Synthetic Aperture Radar (SAR) Raw Signal Simulation

A Thesis Presented to the Faculty of the
California Polytechnic State University
San Luis Obispo, California

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Electrical Engineering

By
Amin Shoalehvar
June 2012

Supported by Raytheon Space and Airborne Systems Division
Committee Membership

Title: Synthetic Aperture Radar (SAR) Raw Signal Simulation

Author: Amin Shoalehvar

Date Submitted: June 2012

Committee Chair: Dr. John Saghri

Committee Member: Dr. Jane Zhang

Committee Member: Dr. Xiao-Hua Yu
Abstract

Synthetic Aperture Radar (SAR) Raw Signal Simulation

Author: Amin Shoalehvar

Synthetic aperture radar (SAR) raw signal simulation is a useful tool for SAR system design, mission planning, processing algorithm testing, and inversion algorithm design. This thesis explores a SAR raw signal simulation. The raw signal simulation is the simulated received signal before any processing with exception of the down-converter. The simulation plays a significant role in studies concerning noise and clutter rejection and contributes toward optimizing SAR system parameters.

To simulate SAR raw data, a Chirp Scaling (CS) method is used. This method [3] first stretches the input surface reflectivity of the target in the azimuth and range direction respectively. Then it derives the raw data by inverse equalizing the signal based on CS principle. This method avoids the time-domain integral operation and improves the computational efficiency. A simulation diagram, calculation and systematic process are proposed in this thesis. Finally, simulation results are presented to verify the accuracy of calculations and the efficiency of the process.
Acknowledgements

I would like to thank the people and organizations that made this possible: Cal Poly University, Raytheon Space and Airborne Systems and Dr. Saghri for help, support guidance and sponsorship of this project. I would like to thank my wife and my family for strength, energy and love they provide me. Finally, I would like to thank Dr. Jane Zhang and Dr. Xiao-Hua Yu for volunteering their time to be part of the thesis committee.
# Table of Contents

Table of Figures.............................................................................................................................. vii

Introduction ........................................................................................................................................ 1

Prior SAR Progress .......................................................................................................................... 5

SAR Signal ........................................................................................................................................ 7

SAR Raw Signal Module and Analysis in Frequency Domain ............................................................. 11

Simulation Program ........................................................................................................................ 17

Simulation Results .......................................................................................................................... 19

Conclusion ........................................................................................................................................ 23

Future Work ..................................................................................................................................... 25

References ....................................................................................................................................... 26

Appendix A: SAR Simulation Code in MATLAB ................................................................................ 27

Appendix B: Fourier and Inverse Fourier Transform Coding in MATLAB used in SAR Raw data
Simulation Process ........................................................................................................................... 35
# Table of Figures

<table>
<thead>
<tr>
<th>Figure 1:</th>
<th>The surface of Venus as imaged by the Magellan Probe using SAR $^1$</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2:</td>
<td>SAR Geometry ($\theta=$Look Angle, $r=$closest distance from P to flight path, $R=$ distance from the P to the radar)</td>
<td>7</td>
</tr>
<tr>
<td>Figure 3:</td>
<td>Simulation diagram of raw data based on CS principle</td>
<td>16</td>
</tr>
<tr>
<td>Figure 4:</td>
<td>Cuts of the phase curves comparison in azimuth direction and, average error of the phase curve</td>
<td>20</td>
</tr>
<tr>
<td>Figure 5:</td>
<td>Cuts of the phase curves comparison in range direction, and the average error of the phase curve</td>
<td>21</td>
</tr>
<tr>
<td>Figure 6:</td>
<td>(a) Surface Reflectivity of Target (CalPoly Map) (b) Amplitude of the raw data of extended scenes (c) Simulated SAR image of raw data</td>
<td>22</td>
</tr>
<tr>
<td>Figure 7:</td>
<td>Surface reflectivity of target, amplitude of raw data and focused image</td>
<td>24</td>
</tr>
</tbody>
</table>
Introduction

Synthetic Aperture Radar (SAR) is one of the advanced techniques of radar imaging that was developed in 1950s, and the technology was released to the civilian communities in 1970s. SAR is usually implemented by mounting a single beam-forming antenna on a moving platform such as an aircraft or spacecraft, from which a target scene is repeatedly illuminated with pulses of radio waves at wavelengths anywhere from a meter down to millimeters. The many echo waveforms received successively at the different antenna positions are detected and stored and then post-processed together to resolve elements in an image of the target region. SAR is applied widely in many areas such as military, ocean and agriculture. Software simulation, which produces simulative echo and images, is a very important for many purposes like testing different image formation algorithms, studying the interaction of electromagnetic waves with a scene that is being imaged, testing and validating of different system design parameters and economical method in research of SAR systems.

At the start of the project in September 2007, the Synthetic Aperture Radar (SAR) simulation from previous Master’s theses was capable of simulating and imaging point targets in a two dimensional plane with limited mobility. Through the course of this project, the focus was on improving the computational efficiency and accuracy of the SAR simulation so that it could be applied to more complex, time-sensitive two-dimensional targets.
Figure 1: The surface of Venus as imaged by the Magellan Probe using SAR

All these simulations can be roughly categorized into three groups. The first group works in time domain (previous thesis). It creates a high-precision raw data set. However, this method has low computational efficiency. The second group concentrates on the raw data generation of extended scenes that operates in the two-dimensional frequency domain. The third group can simulate SAR raw data by using inverse imaging algorithm in hybrid domain. The limitation for this method is the need of the real SAR image as input, so it lacks the flexibility to simulate SAR raw data of the artificial targets.

This paper proposes the work that has been done in reference [3]. Future work on this ongoing project will include an algorithm to calculate line of sight limitations of point targets. Another field is to optimize the process of generating the radar information, so the more complex and realistic targets can be simulated. In addition, a motion compensation method can introduce a real arbitrary trajectory deviation error into the raw data simulation.

The readers will find most of the technical information needed for concise understanding and designing of high quality and high throughput SAR processors in Reference [1].

1. This image or video was catalogued by Jet Propulsion Lab of the United States National Aeronautics and Space Administration (NASA) under Photo ID: PIA00104.
This book [1] is divided into three parts. The first part contains information related to signal processing fundamentals, pulse compression of linear FM signal, synthetic aperture concepts, and SAR signal properties. The second part contains information regarding Range Doppler Algorithm (RDA), the Chirp Scaling Algorithm (CSA), the Omega-K Algorithm, the SPECAN Algorithm, and processing Scan SAR data. The third part contains Doppler Centroid estimation, and azimuth FM rate estimation. The second reference [2] talks about the basic principles behind one-dimensional range and cross-range imaging via application of radar bandwidth and synthetic aperture respectively. It also discusses the special role of the radar radiation pattern and its Fourier properties using the wavefront reconstruction theory. Then it provides the principle behind SAR imaging, and discusses system modeling and imaging for squint spotlight SAR and stripmap SAR. Finally, it discusses the SAR geometry and monopulse SAR system (a SAR system that uses two or more receiving radars to record echoed data due to transmission from single radar). The reference [4] provides a brief description of chirp scaling (CS) algorithm and it presents image processing using CS algorithm. The references [5] and [6] discuss a generalized formulation of extended chirp scaling algorithm, which is applicable for air and space borne SAR processing.

In this paper, the readers will get familiar with the prior SAR projects. Then, they will read a brief summary of SAR raw signal formation under SAR Signal chapter. In next chapter (SAR Raw Signal Module and Analysis in Frequency Domain), they will be introduced to equations, process and simulation algorithm. In “Simulation Program” chapter, the readers will
see tutorial for the MATLAB code in appendix A. Finally, the readers will see the simulation results and conclusion.
Prior SAR Progress

The ultimate goal of this ongoing project is to develop a piece of code which uses raw SAR signals to generate images of targets that can be used for Automatic Target Recognition (ATR) implementations. However, this project started out as a one-dimensional implementation of radar range finding. Lynn Kendrick developed the one-dimensional radar range finding simulation in 2005 as her Cal Poly Senior project. Later Brian Zaharris developed the full two-dimensional SAR simulation and range-Doppler algorithm in 2007 as a Master’s thesis. Zaharris’ simulation arrayed point targets in a two-dimensional plane of azimuth and range, with the platform traveling in the same plane along the azimuth. Zaharris’ simulation also made use of Kalman filtering of the raw SAR signal portion of the code to allow accurate imaging of limited mobility point targets at their final positions. Paul Mason used Brian Zaharris’ two-dimensional range-Doppler algorithm and adapted it to two-dimensional objects such geometrical shapes and letters composed of arrays of point targets with the azimuth and range location of each point defined in input profiles. Each point target requires a reflection to be calculated during each stage in the flight over the duration of the flight. With the realistic SAR parameters used, 900 reflections are calculated for each point target in the two-dimensional SAR simulation and due to the complexity of the radar reflection equation used, without optimization of the code each point target would require 30 seconds to calculate each reflection. For larger images, such as MSTAR images, which are of size 128x128, the simulation would take over five days to complete. Matthew Schlutz used Zaharris’ MATLAB code of point target SAR simulation to develop more complex two-dimensional and eventually three-dimensional target SAR
simulation. In order to move into ATR, moving target and more complex three-dimensional SAR simulations, a major flaw in the two-dimensional SAR simulation needed to be solved first. In this paper, a new simulation method in hybrid domain will be discussed.
SAR Signal

In this section, the theory of SAR raw signal formation will be briefly summarized. The theory of SAR systems has been worked out in detail elsewhere, so the preliminaries will be kept at a minimum and just the basics will be presented to make the paper self-consistent.

The SAR system is conventional pulsed radar, which takes advantage of the relative motion between sensor and target to synthesize a very long antenna and to achieve a high cross-range (azimuth) resolution. Each echo retains both its amplitude and its phase. It is usual to adopt the complex envelope representation in which the received signal is complex, with its real and imaginary parts obtained through quadrature demodulation from the incoming bandpass signal.

Figure 2: SAR Geometry ($\theta=$Look Angle, $r=$closest distance from $P$ to flight path, $R=$ distance from the $P$ to the radar)
Let us consider the radar sensor flying over the earth as shown in Fig. 2 with constant velocity $V_p$ at an altitude “$h=R_0 \cos \theta_o$” ($\theta_o =$ Look Angle) along $y'$ direction (Azimuth). A single scatterer, as long as stays within the antenna footprint, produces a series of echoes with arrival times and phase delays that are functions of the sensor position with respect to the elementary scatterer. In SAR, the relative motion between sensor and target is supposed known. SAR algorithms estimate the backscattering coefficient of an elementary cell by picking up from the received data that samples with the right sequence of time delays and correlating them with the corresponding sequence of phase delays. The change from one scan to another in the echo time delay is known as “range migration”. The sequence of phase delays of the echoes that coming from a single scatter are known as “target Doppler history”.

As mentioned previously the spacecraft (or the aircraft) moves at a constant velocity $V_p$ and emits signal at time $\tau$ given by:

$$p(\tau) = \text{rect} \left[ \frac{\tau}{T_p} \right] \exp[j2\pi f \tau - j\pi k \tau^2] \quad (1)$$

Wherein $f$ is the carrier frequency, $T_p$ and $k$ are the chirp duration and rate, respectively. The expression of the received signal after the demodulation steps is as follow:

$$h(y', r') = \iint dydr \gamma(y, r) \exp \left[ -j \left( \frac{4\pi}{\lambda} \right) R - j \left( \frac{4\pi}{\lambda} \right) \left( \frac{\Delta f}{T} \right) (r' - R)^2 \right] \text{rect} \left[ \frac{r' - R}{cT_p} \right] \omega^2 \left( \frac{y' - y}{\gamma} \right) \quad (2)$$

Wherein $(y', r')$ are the (output) azimuth and range coordinates. $(y, r)$ are the corresponding coordinates over the ground. In addition, $\gamma(y, r)$ is the equivalent backscattering coefficient.
is the carrier wavelength. \( \omega(.) \) is the antenna ground illumination pattern (usually approximated to \( \text{rect} \left[ \frac{y-y'}{y} \right] \)). \( Y = \frac{\lambda r}{L} \) is the azimuth footprint. L is the effective azimuth length of the physical antenna.

R is a distance from sensor position to the generic point of the scene. \( R_0 \) is a distance from the line of flight to the center of the scene. \( \Delta f \) is chirp bandwidth. C is the speed of light. \( T_p \) is the pulse duration. \( r' \) is \( \frac{c}{2} \) times the time elapsed from the pulse transmission (Sampling Coordinate).

\[
\text{rect} \left[ \frac{t}{T} \right] = \begin{cases} 1 & \text{if } |t| \leq \frac{T}{2} \\ 0 & \text{otherwise} \end{cases} \quad (3.1)
\]

\[
R^2(\eta) = R_0^2 + V_p^2 \eta^2 \quad (3.2)
\]

The instantaneous slant range, \( R(\eta) \), changes with azimuth time, \( \eta \), according to equation (3.2), in which \( R(\eta) \) is expressed as a hyperbolic function of \( \eta \). The equation represents the target trajectory, in distance units, as a function of azimuth time. The separation between range samples is \( \frac{c}{2F_r} \), where \( F_r \) is the range sampling rate. This means the trajectory migrates through range cells during the exposure time of the target in signal memory; hence, the name “Range Cell Migration” or RCM comes from. This migration complicates the processing, but ironically, it is an essential feature of SAR. This variation of slant range with time imposes an FM characteristic on the signal in azimuth direction. The hyperbolic form of the slant range equation can be expanded in a power series, resulting in a linear RCM component, a quadratic
RCM component and higher order terms. The first generation of satellite SAR processors used the power series expansion of the range equation in the time domain and the range Doppler domain. Later it was discovered that the hyperbolic form could be kept in all domains, thereby improving the processing accuracy. However, the power series expansion is sometimes useful for analysis purposes.

The range Doppler algorithm was the first algorithm developed for civilian SAR satellite processing. It is still the most widely used algorithm because of its favorable tradeoff between maturity, simplicity, efficiency, and accuracy. However, under certain conditions, its two disadvantages can become apparent. First, a high computing load is experienced when a long kernel is used to obtain high accuracy in the Range Cell Migration Correction (RCMC) operation. Second, it is not easy to incorporate the azimuth frequency dependence of SRC, which can limit its accuracy in certain high squint and wide-aperture cases.

A chirp is a signal in which the frequency increases (up-chirp) or decreases (down-chirp) with time. The chirp-scaling algorithm was developed specifically to eliminate the interpolator used for RCMC. It is based on a scaling principle whereby a frequency modulation is applied to a chirp-encoded signal to achieve a shift or scaling of the signal. Using this chirp-scaling principle, the required range-variant RCMC shift can be implemented by using phase multiplies instead of a time-domain interpolator. The algorithm has the additional benefit that Secondary Range Compression (SRC) can be made in azimuth frequency dependent. This benefit arises because the data are available in the two-dimensional frequency domain at a convenient stage in the processing.
SAR Raw Signal Module and Analysis in Frequency Domain

The expression of the received signal after demodulation steps is shown in equation (4).

The Fourier transform (FT) of (4) along the range direction is shown in (5).

\[
h(y', r') = \iiint dy \, dr \, \gamma(y, r) \exp\left[-\frac{j(4\pi)}{\lambda} R - \frac{j(4\pi k)}{c^2} (r' - R)^2 \right] \text{rect} \left[ \frac{r'-R}{cTp} \right] \omega^2 \left( \frac{y'-y}{Y} \right) \quad (4)
\]

\[
H(y', \eta) = \int h(y', r') \exp(-j\eta r') \, dr'
\]

\[
H(y', \eta) = \iiint dy \, dr \, \gamma(y, r) \exp\left[-\frac{j4\pi R}{\lambda} \right] \omega^2 \left( \frac{y'-y}{Y} \right) \int \exp\left[-\frac{j4\pi}{c^2} \right] K(r') - R)^2 \right] \text{rect} \left[ \frac{2(r'-R)}{cTp} \right] \exp[-j\eta r'] \, dr'
\]

\[
b = \frac{4\pi}{c^2} K \quad , \quad r' - R = U \quad , \quad dr' = dU \quad , \quad r' = U + R
\]

\[
H(y', \eta) = \iiint dy \, dr \, \gamma(y, r) \exp\left[-j(\eta + \frac{4\pi}{\lambda} R \right] \omega^2 \left( \frac{y'-y}{Y} \right) \quad (5)
\]

Where:

11
R is distance from A to the generic point of the scene \((R = \sqrt{r'^2 + (y' - y)^2})\). \(\gamma(y, r)\) is scene reflectivity pattern, including the phase factor \(\exp[-j(4\pi/\lambda)]\). \(\Delta f\) is chirp bandwidth. C is speed of light. \(T_p\) is pulse duration time. \(\omega(.)\) is azimuth illumination diagram of the real antenna over the ground. \(Y\) is real antenna azimuth footprint \((Y = \frac{\lambda R_o}{L})\). \(R_o\) is the distance from the line of flight to the center of the scene. \(L\) is the azimuth dimension of the real antenna. \(r'\) is \(c/2\) times the time elapsed from the pulse transmission. \(\eta\) is the variable in range frequency domain, \(b = \frac{4\pi k}{c^2}\), and \(\text{rect}[t/T]\) is standard rectangular window function.

The expression of the FT of equation (5) along the azimuth direction is shown below (6):

\[
H(\xi, \eta) = \text{rect} \left[ \frac{\eta}{bcT_p} \right] \exp \left[ \frac{j\xi^2}{4b} \right] \int dy \int dr \gamma(y, r) \omega^2 \left[ \frac{-\xi r}{Y\sqrt{\eta^2 - \xi^2}} \right] \exp(-j\xi y) \exp \left( -jr\sqrt{\eta^2 - \xi^2} \right) 
\tag{6}
\]

By using (6), the integral operator cannot be avoided because reflectivity of the target raw data is a function of range direction. Now, we can rewrite the equation (6) as follow:

\[
H(\xi, \eta) = \text{rect} \left[ \frac{\eta}{bcT_p} \right] \exp \left[ \frac{j\eta^2}{4b} \right] \int dr \Gamma(\xi, r) G(\xi, \eta, r) 
\tag{7}
\]

Where:

\[
\Gamma(\xi, r) = \int \gamma(y, r) \exp(-j\xi y) dy \tag{8}
\]

\[
G(\xi, \eta, r) = \omega^2 \left[ \frac{-\xi r}{Y\sqrt{\eta^2 - \xi^2}} \right] \exp \left( -jr\sqrt{\eta^2 - \xi^2} \right) \tag{9}
\]
$$\eta' = \eta + \frac{4\pi}{\lambda}$$

Based on Taylor series expansion, the expansion of the raw data two-dimensional spectrum phase (neglecting three, and higher order terms), around zero (Maclaurin Series) is as follow:

$$\text{Phase } (H(\xi, \eta)) = \phi = r \sqrt{\eta^2 + \frac{8\pi \eta}{\lambda} + \left(\frac{4\pi}{\lambda}\right)^2 - \xi^2 - \frac{\eta^2}{4b}}$$

$$\phi(a) + \frac{\phi'(a)}{1!}(\eta - a) + \frac{\phi''(a)}{2!}(\eta - a)^2$$

$$\phi' = \frac{r(\eta + \frac{4\pi}{\lambda})}{\sqrt{\eta^2 + \frac{8\pi \eta}{\lambda} + \left(\frac{4\pi}{\lambda}\right)^2 - \xi^2}} - \frac{\eta}{2b}$$

$$\phi'' = \frac{r \sqrt{\eta^2 + \frac{8\pi \eta}{\lambda} + \left(\frac{4\pi}{\lambda}\right)^2 - \xi^2 - r(\eta + \frac{4\pi}{\lambda}) \left(\frac{\eta + \frac{4\pi}{\lambda}}{\sqrt{\eta^2 + \frac{8\pi \eta}{\lambda} + \left(\frac{4\pi}{\lambda}\right)^2 - \xi^2}}\right)} - \frac{1}{2b}$$

$$T = \frac{4\pi r D}{\lambda} + \frac{r \eta}{D} + \left(\frac{r \lambda (D^2 - 1)}{8\pi D^3} - \frac{1}{4b}\right)\eta^2 \text{ for } a = 0$$

$$H(\xi, \eta) \approx \text{rect} \left[\frac{\eta}{bcTp}\right] H_1(\xi, \eta)$$

$$H_1 = \int dr \ \Gamma(\xi, r) \omega^2 \left(\frac{-j\xi r \lambda}{Y4\pi}\right) \exp \left[-j \left(\frac{4\pi r D}{\lambda}\right) - j \left(\frac{r}{D}\right)\eta + j \left[\frac{\lambda r (1 - D^2)}{8\pi D^3} + \left(\frac{1}{4b}\right)\eta^2\right]\right]$$

(10)
Where:

\[ D = \sqrt{1 - \left(\frac{\lambda \xi}{4\pi}\right)^2} \]

We can get the analytical solution in azimuth frequency domain and in range time domain by inverse Fourier transforming in range direction as follow:

\[
h_1(\xi, r') = \int dr \Gamma(\xi, r) \omega^2 \left[ -\frac{\xi}{Y4\pi} \right] \exp \left[ -\frac{j(4\pi r D)}{\lambda} - j\pi K_s \left( -\frac{r' - R_s(\xi)}{c} \right)^2 \right]
\]

(11)

Where:

\[
K_s = \frac{K_r}{1 + 2\lambda r K_r \left( \frac{1 - D^2}{c^2 D^3} \right)}
\]

is the effective FM chirp rate in range. \( R_s \) is

\[
R_s(\xi) = r (1 + C_s)
\]

(11.1)

equivalent range modulation in azimuth frequency domain, where:

\[
C_s(\xi) = \frac{1}{D} - 1
\]

\( C_s \) is curvature factor, which describes the Doppler frequency-dependent part of the signal trajectory.

The scaling principle that was described by Papoulis[4], whereby a frequency modulation is applied to a chirp-encoded signal to achieve a shift or scaling of the signal. Using
this “chirp scaling” principle, the required range-variant RCMC shift can be implemented; using phase multiplies instead of a time-domain interpolator. The algorithm has the additional benefit that SRC (Secondary Range Compression) can be made azimuth frequency dependent. This benefit arises because the data are available in the two-dimensional frequency domain at a convenient stage in the processing. By using Chirp Scaling principle, linear frequency modulated signal multiplied by correlated frequency modulated signal (CS factor), result is still a frequency-modulated signal, only the phase center and the frequency modulation rate changes. After range is compressed with the new frequency modulated rate, displacement occurs at the location of the signal, which makes target range curvature in frequency domain that has the same shape in different range. This is the purpose of chirp scaling to equalize all the range migration trajectories to a reference range ($r_{ref}$). By using inverse-CS principle, we can first simulate raw data, which has the same range curvature, and then we can get precise raw data by inversed-equalizing with chirp scaling factor. Accordingly, we can derivate curvature equalizing phase factor from (11) and (11.1) by using this idea.

\[
\Phi_1(\xi, r', r_{ref}) = e^{-j\left(\frac{4\pi}{c^2}\right)K_s C_s \left(r' - r_{ref} (1 + C_s)\right)^2} \tag{12}
\]

\[
h_{cs}(\xi, r') = h_1(\xi, r')\Phi_1(\xi, r', r_{ref})
\]

\[
h_{cs} = \int dr \left[ \Gamma(\xi, r) W^2 \left(\frac{-\xi r \lambda}{Y, 4\pi}\right) \Phi_2(\xi, r) \right] e^{-j\frac{4\pi}{c^2}K_s(1+C_s)(r'-(r+C_s r_{ref}))^2} \tag{13}
\]

\[
\Phi_2(\xi, r) = e^{-j\frac{4\pi}{\lambda}D - \frac{4\pi}{c^2}K_s(1+C_s)C_s (r-r_{ref})^2}
\]
\[
H_{cs}(\xi, \eta) = \phi_3(\xi, \eta, r_{ref}). \text{FT}_r \left\{ \Gamma(\xi, r) W^2 \left( -\frac{\xi r \lambda}{Y_4 \pi} \right) \phi_2(\xi, r) \right\} \quad (14)
\]

Where the \( \phi_3 \) is:

\[
\phi_3(\xi, \eta, r_{ref}) = e^{j \frac{c^2 \eta^2}{16 \pi \eta K_0 (1 + c^2)}} e^{-j \eta C_3 r_{ref}}
\]

The simulation program of the target raw data in extended scenes can be extracted from equations (11), (13), and (14). The simulation algorithm is shown in figure 3.

\[\gamma'(x, r)\]

\[\text{FT}_x\]

\[W^2(-\xi \cdot r \lambda / \eta_4 \pi) \quad \Phi_2(\xi, r)\]

\[\text{FT}_r\]

\[\Phi_3(\xi, \eta, r_{ref}) \cdot \text{rect} \left[ \frac{\eta}{b \cdot c \cdot T_p} \right]\]

\[\text{IFT}_\eta\]

\[\Phi_1^{-1}(\xi, r', r_{ref})\]

\[\text{IFT}_\xi\]

\[h(x', r')\]

\textit{Figure 3: Simulation diagram of raw data based on CS principle}
**Simulation Program**

The simulation program was written in MATLAB code based on an algorithm introduced in the previous section (figure 3). The raw data simulation program placed in appendix A.

Appendix A shows the SAR raw data simulation in MATLAB coding. The coding is broken into a few self-explanatory sections. At the beginning user can setup plotting in time-domain, frequency-domain, or either one by setting Time Plots and Spectral Plots parameters to one. In addition, the user can set the SAR airborne parameters such as carrier frequency, sampling rate, pulse repetition frequency, flight duration, Doppler centroid, platform velocity, chirp pulse duration, chirp bandwidth, etc in this section. (Note: One of the key parameters in the SAR processing is Doppler centroid ($f_{dc}$). The Doppler Effect is due to the relative motion between the pointing angle of the antenna and the target, where the frequency of the received echoes changes compared to the transmitted signals. Equation 15 expresses the Doppler centroid, where the $\epsilon$ is the angle between the antenna main beam and zero Doppler plane. All the simulations in this thesis were calculated based on zero Doppler centroid.) Then, the program reads the target-backscattering coefficients, and aligns the target area to fit into SAR scan area. From this point, the program calculates SAR raw data based on chirp scaling algorithm that was explained in the previous section.

$$f_{dc} = \frac{2V_p}{\lambda} \sin \epsilon \quad (15)$$
Correlation between equations in this paper and parameters in the MAT-LAB code are as follows:

\[ f_a = \xi \]
\[ f_r = \eta \]
\[ \text{PHI}_x = \Phi_x \text{ where: } x = 1, 2, 3 \]
\[ \gamma = \gamma(y, r) \]
\[ \lambda = \lambda \]
\[ w_2 = W^2(.) \]
Simulation Results

In this section, the simulation results are presented. For convenience, a point target (3x3) has been used to make comparison between Time Domain method and Frequency Domain method. The system parameters are shown in Table 1 (values are based on real application).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Pixel Parameters</th>
<th>3x3</th>
<th>1171x747</th>
<th>Units</th>
<th>Pixel Parameters</th>
<th>3x3</th>
<th>1171x747</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrier Frequency</td>
<td>5.330E+09</td>
<td>5.330E+09</td>
<td>Hz</td>
<td>Near Range</td>
<td>7.99E+05</td>
<td>7.99E+05</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>Sampling Frequency</td>
<td>1.920E+07</td>
<td>1.920E+07</td>
<td>Hz</td>
<td>Platform Velocity</td>
<td>8.55E+02</td>
<td>8.55E+02</td>
<td>m/s</td>
<td></td>
</tr>
<tr>
<td>PRF</td>
<td>1.070E+03</td>
<td>1.174E+03</td>
<td>Hz</td>
<td>Chirp Pulse Duration</td>
<td>2.16E-05</td>
<td>2.16E-05</td>
<td>Sec</td>
<td></td>
</tr>
<tr>
<td>dur</td>
<td>1.000E+00</td>
<td>1.000E+00</td>
<td>Sec</td>
<td>Chirp Bandwidth</td>
<td>1.62E+07</td>
<td>1.62E+07</td>
<td>Hz</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: SAR System Parameters

Based on the result in figure 4 and figure 5, the simulation method can get average phase error within 60 degrees in range direction and within 40 degrees in azimuth direction, so we can get the precise phase in comparison to the time domain. This result shows how close phase calculations in frequency domain are to the time domain calculations. This means the frequency domain can get phase error no more than 60 degree in range direction and no more than 40 degree in azimuth direction.

To verify the efficiency of the Chirp Scaling method, with the system parameters shown in table 1, the raw data of extended scenes has been simulated and shown in figure 6. It takes up to 8.5864 seconds to simulate 1171x747 pixel raw data, based on the chirp scaling principle method in figure (3) on Intel® Core™2 Dual CPU with 2.2GHz frequency. The results are shown in table 2.
Figure 4: Cuts of the phase curves comparison in azimuth direction and, average error of the phase curve.
Figure 5: Cuts of the phase curves comparison in range direction, and the average error of the phase curve

<table>
<thead>
<tr>
<th>Methods</th>
<th>Pixels</th>
<th>Time Domain</th>
<th>Frequency Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 x 3</td>
<td>15.06 Sec</td>
<td>3.52 Sec</td>
</tr>
<tr>
<td></td>
<td>1171 x 747</td>
<td>+ 10 Hrs</td>
<td>4.83 Sec</td>
</tr>
</tbody>
</table>

Table 2: Calculation Efficiency Result
Figure 6: (a) Surface Reflectivity of Target (CalPoly Map) (b) Amplitude of the raw data of extended scenes (c) Simulated SAR image of raw data
Conclusion

We can see from simulation results that the chirp scaling (CS) principle method can get precise result compare to time domain simulation. The results showed the phase error could not be more than 60 degree in range direction and 40 degree in azimuth direction. This method avoids the time domain integral and because it processes in frequency domain, it has great computational efficiency. To verify the efficiency, I ran a simulation (see MATLAB code in Appendix A) of extended scenes with the surface reflectivity that is shown in figure 7 and system parameters that are shown in table 3. The simulation took about 2.14 seconds to simulate 600x932 pixel raw data on Intel® Core™2 Dual CPU with 2.2GHz frequency.

In addition, this method can generate simulated raw data based on any target reflectivity matrixes without using inverse SAR image algorithm.

<table>
<thead>
<tr>
<th>Carrier Frequency</th>
<th>5.33E+09 (Hz)</th>
<th>Chirp Bandwidth</th>
<th>1.60E+07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Sampling Rate</td>
<td>1.92E+07 (Hz)</td>
<td>Platform Velocity</td>
<td>750 (m/s)</td>
</tr>
<tr>
<td>Flight Duration</td>
<td>2 Second</td>
<td>Chirp Pulse Duration</td>
<td>2.16E-05</td>
</tr>
</tbody>
</table>

*Table 3: SAR Simulation System Parameters*
Figure 7: Surface reflectivity of target, amplitude of raw data and focused image
**Future Work**

There are several areas of future work for continuations on this project. An important area of future work is to add SAR sensor trajectory deviations so the program can simulate more realistic target and platform. Second area is to add target motion to the simulation, so the raw data can be generated for ATR application. Third, extend the code to include more algorithms. It will be useful to include other SAR processing algorithms and run a comparison between the results obtained from different algorithms. Fourth, to include known satellite data formats, or option of selecting known data formats.

This program runs under MATLAB. However, it is not very difficult to convert the whole program to another environment such as C++ or Java. This is the best solution to get rid of memory matters and MATLAB limitations of array and matrix size.
References


Appendix A: SAR Simulation Code in MATLAB

% =========================================================================
% Author A. Shoalehvar 12/1/2009
% The purpose of this script is to generate SAR raw data using Chirp Scaling Principal. The raw data will be saved in a file on the computer hard-drive for future use by other script or programs.
% Last Update 4/30/2012
% =========================================================================
clear all
tic;
TimePlots = 0; % To draw Time Plots, TimePlots should equal to 1 otherwise 0
SpectPlots = 0; % To Draw, SpectPlots should be 1 otherwise 0
PLOTS = 1; % To Draw plots, PLOTS should be 1 otherwise 0

% =========================================================================
% SAR Airborne Parameters
% =========================================================================
c = 3.00e8; % speed of light in vacuum
f_c = 5.33e9; % carrier frequency (Hz)
f_s = 1.92e7; % data sampling rate (Hz)
PRF = 0.300e3; % pulse repetition frequency
dur = 2.0; % flight duration
t_p = 2.16e-005; % chirp pulse duration
B = 1.60e7; % chirp bandwidth (Hz)
echoes = PRF*dur; % number of radar echoes in data file
near_range=7.99e5; %Near Range (m)
X0=200; % Half Target Area Width (Target is located within [Xc-X0,Xc+X0])
Xc= near_range -X0;
t_near = (2*near_range)/c; % near range fast time (2 x range)
k_r=-B/t_p; % Range Chirp Rate
dt=1/(2*f_s); % Time Domain Sampling Interval
Ts=(2*(Xc-X0))/c; % Start time of sampling

...
\[ \text{Tf} = \frac{2(Xc + X0)}{c + t_p}; \quad \% \text{End time of sampling} \]

\[ \text{samples} = 2 \times \text{ceil} \left( \frac{0.5 \times (\text{Tf} - \text{Ts})}{\text{dt}} \right); \quad \% \text{number of samples per radar echo (Range)} \]

\[ \tau = \text{Ts} + (0 : \text{samples} - 1) \times \text{dt}; \quad \% \text{Time array for data acquisition (fast time)} \]

\[ f_{\text{dc}} = 0; \quad \% \text{Doppler centroid (squint angle)} \]

\[ v = 7.50e2; \quad \% \text{SAR satellite platform velocity} \]

\[ r_{\text{ref}} = (\tau(1) + \text{samples}/2/f_s)/2c; \]

\[ \alpha = 1.0; \]

\[ f_{\text{a}} = -\text{PRF}/2 + f_{\text{dc}} : \text{PRF} / \text{echoes} : f_{\text{dc}} + \text{PRF}/2 - \text{PRF}/\text{echoes}; \]

\[ f_{\text{r}} = -f_s/2 : f_s/\text{samples} : f_s/2 - f_s/\text{samples}; \]

\[ \lambda = c/f_c; \]

\[ L_a = 1.0; \quad \% \text{Antenna Length (m)} \]

\[
\% \text{=================================================================}
\% \text{Target Area}
\%
\% \text{=================================================================}
\]

\[ \text{target\_name} = 'TG1'; \]

\[ [\text{target1}, \text{map}] = \text{imread} (\text{target\_name}, 'gif'); \]

\[ \text{target} = \text{double} (\text{ind2gray} (\text{target1}, \text{map})); \]

\[ \text{gamman} = \text{double} ((\text{target}) / 255); \]

\[ \text{Naz} = \text{size} (\text{gamman}, 1); \quad \% \text{row, X direction} \]

\[ \text{Nrg} = \text{size} (\text{gamman}, 2); \quad \% \text{column, Y direction} \]

\[ \text{Ntarget} = \text{Naz} \times \text{Nrg}; \]

\[ \text{clear target target1 map target\_name} \]

\[
\% \text{Target Area Alignment}
\]

\[ \text{data} = \text{zeros} (\text{echoes}, \text{samples}); \]

\[ \text{data} (\text{ceil} ((\text{echoes} - \text{Naz}) / 2) : (\text{Naz} + \text{ceil} ((\text{echoes} - \text{Naz}) / 2) - 1), \text{ceil} ((\text{samples} - \text{Nrg}) / 2) : (\text{Nrg} + \text{ceil} ((\text{samples} - \text{Nrg}) / 2) - 1)) = \text{gamman}; \]

\[ \text{gamman} = \text{data}; \]

\[ \text{clear data} \]

[if \ (\text{PLOTS} == 1)\]

\[ \text{figure}; \]
```matlab
colormap(gray(256))
imagesc(gamman);
axis('image');axis('xy')
xlabel('range')
ylabel('azimuth')
title('SAR Input Reflectivity')
end;

% Data File Location
OutputFileName='SAR_RawData';

% ************************************************************************
% Chirp Scaling Algorithm
% Azimuth FFT
% ************************************************************************
gamman = ftx(gamman);

if (SpectPlots==1)
    G=angle(gamman);
    xg=max(max(G)); ng=min(min(G)); cg=255/(xg-ng);
    figure;
    colormap(gray(256))
    image(256-cg*(G-ng));
    axis('image');axis('xy')
    xlabel('range')
    ylabel('Doppler frequency phase')
    title('range signal/Doppler domain')
end;

% ************************************************************************
% Chirp Scaling
% ************************************************************************
D = (1 - (f_a*lambda/2/v).^2).^0.5;
C = 1./D - 1;
```
\[ R = \frac{r_{\text{ref.}}}{D}; \]
\[ C_{\text{scl}} = C + (1 - \alpha) \cdot (1 + C) \cdot \frac{1}{\alpha}; \]
\[ k_{\text{inv}} = \frac{1}{k_r} - \frac{2 \cdot \lambda \cdot r_{\text{ref.}} \cdot (D - 1)}{c^2 \cdot D^3}; \]
\[ k = \frac{1}{k_{\text{inv}}}; \]

% =========================================================================
% Stretch surface reflectivity in azimuth direction \( H1 \cdot H2 = \Phi2 \)
% =========================================================================
\[ r_0 = \frac{\tau}{2c}; \]
\[ L_s = \frac{\lambda \cdot r_0}{L_a}; \]
\[ x = k \cdot C_{\text{scl}} \cdot (1 + C)^2 \cdot \frac{1}{c^2 \cdot (1 + C_{\text{scl}})}; \]
\[ X = x(1) \cdot \text{ones}(1, \text{samples}); \]
\[ z = (r_0 - r_{\text{ref.}})^2; \]
\[ Z = \text{ones}(\text{echoes}, 1) \cdot z; \]
\[ dphi = 4 \pi X \cdot Z; \]
\[ H1 = \exp(-1i \cdot dphi); \]
\[ r_{0,\text{scl}} = r_{\text{ref}} + \frac{r_0 - r_{\text{ref}}}{\alpha}; \]
\[ X = \text{ones}(\text{echoes}, 1) \cdot r_{0,\text{scl}}; \]
\[ Z = (D(1) - 1) \cdot \text{ones}(1, \text{samples}); \]
\[ H2 = \exp(-1i \cdot 4 \pi / \lambda \cdot X \cdot Z); \]
\[ w_2 = (\text{sinc}(-\lambda \cdot f_a) / (L_s \cdot 4 \pi))^2 \];
\[ \Phi2 = H1 \cdot H2; \]
\[ \text{gamman} = \text{gamman} \cdot \Phi2; \]
\[ \text{gamman} = \text{diag}(w_2) \cdot \text{gamman}; \]

% =========================================================================
% Range FFT
% =========================================================================
\[ \text{gamman} = \text{fty}(\text{gamman}); \]

if (SpectPlots==1)
\[ G = \text{angle}(\text{gamman}); \]
\[ xg = \text{max} \left( \text{max}(G) \right); \]
\[ ng = \text{min} \left( \text{min}(G) \right); \]
\[ cg = 255 / (xg - ng); \]
\[ \text{figure}; \]
```matlab
% colormap(jet(256))
image(256-cg*(G-ng));
axis('image');axis('xy')
xlabel('signal frequency phase')
ylabel('Doppler frequency phase')
title('2D spectral domain')
end;

% Bulk RCMC, Range Compression: PHI3

x = 1./(k.*(1+C_scl));
X = x(:)*ones(1,samples);
y = f_r.^2;
Y = ones(echoes,1)*y;
z = f_r;
Z = ones(echoes,1)*z;
A = C(:)*ones(1,samples);
Z = Z .* A;
PHI3 = exp(1i*pi*X.*Y) .* exp(-1i*4*pi*r_ref/c.*Z);
w_3=sinc((f_r*c)/(4*pi*k_r*t_p))*1 - 0;
Rect=ones(echoes,samples)* diag(abs(fftshift(fft(fftshift(w_3.'))).'));
gamman = gamman .* PHI3;
gamman = gamman .* Rect;

if (SpectPlots==1)
G=angle(gamman);
xg=max(max(G)); ng=min(min(G)); cg=255/(xg-ng);
figure;
colormap(jet(256))
image(256-cg*(G-ng));
axis('image');axis('xy')
xlabel('signal frequency phase')
ylabel('Doppler frequency phase')
title('bulk rcmc and range compression')
```

end;

if (TimePlots==1)
    figure;
    colormap(jet(256))
    mesh(abs(iftx(ifty(gamman)))):
    xlabel('range')
    ylabel('azimuth')
    zlabel('intensity')
    title('after H2')
    axis('tight');
end;

% =========================================================================
% Range IFFT
% =========================================================================

gamman = ifty(gamman);

if (SpectPlots==1)
    G=angle(gamman);
    xg=max(max(G)); ng=min(min(G)); cg=255/(xg-ng);
    figure;
    colormap(jet(256))
    image(256-cg*(G-ng));
    axis('image');axis('xy')
    xlabel('range')
    ylabel('Doppler frequency phase')
    title('range Doppler Domain after range compression')
end;

% =========================================================================
% Chirp Scaling (PHI1)
% =========================================================================

x = k.*C_scl;
X = x(:).*ones(1,samples);
\[ \text{Tau} = \text{ones}(\text{echoes}, 1) \times \text{tau}; \]
\[ y = 2. \times \text{R./c}; \]
\[ Y = y(:, :) \times \text{ones}(1, \text{samples}); \]
\[ Z = (\text{Tau} - Y) .^2; \]
\[ \text{PHI1} = (\exp(-1i \times \pi \times X \times Z)).^(-1); \]
\[ \text{gamman} = \text{gamman} \times \text{PHI1}; \]

if (SpectPlots==1)
    \[ G = \text{angle}(\text{gamman}); \]
    \[ xg = \text{max}(\text{max}(G)); \ ng = \text{min}(\text{min}(G)); \ cg = 255/(xg - ng); \]
    figure;
    \text{colormap(jet(256))}
    \text{image(256 - cg*(G - ng));}
    \text{axis('image'); axis('xy')}
    \text{xlabel('range')}
    \text{ylabel('Doppler frequency phase')}
    \text{title('chirp scaling and range scaling')}
end;

if (TimePlots==1)
    figure;
    \text{colormap(jet(256))}
    \text{mesh(abs(\text{iftx(gamman)}));}
    \text{xlabel('range')}
    \text{ylabel('azimuth')}
    \text{zlabel('intensity')}
    \text{title('after H1')}
    \text{axis('tight');}
end;

\%========================================================================
\% Azimuth IFFT
\%========================================================================
\text{gamman} = \text{iftx(gamman);}
if (PLOTS==1)
    G=abs(gamman);
    xg=max(max(G)); ng=min(min(G)); cg=255/(xg-ng);
    figure;
    colormap(gray(256))
    imagesc((cg*(G-ng))./256);
    axis('image');axis('xy')
    xlabel('range')
    ylabel('azimuth')
    title('Raw data')
end;
toc

end of Raw Data Generation

data = gamman;
PT=toc;
save(OutputFile_Name, 'data', 'echoes', 'samples', 'dur')

End
Appendix B: Fourier and Inverse Fourier Transform Coding in MATLAB

used in SAR Raw data Simulation Process

Fast Fourier Transform (FFT) in X direction:
%====================================================================
% Forward FFT w.r.t. the first variable %
%====================================================================
function fs=ftx(s)
fs=fftshift(fft(fftshift(s)));

Fast Fourier Transform in Y direction:
%====================================================================
% Forward FFT w.r.t. the second variable %
%====================================================================
function fs=fty(s)
fs=fftshift(fft(fftshift(s.'))).';

Inverse Fast Fourier Transform (IFFT) in X direction:
%====================================================================
% Inverse FFT w.r.t. the first variable %
%====================================================================
function s=iftx(fs)
s=fftshift(ifft(fftshift(fs)));

Inverse Fast Fourier Transform (IFFT) in Y direction:
%====================================================================
% Inverse FFT w.r.t. the second variable %
%====================================================================
function s=ifty(fs)
s=fftshift(ifft(fftshift(fs.'))).';