RapidSCAT Slice Spatial Response Function Contour Parameterization

John Clyde Niedfeldt

A thesis submitted to the faculty of Brigham Young University in partial fulfillment of the requirements for the degree of

Master of Science

David G. Long, Chair Greg Nordin D.J. Lee

Department of Electrical Engineering Engineering Brigham Young University

2016

Copyright © 2016 John Clyde Niedfeldt All Rights Reserved

ABSTRACT

RapidSCAT Slice Spatial Response Function Contour Parameterization

John Clyde Niedfeldt Department of Electrical Engineering Engineering, BYU Master of Science

The spatial response function (SRF) of the backscatter measurements for a radar scatterometer is often used in reconstruction. It has been found that in many cases the SRF can be approximated as a binary function that is 1 inside the - 6 dB contour of the SRF and 0 outside. This improves the computation speed of reconstruction. Computing the SRF contour can still be a lengthy computation, which can be simplified by precomputing and tabulating key SRF contours. The tabular parameterization for many spinning scatterometers, i.e., QuikSCAT, is straight-forward. For RapidSCAT, this estimation is more involved than other radars due to the irregular orbit of its host platform, the International Space Station (ISS). This thesis presents a process for parameterizing the slice contours for Rapid-SCAT that are acceptable for reconstruction purposes. This thesis develops a new process for parameterizing slice contours. First, RapidSCAT SRFs are calculated using XfactorRS3, and -6 dB slice contours are found using *matplotlib*. Then, a suitable filter is found for reducing noise present in slice contours due to quantization error and interpolation inaccuracies. Afterwards, the polygon comparison algorithm is used to determine a set of approximation points. With the approximation points selected, the 3-rd order linear approximation is calculated using parameters available in the L1B data files for RapidSCAT. Finally, analysis of the parameterization is performed.

Overall, I developed a process that parameterizes RapidSCAT slice contours with an average root mean square (RMS) error of roughly 1.5 km. This is acceptable for the application of the slice parameterization algorithm and significantly reduces computation compared to fully computing the SRF.

Keywords: RapidSCAT, Parameterization, Slice, Contour

ACKNOWLEDGMENTS

First and foremost, thanks to Dr. Long, who was an example, mentor, and friend to me as I worked on my thesis. I also thank my family for their encouragement and support. Lastly, I thank my coworkers, past and present, in the Microwave Earth Remote Sensing (MERS) Laboratory for being phenomenal colleagues with great potential.

TABLE OF CONTENTS

LIST OF TABLES				
LIST O	\mathbf{F} FIGURES	i		
Chapte	r 1 Introduction	L		
Chapte	r 2 Background	3		
2.1	Scatterometers	3		
2.2	SeaWinds on QuikSCAT	1		
2.3	RapidSCAT	7		
2.4	XfactorRS3)		
2.5	Δf	2		
2.6	Eggs and Slices	3		
2.7	Reconstruction	5		
Chapte	r 3 Contour Preparation	7		
3.1	Data Used	7		
	3.1.1 Xfactor	3		
	3.1.2 L1B	3		
3.2	Find the Contours	9		
3.3	Filter the Contours	1		
3.4	Relation of Contours	3		
3.5	Approximation	5		
	3.5.1 Determining Important Points	3		
3.6	Determine Valid Slices)		
Chapte	r 4 Parameterization and Results	7		
41	Parameterization 37	7		
1.1	$4 1 1 \text{Calculate Weightings} \qquad \qquad$	2		
4 2	Analysis of Parameterization 30	ן ג		
7.2		,		
Chapte	r 5 Conclusion	5		
5.1	Future Work 47	7		
0.1		,		
REFER	49 AENCES)		
Append	lix A Comparing Polygonal Shapes	2		
Append	lix B Contour Encircling Approximation (CEA) Method 57	7		
Append	lix C Analysis of the QuikSCAT Post Wind Mission (PWM) Data 61	L		

LIST OF TABLES

2.1	RapidSCAT and QuikSCAT Basics Compared [1]	7
$3.1 \\ 3.2$	NASA Processing Levels [2]	19 20
B.1	RMS Error for Dynamic vs. 1.5776 Fixed Encircling Radius	59

LIST OF FIGURES

2.1	Illustration of SeaWinds aboard QuikSCAT [4].	5
2.2 2.3	Diagram of SeaWinds on QuikSCAT geometry [5]	5
2.4	compensate for rotation around the sun	6
2.5	course of two months	8
2.6	orbit	9 10
2.7	Example integration grid for a RapidSCAT egg and slice 6	11
2.8	X plotted as a function of Δf . The red line is the 3^{rd} order fitted curve.	12
2.9	The BapidSCAT Xfactor egg and slice processor responses. In practice, the	
2.0	egg response is calculated using a sum of the slice responses	14
2.10	All azimuth angles, both beams, an value liner snees at a position hear the	1 /
0.11	equator. The spacecraft is moving northeast.	14
2.11	the sampling and reconstruction process. The sampling process is already determined by instrument design and orbit geometry. For reconstruction, $\sigma^0[x, y]$ is represented from the QuikSCAT measurements, subject to param- eters such as pixel size and reconstruction algorithm. [7]	15
2.12	Examples of grid algorithm (left) and ave algorithm (right). Using a knowl- edge of the contours of the egg contours (green ellipses), the ave algorithm can determine information on a finer grid than the grid method, which is also known as the "drop in a bucket" method [7]	16
		10
3.1	The RapidSCAT X factor slice processor response at a particular azimuth angle and orbit position for Slice 6.	21
3.2	An example of the inner 8 out of the 12 slices for RapidSCAT. Slices 1 and 12 are rarely used as they are considered "guard bands". The axes "km north" and "km east" are defined with respect to the boresight of the antenna. Measurements taken at 0 degrees azimuth ascending pass of equator, with the	
	spacecraft heading portheast	იე
22	An example glice contour and its filtered version. Notice that the filtered	
J.J	contour closely resembles that of the original.	23
3.4	Plots of slice contours and filtered slice contours	24
3.5	Rotation angle for all 32 orbit times and 36 azimuth angles in a given orbit for the inner beam. The red point indicates the 0 orbit time, 0 azimuth angle location. Each vertical set of points span the 32 orbit times for that angle,	0.0
	and more in depth view of the vertical set of points can be seen in Figure 3.6.	26

3.6	A characterization of the effect of orbit time on rotation angle. The apparent sinusoidal nature is a result of the change in direction of the ISS with respect	
	to ground as it orbits around Earth	27
3.7	The residual rotation angle in degrees for a slice after compensating for az-	
	imuth angle rotation when compared to the 0 orbit time, 0° azimuth angle	00
20	case. The red point indicates the U orbit time, U azimuth angle case These plots are produced from an ideal orbit of PapidSCAT at an arbit time.	28
5.0	These plots are produced from an ideal orbit of RapidSCAT at an orbit time of 0. One interesting thing to note is that the contours for $4z = 100^{\circ}$ are	
	still not well aligned according to human intuition though the least squares	
	orientation is certainly maximized.	29
3.9	36 approximation points evenly spaced according to θ	30
3.10	The method for determining important points graphically demonstrated. Here	
	8 approximation points are used to define the contour shown above. The black,	
	green, and red points are all approximation points. The black and green dots	
0.11	are given in order to correlate the points on the two plots.	31
3.11	Filtering noisy slices based on g-factor. It is evident that the g-factor method filters more glices than gliceman (see Figure 2.12) and a vigual comparison	
	with some of the slices the Slice Max method shows that some valid slice	
	contours are being discarded.	32
3.12	A scatter plot of the g-factor as a function of l_2 polygon distance for the over	
	94,000 pulses analyzed, with points plotted for each slice (top) and for only	
	slice 5 (bottom) in each pulse. A histogram is overlaid where a bin has a	
	count greater than 10. The top left corner shows slivers of color where most	.
9 1 9	of the measurements are located.	34
3.13	A scatter plot of the maximum slice value in the spatial response function	39
0.14	as a function of l_2 polygon distance for the over 94.000 pulses analyzed, with	
	points plotted for each slice (top) and for only slice 5 (bottom) in each pulse.	
	A histogram is overlaid where a bin has a count greater than 10	36
11	Status of each parameterization for the inner beam. Creen indicates that	
4.1	the parameterization used incorporates the slice selection limits mentioned in	
	Section 3.6, orange indicates the slice selection limits had to be loosened in	
	order to create a usable parameterization, and yellow indicates that creating	
	a slice parameterization was not possible due to range gate clipping	40
4.2	Status of each parameterization for the outer beam. Green indicates that	
	the parameterization used incorporates the slice selection limits mentioned in	
	Section 3.6, orange indicates the slice selection limits had to be loosened in order to create a washle parameterization, and valley indicates that creating	
	a slice parameterization was not possible due to range gate clipping	41
4.3	Average of l_2 RMS errors in km for each curve used for each parameterization	TI
1.0	for the inner beam.	42
4.4	Average of l_2 RMS errors in km for each curve used for each parameterization	
	for the outer beam. \ldots	43
4.5	A good estimate obtained using parameterization.	44

4.6	A bad estimate obtained using parameterization	44
A.1 A.2	An example turning function for the original and filtered contour cases An example of applying the algorithm discussed in this section to two different turning functions in order to find the minimum distance metric between the	53
A.3	shapes	55 56
B.1	The Contour Encircling Approximation (CEA) method. The CEA method works by first encircling the contour to be approximated with a finite number of points evenly distributed in a circle. Then the nearest points to the curve are found by determining the point on the original contour with the smallest l_2 distance for each encircling point. The green lines in the image show the nearest neighbor.	58
B.2	The plot of the encircling radius and the resulting estimated contour area. Notice that the maximum point on the "fitted area" curve is not at the ends but in the middle	50
B.3	A histogram of the original contour radius plotted against the optimum en- circling radius for each of the slices. The data is approximately 45 revolutions of data where revolutions with different Doppler tables are selected. The red line indicates the unbiased linear fit that minimizes the least square error	59 60
C.1	JPL produced figure which shows that the variance in QuikSCAT average daily σ^0 is not an annual cycle as was expected. This suggests that QuikSCAT	60
C.2	might be inaccurately calibrated	62
		00

Chapter 1

Introduction

Radars have been used for decades to gather information about what is around us. Military radars are used to detect incoming missiles, aircraft, maritime vessels, and other threats and is also used for surveillance. Weather radars are a class of radar used for measuring current weather conditions. Data from space-based radars is used for measuring windspeed over the ocean, composition of gases in our atmosphere, tracking icebergs, oil-spill extent, and more. Truly, radars are essential to many parts of our society today.

One of the inherent difficulties of a space-based radar—such as a scatterometer—is its relatively low resolution. We desire to generate images with increased resolution compared to conventional radar, since generally each low resolution radar footprint can cover many square kilometers. One method to increase the image resolution is to use "reconstruction". Reconstruction entails taking measurements with overlapping footprints and using an inversion algorithm to calculate the backscatter at a each of the overlapping locations that make up the measurements. To make the process of reconstruction simpler, there are some approximations and estimations that can be applied. In practice, these contours are usually pre-calculated [8]. For prior space-based radars, such as QuikSCAT, this is a straight forward process due to the consistent orbit of the radar, but RapidSCAT is a different story.

Unlike other radars, RapidSCAT is located on the International Space Station (ISS) which has a variable attitude and orbit height. This, in addition to other challenges associated with being hosted by the ISS, makes estimating contours to be used for reconstruction more difficult with RapidSCAT than for almost any other scatterometer. Accounting for

these variations and determining a reliable method of calculating pulse contours is the focus of this thesis.

In this thesis I calculate a 3-rd order linear parameterization of slice contours versus orbit position and antenna rotation for RapidSCAT for use in reconstruction. To accomplish this, first the spatial response functions (SRFs) of selected measurements are calculated. From the SRFs, the -6 dB slice contours are found, which are the main building block for this thesis. The -6 dB slice contours are then filtered to reduce noise. Afterwards, the contours are approximated using a predetermined number of points. The parameterization of the approximated slice contours is then calculated. Finally, an analysis of the parameterization is presented. The parameterization enables rapid computation of measurement SRF contours for any orbit position and antenna angle.

An outline of this thesis is as follows. Basic concepts necessary to the thesis are introduced in Chapter 2. The calculation of SRFs, and approximation of slice contours is explained in Chapter 3. The parameterization and analysis are covered in Chapter 4. Finally, a summary and ideas for future work are presented in Chapter 5.

Chapter 2

Background

This chapter provides some background for this thesis. Scatterometers are reviewed in Section 2.1. The two primary scatterometers used in this text, QuikSCAT and Rapid-SCAT, are reviewed in Sections 2.2 and 2.3, respectively. Thereafter an overview of image reconstruction is presented in Section 2.7.

2.1 Scatterometers

A radar scatterometer is defined as a device used for measuring the radar scattering coefficient quantitatively [9]. RapidSCAT is such a device. It sends out electromagnetic waves which hit the earth, part of which typically reflects back to RapidSCAT which then measures the return power. Utilizing the profile of the power return, the backscatter characteristics of the surface are calculated. The main measure of the scattering characteristics of the surface used in this study is the normalized radar scattering coefficient (σ^0), also known as the backscatter coefficient, or normalized radar cross-section. σ^0 differs from the radar cross-section (σ) in that it compensates for the area illuminated by the beam so that measurements can accurately be related one to another.

Scatterometers can be used to gather information about geophysical properties, which can be used for various studies that benefit society. Some examples of data that can be collected, as well as applications for said data, are given hereafter. Ocean wind measurements are used in weather prediction and ocean studies [10, 11]. Iceberg tracking [12–14] is essential for safe maritime transport of goods. Forest extent measurements enable study of forests, such as the Amazon rainforest, and used for measuring deforestation. Snow coverage extent [15, 16], as well as Antarctic and Arctic ice extent [17, 18], are used in studies of global warming. Oil spill mapping [19–21] is invaluable for nature preservation as it aids in determining containment boundaries as well as estimating the effectiveness of cleanup efforts.

2.2 SeaWinds on QuikSCAT

This section covers the essential information pertaining to the SeaWinds scatterometer on QuikSCAT. The SeaWinds scatterometer on QuikSCAT-hereafter referred to as QuikSCAT-commenced its mission in 1999 [22]. It was originally launched to fill the gap in Ku-Band ocean wind data left by the failure of NSCAT—the NASA Scatterometer launched in August of 1996—and was to be decommissioned with the launch of *SeaWinds* aboard *ADEOS-II*—referred to as SeaWinds—2.5 years later. QuikSCAT went on to surpass expectations and its data was used for calculating sea winds for the next 10 years—many years after *SeaWinds* on *ADEOS-II* failed—until 2009 when a ball bearing seized up, causing the antenna to stop spinning. An illustration of the scatterometer aboard its host instrument can be seen in Figure 2.1.

QuikSCAT, whose geometry can be seen in Figure 2.2, is a rotating, pencil-beam scatterometer. Pencil-beam scatterometers illuminate an approximately elliptical area on the incident surface, which in the case of QuikSCAT is the result of the antenna being a parabolic reflector. It also has a sun-synchronous polar orbit, which means it maintains the same orientation with respect to the sun year-round. A figure demonstrating the definition of a sun-synchronous orbit is shown in Figure 2.3. The importance of a sun-synchronous orbit is that the radar's measurements in this orbit are then all at the same local time of day. For instance, QuikSCAT's measurements near the equator are taken at 6 am or 6 pm local time. This is important for ensuring that data from subsequent orbits can be related to a particular orbit with regard to diurnal effects (which is discussed more in Section 2.3).

QuikSCAT's orbit is Keplerian and can therefore be described by the six Keplerian Orbit Parameters: eccentricity, semimajor axis length, inclination angle, longitude of ascending node, argument of periapsis, and the mean anomaly. This allows for an efficient means



Figure 2.1: Illustration of SeaWinds aboard QuikSCAT [4].



Figure 2.2: Diagram of SeaWinds on QuikSCAT geometry [5].



Figure 2.3: Sun Synchronous Orbit diagram [6]. The green line represents a sun-synchronous orbit, which means that it maintains the same orientation with respect to the sun throughout the year. The red line indicates an orbit which does not compensate for rotation around the sun.

of defining the orbit with only a few parameters. QuikSCAT's orbit altitude is located at an average of 800 km above the earth's surface. At this distance from the Earth's surface there is little atmospheric drag on the satellite, allowing QuikSCAT to maintain a near constant orbit over its 17 year lifespan.

The consistency of QuikSCAT's orbit, together with the accuracy and reliability of its ocean wind measurements, have made its data invaluable to the remote sensing community at large. Access of various datasets from QuikSCAT can be found at podaac.jpl.nasa.gov. QuikSCAT is currently in its Post Wind Mission phase, whose main purpose is to calibrate the RapidSCAT scatterometer. QuikSCAT is scheduled to be decommissioned in Fall of 2016.

Parameter	RapidSCAT	QuikSCAT	Unit
Orbital altitude	435	800	km
Antenna size	0.75	1	m
Pulse width	1.0	1.5	ms
PRI	6.0	5.4	ms
Peak radiated power	80	80	W
Incidence angle, 2 beams	49, 56	46, 54	degree
Look angle, 2 beams	45, 50.5	40, 46	degree
Ground-range resolution	0.79, 0.73	0.55, 0.49	km
Azimuth resolution	15.5, 17.3	24.5, 26.0	km
Slant range	600,678	1095, 1242	km
Ground swath	900, 1100	1410, 1800	km
Data window length	1.4	1.8	ms
NE σ^0	-32.8, -31.5	-31.2, -32.2	dB

Table 2.1: RapidSCAT and QuikSCAT Basics Compared [1]

2.3 RapidSCAT

After QuikSCAT stopped spinning in 2009, NASA decided that a Ku-Band scatterometer should be flown on the ISS. To speed up development and save costs, the engineering model of QuikSCAT was modified and used as the replacement scatterometer. The mission was called RapidSCAT. RapidSCAT was launched in the fall of 2014, and is currently aboard the International Space Station (ISS). RapidSCAT is due to be decommissioned in Fall of 2016.

A comparison of some of the basic parameters for QuikSCAT and RapidSCAT can be seen in Table 2.1. Only a few of the parameters of RapidSCAT were modified relative to QuikSCAT, and as a result RapidSCAT is quite similar to QuikSCAT, which facilitates calibration, but with some distinct differences. For instance, RapidSCAT's smaller antenna size and increased look angle of its beams—due at least partially to the much lower altitude of RapidSCAT than QuikSCAT—give RapidSCAT a larger ground swath and measurement area than if the QuikSCAT instrument were at RapidSCAT's . However, as can be seen in Table 2.1, the ground swath is still smaller for RapidSCAT.

RapidSCAT has a unique orbit due to its host platform, the ISS. The orbit is a nonsun-synchronous, non-polar orbit that precesses much more rapidly than QuikSCAT. The



Figure 2.4: Graphs showing the progression of the ISS orbit [1]. The ISS orbit slowly evolves and eventually covers every latitude—for all longitudes—over the course of two months.

overlay of orbits spanning various time periods is shown in Figure 2.4. While generally a precession of an orbit into different local times of day—such as that shown in Figure 2.4—might not be desirable for most applications, this phenomenon allows the measurement of diurnal cycles. Diurnal cycles are processes that vary throughout the course of the day, such as temperature or wind velocity. Every two months, the ISS orbit processes such that RapidSCAT passes above each location between ± 60 latitude at most times of the day, as seen in Figure 2.4. This allows us to look at various parameters as never before, such as modeling how the winds over the ocean vary from month-to-month and season-to-season. While it would be excellent to get a measurement at every location for each time of day, it is not possible with RapidSCAT.



Figure 2.5: Plot of ISS height versus the revolution number for RapidSCAT. The revolution number is the number of orbits since the instrument was placed in orbit.

In addition to the other non-standard orbit qualities, RapidSCAT additionally has a non-Keplerian orbit. Simply stated, this means that calculations become more time consuming. The altitude of the spacecraft also varies significantly over time, which has a big impact on the Xfactor (which is introduced in Section 2.4). A plot of the spacecraft height can be seen in Figure 2.5.

Another attribute of RapidSCAT that makes correlating data between orbits more difficult is the large attitude range when compared to QuikSCAT. Figure 2.6 shows the roll, pitch, and yaw values for the ISS across various orbits. The roll and pitch variations seen here are much larger than QuikSCAT's, which affects Xfactor calculations. The Xfactor is introduced in Section 2.4.

2.4 XfactorRS3

The XfactorRS3 program and equations are referenced at various points in this work and are introduced here. The Xfactor is derived from the Radar Equation following the derivation in [8].



Figure 2.6: Plots of ISS roll, pitch, and yaw for various orbits. QuikSCAT attitudes would appear as lines with 0° slopes.

The monostatic radar equation is defined in [9] as

$$P_r = \frac{\lambda^2}{(4\pi)^3} \sum_{i \in F} \frac{P_{ti} G_i^2 \sigma^o \delta A_i G_{F,i}}{R_i^4},$$
(2.1)

where *i* indicates a specific cell in the footprint *F*, P_t is the power transmitted, P_r is the power received, *G* is the antenna gain, σ^0 is the normalized radar cross-section (normalized backscatter coefficient), *R* is the range, λ is the wavelength, δA is the area of each cell, and G_F is the gain of the slice filter band for the signal. This is computationally expensive, so in order to simplify and reduce computation time, three variables— σ^0 , P_t , and G_p —are assumed constant over the footprint. Implementing this into the equation results in

$$\bar{P}_r = \frac{\lambda^2}{(4\pi)^3} P_t G_p^2 \sigma^o \sum_{i \in F} \frac{\delta A_i G_{F,i}}{R_i^4} = \frac{\lambda^2}{(4\pi)^3} P_t G_p^2 \sigma^o \sum_{i \in F} SRF(i)$$



Figure 2.7: Example integration grid for a RapidSCAT egg and slice 6.

with

$$SRF(i) = \frac{\delta A_i G_{F,i}}{R_i^4},\tag{2.2}$$

where \bar{P}_r is the approximate return power and SRF(i) is the spatial response function. Further expanding the equation we find

$$P_r = \frac{\sigma^0}{X},\tag{2.3}$$

where

$$X = \frac{C_{cal}}{NN_{p,t}} \sum_{i \in F} \frac{P_{t,i}g_i^2 \delta A_i G_{F,i}}{R_i^4} = \frac{C_{cal}}{NN_{p,t}} \sum_{i \in F} SRF(i)$$

and

$$C_{cal} = \left(\frac{\lambda^2}{(4\pi)^3}\right) \left(\frac{G_p^2 L_{cal} P_{cal}}{L_{sys}}\right),\,$$

where X is the XfactorRS3, C_{cal} is a calibration variable, L_{cal} is the product of the calibration losses, L_{sys} is the product of the system losses, and P_{cal} is the calibration power used.

Currently, the radar equation cannot accurately be computed in real time. Thus, the Xfactor is precalculated for various positions and other needed positions are interpolated. An example of a nominal (or standard position) Xfactor table for both an egg and a slice of RapidSCAT can be seen in Figure 2.7.

In the event of an attitude perturbation, it is desirable to have an easy correction scheme so that the Xfactor does not need to be recalculated. After various tests, it was



Figure 2.8: X plotted as a function of Δf . The red line is the 3^{rd} order fitted curve.

found that parameterizing the Xfactor with respect to Δf is a reliable parameterization, where the different data are collected by calculating X for various attitude positions from nominal. Δf is explained in Section 2.5. An example of a relationship between X and Δf can be seen in Figure 2.8.

2.5 Δf

The frequency shift, also known as Δf , is essential to the definition of slices as well as defining the effect of perturbations on the Xfactor. Δf is a combination of range and Doppler bins, where a "bin" is a select set of time delays and frequency ranges. This section explains the mathematical definition of Δf .

 Δf is given as

$$\Delta f = (f_{nom} - f_{pert} + f_{err}) \cdot T \cdot N_{FFT},$$

where

$$f_{nom} = f_{d_{nom}} + R_{constant} \cdot r_{nom},$$

$$f_{pert} = f_{d_{pert}} + R_{constant} \cdot r_{pert},$$

and

$$R_{constant} = \frac{2 \cdot ChirpRate}{SpeedLight}$$

Reorganized this is

$$\Delta f = \left(f_{d_{nom}} - f_{d_{pert}} + \frac{2 \cdot ChirpRate}{SpeedLight} \cdot (r_{nom} - r_{pert}) \right) \cdot T \cdot N_{FFT}.$$

In these equations T is the sample period and N_{FFT} is the number of points in the FFT, which are 2.114e-6 seconds and 1024, respectively, according to the Xfactor code. The *Chirp-Rate* is approximately 250 kHz/ms.

2.6 Eggs and Slices

Throughout this work "eggs" and "slices" are referred to. These have relation to the spatial response function. The spatial response function, eggs, and slices are explained in the next few paragraphs.

The spatial response function (SRF) is a measure of the power returned from each location on the surface being imaged as seen by the instrument, and is determined according to 2.2. Various aspects, which are primarily due to the radar equation, feed into determining the spatial response function, given in Equation (2.1), including: antenna response at transmit, antenna response at receive, topography of the imaged surface, and range gate clipping. When all these aspects are taken into account, the spatial response can be calculated, as shown in the left image of Figure 2.9. The left image is a plot of what is considered the spatial response of the sensor, and also the image from which the egg is derived.

The egg is the contour of the main lobe response within -6 dB of the maximum power return, and is shown by the black contour in Figure 2.9. This value is commonly used for low-resolution, low-noise imaging as well as for calibration. For QuikSCAT and RapidSCAT, the egg return power is not measured directly, but instead is a sum of the inner 8 of 12 slices.

A slice is the power return over a specific set of frequency shift bins (Δf) of the dechirped echo. An overview of Δf is given in Section 2.5. On QuikSCAT and RapidSCAT there are range bins—which differentiate locations on the ground by looking at the return



Figure 2.9: The RapidSCAT Xfactor egg and slice processor responses. In practice, the egg response is calculated using a sum of the slice responses.



Figure 2.10: All azimuth angles, both beams, all valid inner slices at a position near the equator. The spacecraft is moving northeast.

time—and frequency bins, which differentiate locations on the ground utilizing the Doppler effect. A slice, in essence, is a certain combination of range and Doppler bins. Plotting the slice contours for 36 different azimuth angles produces the plot shown in Figure 2.10.



Figure 2.11: The sampling and reconstruction process. The sampling process is already determined by instrument design and orbit geometry. For reconstruction, $\sigma^0[x, y]$ is represented from the QuikSCAT measurements, subject to parameters such as pixel size and reconstruction algorithm. [7]

2.7 Reconstruction

Reconstruction is the ultimate purpose of accurately determining the SRF slice contours. Reconstruction is a process that takes individual measurements and tries to recreate the original backscatter image on a fine resolution grid. This process is outlined in Figure 2.11.

An example of this process is shown in Figure 2.12. In the left image in this plot, the boresight locations of the σ^0 measurements on the surface are shown. In the right side image, the -6 dB contours of the SRF are shown. These approximately determine which parts of the surface contribute to the measurement's value. Using this information allows the reconstruction to infer a finer grid [23].



Figure 2.12: Examples of grid algorithm (left) and ave algorithm (right). Using a knowledge of the contours of the egg contours (green ellipses), the ave algorithm can determine information on a finer grid than the grid method, which is also known as the "drop in a bucket" method. [7]

Chapter 3

Contour Preparation

This section describes the methodology used in finding a parameterization for slice contours. Determining a reliable method of slice parameterization improves computation speed of image reconstruction methods while maintaining reasonable accuracy. For an overview of image reconstruction and the role of eggs and slices in reconstruction see Section 2.7.

First, the various data used are introduced in Section 3.1. The calculation of slice contours is explained in Section 3.2, and filtered in Section 3.3. After filtering, the contours are correlated to one another as described in Section 3.4, which enables slice approximation according to the method explained in Section 3.5. After approximation, the method of selecting valid contours is introduced in Section 3.6. The parameterization is calculated in Section 4.1. Lastly, the results of the parameterization are compared to the original slice contours in Section 4.2.

3.1 Data Used

This section covers the data used in producing the slice contours used for parameterization as well as providing an introduction to L1B data. Section 3.1.1 explains what data is used and produced as well as metadata generated from X factor RS3, while Section 3.1.2 introduces the L1B dataset.

3.1.1 Xfactor

The XfactorRS3 program is essential in generating the data used in this research. The main data required for this research are the slice integration grids, such as the one shown in Figure 3.1, which is metadata produced while running XfactorRS3. The integration grids are defined in the Brigham Young University XfactorRS3 program. The antenna pattern, range and Doppler tracking tables, and ephemeris and attitude data are evaluated for each location and time of interest.

The NASA Jet Propulsion Laboratory (JPL) supplies the inner and outer beam antenna pattern files which are read in *XfactorRS3*. The antenna gain patterns seen in Equation (2.1) are assumed constant across all orbits and are essential for determining the SRF. The input attitude data—the roll, pitch, and yaw—can be seen in Figure 2.6. The x, y, and z positions and velocities are given by the ephemeris data file.

Another important set of information includes the Doppler and range tracking tables. As the ISS orbit changes, the Doppler frequency range as well as gate timing range varies. If uncompensated, clipping of the main lobe of the processor response can occur resulting in poor estimates of the Xfactor as well as an inaccurate estimate of the slice contours. In order to compensate, range and Doppler tracking tables are created and used to change the range-gate timings and Doppler frequencies being used. Using these tables avoids clipping of the main lobe of the processor response and allows accurate calculations of the Xfactor.

With this information, the *XfactorRS3* program is run. The *XfactorRS3* program produces the integration grids for eggs and slices from which the contours are calculated.

A total of 45 different orbits are used to provide the integration grids used in this thesis. The orbits are spread in time with different attitude and ephemeris data for each revolution and selected based on having different Doppler tracking tables.

3.1.2 L1B

L1B is one of the data formats for QuikSCAT and RapidSCAT data distributed by JPL. The L1B data layout, which is explained hereafter, is representative of the data from which a parameterization is made.

 Table 3.1: NASA Processing Levels [2]

Data Level	Description
Level 0	Reconstructed, unprocessed instrument and payload data at full reso-
	lution, with any and all communications artifacts (e.g., synchronization
	frames, communications headers, duplicate data) removed.
Level 1A	Reconstructed, unprocessed instrument data at full resolution, time-
	referenced, and annotated with ancillary information, including radio-
	metric and geometric calibration coefficients and georeferencing parame-
	ters (e.g., platform ephemeris) computed and appended but not applied
	to Level 0 data.
Level 1B	Level 1A data that have been processed to sensor units, e.g. including
	QuikSCAT.
Level 2	Derived geophysical variables at the same resolution and location as
	Level 1 source data.
Level 3	Variables mapped on uniform space-time grid scales, usually with some
	completeness and consistency checks.

The basic data levels of interest, as defined by NASA, are shown in Table 3.1. L1B provides the highest level processed data available to the public before wind processing. The developed parameterization is based on the parameters of the RapidSCAT L1B file.

An explanation of the parameters in a QuikSCAT L1B file is given in [3], which is similar to the L1B files of RapidSCAT. For reference, the L1B parameters of interest in this thesis are given with their descriptions in Table 3.2, where descriptions are obtained from [3].

3.2 Find the Contours

Once the slice integration grids have been determined and the SRF calculated, the contours are then determined. A contour is defined as the -6 dB contour of the normalized SRF for a given slice. The SRF is introduced in Section 2.6. The contours are determined using linear interpolation between grid points and then finding where the interpolated lines intersect the contour level using the *matplotlib.pyplot.contour* package in *Python 3.5*. After finding the intersection points, the points are then arranged into contours. A contour overlaid

Table 3.2: Se	lected L1B	File Parameters	from	[3]
---------------	------------	-----------------	------	-----

L1B Parameter	Description
rev_number	An assigned revolution number based on the spacecraft orbital his-
	tory. Each revolution begins and ends at the southernmost orbital
	latitude.
orbit_time	The spacecraft time which the SeaWinds Command and Data Sub-
	system (CDS) assigns to the telemetry data packet. Orbit_time
	records the clock counts as they are reported in the telemetry
	packet.
sc_lat	The geodetic latitude of the location on the spacecraft nadir track.
sc_lon	The east longitude of the location on the spacecraft nadir track.
sc_alt	The spacecraft nadir altitude relative to the reference ellipsoid at
	the time specified in data element frame_time.
x_pos	The X component of spacecraft position in the Earth Centered Ro-
-	tating (ECR) coordinate system.
y_pos	The Y component of spacecraft position in the Earth Centered Ro-
	tating (ECR) coordinate system.
z_pos	The Z component of spacecraft position in the Earth Centered Ro-
	tating (ECR) coordinate system.
x_vel	The X component of spacecraft velocity in the Earth Centered Ro-
	tating (ECR) coordinate system.
y_vel	The Y component of spacecraft velocity in the Earth Centered Ro-
	tating (ECR) coordinate system.
z_vel	The Z component of spacecraft velocity in the Earth Centered Ro-
	tating (ECR) coordinate system.
roll	The angular rotation about the y-axis of the ISS spacecraft fixed
	right handed coordinate system.
pitch	The angular rotation about the x-axis of the ISS spacecraft fixed
	right handed coordinate system.
yaw	The angular rotation about the z-axis of the ISS spacecraft fixed
	right handed coordinate system.
frequency_shift	The shift in the baseband frequency of a scatterometer pulse due
	to various measurement conditions.
antenna_azimuth	The calculated azimuth of the SeaWinds antenna at the instant
	when the radiation from the scatterometer pulse impacts the earths
	surface.
x_factor	The conversion factor from the energy measurement to the nor-
	malized radar cross section value for each of the slices of a whole
	scatterometer pulse.



Figure 3.1: The RapidSCAT Xfactor slice processor response at a particular azimuth angle and orbit position for Slice 6.

on the integration grid from which it is derived can be seen in Figure 3.1. Figure 3.2 shows an example of the slice contours for a particular pulse for the inner beam.

3.3 Filter the Contours

The calculated slice contours are filtered to mitigate the effects of quantization error inherent in digitized data. When generating the contour for a slice or egg the contour locations are quantized to grid centers. This induces noise in the reported location. In order to reduce this noise—which encourages greater accuracy in parameterization of contours while enabling a more conservative estimate of the slice contours, a Blackman filter of length 7 is applied to each array of km east and km north locations of the -6 dB contours. This filter length is sufficiently long to reduce the effects of noise and quantization error without



Figure 3.2: An example of the inner 8 out of the 12 slices for RapidSCAT. Slices 1 and 12 are rarely used as they are considered "guard bands". The axes "km north" and "km east" are defined with respect to the boresight of the antenna. Measurements taken at 0 degrees azimuth, ascending pass of equator, with the spacecraft heading northeast.

significantly altering the shape of the contour. A Blackman filter is used due to the wide main lobe and the low side lobes in the Fourier transform, which in essence means the filter accepts slow changes but resists (smooths) fast (noisy) changes. The original contour, along with its filtered version, can be seen in Figure 3.3. For a more in depth view, including turning functions and derivatives, see Figure 3.4.



Figure 3.3: An example slice contour and its filtered version. Notice that the filtered contour closely resembles that of the original.

3.4 Relation of Contours

To approximate the slice contours, the filtered contour is compared to a pre-defined standard contour. In order to correlate two contours, we must determine the orientation difference and the corresponding points on each contour.

In order to determine the rotation angle of a slice contour, first a reference curve must be chosen. For this work, the 0 orbit time, 0° azimuth contours from an idealized RapidSCAT orbit are used as the reference case. This enables accurate correlation of points between all the various RapidSCAT orbits, which vary greatly in orbital and attitude parameters. The method used to determine slice rotation is the polygon comparison method (PCM) outlined in Appendix A. The algorithm, originally developed in [24], is explained in Appendix A and



Figure 3.4: Plots of contours (blue) and filtered contours (red). (a) the contours plotted in latitude and longitude; (b) the turning functions (see Appendix A); (c) the first derivative of the turning function for the original contour; (d) the first derivative of the turning function for the second derivative of the turning function for the original contour; (f) the second derivative of the turning function for the filtered contour.

the reader is encouraged to review that section. The PCM method additionally gives the correlated starting points between the contours.

The PCM works well and closely matches human intuition of the best fit, but that is not always desirable. The best fit is sometimes at an angle that is 180° off from the rotation angle we want it compared to. To account for this possibility, a weighting is applied to the PCM that favors rotations that are closer to a reference angle, which is chosen to be an estimate of the expected rotation angle.

The calculation of the expected reference angle is dependent on two factors: the orbit position and the azimuth angle. Figure 3.5 illustrates the rotation angle difference with respect to antenna azimuth angle and shows that determining the reference angle is not as simple as many would assume. The wide range of rotation angle values across orbit positions for a given azimuth angle can be understood through an analysis of Figure 3.6. The figure shows an example of the effect of orbit position for a single azimuth angle, and in fact is essential in producing the figure shown in Figure 3.5. The effect of azimuth on Figure 3.5 is negligible.

3.5 Approximation

The original number of points used to approximate a slice contour varies, though a good-quality valid contour generally consists of approximately 400 points depending on the slice and pulse in question. Parameterization requires that we relate comparable points in different contours in order to ascertain the parameters of interest and to determine consistent, reliable contour descriptions. To this end, an estimate of each slice contour needs to be made with a consistent number of points (arbitrarily chosen to be 36, which also matches the number of azimuth angle measurements) and in a consistent fashion.

This is a more difficult task than originally anticipated due to the change in slice shape for different azimuth angles. Figure 3.8 shows the effects of rotation on slice 6. It should be noted that the results shown in Figure 3.8 are a best case scenario as they are calculated based on an idealized ISS orbit with ideal attitude parameters.

For QuikSCAT, the method developed by Jvan S. Ashcraft in [25] was used, which employs an approximation based on a polar coordinate defined version of the contour. Instead



Figure 3.5: Rotation angle for all 32 orbit times and 36 azimuth angles in a given orbit for the inner beam. The red point indicates the 0 orbit time, 0 azimuth angle location. Each vertical set of points span the 32 orbit times for that angle, and more in depth view of the vertical set of points can be seen in Figure 3.6.

of that method, a new method based upon the polygon comparison method explained in Appendix A is used because the method used in [25] does not compensate for orbit and attitude changes present in RapidSCAT's orbits. The first step is to create an ideal orbit and calculate, using XfactorRS3, the integration grids for at least one position, that being the 0° azimuth, 0 orbit time case. This is essential since in order to use the method outlined in Appendix A there needs to be a second contour, which in order to ensure accurate correlation of points between different curves, means that we need a reference set of slice shapes. To satisfy this requirement, the aforementioned ideal slice contours at 0° azimuth, 0 orbit time are used as reference contours.



Figure 3.6: A characterization of the effect of orbit time on rotation angle. The apparent sinusoidal nature is a result of the change in direction of the ISS with respect to ground as it orbits around Earth.

After the ideal case slices have been calculated, the contour points need to be aligned. When calculating contours, the contour arrays rarely start at the same relative position in the contour. A useful metric to see how similar the curves are and to determine the orientation of one with respect to the other is needed. The polygon comparison algorithm explained in Appendix A fulfills these requirements. While the main purpose of Appendix A is to define a metric for the difference in shape between two polygons, the method has the side benefits of finding the correct orientation and correct position of each curve that minimizes the polygon distance metric introduced in [24]. The start point of the ideal curve is fixed in order to speed computation as well as to enable easy and accurate correlation. The correlated starting index of each array—in addition to the rotation compensated curves—can be seen in Figure A.3.



Figure 3.7: The residual rotation angle in degrees for a slice after compensating for azimuth angle rotation when compared to the 0 orbit time, 0° azimuth angle case. The red point indicates the 0 orbit time, 0° azimuth angle case.

3.5.1 Determining Important Points

In approximating the shapes of slices, one of the essential operations is to ensure that critical defining points are determined in a consistent manner between slices so that accurate correlation is possible. To this end, the method is to use the turning function and look for the first crossings at intervals along the curve, e.g., for 36 approximation points an point is chosen every 10° of rotation. The results of this approximation can be seen in Figure 3.9. Notice that this method of approximation handles both sharp corners and gentle slopes very well.

It should be noted that this method of approximating shapes is only useful if the function is a monotonic increasing function. The slice contours meet this specification in nearly all cases, and those that are not monotonic increasing are usually erroneous slices.


Figure 3.8: These plots are produced from an ideal orbit of RapidSCAT at an orbit time of 0. One interesting thing to note is that the contours for $Az = 100^{\circ}$ are still not well aligned according to human intuition, though the least squares orientation is certainly maximized.



Figure 3.9: 36 approximation points evenly spaced according to θ .

It is also for this reason that the original slice contour needs to be filtered. The effects of filtering on the turning function can be seen in Figure 3.4. From the set of defining points, a smaller subset is chosen on which to focus.

3.6 Determine Valid Slices

In parameterizing contours it is necessary to consider that not all contours are valid. A reliable method of determining slice validity is presented here.

As visually verifying all the accurate slices is impractical. In order to programatically determine validity of a slice the polygon metric introduced in [24] is used. This metric is used based on the fact that if the slice has a shape that is greatly dissimilar from the ideal then the slice is not valid. It should be noted that the data used for determining valid slices for use in parameterization is not important so long as it gives a reliable estimate.



Figure 3.10: The method for determining important points graphically demonstrated. Here 8 approximation points are used to define the contour shown above. The black, green, and red points are all approximation points. The black and green dots are given in order to correlate the points on the two plots.



Figure 3.11: Filtering noisy slices based on g-factor. It is evident that the g-factor method filters more slices than slicemax (see Figure 3.13)., and a visual comparison with some of the slices the Slice Max method shows that some valid slice contours are being discarded.

One key parameter for determining a slice's validity is the g-factor. The g-factor is a measure of the range-gate clipping, or the amount of the egg that is not included in the power returned. If the valid contours, defined as those having a certain range-gate clipping threshold, for 36 uniform azimuth angles and 1 orbit position are plotted a figure similar to Figure 3.11 results. Originally this plot was made with a threshold of 0.9999, which allows almost no range-gate clipping, but by inspection it was apparent that valid slices were being thrown out.

Plotting the g-factor with respect to the polygon metric for all slices and for a single slice can be seen in Figure 3.12. From Figure 3.12 we conclude that the g-factor is insufficient in determining if slices are valid. One reason the g-factor alone does not accurately determine

the validity of slices is due to the fact that the g-factor is the fraction of power from the egg that is present in the used frequency bins, but a particular slice may not include the egg power lost due to clipping. Applying a more stringent requirement on g-factor results in losing all but 2 or 3 slices for some aft azimuth angles.

Another option considered for determining the validity of slices is to use the maximum gain of the SRF, which here is also referred to as slicemax. If the slicemax is too low then parts or all of the egg are below the noise floor, meaning that the signal-to-noise ratio (SNR) are too poor to reliably use the data. Basing slice selection by setting a threshold based on the slicemax results in the plot shown in Figure 3.13, which is made from the same set of pulses as Figure 3.11. As can be seen, Figure 3.13 has more slices than Figure 3.11 and the slices contour shapes still resemble the expected valid slice shapes.

Furthermore, comparing the slicemax to the polygon metric from [24] for all slices and for a single slice results in the plots shown in Figure 3.14. In Figure 3.14 it can be seen that contours that are greatly dissimilar to the ideal contour—contours that have a large l_2 distance—are appear to be limited to slicemax values below -80 dB. Using a combination of thresholding slicemax, g-factor, and the polygon comparison metric, a set of valid slices is identified, which are then used for parameterizing slices.



Figure 3.12: A scatter plot of the g-factor as a function of l_2 polygon distance for the over 94,000 pulses analyzed, with points plotted for each slice (top) and for only slice 5 (bottom) in each pulse. A histogram is overlaid where a bin has a count greater than 10. The top left corner shows slivers of color where most of the measurements are located.



Figure 3.13: Filtering noisy slices based on slice max amplitude.



Figure 3.14: A scatter plot of the maximum slice value in the spatial response function as a function of l_2 polygon distance for the over 94,000 pulses analyzed, with points plotted for each slice (top) and for only slice 5 (bottom) in each pulse. A histogram is overlaid where a bin has a count greater than 10.

Chapter 4

Parameterization and Results

4.1 Parameterization

After all contours have been approximated, and valid slices have been determined, a parameterization can be calculated. The method chosen for performing the parameterization is the least-squares linear fit of polynomial order 2. Originally a fit for all azimuth angles for each slice was created, but due to the error of the fit a parameterization is created for each slice for each azimuth angle. An explanation of the calculation follows.

The base algorithm for a linear fit is of the form

$$Ax = b$$

with

$$A = \begin{bmatrix} p_1[1] & p_2[1] & \dots & p_m[1] \\ p_1[2] & p_2[2] & \dots & p_m[2] \\ \vdots & \vdots & \ddots & \vdots \\ p_1[n] & p_2[n] & \dots & p_m[n] \end{bmatrix}$$
$$b = \begin{bmatrix} kmeast[1] & kmnorth[1] \\ kmeast[2] & kmnorth[2] \\ \vdots & \vdots \\ kmeast[n] & kmnorth[n] \end{bmatrix},$$

and

where p_m is the m-th parameter row vector, *kmeast* represents the km east row vector from the egg center for the approximation points, *kmnorth* represents the km north row vector from the egg center for the approximation points, and *n* selects the n-th pulse from the pulses being used for the parameterization. The parameters represented in p_m include: roll, pitch, yaw, g-factor, polygon distance, orbit position, slant range of boresight, slicemax, xfactor for the egg, xfactor for the slice. In order to ensure an accurate linear approximation, the data used for the parameterization is defined as the deviation from the mean for the data. The mean referred to is the mean calculated for a particular variable, azimuth, and slice combination.

The m-th parameter row vector, p_m , is

$$p_m = \begin{bmatrix} v_{centered}^0 & v_{centered}^1 & \dots & v_{centered}^O \end{bmatrix},$$

where O is the order of the variable and $v_{centered}$ represents the data deviation from the mean. For this thesis the order was arbitrarily chosen to be 3.

4.1.1 Calculate Weightings

In order to solve for the weightings, x, the pseudo inverse must be calculated. The pseudo inverse is defined as

$$x = (A^T A)^{-1} A^T b,$$

where A and b have been defined previously. It is important to note that the pseudo inverse only exists if the inverse of $A^T A$ exists, which means that A must be full rank, which in the overdetermined case which means the columns of A must be independent. For certain azimuth angles, slice combinations the pseudo inverse cannot be calculated. In these cases, the restraints on what constitutes a valid slice are relaxed which in turn increases the number of slices used in parameterization. If the relaxed restraints still do not yield enough slice contours to work with, then the parameterization cannot be calculated. Otherwise, a set of model coefficients are computed that enable computation of the slice contour at an arbitrary orbit position and azimuth angle given the input parameters. To see the results of this parameterization, see Section 4.2.

4.2 Analysis of Parameterization

In analyzing the results of parameterization we first consider the quality of the slices used for parameterization. The quality for the inner and outer beams of RapidSCAT can be seen in Figure 4.1 and Figure 4.2, respectively. For the inner 8 slices the results are quite good for both inner and outer beams, with inner having more parameterization difficulties than the outer beam.

The metric used for determining the accuracy of the parameterization is explained here. First, for each contour used in the parameterization the root mean square (RMS) error is calculated for the l_2 distance between the approximation points and the points estimated using the parameterization. Afterwards, the mean of the curve l_2 RMS error for all curves utilized in the parameterization is calculated. The results of this calculation for the inner and outer beams is shown in Figure 4.3 and Figure 4.4, respectively. A white section means that a parameterization could not be calculated for that case. Here we see that in general the RMS error across the inner 8 slices averages around 1.5 km l_2 RMS error for the inner beam and a little higher for the outer beam. This is an acceptable level of performance.

The estimations produced are, in general, visually close estimates if the estimation flags shown in Figures 4.1 to 4.2 are "Good", and inaccurate estimates if "Warning". Examples of a good and a bad estimate can be seen in Figure 4.5 and Figure 4.6, respectively.



Figure 4.1: Status of each parameterization for the inner beam. Green indicates that the parameterization used incorporates the slice selection limits mentioned in Section 3.6, orange indicates the slice selection limits had to be loosened in order to create a usable parameterization, and yellow indicates that creating a slice parameterization was not possible due to range gate clipping.



Figure 4.2: Status of each parameterization for the outer beam. Green indicates that the parameterization used incorporates the slice selection limits mentioned in Section 3.6, orange indicates the slice selection limits had to be loosened in order to create a usable parameterization, and yellow indicates that creating a slice parameterization was not possible due to range gate clipping.



Figure 4.3: Average of l_2 RMS errors in km for each curve used for each parameterization for the inner beam.



Figure 4.4: Average of l_2 RMS errors in km for each curve used for each parameterization for the outer beam.



Figure 4.5: A good estimate obtained using parameterization.



Figure 4.6: A bad estimate obtained using parameterization.

Chapter 5

Conclusion

A knowledge of the spatial response function (SRF) of the RapidSCAT σ^0 measurements allows a user to reconstruct a high resolution image from densely sampled measurements. While a precise evaluation of the SRF for every given point would produce the highest resolution, calculation of the SRF for every pulse is impracticable as it is too computationally intensive. An alternate approach is to calculate the -6 dB contour for each slice and assume that the return is uniform inside the contour and negligible outside the contour. The contour can then be parameterized to enable a much faster computation of the contours for use in reconstruction.

Previously for QuikSCAT, the parameterization was simple and highly accurate due to QuikSCAT's consistent, easily defined orbit. RapidSCAT's orbit is much less consistent than that of QuikSCAT such that parameterization is less straight forward. Here we review the parameterization method used for RapidSCAT, analyze the results, and propose future work to be done.

The basic process for parameterizing the slice contours is as follows:

- 1. Compute the SRFs for each slice at multiple azimuths and orbit positions,
- 2. Compute the -6 dB contours of each slice
- 3. Filter (smooth) the contours,
- 4. Approximate contours, aligning the contours appropriately,
- 5. Determine valid slices to be used in paramterization,

6. Calculate the parameterization coefficients.

The following paragraphs explain the findings regarding each of these steps, as well as a synopsis of the analysis for the parameterization.

Generation of the SRF for multiple orbits is done using the *XfactorRS3* program, developed at BYU. As inputs, real-world data from the ISS and RapidSCAT are passed into *XfactorRS3*. The outputs of the program, for the single output mode, include the spatial response function, Xfactor, and various other instrument parameters, from which the slice contours are calculated and the parameterization defined.

After the SRFs are generated, the contours are calculated. To do this the maximum SRF slice response is found. Using the maximum slice SRF, the contour -6 dB down from the slice maximum SRF is calculated utilizing the *matplotlib* package in *Python 3*. The contour generated is noisy because of quantization error and the imperfect linear estimation features.

In order to compensate for the noisy contour locations, the contours are filtered. A Blackman filter of length 7 is used to smooth while maintaining the general shape of the contours. This aids the accuracy of the parameterization.

Having filtered the contours, the next step is to reduce the number of points along the contour. The number of contour defining points changes for each pulse and slice. To approximate the contours and relate them one to another, the polygon comparison method is used to compute 36 contour approximation points.

Valid slices are determined using thresholds on a combination of slice maximum SRF, slice gfactor, and polygon l_2 distance. Details are elaborated on in Section 3.6.

Having found the approximation points and valid contours, a linear least-squares parameterization based off of the equation Ax = b is computed. The matrix A has dimensions of mx25 where m is the number of contours used in parameterization, which is between 2 and 1312 for this thesis. In addition, x is 25xn where n is the number of approximating points, and b—which represents km east or km north—is mxn. The columns of this include 1 column of ones (the zero-th order) and 3 columns (for orders 1, and 2, 3) each of roll, pitch, yaw, g-factor, orbit position, slant range, Xfactor of the egg, and Xfactor of the slice.

The estimates are visually close for parameterizations that are classified as "good" according to the status estimation flag from Figures 4.1 to 4.2. These parameterizations

can be used to greatly increase the speed of contour estimation for use in reconstruction algorithms.

5.1 Future Work

There are a number of things that could be done to improve the results. Azimuth rotation due to change in direction of the spacecraft is not currently well accounted for in the code. The contours to be approximated are place on a km east, km north grid that is defined with regard to latitude-longitude. In order to better account for the variation in spacecraft direction, the contours could be rotated to compensate for the change in spacecraft direction, which could simplify the parameterization, and then the contours can be rotated back to their original position. This can be expected to produce a smaller spread of points, which may be easier to approximate. In addition, rotation could compensate for rotation due to spacecraft direction, which the current code does not. A variation on this would be to define the integration grid in *XfactorRS3* in terms of along-track and cross-track km. This should negate the need for a rotation matrix.

Another method to improve the contour parameterization is to include more L1B parameters in the parameterization. Due to the non-linear variation of the slice contours, it is likely not possible to have an exact parameterization with a linear model. However, the more data that can be used in the parameterization the more accurate the estimation becomes. Various parameters including spacecraft position in x-y-z, velocity in x-y-z, latitude, longitude, and height—as well as various other parameters available in L1B files—could be used as parameters to improve the parameterization.

Yet another possible method to improve approximation is to use the Contour Encircling Approximation (CEA) method developed. The CEA method is explained in Appendix B. The advantage of this method—in contrast to the polygon comparison method from Appendix A—is that CEA works to maximize the encompassed area, which indicates that the most critical defining points are used in the approximation.

Another option is to generate a "super table". The premise of a super table is to generate a set of contours based on a certain combination of parameters. The parameters would be varied in set amounts in all possible combinations for a set range. With this set of tables, the user would simply need to find the tables that most closely match the parameters of the pulse in question (a simple lookup hierarchy being set up to increase search speed) and interpolate between the two. This would be a memory intensive method, but would likely produce a good fit.

REFERENCES

- [1] N. J. P. Laboratory. RapidSCAT. [Online]. Available: http://winds.jpl.nasa.gov/ missions/RapidScat/
- [2] Data processing levels for eosdis data products nasa science. National Aeronautics and Space Administration. [Online]. Available: http://science.nasa.gov/earth-science/ earth-science-data/data-processing-levels-for-eosdis-data-products/
- [3] Level 1B Data Software Interface Specification (SIS-2), SeaWinds Processing and Analysis Center (SeaPAC), January 2000, rev. A.
- [4] QuikSCAT_high.jpg (24003452). Jet Propulsion Laboratory. [Online]. Available: https://winds.jpl.nasa.gov/files/winds/QuikSCAT_high.jpg
- [5] M. W. Spencer, C. Wu, and D. G. Long, "Improved resolution backscatter measurements with the SeaWinds pencil-beam scatterometer," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 38, no. 1, pp. 89–104, 2000.
- [6] W. Commons. (2014) Heliosynchronous orbit. File:Heliosynchronous orbit.svg. [Online]. Available: https://commons.wikimedia.org/wiki/File:Heliosynchronous_orbit.svg
- [7] R. D. Lindsley, "Enhanced-resolution processing and applications of the ascat scatterometer," Ph.D. dissertation, Brigham Young University, 2015.
- [8] I. S. Ashcraft, D. G. Long, A. Anderson, S. L. Richards, M. Spencer, and B. Jones, "σ⁰ retrieval from SeaWinds on QuikScat," in SPIE's International Symposium on Optical Science, Engineering, and Instrumentation. International Society for Optics and Photonics, 1999, pp. 171–178.
- [9] F. Ulaby, D. G. Long, W. Blackwell, C. Elachi, A. Fung, C. Ruf, K. Arabandi, H. Zebker, and J. van Zyl, *Microwave Radar and Radiometric Remote Sensing*. Ann Arbor: University of Michigan Press, 2014.
- [10] F. Naderi, M. Freilich, and D. G. Long, "Spaceborne radar measurement of wind velocity over the ocean-an overview of the NSCAT scatterometer system," *Proceedings of the IEEE*, vol. 79, no. 6, pp. 850–866, 1991.
- [11] R. Atlas, R. N. Hoffman, J. Ardizzone, S. M. Leidner, J. C. Jusem, D. K. Smith, and D. Gombos, "A cross-calibrated, multiplatform ocean surface wind velocity product for meteorological and oceanographic applications," *Bulletin of the American Meteorologi*cal Society, vol. 92, no. 2, p. 157, 2011.

- [12] J. Ballantyne and D. G. Long, "A multidecadal study of the number of antarctic icebergs using scatterometer data," in *Geoscience and Remote Sensing Symposium*, 2002. *IGARSS'02. 2002 IEEE International*, vol. 5. IEEE, 2002, pp. 3029–3031.
- [13] D. G. Long, J. Ballantyne, and C. Bertoia, "Is the number of icebergs around Antarctica really increasing?" *Eos Trans. AGU*, vol. 83, pp. 469–480, 2002.
- [14] K. Stuart and D. G. Long, "Tracking large tabular icebergs using the SeaWinds Ku-band microwave scatterometer," *Deep Sea Research Part II: Topical Studies in Oceanography*, vol. 58, no. 11, pp. 1285–1300, 2011.
- [15] S. V. Nghiem and W. Tsai, "Global snow cover monitoring with spaceborne Ku-band scatterometer," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 39, no. 10, pp. 2118–2134, 2001.
- [16] E. Schanda, C. Matzler, and K. Kunzi, "Microwave remote sensing of snow cover," *International Journal of Remote Sensing*, vol. 4, no. 1, pp. 149–158, 1983.
- [17] Q. P. Remund and D. G. Long, "Sea ice extent mapping using Ku band scatterometer data," *Journal of Geophysical Research: Oceans*, vol. 104, no. C5, pp. 11515–11527, 1999.
- [18] W. N. Meier and J. Stroeve, "Comparison of sea-ice extent and ice-edge location estimates from passive microwave and enhanced-resolution scatterometer data," Annals of Glaciology, vol. 48, no. 1, pp. 65–70, 2008.
- [19] R. D. Lindsley and D. G. Long, "Estimating surface oil extent from the deepwater horizon oil spill using ascat backscatter," in *Radar Conference (RADAR)*, 2011 IEEE. IEEE, 2011, pp. 559–563.
- [20] —, "Mapping surface oil extent from the deepwater horizon oil spill using ASCAT backscatter," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 50, no. 7, pp. 2534–2541, 2012.
- [21] V. Wismann, M. Gade, W. Alpers, and H. Huhnerfuss, "Radar signatures of marine mineral oil spills measured by an airborne multi-frequency radar," *International Journal* of *Remote Sensing*, vol. 19, no. 18, pp. 3607–3623, 1998.
- [22] W. Tsai, M. Spencer, C. Wu, C. Winn, and K. Kellogg, "SeaWinds on QuikSCAT: sensor description and mission overview," in *Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International*, vol. 3. IEEE, 2000, pp. 1021– 1023.
- [23] D. S. Early and D. G. Long, "Image reconstruction and enhanced resolution imaging from irregular samples," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 2, pp. 291–302, 2001.
- [24] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, and K. Kedem, "An Efficiently Computable Metric for Comparing Polygonal Shapes," *IEEE Transactions on Pattern Anal*ysis and Machine Intelligence, vol. 13, no. 3, pp. 209–216, 1991.

[25] I. S. Ashcraft, "Shape of slices for use in scatterometer image reconstruction," Brigham Young University, Microwave Earth Remote Sensing Laboratory, 459 CB, Provo, UT, 84602, Tech. Rep., June 1998, internal Report.

Appendix A

Comparing Polygonal Shapes

Throughout this text, contours of slices—the polygons used in this work—are compared to each other. The original methods used for comparison of slices for QuikSCAT, as documented in [25], are ineffective for RapidSCAT. A new method for comparing contours is developed in this thesis based on the metric and algorithm defined in [24] for comparing polygons. The rest of this appendix give an explanation of the methods outlined [24], as well a more verbose version of the motivation.

Previously, QuikSCAT slice -6 dB contours were parameterized to compensate for slice rotation, transformed to compensate for warping due to differences between azimuth angle of the instrument and the actual slice rotation, and approximated using eight points defined using a polar definition of the slice contour, as documented in [25]. While the method is effective for parameterizing QuikSCAT slices, it is not as effective for RapidSCAT due to the large attitude and orbit perturbations of RapidSCAT which cause significant slice shape variation. A primary motivation for using the algorithm in [24] for RapidSCAT parameterization is that it compares polygonal shapes irrespective of scaling and rotation, and an optimum least-squares rotation is found in the process. An example of this can be seen in the left plot in Figure A.3, which shows how azimuth angles affect slice shape.

The basic procedure detailed in [24] is as follows: create a normalized turning function for each contour; find the optimum rotation angle that minimizes the least squares error; determine the starting point that minimizes the least squares error between two turning functions. This process is expounded upon in the next few paragraphs.



Figure A.1: An example turning function for the original and filtered contour cases.

A turning function ($\Theta_A(s)$) measures the angle of the counterclockwise tangent as a function of the arclength s, where some reference point O is chosen on A's boundary for reference and A is the contour [24]. An example is shown in Figure A.1. One unique quality of turning functions that makes them especially useful is that turning functions are neither spatially dependent nor dependent on absolute rotation. In addition, in Figure A.1 we also see that the arc length is normalized to account for scaling, making this definition also unaffected by scaling.

With the turning function defined, the algorithm in [24] can be applied. The method finds the minimum l_2 distance between two turning functions. Although this method of defining a contour is sensitive to non-uniform noise, in this project it is reasonable to assume small spatially uniform noise.

First, assume that f(s) and g(s) are turning functions, with s being the normalized arc length. The sum squared error between two turning functions is defined as

$$h(t,\theta) = \int_0^1 (f(s+t) - g(s) + \theta)^2 ds,$$
 (A.1)

where t is the offset arc length and θ is the offset angle. The goal is to find the minimum l_2 distance between two turning functions functions according to

$$d_2(A, B) = \left\{ \min_{t \in [0,1], \theta \in \mathbb{R}} h(t, \theta) \right\}^{1/2}.$$

This is a minimization problem with two variables, which is more computationally intensive to solve than an equation with one variable. In order to obtain an equation with one unknown, the optimum offset angle is used. The optimum offset angle—as derived in [24] is

$$\theta^*(t) = \int_0^1 (g(s) - f(s+t))ds$$

= $\alpha - 2\pi t$, (A.2)

where $\alpha = \int_0^1 g(s)ds - \int_0^1 f(s)ds$. By integrating Equation (A.2) into Equation (A.1), an equation with 1 variable is obtained:

$$d_2(A, B) = \left\{ \min_{t \in [0,1]} h(t, \theta^*(t)) \right\}^{1/2} \\ = \left\{ \min_{t \in [0,1]} \int_0^1 (f(s+t) - g(s) + \theta^*(t))^2 ds \right\}^{1/2}.$$

An example of applying this algorithm to turning functions to achieve the best shape alignments can be seen in Figure A.2, which then produces shape alignments similar to those in Figure A.3.



Figure A.2: An example of applying the algorithm discussed in this section to two different turning functions in order to find the minimum distance metric between the shapes.



Figure A.3: The original contour and rotated slices. Notice how the slice shape changes for different azimuth angles. These contours are from an idealized orbit for RapidSCAT.

Appendix B

Contour Encircling Approximation (CEA) Method

The contour encircling approximation (CEA) method is based on a technique for approximating a contour by determining it's proximity to the contour encircling points. An example of how CEA method works can be seen in Figure B.1. The method defines a circle outside of the contour made of a discrete number of points whose number is less than that of the original contour. The approximation points are determined by finding the point in the original contour whose l_2 distance is smallest for each of the contour encircling points.

This method works remarkably well, with the encompassed area having a root mean square (rms) error of 2.064 km² for a static radius of 500 km. This radius was chosen with the assumption that if a chosen static radius is too small then the CEA points are no longer "encircling" and the approximation is poor. To avoid under-approximating, I chose a radius that was much larger than any expected slice contour radius to avoid this error.

To verify that that was the best choice, I generated estimation contours for multiple radii which were at least as large as the contour being measured. The result is shown in Figure B.2. This figure very clearly shows that the optimum radius—where the optimum radius is determined to be where the contained area is maximized—is closer to the original contour than initially thought. Testing for multiple contours at different azimuth angles, orbit times, and revolutions shows that the optimum radius is dynamic and not static. While the distinction between static and dynamic may appear to be small, it affects the



Figure B.1: The Contour Encircling Approximation (CEA) method. The CEA method works by first encircling the contour to be approximated with a finite number of points evenly distributed in a circle. Then the nearest points to the curve are found by determining the point on the original contour with the smallest l_2 distance for each encircling point. The green lines in the image show the nearest neighbor.

noise introduced by estimating the contour. In the next paragraphs I explain a method to correct for the imprecise estimation of a contour.

I create the histogram shown in Figure B.3 by binning the optimum encircling radius with respect to original contour radius from multiple revolutions for each slice. Figure B.3 shows that the the original radius is proportional to the optimum encircling radius, and the approximate scale factor for all the slices is 1.5776. Table B.1 shows the difference between the dynamic rms encircling radius and the fixed 1.58—where the fixed radius was chosen as the average linear fit for the inner 8 slices—factor for the encircling radius. As can be seen the the aforementioned table, the difference between the rms error for dynamic and fixed radius is minimal and therefore 1.5776 * max(contourRadius) can reliably be used for approximating a slice.



Figure B.2: The plot of the encircling radius and the resulting estimated contour area. Notice that the maximum point on the "fitted area" curve is not at the ends but in the middle.

	Dynamic	Fixed	Difference
Slice 1	2.2120	2.4229	0.2109
Slice 2	2.3241	2.3842	0.0601
Slice 3	2.2363	2.2535	0.0172
Slice 4	2.0996	2.1041	0.0046
Slice 5	1.9678	1.9679	0.0001
Slice 6	2.0408	2.0447	0.0039
Slice 7	2.0760	2.0984	0.0224
Slice 8	2.0067	2.0479	0.0412
Slice 9	2.0359	2.0881	0.0523
Slice 10	2.1611	2.1863	0.0252
Slice 11	2.2070	2.2074	0.0004
Slice 12	2.2246	2.2421	0.0175

Table B.1: RMS Error for Dynamic vs. 1.5776 Fixed Encircling Radius



Figure B.3: A histogram of the original contour radius plotted against the optimum encircling radius for each of the slices. The data is approximately 45 revolutions of data where revolutions with different Doppler tables are selected. The red line indicates the unbiased linear fit that minimizes the least square error.

Appendix C

Analysis of the QuikSCAT Post Wind Mission (PWM) Data

In 2009, a worn bearing caused QuikSCAT to stop rotating and ushered in QuikSCAT's (QS) Post Wind Mission (PWM). The PWM is marked by very narrow tracks that separately measure horizontal and vertical polarizations. While QS cannot rotate, the satellite attitude can be modified to enable the sensor to "look" in different directions.

Currently, QuikSCAT PWM data is being used to help calibrate RapidSCAT as well as other sensors. But, concern has been expressed that QuikSCAT might not be as accurately calibrated for it's post wind mission as it was for its nominal wind mission. One way which can be used to determine the accuracy of QS's calibration is to consider the effect of spin on the X-factor (X).

The Xfactor (X) is part of an approximation of the radar equation, with X relating the power received to radar backscatter coefficient, or radar cross-section (σ^0) . The derivation of the X can be found in Section 2.4

This appendix aims to analyze the Xfactor for QuikSCAT spinning and non-spinning modes to determine if the sensor is accurately calibrated and can be used for RapidSCAT calibration.

The Brigham Young University (BYU) Microwave Earth Remote Sensing (MERS) Laboratory received the plot shown in Figure C.1 from the Jet Propulsion Laboratory (JPL). An annual cycle in average daily σ^0 was expected, but Figure C.1 shows that that is not



Figure C.1: JPL produced figure which shows that the variance in QuikSCAT average daily σ^0 is not an annual cycle as was expected. This suggests that QuikSCAT might be inaccurately calibrated.

the case. The difference between expected and actual annual cycle may mean that the QuikSCAT PWM is inaccurately calibrated.

One possible cause for the variation could be that the Xfactor for the non-spinning case introduces error when compared to the spinning case. To test the theory that the Xfactor may be responsible, XfactorRS3 is used to calculate the difference between the spinning and non-spinning cases. The only required change to the code in order to simulate the non-spinning case is to set the revolutions-per-minute variable to 0. No other changes are required between the original and PWM modes when computing X. The nominal Xfactor (X) "egg" values are calculated and shown in Figure C.2. This figure contains nominal X "egg" images for the (from top to bottom) spin, no-spin, and difference cases for (from left to right) the inner and outer beams. It can be seen from the difference images for the inner and outer beams that the non-spinning case can be roughly translated into a fixed bias, with an accuracy of ± 0.04 dB for both the inner and outer beams.



Figure C.2: Images of nominal X "egg" for the (from top to bottom) spin, no-spin, and difference cases for (from left to right) the inner and outer beams. It can be seen from the difference images for the inner and outer beams that the non-spinning case can be roughly translated into a fixed bias, with a small error that is a function of orbit time and azimuth.

Checking with JPL, the PWM is indeed being compensated for with a bias. With this knowledge and the obtained results of our study, it can be affirmed that the QuikSCAT Xfactor is properly computed for both the spinning and non-spinning cases.