Time-Frequency Analysis of Synthetic Aperture Radar Signals

Brooks Johnston
(MS Thesis)

August 1, 1996
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
Time-Frequency Analysis of Synthetic Aperture Radar Signals

Brooks Johnston
(MS Thesis)

Manuscript date: August 1, 1996
Time-Frequency Analysis of
Synthetic Aperture Radar Signals

By

Brooks Johnston

B.S. (University of California, Davis) 1994

THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical and Computer Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Gary E. Ford, Ph.D. - Committee Chair ________________________________

V. Ralph Algazi, Ph.D - Committee Member _____________________________

Richard E. Twogood, Ph.D - Committee Member _________________________

James V. Candy, Ph.D - Technical Advisor ______________________________

Committee in Charge

1996

i
Abstract

Synthetic aperture radar (SAR) has become an important tool for remote sensing of the environment. SAR is a set of digital signal processing algorithms that are used to focus the signal returned to the radar because radar systems in themselves cannot produce the high resolution images required in remote sensing applications. To reconstruct an image, several parameters must be estimated and the quality of output image depends on the degree of accuracy of these parameters. In this thesis, we derive the fundamental SAR algorithms and concentrate on the estimation of one of its critical parameters. We show that the common technique for estimating this particular parameter can sometimes lead to erroneous results and reduced quality images. We also employ time-frequency analysis techniques to examine variations in the radar signals caused by platform motion and show how these results can be used to improve output image quality.
ACKNOWLEDGMENTS

I owe a great deal of thanks to those who assisted and supported me during my thesis work. Dr. James V. Candy deserves the greatest thanks for challenging me with new ideas and for his patience and guidance throughout the writing of this thesis. I would also like to thank the members of my thesis committee: Dr. Gary Ford, Chair of my committee and Dr. Ralph Algazi of CIPIC, and Dr. Richard Twogood of the Imaging and Detection Program (IDP) at Lawrence Livermore National Laboratory. I would also like to express my gratitude to Dr. Douglas Lake from the Office of Naval Research for support received to perform the work involved in writing this thesis.

I am indebted to the leadership of IDP, Dr. Twogood and Jim Brase for their support, and also the members of IDP, especially those with direct contributions to my work. I'm grateful for the help I've received from Dr. David Chambers, Dr. Dennis Goodman, Mr. Holger Jones and Ms. Carmen Mullenhoff for sharing their insights into the complexities of radar imaging. I also appreciate the input from the “guys” down the hallway: Mel, Jeff and Lee.

Finally, I would like to thank my wife Sandy for supporting me and enduring the long hours while I was “locked up in my room”. Most thanks of all go to God, who “holds all things together” and without whom, none of this would have been possible.
1 INTRODUCTION

A radar is a system that emits radiation in the microwave region of the electromagnetic spectrum. Since it provides its' own illumination, the radar operates equally well day or night and is not significantly affected by fog, clouds or precipitation. These properties of radar make it highly desirable for remote sensing applications.

Radar systems have been an important tool for remote sensing since they were first widely used in the World War II era. At that time, the primary role of radar was the remote detection of airplanes and ships. For many years, radar played the important but simplistic role of determining relative distances to targets. More recently, however, substantial progress has been made in the understanding of the signal processing required to make the radar system capable of reconstructing high resolution images from low resolution microwaves. In Chapter 2 of this thesis, we briefly discuss radar system design and engineering issues to provide a basis from which to build an imaging system. These topics are discussed thoroughly in Skolnik [1] and Eaves [2]. We then develop the principles of synthetic aperture radar (SAR), the set of digital signal processing algorithms from which we obtain high resolution images. The literature on SAR processing is often confined to a narrow topic but some references which provide an introductory overview are Curlander[3], Oliver[4], Hovanessian[5] and Fitch[6].

The SAR algorithms require the estimation of certain parameters to reconstruct an image properly. The latter half of Chapter 2 develops the common method for estima-
ing the parameter known as the doppler centroid. Various papers such as Barber[7] and Munson[8] describe the implementation of a SAR processor and the issues surrounding the estimation its parameters.

Estimation of the doppler centroid is usually accomplished by examining the doppler frequency spectrum, calculated using the Fast Fourier Transform (FFT). We will show in Chapter 3 that in some circumstances, this method for determining the doppler centroid suffers from a lack of performance. This performance can be evaluated under different circumstances by using time-frequency analysis and parametric (model-based) signal processing techniques. The literature in time-frequency analysis is well developed in texts such as Oppenheim[9], Cohen[10], Deller[11] and Rabiner[12]. Model based signal processing and spectral estimation techniques are detailed in Candy[13] and Kay[14], respectively.

Finally, in Chapter 4 we present the results of simulations of SAR data. A model of the radar described in Chapter 2 is simulated to create return image data sets. The data sets contain the signals received from a radar viewing a point object in space, the most simple scenario. The radar data are then analyzed using time-frequency analysis techniques that we developed in Chapter 3.

In Appendix A, we present a time-frequency tool that was created using the MATLAB [16] software package by implementing the algorithms derived in Chapter 3 of this thesis. It is used as an aid in understanding the time-varying spectral shifts seen in SAR data.
2 FUNDAMENTALS OF RADAR IMAGING

RADAR is an acronym used for Radio Detection And Ranging. As mentioned, it was primarily used to detect ships and airplanes during World War II. However, with the evolution of modern remote sensing platforms (satellite, aircraft, drones, etc.) the application of radar as an imaging system has evolved as a technology of high interest both from a military (reconnaissance) and non-military (forest growth, earth surface mapping) perspective. A drawback of using radar as an imaging source is the low resolution obtained as a result of the long wavelength microwaves. The goal of radar image processing is to reconstruct the low resolution images produced by the radar into high resolution images that resemble photographs from the visible spectrum. To scan large areas, radars are often mounted on moving platforms remote from the area of interest. As the platform travels, the radar transmits, receives and stores the signals. The signals in modern radars are digitized so that digital signal and image processing techniques may be employed to reconstruct the image.
2.1 Transmitting and Receiving Radar Signals

Radars emit energy either continuously or in pulses. Continuous wave (CW) systems are generally used for determining target range and position, and pulsed signals are usually used for imaging systems. As was mentioned, most radar imaging systems are mounted on moving aircraft platforms such as satellites or airplanes. The radar scans a region by pulsing it with electromagnetic waves, and then receiving and digitally processing the waves as they return from reflective objects on the planet’s surface. The direction that the platform is moving is called the azimuth while the direction orthogonal to motion is called range. The electromagnetic pulse is characterized by its power, bandwidth and shape as it is emitted from the radar antenna.

The power of the pulse decreases with the distance squared and when it illuminates a point target, the signal power reflected from the target again decreases as the distance squared. The received power is then on the order of $\frac{P_t}{r^4}$ where $P_t$ is the transmitted power and $r$ is the distance from the transmitter to the target. Power as a function of position on the illuminated surface is known as the antenna gain pattern. If the transmitter and receiver share the same physical antenna, the radar is called monostatic, whereas if they are in different locations it is called a bistatic configuration.

The bandwidth of the pulsed radar signal is a function of the rise time of the transmitted pulse, that is, time required to raise the output power on the antenna from a ‘low’ state to a ‘high’ state as shown in Figure 1. Increasing the slope of the leading edge in order to reduce the rise time $\tau_r$ adds higher frequency components to the output signal and reduces the pulse length. The difficulty in building electronics
with high bandwidth as well as high output power has prompted engineers to use narrowband methods of radar pulsing. The narrowband signals are phase coded, i.e. the output phase changes with a predetermined function. A common coding scheme is called linear frequency modulation (LFM) or “chirp”. This function starts with a narrowband signal and linearly increases the frequency with time as shown in Figure 2. Linear FM is described mathematically by

\[ s(t) = \cos \left( 2\pi \left( f_c t + \frac{Kt^2}{2} \right) \right) \]  \hspace{1cm} (1)

where \( f_c \) is the carrier frequency, and \( K \) is the chirp rate or slope of the linear modulation. The high frequency carrier takes advantage of the all weather properties of
microwaves and the chirp encoding can be processed to improve range resolution which we describe subsequently.

The shape of the beam pattern is considered to be conical, with conic angle

$$\theta = \frac{\lambda}{L}$$

(2)

where $\lambda$ is the wavelength of the carrier frequency and $L$ is the length of the antenna (see Figure 3). The outer surface of the cone is defined as the point at which the power is half of that in the center of the cone. The ground surface illuminated by the cone is called the radar footprint.

The most important capability of an imaging radar is its ability to resolve objects which lie in close proximity to one another both in the range direction as well as the azimuth direction. These parameters are appropriately called range and azimuth resolution.
2.1.1 Range Resolution

Figure 3: **Range resolution.** *The radar illuminates a non-uniform footprint as it travels along its path.*

*Range resolution* is defined by the ability of the radar system to discern two reflective objects lying in close proximity in the range direction. The pulse duration $\tau_p$ and the look angle $\eta$ establish the range resolution of the radar by

$$\Delta R_g = \frac{\Delta R}{\sin(\eta)}$$  \hspace{1cm} (3)
where $\Delta R = c \tau_p$ is the resolution in slant range and the look angle $\eta$ is the angle between the radar beam and the normal to the earth's surface (see Figure 3). This equation states mathematically that which is already intuitive: as the look angle *increases* and the radar beam is pointed further away, the size of the radar footprint increases and the resolution *decreases*. Also, as the pulse-width increases, the area illuminated by the pulse increases and the smearing of the return signal from a point target increases. That is, two targets separated by less than $\Delta R_g$ will return overlapping signals. From the figure we can also define the *slant range*, $R$ as the distance from the radar to target and *ground range*, $R_g$ as the horizontal projection of $R$.

### 2.1.2 Azimuth Resolution

Azimuth resolution is the distance at which two targets lying along the track of the radar can be distinguished. Therefore the resolution is defined as the *width* of the radar footprint as shown in Figure 4. Since the width of the footprint is subject to the conical angle $\theta$, it is therefore a function of the length of the antenna. In particular,

$$W = 2R \sin \left( \frac{\theta}{2} \right) \approx \frac{\lambda R}{L}$$  \hspace{1cm} (4)$$

where $W$ is the footprint width, $R$ is the slant range to target, $\lambda$ is the carrier wavelength, $L$ is the length of the antenna and as before $\theta$ is the conic angle. The azimuth resolution of the radar system can be improved by reducing the size of the footprint. However, to reduce the size of the footprint for a particular $\lambda$, the length of the antenna $L$ must be increased or $R$ decreased. Since the slant range is often constrained by the type of aircraft platform and the specific imaging application, the antenna length
Figure 4: **Azimuth resolution.** The radar footprint simultaneously illuminates targets separated by $W$.

must be increased in order to resolve smaller features. In many cases, the resolution requirement of the imaging system calls for extraordinarily large antenna sizes.

Fortunately, the resolution in both range and azimuth directions can be increased for this system using digital signal processing techniques. The problem is *separable* and can be treated first in the range direction and next in the azimuth direction. The following sections detail a signal processing system to obtain a high resolution image from the radar system.
2.2 Improving Image Resolution

Figure 5 depicts an overview of a high resolution radar imaging system. To produce an image, a linear chirp signal is generated and modulated (i.e. mixed, or multiplied) onto the carrier frequency. The signal is then amplified and transmitted through the radar antenna. Scatterers in the scene reflect the chirped pulse back to the receiver, where it is demodulated to obtain the chirp signal. The received signal is recorded with two components: the in-phase and quadrature, collectively known as I/Q. The quadrature component is recorded 90 degrees out of phase with respect to the in-phase component, retaining a record of the phase of the input signal. As we will see, the phase history is a crucial element to the generation of high resolution images. In the diagram above, digital signal processing is used to improve the resolution, first in the range direction and then in the azimuth direction. The details of the signal processing...
block are discussed in this section.

2.2.1 Improved Range Resolution

From the previous discussion, the resolution in the range direction does not depend on the size of the antenna. The dominant factor is the pulse width which is determined by the radar hardware. This resolution can be increased however by using digital processing techniques on the received signal. The term used for the signal processing in the range direction is *pulse compression* and is illustrated in Figure 6.

---

**Figure 6: Pulse compression.** The wide pulse-width in (a) illuminates both targets simultaneously and returns the pulse shown. The compressed pulse-width (b) illuminates only one target at a time.

---

To compress the pulse-width a matched filter is used. A *matched filter* is simply the
convolution of the input data stream with a filter or replica designed for a particular 
"shape". When the shape of the data "matches" that of the replica, a large output 
results; otherwise the filter produces relatively small outputs.

![Convolution Examples](image)

Figure 7: **Matched filtering.** Matched filter outputs for three different 
signals. a) Width of cosine burst is equal to original burst. b) Width 
of chirp burst has narrow convolution output. c) Ideal high bandwidth 
burst.

Examples of matched filter replicas and their corresponding matched filter outputs 
are shown in Figure 7. Note that the narrowband output is the same duration as the 
original signal, so there is no effective time compression. The chirped pulse however,

12
has an output which is compressed in time after matched filtering. The smallest time
duration, and therefore the greatest range resolution, is obtained by the wideband
pulse. Though the wideband pulse has the ideal resolution characteristic it is diffi-
cult to produce, especially with high power and is impractical to implement in many
applications.

The shape of the replica used in the matched filter is that of the impulse response of
the radar. The impulse response can be calculated by using known parameters. For a
chirped output signal, the parameters are the chirp bandwidth and the pulse duration.

The matched filter is implemented by time reversing and conjugating the impulse
response. When the signal is digitized, the reverse time impulse response 'slides' along
the input data stream summing the products of the two signals to produce the com-
pressed pulse return. This is best described mathematically by

\[ g(t) = \int_{-\frac{T_p}{2}}^{\frac{T_p}{2}} s^*(t - t')r(t')dt' \]  

(5)

where \( s^*(-t) \) is the time-reversed complex conjugate of the impulse response and \( r(t) \)
is the received data.

2.2.2 Range Processing Algorithm

We now develop a procedure to process the time delayed signal by first basebanding,
and then “time compressing” the returned signal using matched filtering. Consider the
transmitted chirp or linear FM signal

\[ s(t) = \cos \left[ 2\pi \left( f_c t + \frac{Kt^2}{2} \right) \right] \]  

(6)
where \( f_c \) is the carrier frequency and \( K \) is the chirp rate. The received pulse is delayed by time \( \tau_d = \frac{2R}{c} \) where \( R \) is the slant range to a point target and \( c \) is the speed of light.

The result is

\[
    r(t) = \cos \left[ 2\pi \left( f_c (t - \tau_d) + \frac{K(t - \tau_d)^2}{2} \right) \right]
\]  

(7)

and since the wavelength \( \lambda = \frac{c}{f_c} \), then

\[
    f_c (t - \tau_d) = f_c t - \frac{2R}{\lambda}
\]  

(8)

we can write

\[
    r(t) = \cos \left[ 2\pi \left( f_c t - \frac{2R}{\lambda} + \frac{1}{2} K \left( t - \frac{2R}{c} \right)^2 \right) \right]
\]  

(9)

To extract the chirp, this signal is basebanded by multiplying by the carrier \( f_c \) term

\[
    r_b(t) = \cos(2\pi[f_c t + \phi(t)]) \cos(2\pi f_c t)
\]  

(10)

where \( r_b \) is the basebanded return signal and for generality, the chirp signal has been symbolized by \( \phi(t) \). The basebanded result can easily be computed by using the complex exponential equivalent

\[
    r_b(t) = \frac{1}{2} \left[ e^{-j(\omega_c t + \phi(t))} + e^{j(\omega_c t - \phi(t))} \right] \cdot \frac{1}{2} \left[ e^{-j\omega_c t} + e^{j\omega_c t} \right]
\]

\[
    = \frac{1}{4} \left[ e^{-j(\omega_c t + \phi(t)) + \omega_c t} + e^{-j((\omega_c t + \phi(t)) - \omega_c t)} + e^{j((\omega_c t + \phi(t)) - \omega_c t)} + e^{j((\omega_c t + \phi(t)) + \omega_c t)} \right]
\]

\[
    = \frac{1}{4} \left[ e^{-j(2\omega_c t + j\phi(t))} + e^{-j\phi(t)} + e^{j\phi(t)} + e^{-j(2\omega_c t + j\phi(t))} \right]
\]  

(11)

The terms containing \( 2\omega_c \) can be removed by low pass filtering so that the result of basebanding becomes

\[
    r_b(t) = \frac{1}{2} \left[ e^{-j\phi(t)} + e^{j\phi(t)} \right]
\]  

(12)
Substituting the results from basebanding of Eqn. 9 we can write in terms of the positive frequencies

$$r(t) = Ae^{-j\frac{4\pi R}{\lambda}} e^{j\pi K(t-\frac{2R}{c})^{2}}$$  \hspace{1cm} (13)

To compress, we want to convolve the received signal with the conjugated form of the transmitted signal

$$h(t) = e^{j\pi K t^2}$$  \hspace{1cm} (14)

to obtain

$$g(t) = e^{-j\frac{4\pi R}{\lambda}} \text{sinc} \left( \pi K \tau_p \left( t - \frac{2R}{c} \right) \right)$$  \hspace{1cm} (15)

which can be approximated (for details see [3]) by

$$g(t) = e^{-j\frac{4\pi R}{\lambda}} \delta \left( t - \frac{2R}{c} \right).$$  \hspace{1cm} (16)

This gives us the phase and range bin of the reflector.

Since the radar platform is moving and taking measurements at a spatial interval $ds$, the range compression function $g(t)$ can be written as a two dimensional function. One dimension is still the time in which the radar signal propagates and the other is the axis of radar platform motion. We can write the location of the point target relative to the radar platform as

$$f(s, \tau_d) = e^{-j\frac{4\pi R}{\lambda}} \delta \left( t - \frac{2R}{c} \right)$$  \hspace{1cm} (17)

where $s$ is the position along the radar track and $\tau_d$ is the round trip delay time of the signal. For consistency (and convention) we make both parameters of $f$ spatial, i.e.

$$f(s, R) = e^{-j\frac{4\pi R}{\lambda}} \delta \left( t - \frac{2R}{c} \right).$$  \hspace{1cm} (18)
where $R = \frac{2\pi c}{\lambda}$. Thus, the time bin of the compressed pulse is found at the delay interval $\frac{2R}{c}$ with phase $e^{-j\frac{4\pi R}{\lambda}}$.

In summary, the scatterer reflects the transmitted chirp back to the SAR processor where it is first demodulated and then "compressed" in range using a matched-filter. Therefore, the range compressed returned wavefront can be described by

$$g(s, R) = e^{-j\frac{4\pi R(s)}{\lambda}},$$

and $R(s) = R_0 + \Delta R(s)$ where as before $s$ represents position along the azimuth track.

### 2.2.3 Summary of Range Processing

Figure 8 reviews the processing procedure described in this section which was used to improve the resolution of the radar signal in the range direction. First, a high bandwidth chirp signal is generated. The chirp is modulated with a carrier frequency $f_c$ to take advantage of the properties of microwaves. This signal is then amplified and broadcast through the output transmit antenna. The electromagnetic wave travels through the propagating medium (usually air and/or space) until it encounters a reflective object. The object scatters the wave back to the radar receiver which receives it $\frac{2R}{c}$ seconds after the pulse was first transmitted. The signal is demodulated or basebanded as discussed as above, so that $f_c$ is removed from the signal and only the chirp remains. The chirp is then matched filtered so that the point in time when the retransmitted pulse was received is more closely approximated. This sequence of events is repeated at a frequency determined by the desired azimuth resolution discussed in the next section.
2.2.4 Improved Azimuth Resolution

As seen in the range compressed image in Figure 8, a point target signal is smeared across several range lines for as long as the target remains in the footprint of the radar.

We seek to convert the smeared impulse response back to its original form, as a single point in the scene.

To improve azimuth resolution the radar beam could be focused by increasing the
length of the antenna. Recall that the resolution of the radar is

\[ W = R \sin \theta \approx \frac{R\lambda}{L} \]  

(20)

where \( W \) is the width of the footprint.

Equation 20 indicates that practical resolution is obtained only by extraordinarily large antennas. For example, a typical L-band (1.25GHz) radar imaging system mounted on satellite platform has wavelength \( \lambda = 24cm \) and \( R = 850Km \). For a resolution of 50 meters, an antenna 4Km long would be required.

Instead, alternative techniques can be used to significantly improve the azimuth resolution of the beam to \( \Delta x \) (as shown in Figure 9) where \( \Delta x \) represents the improved resolution. One method of improving resolution would be to create the single large antenna with an array of smaller antennas that would "sample" the return signal. The array elements would be spaced appropriately so that the surface of the large antenna was approximated by the smaller elements. Each element would need to be "phased" together, i.e. a received impulse would travel from each antenna over various length delay lines so that the signals would be combined simultaneously at the radar receiver. Figure 10 depicts the phase delayed arrival times of pulse returns at evenly spaced azimuth positions from a single point target passing across the radar footprint. The phases must be aligned before processing. As can be seen in the diagram, the returns from the point source arrive sooner when the platform is near the center of the radar cone, and later when the target is near the outside of the cone. Phasing can be done by digitally delaying each signal and since the data was recorded with phase history intact (i.e. I/Q recorded), the phase of each return is known. This type of processing
Figure 9: **Azimuth resolution.** For a radar emitting wavelengths $\lambda$ and with an antenna length $L$, the footprint width is $W$. Signal processing techniques can “compress” the beam width to $\Delta x$. In this diagram, two point targets are illuminated by the radar footprint simultaneously, but can still be resolved since their distance is greater than $\Delta x$.

is referred to as **azimuth compression**.

A synthetic aperture radar (SAR) uses the idea of an array of antennas, but uses only a single antenna mounted on a moving platform, sampling the returns of an object for as long as the object remains in the footprint of the radar. SAR achieves high resolution in the along track (azimuth) direction because the size of the synthetic antenna can be as large as the footprint of the radar. To see this, consider the antenna
in Figure 11 which depicts the physical radar in locations P1 and P2. At location P1, the point target is first illuminated by the radar. At location P2, the point target reflects the final return to the receiver. The distance \( D = P2 - P1 \) is the width of the synthetic antenna. If the antenna is pointed orthogonally to the direction of platform travel (i.e. broadside), the point target will be exactly in the center. Recalling that the footprint of the radar has width

\[
W \approx \frac{R\lambda}{L}
\]  

(21)

which is inversely proportional to the length of the physical antenna, we now see that
Figure 11: Size of synthetic aperture. \( P1 \) is the point at which the radar first illuminates the point target. \( P2 \) is the last point in which the radar illuminates the target. The SAR aperture has width \( D \).

The width of the synthetic antenna is also inversely proportional to the length of the antenna by

\[
L_{\text{synthetic}} = W = \frac{R\lambda}{L}
\]  \hspace{1cm} (22)

where \( L_{\text{synthetic}} \) is the length of the synthetic aperture. We obtain the width of the footprint from the synthetic antenna by substituting \( L_{\text{synthetic}} \) into Eqn. 21 to get

\[
W_{\text{synthetic}} = \frac{R\lambda}{L_{\text{synthetic}}} = L.
\]  \hspace{1cm} (23)

This is the surprising result that allows SAR to create high resolution images: the
smaller the size of the physical antenna, the greater the resolution.

2.2.5 Azimuth Processing Algorithm

As the radar moves along its’ path illuminating the scene on one side of the track, a point target within the footprint of the radar returns a signal proportional in time to the distance between antenna and target. As the target approaches broadside, the distance minimizes and after passing broadside the distance increases again until it is out of the radar cone. Figure 12 depicts what is known as range migration. The target migrates closer and then farther from the radar as the radar approaches and then passes the targets location. The azimuth impulse response is the phase differences as a function of position and can be used in a matched filter for compression. If the distance
between the maximum and the minimum range is more than the range resolution, the returns appear to be in different (but adjacent) range bins. In these cases, special attention is required to apply the matched filter to the correct data points. This is known as range curvature correction.

We first need to develop the replica for the azimuth direction by examining the phase relationship to the azimuth position $s$.

Recall that the pulse compressed returned wavefront was described by

$$g(s, R) = e^{-j \frac{4\pi R(s)}{\lambda}}$$  \hspace{1cm} (24)$$

and from Figure 13 $R(s) = R_0 + \Delta R(s)$ where $s$ represents position along the azimuth.

Figure 13: SAR phase delay. Phase of incoming signal as a function of the range $R(s)$. The point target enters the radar cone at $s_{\text{enter}}$, is at broadside at $s_0$ and exits the cone at $s_{\text{exit}}$.  

Point Reflector
track. The phase is
\[ \phi(s) \equiv -2\pi \left( \frac{2R(s)}{\lambda} \right) \] (25)
and its derivative is
\[ \phi'(s) = -2\pi \left( \frac{2R'(s)}{\lambda} \right) \] (26)
which represents the change of phase with respect to position \( s \). The second derivative of \( \phi(s) \) is
\[ \phi''(s) = -2\pi \left( \frac{2R''(s)}{\lambda} \right) \] (27)
which represents the rate of phase change with respect to position \( s \). From these derivatives, two important parameters of SAR are defined:
\[ f_d \equiv -\frac{2R'(s)}{\lambda} \] (28)
which is known as the doppler frequency and
\[ f_r \equiv -\frac{2R''(s)}{\lambda} \] (29)
which is known as the doppler rate. Solving for \( R' \) and \( R'' \) in terms of \( f_d \) and \( f_r \) we have
\[ R' = -\frac{\lambda f_d}{2} \] (30)
\[ R'' = -\frac{\lambda f_r}{2} \] (31)
where \( R' \) is the velocity vector to a point target and \( R'' \) is the corresponding acceleration. The doppler frequency and rate can approximate components of the range equation. To see this, consider the second order Taylor series about \( s_0 \)
\[ R(s) \approx R_0 + R'(s - s_0) + \frac{R''(s - s_0)^2}{2} \] (32)
Equations 30, 31 and 32 can be used in Eqn. 24 to obtain

\[ g(s, R) = e^{-j 4\pi R_0 / \lambda} e^{-j 2\pi \left( f_{dc} (s - s_0) + \frac{1}{2} f_r (s - s_0)^2 \right)} \] (33)

The quadratic term in this equation indicates that like the range compression signal, the azimuth signal is a linear FM chirp with center frequency \( f_{dc} \) at \( s = s_0 \). To compress a point at \((s, R)\) we need to first construct the replica. Recall that in the range compression algorithm we first basebanded the input signal to remove the modulating frequency \( f_c \). The azimuth signal in Eqn. 33 also has a modulating signal: \( e^{-j \frac{4\pi R_0}{\lambda}} \). To demodulate we multiply our input by \( e^{j \frac{4\pi R_0}{\lambda}} \) by selecting the point in range phase-delayed by \( \frac{2R_0}{\lambda} \). The result is conjugated to obtain the matched filter replica function

\[ h^{-1}(s, R) = e^{j 2\pi \left( f_{dc} s + \frac{1}{2} f_r s^2 \right)} \] (34)

Matched filtering is performed just as it was for range processing, by convolving the replica with the returned data. If the parameters \( f_{dc} \) and \( f_r \) are selected correctly, the replica will match the data and the filter will output a signal indicating the azimuth position of the point target. The arc seen in the range compressed image will be compressed to a single point representing the point target in the output image.

2.2.6 Summary of Azimuth Processing

Figure 14 shows the processing procedure used to compress a number of accumulated range processed signals. The number \( N_p \) represents the number of pulses which illuminate a point target on the ground. The received range processed image is \( g(s, R) \). An azimuth replica is formed based on the doppler frequency and rate and is convolved with corresponding points in the range compressed image \( g(s, R) \). In the case shown
above, there is only a single point target at location \((s_t, R_t)\) in the scene, the matched filter has only one “match” and outputs accordingly at the final image output location \(I(s, R)\).

2.3 SAR Parameter Estimation

It is clear from Eqn. 34 why the parameters \(f_{dc}\) and \(f_r\) are critical in SAR processing. These parameters must be accurately estimated to create a matched filter and thus focus the image correctly, that is, to align the phase arrivals of the reflected pulses. Analytical methods for determining the doppler parameters can be used but are limited
by the accuracy of the instrumentation used to obtain exact positional coordinates. Methods have been developed to estimate the SAR parameters directly from the data, thus bypassing the need for analytical constraints. The procedures used to estimate $f_{dc}$ and $f_r$ are called clutterlock and autofocus respectively.

2.3.1 Clutterlock

The parameter $f_{dc}$ characterizes the linear portion of the range curvature called the range walk. Range walk is most often caused by radar "squinting", i.e. an angle $\beta \neq 90$ degrees exists between the center of the radar beam and the track (see Figure 3). The squint angle of an airborne radar platform is continuously changing due to accelerations of the aircraft. The doppler center frequency will be a function of position along the travel path and must be estimated using a clutterlock algorithm to correctly focus the SAR image. The clutterlock algorithm gets its name from the fact that it examines the "clutter" in the scene to estimate the doppler center frequency. Clutter is defined as the non-descript scatterers positioned randomly in the field of view of the radar as shown in Figure 15.

Recall that the doppler frequency response is defined as

$$f_d \equiv -\frac{2R'(s)}{\lambda}$$  \hspace{1cm} (35)

where $R'(s)$ is the velocity with respect to a point target and $s_0$ is the azimuth position at which the target is broadside. For a cluttered scene, many point scatterers exist in the radar footprint and a broad spectrum of narrowband frequencies are generated as the airborne platform passes by. This is because at each instant all of the illuminated
Figure 15: Clutter scene and subsequent radar returns.

Point scatterers lie at different ranges and therefore represent different velocities in Eqn. 35. Figure 16 illustrates this idea by mapping doppler frequencies to regions of the radar footprint. Each scatterer therefore returns its own characteristic chirp at different phases simultaneously to the receiver.

By convention, the azimuth axis is labeled with forward positions as positive numbers and the rear positions as negative numbers as shown in Figure 17. From Eqn. 35, we see that scattered returns from broadside clutter are characterized by low fre-
Figure 16: **Doppler frequency map.** Equal doppler frequencies exist along the contours shown above.

Equal Doppler Frequencies exist because $R'(s)$ is at its minimum, while the returns from scatterers near the edges of the radar cone are at maximum range and have higher frequency content. All of the clutter returns are attenuated by the antenna gain pattern. Assuming an all clutter scene, the clutter spectrum appears as white noise attenuated by the gain pattern of the antenna. Estimation of $f_{dc}$ with clutterlock relies on the premise that as the platform is squinted forward or backward, $R'(s)$ increases or decreases respectively and the doppler center frequency will be offset accordingly as shown in Figure 18.
To estimate $f_{dc}$, a row of the range compressed image data is extracted and transformed into the frequency domain (see Figure 19). The resulting spectrum is assumed to represent the spectrum of the doppler frequency of an equivalent point target because the clutter is randomly distributed. That is, statistically, a row of noisy image data contains the portions of the azimuth chirp appropriate at $R(s)$, and the Fourier transform will therefore represent the doppler spectrum correctly. As shown in Figure 20, the frequency value at the spectral peak is then used for $f_{dc}$ since it represents the center of the antenna pattern, and therefore indicates the squint angle of the radar at each position $s$. Thus, $f_{dc}$ is then used in Eqn. 34 to focus the SAR image. In some cases, the platform motion which causes the squint angle to change cannot be considered constant over the length of the synthetic aperture. The doppler frequency is then no longer linearly increasing but a position dependent function, as shown in Figure 21.
Local perturbations in the azimuth chirp can be seen by examining the spectrum local to the region of the motion. A small disturbance in the platform motion causes range variations in the next example. In the second diagram in Figure 21 we see that the disturbance causes a range deviation in the simulated data. From this data, we can see that the disturbance happens when the target is ahead of the platform and that the platform momentarily shifts away from the target and then back. This is the cause of

Figure 18: **Doppler spectra.** (a) *Doppler spectrum of a broadside (zero squint) azimuth row as seen through the antenna gain pattern.* (b) *Doppler spectrum of a forward squinted azimuth row.*
the short duration increase in range and subsequently, a change in spectral content.
Figure 20: Estimation of doppler center frequency.

Figure 21: A small perturbation in platform motion causes range variation.
2.3.2 Autofocus

Most SAR processors estimate the quadratic term $f_r$ of the chirp frequency in an entirely different way from the linear term $f_{dc}$. These processors use the fact that the image contrast decreases if the velocity parameter is incorrectly chosen. That is, the frequency rate can be approximated\cite{3} by the model

$$f_r = -\frac{2V^2}{\lambda R}$$

if the velocity is known. In general, the instantaneous velocity is not known exactly so successive approximations of $f_r$ are made with guesses of the velocity near the approximate value. The closest guess to the correct velocity will yield the highest contrast image. This is known as the autofocus algorithm. In practice, only a small portion of the image is autofocused with the assumption that this region provides an accurate representation of the image statistics of the image for which $f_r$ is being approximated.

Another method for obtaining the frequency rate in the special case of a single point target would be to measure the difference in spectral peaks over the synthetic aperture. This direct method of measuring $f_r$ is easily made when point target returns can be tracked. Multiple point targets, clutter and the inability to focus the image except where point targets are located severely limit the usefulness of this technique. An example of the single point target case is presented in the results section of this thesis and the more difficult case of multiple scatterers is left for future work.
2.4 Summary of SAR Algorithms

To be used in imaging applications, a radar must have either a very large antenna or use signal processing techniques to improve its resolution. In particular, range compression and azimuth compression represent the principle SAR algorithms. Range compression uses a matched filter based on the radar’s linear frequency modulated (chirp) signal to effectively reduce the width (i.e. the duration) of the signal returned from a target. The range compressed signal is

\[ g(t) = e^{-j \frac{4\pi R}{\lambda} \delta \left( t - \frac{2R}{c} \right)} \]  (37)

which identifies the target at a range \( R \) with a phase \( e^{-j \frac{4\pi R}{\lambda}} \). The imaging radar travels along a track sampling the scene by transmitting and receiving the return signals. The resulting two dimensional image is given by

\[ g(s, R) = e^{-j \frac{4\pi R(s)}{\lambda}} \]  (38)

where \( s \) is the azimuthal position and \( R(s) \) is the function describing the range to the target as the radar passes by.

Azimuth compression also uses a matched filter to improve image resolution. The algorithm makes use of the fact that as the radar passes by a point target, the range changes and therefore the phase of the incoming return signals is also changing. More specifically, the range to target first decreases, reaches a minimum and then increases as the radar platform passes by the target. The phase of the return signals increases over the entire aperture. A matched filter must be created to compress the signal and the critical parameters of the filter are the doppler frequency \( f_d \) and the doppler rate...
$f_r$. The doppler frequency can be estimated by using a technique known as clutterlock. This method estimates the center of the doppler spectrum for a clutter scene by finding the peak spectral value. Since the spectrum of a clutter scene is wideband, the center of the spectrum represents the center of the antenna gain pattern and therefore the pointing vector of the radar platform. This technique is vulnerable to noise in the estimated spectrum.

The frequency rate is often estimated by iteratively selecting a value for $f_r$, reconstructing a portion of image, and calculating the contrast. The value of $f_r$ yielding the highest contrast is then used for reconstructing the entire image. This technique is known as autofocus. In the special case of a single point target in the scene, the frequency rate can be measured by measuring the slope of the spectral peaks.
3 DOPPLER FREQUENCY ESTIMATION AND TIME-FREQUENCY ANALYSIS

In order to more closely examine the doppler frequency, time-frequency methods of spectral estimation can be used. An effective tool used to analyze changes in spectral content over time is called a spectrogram. It assumes that frequency content of an input data stream is stationary for the time window for which the spectrum is computed. The window is then shifted down the input stream to view a new spectrum. As this process is repeated, an indication of the changes in spectral content can be seen by viewing the computed spectra side by side. In the case of doppler frequency we expect to see a linear ramp as a result of the azimuth chirp signal. The FFT is usually used to estimate the spectral content of an azimuth row of SAR data, but because the data is inherently noisy the estimated spectrum is also noisy.

In this section, we develop spectrogram estimation methods for the analysis of SAR parameter estimation. Specifically, we examine the FFT in a noisy environment and then compare the results to an optimized spectral estimator. Also, a tool created for spectrogram analysis in the MATLAB signal processing software package [16] is
3.1 Calculating the Spectrogram

To calculate a spectrogram, an analog signal must first be sampled in time, quantized and stored in the computer’s memory. The \( n^{th} \) sample is designated by \( n \Delta t \) where \( \Delta t \) is the time interval between samples. Since the input signal \( y(n) \) could be of indefinite length, the computer must operate on sections of \( M \) data points at a time. After the \( M \) samples have accumulated, it can compute a Discrete Fourier Transform (DFT) of the data set in order to determine the frequency content of the signal

\[
Y(n, e^{j\omega}) = \sum_{m=0}^{M-1} y(n + m)e^{-j\omega m}
\]  

(39)

where \( Y(n, e^{j\omega}) \) is the Fourier transform of the windowed sequence starting at time \( n \) and continuing through time \( n + M \). This is equivalent to using a rectangular window to extract a frame of data and computing the DFT. After accumulating more samples, the computer can compute the next spectrum by sliding and transforming the window in the time direction as shown in Figure 22. Note that the window in the figure slides by less than \( M \) points, but could slide by any number of points so that the windows could overlap, have gaps in between or align side by side.

Rectangular windowing in the time domain causes large sidelobes to occur in the frequency domain, and therefore reduces the frequency resolving capability of the spectrogram. To obtain increased time localization of frequency components, a more suitable windowing function can be used such as the standard Hamming, Hanning or Blackman
Figure 22: Sliding window. Window begins at \(n\) and goes to \(n+M\), then slides to next window position, overlapping the end of the current window.

Windows [9]. This operation can be described mathematically by

\[
Y(n, e^{j\omega_k}) = \sum_{m=0}^{M-1} y(n+m)w(m)e^{-j\omega_km}
\]  

(40)

where \(w(m)\) is the non-zero window function which emphasizes the signal at the time of interest and suppresses it at all other times. This equation represents the Short-Time Fourier Transform (STFT) and by replacing the \(\omega_k\) with \(\left(\frac{2\pi k}{M}\right)\) we get the Short-Time Discrete Fourier Transform (STDFT)

\[
Y(n, k) = \sum_{m=0}^{M-1} y(n+m)w(m)e^{-j\left(\frac{2\pi km}{M}\right)}
\]  

(41)
The inverse is then

\[ y(n + m)w(m) = \frac{1}{M} \sum_{k=0}^{M-1} Y(n, k) e^{j\frac{2\pi k}{M}m} \]  

(42)

and since the window function was defined earlier as a non-zero sequence, the sequence \( y(n + m) \) can be obtained from this definition exactly by dividing through by \( w(m) \) (see [12], pg. 253)

\[ y(n + m) = \frac{1}{w(m)M} \sum_{k=0}^{M-1} Y(n, k) e^{j\frac{2\pi k}{M}m} \]  

(43)

The STDFT samples in the frequency domain at evenly spaced increments of \( \omega \) over the discrete time \( n \) to \( n + M - 1 \).

---

**Figure 23:** Bandpass filter bank. Bank \( N \) non-overlapping bandpass filters, creating a frequency vs. time output.

Since the STDFT is a function of discrete time and discrete frequency, it is analo-
gous to a digital signal which is processed through a bank of digital band pass filters with equally spaced center frequencies as depicted in Figure 23. The spectral characteristics of the time-dependent Fourier transform can be considered stationary for a small duration of time, depending on the application. The width of the window function must be chosen carefully if the STDFT is to work correctly. The width must be small enough so that the spectral components do not change significantly, but if it is too small then frequency resolution will be lost (i.e. the intervals between frequency samples will be too large). Conversely, the window width may be made large to preserve frequency resolution, but at the expense of event resolution (i.e. the ability to resolve to distinct frequency events in time). To improve event resolution, the window functions can be overlapped so that the time-varying frequency content is sampled more often. The amount of sliding should be carefully considered as well because it too has undesirable side effects if inappropriate values are chosen. If the window overlaps the previous window too much, any distinct frequency event will be blurred because it is represented in many windows. If the overlap is too small, the frequency event may only appear in one computed spectrum and therefore may not be easily recognized as an event.

Given the algorithm for the STDFT, the computation of the spectrogram is relatively simple. The input data is windowed using the chosen function, the DFT is applied and the results (the spectra for time n) are accumulated. The process is repeated until the input signal has been completely processed. The spectrogram can then be viewed as a two dimensional image of the accumulated STDFT results, as in Figure 24.
Figure 24: Spectrogram and corresponding time signal. As the spectrum changes in the time signal, the spectral changes appear in the time-frequency plot (i.e. the spectrogram).
3.2 Alternative Approaches to Spectrogram Estimation

When constructing the spectrogram, the idea of a "local spectrum" is central to resolving frequency events in time. The STDFT is well suited in this respect for the construction of a spectrogram and is indeed the most common method of doing so. A disadvantage of the STDFT is that when the transform of a signal containing random noise is computed, the resulting spectrum (and therefore the spectrogram) also contains noise. An alternate approach which uses parametric signal processing techniques, seeks to reduce the noise and enhance the signals in the spectrum. We first build a foundation of spectral relationships for random signals. We then use those relationships in creating an optimum FIR filter called MVDR.

3.2.1 Spectral Relationships of Random Signals

We must first determine a general expression for the spectrum of a FIR filter which is defined by the equation

\[ y(n) = \sum_{k=0}^{\infty} h(k)x(n - k) \]  (44)

Taking the Z-transform and dividing gives us the transfer function

\[ H(z) = \frac{Y(z)}{X(z)} \]  (45)

To determine the spectrum in terms of this transfer function, we must recall that the covariance of the input signal is defined by

\[ R_{xx}(m) = E\{x(k)x(k + m)\} \]  (46)

where we assume a case in which the input signal has zero mean. We would like to obtain the relationship of \( R_{yy}(n) \) to the input \( R_{xx}(n) \) and a transfer function \( h(n) \). We
find that if we convolve $R_{xx}(n)$ with $h(n)$ we get

$$R_{xx}(n) * h(n) = \sum_{k=0}^{\infty} R_{xx}(k)h(n - k)$$

$$= \sum_{k=0}^{\infty} E\{x(n)x(n - k)\}h(n - k)$$

$$= \sum_{k=0}^{\infty} E\{x(n - k)x(n)\}h(n - k)$$

$$= \sum_{k=0}^{\infty} E\{x(n - k)y(n)\}$$

which can be defined as $R_{xy}$. Convolving $R_{xy}$ with $h(n)$ in the same way will result in the relationship

$$R_{yy}(n) = h(n) * h(-n) * R_{xx}(n)$$

(48)

Applying the Wiener-Kintchine relationship we get the desired relationship

$$S_{yy}(z) = \sum_{k=-\infty}^{\infty} R_{yy}(k)z^{-k}$$

(49)

where $z = e^{j\Omega}$ or in the discrete-time Fourier frequency domain

$$S_{yy}(\Omega) = H(\Omega)H^*(\Omega)S_{xx}(\Omega)$$

(50)

We can now use this relationship to create an optimal FIR filter.

3.2.2 Estimating the Spectrum with Minimum-Variance Distortionless Response (MVDR)

The MVDR approach to estimating a spectrum assumes that the input signal $x(n)$ is composed of a linear combination of complex exponentials with additive noise $v(n)$ as shown in Figure 25 such that

$$x(n) = Ae^{j\Omega_m} + v(n)$$

(51)
Figure 25: **MVDR processing diagram.** A sinusoid with additive noise is processed through the MVDR FIR filter which minimizes the output power while constraining the gain to unity.

Note that $x(n)$ has zero mean since the sinusoid has exactly zero mean and the noise $v(n)$ is assumed to be white. MVDR estimates the power at each discrete frequency $\Omega_m = \left(\frac{2\pi}{M}\right) m$ with a bank of narrowband finite impulse response (FIR) filters as shown in Figure 26. This is the same analogy used in constructing the FFT spectrum but in this case the filter coefficients adapt at each time interval to minimize noise.

This is accomplished by suppressing the sidelobes of the corresponding window, and therefore the associated noise, by minimizing the output power while simultaneously constraining the gain of the filter to unity. Since the sidelobes are a result of windowing the input data stream, the MVDR filter can be considered as the optimal windowing function for minimizing power as shown in Figure 27. The coefficients of the subsequent filter are optimal for the frequency $\Omega_m$ at time $n$, and must be recomputed at each point $(\Omega_m, n)$.

This method of estimating the spectral response constructs a new set of unity-gain, minimum-variance narrowband filters for each windowed input signal segment. The frequency response for the $m^{th}$ filter (i.e. the $\Omega_m$ frequency bin) is defined as
Figure 26: **Time-varying filter bank.** Similar to the filter bank in Figure 23, this filter bank changes coefficients at each value of $n$ to minimize power.

\[
|H_m(n, \Omega_m)| = \sum_{k=0}^{N-1} h_m(k)e^{-j\Omega_m k} = 1
\]

and under this constraint we want to minimize the output power by adjusting the values $\hat{h}_m(k)$. The estimated spectrum is the convolution of the actual spectrum and our unity gain filter

\[
\hat{g}_m(n) = h_m(n) * x(n)
\]

and the variance is then

\[
\sigma^2 = E\{\hat{y}^2\} = E\{(h_m(n) * x(n))^2\}
\]
Figure 27: MVDR sidelobe suppression. Sidelobes in narrow-band frequency response the FFT window are suppressed when using the MVDR adapting filter technique by minimizing power.

since the process is zero-mean. At this point it is convenient to define the Toeplitz matrix for $0 \leq n \leq N - 1$

$$X = \begin{bmatrix}
x(0) & x(1) & x(2) & \cdots & x(N - 1) \\
x(1) & x(0) & x(1) \\
x(2) & x(1) & x(0) \\
\vdots & \ddots & \ddots \\
x(N - 1) & & & \cdots & x(0)
\end{bmatrix}$$

(55)
and
\[ \vec{h}' = [h_0 \ h_1 \ h_2 \ \cdots \ h_{N-1}] \] (56)

We use output power for the cost function
\[ J = \vec{y}^2 = (\vec{h}(n) * \vec{x}(n))^2 \] (57)
written in vector form
\[ J = (\vec{h}\vec{x})^2 = \vec{h}'R_{xx}\vec{h} \] (58)

where
\[ R_{xx} = XX' \] (59)

we follow Candy [13], minimizing the function with the constraint from Eqn. 52
\[ \vec{V}'(\Omega_m)\vec{h} = 1 \] (60)

where
\[ \vec{V}(\Omega_m) = [1 \ e^{-j\Omega_m} \ \cdots \ e^{-jN\Omega_m}] \] (61)

The constrained minimization problem can be solved with the use of the Lagrange multiplier technique. The Lagrange operator \( \Lambda \) is multiplied by the constraining function in Eqn. 60 to obtain
\[ J = \vec{h}'R_{xx}\vec{h} + \Lambda(\vec{V}'(\Omega_m)\vec{h} - 1) \] (62)

Taking the gradient with respect to \( \vec{h} \), setting to zero and then solving for \( \vec{h} \)
\[ \nabla_h J = 2R_{xx}\vec{h} + \Lambda\vec{V}'(\Omega_m) = 0 \] (63)
\[ \vec{h} = \frac{\Lambda}{2} R_{xx}^{-1}\vec{V}(\Omega_m) \] (64)

48
Substituting $\tilde{h}$ into Eqn. 60 we can solve for $\Lambda$:

$$\Lambda = \frac{2}{\tilde{V}'(\Omega_m) R_{xx}^{-1} \tilde{V}(\Omega_m)} \quad (65)$$

which can be used in Eqn. 64 to obtain the filter weights

$$\tilde{h} = \frac{R_{xx}^{-1} \tilde{V}(\Omega_m)}{\tilde{V}'(\Omega_m) R_{xx}^{-1} \tilde{V}(\Omega_m)} \quad (66)$$

$\Lambda$ and $\tilde{h}$ can be substituted into Eqn. 62 to obtain the variance of the MVDR spectral estimator

$$\sigma^2 = \frac{1}{\tilde{V}'(\Omega_m) R_{xx}^{-1} \tilde{V}(\Omega_m)} \quad (67)$$

Recall that our initial assumptions were that the input sequence was zero-mean and white and that the gain of the filter $|H(z)| = 1$, so from Eqn. 50 we have our final result

$$S_{MVDR}(n, \Omega_m) = \frac{1}{\tilde{V}'(\Omega_m) R_{xx}^{-1} \tilde{V}(\Omega_m)} \quad (68)$$

Reciprocating the denominator gives $S_{MVDR}(n, \Omega_m)$ as in Eqn. 68. Note that since the covariance matrix $R_{xx}$ is Toeplitz, it can be efficiently inverted using the Levinson-Durbin recursion [12], [13].
In Chapter 2, we derived the image formation equations central to synthetic aperture radar. In doing so, we required the estimation of certain parameters which we called the SAR parameters. Of particular interest was the parameter called the doppler center frequency or $f_{dc}$. This parameter was obtained by estimating the spectral peak of the azimuth chirp signal but as we saw in Chapter 3, the FFT method of spectral estimation can sometimes lead to erroneous results. In this chapter, we apply the time-frequency analysis developed in Chapter 3 to a simulated SAR azimuth chirp signal. We begin by providing the details of how the simulated data was created and validate the simulation by applying the SAR compression filters to it. The result is a focused image, measured by the size of a point target.

Since the scene is composed of a combination of point targets and clutter, we consider these at first separately to obtain ideal results and then together to measure the differences in doppler frequency. We then alter our model to simulate platform motion of the radar. Each deviation from the linear chirp in doppler frequency represents a
platform motion which can be quantified through the use of time-frequency methods.

4.1 Simulation using a Radar Model

A simulation of the radar system and SAR processor is presented here corresponding to the principles prepared in Chapter 2. A first order approximation is used to simulate the return image. By first order, we mean that secondary scattering mechanisms such as multiple bounce and Bragg scattering [3] are not included in the model. Figure 28 depicts the simulation in block diagram form.

A point target is the simplest reflector model and is used in the first block to validate the radar and SAR processor. The simulation creates the "forward projection", i.e. the two dimensional radar return of the point target, by calculating the arrival time of the range chirp and placing it in the columns of the image accordingly. Recall that the range chirp is defined by

\[
s(t) = \cos \left[ 2\pi \left( f_c t + \frac{K t^2}{2} \right) \right]
\]

Each return is calculated by using Eqn. 19 to form the complex equation

\[
\tau_d = e^{-j \frac{4\pi R}{\lambda}}
\]

which gives us the phase of the incoming signal. We assume in the simulation that the return signal has been basebanded as before in Eqn. 11 to remove the carrier frequency. Noise can optionally be added to the result to simulate instrumentation, thermal or quantization noise. The resulting forward projection is the point target smeared in the range direction because of the range pulse duration (i.e. the range impulse function)
and smeared in the azimuth direction due to the changes in range (the azimuth impulse function).

The SAR algorithms provide us with the compression filters needed to improve resolution. First, we apply pulse compression to the columns (range direction) of the range-azimuth image. Recall that the compression filter is the time-reversed conjugate
of the impulse response

\[ h(t) = e^{-j\pi R(t - \frac{2R}{c})^2} \]  

(71)

which is convolved with each column (position s) of the image to obtain the range compressed image \( g(s, R) \).

Next, we apply the azimuth compression filter

\[ h(s, R) = e^{-j2\pi(s + \frac{1}{2}f_r s^2)} \]  

(72)

to the range compressed data. We generate the azimuth chirp signal expected for the range at which the point target exists, time-reverse, conjugate, and convolve with a row of azimuth image data. If the range curvature exceeds the range bin of the point target, range curvature correction must be applied so that the convolution can be performed on contiguous computer memory addresses. If the SAR parameters \( f_{dc} \) and \( f_r \) are correctly chosen, azimuth compression will yield a small point size at the location of the original point target. This is our main objective: to estimate the SAR parameters \( f_{dc} \) and \( f_r \) using time-frequency techniques and therefore enhance the target localization.

4.2 Demonstration of Radar Model

We now demonstrate the simulated radar with sample outputs from a scene with one point target in a background free of clutter. In Figure 29 we see the point target at location \((R_0, s_0)\) in the scene located at \( R_0 = 4000m \) and \( s_0 = 0m \), with an aperture width of 844m. Range and azimuth directions are indicated in the figure, and we show the width of the synthetic aperture in the azimuth direction. The simulated radar
scans the scene in the azimuth direction, accumulating return signals from the point target. The final return image is shown in the figure. The returns from the point target are smeared both in the range direction (due to pulse width) and the azimuth direction (due to radar conical angle). Note that the antenna gain pattern attenuates the signal at the outsides of the aperture. Noise is added to the model at this point to simulate clutter and instrumentation uncertainty.

The first step of SAR processing is range compression. In Figure 30 we show a single compressed range bin in (a) which is the result of matched filtering. The radar emits a chirped frequency with a predefined starting frequency and chirp rate. A replica can easily be synthesized for a matched filter from these parameters and applied to the returned signal. The fully range compressed image is shown in (b).

The second step to SAR processing is azimuth compression. Since this is a simulation, we can create an ideal matched filter for azimuth compression because we know the velocity and pointing direction exactly at all times. In this example, the doppler center frequency $f_{dc} = 0$, i.e. the radar is pointed broadside. The azimuth doppler signal shown in Figure 31(a) is attenuated by the antenna gain pattern. The compressed output is shown in (b) of the figure and the azimuth compressed image is shown in 31(c). The effects of reconstruction can be seen more clearly in the magnified view of the image in Figure 32. In this figure, the sidelobes resulting from both the range and azimuth compression filters can be seen. The final result is the impulse (point target) response of the radar and signal processing system.

To see the effects of a doppler frequency shift, we simulate the radar with carrier frequency $\lambda = 0.24m$ moving at 200m/s with a range of 3500m, pointed forward off
broadside by 1.72 degrees (0.03 radians). The frequency shift is approximated by

\[ f_{dc} = -\frac{2R'(s)}{\lambda} \approx \frac{-2V}{\lambda} \sin \theta_{sq} \]  

(73)

where \( \theta_{sq} \) is the squint angle and \( V \) is the velocity of the platform. This squint angle results in a 50Hz doppler shift.

For this platform velocity and range to target the squint angle effectively distorts the point target by over 100 meters, if not processed with the appropriate matched filter. Even if the doppler center frequency is estimated within only a 10Hz error, the equivalent target shift is still 21 meters.

### 4.3 Estimation of Doppler Center Frequency from Simulated Data.

Recall from the discussion of clutterlock, that a scene filled with randomly placed scatterers produces a continuous doppler spectrum because scatterers at all ranges from within the radar footprint contribute a return signal. Using this information and the fact that the spectrum is weighted by the antenna gain pattern, we can estimate the doppler center frequency at the location of the spectral peak and thus, the pointing direction of the radar platform. Perhaps the most common method for obtaining the doppler spectrum is simply to apply the FFT to the average of several neighboring azimuth signals [15]. As we discussed in Chapter 3, a problem arises when the azimuth signal has noise added to it and we wish to estimate the spectral peak accurately. A noise spike off center may have greater amplitude than the actual center frequency leading to erroneous formation of the azimuth replica function. One approach to this
problem would be to apply a low pass filter to remove the noise. Selection of the appropriate filter is important because filters apply various distortions to the input signal used for estimating. We use the MVDR filter here not only because of its noise reduction capabilities, but also because of its' "distortionless response".

In Figure 33 we see the ideal azimuth chirp signal, a typical antenna gain pattern, their product and its spectrum. The center of the spectrum is easily estimated by finding the maximum value of the signal. We compare this ideal value to the values obtained from noisy spectra generated by the FFT, a smoothed version of the FFT and our optimal noise reducing filter MVDR.

In Figure 34 we see the sample chirp spectra with additive noise of an FFT, a smoothed-FFT and the MVDR filters. We generate 1000 Monte Carlo runs of azimuth spectra and estimate the center values of each by finding the maximum value. The standard error is computed by finding the square root of the variance, i.e.

\[ \sigma = \sqrt{E[x^2] - E^2[x]} \] (74)

where \( E \) is the expectation operator and \( x \) represents the measured doppler center values. The standard error is then used to obtain the equivalent squint angle error (see Eqn. 73)

\[ \theta_{err} = \sin^{-1} \frac{\sigma \lambda}{2V} \] (75)

For a particular range to point target \( R_0 \), the squint angle corresponds to an azimuth distance from the point target by

\[ d \approx R_0 \sin(\theta_{err}) \] (76)
This experiment is duplicated with the smoothed spectrum such as the example in the center plot in Figure 34 and a final time with the MVDR spectrum shown in the lower plot in Figure 34. A sample of the results are shown in Figure 35 with several plots overlaid to represent the variance. The statistics are recorded in Table 1.

The same experiments were run with a biased doppler frequency so that the results could be compared with those of the zero bias data set. Those results are recorded in Table 2.
Figure 29: Point reflector and return image. (a) Ideal point reflector. (b) Results of simulated radar returns.
Figure 30: **Range compression.** (a) *Output of the range compression matched filter.* (b) *Fully range compressed image.*
Figure 31: **Azimuth compression.** (a) The real portion of the azimuth doppler signal. (b) The result after compression. (c) The fully compressed image.
Figure 32: Point target impulse response.
Figure 33: Ideal azimuth chirp spectrum. The ideal azimuth chirp signal attenuated by a typical antenna gain.
Figure 34: Comparison of azimuth chirp spectra. (a) Example of the azimuth chirp spectrum with approximately 800Hz doppler bandwidth. The signal to noise ratio is approximately 12db due to additive noise. (b) The FFT-smoothed spectrum. (c) Example of the MVDR spectral estimate.
Figure 35: Results of azimuth center frequency estimation. The results of a broadside azimuth spectrum Monte Carlo simulation. The spectra of FFT-generated, FFT-smoothed and MVDR are overlaid to show variance.
Results of Monte Carlo Simulation with 0Hz Doppler Center Frequency

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Average center (Hz)</th>
<th>Std Error (Hz)</th>
<th>Squint angle (Rad)</th>
<th>Error Distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>399.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>FFT</td>
<td>394.72</td>
<td>20.69</td>
<td>6.21×10⁻³</td>
<td>21.72</td>
</tr>
<tr>
<td>Smooth FFT</td>
<td>399.90</td>
<td>16.49</td>
<td>4.95×10⁻³</td>
<td>17.31</td>
</tr>
<tr>
<td>MVDR</td>
<td>399.01</td>
<td>0.97</td>
<td>0.03×10⁻³</td>
<td>1.01</td>
</tr>
</tbody>
</table>

Table 1: Results of Monte Carlo simulations. The results of 1000 Monte Carlo runs shows that the MVDR spectral estimator is capable of determining the doppler center frequency more consistently than the FFT-smoothing method. This table represents the results from a non-biased (zero doppler shift) azimuth spectrum. The MVDR spectral estimator has better performance than the smoothed-FFT when comparing the standard error.

As can be seen from the table, the unprocessed FFT spectra is very noisy and a wide variance of center values are estimated as a result. The error is defined as the distance in the azimuth direction between where that the SAR algorithm focuses and the true location of the point target. In the zero frequency bias simulation the unprocessed spectra had a standard deviation of center frequency of ±20Hz which would cause the focusing algorithm to assume an additional squint angle of ±6.21×10⁻³ radians. For
Results of Monte Carlo Simulation with 100Hz Doppler Center Frequency

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Average center (Hz)</th>
<th>Std Error (Hz)</th>
<th>Squint angle (Rad)</th>
<th>Error Distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>449</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>FFT</td>
<td>448.33</td>
<td>23.84</td>
<td>7.15×10⁻³</td>
<td>25.03</td>
</tr>
<tr>
<td>Smooth FFT</td>
<td>449.51</td>
<td>14.97</td>
<td>4.49×10⁻³</td>
<td>15.72</td>
</tr>
<tr>
<td>MVDR</td>
<td>450.32</td>
<td>2.02</td>
<td>0.06×10⁻³</td>
<td>2.12</td>
</tr>
</tbody>
</table>

Table 2: Results of Monte Carlo simulations. This table shows the results from a biased (100Hz doppler shift) azimuth spectrum.

A point target 3500m away, the focusing error is 21 meters. The smoothed version of the same spectra has a reduced variance but still has a significant error of 17 meters. This variance can be reduced by averaging over more neighboring azimuth lines but since the center frequency changes as a function of range, a smearing effect occurs. The MVDR spectra have the least center frequency variance because of the high noise rejection in the spectral estimator. In this simulation, the error was reduced to only one meter.

The experiments were duplicated with a forward bias of 100Hz and the results are shown in Figure 36. The corresponding statistics are compiled in Table 2. The results are consistent with the results of the zero bias experiment in that the MVDR spectral estimator had lower center frequency variance than the smoothed FFT spectral
estimator.
Figure 36: Results of azimuth center frequency estimation with doppler shift. The results of a 100Hz biased azimuth spectrum Monte Carlo simulation. The spectra of FFT-generated, FFT-smoothed and MVDR are overlaid to show variance.
4.4 Time-Frequency Analysis of the Doppler Spectrum

In the previous demonstrations, we used the MVDR spectral estimator to determine if the radar produced a doppler shift across the synthetic aperture. While the antenna may indeed encounter a constant or near constant squint angle across the aperture, it is likely that small perturbations in the position of the platform will cause short duration frequency shifts. These frequency shifts are not visible in the spectrum taken across the entire synthetic aperture because they represent doppler bandwidth already found in the azimuth doppler chirp. To detect the effects of small duration pointing changes, time-frequency analysis of the doppler chirp may be employed. We use the spectrogram that we presented in Chapter 3 in a software tool called SpecView which was created to examine the time-varying spectrum. A more detailed description of the tool is found in Appendix A.

As an example, consider the results obtained by the simulation of a single point target. As we have previously stated, the signals from the point target return with phase changes that collectively represent a chirp frequency. That is, as the platform passes by, the phase changes become smaller and reach a minimum at broadside, and then increase until the target is no longer illuminated by the cone of the radar. By convention, the phase of the returned signals is negative before the platform reaches the target, and positive afterward. For this reason, there are “negative” doppler frequencies and “positive” doppler frequencies which comprise the doppler chirp. This is seen graphically in the output of SpecView as a linearly increasing frequency (see Figure 37.)
The range change induces additional phase changes in the returned signal as shown in Figure 38. The spectral changes are clearly seen in the output of SpecView. This demonstrates how a small change in platform position translates to a significant frequency shift when viewed with time-frequency analysis.
Figure 37: SpecView output of ideal azimuth chirp.
Figure 38: SpecView output of local deviation in azimuth chirp. The azimuth chirp has a small frequency variation which leads to local spectral changes.
4.5 Time-Frequency Analysis of Doppler Rate

As a final result, we briefly examine a special case of doppler rate estimation. In Chapter 2, we mentioned that the frequency rate could be estimated from the slope of the doppler frequency peaks over the synthetic aperture of the radar. We present a simulation of the case in which a single point target is unobstructed by either clutter or other point targets.

Time-frequency analysis is used to obtain a spectrogram of the azimuth chirp signal. Because the azimuth chirp is a spatial function as opposed to a temporal function, we apply time-frequency techniques to actually obtain a space-frequency representation of the azimuth signal. So we will replace the temporal index $t_n$ with the spatial index $s_n$ in this analysis to be consistent with our SAR problem. Since the point target is the only source of signal, a single sinusoid is present at each index in the spectrogram and can be located by determining the position of peak power. Once again, the MVDR spectral estimator can be used to obtain the spectrogram so that noise spikes do not interfere with the determination of spectral peaks. The peaks for each time position in the spectrogram are accumulated and a polynomial fit to the resulting data. Ideally, the data forms a line of the form

$$y = mx + b$$

(77)

where $m$ is the slope of the line, $b$ is the offset of the line and $x$ is the time index. An example of a spectrogram and its estimated frequency polynomial are shown in Figure 39.

The slope of the line is related to the frequency rate by the frequency resolution of
the spectrogram. Since the quadratic portion of the phase of the chirp signal is

$$\phi(s) = \frac{Ks^2}{2}$$  \hspace{1cm} (78)

the derivative is the frequency

$$f(s) = Ks$$ \hspace{1cm} (79)

where $K$ is the frequency rate. This is the parameter used to construct the matched filter for azimuth compression.
To determine the frequency rate, we first construct the polynomial in space.

\[ f(s) = \sum_{l=1}^{L} \alpha_l s^l = \alpha_0 s^0 + \alpha_1 s^1 + \cdots + \alpha_L s^L \]  

(80)

In our problem we are given a spectrogram \( S(f, s) \) and we would like to estimate the polynomial \( f(s) \) which should be linear, i.e.

\[ f(s) = K s + f_{dc} \]  

(81)

where \( K = \alpha_1 \) is the desired frequency rate required to design the azimuth compression filter. Our problem is to estimate the polynomial \( f(s) \) from the noisy spectrogram data.

Our approach is to estimate an enhanced spectrogram, \( \hat{S}(f, s_n) \) using the MVDR processor discussed previously, detect and select the peak of a particular time bin (assuming a single point target), i.e.,

\[ p(s) \equiv \text{max}[\hat{S}(f, s)] \]  

(82)

which essentially "transforms" the two dimensional spectrogram to a one dimensional time series measurement of the required polynomial. Thus, we can consider the peak measurement model as

\[ p(s_n) = f(s_n) + n(s_n) \]  

(83)

where \( p \) is the noisy measurement, \( f \) is the desired signal modeled by the polynomial of Eqn. 80 and \( n \) is the random noise.

The least squares problem for this model is to find the "best" values of the polynomial parameters \( \alpha_l \) to minimize the sum-squared error given a set of noisy measurements \( p(s_n) \) for \( n = 1, \cdots, N \). That is,

\[ \min_{\alpha} J = \sum_{n=1}^{N} \epsilon^2(s_n) \]  

(84)
for the error
\[ \epsilon(s_n) = p(s_n) - \hat{f}(s_n) = p(s_n) - \hat{s}^T \hat{a} \]  
(85)

with \( \hat{s} = [s_0 s_1 \cdots s_N]^T \) and \( \hat{a} = [\alpha_0 \alpha_1 \cdots \alpha_N]^T \). We can then write the vector-matrix measurement equation

\[
\begin{bmatrix}
  p(s_1) \\
  \vdots \\
  p(s_N)
\end{bmatrix} =
\begin{bmatrix}
  1 & s_1 & \cdots & s_1^N \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & s_N & \cdots & s_N^N
\end{bmatrix}
\begin{bmatrix}
  \alpha_1 \\
  \vdots \\
  \alpha_N
\end{bmatrix}
\begin{bmatrix}
  n(s_1) \\
  \vdots \\
  n(s_N)
\end{bmatrix}
\]  
(86)

or using vector-matrix notation

\[ \bar{p} = S \bar{a} + \bar{n} \]  
(87)

and the corresponding cost becomes simply

\[ J = \epsilon^T \epsilon = (\bar{p} - S \bar{a})^T (\bar{p} - S \bar{a}) \]  
(88)

The expression can be easily minimized by differentiating with respect to \( \alpha \) (see [13] for details) and setting the result to zero. That is,

\[-2S^T(\bar{p} - S \bar{a}) = 0 \]  
(89)

or solving for \( \bar{a} = \bar{a}_{LS} \) we obtain

\[ \bar{a}_{LS} = (S^T S)^{-1} S^T \bar{p} \]  
(90)

The polynomial estimate becomes

\[ \hat{f}(s_n) = \hat{s}^T \bar{a}_{LS} \]  
(91)

and since we choose a linear function for our problem, \( N = 2 \) and \( f(s) = \alpha_0 + \alpha_1 s \) and the least squares estimate is \( \hat{a}_{LS} = [\hat{\alpha}_0 \hat{\alpha}_1]^T \) with the desired frequency rate given by

\[ \hat{K}_{LS} = \hat{\alpha}_1 \]  
(92)
To summarize, our algorithm to estimate the frequency rate for a single point scatterer is:

1. estimate the spectrogram $\hat{S}(f, s_n)$ using the MVDR processor
2. detect the spectral peaks in each temporal bin $(s_n)$
3. choose the maximum peak to extract $p(s_n)$ for $n = 1, \ldots, N$
4. construct the least squares estimator for $\hat{\alpha}_{LS}$

We can use the frequency rate detector to detect perturbations in platform motion. By calculating the residual between the measured frequency rate and the estimate, a threshold can be applied to detect the deviation from linearity due to platform motion. In many applications, this method for obtaining the frequency rate is inadequate because it requires a single point target that can be isolated from the effects of adjacent point targets and clutter. More sophisticated signal processing techniques must be employed, however, to isolate the doppler frequencies from a single point target in a multi-scatterer environment. This is an open problem that will require further research.
5 CONCLUSIONS AND FUTURE WORK

We have described the fundamentals of a coherent radar system and the principles underlying synthetic aperture radar imaging. A spectral estimation technique was developed to analyze the return signals from the radar. A radar simulation, spectral estimator and time-frequency analysis tool were all created to examine these signals.

We found that in order to focus an image, parameters derived from platform velocity and position are required by the SAR reconstruction processor. One parameter could be estimated by examining the spectral content of the returned radar signals and we showed that by applying optimal spectral estimation techniques, we could decrease the amount of error induced from using the more common fast fourier transform (FFT) method of obtaining the spectrum. We also used our spectral estimation technique for time-frequency analysis of the doppler frequency to view short duration motion of the aircraft platform and detect platform pointing errors.

In the future, applications of time-frequency processing of radar signals could be directed towards more conventional techniques such as the Wigner-Ville or Discrete Cosine Transform (DCT) [10]. Also, there are other spectral estimators that could
yield higher performance noise rejection than MVDR such as MUSIC and MEM [14]. We could also apply time-frequency techniques to the estimation of the frequency rate component $f_r$ of the azimuth chirp in the case of multiple point targets and clutter in the field of view of the radar.
References


[17] M. Tabiani and S.H.R. Sadjadpour, Doppler Estimation by MA Model and its Comparison with FFT, Sharif University of Technology, Teheran, Iran


Appendix
A SpecView: A Spectrogram Analysis Tool

A tool for spectrogram analysis was created using the MATLAB signal processing software package [16]. This tool is capable of reading a time series signal and constructing the corresponding spectrogram using the methods developed in this thesis as well as some others. The tool as shown in Figure 40 is divided into display and control sections. The display portion of the window is subdivided into the spectrogram window, the time-selected frequency window (plotted sideways, corresponding to the frequency axis of the spectrogram image), and the time axis. The control section has buttons and edit windows which allow the user to load files, select spectrogram models and display options, move a cursor in the time window and zoom in on a specific location in the spectrogram. These features are each explained in detail below and a snapshot of the graphical user interface (GUI) is shown in Figure 40.

Estimators

This pull down menu allows the user to select the type of spectrogram to create. The choices include DFT for the standard spectrogram, MEM for the all pole spectral estimator and MVDR for the Minimum-Variance Distortionless Response model described in this thesis.

Plot Type

A choice of linear or log plots can be made with this menu.

Load Data

This button allows a user to select a time series file in MATLAB format. Depending on the spectrogram model selected, the user is prompted for processing parameters on
the command line.

**Time Index**

Allows the user to increment or decrement the red cursor in the time window which corresponds to the frequency plot. The cursor increments/decrements according to the window size used to create the spectrogram. The "skip" selector sets the number of time indices to move when a Time Index button is pressed.

**Multiple Slice Plots**

Figure 40: SpecView time-frequency analysis tool.
The multiple slice selector specifies how many spectra to plot in the frequency window starting with the current time index. This feature can be turned off by entering a zero in the window or by clicking the check box off.

**Color Map Selection**

A menu allows the user to select the color map from the standard set of maps offered by the MATLAB environment.
B Radar Simulator Source Code
% sim.m: This is the simulation of the radar and SAR processor
% author: Brooks Johnston
% date: May 5, 1996

% usage: sim(displayFlag, noiseFlag, deviationFlag, rangeCurvatureFlag, ...
%     noiseFactor, fdc, filestr)
% function sim(displayFlag, noiseFlag, deviationFlag, rangeCurvatureFlag, ...
%     noiseFactor, fdc, filestr)

% clear all;
displayFlag = 1;
noiseFlag = 0;
printFlag = 0;
saveFlag = 1;
deviationFlag = 0;
rangeCurvatureFlag = 0;
noiseFactor = 0.5;
fdc = 0;
filestr = 'tst';

% simulated radar parameters
c = 3.0e8; % speed of light in air, meters
a = 0; % chirp freq
K = 1.256e12; % chirp rate
Tp = 10e-6; % chirp duration
Np = 128; % number of samples in pulse
RO = 1500; % point object range in meters
minRange = 1000; % min swath, in meters
maxRange = 10000; % max swath, in meters
numRangePix = 512; % in pix... watch out for aliasing
RpixToMeters = maxRange/numRangePix; % conversion
RmetersToPix = numRangePix/maxRange; % conversion
ROP = RO*RmetersToPix; % p.o. range in pix
fc = 1.25e9; % carrier freq
lambda = c/fc; % carrier wavelength
L = 1; % antenna length, meters
theta = lambda/L; % cone angle, radians
minAz = 2*RO*tan(theta/2); % Az row length at point object, meters
maxAz = minAz*1.25;
s0 = maxAz/2; % locate the point object in the middle of az
forwardVel = 200; % meters/sec
prf = 500; % Az sampling rate, samps/sec
dt = 1/prf; % Az sampling interval, seconds
ds = forwardVel/prf; % Az sampling interval, meters
str = sprintf('sampling interval %3.2f meters\n', ds);
disp(str);
S = minAz/forwardVel;
str = sprintf('integration time= %3.2f seconds\n', S);
disp(str);
minAzP = floor(minAz/ds);
maxAzP = floor(maxAz/ds);
minRangeP = minRange/numRangePix;
maxRangeP = maxRange/numRangePix;
str = sprintf('Number az points: %d\n', maxAzP);
disp(str);

tic;
% Create radar chirp pulse
disp('Create radar chirps\n');
t = 0:Tp/Np:Tp;
chirp = sin(a*t + .5*K*t.^2);
m = (maxRangeP-minRangeP-length(chirp))/2;
r = [ zeros(m,1) ; chirp' ; zeros(m,1)];

% create chirp pulse matched filter replica
rReplica = [ flipud(r) ; zeros(1, numRangePix-size(r))]; % reverse time

% determine range for each pulse
disp('determine range for each pulse');
s = -maxAz/2:ds:(maxAz/2)-ds;
Rs = sqrt(s.^2+R0^2);
if deviationFlag == 1
    hann = .05*sin(pi*(0:.005:.995)).^2;
squintDeviation = hann*R0*.005; % 2 percent of R0
    m = zeros(1,length(Rs));
    first = floor(length(Rs)/4);
    last = first + 199;
    m(first:last) = squintDeviation;
    Rs = Rs + m;
    str = sprintf('Squint deviation turned on at %d\n', first);
disp(str);
end
end;

m = 2*(Rs(1)-Rs(2))/lambda;
str=sprintf('phase change/meter = %3.4f radians, %3.2f samples/meter', m/ds, i/ds);
disp(str);
str=sprintf('phase change/sample = %3.4f radians', m*2*pi);
disp(str);
if m>.5
    disp('WARNING: prf needs to be increased');
end;
str=sprintf('doppler frequency = %4.5f Hz', ddt);
disp(str);
maxConeRange = RO*acos(theta/2);
%i = find(Rs>maxConeRange);
%Rs(i) = zeros(length(i),1); % zero out the ranges that are out of view
RsP = Rs*RmetersToPix;

%% Antenna Gain Pattern
agc = ((.5 - .5*cos(2*pi*(0:(maxAzP-1))/(maxAzP-1)))/1)^2./Rs.^4;
%AGC = sparse(1:length(agc),1:length(agc), agc);
%r = AGC * r;  %% multiply by antenna gain pattern

%% offset pulses to make 2d return image
disp('Create 2d pulse return image');
r = zeros(numRangePix, maxAzP);
t = length(chirp);
if rangeCurvatureFlag == 1
    disp('computing range curvature image');
    for i=1:maxAzP
        if RsP(i) > 0
            r(RsP(i):(RsP(i)+t-1),i) = agc(i)*chirp';
        end;
    end;
else
    disp('computing image without range curvature');
    for i=1:maxAzP
        if RsP(i) > 0
            r(RsP(i):(RsP(i)+t-1),i) = agc(i)*chirp'; % inhibit range curvature
            r(RsP(i):(RsP(i)+t-1),i) = chirp'; % inhibit range curvature
        end;
    end;
end;
% add noise to system
if noiseFlag == 1
    v = randn(size(r))*noiseFactor;
    r = r + v;
    clear v;
end;

% Page 1
if displayFlag == 1
    figure(1);
    subplot(211);
    plot(0,RO,'o');
    title('Point target at range RO, in cluttered background');
    ylabel('range');
    xlabel('azimuth');
    subplot(212);
    imagesc([-maxAz/2 maxAz/2],[minRange maxRange],r);
    surf(r);
    a = axis;
    axis([0 length(r) a(3) a(4)]);
    colormap(jet);
    title('Returned image');
    ylabel('range');
    xlabel('azimuth');
end;
if saveFlag == 1
    save page1 RO s v r maxAz minAz minRange maxRange
end;

disp('start range ffts');
R = fft(r,numRangePix);
fftReplica = conj(fft(rReplica)');

disp('Starting range compression');
tt = exp(-j*4*pi*Rs/lambda);
%tt(length(tt)+1) = tt(length(tt)); % compensate length for diff
f = sparse(1:length(tt),1:length(tt), tt);
%f = sparse(diag(tt));
%G = sparse(diag(fftReplica)) * R;
G = sparse(1:length(fftReplica), 1:length(fftReplica), fftReplica) * R;
G = (f*G');
clear f
g = ifft(G');
disp('finished range compression');
imageSize = size(g);
clear G

%% page 2
if displayFlag == 1
    figure(2)
    subplot(211);
    plot(1e14*abs(g(:,100)),'w');
    a = axis;
    axis([0 length(g(:,100)) a(3) a(4)]);
    title('Pulse compressed range chirp');
    ylabel('amplitude');
    xlabel('time');

    subplot(212);
    imagesc([-maxAz/2 maxAz/2], [minRange maxRange], abs(g));
    colormap(jet);
    surf(abs(g));
    title('Range compressed radar return image');
    % axis([ s(1) s(length(s)) minRange maxRange]);
    ylabel('range');
    xlabel('azimuth position');
end;
if saveFlag == 1
    save page2 RO s g maxAz minAz minRange maxRange
end;
disp('examining processed data');
azRowPeak = find(g(:, length(g)/2) == max(g(:, length(g)/2)));
azRow = g(azRowPeak,:);
if saveFlag == 1
    sig = (azRow);
    save azRowfd sig;
    str = sprintf('mv azRowfd.mat azRowfd%d%s.mat', fdc, filestr);
    unix(str);
    clear sig;
end;
azRow = azRow/max(azRow);
%plot(angle(azRow(1:length(azRow)/2)));
%disp('enter to continue');
%pause;

%%% make azimuth replica
disp('Start Az compression');
% take fdc from arglist
str=sprintf('doppler centroid = %4.1f',fdc);
disp(str);
fr = -2*forwardVel^2/(lambda*R0);
first = maxAzP/2 - minAzP/2;
last = first+minAzP;
t = s/forwardVel;
h = exp(j*2*pi*(fdc*t+.5*fr*t.^2));

azFFTsize = 2^nextpow2(length(h));
azRef = zeros(azFFTsize,1);
azRef(first,last) = h(first:last);

azRef = h;
%plot(azRef);
keyboard;
subplot(311);
plot(real(azRef));
subplot(312);
plot(real(azRow));
subplot(313);
plot(abs(conv(azRef, azRow)));
keyboard;

% azimuth compression, not enough memory to do all range bins
clear y;
H = conj(fft(azRef));
y=zeros(512,length(H));
for i=azRowPeak-25:azRowPeak+25
%  disp(sprintf('az compress row %4.0f',i));
    GG = fft(g(i,:));
    y(i,:) = ifft(GG .* H);
end;

clear GG H;
disp('finished az compression');
runtime = toc;

%%% Page 3
if displayFlag == 1
    figure(3);
    subplot(211);
    plot(imag(azRow), 'w');
    a = axis;
    axis([a(1) a(1)+length(azRow) a(3) a(4)]);
    title('Azimuth doppler chirp');
    ylabel('amplitude');
    xlabel('az position');

    subplot(212);
    imagesc([-maxAz/2 maxAz/2],[minRange maxRange],abs(y));
    colormap(jet);
    % surf(abs(y));
    a = axis;
    title('Azimuth compressed data');
    ylabel('range');
    xlabel('azimuth');
end;

if saveFlag == 1
    disp('saving data');
    save page3 RO s fdc azRef azRow y runtime maxAz minAz minRange maxRange
end;

%clear G R h g t r
%outimg = sparse(zeros(1024, maxAzP));
%outimg(RO*unitsToPix, :) =
% abs(y(length(y)/2-maxAzP/2:length(y)/2+maxAzP/2-1));
% Page 4
if displayFlag == 1
    figure(4);
    subplot(213);
    % surf(full(outimg));
    imagesc(full(outimg));
    colormap(jet);
    title('Range and azimuth compressed output image');
    ylabel('range');
    xlabel('az position');
end;

if printFlag == 1
    str = sprintf('print -deps resultsFdc%d.eps', fdc);
    disp(str);
end;
eval(str);
end;
C SAR Simulation Source Code
% sim2.m: This code creates the model of azimuth chirp spectrum as described in this thesis. The chirp is frequency-biased by assigning the doppler frequency centroid (fdc) a value in Hz. Other parameters are arbitrarily chosen but are within reasonable limits of an airborne SAR.

% author: Brooks Johnston
% date: May 9, 1996
% 5/26/96 Added noise and lpf

clear all
clg
noiseFlag = 1;
noiseFactor = .1;
printFlag = 0;
displayFlag = 1;
tic;
c = 3e8; % speed of light, meters/sec
L = 1; % antenna length, meters
V = 400; % platform velocity
fc = 1.25e9; % carrier frequency
lambda = c/fc; % carrier wavelength
beta = .0; % squint angle in radians
R0 = 3500; % broadside range, in meters
fdc = 50;
fr = -2*V*V/(lambda*R0);
Rp = -lambda*fdc/2;
Rpp = -lambda*fr/2;
thetaC = lambda/L; % cone angle
maxX = 2*R0*sin(thetaC/2); % cone width, meters
maxR = sqrt((maxX/2)^2 + R0^2);
T = maxX/V; % integration time
prf = abs(T*fr); % pulse repitition frequency
ds = V/prf; % azimuth sampling interval, meters
dt = ds/V; % sample time interval
naz = floor((maxX/ds)*1.25);
disp(sprintf('BW= %4.1f',BW));

disp(sprintf('naz= %4.1f',naz));

disp(sprintf('fdc= %4.1f',fdc));

disp(sprintf('RO= %4.1f',RO));

s = -maxX/2:ds:maxX/2;
t = -T/2:dt:T/2;

R = sqrt(s.^2 + R0^2);  % do range by distance

dR = diff(R);

%R = s./sqrt(s.^2 + R0^2);

R = R0 + Rp*t + .5*Rpp*t.^2;  % range by SAR params

i = find(R > maxR);

R(i) = zeros(length(i),1)+1e-6;  % zero out ranges outside of cone

R = R0 + Rpt + .5*Rpp*t.^2;  % range by SAR params

i = find(R > maxR);

R(i) = zeros(length(i),1)+1e-6;  % zero out ranges outside of cone

for i=1:100
    if rem(i,10) == 0
        disp(sprintf('loop %d',i));
    end;
end;

% Create return signal with correct amplitude/phase

rt1 = 1e8*abs(R.^(-4)) .* exp(-j*4*pi*R/lambda);

peakTrue(i) = find(abs(rt1) == max(abs(rt1)));  % range by SAR params

peakTrue(i) = BW*peakTrue(i)/length(rt1);

if noiseFlag == 1
    noise = randn(1,length(rt1)) * std(rt1) * noiseFactor;
    rt = rt1 + noise;
    snr = 10*log10((std(rt1)^2)/(std(noise)^2));
    if i==1
        disp(sprintf('Adding noise: SNR=%4.1f db',snr));
    end;
end;

wf = hanning(length(rt));

wf = .5 - .5*cos(2*pi*(0:length(rt)-1)/length(rt));

r = rt .* wf;

x = fftshift(abs(fft(r)));
if (displayFlag == 1) & (i==10)
    subplot(311);
    hold on;
    plotf(x,BW);
    title('Azimuth spectral response: FFT');
    ylabel('amplitude');
    xlabel('frequency');
    title('Azimuth spectrum weighted by antenna gain pattern');
end; % displayFlag

peakNoise(i) = find(abs(x) == max(abs(x)));
peakNoise(i) = BW*peakNoise(i)/length(x);
%disp(sprintf('noise: missed by %4.1f Hz',...
% BW*abs(peakNoise-peakTrue)/length(x));

%% Smooth spectrum by lpf
[b,a] = butter(5,.25);
b = [.2 .2 .2 .2 .2];
a = 1;
%y = medfilt1(r,5);
if i == 1
    last = x;
end;
t1 = (x+last)/2;  % average w/previous az line
last = x;
y = filtfilt(b,a,t1);
if (displayFlag == 1) & (i==10)
    subplot(312);
    plotf(y,BW);
    hold on;
    title('Smoothed azimuth spectral response: FFT');
    xlabel('azimuth position index');
    ylabel('amplitude');
end; % displayFlag

t = find(abs(y) == max(abs(y))); % average is center pt
peakSmooth(i) = sum(t)/length(t);
peakSmooth(i) = BW*peakSmooth(i)/length(y);
%disp(sprintf('smooth: missed by %4.1f Hz',...
% BW*abs(peakSmooth-peakTrue)/length(y));

M = fftshift(mvdr(r,5,1024));
if (displayFlag == 1) & (i==10)
subplot(313);
hold on;
plotf(M,BW);
a = axis;

axis([a(1) length(M) a(3) a(4)]);
title('Azimuth spectral response: MVDR');
ylabel('amplitude');
xlabel('frequency');
title('Azimuth MVDR spectrum weighted by antenna gain pattern');
end; % displayFlag
drawnow;

% t = find(M == max(M));
peakMVDR(i) = sum(t)/length(t); % average is center pt
peakMVDR(i) = BW*peakMVDR(i)/length(M);

if printFlag == 1
    print -deps azChirps.eps
end;

peakNoiseSTD = std(peakNoise);
peakSmoothSTD = std(peakSmooth);
peakMVDRSTD = std(peakMVDR);
runtime = toc;
save peakResults peakNoise peakSmooth peakMVDR snr runtime

thetaNoise= asin(lambda*(abs(mean(peakNoise-peakTrue)))/(-2*V)));
thetaNoiseSTD= asin(lambda*(abs(std(peakNoise))/(-2*V)));
disp(sprintf('peakNoise: avg %4.2f std Hz %3.2f theta %2.3d d %2.4f',
    mean(peakNoise),peakNoiseSTD, abs(thetaNoiseSTD), ...
    abs(R0*sin(thetaNoiseSTD))));

thetaSmooth= asin(lambda*(abs(mean(peakSmooth-peakTrue)))/(-2*V)));
thetaSmoothSTD = asin(lambda*(abs(std(peakSmooth))/(-2*V)));
disp(sprintf('peakSmooth: avg %4.2f std Hz %3.2f theta %2.3d d %2.4f',
    mean(peakSmooth),peakSmoothSTD, abs(thetaSmoothSTD), ...
    abs(R0*sin(thetaSmoothSTD))));

thetaMVDR= asin(lambda*(abs(mean(peakMVDR-peakTrue)))/(-2*V)));
thetaMVDRSTD = asin(lambda*(abs(std(peakMVDR))/(-2*V)));
disp(sprintf('peakMVDkMvDR: avg %4.2f std Hz %3.2f theta %2.3d d %2.4f',
mean(peakMVDR),peakMvdrSTD, abs(thetaMVDRstd), ... 
abs(R0*sin(thetaMVDRstd))));

disp(sprintf('Simulation took %4.1f minutes', runtime/60));