Modeling and Simulation Analysis of an FMCW Radar for Measuring Snow Thickness

By

Sudarsan Krishnan

B.E., Electronics and Communication Engineering,

University of Madras, 2000

Submitted to the Department of Electrical Engineering and Computer Science and the Faculty of the Graduate School of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

Chair: Glenn Prescott

Sivaprasad Gogineni

David Braaten

Date Thesis Accepted

To the lotus feet of Lord Sriman Narayana

Acknowledgements

"Om Namo Narayanaya"

I would like to thank my advisor Dr. Glenn Prescott, Dr. Prasad Gogineni and Dr. David Braaten for giving me an opportunity to work on this thesis and for guiding me through my work. I would specially like to thank Dr. Pannirselvam Kanagaratnam for helping me understand a lot about radar.

This work would not have been possible without the strong support I received from Amma, Appa, Amrutha and Priya. They have been instrumental in helping me to get back to good spirits whenever I am down and have encouraged me on this work like no one else.

I am indebted to my professors, especially Dr. Zacharia, Dr. Sethuraman and Dr. Shanmugan. Their teaching has given me the fundamentals to approach this thesis. I express my admiration to my teachers who from my very young age have helped me love science and mathematics.

I would like to thank my colleagues from Midas Comm. Tech. Pvt. Ltd. who encouraged me to work on my graduate studies. Specifically, I would like to thank Ms. Sampoornam for having been the best supervisor an employee can ask for.

I also express my gratitude to all my friends who have made my stay at KU a memorable one. It has been great fun being here.

Table of Contents

List of Figures

List of Tables

Abstract

Global warming is one of the most important issues facing humanity. The depth and extent of sea-ice are clear indicators of global warming. Snow cover on sea-ice is an important variable that affects the volume of sea-ice and indirectly global warming as a whole. The Radar and Remote Sensing laboratory at the University of Kansas has developed an Ultra Wideband Frequency Modulated Continuous Wave Radar to determine snow thickness over sea-ice.

In any radar system, including the snow radar, the return signal from the medium is corrupted by the effects of the radar system itself, and also by the effects of surface and volume scattering. Therefore, the return signal cannot be analyzed directly. The research described here focuses on a numerical analysis and simulation of the snow-radar that includes the effects of the system and those of scattering.

The system effects are accounted for by including the point-spread function into the simulation. The Point Spread Function is determined by empirical measurement and calibration using the prototype radar. The effects of surface and volume scattering are modeled from theory and are also incorporated into the simulation.

This research completed here will help us in analyzing the actual radar returns and in deconvolving the effects of the system and the effects of surface and volume scattering. The application of deconvolution on the radar return signal will help in extracting range information easily as the layers will be clear.

Chapter - 1

Introduction and Background

One of the major issues facing the world is global warming. Scientists have suggested that the Polar Regions will exhibit the first and clearest indication of global warming compared to other regions on the planet [1]. This is because the Polar Regions represent an extreme in climate. In addition, the Polar Regions are relatively untouched by human influence.

Scientists have determined that the extent and thickness of sea-ice is an important indicator of global climate change [2], as it directly influences the energy transfer between the ocean and the atmosphere. An understanding of this integrated system, consisting of atmosphere, hydrosphere and cryosphere, can help us analyze global climate change.

"Snow cover on sea-ice is an important physical variable that impacts energy, physical and chemical processes"[3] that occur on the integrated system consisting of the Ocean, Sea-Ice and the Atmosphere. Shrinking of snow cover in the Polar Regions leads to a faster loss of sea-ice to melting. Sea-Ice melting poses a major threat to coastal ecosystems around the world by contributing to sea level increase.

1.1 Snow Cover on Sea-Ice

1.1.1 Significance

The presence of snow cover on sea-ice alters the interaction of the Ocean – Sea-Ice - Atmosphere (OSA) interface. The factors that influence these interactions are the physical and electromagnetic characteristics of snow and sea-ice. Snow has a lower thermal conductivity, almost a tenth of that of the sea-ice below it. Thus, it acts as an insulating thermal blanket affecting the accretion rate and ablation rate of the sea-ice. This is known to affect heat exchanged between ocean and the atmosphere. The presence of snow also creates a high albedo(ratio of reflected to incident electromagnetic power) surface on the sea-ice. This reflects a major portion of the incident energy, thus slowing down ablation of sea-ice. The effect of the high albedo surface is only felt during summer when the sun shines for months at a time.

Snow also affects the ecology of sub-ice organisms by the extinction of Photosynthetically Active Radiation (PAR) necessary for the organisms. It also affects the habitat selection of animals like seals and polar bears [4]. Therefore, we are interested in determining the magnitude, distribution and change in distribution of snow over sea-ice.

1.1.2 Properties

Physically, snow is a mixture of air, ice and liquid water. Electromagnetically, snow is characterized by a dielectric constant that depends on the temperature, water content, density and shape of ice-particles and water inclusions [5]. The dielectric constant also varies with signal frequency. The equations that characterize snow as a factor of these parameters are explained in Chapter 4.

Dry snow has very little liquid water and mostly consists of ice and air. It is almost transparent to electromagnetic waves in the microwave region. Its dielectric constant lies between that of air ($\varepsilon_r = 1$) and of ice ($\varepsilon_r = 3.15$). Wet snow, on the other hand offers significant attenuation to electromagnetic waves and is also characterized by a much higher dielectric constant than dry snow.

An appropriately designed radar sensor can be used to study snow cover over sea-ice; since the dielectric constant of air, snow and sea-ice are different. This dielectric contrast can be seen in the reflected signal. This helps in determining snow depth over sea-ice. As an example, consider a snow layer of height 0.5 meters on top of a sea-ice half space as shown below.

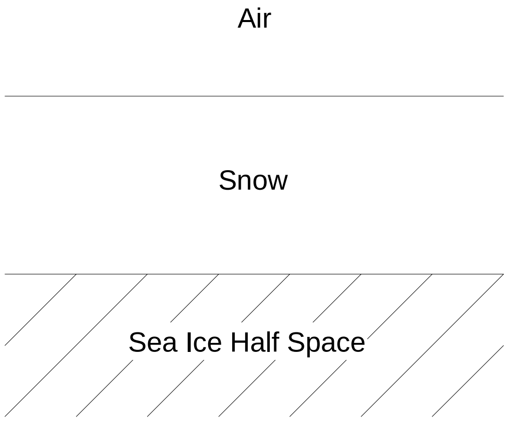


Figure 1.1 - Simple Geophysical Model of Snow Covered Sea-Ice

Here, each layer is characterized by its dielectric constant; say 1 for air, 1.5 for snow and 3.15 for sea-ice. These dielectric contrasts result in two interfaces, the air-snow interface and the snow-sea-ice interface. An incident signal is reflected at these two interfaces and this reflected signal can be processed by the radar receiver to identify the distance of each layer from the radar.

This example assumes single ray reflection from perfectly flat interfaces. In actuality, such an interface doesn't exist and the antennas have a finite beam width. Therefore, the return will be corrupted by returns from side angle surface scattering and volume scattering. In order to clearly identify the interfaces, these scattering phenomena need to be taken into account. These phenomena are discussed in detail in Chapter 4.

1.2 Measurement of Snow Thickness over Sea-Ice

In-situ measurements have been the norm for collection of data on the magnitude and distribution of snow thickness over sea-ice. This information is very limited and inadequate considering the vast expanses of sea-ice. Remote sensing is the only effective solution to collect data over such large areas.

Snow thickness measurements over large areas like the Polar Regions can be made effectively only using space-borne measurements. The AMSR-E (Advanced Microwave Scanning Radiometer for EOS) can measure snow depths with accuracy of less than 5 cm and a spatial resolution of 12.5 km. The height of snow can be retrieved from AMSR data using algorithms developed by Markus *et al.* [6]. However, before these algorithms can be used they need to be validated over a reasonably large area. Again, in-situ measurements generate only sparse data.

Therefore, they cannot be used to validate the algorithm. What is needed is an ultra wideband radar operated from an aircraft or helicopter. The operation from aircraft or helicopter will enable the radar to collect data over a reasonably large area. The wide bandwidth will help in achieving high range resolution, while the appropriate choice of operating frequency will allow the signal to penetrate snow without significant attenuation. Lower microwave frequencies (between 2 GHz and 8 GHz) are chosen because the depth of penetration in snow is about 1m at these frequencies [7].

1.3 KU Snow Radar Prototype

The Radar and Remote Sensing Laboratory has developed an Ultra Wideband Frequency Modulated Continuous Wave Radar (Snow Radar) to address this requirement [8]. The radar operates over a frequency range of 2 to 8 GHz. This bandwidth gives it a range resolution of about 4 cm. In addition, operating in the lower microwave band allows the radar signal to propagate through snow without significant attenuation. A Radar system functional block diagram is given in Figure 1.2. This illustrates the functional components that make up the Snow Radar.

The transmitter generates the 2-8 GHz chirp using a YIG oscillator, whose tuning characteristic is linearized by using a phase locked loop and a low frequency digital chirp generator. Using a combination of external and internal frequency dividers, an error signal is generated, which is used to generate an extremely linear chirp. The chirp is fed to a directional coupler, whose output is amplified and transmitted through the antenna. The coupled port output is used as the local oscillator reference in the receiver. The receiver amplifies the low-level signal from the antenna and feeds it into the mixer. Care is taken not to radiate the local oscillator signal by using a high reverse-isolation amplifier. The attenuators in the front end of the receiver buffer the signal and reduce mismatches.

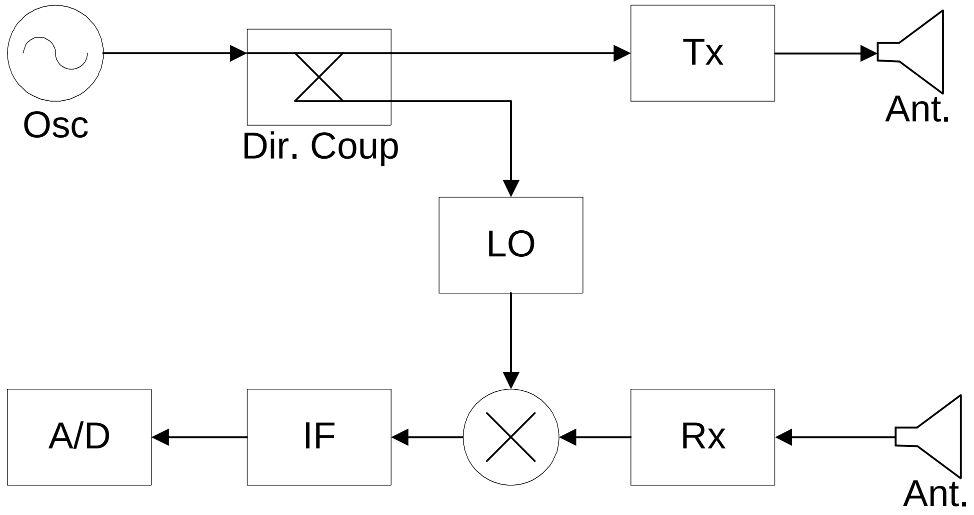


Figure 1.2 - Radar System Functional Block Diagram

The intermediate frequency (IF) mixer output signal is passed through a band pass filter of bandwidth 4-400 KHz, amplified and digitized at five MHz using a 12-bit A/D converter. The parameters of the snow radar are shown in Table 1.1.

Table 1.1 - KU Snow Radar Parameters

Characteristic	**Value**
Radar Type	FM-CW
Sweep Frequency	2-8 GHz
Range Resolution	≅4 cm
Sweep Time	10 ms
Transmit Power	13 dBm
PRF	25 Hz
A/D Dynamic Range	12-bit, 72 dB
Sampling Rate	5 MHz
Antenna	TEM Horn Antenna

1.4 Objective and Approach

The objective of this research is to analyze by simulation, the measured system characteristics of the snow radar and the propagation effects of surface scattering and the effects of volume scattering. The approach used here to simulate the snow radar is as follows:

1. Model the system effects by taking measurements on and with the snow radar.

2. Using simulation techniques, numerically analyze the radar system while incorporating a propagation model that includes surface scattering and volume scattering.
3. Compare the simulation with actual results from laboratory setup and field experiments.

These steps form the basis of the research. A brief technical background on FMCW radar is provided in Chapter 2. The system modeling required for the simulation is covered in Chapter 3. The surface and volume scattering effects are analyzed in Chapter 4. The simulation methodology and its numerical basis are covered in Chapter 5. Results are analyzed in Chapter 6. Finally, Chapter 7 concludes this thesis with some recommendations for future work including deconvolution.

Chapter - 2

FMCW Radar

This chapter describes the basic principle of operation of FMCW Radar, including the approach used to simulate the radar for this project.

2.1 FMCW Radar Basics

Frequency Modulated Continuous Wave Radar transmits a frequency sweep, often called a chirp. The signal is reflected from distant targets and detected by the receiver where the return signal is mixed with a copy of the transmitted signal to determine the range of the target.

The transmitted waveform has a time varying frequency f(t) given by

$$f(t) = f_0 + \alpha \cdot t \qquad \forall \quad t < T \tag{2.1}$$

where f_0 is the initial frequency, α the Rate of Change of Frequency and T is the Sweep time.

This is a linearly increasing frequency sweep as shown in Figure 2.1. The frequency is f_0 at the start of the sweep and increases to f_1 at the end of the sweep after a time T called the sweep period. The bandwidth B is the difference between f_1 and f_0.

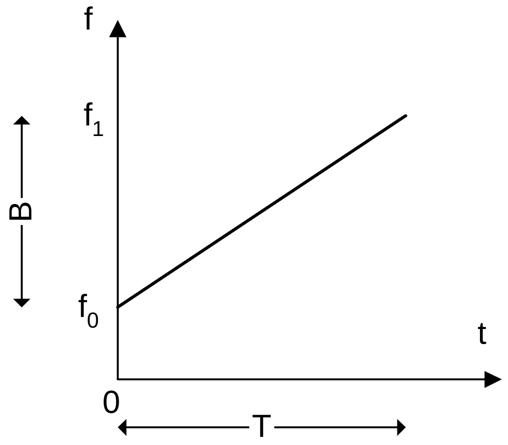


Figure 2.1 - FMCW Radar Sweep

The rate of change of frequency is then given by

$$\alpha = \frac{B}{T} \tag{2.2}$$

The phase of the waveform is given by

$$\phi(t) = 2\pi \int f(t)dt = 2\pi \cdot (f_0 t + \frac{1}{2}\alpha \cdot t^2) \tag{2.3}$$

The transmitted waveform travels to the target at distance R and returns after a time delay τ given by

$$\tau = \frac{2R}{c} \tag{2.4}$$

where c is the velocity of light in the medium. The process of generating the beat frequency from the return signal can be visualized as shown in Figure 2.2.

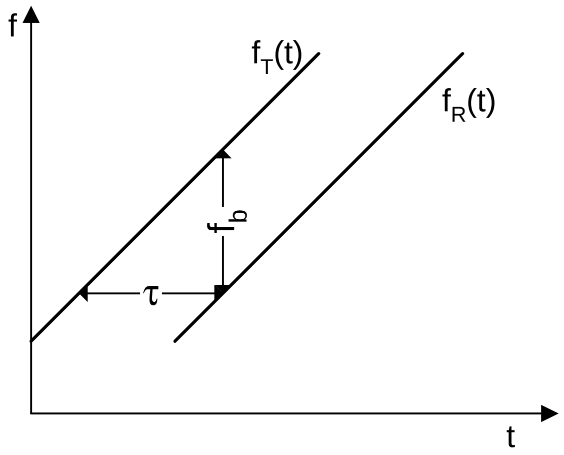


Figure 2.2 - Beat Frequency

Given that the transmit signal frequency f_T is

$$f_T(t) = f_0 + \alpha \cdot t \qquad \forall \quad 0 < t < T \tag{2.5}$$

and assuming an ideal point target, the received signal frequency f_R is given by

$$f_R(t) = f_0 + \alpha \cdot (t - \tau) \qquad \forall \quad \tau < t < T \tag{2.6}$$

Mixing of these two signals produces sum and difference frequencies, $f_T + f_R$ and $f_T - f_R$. The resultant signal is then low pass filtered to remove the $f_T + f_R$ term. The term that remains is the beat frequency term f_B. This process is modeled in Figure 2.3. The beat frequency term is directly related to the range by Equation 2.7.

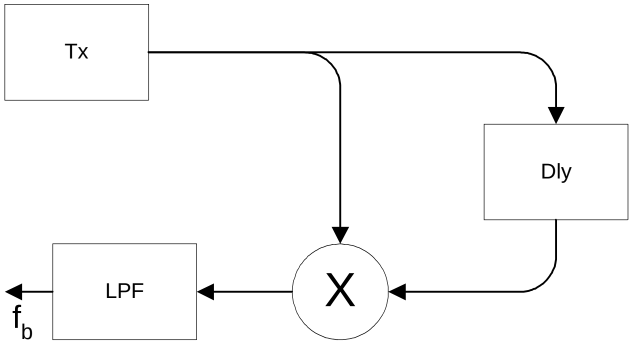


Figure 2.3 - Conceptual FMCW Radar Model

$$\begin{aligned} f_B &= f_T - f_R \\ &= \alpha \cdot \tau \\ &= \frac{B}{T} \cdot \frac{2R}{c} \\ &= \frac{2RB}{cT} \end{aligned} \tag{2.7}$$

Thus knowing the beat frequency and the radar parameters of B and T, we can retrieve range information from the return signal. FMCW radars often store the beat frequency signal to allow for off-line processing using Fourier transform techniques. The Fast Fourier Transform (FFT) is the mathematical tool used to interpret the spectrum of the beat frequency signal in terms of radar range.

The point target scenario can be extended to multiple targets also. For each target, there will be a return with a frequency corresponding to the distance. Each return will also have an amplitude corresponding to the round trip attenuation to its specific target. These can be seen as peaks in the frequency domain. This can be further extended to spread targets. In this case, the return will not consist of only individual peaks but will have spread peaks corresponding to returns from other scatterers.

2.2 Simulation

A simple simulation of an FMCW system can be accomplished as follows:

1. Generate an FMCW Transmit waveform.
2. Introduce a delay (τ) to the waveform. This is then the Receive Waveform.
3. Multiply it with the Transmit waveform. This is the Mixer Output.
4. Low Pass filter the waveform. This gives us the beat frequency signal.
5. Apply an FFT and display.

The output of this simple simulation is shown in Figure 2.4.

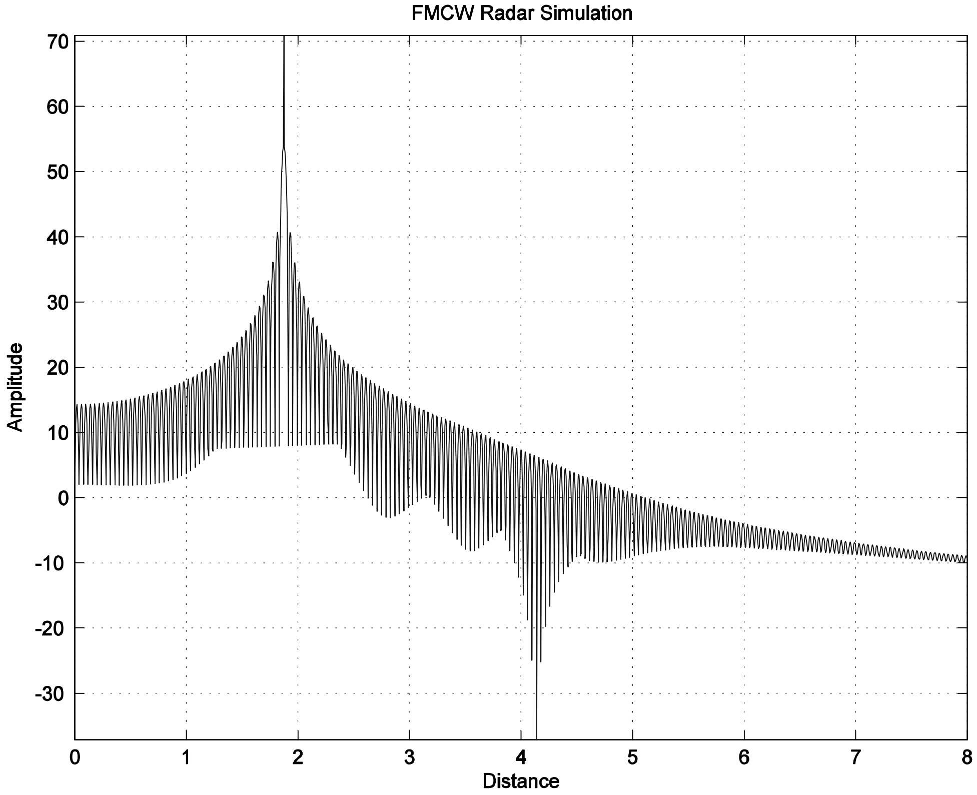


Figure 2.4 - Simple FMCW Simulation Result

Here we can see a peak at a distance D (about 1.9m) corresponding to the delay (τ) taken for the reflected signal to return to the radar. This would also be the return signal expected from an ideal radar. An amplitude scope (A-scope) from actual field trials in Antarctic sea-ice is shown in Figure 2.5 for comparison with this ideal simulation. It can be seen that the A-scope has multiple peaks. Some of these peaks correspond to actual targets. The other peaks correspond to clutter targets, noise and multiple returns.

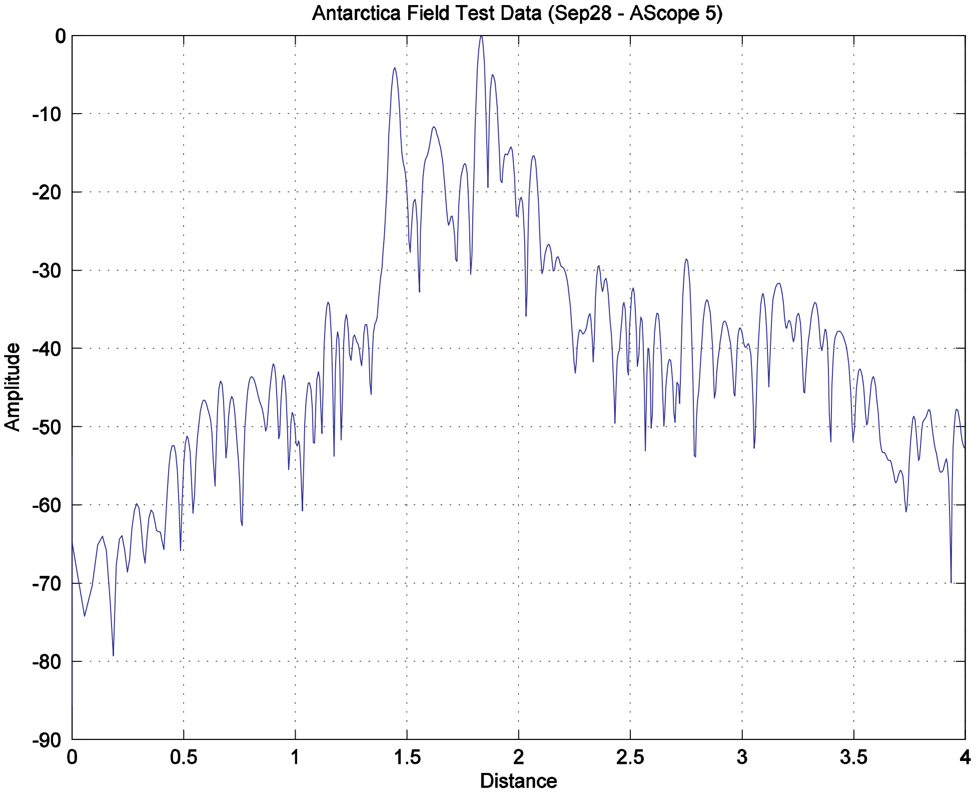


Figure 2.5 – A-Scope (amplitude versus range) of an actual field data set

Chapter - 3

Radar System Modeling

The goal of system modeling is to determine the point spread function of the radar. The point-spread function is a characterization of the effects of the system on the signal. If the system were ideal, it would have a flat frequency response. This would correspond to an impulse in the time domain. However, if the system were not ideal, the frequency response would not be flat, and the impulse would be spread out in the time domain. Therefore, this point-spread function is an important indicator of the influence of the system on the received signal. The inclusion of the point spread function of the radar in an ideal simulation allows the designer to identify and characterize the effects of the transmitter and receiver system on the signal reflected from the target.

Radar system modeling can be done in two ways. The first method is modeling by measurement and simulation. In this method, the components are characterized in a section-by-section approach and the complete system is simulated to observe the effects of the combination of all the sections. The second method is modeling by using a calibration target. This involves measuring the radar return from a calibration target and determining the effects of the system by observing the changes in the radar return.

3.1 Modeling by Measurement

The snow radar designed and constructed at the University of Kansas consists of an oscillator section, a transmitter section, an antenna section, a local oscillator section, the receiver front end, a mixer and the intermediate frequency section as shown in Figure 3.1.

The oscillator section generates the 2-8 GHz chirp. The generated chirp should ideally have constant amplitude and a linear time-frequency relationship as shown in Figure 2.2. In reality, the chirp will be corrupted by amplitude and phase errors. A non-constant chirp envelope results when the output of the oscillator does not have equal amplitude at all frequencies. In addition, the effects of the transmitter components can modulate the chirp envelope. These are called Amplitude errors. In addition, the relationship between time and frequency at the oscillator output is not precisely linear. The transmitter components also introduce frequency selective delays that cause phase fluctuations. These are called phase errors. For a simulation to be meaningful, these errors must be included. The procedure to determine the amplitude and chirp errors and to incorporate them into the simulation is described in the next section.

The transmitter, receiver and local oscillator sections are composed of two port or three port components that can be characterized by their transfer functions. The transfer functions are determined from the s-parameters of the sections. This will be further explained later in this chapter. The intermediate frequency section and the antenna section are considered ideal because of limitations explained in the discussion later in the chapter.

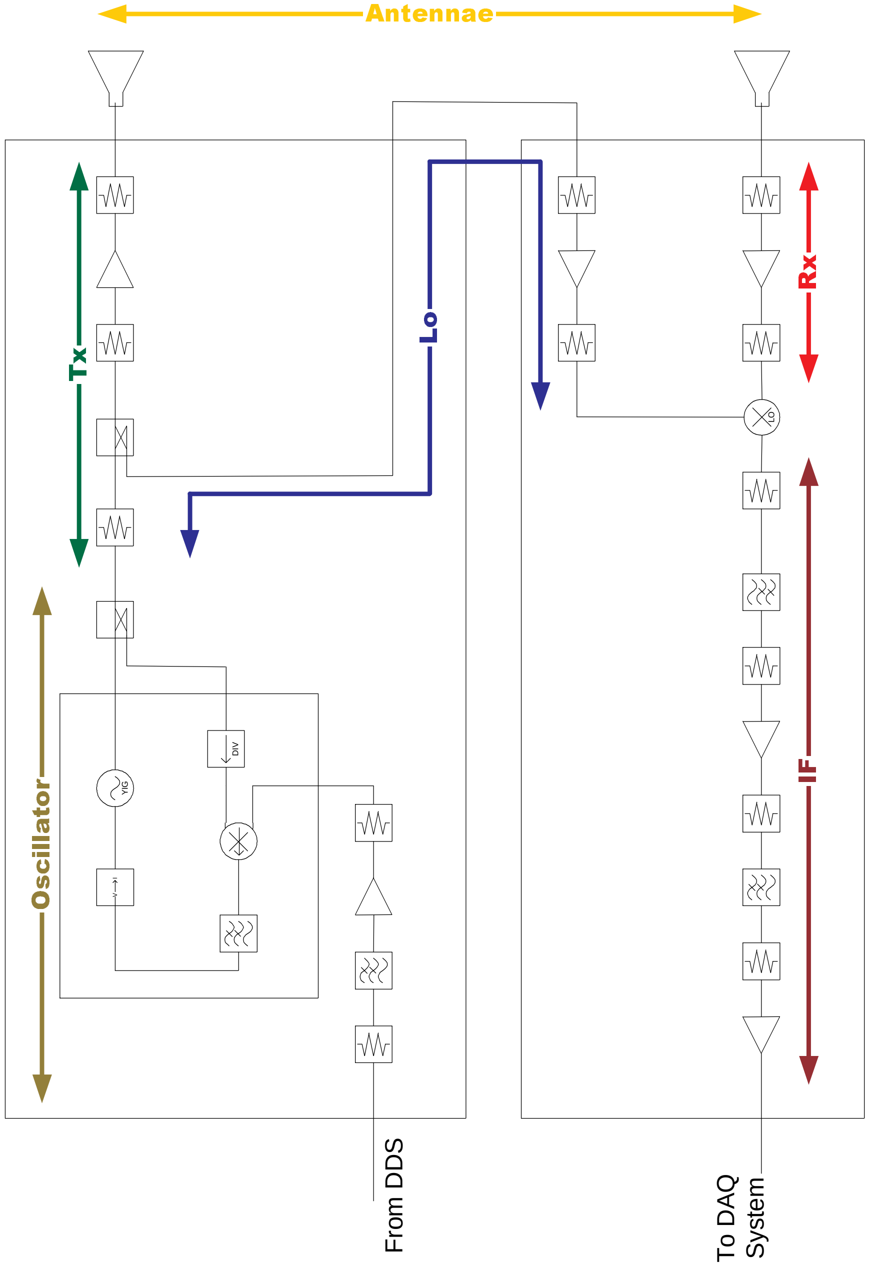


Figure 3.1 – KU Snow Radar

3.1.1 Modeling the Oscillator Section

The oscillator generates a 2-8 GHz chirp in response to an error voltage sweep of 0-5 volts in the PLL. This can be visualized as shown below.

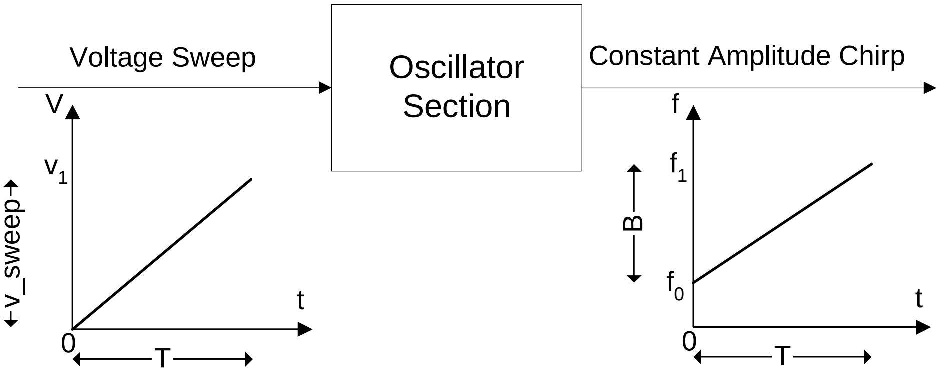


Figure 3.2 - Visualization of Oscillator Section

The goal is to model amplitude and phase characteristic of the output chirp. This is done by measuring the output power and frequency at different values of the input voltage sweep.

3.1.1.1 Amplitude Error Modeling

The measurement of power versus sweep voltage is used to model the amplitude errors. Ideally, the relationship between power and sweep voltage should be constant. This would indicate that the chirp output from the oscillator has constant amplitude at all frequencies. In actuality, this is not the case as the oscillator is not ideal [9], [10].

The measurement setup to measure the output power at different sweep voltages will also introduce amplitude fluctuations. So, the voltage of the generated chirp is corrected for the measurement setup. This involves correcting for the power loss introduced by the measurement setup. Then a discrete cosine transform (DCT) is performed on the voltage, after removing the constant term, to look at frequency components of the envelope errors. The DCT is chosen over the FFT as the tool to determine significant components because its orthogonolization properties are better. So a lesser number of DCT coefficients can model the amplitude errors compared to the FFT. The significant peaks of the DCT are selected as the model of the amplitude.

This operation is shown in Figure 3.3. Here the measured amplitude error signal is shown in Figure 3.3a, its DCT shown in Figure 3.3b. The points on the original DCT with amplitude greater than 0.1 are assumed to be significant peaks. This value was chosen by trial and error. A threshold of 0.1 gives 19 coefficients. The rest of the values are set to zero. This is the modeled DCT and is shown in Figure 3.3c. From the modeled DCT we obtain the modeled amplitude errors by performing the Inverse DCT is shown in Figure 3.3d.

The actual and modeled amplitude of the sweep are plotted in Figure 3.4 for comparison. The modeled envelope errors determined by applying the inverse DCT on the modeled DCT can be superimposed on the constant amplitude of an ideal simulated chirp to include amplitude errors in the simulation. It can be seen that there is a close match between the actual and modeled amplitude errors.

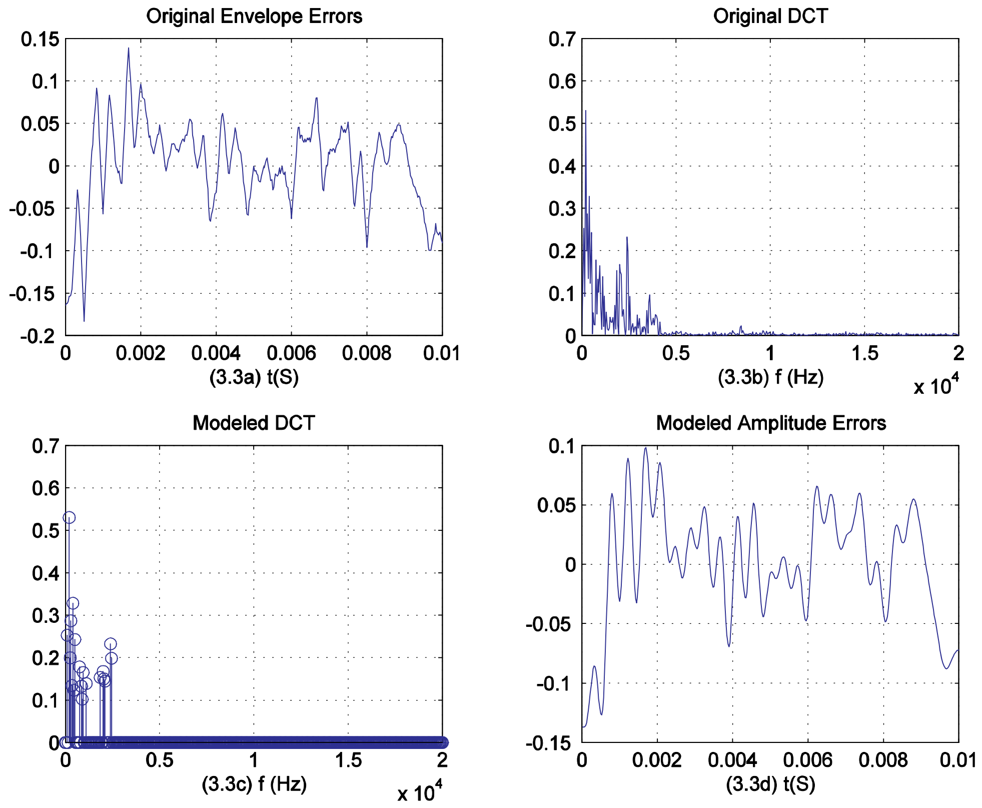


Figure 3.3 - Amplitude Error Modeling

The modeled DCT (amerr) values are passed as parameters to the simulation. The simulation then expands these into the amplitude errors and superimposes the amplitude fluctuations on an ideal chirp with constant amplitude.

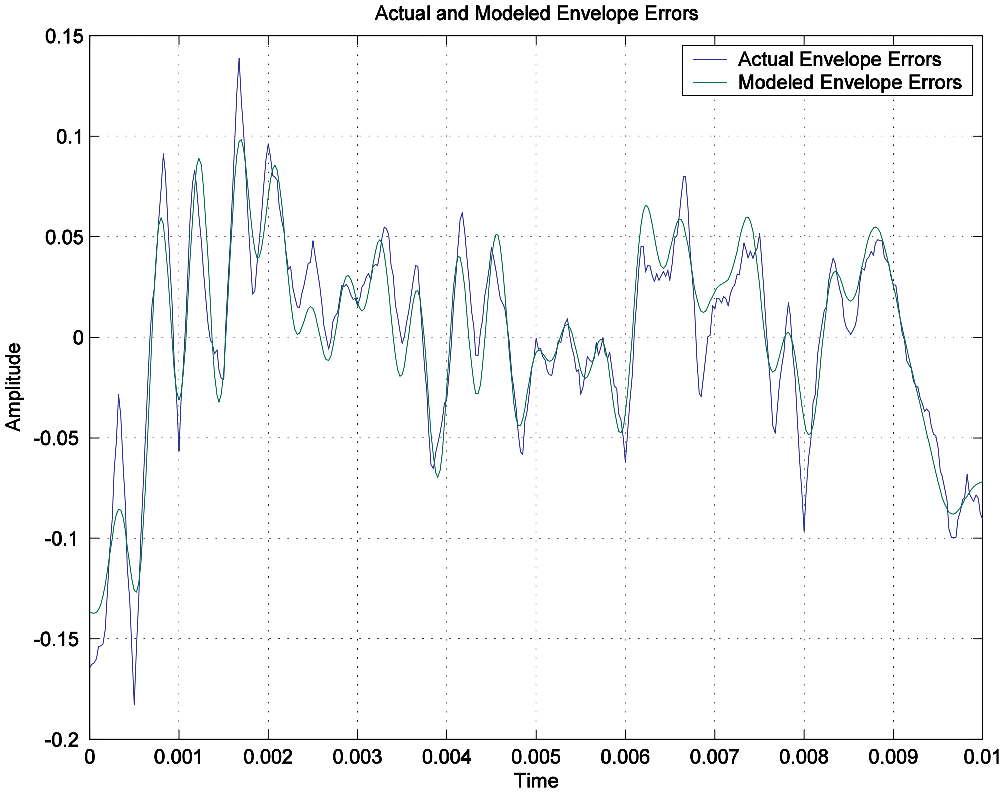


Figure 3.4 - Actual vs. Modeled Errors

3.1.1.2 Phase Error Modeling

Ideally, the phase of the chirp is given by

$$\phi(t) = 2\pi \int f(t)dt = 2\pi * (f_0 t + \frac{1}{2}\alpha * t^2) \tag{3.1}$$

However, the oscillator output chirp will not have this exact phase characteristic. The phase of the generated chirp is determined as follows. From the measurement of frequency vs. input voltage of the oscillator section and the voltage-time relationship, we can fit the measured frequency as a polynomial of time.

$$f(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0 \tag{3.2}$$

An third order polynomial was found to be sufficient to model the linear time frequency relationship. The third order polynomial was used because the higher order coefficients were found to be zero. A plot of the measured frequency versus time characteristic is compared with the polynomial fitted characteristic and the ideal mathematical characteristic is shown in Figure 3.5. In Figure 3.5a, the red line is the ideal linear frequency vs. time plot, the blue line is the measured frequency vs. time plot and the green line the modeled frequency vs. time plot. The differences between these three frequency-time plots are shown in Figure 3.5b for clarity.

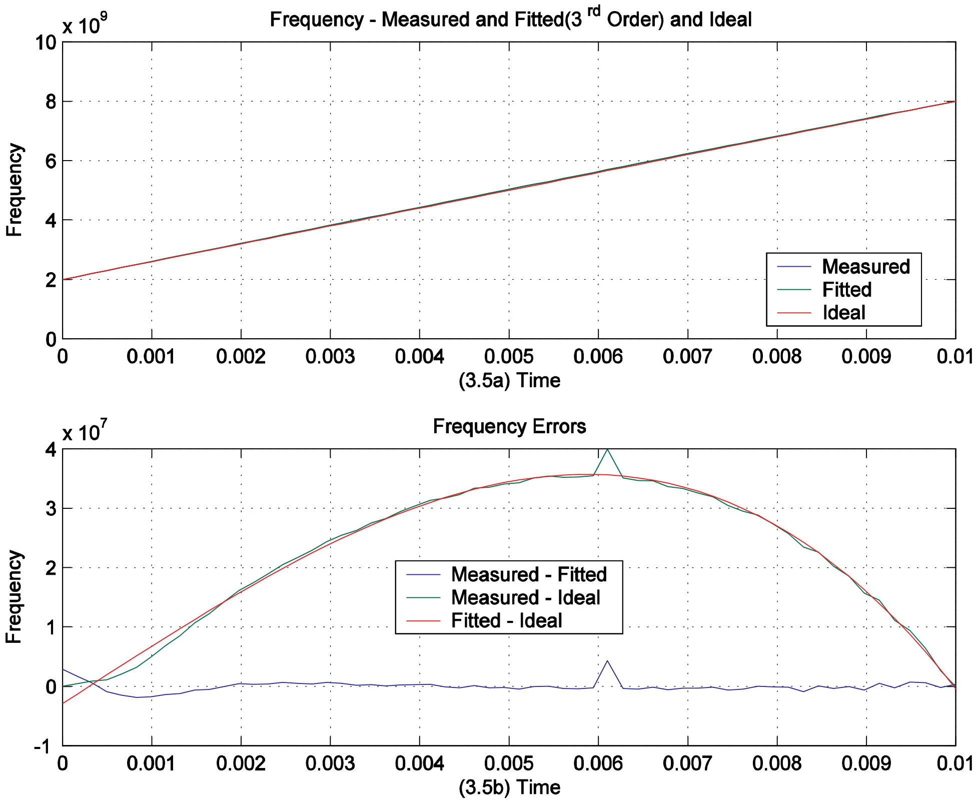


Figure 3.5 - Frequency Errors

Using the polynomial for frequency and assuming zero initial phase we can determine the phase as a polynomial of time.

$$\phi(t) = 2\pi \int f(t)dt = b_3 t^4 + b_2 t^3 + b_1 t^2 + b_0 t \qquad (3.3)$$

The polynomial fitted phase and the ideal phase are plotted in Figure 3.6. In Figure 3.6a, the green line is the ideal phase and the blue line is the modeled phase. If they were equal, the phase errors would be zero. The actual phase errors are found as the difference between the modeled and ideal phase and shown in Figure 3.6b.

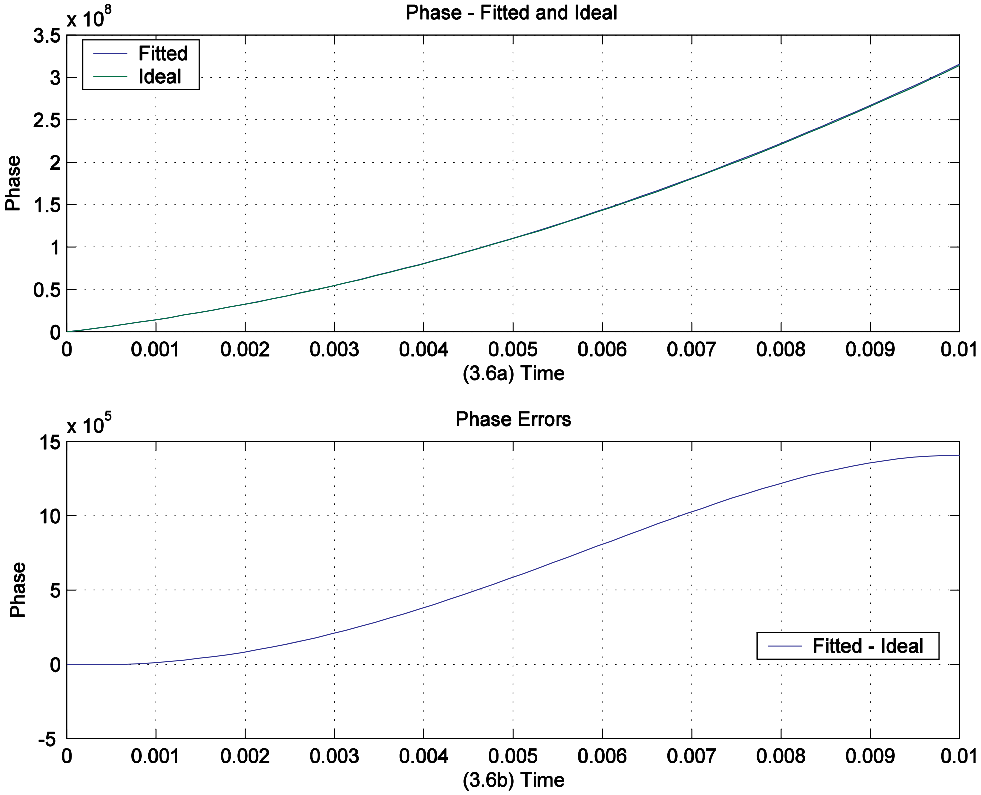


Figure 3.6 - Phase Errors

The coefficients of the polynomial are passed as a parameter to the simulation. The simulation forms the polynomial and uses it as the phase model used to generate the chirp. Thus, the generated chirp includes the phase errors.

3.1.2 Modeling the Transmitter, Local Oscillator and Receiver Sections

The Transmitter, Local Oscillator and Receiver Sections are modeled using their s-parameters. A brief discussion of s-parameters is given below followed by a discussion on modeling using s-parameters.

3.1.2.1 S - Parameters

For an arbitrary N-port network the scattering matrix gives the relation between the incident and the reflected waves at each port.

$$[V^-] = [S][V^+] \tag{3.4}$$

For example, when we consider a 2-port network the relationship is given by

$$\begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix} \tag{3.5}$$

Specifically, the parameter s_{21} is the parameter that links the output wave to the input wave.

$$S_{21} = \left. \frac{V_2^-}{V_1^+} \right|_{V_2^+ = 0} \tag{3.6}$$

This is almost similar to the Transfer Function H(f) of a system.

$$H(f) = \frac{V_o(f)}{V_i(f)} \tag{3.7}$$

In fact, the transfer function is equal to s_{21} if s_{11} and s_{22} are zero. This happens when the ports are matched. The radar is designed with a good impedance match, so we use the s_{21} parameter as the transfer function. So to determine the transfer function of a component, it will suffice if we know the overall S parameters of that component. Moreover, this can be easily measured using a network analyzer.

Once we know the s-parameters, s_{21} can be considered the transfer function and the output Y(f) of that components to an input X(f) can be determined as

$$\begin{aligned} Y(f) &= X(f) * H(f) \\ &= X(f) * S_{21}(f) \end{aligned} \tag{3.8}$$

3.1.2.2 Modeling Using S-Parameters

Using a network analyzer, the s-parameters of these sections are measured. The s_{21} parameter is then interpolated to determine the transfer function of each section. The effects of the antenna section are not included in “system modeling by measurement” as this modeling uses a delay line connected between the transmitter and receiver, thus bypassing the antenna. The simulation of a high bandwidth of a 2-8 GHz forces a high sampling rate of 20 GHz. The intermediate section has a frequency response between 4 KHz and 400 KHz. The frequency response of the IF section is assumed ideal, as this bandwidth requires a very high number of samples at the sampling rate used.

3.2 Modeling by Calibration

As previously mentioned, the point-spread function can be measured by using a calibration target such as a aluminum screen. The spacing of conductors in the screen is 2 mm, which is smaller than one-eighth the smallest wavelength of the transmitted chirp signal. Thus, it acts as a metal reflector. The return from the screen is processed, and will have a single main peak corresponding to the return from the net. This peak is filtered out and the envelope of its inverse Fourier transform is taken as the Point Spread Function. This method of finding the point-spread function is especially effective in that, it includes all the anomalies that might be present in the complete system, including those that might not be effectively measured in the previous technique.

A delay line could also be used as the target, but care should be taken to remove the effects of the delay line before determining the point-spread function. A point-spread function derived using a delay line as the target will not include the effects of the antennas. The point-spread function obtained from one such calibration is shown in Figure 3.7. This point-spread function can be convolved with an ideal return signal to produce the return signal that is corrupted by system effects.

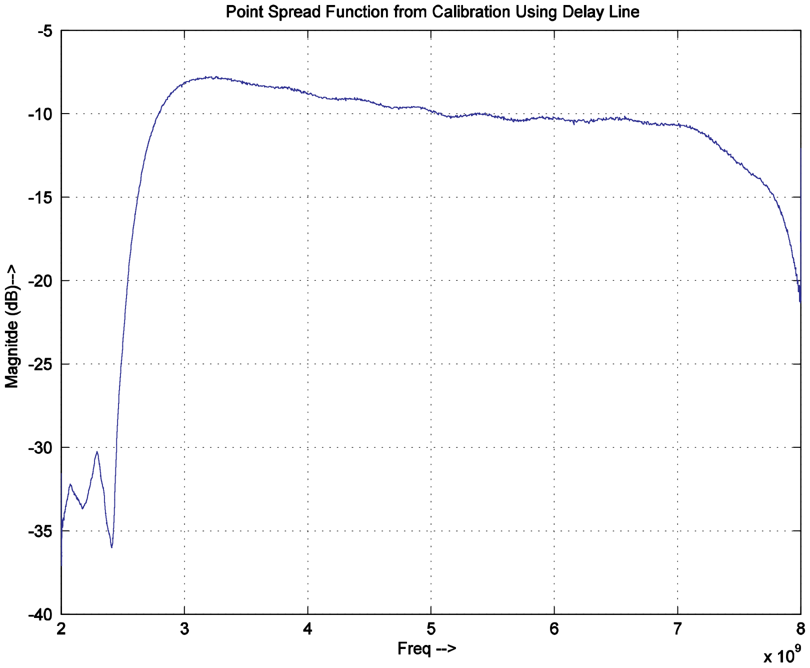


Figure 3.7 - Point-Spread Function by Calibration

3.3 Summary

In this chapter, we looked at two methods of modeling the system. Modeling by measurement involves determining the point spread function by combining the effect of the source model and the transfer functions of the component sections of the radar. Modeling by calibration involves determining the point spread function from the return obtained using a calibration target.

Chapter - 4

Propagation Modeling

The radar return signal from snow-covered sea-ice is characterized by a reflection term and a scattering term. The reflection term arises due to dielectric contrast between the layers that make up the medium. This reflected signal is modeled with the assumption that the layers are perfectly planar and that the antenna beam is a pencil beam. However, the surface is not perfectly smooth and the antenna has a finite bandwidth. This gives rise to the phenomenon of surface scattering. Surface scattering accounts for the fact that the antenna will pick up back-scattered signals from angles other than nadir. Another phenomenon to be considered is volume scattering. Volume scattering occurs because the layers are not homogenous. Rather, snow consists of ice particles suspended in air. The signal entering a layer will be subjected to the effects of scattering and absorption.

The propagation model incorporates the physical and electromagnetic effects of the medium into the simulation. The return waveform can be simulated knowing the effects of the medium on the transmit waveform.

4.1 Geophysical Data

The starting point for the propagation model is the geophysical data used for the simulation. The geophysical data defines the medium. The geophysical data includes information about the depth of each layer, the constituent media of each layer and its properties.

This is best explained using an example. Consider the data in Table 4.1.

Table 4.1 - Geophysical Data of Snow Covered Sea-Ice

Depth **(m)**	**Medium**	**Density** **(gm/cc)**	**Temperature** **(°C)**	**Salinity** **PPT**
1	Air	-	-	-
0.5	Snow	0.500	-8	-
5	Sea-Ice	0.915	-12	40

Snow layer roughness: rms height = 1cm, correlation length = 40cm

Sea-Ice layer roughness: rms height = 1mm, correlation length = 10cm

Here the data says there are three layers in the medium. The first is one meter of air. The second layer is half a meter of snow with a density of 0.500 gm/cc and a temperature of -8 centigrade. The third layer is a five meters of sea-ice with a density of 0.915 gm/cc at -12 centigrade and 40 PPT salinity.

Roughness information about the interfaces like the rms height and the correlation lengths are also given. These parameters are used to determine the scattering return. These will be introduced later in the chapter.

4.2 Dielectric Profile

The geophysical data must be converted into a dielectric profile before it can be used in the simulations. This entails determining the dielectric constant of every layer. This dielectric constant depends on physical parameters and frequency.

4.2.1 Dielectric Constant of Dry Snow

The dielectric constant of dry snow can be computed using the Tinga mixing model for dry snow [5]. The real part is computed in Equation 4.1a and the imaginary part in Equation 4.1b.

$$\varepsilon'_{ds} = 1 + \frac{2\upsilon_i(\varepsilon'_i - 1)}{(2+\varepsilon'_i) - \upsilon_i(\varepsilon'_i - 1)} \tag{4.1a}$$

$$\varepsilon''_{ds} = \frac{9\upsilon_i\varepsilon''_i}{[(2+\upsilon_i)+\varepsilon'_i(1-\upsilon_i)]^2} \tag{4.1b}$$

Here ε_{ds} is the dielectric constant of dry snow, ε_i is the dielectric constant of ice and ν_i is the ice volume fraction.

4.2.2 Dielectric Constant of Wet Snow

The dielectric constant of wet snow can be computed using the empirical Debye like model [5]. The real part is computed in Equation 4.2a and the imaginary part in Equation 4.2b.

$$\varepsilon'_{ws} = A + \frac{0.073\, m_v^{1.31}}{1 + \left({}^{f}\!/\!{}_{f_0} \right)^2} \tag{4.2a}$$

$$\varepsilon''_{ws} = \frac{0.073\left(f \big/ f_0 \right) m_v^{1.31}}{1+\left(f \big/ f_0 \right)^2} \qquad (4.2b)$$

Here ε_{ws} is the dielectric constant of wet snow, f_0 the relaxation frequency, f the frequency and m_v is the volume fraction of water.

The dielectric constant of air is assumed one and the dielectric constant for sea-ice is computed using an implementation of the Tinga Mixing Model in the RSL simulation package for Radar Systems [11].

4.3 Plane Surface Reflection

The first order reflected term is found by assuming that we are dealing with a pencil beam incident on perfectly flat layers. Consider that a dielectric profile as shown in Figure 4.1 was derived from the shown geophysical data.

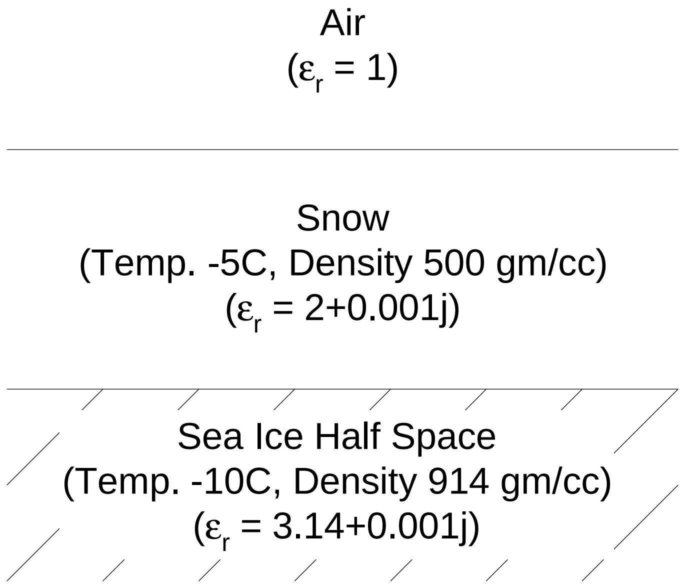

Air
(ε_r = 1)
Snow
(Temp. -5C, Density 500 gm/cc)
(ε_r = 2+0.001j)
Sea Ice Half Space
(Temp. -10C, Density 914 gm/cc)
(ε_r = 3.14+0.001j)

Figure 4.1 - Geophysical data and its Dielectric Profile

This multilayered medium with distinct dielectric constants can be considered similar to a transmission line with multiple sections. It is then characterized by an equivalent reflection coefficient at the surface. The power returned from such a surface is given by

$$P_r = \frac{P_t \lambda^2 G^2 \left|\Gamma\right|^2}{(4\pi)^2 (2R)^2} \tag{4.3}$$

This expression for return power assumes all the layers are perfectly planar. In the presence of roughness, the specular reflection Γ_{sp} coefficient needs to be changed to account for the surface as follows [12].

$$\Gamma = \Gamma_{sp} e^{-4k^2\sigma_h^2} \tag{4.4}$$

Here Γ is the effective reflection coefficient, k is the wave number and σ_h is the rms height.

4.4 Surface Scattering

The surface of snow-covered sea-ice is not perfectly smooth but is characterized by roughness. This roughness is specified in terms of the rms height and the correlation length. A signal ray incident normally on this rough surface does not undergo specular reflection. Instead, it is scattered at different angles. This would also be acceptable if we had a single ray incident on the target as it will only affect the amount of power returned to the receive antenna. However, as the transmit beam width is finite, rays scattered from the surface at angles other than nadir also return to the receive antenna. These might be mistaken for returns from nadir.

The rms height is the standard deviation of the height of a surface. The correlation length is the distance at which the normalized autocorrelation of the surface height falls to (1/e) where e is the exponential constant (base of natural logarithms). As the rms height of a surface increases, it becomes rougher and as the correlation length of a surface increases it becomes smoother. Sometimes the ratio of the rms height to the correlation length called the surface slope is used as an indicator of roughness.

A roughness measurement made in Antarctica during the September 2003 field trip and its autocorrelation is shown in Figure 4.2. The data were measured by taking photographs of a ruler inserted into the snow. The photos were digitized and the height of the snow was stored. The digitized data from the photos were then combined and the rms height and the correlation length were computed. The rms height measured for this dataset was 59.86 cm and the correlation length was 3.73

mm. We assume that the underlying ice has similar but lower roughness compared to the surface. The ice surface roughness is assumed to be lower as it is not affected by wind effects.

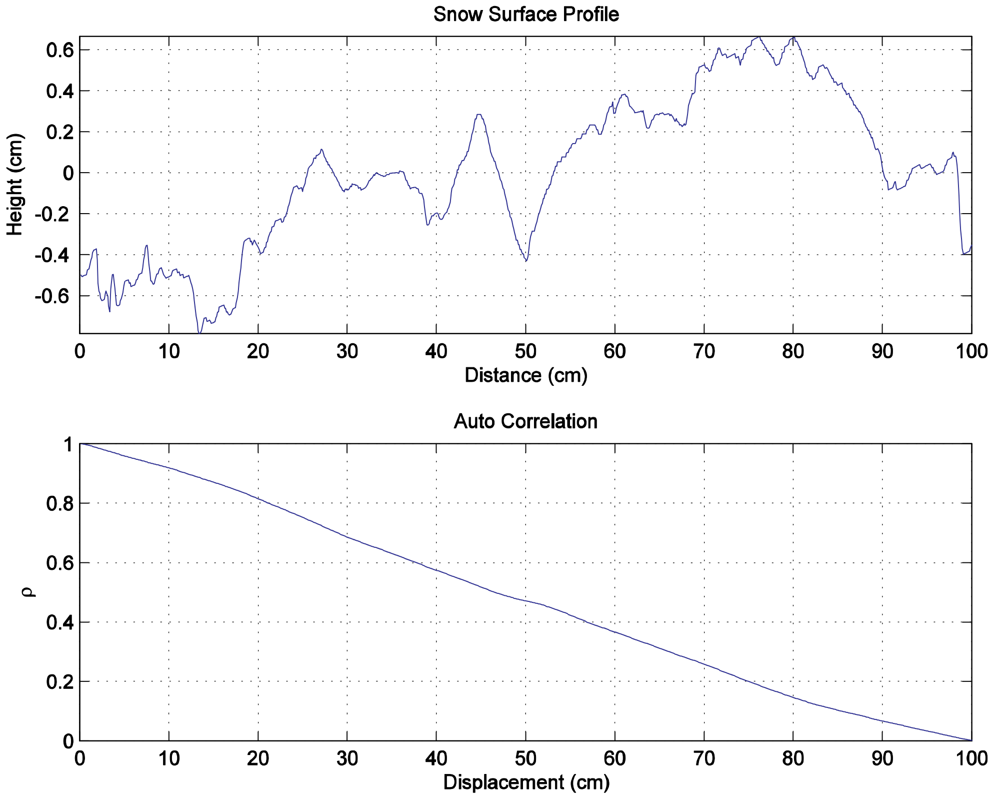


Figure 4.2 - Roughness Profile and Autocorrelation

This rough surface return is modeled for two scales of roughness. For surfaces with high roughness scales where the ratio of rms height to correlation length is greater than one we use the Kirchhoff model. For small scales of roughness, we use a special case of Integral Equations Method (IEM) for low roughness [13]. For the Kirchhoff Model the scattering coefficient is given as follows:

$$\sigma_{pp}^{0}(\theta) = \left(\frac{|R(0)|^2}{\left(2\sigma_s^2 \cos^4\theta\right)} \right) e^{\left(\frac{-\tan^2\theta}{2\sigma_s^2} \right)} \tag{4.5}$$

Here σ_s is the surface slope and R(0) the Fresnel reflection coefficient.

For the IEM model applied to small roughness media the scattering coefficient is

$$\sigma_{pp}^{0} = \frac{k^2}{2} e^{\left(-2k_Z^2\sigma^2\right)} \sum_{n=1}^{\infty} \left| I_{pp}^{n} \right|^2 \frac{W^{(n)}(-2k_x,0)}{n!} \tag{4.6a}$$

$$I_{pp}^{n} = \left(2k_z\sigma\right)^n f_{pp} e^{\left(-k_z^2\sigma^2\right)} + \frac{\left(k_z\sigma\right)^n \left[F_{pp}\left(-k_x,0\right) + F_{pp}\left(k_x,0\right)\right]}{2} \tag{4.6b}$$

$$\begin{aligned} &F_{vv}\left(-k_x,0\right) + F_{vv}\left(k_x,0\right) \\ &= \frac{2\sin^2\theta\left(1+R_{\parallel}\right)^2}{\cos\theta} \left[\left(1 - \frac{1}{\varepsilon_r}\right) + \frac{\mu_r\varepsilon_r - \sin^2\theta - \varepsilon_r\cos^2\theta}{\varepsilon_r^2\cos^2\theta} \right] \end{aligned} \tag{4.6c}$$

Here $k_z = k \cos \theta$, $k_x = k \sin \theta$ and pp = vv for vertical or hh for horizontal polarization and R the Fresnel reflection coefficient.

4.5 Volume Scattering

The above calculations of reflected power and surface scattering assume that the layer media are homogenous. However, that is not the case. They are made up of independent particles suspended in a background medium. For example, snow can be considered as ice suspended in a background medium of air.

If such a scenario is considered, it will be seen that a signal incident on such a media will undergo scattering and absorption. The volume backscattering coefficient is given by [12]

$$\sigma_v = N\sigma_b \tag{4.7}$$

And the extinction coefficient is

$$k_e = NQ_e \tag{4.8}$$

Here N is the number scattering particles per unit volume and σ_b and Q_e are the backscattering and extinction cross-sections respectively. The one-way loss factor is given by:

$$L(\theta) = e^{k_e h \sec\theta} \tag{4.9}$$

And the effective volume backscattering coefficient is

$$\sigma_v^o = \frac{\sigma_v \cos(\theta)}{2k_e}\left(1 - \frac{1}{L^2(\theta)}\right) \tag{4.10}$$

4.6 Combining Scattering Coefficients

For the case of a single layer of snow on top of sea-ice, we see that the major scattering terms are σ_{ss} from the snow surface, σ_{sv} from the snow volume and σ_{is} from the ice surface. These three scattering coefficients are combined to derive a composite scattering coefficient as shown below [14]:

$$\sigma^0(\theta) = \sigma_{ss}^0(\theta) + T_s^2(\theta)\left[\sigma_{sv}^0(\theta') + \frac{1}{L^2(\theta')} \cdot \sigma_{is}^0(\theta')\right] \tag{4.11}$$

This scattering coefficient can be used to determine the backscattered power using the familiar radar equation.

$$P_r = \frac{P_t \lambda^2 G^2 \sigma^0 A}{(4\pi)^3 (R)^4} \tag{4.12}$$

Simulated values of σ_{ss}, σ_{sv} and σ_{is} for specific values of the correlation length and rms heights are shown in Figure 4.3. The gain is assumed to be of a Gaussian shape with a maximum value of one. The range also varies with distance.

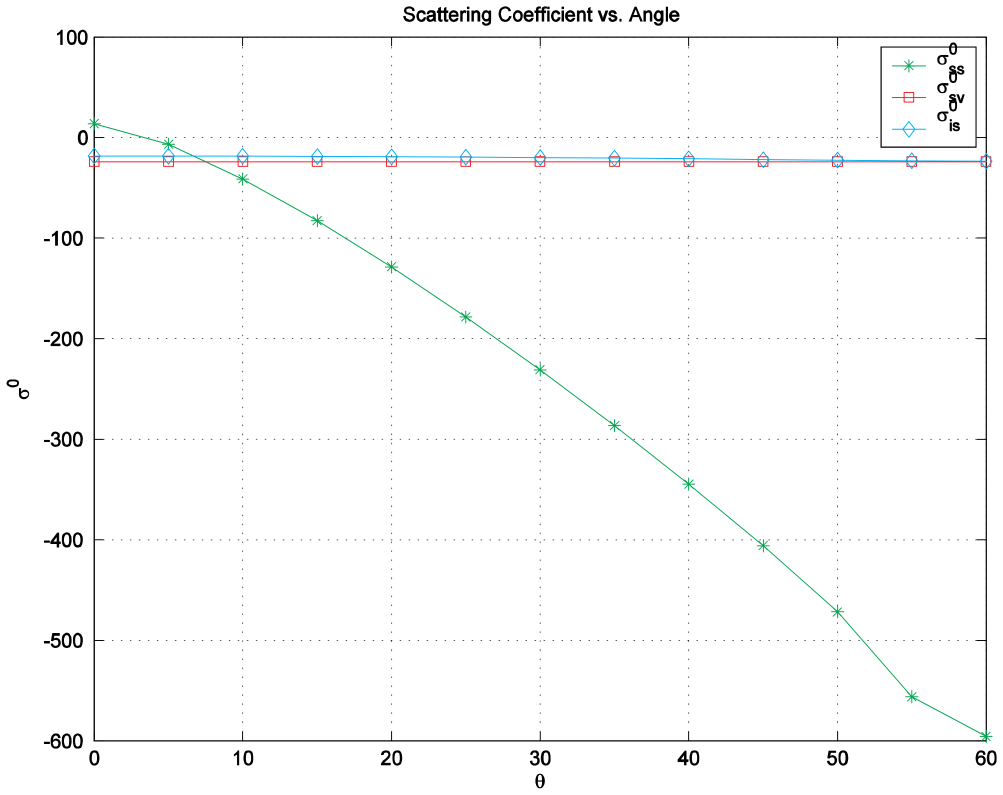


Figure 4.3 - Scattering Coefficient vs. Angle

4.7 Summary

In this chapter, we looked at propagation modeling. Starting with the given geophysical data, we generate the dielectric profile. From the dielectric model, we can compute the Fresnel reflected power. Surface roughness parameters are introduced and the surface scattering coefficient is computed. Volume scattering coefficient is also computed. Then these are combined to determine the overall scattering coefficient from which we can compute the backscattered power.

Chapter - 5

Simulation of the Snow Radar

The goal of this research is to simulate the snow radar, starting with a simple and ideal mathematical model and then increasing the complexity of the model by incorporating system effects and scattering effects. The inclusion of these effects makes the output of the simulation more realistic. The simulations are performed in MATLAB™, an interactive mathematical package from The Mathworks, Inc.

5.1 Simulation Parameters

The primary simulation parameters are:

1. Sampling frequency / Sample time.
2. Number of samples / Chirp time.
3. Radar Parameters and Data.
4. Target Information.

5.1.1 Sampling Frequency

The sampling time has to be chosen to satisfy the Nyquist criterion, which requires that the sampling frequency be at least twice the maximum frequency of the sampled signal. As a general rule, simulations are usually run at a higher rate than the minimum required by Nyquist rate.

The signal to be simulated is a 2-8 GHz chirp. The maximum frequency is 8 GHz. Therefore, the Nyquist sampling rate is 16 GHz, so we have chosen 20 GHz as the sampling frequency of the simulation.

5.1.2 Number of Samples

The number of samples is a very important simulation parameter. The higher the number of samples the longer the simulation takes. In the case of the FMCW radar the number of samples is also directly related to the chirp time. The chirp time is the product of the number of samples and the sample time.

The actual chirp time of the 2-8 GHz chirp signal is 10 ms. This gives us an inordinately high number of samples as shown below.

Chirp Time	=	10 ms
Sampling Frequency	=	20 GHz
Number of Samples	=	Chirp Time * Sampling Frequency
	=	200 Mega Samples

To avoid this problem, we choose to simulate the signal with a specific number of samples. By fixing the number of samples and the sampling frequency of the simulation, the simulation chirp time is not equal to actual chirp time. With a sampling frequency of 20GHz, the number of samples was chosen to be 2^{17} as it is found that 2^{17} samples is sufficient to simulate the distances involved.

5.1.3 Radar Parameters and Data

Other inputs to the simulation include information about the radar system. These include radar parameters such as the start frequency and stop frequency of the radar, radar source models such as the amplitude and phase error models and radar system information such as the point spread function.

5.1.4 Target Information

Information about the target such as delay line length and transfer function or the geophysical data are also inputs to the simulation. The geophysical data includes information about the constituent media of the target and its scattering parameters such as surface rms height and correlation length.

5.2 Simulation Technique

The simulation proceeds according to the following steps. First the simulation parameters such as the sampling frequency and number of samples are initialized. Then the radar system information and target information are read in. The transmit waveform is then generated according to the source model. The simulation then follows two branches.

In the first branch, The reflection coefficient is generated from the geophysical data. The reflected wave is then generated using the reflection coefficient. The system effects are then included by convolving this signal with the point spread function. This is then mixed with the transmit waveform and the reflected power is determined. Random noise is also incorporated at every range cell.

In the second branch, the scattering coefficient is generated from the geophysical data. Random phase noise is incorporated into the scattering coefficient and the backscattered wave is determined. Systems effects are included by convolving with the point-spread function. Multiple backscattered waves are generated and the mean backscattered power is computed as the mean power of the multiple backscattered waves mixed with the transmit wave.

The total return power is computed as the sum of reflected and backscattered powers. The distance axis is computed after compensating for the variation in wave velocity due to difference in dielectric constant. This return power is then plotted versus the distance. This process can be represented by the flowchart below. The mathematical foundation of the procedure is explained in detail in the following section. The actual simulation code is shown in Appendix A.

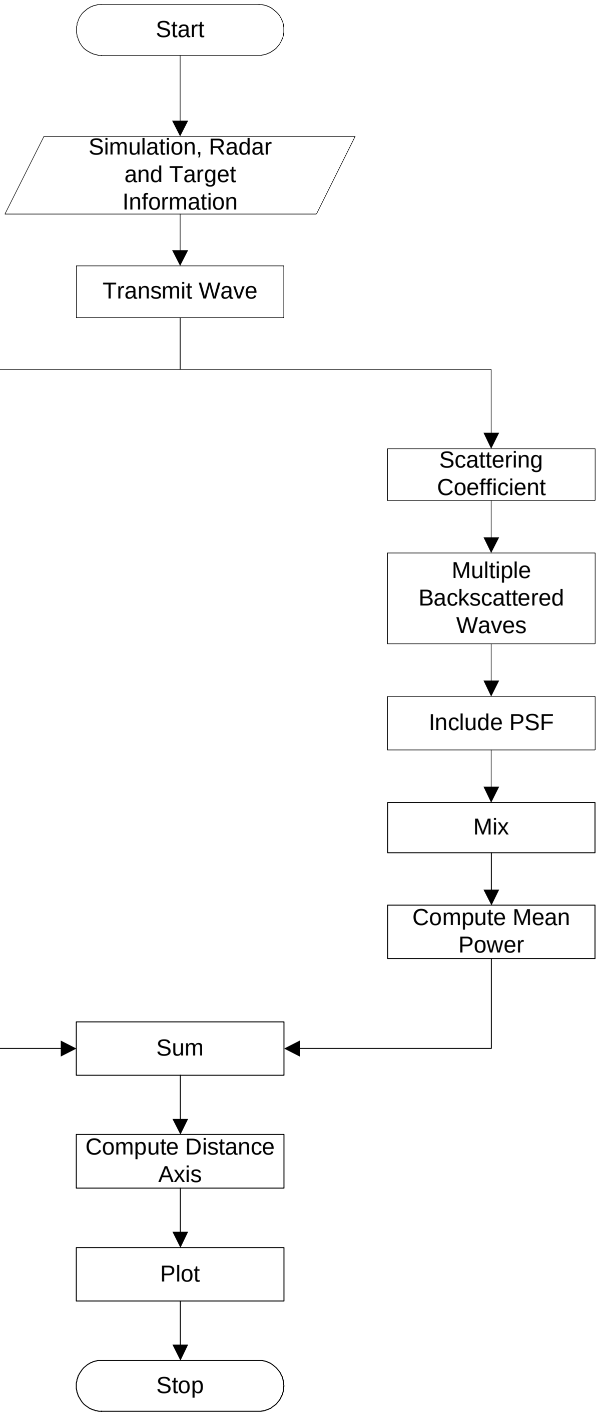


Figure 5.1 - Simulation Flowchart

5.3 Numerical Basis of the Simulation

5.3.1 Generation of the Transmit Waveform

The transmit waveform is a 2-8 GHz chirp. The ideal chirp x(t) is generated according to Equation 5.1. Here, f_0 is the start frequency and α the sweep rate.

$$x(t) = \cos\left(2\pi\left(f_0 t + \tfrac{1}{2}\alpha \cdot t^2\right)\right) \tag{5.1}$$

To generate the waveform with amplitude errors the amplitude model is used. To use the amplitude error model the inverse DCT (idct) of the model has to be taken. If “amerr” is the amplitude error model (Sec. 3.1.1.1), the transmit wave is given by

$$x(t) = \left(1 + idct(amerr)\right)\cos\left(2\pi\left(f_0 t + \tfrac{1}{2}\alpha \cdot t^2\right)\right) \tag{5.2}$$

To include phase errors into the simulations the phase error polynomial (Equation 3.3) is used. If the phase polynomial is ϕ(t) the transmit waveform is given by

$$x(t) = \left(1 + idct(amerr)\right)\cos\left(\phi(t)\right) \tag{5.3}$$

5.3.2 Computation of Reflection Coefficient

The reflection coefficient of a ray normally incident on an interface between two media with dielectric constants ε_1 and ε_2 is given by [15]

$$\Gamma = \frac{\sqrt{\varepsilon_2} - \sqrt{\varepsilon_1}}{\sqrt{\varepsilon_2} + \sqrt{\varepsilon_1}} \tag{5.4}$$

Multilayered media can be a treated as a single interface by incorporating one layer at a time. An illustration of this is described with an example of a three-layered medium [16]. Consider a ray incident on a three-layered medium as shown in Figure 5.2. With Γ the reflection coefficient and T the transmission coefficient, the effective reflection coefficient is given as

$$\Gamma_{eff} = \Gamma_{12} + T_{12} \frac{\Gamma_{23} T_{12} e^{-j2k_2 d_2}}{1 - \Gamma_{23}\Gamma_{21} e^{-j2k_2 d_2}} \tag{5.5}$$

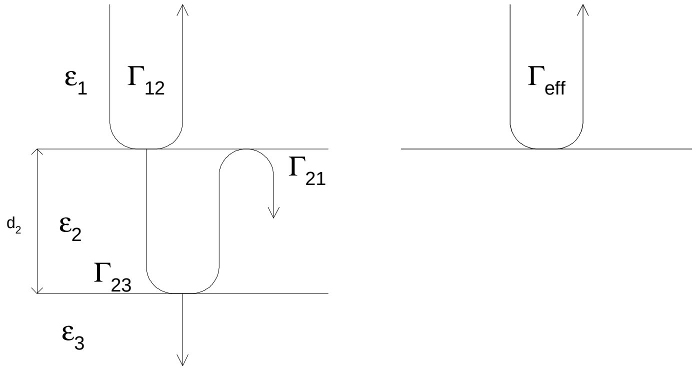


Figure 5.2 - Multilayered Media

5.3.3 Generation of Reflected Waveform and Power

The reflected waveform is generated as the convolution of the transmit waveform with the effective reflection coefficient. For a three layer (two interface) medium the process is illustrated in Figure 5.3. The two interfaces have reflection coefficient peaks corresponding to their distance.

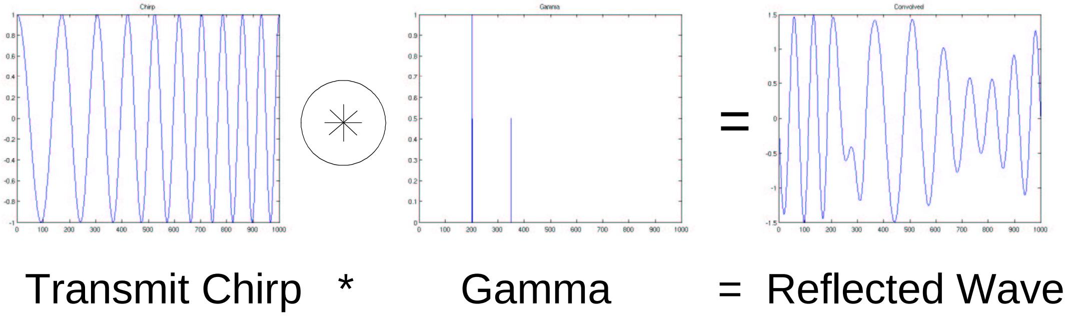


Figure 5.3 - Generation of Return Waveform

The IF signal is then computed by mixing this reflected signal with the transmit waveform and convolving with the Point Spread Function. Then the Fast Fourier Transform of the IF signal is computed. This gives the reflected power vs. frequency. We know that range is directly proportional to frequency for an FMCW radar as shown in Equation (2.7). Therefore, the obtained FFT is also a measure of reflected power as a function of distance. In order to incorporate random noise in to the simulation we now add random noise power at every range bin of the distance. The power level of this random noise is taken from the noise floor of actual radar data.

5.3.4 Generation of Backscattered Waveform and Power

The scattering coefficient is calculated as a function of scattering parameters like the rms height and correlation length using Equations (4.6), (4.7), (4.10), and (4.11). A random phase is added to this scattering coefficient σ^0. This is then convolved with the transmit wave to determine the backscattered wave. This process is repeated to generate multiple backscattered waves. Each of these waves is then mixed with the transmit waveform and convolved with the point spread function. Then power is calculated. Finally, their mean is taken as the backscattered power versus range.

5.3.5 Computation of Distance Axis and Display

The distance axis is computed by incorporating the velocity of EM waves in each medium. The velocity of EM waves depends on the dielectric constant of each medium.

$$c = \frac{1}{\sqrt{\varepsilon_r \varepsilon_0 \mu_0}} = \frac{c_0}{\sqrt{\varepsilon_r}} \tag{5.6}$$

Here c_0 is the speed of light in free space. The distance axis is found by using the relationship between range and frequency shown in Equation 5.7.

$$D = \frac{f \cdot c \cdot T}{2 \cdot B} \tag{5.7}$$

Here f is the frequency, c the velocity of EM waves in each medium, T the sweep time and B the sweep bandwidth. The sum of the reflected and scattered waves is plotted versus distance as the final output of the simulations. The results of these simulations will be shown in the next chapter.

5.4 Summary

In this chapter, the simulation technique was explained in detail. The input to the simulation includes simulation parameters, radar data and target data. The process of simulation is explained with its mathematical basis. The output of the simulation is the total return power versus distance.

Chapter - 6

Simulation Results

The simulations explained in the previous chapter are performed on a systematic basis. The first simulations involved are ideal simulations with a sample delay. This was introduced in Chapter 2. The first step is to include the system effects by including the point spread function of the system for a delay line case. Then the propagation effects are included in the simulation. Finally, the simulation results are compared with actual results for both delay line and actual snow measurements.

6.1 Delay Line Simulation

A simulation model of the radar system for a delay line target is shown in Figure 6.1. It consists of the all the sections of the radar except the antennas. The target is actually a delay line instead of snow over sea-ice. This model is simulated and the result obtained is plotted in Figure 6.2.

The delay line simulation is the green plot and the ideal simulation the blue plot. The simulated response is delayed due to delays introduced by the components in the system. The main lobe is also broader due to amplitude errors. Phase errors contribute to unequal side lobe levels and the filling of nulls in the plot.

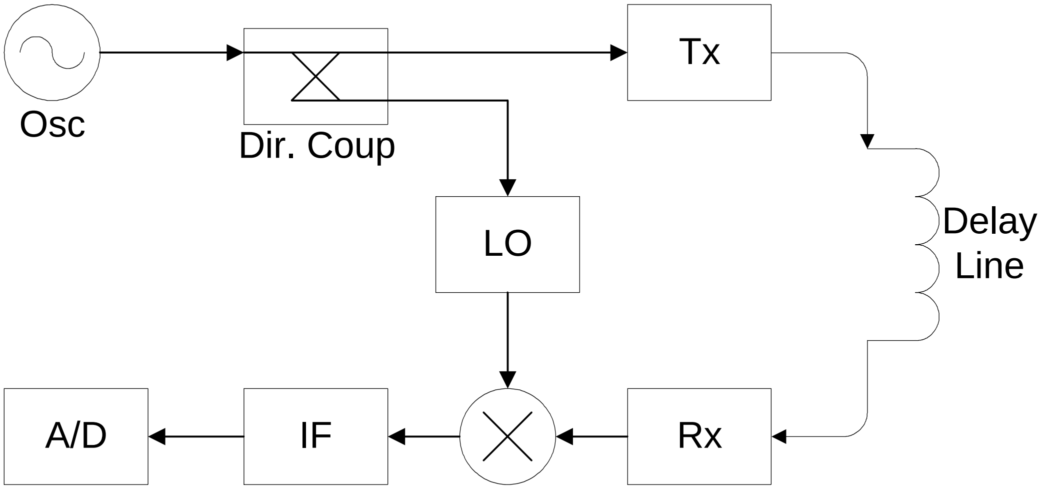


Figure 6.1 - Delay Line Simulation Model and Measurement Setup

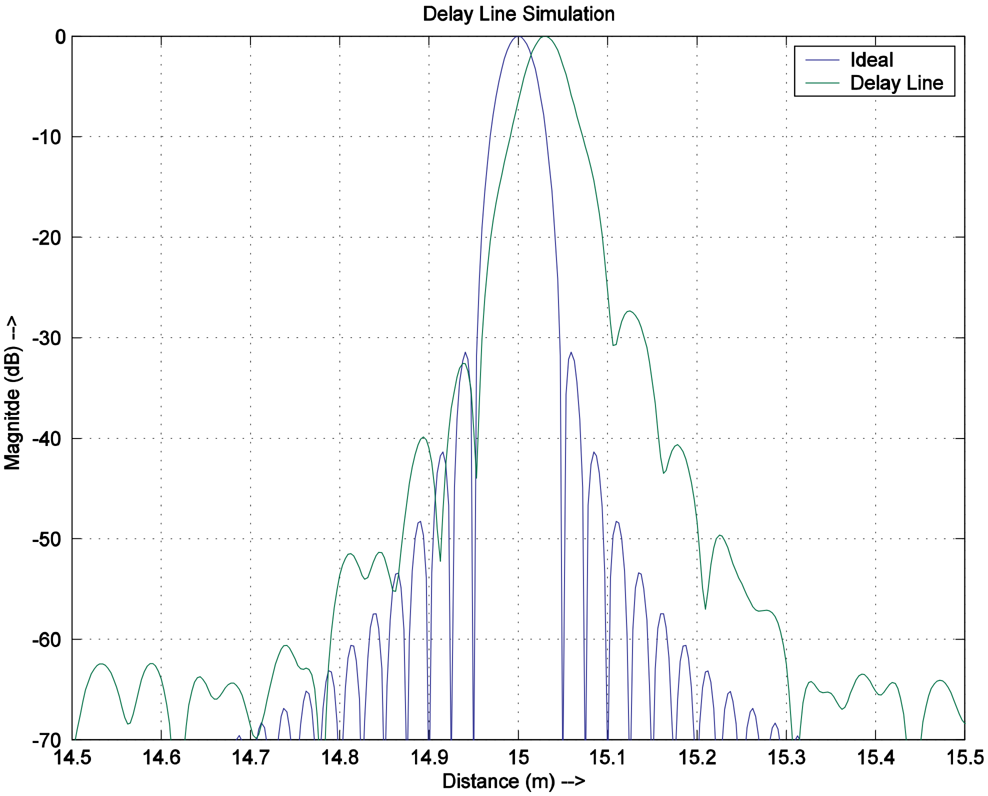


Figure 6.2 – Comparison of Delay Line Simulation Result with Ideal Result

6.2 Snow Simulation

A simulation model of the radar system for simulation over snow-covered sea-ice is shown in Figure 6.3. The difference between this model and the previous model is the inclusion of antennas and also the inclusion of a simulation target of snow over sea-ice.

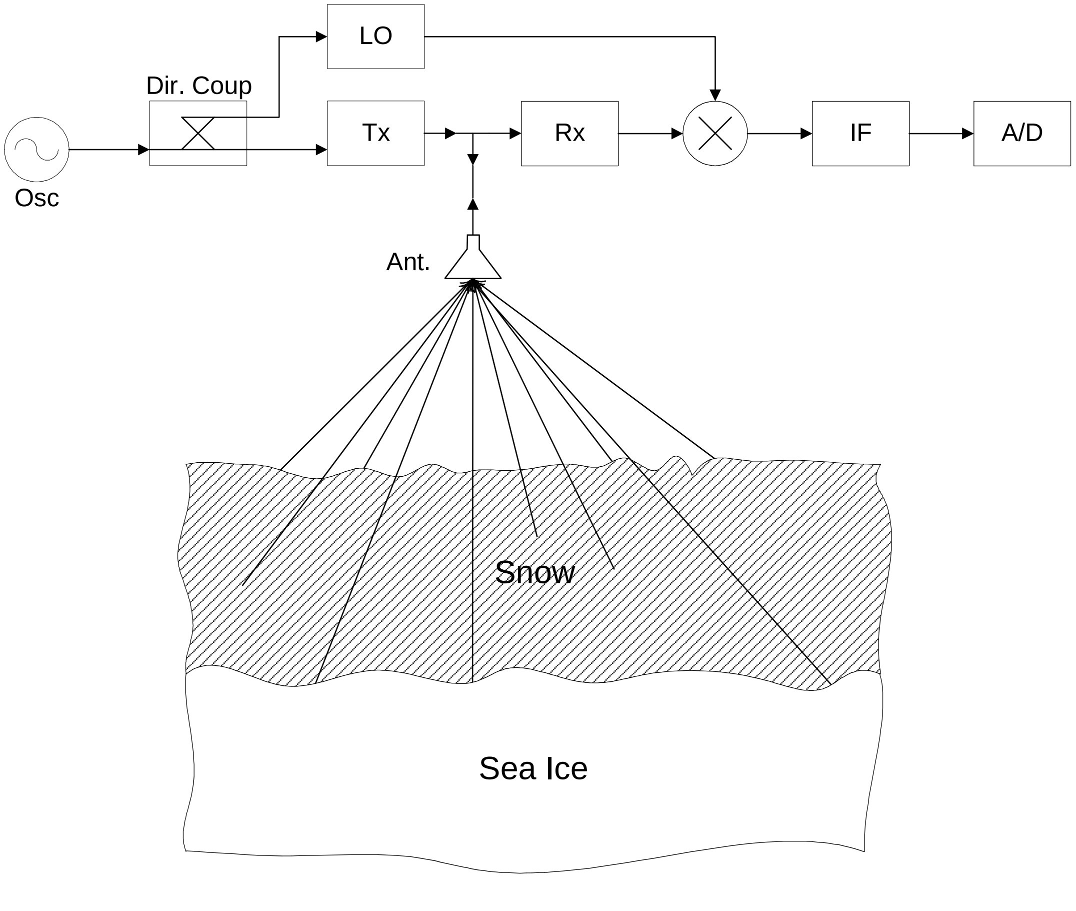


Figure 6.3 - Simulation Model of Snow over Sea-Ice

The results of this simulation are generated for a variety of cases. The simplest case is when the snow surface and the sea-ice surface are assumed planar. The result for this case is shown in Figure 6.4. The next case is when the snow surface is rough but the sea-ice surface is assumed planar. This result is shown in Figure 6.5. The next case is when the snow surface is planar and the sea-ice surface is assumed rough. This result is shown in Figure 6.6. Finally, we look at the case when both surfaces are rough and result for this case is plotted in Figure 6.7.

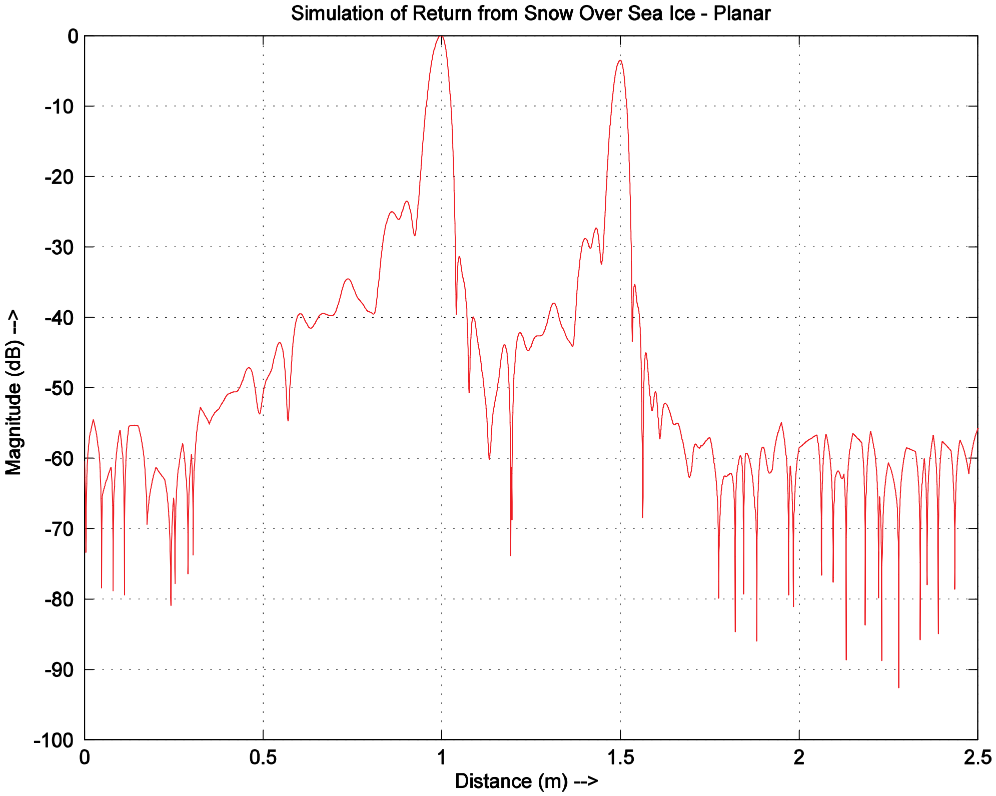


Figure 6.4 - Simulation of Return from Snow over Sea-Ice - Planar

Here the antenna height is assumed as 1 m. The snow layer depth is assumed as 0.5 m. These assumptions are typical of a scenario of ground based testing of the snow radar. Both the surfaces are assumed planar. In addition, volume scattering is ignored. The plot shows two peaks corresponding to the distances of 1 m and 1.5 m.

The peaks are not symmetric. This is due to the effects of the system that have been incorporated by computing the point spread function. For this example the point spread function has been computed from a calibration file. In addition, random noise has contributed to the noise floor seen at about –60 dB.

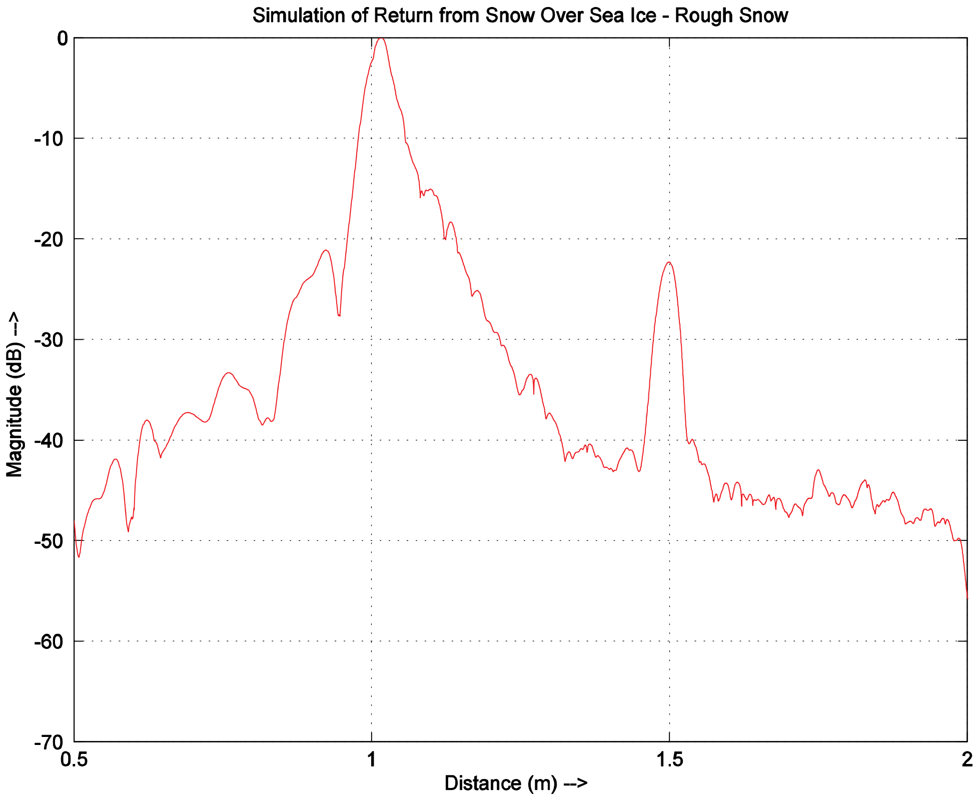


Figure 6.5 - Simulation of Return from Snow over Sea-Ice - Rough Snow Surface

In this simulation, the snow surface is assumed be rough with an rms height of 2 mm and correlation length of 2 m. The effects of snow surface scattering and volume scattering are included. The simulation also includes phase noise. The effect of snow surface roughness on the return is clearly seen. The snow surface return is no longer sharp but falls as a function of distance. This is because the scattered responses from the snow surface at side angles reach the antenna after a time delay corresponding to a greater depth than the actual layer depth.

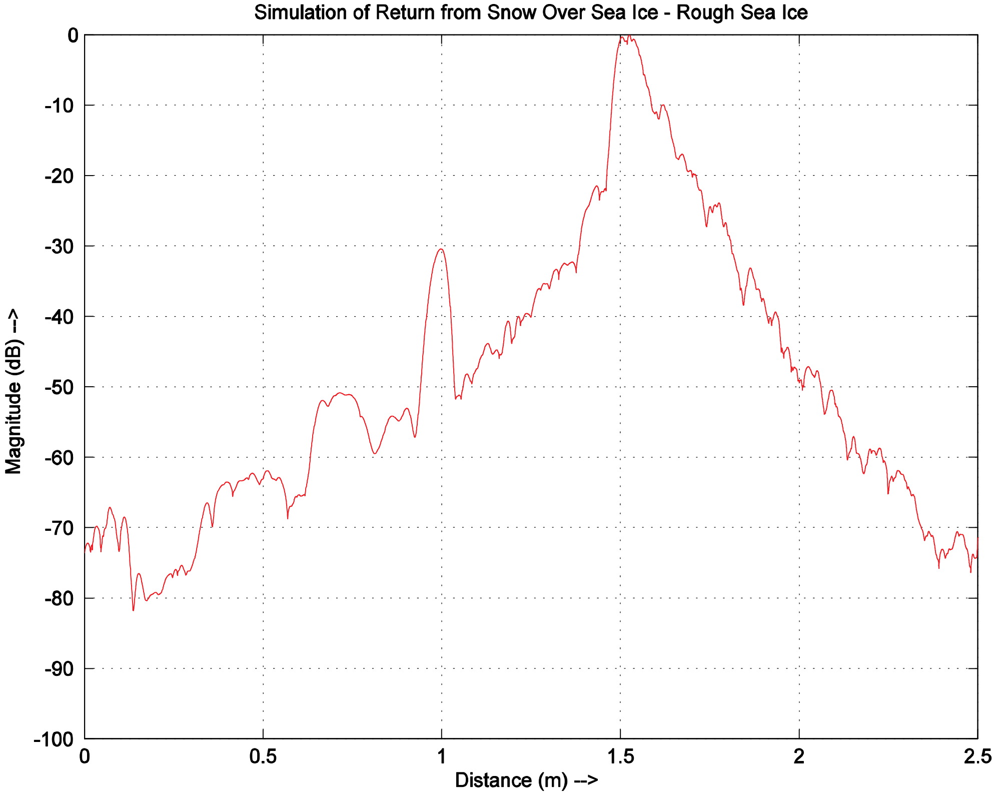


Figure 6.6 - Simulation of Return from Snow over Sea-Ice – Rough Sea-Ice surface

In this simulation, the sea-ice surface is assumed be rough with an rms height of 5 mm and correlation length of 1 m. The effects of sea-ice surface scattering and volume scattering are included. The simulation also includes phase noise. The effect of sea-ice surface roughness on the return is clearly seen. The second peak drops off with distance gradually. This is because the scattered responses from the sea-ice surface at side angles reach the antenna after a time delay corresponding to a greater depth than the actual layer depth.

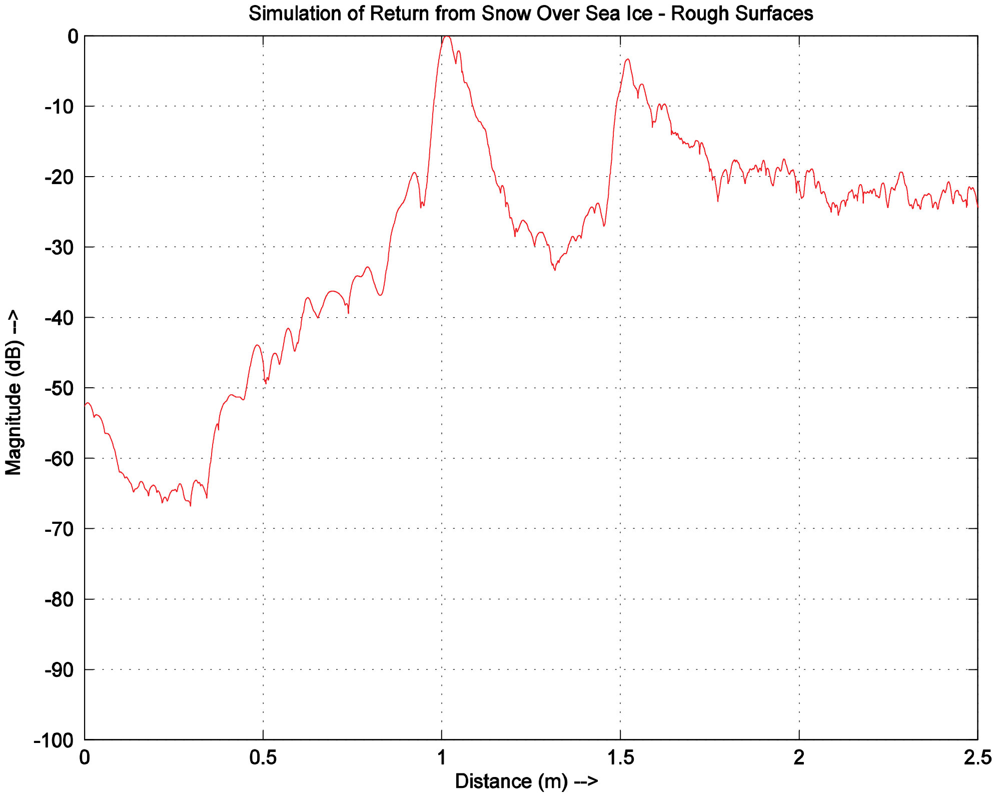


Figure 6.7 - Simulation of Return from Snow over Sea-Ice - Rough Surfaces

In this simulation, both the snow and the sea-ice surfaces are assumed rough. The gradual falloff is seen for both the peaks.

6.3 Comparison of Delay Line Simulation and Results

The delay line simulation result is compared with the actual delay line measurements taken with the snow radar. The measurement setup is as shown in Figure 6.1. The antennas are removed from the radar and delay line is connected instead. The comparison result is shown in Figure 6.8.

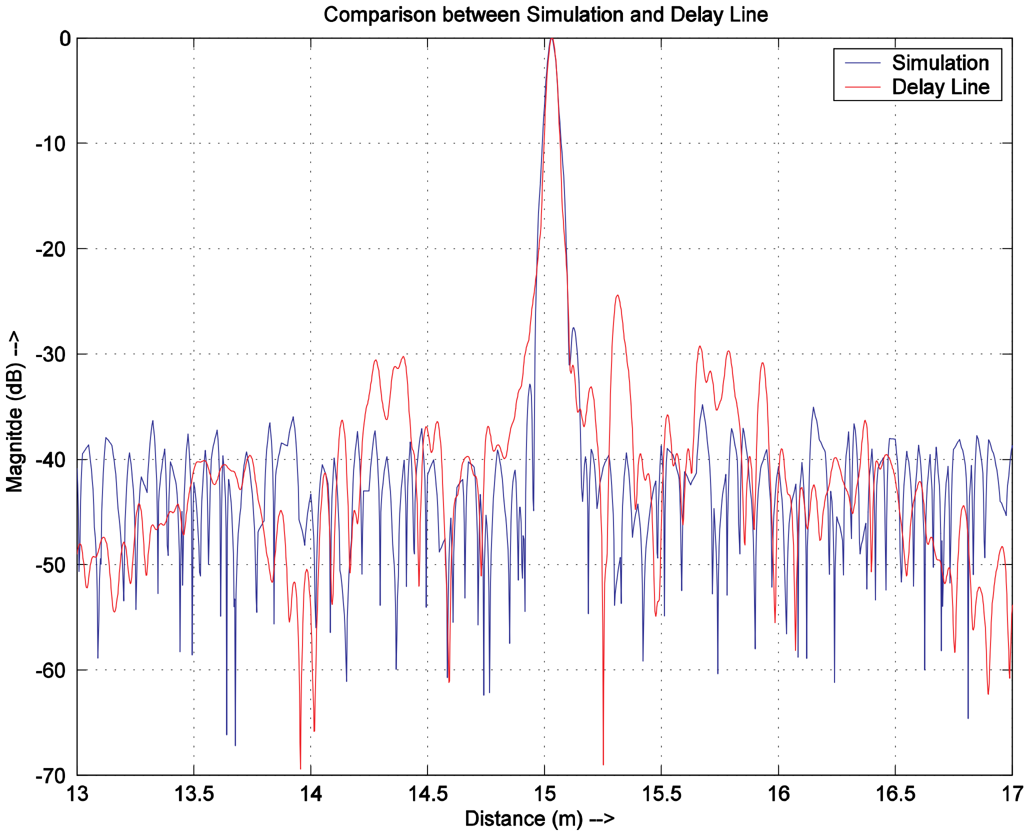


Figure 6.8 - Comparison between Delay Line Simulation and Radar Data

The data clearly shows that the simulation peak matches the measured delay line peak. The noise floor also lies on the same level. There are two amplitude sidelobes (between 14.25m and 14.4m and between 15.6m and 15.8m) in the actual data that are not seen in the simulation. These could be because of some discontinuity in the delay line that appears as an additional target. Otherwise these could be due to the effect of the amplitude error model not being accurate enough.

6.4 Comparison of Snow Simulation and Results

The simulation of snow over sea-ice is compared with the actual data sets from measurements taken with the snow radar. The measurement setup is as shown in Figure 6.9. The significant difference between the simulation model and the actual radar setup is the presence of two antennas. The measurement was conducted during field trials in Antarctica in September – October 2003. Some results from the field trial are shown in Appendix B.

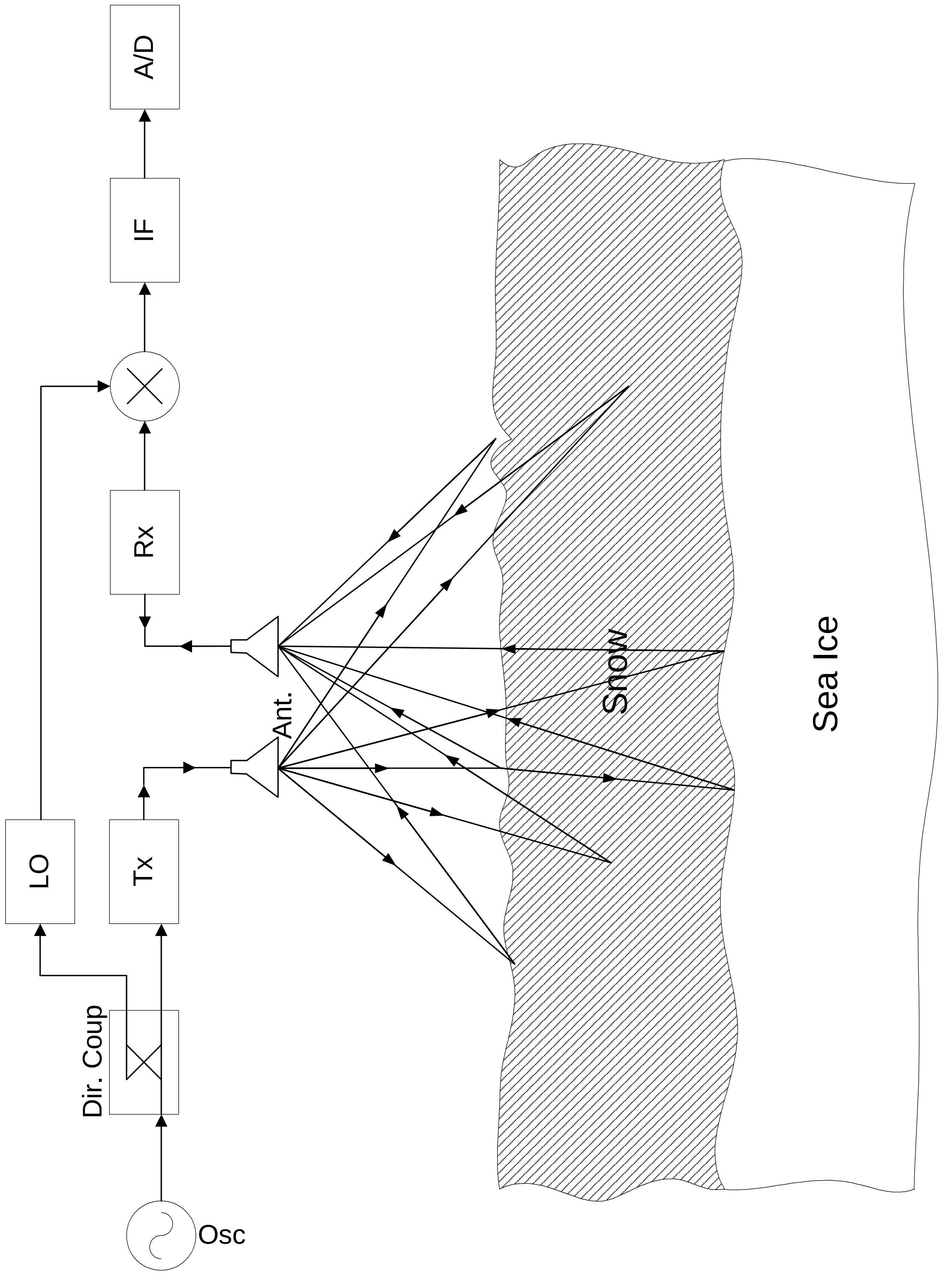


Figure 6.9 - Actual Snow Radar Measurement Setup

The actual radar result is compared with simulation without scattering in Figure 6.10. The actual radar data is compared with simulation including scattering is shown in Figure 6.11.

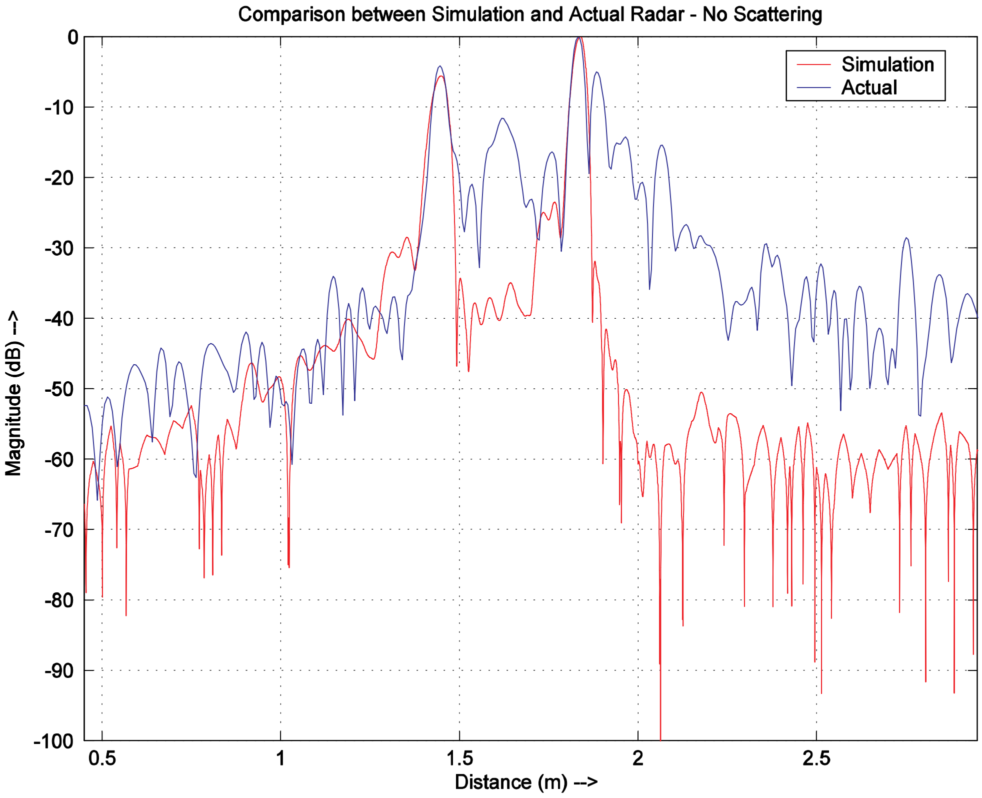


Figure 6.10 - Comparison of Actual Radar Data with Simulation without scattering.

This case clearly shows the effect of not including scattering effects into the simulation. As can be seen the peaks match very well. However, the simulation is unable to reproduce the falloff due to scattering. The next result shows the effect of including scattering into the simulation.

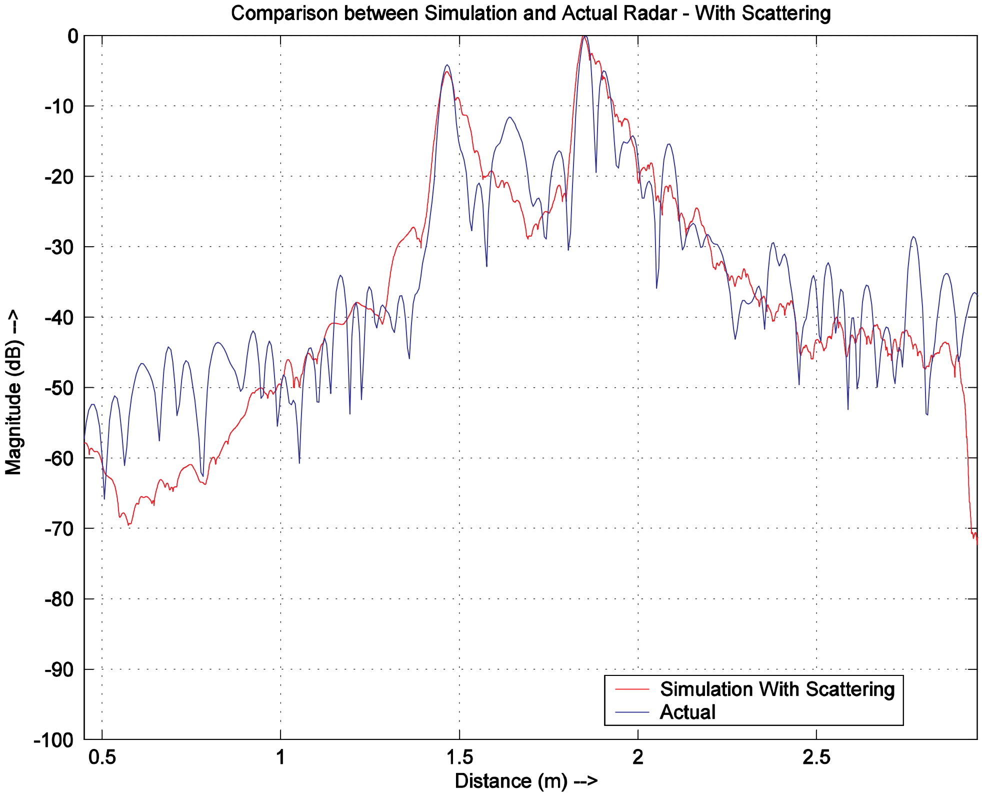


Figure 6.11 - Comparison of Actual Radar Data with Simulation with Scattering Result

The result shows the falloff with distance cause due to scattering. While it is clearly impossible to reproduce the actual phase noise peaks, the simulation does a good job in predicting the rate of falloff. This is because of the introduction of mean random phase noise while computing the scattered power. The simulation was done with a two-layer snow over sea-ice model. The measured depth profile consists of an additional snow layer at around 1.62m and this is seen as the peak at 1.62m. This peak is not seen in the simulated result as the simulation considers only one snow layer.

6.5 Summary

The results of simulations performed are analyzed in this chapter. For each simulation, the Simulation Model is presented and explained first. Then the results are presented and explained. Actual test measurements from delay line and from field trials in Antarctica are shown.

Chapter - 7

Summary and Future Work

Measurement of thickness of snow over sea-ice is an important task in learning about global environmental effects. The Radar and Remote Sensing Lab at the University of Kansas has developed the Snow Radar to measure Snow thickness. This would be used to validate large-scale measurements of snow thickness from satellite-based sensors.

7.1 Summary

The goal of this research work is to simulate the snow radar system and incorporate a propagation model that includes scattering effects. The snow radar system has been simulated and the results have been compared with actual measurements to validate the simulations.

The system modeling effort involved in the simulation has been explained in Chapter 3. The effects of amplitude and phase errors of the source are modeled and included in the simulation. The system modeling is achieved either by measurement or by calibration.

The propagation model is detailed in Chapter 4. The determination of the reflection coefficient aids in determining the reflected component. The determination of the surface and volume scattering coefficient helps in determining the scattered component.

The actual method of simulation is explained in detail in Chapter 5. Results of the simulation and comparisons with actual measurements are shown in Chapter 6.

7.2 Recommendation for Future Work

7.2.1 Higher rate sampling to incorporate the IF section effects

The modeling of the system is curtailed by the fact that the IF section is not included. This is because the sampling rate of simulation is limited by the performance of the systems used in simulation. The use of much higher sampling rate will therefore improve the performance of the simulation.

7.2.2 Application of Deconvolution

The system data can be deconvolved from the actual radar data set to improve the results. The mathematical foundation for this is described in Appendix C.

7.2.3 Incorporation of Forward Scattering Models

The current simulations work with back scattering models to determine the scattering coefficients. This can be further extended to include forward scattering models which will give a more accurate representation of the snow radar because the snow radar uses a bistatic arrangement.

References

[1] Barber, D.G.; Fung, A.K.; Grenfell, T.C.; Nghiem, S.V.; Onstott, R.G.; Lytle, V.I.; Perovich, D.K.; Gow, A.J.; “The role of snow on microwave emission and scattering over first-year sea ice”; Geoscience and Remote Sensing, IEEE Transactions on , Volume: 36 , Issue: 5 , Sept. 1998; Pages:1750 - 1763

[2] Tjuatja, S.; Fung, A.K.; Bredow, J.; “A scattering model for snow-covered sea ice”; Geoscience and Remote Sensing, IEEE Transactions on , Volume: 30 , Issue: 4 , July 1992; Pages:804 - 810

[3] Yackel, J.J.; Barber, D.G.; “Validation of a snow water equivalence algorithm over landfast first-year sea ice using RADARSAT-1”; Geoscience and Remote Sensing Symposium, 2002. IGARSS '02. 2002 IEEE International , Volume: 1 , 24-28 June 2002; Pages:234 - 236 vol.1

[4] Iacozza, J. and D.G. Barber. 1999.; “An examination of the distribution of snow on sea ice “; Atmosphere-Ocean, vol. 37(1); 21-51.

[5] Hallikainen, M.; Ulaby, F.; Abdelrazik, M.; “Dielectric properties of snow in the 3 to 37 GHz range”; Antennas and Propagation, IEEE Transactions on, Volume: 34 , Issue: 11 , Nov 1986; Pages:1329 - 1340

[6] Markus, T. and D. Cavalieri. ; “Snow depth distribution over sea ice in the Southern Ocean from satellite passive microwave data. Antarctic Sea Ice: Physical Processes, Interactions, and Variability.”; Antarctic Research Series 1998. 74:19-39. Washington, DC, USA: American Geophysical Union.

[7] Hallikainen, M.;“Review of the Microwave Dielectric and Extinction Properties of Sea Ice and Snow”; Geoscience and Remote Sensing Symposium, 1992. IGARSS '92. International , Volume: 2 , 1992; Pages:961 - 965

[8] Gogineni S., Wong K., Krishnan S., Kanagaratnam P., Markus T. and Lytle V. ;“An Ultra-wideband Radar for Measurements of Snow Thickness over Sea Ice”; IGARSS'03; Toulouse, France, July 2003.

[9] Klauder J R, Price A C, Darlington S and Albersheim W J. ; “The theory and design of chirp radars”; Bell Syst. Tech. J. 1960 39; 745-808

[10] Griffiths, H.D. ; “The effect of phase and amplitude errors in FM radar”; High Time-Bandwidth Product Waveforms in Radar and Sonar, IEE Colloquium on , 1 May 1991; Pages:9/1 - 9/5

[11] Vijaya C Ramasami ;“A Simulation Package for Radar Systems”; RSL Report, University of Kansas

[12] Ulaby, F.T., R. K. Moore, and A. K. Fung; Microwave Remote Sensing: Active and Passive, vol. 2, Artech House, Norwood, MA, 1986.

[13] Fung, A. K.; Microwave Scattering and Emission Models and Their Applications, Artech House, Boston, 1994.

[14] Young-Soo Kim; Onstott, R.; Moore, R. ; “Effect of a snow cover on microwave backscatter from sea ice”; Oceanic Engineering, IEEE Journal of , Volume: 9 , Issue: 5 , Dec 1984; Pages:383 - 388

[15] Jordan E. C. and Balmain K. G.; Electromagnetic Waves and Radiating Systems. PrenticeHall, second ed., 1990.

[16] Jan Askne;(2002, Dec 17) Remote Sensing using Microwaves ; [online]
Available: "http://www.rss.chalmers.se/rsg/Education/RSUM/"

Appendix – A

MATLAB Code

The MALTAB code is in two groups .The Library code and the Simulation code. The Library code contains functions used by the Simulation Code.

Library Code

ampm_model.m

```
%% AMPM Model
%% Determines Amplitude and Phase Error Models
function [PMModel,AMModel] = ampm_model(NoArg);

%% Freq_Volt_Pow.txt is the Modeling Measurement
X=load('Freq_Volt_Pow.txt');
f=X(:,1);% - 20e6;
v=X(:,2);
PdBm = X(:,3);

%% AMPM.s2p charachterizes the AMPM Setup.
data = load('AMPM.s2p');
fr =  data(:,1);
s_db = data(:,4);
s_m = 10.^(s_db/20);
s_a = data(:,5);
[s_r,s_i] = pol2cart(s_a*pi/180,s_m);
S21 = s_r + j*s_i;

t_end = 10e-3;
t_0 = 0;

m=(t_end-t_0)/(v(end)-v(1));
t=t_0+m*(v-v(1));

f_t = polyfit(t,f,1);

ff_t = polyval(f_t,t);
ff_ideal = f(1) + (f(end) - f(1))/t_end*t;

f_end = 8e9;
f_0 = 2e9;
phi_ideal = 2*pi*(f_0*t + 0.5*(f_end - f_0)/t_end*t.^2);

phi_t = 2*pi*polyint(f_t);
phi = polyval(phi_t,t);

PMModel = phi_t;
```

```
Phse = interp1(f,phi,fr,'','extrap');

R = 50;
P = 10.^((PdBm-30)/10); % Power in watts
A = sqrt(P*R); % Amplitude in volts.

Amp = interp1(f,A,fr,'','extrap');
Amp_Setup = abs(Amp./S21);

%A_Setup = interp1(fr,Amp_Setup,f,'','extrap');
%A_DC = A_Setup - mean(A_Setup);

Amp_DC = Amp_Setup - mean(Amp_Setup);
Amp_Comps = dct(Amp_DC);

%% Compute DCT freq. axis;
dct_t_ax = linspace(0,.01,401);
dct_Fs = 1/(dct_t_ax(2)-dct_t_ax(1));
dct_ax = linspace(0,dct_Fs/2,length(Amp_DC));

Amp_Comps1 = zeros(size(Amp_Comps));
%Amp_Comps1(1:100) = Amp_Comps(1:100);
%Significant_Components_Index = [5,9,11,16,20,49,50,73];
Significant_Components_Index = find(abs(Amp_Comps) > 0.1);
Amp_Comps1(Significant_Components_Index) =
Amp_Comps(Significant_Components_Index);
Amp_DC1 = idct(Amp_Comps1);

AMModel = Amp_Comps1;
```

compute_Dprofile.m

```
% -------------------------------------------------------------------
--------- %
% Compute Distance Axis
% -------------------------
% Computed the Distance Axis
%
% Paramters :
% -----------
% radarParams -> The type of the radar used affects the velocity
profile.
% erLayer -> Dielectric Information too affects the velocity
profile.
% Depth -> Depth profile of the geo-physical data.
%
% -------------------------------------------------------------------
--------- %

function [dProfile] = ComputeDProfile(radarParams, erLayer,
Depth,fAxis)

Epsilon0 = 1e-9/(36*pi);
```

```
Mu0 = (4e-7)*pi;

nLayers = length(erLayer(1,:));

% Sampling Parameters ...
nT = radarParams(2);
tS = radarParams(3);
fR = 1/(nT*tS);
f0 = radarParams(5);
f1 = radarParams(7);
BW = f1-f0;

% Compute the Speed of Light in each layer.
Vel = real(1./sqrt(mean(erLayer)*Epsilon0*Mu0));
Vel = [Vel,Vel(end)];

% Generate Cumulative Depths!!
CumDepth = [cumsum(Depth);Inf];

% Computing the Distance Axis.
disp('Compute Distance Axis...');
indx1 = 1;
d_cur = 0;

dProfile = zeros(size(fAxis));
fStep = fAxis(2)-fAxis(1);

for indx2 = 2:length(fAxis)
      d_cur = d_cur + Vel(indx1).*fStep/(2*fR*BW);
      if(d_cur > CumDepth(indx1))
            indx1 = indx1 + 1;
            if (indx1 > nLayers)
                  break;
            end;
      end;
      dProfile(indx2) = d_cur;
end;

dProfile(indx2:end) = d_cur + Vel(indx1).*(fAxis(indx2:end)-
fAxis(indx2-1))./(2*fR*BW);
return;
```

ComputeGamma.m

```
%% Determine Gamma
function [Gamm_long] = ComputeGamma(radarParams,
erLayer,Depth,tAxis,Wv_Len,Sigh)

erLayer_m = mean(erLayer);
Vel_m = 3e8./sqrt(erLayer_m);

Range = cumsum(Depth);
Dist = Range(1:end-1);
```

```
for index_var = 1:length(Depth)-1
      Gamm(index_var) = (sqrt(erLayer_m(index_var+1)) -
sqrt(erLayer_m(index_var)))/...
                        (sqrt(erLayer_m(index_var+1)) +
sqrt(erLayer_m(index_var)));
end;

depths = ComputeDvsT(radarParams, erLayer, Depth);

Wv_Len_m = Wv_Len ./ sqrt(erLayer_m);
K = 2*pi./Wv_Len_m(1:end-1);
Gamm = Gamm.*exp(-4*K.^2.*Sigh.^2);

% Interpolate Gamma to all values!
Gamm_long = zeros(size(tAxis));
Gamm_long(max(find(depths < Dist(1)))) = Gamm(1);
Gamm_long(max(find(depths < Dist(2)))) = Gamm(2);

return;
```

PSFromCalib.m

```
%% PSFromCalib.m
%% Takes in the calibration file
%% And Generates Transfer Function
function [PSF] = ComputePSFromCalib(nT,Fname1);

Num_Fft=2^19;
BW = 6e9;
T_swp = 10e-3;
N_swp = 50000;

fid1 = fopen(Fname1,'r');
%% Read in Headers
header1 = fread(fid1,3,'float64')';
header1 = [header1 fread(fid1,3,'int32')'];
header1 = [header1 fread(fid1,2,'float64')'];
numsam1 = header1(4);
Fs1 = header1(3); %Sampling Rate
F_ax1=-Fs1/2:Fs1/Num_Fft:Fs1/2-Fs1/Num_Fft;
F_ax1_p=F_ax1(end/2+1:end);
D_ax1=F_ax1_p*3e8*T_swp/(2*BW);
Ts1 = 1/Fs1;

fAxis = linspace(2e9,8e9,nT);

Cur_Pos1 = ftell(fid1);
No_AScopes1 = length(fread(fid1,inf,'int16'))/64006;
fseek(fid1,Cur_Pos1,-1);

%% AScope_Num = ceil(No_AScopes1);
AScope_Num = 5;
```

```
for i=1:No_AScopes1
      Data1 = fread(fid1,numsam1,'int16');
      Time1 = fread(fid1,3,'int32');
      Upsweep1 = Data1(1:N_swp);
      Data_Arr1(:,i) = Upsweep1;
end

fclose(fid1);

Calib_data = Data_Arr1(:,AScope_Num)/65536; % Normalize the Data
Calib_data = resample(Calib_data,nT,N_swp);

Calib_data_f = fftshift(fft(Calib_data',Num_Fft));
Calib_indx = find(Calib_data_f(end/2+1:end) ==
max(Calib_data_f(end/2+1:end)));
F_calib = F_ax1_p(Calib_indx);

Filter_Wn = [Calib_indx-50 Calib_indx+50]/Num_Fft;
[Filt_B,Filt_A] = butter(2,Filter_Wn);
[Filt_H,Filt_W] = freqz(Filt_B,Filt_A,Num_Fft);

Calib_data_filt_f = (ifftshift(Calib_data_f).*abs(Filt_H)')';

Calib_data_filt = real(ifft(Calib_data_filt_f'));
Calib_data_filt_m = hilbert(Calib_data_filt(1:nT));

System_m = abs(Calib_data_filt_m);
System_p = abs(unwrap(angle(Calib_data_filt_m))) -
2*pi*F_calib*T_swp*(0:nT-1)/nT;
System_func1 = System_m.*exp(j*System_p);

System_m = SysDivMag(System_m',2e9,8e9,'IdMag_50k.txt');
System_func2 = System_m'.*exp(j*System_p);

PSF = System_func1;

return;
```

PSFromMeas.m

```
%% Compute Point Spread Function H(f) from Measurement Using NA

%% Equivalent to Simulation with an ideal sample delay
%% with system effects included using NA measured files.
% Load Models

% Sample Radar Parameters.
function [PSF] = ComputePSFromMeas(radarParams);

% Sampling Parameters ...
nT = radarParams(2);
tS = radarParams(3);
fR = 1/(nT*tS);
tAxis = 0:tS:(nT-1)*tS;
```

```
f0 = radarParams(5);
f1 = radarParams(7);
BW = f1-f0;
fc = (f0+f1)/2; % Center Frequency.

fAxis = (0:fR:(nT/2)*fR)';
freq_Axis = [flipud(-fAxis(2:end));fAxis(1:end-1)];

[Wv] = GenerateChirpWaveform(radarParams);
Tx_Wv = SysPass(Wv,nT,tS,'Tx.txt');
Rx_LO = SysPass(Wv,nT,tS,'TxLo.txt');
Rx = [zeros(2000,1);Tx_Wv(1:end-2000)];
Rx_Wv = SysPass(Rx,nT,tS,'Rx.txt');
Mix_Wv = Rx_Wv.*Rx_LO;
Rx_Mix = Mix_Wv;

Num_Fft = 2^21;
freq_Axis_range = max(freq_Axis) +fR - min(freq_Axis);
freq_Axis_res = freq_Axis_range/Num_Fft;
freq_Axis_fft = [-freq_Axis_range/2:freq_Axis_res:freq_Axis_range/2-
freq_Axis_res];
freq_Axis_pos = freq_Axis_fft(end/2+1:end);

Range_Prof = fftshift(fft(Rx_Mix,Num_Fft));

Calib_indx = find(Range_Prof(end/2+1:end) ==
max(Range_Prof(end/2+1:end)));
F_calib = freq_Axis_pos(Calib_indx);

Filter_Wn = [Calib_indx-100 Calib_indx+100]/Num_Fft;
[Filt_B,Filt_A] = butter(2,Filter_Wn);
[Filt_H,Filt_W] = freqz(Filt_B,Filt_A,Num_Fft);

Calib_data_filt_f = ifftshift(Range_Prof).*abs(Filt_H);

Calib_data_filt = real(ifft(Calib_data_filt_f'));
Calib_data_filt_m = hilbert(Calib_data_filt(end:-1:end-nT+1));

System_m = abs(Calib_data_filt_m);
System_p = abs(unwrap(angle(Calib_data_filt_m))) -
2*pi*F_calib*tAxis;
PSF = System_m.*exp(j*System_p);

%% %% Plot Tranfer Function
%% figure;
%% plot(linspace(2e9,8e9,nT),20*log10(System_m));
%% xlabel('Freq -->');
%% ylabel('Magnitde (dB)-->');
%% grid;
%% title('Point Spread Function from Measuremnt using NA');

return;
```

GenerateAMPMChirpWaveform.m

```
% ------------------------------------------------------------------
--------- %
% Generate AMPM Chirp Waveform
% ========================
%
% Returns :
% ---------
% Waveform -> Chirp Waveform (time-domain).
%
% ------------------------------------------------------------------
--------- %

function [Waveform] =
GenerateAMPMChirpWaveform(radarParams,PMPoly,AMDCT)

% Load Paramters.
nT = radarParams(2); % # Time Points.
tS = radarParams(3); % Time Step.
t0 = radarParams(4); % Time Offset (not user in Ver 1.0).
f0 = radarParams(5); % Initial Frequency.
tD_act = radarParams(6); % Time Duration (not used in Ver 1.0).
tD_sim = nT*tS;        % .. Set to nT x tS.
f1 = radarParams(7); % Final Frequency.
fc = (f0 + f1)/2;    % Center Frequency.
B = (f1 - f0);         % 3 dB bandwidth.

% Generate a time Scale.
t = (0:tS:(nT-1)*tS)'; % Time Vector.

% Generate the Phase
PMPoly_Sim = PMPoly.*(tD_act/tD_sim).^(length(PMPoly)-2:-1:-1);
Phse = polyval(PMPoly_Sim,t);

% Debug
% k = B/(nT*tS); % Sweep Rate.
% Phse1 = (pi*k*t.^2+2*pi*f0*t);
% plot(t,Phse,t,Phse1);

% Generate the Waveform
Waveform = real(exp(j*Phse));
Waveform =(1+idct(AMDCT,length(Waveform))).*Waveform;
```

ComputeDvsT.m

```
function [dProf] = ComputeDvsT(radarParams, erLayer, Depth)

Epsilon0 = 1e-9/(36*pi);
Mu0 = (4e-7)*pi;

nLayers = length(erLayer(1,:));

% Sampling Parameters ...
nT = radarParams(2);
tS = radarParams(3);
```

```
tAxis = 0:tS:(nT-1)*tS;

% Compute the Speed of Light in each layer.
Vel = real(1./sqrt(mean(erLayer)*Epsilon0*Mu0));
Vel = [Vel,Vel(end)];

% Generate Cumulative Depths!!
CumDepth = [cumsum(Depth);Inf];

% Computing the Distance Axis.
indx1 = 1;
d_cur = 0;

dProf = zeros(size(tAxis));

for indx2 = 2:nT
      d_cur = d_cur + Vel(indx1).*tS/2;
      if(d_cur > CumDepth(indx1))
            indx1 = indx1 + 1;
            if (indx1 > nLayers)
                  break;
            end;
      end;
      dProf(indx2) = d_cur;
end;

dProf(indx2:end) = d_cur + Vel(indx1).*(tAxis(indx2:end)-
tAxis(indx2-1))/2;
return;
```

ComputeSigma0.m

```
function [Sigma0,Distance] = ComputeSigma0(Freq, Ht, Fs, R_Res, ...
                      Sig_h_1, Sig_h_2, L_c_1, L_c_2,geoData)

addpath('../Library/PermCalc');

% Omni-Present Definitions !!
Epsilon0 = 1e-9/(36*pi);
Mu0 = (4e-7)*pi;

Angle_Rad = [0:5:60]'*pi/180;
Angle_Rad_2s = [-60:5:60]'*pi/180;

% To Apply Antenna beam pattern
Gain_2s = gausswin(length(Angle_Rad_2s),7.17); % Coeff adjusted to
get 10deg BmW

% Range from Angle
Layer_Ht = sum(geoData(2:end-1,1)); % Exclude Air and Sea Ice to get
snow height.
Ranges_1 = Ht./cos(Angle_Rad);
```

```
Cell_Num_1 = floor((Ranges_1 - Ht)/R_Res);
Ranges_2 = (Ht+Layer_Ht)./cos(Angle_Rad);
Cell_Num_2 = floor((Ranges_2 - Ht - Layer_Ht)/R_Res);

% Define
Dens_Ic = 918; % Ice Densit
Er_Air = 1;
Er_Ice = 3.15 - 0.001j;
K = (Er_Ice - Er_Air)/(Er_Ice + Er_Air); %Constant

% Compute WvNum
WvLen = 3e8/Freq;
WvNum = 2*pi./WvLen;

% Extract Snow Info
Temp_Sn = geoData(2,3);
Dens_Sn = geoData(2,4);
Radi_Sn = 0.5e-3; %% Assume Now, Later include in geoData.
Er_Layer = ComputeDrySnowEr(Dens_Sn,Temp_Sn,Freq);

% Number of Scatterers
Vol_Frct = Dens_Sn / Dens_Ic;
Vol_Snow = 4*pi*Radi_Sn^3/3;
Num_Scat = Vol_Frct/Vol_Snow;
Eq_Radi_Sn = (1.2+Vol_Frct-2*Vol_Frct^2)*Radi_Sn;

% Extract Sea Ice Info
Temp_SI = geoData(end,3);
Sali_SI = geoData(end,4);
Dens_SI = geoData(end,5);

% Mie Parameter
Chi = 2*pi*Radi_Sn*sqrt(Er_Air)./WvLen;

% Initialize Return Variable 1
Distance = [min(Ranges_1):0.01:max(Ranges_2)];

VScatTerm_1 = zeros(size(Angle_Rad));
VScatTerm_2 = zeros(size(Angle_Rad));
Area_1 = zeros(size(Angle_Rad));
Area_2 = zeros(size(Angle_Rad));
Sigma_0_Sn = zeros(size(Angle_Rad));
Sigma_0_Ic = zeros(size(Angle_Rad));
Sigma_0_ss = zeros(size(Angle_Rad));
Sigma_0_sv = zeros(size(Angle_Rad));
Sigma_0_is = zeros(size(Angle_Rad));

% Some Calcs out of the Loop
Sigma_B = 8*pi*WvNum^4*Eq_Radi_Sn^6*abs(K)^2/3;
Sigma_V = Num_Scat * Sigma_B;

Qs = 2*WvLen^2*Chi^6*abs(K)^2/(3*pi);
Qa = WvLen^2*Chi^3*imag(-K)/(pi);
```

```
ks = Num_Scat * Qs;
kai = Num_Scat * Qa;
kab = 2*WvNum*(1-Vol_Frct)*imag(sqrt(Er_Air));

ka = kai + kab;
ke = ks + ka;

for index_var = 1:length(Angle_Rad)

    Sigma_0_ss(index_var) = 
ReturnScatterCoeff(Angle_Rad(index_var),Freq,Sig_h_1,L_c_1,1,Er_Laye
r);

    Angle_Vol = 
real(asin(sin(Angle_Rad(index_var))/sqrt(Er_Layer)));

    Er_Final = 
ComputeSeaIceEr(Dens_SI,Sali_SI,Temp_SI,rad2deg(Angle_Vol),Freq,1);

    Loss = exp(ke*Layer_Ht*sec(Angle_Vol));

    Sigma_0_sv(index_var) = Sigma_V * cos(Angle_Vol) * (1 - 
1/Loss^2) / (2*ke);

    Refl = (sqrt(Er_Air)*cos(Angle_Rad(index_var))-
sqrt(Er_Layer)*cos(Angle_Vol))/...

(sqrt(Er_Air)*cos(Angle_Rad(index_var))+sqrt(Er_Layer)*cos(Angle_Vol
));

    Transm_s = 1-abs(Refl)^2;

    Sigma_0_is(index_var) = 
ReturnScatterCoeff(Angle_Vol,Freq,Sig_h_2,L_c_2,Er_Layer,Er_Final);

    Sigma_0_Sn(index_var) = Sigma_0_ss(index_var) + 
Transm_s*(Sigma_0_sv(index_var));
    Sigma_0_Ic(index_var) = Transm_s*Sigma_0_is(index_var)/Loss;

    if (Cell_Num_1(index_var) == 0)
        Radius_2 = (Ht + R_Res)^2 - Ht^2;
        Area_1(index_var) = pi * Radius_2;
        VScatTerm_1(index_var) = 
Sigma_0_Sn(index_var)*Area_1(index_var)/((4*pi)^3*Ranges_1(index_var
)^4);
    else
        Radius_1_2 =  (Ht + Cell_Num_1(index_var)*R_Res)^2 - Ht^2;
        Radius_2_2 =  (Ht + (Cell_Num_1(index_var) + 1)*R_Res)^2 - 
Ht^2;
        Area_1(index_var) = pi * (Radius_2_2 - Radius_1_2);
        VScatTerm_1(index_var) = 
Sigma_0_Sn(index_var)*Area_1(index_var)/((4*pi)^3*Ranges_1(index_var
)^4);
    end;
    if (Cell_Num_2(index_var) == 0)
        Radius_2 = (Ht+Layer_Ht + R_Res)^2 - Ht^2;
        Area_2(index_var) = pi * Radius_2;
        VScatTerm_2(index_var) =
Sigma_0_Ic(index_var)*Area_2(index_var)/((4*pi)^3*Ranges_2(index_var
)^4);
    else
        Radius_1_2 =  (Ht+Layer_Ht + Cell_Num_2(index_var)*R_Res)^2
- Ht^2;
        Radius_2_2 =  (Ht+Layer_Ht + (Cell_Num_2(index_var) +
1)*R_Res)^2 - Ht^2;
        Area_2(index_var) = pi * (Radius_2_2 - Radius_1_2);
        VScatTerm_2(index_var) =
Sigma_0_Ic(index_var)*Area_2(index_var)/((4*pi)^3*Ranges_2(index_var
)^4);
    end;

end;

% figure;
% plot(Angle_Rad*180/pi,10*log10(Sigma_0_ss),'g*-
',Angle_Rad*180/pi,10*log10(Sigma_0_sv),'rs-
',Angle_Rad*180/pi,10*log10(Sigma_0_is),'cd-');
% title('Scattering Coefficient vs. Angle');
% xlabel('\theta');
% ylabel('\sigma^0');
% legend('\sigma^0_s_s','\sigma^0_s_v','\sigma^0_i_s');
% grid;

VScatTerm_2s_1 = [flipud(VScatTerm_1(2:end));VScatTerm_1];
VScatTerm_2s_2 = [flipud(VScatTerm_2(2:end));VScatTerm_2];

% Convolve with gain
VScatTerm_Gain_1 = flipud(fftfilt(VScatTerm_2s_1,Gain_2s));
VScatTerm_Gain_2 = flipud(fftfilt(VScatTerm_2s_2,Gain_2s));

VScatTerm_Gain_1 = VScatTerm_Gain_1(1:length(Angle_Rad));
VScatTerm_Gain_2 = VScatTerm_Gain_2(1:length(Angle_Rad));

%% % Normalization : Match Maxmimum After Gain Convolution!
%% VScatTerm_Gain_1 =
VScatTerm_Gain_1/max(VScatTerm_Gain_1)*max(VScatTerm_2s_1);
%% VScatTerm_Gain_2 =
VScatTerm_Gain_2/max(VScatTerm_Gain_2)*max(VScatTerm_2s_2);

VScatTerm_Gain =
interp1(Ranges_1,VScatTerm_Gain_1,Distance,'',0)+...
                 interp1(Ranges_2,VScatTerm_Gain_2,Distance,'',0);

% Include Lamda Square Term
VScatTerm = VScatTerm_Gain.*WvLen^2;
Sigma0 = sqrt(VScatTerm);
```

```
return;
```

Simulation Code

Sim_AllRuff.m

```
clear;clc;close all;

addpath('../Data');
addpath('../Library');

radarParams = [1, 50000, 5e-11, 0, 2e9, 10e-3, 8e9];
win_type = 'hanning';

geoData = load('air_snow_seaice.csv');
Sig_h_1 = 0.009; % Standard deviation of height (Snow)
L_c_1 = 0.5; % Correlation Length (Snow)
Sig_h_2 = 0.007; % Standard deviation of height (Sea Ice) (2 to 5
cm)
L_c_2 = 0.02; % Correlation Length (Sea Ice) (40 cm) (2 to 90cm)

% Sampling Parameters ...
nT = radarParams(2);
tS = radarParams(3);
fS = 1/tS;
fR = 1/(nT*tS);
tAxis = 0:tS:(nT-1)*tS;
f0 = radarParams(5);
f1 = radarParams(7);
BW = f1-f0;
fc = (f0+f1)/2; % Center Frequency.
fAxis = (0:fR:nT/2*fR-fR)';

% Geo-Physical Model: Conversion to Electrical Model
disp('Converting Geo-Physical Model to Electrical Model ...');
[erLayer, Depth] = ComputeEr_General(geoData, nT, tS);

% Generating the Chirp Waveform.
disp('Generating Chirp Waveform ...');
[PMModel,AMModel] = Ampm_Model('');
[Wv] = GenerateAMPMChirpWaveform(radarParams,PMModel,AMModel);

% Compute Tx WvLen , and estimate initial order of Rng
f_t = linspace(f0,f1,length(tAxis))';
lam_t = (3e8/mean(f_t));
Rng = Depth(1); % Bcuz we are using equivalent gamma @ depth of
first layer.

disp('Computing Return Response ...');
R_Res = 3e8/(2*BW); % Range Resolution
Freq = 5e9; % Frequency to calculate scattering Power
```

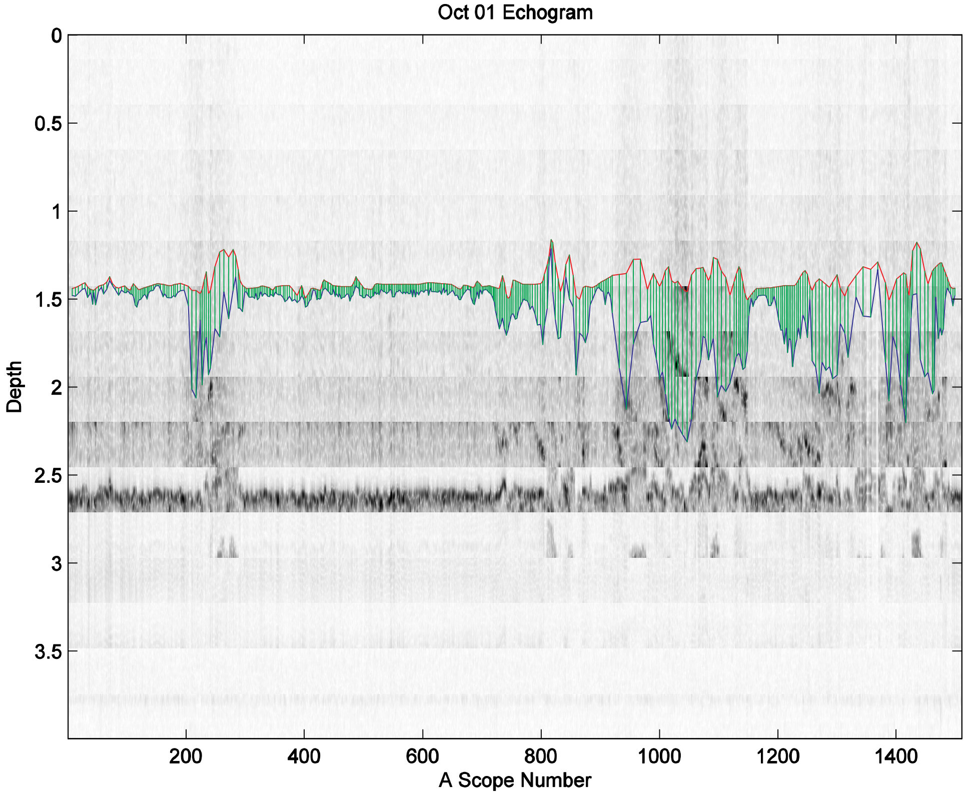
```
Ht = Depth(1); % Ht of First interface.Depth
num_fft = 2^21;

disp('Compute Point Spread Function...');
%% Either PSF Can be Used!
%% PSF = ComputePSFromMeas(radarParams)';
PSF = ComputePSFromCalib(nT,'/rsl1/antarctica2003/sep28/snow2.1')';

disp('Computing Return due to Reflection...');
Gamma_Sc = ComputeGamma(radarParams,
erLayer,Depth,tAxis,lam_t,[Sig_h_1,Sig_h_2])';
Rx_Sc = real(ifft(fft(Gamma_Sc).*fft(Wv)));
Rx_Wv = Rx_Sc.*lam_t./(2*Rng)/(4*pi);
IF1 = Wv.*Rx_Wv.*PSF;
IF1 = ApplyTimeWindow(IF1,win_type);
rProfile1 = 20*log10(abs(fftshift(fft(IF1,num_fft))/(length(Wv))));
rProfile1 = rProfile1(end/2+1:end);
power1 = 10.^(rProfile1/10);

disp('Computing Return due to Scattering...');
[Sigma0vsD, Dists] = ComputeSigma0(Freq, Ht, 1/tS, R_Res,Sig_h_1,
Sig_h_2, L_c_1, L_c_2,geoData);
DvsT = ComputeDvsT(radarParams, erLayer, Depth);
Sigma0 = interp1(Dists,Sigma0vsD,DvsT,'',0)';
Sigma_1 = zeros(size(Sigma0));
Sigma_1(1:4:end) = Sigma0(1:4:end);

Num_Avg = 20;
rProfile2_avg = zeros(num_fft/2,1);
for indx_var = 1:Num_Avg

      Sigma_ph = Sigma_1 .*exp(j*2*pi*rand(size(Sigma_1))); %% Phase
Fluctuations
      Rx_s1 = real(ifft(fft(Sigma_ph).*fft(Wv)));
      IF2 = Wv.*Rx_s1.*PSF;
      IF2 = ApplyTimeWindow(IF2,win_type);
      rProfile2 =
20*log10(abs(fftshift(fft(IF2,num_fft))/(length(Wv))));
      % rProfile2 = rProfile2(end/2+1:end);
      rProfile2_avg = rProfile2_avg +
rProfile2(end/2+1:end)/Num_Avg;
      if mod(indx_var,10) == 0
            disp(indx_var);
      end;

end;

power2 = 10.^(rProfile2_avg/10);

% Computing the Distance Profile.
fr_Axis = 0:fS/num_fft:fS/2-fS/num_fft;
dProfile = ComputeDProfile(radarParams, erLayer, Depth, fr_Axis);

Noise_Dist = [0:R_Res:max(dProfile)];
```

```
Noise = mean(power1)*0.01*randn(size(Noise_Dist)); % -60dB Noise @
every range cell.
Noise_dProfile = interp1(Noise_Dist,Noise,dProfile)';

power1 = power1+Noise_dProfile; % Noise is added here ALONE!
rProfile1 = 10*log10(abs(power1));
power3 = power1+power2;
rProfile3 = 10*log10(abs(power3));

rProfile1 = rProfile1 - max(rProfile3);
rProfile2_avg = rProfile2_avg - max(rProfile3);
rProfile3 = rProfile3  - max(rProfile3);

% Profile Vs. Distance.
figure;
plot(dProfile,rProfile3-max(rProfile3),'r');
xlabel('Distance (m) -->');
ylabel('Magnitude (dB) -->');
title('Simulation of Return from Snow Over Sea Ice - Rough
Surfaces');
axis([geoData(1,1)-1,geoData(1,1)+1.5,-100,0]);
grid on;
```

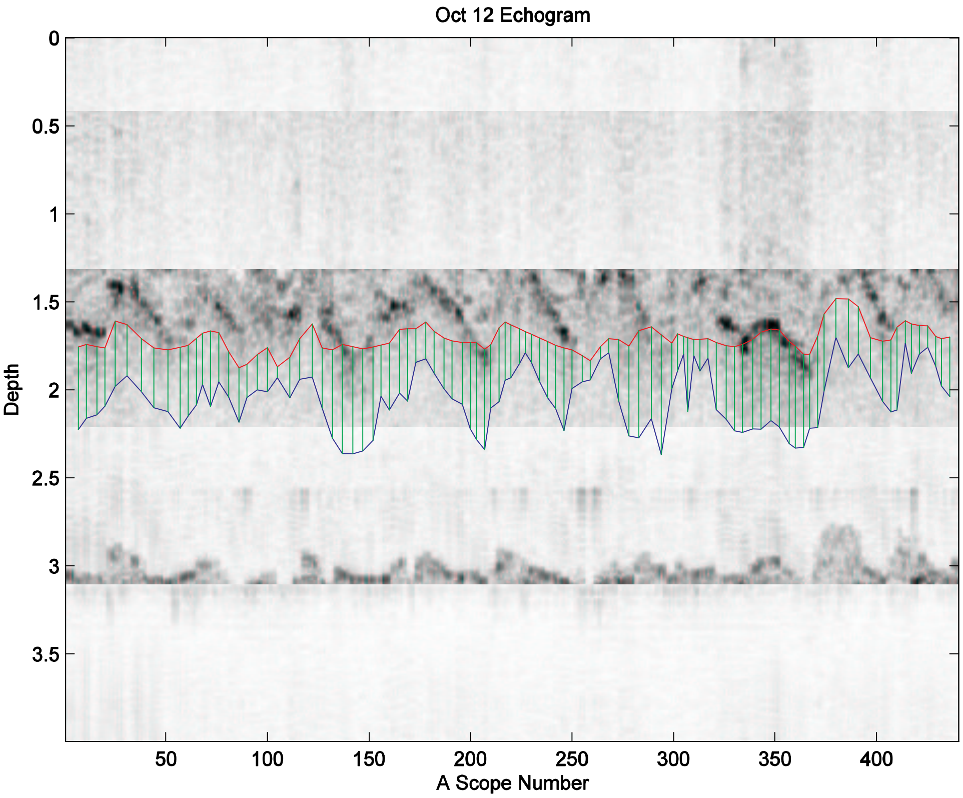
Appendix –B

Antarctica Field Experiment Results

Field trials were conducted using the snow radar experiment and the results from some of the days are shown below. The echogram is shown with additional red, blue and green lines. The red line is the snow top, the blue line is the snow bottom interpolated between actual depth measurements that are shown in green.



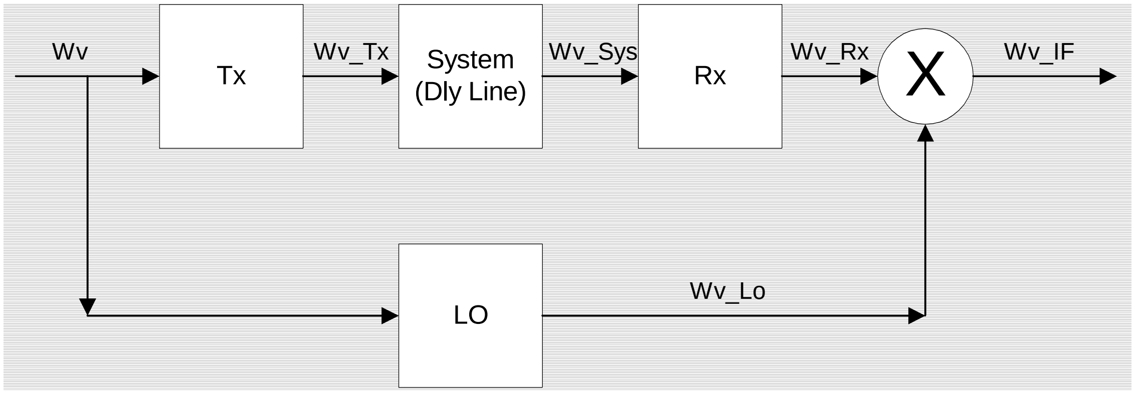
This echogram is from the data collected on Oct 01, 2003. The next echogram shows data collected on Oct 12, 2003.


Oct 12 Echogram
Depth
0
0.5
1
1.5
2
2.5
3
3.5
50
100
150
200
250
300
350
400
A Scope Number

Appendix – C

Deconvolution Technique

The Model used in the deconvolution is shown here.



The Math involved is given below. Assume that Wv is the Input Chirp

$$Wv = \cos\left(2\pi\left(f_0 t + \alpha * t^2/2\right)\right)$$

Then the output of the transmitter is given by

$$Wv\_Tx = |H_{Tx}(t)| \cos\left(2\pi\left(f_0 t + \alpha * t^2/2\right) + \phi_{Tx}(t)\right)$$

Note that this is because the FMCW radar has a direct relationship between frequency and time. The system is characterized by a time delay τ and a reflection coefficient of Γ. The output of the system is therefore given by

$$Wv\_Sys = |H_{Tx}(t)||\Gamma| \cos\begin{pmatrix} 2\pi\left(f_0 t + \alpha * t^2/2\right) + \phi_{Tx}(t) + \phi_{\Gamma}(t) + \\ \left\{2\pi * \alpha * t * \tau - 2\pi * f_0 * \tau - \pi * \alpha * \tau^2\right\} \end{pmatrix}$$

The Received signal is therefore

$$Wv\_Rx = |H_{Tx}(t)||\Gamma||H_{Rx}(t)| \cos\begin{pmatrix} 2\pi\left(f_0 t + \alpha * t^2/2\right) + \phi_{Tx}(t) + \phi_{Rx}(t) + \phi_{\Gamma}(t) + \\ \left\{2\pi * \alpha * t * \tau - 2\pi * f_0 * \tau - \pi * \alpha * \tau^2\right\} \end{pmatrix}$$

The local oscillator signal is

$$Wv\_Lo = \left|H_{Lo}(t)\right| \cos\left(2\pi\left(f_0 t + \alpha * t^2/2\right) + \phi_{Lo}(t)\right)$$

Let us make the following substitutions

$$\prod \left|H_{xx}(t)\right| = |H|$$

$$\sum \phi_{xx}(t) = \phi_H$$

$$\alpha * \tau = f_b$$

Case 1

Assume that LO has no effect i.e. Wv_Lo =Wv. Then the output of the mixer after low pass filtering is

$$Wv\_IF = \left(\tfrac{1}{2}\right)|H||\Gamma| \cos\left(2\pi * f_b t + 2\pi * f_0 * \tau - \pi * f_b * \tau - \phi_H(t) - \phi_\Gamma(t)\right)$$

Case 2

Assume that LO is not ignored.

$$Wv\_IF = \left(\tfrac{1}{2}\right)|H||\Gamma||H_{LO}| \cos\left(2\pi * f_b t + 2\pi * f_0 * \tau - \pi * f_b * \tau - \phi_H(t) - \phi_\Gamma(t) + \phi_{LO}(t)\right)$$

Deconvolution

As the frequency of FMCW radar is a function of time, we can find the transfer function as a function of time. In Case I the deconvolution is to be attempted by dividing the received signal by H interpolated as a function of time. In Case II the deconvolution is to be attempted by dividing by H and also H_{LO}. What has to be taken note is that, because the sign of ϕ_{LO} is opposite to the one of H, we need to divide the conjugate of the Wv_IF by H_{LO} interpolated as a function of time.