DSSS Detection and Direction Finding Methods for a DSP-based Small Aperture DF System

David Durrett

A dissertation submitted to the Department of Electrical Engineering,
University of Cape Town, in fulfilment of the requirements
for the degree of Master of Science in Engineering.

Cape Town, December 2005

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own.

Signature of Author signature removed

Cape Town

20 December 2005

Abstract

This dissertation discusses an investigation into methods for detection and direction finding of direct-sequence spread-spectrum signals (DSSS), appropriate for a particular DSP-based direction finding platform. Relevant details on DSSS and radio direction finding are presented. These are used as the theoretical basis for the development of a software-based simulator of the direction finding platform, which in turn is used to develop and investigate various algorithms. The algorithms included use a least-squares model-based approach. The results of experiments run on simulated data indicate that this approach has merit for the automated detection and direction finding of DSSS signals. The least-squares algorithm applied to a single capture of power-spectrum data is shown to perform comparably with human experts.

Acknowledgements

I would like to express my gratitude to the following people for their assistance:

My supervisor, Dr. Andrew Wilkinson.

Peralex Electronics (Pty) Ltd for making this research possible.

All those at "The Firm", who've helped in various ways, from discussing technical details to participating in experiments. In particular, Trevor, Mark and Mike for their sustained interest and input.

Contents

List of Figures

List of Tables

List of Symbols

Δt	—	Time difference
$\Delta \psi$	—	Phase difference
ϕ	—	Direction of arrival (azimuth)
ω	—	Frequency (radians/s)
λ	—	Wavelength
a_x	—	*xth* antenna, where x is a natural number
ap	—	Aperture-product
B	—	Transmitted RF bandwidth
c	—	Speed of light
C	—	Channel capacity
d	—	Antenna separation distance
f	—	Frequency (Hz)
f_s	—	A/D sampling frequency
f_c	—	Signal centre frequency
S/N	—	Signal to noise ratio
$v_a(t)$	—	Time domain signal on antenna a
$V_a(\omega)$	—	Frequency domain signal on antenna a
W	—	Bandwidth (Shannon-Hartley)

Nomenclature

ADC — Analogue to Digital Converter.

Aperture — a pair of antennas, or a baseline.

Aperture-product — the product of the Fourier-transform of the signal on one antenna with the conjugate of the Fourier-transform of the signal on another antenna.

Azimuth — Angle in a horizontal plane, relative to a fixed reference, usually north or the longitudinal reference axis of the aircraft or satellite.

BPSK — Binary Phase Shift Keyed.

DF — Direction Finding.

DOA — Direction Of Arrival.

DSP — Digital Signal Processor. A specialised micro-processor optimised to perform mathematic intensive processing on large-amounts of data, at high speed.

DSSS — Direct Sequence Spread Spectrum.

DS/SS — see DSSS.

DS-SS — see DSSS.

FHSS — Frequency Hopped Spread Spectrum.

IF — Intermediate Frequency.

LPD — Low Probability of Detection.

LPI — Low Probability of Interception.

PN — Pseudonoise.

PSD — Power Spectral Density

PSK — Phase Shift Keyed.

RF — Radio Frequency.

SNR — Signal-to-noise ratio.

VHF — Very High Frequency (30 - 300 MHz).

XOR — Exclusive Or.

Chapter 1

Introduction

1.1 Objective

This dissertation describes an investigation into methods suitable for implementation on a *particular*[1] Digital Signal Processing (DSP)-based platform that are applicable to the problem of detection and direction finding of Direct Sequence Spread Spectrum (DSSS) signals. Specific objectives were to:

- Review the theory of DSSS signals
- Study radio direction finding, and the target direction finding platform
- Implement a DSSS simulator which produces outputs in the same format as the real DF platform
- Use the simulator to investigate algorithms suitable for DSSS detection and direction finding.

While it is more usual to consider signal detection separately from direction finding, the low probability of intercept (LPI) nature of DSSS means a combined approach has merit - making use of direction information as an aid to detection.

[1] The radio direction-finding system on which this research is based was made available by a local company, and is described more fully in chapter 4. For convenience, the relevant features of the system are listed here: 5 channels, analog VHF receiver mixed down to an intermediate frequency (IF), IF bandwidth of 12.8 MHz, blocksampled (80 μs capture every 1ms), DSP-based processing.

1.2 Motivation

The radio spectrum is a resource that is monitored [30] by government agencies to ensure:

1. Intentional/unintentional interference does not impede a user's access to a radio-based service.
2. Only users with paid-up licences have access to non-public portions of the spectrum (i.e. policing pirate radio stations).

In order to do this, the relevant agencies use spectrum-monitoring and radio-direction-finding equipment. To aid the operator with the monitoring task, the equipment can perform a number of functions automatically. One example of this is scanning a frequency range and marking frequencies whose power is above a certain threshold, i.e. simplistic signal detection. On a modern DSP-based platform this is carried out by applying a threshold to an estimate of the power spectral density.

Traditionally, radio signals typically found in the spectrum have been "conventional", in the sense that (from the intended receiver's point-of-view) the signal-power is strong relative to the noise-floor. That is, the signal-to-noise ratio is probably +6dB or more. These types of signals can easily be differentiated from noise by setting a threshold that is say +4dB above the noise-floor.

However, in the last 20 years, and particularly the last decade, there has been increasing interest in commercial spread-spectrum systems.

The two main characteristics of spread-spectrum signals (averaged over time) are: a) their bandwidth is much larger than is required to transmit the information they carry, and b) the average signal-power over this wider bandwidth is weak (sometimes weaker) relative to the noise power over the same bandwidth. The significance of this is that detection and direction-finding of spread-spectrum signals is more challenging.

There are two *main* kinds of spread-spectrum signals: frequency-hopped spread-spectrum (FHSS) and direct-sequence spread-spectrum (DSSS) (although chirp signals and ultra-wideband/time-hopping can also be classified as spread-spectrum, we do not consider them here).

Since the *instantaneous* power-spectrum of FHSS signals can be considered that of a conventional narrow-band signal, detection of these kinds of signals is not difficult with modern wideband, fast-sampling systems.

DSSS signals on the other hand, can have an instantaneous power-spectrum in which the signal component is below the noise-floor. Thus there is a need for techniques to aid in the detection and direction finding of DSSS signals.

1.3 Scope

The topic of this dissertation touches on a wide variety of topics, including: signal processing, antenna theory, array-based processing, statistics, radio direction-finding. In order to narrow the focus, it was decided to limit investigations to binary-phase-shift-keyed (BPSK) DSSS signals, in the VHF band (30 to 300 MHz). A maximum capture bandwidth of 10 MHz is considered, and it is assumed that any signals of interest are clear of the edge[2] of the band. While this last assumption is not realistic, in that real signals could occur anywhere in the band, a strategy to intelligently *scan* a wider bandwidth than that of the receiver available is beyond the scope of this thesis.

Since the investigation is based around a specific hardware platform, this constrains the approaches taken. For example, since the platform is based on *block-sampling* rather than *continuous-sampling*, methods which required a configurable capture time were not investigated.

1.4 Organisation of this dissertation

This dissertation ties together DSSS signals and radio direction finding, on particular DSP-based DF hardware. It begins by giving relevant background information on both of these topics in chapters 2 and 3.

Chapter 4 briefly describes the architecture and relevant implementation details of the DF platform.

Together, these three chapters formed the basis for the development of a DSSS signal simulator, described in chapter 5.

Some basic pre-processing in the form of incoherent and coherent averaging can be performed on the output data of the DF system, and this forms the content of chapter 6.

Chapter 7- describes and discusses the algorithms developed, and their evaluation.

Results are presented in chapter 8.

Finally, conclusions drawn are summarised in chapter 9, and possible future work is listed in chapter 10.

[2] A signal at the edge of the capture band could be handled by retuning the receiver so that the signal is centred in the capture band.

Chapter 2

Direct Sequence Spread Spectrum (DSSS) Signals

2.1 What is DSSS?

DSSS is a type of *spread-spectrum* technique. Various definitions exist for spread-spectrum. The most general definition would be to say a spread-spectrum signal is one for which the transmission bandwidth is much greater than the minimum bandwidth required to transmit the information.

Types of spread-spectrum signals include: chirp signals, direct-sequence spread-spectrum (DSSS), frequency-hopped spread-spectrum (FHSS), and ultra-wideband (UWB).

2.1.1 DSSS - a definition

It is difficult to define DSSS concisely. Ryan and Frater [22] put it like this:

> "A DSSS system spreads the transmitted spectrum ... by modulating the base-band signal with the digital code sequence produced by a pseudonoise code generator."

Another definition:

> "A signal structuring technique utilizing a digital code sequence (pseudo-noise sequences) having a chip rate much higher than the information signal bit rate. Each information bit of a digital signal is transmitted as a pseudo-random sequence of chips."[34]

It is generally accepted in the modern communications environment that a DSSS implementation will be wholly digital, but as pointed out by [28], analogue message modulations are possible. However, due to certain short-comings (lack of message privacy, synchronisation problems) analogue message modulations are rarely used.

Typically, M-ary PSK is used, and of these forms, BPSK is very common. In this dissertation, only BPSK-DSSS is considered.

2.2 Background

Spread-spectrum technology is widely cited as a development in response to military requirements for a radio communications which were difficult to detect and/or difficult to jam.

One of the earliest conceptions of a spread-spectrum system is that of Hedy Lamarr and George Antheil, as recorded in US Patent 2,292,387 [23, 18] filed in 1941[1].

The first example of a working spread-spectrum system is probably SIGSALY, the system used to secure communications between Churchill and Roosevelt during WWII [32, 15].

More recently, as the technology has become more widely known, and advances in modern electronics have reduced the cost of hardware, there have been a number of commercial applications of spread-spectrum: Modern cordless-phones, IEEE 802.11 ("WiFi") [12], CDMA for mobile phones (e.g. IS95, 3G-GSM-WCDMA) [1, 6], are just a few examples.

Possibly, the reason for the increase in commercial spread-spectrum systems in the last 20 years has been a combination of factors: the falling cost of receiver technology (due to advances in manufacturing), an increase in mobile applications, and declassification of the technology (very little was published on spread-spectrum in the open literature prior to 1980).

2.3 How it works (BPSK-DSSS)

Figure 2.1 shows a block diagram of a binary phase-shift-keyed DSSS system, with power spectra of the signals at various points in the system.

[1]The Lamarr-Antheil story is fascinating. As David Kahn [15] says of their invention: *"Though only a sidelight in the history of spread spectrum, because it had no direct influence on the evolution of the technology, the frequency-hopping invention did impart to that field its most glittering bit of glamour"*.

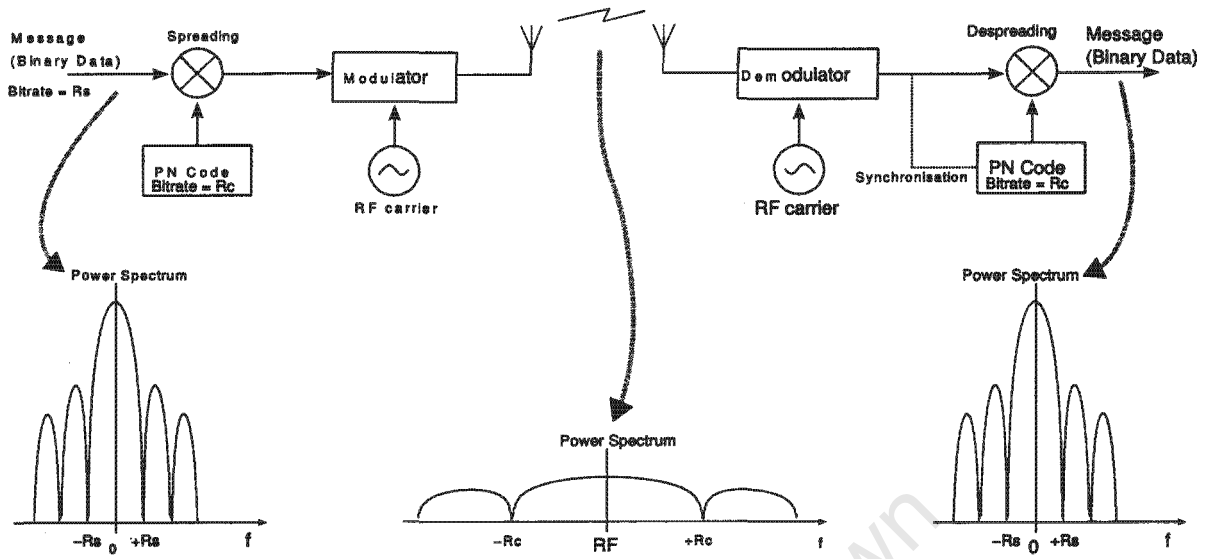


Figure 2.1: DSSS Overview - Block diagram with power spectra

Consider a narrow band BPSK signal at baseband, which is just a non-return to zero (NRZ) binary signal. Figure 2.2 shows such a signal (normalised to +1 and -1 volts), with its magnitude spectrum.

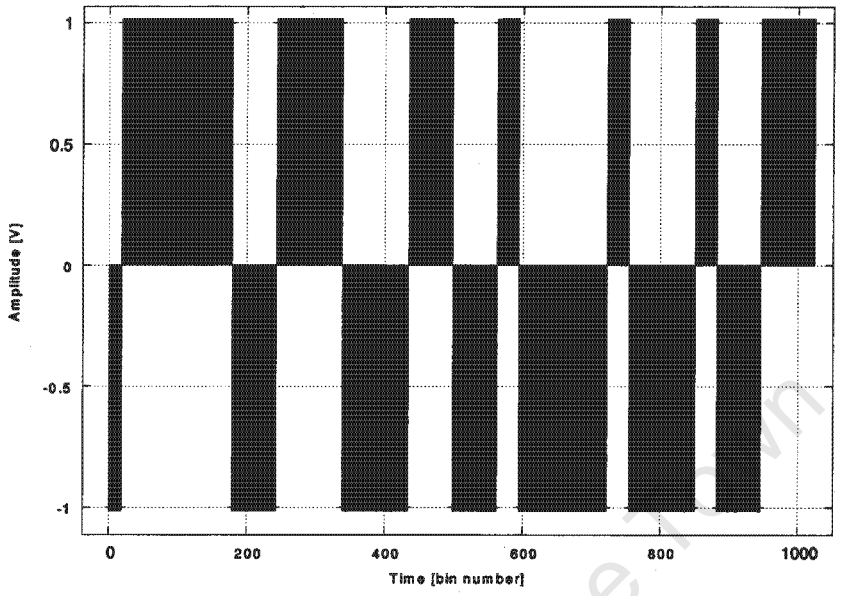
The power spectral envelope for the NRZ binary signal has a $Sinc^2$-squared form. The main-lobe of the power-spectrum has a null-to-null bandwidth of twice the bit-rate of the signal.

Thus if we want to occupy more bandwidth, we could (naively) simply increase the bit-rate. Doubling the bit-rate will give us double the bandwidth. Assuming we transmit with the same power, the total signal power doesn't change – but since we have increased the bandwidth, the power-per-Hz decreases. Doubling the bandwidth will halve the average signal power-per-Hz over a given bandwidth. If noise is assumed to be additive white Gaussian, with a flat PSD, increasing the signal bandwidth also has the effect of increasing the noise power in the signal band (noise power-per-Hz doesn't change). Doubling the signal bandwidth, means the in-band noise power will double. The net effect of this is to reduce the signal to noise ratio.

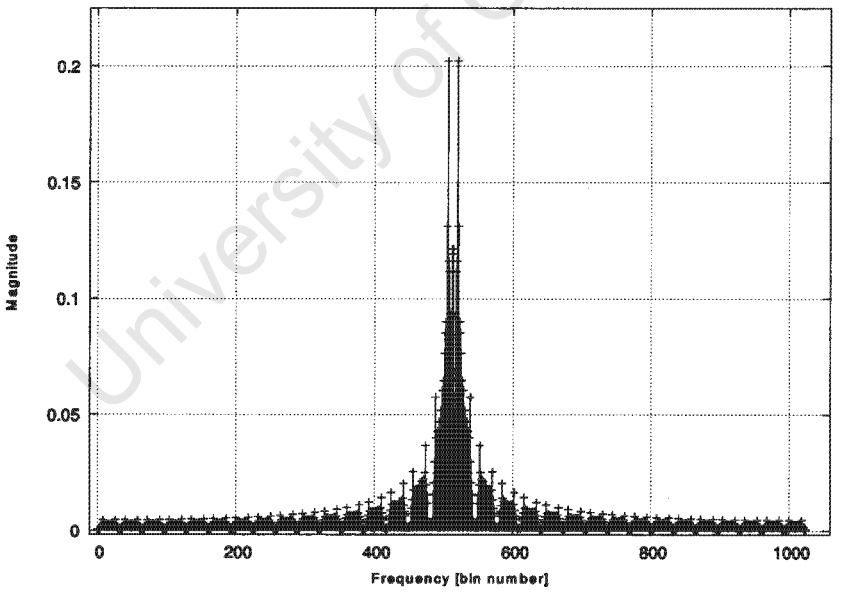
It might be thought if *low probability of detection/low probability of interception* (LPD/LPI) signals are desired, that this would be the way to do it. But there is a (somewhat obvious) problem.

The Hartley-Shannon theorem [25] states that the capacity of a band-limited channel in

[2] $Sinc(x) = sin(x)/x$. Also sometimes referred to as $Sa(x)$.



a)



b)

Figure 2.2: A narrow-band NRZ signal at baseband in a) time, and its FFT, b)

the presence of white thermal noise, given sufficient data encoding (to achieve arbitrarily low error-rates) is:

$$C = W \log_2 \left(1 + \frac{S}{N}\right) \tag{2.1}$$

where C is the channel capacity in bits per second (bps), W is the bandwidth in hertz, N is the noise power, and S is the signal power.

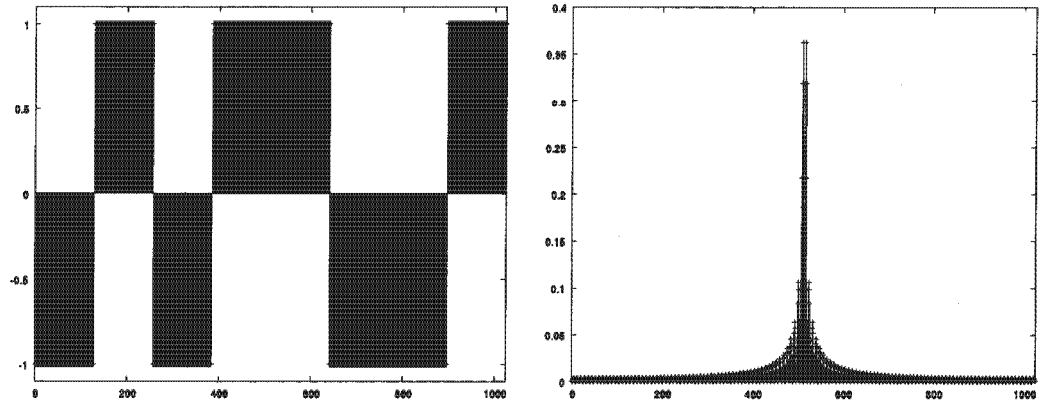
This means, given a channel operating at full capacity, at a particular bit error rate (BER), the capacity (the bit rate) of the channel can be increased without increasing the BER, provided *either* the bandwidth is increased (and S/N held constant) *or* the signal-to-noise ratio is increased (and bandwidth unchanged).

The problem with simply increasing the bit-rate, is that while the bandwidth of the channel increases, S/N falls, and thus we cannot expect to achieve the increased capacity without a corresponding increase in BER. The obvious answer to reducing errors, is to use a form of error-correction coding – which involves introducing a certain amount of redundancy. Simplistically, redundancy means sending the same data bit multiple times, and has the effect of lowering the actual data bit rate.

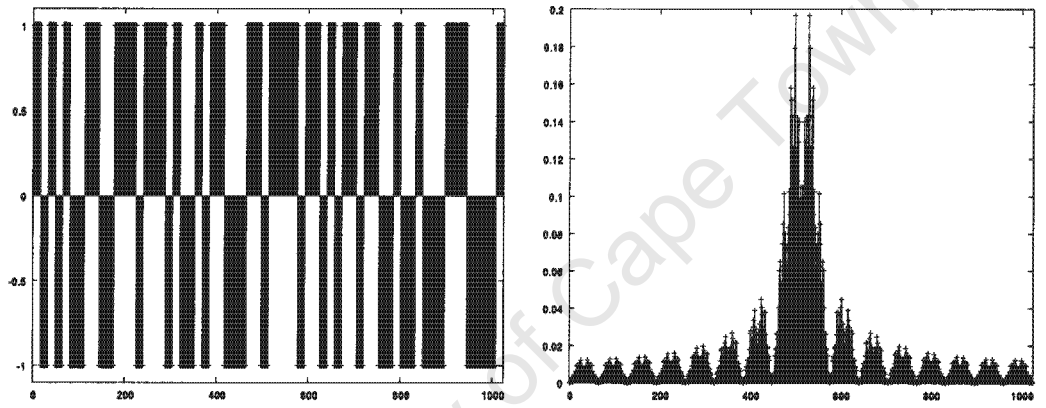
With DSSS, every data bit is "encoded" by multiplying it with a higher bit-rate code-sequence. In order to avoid confusion, the bit-rate of the code-sequence is referred to as the *chip-rate*, and the bits of the code sequence are referred to as *chips*. The higher chip-rate means a wider bandwidth, but the fact that each data bit is sent over multiple chips (redundancy) means error free communication. The ratio of the chip-rate to the data bit-rate is known as the *processing-gain*, and is also the factor by which the bandwidth of the signal is spread. If the processing gain is 1, we have one chip per bit, and the signal is not spread at all (it is however encrypted). If the processing gain is 10, we have 10 chips per bit, and the signal is spread by a factor of 10. The spreading process is illustrated in Figure 2.3.

The receiver has a copy of the same code-sequence. By multiplying the incoming signal with the code-sequence again, the signal is despread. In practice, this is made tricky by the need to first synchronise the stored code-sequence with the incoming chip-stream.

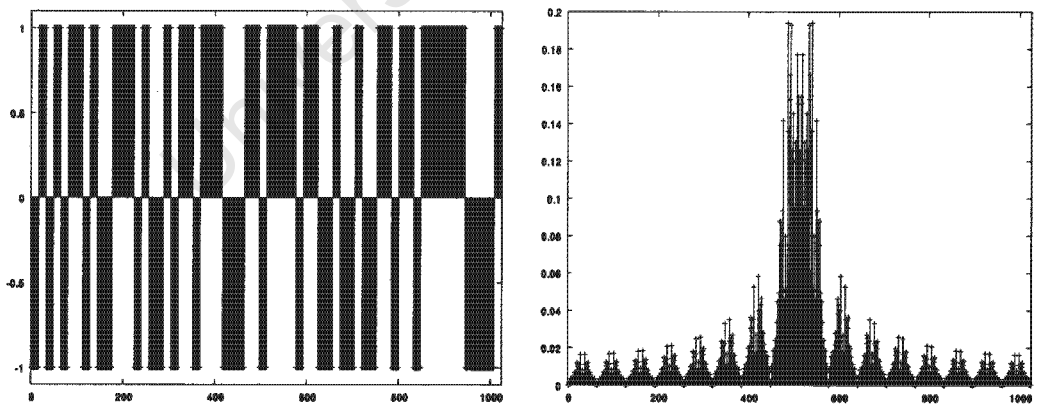
The choice of what code-sequence to use is important. For example a code that was just a sequence of ones when multiplied by the data doesn't achieve anything. Using a code sequence that alternates between +1 or -1 is better, but still not ideal, since the energy in the frequency spectrum is grouped at distinct frequencies (see Figure 2.4). We need a code with certain properties, that results in good frequency spreading. These properties are discussed in more detail in section 2.4.



data



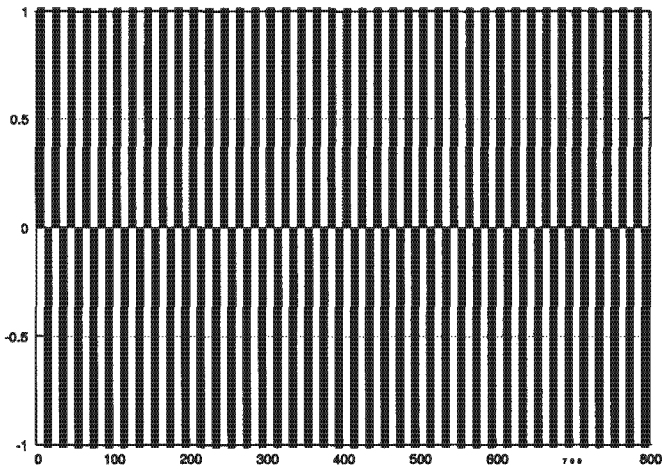
code



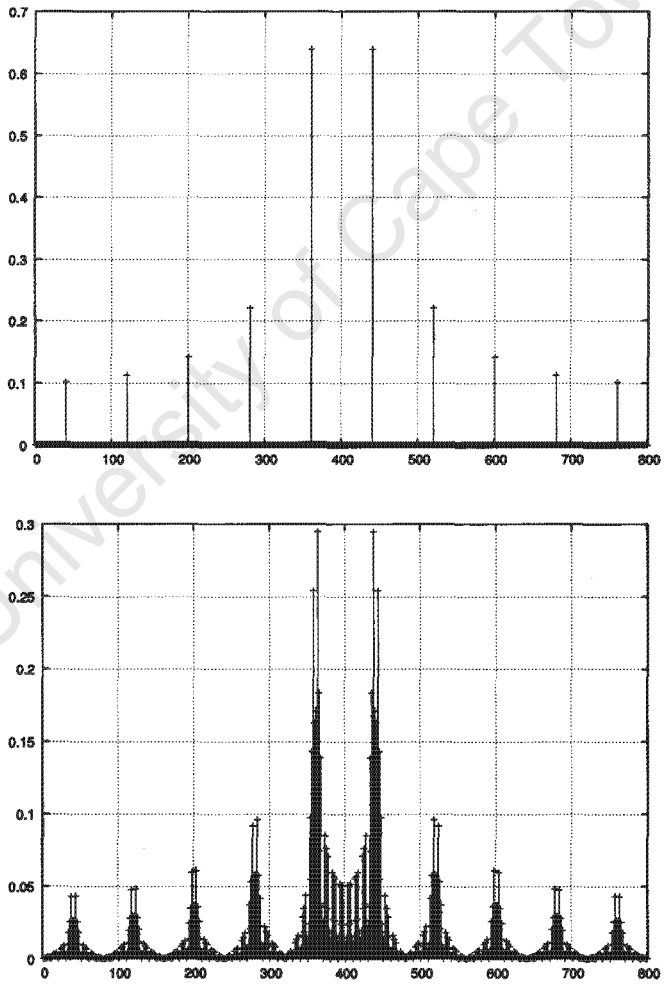
data × code

a) b)

Figure 2.3: The spreading process in a) time, b) frequency



Code of alternating +1s and -1s in time domain (above) and the resulting FFT (below)



The FFT of the result after multiplying the above code with the data of figure 2.3 (process-gain of 10)

Figure 2.4: Effect of code choice on frequency spreading

2.3.1 Additional Notes

In practice, the spreading operation is performed by the multiplication of two bi-polar sequences, or else, if the sequences are the more usual zero-one sequences, then they are combined by means of modulo-2-addition (XOR).

Some DSSS systems filter the signal to suppress side-lobes, for example IEEE802.11.

2.4 Codes

The spreading codes used for DSSS are pseudo-random, or pseudo-noise (PN) – that is they have the statistical properties of random noise, but they are deterministic. They are periodic signals known to both transmitter and receiver.

DSSS systems are classified as "short code" or "long code". Short code systems use the same PN sequence for each data bit (chip-period × number of chips in sequence = data bit-period). Long code systems have a PN sequence period much longer than the data bit period, thus adjacent bits are multiplied by a different chip sequence.

2.4.1 Required properties

Balance property

The total number of "ones" in the code differs from the total number of "zeros" by no more than 1. This property means the DC component of the code is minimal. This has significance for carrier suppression.

Run-length distribution

A run is a sequence of chips of the same value. The run-length distribution of a code is an indication of its randomness properties, which affects the ability of the code to spread the energy of a signal evenly in frequency.

For example, if a code sequence had only one run-length (the same) for both ones and zeros, it would be a simple square wave. As was illustrated above, this results in a power-spectrum with power "bunched" at discrete frequencies - spectral lines.

Meel [19] says "among the runs of ones and zeros in each [code] period it is desirable that about one-half the runs of each [value] are of length 1, about one-fourth are of length 2, one eighth are of length 3, and so on."

Auto-correlation

The auto-correlation function of a code is a measure of how well the code matches a phase-shifted replica of itself. Ideally, the auto-correlation function should have strong peaks at shifts of zero and multiples of the code period, and be minimal elsewhere. This property allows the receiver to correctly synchronise.

Cross-correlation (for multiple-access systems)

The cross-correlation of two codes is a measure of the similarity between them. This property is important for multiple-access systems, where a number of transmitters using different codes share a frequency band. Ideally codes used in a multiple-access system should be orthogonal (that is the cross-correlation is zero for all shifts of the codes relative to each other).

In practice, the number of orthogonal codes is small, and a set of non-orthogonal codes is used. The cross-correlation of these non-orthogonal codes causes performance degradation, and places an upper limit on the number of simultaneous users.

2.4.2 Types of codes

A number of different types of codes can be used for a DSSS system. The choice of which particular type to use depends on the requirements of the system. The considerations for making this choice include: single-access vs. multi-access, whether the message needs to be secure, whether the transmission need to be hidden (LPD), and the cost/complexity of the receiver. Some of the more common types of codes are:

Maximal length sequences

Also known as *m*-sequences. These are a popular choice, and as Dixon [5] says "are unexcelled for general use in communications and ranging. (Other codes can do no better than equal their performance.)" However, *m*-sequences can be predicted from even a small portion of the sequence – thus are not used for secure systems.

Barker codes

Barker codes are commonly used for pulse compression of radar signals. They are also used by IEEE802.11b for 1Mbps and 2Mbps rates [12].

Composite codes

These are code sequences generated by a combination of linear maximal sequences, and include JPL ranging codes, Gold codes and Kasami codes [19]. JPL ranging codes have special correlation properties to allow rapid synchronisation. Gold and Kasami codes are useful for CDMA, since they are large code families.

2.5 Properties

DSSS signals have the following properties:

Low Probability of Detection (LPD)

Since the energy of the signal is spread over a wide bandwidth, DSSS signals have a low power per Hz, often of the same magnitude or less than the background noise. They thus have a low probability of detection.

Secure/Private/Low Probability of Interception (LPI)

Depending on the complexity of the code employed, DSSS signals offer differing degrees of protecting a message from being intercepted by unintended receivers.

Interference Rejection

DSSS signals are unaffected by narrow band interference.

Multipath Rejection

Since the receiver is synchronised to the signal, multipath signals which have delays of a chip period or greater are rejected.

Code Division Multiplexing

By using orthogonal codes, multiple transceivers can share the same frequency band at the same time.

High Resolution Ranging

DSSS is used for position location systems, for example GPS and commercial vehicle tracking systems.

Chapter 3

Radio direction-finding

Radio direction finding refers to the techniques and systems used to find the direction to a radio source from a point.

Radio direction finding has a rich history, almost as long as radio itself. Initial experiments on radio direction finding were conducted over 100 years ago [13]. Since "radio" is implied by the context, "radio direction finding" is frequently abbreviated to direction-finding (DF).

The earliest radio DF systems used an antenna with a directional response pattern which was physically rotated. The DOA of a signal was determined by noting the direction of the antenna for which the received signal was strongest.

Modern DF systems are based on high-performance ADCs and DSPs. Most make use of arrays of omnidirectional antennas.

This chapter presents a very brief summary of radio direction finding methods, focusing on the DF methods most appropriate for this project.

3.1 Applications of Radio Direction Finding

The main application areas of radio direction finding include:

- civil: radio-monitoring, searching for interference sources, localization of non-authorized transmitters, air & marine navigation, wildlife tracking
- security & safety services: fighting organized crime, search & rescue (emergency beacon location), personnel and vehicle location

- military: communications intelligence, force strength assessments, gaining information on an opponent's order of battle (signal intelligence), friendly force location

3.2 Basic Principles

Consider Figure 3.1, which shows the spatial geometry for a DF system located at the origin.

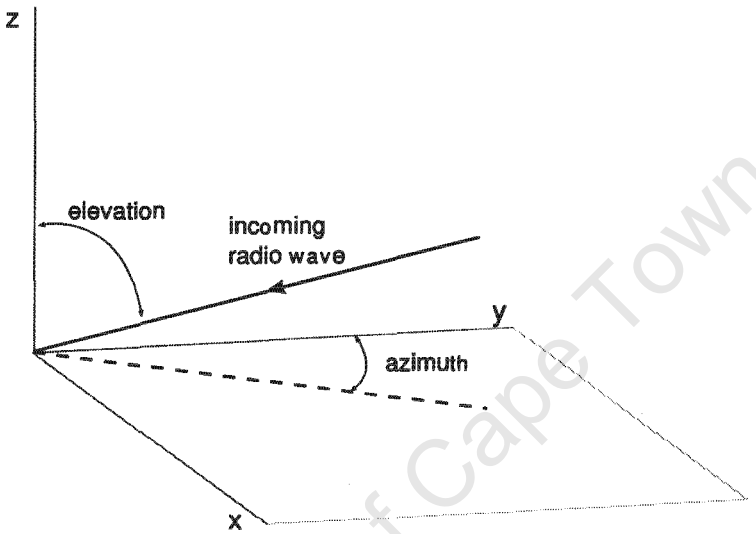


Figure 3.1: DF co-ordinate convention

While in practice the incoming radio wave is usually non-planar, most DF methods make the simplifying assumption that it is far-field planar. For this dissertation, we are only concerned with the azimuth component of the DOA, and thus will assume all incoming signal paths are parallel with the earth's surface, that is of elevation $90°$.

Figure 3.2 shows the fundamental principle of radio direction finding for 2-dimensions. Consider 2 omnidirectional antennas separated by a distance d, with an incoming (planar) radio wave at an angle of ϕ relative to the antenna baseline normal. The plane wave arrives first at antenna 1, and then at antenna 2 after a time Δt, given by

$$\Delta t = \frac{d\sin(\phi)}{c} \tag{3.1}$$

where c is the speed of light.

This delay produces a frequency dependent phase shift, $\Delta\psi$, in the signal received at antenna 2, relative to antenna 1, given by

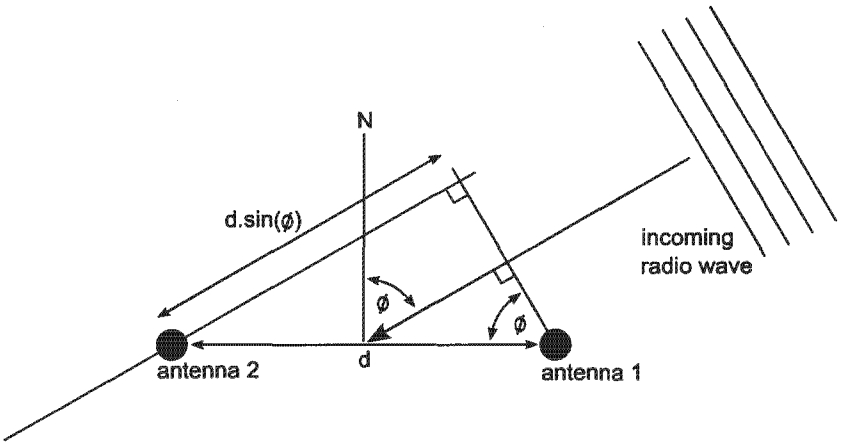


Figure 3.2: DF principle - delay

$$\Delta\psi = \frac{d\sin(\phi)}{\lambda}2\pi \tag{3.2}$$

Note that in this case, this phase-difference is actually a phase-*delay*, that is $\psi_{ant1} - \Delta\psi = \psi_{ant2}$. Substituting $\frac{c}{f}$ for λ gives

$$\Delta\psi = \frac{d\sin(\phi)}{c}2\pi f = 2\pi f\Delta t \tag{3.3}$$

re-arranging

$$\sin(\phi) = \frac{c}{2\pi fd}\Delta\psi = \frac{c}{d}\Delta t \tag{3.4}$$

Thus DOA, ϕ, can be determined by either measuring the phase-difference or the time-of-arrival (TOA) difference.

Due to the difficulty of measuring small time-differences accurately, TOA difference techniques are generally not used on small baseline systems [13], and thus will not be considered further.

3.2.1 Some notes on phase-difference methods

Figure 3.3 shows a plot of $\Delta\psi$ versus angle of arrival ϕ (equation 3.2), with $\lambda = 10\,m$ ($f = 30\,MHz$), for 3 different baselines. The different baselines of 1, 4.5 and 6 metres thus correspond to 0.1λ, 0.45λ and 0.6λ. This figure reveals some interesting points about phase-difference DF methods.

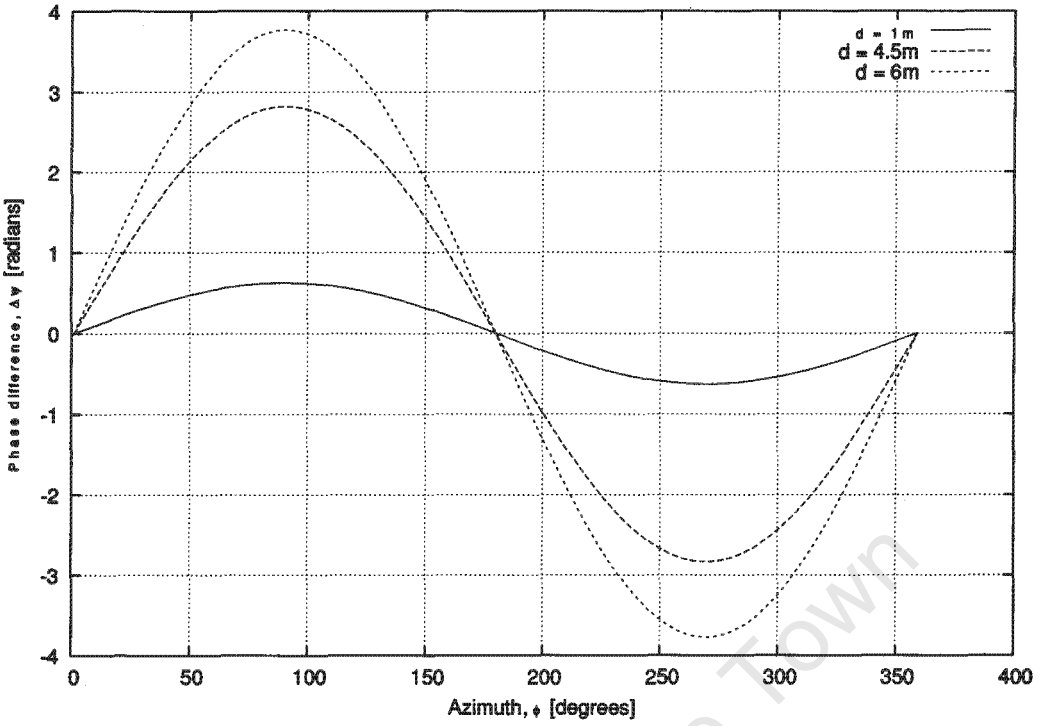


Figure 3.3: Plots of $\Delta\psi$ versus ϕ for different d, at frequency 30 MHz

The first thing to notice is that, with only 2 antennas, there is a $180°$ DOA ambiguity; We cannot tell the difference between a signal with a DOA of $\phi°$ and $(180 - \phi)°$. This is easily resolved by using an additional antenna. An additional antenna also helps improve accuracy over ranges where the rate of change of $\Delta\psi$ is small.

The second thing to notice is fairly obvious - that the maximum phase-difference increases with increasing d. This impacts on precision. A system with $d = 0.45\lambda$ is preferred to one with $d = 0.1\lambda$, since for a given DOA error, the "phase-difference resolution" requirements are more relaxed. Or in other words, given a fixed "phase-difference resolution", the first system will have smaller error on DOA measurements.

Unfortunately, there is a catch. Phase-difference measurements are restricted to a 2π range. Thus the plot for $d = 6\,m$ should really be as is shown in Figure 3.4.

So thirdly, for $d > 0.5\lambda$ an additional ambiguity is introduced. For example, a phase-difference of 3π corresponds to azimuths of approximately 53, 127, 241 and 299 degrees. Two of these are due to the previous ambiguity mentioned above.

Another way of looking at this is if we plot the magnitude of the sum of the signals on the two antennas - essentially the response of the antenna array. This is shown in Figure 3.5 for 5 different baselines. *Note that these plots show azimuth according to conventional bearing (relative to North), that is in a clockwise direction from the y-axis.*

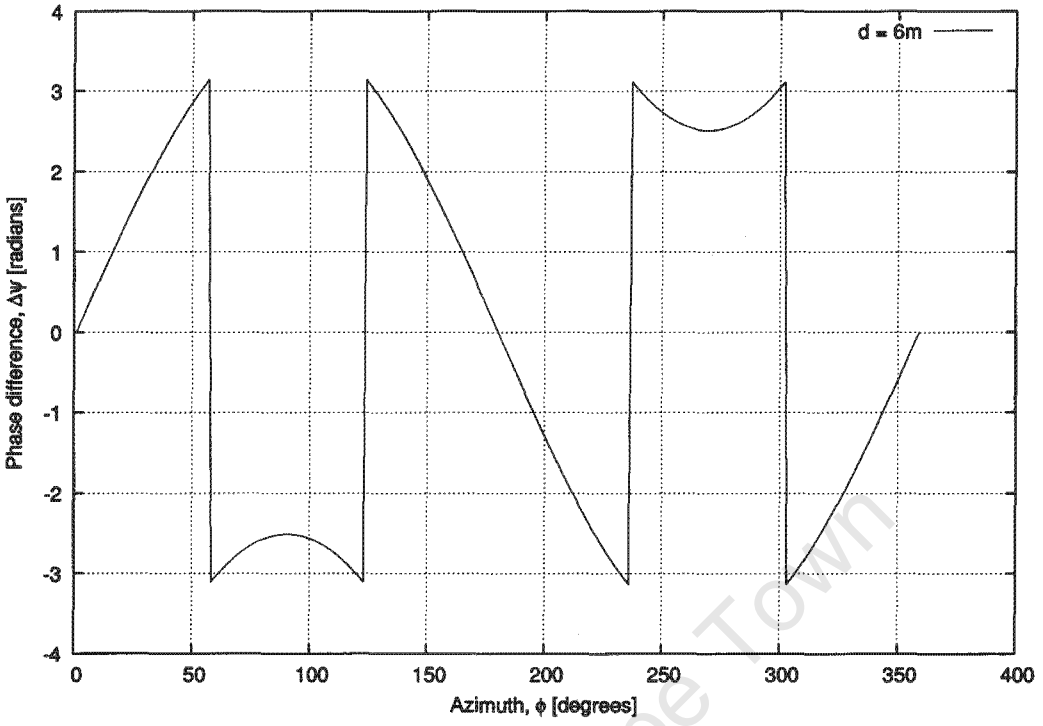


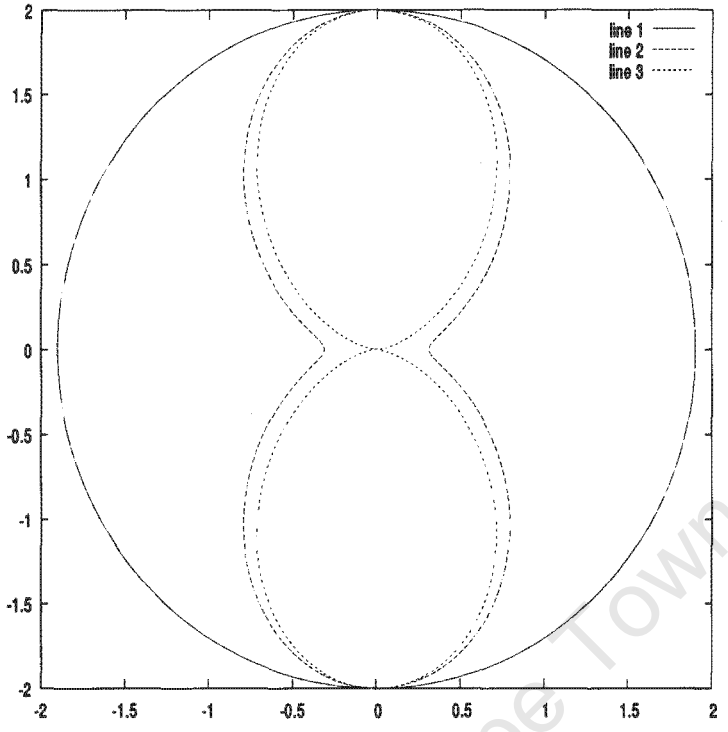
Figure 3.4: Plots of $\Delta\psi$ versus ϕ showing wrapping (f= 30 MHz, d= 6m)

Consider what happens to the response as d is increased. Initially, when the baseline is 1 m, the response has virtually the same shape as that of a single antenna, except that the gain has doubled. The response has no directionality to speak of.

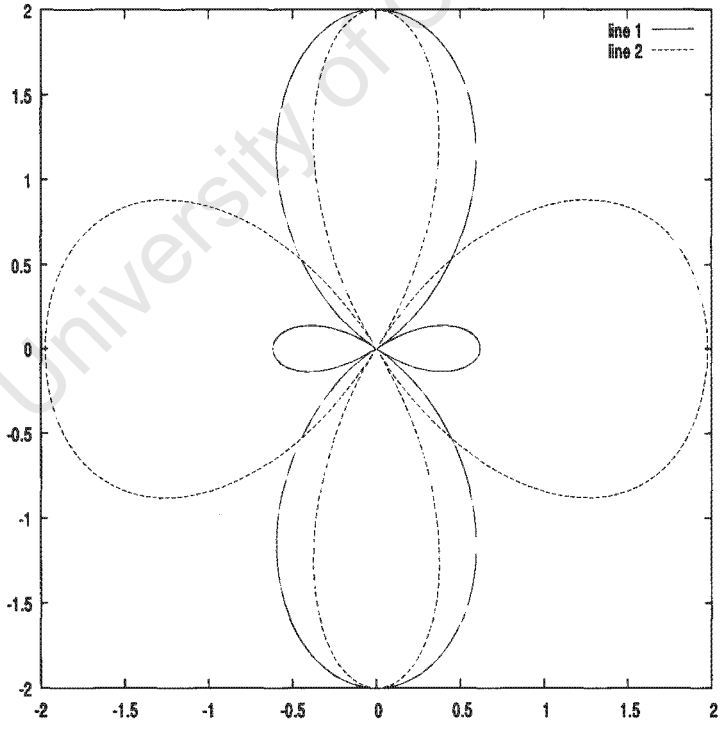
At a baseline of 4.5m, the situation is very different. The response is strong for directions perpendicular to the baseline, and weak for directions parallel to the baseline. The 180° DOA ambiguity mentioned earlier is evident.

As d is increased past 0.5λ, two additional lobes appear, which increase in size. These additional lobes correspond to the ambiguity mentioned in the third observation above. Also notice that the original lobes are narrower – this corresponds to increased angular resolution. Increasing d past λ results in the appearance of more lobes, and introduces further ambiguity.

To summarise, a DF system making use of phase-difference methods will need at least 3 antennas. At least one pair of antennas should have a baseline less than 0.5λ, and in practice this is typically further reduced to 0.4λ to compensate for measurement noise. Having additional antenna pairs with larger baselines improves angular resolution.



a) baselines: line 1 = 1m, line 2 = 4.5m, line 3 = 5m



b) baselines: line1 = 6m, line2 = 9.5m

Figure 3.5: Antenna response plots for 2-element array at 30 MHz

3.3 Different DF Techniques

Direction finding techniques can be broadly classified into 4 types:

Amplitude based

This includes systems with mechanically rotated antennas, as well as systems which use the Watson-Watt method.

Phase based

Most modern DF systems are phase based. These systems include:

- The classic interferometer, using an L-antenna array (number of elements typically varies from 3 to 9).
- Sensor array based systems. Reference [21] suggests two sub-types: i) beam-forming methods (classic beam-forming, as well as correlative interferometry) and ii) subspace methods (for example MUSIC, ESPRIT).

The DF system considered in this dissertation uses correlative interferometry, and will be considered in more detail below.

Time Difference based

These systems measure the TOA difference at spatially separated antennas. To get accurate results, the separation must usually be quite large (at least 10 m for a system with timing accuracy of $\pm 0.5\, ns$ [13]). These systems tend to be used for pulsed signals.

Doppler based

These systems make use of the fact that the received frequency at a moving antenna experiences a Doppler shift. This shift is a maximum for motion directly toward or away from the emitter, and is zero for motion tangential to the emitter (moving in a circle of fixed radius centred on the emitter). For stationary DF platforms the antenna can be physically rotated, or else a pseudo-doppler technique is used, where the antenna input to the DF system is switched rapidly between a number of fixed antennas.

3.4 A closer look at the correlative method

Consider a 5 element circular antenna (Figure 3.6).

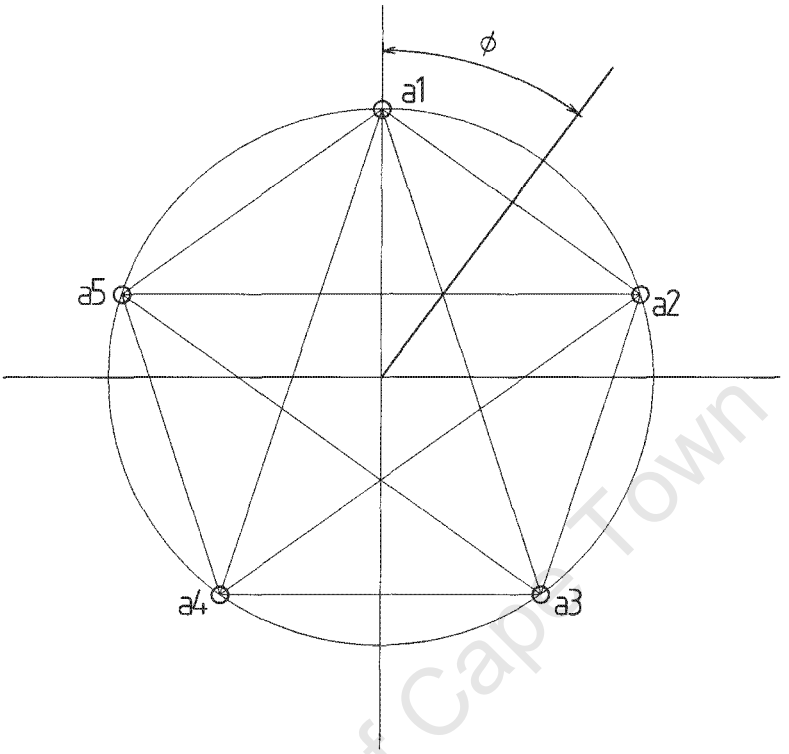


Figure 3.6: 5-element circular antenna geometry

An *aperture* can be defined as a pair of antennas, or a baseline. Since modern DF systems use the FFT to perform DF processing for multiple frequencies in parallel, in this dissertation, the term *aperture-product* will be used to refer to the product of the Fourier-transform of the signal on one antenna with the conjugate of the Fourier-transform of the signal on another antenna:

$$ap_x = V_a(\omega)V_b^*(\omega) \tag{3.5}$$

where ap_x is the *xth* aperture-product, and $V_a(\omega)$ and $V_b(\omega)$ are the respective Fourier-transforms of the signals from two antennas *a* and *b*.

The phase of the aperture-product (at a particular frequency) is the phase-difference between the two component signals (at that particular frequency).

For a 5 element array, we can define 10 apertures, as shown in the table 3.1:

Aperture	ap_1	ap_2	ap_3	ap_4	ap_5	ap_6	ap_7	ap_8	ap_9	ap_{10}
Antenna Pair	a3, a1	a4, a2	a5, a3	a1, a4	a2, a5	a2, a1	a3, a2	a4, a3	a5, a4	a1, a5

Table 3.1: Aperture list

Apertures 1 to 5 will be referred to in this dissertation as *pentagram* or *star* apertures, while apertures 6 to 10 are referred to as *pentagon* apertures.

For the pentagram apertures, the relationship between antenna spacing, d, and radius, r, is:

$$d_{pentagram} = 2r\sin(\frac{\pi}{5}) \tag{3.6}$$

And for the pentagon apertures:

$$d_{pentagon} = 2r\sin(\frac{2\pi}{5}) \tag{3.7}$$

Basically, to estimate the DOA of an incoming signal, the aperture-products for the above apertures are formed. This set of 10 aperture-products for the unknown DOA is then compared to a table of sets of aperture-products for known directions (the table size is $10 \times N$, where N is the number of known directions). The estimated DOA is obtained by finding the entry in the table that most closely matches the aperture-product set for the incoming signal. Choice of N affects angular resolution, for example with $N = 360$ (equally spaced azimuths), we can estimate DOA to nearest degree.

In practice, the comparison is performed by correlating the unknown aperture-product set with the conjugate of the table. The correlation can be thought of as a matrix product: $(1 \times 10)(10 \times N) = (1 \times N)$. Since aperture-products are complex, this means the result of a perfect match is wholly real. The column of the output correlation vector which has the maximum real part is the best match.

Both the table and the unknown aperture-product set are first normalised before performing the correlation. This means that the real part of the output correlation vector has range -1 to $+1$. A perfect match results in a value of $+1$. The value of the correlation vector associated with the DOA estimate is called the *quality factor*.

The table of aperture-products for known DOAs can be generated using theoretical values, given the antenna geometry. However, in practice, the table is usually generated by recording actual aperture-product values for an antenna with signals of different DOA. This process is known as *characterisation*. It is necessary because a) due to manufactur-

ing flaws, an antenna is unlikely to have perfect theoretical geometry, and b) because of element shadowing.

Lambert-Porter [16] mentions some problems with the correlative method when $d_{pentagon} > \lambda/2$. Since this dissertation is concerned with detection/DF as it relates to DSSS signals, and less about the specifics of particular DF techniques, it makes sense to eliminate other sources of possible uncertainty – thus this impacts on choice of frequencies for simulations and experiments.

Chapter 4

The DF Platform

This chapter briefly describes the components of the target DF platform. The limitations of the hardware constrain the possible approaches to solving the problem.

4.1 System Overview

Figure 4.1 presents an overview of the DF platform around which this research is based.

4.1.1 Antenna

The antenna most typically used with the platform, and the one considered for this thesis, is intended for use with a mobile DF system. It consists of 2 circular arrays, arranged in concentric rings. Each array has 5 equally spaced elements, as illustrated in figure 3.6. The inner array has radius of 125mm, the outer array has radius of 500mm.

4.1.2 Receiver

The receiver used is a 5-channel VHF/UHF receiver, tunable from 30MHz to 3GHz. Output is at an intermediate frequencyy (IF) centred on 12.8 MHz, with an IF bandwidth of 12.8 MHz.

4.1.3 Analogue/Digital Conversion and Digital Down-conversion

A block-sampling scheme is used to capture data. A capture of length $80\mu s$ is made on each channel simultaneously. This capture is digitally down-converted, resulting in

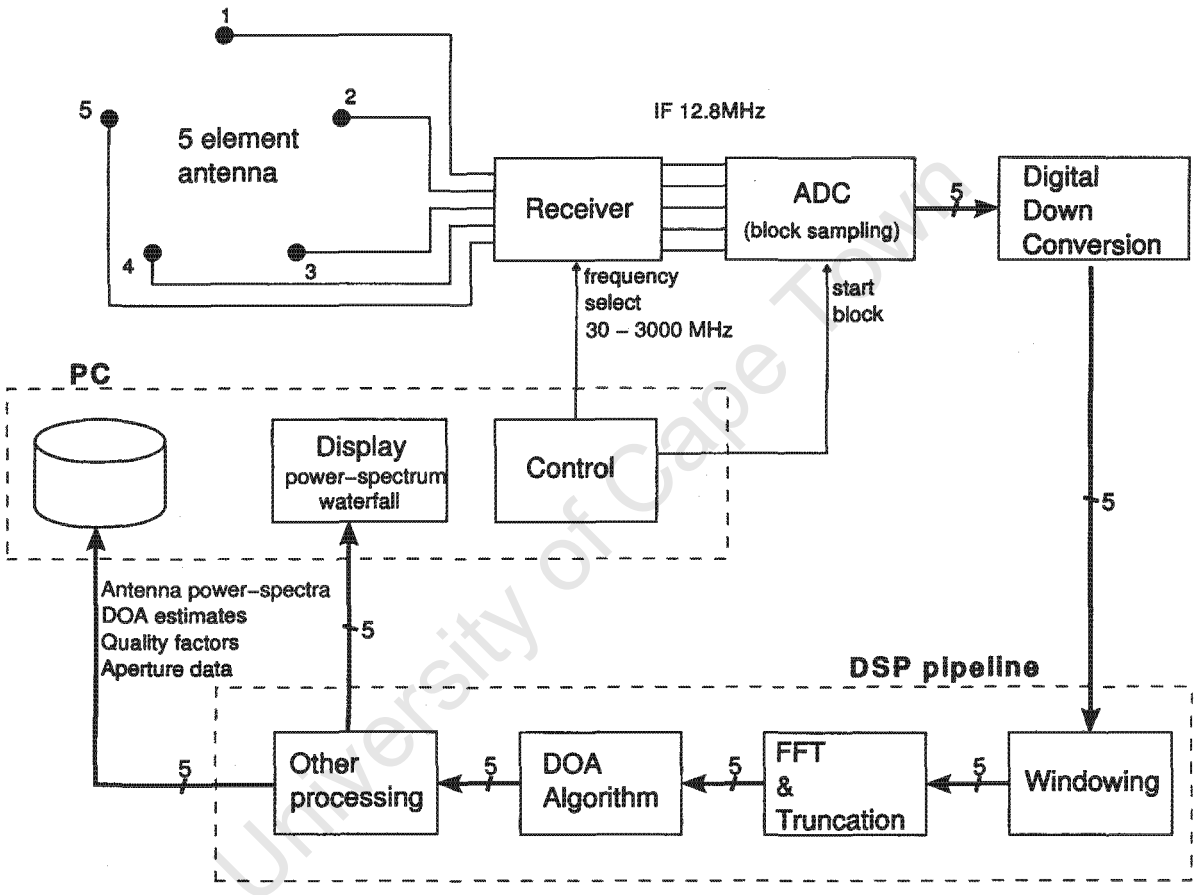


Figure 4.1: Block diagram of the DF platform

a complex base-banded signal, consisting of 1024 complex samples, representing 12.8 MHz. The period between captures is guaranteed to be within 2 milliseconds - that is the period could be slightly less than 2 milliseconds, but won't be more than 2 milliseconds.

4.1.4 DSP pipeline

Once the data is in digital format, it is fed into a processing pipeline consisting of a chain of high-performance DSPs. Each capture is windowed (typically Hanning, although other windows can be applied), the FFT computed, and then truncated to 800 points. These 800 points represent the RF signal at the receiver centre-frequency ±5MHz, now base-banded (-5 MHz to +5 MHz).

4.1.5 DOA Algorithm and other processing

Next, the 5 channels are fed into the direction-of-arrival algorithm, and a DOA (and associated *quality-factor*) computed for each frequency bin, as was discussed in section 3.4. Other processing performed includes: combining the 5 channels and computing a dB-scaled power-spectrum and some simple threshold-based signal detection.

4.1.6 Output

For each capture, 800 bins of power-spectrum, DOA, and quality-factor information are sent to the display. In addition, the following data can optionally be recorded to disk:

- The 800 bin power-spectrum for each channel (log scaled)
- DOA and quality-factor for each frequency bin
- The 10 *aperture-products* computed by the DOA algorithm

Figure 4.2 is a screen-capture from the target DF platform display, taken while monitoring a 10MHz portion of the VHF band. It shows the FFT of the current capture in the lower portion of the display, and a scrolling spectrogram of recent captures in the upper portion of the display. In the spectrogram, frequency is on the x-axis, time on the y-axis, with the most recent capture at the bottom and the least recent at the top. A conventional narrow-band signal is visible on the left-hand side of the display, while a frequency-hopping signal is visible in the centre.

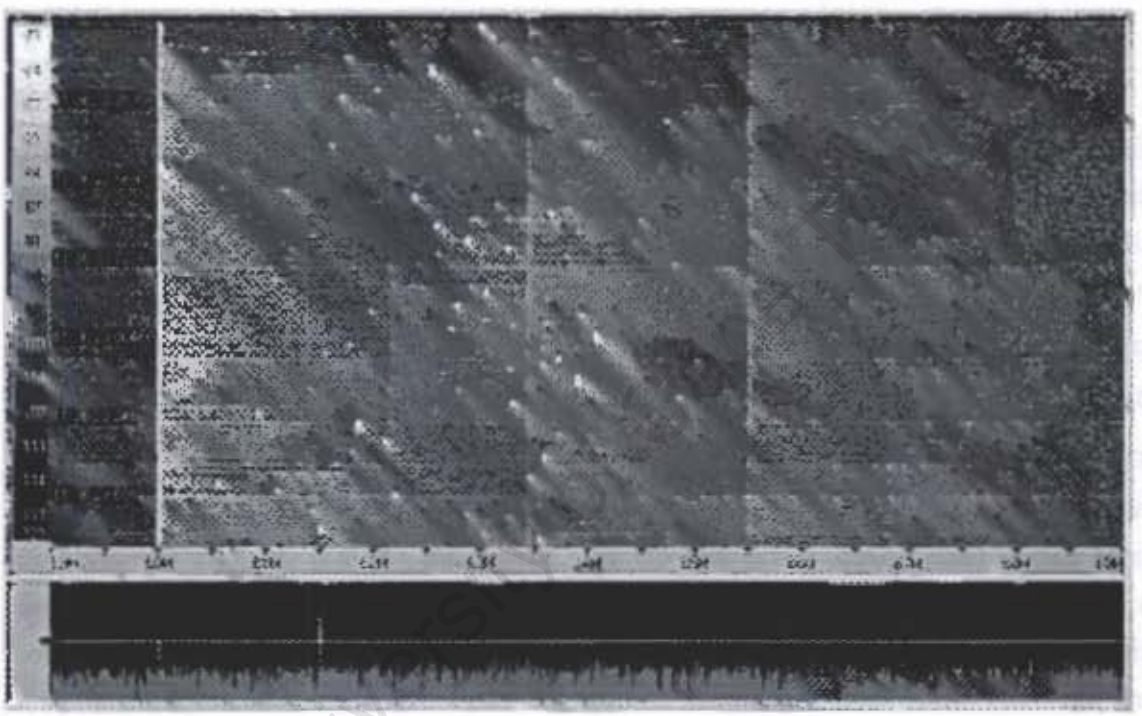


Figure 4.2: Screen-capture from DF platform display showing conventional narrow-band signal (left) and frequency-hopping signal (centre)

Chapter 5

The Simulator

Since sufficient amounts of real data with the required parameters is difficult to acquire, code was written in *Octave*[1] to simulate DSSS signals, and the capture and processing of these signals by the target DF platform. This code was also invaluable for furthering understanding of DSSS and direction finding.

Figure 5.1 presents an overview of the process of generating a single capture of simulated data.

The simulator begins by generating data representative of a typical capture on the real platform as it would appear at the outputs of the digital-down-converters. That is, simulated data is basebanded, and is initially 12.8 MHz wide. This is done by first creating a DSSS-BPSK reference signal, which is then replicated 5 times, one for each antenna channel. Each of these channels is then delayed appropriately, given the antenna geometry, the DOA of the incoming signal, and the RF frequency. It must be emphasised, that while the simulator does at no point directly synthesise data at RF, it does correctly model propagation delays at RF as they would appear at baseband.

Signals in the actual DF platform are corrupted by 2 major noise sources. One is noise in the RF spectrum, which includes sky noise and atmospheric noise (lightning). The other is uncorrelated thermal noise in each receiver channel. To simulate this, uncorrelated additive white Gaussian noise (AWGN) was added to each channel.

Next, each channel is windowed, with a *Hanning* function, to reduce sidelobe leakage due to endpoint discontinuities [17], and then fast-Fourier-transformed. The resulting frequency data is represented by five 1024 point arrays. Since the real platform outputs 800-points of frequency data per channel, the 1024 points are truncated to 800.

[1] http://www.octave.org - "GNU Octave is a high-level language ... that is mostly compatible with Matlab." Octave is free software and open-source.

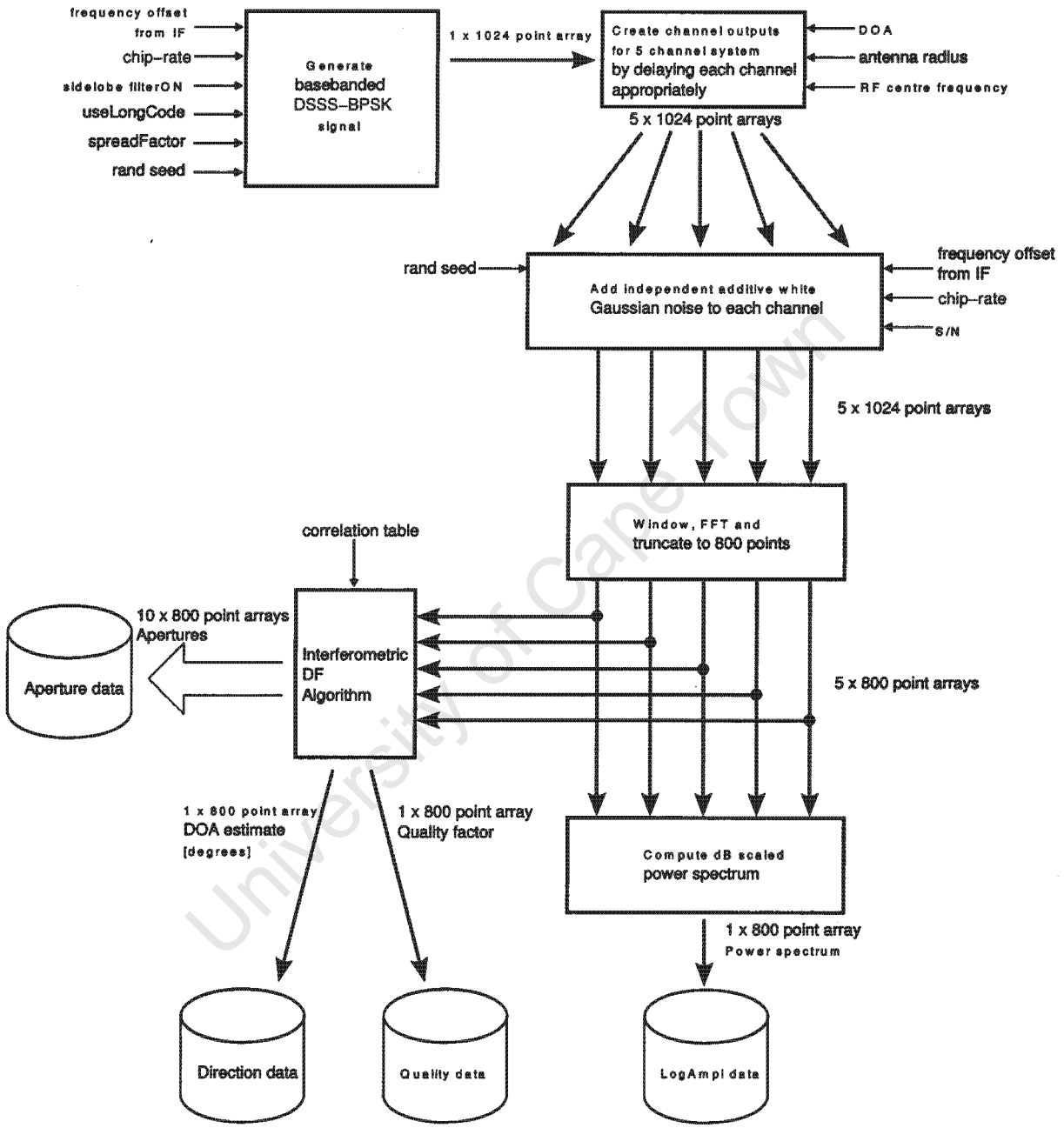


Figure 5.1: Block diagram of the simulator used to generate test data

At this point, the 5 antenna channels are fed into the correlative DF algorithm, while their power-spectra are saved to disk. The DF algorithm creates 10 aperture-products, 1 array of DOA estimates with an associated array of quality factors. These are all written to disk.

5.1 Inputs

frequency-offset-from-IF A value in Hz. This controls the position of the DSSS signal in the frequency spectrum. A value of zero results in the signal being centred (0 Hz baseband, e.g. bin 401 in 800 point power-spectrum), while a positive value shifts the signal right (bins 402 to 800).

chip-rate in chips per second. This controls the bandwidth of the signal. A value of 3 Mcps results in a signal with a null-to-null bandwidth of 6 MHz (480 bins).

sidelobe-filterON A Boolean value. If set to true will suppress the signal sidelobes.

useLongCode A Boolean value. As mentioned in chapter 2, DSSS systems can be either short-code or long-code based. Setting this true results in a long-code being used.

spreadFactor This relates the data bit-rate to the chip-rate, that is the number of chips per data bit. It only has a significant effect when used with short-codes, in which case it defines the length of the code.

rand-seed This enables the random number generator to be set to a known state, so experiments can be repeated with *exactly* the same results if necessary.

DOA in degrees of the incoming signal.

antenna-radius in metres, of the circular array.

RF-centre-frequency in Hz, that the data represents. This is necessary to compute the appropriate phase delays for the antenna signals.

S/N The signal-to-noise ratio.

5.2 Outputs

The simulator produces a series of binary output files, containing a number of 800 point arrays. Each point is represented in double-precision (8 bytes). For *each* capture, output

files contain 10 aperture-product arrays, 1 DOA array, 1 quality factor array and 5 dB-scaled power-spectrum arrays (one for each antenna channel).

Aperture data These are the aperture-products as detailed in table 3.1.

The output is complex. Figure 5.2 shows magnitude and phase of two aperture-products.

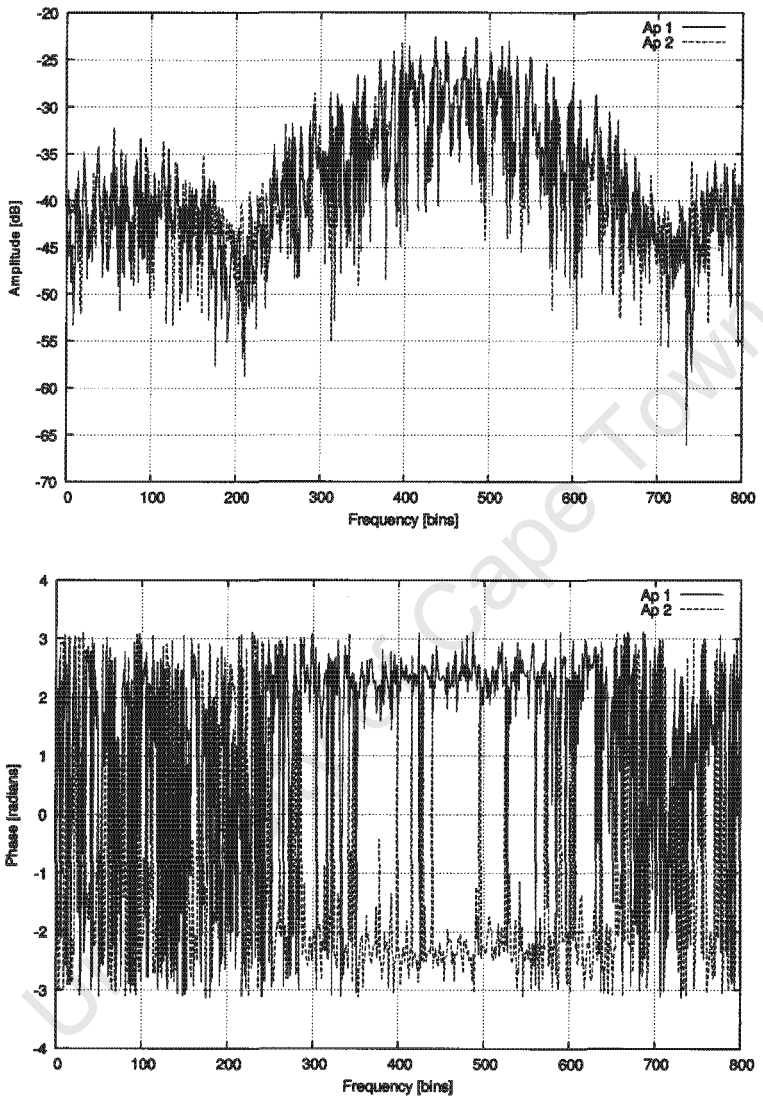


Figure 5.2: Example simulator output - 2 aperture-products, magnitude and phase (S/N=12dB)

Direction data

See Figure 5.3.

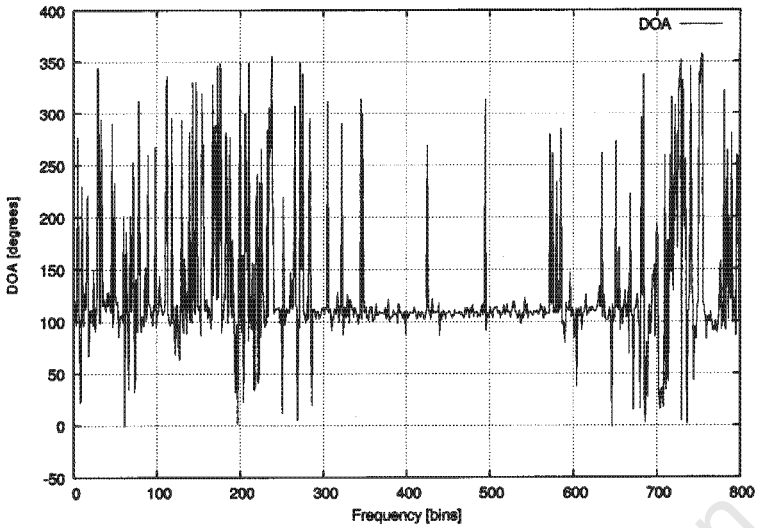


Figure 5.3: Example simulator output - direction data (S/N=12dB)

Quality data

See Figure 5.4.

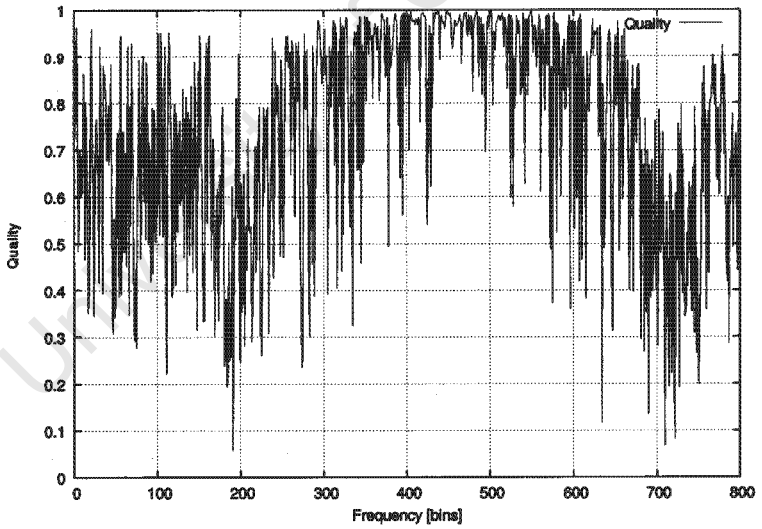


Figure 5.4: Example simulator output - quality data (S/N=12dB)

LogAmpl data

This is the dB-scaled power-spectrum of each antenna channel. Figure 5.5 shows example output for two channels.

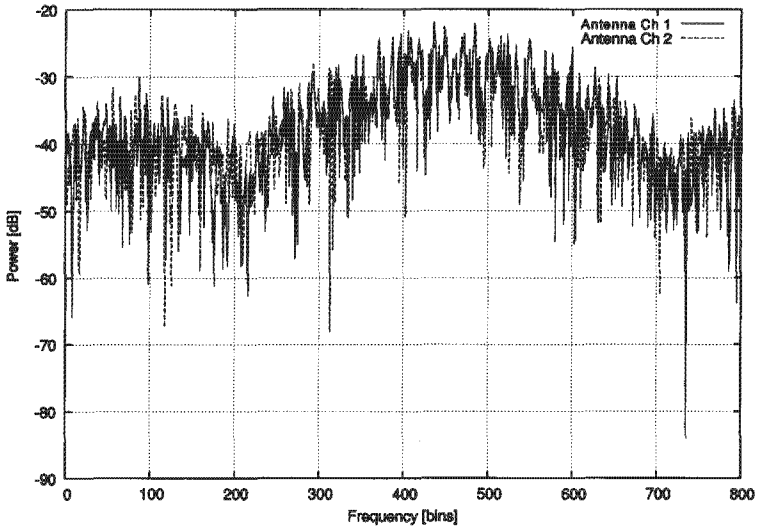


Figure 5.5: Example simulator output - LogAmpl data (S/N=12dB)

settings All input settings were saved to a text log-file for reference.

5.3 Some implementation details

5.3.1 Generating DSSS-BPSK

It was decided to use a maximal length sequence as the spreading code, as m-sequences have the desired spectral properties, and since of the code types suitable for DSSS, they are simplest to generate. Matlab code [9] to generate m-sequences was found at the Matlab file exchange[2], and adapted for use with Octave. Since calls to this function are computationally expensive, for situations where a number of captures are required, two codes are generated at the start of the simulation run and used for all captures. In the case of short-codes, repetition of the code is expected. In the case of long-codes, we would expect to capture a different portion of the code on each capture. Thus a code of length twenty times the short-code case is generated, and an appropriate length portion of this randomly selected.

The spreading code is multiplied with an arbitrary stream of data bits, whose period is *spreadFactor* times the chip period. This data stream represents the unknown message of the signal. Since in practice, the message is likely to be encrypted anyway, the actual content of the data stream is not significant.

[2]www.mathworks.com/matlabcentral/fileexchange/

5.3.2 Simulating appropriate delays

The time delays that need to be applied to the reference signal to generate the various antenna signals are very small: for a 5-element circular antenna of radius 0.5 m, the *maximum* delay between elements is ~2 nanoseconds. Since the sample period is ~78 nanoseconds, these delays cannot be created by simply shifting samples in time-domain. Instead, the reference time-domain signal is fast-Fourier-transformed, each frequency component is phase shifted by an appropriate angle, and then inverse-Fourier-transformed. The phase shift is implemented by multiplying by $e^{j\omega T_0}$, where T_0 is the time-delay, and ω is the frequency variable *at RF*.

5.3.3 Generating additive white Gaussian noise with required S/N

For the purposes of this dissertation, we treat the *signal-to-noise ratio* (S/N) of DSSS signals as the ratio of the average signal power per Hertz ($\overline{P^{Hz}_{signal}}$) *over bandwidth of the main lobe* of the signal, to the average noise power per Hertz ($\overline{P^{Hz}_{noise}}$). Strictly, $\overline{P^{Hz}_{noise}}$ should also only be over the band containing the main lobe, but in practice, it is assumed that $\overline{P^{Hz}_{noise}}$ is constant over the entire available bandwidth, and so to achieve a more accurate result, $\overline{P^{Hz}_{noise}}$ is computed using the entire available bandwidth.

The reason $\overline{P^{Hz}_{signal}}$ is only computed over the main lobe of the signal, is that some (but not all, e.g. IEEE 802.11b[3]) DSSS signals are spectrally masked so that the side lobes are significantly attenuated. The algorithms developed do not assume the presence of side-lobes.

For zero-mean functions, the *variance* of the time-domain function is the same as the mean square of the time-domain function – which is the power of the function. This means, if we know $\overline{P^{Hz}_{signal}}$, and the desired S/N, we can calculate the required noise power, and thus the variance of the noise:

$$Variance,\ \sigma^2 = P_{noise} = \frac{\overline{P^{Hz}_{signal}} \cdot B_{noise}}{10^{(dSN/10)}} \tag{5.1}$$

where dSN is the desired S/N in dB, and B_{noise} is the bandwidth of the noise.

[3] The 802.11b and 802.11g standards do not specify the width of a channel. Rather, they specify the center frequency of the channel and a spectral mask for that channel. The spectral mask for 802.11b requires that the signal be at least 30 dB down from its peak energy at ±11 MHz from the center frequency and at least 50 dB down from its peak energy at ±22 MHz from the center frequency. [35]

We multiply $\overline{P_{signal}^{Hz}}$ by B_{noise}, since we wish to compute the *total* noise power, across the entire bandwidth, such that the "per Hertz S/N" is correct in the band where the main lobe of the signal is present.

It is easy to generate noise samples of the required variance in *Octave* by using the *randn* function, and scaling by σ, the standard-deviation.

In practice, both signal and noise are windowed in the time domain, which means their absolute values for power are not the same as used/calculated in 5.1. Since it is likely that both are attenuated by the same ratio, this should not affect S/N greatly. This was verified experimentally: signal and noise were generated at a required S/N, and this compared with the actual S/N after windowing. This was repeated 10000 times. The standard-deviation was found to be 0.27 dB, and worst case difference was 1 dB.

Chapter 6

Incoherent and Coherent Averaging

It is well known that uncertainty on a measurement (whether due to measurement error or the presence of a random disturbance) can be reduced by averaging a number of repeated measurements.

In the context of signal-processing (where we are dealing with sampled time-domain and frequency-domain representations of signals), it is important to distinguish between what the literature refers to as *incoherent averaging* and *coherent averaging*.

6.1 Incoherent averaging

As suggested by Lyons[17], this is better referred to as *averaging data that is obtained incoherently*. This could be extended to refer to averaging data in such a way that coherency in the data is not made use of.

Coherency means that in a set of block-captures of a particular sinusoidal signal, the phase of the signal is the same at the start of each capture.

A good example of incoherent averaging (in the context of the platform used for this project) would be averaging the power-spectrum of a number of captures for a single channel. The raw channel data is 800-points per capture in the time-domain, representing 10 MHz of bandwidth. While the DF platform specifies captures are each 80 microseconds long, it doesn't guarantee a specific period between captures (see section 4.1.3). For this reason, this data can be regarded as *incoherent*. To elaborate further, consider a sinusoidal signal of frequency 12.8MHz. The period of this signal is 78.125 nanoseconds. In order to coherently average $80\mu s$ block-samples of this signal, it would be necessary to specify the start time of each block to a resolution of 217 picoseconds in order for the phase of the start of each block to be within 1 degree. This kind of precise timing is not

possible with the available hardware. Since the channel data is incoherent, we can't simply take the mean of a series of captures - as they will not necessarily add constructively. But we can take the mean of the power-spectra of a series of captures.

In Figure 6.1, two plots are shown. The first shows the power-spectrum for a single capture of a channel. The second shows the result of averaging the power-spectra of a series of 10 captures. It can be seen that this kind of averaging - incoherent averaging - reduces the variance of the noise, enabling a better estimate of the noise-floor and the average signal power. There is no improvement in signal-to-noise ratio.

Other applicable examples would be averaging the phase-angle or magnitude of an aperture-product. These are both examples of averaging where the coherency in the data is not utilised.

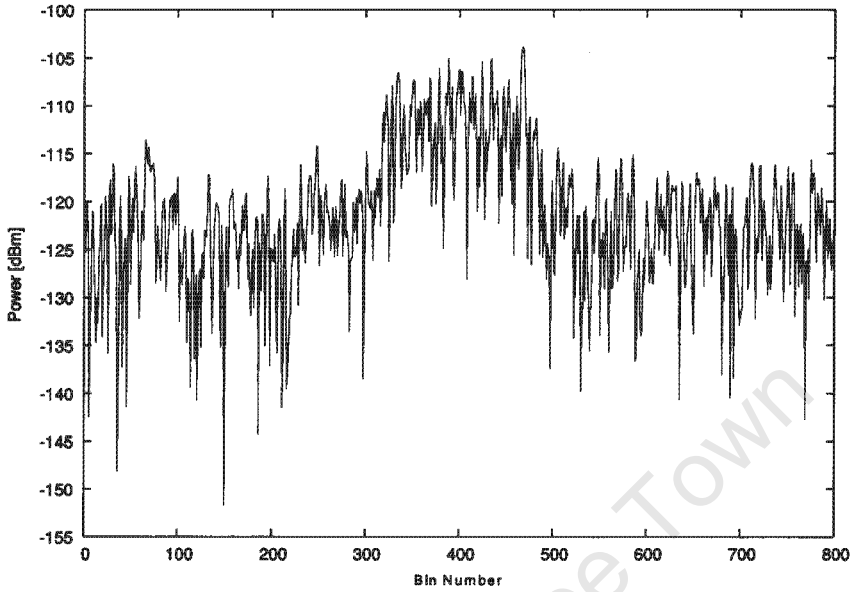
6.2 Coherent averaging

Consider three vectors, call them *Resultant 1, Resultant 2* and *Resultant 3,* being made up of three signal vectors, *Signal 1, Signal 2* and *Signal 3*, each with a small amount of additive noise, *Noise 1, Noise 2* and *Noise 3* respectively.

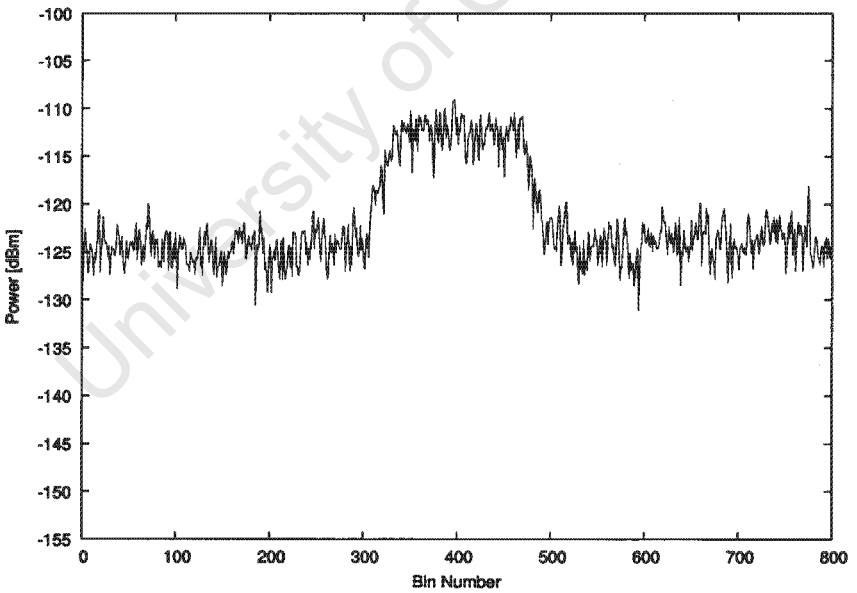
If these signals were captured *incoherently,* then the phase-differences of the signal vectors are arbitrary with respect to each other, and we have the situation in Figure 6.2. Taking the mean in this situation is unhelpful - the mean of the resultants is worse than any of the resultants on their own. This is the situation with the system's raw channel data.

If, however, these signals were captured *coherently*, then the phase-differences of the signal vectors are zero with respect to each other. We have the situation in Figure 6.3, where the mean of the three resultants is much closer to the true signal vector than any of the original resultants. The signal component of the mean has not changed, but the noise components, being uncorrelated, have "cancelled out" to a certain degree, thus there is a gain in signal-to-noise ratio. It is this kind of averaging that is meant by the term *coherent averaging,* sometimes also referred to as *integration.*

Although the raw channel data is incoherent, *aperture-product* data can be regarded as *coherent.* This is because an aperture-product is a measure of the *phase-difference* of a signal between two channels, and this is dependent on the frequency and direction-of-arrival of the signal. Provided the DOA remains constant over the captures, aperture-products derived from multiple captures can be averaged in their complex (vector) form. This results in an improved signal-to-noise ratio, as can be seen in Figure 6.4.



a)



b)

Comparison of power-spectra for a channel for a) single capture, b) mean of 10 captures of a real DSSS signal.

Figure 6.1: Effect of averaging channel captures (incoherent)

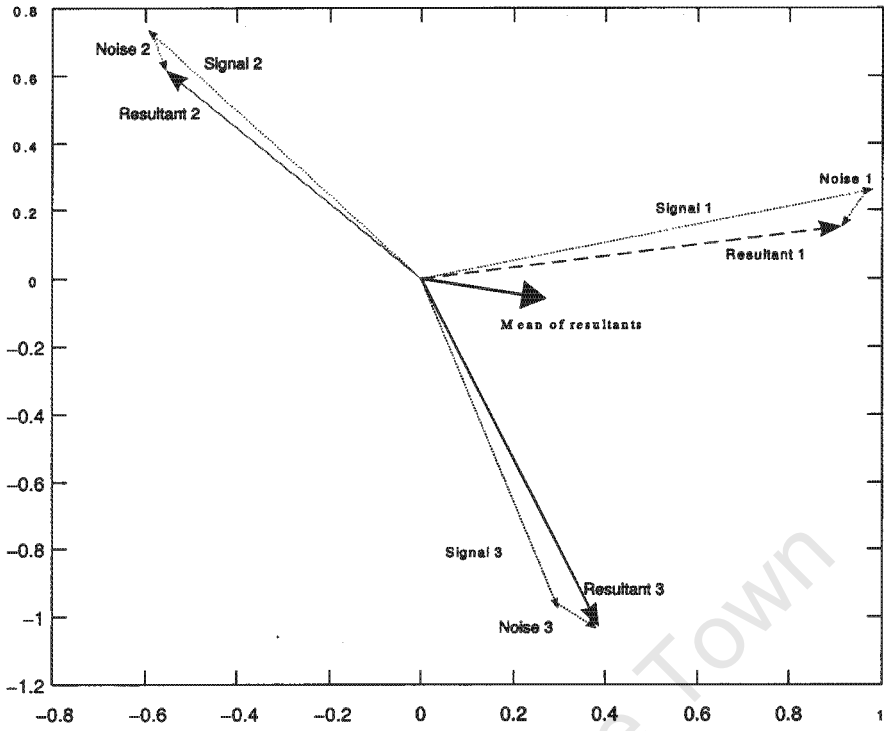


Figure 6.2: Averaging three vectors from incoherent captures

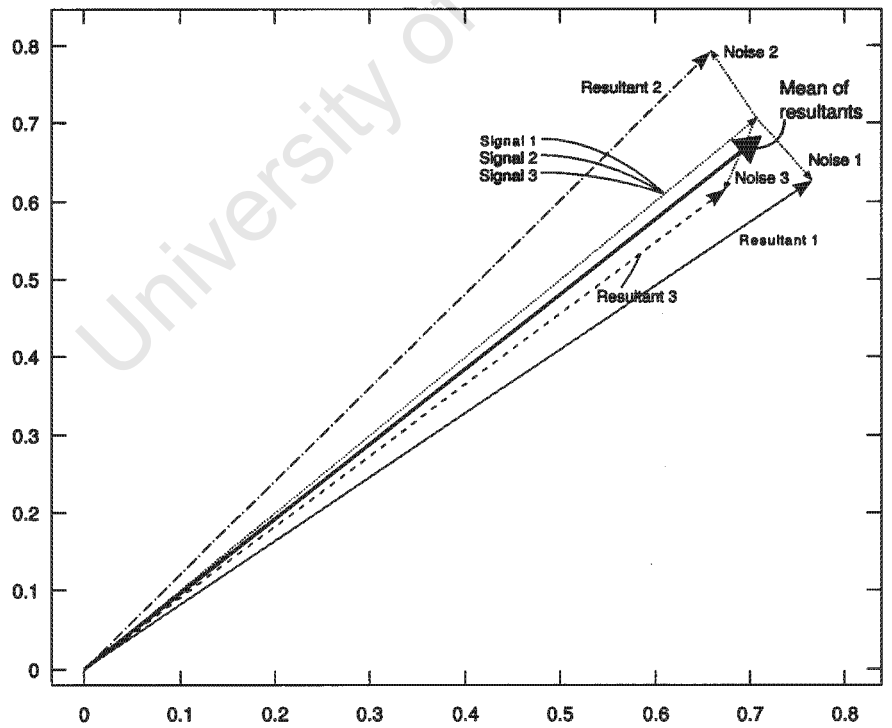
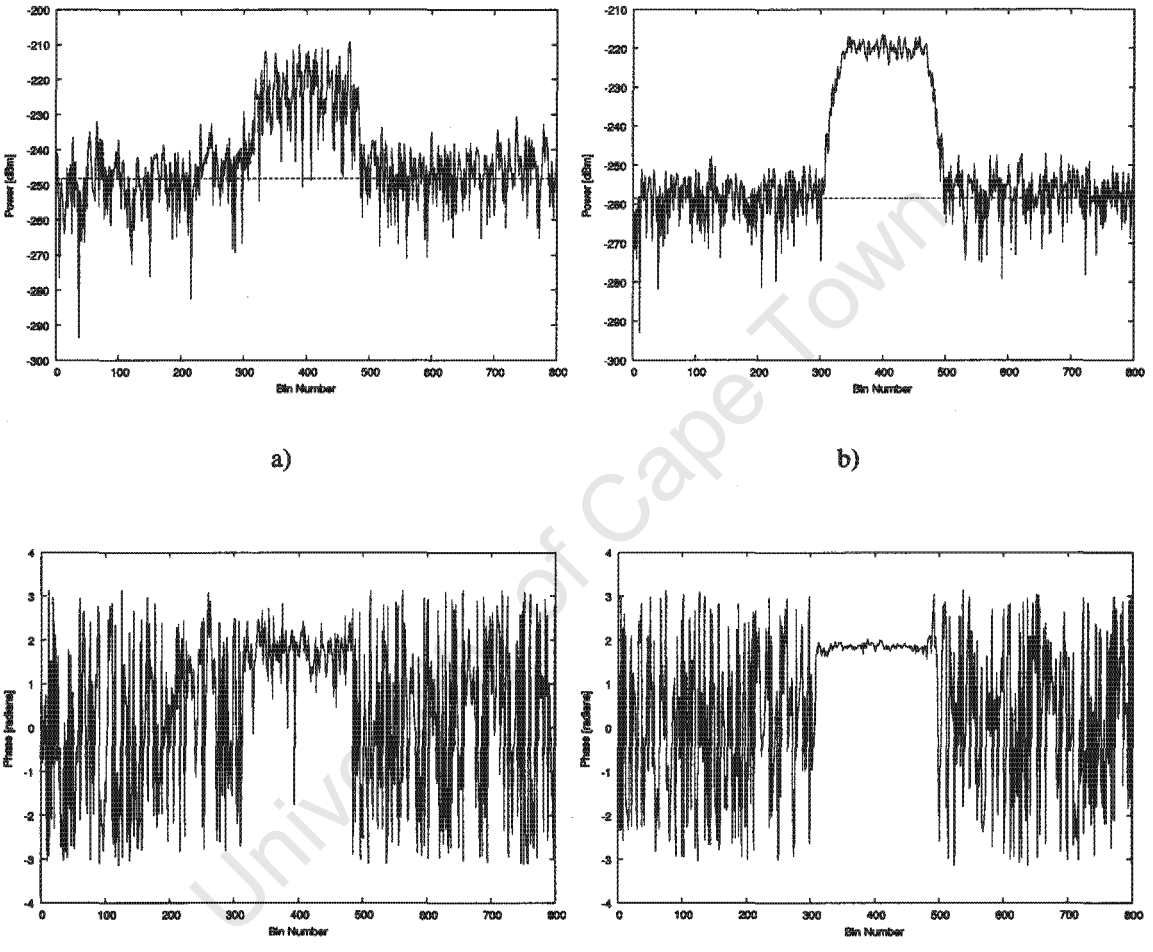


Figure 6.3: Averaging three vectors from coherent captures



a) b)

c) d)

Comparison of power-spectra and phase of an aperture-product for a) & c) single capture, b) & d) mean of 10 captures of a real DSSS signal. [Note: the horizontal line in a) and b) represents the mean of the noise.]

Figure 6.4: Effect of averaging aperture captures (coherent)

Chapter 7

Approach to Detecting DSSS

It was decided to approach the detection of DSSS by making use of two sources of information: firstly, the shape of the DSSS signal power spectral envelope, which has a characteristic *Sinc*-squared form (section 2.3) and secondly, by making use of the direction information generated by the DF platform. Direction data generated for frequency bins where no signals were present would be expected to have a uniform distribution - no DOA would dominate over any other. But direction data generated for a band where a signal was present (even a weak one) would be expected to be "coloured" by this signal. The frequency bins occupied by this signal would have a non-uniform DOA probability distribution. The direction with the highest probability would be the DOA of the signal.

Five different algorithms were considered:

1. A least-squares fit of a top-hat model to power-spectrum data from a single channel.
2. A least-squares fit of a sinc-squared-based model to power-spectrum data from a single channel.
3. A least-squares fit of a top-hat model to the mean of the magnitudes of all ten aperture-products (ap_1 to ap_{10}) data, from 5 channels.
4. A least-squares fit of a top-hat model to pre-processed direction data.
5. A combination of algorithms 3 and 4.

7.1 Test harness for evaluating algorithms

An overview of the experimental method for evaluating the algorithms mentioned in this chapter is shown in Figure 7.1.

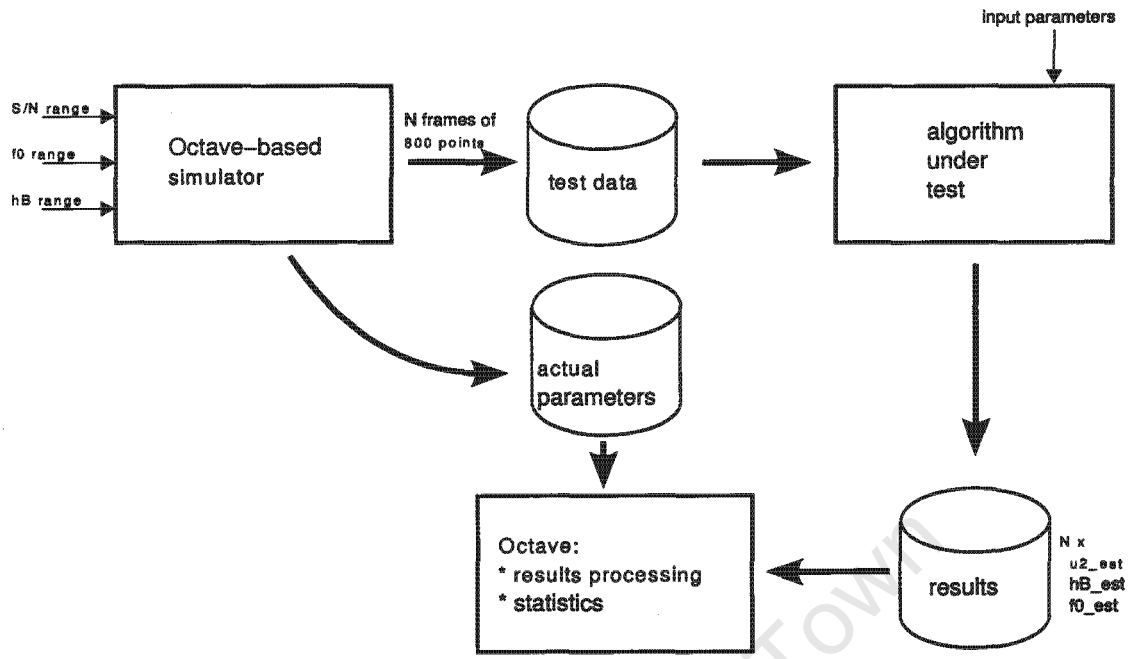


Figure 7.1: Block diagram of experimental method for least-squares

A Monte-Carlo approach was used where a large number of frames of data were generated, and the algorithm under consideration run on each of them.

The simulator discussed in chapter 5 was used to generate test data. Over the course of a Monte-Carlo run, the simulator input parameters varied were S/N ratios, signal frequency-offsets-from-IF, $f0$, and different half-bandwidths, hB. The other simulator input parameters (section 5.1) were held constant. Twice hB gives the null-to-null bandwidth of the DSSS signal. The input parameters $f0$ and hB are specified in bins - these were converted to appropriate values in Hz and chips per second (cps) before being passed to the simulator module.

Octave was used to compare the algorithm output against the actual input parameters, and generate statistics.

7.2 Using a single channel

Since variations of the target DF platform are "detection-only" systems, that is they only have a single antenna, it was felt there was merit in looking at algorithms which operated purely on the power-spectrum. These algorithms could then be used as a starting point for incorporating DOA information.

The general approach taken was to attempt to fit a model to the data using least-squares

[36]. Two models were considered: a) a tophat function and b) a dB-scaled sinc-squared function. These are illustrated in Figures 7.2 and 7.3.

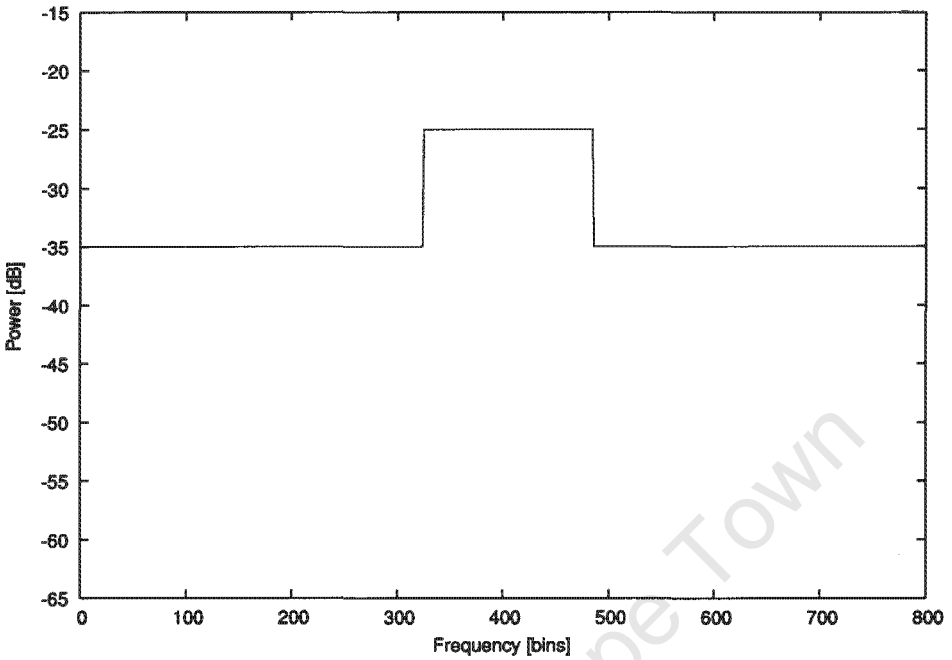


Figure 7.2: Tophat model used for least-squares fitting to data

These two models can be defined as follows:

Let n be the frequency bin index in the range 1 to N. Then the tophat model, M_{tophat}, is given by

$$M_{tophat}(n) = \begin{cases} u1, 1 \leq n < f_0 - hB \\ u2, f_0 - hB \leq n < f_0 + hB \\ u1, f_0 + hB \leq n < N \end{cases} \tag{7.1}$$

Let

$$x = n - 2 - N - f0 \tag{7.2}$$

and

$$\kappa = 10^{\left(\frac{u1-u2}{10}\right)} \tag{7.3}$$

then the dB scaled Sinc-squared model, M_{sinc}, is given by

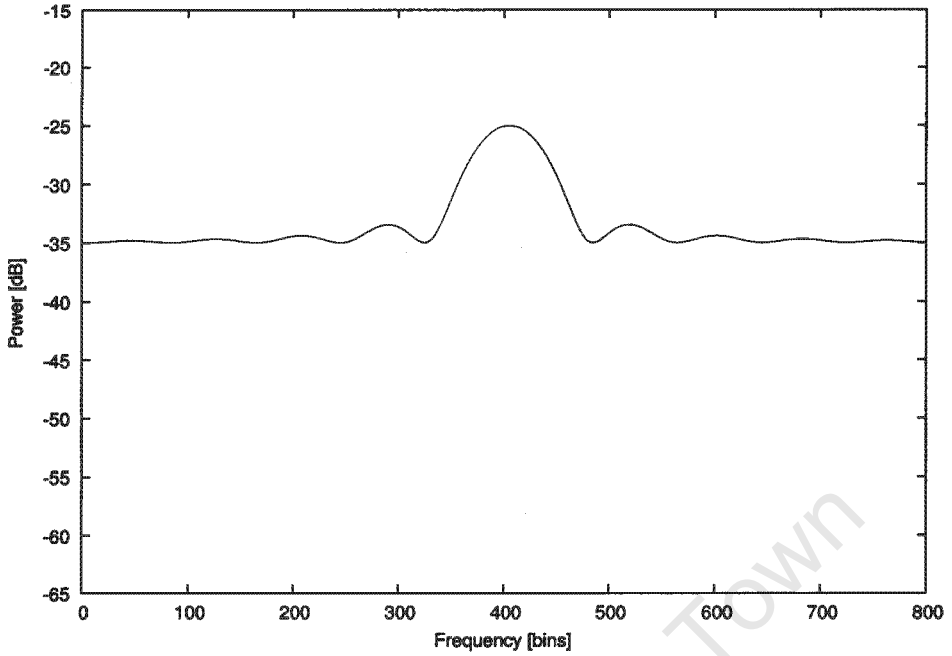


Figure 7.3: dB scaled Sinc-squared model used for least-squares fitting to data

$$M_{sinc} = u2 + 10\log_{10}\left((1-\kappa)\left(\frac{\sin\left(\frac{\pi}{x}\right)}{\left(\frac{\pi}{x}\right)}\right)^2 + \kappa\right)$$

where

u1 - the lower power value, the noise-floor (i.e. -35 in Figures 7.2 and 7.3)

u2 - the upper power value, the signal + noise peak (i.e. -25 in Figures 7.2 and 7.3)

hB - the half-bandwidth (i.e. 80 in Figures 7.2 and 7.3)

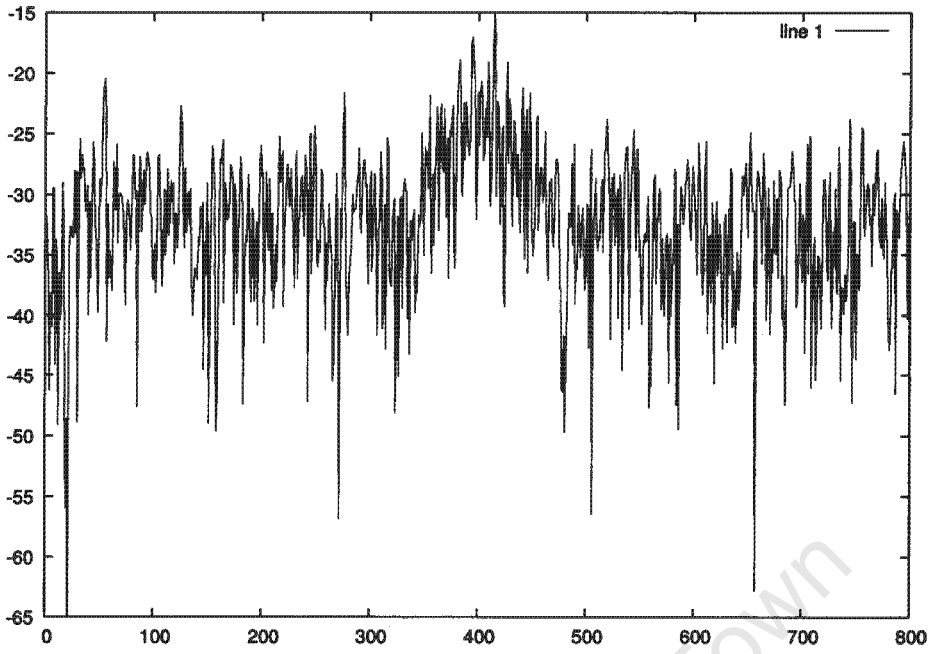
f0 - the centre-frequency (i.e. 405 in Figures 7.2 and 7.3)

For both these functions, the left and right "edges" (that is $f0 - hB$ and $f0 + hB$) are restricted to fall within the range 1 to N.

Figure 7.4 shows a single-capture, a "snapshot", of the power-spectrum of a simulated DSSS signal for a single channel. This is the type of data the two models will be fitted to.

7.2.1 Least-squares implementation

The least-squares approach aims to find the model parameters that minimise:



Signal centre is bin 405, bandwidth 160 bins, S/N is 4dB

Figure 7.4: Power-spectrum of a simulated DSSS signal

$$C(\theta, n) = \sum_{n=1}^{N} (d(n) - M(\theta, n))^2 \tag{7.4}$$

where C is the *cost*, M is the model (in this case either tophat or dB-scaled-Sinc-squared), n is the bin-index, d represents the actual data and θ represents the model parameters *u1*, *u2*, *hB* and *f0*.

To reduce the search space, parameter *u1* was fixed at the mean of the input data. It should be emphasised here that the algorithm does not transform input data to/from the log/linear domain. If the input data is log-scaled power-spectrum data, then the mean and model-fitting calculations are on a log scale. While taking the mean in the log domain is less accurate than in the linear domain, it is considered to be acceptable, since only a rough approximation of the noise floor is required.

Thus the *cost* in equation 7.4 can be considered a function of 3 variables: *u2*, *hB* and *f0*. We wish to find the global minimum of this function.

Borse [3] suggests some methods for rapidly finding minima of multivariable functions, however most of these methods cannot *guarantee* that the minimum returned is the global minimum - some of them get "stuck" on local minima. While speed of execution of the algorithm is an important issue for possible future practical implementation, at this stage

the focus is more on the accuracy of the results. For this reason, it was decided to use a "brute-force" iterative search to locate the global minimum.

The algorithm was implemented in C++, and used a 2-step process of 3 nested for-loops to iterate through the search-space. Step 1 was a coarse search, with larger increments/steps between subsequent variable value. Step 2 was a fine search conducted over a smaller area centred on the minimum returned by step 1.

7.2.2 Human experts – experimental method

To get a qualitative "feel" for how well the least-squares algorithm performs, the results were compared to those of human experts.

Candidates were presented with print-outs of twenty power-spectra (see Appendix A) and an answer sheet. They were asked to a) state whether a DSSS signal was present in the spectrum, and b) if so, identify the centre-bin and bandwidth (in bins) of the main lobe of the signal.

Six power-spectra represented signals with a signal-to-noise ratio of 0 dB, another six power-spectra represented signals with a signal-to-noise ratio of 2 dB, and 5 power-spectra represented signals with a signal-to-noise ratio of 4 dB. Three power-spectra were pure noise and did not contain a signal. The DSSS signals were generated with random centre-bin, and with bit-rates picked from 1, 1.5, 2, 2.5 and 3 Mbps. These bit-rates correspond to main-lobe null-to-null bandwidths of 160, 240, 320, 400 and 480 bins.

Precautions:

- The power-spectra presented to the candidates were arranged randomly – so candidates would not observe a sequential trend based on any parameter such as SNR or bandwidth.
- All power-spectra had the same scale/range.
- Power-spectra were presented one at a time.

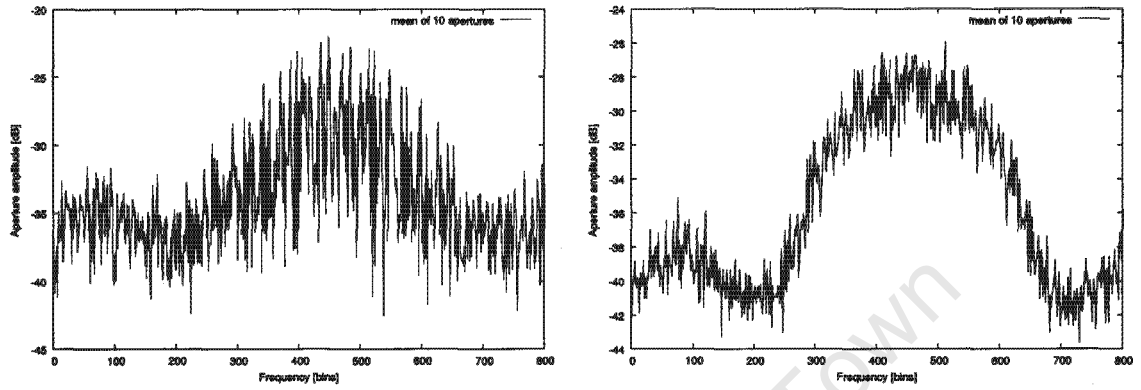
7.3 Using multiple channels

Since a DF system has multiple antennas, we can use coherent averaging, as described in section 6.2 to improve the S/N ratio. Averaging the aperture-products from 10 sequential

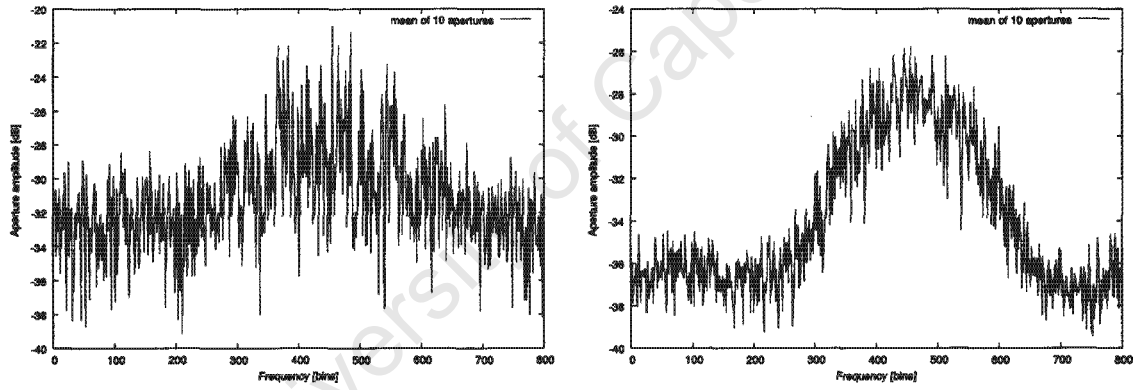
captures gives an improvement of ~10dB, and is a good trade-off of S/N improvement versus the increased capture time.

Figure 7.5 shows aperture-product amplitudes for different signal-to-noise ratios, and the effect of coherent averaging.

S/N = 4dB



S/N = 0dB



Single capture | Coherent average of 10 captures

All plots are of mean of amplitudes of the ten apertures, ap_1 .. ap_{10}

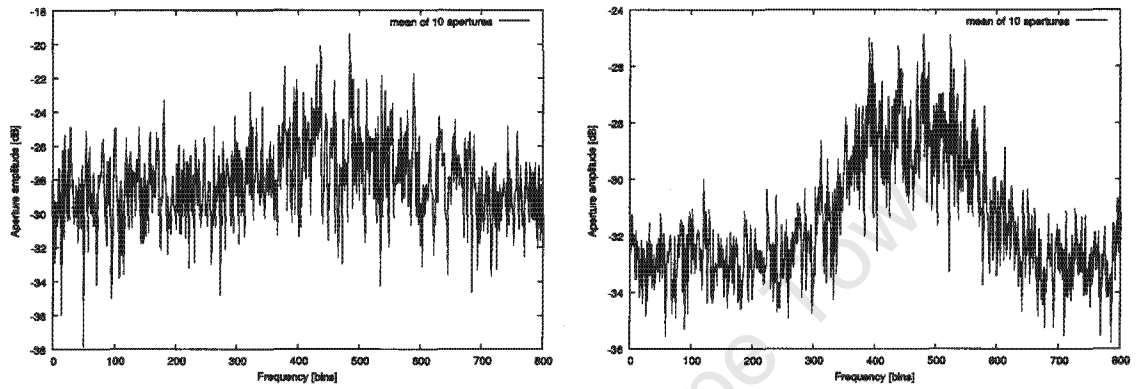
Figure 7.5: Aperture amplitude plots for various S/N

7.3.1 Incorporating direction information

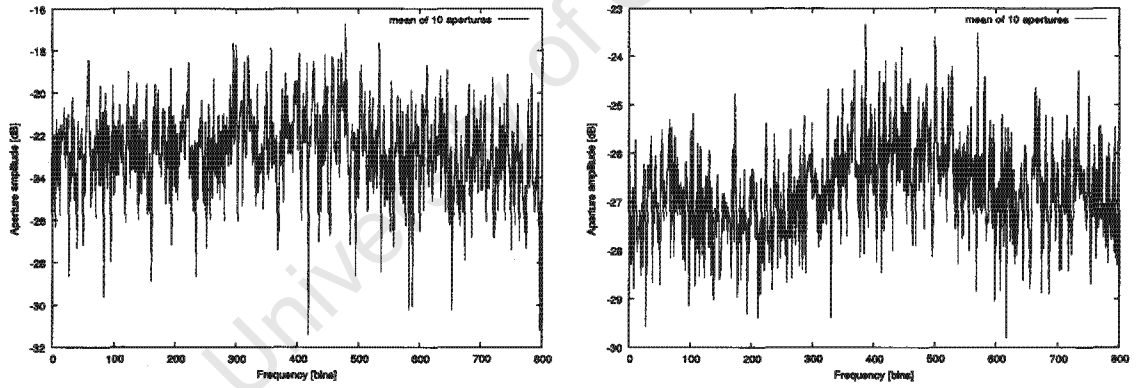
Figure 7.7 shows plots of direction-data on the y-axis of range 0 to 360° for different signal-to-noise ratios, and the effect of coherent averaging.

The most straight-forward way to incorporate direction information is to generate a cost function for direction data by applying a least-squares algorithm to it. Then, the power-

S/N = -4dB



S/N = -10 dB

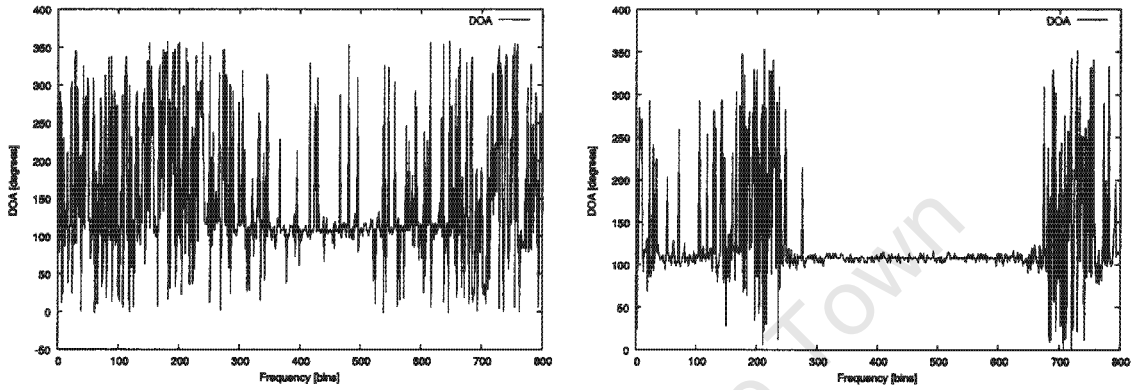


Single capture Coherent average of 10 captures

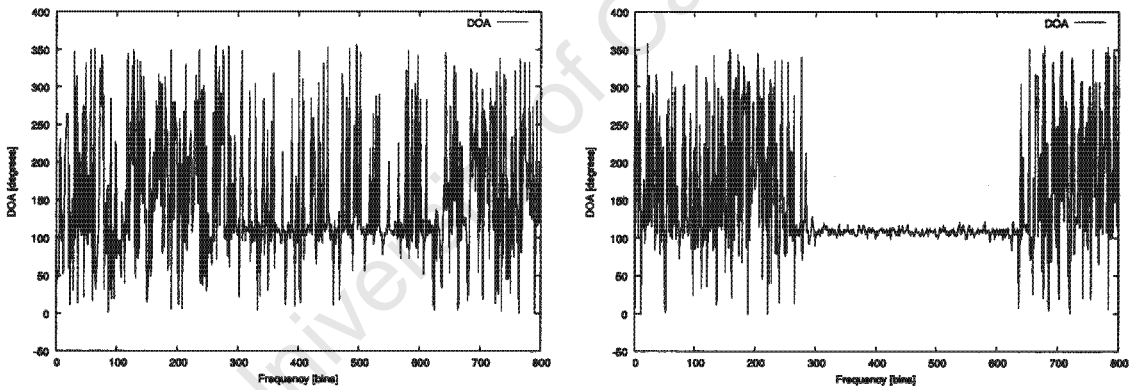
All plots are of mean of amplitudes of the ten apertures, ap_1 .. ap_{10}

Figure 7.6: Aperture amplitude plots for various S/N (continued)

S/N = 4dB



S/N = 0dB



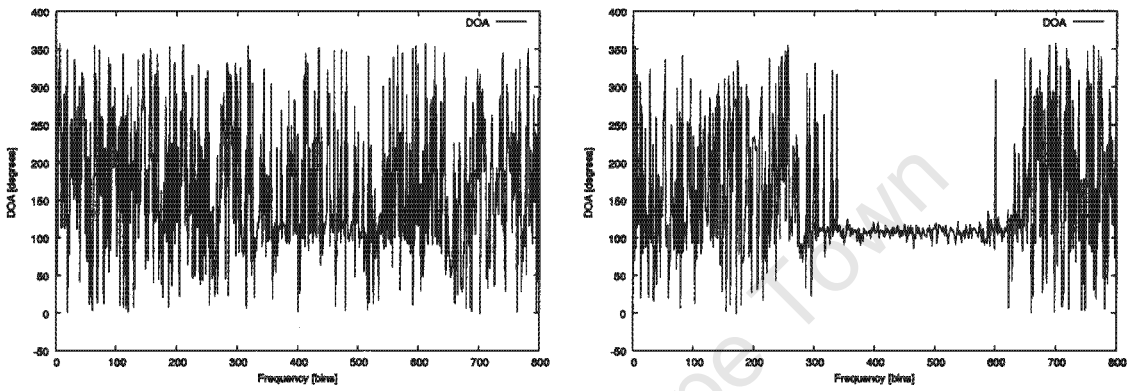
Single capture

Average of 10 captures

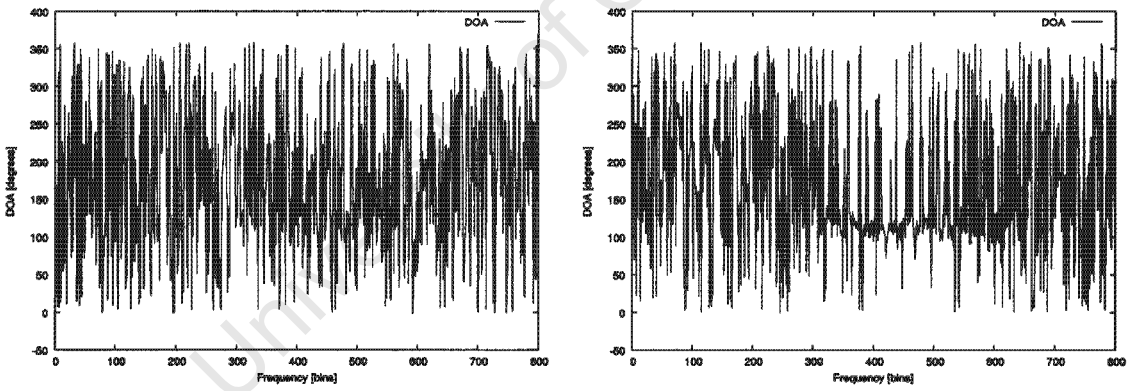
y-axis is estimated DOA in degrees, x-axis is frequency in bins

Figure 7.7: DOA plots for various S/N

S/N = -4dB



S/N = -10 dB



Single capture Average of 10 captures

y-axis is estimated DOA in degrees, x-axis is frequency in bins

Figure 7.8: DOA plots for various S/N (continued)

spectrum cost function and the direction cost-function are normalised, weighted and combined:

$$C(n, \theta_{ps}, \theta_{dir}) = W_{ps} \frac{C_{ps}(n, \theta_{ps})}{\max(C_{ps}(n, \theta_{ps}))} + W_{dir} \frac{C_{dir}(n, \theta_{dir})}{\max(C_{dir}(n, \theta_{dir}))} \tag{7.5}$$

where W_{ps}, W_{dir} are weighting factors that can be used to allow a component cost-function more/less say in the final outcome. C_{ps} is given by equation 7.4, with model parameters, $\theta_{ps} = \{u1, u2, f0, hB\}$ where

u1 - the lower power value, the noise-floor

u2 - the upper power value, the signal + noise peak

hB - the half-bandwidth

f0 - the centre-frequency

C_{dir} is given by

$$C_{dir}(n, \theta_{dir}) = \sum_{n=1}^{N} (D(n) - M_{dir}(\theta_{dir}, n))^2 \tag{7.6}$$

where n is the bin-index and D represents the actual direction data.

The model parameters, θ_{dir}, for M_{dir} will be defined shortly.

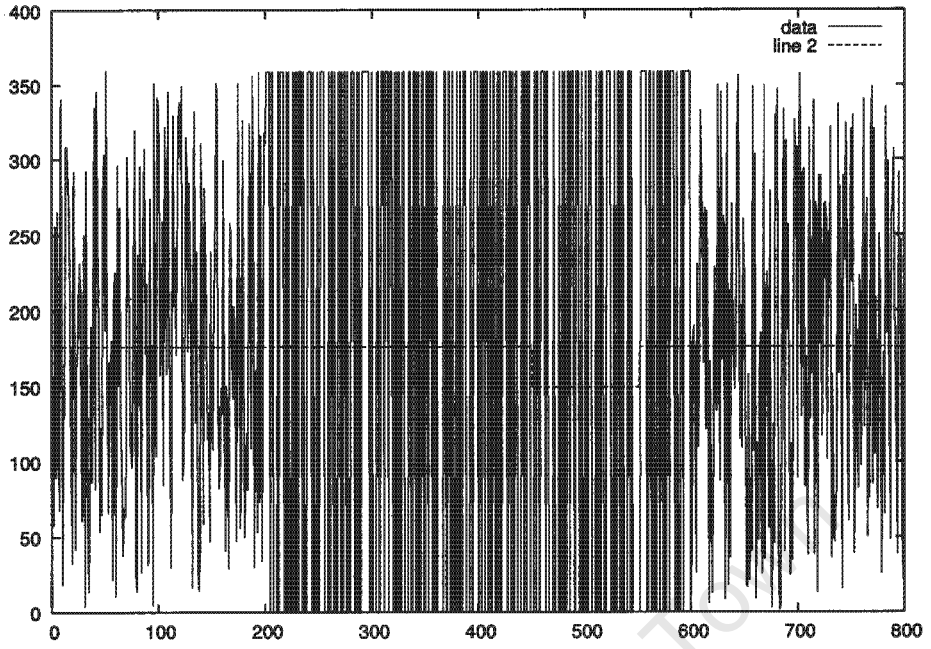
Finding a suitable model M_{dir} turns out to be a little tricky. To begin with, there is the fact that directions wrap, that is $360° \equiv 0°$.

Figure 7.9 shows artificially generated direction data. The centre 400 bins of figure 7.9 are set to either 359° or 0°. Line 2 in the figure shows the result of using the least-squares algorithm as it stands for power-spectrum fitting. The algorithm fails, because it does a simple difference ($359° - 0° = 359°$), rather than the angular (wrapped) difference ($-1° - 0° = -1°$).

This is straight-forward to fix. One method is to use $arg(e^{j(\Phi_1 - \Phi_2)})$, but this is computationally expensive with C++. A method that executes faster, is to add 360° to angles less than $-180°$, and to subtract 360° from angles more than $+180°$. This is easily implemented with 2 successive for loops.

A first attempt at incorporating DOA information was to use a tophat model as above for C_{ps}, with *u2* set to the DOA, and *u1* set to the angular-mean[1] of the direction data, with

[1] That is, $arg(mean(e^{j(\Phi_i \cdots)}))$.



y-axis is estimated DOA in degrees, x-axis is frequency in bins

Figure 7.9: Artificial DOA data demonstrating wrapping problem

the least-squares algorithm correctly computing the angular difference. The results were disappointing - using the tophat model results in consistent overestimates of the bandwidth. This is because the major component of C_{dir} is the sections of the data where the DOA is random; The section where the DOA is consistent (where the signal is) can be well fitted by a line, thus C_{dir} is small over this section. As a result, the algorithm is not biased to finding a good fit to the portion of data containing the signal. Rather, it is biased to reducing the cost over the portions with no signal, at the expense of the section with signal.

An alternative to the tophat model, is to define the model simply as a horizontal line, but only over the bandwidth $f0 - hB$ to $f0 + hB$. The (obvious) problem here is now the global minimum is always given by the smallest hB – since the sum of the squared differences over a few points will always be less than for more points, for a given $f0$. To counter this, $cost_{dir}$ could be scaled by $\frac{1}{hB}$. This was tried. Scaling by the reciprocal of hB helps, but is still not sufficient to produce a fit that returns something close to the actual bandwidth of the signal; The smallest hB is returned.

In the end, it was decided to use a pre-processing step to aid fitting a model to direction

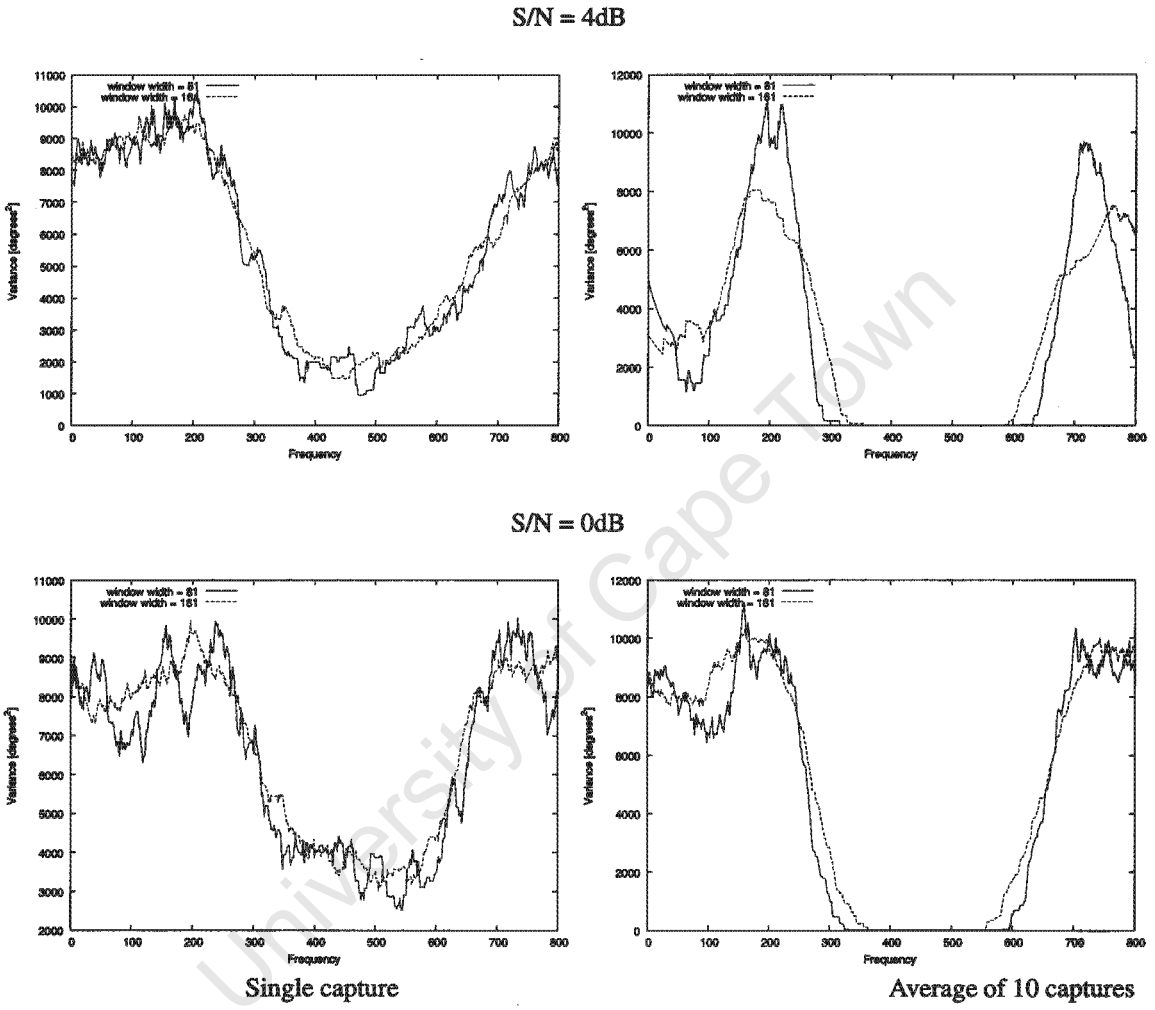
data. On considering a DOA plot for a typical capture of a relatively strong DSSS signal, it can be seen that *variance* is the key to discriminating signal from noise. Bins containing signal DOAs that vary little from one another; Bins with no signal have DOAs that vary wildly. Thus, a *moving-variance-window*[2] was applied to the direction data. For every bin in the input data, the *moving-variance-window* function computes the variance over N bins (where N is the width of the window, centred on the input bin), and assigns this value to the corresponding output bin. (Since in this case, the input was DOA data, care was taken to implement the function using angular variance, that is taking wrapping into account). This is illustrated in Figure 7.10.

Once the direction data is pre-processed in this manner, the existing tophat model least-squares algorithm can be applied to it.

Thus θ_{dir} is defined as $\{f0, hB, P1, P2\}$, where hB is the half-bandwidth, $f0$ is the centre-frequency, $P2$ is the value associated with the central domain of the tophat model (analagous to $u2$), and $P1$ is the value associated with the outer domain of the tophat model (analagous to $u1$).

Note that all calculations performed to fit a model to direction data occur in the linear domain.

[2]Although this function is similar to mean/median boxcar filters, since the output of this function is no longer a DOA, it should be considered a *transform,* rather than a *filter.*



The moving-variance-window function was applied to some of the DOA data illustrated in figure 7.7, with window widths of 81 and 161 bins. In the left-hand column, the function was applied to DOA data from a single capture. In the right-hand column, DOA data from 10 captures was first averaged before the moving-variance-window function was applied.

Figure 7.10: Results of applying moving-variance-window to DOA data

Chapter 8

Results

The primary metrics used to evaluate the performance of different methods are: a) the mean and standard-deviation of the centre-bin estimates, and b) a count of an estimator's centre-bin estimates falling outside of $\pm 20\%$ of the signal's true bandwidth from the true centre. The idea behind this second metric is to obtain an indicator of estimates that are "wrong".

Unless otherwise noted, certain simulator inputs (See section 5.1) were kept constant for all experiments:

sidelobe-filterON - false (Most DSSS signals do not suppress sidelobes.)

useLongCode - true

spreadFactor - arbitrarily set to 10 (Has little effect.)

DOA - arbitrarily set to 108°

antenna-radius - 0.5m (The dimension of the outer-array of the antenna used with the actual DF platform.)

RF-centre-frequency - 200MHz (This frequency ensures that $d_{pentagon} < \lambda/2$, a precaution to avoid possible break-down of the DF algorithm, as mentioned at the end of chapter 3.)

8.1 Model fitting using a single channel

8.1.1 Least-squares fit of tophat to power-spectrum

The input ranges for the 3 variables were:

S/N - [4, 2, 0]

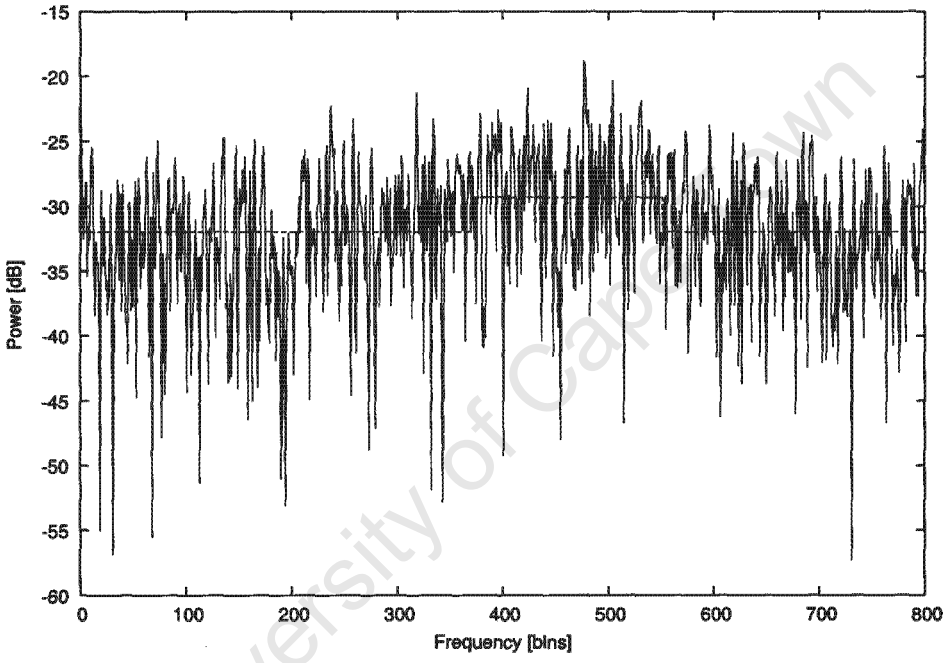
hB - [80, 120, 160, 200, 240]

f0 - 2000 numbers chosen randomly from the range 200 to 600.

This works out to 30000 frames of data - 10000 for every input S/N. In addition 3000 frames were generated which had no signal present, just pure noise.

The test machine (Intel P4 1.8GHz, 512MB RAM) took 1527 seconds to process 33000 frames of data.

Figure 8.1 shows the outcome of a typical fit.



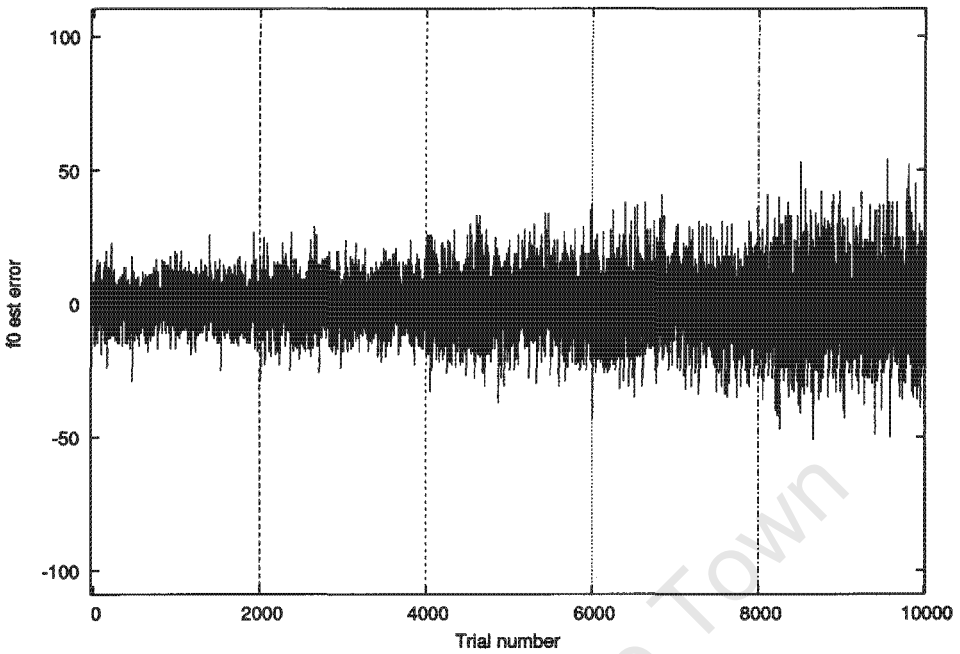
Signal parameters: $f0 = 460$, $hB = 240$, $S/N = 0dB$

Figure 8.1: Example of least-squares fit of tophat model

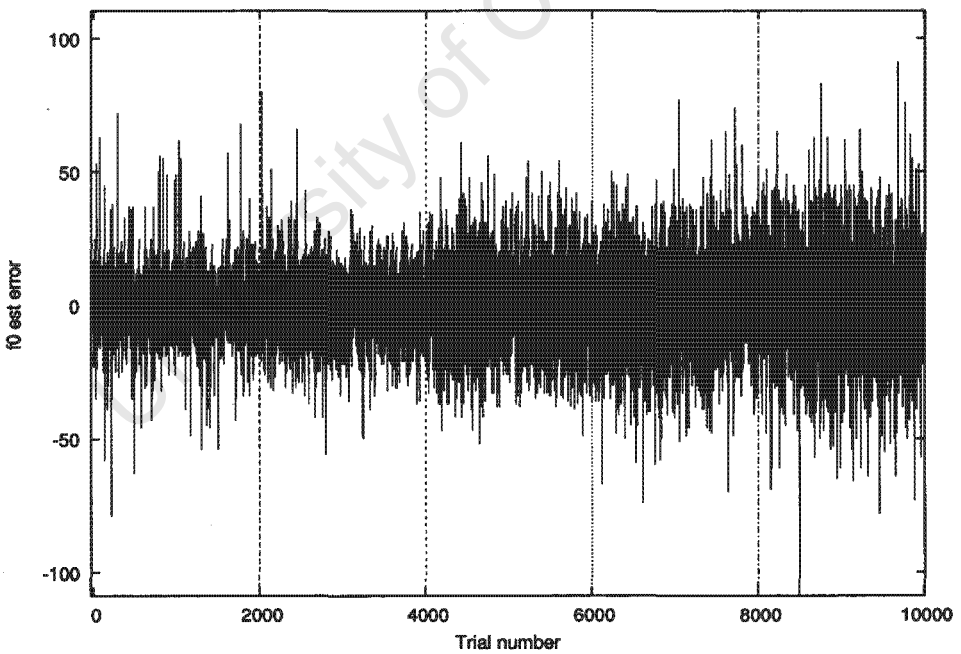
Figure 8.2 shows plots of the centre-bin estimate error for S/N of 4 dB and 0 dB.

Each plot represents 2000 trials at each of 5 different signal bandwidths: 160, 240, 320, 400 and 480 bins. Some observations:

1. the spread of the error is greater for the 0 dB data
2. the 0 dB data also has more estimates which could be considered "wrong" – that is they are not even close to the true value
3. spread of the error increases with increasing bandwidth



a) S/N = 4 dB



b) S/N = 0 dB

Figure 8.2: Comparison of centre-bin estimate error for two different S/N(tophat model)

Observations 1 and 2 are more clearly illustrated in Figure 8.3. Table 8.1 summarises the performance of the centre-bin estimates.

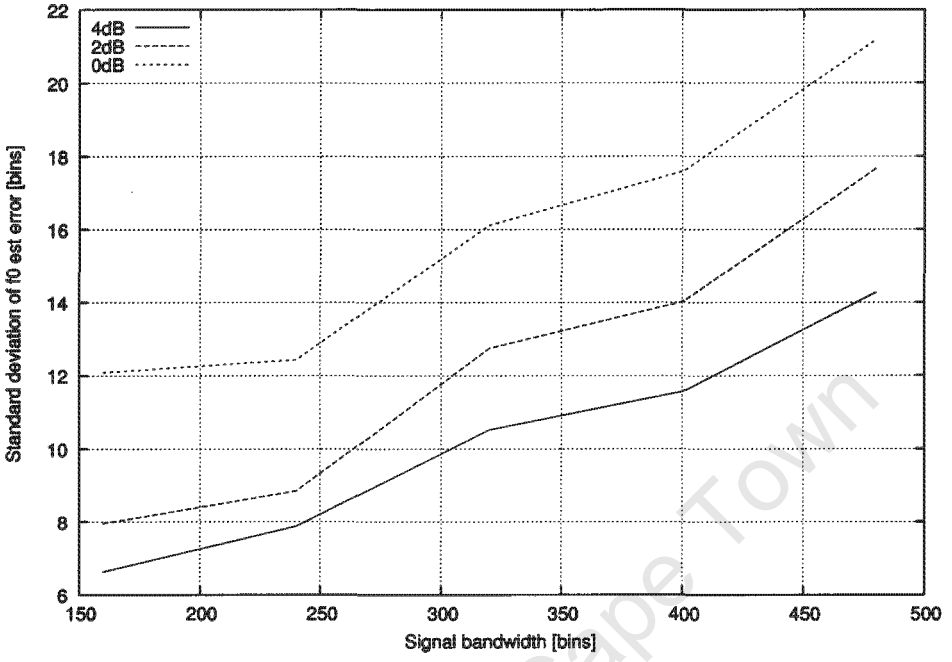


Figure 8.3: Trends of standard deviation of centre-bin estimate error(tophat model)

SNR	4 dB	2 dB	0 dB
Mean of absolute error [bins]	7.68	9.32	12.0
Outside $\pm 20\%$ of BW (count)	0	3	56
Outside $\pm 20\%$ of BW [%]	0	0.03	0.56
Outside ± 50 bins (count)	4	21	74

Table 8.1: Metrics relating to centre-bin estimate(tophat model)

8.1.2 Least-squares fit of dB scaled Sinc-squared to power-spectrum

The input ranges for the 3 variables were the same as those used for the previous section (the tophat model):

S/N - [4, 2, 0]

hB - [80, 120, 160, 200, 240]

f0 - 2000 numbers chosen randomly from the range 200 to 600.

The test machine took 16219 seconds to process 33000 frames of data.

Figure 8.4 shows the relationship between standard deviation, signal bandwidth and S/N. Table 8.1 summarises the performance of the centre-bin estimates.

The most notable thing about these results is that they are only marginally better than those of the previous section. It was expected that the dB scaled Sinc-squared model would produce noticably better results, being a better representation of the actual DSSS power spectral envelope. Since the errors are so small, it is difficult to say with certainty. For this reason, the dB scaled Sinc-squared model was also tested in section 8.2 over a greater S/N range.

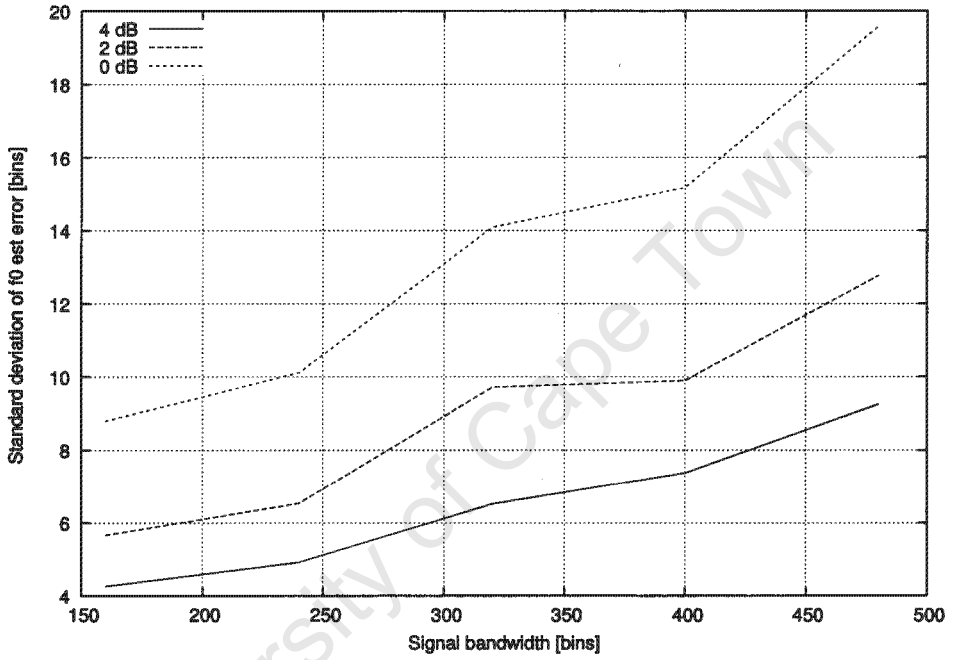


Figure 8.4: Trends of standard deviation of centre-bin estimate error (dB scaled Sinc-squared model)

SNR	4 dB	2 dB	0 dB
Mean of absolute error [bins]	5.08	6.81	9.99
Outside $\pm 20\%$ of BW (count)	0	2	14
Outside $\pm 20\%$ of BW [%]	0	0.02	0.14
Outside ± 50 bins (count)	0	12	87

Table 8.2: Metrics relating to centre-bin estimate (dB scaled Sinc-squared model)(Sinc model)

8.1.3 Human experts

Altogether there were 160 trials (8 candidates, 20 trials each). A *missed signal* occurs when a candidate incorrectly states there is no signal present. A *false detection* occurs when a candidate incorrectly states there is a signal present.

There were 28 missed signals, and 1 false detection.

Table 8.3 summarises the errors candidates made in estimating centre frequency.

SNR	4 dB	2 dB	0 dB
Total trials	40	48	48
Missed signals	3	3	22
Outside ±20% of BW	0 / 37	0 / 45	1 / 26
Outside ±20% of BW [%]	0	0	3.8

Table 8.3: Centre-frequency estimation errors - human experts

Table 8.4 summarises the errors candidates made in estimating signal bandwidths. The bandwidth here of a signal is defined as the main-lobe null-to-null bandwidth. Since the nulls are not visible in the presence of significant noise (see figures 7.5 and 7.6 for typical plots), under-estimates of the bandwidth are expected, and thus the tolerance is set more leniently for under-estimates, but more strictly for over-estimates.

SNR	4 dB	2 dB	0 dB
Estimates > 120%	1	1	2
Estimates < 50%	1	8	3
Total	2	9	5
Out of	37	45	26
Error [%]	5.4	20	19.2

Table 8.4: Bandwidth estimation errors - human experts

8.1.4 Consistency and measure of confidence

The current implementation of the least-squares algorithm cannot give an indication of confidence in its estimate, and thus cannot identify cases where no signal is present. In practice this is not very helpful. However, by comparing the results returned by the least-squares algorithm applied to a series of subsequent snap-shots of the same frequency band, it should be possible to obtain a level of confidence in the estimate. Strong signals

produce a tight grouping of centre bin estimates, while pure noise produces inconsistent results.

This was verified experimentally, by running the least-squares algorithm on 1000 captures for 3 separate cases:

Case 1: DSSS signal present, $f0$ = bin 460, hB = 240 bins, S/N = 4dB.

Case 2: This was the same as case 1, except S/N = -2dB.

Case 3: No signal present.

See Figure 8.5.

8.2 Using multiple channels

To compare the effectiveness of of amplitude only, direction only and combined approaches, three experiments were performed.

The input ranges for the 3 variables for all three experiments:

S/N - [4, 2, 0, -2, -4, -6, -8, -10, -12]

hB - fixed at 240

$f0$ - fixed at 460

1000 frames of data were generated at each S/N level, thus 9000 frames in total.

8.2.1 Least-squares fit of tophat to aperture-amplitude

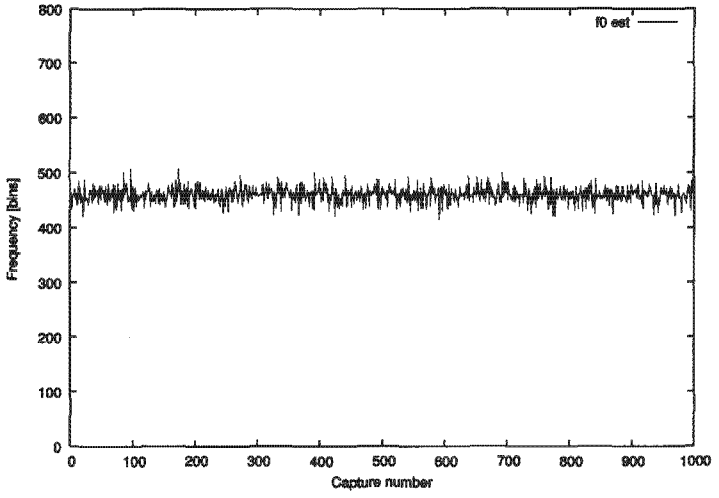
The input data here was the mean amplitude spectrum of the 10 apertures produced per capture. It must be emphasised that this mean is just to combine the 10 apertures magnitudes into a single array, suitable for processing by the least-squares algorithm, and should not be confused with coherent averaging of multiple captures. The averaging process here is incoherent.

The test machine took 779 seconds to process 9000 frames of data.

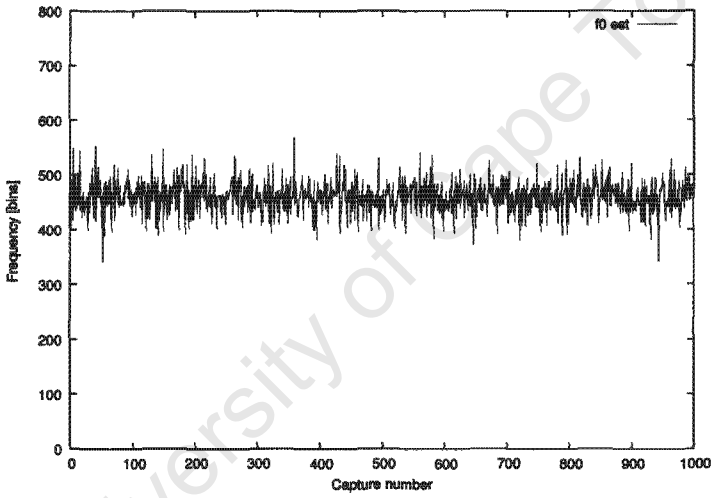
Table 8.5 summarises the performance of the centre-bin estimates.

8.2.2 Least-squares fit of dB scaled Sinc-squared model to aperture-amplitude

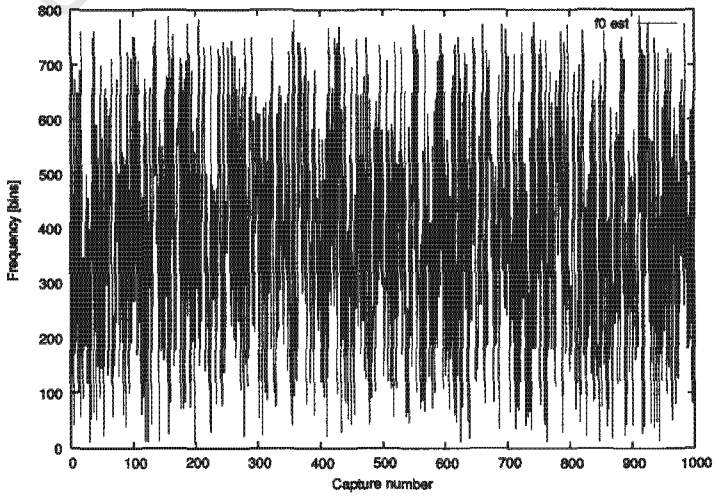
Input data was the same as for section 8.2.1.



a) S/N = 4 dB



b) S/N = -2 dB



c) no signal

Figure 8.5: Comparison of centre-bin estimate for successive captures

SNR	4 dB	2 dB	0 dB	-2 dB	-4 dB	-6 dB	-8 dB	-10 dB	-12 dB
Mean of absolute error [bins]	5.29	6.79	8.65	10.56	15.58	20.62	31.31	57.36	94.89
Outside ±20% of BW (count)	0	0	0	0	0	5	34	147	307
Outside ±20% of BW [%]	0	0	0	0	0	0.5	3.4	14.7	30.7
Outside ±50 bins (count)	0	1	2	2	23	64	181	392	540

Table 8.5: Metrics relating to centre-bin estimate
(mean aperture amplitude only, tophat model)

The test machine took 8101 seconds to process 9000 frames of data. The results are shown in table 8.6.

SNR	4 dB	2 dB	0 dB	-2 dB	-4 dB	-6 dB	-8 dB	-10 dB	-12 dB
Mean of absolute error [bins]	3.17	3.81	5.34	7.26	11.39	19.72	32.48	59.16	95.87
Outside ±20% of BW (count)	0	0	0	0	0	10	45	155	323
Outside ±20% of BW [%]	0	0	0	0	0	1	4.5	15.5	32.3
Outside ±50 bins (count)	0	0	0	0	8	72	216	411	566

Table 8.6: Metrics relating to centre-bin estimate
(mean aperture amplitude only, Sinc-squared model)

8.2.3 Least-squares fit of tophat to variance filtered direction data

The direction data output by the simulator was processed with a moving-variance-window of width 81 (as described in section 7.3.1), before being passed into the least-squares algorithm.

The test machine took 844 seconds to process 9000 frames of data.

Table 8.7 summarises the performance of the centre-bin estimates.

SNR	4 dB	2 dB	0 dB	-2 dB	-4 dB	-6 dB	-8 dB	-10 dB	-12 dB
Mean of absolute error [bins]	10.46	11.59	13.08	18.28	23.13	36.53	61.59	128.21	156.62
Outside ±20% of BW (count)	0	0	0	1	8	60	161	422	549
Outside ±20% of BW [%]	0	0	0	0.1	0.8	6	16.1	42.2	54.9
Outside ±50 bins (count)	0	2	10	41	93	224	400	621	730

Table 8.7: Metrics relating to centre-bin estimate (direction only)

8.2.4 Combined error function

(As in equation 7.5, with W_{ps} and W_{dir} both set to 0.5)

The test machine took 1609 seconds to process 9000 frames of data.

Table 8.8 summarises the performance of the centre-bin estimates.

SNR	4 dB	2 dB	0 dB	-2 dB	-4 dB	-6 dB	-8 dB	-10 dB	-12 dB
Mean of absolute error [bins]	8.11	9.3	11.08	15.7	21.18	32.81	59.28	126.18	156.68
Outside ±20% of BW (count)	0	0	0	0	6	45	150	407	547
Outside ±20% of BW [%]	0	0	0	0	0.6	4.5	15	40.7	54.7
Outside ±50 bins (count)	0	0	3	18	68	201	374	613	728

Table 8.8: Metrics relating to centre-bin estimate (amplitude and direction combined)

8.2.5 Discussion of combined error function results

It was expected that the combined error function would outperform both the component error functions.

It is thought that the reason for the poorer results from the combined algorithm is because the power-spectrum and direction cost functions are given equal weighting. The direction data based centre-bin estimate error has greater variance than the aperture-amplitude based centre-bin estimate error. Incorporating the variances into the weighting-factors W_{ps} and W_{dir} in equation 7.5 could possibly improve these results.

Chapter 9

Conclusions

- DSSS detection from a snap-shot of the power-spectrum by least-squares fitting of a model seems to perform at least as well as humans, in terms of accuracy. (c.f. "Outside $\pm 20\%$ of BW [%]" in Tables 8.1, 8.2 and 8.3).
- The dB-scaled Sinc-squared model doesn't appear to give significantly more accurate results, yet is much more computationally expensive (approximately 10 times more expensive than the tophat model). It was expected that since the dB-scaled Sinc-squared model more closely approximates a DSSS signal power spectral envelope, it would produce noticably better results than the tophat model.
- Comparison of results returned by the least-squares algorithm for data with signal present against data with no signal present does not reveal a discriminant for the no-signal case *for a single capture*.
- By making use of successive *multiple captures* it is possible to identify the no-signal case (Figure 8.5). Using multiple captures, it should be possible to construct a measure of confidence in the model fit.
- Using mean of aperture-product amplitudes data as input to the model-fitting algorithm produces better results than a single channel power-spectrum. This is not unexpected, since the aperture-products are formed by the product of two antenna channels, and the noise on each of these channels is uncorrelated. In addition the aperture-products are combined by averaging, which reduces the variance of the noise.
- Using variance processed direction data as input to the model-fitting algorithm produces poorer results than using mean of aperture amplitudes data. (c.f. Tables 8.6 and 8.7).

- Using an *equally weighted* combined cost function produces poorer results than using the cost function for mean of aperture amplitudes data. (c.f. Tables 8.6 and 8.8).

Chapter 10

Future Work

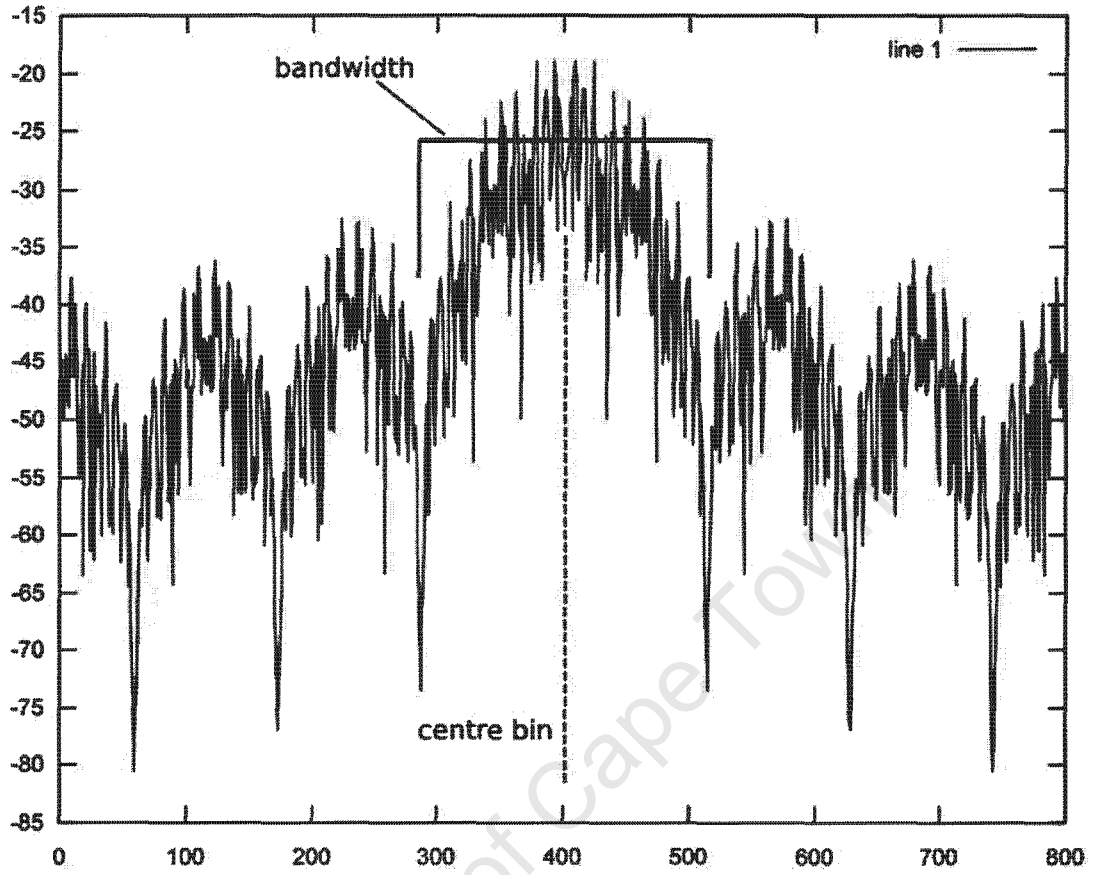
- Establish whether centre-bin estimates based on variance-filtered direction data and aperture-amplitude data are unbiased, independent estimates.
- Investigate the effect of modifying weighting parameters on the combined error function algorithm.
- Rigorous analysis of algorithm processing-cost should be performed in order to identify areas for optimisation.
- Optimising the search for global minimum.
- Optimising the least-squares fit of a tophat model to a single channel's power spectrum / combined aperture-product amplitudes for a particular processor/architecture, such as the exisitng DF platform. This might involve specialised hardware, such as an FPGA configured to perform certain parts of the processing in parallel.
- The algorithm needs to be tested for cases where narrow-band signals are present. A pre-processing step will probably need to be included to identify/eliminate narrow-band signals.
- The simulator needs to be extended to generate data for multiple signals (both narrow-band and DSSS) in the same 10 MHz band.
- Test the algorithms on real data.

Appendix A

Human Expert Questionnaire

Instructions

A DSSS signal looks something like this:



In this image, there is no noise. In the 20 test images, there is substantial additive white Gaussian noise. Typical S/N will be in the range 0 to 5 dB. This means the side-lobes will most-likely not be visible. Just attempt to identify the main-lobe of the DSSS signal, if it is present.

- Spend about 10-20 seconds per image. No more than, say, 30 seconds.
- For each image:
    1. Is there a DSSS signal present?
    2. If so,
        a) What is the centre bin?
        b) What is the bandwith of the signal (in bins)?

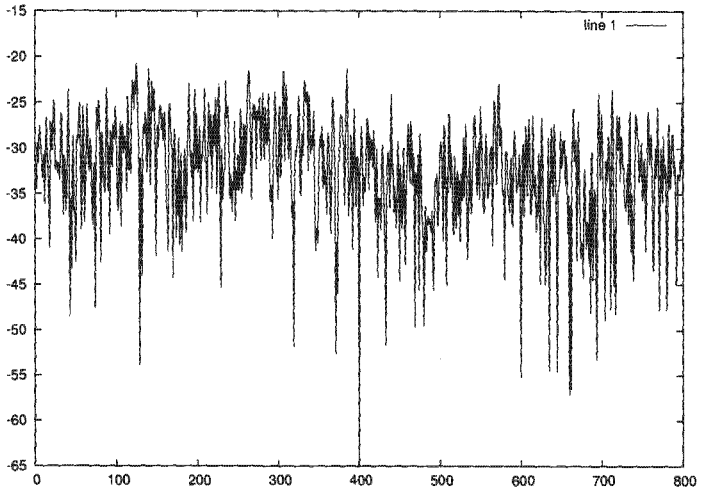


Figure 1: _

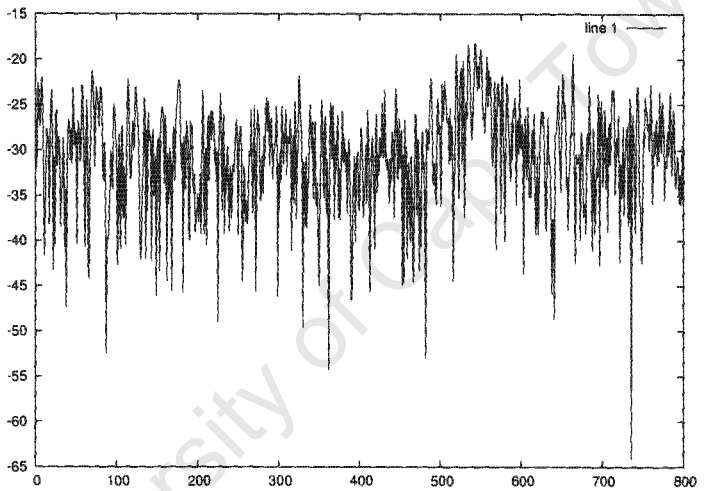


Figure 2: _

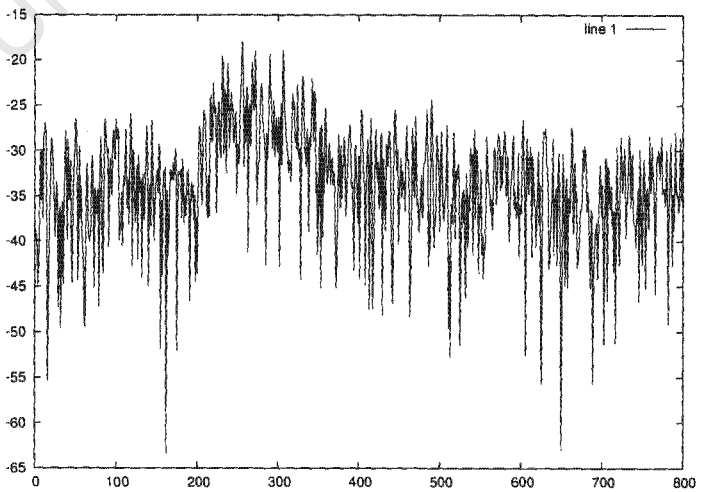


Figure 3: _

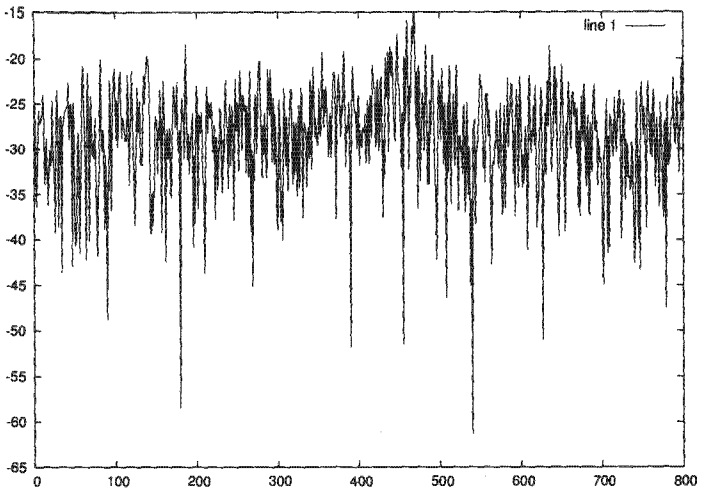


Figure 4: _

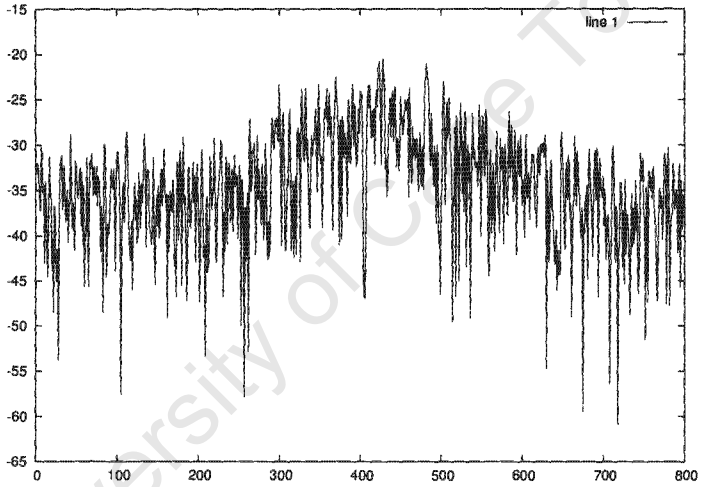


Figure 5: _

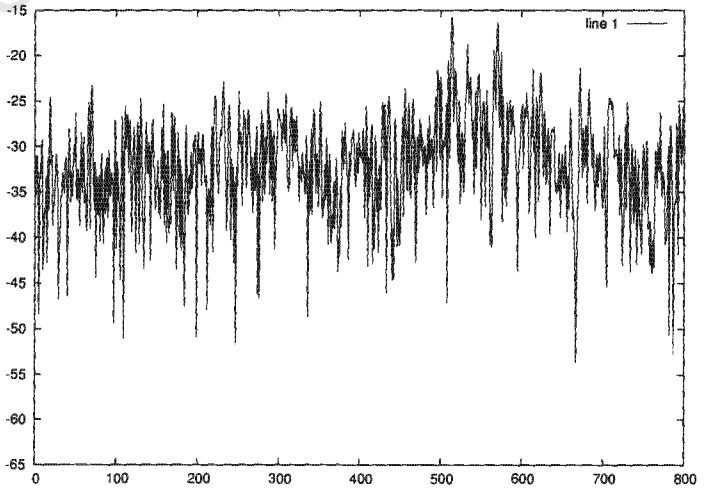


Figure 6: _

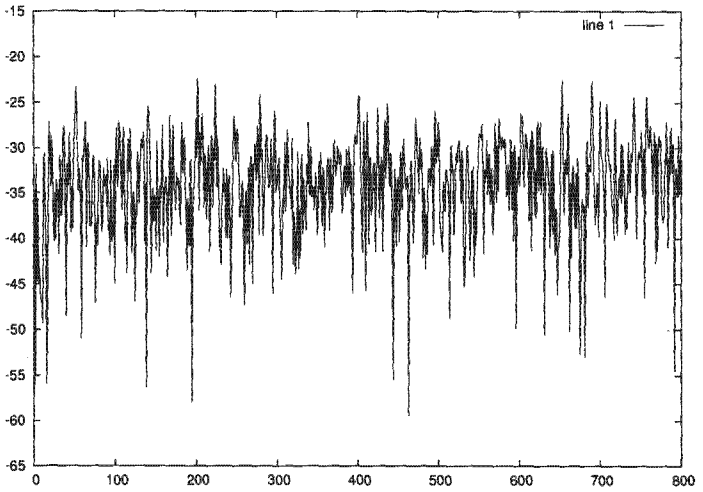


Figure 7: _

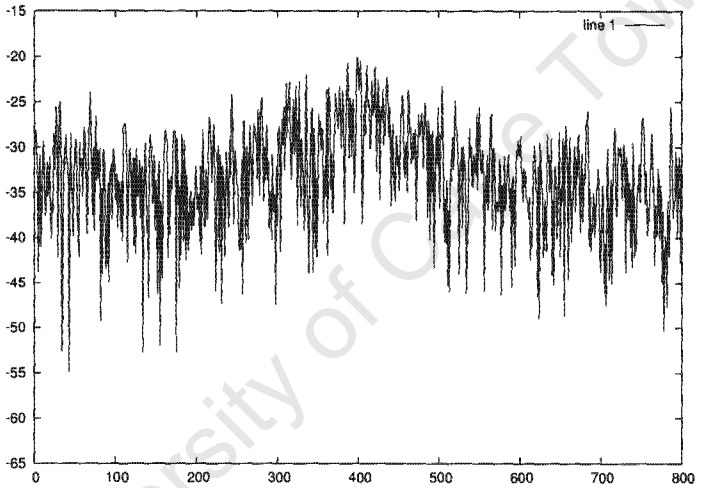


Figure 8: _

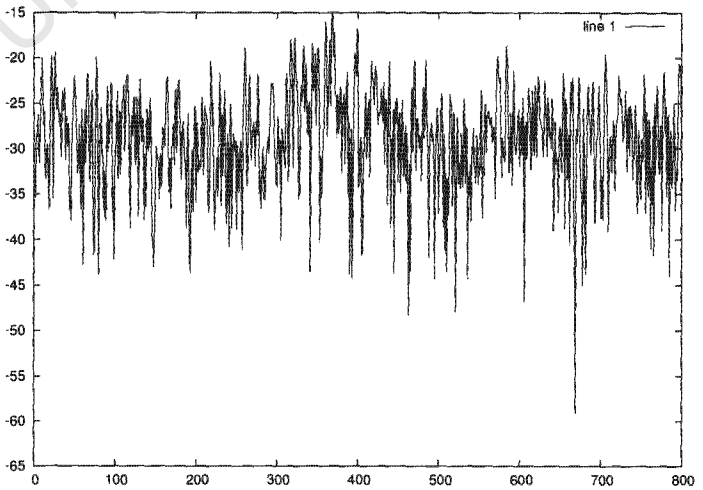


Figure 9: _

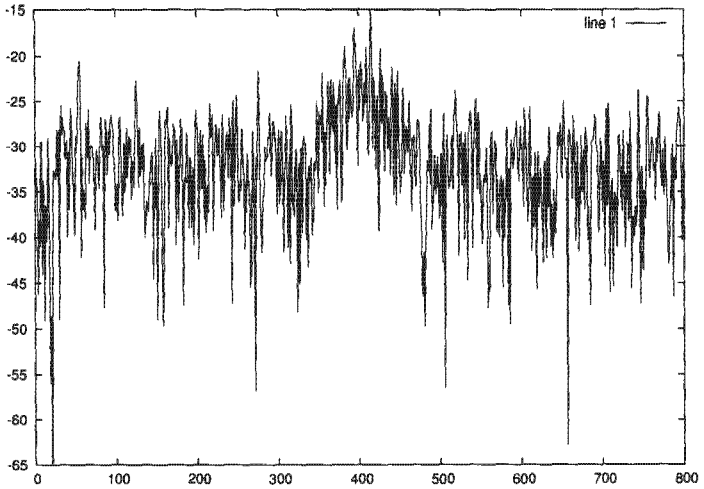


Figure 10: _

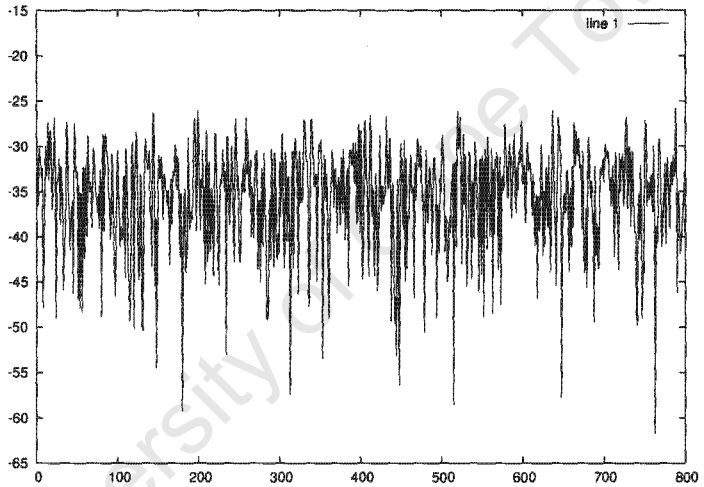


Figure 11: _

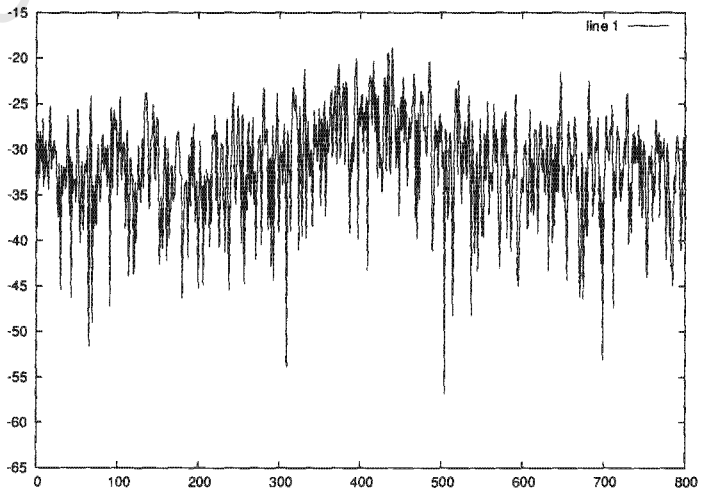


Figure 12: _

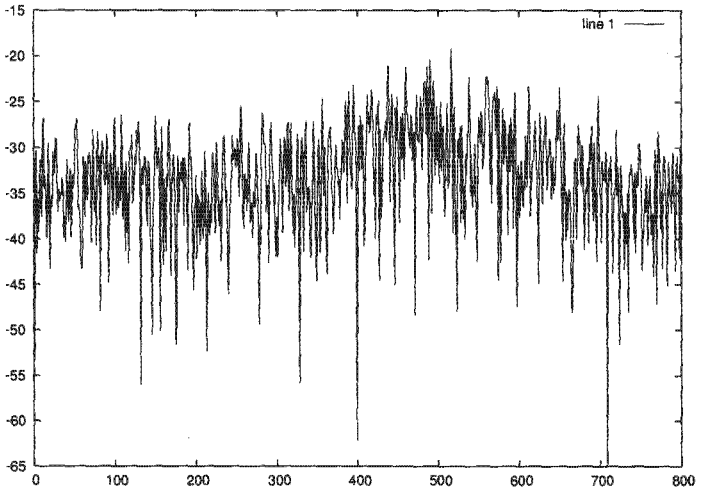


Figure 13: _

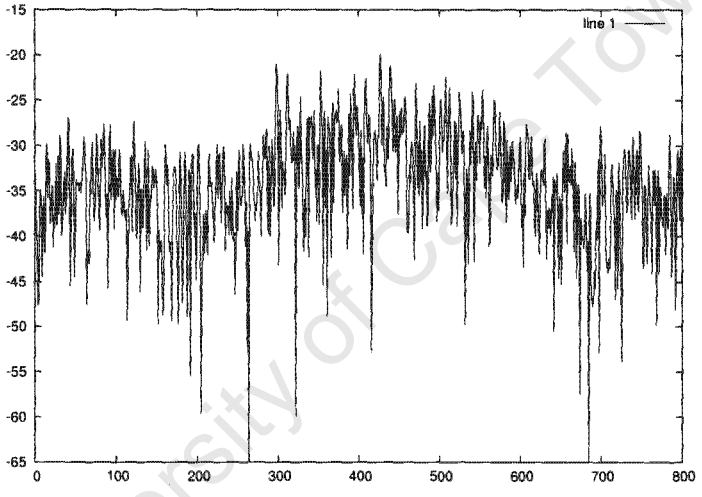


Figure 14: _

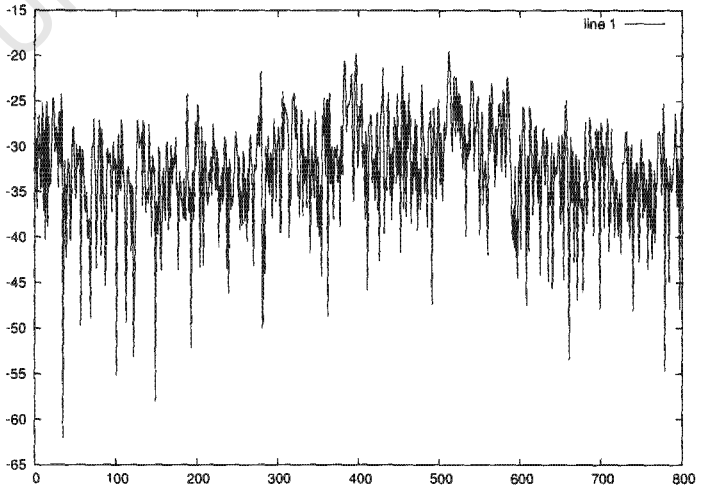


Figure 15: _

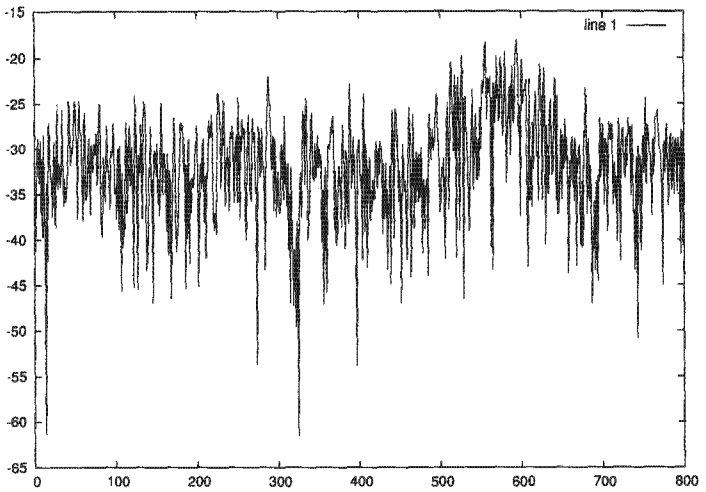


Figure 16: _

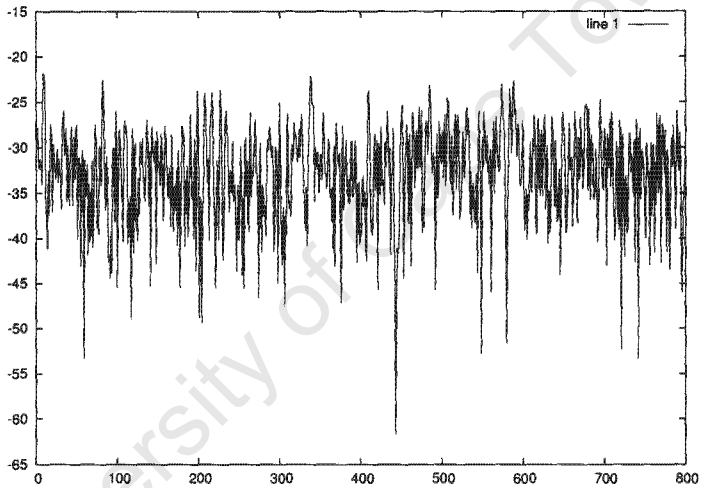


Figure 17: _

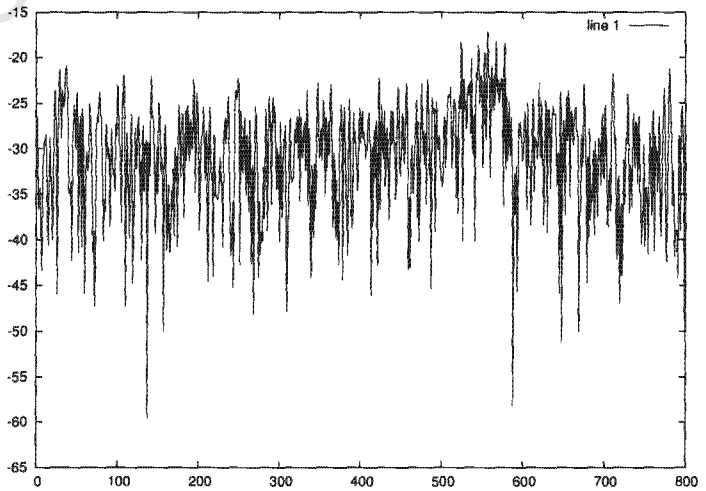


Figure 18: _

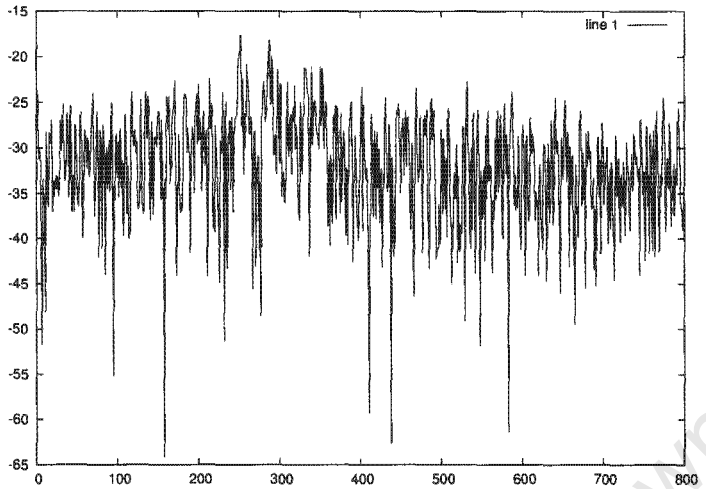


Figure 19: _

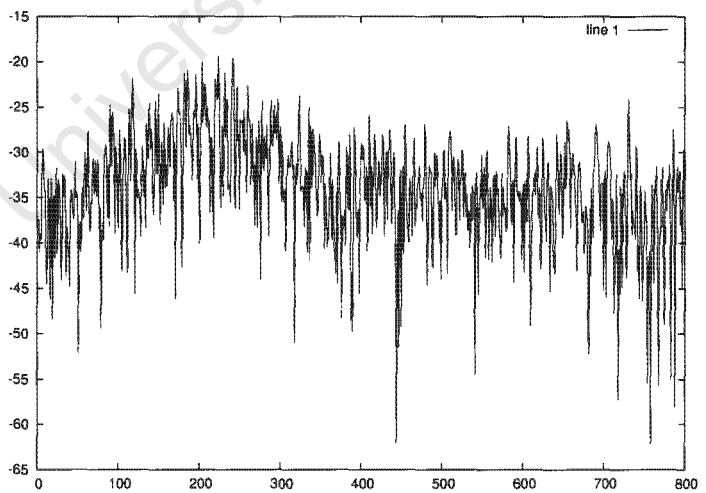


Figure 20: _

Bibliography

[1] 3GPP2. Physical Layer Standard for cdma2000 Spread Spectrum Systems - Revision D v2.0, 2005. Available from: http://www.3gpp2.org/Public_html/specs/tsgc.cfm.

[2] ADAMY, D. *EW101: A First Course in Electronic Warfare*. Artech House, 2001.

[3] BORSE, G. J. *Numerical Methods with MATLAB: A Resource for Scientists and Engineers*. PWS, 1997.

[4] DILLARD, R. A., AND DILLARD, G. M. *Detectability of Spread-Spectrum Signals*. Artech House, 1989.

[5] DIXON, R. C. *Spread spectrum systems with commercial applications*, 3rd ed. John Wiley and Sons, 1994.

[6] ETSI 3RD GENERATION PARTNERSHIP PROJECT (3GPP). Universal Mobile Telecommunications System (UMTS): Spreading and modulation (FDD) (3GPP TS 25.213 version 6.4.0 Release 6). Available from: http://www.3gpp.org/ftp/Specs/html-info/25-series.htm.

[7] FINNE, M. Methods for direction-finding of direct-sequence spread-spectrum signals. Tech. rep., FOA - National Defence Research Establishment, Sweden, May 1996.

[8] GLISIC, S., AND VUCETIC, B. *Spread Spectrum CDMA Systems for Wireless Communications*. Artech House, 1997.

[9] G.M.BOYNTON, G. . Efficient design of event-related fmri experiments using m-sequences. *NeuroImage 16* (2002), 801–813.

[10] HAGGARTY, R. D., KEY, E. L., KRAMER, D. R., AND PALO, E. A. Spread spectrum communications and signal processing requirements. In *Case Studies in Advanced Signal Processing* (September 1979), M. Grant, Ed., IEE, pp. 76–84.

[11] HAHN, B. D. *Essential MATLAB for Scientists and Engineers*. Prentice Hall South Africa, 1997.

[12] IEEE. ANSI/IEEE Std 802.11, 1999 Edition (R2003). Available from: `http://standards.ieee.org/getieee802/802.11.html`.

[13] JENKINS, H. H. *Small-Aperture Radio Direction-Finding*. Artech House, 2002.

[14] JOHANSSON, E. Direction finding of direct-sequence spread spectrum signals. Tech. rep., FOA - National Defence Research Establishment, Sweden, April 1998.

[15] KAHN, D. Cryptology and the origins of spread spectrum. *IEEE Spectrum 21*, 9 (September 1984), 70–80.

[16] LAMBERT-PORTER, J. The detection and tracking of portable gsm handsets using a 5-element circular array. Master's thesis, University of Cape Town, 2004.

[17] LYONS, R. G. *Understanding Digital Signal Processing*. Prentice Hall PTR, 2001.

[18] MARKEY, H. K., AND ANTHEIL, G. Secret communication system. *US Patent 2,292,387* (August 11 1942).

[19] MEEL, J. Spread spectrum introduction, October 1999. De Nayer Instituut.

[20] POISEL, R. A. *Introduction to Communication Electronic Warfare Systems*. Artech House, 2002. Peralex have this book. Good background text on EW. Each chapter has a list of references.

[21] ROHDE & SCHWARZ. Introduction into theory of direction finding. Available: http://www.rohde-schwarz.com.

[22] RYAN, M. J., AND FRATER, M. R. *Tactical Communications for the Digitized Battlefield*. Artech House, 2002.

[23] SCHOLTZ, R. A. The origins of spread-spectrum communications. *IEEE Transactions on Communications COM-30* (1982), 822–854.

[24] SKLAR, B. *Digital Communications*, 2nd ed. Prentice Hall, 2001.

[25] STREMLER, F. G. *Introduction to Communication Systems*, third ed. Addison-Wesley, 1990.

[26] SVANSTROM, M. Methods for detection of direct sequence spread-spectrum signals in a high interference environment. Tech. rep., FOA - National Defence Research Establishment, Sweden, 1995. FOA-R-1995.

[27] TAYLOR, S. *Intel Integrated Performance Primitives - How to Optimize software Applications Using Intel IPP*. Intel Press, 2004.

[28] TORRIERI, D. J. *Principles of Secure Communications*. Artech House, 1985.

[29] VACCARO, D. D. *Electronic Warfare Receiving Systems*. Artech House, 1993.

[30] VARIOUS. ITU Spectrum Monitoring Handbook 2002. Tech. rep., International Telecommunications Union, 2002.

[31] VITERBI, A. J. Spread spectrum communications - myths and realities. *IEEE Communications Magazine* (May 1979), 11–18.

[32] WEADON, P. D. The SIGSALY story [online, cited 13 September 2005]. Available from: `http://www.nsa.gov/publications/publi00020.cfm`.

[33] WHAITS, C. V. Investigation into variable phase shift keying using direct sequence spread spectrum techniques. Master's thesis, University of Cape Town, 1997.

[34] WIKIPEDIA ARTICLE. Direct-sequence spread spectrum - Wikipedia Article [online]. August 2005. Available from: `http://en.wikipedia.org/wiki/Direct-sequence_spread_spectrum`.

[35] WIKIPEDIA ARTICLE. IEEE 802.11 [online]. July 2005. Available from: `http://en.wikipedia.org/wiki/IEEE_802.11`.

[36] WILKINSON, A. J. Discussions concerning least-squares approach.