SHIP TARGET RECOGNITION

Anthony David Robinson [1]

Thesis prepared in fulfillment of the requirements for the Degree of MSc in Electrical Engineering.

September 24, 1996

[1]Radar Remote Sensing Group, University of Cape Town

Abstract

In this report the classification of ship targets using a low resolution radar system is investigated. The thesis can be divided into two major parts.

The first part summarizes research into the applications of neural networks to the low resolution non-cooperative ship target recognition problem. Three very different neural architectures are investigated and compared, namely; the Feedforward Network with Back-propagation, Kohonen's Supervised Learning Vector Quantization Network, and Simpson's Fuzzy Min-Max neural network. In all cases, pre-processing in the form of the Fourier-Modified Discrete Mellin Transform is used as a means of extracting feature vectors which are insensitive to the aspect angle of the radar. Classification tests are based on both simulated and real data. Classification accuracies of up to 93% are reported.

The second part is of a purely investigative nature, and summarizes a body of research aimed at exploring new ground. The crux of this work is centered on the proposal to use synthetic range profiling in order to achieve a much higher range resolution (and hence better classification accuracies). Included in this work is a comprehensive investigation into the use of super-resolution and noise reducing eigendecomposition techniques. Algorithms investigated include the Principal Eigenvector Method, the Total Least Squares Method, and the MUSIC method. A final proposal for future research and development concerns the use of time domain averaging to improve the classification performance of the radar system. The use of an iterative correlation algorithm is investigated.

Acknowledgments

The author is indebted to a number of people: Professor Mike Inggs for his tireless support, guidance and advice. Messrs. Piet Botha, Simon Normal and Johan Theron of IMT, Simon's Town, for providing the data from their facility. Mr. Roy Blatch (then in the SA Navy) integrated the equipment and provided the early data sets. Tanya Douglas, Anton Krantz and Richard Remmington, who all contributed to various phases of the investigations reported here. Alan Langman, for his many tips and ideas on super-resolution techniques. Mr. R. S. Gill for his discussions on stepped frequency waveform design. And finally, all the members of the Radar Remote Sensing Group, for countless small favors, and for making the sixth floor such an enjoyable place to work. Thanks in particular to Norman, Mark, Shaheen, Pete, Paul, Leon, Martin, Tom, Ralph, Gernot and Richard.

Synopsis

This report documents an investigation into ship target recognition. Its purpose is to highlight the author's research in this field during the last three years in fulfillment of the requirements for obtaining an MSc. degree from the University of Cape Town. The objectives are to lay a foundation for the development of a fully operational non-cooperative target recognition radar system.

The work described in this thesis was inspired by earlier research in which an algorithm known as the Fourier Mellin (FM) Transform was used for pre-processing. In this earlier work the resulting feature vector was converted to an equivalent binary word and some standard techniques of binary word recognition were then carried out to implement classification. The work was significant because it pointed to the promise of achieving recognition through the use of standard low resolution radar systems, such as those typically found in most harbours and ports today.

This work has since been superseded by the author's research reported in this thesis, whereby neural network classifiers are used to produce far more robust results.

Although many neural network algorithms are able to operate directly on the raw data by developing their own complex mapping functions (e.g. back-propagation), the FM transform is retained as an important pre-processing stage. In general neural based classifiers always produce superior results when used *in conjunction* with known transforms, than when left to develop their own complex mapping functions. This approach did not compromise the many advantages offered by a completely neural architecture (such as speed), since the transform is in fact well suited to a parallel implementation.

As far as actual classification is concerned, three neural networks are investigated, namely; the Feedforward Network with Back-propagation, Kohonen's Supervised Learning Vector Quantization Network, and Simpson's Fuzzy Min-Max neural network. The three networks are based on widely differing design philosophies, and the relative merits and shortcomings of each approach are compared.

Although all three classifiers were capable of similar classification accuracies, both the back-propagation and LVQ algorithms are found to fall short of the ideal. Back-propagation suffered from various training problems, including an inability to converge and slow training times. Although LVQ showed significant improvements in this regard, both networks re-

quired random guesswork in the tuning process, and were not able to adapt and update themselves on an on-going basis. In this regard the Fuzzy Min-Max neural network is shown to be extremely promising. It is capable of learning new classes and refining existing classes on the fly, without destroying old class information and without the need for complete retraining. In addition, it has only a few well understood training parameters.

Overall, it is shown that classification accuracies of over 90% are feasible with the existing radar system.

Despite the good results achieved, it was felt that the existing system could be significantly improved. The real data sets are limited, and the low range resolution of the radar made accurate discrimination of a wide range of ship targets unlikely. As a result, a considerable amount of groundwork is laid for developing an improved recognition system. This includes a proposal for the use of synthetic range profiles (SRP's), the use of resolution enhancing and noise reduction algorithms, and also the use of time domain averaging through a fast correlation algorithm.

The proposal for the use of synthetic range profile processing is motivated by the ability of this approach to achieve high range resolution with many types of existing search and track radars, which can then be used for target recognition and target imaging in addition to their normal search and track functions. This approach is attractive because it preserves the spirit of the original system, i.e., a system that is modest in its hardware requirements and therefore accessible to a variety of users such as coastguards, search and rescue organizations and harbour masters. A brief theoretical overview of synthetic range profiling is provided, followed by various practical observations, including a look at the effects of matched filtering as well as the effects of target velocity. There is also a practical overview of SRP waveform design.

This research on synthetic range profiling is expanded further through an investigation into the optimal approach for processing synthetic range profiles. Of particular interest here is the use of super resolution techniques, which enable one to achieve much higher range resolution than that of traditional approaches such as the FFT. The use of eigendecomposition to improve the noise resilience of these techniques is also investigated in some detail. Three algorithms are identified as particularly promising. These are; the Principal Eigenvector Method, the Total Least Squares Method, and the MUSIC method. Although producing similar performances, the MUSIC algorithm proves optimal in terms of noise resilience and tolerance to inacuracies in the estimation of the sizes of the noise and signal spaces.

A final proposal for future research and development concerns the use of time domain averaging. Time domain averaging is attractive because it improves the signal to clutter ratio of the ship profiles and also assists in reducing the influence of the target's anisotropic reflectors (which confuse recognition). A corresponding improvement in classification accuracies is predicted. The use of an iterative correlation algorithm is investigated that

promises robust performance. In addition, a fast implementation of the algorithm is developed that is amenable to a neural-based implementation.

Contents

List of Figures

List of Tables

Introduction

This thesis sets out the results obtained to date from an investigation aimed at developing a fully operational non-cooperative ship target recognition radar system. The purpose of the thesis is to highlight the author's research in this field which was carried out over the last three years. The thesis is primarily of an investigative nature, aimed at developing a theoretical basis for future work on the ship target recognition problem. It has been completed in fulfillment of the MSc. degree requirements at the University of Cape Town, South Africa.

Ship target recognition has numerous applications, including coastguard control, sea rescue, the regulation of shipping channels, and naval warfare, and research on the subject is already fairly established [37] [46] [47] [49]. The work described in this thesis was specifically inspired by a paper by Bufa [37]. Bufa used the Fourier Mellin (FM) Transform for pre-processing, and then converted the output to an equivalent binary word. Some standard techniques of binary word recognition were then carried out to implement recognition.

Some preliminary work to this project [49] [47] investigated Bufa's method and some adaptations. This work has now been superseded by research by the author in which neural networks are used to produce far more promising results.

The objectives of this thesis are as follows:

- To provide a description of the radar system and the available data
- To give a brief overview of the classification problem and of existing preprocessing techniques such as the Fourier Mellin Transform.
- To consolidate, develop and expand on the author's research into the use of neural network classifiers.
- To build a comprehensive foundation for future research on ship target recognition.
- To draw conclusions where necessary

The information in this thesis was gathered from books and journal articles obtained through the various University of Cape Town libraries. Private discussions where held with Professor M. Inggs, Alan Langman and other researchers in the Radar Remote Sensing Laboratory. Discussions where also held with R. S. Gill and other authors at the IEEE International

Radar Conference (Washington, D.C.). Except where indicated in the text the contents are entirely original and are not the result of work done in collaboration. Due to the large amount of available literature, as well as the previous work conducted by the author [50], this thesis provides a very superficial summary of the field of neural networks. It is assumed that the reader has a working knowledge of neural networks and also of digital signal processing techniques in general.

The text consists of seven chapters. Chapter 1 begins by laying the groundwork for the research reported in this thesis. Included here is a description of the radar system and the real data that was obtained. This is followed by a summary of the main features of the classification problem, and an overview of the Fourier Mellin transform, whereby *a priori* information on the classification problem is used to achieve feature extraction.

Chapter 2 describes the classification process. Three networks are investigated, namely; the Feedforward Network with Back-propagation, Kohonen's Supervised Learning Vector Quantization Network [40], and Simpson's Fuzzy Min-Max neural network [42]. They are based on widely differing design philosophies, and it is interesting to compare the relative merits and shortcomings of each approach. Classification results of the three networks are presented.

Chapter 3 marks the beginning of several exploratory chapters, aimed principally at laying a theoretical foundation for future research on ship target recognition. In Chapter 3 an improved radar system involving the use of synthetic range profiles (SRPs) is proposed. By employing a stepped frequency waveform, synthetic range profile processing makes it possible to achieve high range resolution with many types of existing search and track radars. A brief theoretical overview of synthetic range profiling is provided, followed by various practical observations. Included here is a look at the effects of matched filtering as well as the effects of target velocity. There is also a practical overview of SRP waveform design.

Chapter 4 documents an investigation into the optimal approach for processing synthetic range profiles. Of particular interest here is the use of super resolution techniques, which enable one to achieve much higher range resolution than that of traditional approaches. The use of eigendecomposition to improve the noise resilience of these techniques is also investigated in some detail.

In the last of the exploratory chapters, Chapter 5, the employment of time domain averaging by means of a fast correlation algorithm is proposed. Time domain averaging is attractive because it improves the signal to clutter ratio of the ship profiles and also assists in reducing the influence of the target's anisotropic reflectors (which confuse recognition). A corresponding improvement in classification accuracies is therefore to be expected.

Chapter 6 contains a summary and final discussion, and the last chapter, Chapter 7, provides recommendations for future work.

The appendices contain mathematical derivations which are too long to

be included in the main text. They are followed by the bibliography which can be found at the end of the thesis.

Chapter 1

Groundwork

This chapter lays the groundwork for the research reported in this thesis. It begins with a summary of the main features of the classification problem. This is followed by a description of the radar system and also the data that was obtained. It concludes with an overview of the use of the Fourier Mellin transform, whereby *a priori* information on the classification problem is used to achieve feature extraction.

1.1 The Classification Problem

The principal factor that distinguishes the ship classification problem from other target classification problems (such as aircraft recognition), is that non co-operative determination of the target orientation with respect to the radar is particularly difficult. This is because slow moving ships can execute rapid course alterations. Even worse, when the target is stationary, this task becomes impossible. It is therefore essential that the classification system not rely on accurate target aspect angle information.

A ship generally has a large length to width ratio, and can be considered to be a linear array of reflectors. In order to quantify how the range profile will change with aspect angle, we note the following:

- If the aspect of the target changes, the relative ranges of points will change. The result will be range migration. i.e., the two points which were once in the same range bin are no longer. If the points are located along one axis in a straight line, (such as on ship targets), then this range migration corresponds to a sinusoidal scaling of the independent variable of the range profile - i.e. the line of sight range profile will become compressed or stretched. The profile will have greatest dimension when the radar signal and the length axis of the ship are aligned (zero aspect angle).

- If the change in aspect angle is not large enough to cause significant range migration, even small changes in the down-range separation of the reflecting features will result in the phenomenon known as speckle, which plagues all coherent imaging systems. The wavelength of the radar is 3cm. This means that a change in the down-range spatial distribution of any two scattering centers in a range bin by just

0.75cm can result in a change from complete destructive interference to complete constructive interference.

- Magnitude fluctuations will also arise due to the fact that the scattering features are generally anisotropic in nature. This applies especially to large, flat reflectors, which only reflect strongly at one specific aspect angle. Also, over large aspect ranges, one can expect certain features to become occluded, and new ones to emerge.

1.2 The Radar System

The pulsed, non-coherent radar is typical of the systems currently in use worldwide for short range ship navigation. In the mode employed for the data capture, each pulse (9.3 GHz) has a length of 80ns, giving a relatively low range resolution of 12m. The pulse repetition frequency (prf) is 3600Hz and the scan rate is 2.2 seconds. Sampling is in the time-domain at 25MHz, using eight bits. The digitization system is able to sample any range-azimuth domain set by the user. The IF and detector system is very close to linear in terms of power. Figure 1.1 shows a typical downrange profile of a ship target. The inter-pulse interval has been made shorter than in practice to give an idea of the formation of successive profiles.

A typical example of one azimuth versus range *scan* is given in Figure 1.2. The characteristic bell shape in azimuth is due to the antenna gain modulation. For the radar system used, the antenna beamwidth was less than a degree.

It is important to note that at a given time, the pulse illuminates only a single strip of length equal to the pulse resolution of the radar (12m in this case). The digitized sample represents the instantaneous vector sum of the scatterers currently being illuminated. In the system used, range samples are taken at 40ns intervals, about two per resolution cell of the radar pulse. The radar bandwidth is not carefully defined in this kind of radar, but is probably wider than the reciprocal of the pulse width, the classic matched filter. The net result is that the higher sampling rate is justified and probably results in some of the finer detail of the target structure being recorded.

1.3 A Description of the Data

With reference to Figure 1.2, it should be clear that the quality of the range profiles was generally high. The probability-density function for the cross section was typically Rayleigh distributed. Magnitude fluctuations in the radar profiles were found to correspond closely to Swerling's first model for calculating detection probabilities [25], i.e., the echo pulses received from a target on any one scan are of constant amplitude throughout the entire scan

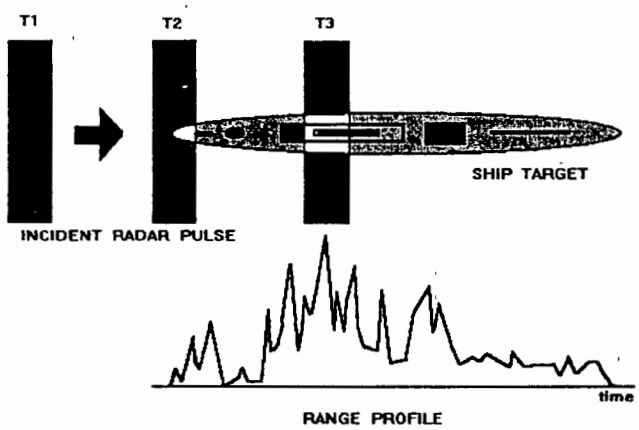


Figure 1.1: Relation between radar range profile and target.

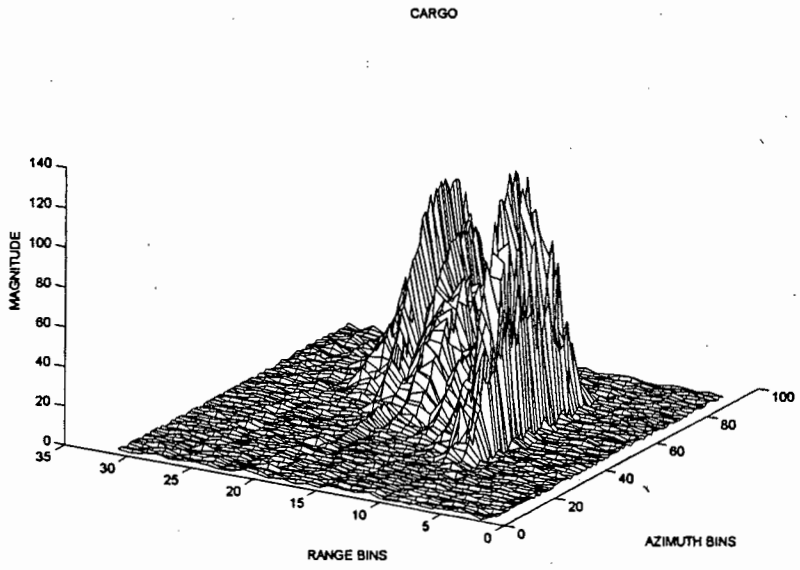


Figure 1.2: 3-Dimensional plot of a typical scan from a cargo ship.

but are independent (uncorrelated) from scan to scan. This assumption of course ignores the effect of the antenna beam shape on the echo amplitude.

For an example of the scan to scan magnitude fluctuations, Figure 1.4 shows several scans for a cargo ship, which was supposedly stationary over the measurement period. Consecutively numbered scans are separated in time by 2.2 seconds. Significant fluctuations caused by small changes in the orientation of the target over this time period can be clearly observed.

This low correlation between scans is most likely the major limiting factor as far as classification accuracy is concerned. One way in which these magnitude fluctuations can be reduced is by improving the resolution of the radar. This is addressed in Chapter 3. Another approach would be to introduce some sort of scan to scan averaging. This is addressed in Chapter 5.

1.3.1 Sea Clutter and Signal-to-Clutter Ratio

Sea clutter is the term used to denote the reflection of radar energy from the surface of the sea. Depending on several factors such as the angle that the incident beam makes with the horizontal, the wavelength and polarization of the radar energy, the state of the sea and wind, sea clutter can seriously affect target recognition.

Various clutter reduction techniques exist, including Moving Target Indication or MTI (not suitable for ship targets), sensitivity time control (STC), or the use of a matched filter or logarithmic receiver [25]. Software based clutter reduction algorithms are time-consuming and will counteract any speed advantage offered by a neural network classifier. For the radar system employed in this work, no attempts were made at sea clutter reduction.

In order to quantify the effects of sea clutter, we define the signal to clutter (S/C) ratio to be the peak amplitude of signal to clutter, given by

$$20log(\frac{A_{signal}}{A_{clutter}})dB \tag{1.1}$$

In general, the degradation of the scans was found to be minimal, and S/C ratios of $+15dB$ were typical (a strong burst of sea clutter is illustrated in figure 1.3). However it should be noted that all of the data was recorded in 'calm', 'smooth' or 'slight' sea conditions, corresponding to wave heights of 0, 1 and $1-3$ meters respectively. No data was recorded for the sea states 'moderate', 'rough', 'very rough' and 'high' (these sea-states correspond to wave heights of $3-5$, $5-8$, $8-12$ and $12-20$ meters respectively). Under these conditions, the influence of sea clutter can be expected to degrade the quality of the range profiles quite significantly.

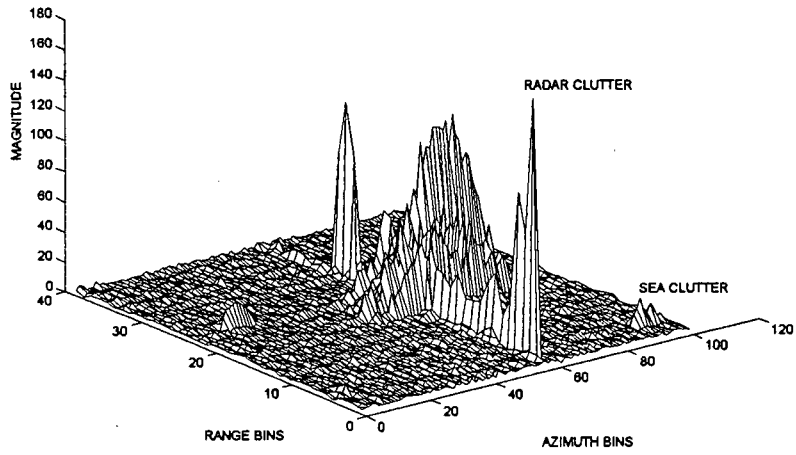


Figure 1.3: A poor quality scan – Corrupted with sea clutter and radar interference.

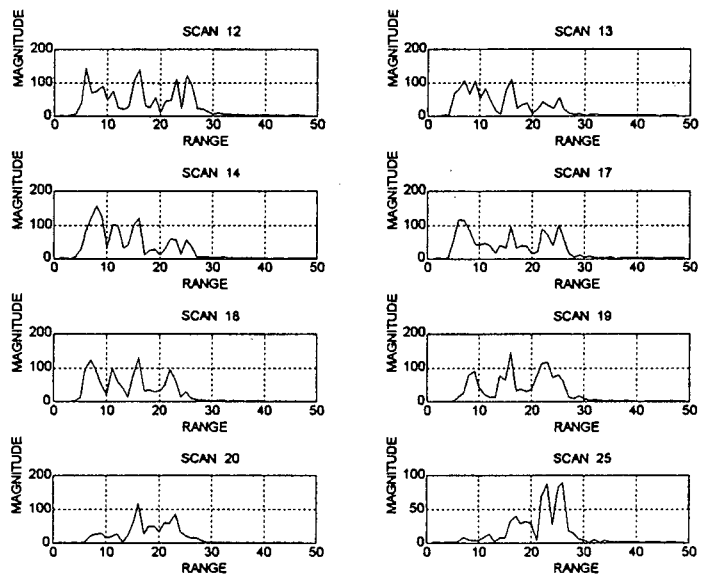


Figure 1.4: Interference between scattering centers for a cargo (tanker) ship. Consecutively numbered scans are separated in time by 2.2 seconds. One range profile is selected from each scan.

1.3.2 Radar Interference

More pronounced than the problem of sea clutter, was that of 'radar clutter', which is caused by interference from other radar sources. Fortunately, it was normally limited to one azimuth bin, making it relatively easy to avoid. Radar clutter is illustrated in figure 1.3.

1.3.3 Selection of Range Profiles

The number of profiles within a scan actually containing a target echo depends on the distance, size and orientation of the target. In theory, only one profile is required since the individual pulses are well correlated. Ideally, the profile with the highest signal-to-clutter ratio should be selected. This will occur when the target is completely enveloped by the radar beam. A simple approach would be to select the profile with the highest mean value. However this method has a tendency to select a profile containing radar interference.

The approach used in thesis was to perform a simple iterative test, based on the fact that each uncluttered bin is almost identical to its neighbour (besides a slight increase/decrease in energy content). The test is as follows:

- Select the bin with the highest energy content.
- Compare this profile to its immediate neighbour by performing a simple subtraction operation.
- If they are not identical (within an arbitrarily specified tolerance) move two bins along and try again.

  A proposed enhancement to the above approach involves the use of compound identification [8] whereby several profiles are fed into the classifier, and the average of the identifications is used to arrive at a single 'best' identification. The identification of a single profile can be considered a 'vote' for one of the targets, and the compound identification then corresponds to choosing the target with the most 'votes'. For the preliminary development described in this thesis, compound identification was not implemented.

1.4 The Fourier-Mellin Transform

As discussed in the previous section, a change in the target aspect angle will cause the range profile to fluctuate according to three basic principles. Only the first (a scaling of the range profile), can be dealt with in an intelligent manner. This is achieved through the use of the Mellin transform, which reduces any linear scaling of the input function to a phase component. It is therefore possible to simply discard this scale information by taking the magnitude of the Mellin transform. In addition, it is necessary to discard

any time-shift information. This is usually achieved through the use of the Fourier Transform.

Given a function $g(t), t \geq 0$, the continuous **Mellin transform** (MT) in one dimension is defined as

$$G(s) = \int_0^\infty g(t)t^{s-1}dt \tag{1.2}$$

There are actually several implementations of the Mellin transform in the literature [7] [46]. These have been studied by the author in [50]. The preferred approach was developed in detail in a paper by Zwicke and Kiss [46], in which it was called the Fourier Modified Discrete Mellin transform (F-MDMT).

The Modified Discrete Mellin Transform (MDMT) can be expressed in matrix notation as

$$\begin{bmatrix} G_M(\omega_1) \\ G_M(\omega_2) \\ \vdots \\ G_M(\omega_p) \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1,N-1} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2,N-1} \\ \vdots & \vdots & & \\ \phi_{P,1} & \phi_{P,2} & \cdots & \phi_{P,N-1} \end{bmatrix} * \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_{N-1} \end{bmatrix}$$

where:

$\phi_{ik} = \cos(\omega_i \ln k) - j \sin(\omega_i \ln k)$
$\omega_i, i = 1, P$ are arbitrary spectral components.
$\Delta_k = g_k - g_{k+1}$.
$g_k =$ the kth sample of the input data sequence.

By calculating the MDMT coefficients beforehand and storing them off-line, the equation is reduced to a differencing operation followed by matrix multiplication. This makes it extremely fast, and also amenable to a parallel (i.e., neural) implementation.

One addition by the author is worth mentioning. It concerns the choice of 'spectral components'. For the classification problem studied here, it was found that components all the way up to 6.25π continued to provide good discrimination between classes, while still exhibiting small in-class variations. This effectively increased the Hamming distance between classes, thereby reducing overlap at the decision surfaces separating each class.

An example of the F-MDMT in action on simulated data is illustrated in Figure 1.5 and Figure 1.6. Despite the within-class variation, a comparison of the first 40 spectral components of Figures 1.5 and 1.6 reveals several components that provide good discrimination, notably component 5, and components 25 to 35. The ships were modeled by assuming point reflectors spaced by dimensions corresponding to the major superstructural elements of the ship type. It was found that these simple models produced realistic looking data (compared to real measurements of similar ships).

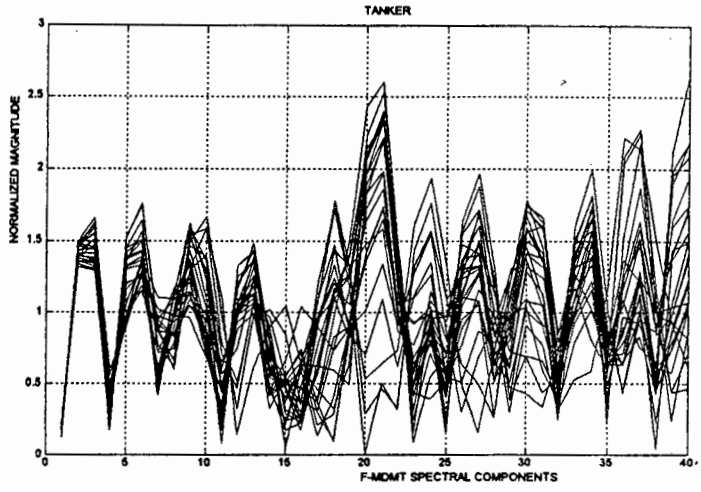


Figure 1.5: Spread of F-MDMT transforms of simulated 'Tanker' profiles (Aspect angle from 0-67.5 degrees, $SNR = 10dB$).

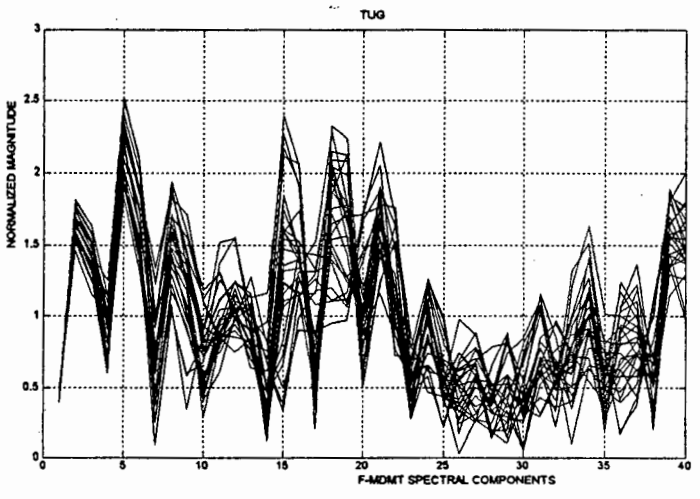


Figure 1.6: Spread of F-MDMT transforms of simulated 'Tug' profiles (Aspect angle from 0-67.5 degrees, $SNR = 10dB$).

1.4.1 Discussion

The main goal of Mellin transform pre-processing is to produce a set of features which are invariant to a linear scaling of the independent variable of the range profile. The idea is that the features extracted from the MT are insensitive to target aspect angle. This is of course a simplification, since only the first of the basic principles mentioned in Section 1.1 (a scaling of the range profile), can be dealt with in an intelligent manner. To recap, the various sources of noise will be as follows:

- The scaling of the range profile with aspect angle is not linear but cosinusiodal. This scaling only approximates to linear when the target is close to head-on to the radar line of sight.
- For ships with low aspect ratios (length divided by width), the effects of range migration will be quite pronounced. This is illustrated graphically in figure 1.7. If the two reflecting features are initially at the same range, and are separated by a distance B, an aspect change of $\Delta\theta$ will cause a change Δr in their down-range separation by approximately $B\Delta\theta$. If the distance between the two features is $10m$, a change in θ of 6 degrees will change their relative ranges by over $1m$.
- Even if the aspect does not change enough to cause migration, changes in the relative range of the different scatterers of a fraction of the radar wavelength will cause the radar profile to fluctuate due to the speckle effect. To illustrate, in the example shown in Figure 1.7, if Δr changes by just $\frac{1}{4}$ of a wavelength, the scatter from the two reflecting features may change from full constructive to full destructive interference, or *vice versa*.
- The reflecting features are not isotropic.

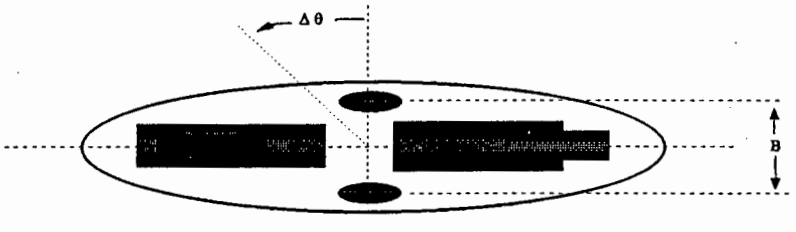


Figure 1.7: Range Migration

Chapter 2

Classification

This chapter begins with a brief overview and discussion of neural classification (section 2.1). It then moves on to review and comment on some of the author's earlier findings (Section 2.2, and then onto the author's latest work in Section 2.3.

2.1 Overview and Discussion

Target recognition of fixed signatures in stationary backgrounds is a straightforward task for which numerous effective techniques have been developed. However, Automatic Target Recognition (ATR) is a problem which involves extraction of critical information from complex and uncertain data where the target signatures or the backgrounds vary in an unlimited or unknown manner.

A number of ATR systems, based on traditional approaches such as signal processing, pattern recognition, and rule-based artificial intelligence (AI) techniques have been developed and tested on very limited data sets. Good classification performance has been reported [7].

However, in practice these efforts have produced high false alarm rates. Some of the key reasons for this are the non-repeatability of the target signature, experience with a very limited data base, and limited use of *a priori* information. Previous ATR systems have also shown a rapid degradation in performance when subjected to the problems of image and background noise, and target and environmental variations (e.g., translation, rotation, scale and context, etc.). Bearing this in mind, the United States Defense Advanced Research Projects Agency (DARPA), in conducting a study of neural networks, has selected ATR as one of the four application areas for evaluating neural network technology [7]. Specifically, neural network learning facilitates two main advances for ATR: automatic knowledge acquisition (unsupervised learning) and continuous system refinement:

- The use of unsupervised learning in system construction can eliminate the enormous amount of time necessary to derive rule-based data bases for targets and environments. Examples of such unsupervised feature extraction techniques are discussed below.

- Continuous system refinement can be incorporated to make changes

necessary to improve the performance of the recognition system. For example the Fuzzy Min-Max Neural Network [42], which is investigated in Section 2.3.1, can learn pattern classes in a single pass through the data, can add new pattern classes on the fly, and can refine existing pattern classes as new information is received.

Selection of appropriate target features is one of the most important tasks for ATR algorithm development. Because it is impractical to match a given input image representation with all the image templates of all possible targets and their variations, it is necessary to find a compact set of features which can represent the critical aspects of the target. Selection of the feature set is thus closely linked to the classification task.

Neural networks can assist in the feature selection task by providing hardware implementations of traditional feature detection algorithms. It has been shown that many well known image processing and feature selection algorithms can be efficiently implemented using neural networks [33]. This massively parallel hardware implementation means that the feature extraction system is very fast, compact and robust. For example Gabor, Fourier and Hough transforms have already been successfully implemented [34] using feedforward networks trained with the back-propagation algorithm.

Neural network technology can also contribute to the feature selection task through automatic discovery of clustered features. For example, Kohonen's self-organizing-maps are neural networks which consist of a set of interconnected adaptive units that have the ability to change their responses in such a way that they will adapt to represent the characteristics of the input signal [40].

Yet another major way in which neural networks can contribute to feature selection is in their ability to automatically integrate a diverse set of features. The back-propagation algorithm, combined with an appropriate training set, has been shown to be very effective here [33]. Subsequent analysis of the neural network weights can determine whether a particular feature has effectively participated in the classification task, or whether it can be removed without loss of overall performance.

In a similar vein, neural network technology can provide tools for developing expert systems, whereby learning algorithms such as the back-propagation algorithm, can be applied to the development of expert system rules. Several neural-network-based expert systems have been directly compared to rule-based expert systems developed for the same problem. In each case, the neural network-based system provided comparable or superior performance but at an enormous reduction in system development and execution time [33].

2.2 Phase 1

For the author's initial investigations into neural-based classification, the Back-propagation algorithm was pitted against Kohonen's Learning Vector Quantization algorithm. The real data was limited (in the sense that a complete range of azimuth angles and sea states was not available), and simulated data (described in detail in [50]) was used to facilitate thorough testing. The work is reviewed and discussed below.

2.2.1 A Brief Overview of Back-propagation

Back-propagation has enjoyed widespread popularity and is therefore well known. It uses an iterative gradient descent technique to minimize the error function equal to the mean square difference between the actual network outputs and the desired outputs. However there are some potential drawbacks, including slow training and the capability to converge to a local minimum rather than the desired global minimum. This is an inherent flaw of backward error propagation because it uses the steepest gradient descent algorithm. In this way it is possible not to find the optimum minimum square error solution, even for classes of input vectors which are linearly separable.

It is necessary to user-define the exact structure of the architecture, such as the number of hidden layers and the number of hidden nodes within each layer.

Networks with multiple hidden layers have been tested quite thoroughly on classification problems similar to the one studied in our work [39] [45]. In general they required a longer training time and exhibited reduced generalizing capabilities. It was not considered worthwhile to investigate this issue any further, and a single hidden-layered network was adopted as the standard configuration.

The problem of how many hidden nodes to use has not yet been solved analytically. Practical solutions range from making an educated guess to performing multiple simulations. Too many nodes tend to impair the generalizing capabilities of the network. Too few will make convergence impossible.

2.2.2 A Brief Overview of Learning Vector Quantization

Kohonen's LVQ classifier is often confused with his Self-Organizing-Map. However the two are in fact quite different. The Self-Organizing-Map is a sheet-like artificial neural network, the cells of which become specifically tuned to various input signal patterns or classes of patterns through an unsupervised learning process. In this way a 2-dimensional visual mapping of any hidden topological striker within a p-dimensional data set is created.

LVQ on the other hand, is essentially a clustering algorithm, (stemming from earlier clustering methods such as McQueen's *hard c means algorithm*) whereby sets of codebook vectors are updated through an iterative procedure to best represent their respective classes. When used in conjunction with the nearest prototype (NP) classifier, these codebook vectors then generate crisp or hard partitions of the data sets. At present there are actually four algorithms to choose from, namely; LVQ1, optimized LVQ1 (for fast initial convergence), LVQ2 and LVQ3. They are all based on different themes of the same underlying philosophy, and generally produce very similar results.

The algorithms require several parameters such as learning rate, size of an update neighbourhood, and a strategy to alter these two parameters during learning. The effect of these parameters is not rigorously defined. They must be determined by trial and error and varied from one data set to another. The number and initial positions of the codebook vectors is also fairly arbitrary. All of these factors have a significant effect on final classification accuracies however. Although Kohonen has developed a set of general rules and guidelines to follow in this regard, the process can hardly be termed scientific.

2.2.3 Results

Table 2.1 shows the percentage of correct classifications for the feedforward network trained with Back-propagation. Five simulated targets were used at different levels of signal to clutter (S/C). The data is described in detail in [50]. The targets were rotated from 0 to 60 degrees of head-on. The network consisted of 40 input nodes, 22 hidden nodes and 4 output nodes.

Zwicke and Kiss [46] expressed reservations about the noise susceptibility of the F-MDMT because there is a differencing operation inherent in its implementation. This explains the testing over a wide range of signal-to-clutter ratios. Its performance was found to be more than adequate, especially since the S/C ratios for the real data are normally $> 15dB$. Table 2.2 shows the performance of the LVQ classifier for the same data used above. The results were achieved after 10 000 iterations. Nine codebook vectors were used per class. The optimized LVQ1 algorithm [20] was used for the first 2000 iterations, and the LVQ3 algorithm was used thereafter. The window size was gradually reduced from a starting value of 0.3 to a final value of 0.05. The learning rate factor, α, was also reduced in rough correspondence to the window size, starting at .003 and ending off with a value of 0.001.

2.2.4 Discussion

Back-propagation is capable of forming more non-linear mapping functions than LVQ. The LVQ classifier forms a decision surface which tends towards

Table 2.1: Percentage correct classifications for the Back-propagation network (simulated data). Each target was rotated through ±60 degrees.

S/C	Tanker	Patrol	Tug	trawler	Average
16 dB	100	100	100	100	100
10 dB	100	100	100	100	100
8.2 dB	100	100	100	100	100
7.3 dB	96	92	92	92	93
6.0 dB	84	80	84	84	83
5.0 dB	84	64	60	92	75
4.3 dB	76	52	44	80	63
Average	91	84	82	93	88

Table 2.2: Percentage correct classifications for Kohonen's LVQ network (simulated data). Each target was rotated through ±60 degrees.

S/C	Tanker	Patrol	Tug	trawler	Average
16 dB	96	100	100	100	99
10 dB	96	100	100	100	99
8.2 dB	96	100	100	96	98
7.3 dB	92	84	92	96	91
6.0 dB	92	80	80	80	83
5.0 dB	88	72	56	80	74
4.3 dB	84	60	44	80	67
Average	91	85	82	90	87

a very close, although piecewise linear, approximation to the theoretically optimal Bayes decision surface. Despite these differences, it was found that both networks produced very similar results. Clearly the FM transform was performing the feature extraction task effectively, and the feature vectors for each class did not differ markedly from the multivariate Gaussian assumption made in the design of the Bayes classifier. In comparing the differences in performance of the two classifiers, we note the following:

- Back-propagation suffered from a training problem when presented with similar looking targets. Data for these targets tends to be ill posed (i.e. more than one output is required for the same input), and the algorithm usually ended up oscillating between the resulting local minima.
- LVQ training times were typically 50-100 times faster than for back-propagation.

Of course the training problem for BP could have been resolved by removing the guilty feature vectors. In addition, recognition times for both classifiers would be very similar once up and running. However this project is eventually aimed at a working system which should be both practical and flexible. The pattern classifier should be able to learn new classes and refine existing classes quickly and effectively. With BP or LVQ, this would involve retraining each time new information is added. In this respect LVQ would be more favorable.

2.3 Phase 2

Despite the advantages offered by LVQ, the classifier has definite shortcomings. An ideal pattern classifier should be capable of learning new classes and refining existing classes on the fly, without destroying old class information and without the need for complete retraining. Retraining (or off-line adaption) not only places heavy demands on memory, but leads to increasingly longer training times. In addition, an ideal classifier should have as few tuning parameters as possible, and if there are parameters, their effect on the system should be well understood.

2.3.1 The Fuzzy Min-Max network

With the aforementioned points in mind, the most recent work by this author has been concerned with an investigation into the Fuzzy Min-Max neural network [42], which was developed by Simpson. It belongs to the class of neural networks where decision boundaries are built by creating subsets of the pattern space, such as the related coulomb energy network and the hyperspherical attractor network.

Overview

There are several properties that the ideal pattern classifier should posses. According to Simpson, each of these has motivated a portion of the development of the fuzzy min-max classification network. These properties are:

- *On-Line Adaption:* A pattern classifier should be able to learn new classes and refine existing classes quickly and without destroying old class information. Many popular neural network and traditional pattern classification techniques utilize off-line adaption (including networks employing BP and LVQ). This leads to a common problem associated with neural network design known as the stability-plasticity dilemma, which concerns the design of neural networks that can remain plastic enough to learn yet still be able to stabilize and recall stored pattern information.

- *Nonlinear separability*: A pattern classifier should be able to build decision regions that separate classes of any shape and size.

- *Overlapping Classes:* A pattern classifier should have the ability to form a decision boundary that minimizes the amount of misclassification for all of the overlapping classes. In this respect the LVQ algorithm is effective since it tends towards the ideal Bayesian classifier [40].

- *Training Time:* A desirable property of a pattern classification algorithm is to be able to learn non-linear decision boundaries in a short training time. This is where many popular algorithms such as BP fall short, since they typically require thousands of passes through the data set.

- *Soft and Hard Decisions:* A pattern classifier should be able to provide both soft and hard decisions. A soft decision would be one that provides a value indicating the degree to which a pattern fits within a class. LVQ in its present form does not provide for this.

- *Tuning Parameters:* A classifier should have as few tuning parameters as possible. In addition, the effect these parameters have on the system should be well understood.

- *Nonparametric Classification:* A classifier should be able to describe the underlying distribution of the data in a way that still provides reliable class boundaries, even if *a priori* knowledge about the underlying probability density functions of each class is not available.

The network developed by Simpson is an extension of the original Adaptive Resonance Theory (ART) neural network developed by Carpenter and

Grossberg [52]. The major difference is that in Simpson's approach, fuzzy sets are used to define the pattern classes in a dataset. It is not the author's intention to provide an overview of fuzzy set theory in this report. Numerous texts exist on the subject [53] and a basic knowledge of fuzzy theory is assumed for the remainder of this chapter. The Fuzzy Min-Max Classifier operates by building fuzzy subsets of the n- dimensional pattern space. It achieves this through the creation of fuzzy hyperboxes, defined by pairs of min-max points and their corresponding membership functions (which define the degree to which a pattern fits within a hyperbox). The aggregation of several hyperboxes in I^2 is illustrated for a two-class problem in figure 2.1. Let each hyperbox fuzzy set B_j, be defined by the ordered set

$$B_j = \{X, V_j, W_j, f(X, V_j, W_j)\} \ \forall X \in I^n \tag{2.1}$$

Using this definition of a hyperbox fuzzy set, the aggregate fuzzy set that defines the k_{th} pattern class C_k is defined as

$$C_k = \bigcup_{j \in K} B_j \tag{2.2}$$

where K is the index set of those hyperboxes associated with class k. Note here that the union operation in fuzzy sets is typically the maximum of all the associated fuzzy set membership functions. The learning algorithm developed by Simpson allows overlapping hyperboxes from the same class and eliminates the overlap between hyperboxes from separate classes. Using this configuration, it is possible to define crisp class boundaries as a special case. These class boundaries are defined as those points where the membership values are equal.

Hypercube Membership Function

The membership function for the j^{th} hyperbox $b_j(A_h)$, $0 \leq b_j(A_h) \leq 1$, must measure the degree to which the h^{th} input pattern falls outside of the hyperbox B_j. It is given by

$$\begin{aligned} b_j(A_h) \quad = \quad & \frac{1}{2n} \sum_{i=1}^{n} [max(0, 1 - max(0, \lambda min(1, a_{hi} - w_{ji}))) \\ & + max(0, 1 - max(0, \lambda min(1, v_{ji} - a_{hi})))] \end{aligned} \tag{2.3}$$

where $A_h = (a_{h1}, a_{h2}, \ldots, a_{hn}) \in I^n$ is the h^{th} input pattern, $V_j = (v_{j1}, v_{j2}, \ldots, v_{jn})$ is the minimum point for B_j, $W_j = (w_{j1}, wj2, \ldots, w_{jn})$ is the maximum point for B_j, and λ is the sensitivity parameter that regulates how fast the membership values decrease as the distance A_h and B_j increases.

Network Topology

The neural network that implements the Fuzzy Min-Max classifier is shown in Figure 2.1. Each F_B node in this three-layer neural network represents a

hyperbox fuzzy set where the F_A to F_B connections are the min-max points and the F_B transfer function is the hyperbox membership function defined by 2.3. The min points are stored in the matrix V and the max points are stored in the matrix W. The connections are adjusted using the learning algorithm described in Appendix A. A detailed view of the $j^{th} F_B$ node is shown in Figure 2.2. The connections between the F_B and F_C nodes are binary valued and stored in the matrix U. The equation for assigning the values to the F_B to F_C connections is

$$u_{jk} = \begin{cases} 1 & \text{if } b_j \text{ is a hyperbox for class } c_k \\ 0 & \text{otherwise} \end{cases}$$

where b_j is the j_{th} F_B node and c_k is the k^{th} F_Cw node. Each F_C node represents a class. The output of the F_C node represents the degree to which the input pattern A_h fits within the class k. The transfer function for each of the F_C nodes performs the fuzzy union of the appropriate hyperbox fuzzy set values.

The algorithm operates by grouping sets of n-dimensional hyperboxes defined by pairs of min-max points in conjunction with fuzzy membership functions in the decision space. The full algorithm can be found in Appendix A. Simpson summarizes the three step learning process as follows:

- *Expansion*: Identify the hyperbox that can expand and expand it. If an expandable hyperbox cannot be found, add a new hyperbox for that class.
- *Overlap Test*: Determine if any overlap exists between hyperboxes from different classes.
- *Contraction*: If overlap between hyperboxes that represent different classes does not exist, eliminate the overlap by minimally adjusting each of the hyperboxes.

This process is ideally suited to on-line adaption, since new classes can be added without retraining, and existing classes can be refined on an ongoing basis. There are two tuning parameters, one for limiting the p-dimensional extent of each individual hyperbox, and the other for defining the slope of the fuzzy membership function. Both affect the classifier in an intuitively predictable manner. When the size of the hyperboxes is set to zero, the classifier collapses to the k-nearest neighbour classifier.

2.3.2 Results

The Fuzzy Min-Max classifier was coded and implemented in Matlab. The source code can be found in Appendix A. Separate training and testing sets of real data were selected in order to test as rigorously as possible the classifiers' abilities to generalize. Where possible, training and testing

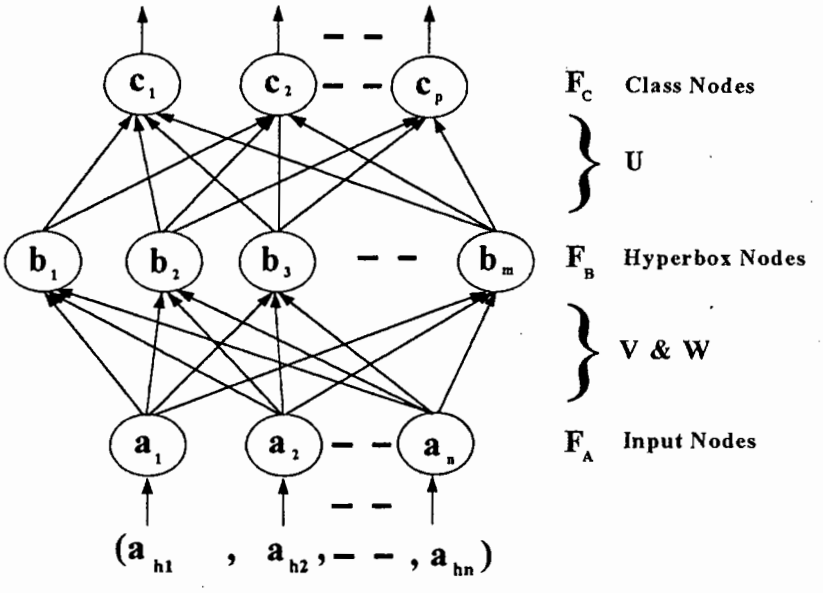


Figure 2.1: The three-layer neural network that implements the Fuzzy Min-Max neural network. The input layer $F_A = (a_1, a_2, \ldots, a_n)$ has n processing elements, one for each of the n dimensions of the input pattern A_h. There are two sets of connections between each input node and each of the m hyperbox fuzzy set nodes found in the layer $F_B = (b_1, b_2, \ldots, b_n)$. These dual connections are adjusted using the Fuzzy Min-Max classification learning algorithm. There are two sets of connections that emanate from F_A and abut the j^{th} F_B nodethe min vector V_j and the max vector W_j. The connections between the F_B nodes and the p output nodes $F_C = (c_1, c_2, \ldots, c_n)$ are binary valued and are determined as each F_B node is added during learning. Each FC node represents a pattern class.

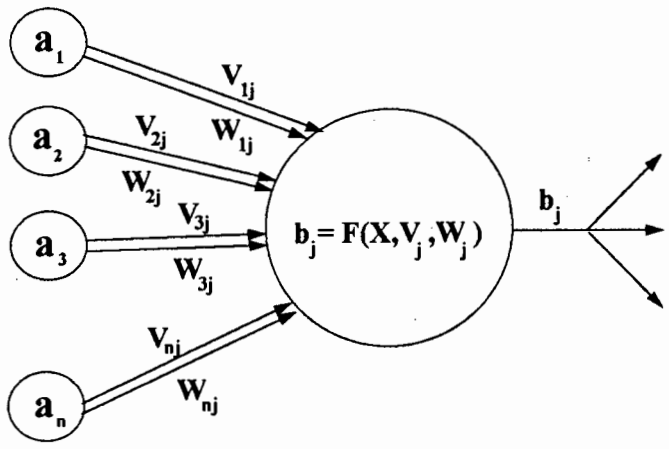


Figure 2.2: The implementation of a hyperbox and its associated membership function as a neural network assembly is shown for the j^{th} F_B node b_j. The input nodes accept each dimension of the h^{th} input A_h. There are two connections from each input node to the output node, one connection represents the min value for that dimension and the other connection represents the max value for that dimension. The connections between i^{th} input node and the j^{th} hyperbox node are v_{ji} and w_{ji}. The min point for the j^{th} F_B node is the vector $V_j = (v_{j1}, v_{j2}, \ldots, v_{jn})$ and the max point is $W_j = (w_{j1}, w_{j2}, \ldots, w_{jn})$. Assuming the input pattern is A_h, b_j's output value $y_i = y_j(A_h)$ is computed using equation 2.3. This entire neural assembly represents a hyperbox fuzzy set.

Table 2.3: Percentage correct classifications for Kohonen's LVQ network (real data).

TARGET NO.	ENTRIES	% ACCURACY
1	110	100.00
2	66	84.73
3	22	72.73
4	55	100.00
AVERAGE		93.68

Table 2.4: Percentage correct classifications for Simpson's Fuzzy Min-Max network (real data).

TARGET NO.	ENTRIES	% ACCURACY
1	110	96.26
2	66	92.42
3	22	63.63
4	55	100.00
AVERAGE		93.28

was performed at different aspect angles. Once again, in all cases separate scans were used. The reader will recall that this was considered particularly important since the scan-to-scan correlation was generally poor and presenting the classifier with unseen scans was considered a harsher test than presenting it with unknown target azimuth angles.

Table 2.3 summarizes the classification results for the LVQ classifier over the aspect range ± 70 degrees of head-on. In order to make the comparison with Simpson's network fair, only the default values recommended in Kohonen's LVQPAK programming package were used [51]. Table 2.4 summarizes the classification results for the Fuzzy Min-Max classifier over the same aspect range. The expansion parameter was 0.15 and the membership parameter was 4.

2.3.3 Discussion

For both classifiers, better accuracies were obtainable through trial and error adjustment of the user-defined parameters, although the process was simpler and more intuitive for Simpson's network. The final positions of the codebook vectors for LVQ varied and depended on their initial placements. In this way the performance of the classifier was not entirely pre-

dictable. The expansion/contraction process of the Min-Max classifier was disappointingly slow and is roughly proportional to the number of created nodes. It is therefore much slower at initial training. However, since it is able to create and update classes without retraining, the speed at which it learns new information is superior, and increases as the size of the overall training set increases. The classification accuracies achieved by the Fuzzy Min-Max network were slightly lower than for Kohonen's LVQ network. This difference is more marked when the optimal settings for the various LVQ tuning parameters are used (found through trial and error). Overall the Min-Max classifier was considered to be more favorable for a final working system, mainly because of its ability to learn and update itself continuously and automatically, in a changing and unpredictable maritime environment.

Chapter 3

Synthetic Range Profile Processing

Despite the good results achieved in Chapter 2, recall that it was felt that they would be somewhat degraded for a real radar system that is required to classify a large array of ship targets. This is because the low resolution of the radar system (12m) makes it difficult to discriminate between targets of a similar size. This applies particularly to smaller targets, whose range profiles tend to show up as a single spike on the current system. To develop a truly effective non-cooperative ship target recognition system, improved range resolution is required.

The approach investigated in this thesis involves the use of synthetic range profiles. A synthetic range profiling system is of particular interest because the University of Cape Town is already actively involved in the development of such a system in conjunction with Reutech Radar Systems. This system is currently under development in an aircraft recognition context, but it would be interesting to apply it to the ship target recognition problem as well.

A brief theoretical overview of synthetic range profiling is provided, followed by various practical observations, including a look at the effects of matched filtering as well as the effects of target velocity. There is also a practical overview of SRP waveform design.

3.1 Literature Review

Synthetic range profiles and their application to target recognition have been investigated by Wehner [54]. Numerous references to synthetic range profiling can also be found in the UCT libraries. Much of this literature has been produced as part of the project aimed at developing an SRP system for aircraft recognition. The research includes several earlier phases [29] [30] [5] [6] as well as more recent work by the author [31] [3] [4]. No references were found relating to the use of synthetic range profiles in the context of ship target recognition.

3.2 Theoretical Overview

Synthetic range profiles (SRP's) contrast to ordinary range profiles in that the target's signature is not measured directly. Instead, a discrete frequency signature that is the frequency domain equivalent of the target's time domain range profile is generated by transmitting narrowband pulses stepped in frequency from pulse to pulse. The additional processing that is required in order to synthesize the time domain signatures is justified by the practical design problems that this approach avoids.

With a stepped frequency waveform, it is possible to achieve a similarly narrow pulse width with a lower instantaneous receiver bandwidth and lower A/D sampling rate. This is made possible by sequentially stepping the carrier frequency over several pulses, instead of within one, single pulse. As a result each individual pulse can be of longer time duration, (in fact the pulse length must be long enough to cover the entire target). Synthetic range profile processing therefore makes it possible to achieve high range resolution with many types of existing search and track radars, which can then be used for target recognition and target imaging in addition to their normal search and track functions.

To understand how it is possible to extract range information from the amplitude and phase of a reflected set of discrete frequencies, let us begin by considering a set of m sources (point reflectors) of horizontally polarized radio frequency (RF) energy positioned in a homogenous media some distance from a radar receiver.

The noiseless signal received for the m^{th} reflector at the i^{th} frequency will then be given by :

$$E_i = \rho_m e^{2\pi f_o \tau_m} e^{2\pi \Delta f \tau_m i} \tag{3.1}$$

where ρ_m is the amplitude, τ_m is the range delay, given by $2r_m/c$, r_m is the range in meters to the m^{th} reflector, and f_o is the start frequency.
It is at this point that the author would like to introduce three new terms, namely the SRP complex amplitude, electrical range and SRP projected wavenumber. Although not found in any of the other literature, they are used here because they help to clarify the mechanism of synthetic range profiles and how they relate to other more traditional forms of frequency domain analysis.

The first term, which we will call the *complex amplitude* (α_m) for the m^{th} reflector, is given by

$$\alpha_m = \rho_m e^{2\pi f_o \tau_m} \tag{3.2}$$

The second term, which we will call the *electrical range* (Φ_m) of the m^{th} reflector, corresponds to the phase shift brought about by each frequency step Δf. It is given by

$$\Phi_m = 2\pi\Delta f \tau_m \tag{3.3}$$

Using the above definitions, it is possible to simplify equation 3.1 to that of equation 3.4.

$$E_i = \Re[\alpha_m e^{\Phi_m i}] \tag{3.4}$$

The ratio $\Phi/\Delta f$ is the third new term. We will call it the *SRP projected wavenumber*, and be denote it by K_{srp}.

From the above equation it is clear that as the operating frequency varies by discrete steps Δf, the received vector for the discrete reflector m will rotate by an amount equal to the electrical range Φ_m. In this way, by determining the SRP projected wavenumber K_{srp} (i.e., the rate at which the electrical range varies with frequency), it is possible to determine the distance to the m^{th} reflector.

3.3 Range Extraction

The reader will immediately notice that the problem of determining the SRP projected wavenumber is directly analogous to finding the projected wavenumber K in linear phased array theory, as well as, of course, the angular frequency ω in spectral analysis theory. It should therefore be clear that extracting range information from the SRP profile is a matter of transforming the received data from the frequency domain into the spatial domain. This involves taking the inner product of the received data set $\boldsymbol{E}$ with a basis function $\boldsymbol{H}$. Since the number and position of the respective reflectors is not known, as well as the exact nature of noise introduced into the received data set, various assumptions have to be made in choosing $\boldsymbol{H}$. This is by no means a trivial task, and the whole of section 4 is devoted to this interesting problem.

3.4 Practical Analysis

In this section we will concern ourselves with a more practical analysis of synthetic range profile processing, taking into account various factors such as the effects of the system hardware (such as a matched filter), as well as the effects of target motion.

3.4.1 The Effect of Target Velocity

We begin by exploring the effects of target motion on the formation of the profile. Consider a target moving at constant velocity v. The target range will change with each pulse as

$$r_i = r_o + viT \tag{3.5}$$

The phase of the SRP complex amplitude then becomes (from equation 3.2)

$$\frac{4\pi\Delta f}{c}(r_o + viT) = \frac{4\pi\Delta f r_o}{c} + \frac{4\pi\Delta f viT}{c} \tag{3.6}$$

and the phase of the electrical range becomes (from equation 3.3)

$$\frac{4\pi f_o}{c}(r_o + viT) = \frac{4\pi f_o r_o}{c} + \frac{4\pi f_o viT}{c} \tag{3.7}$$

The 2^{nd} term in the phase of the SRP complex amplitude represents the Doppler frequency shift due to target motion. When transforming from the frequency to the spatial domain, this frequency shift is mistaken for a shift due to range, resulting in a shifting of the target from its true range position. This range shift is easily calculated as $\frac{vTf_c}{\Delta f}$ where f_c is the center frequency of the step frequency waveform.

The 2^{nd} term in the phase of the electrical range describes the interaction of the stepped frequency waveform with the target motion. The resultant spreading of the stepped frequencies has several negative side-effects, including reduced range resolution, range accuracy and signal to noise ratio.

These negative effects due to target velocity can be eliminated by compensating for the shift in phase before transformation to the spatial domain (although in practice the range profile is still partially degraded due to inaccuracies in the estimation of the target velocity).

Velocity compensation is of importance in aircraft radar systems due to the high velocities that these targets typically attain. However for the ship target recognition case, the effects of target velocity will be negligible. For example, based on the findings in [6], it can be shown that a target velocity of $10ms^{-1}$ radially towards the radar will cause a shift of approximately $30cm$ in the range domain. It is therefore concluded that velocity compensation will not be required for the ship target radar system (this is just as well since the velocity of maritime vessels is difficult to determine accurately).

3.4.2 Waveform Design

Waveform design describes the process of selecting the radar parameters in order to achieve an optimal configuration for the task at hand. Because of the large number of parameters, and their numerous inter-dependencies, the resultant search space is very large. An example of a typical set of radar parameters is given in table 3.1 below. These parameters were taken from an SRP based radar system that is currently under research and development by U.C.T. and Reutech Radar Systems.

Table 3.1: Radar Parameters for the SRP Based Radar System

Start Frequency		1363.75MHz
Stop Frequency		1365.0MHz
Frequency Step	Δf	1.875MHz
Number of Steps	N	55
Radar Bandwidth	B	$54 * 1.875 = 101.25MHz$
Radar Resolution	$\Delta R = c/2B$	1.48m
FFT Range Resolution	$\Delta R_c = c/2N_c\Delta f$	1.25m
ADC Sample Rate	f_s	3.75MHz
Range Bin Length	$R_{bin} = c/2f_s$	1.48m
Radar Compressed Pulse	τ_c	275ns
Radar Pulse Length	$c\tau_c$	82.5m
Pulse Resolution	$c\tau/2$	41.225m
Antenna Rotation Rate	$30rpm$	
Azimuth Beamwidth	$10deg$	
Vertical Beamwidth	$cosec^2$ (Transmit)	
Vertical Beamsidth	$10deg$ (Receive)	

Below is a list of considerations to aid in the selection of the optimal radar parameters:

- Three scenarios are possible in the relationship between the unambiguous range. R_u and the range bin width R_b. Firstly, if $\tau\Delta f$ is unity, the original range bin occupies the entire unambiguous range such that $R_u = R_b$. The second scenario occurs if $\tau\Delta f$ is less than unity. Although the range resolution suffers as a result, this setting is recommended for the recognition of targets with high velocities, since it facilitates cancellation of the clutter space by making room for non-stationary targets to be shifted to the right, *out* of the clutter-space [55]. The third scenario, $\tau\Delta f$ greater than unity is not of any practical value since it causes the range profile to wrap around (aliasing), resulting in distortion.

- To avoid the target wrapping around (aliasing), the unambiguous range R_u should be greater than the maximum target extend L_t. In addition, since R_u should be greater than the pulse width, a tighter constraint is that for good detectability the pulse width should be large enough to encompass the entire target length such that

$$L_t < \frac{c\tau}{2} \tag{3.8}$$

- The fraction of the range domain containing clutter is given by

$$\frac{c\tau/2}{R_u} = \tau\Delta f \tag{3.9}$$

  Choosing a low value for the product $\tau\Delta f$ will increase the cult free space.

3.4.3 Effects of Matched Filtering and A/D Conversion

Up to this point in the discussion it has been convenient to ignore the exact nature of the amplitude term ρ_m in equation 4.3. For a real radar system with pulse length T_p and interpulse period (PRI) T, ρ_m will be of the form given by equation 3.10

$$\rho_m = A(f_i)rect(\frac{t - iT - \tau}{T_p}) \tag{3.10}$$

Where

$$rect(\tau) = \begin{cases} 1 & \text{if } |\tau| < 1/2 \\ 0 & \text{if } |\tau| > 1/2 \end{cases}$$

Once the radar receiver removes the carrier frequency, performs matched filtering and creates digitized $I-$ and $Q-channel$ samples, ρ_m can be described by equation 3.11

$$\rho_m = G(t, f_i)\Lambda(\frac{t - iT - \tau}{T_p}) \tag{3.11}$$

Where $G(t, f_i)$ represents the time and frequency dependency of the transmitted pulses $A(f_i)$, as well as the antenna and receiver gain, and

$$\Lambda(\tau) = \begin{cases} 1 - |\tau| & \text{if } |\tau| < 1 \\ 0 & \text{if } |\tau| > 1 \end{cases}$$

It is important to understand the effect of this matched filtering and A/D sampling on the received signal in the context of SRP processing. This is because it causes some additional complications to the standard time domain pulsed radar range profile. These complications are demonstrated in Figures 3.1 and 3.2, which show a typical case of a received signal from a target consisting of three discrete point reflectors. The reader should observe the following:

- The received energy from the target is contained in at least three range bins. This phenomenon gives rise to some interesting side-effects. For example, in the first figure, notice that only one of the samples (the middle-most one) contains energy from all three point reflectors. Only the range bin associated with this sample will show all three reflectors. By contrast, the range bin associated with the sample to the right will only show the presence of the last point reflector.

- Figure 3.2 shows the output for the same set of reflectors as in Figure 3.1, but with the A/D sampling started slightly earlier. In this case, none of the range bins will show all three reflectors. Although the first two range bins will both be of similar energy content and will both show two point reflectors, the first will contain information on the first two reflectors, while the second will contain information on the last two reflectors.

- Both figures also demonstrate that the magnitude response of each reflector is heavily influenced by the position of the A/D sample in the received signal.

The above points have important repercussions in the context of ship profile classification. For example, the last point demonstrates that the information content of the height of the range profile peaks is low. The most valuable information content is in the *spacing* of these peaks. In addition, the second point rules out the idea of averaging the energy in the central range bin with that of adjacent range bins to increase the SNR. This is because we have no way of intelligently aligning them.

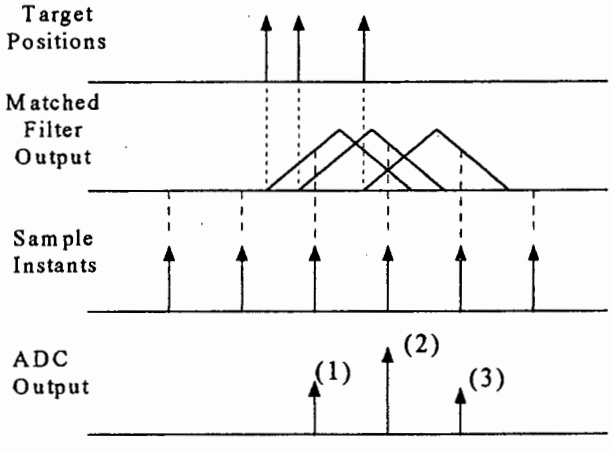


Figure 3.1: Scenario 1. The effect of matched filtering and A/D sampling on the received signal. Targets simulated as point reflectors of zero range extent.

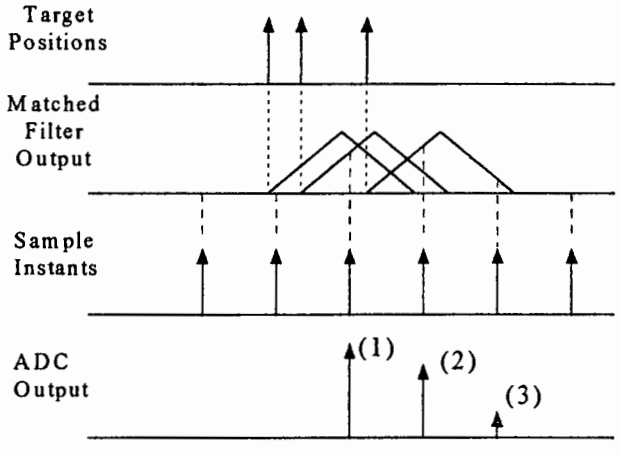


Figure 3.2: Scenario 2. The effect of matched filtering and A/D sampling on the received signal. Targets simulated as point reflectors of zero range extent.

3.4.4 Variable Time Shifts

The frequency domain data contains a variable phase shift which depends on the target's range and velocity. It is not possible to compensate for this accurately, so that the resulting range profiles – usually calculated from the frequency domain data by Inverse Discrete Fourier transform – possess a variable time shift. It is therefore necessary to employ shift invariant pre-processing if reliable recognition is to be achieved (the reader will recall that shift invariant pre-processing was also required for the time domain range profiles studied in Section 2. In that case the F-MDMT was used, and there is no reason why it cannot be applied to the synthetic range profiling case as well).

Chapter 4

Super Resolution Techniques Applied to Range Estimation

As mentioned earlier, the problem of extracting spatial or range information from the frequency domain SRP data is by no means a trivial task. The reader will recall that it involves taking the inner product of the received data set $\boldsymbol{E}$ with a basis function $\boldsymbol{H}$. However since the number and position of the respective reflectors is not known, as well as the exact nature of noise introduced into the received data set, several assumptions have to be made in choosing $\boldsymbol{H}$. This leads to the numerous techniques that are discussed in this section. The objective is to identify those algorithms which are most effective and best suited to the ship target recognition scenario. In the context of SRP neural based recognition, the ideal algorithm should possess the following properties.

- It should achieve the highest resolution possible from the data (i.e. it is assumed that an improvement in resolution will improve the ability to resolve differences between similar targets).
- The algorithm should be fast. In addition, it should be suited to a neural based implementation
- It should be robust and noise resilient.
- It should ideally reduce the size of the data sets by performing some kind of feature extraction.

As is often the case in engineering, the study was conducted by analyzing the theoretical and empirical developments in other related fields and then adapting this knowledge to the problem at hand. This analogy is proven in section 4.1 below, thus making it possible to move beyond the traditional FFT approach to range estimation, to the interesting and promising array of techniques known as super resolution algorithms. Of particular interest in this regard are the various eigendecomposition techniques such as the Total Least Squares method, the MUSIC method, and the Principal Eigenvector method.

4.1 Frequency Estimation, Direction Finding and Range Estimation

It is well known that the problem of estimating the frequencies of complex sinusoids in the field of spectral analysis, and the problem of estimating the direction of arrival (DOA) of incident plane waves on a linear phased array antenna are mathematically analogous.

To clarify this, first consider the standard model used in spectral analysis for a received signal $u(i)$:

$$u(i) = \sum_{l=1}^{L} \alpha_l e^{j\omega_l i} + v(i), \quad i = 0, 1, \ldots, N-1 \tag{4.1}$$

Here L is the number of complex sinusoids, $\alpha_1, \alpha_2, \ldots, \alpha_L$ are their complex amplitudes, and $\omega_1, \omega_2, \ldots, \omega_L$ are their angular frequencies respectively. $v(i)$ is a complex sample of additive noise, and N is the total data length.

Next consider the standard model (defined by Haykin) used to describe the received signal at the ith sensor of a linear phased array, at time k, due to the combined action of L incident waves:

$$u(i,k) = \sum_{i=l}^{L} \alpha_l e^{j\Phi_l i} + v(i,k), \quad i = 0, 1, \ldots, M \tag{4.2}$$

Here α_l and Φ_l are the complex amplitude and electrical angle [1] of the i^{th} incident plane wave, respectively, and $v(i,k)$ is the additive contribution of noise. Equation 4.2 is called the baseband form of the received signal, because all the incident plane waves are assumed narrow-band and centered around a common frequency f_o.

The mathematical analogy between temporal processing and spatial processing is clearly apparent by comparing equations 4.1 and 4.2. Both these topics are extremely well researched and documented, and it is therefore advantageous to derive a similar mathematical analogy for the range estimation (synthetic range profiling) case.

In this light consider what will be termed the *frequency analog* of the temporal and spatial estimation problems. To make the analogy clear, some preliminary groundwork is first required.

To begin, consider a set of sources (point reflectors) of radio frequency (RF) energy: The source output for the m^{th} reflector at the start frequency f_o may be denoted by $\rho_m cos(2\pi f_o \tau_m)$, where ρ_m is the amplitude and τ_m

[1] Given a linear array of sensors for spatial processing, the phase difference between the signals received by the 1^{st} and 2^{nd} sensor, and between every other pair of adjacent sensors in the array, is $2\pi(d/\lambda)sin\theta$, where θ is the angle of incidence of a single plane wave impinging on the array, and λ is the wavelength of the propagating disturbance. This quantity is called the *electrical angle of the incident wave*. The ratio Φ/d is called the *projected wavenumber*, and is denoted by K.

Table 4.1: Analogy between temporal processing, frequency processing and spatial processing

Time Series	Frequency Series	Space Series
Time (sec)	Frequency (Hz)	Distance (m)
Sampling duration, T (sec)	Frequency step, Δf (Hz)	Sensor spacing, d (m)
Angular frequency, ω (rad/sec)	Wavenumber, K_{SRP} (rad/Hz)	Wavenumber, K (rad/m)

is the range delay, given by $2r_m/c$, where r_m is the range in meters to the m^{th} reflector. The noiseless signal received for the i^{th} frequency is then given by

$$\rho_m cos(2\pi f_o \tau_m + 2\pi \Delta f \tau_m i) = \Re[\alpha_m e^{\Phi_m i}] \tag{4.3}$$

Where α_m, is the *complex amplitude* and Φ_m the *electrical range* of the m^{th} reflector. Although both terms were introduced in section 3.2, their definitions are repeated below for the sake of continuity:

$$\alpha_m = \rho_m e^{2\pi f_o \tau_m} \tag{4.4}$$

$$\Phi_m = 2\pi \Delta f \tau_m \tag{4.5}$$

The reader will recall that the ratio $\Phi/\Delta f$ is called the SRP projected wavenumber K_{srp}. It is directly analogous to the projected wavenumber K in linear phased array theory, as well as, of course, the angular frequency ω in spectral analysis theory.

It is now possible to define the complete model for the range estimation/ SRP case. The received signal over N frequencies, at time k, due to the combined action of M point reflectors, is given by

$$u(i,k) = \sum_{m=1}^{M} \alpha_m e^{\Phi_m i} + v(i,k), \quad i = 1, \ldots, N \tag{4.6}$$

where v(i,k) represents the additive contribution of noise. By comparing equations 4.1, 4.2 and 4.6, the analogy between temporal processing, spatial processing, and frequency processing is readily deduced and is summarized in Table 4.1.

It should be noted that there are basic differences between the temporal, spatial and frequency model. The time series described in equation 4.1 is serial in nature and continues for a total observation interval N. The space series of equation 4.2 is formulated on a snapshot-by-snapshot basis,

with each snapshot corresponding to a particular instant in time. The frequency series of equation 4.6 is formulated by recording the response for each frequency over a given observation interval. In this way the frequency series has similarities to both the temporal and the spatial series. The main point to note is that since each frequency is recorded over a time interval, it is also possible to formulate a set of 'snapshots' for the frequency series (although of course each 'snapshot' in this case does not correspond to a particular instant in time, but to a *set* of instances in time.). It should be noted that other than the addition of the extra variable k in the spatial and frequency series, these differences are of no analytic significance to the algorithms that will be investigated.

4.2 Selecting the Optimal Model

The mathematical analogy defined in section 4.1 opens up the vast array of techniques that have been proposed for spectral and DOA processing. Most of the approaches are problem specific and depend quite heavily on how closely the assumed model matches the real data record.

The techniques under investigation will be analyzed from a modeling perspective. In this context, the process becomes one of selecting an appropriate model for the data, estimating the unknown parameters, and then extracting the relevant range and magnitude information. It is hoped that this unified approach will allow for a structured and logical development to the task of selecting the optimal method.

Before beginning, the discussion will be facilitated by expressing equation 4.6 in matrix form as

$$V = \Psi P \tag{4.7}$$

where

$$V = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{M-1} \end{bmatrix}, P = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{M-1} \end{bmatrix},$$

$$\Psi = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{\Phi_0} & e^{\Phi_1} & \cdots & e^{\Phi_{M-1}} \\ \vdots & \vdots & & \\ e^{\Phi_0[N-1]} & e^{\Phi_1[N-1]} & \cdots & e^{\Phi_{M-1}[N-1]} \end{bmatrix},$$

α_m and Φ_m are defined in equations 3.2 and 3.3 respectively.

4.3 Least Squares (LS) Modeling

The process of fitting a model to the data will be achieved through the method of least squares.

In determining a least squares fit to the data, the measured process v_n is modeled by an estimated process $\hat{v_n}$ which is represented by the vector equation

$$\widehat{V} = \Psi \widehat{P} \tag{4.8}$$

where $\widehat{P}$ is a vector of weights and Ψ is a matrix representing the assumed model for the process.

$\widehat{P}$ is then determined by minimizing the total squared estimation error,

$$\sum_{i=0}^{N-1} |v_i - \hat{v_i}|^2 \tag{4.9}$$

to give the minimum Euclidean norm solution

$$\widehat{P} = \Psi^{\#} V \tag{4.10}$$

where $\Psi^{\#}$ is the Pseudoinverse of A, given by

$$\Psi^{\#} = (\Psi^H \Psi)^{\#} \Psi^H \tag{4.11}$$

and V is simply the original data vector

$$V = \begin{bmatrix} \hat{v}_0 \\ \hat{v}_1 \\ \vdots \\ \hat{v}_{N-1} \end{bmatrix}$$

and H denotes Hermitian transpose.

4.4 The Maximum Likelihood Method

The maximum likelihood (ML) estimator is the most general and robust approach to frequency estimation. Unlike other methods where a data model is pre-selected, the ML estimator makes no such underlying assumptions. ML identifies the probability of the data given a set of model parameters as the *likelihood* of the parameters given the data. It is therefore possible to obtain the optimal parameters by maximizing this likelihood. ML always performs better than linear prediction methods when the noise is white Gaussian and the SNR is low. However ML suffers the drawback of being computationally involved, making it unsuitable in many applications.

This is especially true for radar target recognition, where the available processing time is normally very limited. For this reason it is better to use a pre-selected model for the SRP problem under investigation here. The following sections will evaluate which pre-selected model gives the best performance.

4.5 The Harmonic or Fourier Model

The dominant model in the field of spectral analysis is the *harmonic* model, introduced by Jean-Baptise-Joseph Fourier.

According to the standard form of the harmonic model, commonly termed the *periodogram* spectral estimate [2] , an N-point data record x_n is represented by a set of complex sinusoids according to the equation

$$\hat{x}(n\Delta t) = \sum_{k=1}^{N-1} A_k e^{(j2\pi f_k n\Delta t)}, n = 0 \ldots N \tag{4.12}$$

where $\hat{x_n}$ is the model sequence, Δt is the sample interval in seconds, A_k is the amplitude of the complex exponential and f_k is the sinusoidal frequency in Hz.

By making the substitutions $v_i = x_n$, $i\Delta f = n\Delta t$, $\rho_m = A_k$ and $\tau_m = f_k$ equation 4.12 can be made to take on the form of equation 4.6. This match between the proposed model and the underlying theoretical model of the data is intuitively appealing. The harmonic model does however impose certain constraints on the ideal theoretical model proposed in equation 4.6. Specifically, the number of scattering centers is assumed to be equal to the number of data points, N, and their positions in space are assumed to be harmonically related such that $\tau_m = m\Delta\tau$, where $\Delta\tau = 1/N\Delta f$. In other words, by simply pre-assigning values, the harmonic model avoids having to determine the number and positions of the scattering centers

This implicit assumption makes each column (row) vector of $\widehat{\Phi}$ in equation 4.7 orthogonal to all the other column (row) vectors so that

$$(\widehat{\Psi}^H \widehat{\Psi})^{-1} = \frac{1}{N} I$$

where I is the identity matrix. It then becomes a simple matter to solve for the complex amplitude vector $\widehat{P}$:

$$\widehat{P} = \frac{1}{N} \widehat{\Psi}^H V$$

A major advantage of the harmonic model is that it can be implemented by means of the fast Fourier transform (FFT). The FFT is computationally

[2] Another popular implementation of the Harmonic model, popularized by Blackman and Tuckey, uses the Wiener-Khinchen theorem to estimate the Power Spectral Density (PSD) via an autocorrelation estimate.

efficient and produces robust results for a large class of signal processes. There are however several performance limitations for the data sequence under consideration. The most prominent of these will be limited range resolution, i.e., the ability to distinguish between two closely spaced scattering centers. From the inherent assumption that the scattering features are harmonically positioned in space, it is easy to see that the range resolution in meters will be roughly equal to c divided by twice the frequency range in hertz. The second limitation is due to the implicit windowing of the data, which leads to the phenomenon of 'leakage'. In some cases, this can lead to the responses from weaker scattering centers being completely buried in the sidelobes of stronger scatterers. Although the skillful selection of a tapered data window can reduce the sidelobe leakage [21], this is always achieved at the expense of reduced resolution.

It is therefore likely that the harmonic model is not the optimal approach for extracting range information, since, for the case studied here, the data length is relatively short (55 points).

4.6 Prony's Method

The Prony model can be traced back to the work of Gaspard Riche, Baron de Prony [1795]. The contemporary version has evolved somewhat and involves fitting an exponential model to the data (in the least squared sense).

The Prony method estimates the data sequence v_i with an M term complex exponential model. The standard form is given as

$$\hat{v_i} = \sum_{m=1}^{M} A_m e^{[(\alpha_m + \jmath 2\pi f_m)(i-1)T + \jmath\Theta_m]}, i = 0 \ldots N \qquad (4.13)$$

where $\hat{v_i}$ is the estimated sequence, T is the sample interval in seconds, A_m is the amplitude of the complex exponential, α_m is the damping factor in seconds^{-1}, f_m is the sinusoidal frequency in Hz, and Θ_m is the sinusoidal initial phase in radians.

By making the substitution $A_m = \rho_m$, $\alpha_m = 0$, $f_m = -\tau_m$, $\Theta_m = 2\pi f_o \tau_m$ equation 4.18 can be made to take on the exact form of equation 4.6. This model therefore has the same intuitive appeal as the harmonic model. The Prony model can in fact be considered a generalized form of the Fourier model, since the data is once again modeled as a sum of exponentials. However for the Prony case no assumptions are made about their number, phase, frequency or damping factor. The Prony method also does not make the assumptions that the data outside the recorded window is zero.

For this reason the Prony model does not exhibit sidelobes and is also capable of achieving much higher resolution than the Fourier series model (especially for short data sequences).

The Prony method must estimate *four* parameters, namely A_m, α_m, f_m and Θ_m. This is of course a highly non-linear problem, and no known analytic solutions exist. The modern variant of Prony's method, the *least squares* Prony or *extended* Prony method, uses a sub-optimal approximation to the true prediction error.

Determination of the least squares Prony parameters is achieved in three steps. In step one the linear prediction parameters that fit the available data are determined. In step two estimates of damping and sinusoidal frequencies of each of the exponential terms are obtained from the roots of a polynomial formed from the linear prediction coefficients. Finally step three estimates the exponential amplitude and initial phase by finding the solution to a second set of linear equations. It is particularly interesting to note that the the first two steps are identical to solving for the poles of the covariance method of linear prediction (discussed in section 4.7). The Prony method can therefore be seen as an extended AR pole-finding algorithm, with an extra step to determine the magnitudes and phases of the poles.

Another variant of Prony's method, called the modified least squares or Hildebrand Prony method [27], models the data as a set of *undamped* sinusoids (i.e., α is set to zero). (This approach is appealing since the theoretical model does not contain damped sinusoids.) In this case the algorithm simply reduces to an extended version of the *modified* covariance algorithm (also discussed in section 4.7).

Prony's method should not be applied when noise is present. There are many modifications of Prony's methods designed to deal with low SNR and small sample size. The more sophisticated approaches tend to use singular value decomposition, utilizing the fact that a low rank approximation to the noisy correlation matrix helps to mitigate against the effect of noise (the Principal Eigenvector method). The Principle Eigenvector method is discussed in section 4.8.2.

4.7 Rational Transfer Function Models

4.7.1 Introduction and Overview

Rational transfer function models include the now well known Autoregressive (AR) and Moving Average (MA) models [22] [28]. The most general linear model, the ARMA model, is a combination of the two.

These models find application over a considerable range of deterministic and discrete-time processes. The reason lies in their relationship to a fundamental theorem in the decomposition of time series which was proposed by Wold [28]. This theorem states that any real valued, stationary stochastic process allows the decomposition $y_t = u_t + v_t$, where u_t is deterministic and v_t is non-deterministic and has an absolutely continuous spectral distribution function based on a white noise process.

Although Wold's theorem shows that the most general decomposition of a stochastic process is a moving average (MA) one, MA is not well understood and has computational drawbacks which make it unattractive for most applications. Determination of the AR parameters, on the other hand, is a linear problem and is consequently a much simpler process. AR estimation is therefore better understood, faster, and has been more extensively researched than the other models. In addition, Wold's theorem shows that any ARMA or MA process can in fact be represented as an AR process (albeit of possibly infinite order). In terms of rational transfer function models therefore, it is the method of choice in this paper.

Referring back to the previous section, the reader should bear in mind that AR spectral analysis involving a study of the pole locations is essentially identical to the least squares Prony approach without the extra step to determine the amplitude and phase information. AR spectral analysis can therefore be seen as a Prony procedure.

4.7.2 The Autoregressive (AR) Estimator

The importance of the AR method as a spectral estimation technique can be attributed largely to the work of Burg [28], who introduced the concept of maximum entropy (ME). Burg's ME approach moves away from the assumption inherent in conventional spectral estimates (e.g., periodogram and Blackman-Tucky) that the data outside the available record is zero. ME assumes the data is nonzero and that all the information about the system is contained in the finite record.

In AR modeling, each sample of the data record is assumed to be related to M prior samples according to the linear difference equation

$$v_i = \sum_{m=1}^{M} a_m v_{i-m} + e_i \tag{4.14}$$

Determination of the AR parameters is usually achieved through the method of least squares. Note that in order to relate this model to the AR model derived from statistical estimation theory it is necessary to assume the prediction error is a whitened process (although this may or may not be the case). The AR parameters allow for the calculation of an extrapolated and interpolated version of the original data sequence. In order to extract the weights ρ_m and the range delays τ_m, this extended data record is usually then applied to the harmonic model i.e., the z-transform of the AR system function $H(Z)$ between the input sequence and the output sequence is evaluated along the unit circle $z = exp(j2\pi\tau\Delta f)$ for $-1/(2\Delta f) \leq \tau \leq 1/(2\Delta f)$. [3]

[3] In most cases the input driving process is assumed to be a white-noise sequence of zero mean and variance σ^2.

This infinite extrapolation of the auto-correlation function accounts for the improved resolution and absence of sidelobes for the AR estimator. The source of error is no longer in the assumptions made by the harmonic model. Rather it is determined by how the data sequence is extrapolated – bearing in mind there are, of course, an infinite number of valid solutions.

In the case of the Burg method, the extrapolation is made such that the sequence has maximum entropy. The rationale behind this choice is that it places the fewest constraints on the unknown data record by maximizing its randomness. In recent years the Burg method has fallen out of favor, mainly because it suffers from the problem of spectral line splitting and is also very sensitive to phase. Shifts in peak location by as much as 16 percent have been observed [22].

Another popular approach is to first produce estimates of the auto-correlation sequence from the data, and then to apply these to the well known Yule-Walker equations [22]. Various fast recursive algorithms have been developed to speed up this approach. The Levinson-Durban algorithm [22] achieved widespread popularity, mainly because of the useful property that it provides all the lower AR model fits to the data – a useful feature for determining the optimal model order.

It has been found that in practice the Yule-Walker approach only produces good spectral estimates for long data records. This can be explained from the perspective of statistical estimation theory by comparing it to the maximum likelyhood estimator (MLE), which is the most standard and general estimator for a nonrandom set of parameters. It can be shown that for long data records the Yule Walker Method is a good approximation to the MLE. For short records (which are normally encountered in practice), this approximation breaks down. In this regard there are two least squares estimation procedures that operate directly on the data to yield far better results. The first utilizes forward linear prediction, while the second utilizes a combination of both forward and backward linear prediction. They are called the covariance and modified covariance methods respectively.

4.7.3 The Covariance and Modified Covariance Methods

The convariance method makes use of the forward linear predictor, which has the standard form

$$v_i^f = \sum_{m=1}^{M} a_m^f v_{i-m} + e_i \tag{4.15}$$

where the 'f' denotes 'forward prediction'. In this case the matrix Ψ of equation 4.11 consists of the product $V_M^H V_M$ where V_M is defined by

$$V_M = \begin{bmatrix} v[M+1] & \dots & v[1] \\ \vdots & \ddots & \vdots \\ v[N-M] & & v[M+1] \\ \vdots & \ddots & \vdots \\ v[N] & \dots & v[N-M] \end{bmatrix}, \tag{4.16}$$

The modified covariance method, on the other hand, attempts to improve on the performance of the covariance method by taking advantage of the fact that it is also possible to formulate the AR coefficients in terms of a weighted sum of M *future* samples,

$$v_i^b = \sum_{m=1}^{M} a_m^b v_{i-m} + e_i \tag{4.17}$$

where the 'b' denotes 'backward prediction'.

The modified covariance method therefore combines the linear prediction statistics of both the forward and backward errors to generate more error points and consequently a better estimate of the autoregressive parameters.

For the modified covariance case, Ψ is given as $V_M^H V_M + J V_M^T V_M^* J$, where V_M is as defined in equation 4.16, and V_M^* forms a Hankel matrix of conjugated data elements.

The modified covariance method is widely accepted in the literature as the best performer (especially for short data sequences) out of all the AR methods. No examples of spectral line splitting have been reported, and frequency estimation bias has been shown in [27] to be minimal. It is also amongst the most effective (along with the Burg and covariance method) in the mitigation of spurious peaks, since it gives a better resolution for a given order than other AR algorithms [27]. A final advantage is that it is computationally quite efficient (comparable to the Burg algorithm) and for $M \ll N$ is actually more efficient (which is usually the case when dealing with noisy data, as is the case in this paper).

Note that neither the covariance nor the modified covariance methods guarantee a stable linear prediction filter. This is not a cause for concern here, however, since the coefficients are not actually used for filter synthesis.

4.7.4 Determination of AR Order

A crucial point in estimating AR spectra is the determination of the model order, or prediction error operator.

Due to estimation error, the normal equations of most AR methods will generally be full rank, even for high orders, so that solutions for the AR parameters will be obtained even though the true model order may

be much lower. This then gives rise to spurious poles. The model order therefore determines the trade-off between resolution and estimate variance (analogous to the effect of windows in classical spectral estimates). It is therefore necessary to use some sort of criteria for determining the model order.

Numerous criteria have been proposed. Two popular estimates have been proposed by Akaike [23], namely the *final prediction error* (FPE) and the *Akaike information criterion* (AIC). The latter minimizes an information theoretic function, while the former minimizes the average error variance for a one-step prediction. Other methods have been proposed by Rissanen [1983], and Parzen [1974]. In the final analysis, none of these approaches can be said to produce reliable results.

The general consensus is that subjective judgment, rather than science, is the most effective approach for determining the model order for an unknown process.

4.8 Eigendecomposition Techniques

Eigendecomposition techniques are designed to deal with the problem of additive noise. In this regard, two algorithms have achieved widespread popularity. They are multiple signal classification (MUSIC), first proposed by Schmidt [14], and the modified forward-backward linear predictor (modified FBLP), proposed by Tufts and Kumaresan [20]. Both methods utilize the eigendecomposition of the correlation matrix Ψ in order to separate the observation space into a noise and signal plus noise subspace. The eigendecomposition is performed by means of singular value decomposition. Since an understanding of this procedure is necessary for the discussions that follow, it is described in the following section.

4.8.1 The eigendecomposition of the Data Matrix through Singular Value Decomposition.

Consider the matrix Ψ of section 4.7.3 defined for the frequency estimation problem, where the input signal consists of L *uncorrelated* zero-mean complex sinusoids and an additive white noise process of zero mean and variance σ^2, as in Equation 4.1. We denote the angular frequencies of the sinusoids by $\omega_1, \omega_2, \ldots, \omega_L$. By using singular value decomposition (SVD) [4] , the correlation matrix Ψ of Equation 4.8, prediction order L, may be

[4] The singular value decomposition theorem states that there exist positive real numbers $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k > 0$ (the so-called singular values of A), and $m \times m$ unitary matrix $U = [u_1 \ldots u_m]$, and $n \times n$ unitary matrix $V = [v_1 \ldots v_n]$ such that matrix A can be expressed as $A = U\Sigma V^H = \sum_{i=1}^{k} \lambda_i u_i v_i^H$, where the $m \times n$ matrix Σ has the structure

$$\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$$

expressed in the following form:

$$\Psi = \sum_{l=1}^{L} \lambda_l S_l S_l^H = EDE^H = E_s D_s E_s^H + \sigma^2 I \tag{4.18}$$

where

$$E = [S_1, S_2, \ldots, S_L] \quad , E_s = [S_1, S_2, \ldots, S_M],$$

$$D = diag(\lambda_1, \lambda_2, \ldots, \lambda_L), \quad D_s = diag(\lambda_1^s, \lambda_2^s, \ldots, \lambda_M^s).$$

Also,

$$\lambda_1 = \lambda_1^s + \sigma^2 > \lambda_2 = \lambda_2^s + \sigma^2 > \cdots > \lambda_M = \lambda_M^s + \sigma^2 > \lambda_{M+1} = \cdots = \lambda_L = \sigma^2.$$

λ_i are the eigenvectors of R, and S_i are the corresponding eigenvectors.

Note that when there is no noise (i.e., $E = E_S$), the rank of Ψ will be M and there will only be M nonzero elements in the diagonal matrix D. This produces a good solution to Equation 4.10, because the right-hand side vector V in Equation 4.10 lies primarily along the M principal eigenvectors of $\Psi^H \Psi$. Once noise is introduced however (σ is non-zero), Ψ is generally of full rank, with all the eigenvectors making non-zero contributions. The $L - M$ eigenvectors which were originally zero can change directions abruptly depending on the noise perturbation and this effect is amplified by the reciprocal of the usually small eigenvalues $\lambda_{M+1}, \lambda_{M+2}, \ldots, \lambda_l$. These noise subspace eigenvectors can therefore cause significant fluctuations in the prediction filter coefficients, $\widehat{P}$, in Equation 4.10. This leads to the creation of spurious prediction-error filter zeros, and the type of instability commonly encountered in least squares solutions for noisy data.

4.8.2 The Principal Eigenvector Method

The effect described above can be alleviated by using a lower rank approximation to Ψ. This approach, based on a theorem by Eckart and Young [5] and popularized by Kumarsan and Tufts has been termed the modified forward backward linear predictor (M-FBLP) [18] [20]. The M-FBLP is also

and $D = diag(\sigma_1, \ldots, \sigma_k)$ is a $k \times k$ diagonal matrix.

[5] A version of the Eckart-Young theorim can be stated as follows: Let $\widehat{R}$ be an $(L \times L)$ matrix of rank K which has complex-valued elements. Let S_M be the set of all $(L \times L)$ matrices of rank $M < K$. Then for all matrices B in S_M

$$\| \widehat{R} - \widetilde{R} \| - \| \widehat{R} - B \|$$

where

referred to as the minimum norm (ME) method or the principal eigenvector (PE) method in the literature. It essentially involves re-constructing a new correlation matrix Ψ^{KT} by setting to zero all but the M largest singular values in D. Note that this is based on the assumption that the M largest eigenvalues of Ψ and the corresponding eigenvectors are the perturbed versions of the M *nonzero* eigenvalues and the corresponding eigenvectors of Ψ, which may not always be true at low SNR values. Kumarsen and Tufts also used a prediction order filter (L) which is much higher than the number of sinusoids present in the measurement but lower than the number of samples. They observed experimentally that making L large upgraded the frequency estimation performance.

4.8.3 The MUSIC Method

The MUSIC algorithm also relies on the fact that the eigenvectors of Ψ can be divided into two subsets; one set corresponding to the contributions from the noise only, and the other corresponding to the contributions of the signal plus noise. It can be shown [27] that these two sets of eigenvectors span separate subspaces. They are the orthogonal complement of one another. The MUSIC spectrum is therefore obtained by searching for the signal vectors that are most closely orthogonal to the noise subspace. This corresponds to searching for the peaks in the spectrum

$$S_{MUSIC}(e^{\jmath\omega}) = \frac{1}{D(e^{\jmath\omega})} \tag{4.19}$$

where $D(e^{\jmath\omega})$, termed the null spectrum, is given by

$$D(e^{\jmath\omega}) = V^H(\omega)\left(\sum_{l=M+1}^{L} S_l S_l^H\right)V(\omega) = V^H(\omega)P_N V(\omega) \tag{4.20}$$

where the frequency scanning vector $V(\omega)$ is defined by

$$V^T(\omega) = [1, e^{\jmath\omega}, \ldots, e^{\jmath(L-1)\omega}]$$

It is possible to avoid the need for frequency scanning by using a root-finding approach. Through the use of the z-transform, Equation 4.19 becomes

---

$$\tilde{R} = \sum_{k=1}^{M} \hat{\lambda_k} \hat{u_k} \hat{u_k^H}$$

and the $\hat{u_k}$ correspond to the orthonormal eigenvectors of $\hat{R}$.

$$S_{MUSIC}(z) = \frac{1}{D(z)} \tag{4.21}$$

where $D(z)$ is the resulting denominator polynomial. Ideally, $D(z)$ has zeros on the unit circle in the z-plane at locations determined by the frequencies of the complex sinusoids in the input time series. This approach is known as root-MUSIC and has been shown to be superior to the standard MUSIC algorithm [13]. This can mainly be attributed to the fact that radial errors in the estimated signal zeros (which are most prevalent) are eliminated through the use of root music.

4.8.4 The Total Least Squares Method

The most recent addition to the family of linear prediction algorithms is the *total least squares* (TLS) method for solving the LP equation, introduced by Rahman and Yu [10]. The TLS method of frequency estimation differs mainly from the PE method in the approach for solving Equation 4.8. For PE the nonhomogeneous system of equation $\Psi P = V$ is considered, whereas in the TLS method, the homogeneous equation $[\Psi V]\left[\frac{P}{-1}\right] = 0$ is solved. Whereas the PE method considers the noise effect on the data matrix and observation space separately, the total least squares approach attempts to improve performance at low SNR by combating the noise effect from both the data matrix Ψ and the observation vector V simultaneously. The minimum norm solution to the TLS problem can be summarized as follows [10]:

First compute the SVD of a matrix $B = [\Psi V]$, i.e., $B = U\Sigma V^H$. Let $V_1 = (v_{M+1}, \ldots, v_{L+1})$ be a column partition of V. Next compute a Householder matrix Q such that

$$\mathbf{V_1 Q} = \begin{bmatrix} \vdots \ddots \vdots & \mathbf{y} \\ 0 \cdots 0 & \alpha \end{bmatrix} \tag{4.22}$$

The TLS prediction vector is then given by $P_{TLS} = -\mathbf{y}/\alpha$, or, if $\alpha = 0$ there is no solution (corresponds to overestimating the size of M). The reader may have noticed that although the principal eigenvector method compensates for noise perturbations by setting $\sigma = 0$ in the $L - M$ smallest singular values $\sigma^2 = \lambda_{M+1} = \lambda_{M+2} = \ldots = \lambda_L$. However no attempt is made to adjust the eigenvalues $\lambda_1 = \lambda_1^s + \sigma^2 > \lambda_2 = \lambda_2^s + \sigma^2 > \cdots > \lambda_M = \lambda_M^s + \sigma^2$. Rahman and Yu have shown that the TLS method attempts to compensate for perturbations in the M largest eigenvalues as well – by subtracting an estimate of the perturbation. In this way the M largest eigenvalues are also adjusted for noise. It is for this reason that the TLS method has been shown to produce slightly better results than the PE method in the experimental scenario investigated by Tufts/Kumarsen and Rahman/Yu [10]. However it should be noted that at low SNR and large

M the TLS estimate of the noise perturbations is likely to be inaccurate. This is substantiated empirically in section 4.9.3.

4.9 Experimental Results

Using computer simulations, the MUSIC, PE and TLS methods have all been experimentally evaluated in the literature. Tufts and Kumareson [18] compared their *modified* FBLP approach to the standard Nutall/Ulrych-Clayton FBLP method, given the case of two closely spaced sinusoids with a limited amount of data. They showed their method to be superior over a range of SNR values. Schmidt compared his MUSIC algorithm to the Maximum likelihood and Maximum entropy methods, given essentially the same problem studied by Tufts and Kumareson, except in the context of phased array antennas. He showed how under certain conditions MUSIC outperforms ML and ME in both bias error and ambiguity confusion. Rahman and Yu in turn showed how their TLS approach outperforms PE given the same scenario investigated by Tufts and Kumareson.

However all of these empirical evaluations are essentially restricted to specific scenarios that differ substantially from the synthetic range profiling problem considered here. For this reason a set of experiments was devised in order to study the various algorithms in an aircraft recognition context.

4.9.1 Definition of Signal-To-Noise Ratio

Throughout the simulations conducted in this section, additive noise is injected in the form of independent complex Gaussian samples, distributed with zero mean and variance σ^2. This is done in order to simulate the affects of sea clutter. Although not strictly correct for the ship case (the probability density distribution of sea clutter is generally not Gaussian), previous comparisons with real and simulated data have shown that this estimation does not noticeably degrade the accuracy of the simulations [50]. Throughout this portion of the text, the same definition of signal-to-noise ratio (SNR) as described in [20] will be employed. This is given by

$$10log(1/2\sigma^2)$$

dB. Note that in the ship recognition context, it would be more correct to refer to this signal-to-noise ratio as the signal-to-clutter (S/C) ratio.

4.9.2 Part 1: Estimation of the Size of the Signal Space

The majority of empirical evidence on eigendecomposition techniques is based on only one specific scenario i.e., the problem of resolving two closely spaced sinusoids. Here M is known *a priori* to have a value of two. This

is far removed from the typical ship target recognition problem where the value of M is completely unknown and is usually much larger than two.

For the synthetic range profiling problem, the rank of the signal space is not known and must be estimated. Several solutions to this problem have been proposed. These include simple cutoffs, using statistical tests such as minimum description length, and the addition of regularization parameters to create thresholds. The statistical methods are unattractive here because of their computational complexity. Popular cutoff tests include choosing the size of the signal space M such that λ_{M+1} is the first singular value smaller than $s\lambda_1$, with $0 < s < 1$ fixed, or choosing M such that $\lambda_{M+1} - \lambda_M < s(\lambda_1 = \lambda_2)$. These tests are based on the assumption that the noise eigenvalues $\lambda_{M+1}, \ldots, \lambda_L$ will be significantly smaller than the signal eigenvalues and will be centered within a narrow band around σ^2, the noise variance. This assumption breaks down at low signal to noise ratios, or for closely spaced sinusoids [20], where the magnitudes of the signal and noise eigenvectors become comparable and the variance of the noise eigenvectors is high. To investigate this, the following experiments will investigate the problem of determining the true rank of the correlation matrix using the latter of the two cutoff tests described above.

We begin by arbitrarily defining a target consisting of six discrete reflecting features separated in space at 10 meter intervals by the range vector

$$r = [10, 20, 30, 40, 50, 60]$$

Referring to the general SRP equation of Equation 4.6, and assuming a start frequency f_o of $1.3e^9$ and a frequency step Δf of $1.875e^6$, the received signal over $N = 55$ frequencies, at time k, due to the combined action of $M = 6$ point reflectors, is given by

$$u(i,k) = \sum_{m=1}^{M} \alpha_m e^{2\pi \Delta f r_m i} + v(i,k), \quad i = 1, \ldots, N \tag{4.23}$$

where the complex amplitudes, $\alpha_m = \rho_m e^{2\pi f_o \tau_m}$, are assumed to be unity for the sake of simplicity. Here the $v(i,k)$ are independent complex Gaussian samples distributed with zero mean and variance σ^2. As described earlier, the rank of the correlation matrix is estimated by choosing M such that $\lambda_{M+1} - \lambda_M < s(\lambda_1 = \lambda_2)$.

Discussion

Due to the large number of figures in this section, the detailed results have been placed in Appendix B1 (Figures B.1–B.18). Each graph displays the average frequency of occurrence of the estimated signal space size over 100 independent trials. From the figures one can see that this approach performs perfectly in low noise, correctly calculating the size of the signal

space $M = 6$ to 100% accuracy. These results become slightly degraded at a SNR ratio of $0dB$, and unacceptable at a SNR of $-10dB$.

The nature of the problem is illustrated in Figures 4.1–4.3, where the spread of eigenvalue magnitudes at various signal-to-noise ratios is plotted. Notice how, down to a signal to noise ratio of $0dB$, it is possible to make a clear distinction between the noise and signal spaces. Also note that at a SNR of $-10dB$ the problem is not solvable.

Recall that for the ship data used in Section 2, the signal to clutter ratios were typically high ($\pm 15dB$). However this data was recorded in calm sea conditions and the signal to clutter ratio is expected to drop below $0dB$ in rough sea conditions. It is therefore likely that the size of the signal space will not always be predicted with a high degree of accuracy. For robust performance over a wide range of sea conditions, it is therefore important to select a super-resolution algorithm that is tolerant of inaccuracies in the estimation of the signal space size.

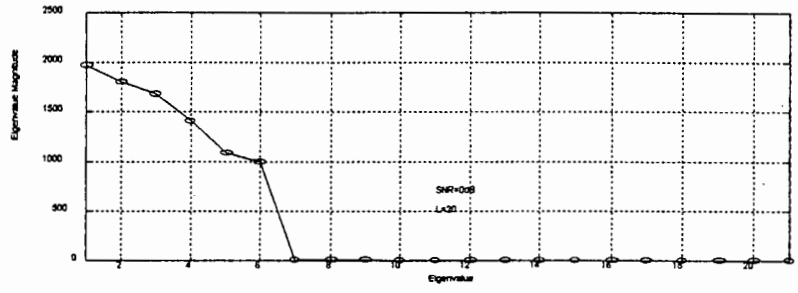


Figure 4.1: Eigenvalue magnitude versus eigenvalue number: A typical example of the spread of values for a SNR of 50dB and a prediction filter order of 20.

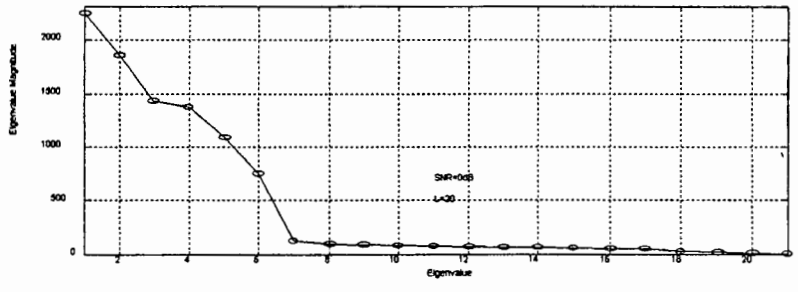


Figure 4.2: Eigenvalue magnitude versus eigenvalue number: A typical example of the spread of values for a SNR of 0dB and a prediction filter order of 20.

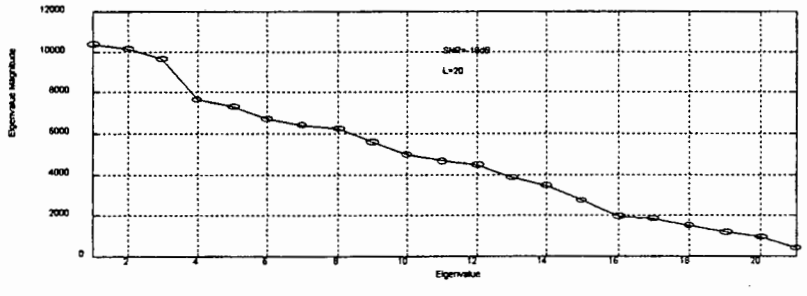


Figure 4.3: Eigenvalue magnitude versus eigenvalue number: A typical example of the spread of values for a SNR of -10dB and a prediction filter order of 20.

4.9.3 Part 2: Algorithm performance at Different Values of M, L and SNR

In this section, experimental results on the range estimation performance of the MUSIC, TLS and PE methods are discussed. The algorithms are evaluated at different values of M, L, and SNR. Of particular interest are their tolerance to inaccuracies in the estimation of the size of the signal space (M), since it was shown in 4.9.2 that this is a requirement for robust performance. Also of interest are their performance at different signal to noise ratios.

As in section 4.9.2 we begin by arbitrarily defining a target, represented by the range vector

$$r = [10, 30.001, 30.002, 30.003, 30.004, 30.005, 30.006, 50, 52, 54]$$

The start frequency f_o, frequency step Δf and number of frequency steps N are defined as before. The received signal at time k, due to the combined action of $M = 10$ point reflectors, is given by

$$u(i,k) = \sum_{m=1}^{M} \alpha_m e^{2\pi \Delta f r_m i} + v(i,k), \quad i = 1, \ldots, N \tag{4.24}$$

where the complex amplitudes, $\alpha_m = \rho_m e^{2\pi f_o \tau_m}$, and complex Gaussian noise samples $v(i,k)$ are also as defined earlier. The 6 features defined over the range $30.001 \ldots 30.006$ are an attempt to represent a reflector that is slightly distributed in space. This is to be expected on real targets.

It is necessary to define a fitness function in order to compare the performance of each algorithm. Rahman and Yu used what they called the 'effective standard deviation' [10]. Kumareson and Tufts used the sample variance of the frequency estimate to obtain a fitness measure [20]. These fitness measures are problem specific however, and lose their validity when used within a ship target recognition context. In formulating a fitness function for ATR, the following points should be noted:

- All of the autoregressive spectral estimation techniques suffer from small inaccuracies in their estimation of the positions of the spectral components. Most fitness measures penalize these inaccuracies heavily. This applies particularly to the two-point reflector problem studied in most of the literature. However with ship profiles it is not necessary to place as much emphasis on the exact range estimation (i.e. frequency estimation) of the dilute reflectors. This is because these errors will typically be of a smaller magnitude than those caused by the phenomenon of *range migration* - the term used to describe the change in the line of sight distribution of the various scattering centers with variations in target aspect angle (see section 1.1).

- Since the size of the signal space M must be derived, it will not be uncommon (especially in rough seas) to overestimate the true rank of the correlation matrix and hence introduce a noise eigenvector into the equation. This can lead to spurious zeros, which if close to the unit circle will be mistaken for prominent reflecting features on the target. Clearly this would severely hamper recognition, and therefore must be penalized in an ATR fitness test.

- Although autoregressive root-finding methods do not yield accurate amplitude information, signal zeros will generally lie close to the unit circle (yielding larger spectral spikes). We can therefore define a measure of the 'certainty' that a particular pole is in fact a signal pole, by measuring its distance to the unit circle. In this way it is possible to scale the fitness test so that only those noise zeros lying close to the unit circle are heavily penalized. In the same way signal zeros drifting off the unit circle will not be strongly rewarded.

The performance function that meets all the above criteria is defined as follows:

$$\sum_{m=1}^{M} \left\{ 2 \sum_{k=1}^{K} ceil\ [max(0, min((r_k + \gamma) - d_m, d_m - (r_k - \gamma)))\ ] \frac{A_m}{\gamma} - A_m \right\} \tag{4.25}$$

where r_k is the k^{th} element of the range vector, $\mathbf{r}$, and d_m is the d^{th} element of the estimated range vector, $\mathbf{d}$, obtained from the poles of the prediction error filter. A_m is the 'certainty' of each pole, defined as the reciprocal of the distance of the m^{th} pole to the unit circle, and γ is a tolerance parameter that sets the minimum accuracy required for the range estimates. Ceil(x) rounds the elements of x to the nearest integers towards infinity.

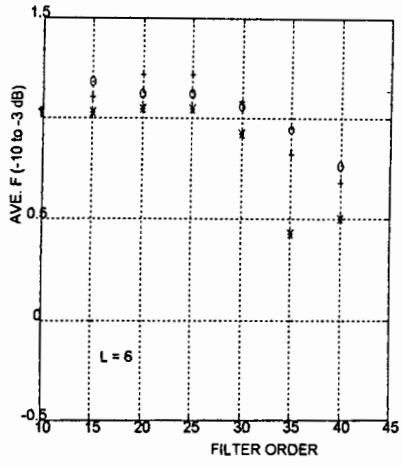
In total, 57 600 statistically independent trials were conducted over a wide range of signal-to-noise ratios, filter prediction sizes and estimated signal space sizes. In all cases γ was set to 0.25. M, the estimated size of the signal space, was varied from a value of 6 to 13. A value of $M = 6$ was determined form the zero noise case to be the true rank of the correlation matrix. Since there exists no theoretical solution to the optimal size of the the prediction filter L, this value was also varied from $L = 40$ to $L = 15$. Although there is no unique method of setting up the correlation matrix, we restrict ourselves to the use of the modified covariance method for reasons discussed in section 4.7.3.

Discussion

A summary of the results is presented in Figures 4.4–4.11. Detailed documentation of the experimental results can be found in [4].

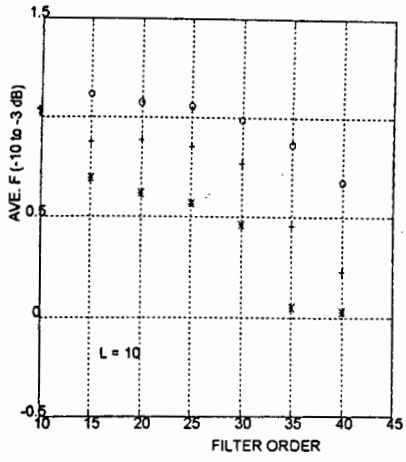
The experiments show that the overall performance of TLS over the complete range of M, L, and SNR values is inferior to PE. This confirms the importance of using the correct fitness criteria and test conditions, since the experimental evidence of Rahman and Yu showed that the performance of TLS would be superior. Recall that their experiments where for the two-point reflector problem. It is interesting that Rahman and Yu concentrated on high signal-to-noise ratios and small values for L. Under these conditions TLS does compete well with PE in the experiments conducted here.

TLS and PE perform well when the SNR is high and the size of M is correct. However their performance drops off sharply at lower SNR, especially when M is overestimated. They also show high sensitivity to the size of the prediction error filter. On average, the performance of MUSIC is superior to PE and TLS. This is largely due to its consistent performance at lower signal-to-noise ratios, and when the value of M is overestimated. MUSIC is also less sensitive to the size of the prediction error filter.



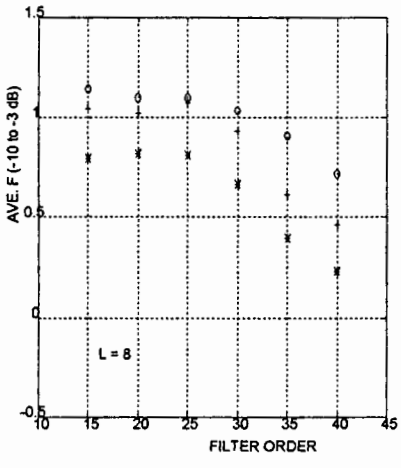
o: MUSIC *: TLS +: PE

Figure 4.4: Average fitness versus prediction filter order (L). SNR range = -5 to 50dB. Assumed number of reflectors (M) = 6.



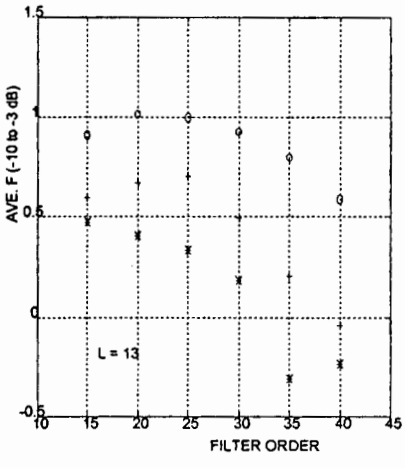
o: MUSIC *: TLS +: PE

Figure 4.6: Average fitness versus prediction filter order (L). SNR range = -5 to 50dB. Assumed number of reflectors (M) = 10.



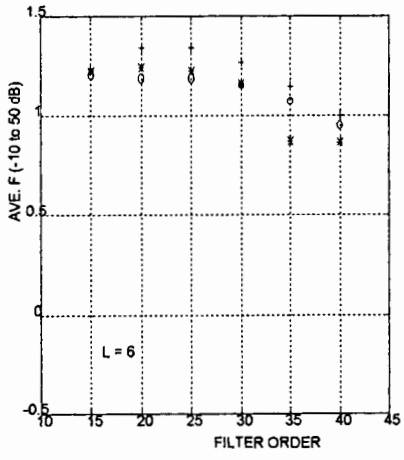
o: MUSIC *: TLS +: PE

Figure 4.5: Average fitness versus prediction filter order (L). SNR range = -5 to 50dB. Assumed number of reflectors (M) = 8.



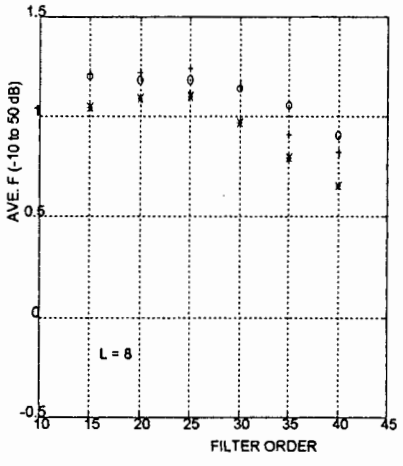
o: MUSIC *: TLS +: PE

Figure 4.7: Average fitness versus prediction filter order (L). SNR range = -5 to 50dB. Assumed number of reflectors (M) = 13.



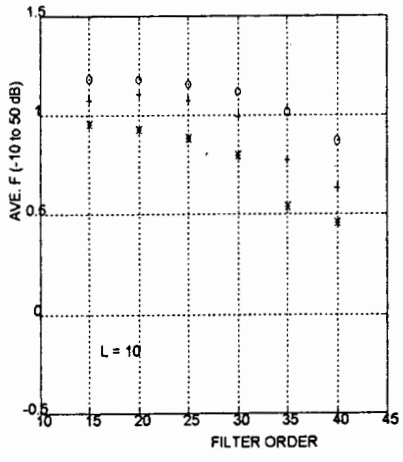
o: MUSIC *: TLS +: PE

Figure 4.8: Average fitness versus prediction filter order (L). SNR range = 0 to 50dB. Assumed number of reflectors (M) = 6.



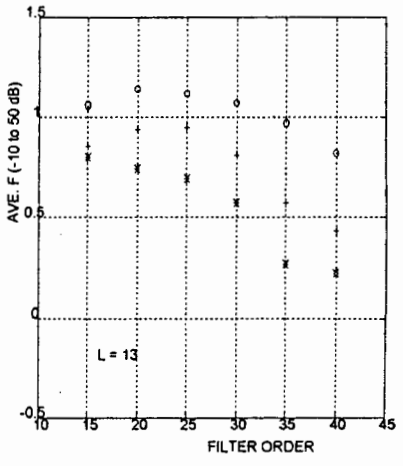
o: MUSIC *: TLS +: PE

Figure 4.9: Average fitness versus prediction filter order (L). SNR range = 0 to 50dB. Assumed number of reflectors (M) = 8.



o: MUSIC *: TLS +: PE

Figure 4.10: Average fitness versus prediction filter order (L). SNR range = 0 to 50dB. Assumed number of reflectors (M) = 10.



o: MUSIC *: TLS +: PE

Figure 4.11: Average fitness versus prediction filter order (L). SNR range = 0 to 50dB. Assumed number of reflectors (M) = 13.

4.9.4 Part 3: Typical Results for Each Algorithm

In this section, the synthetic range profiles produced by each algorithm at signal to noise ratios of $+15dB$, $1dB$ and $-10dB$ are examined., Using the results of the previous two experimental sections, only the optimal values for the prediction error filter length and signal space size – where applicable – were used.

Due to the large number of figures in this section, they have been placed in Appendix B (part 3). The reader should be aware that the various graphs plotted are based on isolated examples and should not be used as an exclusive means of comparing the relative performances of the different algorithms. The purpose of this section is merely to provide the reader with an intuitive feel for the behavior of each algorithm.

We once again use the target arbitrarily defined in section 4.9.2. Recall that the target was represented by the range vector

$$r = [10, 30.001, 30.002, 30.003, 30.004, 30.005, 30.006, 50, 52, 54]$$

The start frequency f_o, frequency step Δf and number of frequency steps N are again defined as before. The received signal at time k, due to the combined action of $M = 10$ point reflectors, is given by

$$u(i,k) = \sum_{m=1}^{M} \alpha_m e^{2\pi \Delta f r_m i} + v(i,k), \quad i = 1, \ldots, N \tag{4.26}$$

where the complex amplitudes, $\alpha_m = \rho_m e^{2\pi f_o \tau_m}$, and complex Gaussian noise samples $v(i,k)$ are as defined earlier. Recall also that the 6 features defined over the range $30.001 \ldots 30.006$ are an attempt to represent a reflector that is slightly distributed in space.

Where applicable, the zeros of the prediction error filter are plotted superimposed on the unit circle. Also where applicable, a stem graph of the estimated range and magnitude (as defined in section 4.9.3) of the reflecting features is given. Note that poles exactly on the unit circle produce infinite magnitudes. For graphing purposes large stems were truncated above a certain value.

An additional note is required with respect to the calculation of the SRP via the Prony method. Normally the Prony procedure terminates with the computation of the amplitude, damping factor, frequency and phase estimates. In order to obtain a Prony 'spectrum' from the exponential estimates, various assumptions can be made. The assumption used here is that the discrete-time exponential sum defined in Equation 4.13 is defined only over the windowed interval $i = 1..200$. This extrapolated data sequence was then processed via a FFT. In this way the predicted targets from the Prony method could be interpolated.

Discussion

Figures B.19 - B.27 illustrate the performance of the MUSIC algorithm. Notice how the reflectors are clearly resolved down to $+1dB$, whilst at $-5dB$ the 4 largest poles are still correctly determined. Compare these results to those plotted for the PE method (Figures B.28 - B.36) and TLS method (Figures B.37 - B.45), where the effects of noise zeros are clearly more intrusive at lower signal to noise ratios.

Figures B.46 - B.63 deal with the modified covariance and covariance methods. Notice how their magnitude/range stem plots are more confused than for the eigendecomposition algorithms. Figures B.64 - B.66 demonstrate the remarkably robust performance of the FFT and the corresponding sacrifice in terms of resolution.

Finally the Prony and Hildebrand approaches are illustrated in Figures B.67 - B.84. The robust nature of the SRP plots can be attributed to the FFT operation implicit in their computation. A look at the actual pole/magnitude stem graphs clearly reveals that both the Prony and Hildebrand approaches are in fact very susceptible to noise. Readers tempted to use the combined Prony/FFT approach because of its noise resilience are cautioned that the calculation of these SRPs are computationally extremely expensive. The Modified Prony Method which utilizes the principal eigenvector method provides a notable improvement in this respect. However its performance still falls far short of the approaches such as the MUSIC method.

4.10 Discussion

As mentioned earlier, the technique of extracting range information from the SRP profile involves taking the inner product of the received data set $\boldsymbol{E}$ with a basis function $\boldsymbol{H}$. However $\boldsymbol{H}$ can never be specified exactly. This is simply because the number and position of the respective reflectors is not known, and neither is the exact nature of noise introduced into the received data set (recall that for the ship recognition problem, the principal source of noise will be from sea clutter). The preceding sections have demonstrated the wide array of techniques that are available for estimating $\boldsymbol{H}$.

These vary from the robust FFT (which pre-defines both the number and position of the target's reflectors), to the noise sensitive Prony approach (which pre-defines only the number of reflectors).

In selecting the optimal algorithm for SRP neural based recognition, it may help to recall the properties that the ideal algorithm should posses:

- It should achieve the highest resolution possible from the data
- It should ideally reduce the size of the data sets by performing some kind of feature extraction
- It should be fast. In addition, it should be suited to a neural based implementation
- It should be robust and noise resilient.

Although the traditional FFT approach is fast and robust, it does not optimally satisfy the first two requirements. In the preceding sections, the use of various super-resolution algorithms were shown to offer promise in terms of satisfying these requirements. However most of them fell well short of the required level of noise resilience expressed in the fourth criteria. This shortfall was remedied through the addition of eigendecomposition techniques, making the use of three super-resolution algorithms quite promising. These where:

- The Principal Eigenvector Method
- The Total Least Squares Method
- The MUSIC method

In selecting the ideal algorithm from the shortlist above, the following additional criteria must be considered:

- The size of the signal space (M) is unknown and must be estimated. This estimation is likely to be inaccurate if the SNR is low (i.e. for rough sea conditions). We seek a method that is tolerant of this.

- The optimal size of the prediction error filter for this set of experiments can be seen to be approximately $L = 20$. However the optimal value of L is closely linked to the value of M, and will therefore vary. We also seek a method that is tolerant of the size of L used.

It is clear from the experimental results that the MUSIC algorithm is best qualified in terms of the above requirements. In terms of super-resolution techniques it was amongst the fastest performers. Although slower and less robust than the FFT, it offers significant gains in terms of improved resolution, as well as the data compression and feature extraction features inherent in its pole-finding approach.

It is important to remember that a fundamental assumption made in the choice of the optimal approach is that accurate amplitude information is not required (refer to section 3). If accurate amplitude information were required, we recall that the Prony method can be seen as an extended pole finding algorithm, with an extra step to determine the magnitudes and phases of the poles. The suggested approach would then be to apply the additional Prony step to the poles obtained through the MUSIC algorithm.

Chapter 5

Time Domain Averaging

In section 1.3 it was mentioned that the low correlation between scans is most likely the major limiting factor as far as classification accuracy is concerned, and that one way of combating this would be to employ some sort of time domain averaging. In this chapter the employment of time domain averaging by means of a fast correlation algorithm is proposed.

Time domain averaging is attractive because it improves the signal to clutter ratio of the profiles and also assists in reducing the influence of the target's anisotropic reflectors (which confuse recognition). For example, for the ship recognition problem, the target signature can be considered to consist of the sum of the quasi isotropic reflectors' contributions, together with a randomly varying component due to the specular reflectors. Time domain processing will suppress the random component and accentuate the target's important distinguishing features (the quasi isotropic ones).

The algorithm used and discussed here was originally developed by Hudson and Psaltis [38], who applied it in conjunction with compound identification to achieve aircraft classification. However, it is also ideally suited for adaption to the time domain averaging problem. This is because it enables the accurate alignment of a set of time domain profiles – a requirement that is crucial if effective time domain averaging is to be achieved. The raw ship target profiles are not, as a rule, aligned. This applies to both synthetic range profiles and normal time domain profiles (the former due to inacuracies in the estimation of the target's velocity and range, the latter due to errors in the synchronization of the time domain sampling with the leading edge of the radar).

Some of the author's work reported in this Chapter has been published in an aircraft recognition context in [31].

5.1 Algorithm Overview

Hudsen and Psaltis used their algorithm as a means of deriving an optimally representative signature to be used as a recognition template for each incoming range profile. Actual recognition was achieved by employing the classic technique of cross-correlating the a-priori representative waveform with the noisy target signature.

The problem in obtaining the recognition templates is in optimizing the

a-priori information, so that the representative waveform or filter can be derived that 'looks like' the profiles of that target as much as possible. For a given target, the filter $f = [f(0), f(1), \ldots, f(K-1)]$ is computed, which maximizes

$$\Phi_1 = E[peak(f \star p)] \,, \tag{5.1}$$

the expected value of the correlation peak between the filter and a randomly chosen profile of the target $p = [p(0), p(1), \ldots, p(K-1)]$.

With N recorded profiles of a target, the expected value of the correlation peak is estimated.

$$\Phi = \frac{1}{N} \sum_{n=1}^{N} peak(f \star p_n) \tag{5.2}$$

The profiles can be range aligned so that the peak correlation occurs at zero shift. It is then only necessary to take the dot product between f and $\acute{p}_n$, a shifted version of p_n.

$$\begin{aligned} \Phi_1 &= \frac{1}{N} \sum_{n=1}^{N} f.\acute{p}_n \\ &= f(\frac{1}{N} \sum_{n=1}^{N} \acute{p}_n) \end{aligned} \tag{5.3}$$

The unit vector that maximizes the dot product is given by

$$f = \frac{\sum_{n=1}^{N} \acute{p}_n}{\left| \sum_{n=1}^{N} \acute{p}_n \right|} \tag{5.4}$$

and the shifts have to be found that maximize 5.4.

They achieved this by means of an iterative procedure, which eliminates the otherwise prohibitive amounts of computation required. The reader is referred to Appendix C for a full breakdown of the approach.

In a nutshell, each iterative step uses a sample profile as a *seed* and cross-correlates it with the average of all the other sample profiles. The shift producing the best correlation is then recorded and used to update the position of this profile before moving on to the next one. In finding the range shifts that maximize 5.4, it is therefore possible to align and average the profiles to obtain a filter that produces the best mean response for the representative set as a whole.

5.2 Discussion

It should be clear that in the same way that the algorithm is used to calculate the range shifts required to maximize the performance of the representative filter, it can also be adapted to align and average the ship profiles *as they come in*, to achieve time domain averaging.

The algorithm has proved remarkably resilient to large specular spikes, as well as inconsistencies in the basic waveform shape, and consistently manages to resolve the underlying signal so as to achieve error-free alignment.

This is illustrated in Figures 5.1 and 5.2. The profiles used consist of real SRP data of an Airbus A320. They were recorded using the SRP radar system mentioned in section 3.4.2. Figure 5.1 shows a set of 16 scans or profiles of the Airbus, and Figure 5.2 shows the same 16 scans after proper alignment and normalization. The last figure, Figure 5.3, shows the resulting time domain averaged profile, obtained by averaging the profiles in Figure 5.2. One can clearly see the reduction in noise, and also how seven prominent reflecting features, which were not at all clear in the original set of 16 profiles, have been resolved.

The algorithm is computationally efficient and therefore fast. In addition it was found that the algorithm could be sped up by a factor of 24 by using an FFT based approach. The FFT implementation also makes this algorithm ideal for a parallel (and therefore neural) implementation. The implementation used by the author can be found in Appendix C

In conclusion, it is recommended that the algorithm be employed in a pre-processing stage to achieve time domain averaging of the ship profiles as they come in. It is highly probable that an improvement in the classification accuracies reported in section 2 will be achieved.

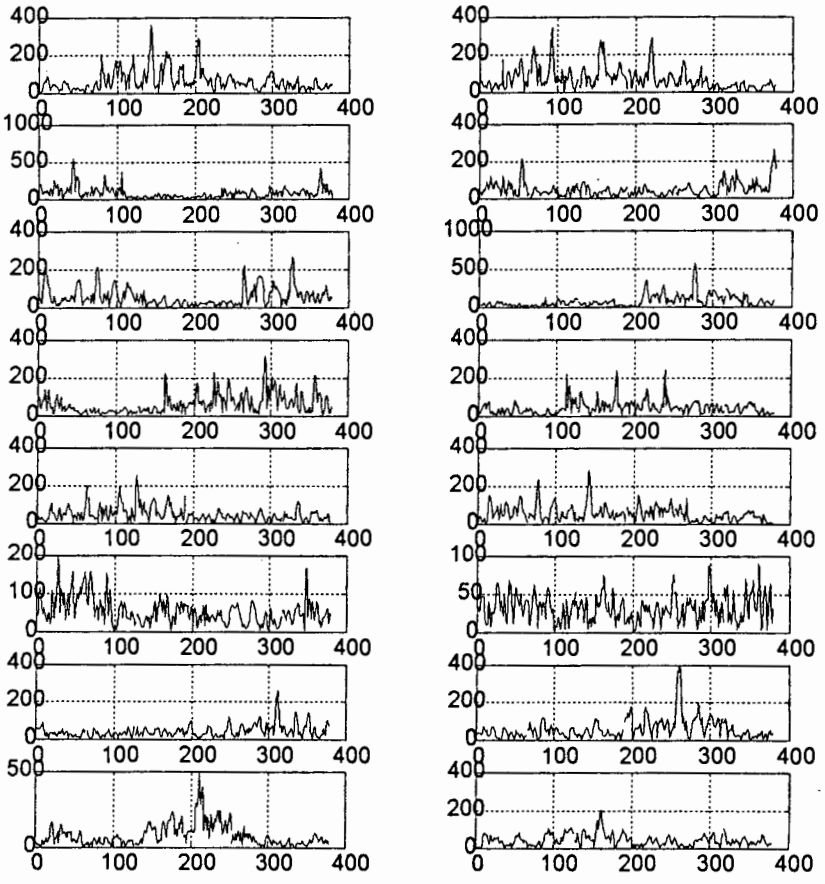


Figure 5.1: 16 scans for an Airbus A320. The vertical axis represents range and the horizontal axis represents magnitude.

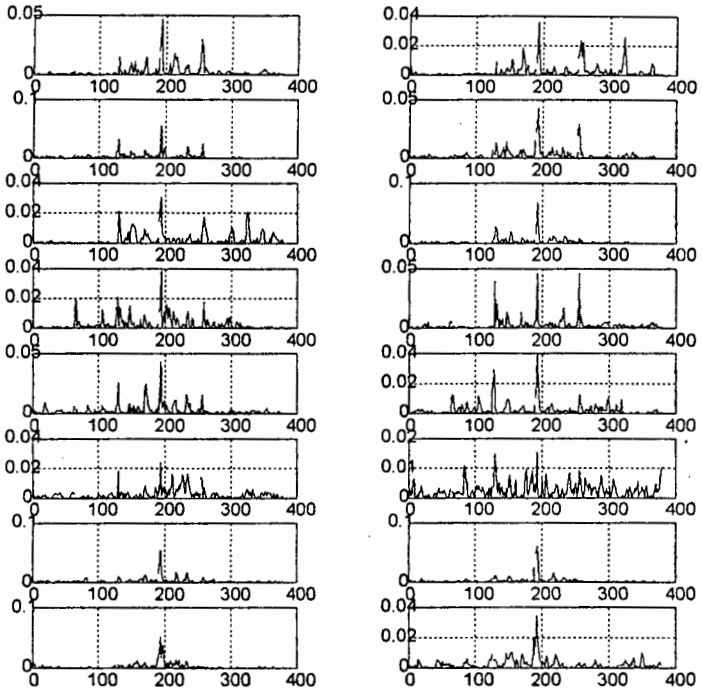


Figure 5.2: The same 16 scans after alignment and normalization.

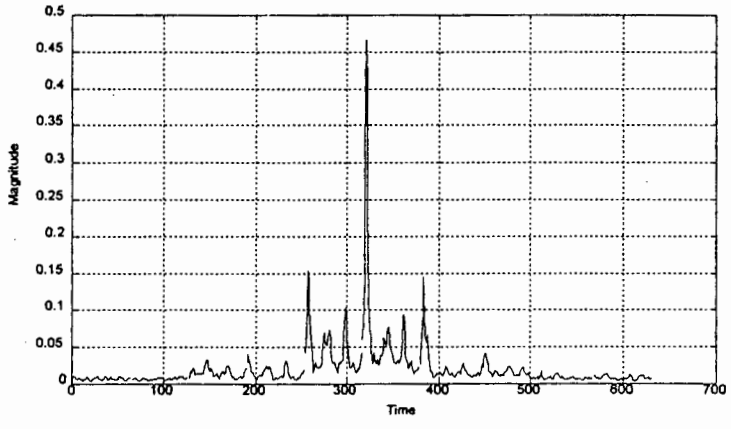


Figure 5.3: The resulting correlation filter.

Chapter 6

Conclusions

Ship target recognition over a wide range of aspect angles has been demonstrated. These results were achieved with a low resolution, non-coherent navigation radar. The modest requirements in terms of computer and radar hardware means that the system shows great potential for providing recognition systems to a variety of users. In addition, a host of new techniques have been investigated. These include the use of synthetic range profiling, the use of various high resolution algorithms, and the use of time domain averaging.

In particular, the following conclusions can be drawn:

- The algorithm known as the Fourier Modified Discrete Mellin transform (F-MDMT) is an effective feature extraction technique in the context of ship target recognition. This was demonstrated by the probability density distribution of the class feature vectors, which was found to closely match the Gaussian assumption made for the Bayes classifier. In addition, it was found that the algorithm remains useful in a neural context. Although neural classifiers are able to generate and learn their own complex mapping functions, it was found that better results could be obtained when specific *a priori* information is used and applied (as is the case for the F-MDMT). Specifically, the algorithm assists the neural classifier in overcoming one of the principal components of the ship classification problem, namely, the inability to accurately determine target aspect angle information.

- In the rapidly advancing field of neural network theory, the Fuzzy Min-Max network has emerged as the current classifier of choice for the ship recognition problem. This classifier was compared to, and found superior to, both Kohonen's Supervised Learning Vector Quantization network, as well as the well known Feedforward network employing Back-propagation. Although all three classifiers were capable of similar classification accuracies, both the Back-propagation and LVQ algorithms were found to fall short of the ideal requirements. Back-propagation suffered from various training problems, including an inability to converge and slow training times. Although LVQ showed significant improvements in this regard, both networks required random guesswork in the tuning process, and were not able

to adapt and update themselves on an on-going basis. By contrast, the Fuzzy Min-Max classifier was found to be capable of learning new classes and refining existing classes on the fly, without destroying old class information and without the need for complete retraining. In addition, it used only a few, well understood tuning parameters.

- Despite the good results achieved, it was concluded that the existing system could be significantly improved. It was felt that the limited size of the real data sets produced somewhat inflated results, and that in reality the low range resolution of the radar would make accurate discrimination of either small or similar targets unfeasible. Suggestions for an improved radar system included the use of high resolution range profiles, and the employment of time domain averaging.

- The use of a synthetic range profile processing radar was found to be a favourable solution to the problem of limited range resolution. In exploring this relatively young field, it is shown theoretically that the problem of extracting range information from the frequency domain data is in fact directly analogous to that of spectral estimation, and also, to that of estimating the direction of arrival for linear phased array antennas. This opened up the large amount of research available in these two fields.

- It was found that the traditional approach of using the FFT to extract range information from the frequency domain data could be significantly improved upon. This lead to an investigation into various super resolution techniques, which enabled the attainment of even better range resolutions. These were in turn combined with eigendecomposition techniques in an attempt to achieve something like the robustness and noise resilience of the FFT. Three algorithms were identified as particularly promising, namely; the Principal Eigenvector Method, the Total Least Squares Method, and the MUSIC method. In comparing their respective performances, it was found that much of the existing evidence was based on scenarios unrelated to that of ship recognition, and therefore were not applicable. A series of tests and simulations was performed in order to evaluate the algorithms under the correct performance criteria. Amongst other things, these criteria included good noise resilience, and tolerance to inaccuracies in the estimation of the sizes of the noise and signal spaces. It was found that the MUSIC method was the optimal choice in this regard.

- A final proposal for future research and development concerned the use of time domain averaging, which is attractive because it improves the signal to clutter ratio and also assists in reducing the random influence of the target's anisotropic reflectors. An iterative correlation algorithm was investigated and developed that promises to produce

robust performance. In addition, a fast implementation of the algorithm was developed that is amenable to a neural-based implementation.

Chapter 7

Recommendations

The main objective of this thesis has been to lay the foundations for future work. The scope of the investigation has covered the design of the radar system, pre-processing, and classification. It has included studies into the use of neural networks, the use of synthetic range profiles, super-resolution techniques, and the use of correlation filters. As a result of this work, various recommendations can be made. These are listed below.

- The existing SRP based aircraft radar system currently under joint development by the University of Cape Town and Reutech Radar Systems should be applied to the ship target recognition problem. Modification of the radar parameters according to the guidelines laid out in Section 3 should be considered.
- A comprehensive set of real data sets needs to be assimilated in order to facilitate thorough testing. This data should include recordings over a wide range of sea states.
- The frequency domain roots should be extracted from the radar data by means of the root-MUSIC algorithm. The size of the signal space should be estimated using the eigenvalue cut-off test investigated in section 4.9.2.
- A pre-processing stage employing the F-MDMT transform should be used to obtain both aspect angle and shift invariant feature vectors.
- Actual classification should be achieved through the use of the Fuzzy Min-Max neural classifier. The classifier should be trained on an on-going basis in order to further investigate its ability to acquire knowledge automatically and refine itself on an on-going basis.
- The high resolution range profiles should be averaged, as they come in, by means of the iterative correlation algorithm detailed in Chapter 5, in order to achieve better recognition accuracies.
- The use of compound identification should be investigated in order to help combat both radar and sea clutter.
- A neural implementation of the F-MDMT should be developed.

- The feasibility of developing a parallel (and hence neural) implementation of the MUSIC algorithm should also be investigated.

Appendix A

The Fuzzy Min-Max Classifier Algorithm

The fuzzy min-max learning algorithm uses an expansion/contraction process. The training set D consists of a set of M ordered pairs $\{X_h, d_h\}$, where $X_h = (x_{h1}, x_{h2}, \ldots, x_{hn}) \in I^n$ is the input pattern and $d_h \in 1, 2, \ldots, m$ is the index of one of the m classes. Note that X_n and A_n are both used to represent input patterns. The learning process begins by selecting an ordered pair from D and finding a hyperbox for the same class that can expand (if necessary) to include the input. If a hyperbox cannot be found that meets the expansion criteria, a new hyperbox is formed and added to the neural network. The final stage of the learning process is one of contraction, whereby any overlap between existing hyperboxes that represent different classes is eliminated by minimally adjusting each of the hyperboxes.

A.1 The Fuzzy Min-Max Classifier Algorithm

Given an ordered pair $X_h, d_h \in D$, find the hyperbox B_j that provides the highest degree of membership, allows expansion (if needed), and represents the same class as d_h. The degree of membership is measured using 2.3. The maximum size of a hyperbox is bounded above by $0 \leq \theta \leq 1$, a user-defined value. For the hyperbox B_j to expand to include X_h, the following constraint must be met:

$$n\theta \geq \sum_{i=1}^{n} (max(w_{ji}, x_{hi}) - min(v_{ji}, x_{hi})) \tag{A.1}$$

If the expansion criteria has been met for hyperbox B_j, the min point of the hyperbox is adjusted using the equation

$$v_{ji}^{new} = min(v_{ji}^{old}, x_{hi}) \forall i = 1, 2, \ldots, n \tag{A.2}$$

and the max point is adjusted using the equation

$$w_{ji}^{new} = max(w_{ji}^{old}, x_{hi}) \forall i = 1, 2, \ldots, n \tag{A.3}$$

A.1.1 Hyperbox Overlap Test

It is necessary to eliminate overlap between hyperboxes that represent different classes. To determine if the expansion step created any overlap, it is necessary to perform a dimension by dimension comparison between hyperboxes. If, for each dimension, at least one of the following four cases is satisfied, then overlap exists between the two hyperboxes. Assume that the hyperbox B_j was expanded in the previous step and that the hyperbox B_k represents another class and is being tested for possible overlap. While testing for the overlap, the smallest overlap along any dimension and the index of the dimension is saved for use during the contraction portion of the learning process. Assuming $delta^{old} = 1$ initially, the four test cases and the corresponding minimum overlap value for the i^{th} dimension are as follows.

- Case 1: $v_{ji} < v_{ki} < w_{ji} < w_{ki}$

$$\delta^{new} = min(w_{ji} - v_{ki}, \delta^{old})$$

- Case 2: $v_{ki} < v_{ji} < w_{ki} < w_{ji}$

$$\delta^{new} = min(w_{ki} - v_{ji}, \delta^{old})$$

- Case 3: $v_{ji} < v_{ki} < w_{ki} < w_{ji}$

$$\delta^{new} = min(min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{old})$$

- Case 4: $v_{ki} < v_{ji} < w_{ji} < w_{ki}$

$$\delta^{new} = min(min(w_{ji} - v_{ki}, w_{ki} - v_{ji}), \delta^{old})$$

If $\delta^{old} - \delta^{new} > 0$ then $\Delta = I$ and $\delta^{old} = \delta^{new}$, signifying that there was overlap for the Δth dimension and overlap testing will proceed with the next dimension. If not, the testing stops and the minimum overlap index variable is set to indicate that the next contraction step is not necessary, i.e., $\Delta = -1$.

A.1.2 Hyperbox Contraction

If $\Delta > 0$, then the Δth dimensions of the two hyperboxes are adjusted. Only one of the n dimensions is adjusted in each of the hyperboxes to minimally impact the shape of the hyperboxes being formed. To determine the adjustment, the same four cases are examined.

- Case 1: $v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta}$

$$w_{j\Delta}^{new} = v_{k\Delta}^{new} = \frac{w_{j\Delta}^{old} + v_{k\Delta}^{old}}{2}$$

- Case 2: $v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta}$

$$w_{k\Delta}^{new} = v_{j\Delta}^{new} = \frac{w_{k\Delta}^{old} + v_{j\Delta}^{old}}{2}$$

- Case 3a: $v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta})$

$$v_{j\Delta}^{new} = w_{k\Delta}^{old}$$

- Case 3b: $v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta})$

$$w_{j\Delta}^{new} = v_{k\Delta}^{old}$$

- Case 4a: $v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta})$

$$w_{k\Delta}^{new} = v_{j\Delta}^{old}$$

- Case 4b: $v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta})$

$$v_{k\Delta}^{new} = w_{j\Delta}^{old}$$

A.2 Fuzzy Min-Max Classifier Program Code

```
% FUZZY MIN-MAX NEURAL CLASSIFICATION ALGORITHM
%
% THIS IS AN IMPLEMENTATION OF THE CLASSIFICATION FUNCTION
% OF THE FUZZY MIN-MAX NEURAL NETWORK
% THE ALGORITHM IMPLEMENTED HERE IS BASED ON THEORY FROM
% "FUZZY MIN-MAX NEURAL NETWORKS-PART1"
% PATRICK K. SIMPSON
% IEEE TRANS. ON NEURAL NETWORKS, 3(5):776-787, SEPTEMBER 92.
%
%
% WRITTEN BY ANTHONY ROBINSON
% UNIVERSITY OF CAPE TOWN, 1995



%%%%%%%% initialization %%%%%%%%%%%%%%%%%%

[classes,ant]=size(u);

bad=zeros(1,classes);
good=zeros(1,classes);
h=0; % no. of input vectors read
[samples,n]=size(x); % n equals size of input vector


%%%%%%%% normalizing %%%%%%%%%%%%%%%%%

x=x';
upperlimit=max(x);
lowerlimit=min(x);

for k=1:samples
x(1:n,k)=x(1:n,k) - lowerlimit(k); %Setting smallest value = 0;
end

upperlimit=upperlimit-lowerlimit;

for k=1:samples
x(1:n,k)=x(1:n,k) ./ upperlimit(k); %Setting largets value = 1;
end

x=x';




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%  main %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%




while h<samples; % continue until all vectors are read
h=h+1

    %%%%%%%%%%%%%%%% hyperbox expansion/creation %%%%%%%%%%%%%%%


          %%% expansion:-check for best matched node %%%

base=0;
disp('checking for best matched node')
[ant,m]=size(u);

for j=1:m

        fb(j)=0;
                temp(1:n)=(max(0,1-max(0,lambda*min(1,x(h,1:n)-w(j,1:n))))...
                + max(0,1-max(0,lambda*min(1,v(j,1:n)-x(h,1:n)))));

                fb(j)=sum(temp)/(2*n); % fb defines each hyperbox membership


end % for j


[ant,node]=max(fb); % node points to best fit node in u.
[ant,class]=max(u(:,node)); % class points to class of node in classlist
if findstr(d(h,:),classlist(class,:))==1
        good(d(h,:))=good(d(h,:))+1;
else
        bad(d(h,:))=bad(d(h,:))+1;
end


end % while samples
```

```
% FUZZY MIN-MAX NEURAL TRAINING ALGORITHM
%
% THIS IS AN IMPLEMENTATION OF THE TRAINING FUNCTION OF
% THE FUZZY MIN-MAX NEURAL NETWORK
% THE ALGORITHM IMPLEMENTED HERE IS BASED ON THEORY FROM
% "FUZZY MIN-MAX NEURAL NETWORKS-PART1"
% PATRICK K. SIMPSON
% IEEE TRANS. ON NEURAL NETWORKS, 3(5):776-787, SEPTEMBER 92.
%
%
% WRITTEN BY ANTHONY ROBINSON
% UNIVERSITY OF CAPE TOWN, 1995



%%%%%%%%% For Testing :- stuff you need before the function call %%%%%%%%

%x=[.1 .321;2 2;1 1;.1 .1; 1.5 1.5; .5 .5; 4 4; 5 5;7 7; 8 8; 9 9; 6 6; 3 3 ];
%d=[1;1;1;1;1;1;2;2;2;2;2;2;1];
%d=str2mat('class1','class1','class2','class2','class1','class1','class1');




%%%%%%%%% initialization %%%%%%%%%%%%%%%%%%%%

m=0; % no. of hyperbox nodes
p=0;  % no. of class nodes
h=0; % no. of input vectors read
lambda=10; % sensitivity parameter
theta=0.4; % expansion parameter (between 0 and 1)
classlist=d(1,:);
[samples,n]=size(x); % n equals size of input vector
                     % samples = size of data set

%%%%%%%%% normalizing %%%%%%%%%%%%%%%%%%

x=x';
upperlimit=max(x);

for k=1:samples
x(1:n,k)=x(1:n,k) ./ upperlimit(k); %Setting largets value = 1;
end

x=x';

%%%%%%%%%%%%%%%% main %%%%%%%%%%%%%%%


while h<samples; % continue until all vectors are read
h=h+1
classmark=0; % marks which class new node belongs to
expand=0; % hyperbox expansion operator
new=1; % hyperbox creation operator
mark=0;

    %%%%%%%%%%%%%%% hyperbox expansion/creation %%%%%%%%%%%%%%%



          %%% expansion:-check for best matched node %%%

if m>0
base=0;
%disp('checking for best matched node')
for j=1:m

        %fb(j)=0;
        %for i=1:n
        %       fb(j)=fb(j)+(max(0,1-max(0,lambda*min(1,x(h,i)-w(j,i))))...
        %       + max(0,1-max(0,lambda*min(1,v(j,i)-x(h,i)))));
        %end
        %       fb(j)=fb(j)/(2*n) % fb defines each hyperbox membership


        fb(j)=0;
                temp(1:n)=(max(0,1-max(0,lambda*min(1,x(h,1:n)-w(j,1:n))))...
                + max(0,1-max(0,lambda*min(1,v(j,1:n)-x(h,1:n)))));

                fb(j)=sum(temp)/(2*n); % fb defines each hyperbox membership


                %%% check expansion constraints %%%


        [ant,class]=max(u(:,j)); % class => class of node j in classlist
        if findstr(d(h,:),classlist(class,:))==1 & fb(j)>base
                mark=j; % marks best fit node of same class
                base=fb(j);
        end


end % for j

%disp('best matched node is node...')
j=mark;



if j>0
        check=0;

        %for i=1:n
        %       check=check+max(w(j,i),x(h,i))-min(v(j,i),x(h,i));
        %end

        temp=max(w(j,1:n),x(h,1:n))-min(v(j,1:n),x(h,1:n));
        check=sum(temp);


        if n*theta>=check; % check to see if selected node fits...
%               disp('best matched node meets expansion constraint')
                expand=1;   ...expansion parameter
        end
end % if j

end % if m

              %%% check if new class is required %%%

for i=1:p

        if findstr(d(h,:),classlist(i,:))==1
                new=0;
%               disp('new class is not required')
                classmark=i; % classmark marks which class matches
        end

end
```

```
                        %%% creation %%%
if new==1 | expand==0

%       disp('creating new node')
        m=m+1; % create new hyperbox node
        if new==1
                p=p+1; % create new class node if one does not exist
                if p>1
                        classlist=str2mat(classlist,d(h,:)); % update...
                end                                          % ...classlist
        end
v(m,1:n)=x(h,1:n); % new hyperbox starts as...
w(m,1:n)=x(h,1:n); % ...point in hyperspace.
if classmark==0
        classmark=p;
end
u(1:p,m)=zeros(p,1); % determines which class this new hyperbox node...
u(classmark,m)=1; % ...belongs to

end


if expand==1 % the rest is only nescessary if we decide to expand
%disp('expansion is required')

                        %%% expansion %%%

                v(j,1:n)=min(v(j,1:n),x(h,1:n)); % adjust min point
                w(j,1:n)=max(w(j,1:n),x(h,1:n)); % adjust max point



       %%%%%%%%%%%%%%% hyperbox overlap test %%%%%%%%%%%%%%%

delta=1;
deltaold=1;

% note that j still points to expanded hyperbox


for k=1:m


[ant,class]=max(u(:,j)); % class points to class of node j in classlist
[ant,match]=max(u(:,k)); % match points to class of node k in classlist


if class~=match  % don't bother about overlap test if hyperboxes...
                 ...belong to the same class
        i=1;
        while i<n+1

                    %%% case 1 %%%

                if v(j,i)<v(k,i) & v(k,i)<w(j,i) & w(j,i)<w(k,i)
                delta=min(w(j,i)-v(k,i),deltaold);
                end

                    %%% case 2 %%%

                if v(k,i)<v(j,i) & v(j,i)<w(k,i) & w(k,i)<w(j,i)
                delta=min(w(k,i)-v(j,i),deltaold);
                end
                    %%% case 3 %%%

                if v(j,i)<v(k,i) & v(k,i)<w(k,i) & w(k,i)<w(j,i)
                delta=min(min(w(k,i)-v(j,i),w(j,i)-v(k,i)),deltaold);
                end
                    %%% case 4 %%%

                if v(k,i)<v(j,i) & v(j,i)<w(j,i) & w(j,i)<w(k,i)
                delta=min(min(w(j,i)-v(k,i),w(k,i)-v(j,i)),deltaold);
                end


                if deltaold-delta<=0 & i==1
                        ch=-1; % indicates contraction not nescessary
%                       disp('contraction unescessary')
                        i=n+1;
                end




                if deltaold-delta>0
                        ch=i; % updates dimension with smallest overlap
                        deltaold=delta;
                end

                i=i+1;

        end % while i
end % if class



    %%%%%%%%%%%%%%% hyperbox contraction %%%%%%%%%%%%%%%


       if ch>0

%disp('starting contraction for dimension ...')
%ch
                        %%% case 1 %%%

                if v(j,ch)<v(k,ch) & v(k,ch)<w(j,ch) & w(j,ch)<w(k,ch)
                        w(j,ch)=(w(j,ch)+v(k,ch))/2;
                        v(k,ch)=w(j,ch);
                end

                        %%% case 2 %%%

                if v(k,ch)<v(j,ch) & v(j,ch)<w(k,ch) & w(k,ch)<w(j,ch)
                        w(k,ch)=(w(k,ch)+v(j,ch))/2;
                        v(j,ch)=w(k,ch);
                end

                        %%% case 3a %%%
                if v(j,ch)<v(k,ch) & v(k,ch)<w(k,ch) & w(k,ch)<w(j,ch)...
                                   & (w(k,ch)-v(j,ch))<=(w(j,ch)-v(k,ch))
                        v(j,ch)=w(k,ch);
                end

                        %%% case 3b %%%

                if v(j,ch)<v(k,ch) & v(k,ch)<w(k,ch) & w(k,ch)<w(j,ch)...
                                   & (w(k,ch)-v(j,ch))>(w(j,ch)-v(k,ch))
                        w(j,ch)=v(k,ch);
```

```
                end

                        %%% case 4a %%%

                if v(k,ch)<v(j,ch) & v(j,ch)<w(j,ch) & w(j,ch)<w(k,ch)...
                                & (w(k,ch)-v(j,ch))<=(w(j,ch)-v(k,ch))
                        w(k,ch)=v(j,ch);
                end


                        %%% case 4b %%%

                if v(k,ch)<v(j,ch) & v(j,ch)<w(j,ch) & w(j,ch)<w(k,ch)...
                                & (w(k,ch)-v(j,ch))>(w(j,ch)-v(k,ch))
                        v(k,ch)=w(j,ch);
                end

        end % if ch
end % for k

end % if expand

%u,classlist,pause

end % while samples
```

Appendix B

Super Resolution Techniques

B.1 Estimation of Signal Space Size

The detailed results obtained from an investigation into deterring the size of the signal space (i.e., rank of the correlation matrix) are presented. The investigation is described in Section 4.9.2. Each graph displays the average Freq. of occurrence of the estimated signal space size over 100 independent trials. The true size of the signal space is six (6). In total 1800 trials were conducted.

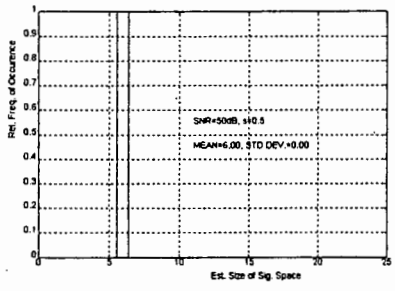


Figure B.1: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=50dB, s=0.5, MEAN=6.00, STD DEV.=0.00

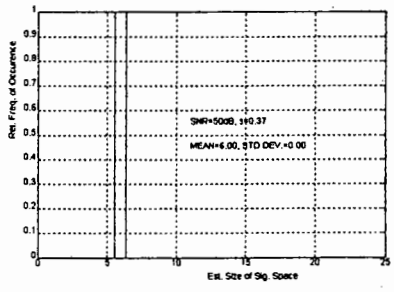


Figure B.4: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=50dB, s=0.37, MEAN=6.00, STD DEV.=0.00

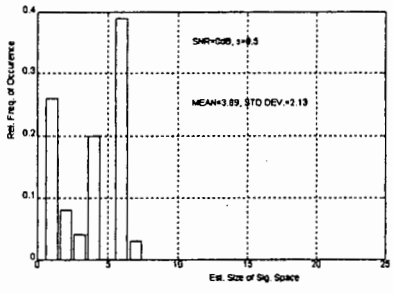


Figure B.2: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=0dB, s=0.5, MEAN=3.89, STD DEV.=2.13

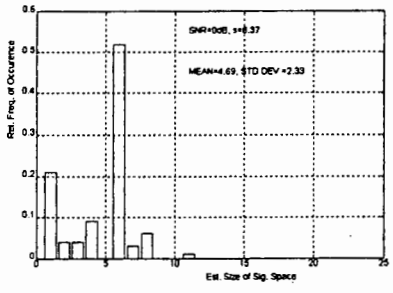


Figure B.5: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=0dB, s=0.37, MEAN=4.69, STD DEV.=2.33

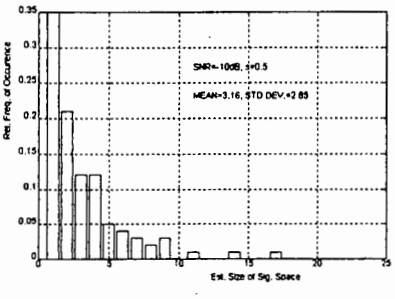


Figure B.3: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=-10dB, s=0.5, MEAN=3.16, STD DEV.=2.85

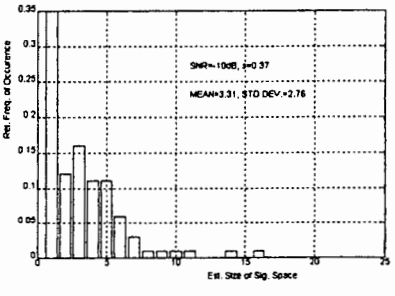


Figure B.6: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=-10dB, s=0.37, MEAN=3.31, STD DEV.=2.76

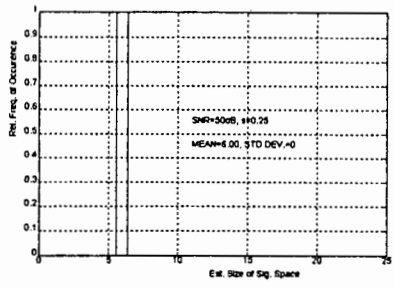


Figure B.7: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=50dB, s=0.25, MEAN=6.00, STD DEV.=0.00

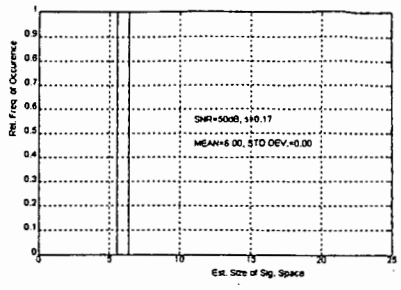


Figure B.10: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=50dB, s=0.17, MEAN=6.00, STD DEV.=0.00

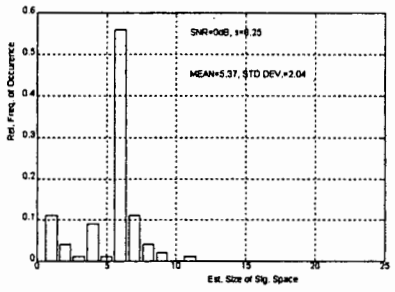


Figure B.8: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=0dB, s=0.25, MEAN=5.37, STD DEV.=2.04

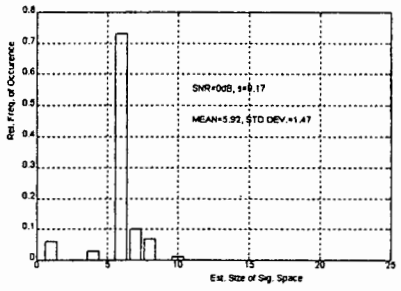


Figure B.11: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=0dB, s=0.17, MEAN=5.92, STD DEV.=1.04

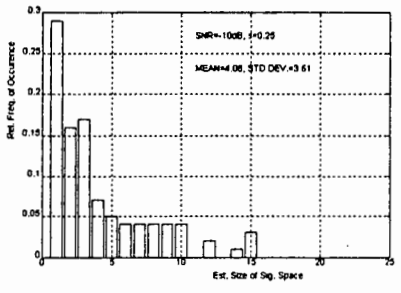


Figure B.9: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=-10dB, s=0.25, MEAN=4.08, STD DEV.=3.61

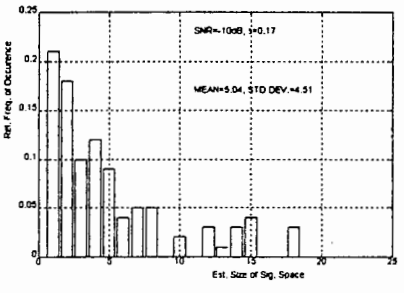


Figure B.12: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=-10dB, s=0.17, MEAN=5.04, STD DEV.=4,51

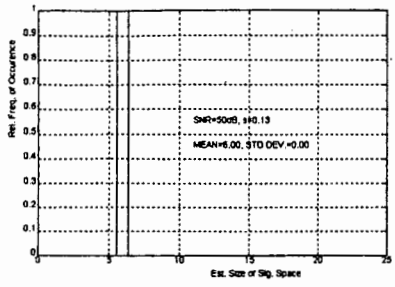


Figure B.13: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=50dB, s=0.13, MEAN=6.00, STD DEV.=0.00

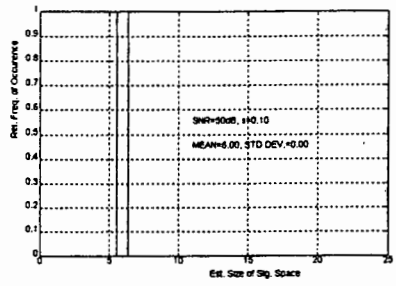


Figure B.16: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=50dB, s=0.10, MEAN=6.00, STD DEV.=0.00

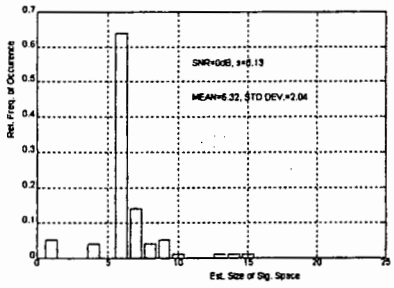


Figure B.14: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=0dB, s=0.13, MEAN=6.32, STD DEV.=2.04

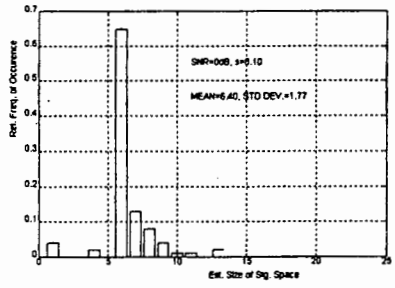


Figure B.17: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=0dB, s=0.10, MEAN=6.40, STD DEV.=1.77

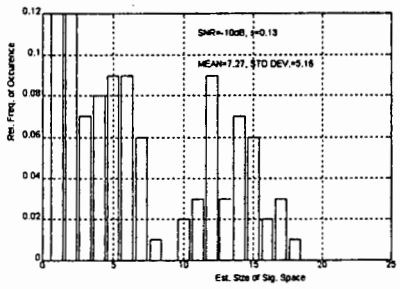


Figure B.15: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=-10dB, s=0.13, MEAN=7.27, STD DEV.=5.16

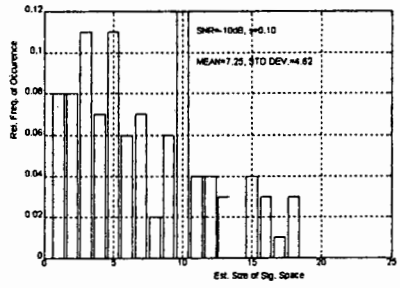


Figure B.18: Relative Freq. of Occurrence vs. Estimated Size of Signal Space. SNR=-10dB, s=0.10, MEAN=7.25, STD DEV.=4.62

B.2 Typical Results for Each Algorithm

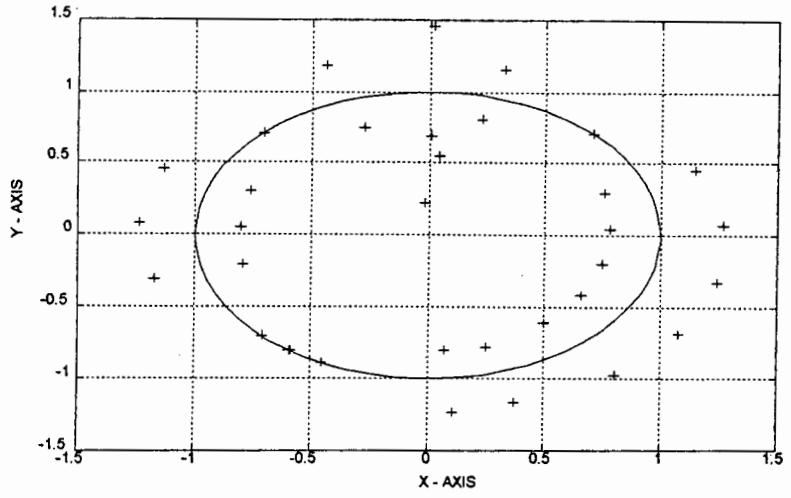


Figure B.19: MUSIC method: SNR = +15dB, L = 20, M = 6. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

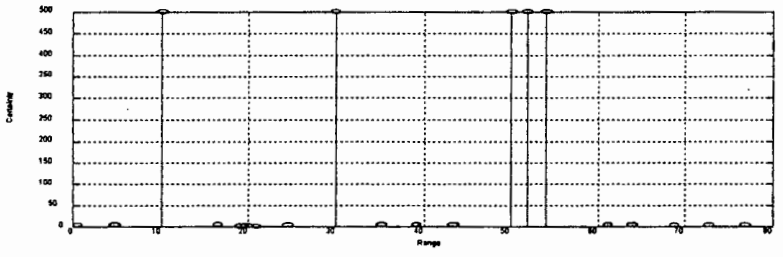


Figure B.20: MUSIC method: SNR = +15dB, L = 20, M = 6. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

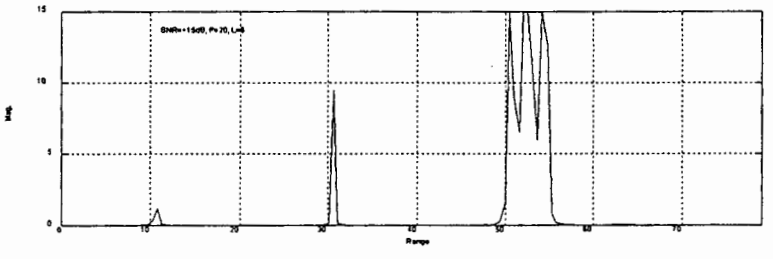


Figure B.21: MUSIC Method: SNR = +15dB, L = 20, M = 6. The corresponding spectral plot, once again derived from the same prediction error zeros.

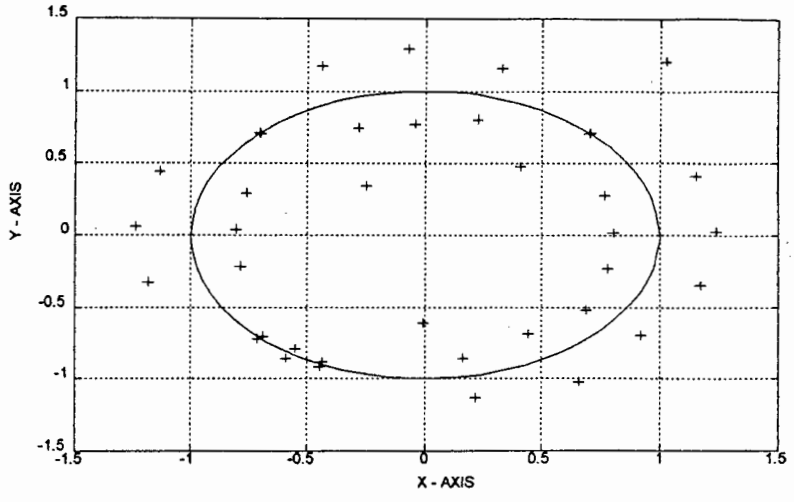


Figure B.22: MUSIC method: SNR = +1dB, L = 20, M = 6. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

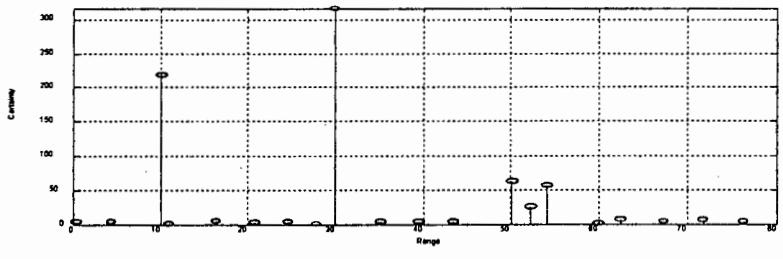


Figure B.23: MUSIC method: SNR = +1dB, L = 20, M = 6. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

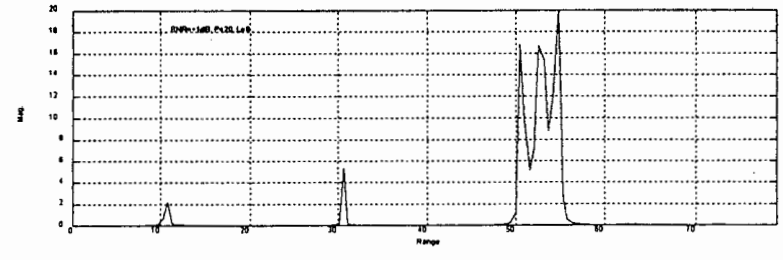


Figure B.24: MUSIC Method: SNR = +1dB, L = 20, M = 6. The corresponding spectral plot, once again derived from the same prediction error zeros.

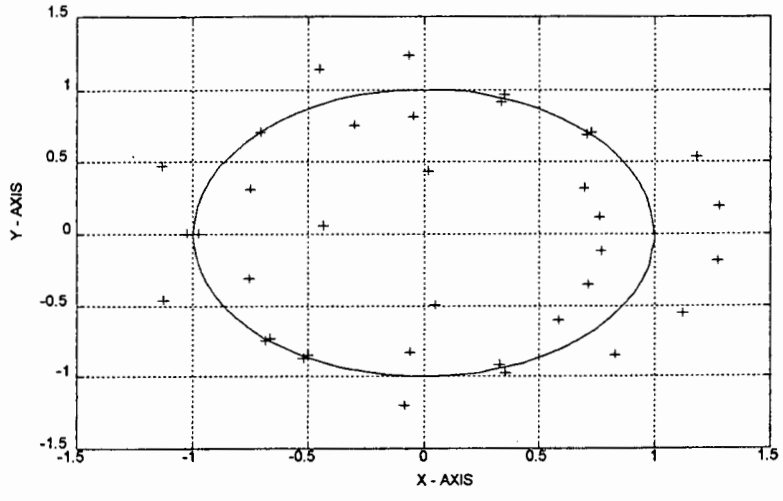


Figure B.25: MUSIC method: SNR = -5dB, L = 20, M = 6. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

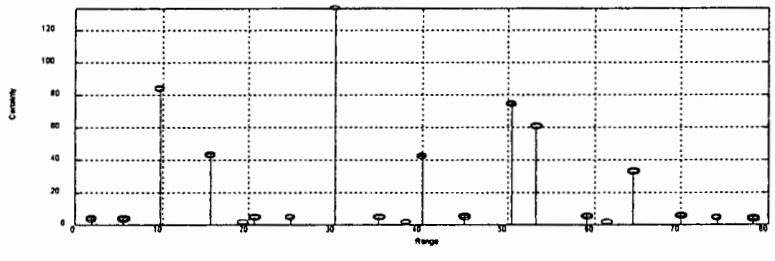


Figure B.26: MUSIC method: SNR = -5dB, L = 20, M = 6. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

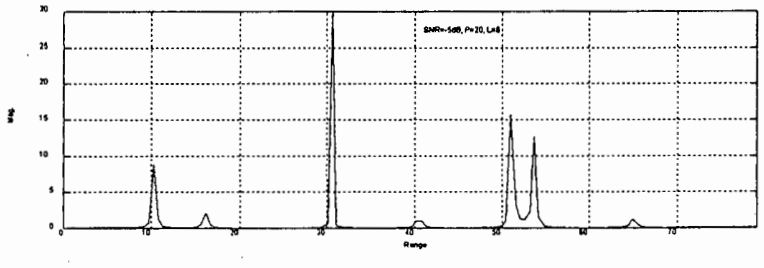


Figure B.27: MUSIC Method: SNR = -5dB, L = 20, M = 6. The corresponding spectral plot, once again derived from the same prediction error zeros.

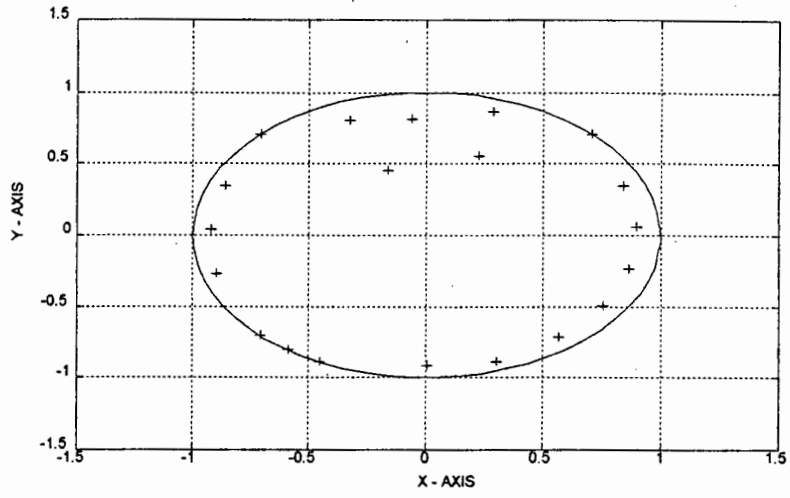


Figure B.28: PE method: SNR = +15dB, L = 20, M = 6. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

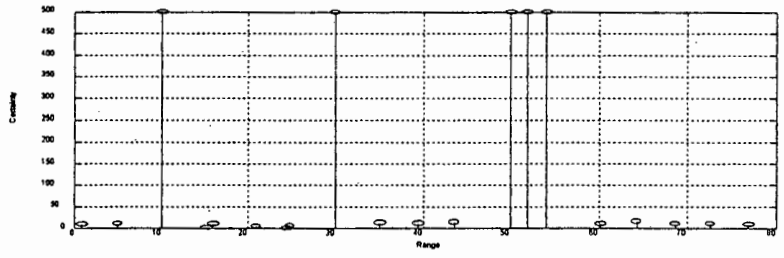


Figure B.29: PE method: SNR = +15dB, L = 20, M = 6. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

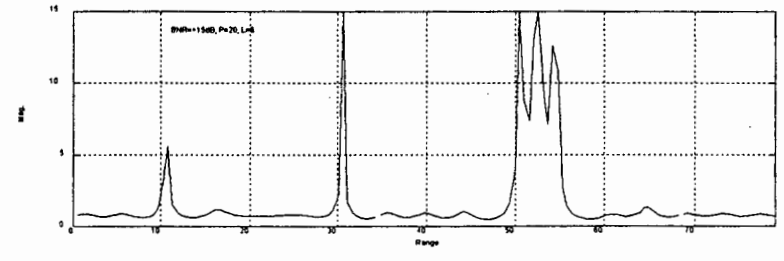


Figure B.30: PE method: SNR = +15dB, L = 20, M = 6. The corresponding spectral plot, once again derived from the same prediction error zeros.

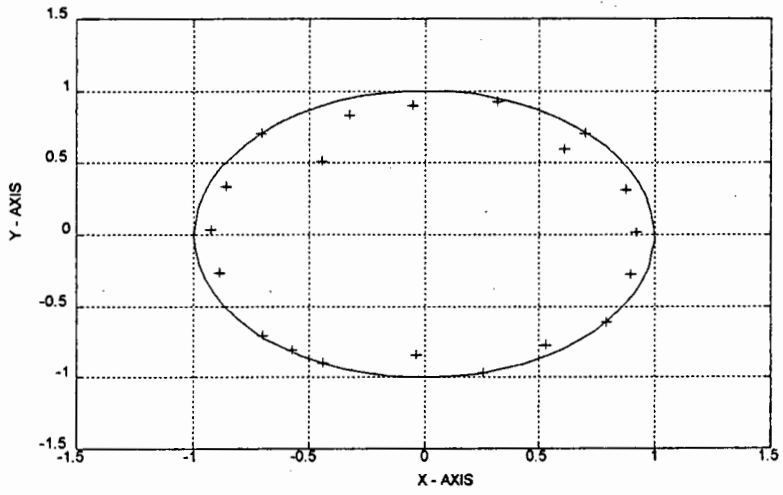


Figure B.31: PE method: SNR = +1dB, L = 20, M = 6. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

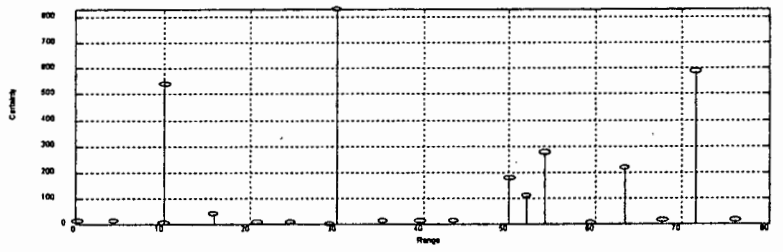


Figure B.32: PE method: SNR = +1dB, L = 20, M = 6. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

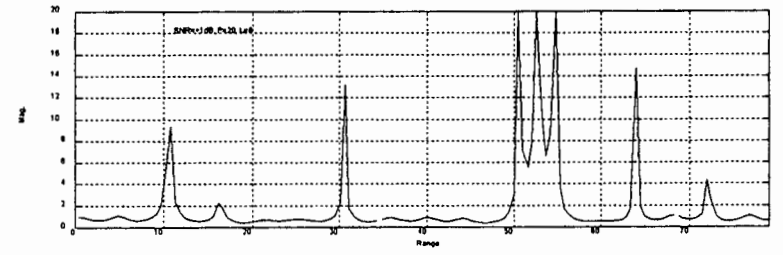


Figure B.33: PE method: SNR = +1dB, L = 20, M = 6. The corresponding spectral plot, once again derived from the same prediction error zeros.

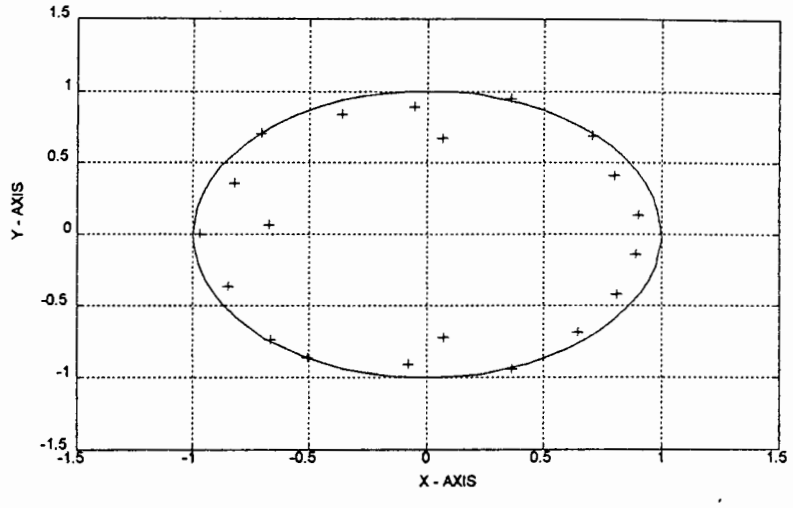


Figure B.34: PE method: SNR = -5dB, L = 20, M = 6. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

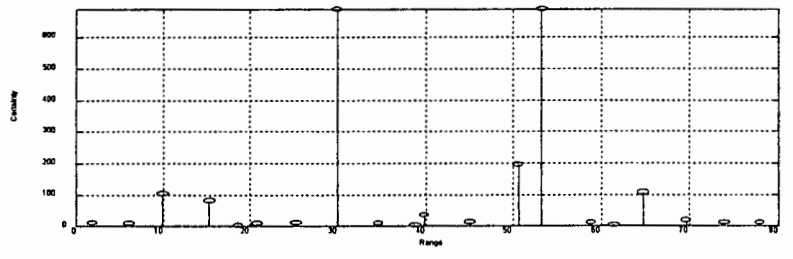


Figure B.35: PE method: SNR = -5dB, L = 20, M = 6. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

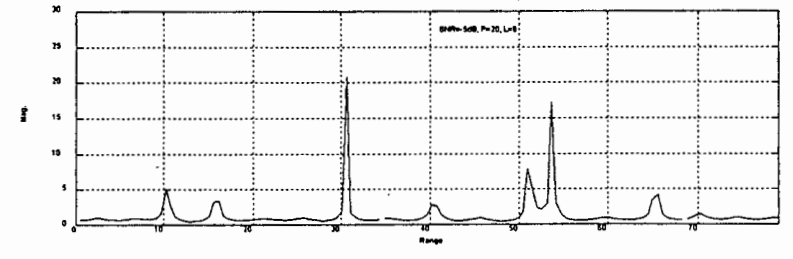


Figure B.36: PE method: SNR = -5dB, L = 20, M = 6. The corresponding spectral plot, once again derived from the same prediction error zeros.

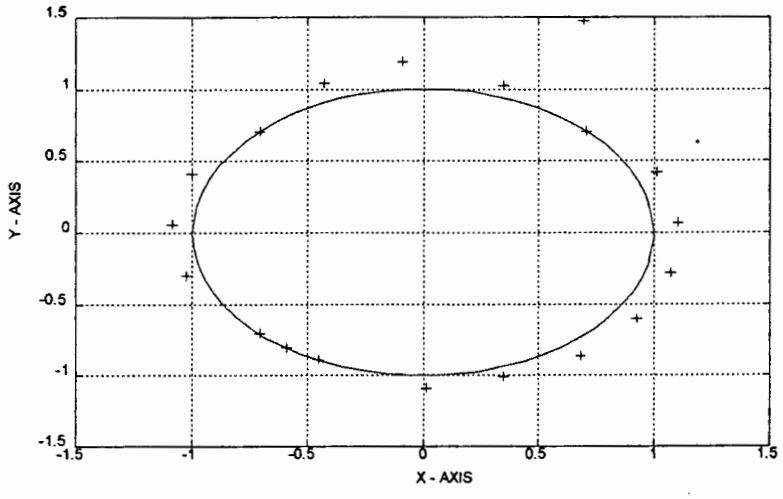


Figure B.37: TLS method: SNR = +15dB, L = 20, M = 6. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

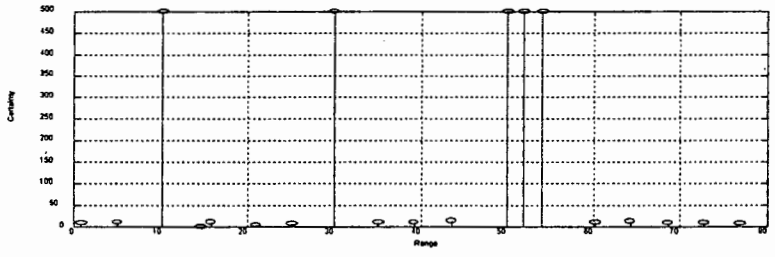


Figure B.38: TLS: SNR = +15dB, L = 20, M = 6. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

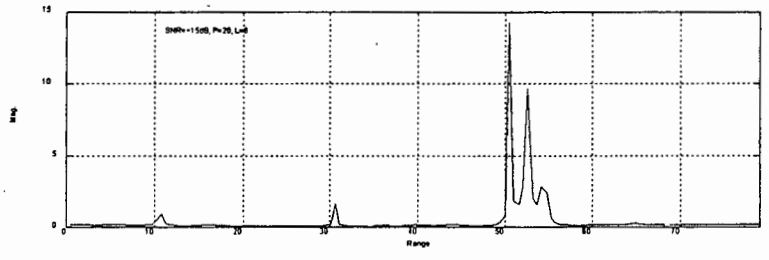


Figure B.39: TLS: SNR = +15dB, L = 20, M = 6. The corresponding spectral plot, once again derived from the same prediction error zeros.

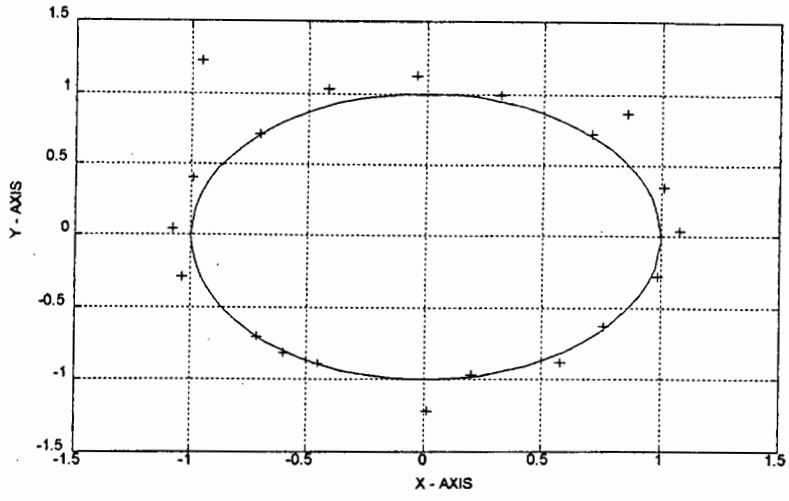


Figure B.40: TLS method: SNR = +1dB, L = 20, M = 6. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

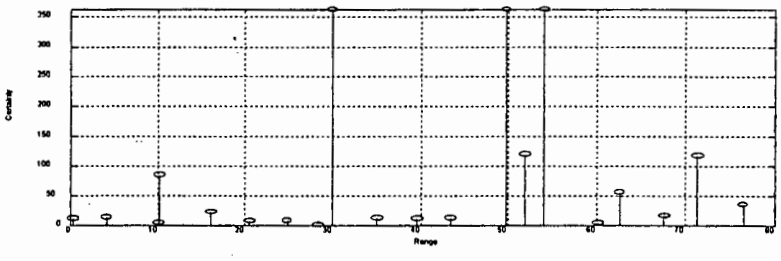


Figure B.41: TLS: SNR = +1dB, L = 20, M = 6. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

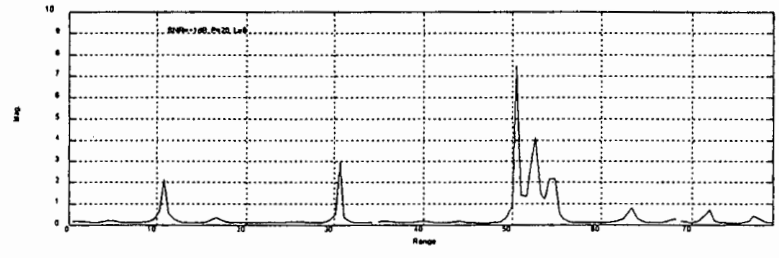


Figure B.42: TLS: SNR = +1dB, L = 20, M = 6. The corresponding spectral plot, once again derived from the same prediction error zeros.

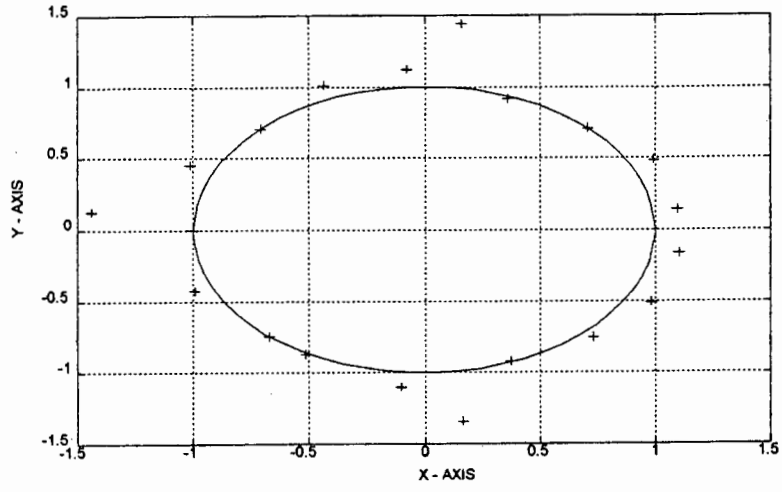


Figure B.43: TLS method: SNR = -5dB, L = 20, M = 6. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

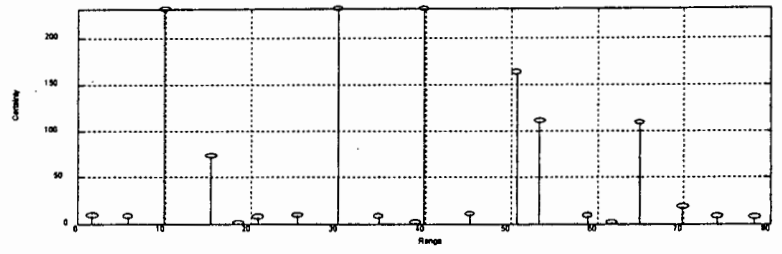


Figure B.44: TLS: SNR = -5dB, L = 20, M = 6. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

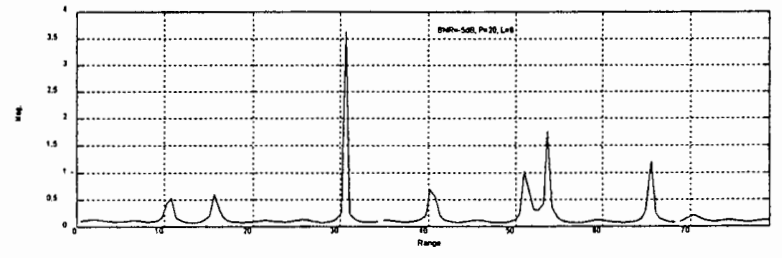


Figure B.45: TLS: SNR = -5dB, L = 20, M = 6. The corresponding spectral plot, once again derived from the same prediction error zeros.

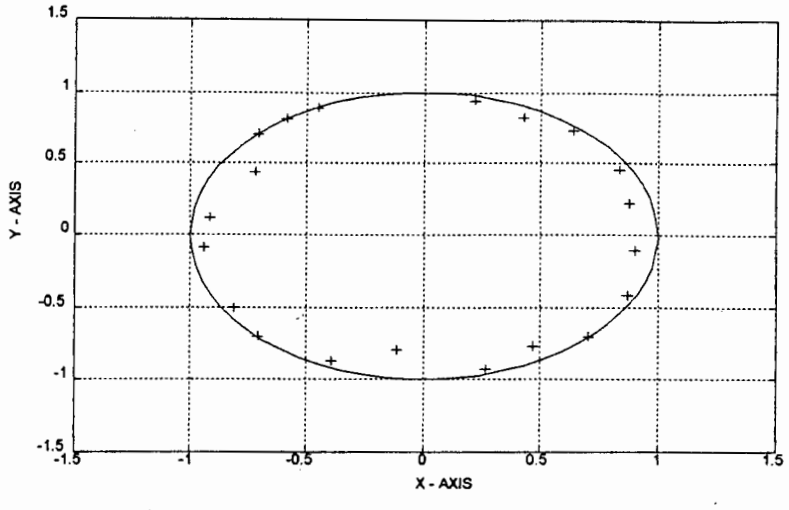


Figure B.46: Modified Covariance Method: SNR = +15dB, L=20. The zeros of the prediction error filter.

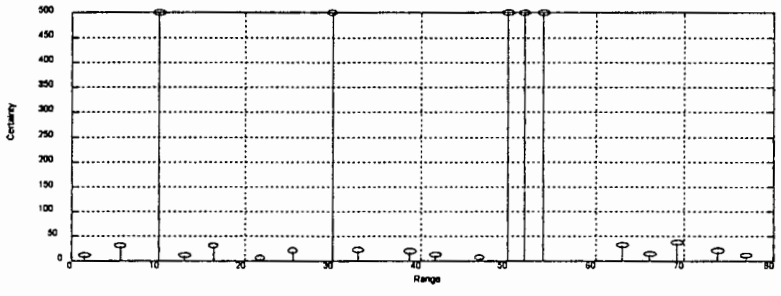


Figure B.47: Modified Covariance Method: SNR = +15dB, L=20. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

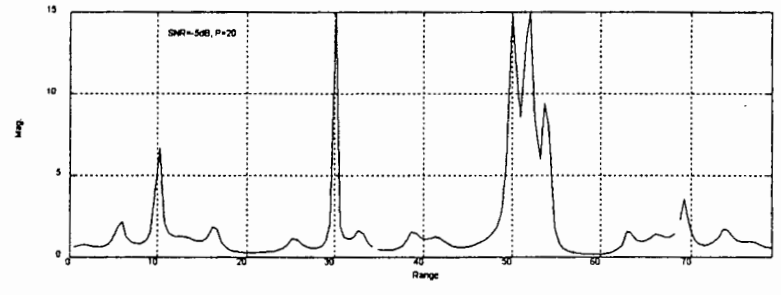


Figure B.48: Modified Covariance Method: SNR = +15dB, L=20. The corresponding spectral plot, once again derived from the same prediction error zeros.

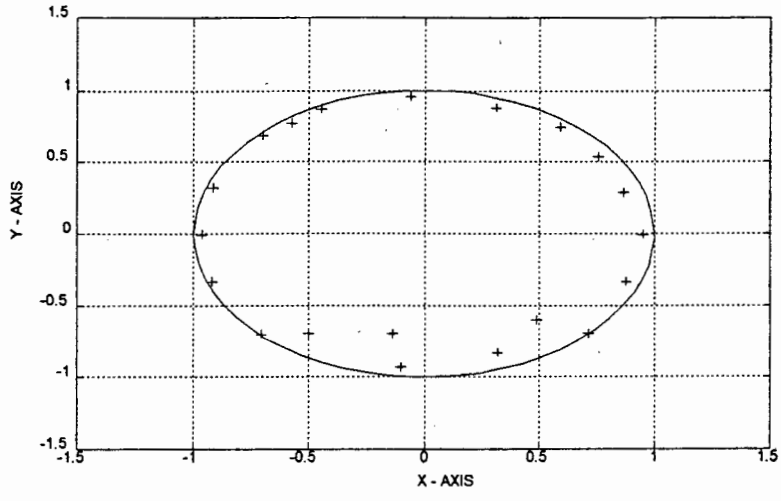


Figure B.49: Modified Covariance Method: SNR = +1dB, L=20. The zeros of the prediction error filter.

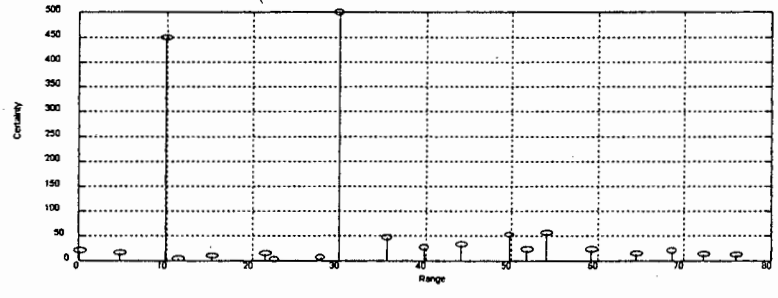


Figure B.50: Modified Covariance Method: SNR = +1dB, L=20. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

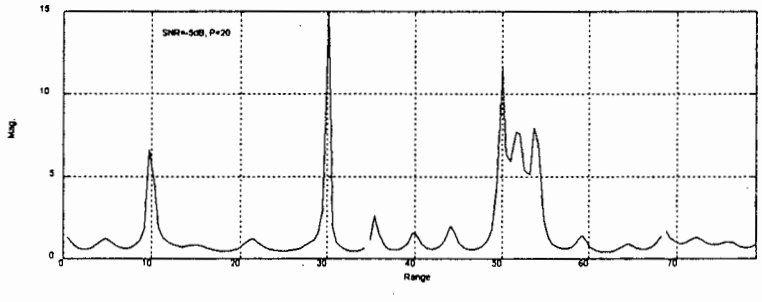


Figure B.51: Modified Covariance Method: SNR = +1dB, L=20. The corresponding spectral plot, once again derived from the same prediction error zeros.

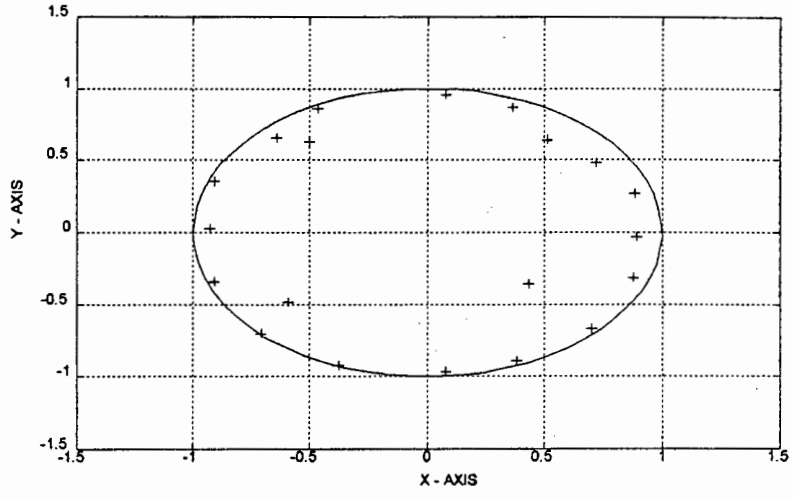


Figure B.52: Modified Covariance Method: SNR = -5dB, L=20. The zeros of the prediction error filter.

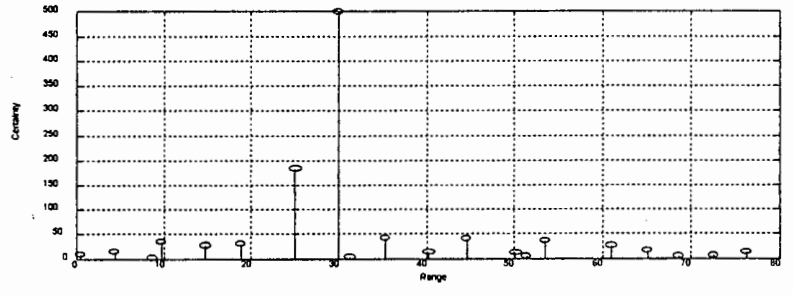


Figure B.53: Modified Covariance Method: SNR = -5dB, L=20. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

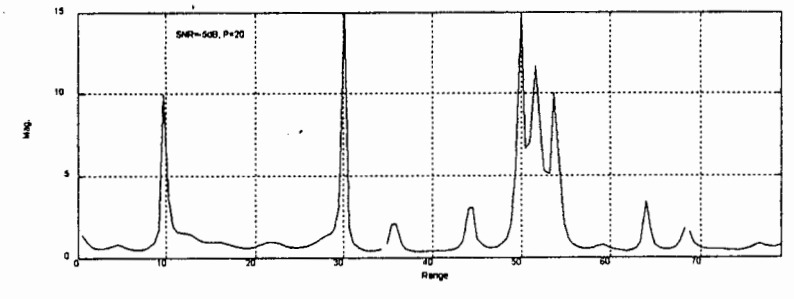


Figure B.54: Modified Covariance Method: SNR = -5dB, L=20. The corresponding spectral plot, once again derived from the same prediction error zeros.

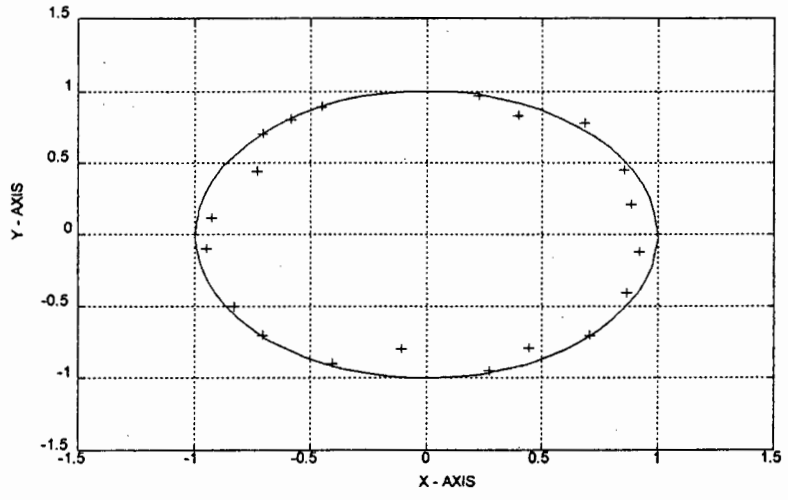


Figure B.55: Covariance Method: SNR = +15dB, L=20. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

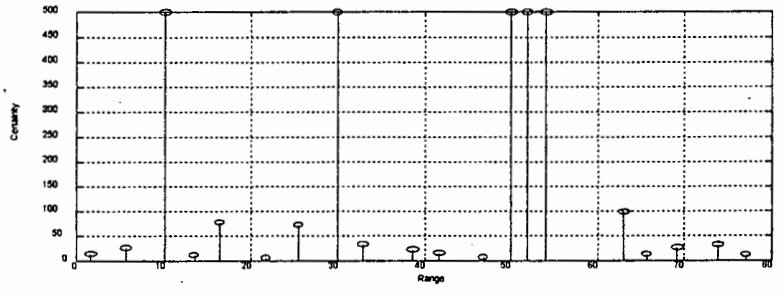


Figure B.56: Covariance Method: SNR = +15dB, L=20. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

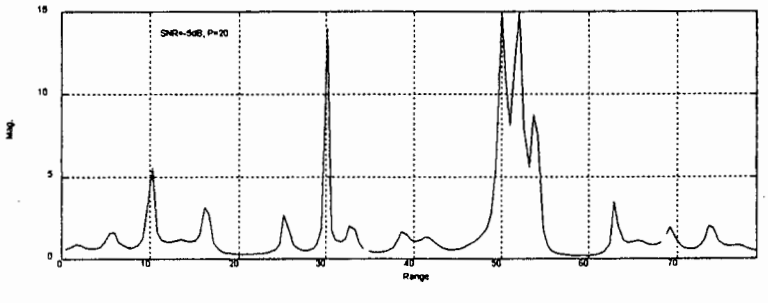


Figure B.57: Covariance Method: SNR = +15dB, L=20. The corresponding spectral plot, once again derived from the same prediction error zeros.

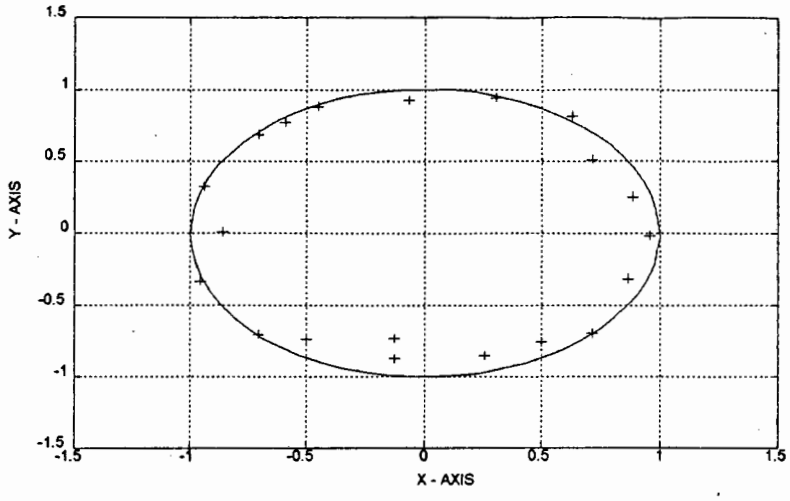


Figure B.58: Covariance Method: SNR = +1dB, L=20. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

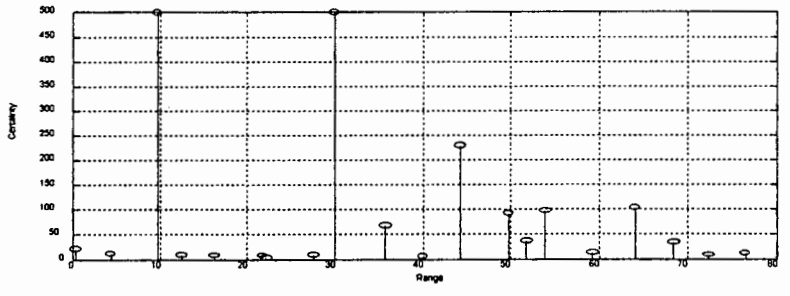


Figure B.59: Covariance Method: SNR = +1dB, L=20. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

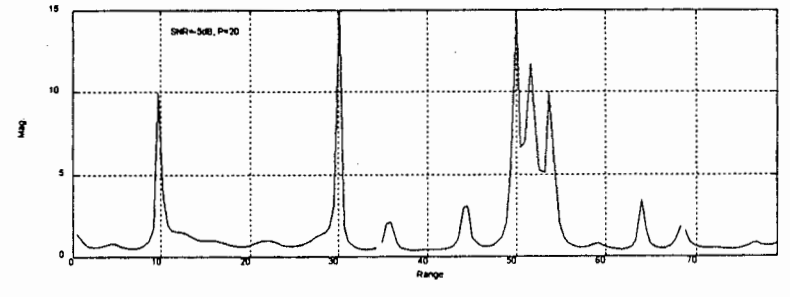


Figure B.60: Covariance Method: SNR = +1dB, L=20. The corresponding spectral plot, once again derived from the same prediction error zeros.

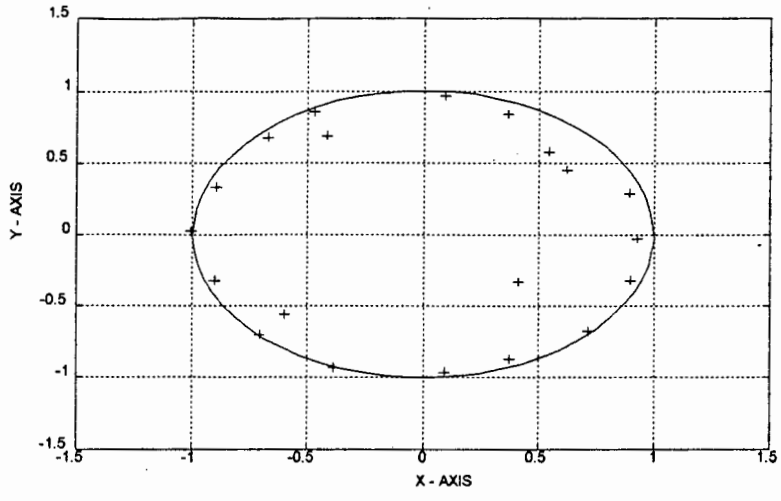


Figure B.61: Covariance Method: SNR = -5dB, L=20. The zeros of the prediction error filter. M was set equal to the true rank of the correlation matrix.

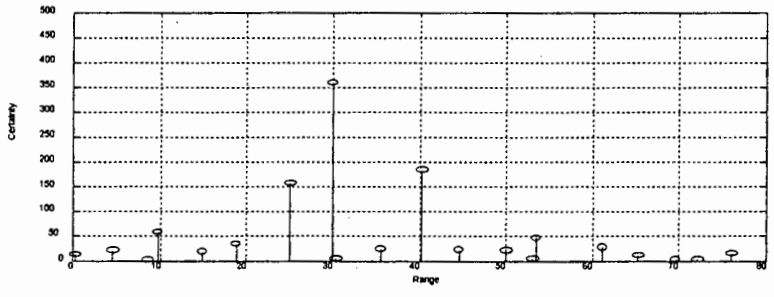


Figure B.62: Covariance Method: SNR = -5dB, L=20. Reflector range position vs. 'certainty'. The range positions of the filter zeros in the previous graph are plotted as stems, whose height or certainty is determined by the reciprocal of each pole's radial distance to the unit circle.

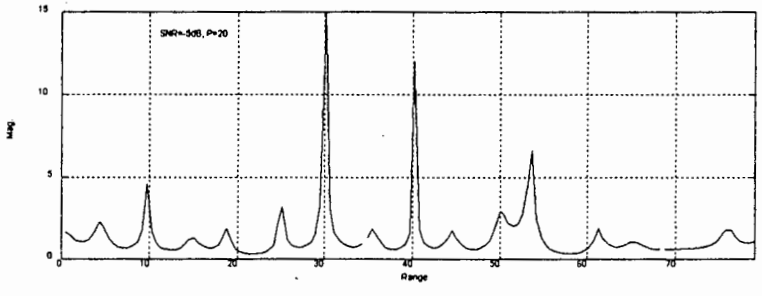


Figure B.63: Covariance Method: SNR = -5dB, L=20. The corresponding spectral plot, once again derived from the same prediction error zeros.

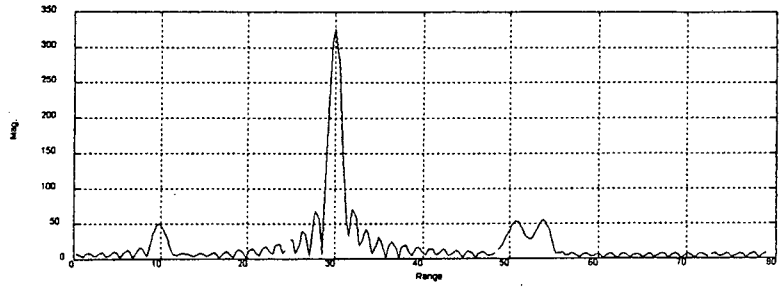


Figure B.64: FFT spectral estimate: SNR = -5dB. 155 padded zeros.

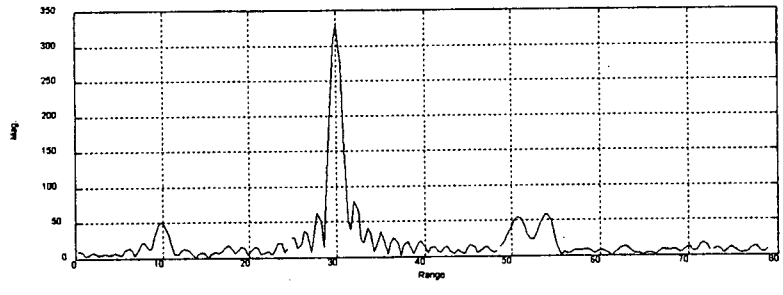


Figure B.65: FFT spectral estimate: SNR = +1dB. 155 padded zeros.

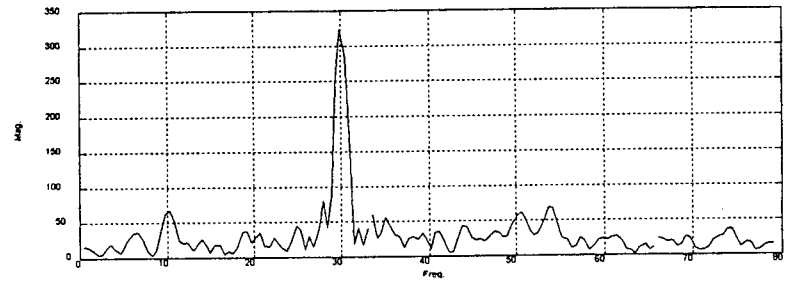


Figure B.66: FFT spectral estimate: SNR = -5dB. 155 padded zeros.

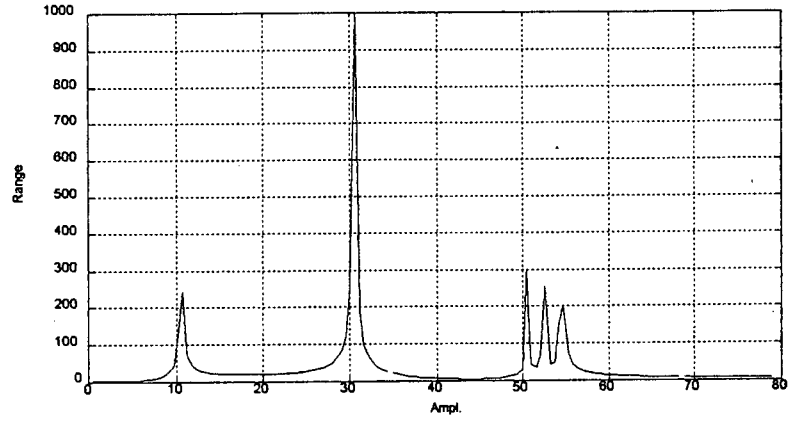


Figure B.67: Prony spectral estimate: SNR = +15dB, L=15.

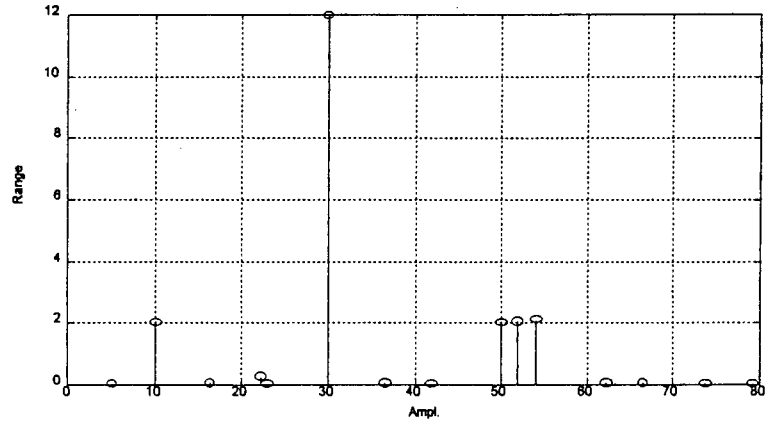


Figure B.68: Prony Method. SNR = +15dB, L=15. Corresponding plot of the Freq. vs. magnitude components of each complex Prony exponential.

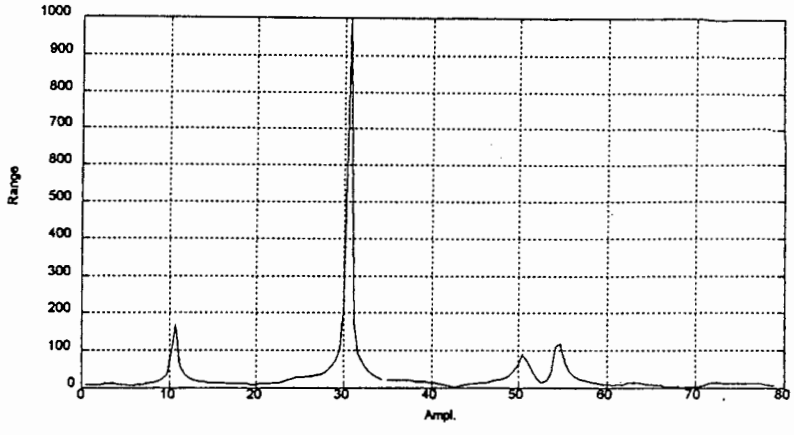


Figure B.69: Prony spectral estimate: SNR = +1dB, L=15.

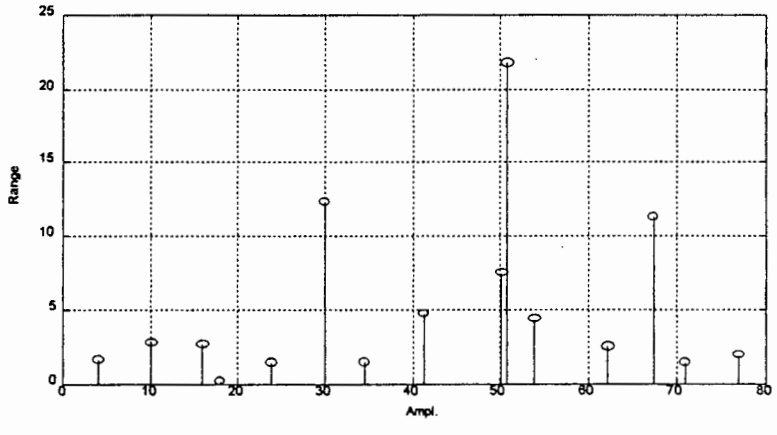


Figure B.70: Prony Method. SNR = +1dB, L=15. Corresponding plot of the Freq. vs. magnitude components of each complex Prony exponential.

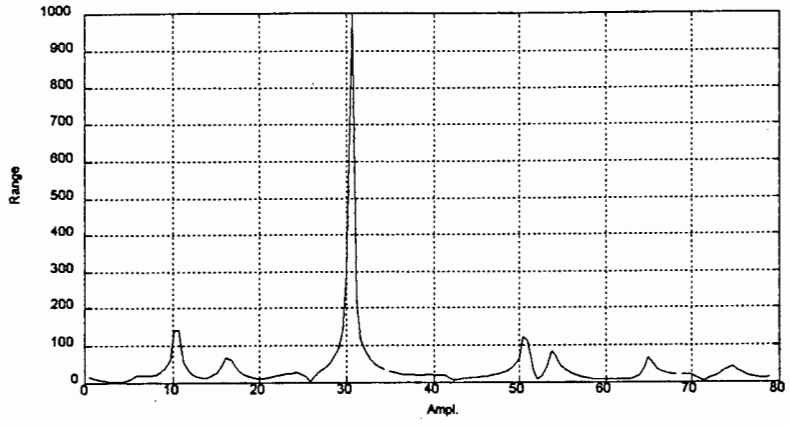


Figure B.71: Prony spectral estimate: SNR = -5dB, L=15.

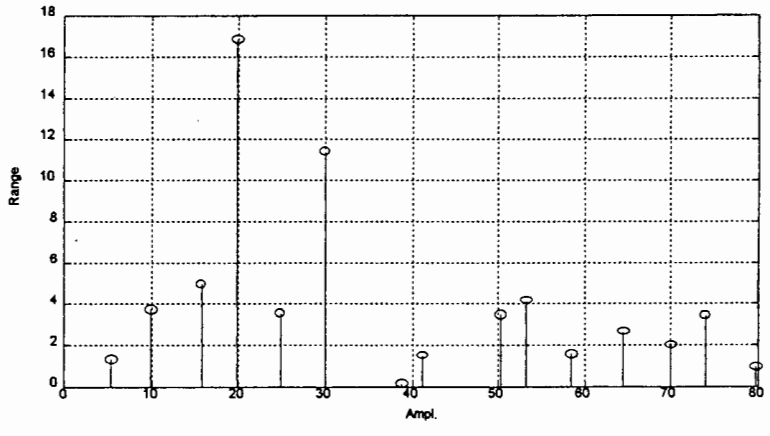


Figure B.72: Prony Method. SNR = -5dB, L=15. Corresponding plot of the Freq. vs. magnitude components of each complex Prony exponential.

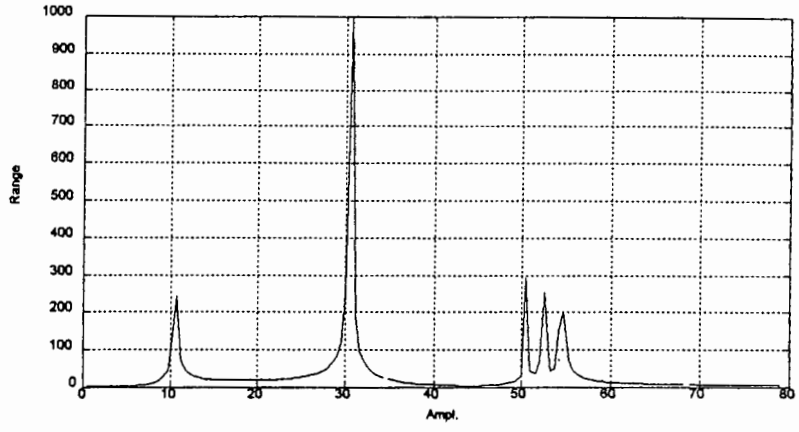


Figure B.73: Hildebrand spectral estimate: SNR = +15dB, L=15.

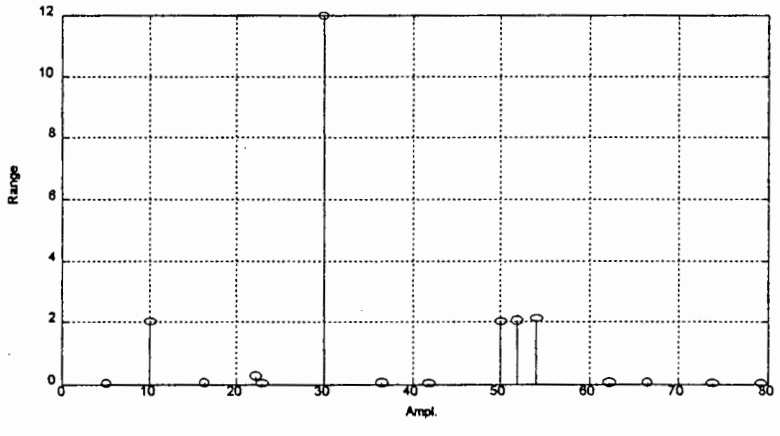


Figure B.74: Hildebrand Method. SNR = +15dB, L=15. Corresponding plot of the Freq. vs. magnitude components of each complex sinusoid.

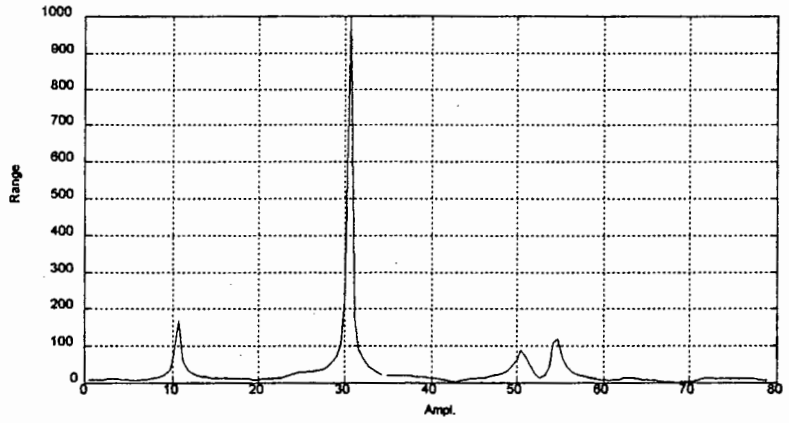


Figure B.75: Hildebrand spectral estimate: SNR = +1dB, L=15.

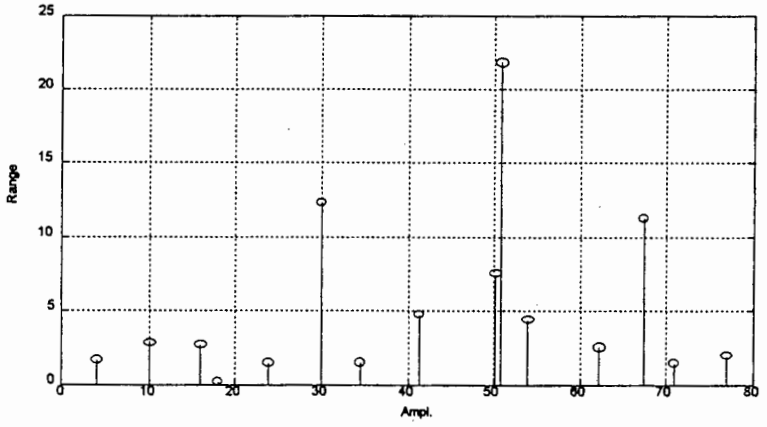


Figure B.76: Hildebrand Method. SNR = +1dB, L=15. Corresponding plot of the Freq. vs. magnitude components of each complex sinusoid.

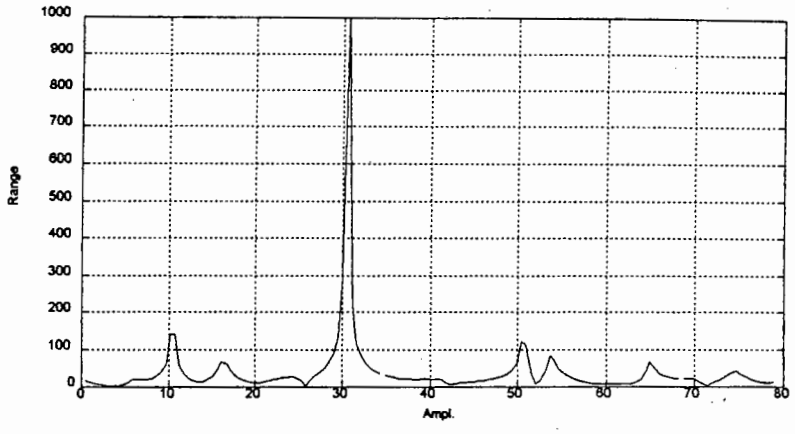


Figure B.77: Hildebrand spectral estimate: SNR = -5dB, L=15.

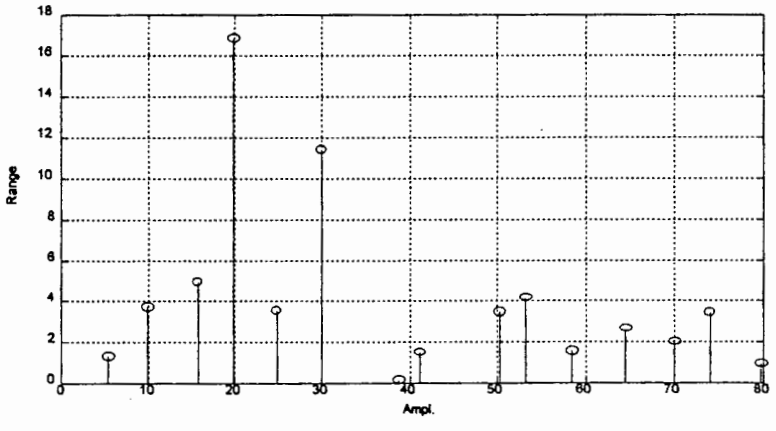


Figure B.78: Hildebrand Method. SNR = -5dB, L=15. Corresponding plot of the Freq. vs. magnitude components of each complex sinusoid.

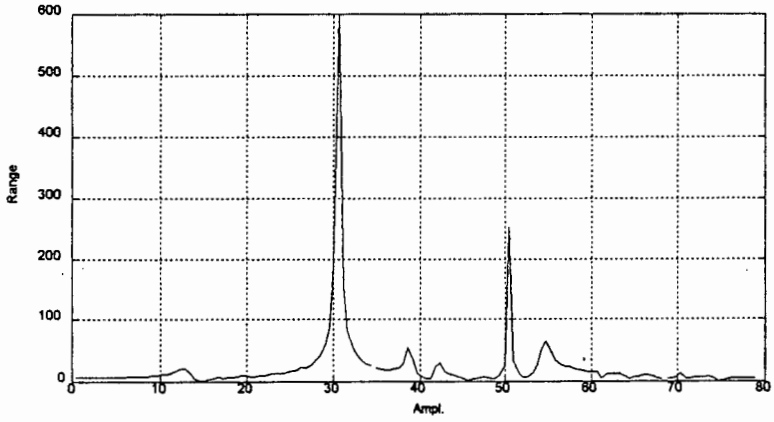


Figure B.79: Modified Prony spectral estimate. SNR = +15dB, L=20. M =6. The value of M was set equal to the true rank of the correlation matrix.

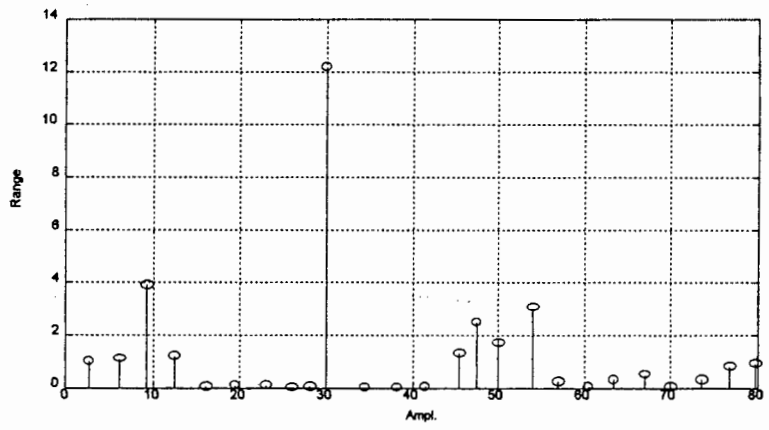


Figure B.80: Modified Prony Method: SNR = +15dB, L=20, M=6. Corresponding plot of the Freq. vs. magnitude components of each complex exponential.

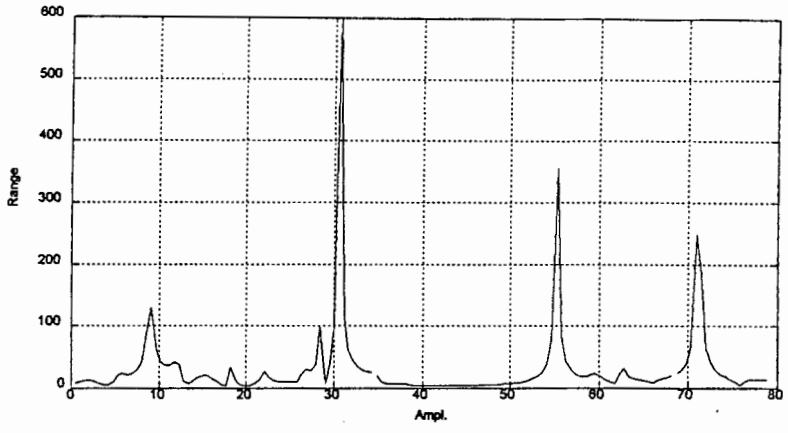


Figure B.81: Modified Prony spectral estimate. SNR = +1dB, L=20. M =6. The value of M was set equal to the true rank of the correlation matrix.

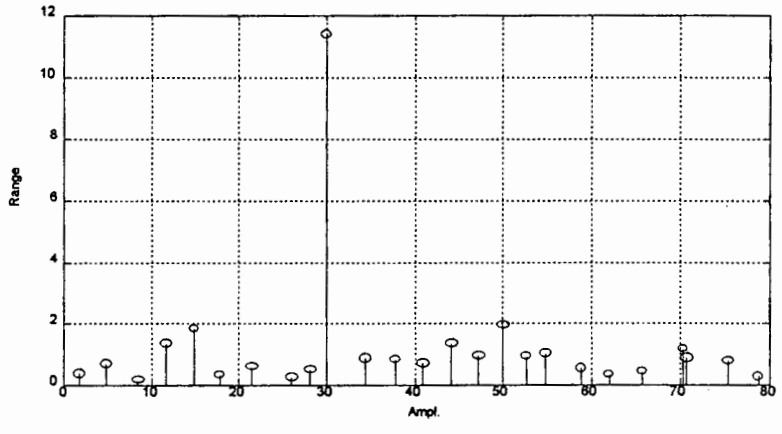


Figure B.82: Modified Prony Method: SNR = +1dB, L=20, M=6. Corresponding plot of the Freq. vs. magnitude components of each complex exponential.

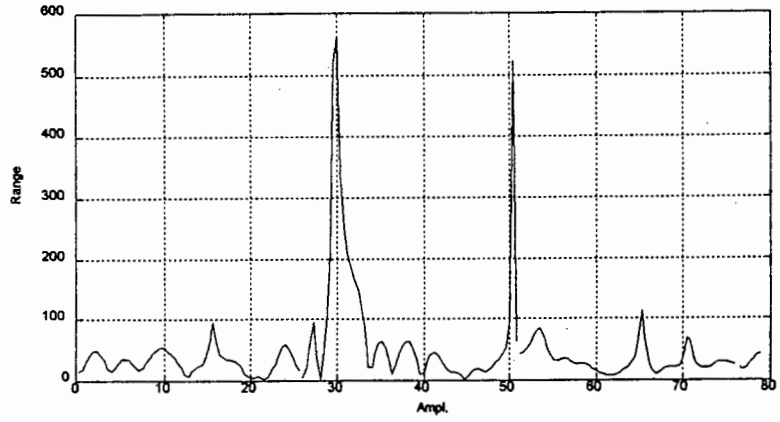


Figure B.83: W
Modified Prony spectral estimate. SNR = -5dB, L=20. M =6. The value of M was set equal to the true rank of the correlation matrix.

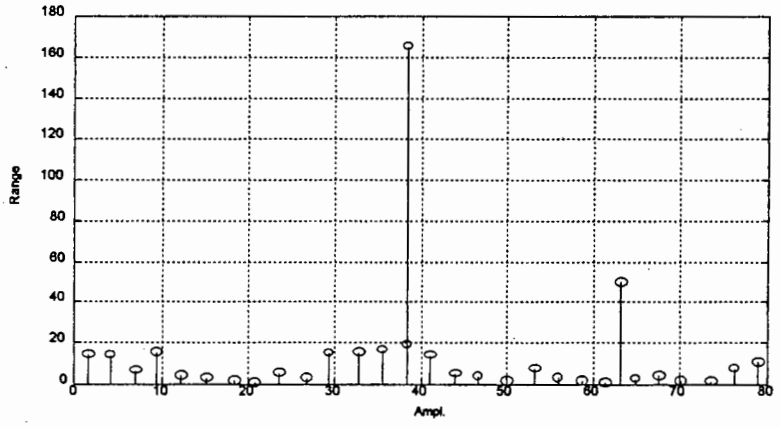


Figure B.84: Modified Prony Method: SNR = -5dB, L=20, M=6. Corresponding plot of the Freq. vs. magnitude components of each complex exponential.

B.3 Algorithms

The *MATLAB* algorithms that were used to implement the MUSIC, PE, TLS, covariance, and modified covariance algorithms are presented. The routines are documented in the code. One note regarding the calculation of the matrix inverses is required:

In all cases the inverse Ψ^{-1} is calculated using the Moore-Penrose pseudoinverse $\Psi^{\#}$, which is defined in terms of the singular value decomposition (SVD) matrix components as the unique matrix

$$\Psi^{\#} = V\Sigma^{\#}U^{H} = \sum_{i=1}^{k} \sigma_i^{-1} v_i u_i^{H}$$

where

$$\Sigma^{\#} = \begin{bmatrix} D^{-1} & 0 \\ 0 & 0 \end{bmatrix} \tag{B.1}$$

```
function [A] = antmusic(X,P,L)

% FUNCTION [A]=ANTMUSIC(X,P,L)
%
% FUNCTION TO DETERMINE THE FREQUENCY SPECTRUM OF
% INPUT VECTOR 'X' USING THE 'ROOT MUSIC' METHOD.
% USES MODIFIED COVARIANCE FOR THE CORRELATION MATRIX
%
% INPUTS:
% X=INPUT VECTOR OR MATRIX (COLUMNWISE IF MATRIX),
% P=AR MODEL ORDER, L=SIZE OF SIGNAL SPACE
%
% OUTPUTS:
% RETURNS DENOMINATOR COEFICIENTS OF D(Z), WHERE
% S(Z)=1/D(Z)
%
% WRITTEN BY ANTHONY ROBINSON
% UNIVERSITY OF CAPE TOWN, 'OCT 1994


% DEFAULT RETURNS

a=[]; A=[];


% ALLOW FOR MATRIX INPUT

[m,n] = size(X);
if min([m n])~=1;
        loops=n;
        else loops=1;,X=X(:);
end


% START LOOP

for j=1:loops

% SETUP VECTOR

x=X(:,j);
N=length(x);


%FORM CORRELATION MATRIX R

firstrow=x(P+1:-1:1);
firstcol=x(P+1:1:N);
Tp=toeplitz(firstcol,firstrow);
R = Tp' * Tp;
R = R + conj(flipud(fliplr(R)));


%COMPUTE AND ORDER EIGENVALUES

[u,s,v] = svd(R);


%SELECT NOISE SUBSPACE

E = v(:,L+1:P+1);


%FORM PROJECTION MATRIX

p = E * E';


%CALCULATE ROOTS USING FAST DIAGONAL METHOD

a = zeros(P+1,1);
for i = 0:P,
        a(i+1,1) = sum(diag(p,i));
end;

a = [conj(flipud(a(2:length(a)))) ; a];
r = abs(freqz(1,a,200,'whole'));

A =[A r];


%END LOOP

end
```

```
function [A] = anttls(X,P,L)

% FUNCTION [A]=ANTTLS(X,P,L)
%
% FUNCTION TO DETERMINE THE FREQUENCY SPECTRUM OF
% INPUT VECTOR 'X' USING A 'TOTAL LEAST SQUARES' APPROACH
% TO THE MINIMUM NORM ALGORITHM.
% USES MODIFIED COVARIANCE FOR THE CORRELATION MATRIX
%
% INPUTS:
% X=INPUT VECTOR OR MATRIX (COLUMNWISE IF MATRIX),
% P=AR MODEL ORDER, L=SIZE OF SIGNAL SPACE
%
% OUTPUTS:
% RETURNS L AR FILTER COEFFICIENTS
%
% WRITTEN BY ANTHONY ROBINSON
% UNIVERSITY OF CAPE TOWN, 'OCT 1994
%
% ALGORITHM BASED ON THE ARTICLE "TOTAL LEAST SQUARES
% APPROACH FOR FREQUENCY ESTIMATION USING LINEAR PREDICTION"
% BY ANISUR RAHMAN AND KAI-BOR YU IN TRANS. ON ACOUSTICS,
% SPEECH, AND SIGNAL PROCESSING, VOL ASSP35, NO 10 OCT '87
% PAGES 1440-1454


% DEFAULT RETURNS

a=[]; A=[];


% ALLOW FOR MATRIX INPUT

[m,n] = size(X);
if min([m n])~=1;
        loops=n;
        else loops=1;,X=X(:);
end


% START LOOP

for j=1:loops

% SETUP VECTOR

x=X(:,j);
N=length(x);



for i=1:N-P
        for j=1:P+1
        B(i,j)=x(i+j-1);
        end
end



% COMPUTE AND ORDER EIGENVALUES

[u,s,v] = svd(B);


% SELECT NOISE SUBSPACE

E = v(:,L+1:P+1);



% COMPUTE AND NORMALIZE Q

[m,n]=size(E);
Q=conj(E(m,:));
Q=Q/sum(abs(Q));



% COMPUTE y AND alpha

Y_ALPHA=E*conj(Q');
y=Y_ALPHA(1:length(Y_ALPHA)-1);
alpha=Y_ALPHA(length(Y_ALPHA));


% COMPUTE c

c=y/alpha;

% GET IN CORRECT FORM

a = [c;1];
a = reshape(a,1,length(a));
r = abs(freqz(1,a,200,'whole'));

A =[A r];


%END LOOP

end
```

```
function [A]=antmdcov(X,P);

% FUNCTION [A] = ANTMDCOV(X,P)
%
% FUNCTION TO DETERMINE THE FREQUENCY SPECTRUM OF
% INPUT VECTOR 'X' USING THE 'MODIFIED COVARIANCE' METHOD.
% MODELS DATA AS AR FILTER
%
% INPUTS:
% X=INPUT VECTOR OR MATRIX (COLUMNWISE IF MATRIX),
% P=NO. FILTER POLES
%
% OUTPUTS:
% A=AR FILTER COEFFICIENTS
%
% WRITTEN BY ANTHONY ROBINSON
% UNIVERSITY OF CAPE TOWN, 'OCT 1994

% DEFAULT RETURNS

a=[]; A=[];


% ALLOW FOR MATRIX INPUT

[m,n] = size(X);
if min([m n])~=1;
        loops=n;
        else loops=1;,X=X(:);
end


% START LOOP

for j=1:loops


% SETUP VECTOR

x=X(:,j);
N=length(x);


% FORM MODIFIED COVARIENCE MATRIX

N=length(x);
toprow=x(P+1:-1:1);
topcol=x(P+1:1:N);
Tp=toeplitz(topcol,toprow);
R = Tp' * Tp;
R = R + conj(flipud(fliplr(R)));


% SOLVE FOR AR COEFICIENTS

[m,n]=size(R);
b = R(2:m,2:n);
c = -(R(2:m,1));
a = pinv(b) * c;


% GET IN CORRECT FORM

a = [1;a];
a = reshape(a,1,length(a));
r = abs(freqz(1,a,200,'whole'));

A =[A r];


%END LOOP

end
```

```
function [A]=antcov(X,P);

% FUNCTION [A] = ANTCOV(X,P)
%
% FUNCTION TO DETERMINE THE FREQUENCY SPECTRUM OF
% INPUT VECTOR 'X' USING THE 'COVARIANCE' METHOD.
% MODELS DATA AS AR FILTER
%
% INPUTS:
% X=INPUT VECTOR OR MATRIX (COLUMNWISE IF MATRIX),
% P=NO. FILTER POLES
%
% OUTPUTS:
% A=AR FILTER COEFFICIENTS
%
% WRITTEN BY ANTHONY ROBINSON
% UNIVERSITY OF CAPE TOWN, 'OCT 1994

% DEFAULT RETURNS

a=[]; A=[];

% ALLOW FOR MATRIX INPUT

[m,n] = size(X);
if min([m n])~=1;
        loops=n;
        else loops=1;,X=X(:);
end


% START LOOP

for j=1:loops


% SETUP VECTOR

x=X(:,j);
N=length(x);


% FORM COVARIENCE MATRIX

N=length(x);
toprow=x(P+1:-1:1);
topcol=x(P+1:1:N);
Tp=toeplitz(topcol,toprow);
R = Tp' * Tp;
R = conj(flipud(fliplr(R)));


% SOLVE FOR AR COEFICIENTS

[m,n]=size(R);
b = R(2:m,2:n);
c = -(R(2:m,1));
a = pinv(b) * c;


% GET IN CORRECT FORM

a = [1;a];
a = reshape(a,1,length(a));
r = abs(freqz(1,a,200,'whole'));
A =[A r];


%END LOOP

end
```

```
function [A] = antmnorm(X,P,L)

% FUNCTION [A]=ANTMNORM(X,P,L)
%
% FUNCTION TO DETERMINE THE FREQUENCY SPECTRUM OF
% INPUT VECTOR 'X' USING THE 'MINIMUM NORM' METHOD.
% USES MODIFIED COVARIANCE FOR THE CORRELATION MATRIX
%
% INPUTS:
% X=INPUT VECTOR OR MATRIX (COLUMNWISE IF MATRIX),
% P=AR MODEL ORDER, L=SIZE OF SIGNAL SPACE
%
% OUTPUTS:
% RETURNS L AR FILTER COEFFICIENTS
%
% WRITTEN BY ANTHONY ROBINSON
% UNIVERSITY OF CAPE TOWN, 'OCT 1994


% DEFAULT RETURNS

a=[]; A=[];


% ALLOW FOR MATRIX INPUT

[m,n] = size(X);
if min([m n])~=1;
        loops=n;
        else loops=1;,X=X(:);
end


% START LOOP

for j=1:loops


% SETUP VECTOR

x=X(:,j);
N=length(x);



% FORM CORRELATION MATRIX R

firstrow=x(P+1:-1:1);
firstcol=x(P+1:1:N);
Tp=toeplitz(firstcol,firstrow);
R = Tp' * Tp;
R = R + conj(flipud(fliplr(R)));


% COMPUTE AND ORDER EIGENVALUES

[u,s,v] = svd(R);


% SELECT NOISE SUBSPACE

E = v(:,L+1:P+1);


% BREAK INTO gn AND GN

[m,n]=size(E);
gnTrans=E(1,:);
gn=conj(gnTrans');
GN=E(2:m,:);


% COMPUTE MINIMUM NORM COEFFICIENTS

w=pinv(gn'*gn)*conj(GN)*gn;


% GET IN CORRECT FORM

a = [1;w];
a = reshape(a,1,length(a));
r = abs(freqz(1,a,200,'whole'));

A =[A r];


%END LOOP

end
```

```
function z = align(y,howmuch)

% FUNCTION [Z] = ALIGN(Y,HOWMUCH)
%
% ALIGN ITERATIVELY ALIGNS VECTORS IN Y BY CORRELLATION PEAKS
% THE ALGORITH IMPLEMENTED HERE IS BASED ON THEORY FROM
% "CORELATION FILTER FOR AIRCRAFT"
% SCOTT HUDSEN AND DEMETRI PSALTIS
% IEEE TRANS. ON AER. AND ELEC. SYST. VOL.29 NO.3 JUL93
%
% INPUTS:
%
% Y=INPUT MATRIX WHERE EACH COLUMN REPRESENTS VECTOR TO
% BE ALIGNED
% HOWMUCH=NO. OF ITERATIONS (1 IS OFTEN SUFFICIENT)
%
% OUPUTS:
%
% Z = OUPUT MATRIX OF SAME FORMAT AS Y, BUT WITH EACH COLUMN
% CIRCULARLY SHIFTED TO ACHIEVE CORRELATION ALIGNMENT.
%
% WRITTEN BY ANTHONY ROBINSON
% UNIVERSITY OF CAPE TOWN, FEB. 1994


% FIRST CHECK INPUT MATRIX

[m,n]=size(y);
if m==1
        m=n;
end

bins=m;


% NORMALIZING

mag=(y.*y);
magtot=sum(mag);
for k=1:n
        y(:,k) = mag(:,k)/magtot(k);
end


% CREATE MEAN FILTER

ytot=y(:,1);
for loop=2:n
ytot=ytot+y(:,loop);
end


for iterations=1:howmuch

for profile=1:n

        home,profile

        %UPDATE MEAN FILTER
        ytot=ytot-y(:,profile);

        %CROSS CORRELATION
        %anthero=conv(y(:,profile),flip(ytot));

        %FASTER CUSTOM-DESIGNED CORRELATION
        ant1=[y(:,profile);zeros(bins,1)];
        ant2=[ytot;zeros(bins,1)];
        anthero=real(ifft(fft(ant1).*fft(flip(ant2))));

        %FIND ALIGN POINTER
        [temp,pointer]=max(anthero);

        %ALIGN
        y(:,profile)=shifter(y(:,profile),pointer);

        %UPDATE MEAN FILTER
        ytot=ytot+y(:,profile);


end     %-PROFILE
end     %-ITERATIONS
z=y;

% END
```

Appendix C

Correlation Filters

For a given target, the filter $f = [f(0), f(1), \ldots, f(k)]$ must be computed which maximizes

$$\Phi_1 = E[peak(f \star p)] \tag{C.1}$$

the expected value of the correlation peak between it and

$$p = [p(0), p(1), \ldots, p(K)]$$

, where p is any randomly chosen profile of the target, and the expected value is over all possible profiles. The correlation between the real-valued vectors f and p is give by

$$(f \star p)(s) = \sum_{j=1}^{K-1} f(j)p(j-s) \tag{C.2}$$

for $0 \leq s \leq K-1$. K is the number of range bins, and it is assumed that the correlation is circular, i.e., $p[j \pm K] = p[j]$. The goal is to create a library of filters, $f^{(1)}, f^{(2)}, \ldots, f^{(m)}$, with M the number of targets to be identified. If a profile from an unknown target is correlated against this library producing numbers

$$\alpha^{(k)} = peak(f^{(k)} \star p) \tag{C.3}$$

(correlation peaks) the hope is that

$$\alpha^{(j)} > \alpha^{(k)} \tag{C.4}$$

for all $k \neq j$, where the j^{th} filter is the "correct" one, i.e., the one which corresponds to the true identity of the unknown target. In practice there are N profiles recorded for a given target, all of which can be used to compute the filter which maximizes the correlation peaks with future, randomly drawn profiles from the same target. Towards this end define an estimate

$$\Phi = \frac{1}{N} \sum_{n=1}^{N} peak(f \star p_n) \tag{C.5}$$

where $p_1, p_2, \ldots, p_N$ is the set of recorded radar profiles for this particular target. Now assume that it is already known what f is, and the peak value of $(f \star p)(s)$ occurs at $s = s_n$. Then it is possible to form

$$\acute{p}_n(k) = p_n(k - s_n) \tag{C.6}$$

so that

$$\begin{aligned} peak(f \star p_n) &= (f \star p)(s_n) \\ &= \sum_j f(j) p_n(j - s_n) \\ &= \sum_j f(j) \acute{p}_n(j) \\ &= f.\acute{p}_n \end{aligned} \tag{C.7}$$

(since circular correlations are assumed, these shifts should also be circular; any data shifted off one end must be shifted into the other. If desired, the profiles can always be zero padded to avoid such edge effects.) In other words, it is possible to 'align' p_n with the filter so that the correlation peak occurs at zero shift. Then

$$\begin{aligned} \Phi_1 &= \frac{1}{N} \sum_{n=1}^{N} f.\acute{p}_n \\ &= f(\frac{1}{N} \sum_{n=1}^{N} \acute{p}_n) \end{aligned} \tag{C.8}$$

This dot product is the quantity that the filter f must maximize. The magnitude of f is unimportant for identification purposes since changing it merely scales its correlation with all vectors, so that there is no loss in the generality if we constrain f to be a unit vector. This unit vector f which maximizes equation C.8 is simply

$$f = \frac{\sum_{n=1}^{N} \acute{p}_n}{\left| \sum_{n=1}^{N} \acute{p}_n \right|} \tag{C.9}$$

and so

$$\begin{aligned} \Phi_1 &= \frac{\sum_{n=1}^{N} \acute{p}_n}{\left(\frac{1}{N} \sum_{n=1}^{N} \acute{p}_n \right)} \left(\frac{1}{N} \sum_{n=1}^{N} \acute{p}_n \right) \\ &= \frac{1}{N} \left| \sum_{n=1}^{N} \acute{p}_n \right| \end{aligned} \tag{C.10}$$

While f is not known *a priori*, the above analysis makes it clear that whatever it is, Φ is given by equation C.10, and the filter that achieves this is given by equation C.9. Therefore, to maximize $E[peak(f \star p)]$ it is only necessary to find the shifts s_n that maximize C.10 and then evaluate C.9 to compute the desired filter.

The practical difficulty is how to find the sets of shifts $[s_1, s_2, \ldots, s_N]$ that take the set $[p_1, p_2, \ldots, p_N]$ into the set $[\acute{p}_1, \acute{p}_1, \ldots, \acute{p}_N]$ that gives the filter via C.9. A global search over all possible shifts is out of the question

because of the astronomical computational costs involved. According to the paper by Hudson and Psaltis [38], an iterative procedure for maximizing C.10 is proposed, in which a global search is performed over only the shifts of one profile at a time:

$$\begin{aligned} \left|\sum_{n=1} N\acute{p}_k\right| &= \left|\acute{p}_j + \sum_{n\neq j} \acute{p}_n\right|^2 \\ &= |\acute{p}_j|^2 + \left|\sum_{n\neq j} \acute{p}_n\right|^2 + 2\acute{p}_j \sum_{n\neq j} \acute{p}_n \end{aligned} \tag{C.11}$$

where $\acute{p}_j$ is one of the profiles. The only term that will change when $\acute{p}_j$ is shifted is $2\acute{p}_j \sum_{k\neq j} \acute{p}_k$, and increasing this will increase the quantity we are trying to maximize. Therefore, with respect to the shift s_j, the aim is to make $\acute{p}_j \sum_{k\neq j} \acute{p}_k$ as big as possible. A global search over s_j can be done efficiently using the fast Fourier transform. Having done this for one profile, it is then possible to do it for the others and start the process over. Each shift increases $|\sum_{k=1} N\acute{p}_k|$, and eventually a maximum is obtained. This result is not guaranteed to be a maximum. The procedure is similar to iterative line maximization techniques used to find extrema of functions of several variables, and it may potentially suffer the same problems of local maxima. However, in the applications investigated by Hudson and Psaltis [38] the algorithm was found to converge to the same result for several different starting configurations.

C.1 Program Code

```
function y = filter2(b,x,shape)
%FILTER2 Two-dimensional filtering.
%
%       Y = FILTER2(B,X) filters the data in X with the 2-D FIR
%       filter in the matrix B.  The result, Y, is computed
%       using 2-D convolution and is the same size as X.
%
%       Y = FILTER2(B,X,'shape') returns Y computed via 2-D
%       convolution with size specified by 'shape':
%         'same'  - (default) returns the central part of the
%                   convolution that is the same size as X.
%         'valid' - returns only those parts of the convolution
%                   that are computed without the zero-padded
%                   edges, size(Y) < size(X).
%         'full'  - returns the full 2-D convolution,
%                   size(Y) > size(X).
%
%       See also CONV2.

%       Clay M. Thompson 9-17-91
%       L. Shure 10-21-91, modified to use conv2 mex-file
%       Copyright (c) 1984-93 by The MathWorks, Inc.

error(nargchk(2,3,nargin));
if nargin<3, shape = 'same'; end

code = [shape,' ']; code = code(1);
if isempty(find(code=='svf')), error('Unknown shape parameter.'); end

[mx,nx] = size(x);
stencil = rot90(b,2);
[ms,ns] = size(stencil);

y = conv2(x,stencil,shape);
```

```
function y = flip(x);

% FUNCTION [Y] = FLIP(X)
%
% FUNCTION FOR USE BY ALIGN TO FLIP A VECTOR
%
% INPUTS:
%
% X = VECTOR OF FORM [X(1),X(2),X(3),...X(M-1),X(M)]
%
% OUTPUTS:
%
% Y = VECTOR OF FORM [X(M),X(M-1),...X(3),X(2),X(1)]
%
% WRITTEN BY ANTHONY ROBINSON
% UNIVERSITY OF CAPE TOWN, FEB. 1994


[m,n]=size(x);
if m==1
        m=n;
end

for i=1:1:m/2
        buffer=x(i);
        x(i)=x(m+1-i);
        x(m+1-i)=buffer;


end

y=x;

% END
```

```
function z = shifter(y,shift)

% FUNCTION [Z] = SHIFTER(Y,SHIFT)
%
% FUNCTION FOR USE BY ALIGN TO CIRCULARLY SHIFT A VECTOR
%
% INPUTS:
%
% Y = VECTOR OF FORM [X(1),X(2),X(3),...X(M-1),X(M)]
% SHIFT = SCALAR NUMBER REPRESENTING THE NUMBER OF
% SHIFTS TO THE LEFT.
%
% OUTPUTS:
%
% Y = VECTOR OF FORM [X(1-SHIFT),X(2-SHIFT),...
%                     X(M-1-SHIFT),X(M-SHIFT)]
%
% WRITTEN BY ANTHONY ROBINSON
% UNIVERSITY OF CAPE TOWN, FEB. 1994


[m,n]=size(y);
if m==1
        m=n;
end
buffer=y;
for i=1:m
        shift_index=rem(i+shift,m);
        if shift_index==0
                shift_index=m;
        end
        y(i)=buffer(shift_index);
end
z=y;

% END
```

Bibliography

[1] A. D. Robinson and M. R. Inggs. Ship Target Recognition Using Low Resolution Radar and Neural Networks. In Press, *IEEE Trans on Aerospace and Electronic Systems*

[2] A. D. Robinson and M. R. Inggs. Neural Approaches to Ship Target Recognition *1995 IEEE Internation Radar Conference, Washington DC*, pp 195-201, May 1995

[3] M.R. Inggs, A.D. Robinson and A. Langman NCTR Phase 4 report. *UCT Radar Remote Sensing Group*, rrsg04:93, May 1993.

[4] M.R. Inggs and A.D. Robinson NCTR Phase 5 report. *UCT Radar Remote Sensing Group*, rrsg05:94, May 1994.

[5] M.W. van Zyl and M.R. Inggs. Theory of Synthetic High Resolution Range Profiles. *UCT Radar Remote Sensing Group*, rrsg02:92, May 1992.

[6] M.R. Inggs. SRP Implemented on Simulated Ideal Radars. *UCT Radar Remote Sensing Group*, rrsg05:92, 1992.

[7] Bir Bhanu. Automatic target recognition: State of the art survey. *IEEE Trans on Aerospace and Electronic Systems*, 22(4):364–378, July 1986.

[8] Scott Hudsen and Demetri Psaltis. Correlation filters for aircraft identification from radar range profiles. *IEEE Trans on Aerospace and Electronic Systems*, 29(3):741–748, July 1993.

[9] J. S. Chen and E. K. Walton. Comparison of Two Target Classification Techniques. *IEEE Trans on Aerospace and Electronic Systems*, Volume AES-22, no. 1, January 1986.

[10] MD. Anisur Rahman and Kai-Bor Yu. Total Least Squares Approach for frequency Estimation Using Linear Prediction. *IEEE Trans of Acoustics, Speech, and Signal Processing*, Volume ASSP-35, no. 10, October 1987.

[11] V. Umapathi Reddy, Arogyaswami Paulraj and Thomas Kailath. Performance Analysis of the Optimum Beamformer in the presence of Correlated Sources and its Behaviour Under Spacial Smoothing *IEEE Trans of Acoustics, Speech, and Signal Processing*, Volume ASSP-35, no. 7, July 1987.

[12] Stanislav B. Kesler and Varaz Shahmirian. Bias and Resolution of the MUSIC and the Modified FBLP Algorithms in the Presence of Coherent Plane Waves. *IEEE Trans of Acoustics, Speech, and Signal Processing*, Volume 36, no. 8, August 1988.

[13] Bhaskar D. Rao and K. V. S. Hari. Performance Analysis of Root-Music. *IEEE Trans of Acoustics, Speech, and Signal Processing*, Volume 37, no. 12, December 1989.

[14] Ralph O. Schmidt. Multiple Emitter Location and Signal Parameter Estimation. *IEEE Trans on Antennas and Propagation*, Volume AP-34, no. 3, March 1986.

[15] Hiroyoshi Yamada and Manabu Ohmiya and Yasutaka Ogawa. Super-resolution Techniques for Time-Domain Measurements with a Network Analyzer. *IEEE Trans on Antennas and Propagation*, Volume 39, no. 2, February 1991.

[16] Rob Carriere and Randolph L. Moses. High Resolution Radar Modeling Using a Modified Prony Estimator. *IEEE Trans on Antennas and Propagation*, Volume 40, no. 1, January 1992.

[17] H. Sakai. Statistical properties of AR spectral analysis. *IEEE Trans of Acoustics, Speech, and Signal Processing*, Volume ASSP-27, August 1979.

[18] Donald W. Tufts and Ramdas Kumareson. Singular Value Decomposition and Improved Frequency Estimation Using Linear Prediction. *IEEE Trans of Acoustics, Speech, and Signal Processing*, Volume ASSP-30, no. 4, August 1982.

[19] R.E.Dubroff. The effective autocorrelation function of maximum entropy spectra. *Proceedings of the IEEE*, Volume 63, November 1975.

[20] Donald W. Tufts and Ramdas Kumareson. Estimation of Frequencies of Multiple Sinusoids: Making Linear Prediction Perform Like Maximum Likelihood. *Proceedings of the IEEE*, Volume 70, no. 9, September 1982.

[21] Fredric J. Haris. On the use of Windows for harmonic analysis with the Discrete Fourier Transform. *Proceedings of the IEEE*, Volume 66, no. 1:51-83, January 1978.

[22] Steven M. Kay and Stanley Lawrence Marple, JR. Spectrum Analysis – A Modern Perspective. *Proceedings of the IEEE*, Volume 69, no. 11, November 1981.

[23] John Makhoul. Linear Prediction: A Tutorial Review. *Proceedings of the IEEE*, Volume 63, no. 4, April 1975.

[24] Scroth, A et al. Ground-based polarimetric radar imaging at DLR: an overview. *Proceedings of the SPIE Radar Polarimetry Conference*, San Diego, Ca., 23-24 July, 1992. Volume 1748, pp97-122.

[25] Merrill I. Skolnik. *Introduction to Radar Systems.* McGraw-Hill, 1962.

[26] Steven M. Kay. *Modern Spectral Estimation: Theory and applications.* Prentice-Hall, 1986.

[27] Jr S. Lawrence Marple. *Digital Spectral Analysis with applications.* Prentice-Hall, 1988.

[28] S. Haykin. *Topics in Applied Physics: Nonlinear Methods of Spectral Analysis.* Springer-Verlag, 1979.

[29] M. R. Inggs, J. Hurwitz and A. Langman. Synthetic Range Profile Measurements of Aircraft. *Procedings of the South African Communications and Signal Processing Conference*, 1993. COMSIG-93 Conference, IEEE South Africa Section.

[30] M. R. Inggs, M.W. van Zyl and A. Knight. A Simulation of Synthetic Range Profile Radar *Procedings of the South African Communications and Signal Processing Conference* , 1993. COMSIG-93 Conference, IEEE South Africa Section.

[31] A.D. Robinson and M. R. Inggs. Correlation Filters Applied to Synthetic Range Profiles of Aircraft Targets *Procedings of the South African Communications and Signal Processing Conference* , 1994. COMSIG-94 Conference, IEEE South Africa Section.

[32] Scroth, A et al. Ground-based polarimetric radar imaging at DLR: an overview. *Proceedings of the SPIE Radar Polarimetry Conference*, San Diego, Ca., 23-24 July, 1992. Volume 1748, pp97-122.

[33] M.W. Roth. Neural Network Technology and its applications. *Heuristics* , 2(1):46-62, 1989.

[34] A. Kabriskey, S. Rogers, D. Ruck and G. Tarr. Artificial neural networks for automatic target recognition. *Proceedings of the SPIE*, 1294 (APRIL 18-20):2-12, April 90.

[35] Cong Deng and Simon Haykin. Classification of Radar Clutter Using Neural Networks. *IEEE Trans on Neural Networks*, 2(6):589–600, November 93.

[36] Nabil H. Farhat. Microwave Diversity Imaging and Automated Taget Identification Based on Models of Neural Networks. *IEEE Proceedings*, 77(5):670–681, May 1989.

[37] Guo Guirong, Zhang Wei, and Yi Wenxian. An Intelligence Recognition Method of Ship Targets. *Institute of Electrical & Electronic Engineers*, 3:1088–1096, 1989.

[38] Scott Hudsen and Demetri Psaltis. Correlation Filters for Aircraft Identification from Radar Range Profiles. *IEEE Trans on Aero and Electronic Systems*, 29(3):741–748, July 1993.

[39] Ismail Jouny, E.D. Garber, and S.C. Ahalt. Classification of Radar Targets Using Synthetic Neural Networks. *IEEE Trans on Aero and Electronic Systems*, 29(2):336–343, April 1993.

[40] Teuvo Kohonen. The Self-Organizing Map. *IEEE Proceedings*, 78(9):1464–1478, September 1990.

[41] H.K. Kwan and Chi Kin Lee. A Neural Network Approach to Pulsed Radar Detection. *IEEE Trans on Aero and Electronic Systems*, 29(1):9–21, January 1993.

[42] Patrick K. Simpson. Fuzzy Min-Max Neural Networks-Part1: Classification. *IEEE Trans on Neural Networks*, 3(5):776–787, September 92.

[43] Merrill I. Skolnik. *Introduction to Radar Systems*. McGraw-Hill, 1962.

[44] Ian Sneddon. *The Uses of Integral Transforms*. McGraw-Hill Book Company, New York, 1972.

[45] G.B. Wilson. Radar Classification Using a Neural Network. *Proceedings of the SPIE*, 1294(APRIL18-20):200–210, April 90.

[46] Philip E Zwicke and Imre Kiss, Jr. A New Implementation of the Mellin Transform and its Application to Radar Classification of Ships. *IEEE Trans on Pattern Anal. and Machine Intelligence*, PAMA-5(2):192-199, March 1983.

[47] T.S. Douglas. Ship Target Recognition System. *University of Cape Town Radar Remote Sensing Group*, November 1992

[48] H. Seretta *University of Cape Town Radar Remote Sensing Group Report*, 1996

[49] Anton Krantz. Target Recognition System. *University of Cape Town Radar Remote Sensing Group*, June 1993

[50] Anthony. D. Robinson. Ship Target Recgnition Using Neural Networks. *University of Cape Town Radar Remote Sensing Group*, June 1994

[51] Teuvo Kohonen, Jari Kangas, Jorma Laaksonen and Kari Torkkola. The Learning Vector Quantization Programming Package Version 2.1., *Internet site cochlea.hut.fi /pub/lvqpak*, October 9, 1992

[52] G. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition structure. *Computer Vision, Graphics, and Image Understanding*, vol. 37, pp. 54-115, 1987.

[53] A. Kandel. *Fuzzy Mathematical Techniques with Applications*. Reading, MA: Addison-Wesley, 1986.

[54] D. R. Wehner *High Resolution Radar* Artech House, Inc., 1987

[55] G. S. Gill Step Frequency Waveform Design and Processing for Detection of Moving Targets in Clutter *1995 IEEE International Radar Conference, Washington DC*, pp 573-578, May 1995.

Publications List

[1] A. D. Robinson and M. R. Inggs. Ship Target Recognition Using Low Resolution Radar and Neural Networks. In Press, *IEEE Trans on Aerospace and Electronic Systems*

[2] A. D. Robinson and M. R. Inggs. Neural Approaches to Ship Target Recognition *1995 IEEE Internation Radar Conference, Washington DC*, pp 195-201, May 1995

[3] A.D. Robinson and M. R. Inggs Correlation Filters Applied to Synthetic Range Profiles of Aircraft Targets *Procedings of the South African Communications and Signal Processing Conference* , 1994. COMSIG-94 Conference, IEEE South Africa Section.

[4] A. D. Robinson and M. R. Inggs. The Application of Neural Networks with Fourier Mellin Transforms to Radar Ship Target Recognition. *South African Workshop on Pattern Recognition, Institute for Maritime Technology (IMT), Simon's town.*, 1993.