A 50 MHz FMCW radar for the study of E-region coherent backscatter

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Physics and Engineering Physics
University of Saskatchewan
Saskatoon

By

Joel Cooper

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Physics and Engineering Physics
116 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5E2

Abstract

A 50 MHz E-region coherent backscatter radar was designed based on frequency modulated continuous wave (FMCW) radar techniques. This thesis presents the theory behind the FMCW technique and its implementation in a practical radar system. The system was designed and constructed at the University of Saskatchewan and was field tested at a radar site a few kilometres from the university. This thesis demonstrates that an FMCW radar is technically possible and functional as a research tool for E-region coherent backscatter studies.

The primary goal of this research is to develop a better understanding of the plasma processes responsible for the radar echoes. FMCW techniques offer a compromise between the pulsed and continuous wave (CW) radar techniques, which have previously been used for E-region experiments. CW techniques provide excellent spectral measurements but are limited in their ability to determine range information. Pulsed techniques offer excellent range resolution but may be limited in their ability to make detailed high resolution Doppler measurements of E-region radar backscatter. The implementation of FMCW techniques provides a simple and effective method of simultaneously obtaining excellent Doppler and range measurements.

The use of FMCW techniques is a novel approach to E-region coherent backscatter studies. Data analysis techniques were developed to extract the range and Doppler information from FMCW radar echoes. In the first few months of operation, the radar observed all four typical E-region radar signatures, Type I to Type IV, plus meteor trail echoes. Observations of each type of radar echo are presented, without interpretation, to illustrate the performance of the radar.

Acknowledgements

Thank you to my supervisor, Glenn Hussey, for giving me the opportunity to work on this project. He deserves thanks for his encouragement, support, advice, hard work, and patience throughout the entire process. I would also like to thank my committee members for their support and comments.

Many people in the ISAS department deserve mention. In particular, Bill Marshal for his contribution to numerous aspects of the project. Without his time and expertise in numerous areas, this project would not have been possible. I would also like to thank Hercules Olivier and Jan Wild for their time and advice. A sincere thank you also goes to Daron Chabot who made significant contributions to the radar control software.

A special thanks to Siobhàn Brennan for her patience, grammar advice, and hours of proofreading.

To Siobhàn for all her love and support.

Contents

List of Tables

List of Figures

List of Abbreviations

AC	Alternating Current
BW	Bandwidth
CW	Continuous Wave
dB	Decibel
dBm	Decibel proportional to milliwatts
DC	Direct Current
DDS	Direct Digital Synthesis
FFT	Fast Fourier Transform
FIFO	First In First Out
FM	Frequency Modulation
FMCW	Frequency Modulated Continuous Wave
GPS	Global Positioning System
HF	High Frequency
I channel	In phase channel
NTP	Network Time Protocol
Q channel	Quadrature phase channel
RF	Radio Frequency
RMS	Root Mean Square
Rx	Received
SNR	Signal to Noise Ratio
Tx	Transmitted
UT	Universal Time
UTC	Coordinated Universal Time
VHF	Very High Frequency
WERA	WEllen RAdar

Chapter 1

Introduction

For this thesis, a radar system was developed based on the FMCW radar technique. The goal was to simultaneously obtain higher resolution in both spatial and temporal domains than previous radar techniques. The radar system was a 50 MHz FMCW radar designed for the study of E-region coherent backscatter. The radar was constructed and operated at a site near Saskatoon, Saskatchewan. This thesis describes the technical details involved in the construction and operation of the radar, including the design of the radar electronics, software and data analysis techniques. Data collected by the radar are also presented as an illustration of radar performance.

1.1 Sun Earth interaction

The sun emits a stream of high speed electrons and protons that form the solar wind. The solar wind is a hot conducting plasma with a large amount of kinetic and electrical energy. The magnetic field from the sun can be trapped by the charged particles and carried with the solar wind. This trapped magnetic field interacts with the Earth's magnetic field. As a result, some of the Earth's magnetic field lines have the potential to connect with the solar wind. Magnetic field lines act as good electrical conductors, and the high energy charged particles of the solar wind can travel down the connected field lines and into the Earth's ionosphere. These high energy particles generate the visual aurora through collisions with particles in the Earth's atmosphere. The interaction of these ionised particles with the atmosphere, as well as the electric and magnetic fields present in the ionosphere, give all the

conditions needed for the generation of plasma processes or instabilities.

Plasma processes are generated throughout the ionosphere; however, the focus of this thesis is the lower E-region at auroral latitudes. The E-region exists at altitudes between 90–150 km, and the lower E-region is at an altitude of 90-120 km. The plasma processes in this region are distinguished by the fact that electrons are considered magnetised because the cyclotron frequency, ω_c, of the electrons ($\omega_c = \frac{qB}{m_e}$ where B is the magnitude of the magnetic field, q is the charge of an electron and m_e is the mass of an electron) is much greater than the frequency of collisions with other particles. On the other hand, the ions are considered collisional because the frequency of collisions with other particles is much larger than the cyclotron frequency.

Small perturbations in the electron density are unstable when the relative motion of the electrons and ions exceeds the ion-acoustic speed ($\sim$360 m/s in the E-region). Instabilities created when the relative motion of the ions and electrons exceeds the ion-acoustic speed are called two-stream instabilities. Perturbations in the plasma density will also grow if the local density gradient is sufficiently large and directed along a component of the electric field. A density gradient parallel to the electric field can activate the gradient drift instability that produces plasma waves with speeds that are (much) lower than the ion-acoustic speed. Radar signals propagating through these waves can scatter from the variations in the plasma densities. The scattering process is similar to Bragg scattering of X-rays from crystals and generates coherent backscatter [*Kelley*, 1989; *Kivelson and Russell*, 1997].

1.2 Radar techniques

The word radar originated as an acronym for *ra*dio *d*etection *a*nd *r*anging. Radar uses electromagnetic radiation to make remote measurements. Position and velocity are most commonly measured, but other reflective properties can be of interest depending on the application. There are various techniques for remote detection, with the most notable being pulsed, continuous wave (CW), and FMCW techniques. Both continuous wave and pulsed radar techniques have been used to study the E-region.

In their most basic form, pulsed systems send out a pulse of energy and wait for it to return. The time it takes for the pulse to return is a measure of the range to the target. Any Doppler shift on the pulse is a measure of motion or velocity. Pulsed radars, by their nature, make very brief observations of a target. As a result, they are very good at determining range, but may be limited in their ability to measure rapid or detailed spectral components of complex motion. Detailed studies of the spectrally complex and often rapidly changing E-region require good temporal resolution abilities. Though pulsed radars work well for E-region studies, the temporal resolution can be limiting when detailed spectral and temporal resolution would help resolve the physics of some of the observations. Examples of pulsed E-region coherent scatter radars include CUPRI [*Providakes et al.*, 1988] and SuperDARN [*Greenwald et al.*, 1995] (although SuperDARN is primarily focused on F-region studies).

CW radars transmit continuous signals, and, as such, simultaneously transmit and receive. CW techniques have the benefit of excellent temporal resolution because the data can be analysed in arbitrarily small segments of time, limited only by the time needed to detect the reflected signals from the ionosphere and the desired frequency resolution. CW techniques do not use a time modulation, and, as such, it is challenging for CW systems to determine range.

CW systems are often bistatic (meaning the transmitter and receiver are in separate locations) in order to overcome problems with feedthough (see Section 3.2.3). Figure 1.2 depicts a bistatic radar system, illustrating the beam patterns of each antenna. Any signal detected by the radar is assumed to come from the region where the antenna patterns overlap. In this sense, some range information can be determined in a CW system; however, this produces poor range resolution relative to that of a pulsed system. It has the further complication of requiring separate radar sites which are often significant distances apart. Receiving scatter solely from this region of overlap also limits the area of coverage of this type of radar. Pulsed radar techniques tend to cover larger areas while simultaneously obtaining better range resolution.

Examples of VHF coherent backscatter CW radars include SESCAT [*Haldoupis*

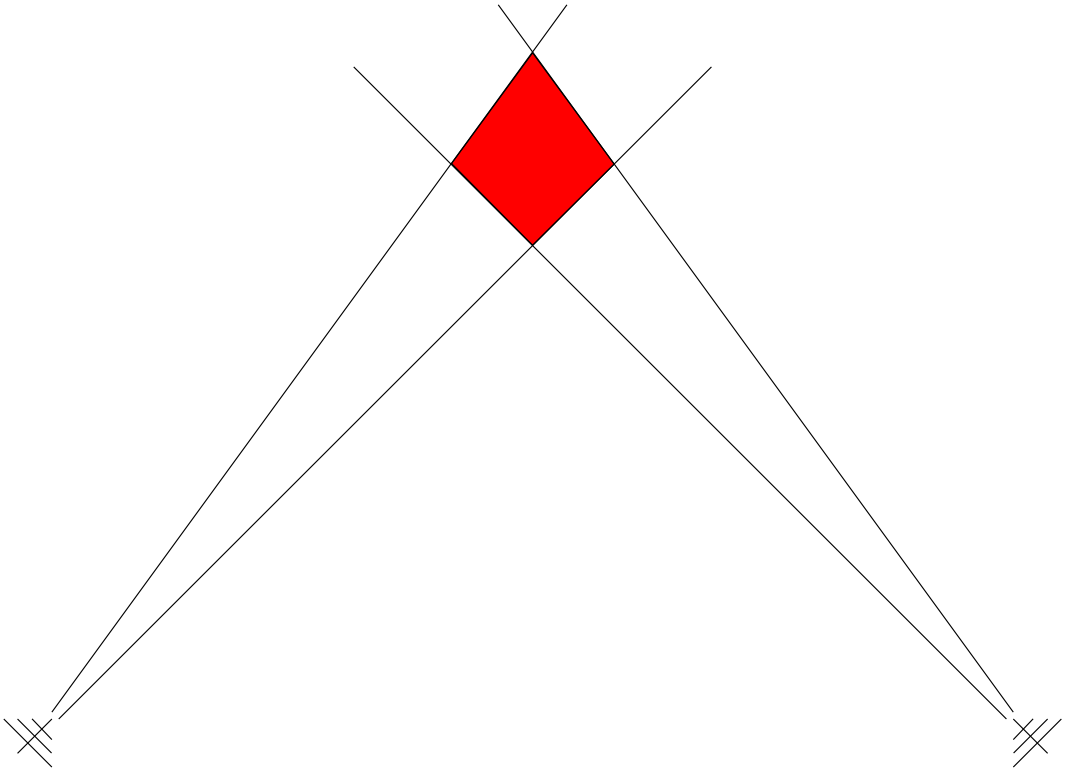

Figure 1.1: Diagram illustrating the antenna pattern for a bistatic CW radar system. Range is determined by the overlap of the transmitting and receiving antenna patterns. Any signal detected by the radar is assumed to originate from this region of overlap.

and Schlegel, 1993] and SAPPHIRE [*Ortlepp*, 1994; *Koehler et al.*, 1997]. The SAPPHIRE radar system was operated by the University of Saskatchewan and is of particular importance for this thesis since equipment from SAPPHIRE was used in the construction of the FMCW radar.

FMCW radar techniques are the focus of this thesis. An FMCW radar continuously transmits and receives a signal, much the same as the CW radar technique; however, with an FMCW radar, the frequency of the transmitted signal is varied as a function of time. This frequency modulation (FM) allows the radar to determine range. Using FMCW techniques for E-region studies offers a compromise between the strengths and weaknesses of the CW and pulsed radar techniques. FMCW radar techniques offer improved temporal and frequency resolution over pulsed techniques by using a continuous signal while simultaneously providing good range resolution. FMCW radar techniques are discussed extensively in Chapter 2.

In summary, CW systems have excellent temporal resolving abilities but have limited spatial resolving capabilities while pulsed systems have excellent spatial resolving capabilities but have limited frequency resolving abilities. Frequency modulated continuous wave (FMCW) systems vary the frequency of a CW signal as a function of time. This allows for range measurements while maintaining the high temporal and spectral benefits associated with a CW system.

1.3 Other FMCW radar

FMCW radar systems have served many functions. A common use of FMCW techniques can be found in altimeters in air-planes where the range of the target is rapidly and continuously changing. FMCW techniques are particularly useful for short-range measurements because they overcome the blind range associated with switching from transmit mode to receive mode in pulsed radar systems. Examples of FMCW radar uses include collision warning systems for cars where range and velocity information from targets within a few hundred meters need to be obtained quickly [*Rasshofer and Biebl*, 1998; *Rohling and Meinecke*, 2001]. A microwave FMCW radar system

designed and operated out of the University of Saskatchewan was used to perform ground based measurements of plant height and moisture in an agricultural setting [*Hussey*, 1989]. FMCW radar techniques have also been used to extract the Doppler spectrum as a function of range for falling raindrops [*Strauch et al.*, 1976].

A notable radar called WERA (WEllen RAdar) is an HF (High Frequency) FMCW radar that measures coherent backscatter from ocean waves [*Gurgel et al.*, 1999, 2000, 2001]. WERA uses sawtooth modulation and extracts Doppler spectra as a function of range using the method outlined in Section 2.1.2. Although it performs measurements that are similar in principle to the E-region measurements required for this instrument (that is, extracting Doppler and range information from coherent backscatter), the targets are very different. The Doppler shift of HF coherent backscatter from ocean waves is a few Hz whereas the VHF coherent backscatter from the E-region can have Doppler shifts as high as a few hundred Hz. When the range to the E-region is also considered, this method becomes impractical as the propagation time makes it very difficult to achieve a sampling rate high enough to resolve the large Doppler shifts expected from the E-region.

Another radar worth noting is the Manastash Ridge radar. It is a passive radar system, which means it uses the signals from commercial FM radio stations to study coherent backscatter from the E-region. The radio station acts as the transmitter. Two receivers are used for this system. The first receiver is placed beside the radio station to capture the transmitted signal. The second receiver receiver is placed on the other side of a mountain to capture signals that have reflected from the E-region. The mountain acts as a natural barrier to minimise direct feed-through from the transmitter. Radio stations produce excellent signals for use in radar because they are essentially unique over time. A correlation of the received signal with the transmitted signal in both time and frequency requires significant data processing, but produces detailed measurements of the range and Doppler spectra of E-region phenomena [*Lind*, 1999].

1.4 E-region signals

Signals scattered from plasma waves in the ionosphere have characteristic echoes that are classified based on their Doppler spectrum and mean Doppler shift. These echoes are placed into four categories labelled Type I through Type IV. Along with the signals that originate from coherent backscatter from plasma waves, radars examining the E-region will also observe meteor trails. Each of the Type I to Type IV and meteor trail echoes are discussed below.

1.4.1 Meteor trail echoes

The Earth is continuously bombarded by meteoroids. Meteoroids are vapourised due to friction between the meteoroid and the constituents of the atmosphere. As the meteoroid vaporises it becomes incandescent. While the meteoroid is vapourising, a streak of light is formed in the sky known as a meteor.

During the process by which a meteoroid is vaporised, an ionisation path is formed that is referred to as a meteor trail. The meteor trail presents radio waves with a change in the refractive index of the atmosphere causing them to be reflected. The type of reflection depends upon the wavelength of the radio wave and the ionisation density of the meteor trail.

A meteor trail generally produces a Doppler spectrum with a width of only a few Hz for 50 MHz radar signals. They form in the D- and lower E-regions, and, as such, have a mean Doppler shift proportional to the neutral wind speed of the D-region [*Reid*, 1983]. Meteor trails can be distinguished from auroral signals because their Doppler spectra are narrower and their mean Doppler shifts tend to be much lower. The neutral winds in the D- and lower E-region are usually much less than 100 m/s [*Prikryl et al.*, 1986].

1.4.2 Type I echoes

Type I echoes are characterised by a narrow spectrum with a mean phase velocity at the ion acoustic speed. The ion acoustic speed in the E-region is approximately 360 m/s. For 50 MHz radar signals the mean Doppler shift is ∼120 Hz. Type I echoes are caused by plasma waves generated by the Farley-Buneman or two-stream instability. A two stream instability is excited when the relative drift velocity of the ions and electrons exceeds the ion-acoustic speed. As discussed in Section 1.1, in the E-region, electrons are magnetised and ions are dominated by collisions with the neutral atmosphere. As a result, the electron velocity is driven by $\vec{E} \times \vec{B}$ drift and the ion velocity is driven by the neutral winds. When the relative velocity of the ion-electron drift reaches the ion acoustic speed, perturbations in the electron density are unstable and will grow. As a result, instabilities form that can be detected by coherent scatter radar [*Kelley*, 1989; *Kivelson and Russell*, 1997].

1.4.3 Type II echoes

Type II echoes are characterised by a broad Doppler spectrum and a low mean Doppler shift. They are thought to be caused by gradient drift instabilities. When the relative velocity is less than the ion-acoustic speed, instabilities can still form if there is a density gradient in the right direction relative to the electric field. In order for a density gradient to be destabilising, the electric field must have a component parallel to the density gradient. In this case, instabilities can form with relative electron speeds that are well below the ion-acoustic speed [*Kelley*, 1989].

1.4.4 Type III echoes

Type III echoes have a narrow spectrum and are typically strong signals. The mean Doppler shift of Type III echoes is below the ion acoustic speed and generally between 150 and 220 m/s. The cause of these echoes is not well understood.

1.4.5 Type IV echoes

Type IV echoes are less common than the other three types of echoes. Typically they have a narrow Doppler spectrum with a large mean phase velocity. Type IV echoes often have a mean phase velocity that is two to three times the normal ion acoustic speed of ~ 360 m/s. *Fejer et al.* [1986] suggested that Type IV events may be caused by the modified two-stream instability, through the same process as Type I echoes. The higher phase velocities are explained by an increased ion acoustic speed caused by higher electron temperatures [*Kelley*, 1989].

1.5 Thesis layout

This thesis describes the design, construction, and operation of an FMCW radar. The radar was constructed using a combination of new and old components. Every step from initial design, to construction, and finally to the collection of actual data and data analysis was performed during the thesis work. The purpose of the thesis is to demonstrate that FMCW techniques provide a valuable tool in the study of E-region coherent backscatter. Emphasis is placed on the design, implementation, and challenges in data analysis that are associated with the use of an FMCW radar for E-region studies.

Chapter 2 gives a theoretical background for understanding FMCW techniques. Different frequency modulation schemes are examined, and the justification for the modulation scheme used in this experiment is given. Chapter 2 also describes the process of separating range and Doppler information with FMCW techniques. The chapter also provides the theoretical basis for the data analysis technique used in this project.

Chapter 3 describes the parameters for the radar and the design of the equipment. The requirements for the radar are outlined and the parameters needed to accomplish those requirements are discussed. The physical design of each component of the radar is then discussed including the transmitter, receiver, software, and data storage.

Chapter 4 provides a detailed description of the data analysis techniques applied to a real example. Overcoming noise and separating range and Doppler information posed significant challenges. Techniques used to process and interpret the data collected by the radar are described in detail.

Chapter 5 presents the results obtained from the operation of the radar. Different events are presented in order to verify that the radar operates and collects data as is described in the other chapters. Each type of E-region radar echo, including meteor trails, is examined. A discussion of some of the challenges specific to FMCW radar is presented and potential solutions are offered, including the implementation and presentation of data from a more complex frequency modulation technique.

Chapter 6 gives a summary of the project, and provides suggestions for future development. As this is the first step in the implementation of a novel instrument for E-region studies, emphasis is placed on suggestions for future work.

Chapter 2

Frequency modulated continuous wave radar

As mentioned in the introduction, FMCW radar has a number of uses, but the use of FMCW radar techniques for E-region coherent backscatter is a novel concept. E-region coherent backscatter poses a number of challenges to an FMCW system. The received signals can have Doppler shifts of up to 500 Hz with a broad Doppler spectrum and shape. Extracting this Doppler information along with good range information is not trivial for any radar. Through a combination of waveform selection and data analysis, FMCW techniques are able to obtain very good measurements of E-region coherent backscatter.

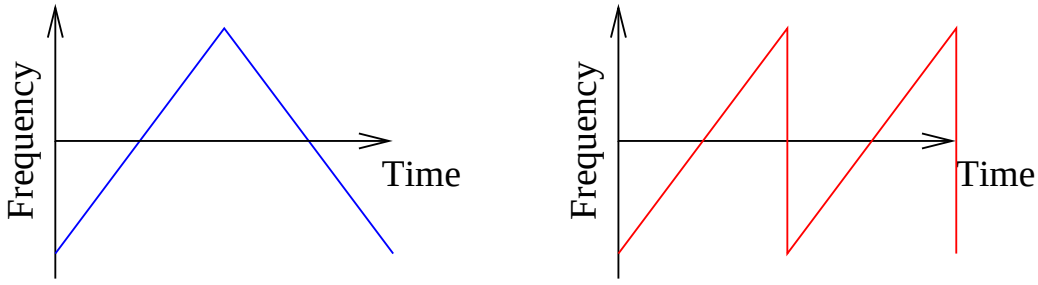

Figure 2.1: Triangular (left) and sawtooth (right) frequency modulation schemes.

2.1 Introduction to FMCW radar

In an FMCW radar system, the frequency of the transmitted signal is modulated as a function of time and, as such, the frequency of the received signal can be used as a measure of propagation time. As a result, range information is measured in the frequency domain along with the Doppler information. This can pose a challenge when determining both range and Doppler information.

Figure 2.1 depicts two common forms of frequency modulation. Both plots have frequency on the vertical axis and time on the horizontal axis. For the triangular frequency modulation scheme, pictured on the left, the frequency of the signal is swept up and then swept down to the original frequency. The sawtooth modulation scheme, on the right, is an example of a modulation in which the frequency is swept up and then immediately returns to the original frequency where the up sweep is repeated. Sawtooth modulation is useful when dealing with stationary or slow moving low Doppler shift targets relative to the modulation frequency. Triangular modulation is better suited for measurements of faster moving targets, because the sign change in the frequency sweep enables the extraction of targets with significant Doppler shifts.

Figure 2.2 demonstrates the basic set up of an FMCW system. All FMCW systems require a signal generator to produce the transmitted signal, as well as a signal processor to interpret the received signal. Most FMCW radar systems mix the transmitted signal directly with the received signal. In this way the receiver for an FMCW system measures the difference in frequency between the transmitted and received signals. The difference in frequency between the transmitted signal and the signal received from a scattering target is referred to as the beat frequency, f_b. The beat frequency is directly related to the range and velocity of the target [*Skolnik*, 1980, 1990; *Eaves and Reedy*, 1987].

The following sections give an introduction to both triangular and sawtooth modulation schemes. As triangular modulation is used for this radar, it will be explored in greater detail. A detailed discussion of the post-mixing signal for various FMCW modulation waveforms can be found in *Tozzi* [1972].

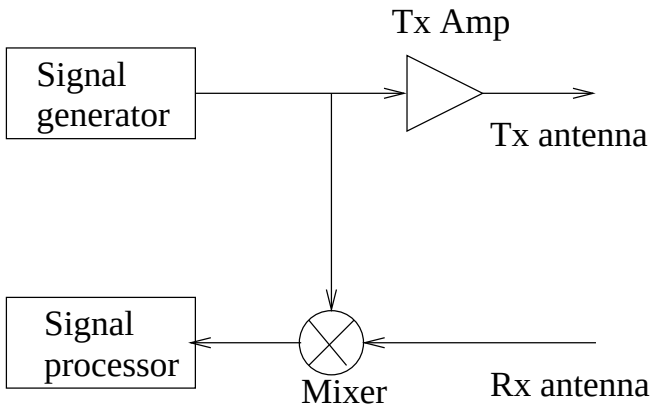

Figure 2.2: Basic block diagram for an FMCW radar system.

2.1.1 Triangular modulation

Triangular modulation techniques use a modulation scheme with a period which is comprised of two intervals. The frequency of the transmitted signal increases during the first interval and decreases during the second interval. As FMCW radar systems measure both range and Doppler information in the frequency domain, the positive and negative slope portions of the triangular frequency modulation scheme can be used to separate this information.

Figure 2.3(a) shows the frequency of the transmitted, Tx, and received, Rx, signals as a function of time for triangular frequency modulation. The transmitted signal (shown in red) can be characterized by three basic parameters: bandwidth B, period of modulation T_m, and a reference frequency value f_0. The bandwidth refers to the width or amplitude of the frequency sweep (defined as the highest frequency minus the lowest frequency), and the period of modulation refers to the total time it takes to complete a full up sweep and down sweep modulation cycle. In this diagram the median value of the frequency sweep is used as the reference point. The received signal (shown in blue) is a time delayed version of the transmitted signal with a Doppler shift caused by the motion of the target. The following analysis is based on *Skolnik* [1980, 1990] and *Eaves and Reedy* [1987] and assumes scatter off of an ideal point target. This discussion is intended as an introduction to FMCW triangular

modulation techniques. Analysis of more complex signals relevant to this project are discussed in Section 2.2.

Figure 2.3(a) also illustrates the effect of a Doppler shift on the received signal. The Doppler shift from motion towards the radar increases the frequency of the Rx signal. An increase in frequency of the Rx signal causes the difference in frequency between the Tx and Rx signals to be lower during the up sweep than the down sweep. For motion away from the radar the Doppler shift causes the difference in frequency between the Tx and Rx signals to be higher during the up sweep than the down sweep. This difference in frequency, f_b, is illustrated in Figure 2.3(b). The receiver of an FMCW radar measures the difference in frequency between the transmitted and received signal.

By applying basic geometry to Figure 2.3(a), f_b between the transmitted and received signals can be related to range and Doppler shift. Before the equations of the range and Doppler shift of a target can be derived, some basic equations need to be defined. The time delay, T, between signal transmission and reception is labelled on Figure 2.3(a). T is related to the range of the target, R, by

$$T = \frac{2R}{c} \tag{2.1}$$

(assuming that the transmit and receive antenna are in the same location) where c is the velocity of an electromagnetic wave in a vacuum. The velocity, v, of the target produces a Doppler shift of

$$f_d = 2v/\lambda \tag{2.2}$$

where λ is the wavelength of the transmitted radar wave.

During an up sweep the rate of change in the frequency of the transmitted signal with respect to time is given by

$$\frac{\Delta f}{\Delta t} = \frac{2B}{T_m}. \tag{2.3}$$

Therefore, the amount that the transmitted frequency will change, Δf, during the time delay between transmission and reception, $\Delta t = T$, is given by

$$\Delta f = \frac{2B}{T_m} T. \tag{2.4}$$

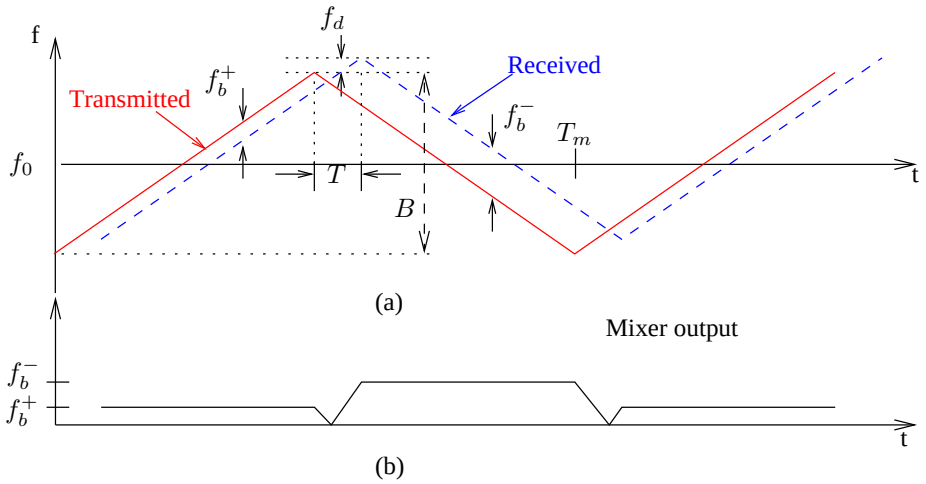

Figure 2.3: Diagram illustrating (a) Linear triangular frequency modulation and (b) beat frequency, f_b, between transmitted and received signals. See text for details.

The Doppler shift acts to move the overall frequency of the received signal up (for motion towards the radar) or down (for motion away from the radar). The equation for the beat frequency during an up sweep, f_b^+, written as a function of range and Doppler shift, is derived by substituting Equation 2.1 into Equation 2.4 and subtracting the Doppler shift, f_d

$$f_b^+ = \frac{4B}{cT_m}R - f_d. \tag{2.5}$$

The first term on the right in Equation 2.5 represents the difference in frequency between the transmitted and received signal due to the range of the target. The second term on the right represents the Doppler shift associated with the velocity of the target.

Using the same analysis, it can be shown that the portion of the sweep with a negative frequency slope has a beat frequency, f_b^-, given by

$$f_b^- = \frac{4B}{cT_m}R + f_d. \tag{2.6}$$

If f_b^+ is considered without reference to f_b^-, there is significant ambiguity between the Doppler and range information. Ignoring practical considerations, a given beat

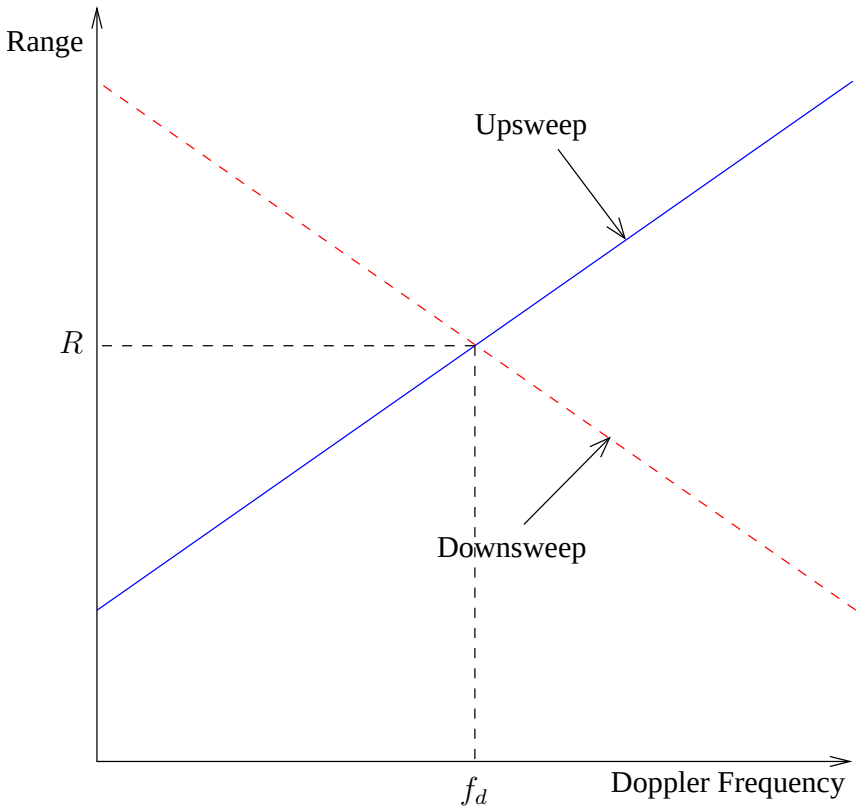

Figure 2.4: A graphical representation of the range and Doppler frequency possibilities for a point object observed during the up sweep (represented by a solid blue line) and the down sweep (represented by a dashed red line). Each line represents all possible combinations of range and Doppler frequency whose contribution to the beat frequency would add up to the measured beat frequency as described in Equations 2.5 and 2.6.

frequency could be produced by a Doppler shift from an object at zero range, $R = 0$, such that

$$f_b^+ = \frac{4B}{cT_m}(0) - f_d = -f_d, \tag{2.7}$$

or it could be from an object with zero Doppler shift, $f_d = 0$, at some range

$$f_b^+ = \frac{4B}{cT_m}R - 0 = \frac{4B}{cT_m}R, \tag{2.8}$$

or any combination of range and Doppler frequencies such that Equation 2.6 is satisfied. Figure 2.4 is a graphical representation of the range and Doppler frequency possibilities for a point object observed during the up sweep (represented by a solid blue line) and the down sweep (represented by a dashed red line). Each line represents all possible combinations of range and Doppler frequency whose contribution to the beat frequency would add up to the measured beat frequency as described in Equations 2.5 and 2.6. On its own, each sweep limits the possible range and Doppler components to a straight line. Due to the positive and negative slopes associated with f_b^+ and f_b^-, the lines intersect at a unique point that gives the true range and Doppler shift of the target.

Mathematically, the range term can be extracted by averaging f_b^+ and f_b^-. The result of this average can be solved to give the line of sight range to the target

$$R = \frac{cT_m}{4B}\frac{f_b^+ + f_b^-}{2}. \tag{2.9}$$

The Doppler shift from the motion of the target is half the frequency difference between f_b^+ and f_b^-. Therefore, the velocity of the target may be expressed as

$$v = \frac{\lambda}{2}\frac{f_b^- - f_b^+}{2}. \tag{2.10}$$

For multiple targets or targets without a clearly defined range and velocity, the analysis is more complicated (see Section 2.2).

Finally, note that the beat frequency becomes ambiguous during the peak and trough portions of the frequency modulation. Information will be lost during the period, T, from the time that the transmitted frequency changes slope to the time that the received signal changes slope. This is illustrated in Figure 2.3(b) by the dips in the beat frequency.

In order to resolve the Doppler shift using the triangular modulation technique, the Doppler shift must be larger than the frequency of the modulation. The signal received is Fourier transformed in intervals that are proportional to the modulation period. Based on sampling theory, the frequency resolution of a Fourier transform is inversely proportional to the sampling interval [*Brigham*, 1988]. Therefore, in order for an FMCW radar to resolve a Doppler shift with a Fourier transform, the Doppler shift must be larger than the twice the frequency of the modulation. For example, if the period of the modulation for a triangular modulation scheme is 0.5 s, then the sampling interval is 0.25 s, and the frequency resolution of a Fourier transform will be 4.0 Hz. If the target does not produce a Doppler shift larger than 4.0 Hz, the Doppler shift information will not be resolved from the range information. Therefore, the modulation frequency of the radar determines the minimum Doppler shift that can be obtained using triangular frequency modulation techniques.

If a triangular modulation scheme is used to measure a stationary or low Doppler shift (low relative to the modulation frequency) target, the up sweep and down sweep will produce the same beat frequency within the resolution of the system. In this case, a single up sweep (or down sweep) provides the range information and the down sweep (or up sweep) provides no further information. For targets with Doppler shifts that are always less than the modulation frequency, a sawtooth modulation technique can be used as described in the next section.

2.1.2 Sawtooth modulation

Doppler information can be extracted from low Doppler shifted targets using sawtooth modulation as demonstrated by WERA [*Gurgel et al.*, 1999, 2000, 2001] (see Section 1.3). Sawtooth modulation extracts Doppler information as a function of range using a process similar to that common in coherent pulsed radar systems. Each sample collected by the receiver of a pulsed system corresponds to a range cell since the sample corresponds to a specific time after a pulse was transmitted. For a given range, samples from subsequent pulses can be combined to produce a time

sequence for that range cell. After multiple pulses, a series of time samples for each range cell has been collected. A fast Fourier transform (FFT) of those samples may then be applied to produce a Doppler spectrum for each range.

A similar analysis technique can be applied to an FMCW system using the sawtooth modulation scheme to obtain Doppler information. The received signal is sampled for a given frequency sweep. An FFT on the data collected during the sampling interval resolves targets as a function of range. Each discrete frequency in the FFT corresponds to a range. The complex value of the FFT represents a phase sample for that range cell. Multiple frequency sweeps produce a series of phase samples for each range cell. A subsequent FFT analysis of the phase samples of each range cell extracts the Doppler spectrum as a function of range [*Gurgel et al.*, 1999, 2000, 2001].

Using this technique, samples are taken over many modulation periods, the sampling interval for the second Fourier transform is large compared to the modulation period. Therefore, the frequency resolution becomes small relative to the modulation frequency, and the Nyquist frequency is the modulation frequency. Using sawtooth modulation, detailed spectra can be obtained from targets that produce a low Doppler shift.

2.2 Soft targets

The signal analysis for the FMCW VHF E-region coherent radar system presented in this thesis is not as basic as was described in Section 2.1. E-region targets do not have a well defined range and phase velocity. As such, it becomes more challenging to resolve the received signal into its range and Doppler components.

In this discussion, a target is defined as a *hard target* if it has a well defined range and Doppler spectrum. For the purpose of clarification, well defined means that the physical size of the scattering volume for a target is small relative to the range resolution of the radar, and that the Doppler spectrum of the target is narrow relative to the frequency resolution. An example of a hard target would be any solid object

like a sheet of metal or an airplane. If these conditions are met, the scatter from such a target will produce a well defined beat frequency and the resolution of that beat frequency into range and Doppler components will be relatively unambiguous. Hard targets observed by triangular modulation can be analyzed as discussed in Section 2.1.1.

A *soft target* is defined as a target that extends over several range cells, or a target with a Doppler spectrum that extends over several frequency cells. A soft target does not have a well defined range and Doppler shift, and, as such, this information can be challenging to extract.

Most E-region coherent backscatter are from soft targets. The processes that cause E-region coherent backscatter typically extend over a few tens of kilometres and have Doppler spectra extending over a few hundred Hz [*Haldoupis et al.*, 2003]. Furthermore, multiple scattering targets can be present in the radar field-of-view at any given time. The large line of sight range to the E-region and the large expected Doppler shifts prevent use of the sawtooth modulation technique discussed in Section 2.1.2. Therefore, triangular modulation was chosen for this radar. The following discussion will explain how information about soft targets can be extracted using a technique which is similar to the method discussed in Section 2.1.1 (note that hard targets are also extracted using this technique).

Because both range and Doppler information are measured in the frequency domain, f_b (as defined in Figure 2.3 of Section 2.1) has a component associated with range and a component associated with the Doppler spectrum. The frequency associated with range refers to the component of the beat frequency that a stationary object would produce at the range of the target. The received signal is a time delayed copy of the transmitted signal, and the frequency of the transmitted signal varies linearly with time. Therefore, the received signal is a different frequency from the signal that is currently being transmitted simply because of the time delay associated with the range to the scattering target (see Figure 2.3 and Equations 2.5 and 2.6). The Doppler spectrum is composed of the Doppler frequencies from the motion of the target and is added onto the range frequency component. Soft targets will

produce a spectral shape or spectrum that extends over several frequency cells. The spectral shape is characteristic of the phenomena that produces it.

Before proceeding with a discussion of how range and Doppler information can be separated for soft targets, a few terms will be defined. The up sweep and down sweep portions of the triangular modulation scheme will be analysed separately and, therefore, are defined separately. The output of the mixer during a down sweep will be defined as $x(t)$, and the output of the mixer during an up sweep will be defined as $y(t)$, with Fourier transforms denoted as $X(f)$ and $Y(f)$ respectively. These signals contain the beat frequencies, f_b^- and f_b^+, that represent the range and Doppler information of the targets.

The signals $x(t)$ and $y(t)$ will be sampled at a regular time interval, T_s, and the sampled waveforms will be defined as $x(k)$ and $y(k)$ respectively, where k is an integer number representing the sample taken at time $t_k = kT_s$. After a set of data are taken over a time interval $(N-1)T_s$, where N is the number of samples, the sampled signal will be transformed to the frequency domain by a fast Fourier transform (FFT) giving $X(n)$ and $Y(n)$ where n is an integer representing the frequency $f = \frac{n}{(N-1)T_s}$.

Signal analysis must identify the beat frequencies in $X(n)$ and $Y(n)$ and then separate the frequency associated with range from the frequency associated with the Doppler spectrum for each beat frequency. For triangular frequency modulation, Equations 2.5 and 2.6 predict that any Doppler spectrum contained in $X(n)$ and $Y(n)$ should be 'reflected' about the frequency cell that corresponds to the mean range. For example, if a target produces a positive Doppler shift (motion towards the radar), the beat frequency in $X(n)$ should be higher than the frequency corresponding to the range and in $Y(n)$ the beat frequency will be lower than the frequency corresponding to the range by the same amount.

The use of triangular modulation separates the Doppler and range information, but the process of extracting this information is still required. In order to extract Doppler and range information using triangular modulation, the Doppler spectrum produced by a target in one frequency sweep (i.e., the up sweep) will have to be matched with the corresponding spectrum from the other frequency sweep (i.e., the

down sweep). If the Doppler spectrum of a target is distinct from the noise and it produces a similar spectrum in both $X(n)$ and $Y(n)$, a correlation technique in the frequency domain will be able to make this match.

The discrete correlation of $X(n)$ and $Y(n)$ is defined as

$$W(n) = \sum_{\phi=0}^{N-1} X(\phi)Y(n+\phi) \tag{2.11}$$

where ϕ is the discrete frequency number. Since, as described above, triangular modulation will reverse the Doppler spectrum of the target between $X(n)$ and $Y(n)$, one of the functions should be reversed in the frequency domain for the correlation. As the Doppler shift from any scattering region will be negative for the beat frequency in the up sweep (see Equation 2.5), it is logical to reverse $Y(n)$. The reversal makes the shape of the Doppler spectrum the same in both the up sweep and the down sweep:

$$W(n) = \sum_{\phi=0}^{N-1} X(\phi)Y(n-\phi) = X(n) * Y(n). \tag{2.12}$$

Equation 2.12 is convolution in the frequency domain. A convolution of $X(n)$ with $Y(n)$ will perform the function of correlation for this radar as a convolution reverses $Y(n)$. Each step in the convolution corresponds to a range. Therefore, convolution of $X(n)$ and $Y(n)$ can be used to identify the range of the target. The width of the distribution in the convolution is proportional to the range extent of the target and the width of the Doppler shift.

Chapter 3

The FMCW Radar Design

The goal of this project was to design and construct an FMCW radar system. Several factors made this goal feasible, with the most notable being the existence of SAPPHIRE. SAPPHIRE was a radar previously operated by researchers at the University of Saskatchewan but it ceased operations in 1997. It was a bistatic 50 MHz CW radar system with a receiver site located 20 km outside of Saskatoon. The receive site for SAPPHIRE was used as the transmit and receive site for the new FMCW radar. The radar site was equipped with 12 antennas, a shed to house radar equipment, and cables between the shed and the antennas. Furthermore, components from SAPPHIRE transmitters, including amplifiers, power supplies, and RF splitters, were available and in working condition. All SAPPHIRE equipment was designed for CW use at 50 MHz. The availability of this equipment along with technical manuals and experienced individuals all contributed to make this project feasible.

The waveform design and data collection parameters determined the range, Doppler frequency and time resolution. Parameters such as bandwidth and sampling time were selected to capture E-region coherent backscatter in a meaningful way. Compromises needed to be considered between the various design parameters.

The hardware equipment for the FMCW radar was comprised of a combination of SAPPHIRE components and new components. The design and initial construction took place on a bench at the University of Saskatchewan. After the equipment had been tested and was working properly, it was installed at the radar site. Further refinement of the equipment took place once the system was operated in the field, but the basic design remained the same.

Radar control and operation was accomplished through software on a PC computer running Linux. The computer needed to interact with the radar equipment on the transmitter portion of the system while concurrently receiving data from the receiver portion.

The following sections will discuss details of design and construction of the FMCW radar system.

3.1 Design parameters

Table 3.1 gives a list of the parameters used in the E-region coherent backscatter FMCW radar. The parameters in Table 3.1 are discussed below.

Bandwidth

The frequencies that the radar operated between were determined by the transmitter licence. The licence allowed a maximum bandwidth of 160 kHz. As with most radars, the bandwidth and the range resolution are inseparable [*Skolnik*, 1980]. A bandwidth of 160 kHz gives a theoretical range resolution of $\Delta R = c/2B$ or approximately 937 m. However, since information is lost during the interval between the change of direction in the frequency sweep and the time that the scattered signal reaches the receiver (see Section 2.1), the actual theoretical range resolution will be less. The effective bandwidth of the signal becomes 154.3 kHz giving an effective theoretical range resolution of approximately 970 m (see the Section 3.1.1 for the determination of effective bandwidth). Furthermore, due to the ambiguity between Doppler and range information, the practical range resolution will be less than that calculated in theory (see Section 2.2). Range and Doppler frequency resolution is discussed in more detail below.

Digital sampling parameters

The parameters used to digitise the received signal determine the limitations of the information extracted from the signal. A number of factors were important in de-

Table 3.1: Table of parameters for the FMCW radar

FMCW radar parameters		
Parameter	Value	Units
Transmit Frequency	49.420-49.580	MHz
Transmit Wavelength	6.0466-6.0662	m
Bandwidth, B	160.0	KHz
Sampling rate	9600	Hz
Samples per FFT	2048	Samples
Sampling time	213	ms
Sampling resolution	24	bits
Frequency resolution	4.69	Hz
Range resolution	970	m
Maximum range	993	km
Velocity resolution	14.1	m/s
DDS parameters		
Frequency step	652	mHz
Step time	900	ns

termining these limitations. Previous CW radar implementations by researchers at the University of Saskatchewan have had approximately 5 Hz frequency resolution [*Hussey*, 1994]. A similar frequency resolution was achieved with the FMCW radar implementation.

A separate Fourier transform is performed for the up sweep and down sweep. The frequency resolution of a Fourier transform is the inverse of the sampling time, therefore, the period of the triangular frequency modulation, T_m, should be 0.4 s to achieve 5 Hz frequency resolution. However, only select sampling rates were available in the receiver system. Efficient FFT algorithms require that the input number of sample points, N, be a power of two [*Brigham*, 1988]. A compromise between the constraints of discrete sampling rates in the sound card (see Section 3.2.5 for details) and a reasonable power of two sample size was found with a sampling rate of 9600 Hz and 2048 samples per FFT. This gives a Nyquist frequency of 4800 Hz, a sampling time of 0.213 s for the 2048 samples, and a frequency resolution of 4.69 Hz.

Velocity, range, and time

Equation 2.9 demonstrates that the range resolution of the radar will vary directly with the frequency resolution; this relationship is expressed in Equation 3.1.

$$\Delta R = \frac{cT_m}{4B}\Delta f \tag{3.1}$$

The frequency resolution of 4.69 Hz gives a theoretical range resolution of ∼970 m. This range resolution accounts for the fact that some information is lost when the frequency sweep changes sign. Similarly, Equation 2.10 can be used to show that the velocity resolution will be given by

$$\Delta v = \frac{\lambda}{2}\Delta f. \tag{3.2}$$

For a radar signal wavelength of ∼6 m (see Table 3.1), this frequency resolution translates to a velocity resolution of 14.1 m/s.

The velocity resolution is limited by the sampling period due to the nature of the discrete Fourier transform. In order to improve velocity resolution, the sampling

period (and, therefore, the modulation period) must be increased. In order to improve range resolution, the bandwidth of the frequency sweep must be increased. The bandwidth can be arbitrarily increased to improve range resolution and is limited only by practical resolution concerns and the transmitter licence. The linearly varying frequency of an FMCW radar is difficult to detect by demodulators without a similarly varying reference frequency. As such, the bandwidth of the signal could likely be increased indefinitely with little concern for interference with other sources. However, if the bandwidth was increased sufficiently, the wavelength of the signal would change too much during the frequency sweep, and the coherent backscatter at the extreme frequencies might come from different phenomena.

The range and velocity resolution can be changed independently without effects on the other parameter. The bandwidth of the radar can be increased to improve range resolution without loss of velocity resolution. Conversely, the velocity resolution can be improved without loss of range resolution. A trade off does exist between velocity resolution and time resolution. Arbitrarily small frequency cells (and, subsequently, arbitrarily narrow velocity resolution) can be achieved by increasing the sampling time, but this reduces the time resolution of the measurements.

An FFT of 2048 samples generates 1024 frequency bins which corresponds to 1024 range cells (Note that the FFT gives 2048 frequency bins, but only 1024 are meaningful after data analysis. See Chapter 4 for more information on the use of the FFT in data analysis). This gives a theoretical maximum line of sight range of 993 km ($1024 * 970$ m). The theoretical maximum range is misleading because both range and Doppler shift signatures are measured in the frequency domain. A scattering region at 993 km without any Doppler shift would produce a beat frequency of 4800 Hz in the receiver. However, any Doppler shift on the target would extend the beat frequency beyond the Nyquist frequency and would make range and Doppler measurements inaccurate. Therefore, the maximum range would be dependent upon the Doppler shift of the target and would have a maximum range of 993 km for zero Doppler shift scattering regions. Alternately, the maximum theoretical Doppler frequency that could be captured by the radar is dependent upon

the range of the target and would have a maximum value of 2400 Hz at a range of 492 km. Therefore, the maximum range and Doppler frequency limitations are such that the beat frequency, made up of frequency components associated with range and Doppler information, does not exceed 4800 Hz.

The maximum range and Doppler frequency limitations given above are adequate for this radar design. The antenna array was designed to have a peak in the vertical antenna pattern at an E-region altitude of $\sim$ 110 km and a line of sight range of about 700 km [*Hussey*, 1994]. This corresponds to an elevation angle of $\sim 6^{\circ}$ for the antenna arrays as illustrated in Figure 3.1. The maximum expected phase velocity for E-region coherent backscatter is approximately 1000 m/s. At 50 MHz, this corresponds to a frequency shift of 333 Hz, therefore allowing for up to +/-500 Hz is more than adequate to capture all expected E-region Doppler shifts. The maximum range for a target with a Doppler shift of 500 Hz would be 890 km; therefore, all E-region backscatter up to a line of sight range of 890 km should be accurately measured. Targets with less Doppler shift can be resolved at ranges greater than 900 km.

3.1.1 Baseband frequency

The baseband signal is the output of the mixing stage in the receiver. The signal at the post mixing stage represents the multiplication of the received signal with the transmitted (local oscillator) signal with the higher order frequency term filtered out. When the radar was implemented, the majority of signals observed came from between 500 and 900 km line of sight range with Doppler shifts of less than 150 Hz. Therefore, the expectations outlined in the previous section were realistic and reasonable. Given the range and Doppler measurement constraints, the values from Table 3.1, and Equations 2.6 and 2.5, the bandwidth of baseband frequencies expected to contain E-region signals were determined. The minimum baseband frequency of a 150 Hz Doppler shifted target from a range of 500 km would be $\sim$ 2260 Hz. The baseband frequency of a similar 150 Hz Doppler shifted target at 900 km would be

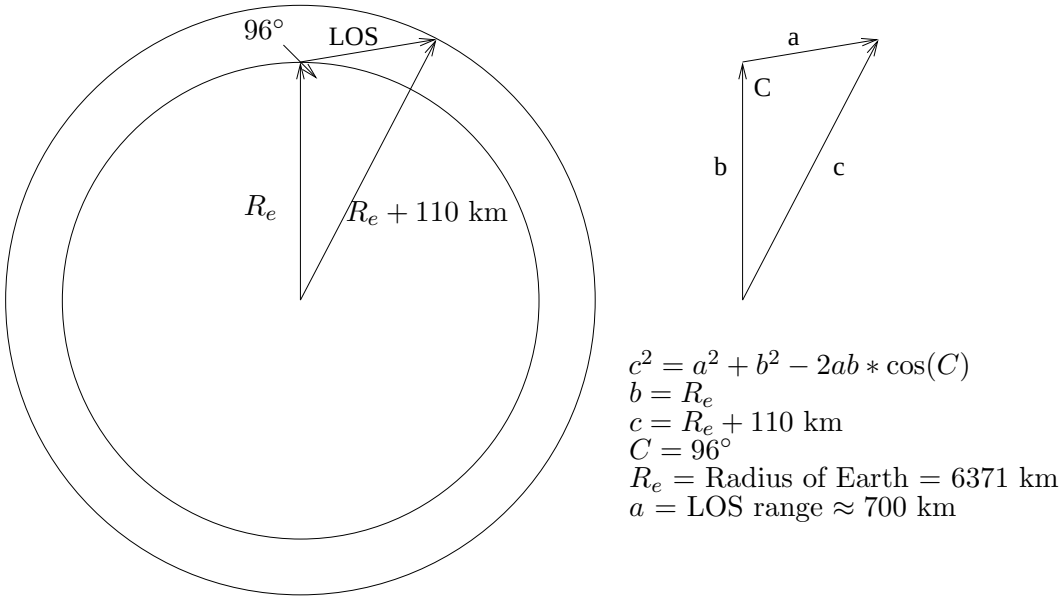

Figure 3.1: Illustration of the calculation of line of sight (LOS) range. A radar signal transmitted at an angle of 6° from horizontal (the elevation angle) will travel ∼700 km before it reaches 110 km altitude.

∼ 4194 Hz. The baseband range of interest, referred to as the *practical bandwidth*, for this study was defined to be between 2000 and 4800 Hz. This bandwidth ensured that targets at the outer boundary of the expected range and Doppler shift were still captured.

DDS parameters

The transmitted signal, including the frequency modulation, is controlled by a direct digital synthesis (DDS) board. The DDS is equipped with linear frequency sweeping capabilities required for the frequency modulation of this radar. The bandwidth of the frequency sweep is defined by an upper and lower frequency. The rate at which the DDS sweeps between these two end frequencies is specified by a frequency step and a timer. The frequency generated by the DDS sweeps between the end frequencies in increments of the frequency step. A step is taken at the completion of each cycle of the timer. For more information on the DDS see Section 3.2.4.

Unless otherwise stated, the data were collected using a frequency step of 652 mHz

and it was stepped every 900 ns. This means that the rate of frequency sweep was 724.4 kHz/s, giving a 160 kHz sweep in 0.221 s. The data were only sampled for 0.213 s, therefore, 8 ms of the sweep was not used in the data collection. 6 ms of this difference between sweep time and sampling time comes from a deliberate delay between a change in direction of the frequency sweep and the start of data collection. A 6 ms time delay corresponds to the time it would take for the transmitted signal to scatter off of a stationary target at 900 km and return to the receiver. The data collected during this time would be ambiguous (see Section 2.1). The remainder of the difference between sweep time and sample time provided a timing buffer for the control of the sweep by the operating system of the computer which had a timing error of up to a millisecond (see Section 3.3).

Note that the rate of the frequency sweep of 724.4 kHz/s and a data collection time of 0.213 s gives the effective bandwidth of 154.3 kHz discussed above. This effective bandwidth gives the same theoretical range resolution ($\Delta R = c/2B$) of $\sim$970 m as discussed in Section 3.1.

3.2 Hardware

Figure 2.2 in Section 2.1 illustrates a basic block diagram for an FMCW system. Section 2.1 placed emphasis on the fact that a copy of the transmitted signal was mixed with the received signal which is important for understanding the data analysis procedure; however, the block diagram has other fundamental radar components such as the signal generator, signal processor, transmitter amplifier, and antennas. This section describes the implementation of these components in the E-region coherent backscatter FMCW radar.

Figure 3.2 represents the full block diagram of the E-region coherent backscatter FMCW radar. A comparison with Figure 2.2 reveals how each basic component was implemented. The signal generation was performed using direct digital synthesis (DDS) techniques. Amplification occurred through a series of SAPPHIRE amplifiers and a commercial Alpha 6 amplifier [*Alpha-Power*]. The receiver was composed of a

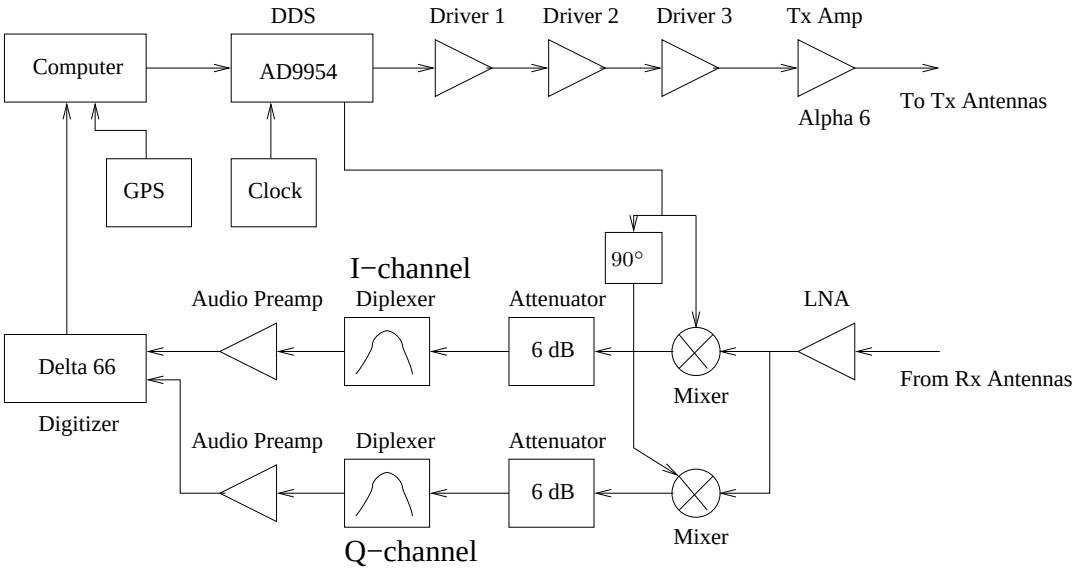

Figure 3.2: Block diagram of FMCW radar.

series of filters and amplifiers culminating in a sound card that digitised the signal for processing in the computer. Each component is discussed below.

3.2.1 Computer

Radar operation and data collection was controlled through software on a computer operating with Linux. The computer initialised the DDS and the sound card before radar operation began. It controlled the frequency modulation through the DDS and coordinated radar transmission with data collection. For more details about software and radar control, see Section 3.3.

The time on the computer was maintained with a Garmin global positioning system (GPS) [*Garmin*, 1999]. Network time protocol (NTP) (see `http://www.ntp.org/`) in Linux interpreted signals from the GPS and used those signals to keep the computer clock synchronised with coordinated universal time (UTC), the absolute time standard from which all time standards are derived, including universal time (UT). All data presented in the thesis are in UT (or UTC). Accurate time is necessary to ensure that data collected by the radar can be compared with data from other

instruments, and NTP will ensure sub-millisecond accuracy.

3.2.2 Antenna array

The antenna array used for this project was composed of horizontally polarised Cushcraft 617 6dB 6 meter (50 MHz) Yagi antennas. The 12 antenna array existed as a receiving antenna array for the SAPPHIRE radar system [*Ortlepp*, 1994; *Koehler et al.*, 1997]. Each antenna had its own cable to the equipment shed. The cables were impedance matched at 50 Ω and phase matched to within 1° of each other. The cables on both the receiver and transmitter portions of the radar had approximately 6 dB of loss.

3.2.3 Feed-through

Since an FMCW system transmits and receives simultaneously (see Chapter 2), there is a need to isolate the receiver from feed-through from the transmitter. Feed-through is the term used to describe power from the transmitter that goes directly into the receiver without reflecting off of a target. It is a problem because receiver equipment needs to be sensitive in order to detect weak signals relative to the strength of the transmitted signal. Therefore, the transmitted signal power can often harm, or at least overload, the sensitive receiver equipment. Feed-through is a particular problem for radar systems used in ionospheric studies where weak signals reflected from the ionosphere must be detected. Pulsed radar systems do not have to contend with feed-through because the receiver is disconnected from the antennas with a switch during the short transmission time. CW systems continuously transmit and receive so these systems must overcome feed-through in other ways. A common solution is to use a bistatic set up where the transmitter and receiver are at physically separated locations. A continuously transmitting system with the transmit and receive antennas co-located will have a powerful self-interference that poses a significant problem to the receiver.

Circulators are devices that allow a transmitter and receiver to share the same

antenna. A circulator causes RF power from the transmitter to be directed to the antenna while isolating the receiver. RF power coming from the antenna is directed to the receiver, while isolating the transmitter. If the transmitter and receiver for this radar were connected to the same antenna it would be necessary to use a circulator to isolate the receiver. Circulators with sufficient isolation at the expected power levels are expensive and difficult to manufacture. Therefore, separate receive and transmit antenna arrays were used instead of transmitting and receiving on the same antennas.

WERA (see Section 1.3), is an FMCW radar system in which the transmit and receive antenna are only separated by a small distance. The isolation achieved by WERA was about 80 dB and this was sufficient for their purposes [*Gurgel et al.*, 1999, 2000, 2001]. Some simple tests with the FMCW radar antenna system using a signal generator and a spectrum analyser revealed that there was about 80 dB of isolation between the transmitter and the receiver when four antennas were used for transmitting and four were used for receiving. To further understand where this isolation comes from, consider that the measurements were made in the equipment shed through the cables to the antennas. The cables themselves have 6 dB of loss giving a total of about 12 dB loss from the signal generator to the spectrum analyser. The remainder of the loss results from isolation between the antennas themselves. The four antennas on the ends of the array were chosen, leaving the central four antennas dormant to allow for a physical separation between the two arrays. The isolation of the antennas comes from the fact that the transmit and receive antenna beam patterns have nulls directed between them. This level of isolation was sufficient to allow for a receiver design that could deal with the remaining feed-through.

3.2.4 Transmitter

Figure 3.3 shows a block diagram of the transmitter side of the radar. The computer controls the operation of the radar through the DDS. The amplifier chain is composed of SAPPHIRE amplifiers and a final Alpha 6 amplifier.

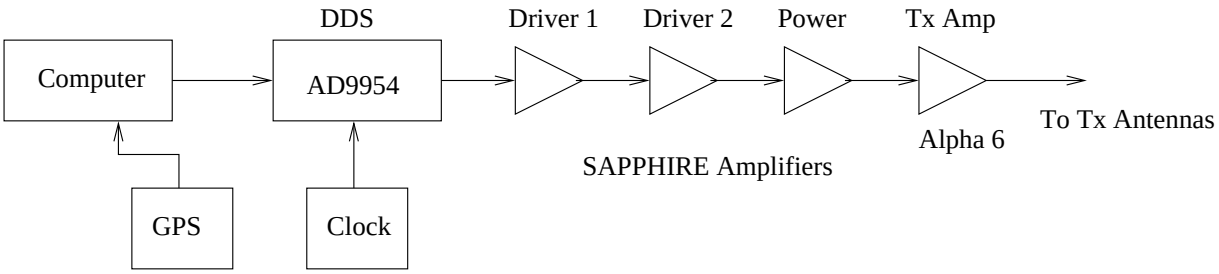

Figure 3.3: Block diagram of transmitter.

Direct digital synthesis

The transmitted signal was produced using an Analog Devices AD9954 direct digital synthesis (DDS) evaluation board. The AD9954 is a 400 MHz, 1.8 V DDS chip and the evaluation board has two AD9954 chips. DDS technology generates a frequency signal from a digital definition of the desired result. Since the signal is generated digitally, precise control over phase and frequency is possible. In its simplest form, a DDS consists of a reference frequency, a phase accumulator, a sine wave amplitude lookup table, and a digital to analog converter. The reference frequency is used as the clock and, through a set of digitally defined rules, is used as a reference to control the rate at which the phase accumulator is changed. The phase accumulator holds the current value of the phase of the signal. The value in the phase accumulator and the rate at which it is changed gives the DDS precise control over phase and frequency. The sine wave lookup table is a memory block that translates the phase word into an amplitude. Sine amplitude values obtained from the lookup table provide the digital input to a fast digital to analog converter [*Harris*, 2003].

A synthesiser previously built at the University of Saskatchewan produced an 80 MHz clock signal for the AD9954 evaluation board. The AD9954 chip has a built in clock multiplier that allows the 80 MHz signal to be multiplied by 5 to generate a 400 MHz internal signal to act as the reference frequency. The same synthesiser signal supplied the reference clock for both AD9954 chips on the evaluation board. This is important as one AD9954 chip supplied the signal for the transmitter path

and the other AD9954 chip supplied an identical local oscillator signal (see below).

The AD9954 chip was programmed to automatically generate the desired linear frequency sweeping. The upper and lower frequency bounds of the sweep are programmed into registers on the AD9954 chip. Unless otherwise stated the lower and upper frequency values were 49.420 and 49.580 MHz. The frequency sweep is performed with small but rapid frequency steps. The size of the step is defined in a register on the chip. The value of a timer is set and each time it completes a cycle, the frequency is stepped by the amount in the frequency step register. Unless otherwise stated, the DDS was programmed to make a 652 mHz frequency step every 900 ns (see Section 3.1). The direction of the frequency sweep (increasing or decreasing) is defined by a bit that is connected to an external pin on the chip. This allows the direction of the frequency sweep to be controlled through the parallel port of the computer (see Section 3.3) [*Harris*, 2003].

SAPPHIRE amplifiers

As mentioned above, a few different components were used from the SAPPHIRE radar system. There are two types of amplifier taken from the SAPPHIRE radar [*Ortlepp*, 1994]. The first is the Driver Amplifier and it uses the Motorola MRF136Y transistor as the active device. The second type of SAPPHIRE amplifier is the High Power Amplifier and it uses a single Motorola MRF141G transistor as the active device.

The DDS was used to generate the signal used in the amplifier chain shown in Figure 3.3. The Alpha 6 amplifier was the final amplifier and it required an input drive of 40 W or 46 dBm. Two SAPPHIRE Driver amplifiers and one SAPPHIRE High Power amplifier chained together provided the drive for the final amplifier.

Alpha 6 amplifier

The Alpha 6 amplifier is a 1.5 kW amplifier that uses the Svetlana 4CX1600B tube as its active device. It was used as the final amplifier stage before the transmitting antennas. As heating was a concern with the Alpha 6 amplifier, particularly during

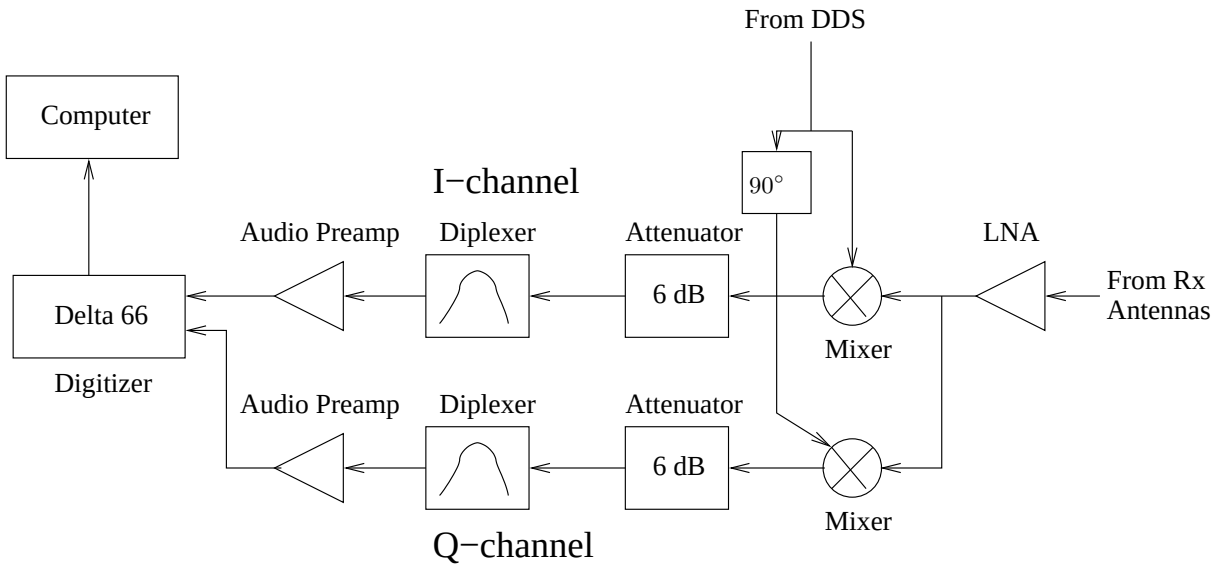

Figure 3.4: Block diagram of receiver.

summer, the device was only operated at about 1.0 kW for the majority of the data collection (for operation at 1.0 kW, the Alpha 6 required a 40 W input drive). The signal from the Alpha 6 amplifier was split into 4 equal components using a power splitter, and each component was sent to one of the four transmitting antennas.

3.2.5 Receiver

Figure 3.4 illustrates the block diagram of the receiver. The receiver design is based on the R2pro receiver [*Campbell*, 1992, 1993; *Campbell and Kelsey*, 2003]. The R2pro receiver is an amateur radio direct conversion receiver designed by Rick Campbell. Direct conversion means the received signal is directly mixed down to baseband without an intermediate frequency stage (i.e. superheterodyne stage) common in receiver designs. An intermediate frequency stage provides an opportunity for filtering and amplification at a frequency above baseband. Without an intermediate frequency stage, the direct conversion receiver poses some design challenges. Using the low noise preamp and downconverter from the R2pro receiver helped to resolve these difficulties.

Low noise amplifier

The low noise amplifier (LNA) in the receiver block diagram in Figure 3.4 is part of the R2pro receiver. The received signal is detected by the antenna array and the signal from the four antennas are combined in a power splitter/combiner. The combined signal passes through the LNA of the receiver where some filtering and amplification takes place. The low noise preamp provides approximately 8 dB of gain as well as acting as a low pass filter on the front end of the receiver.

Local oscillator

After the LNA, the signal is split into an I and Q channel as shown in Figure 3.4. The local oscillator signal is supplied by the second DDS chip on the AD9954 evaluation board. As mentioned in Chapter 2, a copy of the transmitted signal is mixed with the received signal. A common way to implement this is to split the transmitted signal at a point in the Tx amplifier chain where the power level is appropriate for the mixer. However, splitting the signal made it difficult to produce the desired output level on the transmitter side while simultaneously achieving the right level of power for the local oscillator. Since the AD9954 evaluation board had 2 DDS chips, a practical solution was to use one of the chips to generate the input for the transmitter chain and use the second chip to generate a local oscillator signal that was identical to the transmitted signal. The signal generated by the second DDS required amplification and this amplification was performed by a SAPPHIRE Driver amplifier. The DDS provides a signal of -6 dBm and the local oscillator input power required for the mixer is 7 dBm. The amplifier provided about 16 dB of gain which amplifies the signal out of the DDS to 10 dBm. The signal is then split into the I and Q channels using a Minicircuits 90° splitter putting each signal at the desired 7 dBm power level. The use of a separate DDS chip for the local oscillator also allowed the transmitter amplification chain to be manipulated without impact on the strength of the local oscillator signal.

Downconverter

The R2pro downconverter module consists of the RF splitter, mixer, attenuator, diplexer, and audio amplifier shown in Figure 3.4. The RF signal from the preamp is split into two channels that become the I and Q baseband channels after mixing. Each channel is input to a Minicircuit TUF-3 mixer for downconversion with the local oscillator signals. The mixers are terminated with matched 6 dB attenuators. The attenuators serve three primary functions. They provide ideal termination for the mixer stage, reduce 1/f noise, and set the input impedance for the diplexer network. The diplexer network provides filtering of the audio signal out of the mixer. Components (such as capacitors and resistors) are matched between the two channels to preserve the relative phase of the I and Q channels. The preservation of relative phase between the I and Q channels is important for capturing the phase of the received signal [*Campbell and Kelsey*, 2003].

Delta 66 soundcard

The output of the downconverter module is the baseband I and Q channel. An M-audio Delta 66 sound card is used as an analog to digital converter interface for the computer. The Delta 66 has a break out box that allowed multiple channel analog input to the sound card. Only certain sampling rates were available with the Delta 66 sound card [*M-audio*, 2003]. A sampling rate of 9600 Hz was selected since it captured the frequencies of interest (as discussed in Section 3.1). The Delta 66 has a sampling resolution of 24 bits. Once digitised, the data are stored to files on the hard drive of the computer.

Feed-through revisited

An important challenge that the receiver had to overcome was dealing with the strong feed-through signal from the transmitter. The receiver had to be able to detect weak ionospheric scatter beside a very strong transmitted signal. In order to do this, the receiver took advantage of the FMCW frequency sweeping. As discussed in

Section 3.2.2, the separate transmit and receive antennas achieved 80 dB of isolation. Though this isolation was an excellent start to resolving feed-through, the receiver could measure (on the bench) signals that were 80 dB weaker than signals received directly from the transmitter. Due to the frequency sweeping, any signal from the E-region would produce a baseband frequency that is offset from DC. The feed-through signal would be identical in frequency to the local oscillator, and, after the mixer, the feed-through component would be essentially DC. The diplexer and sound card both have capacitors blocking the DC component of the receiver. The fact that the feed-through was essentially DC in the receiver gave the receiver the ability to use analog filters to remove feed-through.

It should be noted that a test of the radar with a pure CW signal, offset in frequency from the local oscillator to bypass the DC filtering, completely swamped the receiver with feed-through.

3.2.6 Power supplies

Power for most of the components was supplied by SAPPHIRE power supplies. The SAPPHIRE power supplies provided a stable 28 V source, and they were designed to supply power for the SAPPHIRE amplifiers. The components for the R2pro receiver required 12 V. A SAPPHIRE power supply provided an input voltage for a 12 V regulator which, in turn, supplied power for the receiver components. This included power for the LNA and the downconverter network. The DDS required a 1.8 V power supply. The output from the 12 V regulator was used as the input to a 1.8 V regulator to produce the power supply for the DDS.

The use of a single SAPPHIRE power supply to provide power for both the DDS and the receiver components (through voltage regulators) was the only point in the receiver where a common power supply was used for transmitting and receiving equipment. The power supplies for the SAPPHIRE amplifiers were separate from those used in the receiver.

The power supplies converted the 120 V AC wall supply into a 28 V DC power

supply. The only components that did not receive power from the SAPPHIRE power supplies and voltage regulators were the computer, synthesiser, and the Alpha 6 power amplifier. Power for the Alpha 6 amplifier was supplied by a 240 V AC wall outlet and power for the computer and synthesiser was supplied by a 120 V AC wall outlet.

3.3 Software

The radar was controlled by a computer using a Linux operating system. The computer controlled all aspects of the radar operation from transmission to data collection. At start-up when power was supplied to the components of the radar, the DDS and the sound card had to be initialised. The DDS needed to be programmed to generate the proper frequency sweep and the sound card needed to be initialised and set to the appropriate sampling configuration. Once these two devices were programmed, the computer controlled the operation of the radar by simply controlling the timing and direction of each frequency sweep. The data collection was initiated by configuring the sound card. After a set of data were collected it was saved to a file. The saved data were then written to a DVD for transport to the University of Saskatchewan where it was analysed.

3.3.1 Control of the DDS

The AD9954 DDS was programmed to perform the desired frequency modulation by loading its registers with the appropriate values. The AD9954 evaluation board was designed to be programmed through the parallel port using a graphical user interface (GUI) supplied by Analog Devices. However, the program was only compatible with Microsoft Windows and no equivalent software was available for Linux. There was a need for the DDS to be programmed automatically without the radar operator manually loading the DDS values with a GUI. Furthermore, the rest of the radar operation was controlled through the Linux operating system. Using the GUI required loading MS Windows, programming the DDS, and then rebooting into Linux. This

created further complications including the need to unplug the parallel port from the DDS board while rebooting into Linux to prevent system checks from resetting the DDS during reboot. Therefore, Linux software was written to program the AD9954 chips through the evaluation board hardware. The evaluation board was already designed to allow the AD9954 chips to be programmed through the parallel port. The software that was written took advantage of this design, programming the DDS by writing the appropriate sequence of bytes to the parallel port registers.

The DDS was programmed to operate in a linear frequency sweeping mode. All values for the linear frequency sweep needed to be set including upper and lower frequencies, frequency step, and the timer that controlled the rate of frequency stepping. Once the appropriate registers in the DDS were set, frequency sweeping could be controlled by a single bit in the DDS that was connected to an external control pin. If the pin was toggled the frequency would sweep from an initial frequency to an end frequency. Once the end frequency was reached the DDS would continue to generate the end frequency until the bit was toggled. Then the DDS would perform a reverse frequency sweep returning to the initial frequency value and holding the bottom frequency until, once again, the bit was toggled. This gave the computer software simple control over the timing of the triangular modulation and allowed it to coordinate with the data collection.

3.3.2 Control of the sound card

Data sampling with the sound card was controlled through the Advanced Linux Sound Architecture API (`http://www.alsa-project.org/`). Similar to the DDS, the sound card needed to be initialised at start up. For data collection it was programmed to a digitisation rate of 9600 Hz. As this was in the audio range of frequencies, a sound card is an economical and efficient digitisation device compared to much more expensive dedicated digitisers. The M-audio Delta 66 sound card is one of high quality, offering 24-bit resolution and 12 independent sampling channels. Only two of these channels are needed for the I and Q channels of the radar receiver.

E-region coherent backscatter can have a dynamic range of 60 dB or more. As such, E-region radar systems often implement an automatic gain control system to keep receiver operation in the linear region. If a receiver system has high sensitivity at low signal levels and stays in the linear region for strong signal levels, increasing the digitisation resolution is equivalent to implementing automatic gain control. The latter approach was taken here to keep the design of this prototype system simple. The excellent rejection of noise and interference of the FMCW design allows for good sensitivity at low signal levels. Although 24-bit resolution was used, 16-bit resolution likely would have been sufficient as this gives a dynamic range of ~ 90 dB; however, it was not possible to select 16-bit resolution on the Delta 66 sound card.

3.3.3 Software timing

Timing between the data collection and the transmitter sweeps needed to be coordinated. As explained, the frequency sweep is controlled by a bit on the DDS chip that can be toggled through the parallel port. Since scatter is expected from a line of sight range of up to 900 km, the maximum expected propagation time of the signal is about 6 ms (see Section 3.1.1). As such, a 6 ms delay between the start of a frequency sweep and the initiation of data collection was desired. Although timing between the radar operation and data collection was important for radar performance, only timing on the order of a millisecond was necessary.

Software timing was accomplished through a combination of threads and scheduling policy. As such, rarely was the timing off by more than a millisecond. This was considered sufficient for the radar, particularly since the data collection occurred over a period of 213 ms and each sweep took slightly more than 221 ms. This ensured that the operations controlling the radar would run at essentially real time without installing a real time operating system. The following sections describe how scheduling policy and threads were used.

Scheduling policy

The Linux operating system is a multitasking system. Processes share access to the central processing unit (CPU) based upon priority and scheduling policy. The software that controlled the radar and data collection was set to top priority on the Linux system and assigned CPU time with a SCHED_FIFO scheduling policy. SCHED_FIFO is a first in first out (FIFO) scheduling policy that queues processes based on priority and the order in which they became ready to execute. Programs that run with SCHED_FIFO scheduling policy will preempt tasks that are scheduled with the normal SCHED_OTHER policy.

Threads

Threads were used to aid the radar control and data collection. A thread is a separate software process or procedure that is started by a main program but runs independent of it. This allows the operating system to schedule the threaded process separately from the main program. For the FMCW system, the main program controlled the radar and initiated data collection. The main program also created two threads for data processing and storage; one for the FM up sweep data (up thread) and one for the FM down sweep (down thread). Each thread was created and then waited for a signal from the main program. Each thread had a buffer that was filled with data collected by the sound card. Once a set of data had been collected the main program activated the appropriate thread depending on whether the data were collected during an up sweep or down sweep. The thread then formatted the data and wrote the data into a data file. Top priority was given to the main program and second highest priority was given to the threads. Setting the main program to operate at a higher priority than the threads ensured continuous radar operation and data collection.

The main program was in charge of all time critical events required for the radar. It toggled the DDS bit to control frequency sweeping and signalled the sound card to start data collection. Once the data were collected the main program sent a

signal to the appropriate thread telling the thread that data are available. The main program then started the next frequency sweep while the thread performed its task of formatting the data and writing it to a file. Due to its higher priority, the main program interrupts a thread when radar control or data collection tasks need to be performed. The main program is idle for the majority of the time while it waits for the sound card to collect data; as such, the threads had ample CPU time to complete their tasks.

The threads performed the non-timing critical tasks of formatting and writing the data. Each thread would wait for a signal from the main program. Once the signal was received, it took data that were placed in the buffer and processed it. The buffer was actually part of a simple structure that stored a time stamp from the start of the frequency sweep and the 12 channels of 2048 samples. As only 2 channels contained the I and Q baseband data, the 10 channels without data were removed. The computer required 4 bytes to store each sample of data (i.e. PC memory can only access bytes of even numbered memory locations) even though the sound card only had 24-bit (3-byte) resolution. This meant that one byte was composed of zero bits. In order to reduce data storage size, the data processing and storage threads removed the all zero bytes from the sample. The thread then wrote the data to a file.

Data files were shared between the two threads to allow the up and down sweep data to be recorded together. Each thread also shared access to its buffer with the main program. In order to control access to these memory segments and ensure that data were not lost or confused between threads, a mutual exclusion object or mutex was associated with each shared item. A mutex is an object that controls access to resources that are shared between processes. Each process that wanted to access the resource had to lock the respective mutex before accessing the resource and then unlock the mutex when the process was done with the resource. If a process tried to access a resource while a second process had locked the mutex, the resource could not be accessed and the first process would have to wait. This ensured that the two threads did not write simultaneously to a data file or that the main program did not

write to a buffer while a thread was processing data from a previous sweep.

Using two threads, instead of just one, allowed each thread to have a buffer that could be filled by the sound card. From an organizational point of view, two threads made sense initially. However, a single thread with two buffers could have performed the same function. In hindsight, a single thread would produce a simpler algorithm. The functional difference of going to one thread would be negligible, however, the change was never made.

3.3.4 Data storage

When a thread wrote data to a data file it used the gzip library to compress the data as it was written. GNU zip or gzip (`www.gzip.org`) is a tool for compressing data. A C library is available to allow a program to open and write gzip files. New data files were generated when the radar was started and after a predetermined number of samples had been collected. A data file stored approximately half an hour of data. Each data file contained an initial header that included a characterization of the frequency sweep as well as the sampling frequency. When a thread wrote data to the file it preceded each data segment with a time stamp indicating the time at the start of the data set and a flag variable indicating whether the data came from an up sweep or a down sweep. Data files were stored on the hard drive of the computer until they could be written to a DVD for permanent storage. Data written to a DVD were then analysed at the University of Saskatchewan.

For the FMCW radar project, no data selection algorithm was implemented, and all data that were collected during radar operation was stored. It was useful at the early design stage of the project to keep all the data since any data could provide information not only on auroral events, but also on radar performance. Furthermore, an algorithm that only kept data containing E-region scatter might be unreliable causing important data to be discarded. Future implementations could significantly reduce the amount of data handling required by using an algorithm to eliminate data with little or no signal. A simple algorithm could measure the spectral SNR as the

data were collected and discard any data that did not exceed a specified threshold or noise-floor.

The threads also wrote the data to a temporary file that could be monitored or analysed as the data were being collected. The temporary data file contained uncompressed data in the full 4 byte data storage format. Data from the temporary file could be observed remotely using a web browser. The FMCW radar system contained an analogue modem for remote access. A web server at the university was set up with a link that would cause it to dial the modem at the radar computer. When this happened, the most recent data set was analysed and an image representing the data were displayed in the web browser. The remote access not only gave information about the data currently being recorded but also provided information about the radar operation.

Chapter 4

Data analysis

This chapter presents the process of obtaining the Doppler and range information of scattering targets in the E-region from the raw data collected by the FMCW system. A significant portion of the data analysis involves the removal of noise. Both systematic and white or random noise must be eliminated or reduced to acceptable levels, and these considerations play a major role in the data analysis.

4.1 Data technical overview

Data collected by the radar were transferred from the computer to a DVD and transported to the University of Saskatchewan. The Python programming language [*Lutz and Ascher*, 2004; *Martelli*, 2003] was used for data processing. Python is a powerful open source and object oriented scripting language. It was chosen for its good numeric extensions and free availability on Linux systems.

Before it could be processed, the data had to be extracted from the gzip format in which it was collected and stored (see Section 3.3.4). The bytes that were removed from the data during data storage were replaced to allow for easier data handling. The task of reformatting the data file from a 24 bit gzipped file to a 32 bit unzipped file was performed by a C program executed from the main Python program.

The header at the beginning of each data file provided the information about parameters used for the data collection. The parameters included the minimum and maximum frequencies of the frequency sweep, DDS frequency step and step interval, and sampling rate. The header data were used to convert the FFTs of the time sampled data into range and Doppler information. The data analysis program could

distinguish data collected during an up sweep from data collected during a down sweep by reading a flag that was written at the start of each data segment (see Section 3.3.4).

4.2 Noise

Distinguishing signals from noise is the basic challenge of any radar experiment [*Skolnik*, 1980]. Noise is any signal that does not convey useful information. As with all radar systems, noise played a central role in the signal analysis for this radar. Noise was divided into two categories: systematic and random. Systematic noise refers to any noise that contributes a signal to the spectrum in a consistent way. It is noise that will be in the spectrum on every frequency sweep. Systematic noise consisted primarily of 60 Hz (and harmonics of 60 Hz) noise from the commercial power grid. The 60 Hz noise combined with 1/f noise dominates the lower part of the spectrum. Random noise is any noise that contributes an unpredictable signal. Random noise in this radar was generated by a variety of sources, including sky noise from cosmic radiation and electrical noise from the thermal motion of electrons in the receiver. Other radio signals in the area would also contribute noise to the system. Sky noise dominated all random noise, and, as such, was uniformly distributed across the bandwidth of the receiver.

4.3 Determination of spectral signal to noise ratio

The concept that an FMCW radar system measures both range and Doppler shift in the frequency domain was explained in Chapter 2. In order to distinguish which part of the spectrum actually contained signals, the data analysis must determine the signal strength relative to the noise as a function of frequency.

The process for determining spectral SNR is outlined in Figure 4.1. It illustrates each step in the process of going from the time sampled data to the spectral SNR. Data from the up sweep and down sweep were analysed separately. For each sweep,

the time series data are Fourier transformed into the frequency domain. The noise floor was reduced by subtracting the portion of the spectrum without signal from the portion of the spectrum with signal. A moving average was used to smooth the data. Multiple sweeps were averaged together to reduce random noise and increase the strength of the signal. The average amplitude associated with noise was determined by taking the root mean square (RMS) of the amplitude over a section of the spectrum that was not expected to receive backscatter from the E-region. Outside of the practical bandwidth, random noise was dominated by cosmic radiation. This RMS noise value was subsequently divided by the amplitude in each frequency cell to give the spectral signal to noise ratio. If the ratio was above a threshold of 2 (or 3 dB), then a signal was determined to be present. The SNR as a function of frequency is referred to as the spectral SNR.

The steps for determining spectral SNR are described in the sections below. A detailed example of the data analysis process accompanies each step. The data collected starting at 02:47:26 UT on July 23, 2004 are used for illustration.

4.3.1 The fast Fourier transform

The analysis begins with the sampled time series of data collected by the receiver. The time series consists of the two baseband I and Q channels. The data were converted to the frequency domain with a fast Fourier transform (FFT) algorithm. The I channel was used as the real input and the Q channel was used as the imaginary input to the complex FFT. The data were sampled at 9600 Hz with 2048 samples taken for each of the up sweep and down sweep (see Section 3.1 for details about waveform and sampling).

Figure 4.2 is a plot of the time series of sampled data collected at 02:47:26 UT on July 23, 2004. The vertical axis represents the value of the normalised 24 bit data. Due to the sign bit, full scale deflection is ± 0.5. Time, in seconds, is along the horizontal axis with 0.0 s as the start of the sampling time. Harmonics of 60 Hz from the AC power grid are present with the strongest harmonic being 120 Hz. Both

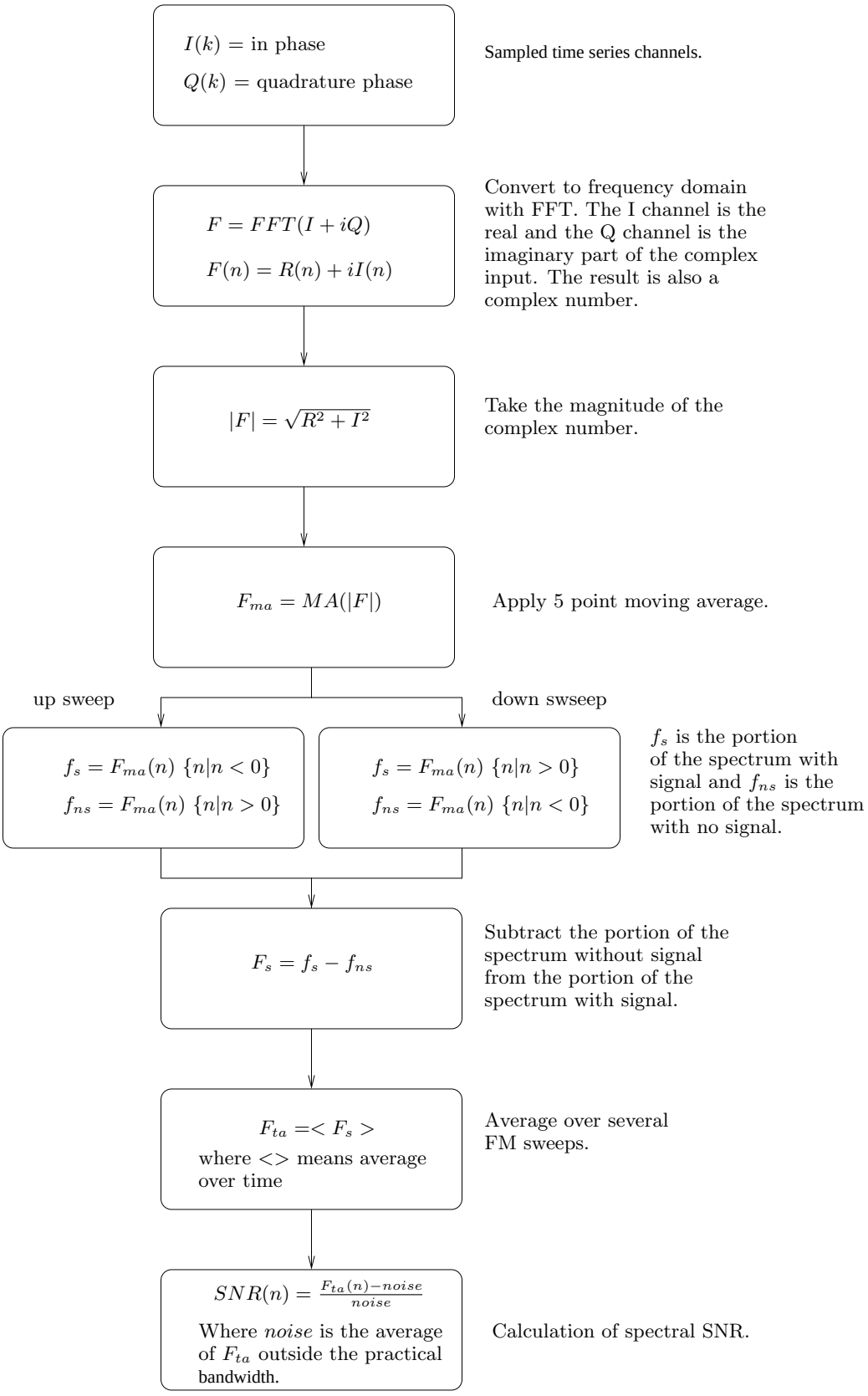

Figure 4.1: Steps used in the calculation of SNR.

the I and Q channel are plotted. The strength of the received signals in the I and Q channel varies over the sampling time. As the amplitude of the I channel decreases, the amplitude of the Q channel increases; however, the overall amplitude (vector addition of the I and Q channels) remains approximately constant.

The time series data were Fourier transformed to the frequency domain as represented in Figure 4.3. Figure 4.3 is a plot of the real and imaginary FFT components as a function of the frequency number. The value of the real (blue) and imaginary (green) components are plotted together. The Nyquist frequency number is 1024 and it divides the FFT between negative and positive frequencies. The frequency numbers along the horizontal axis represent discrete frequencies of [0, 1, 2 ... 1022, 1023, 1024, -1023, -1022 ... -2, -1]. The large spikes in the strength of the signal across the frequency spectrum represent the 60 Hz harmonics interference observed in the time series plot. Figure 4.3 gives an indication of the overall shape of the spectrum. Figure 4.4 is the same diagram as Figure 4.3 with a reduced vertical axis range. The reduced axis range provides a more relevant picture of the spectrum, and reveals the signal of interest (in this case, radar scattering from the E-region) around frequency number 1300. For clarity, the region is highlighted by a red oval. It is important to note that this signal occurs in the negative frequency section for this up sweep example and no signal of interest appears in the positive frequency section. If a down sweep example was used, the opposite would occur, with the signal of interest appearing in the positive frequency section of the spectrum.

4.3.2 Magnitude of the spectrum

The next step was to take the magnitude of the FFT. The magnitude gives the amplitude of the complex FFT values. The magnitude of the FFT for a given frequency cell $|F(n)|$ is calculated by:

$$|F(n)| = \sqrt{R(n)^2 + I(n)^2} \tag{4.1}$$

where $R(n)$ is the real and $I(n)$ is the imaginary component of the FFT at frequency number n. If the phase information of the spectrum is desired, it can be extracted

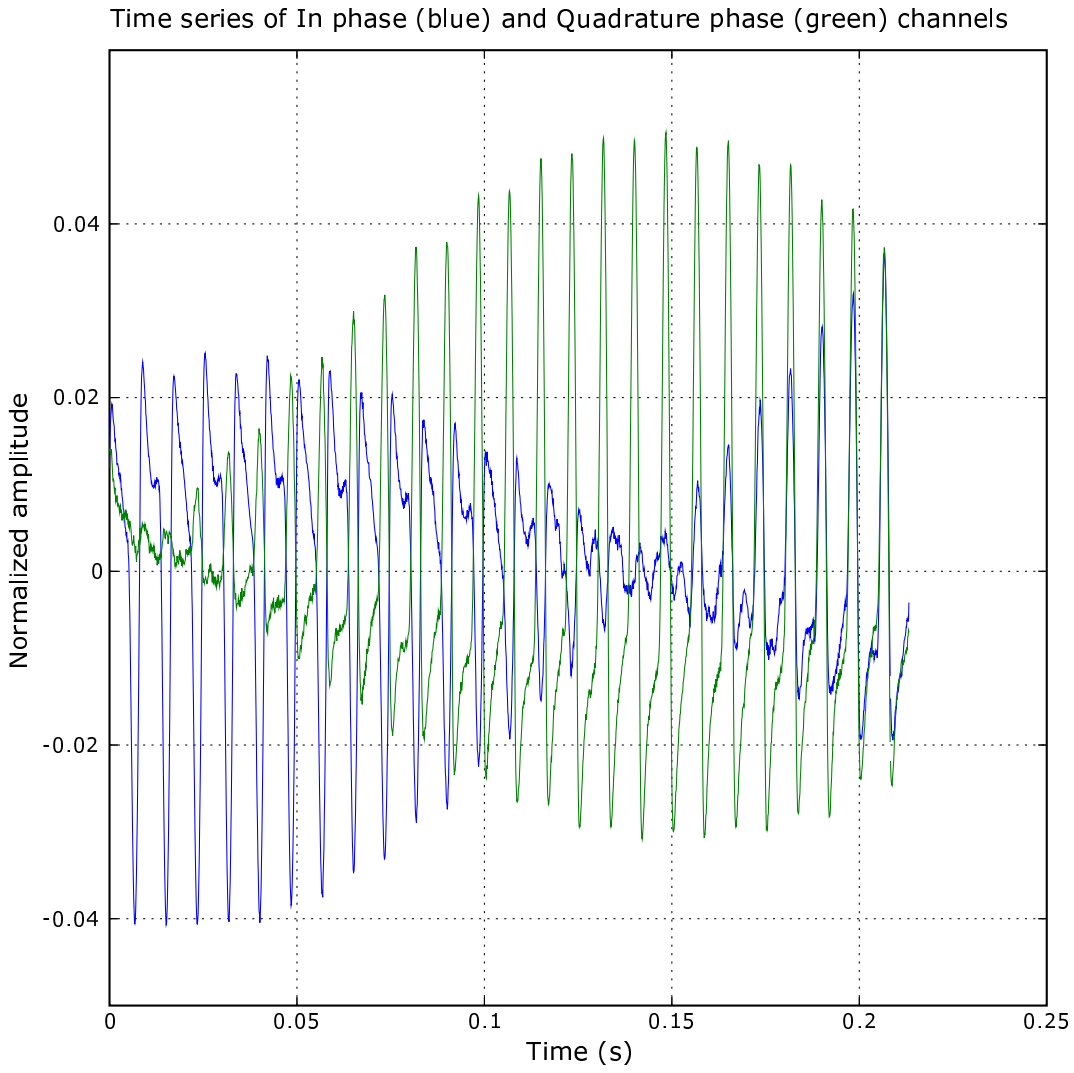

Figure 4.2: Time series of the I (blue) and Q (green) channel from data collected starting at 12:46:26 UT on July 23, 2004.

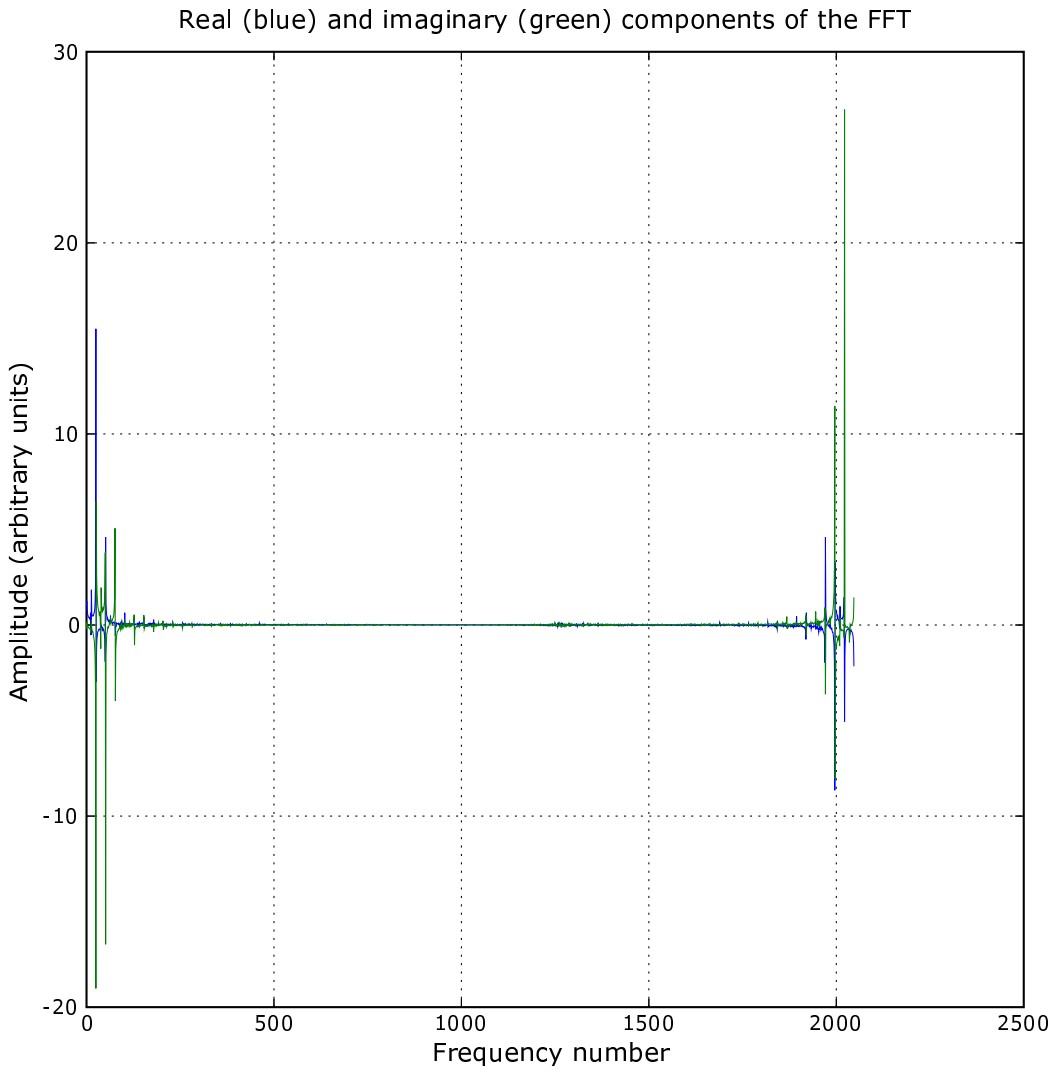

Figure 4.3: Plot of the raw real (blue) and imaginary (green) components of the FFT of time series data presented in Figure 4.2. Raw 2048 point FFT values with frequency numbers representing [0,1,2...1022, 1023, 1024, -1023, -1022...-2, -1].

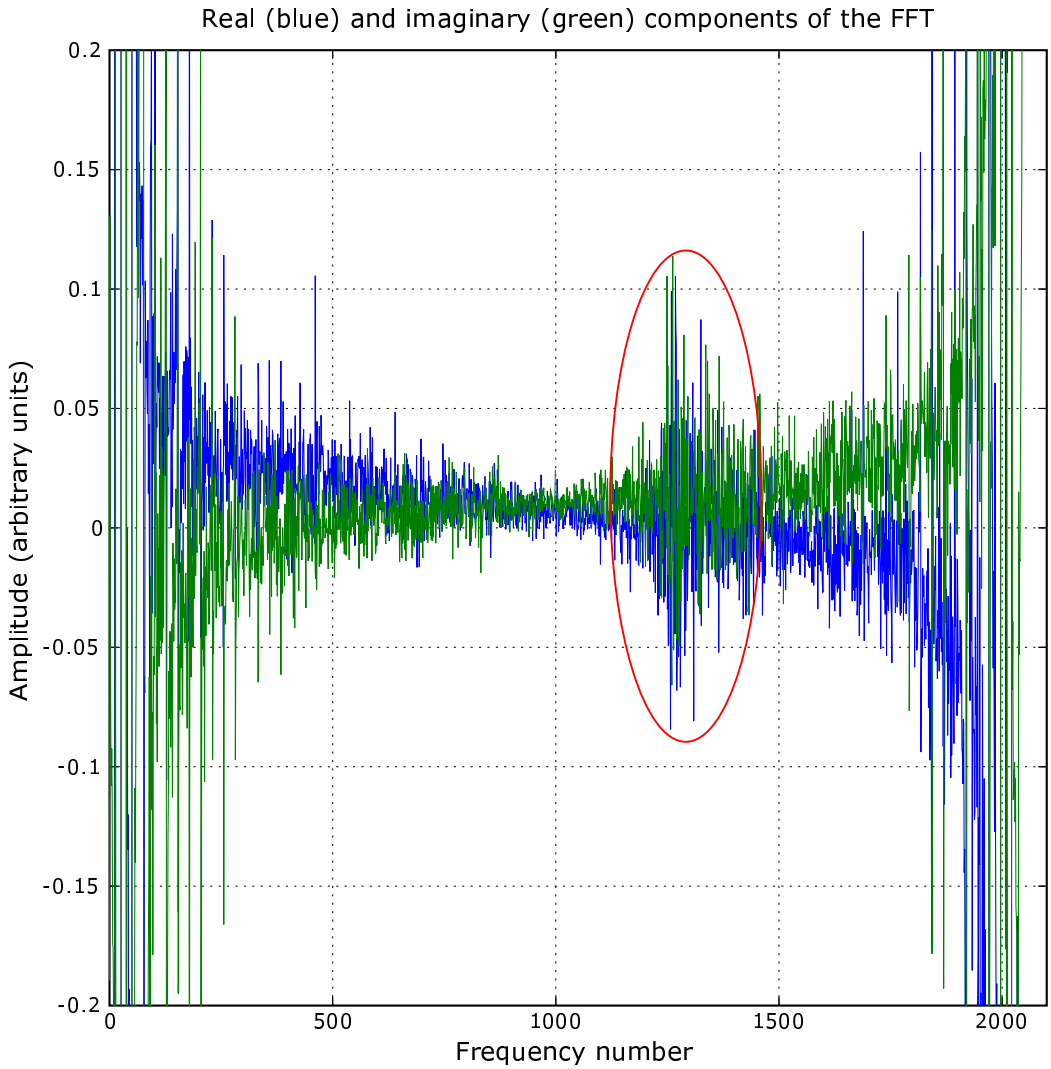

Figure 4.4: Same as Figure 4.3, but the range of the vertical axis has been reduced to show the signal of interest.

from the values of the real and imaginary components in each frequency cell.

Figure 4.5 is a diagram of the magnitude of the FFT data illustrated in Figure 4.3. The horizontal axis has also been converted from frequency number to Hz based on the information about sampling rate and sampling interval. Positive and negative frequencies are plotted as would be expected with negative values on the left and positive values on the right. The full vertical axis is used to show the symmetry in the spectrum between positive and negative frequencies. Figure 4.6 is the same plot as Figure 4.5 but with a reduced vertical axis range which reveals the portion of the spectrum that is of interest. Figure 4.6 shows an increase in the magnitude of the spectrum between the frequencies of -3700 and -3800 Hz and is highlighted by the red oval. No signal is present on the positive frequency portion of the spectrum.

4.3.3 Positive and negative frequencies

Commonly, the Fourier transform is used to transform a set of time sampled data into the frequency domain. However, the FFT is a complex Fourier transform. For many uses of the Fourier transform, the time series samples are treated as the real component of the input to the FFT and the imaginary component is set to zero. The complex Fourier transform gives both positive and negative frequencies (positive frequencies are those frequencies greater than 0 Hz in the Fourier spectrum and negative frequencies are those frequencies below 0 Hz in the Fourier spectrum), but the spectrum is symmetric about zero frequency when only the real component of the time series is available.

When both an I and Q channel are available, the Q channel can be used as the imaginary component of the input to the Fourier transform. Since the I and Q channels are independent samples, the Fourier transform has twice the spectral width of a transform that takes only one component of the time series data. For the FMCW radar system, the I and Q channels are each composed of 2048 time samples. Sampling theory states that each channel yields 1024 frequency bins. As such, when they are combined, they give 2048 frequency bins. As Nyquist's theory still holds,

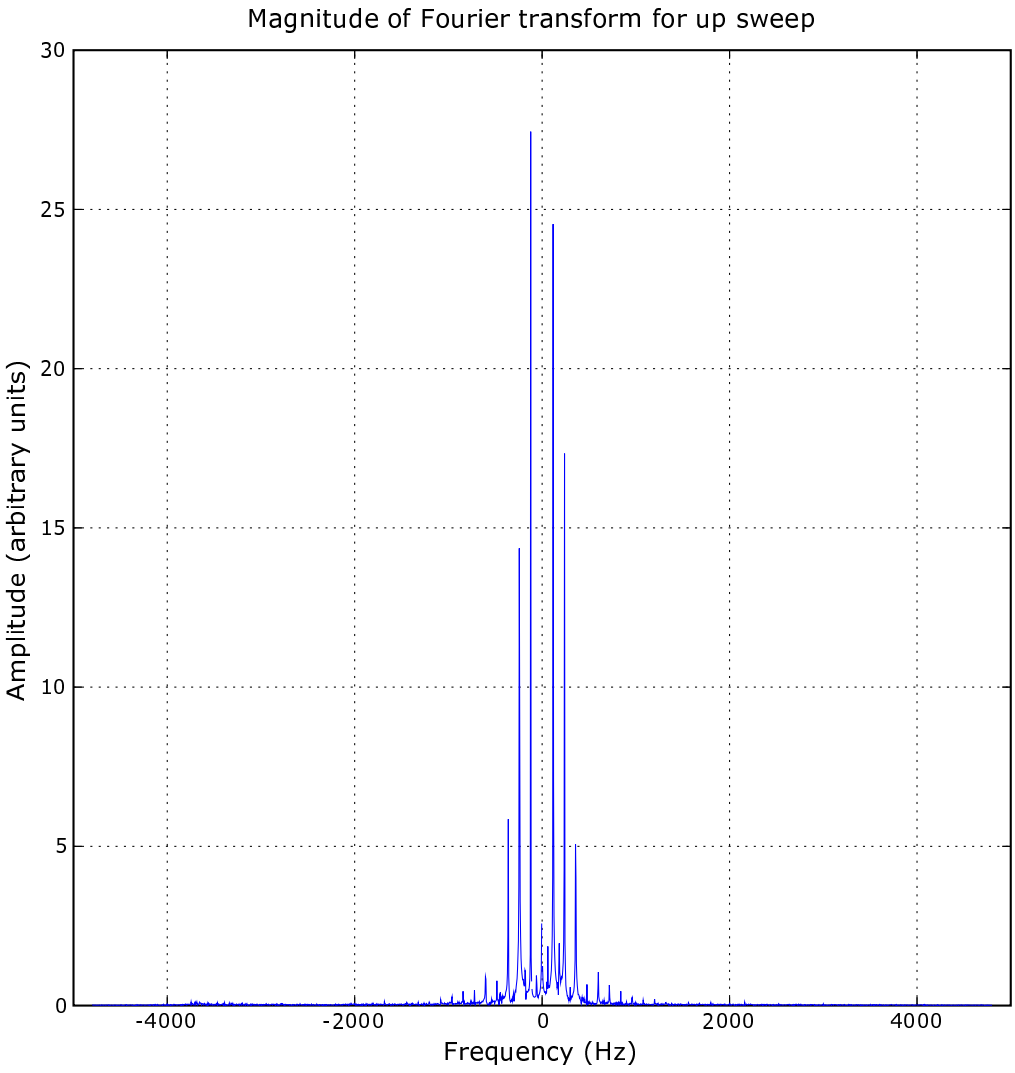

Figure 4.5: The magnitude of the FFT. Frequency number has been converted to frequency in Hz.

each channel provides frequency information up to half of the sampling frequency (4800 Hz for a sampling frequency of 9600 Hz). The extra bandwidth information is available in the form of positive and negative frequencies. With a complex input, the positive and negative frequencies are independent, and, therefore, the effective bandwidth of the discrete spectrum is increased [*Brigham*, 1988].

For a direct conversion radar receiver, such as the one built for this thesis, an RF signal that is at a higher frequency than the local oscillator signal will produce a positive baseband frequency. Conversely, an RF signal that is lower in frequency than the local oscillator will produce a negative baseband frequency. Using Figure 2.3 in Chapter 2 for reference, triangular modulation techniques produce baseband signals that are negative during the up sweep and positive during the down sweep.

Referring to Figure 4.6, the signal of interest is on the negative portion of the spectrum. A similar look at the data from the down sweep shows that the corresponding signal was only present on the positive side of the spectrum, as would be expected. Figure 4.7 is a plot of the absolute value of the Fourier transform of the received signal from the down sweep portion of the FMCW triangular modulation scheme immediately following the up sweep portion whose data were illustrated in Figure 4.6. The axes are defined as in Figure 4.6. Figure 4.7 reveals the presence of a signal on the positive side of the spectrum, once again highlighted by a red oval.

4.3.4 Moving average

A typical baseband spectrum had variations due to noise along with anomalous spikes in the spectrum. In order to smooth this variation, a 5 point moving average was applied to the spectrum.

Figure 4.8 illustrates the spectrum of the practical bandwidth before a moving average has been applied. Figure 4.9 is a plot of the data after a moving average smoothing has been applied. A comparison of Figure 4.9 with Figure 4.8 indicates the effect of having performed the moving average. The moving average significantly reduces noise spikes in the practical bandwidth. Implementing a moving average

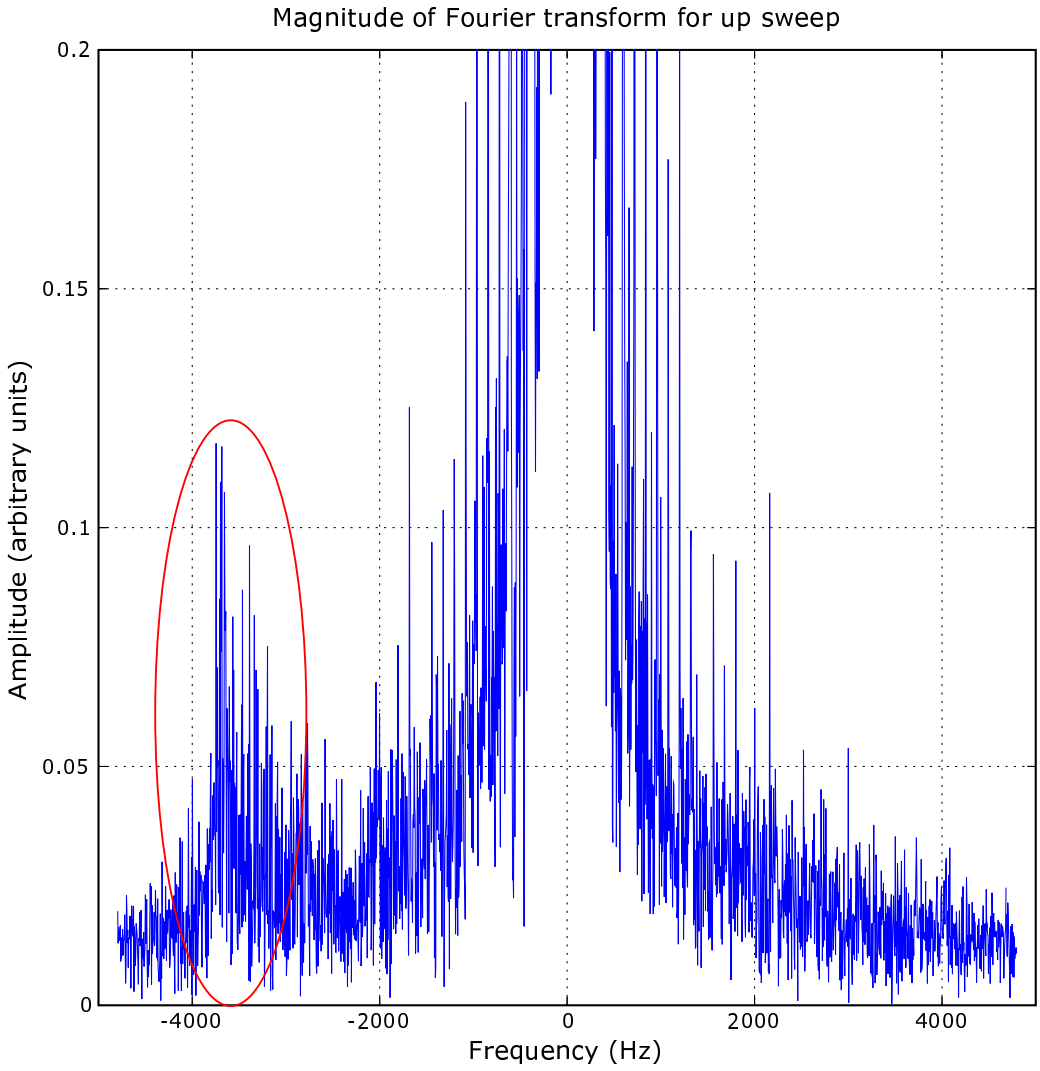

Figure 4.6: A close-up plot of the signal of interest in Figure 4.5. The red oval highlights an anomaly in an otherwise reasonably symmetric spectrum. The anomaly represents the signal of interest.

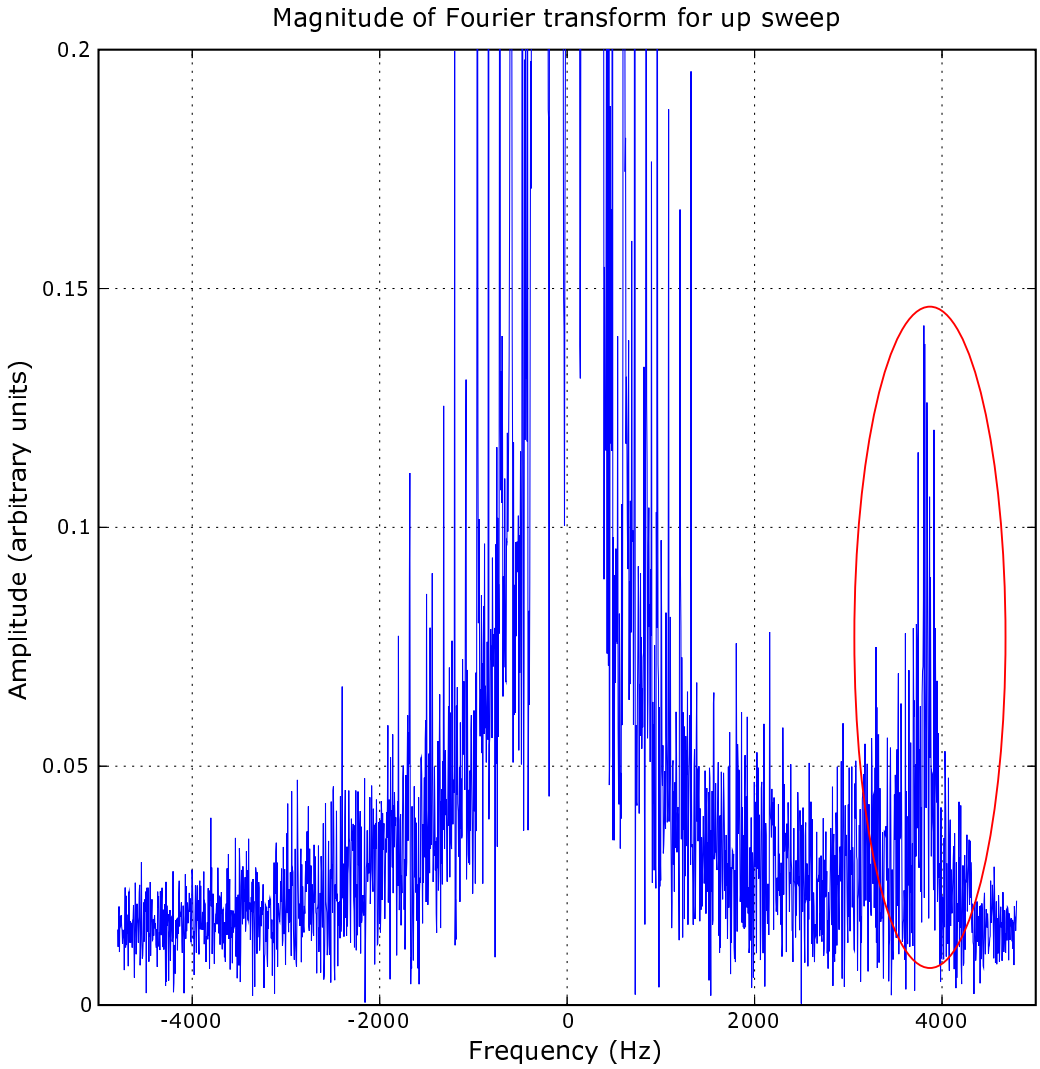

Figure 4.7: Spectrum of the down sweep portion of the triangular modulation scheme immediately following the up sweep portion presented in Figure 4.6. The red oval highlights the presence the received signal of interest.

was an optional analysis step. For most types of E-region observations the results became clearer after a moving average was performed; however, for some observations, particularly meteor data where the signal is strong and has a narrow spectrum, smoothing caused the shape of the spectrum to be significantly altered (i.e. the signal is frequency broadened).

4.3.5 Removal of systematic noise

Since the signal of interest (i.e. the radar signal scattered from plasma irregularities in the terrestrial E-region) was only present in one of either the negative or positive frequency, the side of the spectrum without the signal of interest was used to remove the systematic noise of the system. As can be observed in Figures 4.3 to 4.6, the systematic noise was very strong relative to the signal of interest. The strength of the systematic noise is roughly symmetric between positive and negative frequencies; therefore, a simple way to remove this systematic noise was to subtract the portion of the spectrum without the signal of interest from the portion of the spectrum with the signal of interest.

Referring to the data presented in Figure 4.6, the systematic noise values of the positive frequency cells can be subtracted from their counterparts in the negative frequency cells which contain the signal of interest. Only frequencies between 2000 Hz and 4800 Hz were expected to contain E-region signals, and this section of the spectrum is referred to as the practical bandwidth of the baseband signal (see the discussion in Section 3.1.1). An examination of how the subtraction impacts the systematic noise in the practical bandwidth is illustrated in Figures 4.9 and 4.10. Figure 4.9 presents the practical bandwidth before the subtraction, but after the moving average was performed. A weak E-region backscatter signal exists between -3500 Hz and -3800 Hz. Figure 4.10 shows the same region after the subtraction described above. After subtraction, the noise floor has been noticeably reduced and the signal is more distinguishable from the noise as its magnitude relative to the noise floor has increased. The data for the down sweep data analysed with an identical

process, except any signal of interest will be on the positive side of the spectrum.

4.3.6 Averaging over frequency sweeps

The data from multiple FM sweeps were averaged together during data analysis. Averaging over several frequency sweeps improves the signal to noise ratio and allows weak signals to be better resolved from the noise. As can be seen in Figure 4.10, even after some of the systematic noise has been removed and a moving average smoothing is performed, it is desirable to more clearly distinguish the signal from the noise. Each consecutive sweep in the average systematically increased the amplitude at any frequency with a signal while random noise at any frequency without a signal was reduced. The up sweep data and down sweep data were averaged separately.

The result of averaging 10 up sweeps is depicted in Figure 4.11. Figure 4.11 shows the practical bandwidth of the baseband as well as the section of the spectrum used to calculate noise (see below). A comparison with Figure 4.10 illustrates how averaging multiple sweeps together significantly improves the SNR.

Although averaging over multiple frequency sweeps gives a much improved spectral SNR, it is at the expense of temporal resolution. Averaging over 10 triangular sweeps means averaging the spectrum over $\sim$4.5 s.

4.3.7 Calculation of amplitude SNR

Upon completion of the steps described above, a section of the spectrum at frequencies below the practical bandwidth was selected. Below the practical bandwidth no E-region signals are present, and cosmic background radiation, which is uniform over the receiver bandwidth, completely dominates any instrument noise. Figure 4.11 illustrates the spectrum after 10 consecutive sweeps have been averaged together. The red section in Figure 4.11 depicts a section of the spectrum below the practical bandwidth where noise was determined. The root mean square of the spectrum over this region was calculated to give the amplitude of the noise, *noise*. This noise value was then used to calculate a signal to noise ratio over the practical bandwidth. Any

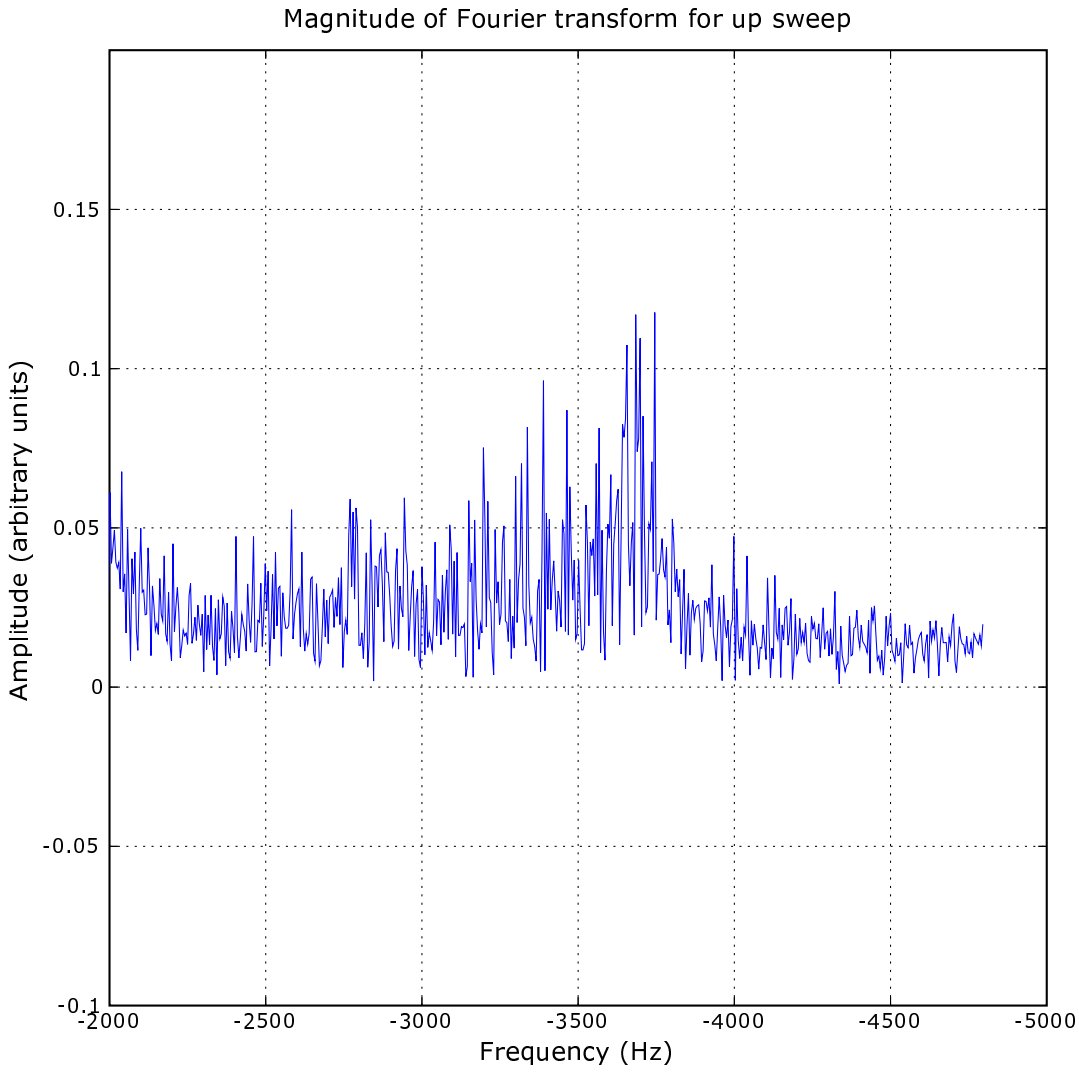

Figure 4.8: The magnitude of the spectrum for the up sweep over the practical bandwidth.

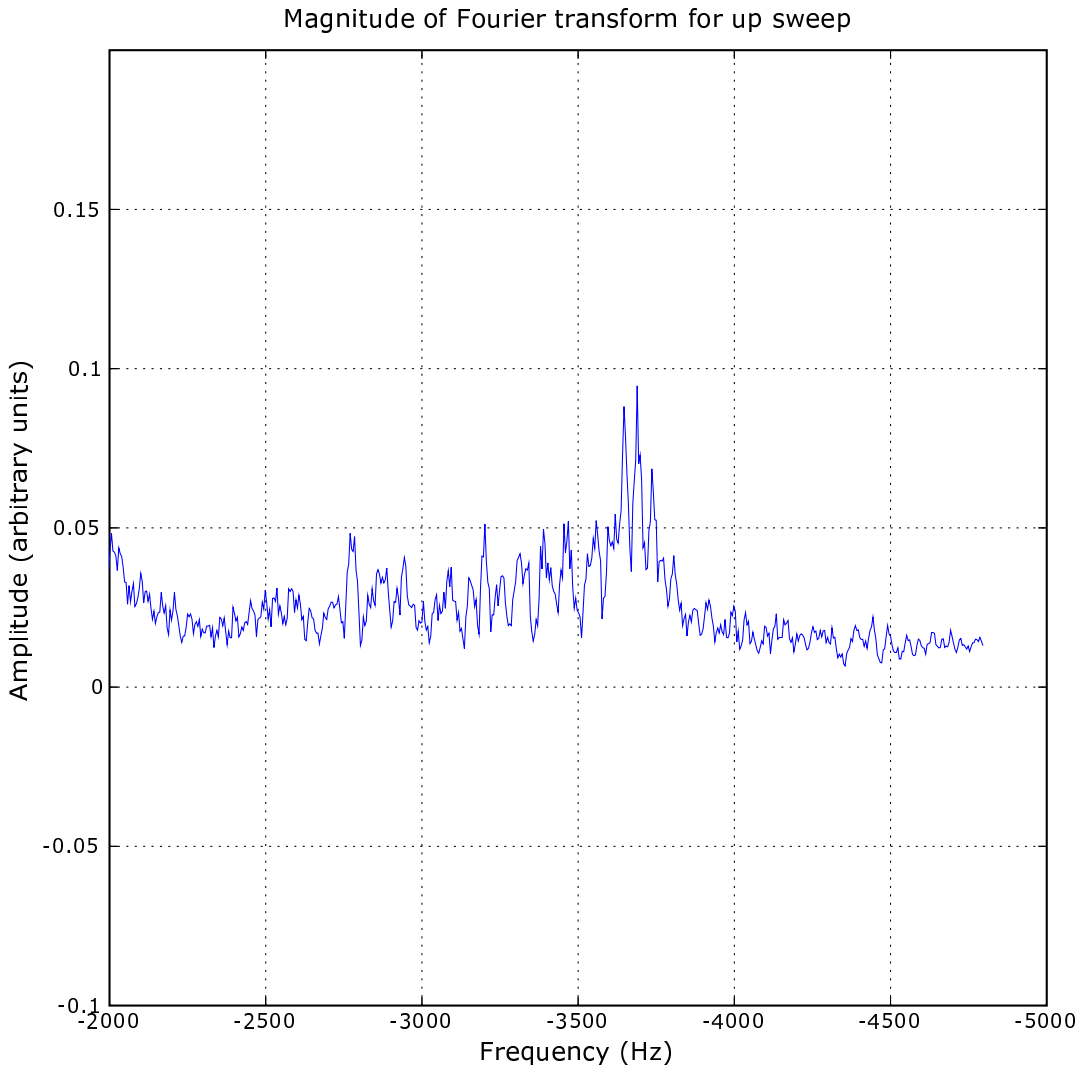

Figure 4.9: The spectrum for the up sweep after a moving average smoothing.

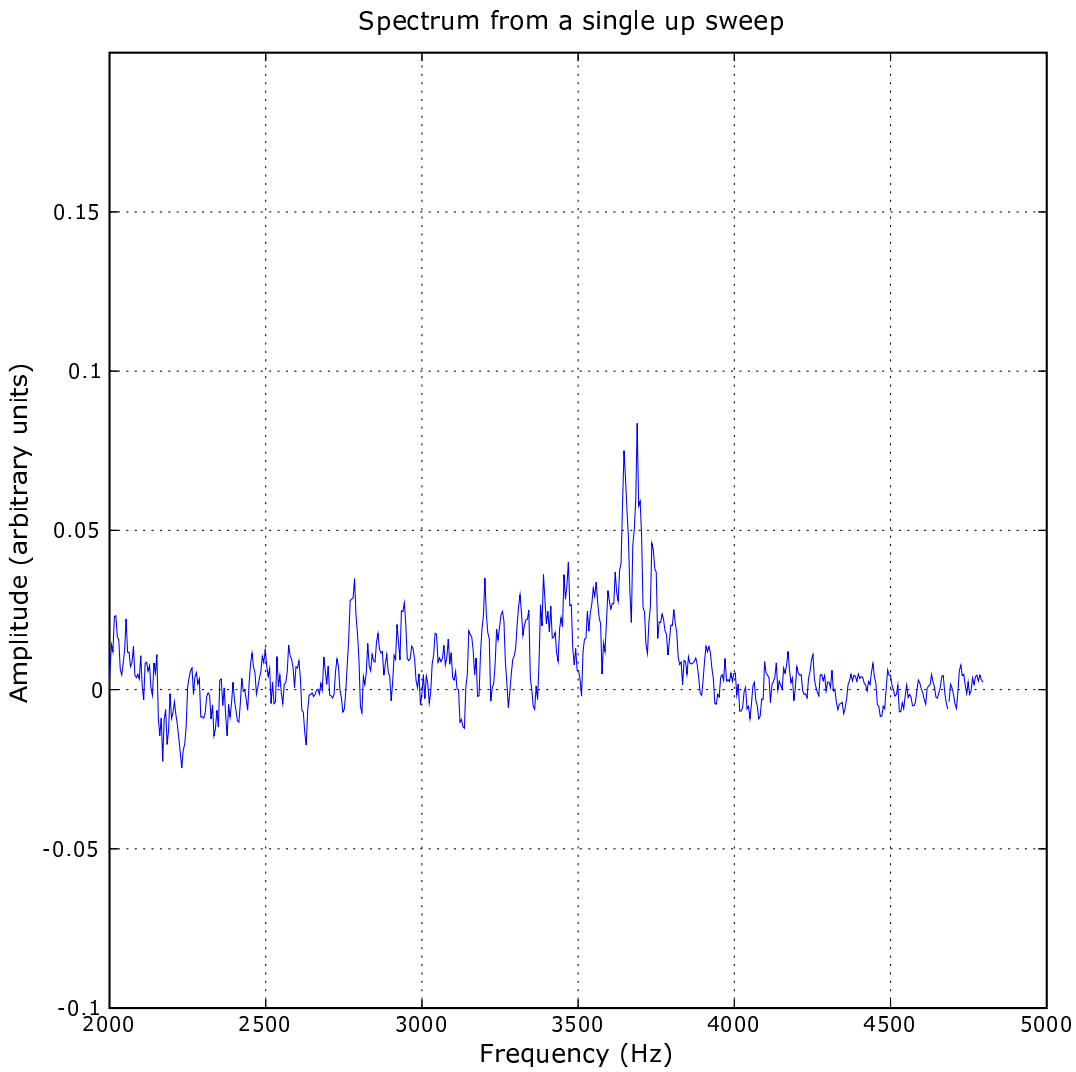

Figure 4.10: The spectrum for the up sweep after the positive region of the spectrum has been subtracted from the negative region of the spectrum. The noise floor has been significantly reduced compared to Figure 4.9.

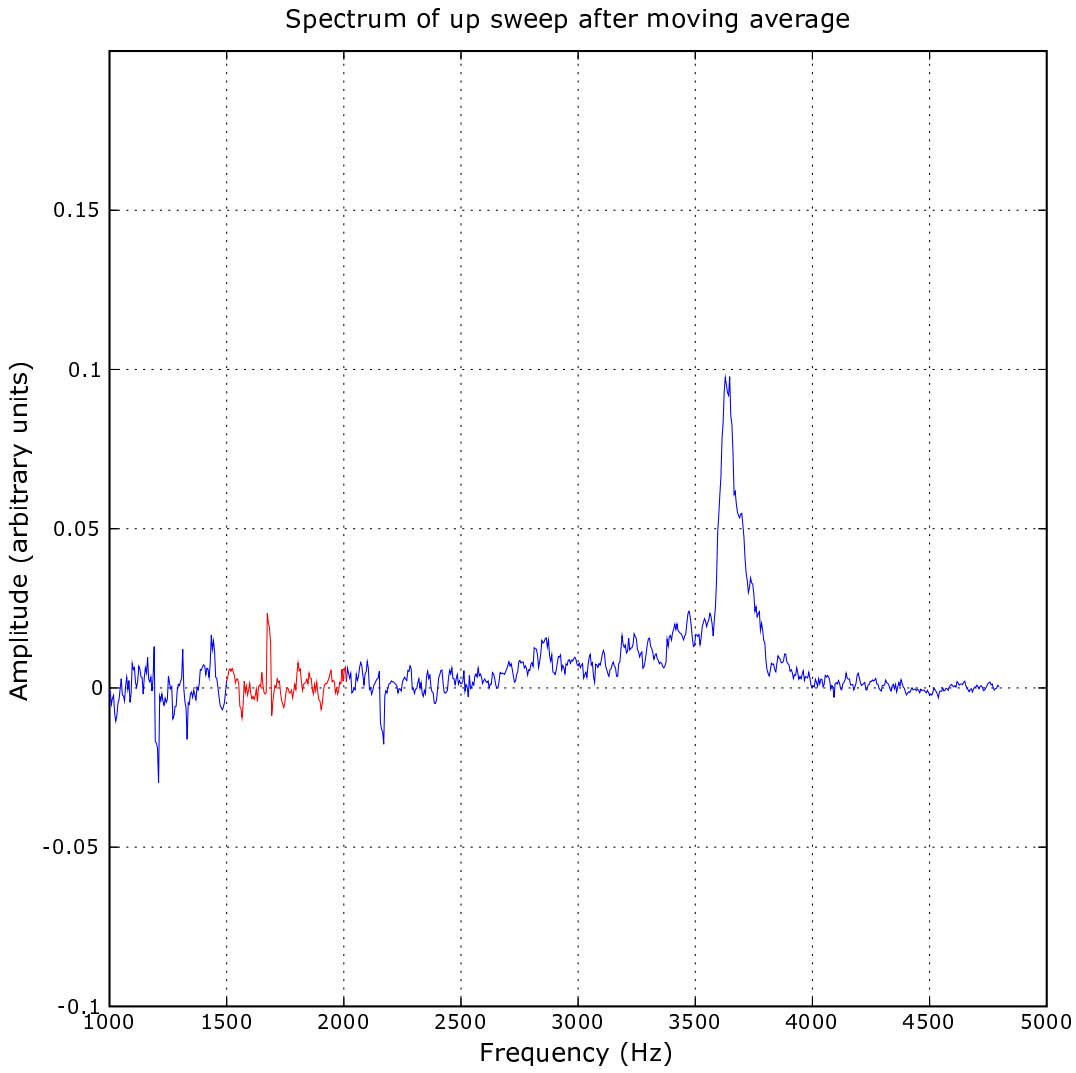

Figure 4.11: Average spectrum of 10 consecutive up sweeps. The red section of the spectrum indicates the region below the practical bandwidth that was used to calculate the RMS noise value.

frequency cell with an SNR value of less than 2 (or 3 dB) was ignored. The SNR was calculated by

$$SNR_{amp}(n) = \frac{F_{ma}(n) - noise}{noise} \tag{4.2}$$

for the linear SNR where $F_{ma}(n)$ is the amplitude of the value at discrete frequency number n after the moving average was performed. The result is the amplitude of the spectral SNR for the received signal.

At this stage of the data analysis, the spectral SNR is expressed in amplitude. A similar calculation is performed for the down sweep data. When the up sweep and down sweep spectral SNR are convolved together, the result of the convolution gives a value that is proportional to the SNR in power (see below).

The result of the spectral SNR calculation for Figure 4.11 is shown in Figure 4.12. All frequencies outside the practical bandwidth (2000 Hz to 4800 Hz) have been ignored. The signal centred on 3600 Hz has a SNR of $\sim$22 (in linear units). Weaker signals with an SNR of less than 5 are present between 3000 Hz and 3500 Hz. The region of the spectrum just below 3500 Hz represents the presence of a weaker scattering region in front of (closer to the radar) the much stronger scattering region centred at 3600 Hz.

4.4 Convolution

The previous section presented the raw data with the final result being the calculation of amplitude SNR across the practical bandwidth. The next step was to translate the SNR data into range and Doppler information by convolution, as discussed in Section 2.2. Recall that both range and Doppler information are measured in the frequency domain by the FMCW radar and that this information must be extracted from the collected data. As discussed in Section 2.2 it can be challenging to separate this information for soft targets such as those due to coherent E-region backscatter. Appendix A presents the convolution process on the same data from the previous sections and presents the problem of ghost targets which are discussed in detail in Chapter 5.

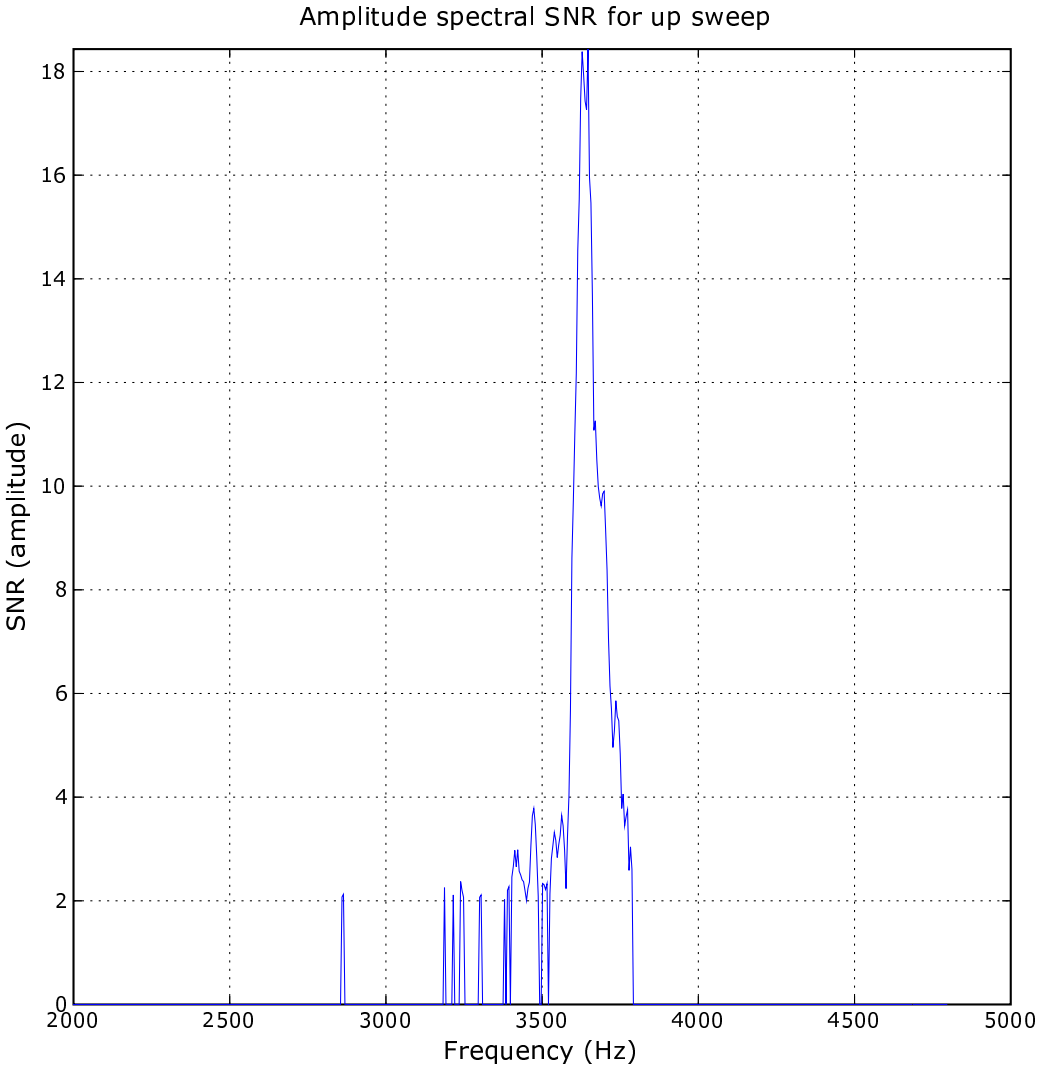

Figure 4.12: The spectral signal to noise ratio.

4.5 Presentation of data

Representing time, range, Doppler frequency, and SNR together is difficult. Only three of the four parameters can be reasonably represented on a single plot.

The primary type of data representation takes the form of a range-Doppler map. The goal is to translate the data from the up and down sweep into a form that represents the Doppler information for each range. The range-Doppler map contains information about all the possible range and Doppler shifts that could have produced the received signal. At each range covered by the radar there is a set of frequencies that 'match up' between the up sweep data and the down sweep data (see Appendix A). The map is produced by displaying the possible spectrum at each range. Figure 4.13 is the range-Doppler map of the radar data corresponding to the examples in the previous sections and Appendix A. Figure 4.13 is a plot with Doppler shift along the horizontal axis and line of sight range along the vertical axis. The amplitude of the signal corresponding to a specific range and frequency is represented by colour. Figure 4.13 covers a range from 500 km to 900 km line of sight from the radar and Doppler shifts between -500 to 500 Hz. As expected from the convolution analysis presented in Appendix A, a strong signal is clearly visible at a range of $\sim$775 km with a Doppler shift of 100 Hz. The width of the Doppler shift appears to be narrow relative to the mean Doppler shift. The magnitude of the SNR is displayed at the top of the graph. The SNR value displayed in Figure 4.13 is produced by the multiplication of the amplitude SNR from the up sweep with the down sweep. As a result, the SNR is the ratio of the power of the signal to the power of the noise.

This is the data presentation format used to plot the FMCW results presented (and discussed) in the next chapter.

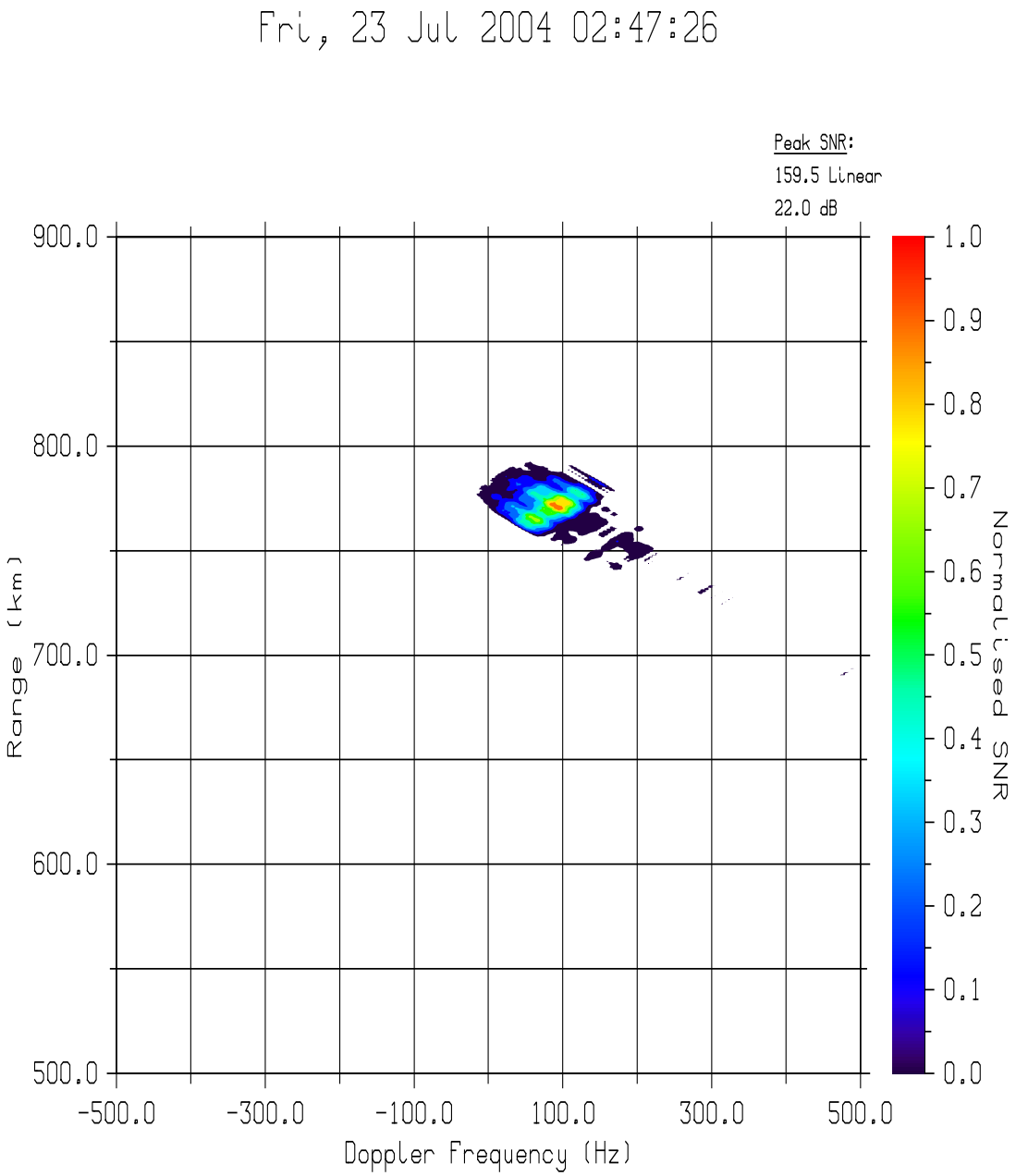

Figure 4.13: The range-Doppler map for data collected on July 23, 2004 02:47:26 UT.

Chapter 5

Results

The FMCW radar began collecting data on June 22, 2004. This chapter presents results from the operation of the radar. Different types of meteor and E-region backscatter observations are presented which verify that the FMCW radar performs as described in previous chapters. An overview of the format for data presentation can be found in Section 4.5.

5.1 Meteor data

The spectral characteristics of meteor trails (see Section 1.4.1 for a discussion of meteor trails) make them a good target for an FMCW radar because they are essentially hard targets relative to auroral phenomena (see Section 2.2 for a discussion of soft targets versus hard targets). During the operation of the FMCW radar, numerous meteor trails were observed. The following represents a typical example of those meteor trail observations.

Figure 5.1 is the range-Doppler map for data obtained on July 23, 2004 at 06:38:54 UT. The figure reveals a signal with a narrow Doppler spectrum and a small mean Doppler shift characteristic of meteor trails. Figure 5.1 indicates a target at a range of $\sim$615 km and a mean Doppler shift of $\sim$15 Hz. A mean Doppler shift of 15 Hz represents a speed of 45 m/s for 50 MHz radar signals.

The data presented in Figure 5.1 are typical of meteor data observed by the radar. The narrow Doppler spectrum of the meteor trail, allows for a clear representation of the range and Doppler characteristics of the meteor trail.

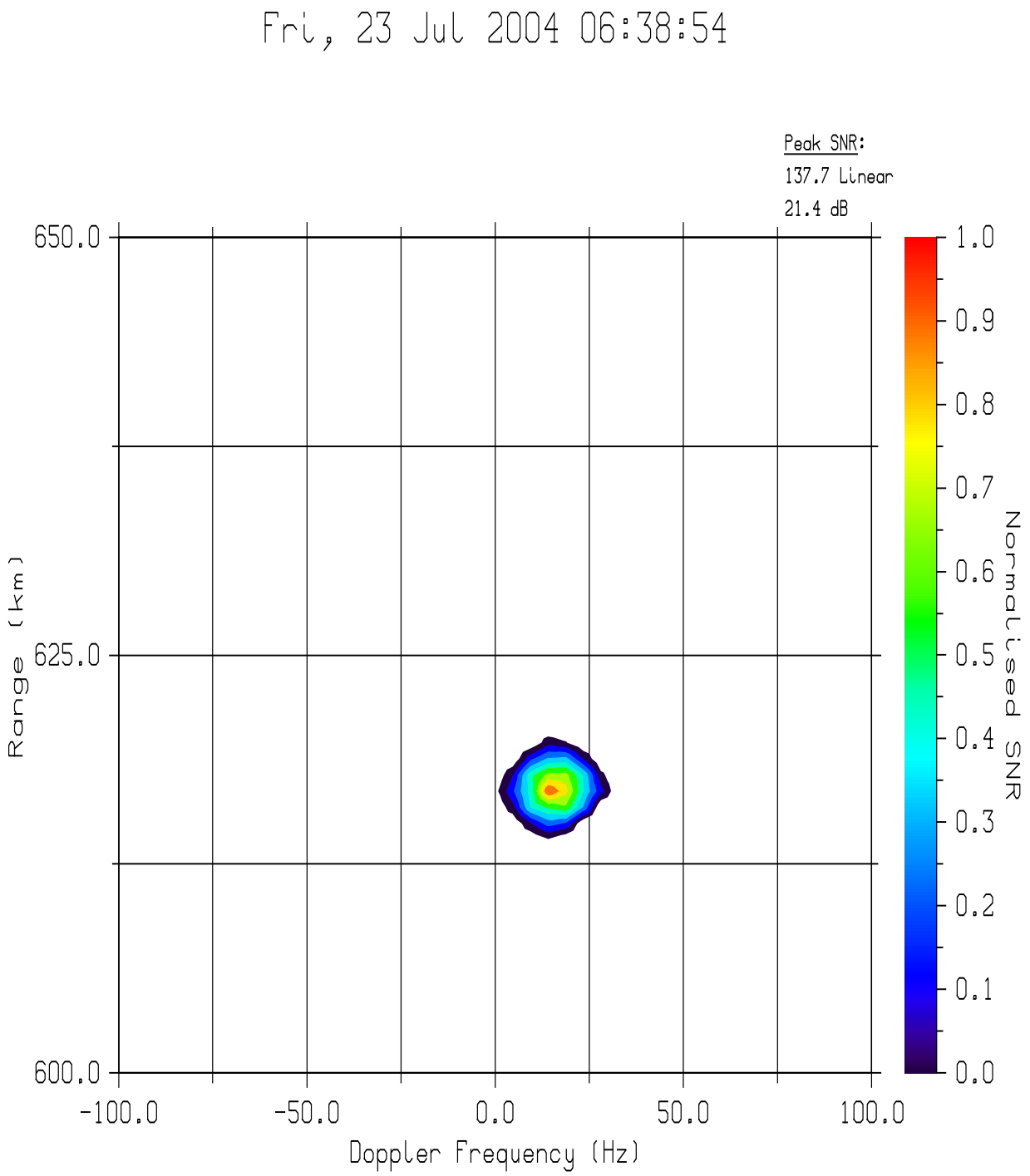

Figure 5.1: A range-Doppler map of a simple meteor echo from 23 July, 2004 06:38:54 UT.

5.2 Type I event

Figure 5.2 illustrates an example of a typical Type I spectrum obtained on July 23 of 2004. Type I waves are characterised by a narrow Doppler spectrum and a mean Doppler shift equivalent to the ion-acoustic speed. In the E-region, the ion acoustic speed is about 360 m/s. The spectrum in Figure 5.2 is narrow with a mean Doppler shift of ~ -120 Hz (~ -360 m/s), and the signal comes from a line of sight range of ~ 850 km.

5.3 Type II event

Type II signatures are characterised by a Doppler spectrum that is broad relative to the mean Doppler shift. Figure 5.3 illustrates the spectrum of scatter obtained on July 23, 2004 at 07:15:25 UT. The strongest signal comes from $\sim$625 km and the spectrum is broad relative to the mean Doppler shift. The mean Doppler shift is close to 0 Hz.

5.4 Type III event

Type III signatures are characterised by a narrow Doppler spectrum and a mean Doppler shift below the ion-acoustic speed that characterises Type I signatures. An example of a Type III signature was collected on July 23, 2004 at 02:46:57 UT and the data from that event are illustrated in Figure 5.4. The signal appears at $\sim$625 km with a mean Doppler shift of $\sim$60 Hz ($\sim$180 m/s).

5.5 Ghost targets

Figure 5.5 illustrates a series of range-Doppler maps. A sequence of 6 consecutive range-Doppler maps are displayed. In this particular event the scatter from a line of sight range of $\sim$520 km decays while weak scatter from a range of $\sim$750 km is

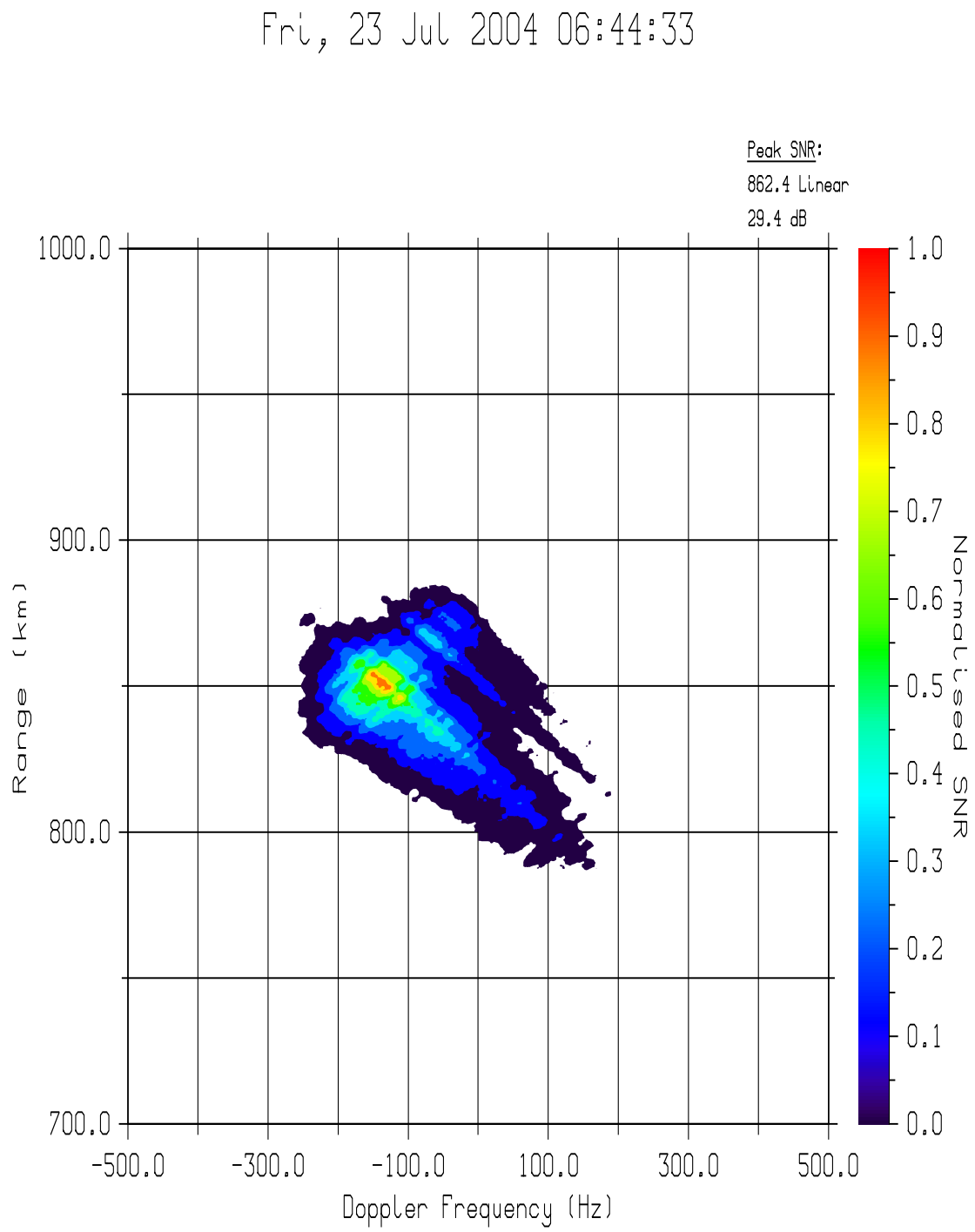

Figure 5.2: Range-Doppler map of Type I echo from 23 July, 2004 06:44:33 UT.

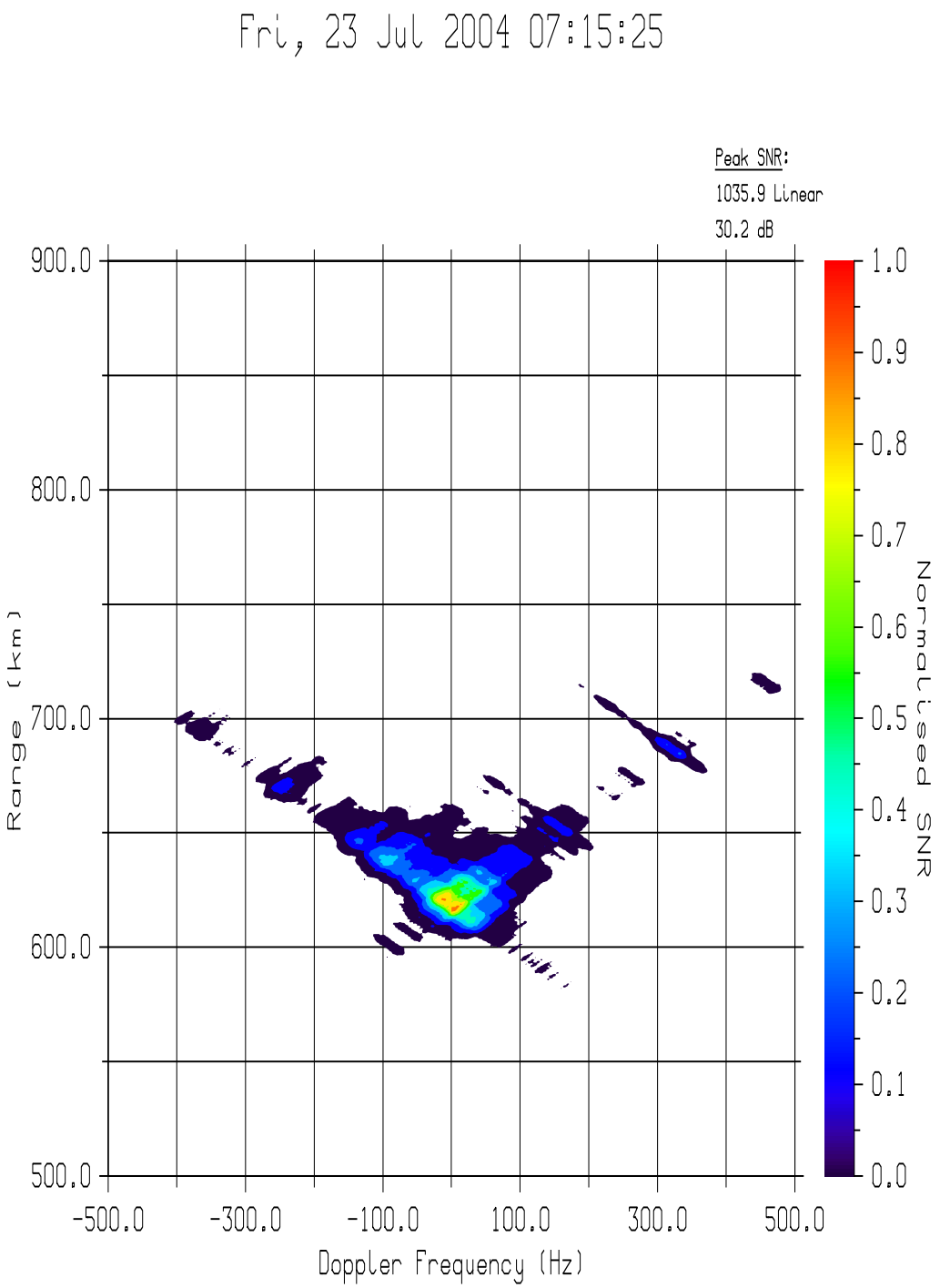

Figure 5.3: Range-Doppler map of Type II echo from 23 July, 2004 07:15:25 UT.

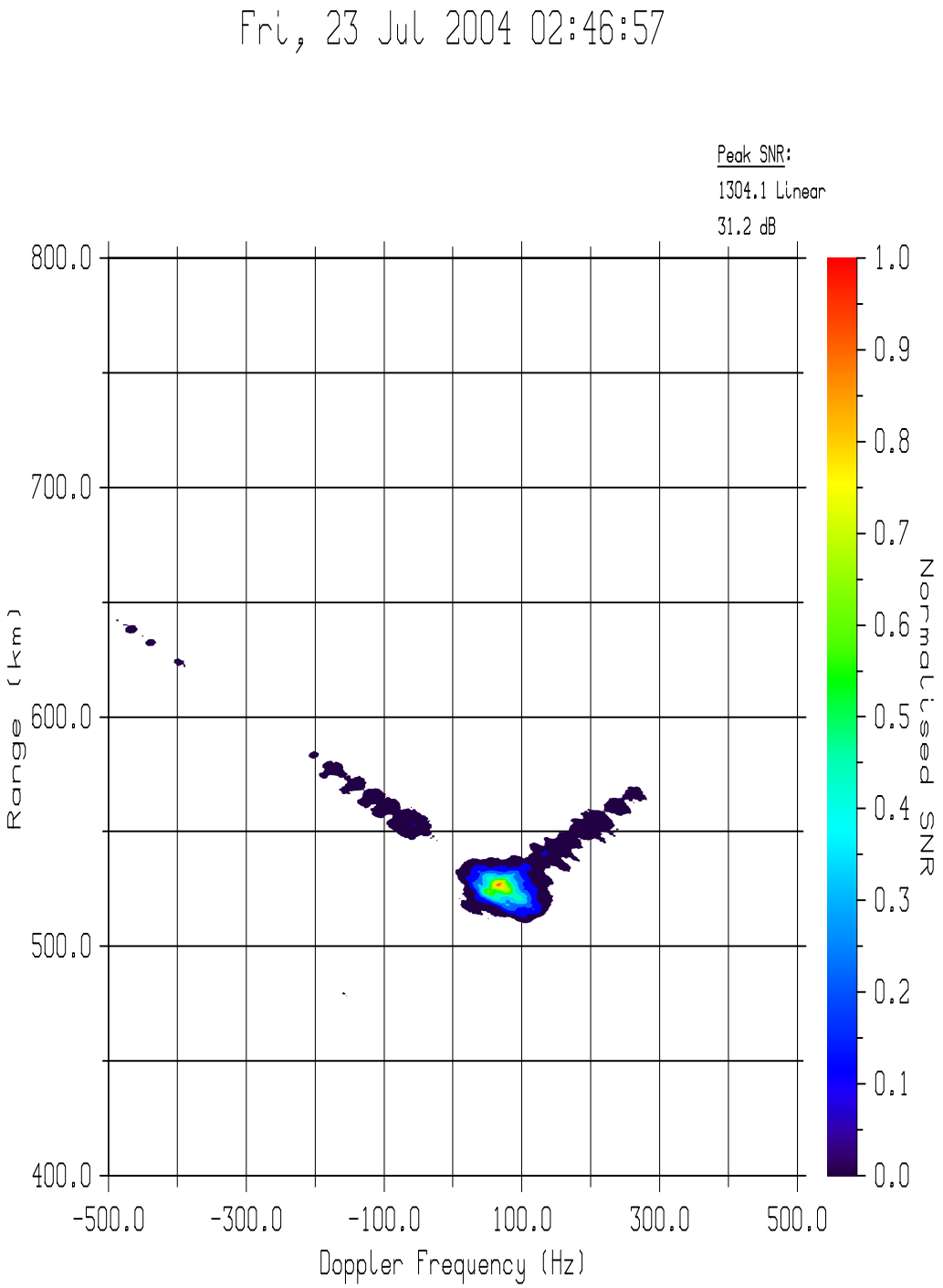

Figure 5.4: Range-Doppler map of Type III scatter from 23 July, 2004 02:46:57 UT.

growing. While these two ranges are active, the radar receives scatter from two distinct ranges. A CW system would detect a two peaked spectrum when both ranges are producing scatter, but it would be unable to determine the ranges of the peaks and would be unable to determine that the two peaks were even coming from distinct ranges unless it has a polarimeter system as discussed by *Hussey* [1994]. A pulsed radar system would be able to clearly identify the ranges, but not with the spectral resolution of the FMCW system. For the FMCW system, when scatter is received from distinct regions, ghost targets appear in the range-Doppler map as discussed in Appendix A. This apparent ambiguity is easily overcome, however, by understanding how ghost targets are formed.

Ghost targets are created when targets from multiple ranges are present. This can be demonstrated graphically by returning to Figure 2.4 in section 2.1, and adding a second target as illustrated in Figure 5.6. Each line in Figure 5.6 represents all possible range and Doppler frequency values that could have generated a particular beat frequency. The solid blue lines represent all possible values of range and Doppler shift if the up sweep was considered by itself and the dashed red lines represent all possible range and Doppler frequency values if the down sweep was considered by itself. It is only at the intersections of the up sweep possibilities with the down sweep possibilities that the range and Doppler shift of the target is determined.

The addition of a second target creates four points of intersection in the range-Doppler map (relative to one point of intersection in Figure 2.4). Since only two targets are present, it is known that two of these points of overlap represent ghost targets. Further examination reveals that the four points exist in pairs and only two distinct possibilities are represented. The target that produced the bottom blue line in the up sweep can only have a real intersection with one of either the bottom dashed red line at intersection 4 or the upper dashed red line at intersection 3. If the true intersection is number 4, then the other blue line must correspond to the other red line at intersection 2 (i.e. both targets in the up sweep can not correlate with the same signal in the down sweep). If the bottom solid blue line corresponds to the upper dashed red line at intersection 3, then the other solid blue line must correspond

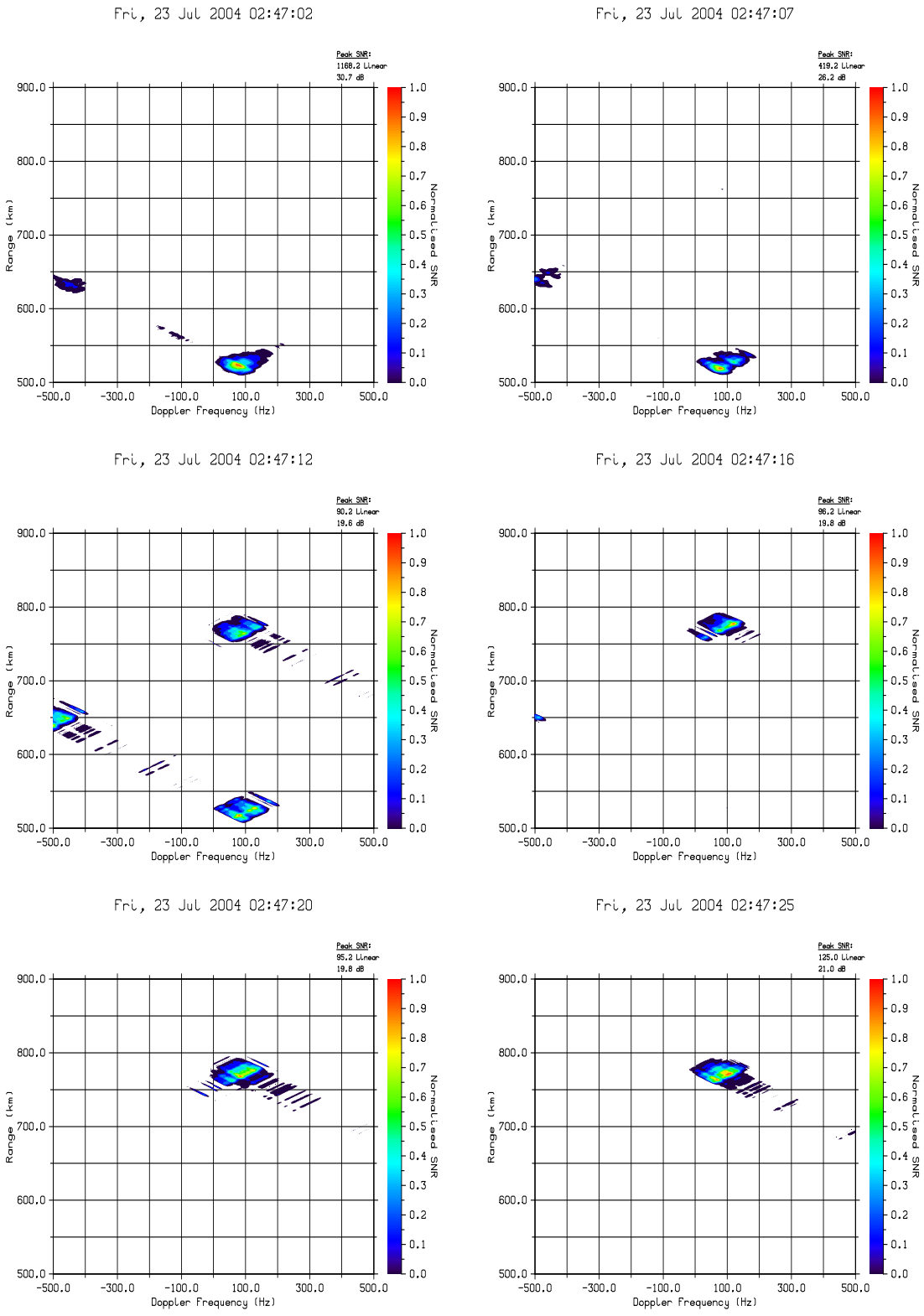

Figure 5.5: Range-Doppler map of scattering event with multiple scattering regions starting at 23 July, 2004 02:47:02 UT.

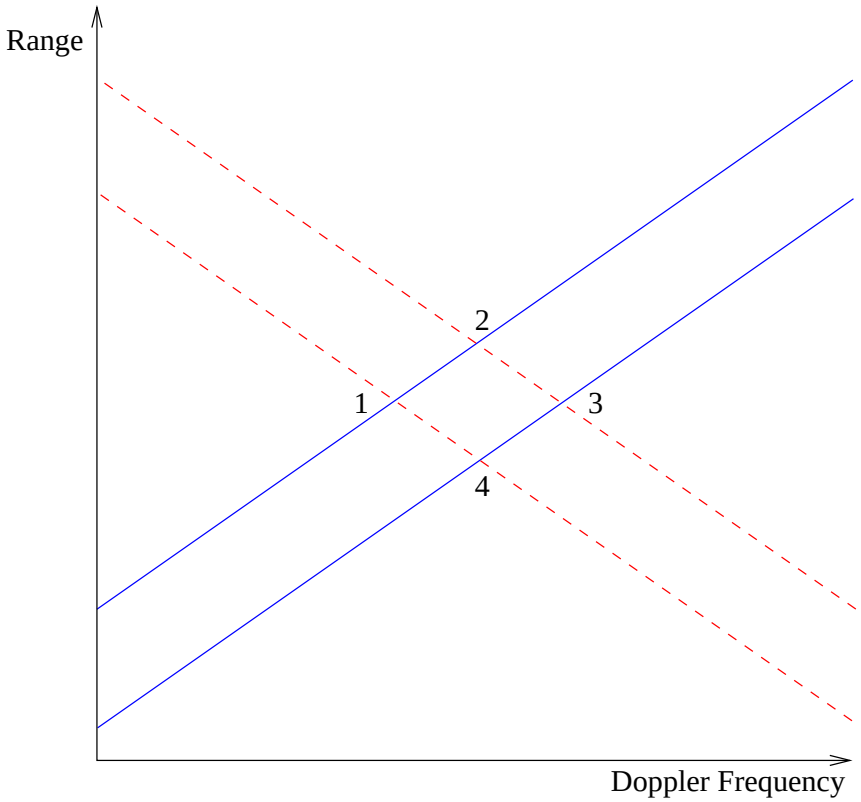

Figure 5.6: If two targets are present 4 points of intersection are present in the range-Doppler map. See text for details.

to the lower dashed red line at intersection 1. In general, this pairing of top and bottom (intersections 2 and 4) or left and right (intersection 1 and 3) always exists, and the identification of one point of intersection as a real target also determines the second point. Real targets such as those in Figure 5.5 give information that indicate which pair is real and which pair is a ghost target. The relative spectral shapes of the target (both magnitude and Doppler spectrum) as well as the temporal information about how the spectrum forms over time gives enough information to identify most targets. Figure 5.5 illustrates how the relative magnitude and the temporal information allows for identification of the top and bottom targets as the real targets.

5.5.1 Resolving ghost targets

It is possible to eliminate the ghost targets by removing the signal(s) from "other" ranges before the convolution process. For example, the data from 02:47:12 UT in Figure 5.5 (middle left) have two distinct scattering ranges producing two distinct regions in the range-Doppler map with approximately equal signal strength. As a result, the ghost targets are also of equal strength. Figure 5.7 gives another view of the data collected at this time. The top plot in Figure 5.7 is the spectral SNR over the practical bandwidth for the up sweep. The x-axis represents frequency in Hz. The second plot is the spectral SNR over the practical bandwidth for the down sweep. The bottom plot is the convolution as a function of range. The spectral SNR reveals the two distinct scattering regions labelled RI and RII in each plot. The convolution gives a clear indication of the range for each of the targets as well as an indication of the range that ghost targets appear. The strength of the convolution is much stronger for the ghost targets because both ghost targets appear at the same range (or, put another way, both targets from the up sweep are correlated with the "other" target from the down sweep).

Once two distinct regions are identified, they can be analysed individually. The result of analysing one target at a time by removing the "other" target is illustrated in Figure 5.8. Two convolutions were performed, one with region RII removed analysing region RI by itself, and another with region RI removed, analysing region RII by itself. The results of the two convolutions were merged to give the range-Doppler map illustrated in Figure 5.8. Clearly the ghost targets have disappeared; however, the excellent range and Doppler information of both targets is still available. Currently, this process must be done by hand, but it could be automated. Automating this process is left for future development.

Sometimes the spectrum of the targets overlap (i.e. do not form two distinct regions in the spectral SNR), and the problem of separating range and Doppler information becomes much more difficult (if not impossible). The study of such events is beyond the scope of this thesis and is a research project of its own.

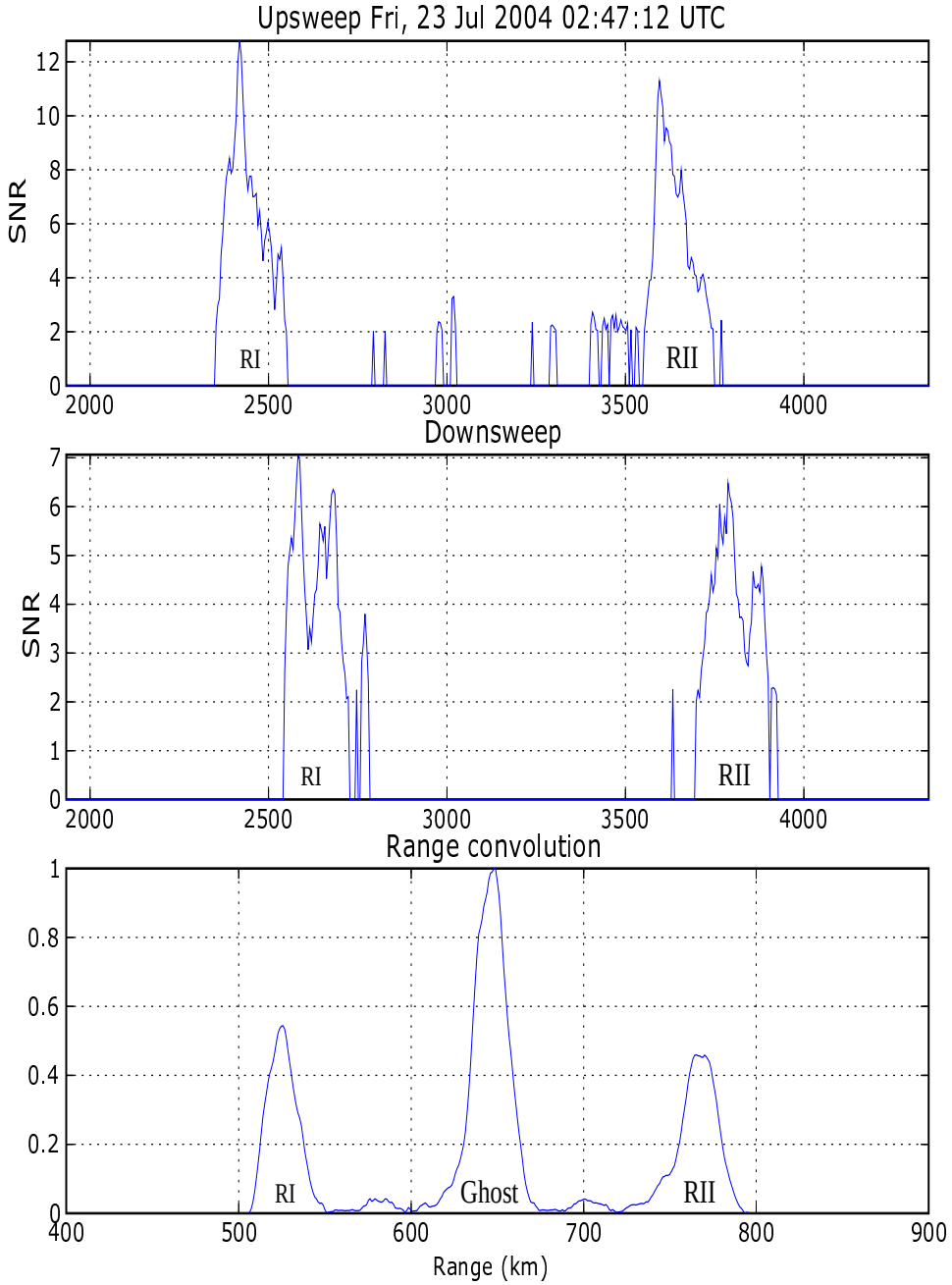

Figure 5.7: Illustration of the spectrum from the up sweep (top), down sweep (middle), and the resulting convolution (bottom) when two distinct targets are detected.

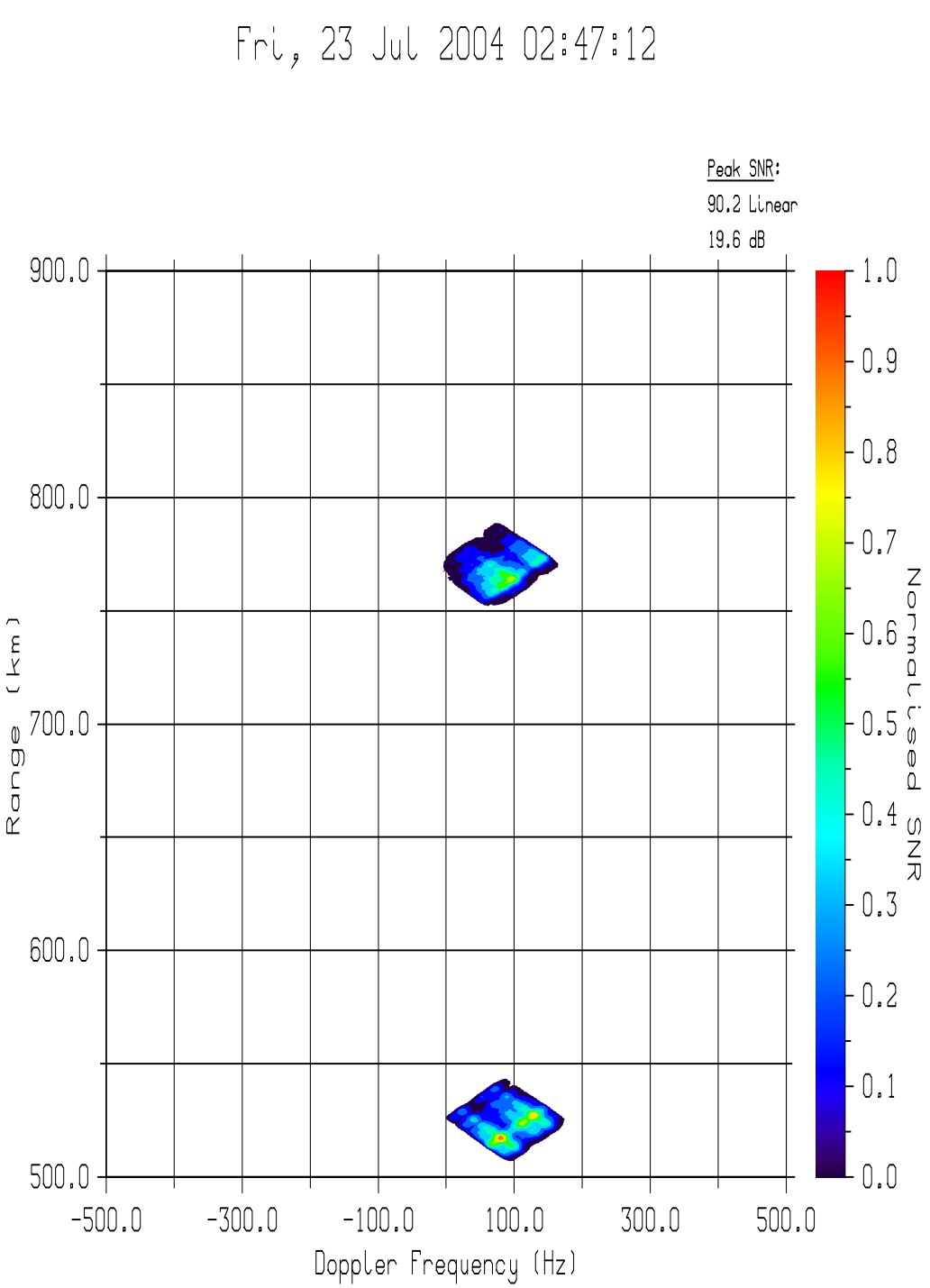

Figure 5.8: Range-Doppler map using the independent analysis of both regions from 23 July, 2004 02:47:12 UT.

5.5.2 Modulation techniques which resolve ambiguity

Ambiguous signals such as ghost targets may also be resolved by using FM waveforms which are able to extract additional information from the data collected. Ambiguity arises because the radar is measuring both range and Doppler information in the frequency domain. If information could be obtained that reduces the ambiguity of either the range or Doppler measurement, it would concurrently reduce the ambiguity of the other parameter. For example, if the Doppler information about a scattering target could be obtained without ambiguity, information about range would also become unambiguous.

By using different FM techniques, the ambiguity in the range and Doppler information can be reduced. Figure 5.9 gives two alternative modulation techniques that would improve the information obtained by an FMCW radar. The top diagram illustrates triangular modulation with the addition of a CW "modulation" section. The technique of adding a CW section to the triangular sweep will be referred to as the sweep-CW technique. The additional CW section would extract only the Doppler information without any of the range information. It would give the complete Doppler spectrum from all targets. With this information, the range-Doppler maps would be significantly improved. Ghost targets would be eliminated because the Doppler shift of these targets would be known to be invalid. Since the Doppler information is known the range information becomes clearer and can be extracted from the sweep portion of the sweep-CW technique.

However, the radar setup for this project was monostatic and with all monostatic CW radar systems the problem of feed-through is significant (see Section 3.2.3 for a discussion of feed-through). In order to overcome the problem of feed-through, the hardware of the receiver was designed to block DC and low frequency output from the mixer. As a result, direct mixing of the transmitted and received signal of a CW system was not possible since low Doppler shift targets would be undetectable. In order to try to capture a CW signal, a local oscillator signal offset by at least 1 kHz from the transmitted signal would have to be used in order to allow the receiver

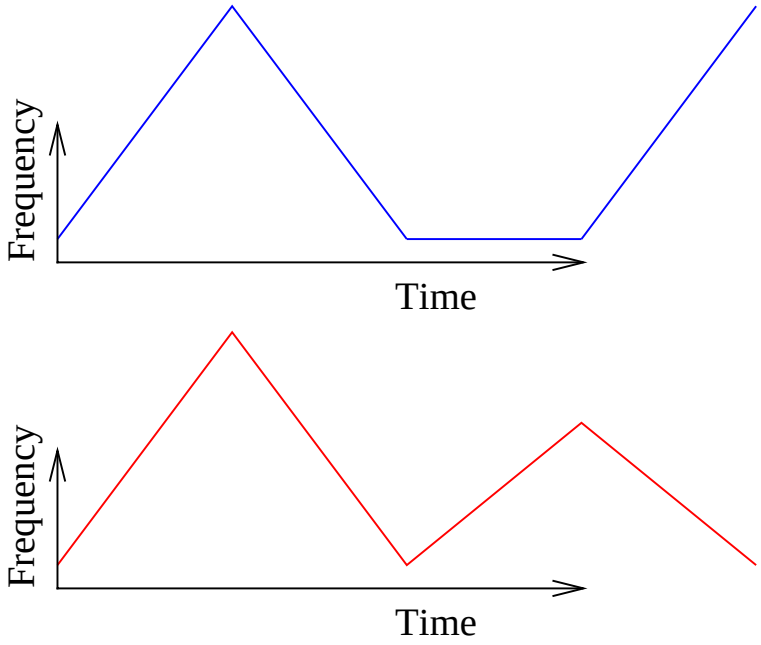

Figure 5.9: Modulation alternatives to triangular frequency modulation. The top diagram illustrates a sweep-CW technique which includes a CW "modulation" section with a triangular frequency sweep. The bottom diagram illustrates a two-BW sweep technique which uses two triangular frequency sweeps with different bandwidths.

to capture E-region signals. Simple measurements using a local oscillator signal with a 1 kHz frequency offset from the transmitted signal produced feed-through that completely swamped the receiver. Therefore, a sweep-CW technique was not feasible for this system and was not used.

If the radar system was designed to handle or eliminate the feed-through power, a sweep-CW technique would be an excellent method of resolving range and Doppler information. Nonetheless, feed-through is a very significant problem for any monostatic CW radar and designing such a radar system to handle the feed-through power would be very challenging.

The bottom diagram in Figure 5.9 illustrates a waveform that consists of two triangular frequency sweeps with different bandwidths. This technique will be referred to as the two-BW sweep technique. A two-BW sweep technique provides a reduction in the ambiguity of the range and Doppler information, and it works within the restrictions of the receiver hardware of this project. Two frequency sweeps produce two independent measurements of Doppler and range information. These independent measurements can be compared, and, as a result, the ambiguity between range and Doppler information is significantly reduced.

If distinct scattering regions are present, ghost targets produced during the convolution process can be removed since the Doppler shift components of the ghost target would necessarily be different between the two sweeps. True targets would have the same range and mean Doppler shift in each sweep. Therefore, ghost targets, at least ones generated by targets that are separated by enough range, would be eliminated.

Another issue that could be improved is the extraction of range and Doppler information from targets that produce scatter from multiple range cells. Recall from Section 3.1 that range resolution is directly proportional to bandwidth. A target that extends over several range cells will have a broadened Doppler spectrum (each range cell corresponds to a frequency cell). However, if that same target is detected by a second frequency sweep with a narrower bandwidth, it would generate a narrower Doppler shifted signature. Therefore, the difference in the Doppler shifted width of

the spectrum between the two frequency sweeps gives a measure of the spatial extent of the scattering region.

5.6 Implementation of two-BW sweep technique

With two frequency sweeps of different bandwidth, there were four separate data segments to analyse. These segments are the up sweep of the large frequency sweep, the down sweep of the large frequency sweep, the up sweep of the small frequency sweep, and the down sweep of the small frequency sweep. The bandwidth of the second frequency sweep was chosen to be half that of the large frequency sweep (80 kHz), and the period was chosen to be the same for both sweeps. This made the range resolution half that of the original waveform but kept the Doppler resolution the same. This format was chosen to allow differences in the received spectrum of each sweep to be easily attributed to measurements associated with range.

Implementation meant changing the program that controlled the radar. Instead of just changing the value of the sweep flag in the DDS, the AD9954 needed to be programmed to perform two frequency sweeps of different bandwidth. Two bandwidth frequency sweeps required changing the upper frequency variable in the AD9954 between sweeps. Therefore, the DDS signal was swept up and down in frequency; then the upper frequency variable was changed and the triangular frequency sweep was repeated.

5.6.1 Two-BW data

Figures 5.10 to 5.12 are the range-Doppler maps produced by using the two-BW sweep technique. These figures illustrate the advantage of using a two-BW sweep technique. Figure 5.10 is the range-Doppler map produced using the wide bandwidth modulation data only, Figure 5.11 is the range-Doppler map using the narrow bandwidth modulation data only, and Figure 5.12 is the range-Doppler map formed by combining the information from the two frequency sweeps.

Figure 5.10 is the range-Doppler map generated using data from the 160 kHz

single bandwidth sweep only. It is essentially equivalent to the data collected using the single bandwidth sweep. In the diagram, a target appears at a range of ∼500 km with a mean Doppler shift of ∼140 Hz. There is also a target at a range approaching 700 km with a mean Doppler shift of 100 Hz. At ∼600 km ghost targets are visible that are typical when two distinct scattering ranges are active.

Figure 5.11 is the range-Doppler map generated using data from the 80 kHz bandwidth sweep only. As the range resolution in this diagram is coarser the targets appear to cover a larger range extent. The target at a range of ∼500 km with a Doppler shift of ∼140 Hz is present as is the target at a range close to 700 km. The ghost targets are also present at a range of ∼600 km once again, but note that the Doppler shift of the ghost targets is significantly different from that of the ghost targets in Figure 5.10. In Figure 5.10 the ghost targets have a mean Doppler shift of ~ -300 Hz and ∼500 Hz. In Figure 5.11 the same targets have a mean Doppler shift of ~ -100 Hz and ∼350 Hz. This difference in Doppler shift of the ghost targets allows them to be easily distinguished.

Figure 5.12 illustrates the result of combining the information from both sweeps. Note that the ghost targets have disappeared but the two main scattering regions at ∼500 km and ∼700 km are still visible. As such, the two-BW sweep technique can be used to extract range and Doppler information from multiple targets.

5.6.2 A Type IV event

Finally, a Type IV event was observed with the FMCW radar system using the two-BW sweep technique. No physical interpretation of this event is discussed as it is beyond the scope of this thesis; however, it is expected that the high temporal and spatial abilities of the FMCW radar will further better understanding of Type IV event in the future.

All of the radar data presented so far have been collected using ∼4.5 s integration time (10 up and down sweeps or 5 complete two bandwidth sweeps). This event, however, lasted a few minutes. In order to view this event in a reasonable number

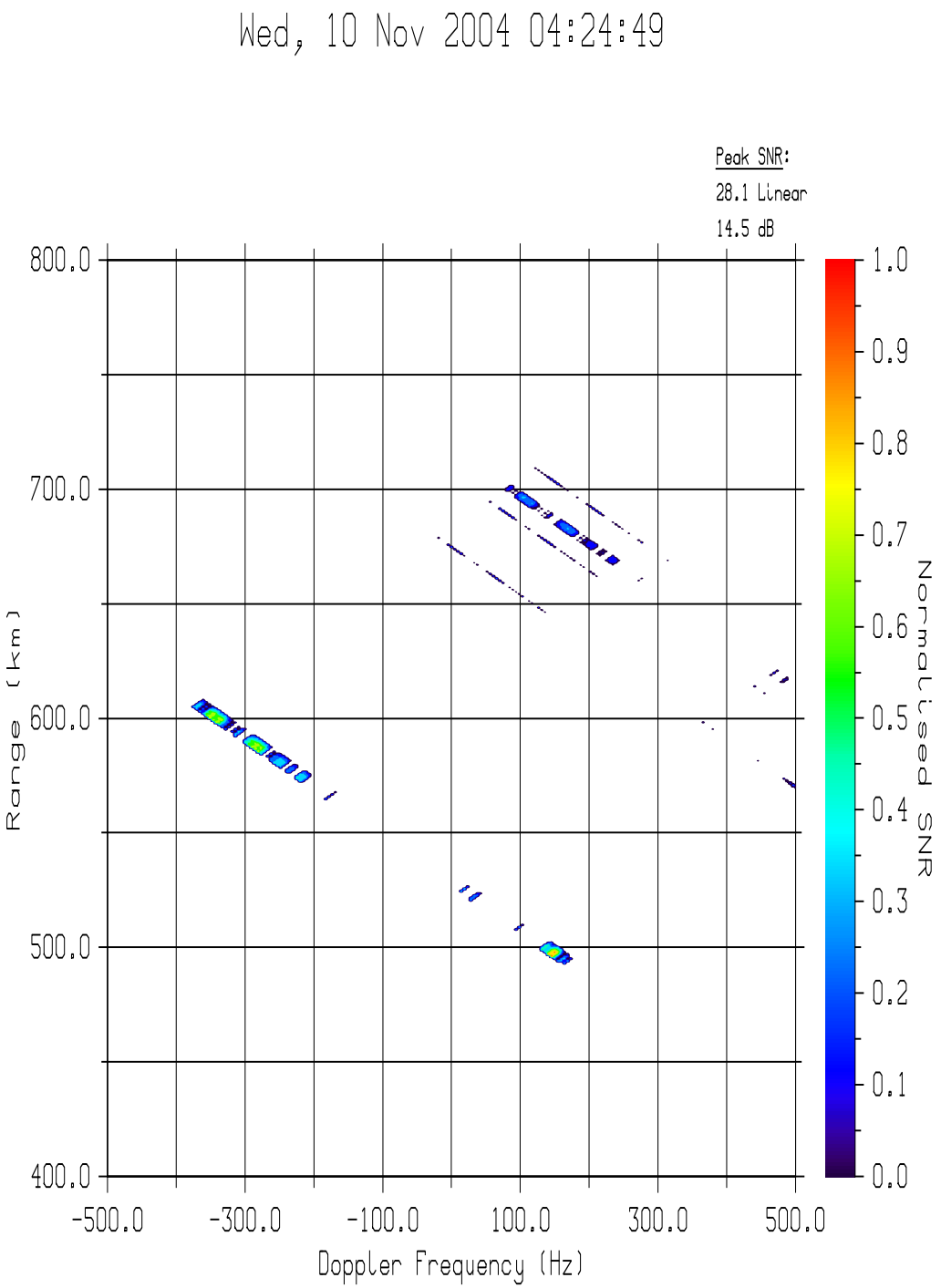

Figure 5.10: Range-Doppler map from 10 November, 2004 04:24:49 UT using only 160 kHz bandwidth data of the two-BW sweep technique.

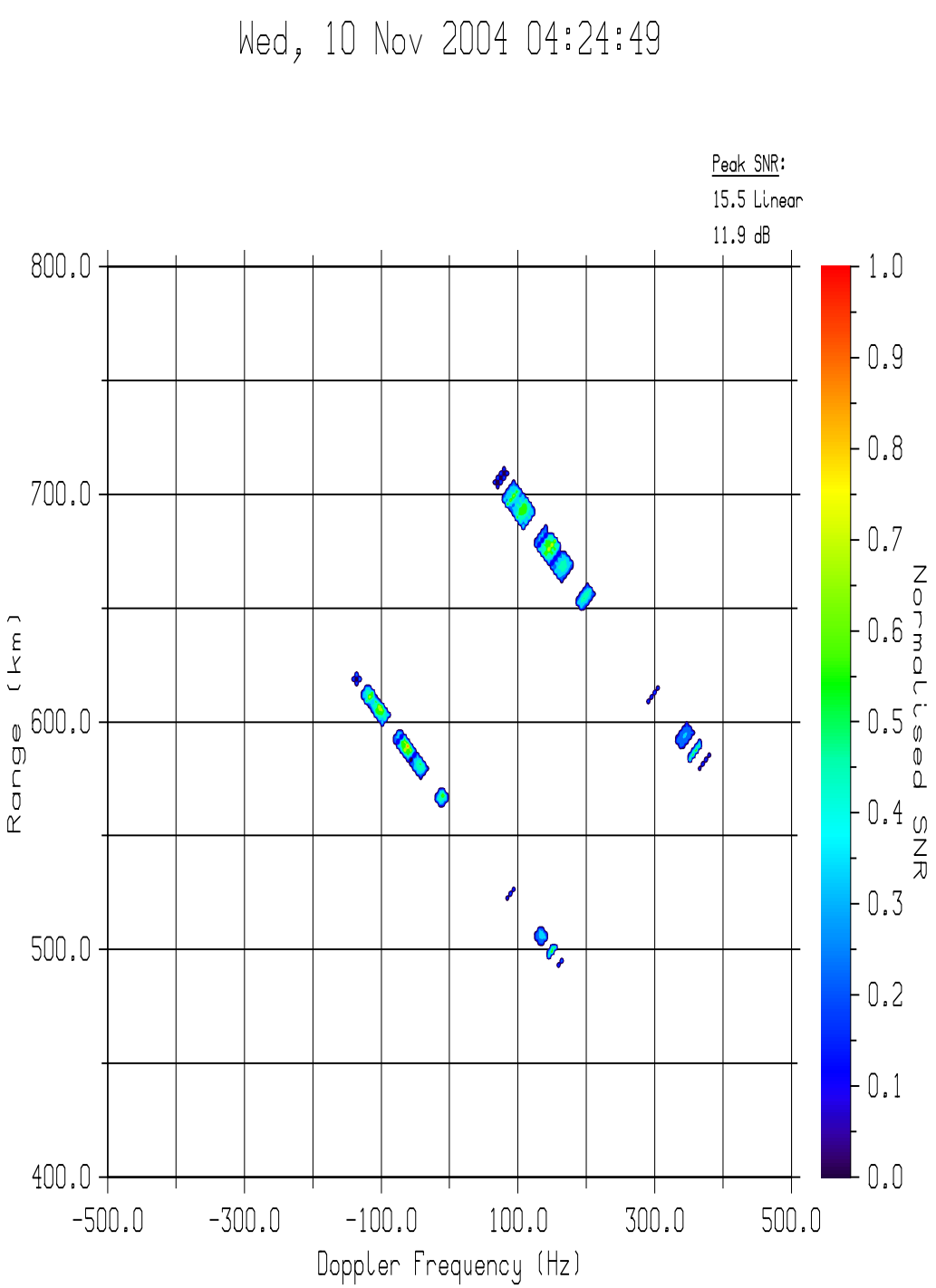

Figure 5.11: Range-Doppler map from 10 November, 2004 04:24:49 UT using only 80 kHz bandwidth data of the two-BW sweep technique.

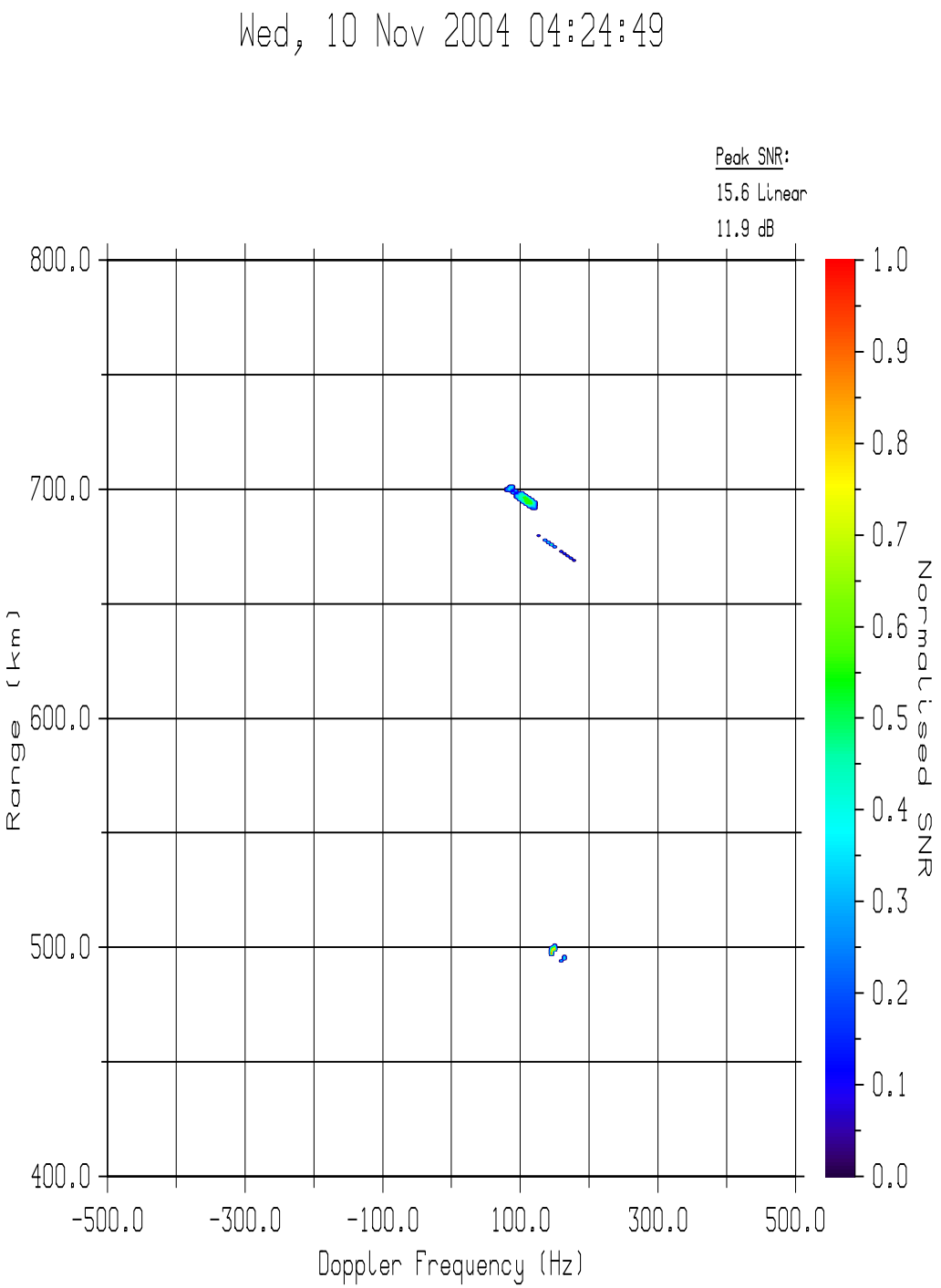

Figure 5.12: Range-Doppler map from 10 November, 2004 04:24:49 UT using 160 kHz and 80 kHz bandwidth data of the two-BW sweep technique.

of diagrams, the integration time was increased. Figures 5.13 illustrate the Type IV event as it evolved over time in ∼18 s intervals (integration of 20 complete two bandwidth sweeps). These diagrams indicate a consistent region of scatter at a range of just under 600 km and a mean Doppler shift between 350 and 400 Hz. This very high Doppler shift corresponds to a mean phase velocity of 1050 to 1200 m/s.

In the range-Doppler maps of Figure 5.13 there is an indication of scatter at 625 km with a mean Doppler shift of 100 Hz. Figure 5.14 shows the next series of range-Doppler maps associated with this event. They reveal that once the Type IV signature decays, there is indeed a scattering region at ∼625 km with a mean Doppler shift of ∼100 Hz. The scattering region appears to move towards the radar until it is less than 500 km range.

This event highlights the excellent range and spectral resolutions which can be simultaneously obtained with the FMCW radar. This is an improvement over CW radars which have equally (or better) spectral resolution, but poor spatial (range) resolution and pulsed radars which typically have relatively poor spectral discrimination and resolution, but excellent spatial resolution. Further, the high Doppler shift observed at the line of sight ranges presented for this Type IV event observed by the FMCW radar could be difficult for a pulsed radar to detect depending on its configuration.

As such, the main reason for developing an FMCW coherent E-region radar was to simultaneously obtain high temporal and spatial resolutions for the study of E-region phenomena. This has been achieved with the FMCW radar system presented in this thesis.

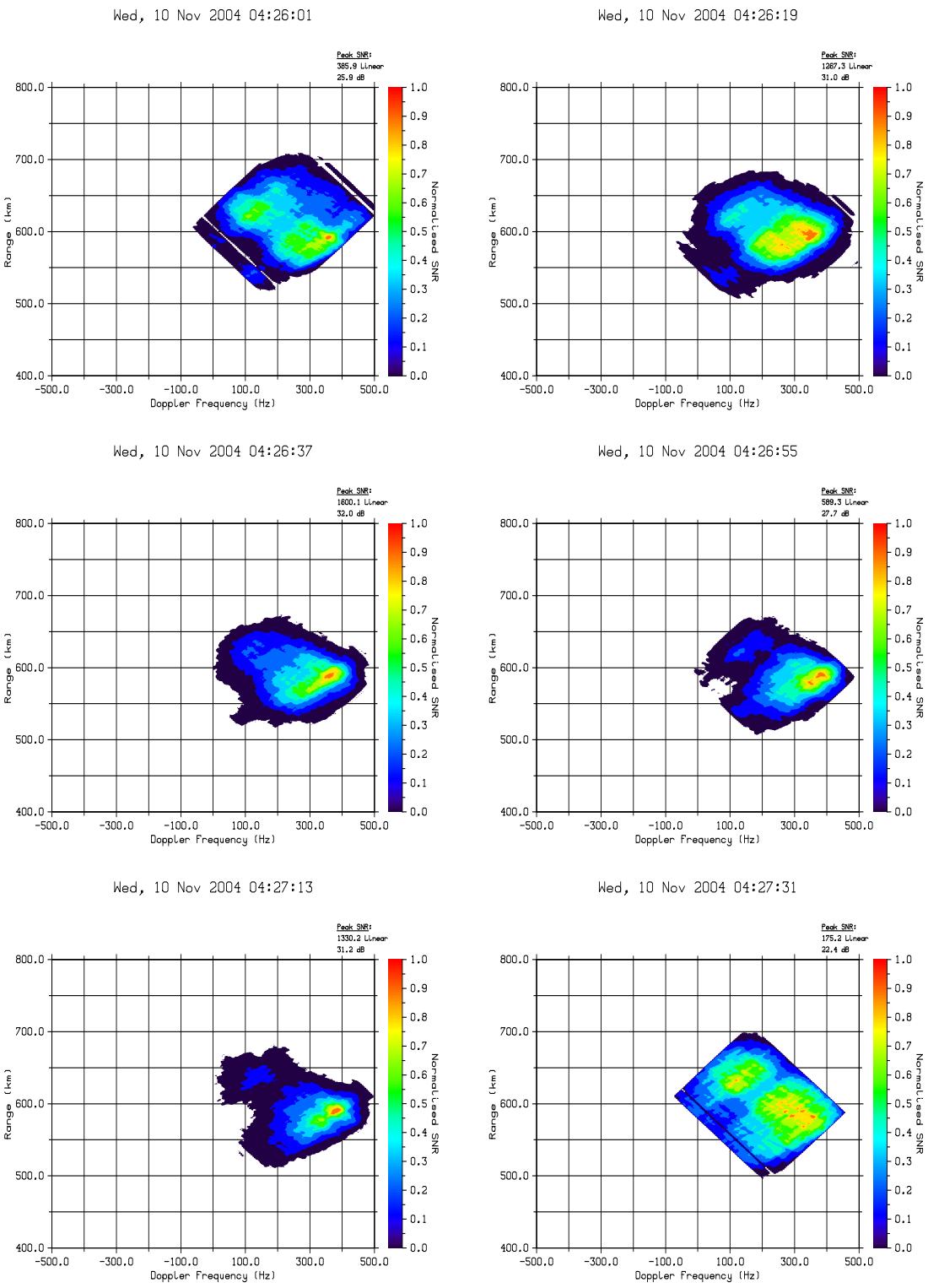

Figure 5.13: Starting at 10 November, 2004 04:26:01 UT, a series of range-Doppler maps generated using the two-BW sweep technique and averaging over 18 s time intervals. A Type IV event is present at a range of ∼600 km.

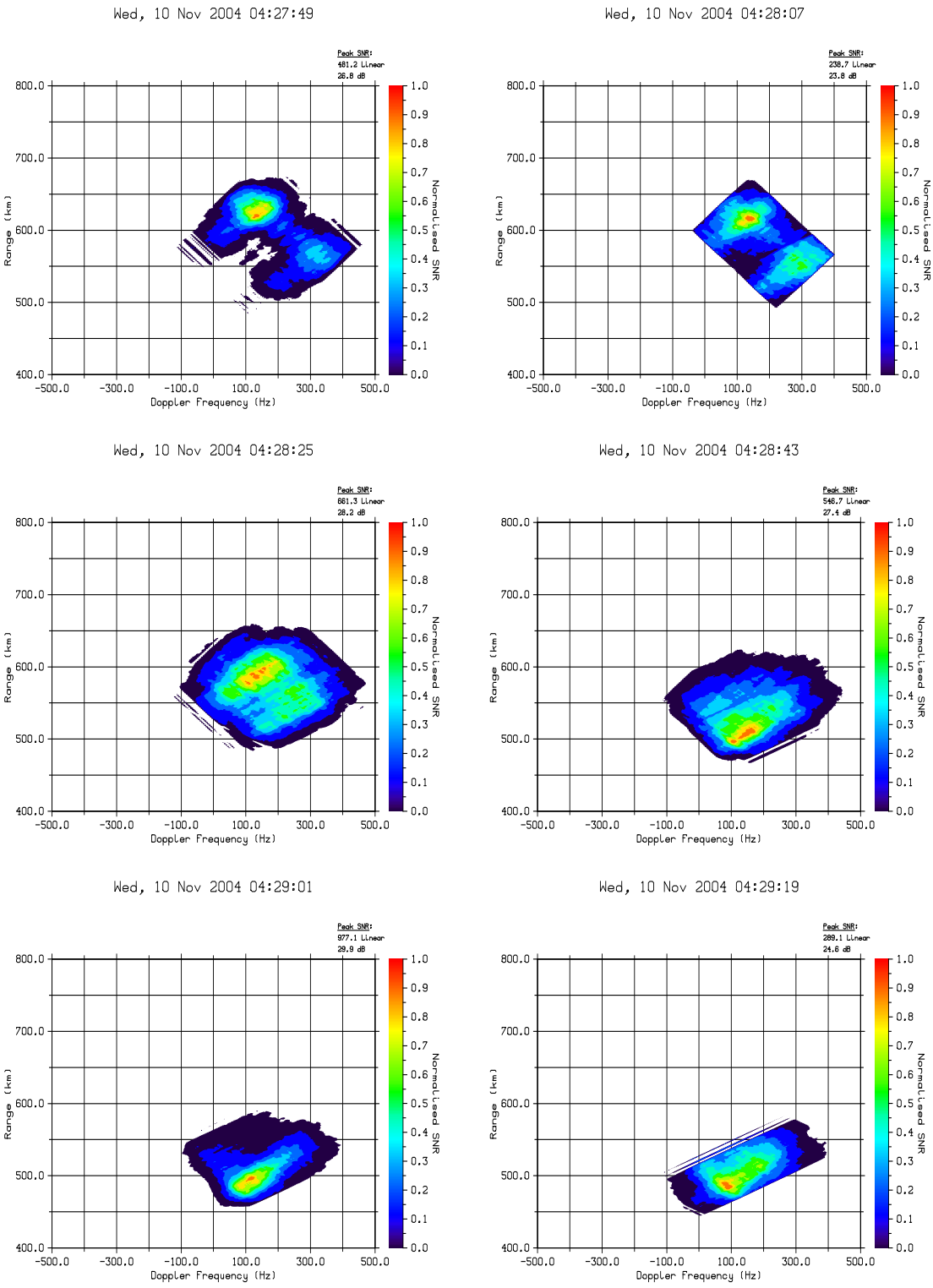

Figure 5.14: A Type IV event decays with time and a Type I event grows with time at a range of ∼600 km. The location of the Type I event then appears to move towards the radar.

Chapter 6

Summary and discussion

FMCW techniques offer a simple and effective method for studying E-region coherent backscatter. For this thesis, an FMCW radar was designed, constructed and operated. The radar provided a monostatic radar setup with Doppler frequency resolution comparable to that of a CW system and range resolution comparable to a pulsed radar system.

The design required analysing different types of FMCW techniques and determining their suitability for E-region studies. The use of triangular frequency modulation techniques, as well as the data analysis techniques that were developed for this thesis, allowed range and Doppler information to be extracted from coherent backscatter from the E-region. Improvements to the triangular frequency modulation technique were also examined. A two bandwidth triangular frequency modulation technique offers opportunities for significant improvements over a single bandwidth triangular modulation scheme.

The availability of DDS technology capable of directly generating the modulated waveform simplified the design of the FMCW radar. DDS lends significant flexibility to the transmitter system as indicated in Chapter 5 where it was demonstrated that a new FM waveform could be implemented with changes only to the software that controlled the radar. The ability to experiment with different FM waveforms is a valuable tool for future work. Range, frequency, and temporal resolution are dictated by the parameters of the modulation. Only a few different modulation parameters were chosen for this project. Future work can experiment with improving (or relaxing) the resolution of different parameters depending on the goals of the radar. These experiments are limited only by the usable bandwidth of the receiver.

The data collected by the radar verified that it would be an effective tool for the study of E-region coherent backscatter. In the first few months of operation, each of the four standard signatures from the E-region, Type I to Type IV, were observed. E-region signals with Doppler shifts between 0 and ±500 Hz and line of sight ranges between ∼400 and ∼1000 km were observed with a frequency resolution of ∼5 Hz and a range resolution of ∼1 km.

The radar constructed for this thesis demonstrates that FMCW techniques will be a useful tool for E-region radar studies. FMCW techniques provide detailed measurements of E-region backscatter and, as such, will improve understanding of the plasma processes that produce the radar echoes. The remainder of this chapter will be dedicated to suggestions for future work.

6.1 Improvement of data analysis

Chapter 5 presented different types of auroral E-region coherent backscatter events that were observed during the operation of the radar. It was difficult to separate range and Doppler information from some of the more complicated signals. In particular, Section 5.5 discussed ghost targets and the challenge of resolving multiple targets. Section 5.5.1 presented a technique where targets were matched between the up sweep and the down sweep and each matched target was analysed separately. The example given in Section 5.5.1 required identifying the targets by hand, but it may be possible to automate that process. Pattern matching algorithms may be able to match targets between the up sweep and down sweep. A few key concepts will be mentioned here to assist future work in this area.

Shape

The shape of the spectrum will aid in the matching of targets. In theory, the spectrum of a target should be reversed between the up sweep and the down sweep. The relative magnitude of each target should also be consistent between sweeps. The strongest target in the up sweep will correspond to the strongest target in the down sweep.

The value of spectral SNR of a target is often different between the up sweep and the down sweep, but the relative magnitude should be consistent. A matching algorithm could normalise the spectral SNR and match targets between the up sweep and down sweep based on relative magnitude and shape.

Minimum percent difference

Another selection technique that may be able to take advantage of the normalised spectrum would be an algorithm that compared the relative difference between two points in the convolution. Any point in the convolution that exceeds a certain minimum percent difference is ignored. Instead of needing to match individual targets between the up sweep and the down sweep, it would be easier to just use the relative magnitude at each point in the spectrum as an approximation. This would not eliminate ghost targets completely, but it would reduce them significantly. It would also clean up some of the spectral broadening that comes out of the convolution. Note, however, that as two targets cross in the convolution, even if they are vastly different in normalised magnitude, there will always be parts of the two spectra that are the same magnitude.

Order in the spectrum

An important concept for identifying and matching targets between the up sweep and the down sweep is the order of targets in the spectrum. The order is the same between the up sweep and the down sweep (in theory this may not be true for very high Doppler shifted targets that have a narrow range gap between them). A target at the lowest part of the spectrum will be matched with the target in the lowest part of the spectrum in the other sweep. As such, a reasonable approximation of a matching algorithm would be to perform the convolution and, as a target is matched between sweeps, the target could be removed from future iterations of the convolution.

Temporal clues

Another important concept is the information provided over time. It is rare that a range and Doppler spectrum from a true target is active for only one integration interval. If multiple targets and ghost targets are present, information from previous (and future) analyses will provide valuable information for the current analysis.

6.2 FM waveform improvements

In Chapter 5, a discussion of different FM waveforms was discussed, along with information about the implementation of a two-BW sweep technique. The use of waveforms with more complexity than simple triangular modulation seems necessary for the extraction of range and Doppler information from complex E-region targets. As discussed in Section 5.5.2, the addition of a CW section to triangular modulation would provide significant improvements in the ability of the radar to separate range and Doppler information. The CW section would provide unambiguous measurements of Doppler information providing an excellent basis for extracting the range information from the triangular frequency sweep. Unfortunately, this poses significant technical problems, primarily with feed-through.

The 2-BW sweep technique offers a good alternative FM waveform that works within the limitations of the current hardware design. This technique offers an improvement over the traditional triangular FM techniques and should be used in future implementations of the radar. Data analysis techniques can be developed that take advantage of the extra information provided by the two independent measurements. For example, the difference in the width of the Doppler spectrum can be used to extract the range extent of targets. From that information the Doppler spectrum can be calculated with more accuracy.

The DDS technology of the FMCW radar design makes it easy to experiment with different FM waveforms. Innovative waveform techniques may provide improvements in the radar performance without requiring changes to the physical design of the

radar.

6.3 Design improvements

Improvements in the design of the radar are dependent on available resources. The following provides a starting point for future improvements to the design.

Circulators

Circulators are three terminal devices that direct RF power from one port to another port while isolating the third port from that signal. They are used to direct power from a transmitter to an antenna while isolating a sensitive receiver. They are also able to direct power from the antenna to the receiver. Using circulators in the design would allow all antennas to be used for both transmitting and receiving, and, as such, would increase the sensitivity of the radar. Circulators could provide the same level of isolation between the transmitter and the receiver as currently exists using separate transmit and receive antennas. However, a number of factors may still make this infeasible, including the strength of the reflected power and the problem of feed-through from neighbouring antennas. The addition of reflected power cancellation techniques may be required to make it feasible to use all antennas for transmit and receive.

Interferometry

Interferometry offers the possibility of extracting more information from detected targets. The purpose of interferometry is to measure the phase difference between the signal received from antenna that are a known distance apart. This phase difference provides information about the angle at which the signal was received. Currently, only the line of sight range to a target can be determined. With interferometry, it would be possible to also determine the elevation and azimuth angle of the originating target relative to the direction of the antenna. The radar site used for the construction of the radar has antennas that have been used for interferometry in pre-

vious radars; therefore, it would be feasible to implement interferometry techniques for the FMCW radar.

Digital receiver

Digital receiver technology provides an interesting possibility for this radar. Just as the DDS technology provides a flexible method for generating the transmitted signal, a digital receiver would provide a flexible receiver for data analysis. Unfortunately, the problem of feed-through is, once again, a limiting factor. Many aspects of the current design of the analog receiver were required to reduce feed-through. A digital receiver may not be able to provide the same flexibility. A digital receiver for this radar would need to be able to capture/handle the relatively strong feed-through signal while simultaneously maintaining the sensitivity required to detect the much weaker signal from the E-region. If this capability exists it may provide a powerful tool for digital filtering without the noise and non-linear effects of an analog mixer. Considering the complexities associated with data analysis, a digital receiver would provide significant flexibility. However, the combination of problems associated with feed-through and the significant cost of the digital receiver relative to the analog receiver design, made use of a digital receiver infeasible at this point in the project.

References

Alpha-Power, *Alpha 6 Manual*, Boulder, Colorado.

Brigham, E. O., *The Fast Fourier Transform and its Applications*, Prentice Hall, 1988.

Campbell, R., High performance direct-conversion receivers, *QST*, pp. 19–28, 1992.

Campbell, R., High performance, single-signal direct-conversion receivers, *QST*, pp. 33–40, 1993.

Campbell, R., and B. Kelsey, A next-generation R2 single-signal direct conversion receiver, `http://www.bright.net/~kanga/r2pro/article.htm`, 2003.

Eaves, J. L., and E. K. Reedy (Eds.), *Principles of Modern Radar*, Van Nostrand Reinhold Company Inc., 1987.

Fejer, B. G., J. F. Providakes, D. T. Farley, and W. E. Swartz, Auroral E region plasma waves and elevated electron temperatures, *Journal of Geophysical Research*, *91:A12*, 13,583–13,592, 1986.

Garmin, *GPS 35/36 TrakPak GPS smart antenna technical specifications*, Olathe, Kansas, 1999.

Greenwald, R. A., et al., DARN/SuperDARN: A global view of the dynamics of high-latitude convection, *Space Science Reviews*, *71*, 761–796, 1995.

Gurgel, K. W., G. Antonischki, H. H. Essen, and T. Schlick, Wellen radar (WERA): A new ground-wave HF radar for ocean remote sensing, *Coastal Engineering*, *37*, 219–234, 1999.

Gurgel, K. W., H. H. Essen, and S. P. Kingsley, HF radars: Physical limitations and recent developments, *Elsevier Preprint*, pp. 1–20, 2000.

Gurgel, K. W., H. H. Essen, and T. Schlick, The University of Hamburg WERA HF Radar - Theory and Solutions, First International Radiowave Oceanography Workshop ROW 2001 Proceedings, 2001.

Haldoupis, C., and K. Schlegel, A 50 MHz radio experiment for mid-latitude E-region coherent backscatter studies, *Radio Science*, *28*, 959, 1993.

Haldoupis, C., A. Bourdillon, A. Kambruelis, G. C. Hussey, and J. A. Koehler, 50 MHz continuous wave interferometer observations of the unstable mid-latitude E-region ionosphere, *Journal of Geophysical Research*, *21*, 1589–1600, 2003.

Harris, T., *400 MSPS 14-bit, 1.8V CMOS Direct Digital Synthesizer*, Analog Devices, 2003.

Hussey, G. C., Multivariate linear regression models of agricultural parameters on scattering cross-section, Ph.D. thesis, University of Saskatchewan, 1989.

Hussey, G. C., The Polarization of 50 MHz Auroral Backscatter, Ph.D. thesis, University of Saskatchewan, 1994.

Kelley, M. C., *The Earth's Ionoshpere*, Academic Press, 1989.

Kivelson, M. G., and C. T. Russell (Eds.), *Introduction to Space Physics*, Cambridge University Press, 1997.

Koehler, J. A., C. Haldoupis, K. Schlegel, and V. Virvilis, Simultaneous observations of E region coherent radar echoes at 2-m and 6-m radio wavelengths at midlatitude, *Journal of Geophysical Research*, *102*, 255–266, 1997.

Lind, F. D., Passive Radar Observations of the Aurora, Ph.D. thesis, University of Washington, 1999.

Lutz, M., and D. Ascher, *Learning Python*, 2 ed., O'Reilly Associates, Inc, 2004.

M-audio, *Delta 66 Manual*, Tewksbury, MA, 2003.

Martelli, A., *Python in a Nutshell*, O'reilly Associates, Inc, 2003.

Ortlepp, A., *SAPPHIRE VHF radar system manual*, Institute of Space and Atmospheric Studies, University of Saskatchewan, 1994.

Prikryl, P., J. A. Koehler, and G. J. Sofko, Simultaneous CW radio measurements of meteor and auroral drifts, *Radio Science*, *22*, 271–282, 1986.

Providakes, J. F., D. T. Farley, B. G. Fejer, J. D. Sahr, W. E. Swartz, I. Häggström, A. Hedberg, and J. A. Nordling, Observations of auroral E-region plasma waves and electron heating with EISCAT and a VHF radar interferometer, *Journal of Atmospheric and Terrestrial Physics*, *50*, 339–356, 1988.

Rasshofer, R. H., and E. M. Biebl, A direction sensitive, integrated low cost Doppler radar sensor for automotive applications, *IEEE MTT-S International Microwave Symposium Digest*, pp. 1055–1058, 1998.

Reid, G. C., The influence of electric fields on radar measurements of winds in the upper mesosphere, *Radio Science*, *18*, 1028–1034, 1983.

Rohling, H., and M.-M. Meinecke, Waveform design principles for automotive radar systems, `http://www.smart-microwave-sensors.de/html/publications.html`, 2001.

Skolnik, M. I., *Introduction to Radar Systems*, Second ed., McGraw-Hill Book Company, 1980.

Skolnik, M. I. (Ed.), *Radar Handbook*, Second ed., McGraw-Hill Publishing Company, 1990.

Strauch, R. G., W. C. Campbell, R. B. Chawick, and K. P. Moran, Microwave FM-CW doppler radar for boundary layer probing, *Geophysical Research Letters*, *3*, 193–196, 1976.

Tozzi, L. M., Resolution in Frequency-Modulated Radars, Ph.D. thesis, University of Maryland, 1972.

Appendix A

Convolution

Chapter 4 presented the manipulation of the raw data with the final result the calculation of amplitude SNR across the practical bandwidth. The translation of the SNR data into range and Doppler information was performed using a convolution as discussed in Section 2.2. There were some challenges associated with extracting range and Doppler information from the data collected by the FMCW radar as both the range and Doppler information are measured in the frequency domain. This means that range and Doppler information are intertwined and they are challenging to separate (see Section 2.2). With each up sweep and down sweep of data there is ambiguity between the frequency associated with the range of the target and the frequency associated with the Doppler spectrum of the target.

Section 2.2 gave a theoretical justification for and interpretation of the convolution of the spectral SNR of the up sweep with the down sweep. Figures A.1 to A.4 illustrate how the process of convolution extracts range and Doppler information from the spectral SNR using real data. The diagrams are each composed of 4 plots. The top plot is the spectral SNR for the up sweep with the spectrum reversed as required by a convolution. The second plot from the top is the spectral SNR for the down sweep. The third plot from the top in each diagram illustrates the potential spectrum at that range. It is the result of the multiplication of the up and down sweep spectral SNR. The bottom plot is the result of the convolution up to that point in the process. Each point in the convolution is the normalized sum of the spectrum at that range.

The y-axis for the top 2 diagrams is the amplitude of the spectral SNR. The third diagram is also spectral SNR, but it is the result of the multiplication of the amplitude from the up sweep with the amplitude from the down sweep and, as such, represents the SNR as a power ratio. The x-axis for the top 3 diagrams is frequency in units of Hz. The x-axis for the bottom diagram is range in units of km.

In each consecutive diagram, the up sweep spectrum is moved to the left while the down sweep spectrum is moved to the right. Each step corresponds to a specific range. The up and down sweep spectrum in each diagram indicates the Doppler spectrum if the target that produced that spectrum was assumed to be at that range. If the spectrum in the up and down sweep do not align, no target at that range could have theoretically produced the observed data. The third plot in each diagram illustrates the overlap of the two spectra and provides the possible Doppler spectrum for that range.

Figure A.1 corresponds to a line of sight range of 688.6 km. The point of peak spectral SNR from the down sweep (second plot from the top) is correlated with one of the weak signals from the up sweep (top plot). The multiplication produces a narrow weak (a peak of only 12) spectrum at a mean Doppler shift of about 500 Hz (third plot from the top) and the convolution indicates a weak correlation (bottom plot). The overlap in this case does not represent a real target. The convolution produces what is referred to as a ghost target because its appearance does not represent a real target. In this case, because the relative strengths of the signals are very different, the Doppler shift (500 Hz) is very large compared to what is expected from an E-region signal, and there is a much stronger signal in the up sweep spectrum that likely corresponds with the strong down sweep signal, it can be concluded that this correlation does not correspond to a real signal. It becomes more difficult to identify ghost targets when two targets of equal strength appear at different ranges at the same time.

Figure A.2 illustrates the convolution point at a range of 756.5 km. At this range, the primary peak in the spectral SNR of the up sweep is beginning to overlap and with the peak in the down sweep. The down sweep peak is aligned with weaker scatter in the up sweep and this generates what is essentially another ghost target. This ghost target, however, manifests itself as a blurring of the range and Doppler shift associated with the primary target. The correlation is getting stronger but has not reached its peak.

Figure A.3 illustrates the convolution point at a range of 773.0 km. The peak

value of the convolution is produced at this range. It is clear that the primary peak in each spectrum is aligned. The shape of the spectrum is a good reflection of the true spectrum at that range. The center of the spectrum is $\sim$ 100 Hz and the spectral width is narrow. Based on Figure A.3, it is clear that the target that produced the strongest spectral SNR was centered at this range.

Figure A.4 illustrates the convolution point at a range of 795.3 km. The diagram completes the illustration of this part of the convolution. The peaks of each spectrum have passed and the value of the convolution is negligible. Note that the overall shape of the convolution is a good indication of the range of the target.

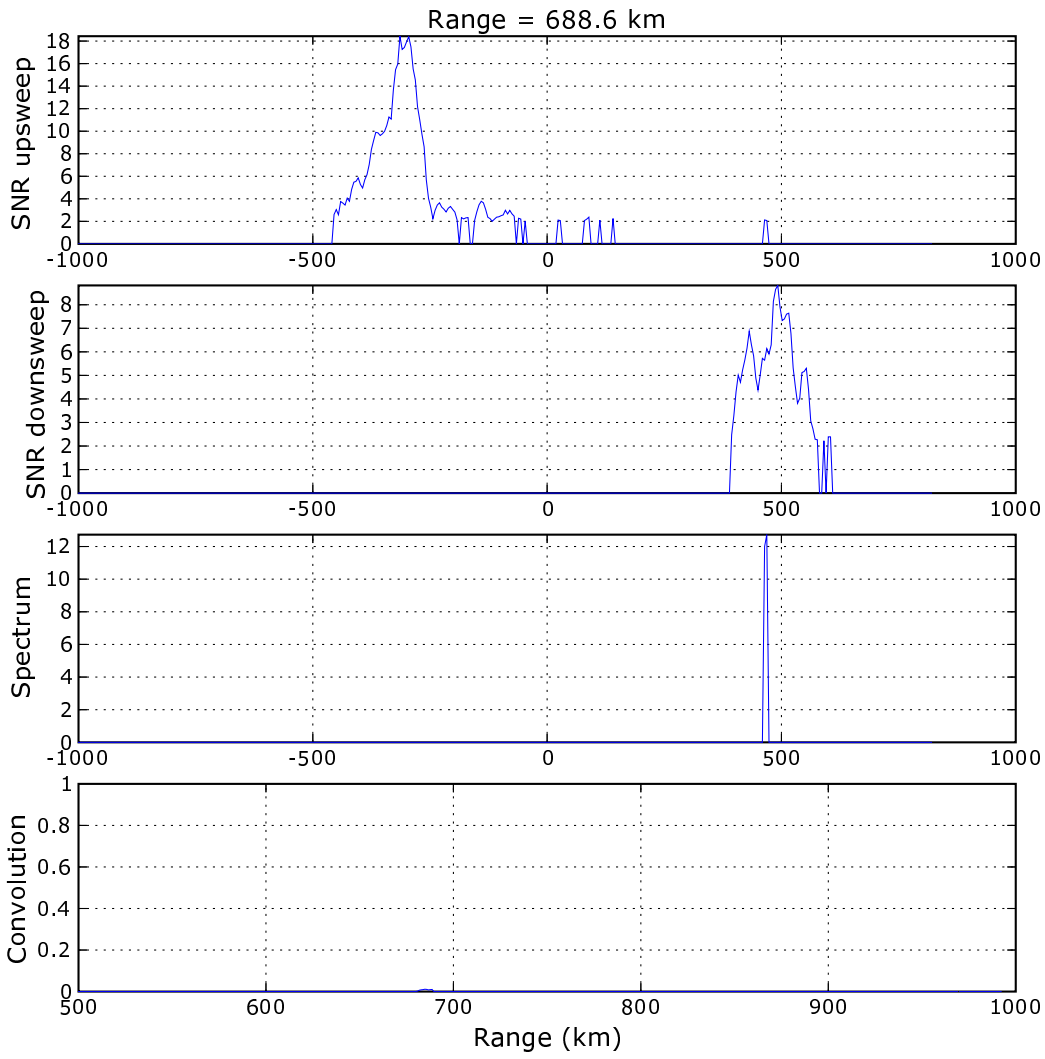

Figure A.1: Illustration of convolution. The data represents the convolution at a range of 688.6 km.

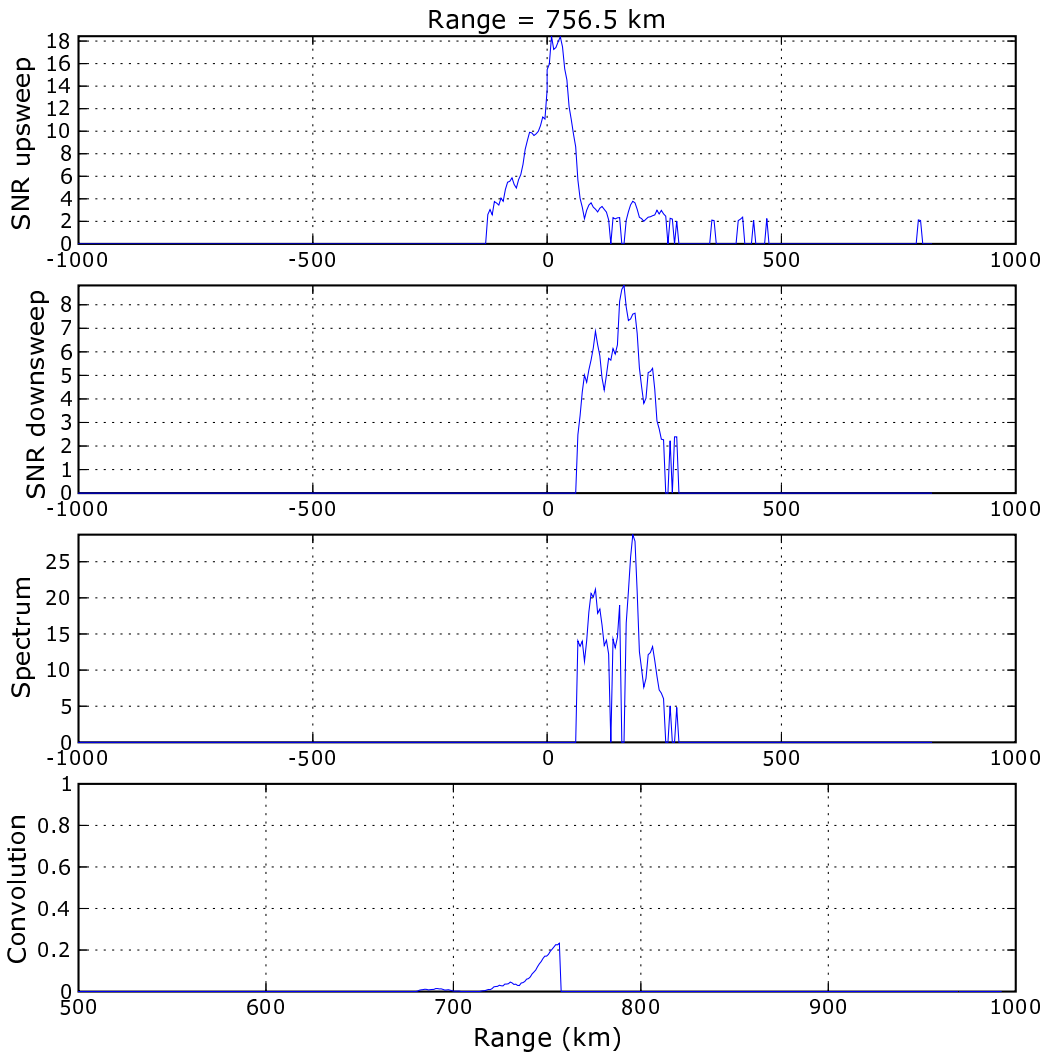

Figure A.2: Illustration of convolution. The data represents the convolution at a range of 756.5 km.

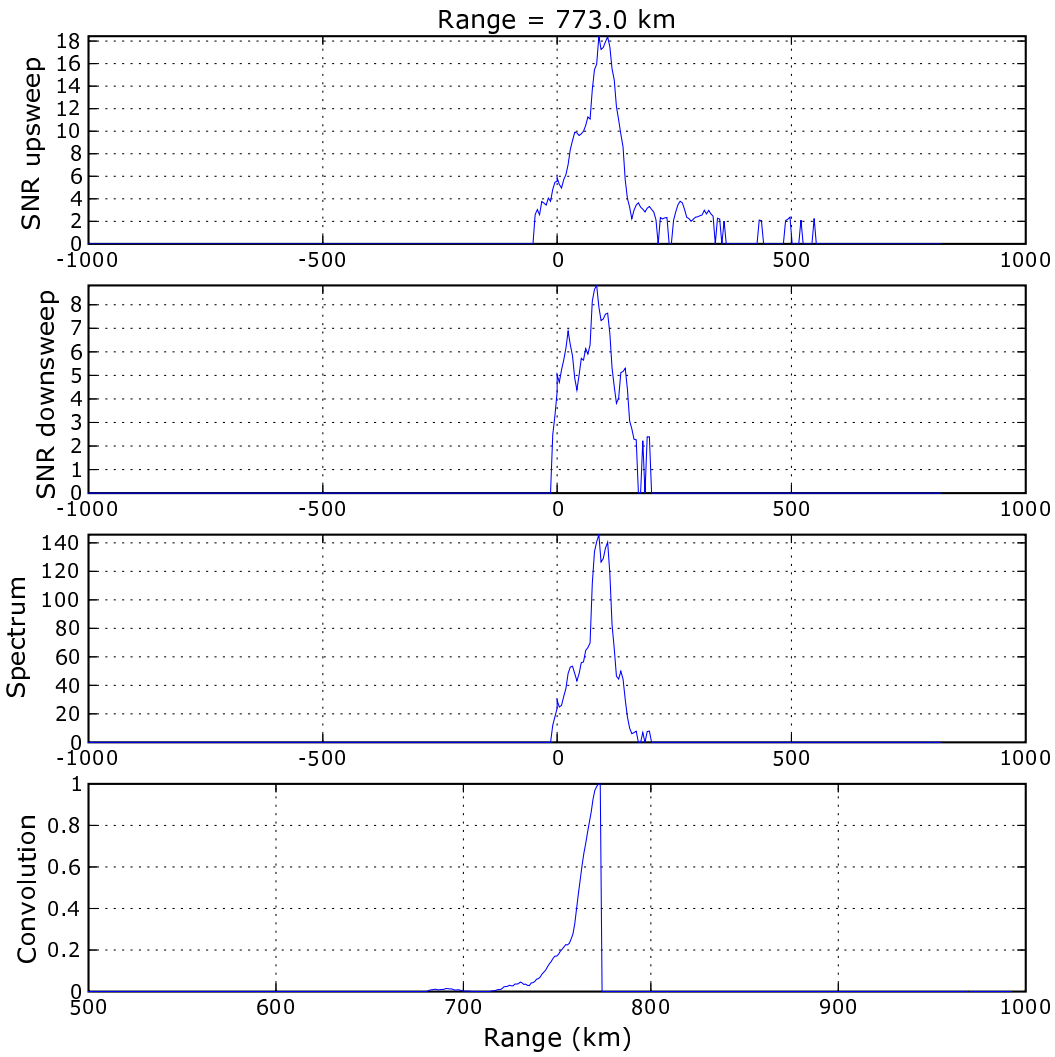

Figure A.3: Illustration of convolution. The data represents the convolution at a range of 774.0 km.

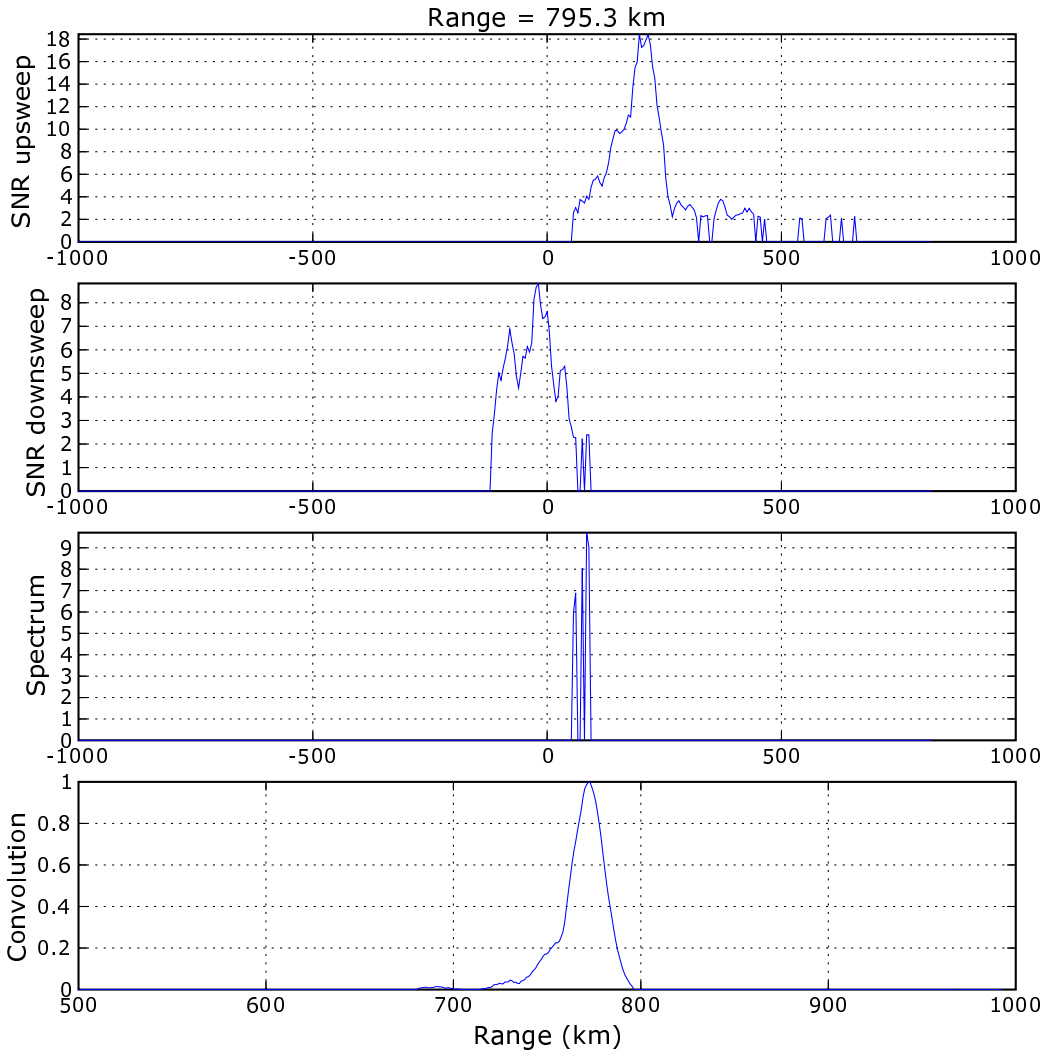

Figure A.4: Illustration of convolution. The data represents the convolution at a range of 688.6 km.

Appendix B

Technical References

Alpha-power, *Alpha 6 Manual*, Alpha Power, Boulder, Colorado.

Beasley, P. D. L., Stove, A. G., and Reits, B. J., Solving the Problems of a Single Antenna Frequency Modulated CW Radar, in *The Record of the IEEE 1990 International Radar Conference*, pp. 391-395, IEEE, 1990.

Campbell, R., High performance direct-conversion receivers, *QST*, pp. 19-28, 1992.

Campbell, R., High performance, single-signal direct-conversion receivers, *QST*, pp. 33-40, 1993.

Campbell, R., and B. Kelsey, A next-generation R2 single-signal direct conversion receiver, `http://www.bright.net/~kanga/r2pro/article.htm`, 2003.

Eaton, F. D., S. A. McLaughlin, and J. R. Hines, A new frequency-modulated continuous wave radar for studying planetary boundary layer morphology, *Radio Science*, *30*, pp. 75-88, 1995.

Garmin, *GPS 35/36 TrakPak GPS smart antenna technical specifications*, Garmin, Olathe, Kanasas, 1999.

M-audio, *Delta 66 Manual*, Tewksbury, MA, 2003.

Harris, T., *400 MSPS 14-bit, 1.8V CMOS Direct Digital Synthesizer*, Analog Devices, 2003.

Intel, Application Notes 102: Alias and spurious responses in DDS systems, `ftp://download.intel.com/design/digital/applnots/24544001.pdf`, 1990.

Osicom Technologies, Direct Digital Frequency Synthesis: A Basic Tutorial, `http://www.ehb.itu.edu.tr/~eepazarc/ddstutor.html`, 1990.

Ortlepp, A., *SAPPHIRE VHF radar system manual*, Institute of Space and Atmospheric Studies, University of Saskatchewan, 1994.

Spectrum Lab, Information about FFT in Spectrum Lab, `http://www.qsl.net/dl4yhf/speclab/fftinfo.htm`.

Appendix C
Radar control program

```
/* This program is designed to control the data collection of the U of S ISAS 50
   MHz FMCW Radar.  It collects data through the soundcard and controls the
   frequency sweeping of the radar by sending signals to the AD9954 DDS
   evaluation board through the parallel port.  The program runs three threads.
   The main thread controls the data collection.  The two secondary threads take
   the collected data and write it to a file using gzip format and after
   stripping the zeros from the 4 byte digital sample (sound card has 24 bit
   resolution but uses 4 bytes with one byte equal to zero).  The threads are
   run using SCHED_FIFO scheduling defined in the POSIX API in order to
   approximate real-time data collection.  The main thread runs at the highest
   priority and the two secondary threads run at a slightly lower priority.  The
   command I am using to compile the program is:

   cc -O -lpthread -lasound -lz -o z24pthsample.o z24pthsample.c
   -O optimizes.  This is recommended for control of parallel port
   -lpthread is required for the POSIX threads API
   -lz is required for the zlib API (gzip)
   -lasound is required for the ALSA API
   The program must be run with root privaleges. */

   /* This particular program was created on May 7, 2004.  I have added the
    * creation of a web accessible file to allow for remote monitoring.  At the
    * same time I removed some of the user interface that I added into the
    * program earlier.  Included is pthreads, gzip, .rdat24, DDS control, and
    * web accessible file */

#include <pthread.h>
#include <sched.h>
#include <alsa/asoundlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <sys/time.h>
#include <sys/io.h>
#include <signal.h>
#include <zlib.h>
#include <sys/stat.h>
#include <sys/sendfile.h>
/* location of parallel port */
   #define DATA 0x378  // Data port
   #define STATUS DATA+1 // Status port -- not used
   #define CONTROL DATA+2 // Control port

   #define FLIPBITS 0x05 /* xor the Data port to flip bits
                            Data port bit 0 controls PS0 DUT1
                            Data port bit 2 controls PS0 DUT2 */
   #define USLEEP_TIME 6000 // useconds between sweep start and data sampling
   #define SCLK 0x02 // sclk bit

    // Basic data structure
    struct databin {
      struct timeval timestamp;  // timestamp
      char *data; // soundcard pcm data
    };

    /* a series of global variables */

    /* DDS variables that define FM waveform */
    long freqbot = 49420; // kHz
    long freqtop = 49580; // kHz
    long freqstep = 652;  // mHz
    long freqclock = 900; // ns
    long DATAF_MAX = 10000; // number of data segments per output file
    gzFile fd; // gz file type

    /* Addition May 7 to allow for web accessible file */
    int fdweb; // file descriptor for web accessible file
    int webdataf_max = 40; //~approximately 10 second data intervals
    int webdataf_size = 0; // File size counter
    char *webdata_file = "/home/radar/fmcwrun/webdata/webdata_file.rdat32"; // data file to be written to
    /* Location to write most recent complete data set for web access */
    char *webdataInter_file = "/home/radar/fmcwrun/webdata/webdataInter_file.rdat32";
    /* end addition */

    int outcond = 1;

    // Data structures
    struct databin dataup;
    struct databin datadn;

    int rate = 9600; /* Sample rate */
```

```
    unsigned int channels = 12; // Delta 66 only seems to support 12 channels
    int periods = 2;        /* Number of periods */
    int frames = 2048;      /* Samples */
    int dataf_size; // variable that keeps track of how much data has been written to file
    int globday = 33; /* day of month variable that determines when a new data directory is created
                         Set to 33 to ensure that directory is created first time */

    /* ALSA defined variables */
    snd_pcm_format_t format = SND_PCM_FORMAT_S32_LE;
    snd_pcm_access_t Access = SND_PCM_ACCESS_RW_INTERLEAVED;

    /* Name of the PCM device, hw:1,0           */
    /* The first number is the number of the soundcard, */
    /* the second number is the number of the device.   */
    char *pcm_name = "hw:1,0";

    /* POSIX defined variables */
    pthread_t thread1; // thread1 handles up sweep data
    pthread_t thread2; // thread2 handles down sweep data
    pthread_mutex_t dataup_mutex;  // control access to dataup
    pthread_cond_t thread1_cond;   // wakeup condition for thread1
    pthread_mutex_t thread1_mutex; // thread1 condition mutex
    pthread_mutex_t datadn_mutex;  // control access to datadn
    pthread_cond_t thread2_cond;   // wakeup condition for thread2
    pthread_mutex_t thread2_mutex; // thread2 condition mutex
    pthread_mutex_t dataf_mutex;   // control data file access

void test_state(snd_pcm_t *pcm_handle)
{/* Test the state of the soundcard and output name of state to screen */
    snd_pcm_state_t state;
    state = snd_pcm_state(pcm_handle);
    printf("state = %s\n", snd_pcm_state_name (state));
}

int set_realtime_priority(void)
{/* set main program to highest priority */
   struct sched_param schp;
   memset(&schp, 0, sizeof(schp));

   // Get and set to max priority on SCHED_FIFO
   schp.sched_priority = sched_get_priority_max(SCHED_FIFO);
   if (sched_setscheduler(0, SCHED_FIFO, &schp) != 0)
   {
     perror("sched_setscheduler");
     return -1;
   }
   return 0;
}

char write_pp(char DataByte)
{ /* Write DataByte to the DATA port and clock bit one of CONTROL port.
     This function is specifically intended to flip the PS0 bits of
     the two DDS chips. */
     DataByte = DataByte^FLIPBITS; // flip bits to be written to DATA port
     outb(DataByte, DATA); // write to DATA port
     // clock data into DDS
     outb(0x01, CONTROL); // note: CONTROL bit 0 is hardware inverted
     outb(0x00, CONTROL);
     return DataByte;
}

int set_hw_params(snd_pcm_t *pcm_handle)
{/* Set the hw parameters on the sound card */

    /* This structure contains information about    */
    /* the hardware and can be used to specify the  */
    /* configuration to be used for the PCM stream. */
    snd_pcm_hw_params_t *hwparams;

    /* Allocate the snd_pcm_hw_params_t structure on the stack. */
    snd_pcm_hw_params_malloc(&hwparams);
    int err;
    unsigned int chan;
    snd_pcm_uframes_t buffsize;
    int dir, newrate;           /* exact_rate == rate --> dir = 0 */
                        /* exact_rate < rate  --> dir = -1 */
                        /* exact_rate > rate  --> dir = 1 */

    /* Init hwparams with full configuration space */
    if (snd_pcm_hw_params_any(pcm_handle, hwparams) < 0) {
      fprintf(stderr, "Can not configure this PCM device.\n");
      return(-1);
    }

    /* Set PCM access to global variable Access */
    /* This can be either     */
    /* SND_PCM_ACCESS_RW_INTERLEAVED or        */
    /* SND_PCM_ACCESS_RW_NONINTERLEAVED.       */
    /* There are also access types for MMAPed access */
    err = snd_pcm_hw_params_set_access(pcm_handle, hwparams, Access);
    if (err < 0)
    {
```

```
    printf("Error setting access: %s\n", snd_strerror(err));
    return(-1);
  }
  printf("Access = %s\n", snd_pcm_access_name(Access));

  /* Set sample format e.g. signed 32 bit little endian */
  err = snd_pcm_hw_params_set_format(pcm_handle, hwparams, format);
  if (err < 0) {
    printf("Error setting format: %s\n", snd_strerror(err));
    return(-1);
  }
  printf("Format set to %s\n", snd_pcm_format_name(format));

  /* Set sample rate. If the exact rate is not supported */
  /* by the hardware, use nearest possible rate.         */
  newrate = rate;
  err = snd_pcm_hw_params_set_rate_near(pcm_handle, hwparams, &newrate, &dir);
  if (newrate != rate)
  {
    printf("Desired sampling rate of %i Hz is not supported.\nSampling rate has been set to %iHz\n",
        rate, newrate);
    rate = newrate;
  }
  else printf("Sampling rate is %iHz\n", rate);

  /* Set number of channels.  Note that the Delta 66 only seems to
     directly support 12 channels */
  chan = channels;
  err = snd_pcm_hw_params_set_channels(pcm_handle, hwparams, chan);
  if (err < 0)
  {
    printf("Error setting channels: %s.\n", snd_strerror (err));
    err = snd_pcm_hw_params_set_channels_near(pcm_handle, hwparams, &chan);
    if (err < 0)
    {
       printf("Error setting channels near: %s.\n", snd_strerror (err));
       return(-1);
    }
  }
  channels = chan;
  printf("Channels set to %i\n", channels);

 /* Set number of periods.  I have found that at least 2 periods should be
    used to prevent problems reading data.
    A period is a division in the buffer.  If there are 2 periods that means
    the sound card buffer is divided into 2 sections.  The soundcard wakes
    up the program to hand it data when each section is full. */
  if (snd_pcm_hw_params_set_periods(pcm_handle, hwparams, periods, 0) < 0) {
    printf("Error setting periods %s.\n", snd_strerror (err));
    return(-1);
  }

  /* Set buffer size (in frames). The resulting latency is given by */
  /* latency = periodsize * periods / (rate * bytes_per_frame)     */
  buffsize = frames;
  if (snd_pcm_hw_params_set_buffer_size_near(pcm_handle, hwparams, &buffsize) < 0) {
    fprintf(stderr, "Error setting buffersize.\n");
    return(-1);
  }
  printf("Buffer size set to %i frames\n", (int) buffsize);

  /* Apply HW parameter settings to */
  /* PCM device and prepare device  */
  if (snd_pcm_hw_params(pcm_handle, hwparams) < 0) {
    fprintf(stderr, "Error setting HW params.\n");
    return(-1);
  }

  /* release hwparams allocated memory */
  snd_pcm_hw_params_free(hwparams);
  return(0);
}

int write_header(int filed)
{
  /* create header of 7 variable type long values in following order
  dataEntries frames bottomFrequency topFrequency frequencySweepStep frequencySweepClock sampleRate
  Each is of size long (4 Bytes) so that it can be read easily by processing program
  */
  if(write(filed, (long *) &webdataf_max, sizeof(long))!=4)
    return(-1);
  if(write(filed, (long *) &frames, sizeof(long))!=4)
    return(-1);
  if (write(filed, &freqbot, sizeof(long))!=4)
    return(-1);
  if (write(filed, &freqtop, sizeof(long))!=4)
    return(-1);
  if (write(filed, &freqstep, sizeof(long))!=4)
    return(-1);
  if (write(filed, &freqclock, sizeof(long))!=4)
    return(-1);
  if (write(filed, (long *) &rate, sizeof(long))!=4)
```

```
    return(-1);
  return(0);
}


int create_file(void)
{/* Creates the data file to be written to by the secondary threads.
    It creates a new directory each day.  Data files are of the format
    hourminutesecond.rdat24.gz */

  char datafile[50];
  char directory[40];
  char *outmode="wb8";
  struct timeval clock;
  void *tz = NULL;
  struct tm *tms;
  gettimeofday(&clock, tz);
  tms = localtime(&clock.tv_sec);
  // Create a new directry each day
  sprintf(directory, "%.2i%.2i%.2i", tms->tm_year-100, tms->tm_mon, tms->tm_mday);
  if (globday != tms->tm_mday)
  {
    mkdir(directory, 0777); // This mkdir call is buggy about setting permissions.
    chmod(directory, 0777); // The chmod call was added to ensure permissions are set properly.
    globday = tms->tm_mday;
  }
  if (sprintf(datafile, "%s/%.2i%.2i%.2i%.2i.24.gz",directory, tms->tm_year-100, tms->tm_mon, tms->
      tm_mday, tms->tm_hour,tms->tm_min)<0)
     return(-1); // tms->year -100 puts year in format "04"

  // open gz file
  fd = gzopen(datafile,outmode);
  if (fd == NULL)
  {
     printf("error opening %s\n", datafile);
     return(-1);
  }
  /* create header of 7 variable type long values in following order
  dataEntries frames bottomFrequency topFrequency frequencySweepStep frequencySweepClock sampleRate
  Each is of size long (4 Bytes) so that it can be read easily by processing program
  */
  if(gzwrite(fd, (long *) &DATAF_MAX, sizeof(long))!=4)
    return(-1);
  if(gzwrite(fd, (long *) &frames, sizeof(long))!=4)
    return(-1);
  if (gzwrite(fd, &freqbot, sizeof(long))!=4)
    return(-1);
  if (gzwrite(fd, &freqtop, sizeof(long))!=4)
    return(-1);
  if (gzwrite(fd, &freqstep, sizeof(long))!=4)
    return(-1);
  if (gzwrite(fd, &freqclock, sizeof(long))!=4)
    return(-1);
  if (gzwrite(fd, (long *) &rate, sizeof(long))!=4)
    return(-1);
  dataf_size = 0; // reset data file size variable
  return(0);
}

int copy_file(char *srcfile, char *destfile)
{
  int srcfd;               /* file descriptor for source file */
  int destfd;              /* file descriptor for destination file */
  struct stat stat_buf;  /* hold information about input file */
  off_t offset = 0;       /* byte offset used by sendfile */
  int rc;                 /* return code from sendfile */

  /* open source file */
  srcfd = open(srcfile, O_RDONLY);
  if (srcfd == -1) {
    fprintf(stderr, "unable to open '%s': %s\n", srcfile, strerror(errno));
    exit(1);
  }

 /* get size and permissions of the source file */
  fstat(srcfd, &stat_buf);

 /* open destination file */
  destfd = open(destfile, O_WRONLY|O_CREAT, stat_buf.st_mode);
  if (destfd == -1) {
    fprintf(stderr, "unable to open '%s': %s\n", destfile, strerror(errno));
    exit(1);
  }

 /* copy file using sendfile */
 rc = sendfile (destfd, srcfd, &offset, stat_buf.st_size);
 if (rc == -1) {
    fprintf(stderr, "error from sendfile: %s\n", strerror(errno));
    return -1;
 }
 if (rc != stat_buf.st_size) {
```

```
   fprintf(stderr, "incomplete transfer from sendfile: %d of %d bytes\n", rc, (int)stat_buf.st_size);
   return -1;
 }

 /* clean up and exit */
 close(destfd);
 close(srcfd);
 return 0;
}


void *threadfuncup(int id)
{ /* Function for thread to deal with data during an up sweep */
       int i, j;
       long sweepflag = 1; // flag for upsweep
       char *data;
       char *webdata; // Addition May 7
       int frameRet;
       webdata = malloc(frames*2*4);
       data = malloc(frames*2*3); // implicit expectation of 2 channels
       if (data == NULL)
       {
          printf("Unable to allocate memory\n");
          pthread_exit(NULL);
       }
       for(;;)
       {
         /* wait until data is ready to be processed */
         pthread_mutex_lock(&thread1_mutex);
         pthread_cond_wait(&thread1_cond, &thread1_mutex);
         pthread_mutex_unlock(&thread1_mutex);

         /* lock dataup and process data */
         pthread_mutex_lock(&dataup_mutex);
         if (dataup.data == NULL)  // condition for exit
         {
           pthread_mutex_unlock(&dataup_mutex);
           pthread_exit(NULL);
         }
       /* Data from the sound card is 32 bytes, but only 24 are
          significant.  In order to save memory, 1 byte is removed here
          and replaced during processing.  Also, 12 channels are collected
          but only 2 are significant. */
         for(i=0; i<frames; i++)
         {
           for(j=0; j<8; j++)
             webdata[8*i+j] = dataup.data[4*i*channels+j];
           data[6*i] = dataup.data[4*i*channels+1];
           data[6*i+1]= dataup.data[4*i*channels+2];
           data[6*i+2]= dataup.data[4*i*channels+3];
           data[6*i+3]= dataup.data[4*i*channels+5];
           data[6*i+4]= dataup.data[4*i*channels+6];
           data[6*i+5]= dataup.data[4*i*channels+7];
         }
         pthread_mutex_unlock(&dataup_mutex);

         pthread_mutex_lock(&dataf_mutex);
         /* Data written with time stamp and sweep flag
            timeInSeconds timeInMicroseconds sweepflag dataArray */

         /* Addition on May 7 to write to web file */
         frameRet = write(fdweb, &dataup.timestamp.tv_sec, sizeof(long));
         frameRet = write(fdweb, &dataup.timestamp.tv_usec, sizeof(long));
         frameRet = write(fdweb, &sweepflag, sizeof(long));
         frameRet = write(fdweb, webdata,frames*2*4);
         webdataf_size++;
         if (webdataf_size >= webdataf_max)
         {
            close(fdweb);
            if (copy_file(webdata_file, webdataInter_file) < 0)
            {
              printf("From process %i: Problem copying webfile\n", id);
            }
            fdweb = open(webdata_file, O_WRONLY);
            write_header(fdweb);
            webdataf_size = 0;
         }
         /* End addition on May 7 */

         /* Wite to gzip file */
         frameRet = gzwrite(fd, &dataup.timestamp.tv_sec, sizeof(long));
         frameRet = gzwrite(fd, &dataup.timestamp.tv_usec, sizeof(long));
         frameRet = gzwrite(fd, &sweepflag, sizeof(long)); // flag for upsweep
         frameRet = gzwrite(fd, data, frames*2*3);
         if (frameRet != frames*2*3)
           printf("write error: frameRet = %i and should be %i\n", frameRet, frames*2*3);

         /* Increment data file size monitor variable.
            If it is greater than DATAF_MAX close the file
            and create a new file. */
         dataf_size++;
         if (dataf_size >= DATAF_MAX)
```

```
        {
            gzclose(fd);
            if (create_file() < 0)
            {
              printf("Process %i stopped since it could not open file", id);
              pthread_exit(NULL);
            }
        }
        // unlock mutexes
        pthread_mutex_unlock(&dataf_mutex);
      }
      pthread_exit(NULL);
}

void *threadfuncdn(int id)
{/* Thread function to deal with data collected during a down sweep */
      int i, j;
      long sweepflag = 0; // flag for downsweep
      char *data;
      char *webdata; // Addition May 7
      int frameRet;
      webdata = malloc(frames*2*4);
      data = malloc(frames*2*3);
      if (data == NULL)
      {
         printf("Unable to allocate memory\n");
         pthread_exit(NULL);
      }
      for(;;)
      {
        /* wait until data is ready to be processed */
        pthread_mutex_lock(&thread2_mutex);
        pthread_cond_wait(&thread2_cond, &thread2_mutex);
        pthread_mutex_unlock(&thread2_mutex);

        /* lock dataup and process data */
        pthread_mutex_lock(&datadn_mutex);
        if (datadn.data == NULL)  // condition for exit
        {
          pthread_mutex_unlock(&datadn_mutex);
          pthread_exit(NULL);
        }
        /* Data from the sound card is 32 bytes, but only 24 are
           significant.  In order to save memory, 1 byte is removed here
           and replaced during processing.  Also, 12 channels are collected
           but only 2 are significant. */
        for(i=0; i<frames; i++)
        {
          for(j=0; j<8; j++)
            webdata[8*i+j] = datadn.data[4*i*channels+j];
          data[6*i] = datadn.data[4*i*channels+1];
          data[6*i+1]= datadn.data[4*i*channels+2];
          data[6*i+2]= datadn.data[4*i*channels+3];
          data[6*i+3]= datadn.data[4*i*channels+5];
          data[6*i+4]= datadn.data[4*i*channels+6];
          data[6*i+5]= datadn.data[4*i*channels+7];
        }
        pthread_mutex_unlock(&datadn_mutex);

        /* lock data file and write */
        pthread_mutex_lock(&dataf_mutex);
        /* Data written with time stamp and sweep flag as follows:
           timeInSeconds timeInMicroseconds sweepflag dataArray */

        /* Addition on May 7 to write to web file */
        frameRet = write(fdweb, &datadn.timestamp.tv_sec, sizeof(long));
        frameRet = write(fdweb, &datadn.timestamp.tv_usec, sizeof(long));
        frameRet = write(fdweb, &sweepflag, sizeof(long));
        frameRet = write(fdweb, webdata,frames*2*4);
        webdataf_size++;
        if (webdataf_size >= webdataf_max)
        {
           close(fdweb);
           if (copy_file(webdata_file, webdataInter_file) < 0)
           {
             printf("From process %i: Problem copying webfile\n", id);
           }
           fdweb = open(webdata_file, O_WRONLY);
           write_header(fdweb);
           webdataf_size = 0;
        }
        /* End addition on May 7 */

        frameRet = gzwrite(fd, &datadn.timestamp.tv_sec, sizeof(long));
        frameRet = gzwrite(fd, &datadn.timestamp.tv_usec, sizeof(long));
        frameRet = gzwrite(fd, &sweepflag, sizeof(long)); // flag for downsweep
        frameRet = gzwrite(fd, data, frames*2*3);
        if (frameRet != frames*2*3)
          printf("write error: frameRet = %i and should be %i", frameRet, frames*2*3);

        /* Increment data file size variable.
           If it is greater than DATAF_MAX close the file
```

```
            and create a new file. */
          dataf_size++;
          if (dataf_size >= DATAF_MAX)
          {
             gzclose(fd);
             if (create_file() < 0)
             {
               printf("Process %i stopped since it could not open file", id);
               pthread_exit(NULL);
             }
          }
          pthread_mutex_unlock(&dataf_mutex);
        }
        pthread_exit(NULL);
}

int init_threads(void)
{/* Creates the two data processing threads */
   int threadid1 = 1;
   int threadid2 = 2;
   pthread_attr_t attr;
   struct sched_param schedparam;
   memset(&schedparam, 0, sizeof(schedparam));
   pthread_attr_init(&attr);
   pthread_attr_setdetachstate(&attr, PTHREAD_CREATE_JOINABLE);
   pthread_attr_setschedpolicy(&attr, SCHED_FIFO);
   /* Set priority level one below data collection task */
   schedparam.sched_priority = sched_get_priority_max(SCHED_FIFO)-1;

   pthread_attr_setschedparam(&attr, &schedparam);
   if (pthread_create(&thread1, &attr, (void *)threadfuncup, (void *) &threadid1)<0)
     return(-1);
   if (pthread_create(&thread2, &attr, (void *)threadfuncdn, (void *) &threadid2)<0)
     return(-1);
   pthread_attr_destroy(&attr);
   return(0);
}

void INThandler(int sig)
{ /* Handles a SIGINT */
  signal(sig, SIG_IGN);
  outcond = 0; // break out of the loop
}

int write_dds(char *data)
{ /* Accepts a string of 1's and 0's and writes those values to
     the DDS.  Both DDS chips are programmed the same. */
  int i, add;
  unsigned long value = 0;
  char DataByte;
  for (i = 0;i<strlen(data); i++)
  {
    if (data[i] == '1')
    {
      DataByte = 0x01;
      add = 1<<(strlen(data)-1-i);
      value = value + add;
    }
    else if (data[i] == '0')
      DataByte = 0x00;
    else
    {
      printf("Improper character %c\n", data[i]);
      return(-1);
    }
    outb(DataByte, DATA); // write to DATA port
    // clock data into D-flipflop
    outb(0x08, CONTROL);
    usleep(100); // small pause between clock up down.  Probably not necessary.
    outb(0x00, CONTROL);
    DataByte = DataByte^SCLK; // flip bits to clock data into dds
    outb(DataByte, DATA); // write to DATA port
    outb(0x08, CONTROL);
    usleep(100); // small pause between clock up down.  Probably not necessary.
    outb(0x00, CONTROL);
  }
  return 0;
}

int ioupdate(void)
{/* Issue an IO Update to the DDS chips */
  outb(0x00, CONTROL);
  outb(0x00, DATA);
  outb(0x01, CONTROL);
  usleep(100);
  outb(0x00, CONTROL);
  usleep(100);
  outb(0xC0, DATA);
  outb(0x01, CONTROL);
  usleep(100);
  outb(0x00, CONTROL);
  return 0;
```

```
}

int convert2binstr(char *word, unsigned long value, int nbits)
{/* Convert a value to a binary representation.  Store the binary
    representation in a string. */
  int i;
  unsigned long temp;
  for (i = 0; i<nbits; i++)
  {
    temp = 1<<(nbits-1-i);
    if (temp <= value)
    {
      word[i] = '1';
      value = value - temp;
    }
    else
    {
      word[i] = '0';
    }
  }
  word[i] = '\0'; // NULL terminate
  return (int) value;
}

int ResetDDS(void)
{  // issue a reset to the DDS
  outb(0x00, CONTROL);
  outb(0xC0, DATA);
  outb(0x02, CONTROL);
  usleep(100);
  outb(0x00, CONTROL);
  usleep(100);
  outb(0x00, DATA);
  outb(0x02, CONTROL);
  usleep(100);
  outb(0x00, CONTROL);
  return 0;
}

int LoadDDS(void)
{ /* Load the DDS.  It sets the linear sweep bit and sets up the registers
     to perform the frequency sweep specified by the user.  The capabilities are
     limited on purpose, but comments are intended to make it easy to add DDS
     capabilities.  Refer to AD9954 datasheet for details. */
  unsigned long num;
  unsigned long L31 = 1<<31; // 2^31
  char *CFR1 = "00000000001000000000000000000000"; // set linear sweep bit
  char *CFR1inst = "00000000";
  // Clock multiplier set to 5 and VCO gain bit set high
  char *CFR2 = "00011000000000000101100";
  char *CFR2inst = "00000001";
  char *ASF = "0000111111001001"; // Related to OSK
  char *ASFinst = "00000010";
  char *ARR = "00111010";  // Related to OSK
  char *ARRinst = "00000011";
  char FTW0[33]; // lower frequency set by user input
  char *FTW0inst = "00000100";
  char *POW0 = "0000000000000000"; // phase offset set to 0
  char *POW0inst = "00000101";
  char FTW1[33]; // Upper frequency set by user input
  char *FTW1inst = "00000110";
  char RSCWdf[33]; // Delta frequency set by user input
  char RSCWrr[9]; // ramp rate timer set by user input
  char *RSCW0inst = "00000111";
  char *RSCW1inst = "00001000";
  char *RSCW2inst = "00001001";
  char *RSCW3inst = "00001010";

  if (ioperm(DATA,3,1))
  {
    printf("Sorry, you were not able to gain access to the ports\n");
    printf("You must be root to run this program\n");
    return(-1);
  }

  ResetDDS();

  /* In order to write the data into the DDS, the binar values to be written to each register
     is put into a string containing 1's and 0's.  These 1's and 0's are then written
     through the serial line of the DDS. */

  // Calculate values assuming 400 MHz clock.  If this changes, these values will be invalid.
  num = (unsigned long) (((float) freqbot)/((float) 400000.0)*((float) L31)*2.0);// rounding error is
       minimal
  if (convert2binstr(FTW0, num, 32) != 0)
    printf("Error converting to binary\n");
  num = (unsigned long) (((float) freqtop)/((float) 400000.0)*((float) L31)*2.0);// rounding error is
       minimal
  if (convert2binstr(FTW1, num, 32) != 0)
    printf("Error converting to binary\n");
  num = (unsigned long) (((float) freqclock)*0.1);// rounding error is minimal
  if (convert2binstr(RSCWrr, num, 8) != 0)
```

```
    printf("Error converting to binary\n");
  num = (unsigned long) ((((float) freqstep)/1000.0)/((float) 400000000.0)*((float) L31)*2.0);// rounding
        error is minimal
  if (convert2binstr(RSCWdf, num, 32) != 0)
    printf("Error converting to binary\n");

  /* Once the binary values are in strings each register must be written to.  The instruction
     byte is written followed by the value of the register.  The data is written MSB first. */
  write_dds(CFR1inst);
  write_dds(CFR1);
  write_dds(CFR2inst);
  write_dds(CFR2);
  write_dds(ASFinst);
  write_dds(ASF);
  write_dds(ARRinst);
  write_dds(ARR);
  write_dds(FTW0inst);
  write_dds(FTW0);
  write_dds(POW0inst);
  write_dds(POW0);
  write_dds(FTW1inst);
  write_dds(FTW1);
  write_dds(RSCW0inst); // Upsweep values
  write_dds(RSCWrr);
  write_dds(RSCWdf);
  write_dds(RSCW1inst); // Down sweep set to same rate as upsweep
  write_dds(RSCWrr);
  write_dds(RSCWdf);
  write_dds(RSCW2inst);
  write_dds(RSCWrr);
  write_dds(RSCWdf);
  write_dds(RSCW3inst);
  write_dds(RSCWrr);
  write_dds(RSCWdf);
  ioupdate();
  return 0;
}
int main(int argc, char **argv)
{
    int err, usleeptime, usweeptime;
    void *tz = NULL;
    struct timeval starttime, endtime; // time structures to control sweeptime
    int happy=0;
    int frameRet;
    char ppdata = 0x00;//, tempByte = 0x00; /* parallel port DATA byte to control radar */

    /* ALSA defined variables */
    snd_pcm_t *pcm_handle;    /* Handle for the PCM device */
    snd_pcm_stream_t stream_capture = SND_PCM_STREAM_CAPTURE; // specify capture stream
    snd_pcm_stream_t pcm_stream;

    /* Sets the ports DATA to DATA+3 to permission 1
       Gives permission to write to the parallel port.
       Must be root to use this function. */
    if (ioperm(DATA,3,1)) {
       printf("Sorry, you were not able to gain access to the ports\n");
       printf("You must be root to run this program\n");
       return(-1);
    }

    // Addition to ensure PP cable can be plugged in without resetting DDS
    outb(0x00, DATA);
    outb(0x0B, CONTROL);

    /* calculate the time of a frequency sweep in useconds
     * Note on units: 1000*kHz*ns/mHz = us
     * Integer math rounds off value */
    usweeptime = 1000*(freqtop-freqbot)*freqclock/freqstep;
    if (usweeptime < (((float) frames)/((float) rate)*1000000+USLEEP_TIME))
    {
       printf("\nWARNING: DDS sweeptime = %i us and sampling interval = %.2f us\n\n", usweeptime,((float)
            frames)/((float) rate)*1000000+USLEEP_TIME);
    }

    LoadDDS();

    // Open sound card pcm device to capture
    err = snd_pcm_open(&pcm_handle, pcm_name, stream_capture, 0);
    if (err < 0)
    {
      printf("Error opening PCM device %s: %s\n", pcm_name, snd_strerror(err));
      return(-1);
    }
    test_state(pcm_handle);

    pcm_stream = snd_pcm_stream(pcm_handle);
    if (pcm_stream != stream_capture)
      printf("Error setting PCM steam\n");

    // set up the hw parameters of the soundcard
    err = set_hw_params(pcm_handle);
    if (err == -1)
```

```
{
  printf("error setting hw parameters %s\n", snd_strerror(err));
  return(-1);
}
test_state(pcm_handle);

err = set_realtime_priority();
if (err == -1)
{
  fprintf(stderr, "error setting priority\n");
  return(-1);
}

/* Initialize mutex and condition variable objects */
pthread_mutex_init(&dataup_mutex, NULL);
pthread_cond_init(&thread1_cond, NULL);
pthread_mutex_init(&thread1_mutex, NULL);
pthread_mutex_init(&datadn_mutex, NULL);
pthread_cond_init(&thread2_cond, NULL);
pthread_mutex_init(&thread2_mutex, NULL);
pthread_mutex_init(&dataf_mutex, NULL);

/* start the threads that will process data */
err = init_threads();
if (err == -1)
{
  printf("error initializing threads\n");
  return(-1);
}


// start pcm to eliminate buggy start
snd_pcm_start(pcm_handle);
test_state(pcm_handle);

/* initialize radar to lower freqeuency by completing one sweep */
ppdata = write_pp(ppdata);
 ppdata = ppdata^0x04;
 outb(ppdata, DATA); // write to DATA port
 //clock data into DDS
 outb(0x01, CONTROL); // note: CONTROL bit one is hardware inverted
 outb(0x00, CONTROL);
usleep(usweeptime);
ppdata = write_pp(ppdata);
ppdata = ppdata^0x04;
outb(ppdata, DATA); // write to DATA port
// clock data into DDS
outb(0x01, CONTROL); // note: CONTROL bit one is hardware inverted
outb(0x00, CONTROL);
usleep(usweeptime);

/* initialize data structures */
dataup.data = malloc(frames*channels*4);
if (dataup.data == NULL)
{
   printf("Unable to allocate memory\n");
   return(-1);
}

datadn.data = malloc(frames*channels*4);
if (datadn.data == NULL)
{
   printf("Unable to allocate memory\n");
   return(-1);
}

/* Create first data file */
if(create_file() == -1)
  return(-1);

fdweb = open(webdata_file,O_WRONLY|O_CREAT);
if (fdweb < 0)
  return(-1);
chmod(webdata_file, 0666);
write_header(fdweb);

printf("\nUse ctrl-c to stop program\n\n");
signal(SIGINT, INThandler); // handle interupt to ensure proper program closure

while(outcond) //(happy < DATAF_MAX*2) &&
{

  // getchar();
   ppdata = write_pp(ppdata); // start frequency sweep
   //sleep(2);
   err = gettimeofday(&starttime, tz); // get time stamp

  //temporary addition to test rader
  /*usleep(4000);
  ppdata = ppdata^0x04;
  outb(ppdata, DATA); // write to DATA port
  //clock data into DDS
```

```
    outb(0x01, CONTROL); // note: CONTROL bit one is hardware inverted
    outb(0x00, CONTROL);*/
   //printf("ppdata = %X\n", ppdata);
    // End temporary addition

    snd_pcm_drop(pcm_handle);  // Stop the sound card and clear the buffer
    snd_pcm_prepare(pcm_handle); // Prepare the sound card

    /* wait for USLEEP_TIME between frequency sweep start to data collection start
       if time interval is off by more the 1 ms */
    err = gettimeofday(&endtime, tz);
    usleeptime=USLEEP_TIME-((endtime.tv_sec-starttime.tv_sec)*1000000+endtime.tv_usec-starttime.tv_usec
        );
   // printf("usleeptime = %i, ppdata = %i\n", usleeptime, ppdata);
    if (usleeptime > 0)
      usleep(usleeptime);


    snd_pcm_start(pcm_handle); // start data collection

    /* Collect the data and pass it to the appropriate thread */
    if (ppdata) // up sweep
    {
      /* lock mutex and read data into dataup */
      pthread_mutex_lock(&dataup_mutex);
      dataup.timestamp = starttime;
      frameRet = snd_pcm_readi(pcm_handle, dataup.data, frames);
      pthread_mutex_unlock(&dataup_mutex);
      /* signal thread that data is ready */
      pthread_mutex_lock(&thread1_mutex);
      pthread_cond_signal(&thread1_cond);
      pthread_mutex_unlock(&thread1_mutex);
    }
    else // down sweep
    {
      /* lock mutex and read data into datadn */
      pthread_mutex_lock(&datadn_mutex);
      datadn.timestamp = starttime;
      frameRet = snd_pcm_readi(pcm_handle, datadn.data, frames);
      pthread_mutex_unlock(&datadn_mutex);
      /* signal thread that data is ready */
      pthread_mutex_lock(&thread2_mutex);
      pthread_cond_signal(&thread2_cond);
      pthread_mutex_unlock(&thread2_mutex);
    }

    /* Check to make sure the right number of bits were written.
       This can be a problem if the program was interrupted during the read function
       or if the number of frames isn't set to a power of 2. */
    if (frameRet != frames)
    {
      fprintf(stderr, "error reading data\n");
      printf("I read %i frames of data and I was supposed to read %i frames\n", frameRet, frames);
    }

//      printf("%i\n", happy);//I am happy:
    happy++;

    /* Check the time again and sleep if pace is off by more than a ms */
    err = gettimeofday(&endtime, tz);
    usleeptime = (endtime.tv_sec - starttime.tv_sec)*1000000 + endtime.tv_usec - starttime.tv_usec;
    if (usleeptime < (usweeptime)-1000)
    {
      usleep(usweeptime-usleeptime);
    }
    //sleep(1);
  }

  /* Clean up functions for exit from program.  First sleep for a second to makes
     sure all threads complete tasks */
  printf("Done!\n");
  sleep(1);
  // Send NULL to thread 1
  pthread_mutex_lock(&dataup_mutex);
  free(dataup.data);
  dataup.data = NULL;
  pthread_mutex_unlock(&dataup_mutex);
  // Send NULL to thread 2
  pthread_mutex_lock(&datadn_mutex);
  free(datadn.data);
  datadn.data = NULL;
  pthread_mutex_unlock(&datadn_mutex);
  // Close sound card
  snd_pcm_close(pcm_handle);
  /* wake up threads and wait until they terminate */
  pthread_mutex_lock(&thread1_mutex);
  pthread_cond_signal(&thread1_cond);
  pthread_mutex_unlock(&thread1_mutex);
  pthread_join(thread1, NULL);
  pthread_mutex_lock(&thread2_mutex);
  pthread_cond_signal(&thread2_cond);
  pthread_mutex_unlock(&thread2_mutex);
```

```
    pthread_join(thread2, NULL);
    // close data file
    pthread_mutex_lock(&dataf_mutex);
    gzclose(fd);
    pthread_mutex_unlock(&dataf_mutex);
    ResetDDS();

    return 0;
}
```