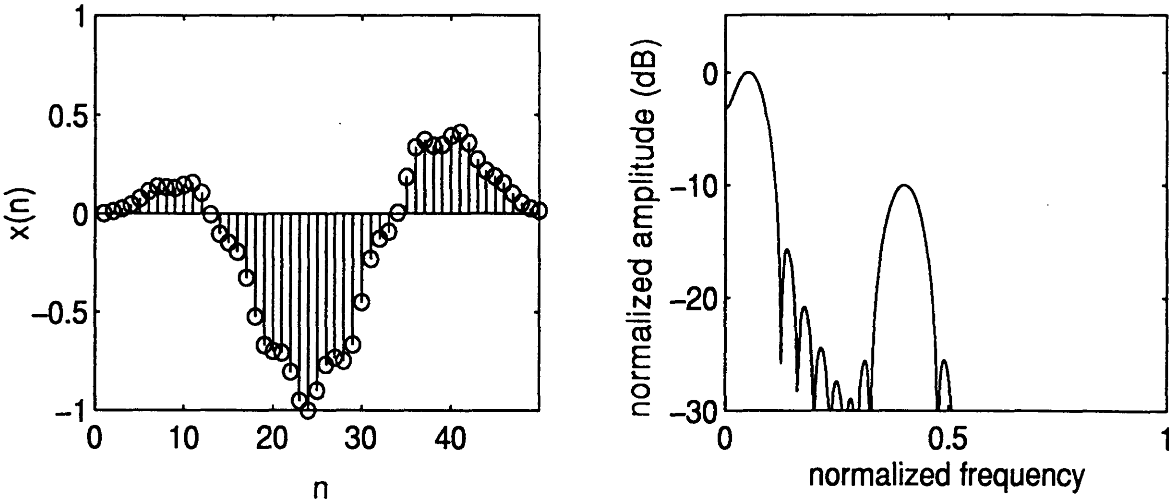

Figure 7-2: True spectrum and time sequence. On the left is the original time sequence fully sampled and the original spectrum is on the right.

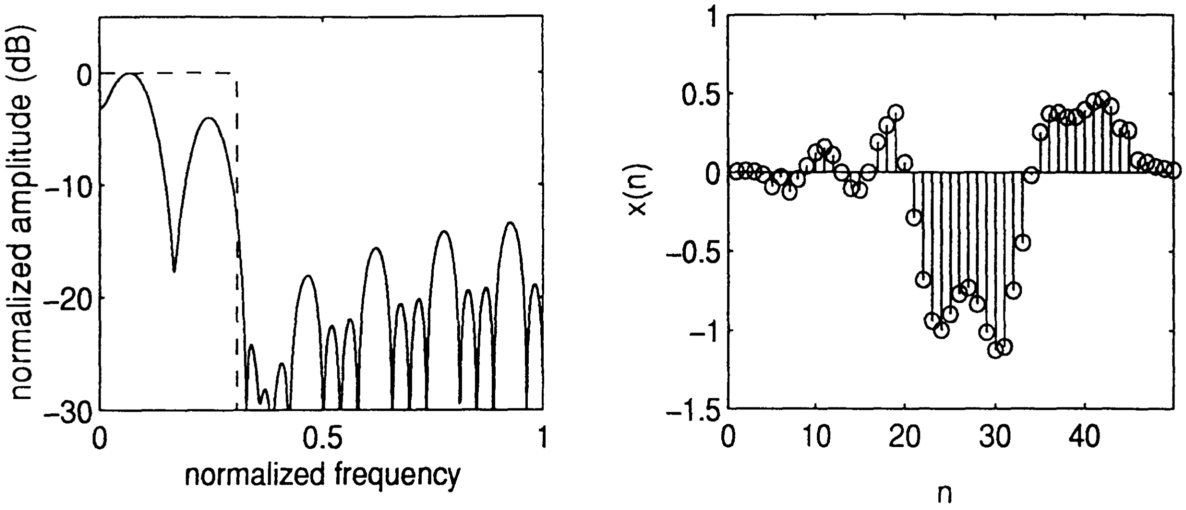

Figure 7-3: Hypothesis 1. On the left is the estimated spectrum under Hypothesis 1 (shown by dashed boxes), with the corresponding estimated time sequence on the right.

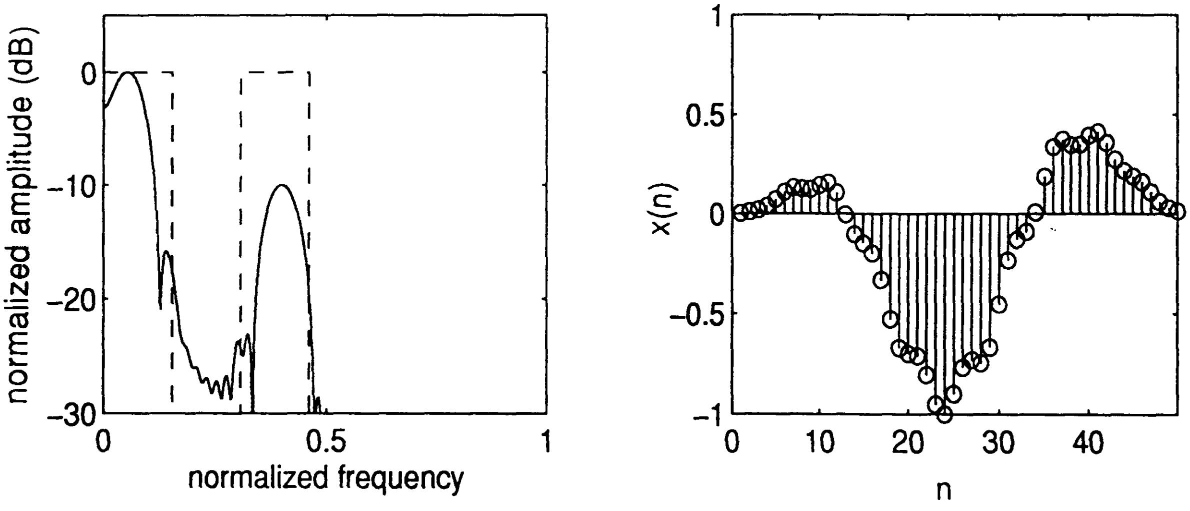

Figure 7-4: Hypothesis 2. On the left is the estimated spectrum under Hypothesis 2 (shown by dashed boxes - and what we know to be the correct hypothesis), with the corresponding estimated time sequence on the right.

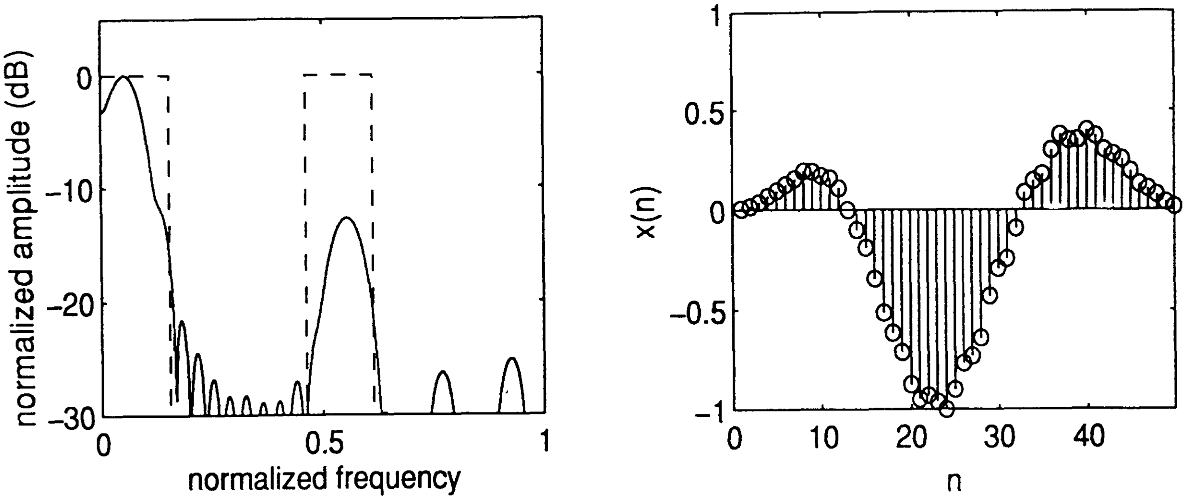

Figure 7-5: Hypothesis 3. On the left is the estimated spectrum under Hypothesis 3 (shown by dashed boxes), with the corresponding estimated time sequence on the right.

All these time sequences have the original bursts of samples. For example, note that the group of four samples (23–26) appear in all of the time sequences ($L = 4$; $M = 13$). Now, let's pick a data point which was not one of the burst samples (for example, sample 20 in Figure 7-2–7-5). Next we take the difference between sample 20 for each case minus the original sample 20. The bigger the difference, the less likely the hypothesis is correct. The values of sample 20 are as follows:

Hypothesis	Sample 20	Difference
Truth	−0.6977	0.0000
Hyp.1	0.0587	0.7564
Hyp.2	−0.7095	0.0117
Hyp.3	−0.8741	0.1763

Hypothesis 2 clearly has the smallest difference. Thus, it would be concluded that it is the correct assumption, and that hypotheses 1 and 3 are incorrect. This is exactly right, as we know that the true spectrum has a peak at 0.4. Under less ideal conditions, random noise could also cause this difference, so taking more than just one extra sample would be a more robust test.

Chapter 8

Summary

This thesis has examined four possible approaches to nonuniform sampling. There were a set of three criteria that were contained in the signal model. Any one of the methods had to perform well under all conditions to be considered a candidate for implementation. The first criterion was phase sensitivity. In the situation of a radar tracking a target, the phase cannot be known apriori, and therefore the algorithms must perform well independent of phase. Secondly, there is the 10dB amplitude ratio of the precession component to the spin component. This is a difficult constraint. In typical examples of spectral estimation methods, we see algorithms that perform remarkably well if the ratio of the two peaks are equal. However, it is a more difficult problem to detect a peak which is 10dB or more below the higher peak. Finally, there is the criterion that the method perform well under noisy conditions.

The first two methods (optimal throw away and autocorrelation) deal with designing a good sample pattern. A good sample pattern means that the pattern produces a spectrum with clearly visible peaks using the DFT. Both these methods performed quite well under certain conditions. The "optimal throw away" method produced two clear peaks keeping only 15 of 50 samples. However, the performance was strongly dependent on the phase of the signal. This fails the test of phase independence, and thus was eliminated early as a contender for implementation.

The second method which dealt with designing a good sample pattern was the autocorrelation or "all-spacings" method. The idea here was to design a sample pattern that may not have been uniform, but that instead had an autocorrelation function that was sampled as uniformly as possible. The phase of the test signal did not considerably alter the performance of the all-spacings method, which passed the phase-independence test. However, it failed to pass the second test. With a spin component 10dB smaller in amplitude than the precession component, there was no way to resolve the smaller peak. Thus, both of the methods which designed sample patterns performed well only if the conditions were right for them. Unfortunately, these conditions were not the signal model in this thesis.

It seems that we may be running into a more fundamental issue here than just that we happened by chance to pick the wrong methods in the first two cases. After all, if data is being eliminated in the time domain, the effect of this will inevitably show up in the frequency domain. The consequence of keeping only 15 of 50 samples was that the noise floor of the spectrum raised up to where the peak 10dB down was no longer visible. To keep this from happening, we need to make some assumptions. In a sense, as we throw away data, we need to assume that we are throwing away data that we do not need. This is a sort of conservation of information issue. With less data, there is less information and therefore a smaller bandwidth.

This leads to the second two methods. Assumptions about the spectrum are made, and the noise floor is lowered. Both of these methods rely on periodic nonuniform sampling. However, a price is paid by making these assumptions. What if the assumptions are wrong? Techniques to avoid this problem can be developed, with one example given in the Chapter 7.

The first method is the interpolation method. The idea is exactly as the name implies. The missing samples are interpolated in the time domain using functions that make some assumptions about the spectrum. This method had a small phase dependent bias, but it was tolerable. The interpolation method also performs well when the amplitude ratio is 10dB, and is able to pick out the peak of the spin frequency quite accurately, even in the radar data.

The last method, and the most difficult to implement, was the filter bank method. Just as with the interpolation method, assumptions were made about the spectrum, but they are less restrictive than those for the interpolation method. Making these assumptions allowed the smaller peak to be resolved. Comparing it to the interpolation method, in simulation the filter bank method was not at all phase sensitive, but was more sensitive to a noisy signal. When used on the real radar data (which had an amplitude ratio of about 23dB), the filter bank method preserved the shape of the bandlimited signal better than the interpolation method. However, the peak at the spin frequency did not have the correct amplitude. More testing is needed to see if these are basic characteristics of the methods.

In conclusion, the interpolation and filter bank methods proved to far outperform the optimal throw away and autocorrelation methods when the amplitude ratios were 10dB. Both the interpolation and filter banks methods had quite comparable performance, and further testing on real radar data is necessary to reach any solid conclusions on which method is better. However, both these methods make assumptions about the spectrum and if these assumptions are wrong, the results become totally unreliable. However, as was shown in Chapter 7, this problem may be alleviated by taking extra samples.

Appendix A

Filter Design

A.1 Eigenfilters

The eigenfilter approach is based on the minimization of an objective function $\Phi(\omega)$. Minimization of the objective function minimizes the squared error between the actual filter and the desired filter, where the desired is:

$$H_{id}(e^{j\omega}) = \begin{cases} 1 & \omega \text{ in passband} \\ 0 & \omega \text{ in stopband} \end{cases} \tag{A.1}$$

Weighting can be applied to the passband and stopband errors. The total error can be expressed in matrix form, as the energy of a symmetric positive definite matrix. The eigenvector corresponding to the minimum eigenvalue is the vector that minimizes the energy of this matrix. It is assumed for simplicity that the filters are of odd length. Fig. A-1 shows how the designed filter tries to approximate the ideal filter in a bandpass example.

The total error, or the objective function, can be described as the weighted sum of the passband and stopband errors.

$$\Phi(\omega) = (1-\alpha)\Phi_p(\omega) + \alpha\Phi_s(\omega)$$

Where the passband error, $\Phi_p(\omega)$, can be expressed in terms of the ideal filter,

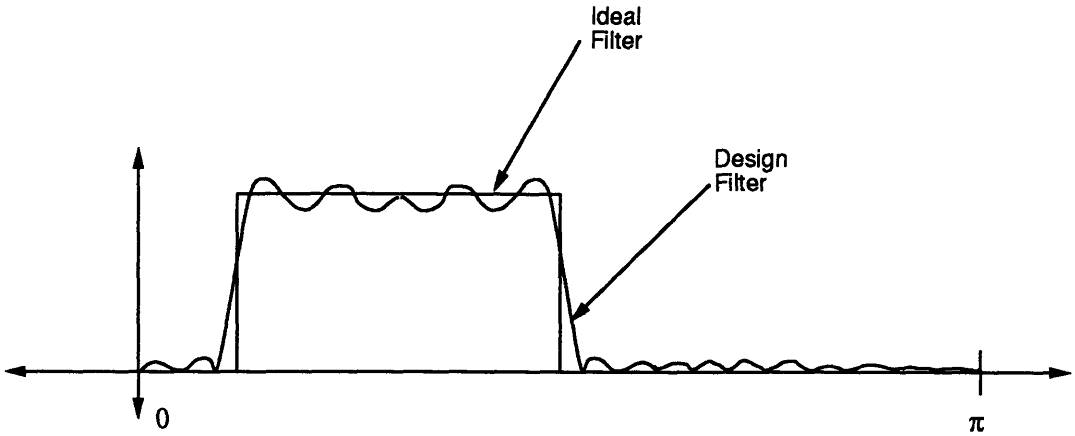

Figure A-1: Comparison of the filter to be designed and the ideal

$H_{id}(e^{j\omega})$, and the filter to be designed, $H(e^{j\omega})$, as

$$\Phi_p(\omega) = \frac{1}{\pi}\int_{\Psi} |H_{id}(e^{j\omega}) - H(e^{j\omega})|^2 d\omega \tag{A.2}$$

where Ψ is the passband region, and the stopband error $\Phi_s(\omega)$ is similarly expressed as:

$$\Phi_s(\omega) = \frac{1}{\pi}\int_{\Sigma} |H_{id}(e^{j\omega}) - H(e^{j\omega})|^2 d\omega \tag{A.3}$$

where Σ is the stopband region.

Now, assuming that the filter is causal linear phase, and the total length, N, is odd, then we have the discrete time Fourier transform (DTFT) relation

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h_n e^{-j(n-M)\omega} = \sum_{n=0}^{M} b_n \cos n\omega \tag{A.4}$$

where $M = (N-1)/2$ and

$$b_n = \begin{cases} 2h_{M-n} & n \neq 0 \\ h_M & n = 0 \end{cases} \tag{A.5}$$

Now, (A.4) can be expressed in matrix form as:

$$H(e^{j\omega}) = \mathbf{b^t c}(\omega) \tag{A.6}$$

where

$$\begin{aligned} \mathbf{b} &= [b_0\, b_1\, \cdots\, b_M]^t \\ \mathbf{c}(\omega) &= [1\ \cos\omega\ \cos 2\omega\ \cdots\ \cos M\omega]^t \end{aligned} \tag{A.7}$$

in the stopband, $H_{id}(e^{j\omega}) = 0$, thus

$$\Phi_s = \frac{1}{\pi}\int_{\Sigma} \mathbf{b}^t\mathbf{c}\mathbf{c}^t\mathbf{b}\,d\omega = \mathbf{b}^t\mathbf{P}_s\mathbf{b} \tag{A.8}$$

where

$$\mathbf{P}_s = \frac{1}{\pi}\int_{\Sigma} \mathbf{c}\mathbf{c}^t d\omega \tag{A.9}$$

and $\mathbf{P}_s$ is a real, symmetric, positive definite matrix.

Similarly, since $H_{id}(e^{j\omega}) = 1$ in the passband, the error, Φ_p can be expressed as

$$\Phi_p = \mathbf{b}^t\mathbf{P}_p\mathbf{b} \tag{A.10}$$

where

$$\mathbf{P}_p = \frac{1}{\pi}\int_{\Psi} (\mathbf{1} - \mathbf{c}(\omega))(\mathbf{1} - \mathbf{c}(\omega))^t d\omega \tag{A.11}$$

and $\mathbf{1}$ is the vector $[1, 1, \cdots, 1]^t$. Thus, the total (weighted) error can be expressed as

$$\Phi = (1-\alpha)\Phi_p + \alpha\Phi_s = \mathbf{b}^t\mathbf{P}\mathbf{b} \tag{A.12}$$

where

$$\mathbf{P} = (1-\alpha)\mathbf{P}_p + \alpha\mathbf{P}_s \quad \text{with} \quad 0 \le \alpha \le 1$$

The variable α is used to weight the errors in the passband and the stopband, allowing the designer to choose where he wants most of the total error to be.

Thus, (A.12) describes the total error in quadratic form. Using Raleigh's principle [8], we find that the vector $\mathbf{b}$ which minimizes the total error, Φ, is the eigenvector corresponding to the minimum eigenvalue of the matrix **P**. Once we have the **b** vector, we can immediately write down h_n, which is the impulse response of the eigenfilter. Fig. A-2 compares equal length filters that have been produced by both the widely used Parks-McClellan algorithm and the Eigenfilter method. The equiripple of the Parks-McClellan filter is apparent, while you can see that the Eigenfilter falls off as the frequency increases.

A.2 Finding the Eigenvector

In [16], the method used to find the eigenvalue and eigenvector suggested is the power method. The reason for this choice was to take advantage of the fact that we are only looking for the minimum eigenvalue, and a complete factorization of the matrix is unnecessary. However, it was found that for this thesis, the computation time was not intolerable if the entire matrix was factorized using a singular value decomposition (SVD). Because the matrix is a symmetric, positive definite matrix, a SVD is equivalent to an eigendecompostion of the matrix [13]. The result of the SVD on any symmetric positive definite matrix is to decompose the matrix **Q** into the following form:

$$\mathbf{Q} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1} \tag{A.13}$$

where $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues of **Q** in decreasing order along the diagonal. The columns of **S** contain the corresponding eigenvectors. Thus, it is relatively straight forward to find the eigenvector **b** by taking the singular value decomposition of **P** and choosing **b** as the last column of **S**.

A.3 Nyquist-Eigenfilters

The extension from the general eigenfilter to the Nyquist filter design problem is relatively straight forward. The Nyquist property is defined in the time domain. For most filter design techniques, time domain constraints cannot be incorporated.

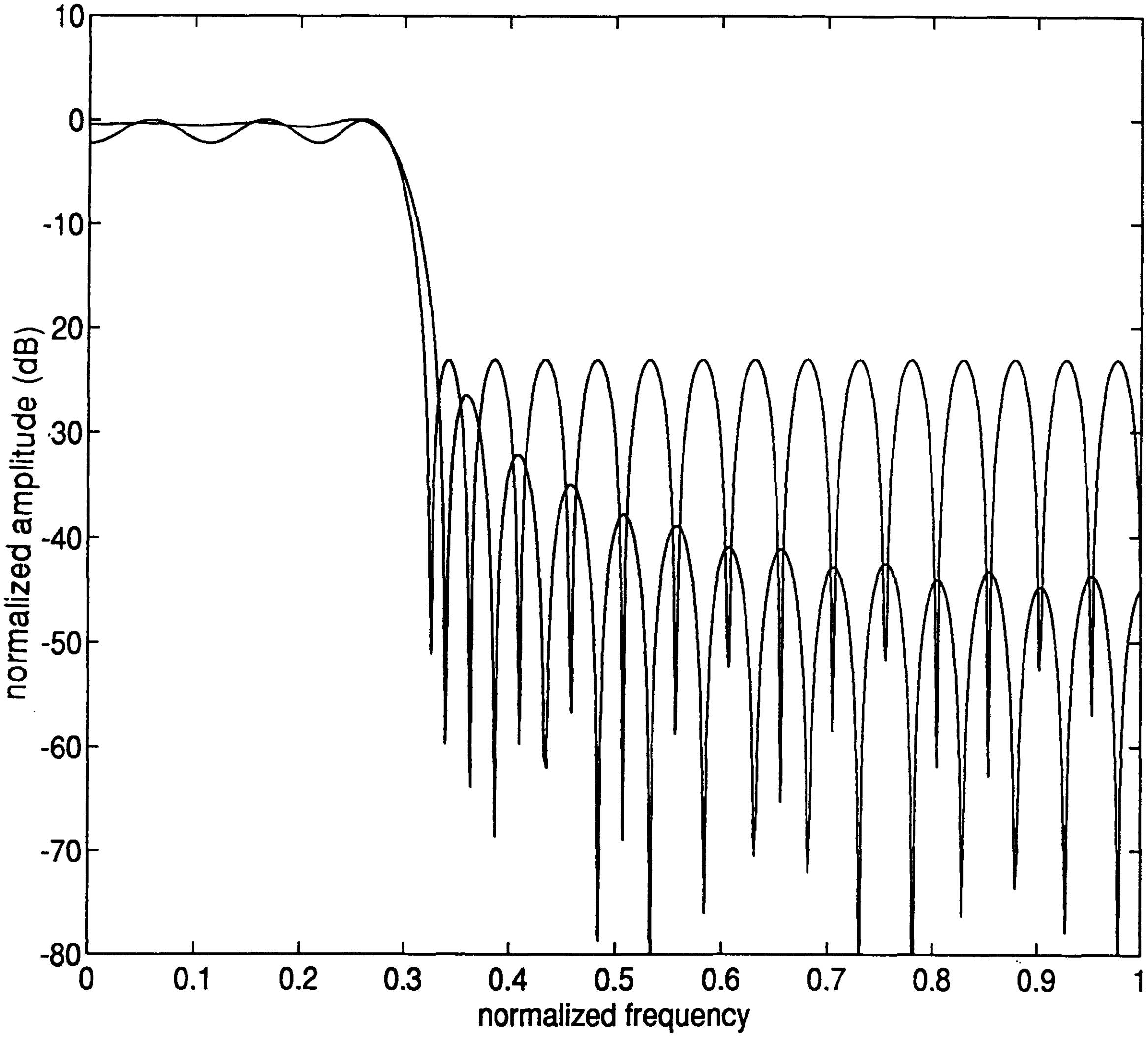

Figure A-2: Comparison of the standard Parks-McClellan algorithm (equiripple plot) and the Eigenfilter method with $(1-\alpha) = 0.3$ (the filter cutoff frequency). Note that the Eigenfilter has a slightly flatter passband, a slightly wider transition band, and a stopband that continues to fall off as the frequency increases.

However, the definition of a Nyquist (or M^{th} band) filter is

$$h_{Mn} = \begin{cases} 1/M & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{A.14}$$

which is a time domain constraint. Because the Eigenfilter approach can incorporate time domain constraints [16], it is particularly suitable for this problem.

If we incorporate the M^{th} band property into the Eigenfilter method, we can see from (A.5) that the minimization vector $\mathbf{b}$ will be of the form

$$\mathbf{b} = [b_0, b_1 \cdots, b_{M-1}, 0, b_{M+1}, \cdots, b_{2M-1}, 0, b_{2M+1}, \cdots]^t \tag{A.15}$$

Therefore, both the rows and the columns of $\mathbf{P}$ that are multiples of M do not contribute to the total error $\Phi = \mathbf{b}^t\mathbf{P}\mathbf{b}$. Thus, we can completely eliminate these rows and columns of $\mathbf{P}$ and every M^{th} element of $\mathbf{b}$. Doing this, we have a new error function $\Phi' = \mathbf{b}'^t\mathbf{P}'\mathbf{b}'$ where

$$\mathbf{b}' = [b_0, b_1 \cdots, b_{M-1}, b_{M+1}, \cdots, b_{2M-1}, b_{2M+1}, \cdots]^t \tag{A.16}$$

and

$$\mathbf{P}' = \begin{pmatrix} P_{0,0} & \cdots & P_{0,M-1} & P_{0,M+1} & \cdots & P_{0,2M-1} & P_{0,2M+1} & \cdots \\ P_{1,0} & \cdots & P_{1,M-1} & P_{1,M+1} & \cdots & P_{1,2M-1} & P_{1,2M+1} & \cdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ P_{M-1,0} & \cdots & P_{M-1,M-1} & P_{M-1,M+1} & \cdots & P_{M-1,2M-1} & P_{M-1,2M+1} & \cdots \\ P_{M+1,0} & \cdots & P_{M+1,M-1} & P_{M+1,M+1} & \cdots & P_{M+1,2M-1} & P_{M+1,2M+1} & \cdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ P_{2M-1,0} & \cdots & P_{2M-1,M-1} & P_{2M-1,M+1} & \cdots & P_{2M-1,2M-1} & P_{2M-1,2M+1} & \cdots \\ P_{2M+1,0} & \cdots & P_{2M+1,M-1} & P_{2M+1,M+1} & \cdots & P_{2M+1,2M-1} & P_{2M+1,2M+1} & \cdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{pmatrix} \tag{A.17}$$

The eigenvector corresponding to the minimum eigenvalue of this matrix is then found. This gives the vector $\mathbf{b}'$, which is not quite what we want. We still need to insert zeros back at every M^{th} element of the vector to give $\mathbf{b}$, which we can finally use to find the h_n.

Appendix B

Programs

B.1 Optimal Throw Away

```
%%%%%%%%%%%%%%%%%%%%%%%%%% optimal throw away %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Here the test spectrum is formed for testing f1 and f2.
% This is the test signal and is useful because it
% contains only those frequencies which we expect to see.
% (This is indicated by the variable Intervals)
% (We make the assumption here that the first
% frequency interval extends from 0 to some frequency and that there
% are two frequencies)

dftlength = 256;

% Assume these intervals contain all the significant frequency information:
% (scaled by 2pi and dftlength)
%Interval1:
beg1 = 0.0  * dftlength + 1;
end1 = .1/2 * dftlength + 1;
%%Interval2
beg2 = .3/2 * dftlength + 1;
end2 = .5/2 * dftlength + 1;

%%Create test signal
A1 = 1.;
f1 = .025;
phi1 = pi/3;

t = 1:50;
A2 = 1.;
f2 = .2;
phi2 = 0.0;

x = A1*sin(2*pi*f1*t + phi1)+A2*sin(2*pi*f2*t + phi2);

% zero pad for the FFT
xpad = zeros(1,dftlength-length(x));
x = [x,xpad];

% nset is the set of sample times
tt = t;
xt = x;
Xdfta = fft(x);

count = 1;
```

```
epsilon = [];

while (count <= 35)
% Throw away a sample and check the total error...
minerr = 99999999.;
for i = 1:length(xt)
if (xt(i) ~= 0)
tt(i)=0;
xt(i) = 0;
Xdft = fft(xt);
err = abs(sum((Xdft(beg1:end1) - Xdfta(beg1:end1)).^2)) ...
+ abs(sum((Xdft(beg2:end2) - Xdfta(beg2:end2)).^2));
%pick sample with minimum error..
if (err < minerr)
minerr = err;
besttt = tt;
bestxt = xt;
bestX = Xdft;
end
tt(i)=t(i);
xt(i)=x(i);
end
end

epsilon(count)=abs(minerr);
abminerr = abs(minerr)
tt = besttt;
xt = bestxt;
count = count+1

end
```

B.2 Autocorrelation

```
%%%%%%%%%%%%%%%%%%%%%%%% all-spacings %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%  The purpose of this program is to use an almost
% uniform autocorrelation set of data to estimate the
% frequency of some non-uniform data by using a sample
% pattern which includes almost all spacings from 0 to 64.

% Signal Parameters
A1 = 1;
f1 = .2;
```

```
phi1 = pi/2;
phi1 = 2*(-2*pi*f1)+pi/2;

A2 = 10;
f2 = .025;
phi2 = pi/3;
phi2 = 1*(-2*pi*f2)+pi/2;

% List of sample times:
t = [1 8 19 37 47 52 56 57 59 63 69 82 96 98 115]; %Almost every spacing
%t = [1 2 5 12 27 33 57 69 77 116 118 135 151 164 169 178]; %Spanning Ruler

%%  Make a set of ones and zeros so you can stem the sample times
ts = zeros(1,max(t));
j=1;
for i=1:max(t)
if (i == t(j))
ts(i)=1;
j=j+1;
end
end

tr = 1:max(t);  %Uniform sampling for comparison
x = A1*sin(2*pi*f1*t + phi1) + A2*sin(2*pi*f2*t + phi2);
xr = A1*sin(2*pi*f1*tr + phi1) + A2*sin(2*pi*f2*tr + phi2); %Uniform x

% Calculate sum of x(n)x(n+l) for all l's. This is a(n). (the autocorelation)
a = zeros(1,max(t));
beg = 1;
while (beg <= length(t))
for i=beg:length(t)
a(abs(t(beg)-t(i))+1) = a(abs(t(beg)-t(i))+1) + ...
x(beg)*x(i);
end
beg = beg + 1;
end

%  finish with a hanning window
ahan = a.*hanning(length(a))';
```

B.3 Interpolation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Interpolation %%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Interpolation Method from Scouler's Sampling Paper

L=3;
M=10;

%Create test signal

A1 = 10;
f1 = .025;
phi1= 3*pi/4;

A2 = 1;
f2 = .3;
phi2= pi/3;

%  Must make sure that the total duration is a multiple of M
JJJ=M*5;
t = 1:JJJ;
sigmanoise = 0.5;

x = A1*sin(2*pi*f1*t + phi1) + A2*sin(2*pi*f2*t + phi2);
noise = randn(1,length(x))*sigmanoise;
x = (x + noise).*hanning(length(x))';

%load radar data (for use with real radar data)
%load range.dat
%x=range;
%x=x(1:140);

% Get the polyphase (nonuniform samples) of input signal
fc = .1;
X = polyphase(x,M);
X = X(1:L,:);

%Implementation of Scoular's formula
for l = 1:length(x)
for n = 1:length(x)/M
Psi1(l,n) = (sin(2*pi*(fc)*(l-M*n-1))/sin(2*pi*fc*(0-1)))*...
    (sin(2*pi*(fc)*(l-M*n-2))/sin(2*pi*fc*(0-2)))*...
    sinc(.1*(l-M*n));
end
end
for l = 1:length(x)
for n = 1:length(x)/M
Psi2(l,n) = (sin(2*pi*(fc)*(l-M*n))/sin(2*pi*fc*(1-0)))*...
    (sin(2*pi*(fc)*(l-M*n-2))/sin(2*pi*fc*(1-2)))*...
    sinc(.1*(l-M*n-1));
end
end
for l = 1:length(x)
for n = 1:length(x)/M
Psi3(l,n) = (sin(2*pi*(fc)*(l-M*n))/sin(2*pi*fc*(2-0)))*...
    (sin(2*pi*(fc)*(l-M*n-1))/sin(2*pi*fc*(2-1)))*...
    sinc(.1*(l-M*n-2));
end
end

% Summation of Inperpolating functions
xf = Psi1*X(1,:)' + Psi2*X(2,:)' + Psi3*X(3,:)';
xf = xf';

% These are used to draw the "assumption boxes"
fbox = [0 .0999 .1001 .2999 .3001 .4999 .5001 1];
abox = [0 0 -30 -30 0 0 -30 -30];
```

B.4 Filter Banks

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Filter Banks %%%%%%%%%%%%%%%%%%%%%%%%

% General Method for Multiband case

M = 13;
L = 4;

jimag = sqrt(-1);
Wm = exp(-2*jimag*pi/M);

%This is the set of 2pi/M wide intervals from zero to 2pi
%that are the support of X(w).  For real signals, it should be
%symmetric about its midpoint

Intervals = [1 0 0 1 0 0 0 0 0 1 0 0 1];
Lset =[];
for i = 1:M
if (Intervals(i) == 1)
```

```
Lset = [Lset,(i-1)];
end
end

%We create the matrix C=B * Wdagger * diag(Wm^(1/2)k
% Need B matrix B = [b0 b1 ...bL-1]
% so first create bp's
% bp = diag(Wm^p*k) * uq  if p=1 on Intervals
% bp = [0 0 .. 0]t  if p=0 on Intervals
% So we need U !!!
% inv(U) = V
% So we need V !!
% V = eqn 3.9 in Liu's Phd Thesis
for i = 1:L
for j = 1:L
V(i,j) = Wm^((j-1)*(Lset(i)));
end
end

U = inv(V);

for p = 1:M
Dflag = 0;
for q = 1:L
if (((p-1) == Lset(q)) & Dflag == 0)
lam = [];
for i = 1:L
lam(i,i) = Wm^((p-1)*(i-1));
end
B(:,p) = lam*U(:,q);
Dflag = 1;
elseif (Dflag == 0)
B(:,p) = zeros(L,1);
end
end
end

% We have B, so now we need Wdagger
for i=1:M
for j=1:M
Wdagger(i,j) = Wm^(-(i-1)*(j-1));
end
end

% and last we create that diagonal matrix
```

```
for i=1:M
diag(i,i) = Wm^((-1/2)*(i-1));
end

% Now we're ready for C

C = B * Wdagger * diag;

%%Now, the second part is to design the filters G
%% First, we should design the prototype filter, P
% which is defined to be low pass , wc = 2pi/M
% P is also a Mth band filter

Fp = [0 (2/M-.034/4) (2/M+.034/4) 1];
Mp = [1 1 0 0];
Wp = [1/M (1-1/M)];

%We need to pick a length that makes the polyphase work out right
% Pick initial length:
ilength = 200;
flag = 0;
ii = 1;
while (flag == 0)
        if (ilength < (2*M+1)+(2*M*ii))
                len = (2*M+1)+(2*M*ii);
                flag = 1;
        end
        ii = ii+1;
end

p = nyqfilt(len,Fp,Mp,Wp,M);

% Find Polyphase components of Mth band Filter, P
P = [];
P = polyphase(p,M);

%Now find the G's using eqn 3.29
G = [];
for i=1:M
G(i,:) = P(i,:).*(M/(2*cos((i-1)*pi/M)));
end

%But we need upsampled G's ...
Gtemp=[];
```

```
for i=1:M
        Gtemp(i,:) = upsample(G(i,:),M);
end
G = Gtemp;

% The rest involves implementing Fig. 3.10

%Create test signal
A1 = 10.;
f1 = .025;
phi1 = pi/3;

A2 = 1.;
f2 = .2;
phi2 = 0.0;

% Make sure duration is a multiple of M
JJJ = M*4;
t = 1:JJJ;

x = A1*sin(2*pi*f1*t + phi1)+A2*sin(2*pi*f2*t + phi2);
x = (x).*hanning(length(x))';

%Import radar signal for testing
%load range.dat;
%x = range;
%x=x(1:143);

U0 = polyphase(x,M);
U1=[];
U1(1,:) = U0(1,:);
for i = 2:L
        U1(i,:) = U0(M-(i-2),:);
end

U2 = [];
U2(1,:) = [U1(1,:),0];
for i=2:L
        U2(i,:) = delay(U1(i,:),2);
end
U3 = [];
for i=1:L
        U3(i,:) = upsample(U2(i,:),M);
end
U4 = [];
```

```
for i=1:L
        U4(i,:) = [delay(U3(i,:),L-i+1),zeros(1,i)];
end
U5 = [];
% need to convert C to traspose to fit the IEEE paper
C = C';
CO=C;
C = real(C);
U5=C*U4;
U6 = [];
for i=1:M
        U6(i,:) = conv(U5(i,:),G(i,:)).*M;
end

xhat = zeros(1,length(U6)+M-1);
for i=1:M
        xhat = xhat + [delay(U6(i,:),i),zeros(1,M-i)];
end
```

B.4.1 Filter Design

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Filter Design: Nyquist Filters %%%%%%%%%%

% This is used to draw the "assumption boxes"
fbox = [0 .15383 .15384 .46153 .46154 .61538 .61539 1];
abox = [0 0 -30 -30 0 0 -30 -30];
function [h] = nyqfilt(N,F,M2,W,K)
%nyqfilt Nyquist Filter Design using the Eigenfilter Method.
% h = nyqfilt(N,F,M2,W,K)
% Kth band Nyquist filter, designed as follows:
% (Same as Eigenfilter):
% Gives the impulse response of the Nth order filter that
% minimizes weighted passband and stopband errors
% by using the Eigenfilter method.
% Frequency response is specified by F and M2.
%          Vectors F and M2 specify the frequency and magnitude
%          breakpoints for the filter such that PLOT(F,M2) would show a
%          plot of the desired frequency response. The elements of M2 must
%          appear in equal-valued pairs.  The frequencies in F must be
%          between 0.0 < F < 1.0, with 1.0 corresponding to half the
%          sample rate. They must be in increasing order, start with 0.0,
%          and end with 1.0.
% M2 must have elemens 1 or 0 with 1's and zeros in blocks of
% 2.  i.e M2 = [0 0 1 1 0 0 1 1 0 0]
%         W is used to specify weighting in each
%         of the pass or stop bands corresponding to vectors F and M2,
% and has half the number of elements of F and M2.
% The sum of the elements in W must add up to 1.

M = (N-1)/2;
P = []; %Initialize P
%The theory behind the following is from the original paper on eigenfilters
%by Truong Nguyen (see bibliography)

%%% Make elements of P
flag=0;
for i = 1:M+1
for j = 1:M+1
P(i,j)=0.0;
for k=1:2:length(F)-1
 if (M2(k) == 0)
          P(i,j) = P(i,j)+(W((k+1)/2)/pi)*Pstop(i-1,j-1,F(k),F(k+1));
 elseif (M2(k) == 1)
  if (F(k) == 0)
   P(i,j) = P(i,j)+(W((k+1)/2)/pi)*Ppass(i-1,j-1,F(k),F(k+1)...
,0);
  elseif (F(k+1) == 1)
   P(i,j) = P(i,j)+(W((k+1)/2)/pi)*Ppass(i-1,j-1,F(k),F(k+1)...
,1);
  else
   if flag==0
   temp1=F(k);
   temp2=F(k+1);
   flag=1;
   end
   P(i,j) = P(i,j)+(W((k+1)/2)/pi)*Ppass(i-1,j-1,F(k),F(k+1)...
,(temp1+temp2)/2);
  end
 end
end
end
end

%%      get rid of Kth columns of P to get Q
Q=[];
Q=[Q,P(:,1:K)];
for i=1:M/K
        if (M+1 > (i+1)*K-1+1)
```

```
         Q = [Q,P(:,(K*i+1+1):((i+1)*K-1+1))];
        else
         Q = [Q,P(:,(K*i+1+1):M+1)];
        end
end

%%      get rid of Kth rows of Q to get R
R=[];
R = [R;Q(1:K,:)];
for i=1:M/K
        if (M+1 > (i+1)*K-1+1)
         R = [R;Q((K*i+1+1):((i+1)*K-1+1),:)];
        else
         R = [R;Q((K*i+1+1):M+1,:)];
        end
end

% now we have Pprime = R
P = R;
%and we have to redefine M for the new P

M = length(P)-1;
[U,S,V] = svd(P);
b = U(:,M+1);

% now we have to add zeros at approprate locations on b
d = [];
K=K-1;

d = [d;b(1:K+1)];
d = [d;0.0];
for i = 1:M/K+K+1+1
        if (M+1 > (i+1)*K+1)
         d = [d;b(K*i+1+1:(i+1)*K+1)];
         d = [d;0.0];
        else
         d = [d;b(K*i+1+1:M+1)];
                if ((N-1)/2+1 > length(d))
                 d = [d;0.0];
                end
        end
end
b=d;
%be carefull!! set M back to original!!
M = (N-1)/2;
```

```
h = [];
for i=1:M %find h(n) from b(n)
h(i) = 0.5*b(M+1-(i-1));
end
h(M+1)=b(1);
for i=2:M+1
h((M+1)+(i-1)) = 0.5*b(i);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%% Filter Design: Ppass matrix %%%%%%%%%%%%%

function [p] = Ppass(n,m,wp1,wp2,w0)
%Ppass function to calculate the n,mth element of the Ppassband matrix
% for the Eigenfilter P matrix
% wp1 and wp2 are assumed to be normalized w.r.t pi
% it can be thought of as the integral FROM wp1 TO wp2
% w0 is the center frequency of the passband

wp1 = wp1*pi;
wp2 = wp2*pi;
w0  = w0*pi;

if (n == 0 | m == 0)
 p=0.0;
elseif (n == m)
 p = ((cos(n*w0)^2)*n*wp2 - 2*cos(n*w0)*sin(n*wp2) + ...
0.5*cos(n*wp2)*sin(n*wp2) + 0.5*n*wp2)/n ...
   - ((cos(n*w0)^2)*n*wp1 - 2*cos(n*w0)*sin(n*wp1) + ...
0.5*cos(n*wp1)*sin(n*wp1) + 0.5*n*wp1)/n;
else
 p = (0.5*cos(n*w0-m*w0)*wp2 + 0.5*cos(n*w0+m*w0)*wp2  ...
+ 0.5*(sin(n*w0-m*wp2)/m) - 0.5*(sin(n*w0+m*wp2)/m) ...
- 0.5*(sin(n*wp2-m*w0)/n) - 0.5*(sin(n*wp2+m*w0)/n) ...
+ 0.5*(sin(n*wp2-m*wp2)/(n-m)) + 0.5*(sin(n*wp2+m*wp2)/(n+m))) ...
   - ...
(0.5*cos(n*w0-m*w0)*wp1 + 0.5*cos(n*w0+m*w0)*wp1  ...
+ 0.5*(sin(n*w0-m*wp1)/m) - 0.5*(sin(n*w0+m*wp1)/m) ...
- 0.5*(sin(n*wp1-m*w0)/n) - 0.5*(sin(n*wp1+m*w0)/n) ...
+ 0.5*(sin(n*wp1-m*wp1)/(n-m)) + 0.5*(sin(n*wp1+m*wp1)/(n+m)));

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%% Filter Design: Pstop matrix %%%%%%%%%%%%%

function [p] = Pstop(n,m,wp1,wp2)
```

```
%Pstop function to calculate the n,mth element of the Pstopband matrix
% for the Eigenfilter P matrix
% wp1 and wp2 are assumed to be normalized w.r.t pi
% it can be thought of as the integral FROM wp1 TO wp2
if (n == 0 & m == 0)
 p = wp2*pi-wp1*pi;
elseif (m == 0)
 p = (sin(pi*n*wp2))/n - (sin(pi*n*wp1))/n;
elseif (n == 0)
 p = (sin(pi*m*wp2))/m - (sin(pi*m*wp1))/m;
elseif (n == m)
 p = (0.5*cos(pi*m*wp2)*sin(pi*m*wp2) + .5*m*pi*wp2)/m ...
   - (0.5*cos(pi*m*wp1)*sin(pi*m*wp1) + .5*m*pi*wp1)/m;
else
 p  = (sin(pi*(m+n)*wp2)/(2*(m+n)) + sin(pi*(m-n)*wp2)/(2*(m-n)))   ...
    - (sin(pi*(m+n)*wp1)/(2*(m+n)) + sin(pi*(m-n)*wp1)/(2*(m-n)));
end
```

B.4.2 Matlab Implementation

```
%%%%%%%%%%%%%%%%%%%%%%%%%% Generic .m functions:  Delay %%%%%%%%%%%%

function [yd] = delay(y,d);
%delay Delays the vector y by d units

for i = 1:d-1
u(i)=0;
end
u(d)=1;
yd=conv(u,y);

%%%%%%%%%%%%%%%%%%%%%%%%%% Generic .m functions:  Polyphase %%%%%%%%%%%%

function [p] = polyphase(q,M)
%polyphase Performs an M-fold polyphase decompostion of the
% time series q, returning the matrix p where the ith
% row of p is the ith polyphase component of q

for i = 1:M
for j = 1:length(q)
if ((j-1)*M + i) > length(q)
 junk=1;
else
```

```
 p(i,j) = q((j-1)*M + i);
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%% Generic .m functions:  upsample  %%%%%%%%%%%%

function [y] = upsample(x,N);
%upsample upsample(x,N):
% add N-1 zeros between samples of x

y=[];
for i = 1:length(x)-1
y = [y,x(i),zeros(1,N-1)];
end
y(length(y)+1) = x(length(x));
```

Bibliography

[1] M. Bellanger, G. Bonnerot, and M. Coudreuse. Digital filtering by polyphase network: application to sample rate alteration and filter banks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24:109–14, April; 1976.

[2] R. E. Crochiere and L.R. Rabiner. *Multirate Digital Signal Processing.* Prentice Hall, Englewood Cliffs, NJ, 1983.

[3] P. Jarske, T. Saramaki, S.K. Mitra, and Y. Neuvo. On properties and design of nonuniformly spaced linear arrays. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(3):372–80, March 1988.

[4] Steven M. Kay. *Modern Spectral Esimation.* Prentice Hall, Englewood Cliffs, NJ, 1988.

[5] A. D. Keedwell. *Surveys in Combinatorics, 1991*, pages 29–35. Cambridge University Press, Cambridge, England, 1991.

[6] Vincent Cheng-Teh Liu. *One and Two-Dimensional Digital Multirate Systems with Applications in Subpampling and Bandlimited Signal Reconstruction.* PhD thesis, California Institute of Technology, 1990.

[7] Robert J. II Marks. *Introduction to Shannon Sampling and Interpolation Theory.* Springer-Verlag, New York, NY, 1991.

[8] B. Nobel and J.W. Daniel. *Applied Linear Algebra.* Prentice Hall, Englewood Cliffs, NJ, 1977.

[9] J.M. Nohrden and T.Q. Nguyen. Constraints on the cutoff frequencies of mth band filters. to be submitted for publication, 1994.

[10] P.R.M. Rao, P.V.S.S. Meher, N. Ramanjaneyulu, Y. Ravindranath, and G.S.N Raju. Sidelobe reduction using element space distribution. In *Proceedings of the 1989 Int. Symposium on Antennas and Propogation*, volume 3, pages 565–8. ISAP, Inst. Electron. Inf. Commun. Eng., 1989.

[11] S.C. Scoular and W.J. Fitzgerald. Peridic nonuniform sampling of multiband signals. *Signal Processing*, 28:195–200, 1992.

[12] W. M. Siebert. *Circuits, Signals, and Systems.* MIT Press and McGraw Hill, Cambridge, MA, 1986.

[13] Gilbert Strang. *Introduction to Applied Mathematics*, pages 79–79. Wellesley-Cambridge Press, Wellesley, MA, 1986.

[14] P.P. Vaidyanathan. *Multirate Systems and Filter Banks.* Prentice Hall, Englewood Cliffs, NJ, 1992.

[15] P.P. Vaidyanathan and V.C. Liu. Efficient reconstruction of band-limited sequences from nonuniformly decimated versions by use of polyphase filter banks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(11):1927–1936, November 1990.

[16] P.P. Vaidyanathan and T.Q. Nguyen. Eigenfilters: A new approach to least squares fir filter design and applications including nyquist filters. *IEEE Transactions on Circuits and Sustems*, CAS-34(1):11–23, January 1987.