GPS-DENIED NAVIGATION USING SYNTHETIC APERTURE RADAR

by

Tucker Hathaway

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

| | |
|---|---|
| Todd Moon, Ph.D.<br>Major Professor | Jacob Gunther, Ph.D.<br>Committee Member |
| Mark Fels, Ph.D.<br>Committee Member | D. Richard Cutler, Ph.D.<br>Vice Provost of Graduate Studies |

UTAH STATE UNIVERSITY
Logan, Utah

2023

# ABSTRACT

GPS-Denied Navigation Using Synthetic Aperture Radar

by

Tucker Hathaway, Master of Science

Utah State University, 2023

Major Professor: Todd Moon, Ph.D.
Department: Electrical and Computer Engineering

GPS-denied (GPS-D) navigation is the process of determining the precise location of an object without the use of GPS. Due to the ease with which GPS signals can be blocked, spoofed, or jammed, GPS-D navigation is a very active area of research. Many methods of performing GPS-D navigation have been proposed and explored, but this thesis presents a novel approach using synthetic aperture radar (SAR). Whereas other SAR-based, GPS-D methods assume point sources in the scene or the use of onboard data, the method presented in this paper assumes only that the statistical structure of a well-formed SAR image differs from that of a poorly formed image.

To measure the difference in the statistical structure of a SAR image, an image quality measure is first determined. Image quality measure testing is performed using kurtosis and entropy, and kurtosis is shown to be more promising than entropy. Next, a GPS-D navigation algorithm is developed that uses the image quality measure. This algorithm is tested on one simulated dataset and two real datasets, and the results are analyzed. The algorithm is determined to have variable performance depending on the dataset used and the parameters of the algorithm.

(62 pages)

# PUBLIC ABSTRACT

GPS-Denied Navigation Using Synthetic Aperture Radar

Tucker Hathaway

In most modern navigation systems, GPS is used to determine the precise location of the vehicle; however, GPS signals can easily be blocked, jammed, or spoofed. These signals can be blocked by canyons or tall buildings. Additionally, adversaries can transmit signals that either make GPS signals difficult to interpret or that imitate real GPS signals and cause a navigation system to think it is somewhere other than its true location. GPS-denied (GPS-D) navigation is the process of navigating in the absence of GPS.

Many methods of performing GPS-D navigation have been proposed and explored. One such method is to use synthetic aperture radar (SAR) to provide information lost in the absence of GPS. SAR is a technique that uses radar to form images. To create high-quality SAR images, precise location information must be used. This thesis explores using the quality of SAR images to improve position accuracy. First, a method of measuring the quality of a SAR image is determined and tested. Next, a GPS-D algorithm is developed that uses this measure of SAR image quality. The algorithm is then tested on multiple sets of SAR data. The results show that the algorithm performs variably depending on the data set and the parameters of the algorithm.

# ACKNOWLEDGMENTS

I would like to thank my major professor, Dr. Todd Moon, and the other members of my committee, Dr. Jacob Gunther and Dr. Mark Fels. Additionally, I would like to thank Sandia National Lab for funding most of this research and Adam Fletcher for helping immensely with performing the research.

Tucker Hathaway

CONTENTS

# ACRONYMS

| | |
|---|---|
| BPA | back-projection algorithm |
| BYU | Brigham Young University |
| DEM | digital elevation model |
| EKF | extended Kalman filter |
| GPS | global positioning system |
| GPS-D | GPS-denied |
| IMU | inertial measurement unit |
| INS | inertial navigation system |
| InSAR | interferometric SAR |
| LFM | linear frequency-modulated |
| SAR | synthetic aperture radar |
| SDL | Space Dynamics Lab |
| VRW | velocity random walk |

# CHAPTER 1

# INTRODUCTION

Many critical navigation systems largely rely on a global positioning system (GPS) to determine the precise location of an object on Earth. However, GPS signals can easily be blocked, jammed, or spoofed. The signals may be blocked by tall buildings (urban canyons), natural canyons, or other tall objects that obstruct the view of the sky. Additionally, GPS signals can easily be jammed or spoofed. Jamming prevents a navigation system from using GPS signals by transmitting signals of much higher strength in the same frequency spectrum. Spoofing allows an adversary to trick a navigation system into thinking it is in a location other than its true location by transmitting signals that imitate those transmitted by GPS satellites. For this reason, GPS-denied (GPS-D) navigation is a highly active area of research.

GPS-D navigation is the process of navigating without the use of GPS. To successfully navigate in the absence of GPS, the information conveyed by GPS must be replaced using alternative sensors. An inertial measurement unit (IMU) is commonly used to provide approximations of velocity; however, IMUs provide noisy measurements that may include biases. This results in an accumulation of positional errors over time, which necessitates additional sensors to correct these errors. Various sensors are being explored to augment a GPS-D navigation system, such as radar, lidar, optical cameras, and range finders. This paper explores the use of synthetic aperture radar (SAR) as an augmentation sensor.

SAR is a method of forming images using radar pulses. SAR has multiple advantages. It is self-illuminating, meaning it can form images of equal quality regardless of the lighting conditions. Also, SAR operates at frequencies that can penetrate obstructions such as clouds, rain, and smoke.

Previous research on SAR-based GPS-D navigation has been performed by either identifying and tracking point sources in the scene being imaged or by comparing the SAR

data to onboard reference data, such as a digital elevation model (DEM) [1–16]. This thesis presents a novel method of SAR-based GPS-D navigation that does not assume point sources in the scene or use onboard reference data. Instead, the only assumption made is that a well-formed SAR image has a different statistical structure than a poorly formed SAR image.

SAR works by periodically sending pulses from a moving radar platform. The pulses reflect off the scene, and the reflections are recorded for later processing. SAR image formation algorithms then use the received pulses and the relative motion of the platform between pulses to construct an image. One SAR image formation algorithm is the backprojection algorithm (BPA). BPA is a time-domain algorithm that is dependent on the position of the radar platform at the time each pulse is transmitted and received.

This paper exploits the dependence of BPA on radar platform position to infer the best position out of a number of candidate positions. The candidate positions are scored based on the quality of the images formed over each candidate position. There are two objectives of this research. The first objective is to determine a robust measure of SAR image quality. The desired behavior of a good measure is one where a better score corresponds to the image being formed with more accurate position data. The second objective is to develop and test a GPS-D algorithm that uses the image quality measure to correct errors in position data.

This paper is organized as follows. Chapter 2 is a literature review, which highlights how this research differs from previous research in the area. Chapter 3 provides background on SAR and the BP algorithm, discusses the image quality measures considered, develops the GPS-D SAR algorithm, and explains how testing was performed. Chapter 4 shows the results of testing and a discussion of their implications. Finally, chapter 5 presents conclusions.

# CHAPTER 2

# LITERATURE REVIEW

GPS-D navigation is a highly active area of research due to the ease with which GPS can be blocked, spoofed, or jammed. Much work has been performed using SAR to replace the information lost in a GPS-D environment.

Nitti et al. considered an amplitude approach and a phase approach [1]. The amplitude approach compared SAR images with a terrain landmark database. Automatic target detection and recognition was used to find landmarks in the SAR image, then the landmarks were correlated with landmarks in the database. The phase approach used interferometric SAR (InSAR) and comparison with a DEM to infer position information.

Greco et al. presented a similar approach as a part of the SARINA project in [2, 3]. They proposed comparing SAR images to a digital landmark database and using InSAR and land topography when there are no prominent landmarks in the terrain. In [4], they tested the SARINA system on simulated data and showed the system is able to outperform a traditional inertial navigation system (INS) in a GPS-D environment.

In [5, 6], Kauffman et al. developed an algorithm for efficiently detecting and tracking sparse targets using raw SAR data, and they tested the algorithm in simulated scenes. In [7], they studied the feasibility of using this algorithm for navigation, and in [8], they examined the performance when using an extended Kalman filter (EKF) to estimate 2D location. In [9], IMU measurements were added as inputs to the EKF, and the system was increased from 2D to 6D. The simulations performed well, showing that the SAR-aided navigation system performs much better than a standard INS. This research relies on strong point scatterers located uniformly throughout the scene. Next, [10] replaced the original target tracking algorithm with an improved algorithm. The new algorithm has multiple improvements including using an M/N detector that removes the need for tuning the noise level to the specific scene. More testing was done in [11], which showed 1.2 meters of

simulated drift over a 600 second flight using a tactical grade IMU.

Christensen et al. analyzed the effects of position and attitude errors on SAR images using BPA [17]. They demonstrated that BPA-SAR images are sensitive to all types of navigation errors tested. They attempted to characterize the type of error in the image and then infer navigational errors based on the specific type of image error. Ambiguity was found in the association between image error and navigational errors, but this work suggests that formed BPA-SAR images could be used to determine errors in position data.

Lindstrom also explored radar-aided GPS-D navigation assuming prominent point targets [12]. They first developed and validated an INS and EKF that incorporates radar telemetry as an input, showing that radar-aided GPS-D navigation is feasible. Next, they surveyed the effect of different types of navigation errors on formed BPA-SAR images with the hope of being able to use BPA-SAR image errors to infer navigation errors. They determined that the effects on the formed image are not unique to one type of navigation error. Finally, they developed an algorithm that uses known locations of targets in the scene to extract position data and correct for navigation errors. This method resulted in converging and bounded position estimates.

In [13, 15], Quist and Beard used radar to improve navigational errors. They used the Hough transform, which is a data association technique used to identify multiple point scatterers in the terrain. Their research does not rely on known reflectors. Instead, they used environmental point scatterers, like cars or buildings. These papers include successful results from real test flights. Due to the computational complexity of the Hough transform, [14,16] replaced the Hough transform with the recursive-random sample consensus (R-RANSAC) algorithm.

Sjanic and Gustafsson tested a method of simultaneous navigation and auto-focusing in [18]. They used a modified BPA called Fast Factorised Back-projection to form SAR images. They then used two varieties of entropy along with a grid search algorithm to find the minimum entropy value. This value is hypothesized to correspond to the true position of the aircraft. They tested their algorithm on simulated SAR data made of point targets

and showed favorable results.

Most of the previous work presented here focuses on point targets in the scene and comparison with onboard reference data. By contrast, the research presented here does not require any point targets in the scene or the use of reference data. Instead, this research assumes simply that a well-formed SAR image has different statistics than a poorly formed SAR image. This means that the method presented in this thesis may apply to more classes of scenery, such as water, ice, and sand dunes, which lack point targets. This research is most closely related to the research done in [18]; however, this research incorporates different measures of image quality and a different search method. This paper also provides the results of tests on real data.

# CHAPTER 3

# RESEARCH AND DESIGN METHODS

This chapter gives background information on SAR and BPA in Section 3.1. Then, Section 3.2 describes the first objective of this research, which is to determine a measure of SAR image quality. Section 3.3 describes the second objective of this research. The second objective is to create a GPS-D SAR algorithm that uses the image quality measure to aid in navigation in a GPS-D environment. Finally, Section 3.4 outlines how each objective was tested.

The main difference between this research and other research in the area is that this method of GPS-D navigation using SAR does not assume any point sources in the landscape or the use of a DEM. Instead, this research assumes a well-formed SAR image will be either less Gaussian or less uniform than a poorly formed SAR image.

## 3.1 Background

Fundamental to this research are the concepts of SAR and BPA. This section gives background on each of these concepts. For more information on SAR and BPA, see [19].

### 3.1.1 SAR

SAR is a method of forming images using radar and relative motion of the radar platform. SAR has the advantage of being able to form images through clouds or similar impediments and in the dark. Traditionally, a SAR platform involves mounting an antenna on a vehicle, generally an aircraft or a satellite. The antenna is mounted so that the beam pattern is pointing downward and to the side. Fig. 3.1 shows an example of the antenna geometry.

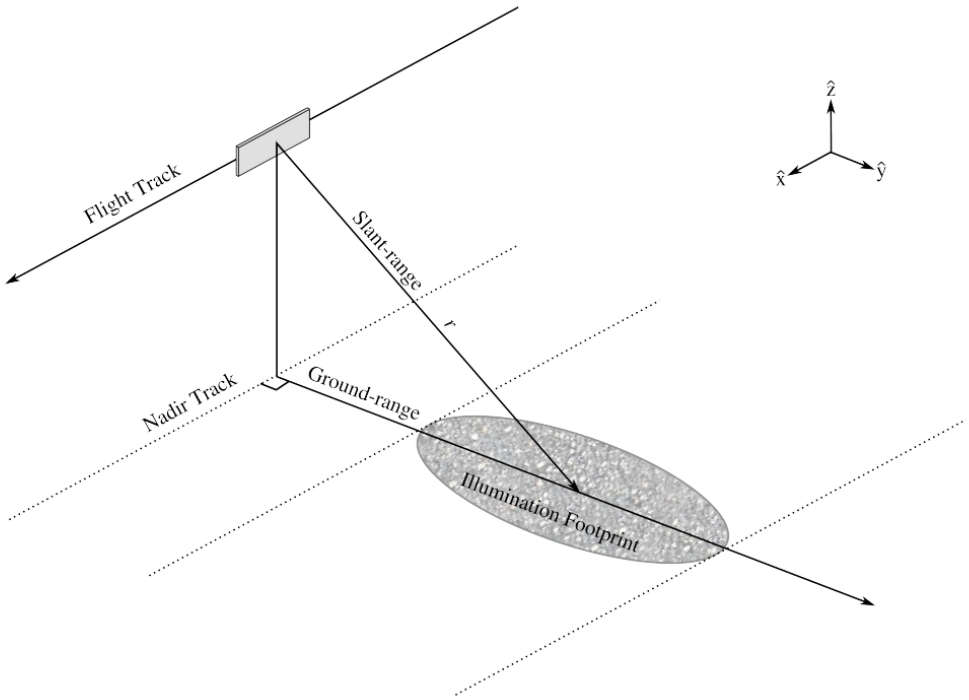



Fig. 3.1: Side-looking SAR antenna geometry (copied from Fig. 2.1 of [19])

In this figure, the antenna is pointing along the arrow titled "slant-range," and its beam pattern produces the illumination footprint. The antenna points to the side, rather than straight down, producing a larger illumination footprint. The flight path is often called the azimuth direction or the along-track direction. The "ground-range" direction is often called the cross-track direction or the range direction.

The platform transmits pulses periodically as it moves along the flight path. The pulse travels through the air at the speed of light, which for typical operating scenarios is fast enough that the motion of the platform is negligible. This allows the assumption that the position of the platform when the pulse is transmitted is the same as the position when the pulse is received. As the platform flies along the flight path and transmits periodic pulses, the illumination footprint moves. The pulse repetition frequency is fast relative to the motion of the platform, such that the illumination footprint of one pulse partially overlaps with that of many other pulses. Since each received pulse corresponds to a different location

of the platform, each pulse contains different information about the ground. This differing information allows for images to be formed from the collected data.

In this paper, all pulses are linear frequency-modulated (LFM) chirps. An LFM chirp is a sinusoid that increases in frequency linearly over time. In this paper, all chirps are complex exponentials, and Fig 3.2a shows an example of the real part of an LFM chirp.

The data can be acquired using different geometries. For example, in `spotlight mode`, the radar antenna can be rotated on a gimbal to focus on a single location as the platform passes. The data are processed to form an image of the area of focus. In `stripmap mode`, the antenna points in a fixed direction relative to the SAR platform, and an image is formed of the strip of terrain that it passes by. Since the purpose of SAR in the present application is to augment the navigational capability of a moving platform, the image formation algorithm uses stripmap mode. It is possible, however, to adapt these ideas to other modes, such as spotlight.

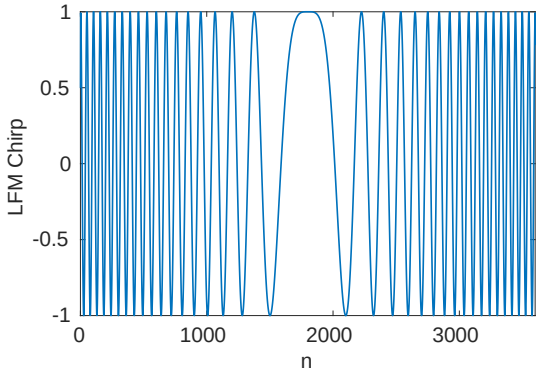
### 3.1.2 BPA

After the SAR data has been collected, it has to be processed to form an image. SAR processing can be performed in the frequency domain or the time domain, and multiple algorithms exist for each method. For this paper, time domain processing is used in the form of BPA. Time domain processing is generally more computationally expensive than frequency domain; however, it allows contributions to the SAR image to be added on a pulse-by-pulse basis, which is ideal for this research. Additionally, BPA is highly parallelizable, which can reduce the disadvantage of its computational complexity.

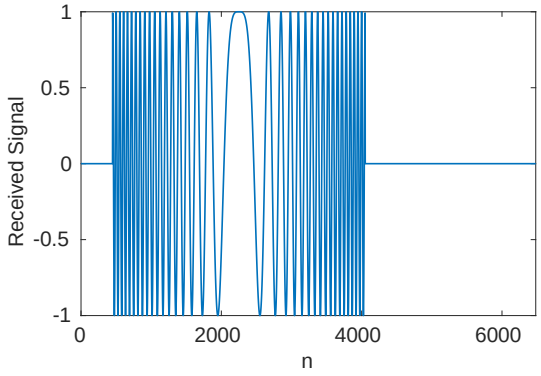
When images are formed using BPA, the first step is to matched-filter the data. This is done by correlating the received pulse with the transmitted pulse. This is referred to as the range compression step. Since each pulse reflects off the entire illumination region, the received pulse will have multiple copies of the transmitted pulse overlaid at different time offsets (corresponding to different distances from the platform) with different amplitudes (corresponding to the reflectivity of the illumination footprint). The result of range compression is the magnitude of the reflected signal can be better compared between various

objects on the ground.

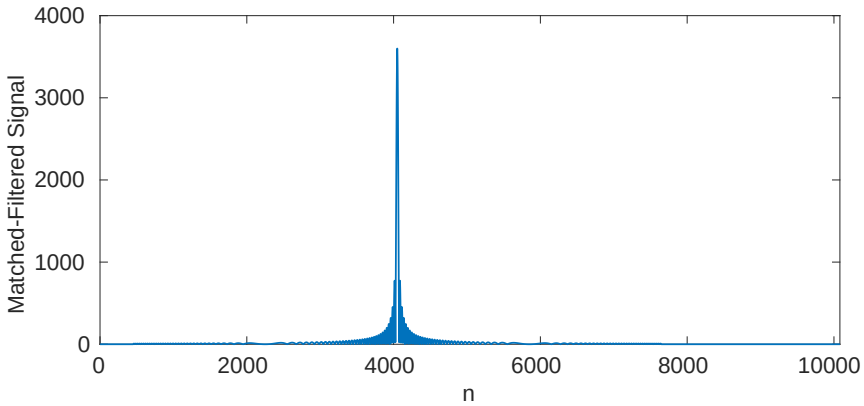
As an example of the effect of matched-filtering, consider the case where the transmitted pulse reflects off a single point source and nothing else. The received signal will contain one instance where it looks a lot like the transmitted signal at a delay corresponding to the distance from the radar platform to the point source, so the matched-filtered signal will have a single peak. Fig. 3.2 shows this situation.




(a) Real part of an LFM chirp




(b) Real part of received signal




(c) Magnitude of matched-filtered signal

Fig. 3.2: Result of matched-filtering received signal from a point source

Since the transmitted pulse is a complex exponential with constant magnitude, only the real part of the transmitted and received signals are shown. The magnitude of the range compressed signal is shown because it best demonstrates the desired characteristics. The received signal in Fig 3.2b shows a copy of the transmitted chirp, which is the reflection

of the pulse off of the point source. Fig 3.2c shows a peak where the received signal looks like the transmitted chirp. From this peak, the time difference between when the pulse was transmitted and received can be calculated and used to find the distance from the platform to the point source.

The next step in BPA is azimuth compression. First, the union of the illumination footprint of each pulse is mapped to the pixels of an image, represented by a matrix $I$ where each pixel is indexed as $I(i, j)$. Each pixel of the image corresponds to a location of the physical region, which will be referred to as $(x(i), y(j))$. Each coordinate $(x(i), y(j))$ is called a scattering cell. (Technically, a scattering cell is the smallest region of physical space that can be differentiated given the characteristics of the transmitted pulse and the radar system, but when using pixels that are larger than the minimum possible size, the entire region corresponding to a pixel can be called a scattering cell.) It is assumed that when each pulse was transmitted, the location of the platform at that time was also recorded. Let the location of the radar platform at pulse $n$ be represented by $(r_x(n), r_y(n), r_z(n))$, where $r_x(n)$ is the along-track position, $r_y(n)$ is the cross-track direction, and $r_z(n)$ is the altitude above the ground.

For each received and range compressed pulse, the entire image is traversed, and contributions from that pulse are added to each pixel. To do this for the $n$th pulse, the Euclidean distance between the radar platform at pulse $n$ and each scattering cell is calculated as

$$d(n, i, j, k) = \|(r_x(n), r_y(n), r_z(n)) - (x(i), y(j), z(k))\|.$$

It is often assumed that any scattering cells are located on the ground, and the ground is approximately level, meaning that $z(k) = 0\ \forall k$. This reduces the equation to

$$d(n, i, j) = \|(r_x(n), r_y(n), r_z(n)) - (x(i), y(j), 0)\|.$$

If additional information is known about the altitude of each scattering cell, such as from a DEM, it can be incorporated in $z(k)$.

Each calculated distance corresponds to a time delay between the transmission of a radar pulse and the reception of the reflection of that radar pulse off a scattering cell. This time delay is calculated as

$$\begin{aligned}\tau(n,i,j) &= \frac{2}{c}d(n,i,j) \\ &= \frac{2}{c}\|(r_x(n), r_y(n), r_z(n)) - (x(i), y(j), 0)\|\end{aligned} \tag{3.1}$$

where $c$ is the speed of light. The factor of $\frac{2}{c}$ arises from the pulse traveling at the speed of light from the radar platform to the location of the scattering cell and back to the radar platform.

The time delay $\tau$ is used to index into the range compressed pulse. The value of the range compressed pulse at the calculated time delay is multiplied by a complex exponential and accumulated into the corresponding pixel of the image. The reason for the multiplication with a complex exponential is discussed in [19]. Putting this together, for each pulse, the value added to $I(i,j)$ is

$$e^{j\pi f_0 \tau(n,i,j)} g(n, \tau(n,i,j))$$

where $f_0$ is the initial frequency of the LFM pulse, and $g(n,\tau)$ is the value of the $n$th range compressed pulse at time $\tau$.

After every pulse that covers the physical region of interest has been accumulated into the image, the image formation process is complete. Pseudocode that implements BPA is given in Algorithm 3.1.

BPA is able to account for any platform motion since it does not assume a fixed trajectory. Instead, BPA uses the position of the platform at the time of each pulse to generate the image. Because of this, BPA is very sensitive to radar platform motion, and if the precise position of the platform is not known, the formed image will be inferior in some regard to images formed when the true platform position is known. This means errors in the image can be exploited to infer errors in position.

**Algorithm 3.1** BPA

Create zero image $I$
For each pulse $n$
    Perform matched-filtering to create range compressed pulse $g(\tau)$
    For each pixel index $i$
        For each pixel index $j$
            $\tau = \frac{2}{c}\|(r_x(n), r_y(n), r_z(n)) - (x(i), y(j), 0)\|$
            $I(i,j) += \exp(j\pi f_0 \tau) * g(\tau)$
        End for $j$
    End for $i$
End for $n$

BPA can be implemented in real-time (up to computational speed) as each pulse arrives because no information about future pulses is necessary to accumulate the current pulse information into the image. This is ideal for a navigation algorithm because it can provide the most current position estimates. While BPA is computationally expensive, it contains a high degree of parallelism because each pixel value is independent of all others. This parallelism can be exploited to accelerate computation time in real-time applications. Additionally, the computational speed can be increased by performing the matched-filtering using frequency domain convolution.

## 3.2 SAR Image Quality Measure

The first objective of this research was to determine a robust measure of SAR image quality. The desired behavior of a good measure is one where a better image score corresponds to the image being formed with more accurate position data. The two methods tested were kurtosis and entropy.

### 3.2.1 Kurtosis

Kurtosis is a scaled version of the fourth moment of a probability distribution. It is a measure of how well the tails of a probability distribution match that of a Gaussian distribution. One important property of kurtosis is that the kurtosis of a Gaussian random variable is zero. For a non-Gaussian random variable, the kurtosis may be greater than or

less than zero. Thus, the absolute value of the kurtosis of a random variable is essentially a measure of how closely the probability distribution of the random variable resembles a Gaussian distribution.

The kurtosis of a zero-mean real random variable $y$ is given by Equation (3.2), and the kurtosis of a zero-mean complex random variable is given by Equation (3.3) [20].

$$K_r(y) = E[y^4] - 3E[y^2]^2 \tag{3.2}$$

$$K_c(y) = E[|y|^4] - 2E[|y|^2]^2 - |E[y^2]|^2 \tag{3.3}$$

For a sequence of $N$ trials of a random variable denoted by $y_1, y_2, \ldots, y_N$, the expected value may be approximated by averaging the trials.

$$\hat{K}_r(y) = \frac{1}{N}\sum_{i=1}^{N} y_i^4 - 3\left[\frac{1}{N}\sum_{i=1}^{N} y_i^2\right]^2 \tag{3.4}$$

$$\hat{K}_c(y) = \frac{1}{N}\sum_{i=1}^{n} |y_i|^4 - 2\left[\frac{1}{N}\sum_{i=1}^{N} |y_i|^2\right]^2 - \left|\frac{1}{N}\sum_{i=1}^{N} y_i^2\right|^2 \tag{3.5}$$

Kurtosis was chosen as a measure of image quality because of the relationship between kurtosis and a Gaussian distribution. It is conjectured that the worse the image quality, the more closely it resembles Gaussian noise. Conversely, the better the image quality, the higher the magnitude of the kurtosis. For scoring purposes, (3.5) was used where the sums are over the pixels in the formed SAR image.

The pixels of a SAR image are not necessarily zero-mean; however, the mean of the pixels is very small when compared to each individual pixel (approximately three orders of magnitude smaller). Thus, the zero-mean equation provides an accurate estimate of the true kurtosis value. Plots supporting this fact are given in Section 4.1.

### 3.2.2 Entropy

Entropy is the average amount of information conveyed by the outcome of a random

trial, and it is related to the level of surprise of the outcome of a random trial. The equation for entropy is given in (3.6), where $X$ is a discrete random variable with $N$ outcomes, $p$ is the probability measure of $X$, and $x_i \in X$ [21].

$$H(X) = -\sum_{i=1}^{N} p(x_i) \log_2(p(x_i)) \quad \text{[bits]}. \tag{3.6}$$

An important characteristic of the entropy of a discrete random variable is that entropy is highest for a uniform probability distribution.

As an example of the relation between entropy and surprisal, consider tossing a fair coin in which the probability of heads is equal to the probability of tails, 0.5. The entropy is

$$\begin{aligned} H(X) &= -\sum_{i=1}^{2} p(x_i) \log_2(p(x_i)) \\ &= -\sum_{i=1}^{2} \frac{1}{2} \log_2 \left(\frac{1}{2}\right) \\ &= 1 \text{ [bit]} \end{aligned}$$

This may be interpreted as saying that every toss of the coin requires one bit of information to convey the outcome.

Next, consider a coin where the probability of heads is 0.9, and the probability of tails is 0.1. In the case where the coin is tossed and it lands on heads, there is little "surprise" — the outcome is what was expected. On the other hand, if the coin lands on tails, the outcome is unexpected. The entropy is the average of the logarithmic measure of "surprise," computed by

$$\begin{aligned} H(X) &= -\sum_{i=1}^{2} p(x_i) \log_2(p(x_i)) \\ &= -0.9 \log_2(0.9) - 0.1 \log_2(0.1) \\ &= 0.469 \text{ [bits]} \end{aligned}$$

In this example, the entropy of the fair coin is higher than the entropy of the unfair coin. This is because on average, the amount of information conveyed by the unfair coin is very low. It is reasonable to expect the coin to always land on heads, and most of the time, this assumption will be correct. Also, this supports the claim that entropy is highest for uniform probability distributions. (Shannon's source coding theorem relates the entropy to the number of bits that would be required to store or transmit the series of outcomes, showing that asymptotically the entropy determines the amount of storage required.)

A plot of the entropy of a coin with respect to the probability that the outcome is heads is given in Fig. 3.3.

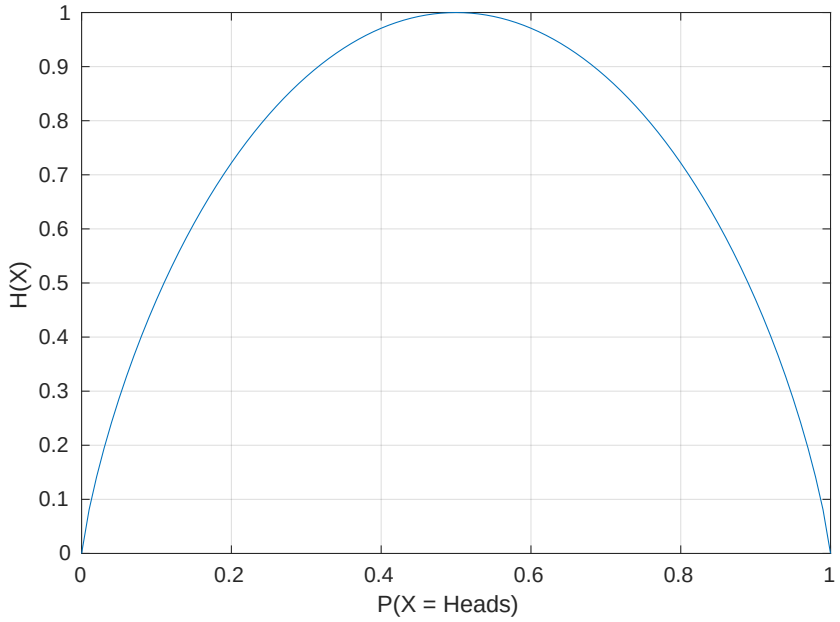



Fig. 3.3: Entropy of an unfair coin

This figure shows that the entropy of a coin toss is highest when the coin is fair and the probability distribution is uniform. As the distribution becomes less uniform, the entropy decreases.

Entropy was considered as a measure of image quality based on the following hypotheses: A well-formed SAR image will have distinctly different pixel amplitudes due to

structural features in the image. A SAR image that is poorly formed (due to inaccurate position measurements) will appear more noise-like than a well-formed image, having a distribution of pixel amplitudes that is more uniform. The extent to which these hypotheses hold depends on the image. An image formed from a uniform scene (e.g., water or sand) may have a fairly uniform distribution.

Due to the equation for entropy requiring the probability of each outcome of a random variable, the probability density function of the image must be estimated. This was done using a histogram. First, the minimum and maximum pixel values of the image were found, and the interval was split into $N$ bins of even width. Then, the number of pixels landing in each bin were counted, and this was used to calculate the probability of landing in each bin. Since in a SAR image, each pixel is a complex number, the magnitude of this number and the sum of the real and imaginary parts were tested.

### 3.3 GPS-D SAR Algorithm

The second objective of this research was to create and test an algorithm that uses the SAR image quality measure to aid in GPS-D navigation. The GPS-D SAR algorithm uses BPA-SAR images in addition to more traditional sensors, such as an IMU. Because BPA is heavily dependent on the position of the radar platform at the time of each pulse transmission, BPA can be used to help determine the relative location of the platform. This concept is illustrated in Fig. 3.4.

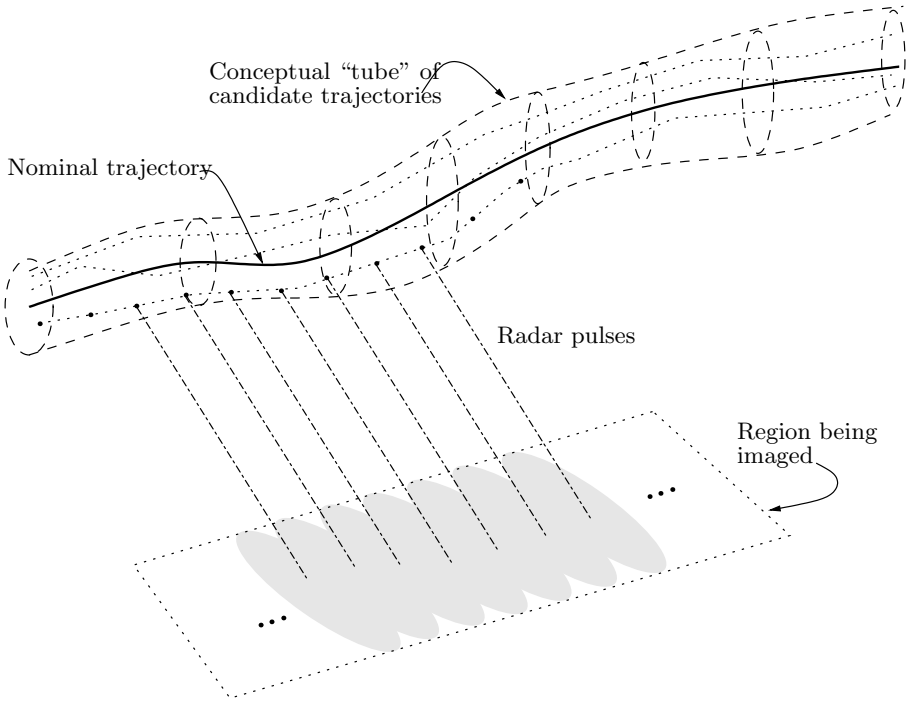



Fig. 3.4: Conceptual idea of SAR-based navigation

As the platform moves, it transmits and receives radar pulses at regular intervals. A nominal flight path is measured using an IMU or other instrumentation. The true flight path is considered to lie in a "tube" of candidate trajectories. Along each candidate flight path, a SAR image is formed by hypothesizing that the radar pulses were emitted along that path. These images are scored using the methods described in Section 3.2. The path used to form the best-scoring image is considered to be the true path. This process is repeated every $S_i$ pulses, where $S_i$ is one degree of freedom in the algorithm. This process is outlined in Algorithm 3.2.

**Algorithm 3.2** GPS-D navigation using SAR

Wait for $S_i$ pulses
Get nominal position from instrumentation
Create candidate positions around nominal position
Form BPA-SAR images using each candidate position
Score each image and select best
Update nominal position with position associated with best scoring image

As the platform moves, the current scene area is built up by radar pulses. Simultaneously, old scene areas shift off the region of interest, so updating is only done on the portion of current interest. The old scene areas can either be discarded or used for other purposes, such as comparing with maps.

The hypothesized positions can be selected around the sample point on the nominal trajectory. Variations can be in the $x$, $y$, or $z$ direction. Quantizing the set of possibilities would result in a large number of images to form and score; however, since velocity errors can integrate into large changes in position, it is assumed that the most significant position errors to compensate for lie in the along-track direction. For this reason, the majority of the testing done was in the along-track direction.

## 3.4 Testing

This section describes how each objective was tested to determine if it was met. First, Section 3.4.1 discusses the three datasets used for testing. Next, Section 3.4.2 outlines how each potential image quality measure was tested for the desired behaviors. Lastly, Section 3.4.3 gives a description of how the GPS-D SAR algorithm was tested on each dataset. This section also describes the design decisions made when testing the GPS-D SAR algorithm.

### 3.4.1 Data Sets

To test the GPS-D SAR algorithm effectiveness, three different datasets were used.

- The first data set, referred to as USU data, is artificial data created based on Fig. 3.5. This data was used as a proof-of-concept. It has the advantage that the features are distinct, and the position of the radar platform at each pulse is known exactly. It has the disadvantage that no natural scenery would have features this clear, but this clarity helps in debugging and establishing baseline performance.

Fig. 3.5: Image from which USU data was created

To generate the data, a coordinate system was first determined for how the image would map onto the simulated scene. Next, a vector of positions was created for the simulated SAR platform to fly through. Each position was looped through, and the pixels that overlapped with the radar beam were selected. Then, each pixel was checked to see whether or not the pixel was illuminated. An illuminated pixel is a pixel that is black in Fig. 3.5. If the pixel was illuminated, the time delay between sending and receiving the pulse was calculated using (3.1). The delay was used to create the simulated received signal from that pixel. This received signal was then added to a vector of received data. After every position was looped through, the received data and the parameters of the simulated SAR platform were exported to a binary file.

- The second set of data came from the Space Dynamics Lab (SDL). Fig. 3.6 shows a formed image from this data. This is real data that was acquired from a flight near Logan, UT. This data has the advantage that it is real, so data collection artifacts have to be handled. It has distinct linear features due to its semi-urban setting, which helps evaluate the algorithm but may not be representative of data encountered in a real-world application. Associated with this data is position data acquired by GPS. These positions are assumed to be relatively accurate.

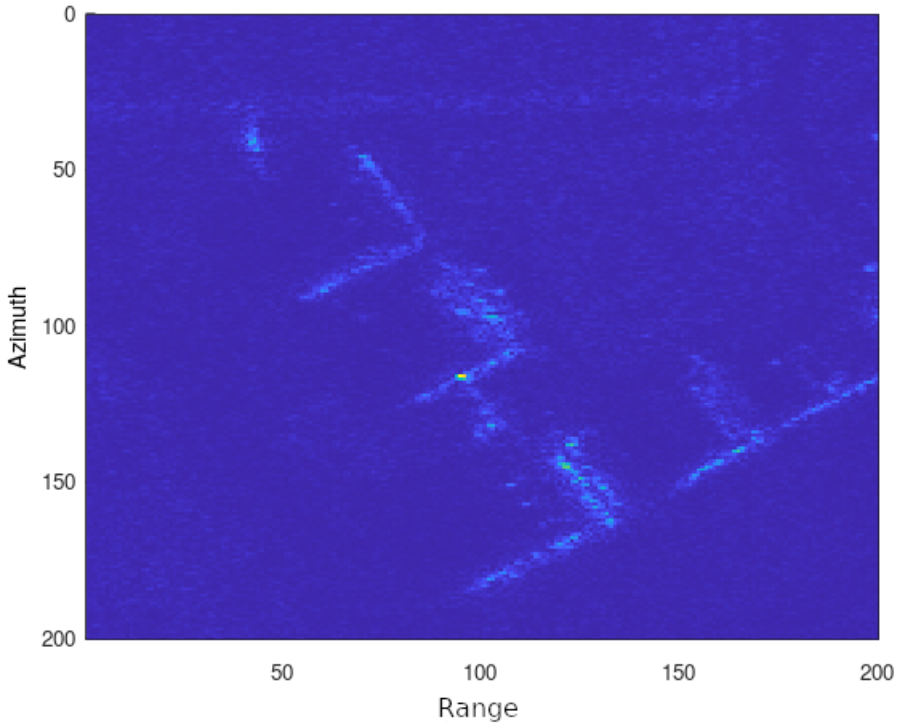



Fig. 3.6: Formed image from the SDL data

- The third set of data is real data acquired from Brigham Young University (BYU) [22] and is displayed in Fig. 3.7. This data was collected from a flight over Arctic sea ice using BYU's microASAR platform. This data also has data acquisition artifacts. It has no artificial linear features, so it may be more typical of data encountered in a real-world application. Associated with this data is position data acquired by GPS.

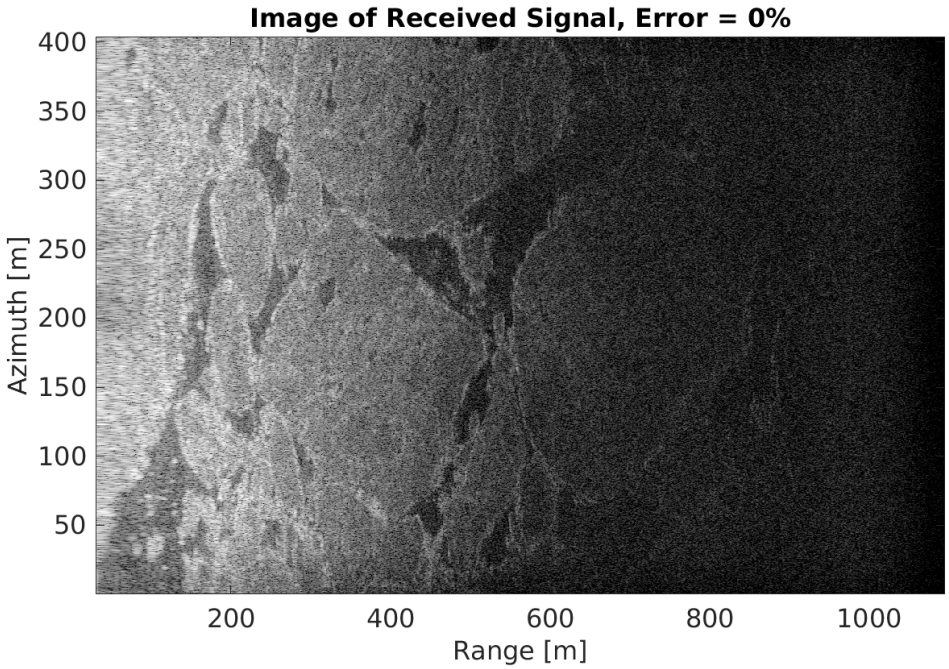



Fig. 3.7: Formed image from the BYU data

### 3.4.2 Image Quality Measure Testing

The first objective of this research was to determine an effective measure of SAR image quality. A desirable measure is one where a better score corresponds to an image being formed with more accurate position data. To test each measure in Section 3.2, BP-SAR images were formed from the data sets described in Section 3.4.1 using the exact positions given. Then, BP-SAR images were formed by adding errors in either the along-track or cross-track velocity of the platform. Kurtosis and entropy scores were computed at different magnitudes of position error, and the scores were plotted against the velocity error. The plots were examined for maximums at zero error in the kurtosis plots and minimums in the entropy plots.

### 3.4.3 GPS-D SAR Algorithm Testing

The second objective of this research was to create and test a GPS-D SAR algorithm that would utilize the image quality measure to correct for errors in position data. To do

this, velocity deviations were only considered in one direction at a time, most frequently, along-track velocity errors. In testing this algorithm, there were eight degrees of freedom:

- Direction of deviation (along-track vs. cross-track)
- Initial number of pulses to form the image over, $S_0$
- Number of pulses per iteration of BPA, $S_i$
- Number of candidate images to form, $L$
- Resolution of the images, $N \times M$
- Variance of noise added to velocity vectors
- Bias (mean) of noise added to velocity vectors
- Velocity spacing of the images around the nominal path

Algorithm 3.3 outlines how the algorithm was tested and the way each parameter was used. With each run, the noisy position vector and the corrected position vector were saved for comparison with the true position vector. The algorithm was run many times with varying parameters. At the end of each run, the variance between the original position vector and the noisy position vector was compared to the variance between the corrected position vector and the original position vector. The corrected position vector is the position vector corresponding to the best image at the end of the program run. Additionally, the differences in ending positions were compared, and the position plots were inspected to see if the result was desirable.

At step 1, the position vector was piecewise linearized $S_i$ samples at a time. This is done to make the comparison with the corrected position vector more accurate. The corrected position vector will always be piecewise linear, so if the variance and bias of the noise added to the velocity vector are small, this algorithm could function almost like a low-pass filter. This would result in a lower variance for the corrected position vector while not necessarily improving position estimates.

**Algorithm 3.3** GPS-D SAR Algorithm Testing

Piecewise linearize the position vector with $S_i$ samples per linear chunk.
Calculate the vector of differences in position per sample. That is, calculate $v[t] = p[t] - p[t-1]$ where $p[t]$ is the position vector. This is referred to as the velocity vector.
Add Gaussian noise to the velocity vector, starting with the $S_0$ element, in the desired direction (along-track or cross-track) with given variance and bias.
Calculate noisy position vector based on noisy velocity vector.
Piecewise linearize this noisy position vector in $S_i$ sample chucks.
Initialize $L$ position vectors to the noisy position vector.
Run BPA on the first $S_0$ pulses of the data using the true position vector, and initialize $L$ images to this image.
Loop through all pulses, $S_i$ at a time.
    Loop through $L$ position vectors and images.
        Starting with the last sample used, create deviations in the position vectors using the noisy velocity vector and the given velocity spacing.
        Run BPA for $S_i$ pulses on $L$ images with corresponding $L$ position vectors
    End loop.
    Score all of the images using the image quality measures
    Copy the best scoring image into all $L$ images, and copy the corresponding position vector into all $L$ position vectors.
End loop.

There are two main options for how the sample chunk could be linearized. One method would be to use all the samples and linear regression to find the line of best fit. The second option would be to take the starting and ending points and draw a line between the two points, discarding all intermediate information. The second option was used because of the assumption that the position information would be provided by an IMU. IMUs provide measurement information at a much lower frequency than the SAR pulse repetition frequency. This means that in a real-world application, the information between the starting and ending points for a sample chunk would never have been provided in the first place.

At step 3, noise was added to the velocity vector starting with element $S_0$ because it is assumed that true positions are known up to a point. This simulates a platform beginning a flight while having GPS signal. Then, at element $S_0$, the platform loses GPS signal. This is the same reason why the initial image is formed with true positions at step 7.

# CHAPTER 4

# RESULTS AND DISCUSSION

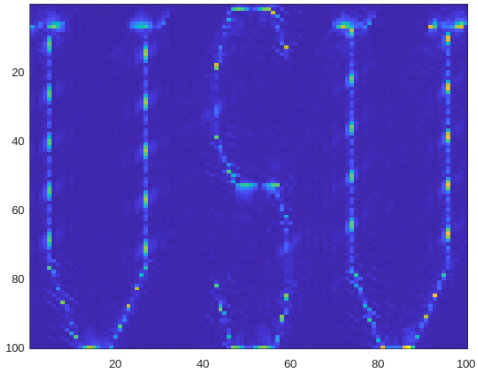
This chapter shows the results of testing the image quality measures according to Section 3.4.2, along with the results of testing the GPS-D SAR algorithm according to Section 3.4.3.

## 4.1 Image Quality Measure Results

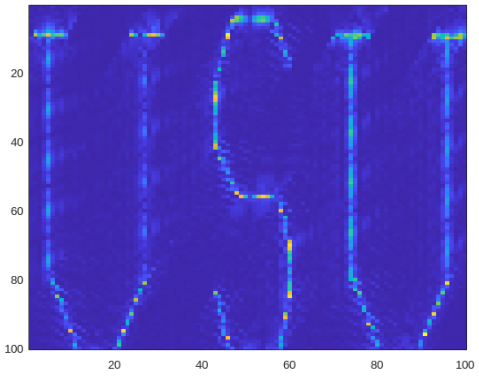
The image quality measures were tested using the three data sets described in Section 3.4.1. The results show that kurtosis is a robust measure of SAR image quality. Entropy also shows promise, but less than kurtosis.

### 4.1.1 USU Data

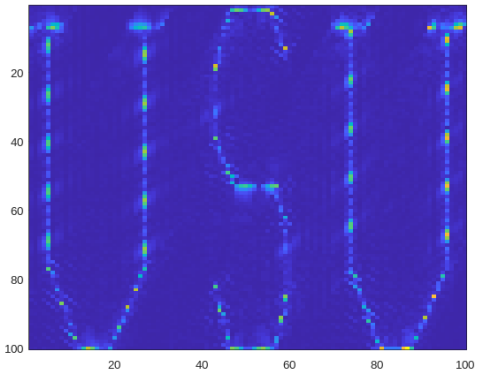
An example of images formed from the USU data with no deviations and small deviations in the along-track and cross-track directions are shown in Fig. 4.1. The images formed with errors appear more blurred than the image formed with no errors. The image formed with along-track velocity errors is also shifted down compared to the image formed with no errors. The along-track velocity errors are shown as a percentage because a percentage gives the most perspective. The cross-track velocity errors are shown as the number of meters traveled in the cross-track direction per meter in the along-track direction. This was done because the data were formed using zero cross-track velocity, so a percentage cannot be calculated.




(a) Nominal trajectory



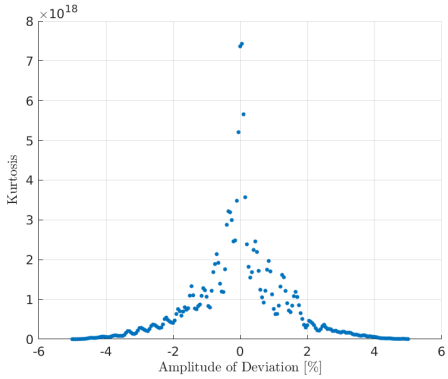

(b) Along-track 0.2% velocity deviation



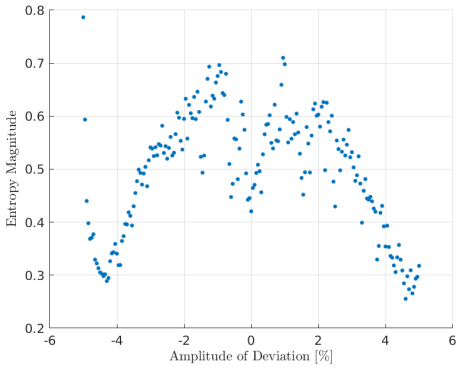

(c) Cross-track 0.002 [m/m] velocity deviation

Fig. 4.1: Formed 100 × 100 USU images

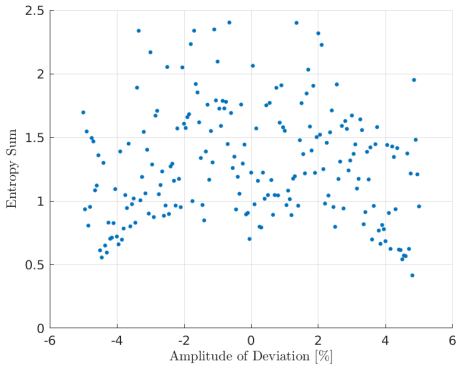
The following image quality measure testing plots were formed according to Section 3.4.2. Images were formed using the true position vector and deviated position vector. Each of the formed images was scored using kurtosis and entropy. When scoring the images using entropy, the magnitude of each complex pixel was tested along with the sum of the real and imaginary parts. These different methods are referred to as entropy magnitude and entropy sum, respectively. Example plots of the image score vs. the amplitude of the deviation are given in Fig 4.2, where the points are plotted as a scatter plot. In future plots, the points are interpolated between because it better shows the trend of the data.




(a) Kurtosis




(b) Entropy magnitude, N=10




(c) Entropy sum, N=10

Fig. 4.2: USU along-track velocity deviation, 100×100 image

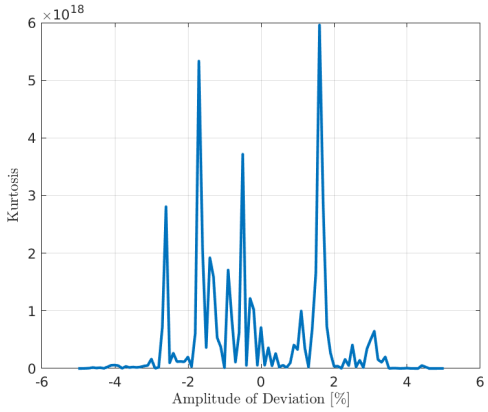
The kurtosis plot, Fig 4.2a, shows a clear trend where the images formed with smaller deviation have a higher score. This shows promise in kurtosis as a measure of image quality. The entropy magnitude plot, Fig 4.2b, also shows desirable behavior, though it is less desirable than the kurtosis plot. There is a trough in the plot near zero deviation, but the data is more scattered surrounding this area. Additionally, the trough around zero is not the global minimum for the data plotted. This suggests that entropy magnitude could be a viable image quality measure as long as the amplitudes of the deviations tested are small enough. The entropy sum plot, Fig 4.2c, has no clear trend and shows virtually zero promise as an image quality measure.

In addition to the magnitude and direction of the velocity errors, other factors that influence the score of the images are the resolution of the image in pixels and the number

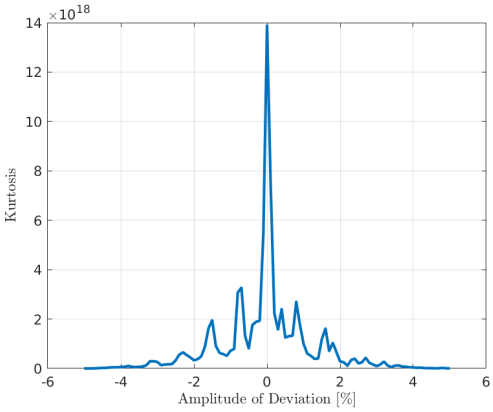
of bins used in the histogram for the calculation of entropy, as described in Section 3.2.2. These parameters are considered in the following sections.

### Image Resolution

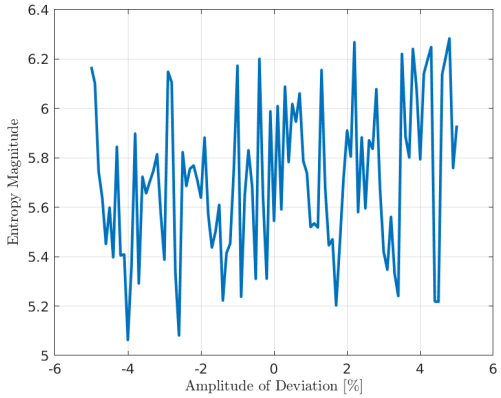
When forming a SAR image, the image resolution in pixels must be chosen. A higher resolution image contains more information about the scene, but it requires more computations, which leads to more time to produce the image. For computational efficiency, the smallest image resolution that gives an easily discernible peak (in the case of kurtosis) or trough (in the case of entropy) is desired. Resolutions from $10 \times 10$ to $300 \times 300$ are tested in the following plots. All entropy plots use $N = 1000$ when calculating the histograms because this number is high enough that it does not negatively impact the performance of entropy as a measure of image quality (This is demonstrated in the section showing the results of testing the value of $N$). Only the along-track deviation plots are shown because the cross-track plots look very similar and provide no additional insight.



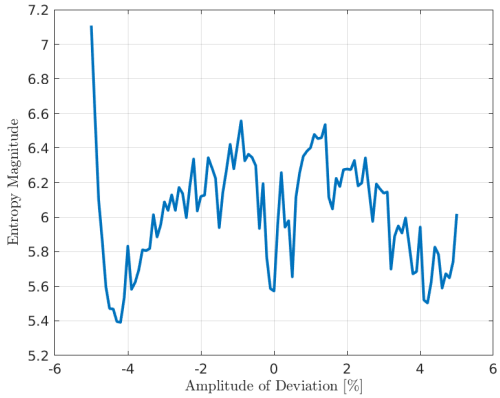

(a) Kurtosis



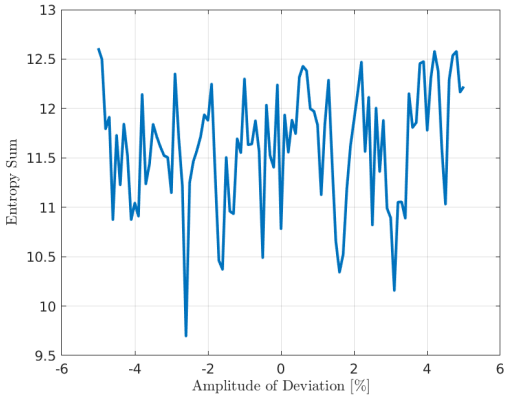

(a) Kurtosis



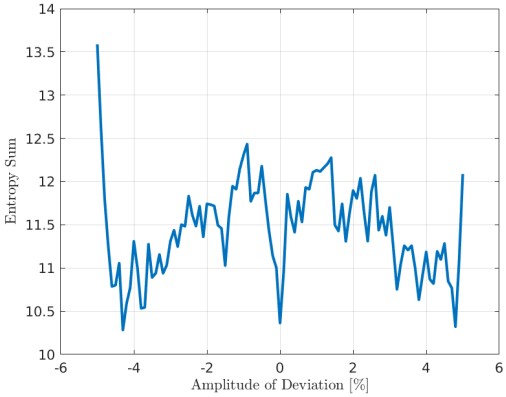

(b) Entropy magnitude




(b) Entropy magnitude



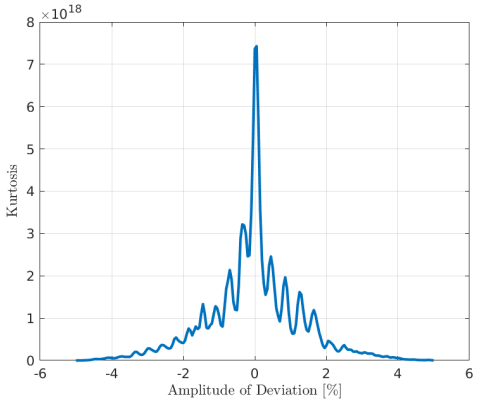

(c) Entropy sum



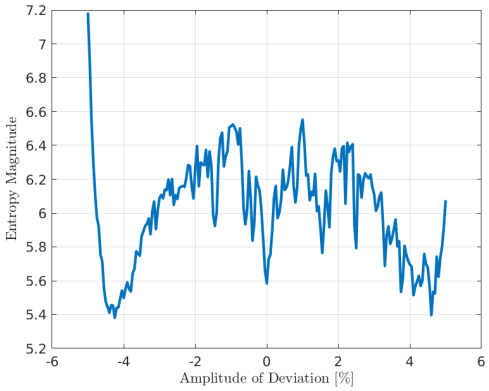

(c) Entropy sum

Fig. 4.3: USU along-track velocity deviation, 10×10 image

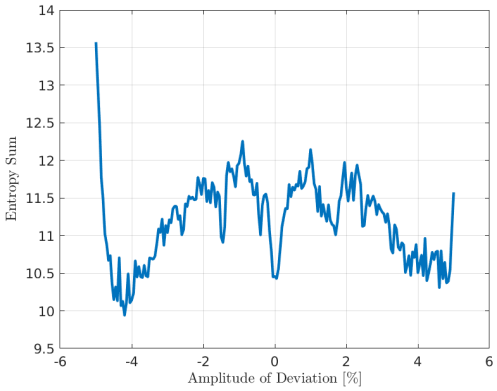
Fig. 4.4: USU along-track velocity deviation, 50×50 image




(a) Kurtosis

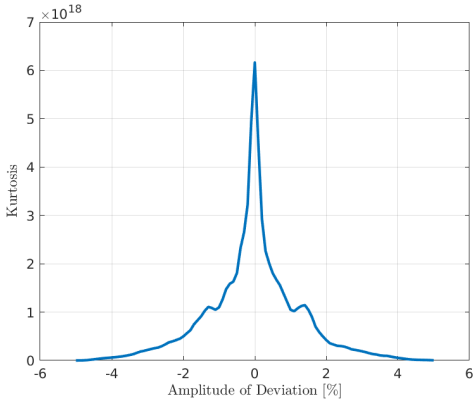



(b) Entropy magnitude

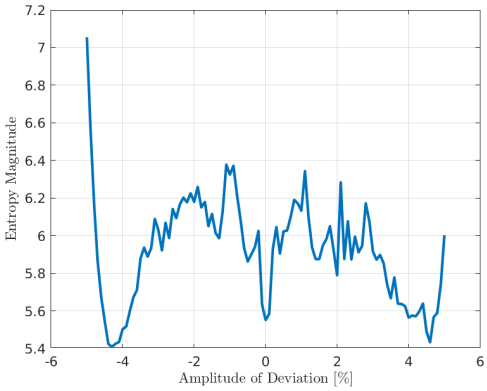



(c) Entropy sum

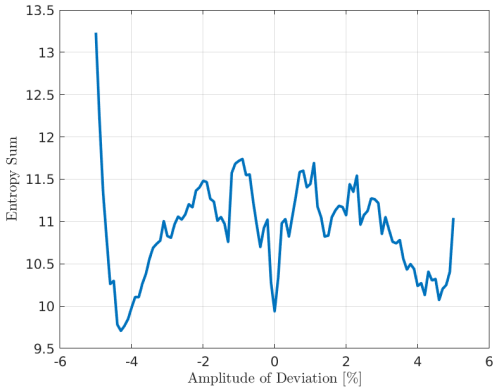
Fig. 4.5: USU along-track velocity deviation, 100×100 image




(a) Kurtosis




(b) Entropy magnitude




(c) Entropy sum

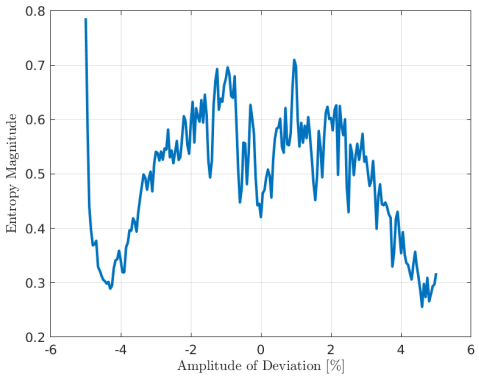
Fig. 4.6: USU along-track velocity deviation, 300×300 image

From Fig 4.3-4.6, it can be seen that higher-resolution images create smoother plots.

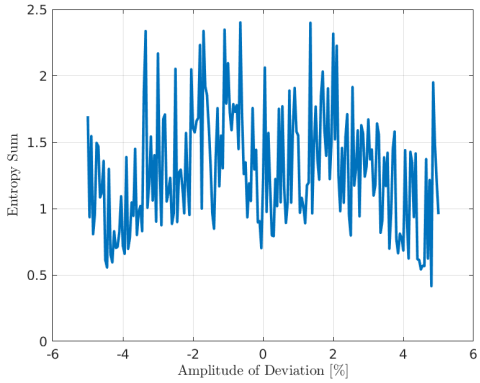
$10 \times 10$ images are not sufficient to determine any trend in the data, but when increasing the resolution to $50 \times 50$, peaks and troughs become detectable. Further increasing the image resolution makes the peaks and troughs more clear and accentuates the overall trend of the data.

**Number of Histogram Bins**

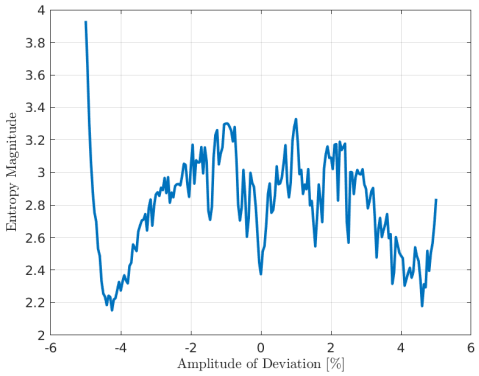
Next, the number of bins used in the histogram calculations was considered. Plots of $N = 10$, $N = 100$, and $N = 1000$ were formed. The plots were created using $100 \times 100$ images because this resolution was high enough to show the desired characteristics. Both along-track and cross-track velocity deviation plots are shown because they contain different information.



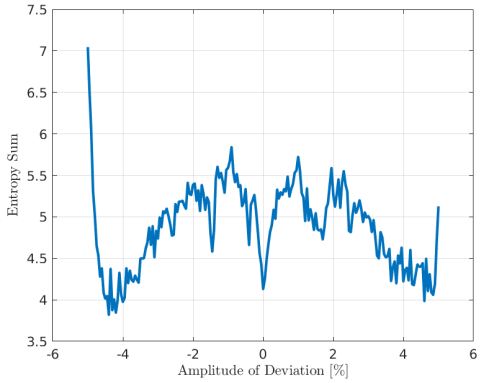

(a) Entropy magnitude, N=10



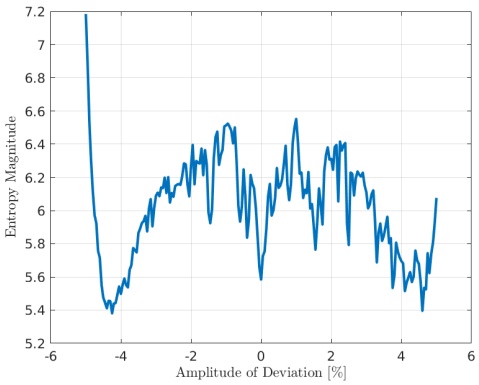

(b) Entropy sum, N=10



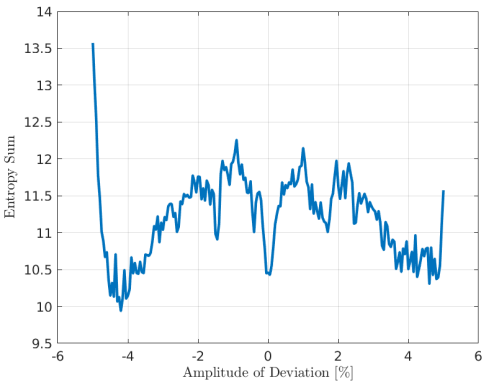

(c) Entropy magnitude, N=100




(d) Entropy sum, N=100

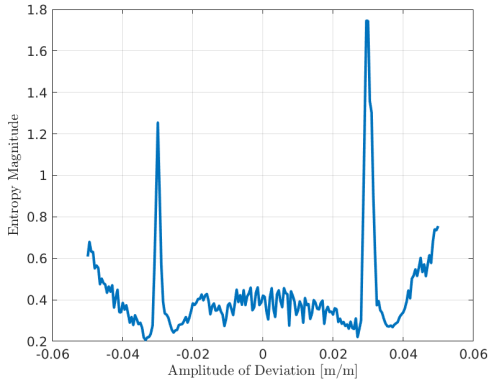



(e) Entropy magnitude, N=1000

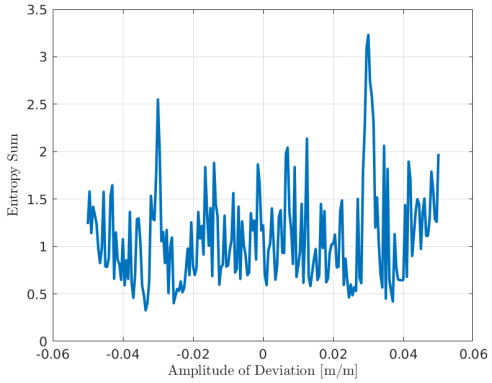



(f) Entropy sum, N=1000

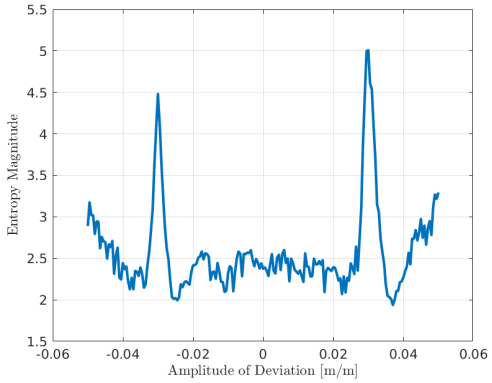
Fig. 4.7: USU along-track velocity deviation, 100×100 image



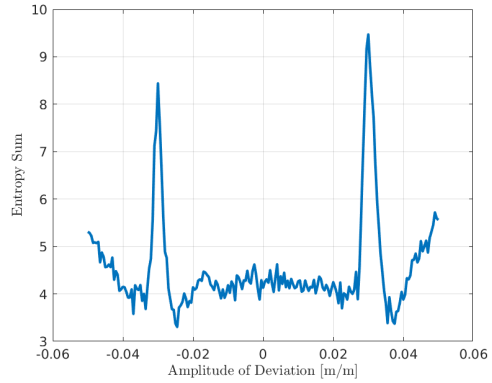

(a) Entropy magnitude, N=10



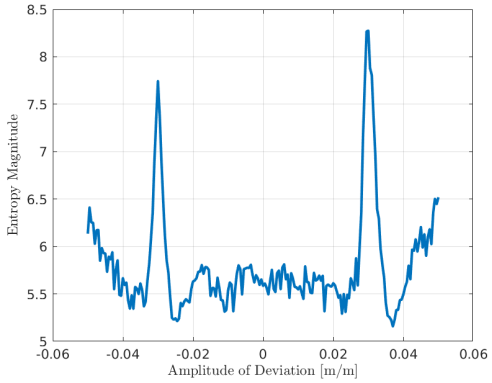

(b) Entropy sum, N=10



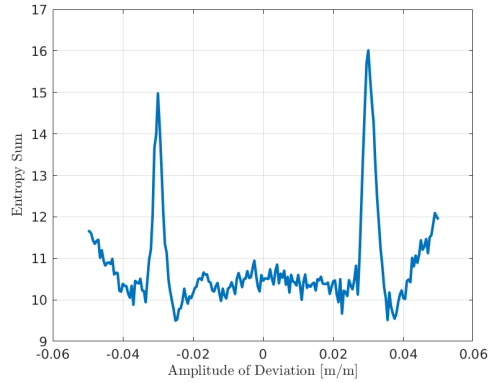

(c) Entropy magnitude, N=100




(d) Entropy sum, N=100




(e) Entropy magnitude, N=1000




(f) Entropy sum, N=1000

Fig. 4.8: USU cross-track velocity deviation, 100×100 image

Fig. 4.7 shows that a value of $N$ as low as 10 creates a distinct trough in the plot of entropy magnitude. Increasing to $N = 100$ provides some benefit over 10, but going

above 100 seems to provide no benefit. With entropy sum, a value of $N = 10$ is insufficient to provide a trough in the plot. Increasing to $N = 100$ makes a trough apparent, and increasing above 100 to 1000 provides no benefit. From Fig. 4.8, the significance of the $N$ value is equivalent to the along-track case. However, in the cross-track case, there is no trough at a deviation of zero.

Overall, the results of the tests of image quality measures with the USU data are promising, specifically with kurtosis. All of the plots of kurtosis (with high enough resolution) show a large peak very close to zero error. There are some local maxima at large deviations, but as long as the approximate position is close to accurate, these local maxima should not be an issue. This trend applies to images of sufficient resolution tested, with the higher resolution images creating smoother plots.

The entropy plots are less promising. Some of the plots show a local minimum near zero error, but other local minima are present without deviating too far from zero error. The entropy sum function seems to perform as well as the entropy magnitude function in cases where the resolution or number of bins is high enough, but the sum function never exceeds the performance of the entropy magnitude function. Increasing the number of bins can create a more pronounced minimum at zero error, but this is not always the case. Similar to kurtosis, higher resolution images create smoother entropy plots.

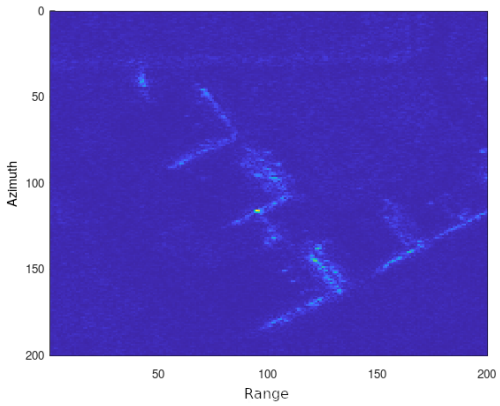
Overall, the testing with the USU data shows that kurtosis could be a viable measure of image quality. Entropy shows some promise but much less than kurtosis. Additionally, the results appear very similar between along-track and cross-track velocity deviations.

### 4.1.2 SDL Data

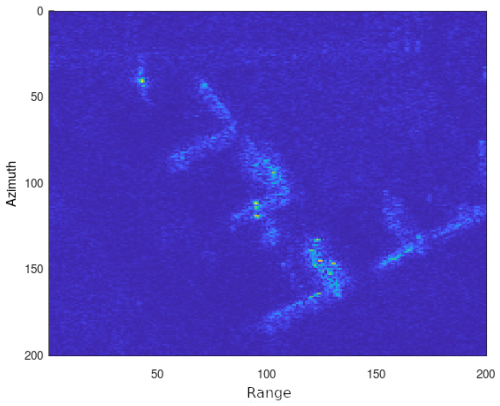
Images formed from the SDL data with -1.7% along-track velocity error, 0% velocity error, and 1.7% velocity error are shown in Fig. 4.9. These images correspond to forming the image assuming the platform was moving 1.7% slower than what GPS recorded, equal to what GPS recorded, and 1.7% faster than what GPS recorded. With this data, only along-track deviations were tested. Additionally, since the USU data showed that entropy sum performs no better than entropy magnitude, only kurtosis and entropy magnitude were

tested.

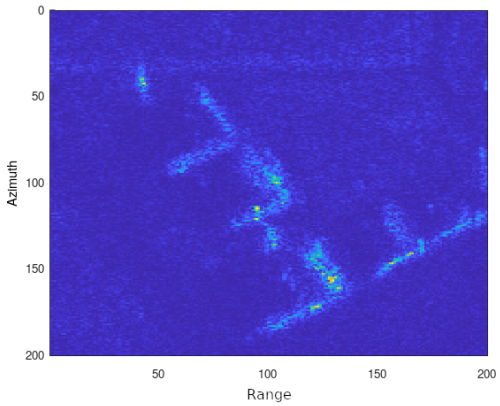
Fig. 4.10a shows the results of testing kurtosis and entropy at different along-track velocity errors. Fig. 4.10b shows the same results but zoomed in. Additionally, Fig. 4.11 shows the results of testing kurtosis on the SDL data but subtracting the mean from every pixel before computing the kutosis value. The resemblance between Fig. 4.10b and Fig. 4.11 supports the claim that the zero-mean kurtosis equation is an appropriate approximation of the true kurtosis value.




(a) Error = 0%

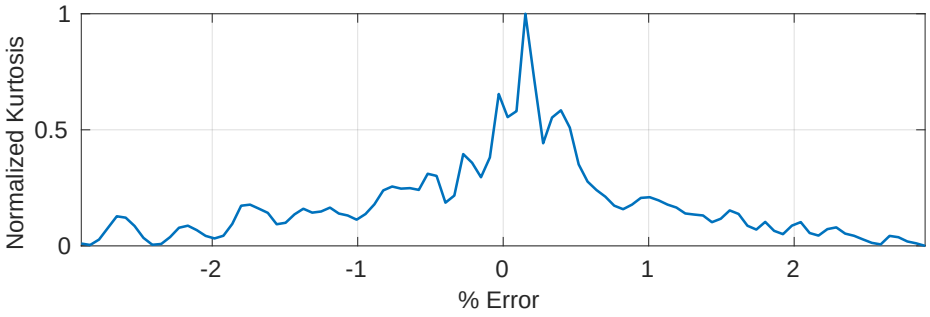


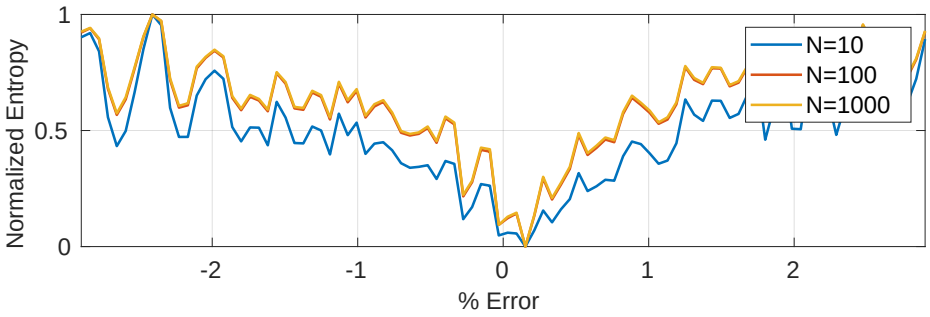

(b) Error = -1.7%



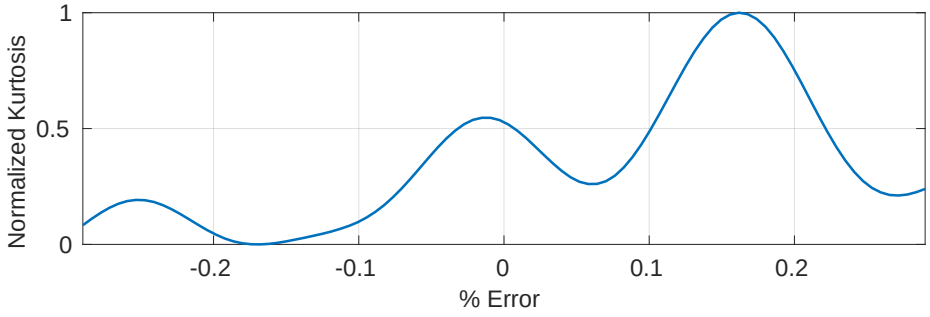

(c) Error = 1.7%

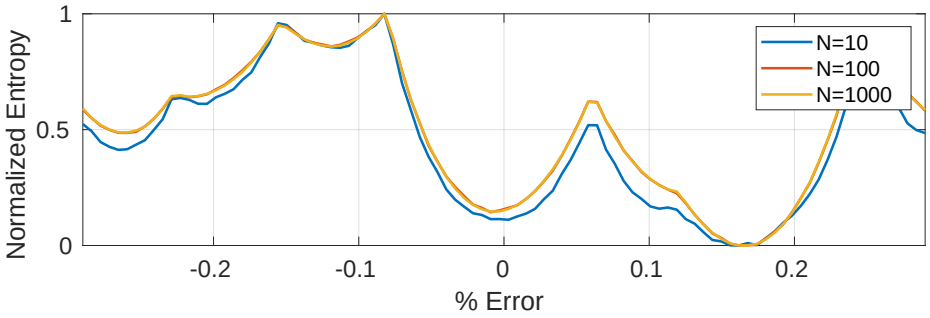
Fig. 4.9: Formed SDL images with along-track velocity errors




(a) Original




(b) Zoomed

Fig. 4.10: SDL data scoring plots

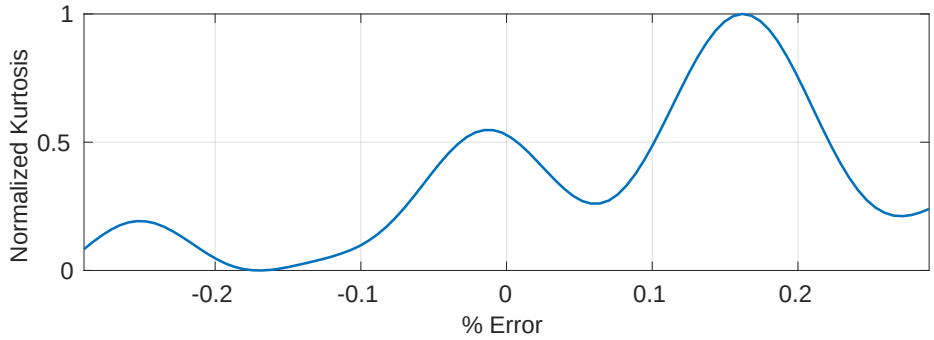



Fig. 4.11: Kurtosis of SDL data with mean removed

Like with the USU data, the images formed at velocity errors show a prominent blurring effect and increased noise. The scoring results of the testing done with the SDL data are very similar to the results with the USU data with two main differences. The first difference is that entropy seems to perform better in the zoomed-out plot than it did with the USU data. There is a more pronounced minimum near zero, and this minimum is the global minimum over the range of values calculated. When looking at the zoomed-in plot, the results seem unusable, though. The second difference is that the maximum of the kurtosis plot and the minimum of the entropy plot are slightly offset from zero. Since this is real data, it is impossible to know without any error what the true velocity of the SAR platform was. These plots could imply that the GPS data was slightly inaccurate.

### 4.1.3 BYU Data

The last set of data to test was the BYU data. Images formed from this data with velocity errors of -1.6%, 0%, and 3.5% are shown in Fig. 4.12-4.14. These error values were chosen due to the characteristics of the kurtosis plot. Like with the SDL data, only along-track velocity deviations were tested. Fig. 4.15 is a plot of kurtosis at different velocity errors, and Fig. 4.16 is a zoomed-in version of this plot. Notice that the zoomed-in plot is not centered at 0% error. Fig. 4.17 is a plot of kurtosis with the mean removed. Again, the resemblance of Fig. 4.16 to Fig. 4.17 supports the claim that the mean is negligible in the calculation of kurtosis. Finally, Fig. 4.18 is a plot of entropy at different velocity errors.

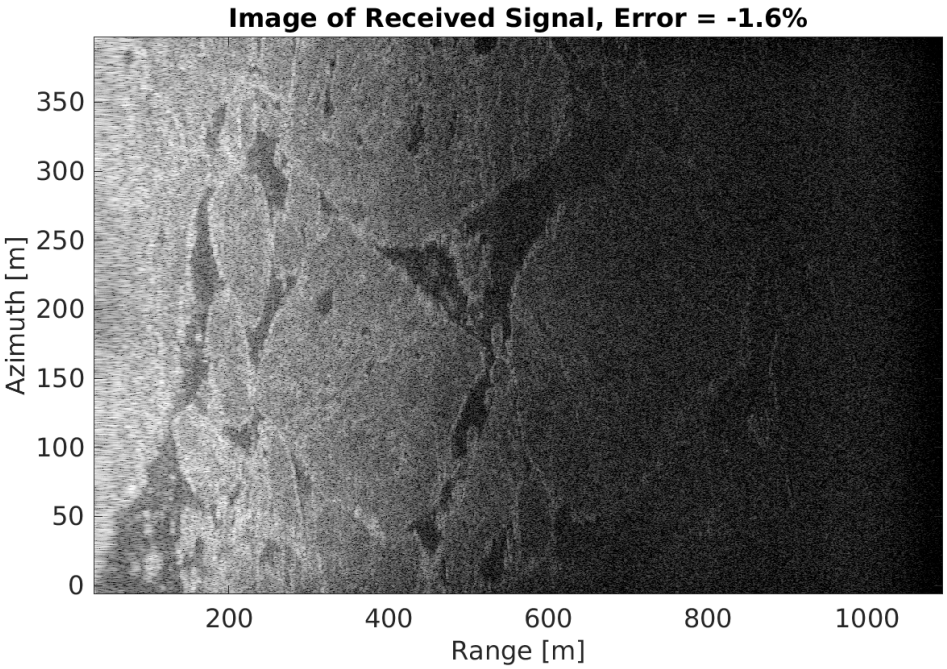



Fig. 4.12: BYU image at error = -1.6%

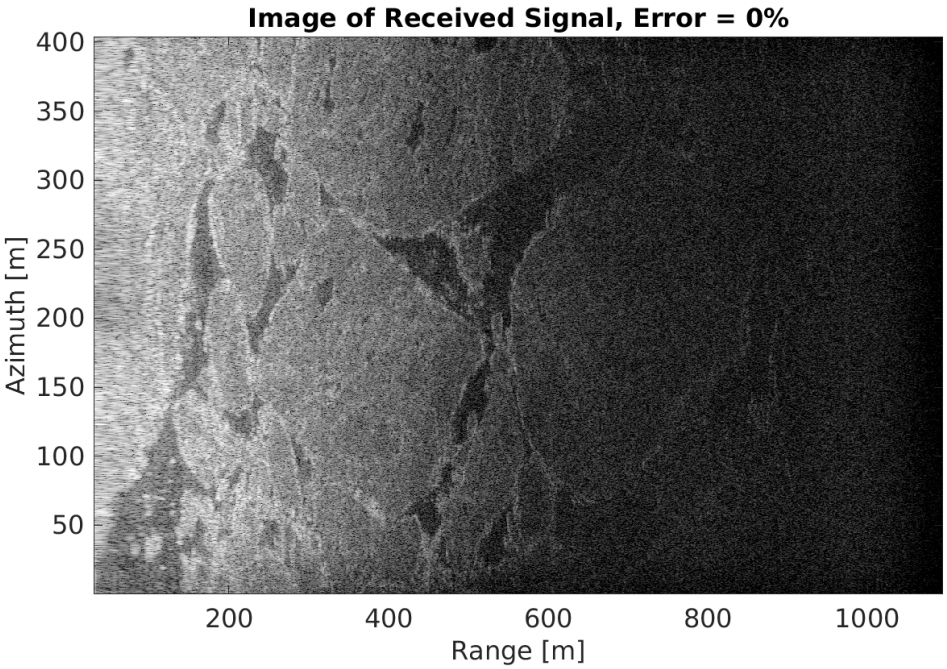



Fig. 4.13: BYU image at error = 0%

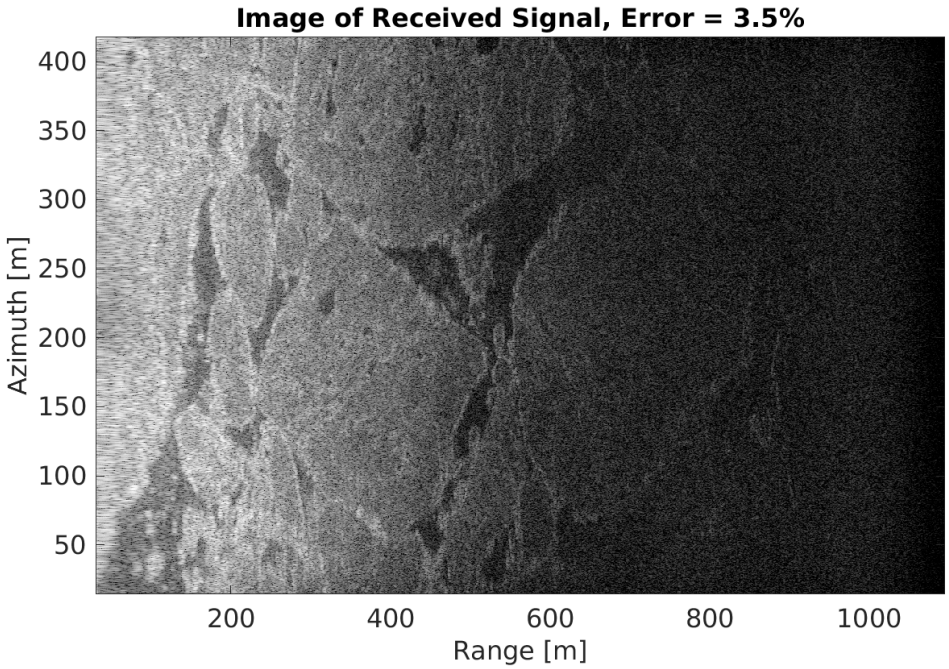



Fig. 4.14: BYU image at error = 3.5%

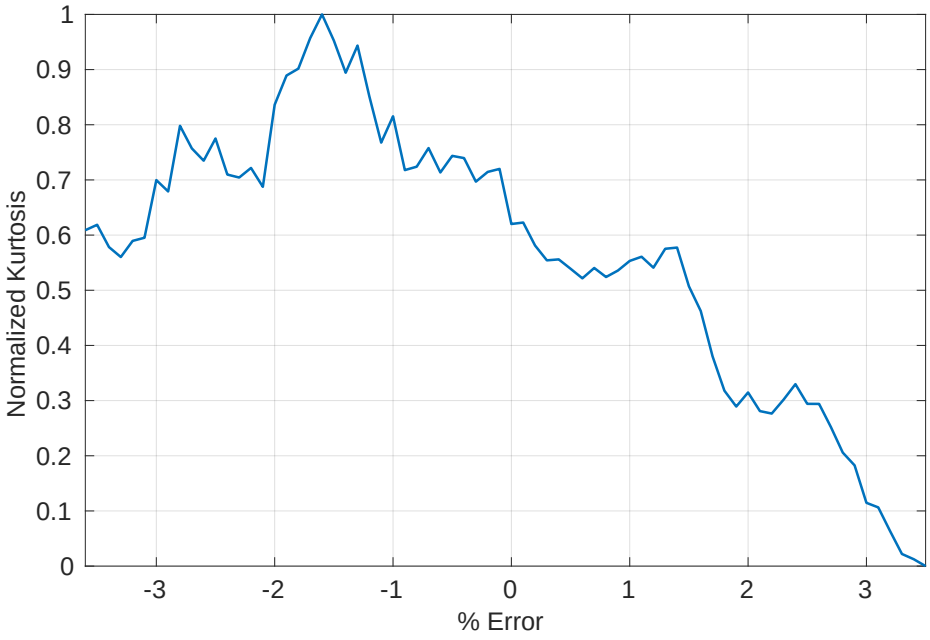



Fig. 4.15: BYU kurtosis plot

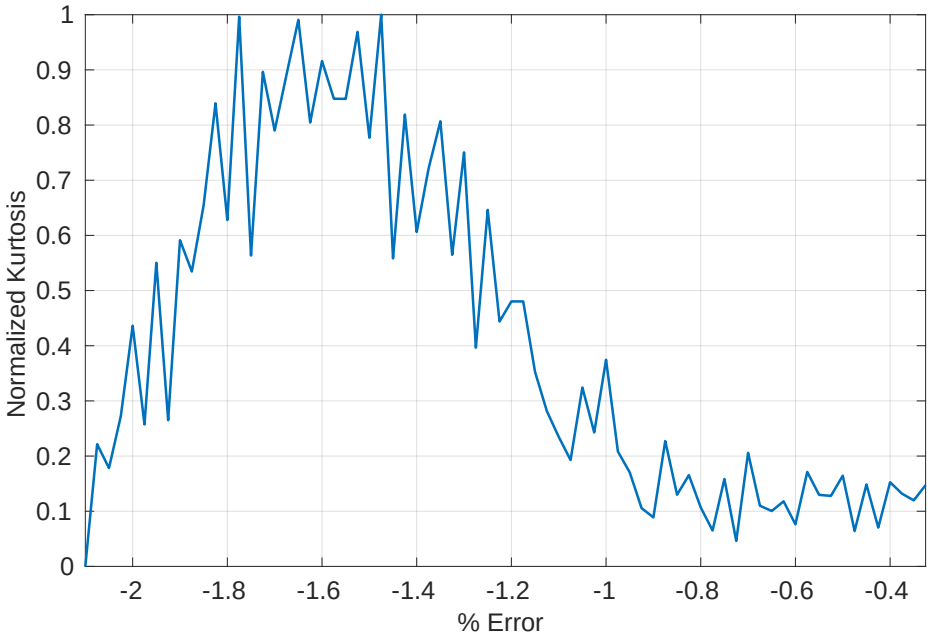



Fig. 4.16: BYU kurtosis plot zoomed

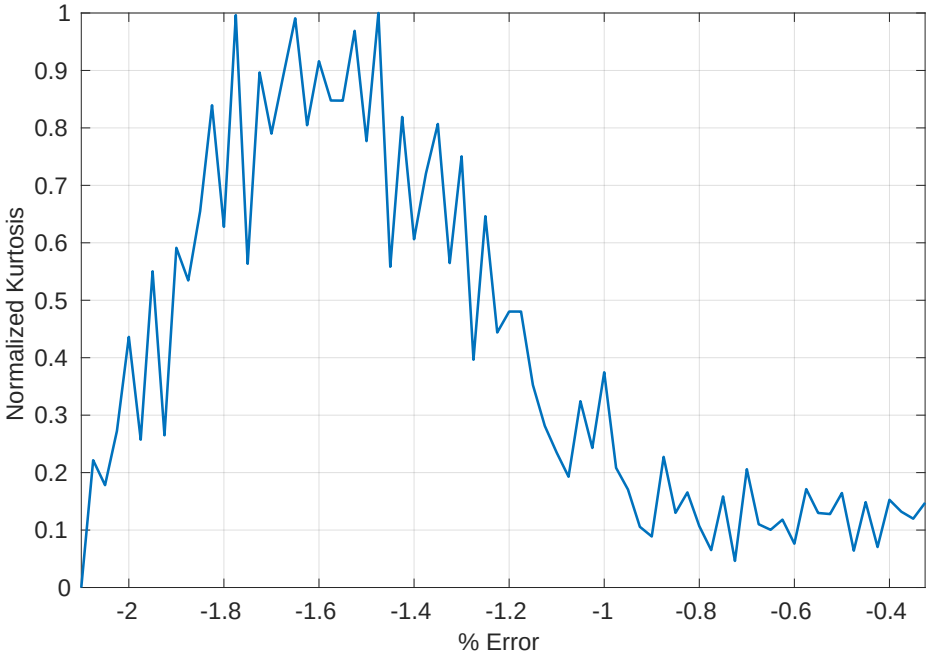



Fig. 4.17: Kurtosis of BYU data with mean removed

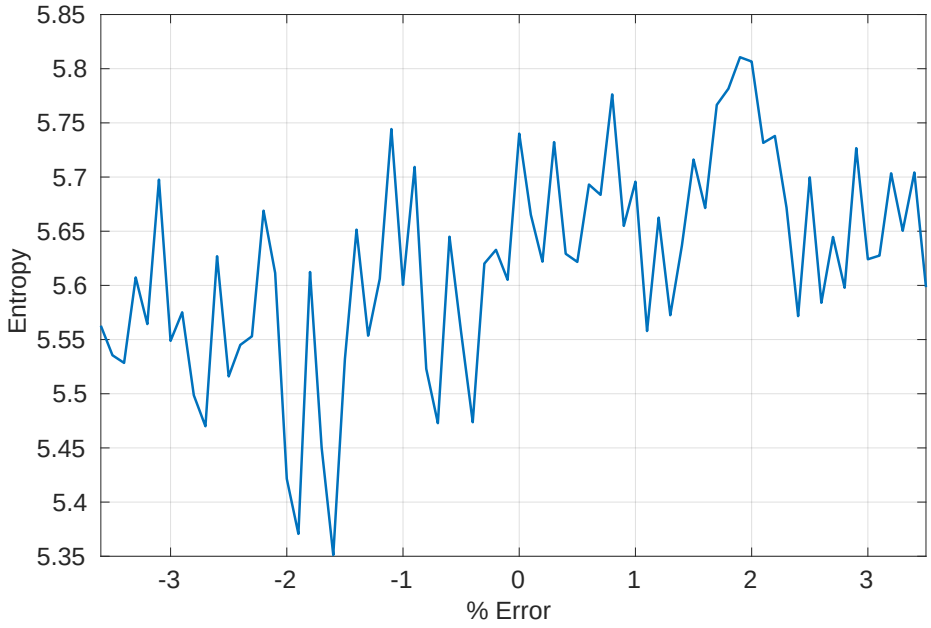



Fig. 4.18: BYU entropy plot

The results of the testing done with the BYU data are very similar to the SDL data. There is a very prominent peak in the kurtosis plot. This peak is further offset from zero error than the peak in the SDL plot, though. As with the SDL data, this offset could be due to a bias in the GPS positions. Additionally, the zoomed kurtosis plot shows that there are many peaks between -1.8% and -1.4% error. The entropy plot further verifies that entropy does not work well as a measure of image quality. There is no pronounced minimum in the plot. When looking at the formed images, it is very difficult to see much of a difference. This shows that kurtosis is better than visual inspection at identifying differences in images. Overall, kurtosis shows promise in being an effective measure of SAR image quality.

## 4.2 GPS-D SAR Algorithm Results

When testing the GPS-D SAR algorithm, the number of degrees of freedom had to be narrowed down for computational expedience. Because of this, all testing was performed in the along-track direction. This was done because along-track velocity errors could accumulate into large position errors over time. The number of initial pulses to form the image

over, $S_0$, was set to 1000 for every test. This number was chosen because it is large enough that the image has already started to form, but it is small enough that there are still a large number of pulses left for the algorithm to run on. The resolution of the images was set to $200 \times 200$ because the image quality measure results showed this was a large enough resolution to achieve good results. Additionally, the bias was held constant at $8.1641 \times 10^{-6}$. All testing was performed on the SDL data.

The number of pulses per iteration, $S_i$, number of candidate images to form, $L$, velocity spacing of the images around the nominal path, and the variance of the noise added to the velocity vector were tested simultaneously.

A tactical and commercial grade IMU can be expected to have a velocity random walk (VRW) value of approximately 0.02 $[\text{m/s}/\sqrt{\text{hr}}]$ and 0.2 $[\text{m/s}/\sqrt{\text{hr}}]$, respectively. The standard deviation can be calculated from the VRW by

$$\sigma = \text{VRW}\sqrt{\frac{1}{3600 * \text{prf}}} \quad [m/\text{sample}]$$

where prf is the pulse repetition frequency, and the factor 3600 converts from hours to seconds. Since the SDL data has prf = 1667, the range of $\sigma$ values corresponding to the VRW range between tactical and commercial grade IMUs is $[8.1641 \times 10^{-6}, 8.1641 \times 10^{-5}]$. For this reason, ten evenly spaced standard deviations within that range were tested. Using Matlab-style syntax, that is $\sigma$ `= linspace(8.1641e-6,8.1641e-5,10)`.

For each standard deviation, 30 velocity spacings were tested, beginning with the standard deviation divided by 10,000, and ending with the starting value multiplied by 30. In Matlab-style syntax, that is `spacings = linspace(`$\sigma$`/10000,`$\sigma$`*30/10000,30)`. For each spacing, $L$ images were formed. These images were formed with velocity deviations beginning with $-$`floor(`$L/2$`)*spacing` and incrementing by `spacing` until there are $L$ elements. If $L$ is odd, this is equivalent to `[-floor(L/2)*spacing:spacing:floor(L/2)*spacing]`, and if $L$ is even, this is equivalent to `[-floor(L/2)*spacing:spacing:floor(L/2-1)*spacing]`.

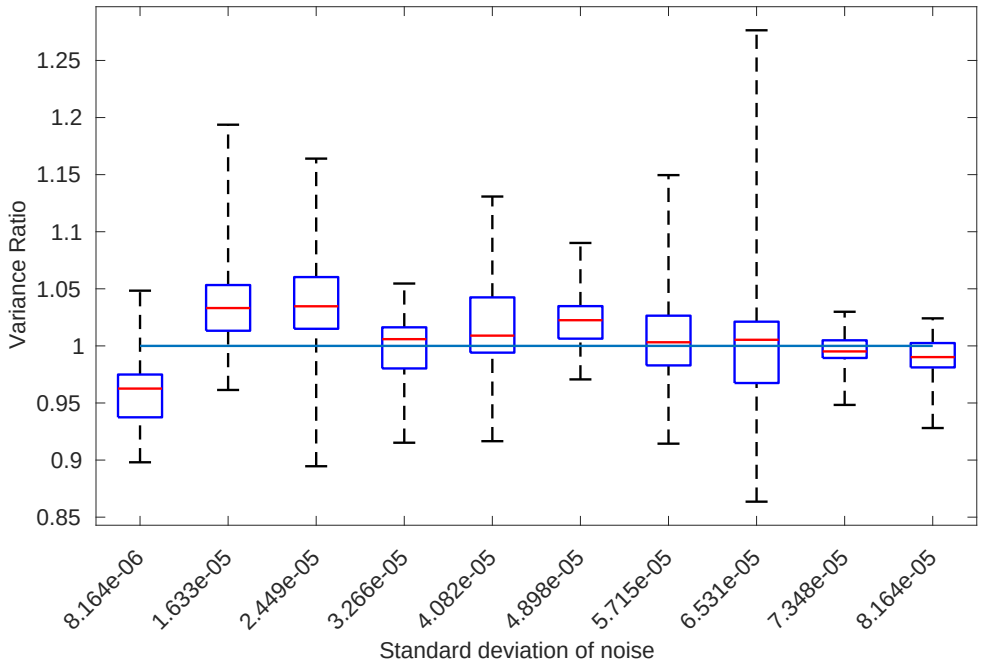
These ranges of standard deviations and velocity spacings were tested using $S_i \in \{100, 500, 1000\}$, and $L \in \{11, 22\}$. Table 4.1 summarizes the parameters tested.

| Parameter | Range |
|---|---|
| Direction | Along-track |
| $S_0$ | 1000 |
| $S_i$ | $\{100, 500, 1000\}$ |
| $L$ | $\{11, 22\}$ |
| Resolution | $200 \times 200$ |
| $\sigma$ | `linspace(8.1641e-6,8.1641e-5,10)` |
| Bias | $8.1641 \times 10^{-6}$ |
| Velocity Spacing | `linspace(`$\sigma$`(i)/10000,`$\sigma$`(i)*3/1000,30)` |

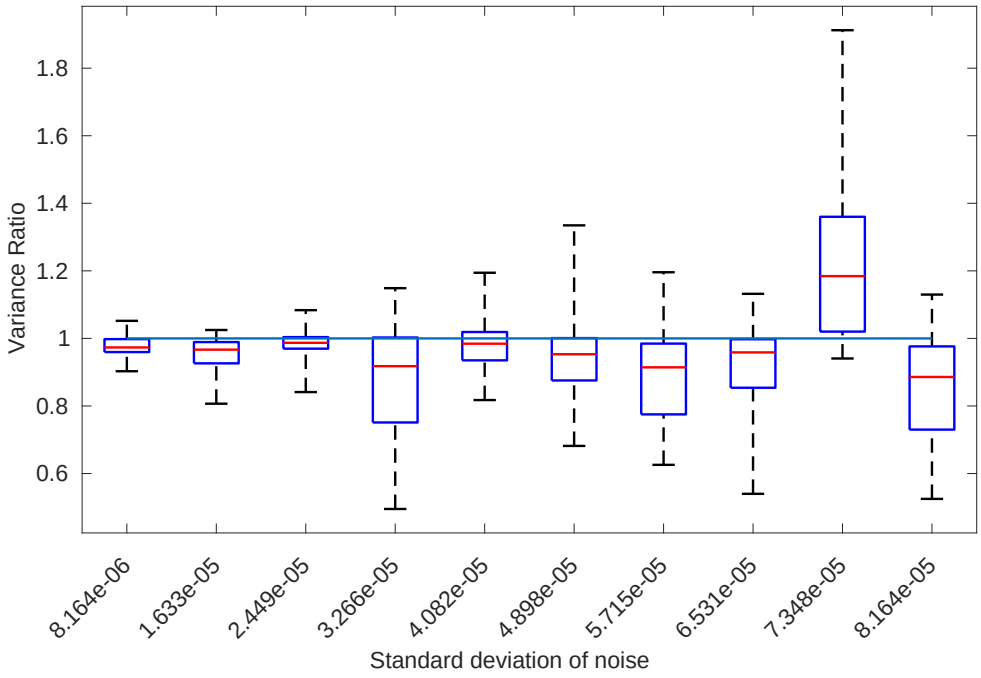
Table 4.1: GPS-D SAR Algorithm Testing Parameters

The testing program was written to test the entire range of standard deviations and velocity spacings in one program run. Each program run corresponds to a single value for $S_i$ and $L$. Thus, testing each combination of $S_i$ and $L$ required six program runs. One of the outputs of a program run is a log file that contains information including the variance of the noise, the velocity spacing, the variance of the noisy position vector minus the original position vector, and the variance of the corrected position vector minus the original position vector. The variance corresponding to the corrected position vector was divided by the variance corresponding to the noisy position vector to get a ratio. A ratio of less than one means the algorithm reduced the variance in position because the corrected position variance is smaller than the noisy position variance.

Whisker plots of the variance ratio were created from each program run and for each variance. The red lines in the plots indicate the medians of the sets of variance ratios, the edges of the dark blue boxes mark the 25th and 75th percentiles, and the ends of the black dotted lines indicate the maximum and minimum values. The horizontal line $y = 1$, colored in light blue, was added to each plot so it is easy to see what portion of the whisker plot lies in the desirable range of variance ratios. Overall, having a majority of the dark blue box below the light blue line means that on average the algorithm improved the position variance for that noise variance. Plots from the six program runs are shown in Figs. 4.19-4.21.

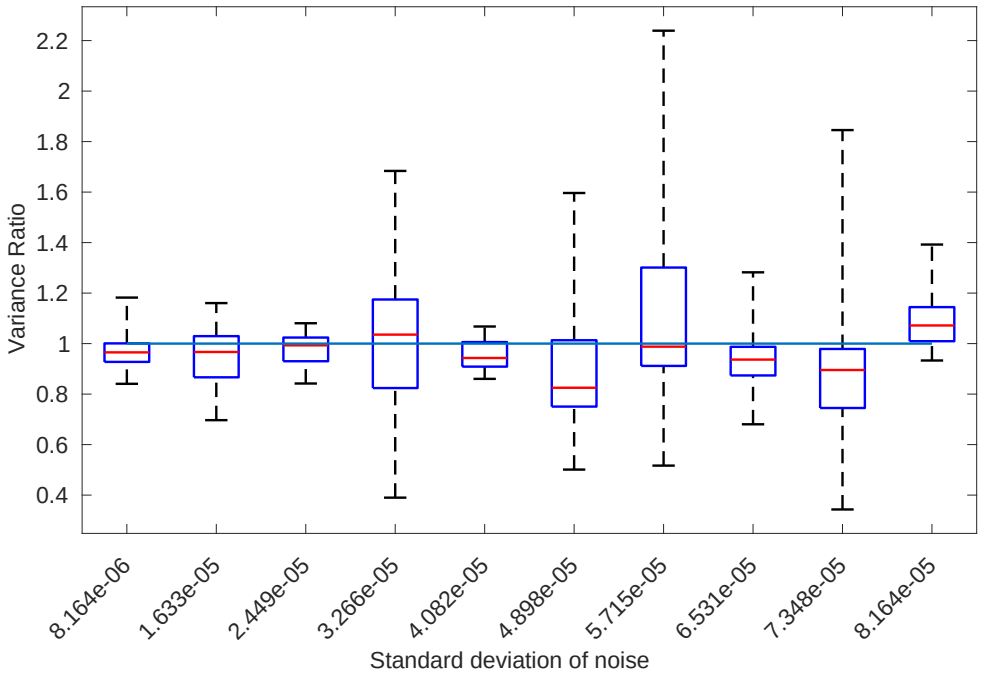



(a) $L = 11$

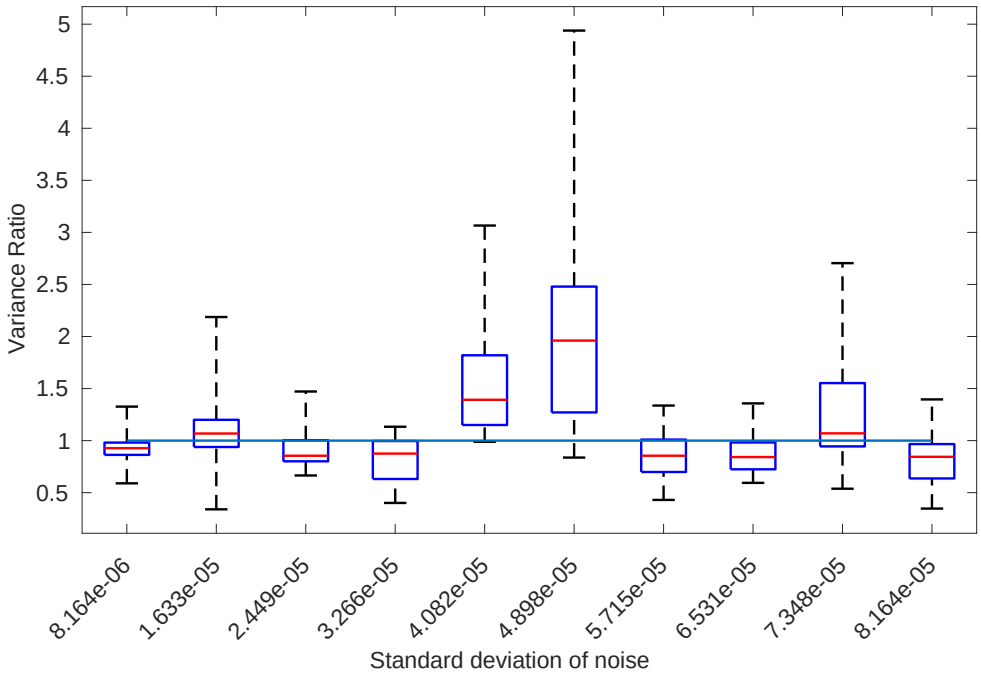



(b) $L = 22$

Fig. 4.19: Whisker plots of variance ratio with $S_i = 100$

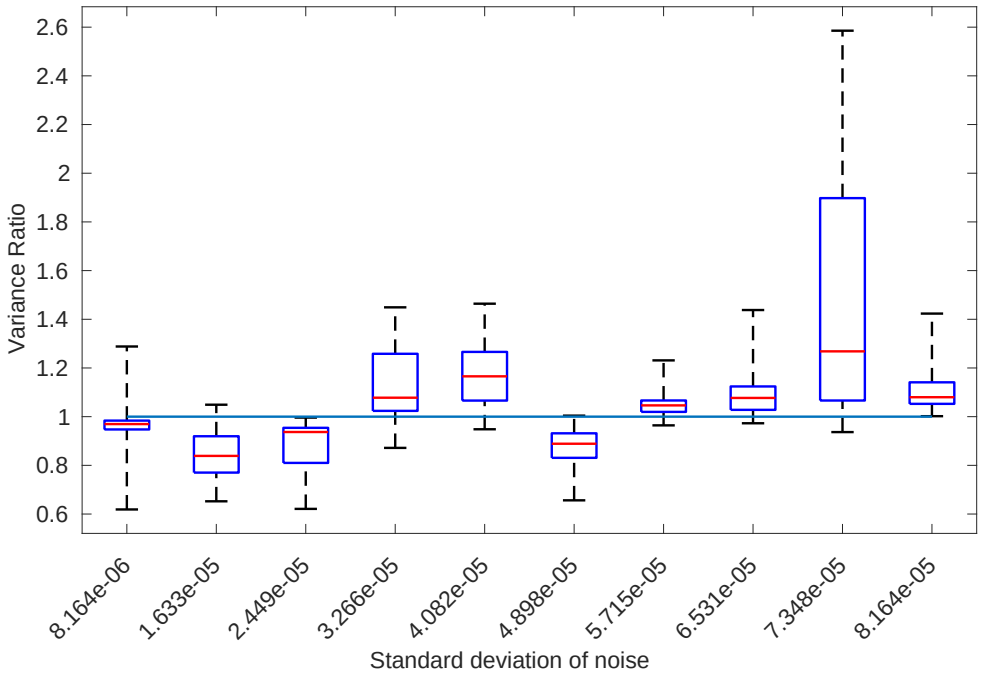



(a) $L = 11$

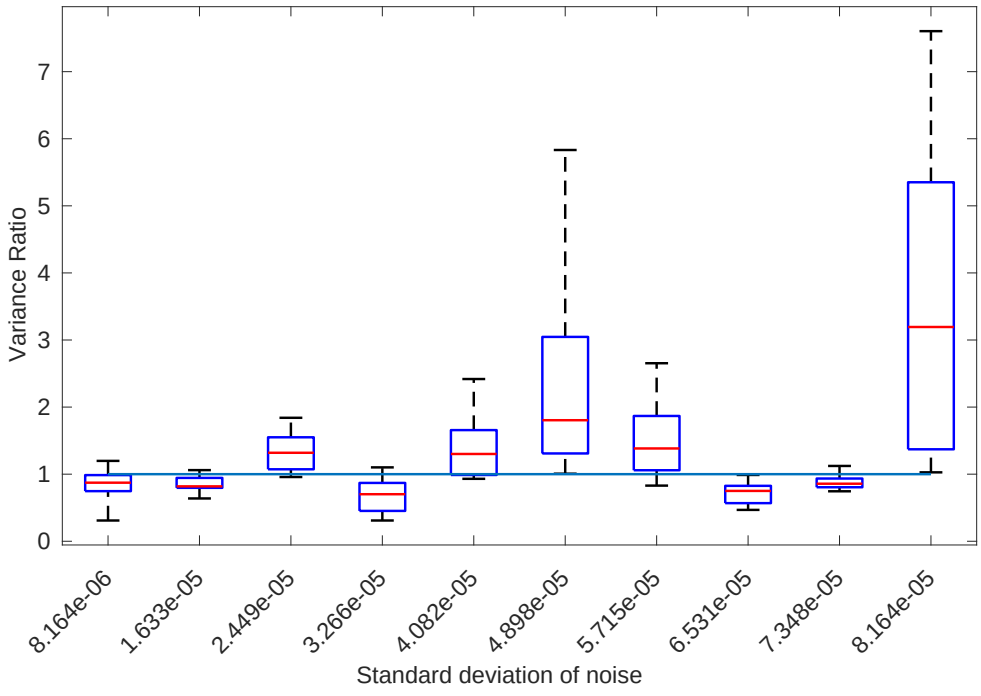



(b) $L = 22$

Fig. 4.20: Whisker plots of variance ratio with $S_i = 500$




(a) $L = 11$




(b) $L = 22$

Fig. 4.21: Whisker plots of variance ratio with $S_i = 1000$

When looking at the whisker plots, the scale of the $y$-axis has been left as the MATLAB

default, so the scales are not the same from one plot to another. This was done because using the same scale for all plots would result in some plots either being squished to the point where no useful information can be seen or cut off without showing all the data.

Examination of the whisker plots shows no general relationship between the variance of the noise and the behavior of the variance ratio. In Fig. 4.21, the algorithm seems to perform better with lower noise variance; however, this is not a strong trend, and the trend does not hold for the other figures. Overall, Figs. 4.19b and 4.20a seem to perform the best across the range of variances tested. The majority of each box falls below the $y = 1$ line, and those that fall above the line are only slightly above. No conclusion can be drawn from these plots alone, though.

To more thoroughly test the algorithm, the program was run ten times while holding $S_i$ and $L$ fixed at $S_i = 500$ and $L = 11$. Each program run generates new random noise, so running the program multiple times provides more insight into the effectiveness of the GPS-D algorithm. Three whisker plots are shown in Figs. 4.22-4.24 representing the best whisker plot of the ten runs, the most compact, and the most variable, respectively. The $y$-axis on each plot has been scaled uniformly for easier comparison between plots.

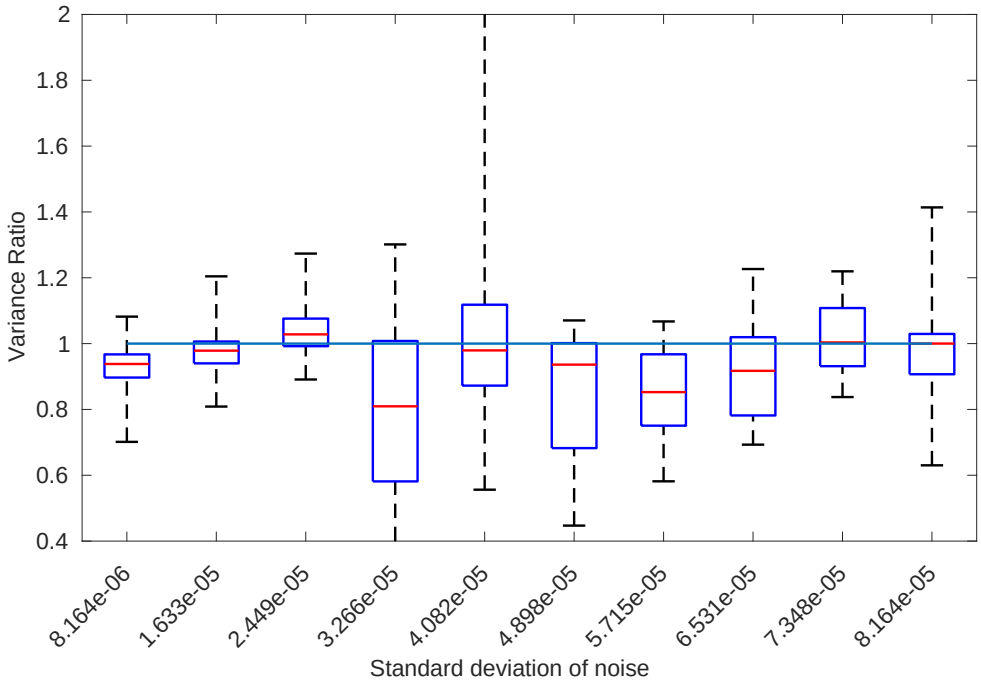



Fig. 4.22: $S_i = 500,\ L = 11$ best whisker plot

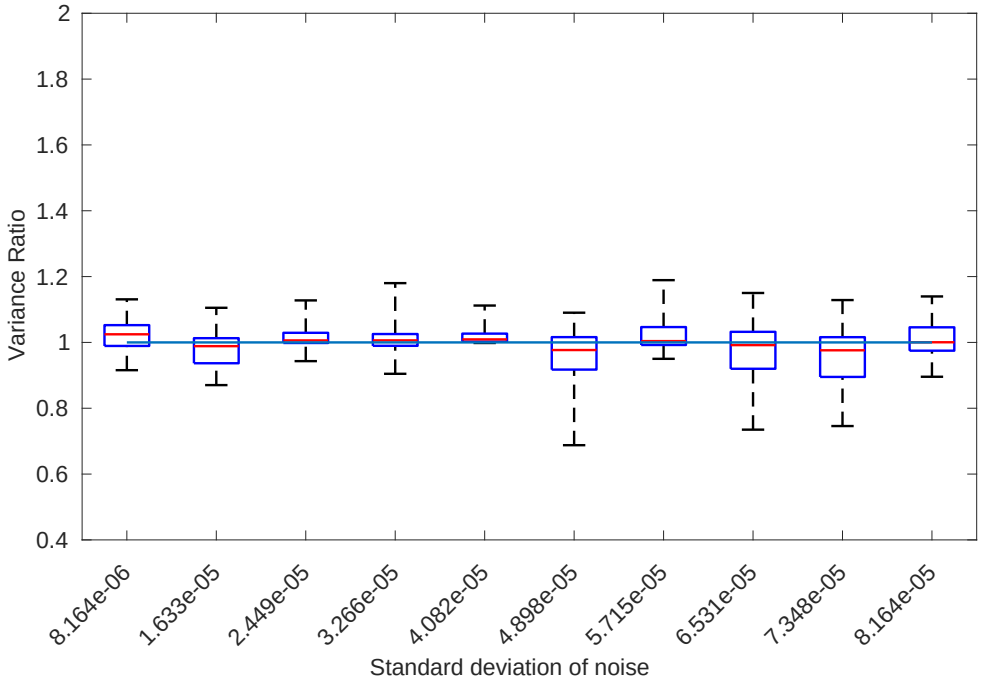



Fig. 4.23: $S_i = 500,\ L = 11$ compact whisker plot

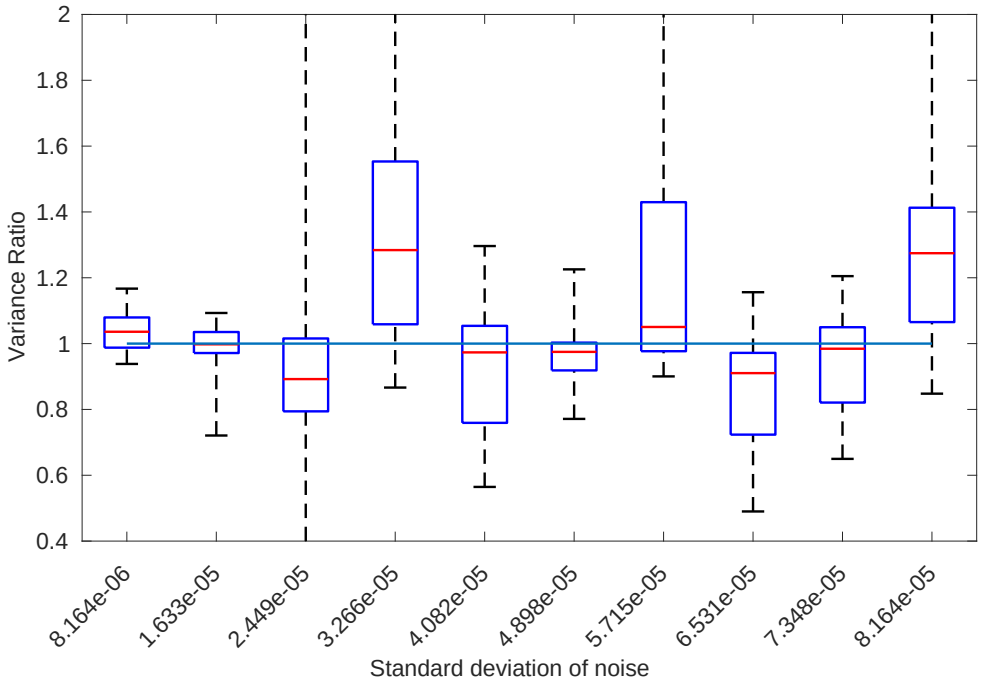


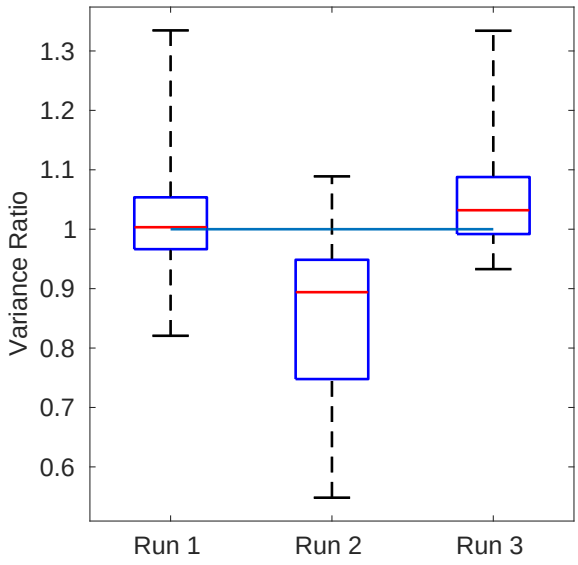

Fig. 4.24: $S_i = 500,\ L = 11$ variable whisker plot

From these whisker plots, it can be seen that the behavior of the algorithm is inconsistent across different samples of random noise. For one magnitude of noise variance, the algorithm can either decrease the variance, leave the variance nearly unaffected, or increase the variance. This is clearly shown with the variance equal to 3.266e-5, where Fig. 4.22 shows an improvement, Fig. 4.23 shows nearly no effect, and Fig. 4.24 shows a negative effect.

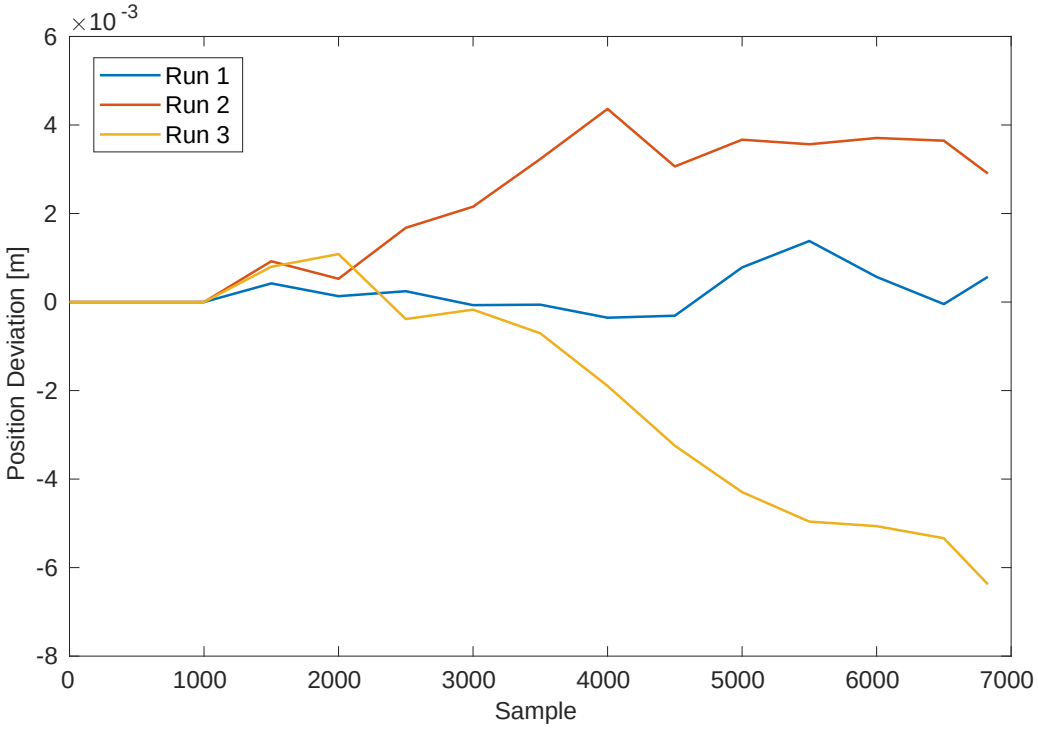
When examining the particular noise added in an attempt to determine when the algorithm works and when it does not, no trend can be seen. A natural conclusion to test is whether the algorithm performs better or worse when the noise causes large deviations from truth; however, there are times when large deviations cause the algorithm to perform better, and there are times when the algorithm performs worse.

For example, Fig. 4.25a shows three runs using the same noise variance. In the first run, the variance ratio is only minimally affected on average. In the second run, the variance ratio is improved. In the third run, the variance ratio is made worse. Fig. 4.25b shows

the noise corresponding to each run. The second and third runs show that when the noise causes large deviations from truth, the algorithm may make position estimates better, or it may make them worse.




(a) Whisker plots




(b) Noise vector

Fig. 4.25: Three runs with constant noise variance

When examining more of the data, the conclusion is the same. Sometimes the algorithm works well, and other times it does not. When the algorithm performs at its best, it can produce variance ratios as low as 0.2735; however, at its worst, the algorithm can produce variance ratios of 2.7707. These numbers come from taking the minimum and maximum variance ratios across every test. When comparing the average variance ratios for a specific noise variance, the minimum variance ratio is 0.8110 and the maximum is 1.3918. Factors that determine whether the GPS-D SAR algorithm provides improvements have not been determined yet and are a topic of future research.

# CHAPTER 5

# SUMMARY AND CONCLUSION

In this thesis, a novel approach to GPS-D navigation using SAR was developed and explored. The primary objective of this research was to address the challenge of navigating in an environment where GPS signals are obscured or unavailable. Since current navigation systems rely heavily on GPS, GPS-D navigation is a very active area of research. This thesis explored a method of GPS-D navigation using SAR that does not rely on point sources in the scene or comparison with onboard data.

First, kurtosis and entropy were tested as SAR image quality measures by forming multiple images with varying positional errors, scoring the images, and comparing the scores. Determining a robust image quality measure was critical to the performance of the GPS-D SAR algorithm. Kurtosis was chosen as an image quality measure due to its relationship to a Gaussian probability distribution, and entropy was chosen due to its relationship to a uniform probability distribution. Through testing, it was determined that kurtosis showed more promise as an image quality measure than entropy. The kurtosis testing plots have prominent peaks near zero positional error, whereas the entropy results are inconsistent.

Next, the GPS-D SAR algorithm utilizing the image quality measure was developed, and the design decisions of the algorithm were outlined. There are many degrees of freedom in this algorithm, so restrictions were placed to make testing the algorithm more computationally feasible. The algorithm was tested by injecting noise into the known position data and examining if the algorithm could correct for this noise and reduce positional variance. A range of noise variances were tested corresponding to the variance range between tactical and commercial grade IMUs. The algorithm was determined to have variable performance results. There are times when the algorithm performs well and can produce a variance ratio of 0.2735. This means the final variance in position after the algorithm was run was 0.2735 times the variance before using the algorithm. On the other hand, there are times

when the algorithm performs poorly and can produce variance ratios of 2.7707, meaning the positional variance was increased by a factor of 2.7707.

This thesis contributes to the field of GPS-D navigation by testing a novel method that assumes only that a well-formed SAR image has a different statistical structure than a poorly formed SAR image. While the results are variable, this research provides a stepping stone for future research in a previously unexplored area.

Due to limited access to real SAR data, the ability to test the algorithm was limited. Consequently, identifying the factors that influence the performance of the GPS-D SAR algorithm is a topic of future research. Additionally, testing multiple degrees of positional freedom is a natural progression of this research.

While the computational complexity would be dramatically increased, future research could test using the Viterbi algorithm in conjunction with the algorithm developed in this thesis. At each time step, the set of states would be the set of candidate positions around the nominal position, and the path metric would be the image quality measure. From this research, the best path metric would be kurtosis, but considering alternative image quality measures would also be valuable.

# REFERENCES


[1] D. O. Nitti, F. Bovenga, M. T. Chiaradia, M. Greco, and G. Pinelli, "Feasibility of Using Synthetic Aperture Radar to Aid UAV Navigation," *Sensors*, vol. 15, no. 8, pp. 18 334–18 359, Aug. 2015, number: 8 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/15/8/18334

[2] M. Greco, K. Kulpa, G. Pinelli, and P. Samczynski, "SAR and InSAR georeferencing algorithms for inertial navigation systems," in *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2011*, R. S. Romaniuk, Ed., vol. 8008, International Society for Optics and Photonics. SPIE, 2011, p. 80081O. [Online]. Available: https://doi.org/10.1117/12.904971

[3] M. Greco, G. Pinelli, K. Kulpa, P. Samczynski, B. Querry, and S. Querry, "The study on sar images exploitation for air platform navigation purposes," in *2011 12th International Radar Symposium (IRS)*, 2011, pp. 347–352.

[4] M. Greco, S. Querry, G. Pinelli, K. Kulpa, P. Samczynski, D. Gromek, A. Gromek, M. Malanowski, B. Querry, and A. Bonsignore, "Sar-based augmented integrity navigation architecture," in *2012 13th International Radar Symposium*, 2012, pp. 225–229.

[5] K. J. Kauffman, "Fast Target Tracking Technique for Synthetic Aperture Radars," Ph.D. dissertation, Miami University, 2009. [Online]. Available: https://etd.ohiolink.edu/apexprod/rws_olink/r/1501/10?clear=10&p10_accession_num=miami1250263416

[6] K. Kauffman, D. Garmatyuk, and J. Morton, "Efficient sparse target tracking algorithm for navigation with UWB-OFDM radar sensors," in *Proceedings of the IEEE 2009 National Aerospace & Electronics Conference (NAECON)*, Jul. 2009, pp. 14–17, iSSN: 2379-2027.

[7] K. Kauffman, Y. Morton, J. Raquet, and D. Garmatyuk, "Simulation Study of UWB-OFDM SAR for Dead-Reckoning Navigation," *Proceedings of the 2010 International Technical Meeting of The Institute of Navigation*, pp. 153 – 160, Jan. 2010. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=8792

[8] K. Kauffman, J. Raquet, Y. Morton, and D. Garmatyuk, "Simulation study of uwb-ofdm sar for navigation using an extended kalman filter," in *Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2010)*, 2010, pp. 2443–2451.

[9] ——, "Simulation Study of UWB-OFDM SAR for Navigation with INS Integration."

[10] ——, "Enhanced feature detection and tracking algorithm for UWB-OFDM SAR navigation," in *Proceedings of the 2011 IEEE National Aerospace and Electronics Conference (NAECON)*, Jul. 2011, pp. 261–269, iSSN: 2379-2027.

[11] ——, "Real-Time UWB-OFDM Radar-Based Navigation in Unknown Terrain," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 1453–1466, Jul. 2013, conference Name: IEEE Transactions on Aerospace and Electronic Systems.

[12] C. P. Lindstrom, "A study in gps-denied navigation using synthetic aperture radar," Master of Science, Utah State University, Logan, UT, 2020. [Online]. Available: https://engineering.usu.edu/ece/research/publications/thesis-sort

[13] E. B. Quist and R. W. Beard, "Radar odometry on fixed-wing small unmanned aircraft," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 1, pp. 396–410, 2016.

[14] E. B. Quist, P. C. Niedfeldt, and R. W. Beard, "Radar odometry with recursive-ransac," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1618–1630, 2016.

[15] E. B. Quist, "Uav navigation and radar odometry," Ph.D. dissertation, Brigham Young University, Provo, UT, 2015. [Online]. Available: https://scholarsarchive.byu.edu/etd/4439/

[16] P. C. Niedfeldt, E. B. Quist, and R. W. Beard, "Characterizing range progression of sar point scatterers with recursive ransac," in *2014 IEEE Radar Conference*, 2014, pp. 0712–0717.

[17] R. S. Christensen, J. Gunther, and D. Long, "Toward gps-denied navigation utilizing back projection-based synthetic aperture radar imagery," in *Proceedings of the ION 2019 Pacific PNT Meeting*, 2019, pp. 108–119.

[18] Z. Sjanic and F. Gustafsson, "Simultaneous navigation and sar auto-focusing," in *2010 13th International Conference on Information Fusion*, 2010, pp. 1–7.

[19] M. I. Duersch, "Backprojection for synthetic aperture radar," Ph.D. dissertation, Brigham Young University, Provo, UT, 2013. [Online]. Available: https://scholarsarchive.byu.edu/etd/4060

[20] A. Hyvärinen, J. Karhunen, and E. Oja, *Random Vectors and Independence*. John Wiley Sons, Ltd, 2001, ch. 2, pp. 13–56. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471221317.ch2

[21] ——, *Information Theory*. John Wiley Sons, Ltd, 2001, ch. 5, pp. 105–124. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471221317.ch5

[22] Sample byu microasar data. Data courtesy of David G. Long at Brigham Young University. [Online]. Available: https://www.mers.byu.edu/microASAR/CASIE_sample/