UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

TRACK-INITIATED BEAM SPOILING FOR IMPROVED TRACKING WITH DIGITAL PHASED-ARRAY RADARS

A THESIS
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for the
Degree of
MASTER OF SCIENCE

By

JOSHUA SHORT
Norman, Oklahoma
2022

TRACK-INITIATED BEAM SPOILING FOR IMPROVED TRACKING WITH
DIGITAL PHASED-ARRAY RADARS

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Nathan Goodman, Chair

Dr. Tian-You Yu

Dr. Justin Metcalf

Acknowledgments

First, I would like to thank my advisory chair, Dr. Nathan Goodman. Dr. Goodman has pushed me to produce my best work while encouraging and mentoring me in my research. Because of Dr. Goodman's mentorship, I will leave the University of Oklahoma as a better engineer. For that, I will always be grateful.

I want to thank the other members of my committee, Dr. Justin Metcalf and Dr. Tian-You Yu. Both have been crucial to my development at the Advanced Radar Research Center (ARRC). Their passion for their work has fostered an environment conducive to curiosity. I would also like to thank Dr. John Dyer for encouraging me to continue my academic career and giving me the personal goal to strive to be a lifelong learner.

I also want to thank my parents, Karen and Rodger, for teaching me the value of hard work and encouraging me to achieve my goals. Thank you, mom and dad, for holding me accountable and teaching me the lesson; that failure is only a failure if you give up. I want to thank my sister Amber for teaching me to always seek joy in life. My friend, David, thank you for being there and for your mentorship that led me to where I am today.

Finally, I would like to thank my beautiful wife, Taylor, who encouraged and supported me through my academic career. Without her strength as a wife and mother, I would not be here today. I want to thank my son Archer, whose curiosity never ceases to amaze me, reminding me every day that there is always something new to learn. To my daughter Opal, thank you for the joy you have for life, reminding me that there is always time for a dance. To my youngest, Aoife, thank you for always greeting me with a smile and

reminding your mother and me why we work so hard.

Table of Contents

List of Tables

List of Figures

Abstract

Radar systems have become highly dynamic with the advancements in all-digital radar architectures. All-digital radar architectures introduce the potential for dynamic beamforming.This thesis will detail the fundamentals that are the foundation of radar signal processing (RSP) and modeling a digital phased array radar. This thesis will detail the techniques used for digital beamspoiling. The intentional beamspoiling is intended to improve the trackers' ability to track a target continuously. When a high-speed target falls out of a beam due to a maneuver, the radar will spoil the transmit beam illuminating a wider scene. The wider illuminated scene allows for a higher likelihood of accurately detecting the target, allowing the tracker to track the target continuously. This thesis will discuss the theory and application of the trackers used in the simulation. With the beamspoiling and trackers, this thesis will analyze the ability of an all-digital phased array to track a target utilizing dynamic beamforming to improve the tracking performance. Finally, it will detail the improvement of the trackers' ability to track when utilizing beamspoiling for specific situations, allowing the radar to track targets for a more extended time. The results varied based on the amount a transmit beam was spoiled due to the loss in $\mathbf{SNR}$ that naturally occurs from the decrease in power density.

Chapter 1

Introduction

The ever-evolving enhancements in technology, such as analog to digital converters (ADCs) and field programmable gate arrays (FPGAs), have led to the increase in efficiency and quality of digital phased array radars [2]. DAR systems can utilize new technology enhancements to enable real-time, adaptive beamforming. [3]. Beamforming is a signal processing technique that utilizes multiple array elements whose collective signals simultaneously contribute to the desired angle [4]. At the same time, beamforming also disseminates non-desired angles, "steering" the beam in desired direction [4]. Technology that has increased the processing power of radar systems has led to various complex algorithms that interpret the information coming in and out of the radar systems. One class of algorithms that benefit from increased processing capacity is tracking algorithms, which can be particularly complex. Tracking algorithms have adaptive and cognitive properties that allow the radar to track a target in non-ideal environments where interference and noise are present [5]. Because of the increased functionality of these radar systems, there is a need to research and study radar signal processing (RSP) algorithms. This thesis will explore how beamforming can be used with existing tracking algorithms to improve the tracking of fast-moving targets.

DARs can "digitally" steer transmit and receive radar beams to specific directions by applying unique amplitude and phase weights to each element of an array [6]. This concept of digitally controlling an array's phase and amplitude weights to steer a transmit or receive beam electronically makes the DAR systems appropriate for Quadratic Phase Spoiling (QPS) [7]. QPS applies unique phase weights to each antenna element, spreading a transmitted beam's power. Spoiled beamforming leads to spreading the transmitted power of the radar to illuminate a wider scene at the cost of the decreased power density of the transmit beam [8]. Because of the broader distribution of transmitted power, there is a trade-off with DAR decreasing the transmit power's contribution to the SNR due to the decrease in power density of the transmitted spoiled beam [9].

Radar trackers can estimate a target's attributes and update that prediction with measured data to converge on a target approximated attributes [10]. For this thesis, the attributes are the position and velocity of the target. With the wide variety of trackers that the radar community utilizes, the two trackers utilized in this thesis are the Unscented Kalman Filter (UKF) and the Extended Kalman Filter (EKF). Both are appropriate for this project because of their ability to process data in a non-linear system and relatively low processing power, as opposed to a particle filter that is much more complex and requires more resources to complete the complex computations [11], [12].

The radar field has researched the performance of various trackers utilizing different adaptive beamforming techniques. Multiple Input Multiple Output (MIMO) radar systems can adaptively assign sections of an array to different targets based on received data, optimizing the ability of a radar to track multiple targets [13]. Using adaptive beamforming to null the effects of a jammer on a received signal to improve the performance of an EKF while tracking [14]. Utilizing automated beamspoiling while tracking a target in cases of fast-moving targets has yet to be widely researched. Utilizing automated beamspoiling while tracking a target allows the radar to illuminate a wider scene. The wider illuminated scene allows the radar to re-detect a lost target, allowing the tracker to re-establish the target's position and to continue tracking a target that would be lost. This thesis will explore an Unscented Kalman Filter (UKF) and an Extended Kalman Filter (EKF) ability to track fast-moving targets escaping a beam. The radar will utilize the QPS, and with each tracker, the radar will have the ability to enable and disable the QPS dynamically. Then the radar will be simulated with the functionality of the QPS for each tracker. With the completion of each scenario for the QPS enabled and disabled, we will analyze the trade-offs of SNR due to the decreased power density on the transmission of a QPS beam.

1.1 Thesis Outline

Chapter 2 will outline the rationale for choosing a DAR to design and simulate, the background of the primary radar concepts used to simulate a signal, and concludes with modeling the Additive White Gaussian Noise (AWGN) for the received signal. Chapter 3 covers the transmit and receive digital beamforming, the manipulation of the weights on transmit to generate a QPS beam, and the process of creating the receive beams necessary to perform monopulse processing. Chapter 4 covers the RSP techniques used to detect a target, beginning with the filter processing used to measure a target's range, Doppler frequency, correlated radial velocity, and the monopulse processing used to estimate a target's direction to the radar. Finally, Chapter 4 concludes with a description of the Cell Averaging Constant False Alarm Rate (CA-CFAR) algorithm used to detect potential targets when AWGN is present. Chapter 5 will cover the Kalman Filter (KF) foundation of the two trackers used in this project. Then Chapter 5 will discuss the Unscented Kalman Filter (UKF) and the Extended Kalman Filter (EKF) in detail, breaking down the theory of the two trackers. With the various algorithms and techniques used in this project, Chapter 6 will detail the Monte Carlo simulation and the conditions used to generate targets for the simulation randomly. Chapter 7 will discuss the preliminary results of each tracker's performance with and without the automated QPS. Chapter 8 will discuss the results and what effects the trade-off of the lower **SNR** had on the tracker's abilities to track a target. To conclude the thesis, Chapter 9 will cover the possible future work that can proceed with the research and analysis completed in this thesis.

Chapter 2

Radar Background, Transmit and Receive Signal Modeling

This chapter describes the mathematical processes and relationships between a pulse-Doppler radar and its intended target. This chapter will discuss the type of radar modeled for this project, introduce the waveforms used, and discuss the model of a delayed signal. Next, the discussion will lead to the design of the pulse train to measure a target's Doppler shift. The last subject this chapter will be adding and introducing Additive White Gaussian Noise (AWGN) into the system model.

2.1 Digital Phased Array

The type of radar model chosen for analysis in this thesis is a digital phased array radar (DAR), which models many similarities to HORUS. HORUS is an all-digital phased-array radar in development at the Advanced Radar Research Center (ARRC) at the University of Oklahoma [15]. The digital phased array radar was birthed in the 1970s when digital beamforming and increased resolution array techniques began to be researched [16]. DAR is rooted in the concept of a phased array antenna that can control the and amplitude at each antenna element, forming transmit and receive beams [17].

The rationale for choosing a DAR for this project is that a DAR allows for arbitrary control of phase weights for digital beamforming on both transmit and receive. The DAR capabilities allow the phase weights used for the transmit beamforming to be manipulated to increase the width of a beam to illuminate a wider scene with a transmitted beam. The wider beam decreases the transmitted beam's power density, decreasing the $\mathbf{SNR}$. With the wider transmitted beams, we can reduce the likelihood that a high-speed target escapes the transmit beam of a track update dwell. However, with the decrease in $\mathbf{SNR}$ created when spoiling a beam. When the $\mathbf{SNR}$ decreases to a certain threshold, the likelihood of a detected target decreases. Therefore, there are limits to spoiling a beam, with a loss in $\mathbf{SNR}$.

2.2 Background and System Model

Figure 2.1 displays how the different concepts and techniques work together to create the system model. The system model is a tool to assess the ability of the EKF and UKF to re-establish high-speed targets that were lost. Figure 2.1 describes radar system generating a waveform and collecting data to form a received data cube. Figure 2.1 displays when the radar signal processing techniques are applied to the data cube. Figure 2.1 explains the conditions for spoiling the transmit beams and ending an iteration as well as scheduling the delta and sum beams on receive. Finally, figure 2.1 shows the high-level process of how the UKF and EKF both take in initial estimates, predict an estimate, and update an estimate based on the detections.

2.3 Waveforms

For this Thesis, there are two types of waveforms that we have modeled. The first is the classic Linear Frequency Modulated (LFM) chirp, and the second is a Pseudo Random Binary Sequence (PRBS) waveform. The LFM signal has a smooth ramp-up and ramp-down time at the beginning and end of the pulse [4]. The LFM waveform allows for large com-

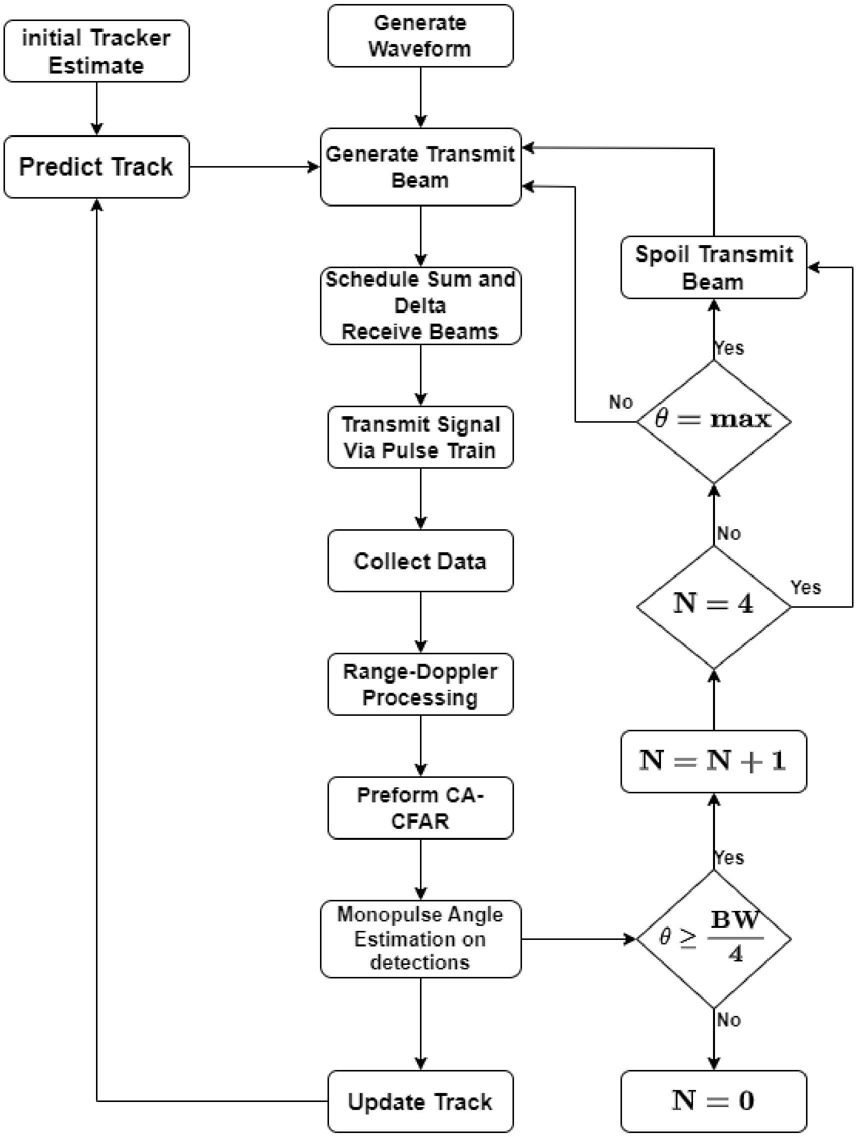


Figure 2.1: System Flow Diagram

pression ratios at the possible cost of ambiguity between the range and Doppler frequency (f_D) [18].

2.3.1 LFM Waveform

Modeling an LFM waveform is straightforward; first, the chirp, defined as a quadratic phase function,

$$\theta(t) = -2\pi \frac{B}{2} t + \pi \gamma t^2. \tag{2.1}$$

where,

$$\gamma = \frac{B}{T_x} \tag{2.2}$$

is the “chirp rate” of the LFM controlled by the Bandwidth (B) of the chirp and the pulse width (T_x). When represented as discrete-time samples prior to an analog waveform, the waveform must be sampled as a sampling frequency (f_s) at least B to satisfy the Nyquist theorem and avoid aliasing [19]. t is in continuous time. With that, the LFM waveform is,

$$\tilde{x}(t) = e^{j\theta(t)} a(t) \tag{2.3}$$

where $a(t)$ is the amplitude envelope whose peak matches the transmitted power (P_t), the envelope creates a smooth ramp-up and ramp-down time for the chirp.

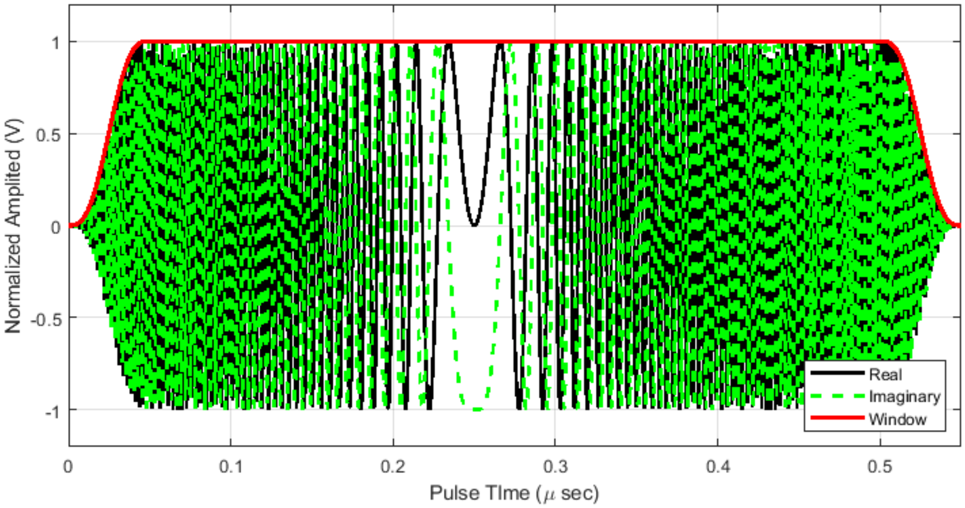


Figure 2.2: Real and Imaginary Parts of LFM with Windowing

Figure 2.2 presents these properties, with the rate of rotation of the real and imaginary parts of the complex LFM signal decreasing in the rate of cycles until $t = \frac{T_x}{2}$ then increasing from $t = \frac{T_x}{2}$ to $t = T_x$. The applied window function clips off the beginning and end of the LFM signal, creating smooth ramp-up and ramp-downs.

2.3.2 PRBS Waveform

The PRBS waveform is easily replicated and has many combinations of the phases 0 and π to create the signal [20], making the PRBS waveform a candidate for DAR systems that transmit multiple waveforms. The construction of the PRBS waveform, as given in the name "Pseudo Random Binary," is performed by generating a series of 1's and 0's with a length,

$$L = 2^N - 1 \tag{2.4}$$

where N is the number of registers in a given system [21]. With the 1's corresponding to a phase of π and 0's corresponding to a phase of 0. Each PRBS waveform consists of a unique "random" sequence ($\theta_{PSRB}(t)$), allowing it to randomly cycle between two "real" values

$$\tilde{x}(t) = e^{j\theta_{PSRB}(t)}. \tag{2.5}$$

where $\theta_{PSRB}(t)$ is a sequence of random π or 0 phases. Figure 2.3 displays the PRBS waveform. A slight delay is added to the changing phases of the PRBS waveform. The challenge with PRBS waveform is that the phase cannot change instantly and must smoothly vary. This challenge requires the signal to move through an amplitude of zero as the signal voltage changes between -1 and +1, resulting in a signal that does not have a constant amplitude. Figure 2.4 displays the inconsistent amplitude of a PRBS waveform.

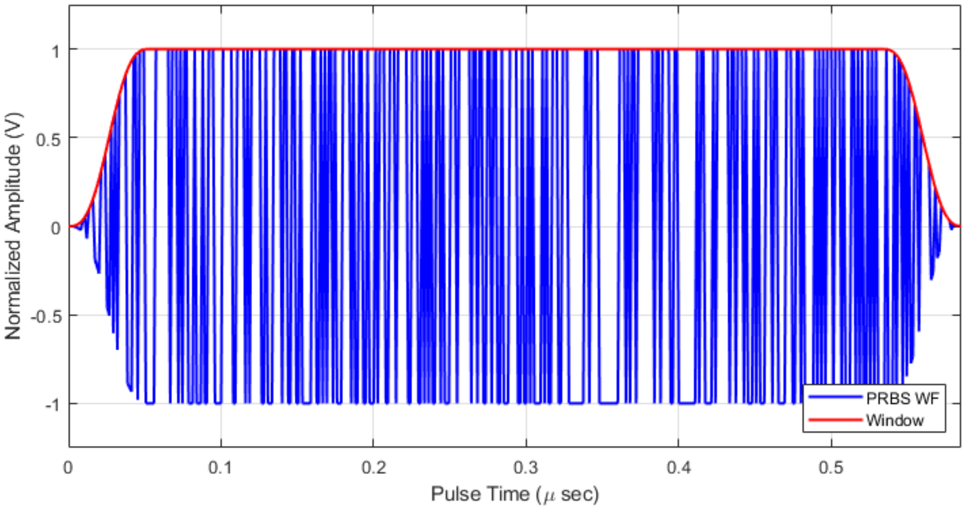


Figure 2.3: PRBS Waveform with Windowing

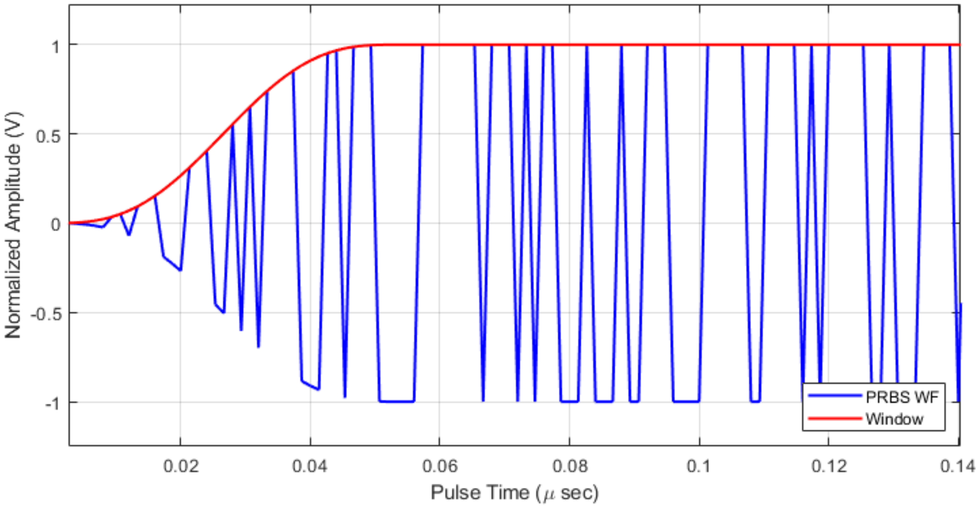


Figure 2.4: PRBS Delayed Slope

2.4 Range Modeling

One of the main goals of radar tracking is to follow the target range or distance from the radar. To determine a target's range, the time it takes the transmitted signal to travel to and from the target is measured and converted to a corresponding distance [4]. The relationship between time and range is

$$\tau = \frac{2R}{c} \tag{2.6}$$

where R is the range of a target, and c is the constant speed of light. The received radar waveform is modeled by applying the required time delay to the transmitted signal as

$$x_{Rx}(t) = \tilde{x}(t - \tau). \tag{2.7}$$

Figure 2.5 represents a delayed signal, shifted due to the target's range.

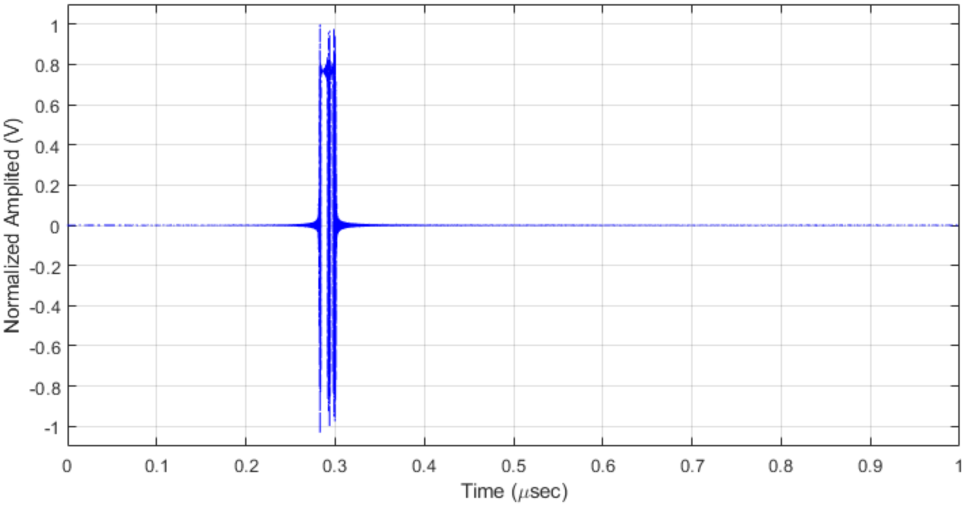


Figure 2.5: Simulated Delay of a PSRB Waveform

2.5 Doppler Modeling

Doppler shift (f_d) is the second of three characteristics that a radar can directly measure. It is challenging to measure accurately f_d with a single pulse due to the lack of time duration in a radar pulse. A pulse train needs to be introduced to more easily measure f_d. The definition is in the name; a radar will send out multiple pulses at evenly spaced intervals creating a “train” of pulses. A target’s radial velocity (V_r) is related to a target’s f_d according to,

$$f_d = \frac{2V_r}{\lambda} \tag{2.8}$$

where λ is the wavelength calculated at the carrier frequency (f_c). Essentially, f_d is the increase or decrease of frequency relative to f_c due to the target’s velocity, which causes relative motion between the radar and the target. For a target with constant velocity, the range of a target on the mth pulse is

$$R_m = R_0 - mV_rT_R, \qquad 0 \leq m \leq M - 1, \tag{2.9}$$

where m denotes a particular pulse number and T_r is the Pulse Repetition Interval (PRI), which is the time between pulses. Therefore the propagation delay on a particular pulse can be derived as

$$\tau_m = \frac{2R_m}{c} = \frac{2(R_0 - mV_rT_R)}{c}. \tag{2.10}$$

The received signal can now be modeled in both range and Doppler by modifying(2.7) to become

$$x_{Rx}(t, m) = a(t - \tau_m)e^{j\theta(t-\tau_m)}e^{-j2\pi f_c\tau_m}e^{j2\pi f_d t}. \tag{2.11}$$

2.6 Noise Model

A good model for radar receiver noise is additive white Gaussian noise(AWGN), which implies that the average noise power is flat over the entire spectrum and that the noise samples are uncorrelated. Additionally, it is assumed that the distribution of the noise is the same across the entire band of interest. “Gaussian” denotes that the noise values follow a normal distribution centered with a mean (μ_n) of 0 and a variance σ_n^2 matched to the noise power. The basic model of the noisy received signal ($x(t)$) is given as

$$x(t) = x_{Rx}(t, m) + q(t, m) \tag{2.12}$$

where $q(t, m)$ is the AWGN within a received signal. The receiver noise power is,

$$P_n = k_b TBF \tag{2.13}$$

where k_b is the Boltzmann constant $(1.38 \times 10^{-23} JK)$, T standard temperature in Kelvin (290K), F is the system's noise figure, and B is the bandwidth.

2.7 Summary

To summarize, Chapter 2 has discussed the design of the two waveforms used in this simulation. The process of simulating a received signal with the Range and Doppler information of a target. Furthermore, the modeling of unwanted AWGN added to a received signal. The next chapter will discuss modeling a target's direction and how that pertains to radar transmit and receive beams.

Chapter 3

Transmit and Receive Beamforming

An essential step for a DAR system is digital beamforming. With a DAR, the ability to form beams (beamforming) depends on the array's multiple phase centers that the signal processor can utilize [4]. Applying phase weights at each antenna element steers the transmit and receive beams in a given direction.

The transmitted and receive beams are pre-designed for our digital array model on each Coherent Processing interval (CPI) before transmitting the waveform. This chapter describes the beamforming model. First, we discuss the array design and how the emitter's coordinates are defined and cataloged. Next, The design of the transmit and receive pencil beams will be detailed. Manipulating the transmitted pencil beams into broader beams using QPS is also discussed. Finally, the design of the receive monopulse sum ($\sum$) and difference (Δ) beams will be detailed.

3.1 Array Design

We modeled a rectangular, uniform-spaced array in a North East Down (NED) coordinate system. NED coordinate systems are standard in radar systems, most commonly used in airborne radar systems. The NED coordinates are a Cartesian coordinate system where the x-axis now points "North," the y-axis points "East," and the z-axis points "Down" towards the center of the earth. A NED coordinate system depends on the local origin. For this model, the local origin is the radar. Figure 3.1 illustrates rotating the coordinate system from a traditional Cartesian to a NED system.

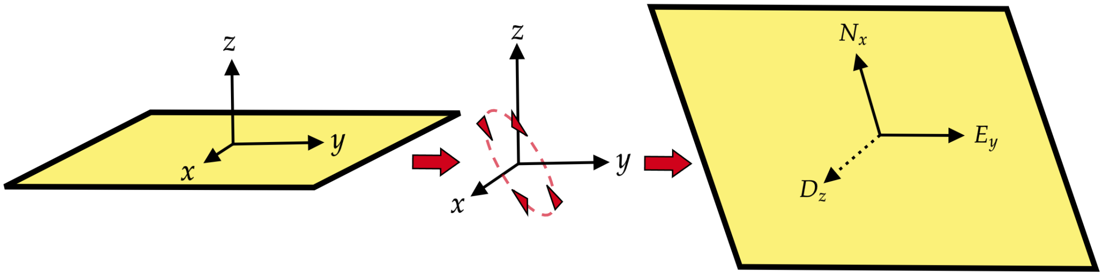


Figure 3.1: North East Down (NED) Rotation

While the introduction of the NED coordinate system is not imperative to the design of the phased array, a NED coordinate-based model allows for the radar and target positions to be accounted for in the same coordinate system. Each element within the array is defined by a NED-based coordinate and treated as a vector.

The elements are spaced $\frac{\lambda}{2}$ apart in both the north and east directions to ensure no aliasing in angular direction [22]. Figure 3.2 shows the rectangular array design.

3.2 Transmit and Recieve Beamforming

This section will cover the creation of both the transmit and receive beams. To begin, the wavenumber (k) is associated with the propagated signal, in units of $\frac{rad}{m}$ given by,

$$k = \frac{2\pi}{\lambda} = \frac{2\pi f_c}{c} \tag{3.1}$$

where f_c is the carrier frequency of the transmit signal and c is the speed of light. The wavenumber, k, is a scalar value, while the 3D wavenumber vector is defined as

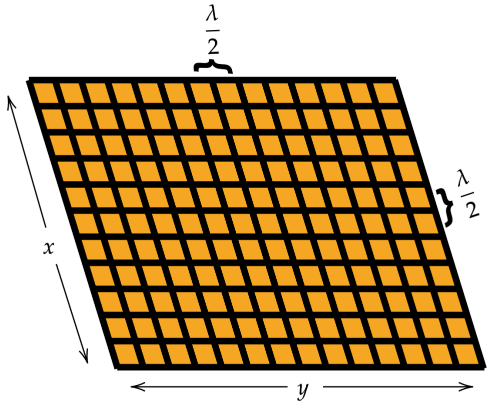


Figure 3.2: Array Layout

$$\bar{\mathbf{K}} = K_x\hat{x} + K_y\hat{y} + K_z\hat{z} \tag{3.2}$$

where

$$|\bar{\mathbf{K}}| = k. \tag{3.3}$$

Because of this, the unit vector in the direction of propagation can be defined as,

$$\bar{u}_k = \frac{\bar{K}}{k} = \frac{K_x\hat{x} + K_y\hat{y} + K_z\hat{z}}{k}. \tag{3.4}$$

A unit vector related to azimuth and elevation angles can be defined as

$$\hat{\mathbf{r}} = \begin{bmatrix} cos(\theta_{el})cos(\theta_{az}) & cos(\theta_{el})sin(\theta_{az}) & -sin(\theta_{el}) \end{bmatrix} \tag{3.5}$$

where $\hat{\mathbf{r}}$ is the spherical unit vector pointing outwards of the array, θ_{el} steers the transmit and receive beams in elevation, and likewise for θ_{az} in the azimuth direction. The transmit steering direction in a NED coordinate system,

$$\hat{\mathbf{k}}_{\mathbf{steer}} = -k\hat{r}. \tag{3.6}$$

The wavenumber vector $\hat{\mathbf{k}}$ is applied to the elements NED coordinates ($\mathbf{P_{NED}}$) and summed, thereby "steering" the array in a direction dictated by θ_{el} and ϕ_{az},

$$\phi_{tx} = e^{-j\mathbf{P_{NED}} \bullet \hat{\mathbf{k}}_{\mathbf{steer}}} \tag{3.7}$$

The model for the target's unit vector is,

$$\hat{\mathbf{u}}_{\mathbf{targ}} = \begin{bmatrix} \hat{x}_{targ} & \hat{y}_{targ} & \hat{z}_{targ} \end{bmatrix}. \tag{3.8}$$

The receive signal complex-valued weights with angles for the array are,

$$\phi_{tar} = e^{-j\mathbf{P_{NED}} \bullet \hat{\mathbf{k}}_{\mathbf{targ}}} \tag{3.9}$$

where $\hat{\mathbf{k}}_{\mathbf{targ}}$ is the wavenumber vector corresponding to the target's position, and we mathematically model it by scaling the target's unit vector ($\hat{\mathbf{u}}_{\mathbf{targ}}$) by k

$$\hat{\mathbf{k}}_{\mathbf{targ}} = k\hat{\mathbf{u}}_{\mathbf{targ}} \tag{3.10}$$

To model the target within the transmit beam,

$$\phi_{tar,tx} = \phi_{tx}^{*}\phi_{targ} = e^{-j\mathbf{P_{NED}} \bullet \hat{\mathbf{k}}_{\mathbf{steer}}} e^{-j\mathbf{P_{NED}} \bullet \hat{\mathbf{k}}_{\mathbf{targ}}} = e^{-j\mathbf{P_{NED}}(\hat{\mathbf{k}}_{\mathbf{steer}} - \hat{\mathbf{k}}_{\mathbf{targ}})}. \tag{3.11}$$

The weights are conjugated so that when the steering vector and the target's wavenumber vector match, the phases are zero, and the coherent sum will add to a peak within the beam corresponding to the direction of the target.

3.3 Quadratic Phase Beam Spoiling

Ideally, beams would never need to be spoiled because of the loss of gain that occurs due to the spread of power that spoiling induces [4]. However, high-speed targets can be re-established utilizing a spoiled beam when a target begins to fall out of a beam due to a high-speed maneuver. A spoiled beam that illuminates a broader scene volume is preferred to re-establish the target location. There are processes used to identify the weights needed for a specific beam shape [8]. One widely-used approach to elevation beam spoiling is to apply quadratic phase weighting vertically across the array elements [23]. We will explore the utilization of the quadratic phase weights in this work.

The phase weights derived to synthesize a spoiled beam in this simulation are from Leifer [23], who modeled his spoiling factor (S_F) after Kinsey [24], [8]. First, we will derive Leifer's S_F from the Quadratic phase of an LFM chirp. A traditional chirp waveform is,

$$x(t) = e^{j\pi\gamma t^2} = x(t) = e^{j2\pi\frac{\gamma}{2}t^2} \qquad 0 \leq t \leq T_p \tag{3.12}$$

where T_p is the pulse time, the chirp rate

$$\gamma = \frac{B_{LFM}}{T_p} \tag{3.13}$$

where γ controls the Quadratic phase with the bandwidth of the chirp (B_{LFM}) and T_p with the units of $\frac{Hz}{s}$. Substituting LFM's γ for γ_x and t for x,

$$\theta(x) = e^{j2\pi\frac{\gamma_x}{2}x^2} \qquad 0 \leq x \leq L_x. \tag{3.14}$$

where L_x is the length of the array in a particular dimension, and x is the exact position of each element (nd_x). The rate at which the weights are spread in the spacial dimension are

$$\gamma_x = \frac{S_F}{L_x^2} = \frac{S_F}{(N\frac{\lambda}{2})^2}. \tag{3.15}$$

Where N is the number of elements in a single dimension, and $\frac{\lambda}{2}$ is the distance between each element, giving γ_x units of $\frac{m^{-1}}{m}$. Knowing that

$$B = \gamma_x L_x \tag{3.16}$$

and

$$B_s = \frac{1}{L_x}, \tag{3.17}$$

then the Spoiling factor (S_F) can be derived from

$$S_F = \frac{B}{B_s} = \frac{\gamma_x L_x}{\frac{1}{L_x}} = \gamma_x L_x^2 \tag{3.18}$$

Substituting (3.15) into (3.14) yields

$$\theta(x) = e^{j2\pi\frac{1}{2}\frac{S_F}{(Nd_x)^2}(nd_x)^2} \tag{3.19}$$

The factors of two introduced (3.19) and the distance between each element cancel out to give the quadratic phase weights derived by Leifer [24]:

$$\theta_{spl}(x) = e^{j\pi\frac{S_F}{N^2}n^2} \qquad 0 \leq n \leq N \tag{3.20}$$

where n is the antenna element that is weighted by S_F and N is the number of elements in a particular dimension.

As for the application in this thesis, S_F is a variable that can alter the weights of $\theta_{spl}(x)$. Creating an initial pencil beam on transmit from (3.7), the vector ϕ_{tx} is then reshaped into an $N_x \times N_y$ matrix where each position within the matrix corresponds to an element's position,

$$\phi_{tx,m} = \begin{bmatrix} \phi_{tx,1,1} & \phi_{tx,1,2} & \cdots & \phi_{tx,1,n_x} \\ \phi_{tx,2,1} & \phi_{tx,2,2} & \cdots & \phi_{tx,2,n_x} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{tx,n_y,1} & \phi_{tx,n_y,2} & \cdots & \phi_{tx,n_y,n_x} \end{bmatrix}. \tag{3.21}$$

Next, $\theta_{spl}(x)$ is applied to the corresponding dimension via point-wise multiplication, whether the transmit beam is spoiled in azimuth, elevation, or both. This process is shown in (3.22) and (3.23), where (3.22) applies the spoiling phases in the elevation and (3.23) spoiling in both.

$$\phi_{tx,EL} = \phi_{tx,m}\theta_{spl,x}(x) \tag{3.22}$$

$$\phi_{tx,,AZ} = (\theta_{spl,y}(x))^T\phi_{tx,m}\theta_{spl,x}(x) \tag{3.23}$$

Similarly to γ and the time-bandwidth product from the LFM chirp, the spoiling factor proves less effective at low spoil factors. Figure 3.3 illustrates an assortment of beams modified by the QPS spoiling factor. Notice how the spoil factor has little effect on the transmitted beam until it reaches a factor of four. At that point, the beams begin to degrade by the intended spoil factor.

Because the transmit beam is spoiled, there is a need for multiple receive beams to populate the spoiled area. The number of receive beams needed is the size of S_F, each receive beam evenly shifted by the $3dB$ beams width in K space by a factor of $K_{x,res}$ and $K_{y,res}$, respectively. $K_{x,res}$ and $K_{y,res}$ are the $3dB$ of the transmit and receive pencil beams in K space,

$$K_{x,res} = \frac{Bw_{EL,3dB}}{(N_{ang}-1)} \tag{3.24}$$

and,

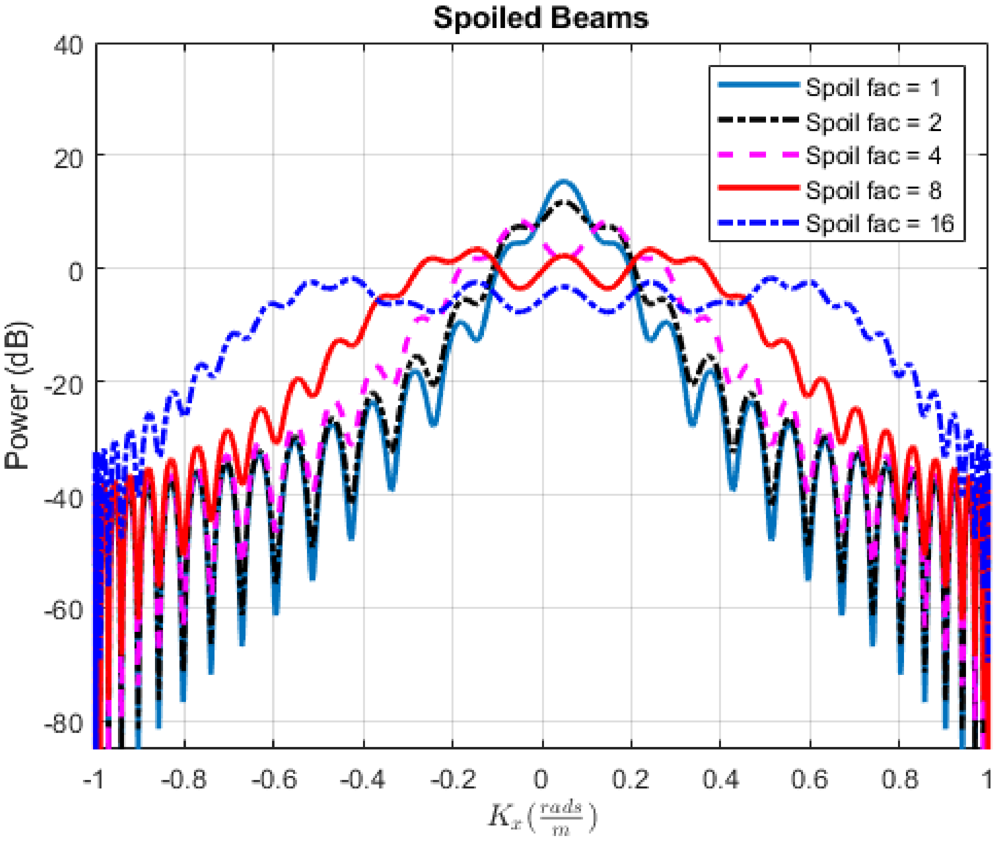


Figure 3.3: Comparison of Different Spoil Factors

$$K_{y,res} = \frac{Bw_{AZ,3dB}}{(N_{ang} - 1)} \tag{3.25}$$

$Bw_{EL,3dB}$ and $Bw_{AZ,3dB}$ are the receive beams' $3dB$ beamwidths in elevation and azimuth, respectively. N_{ang} represents the number of angles in a particular monopulse table.

$$x_{shift} = nK_{x,res} \qquad -\frac{N_x - 1}{2} \leq K_{x,res} \leq +\frac{N_x - 1}{2} \tag{3.26}$$

$$y_{shift} = nK_{y,res} \qquad -\frac{N_y - 1}{2} \leq K_{y,res} \leq +\frac{N_y - 1}{2} \tag{3.27}$$

x_{shift} and y_{shift} are vectors holding the shift factors for each receive beam for a specific spoiling factor.

Similar to (3.5), the unit vectors pointing in the azimuth and elevation directions are found and are obtained by,

$$\hat{\theta} = \begin{bmatrix} cos(\theta_{az})sin(\theta_{el}) & sin(\theta_{el})sin(\theta_{az}) & cos(\theta_{el}) \end{bmatrix}, \tag{3.28}$$

$$\hat{\phi} = \begin{bmatrix} -sin(\theta_{az}) & cos(\theta_{az}) & 0 \end{bmatrix}. \tag{3.29}$$

The angles $\hat{\theta}$ and $\hat{\phi}$ shift the receive beams in elevation and azimuth, respectively. The shift factors from x_{shift} and y_{shift} can be applied to $\hat{\theta}$ and $\hat{\phi}$ giving each receive beam its own unique shift to populate the entire illuminated scene of a spoiled transmit beam,

$$\hat{\theta}_{\mathbf{shift}} = \hat{\theta}x_{shift} \tag{3.30}$$

and,

$$\hat{\phi}_{\mathbf{shift}} = \hat{\phi}y_{shift}. \tag{3.31}$$

$\hat{\theta}_{\mathbf{shift}}$ and $\hat{\phi}_{\mathbf{shift}}$ are then applied to the original $\hat{\mathbf{k}}_{\mathbf{steer}}$, to give each received beam its unique $\hat{\mathbf{k}}_{\mathbf{steer}}$ evenly spaced within the spoiled transmit beams spoiled area;

$$\hat{\mathbf{k}}_{\mathbf{shift}} = \hat{\mathbf{k}}_{\mathbf{steer}} + \hat{\theta}_{\mathbf{shift}} + \hat{\phi}_{\mathbf{shift}}. \tag{3.32}$$

But, due to the addition of $\hat{\theta}_{\mathbf{shift}}$ and $\hat{\phi}_{\mathbf{shift}}$ to $mathbfhatk_{steer}$, $\hat{\mathbf{k}}_{\mathbf{shift}}$ is no longer a true unit vector and needs to be re-weighted by its magnitude,

$$\hat{\mathbf{k}}_{\mathbf{shift}} = \frac{\hat{\mathbf{k}}_{\mathbf{shift}}}{|\hat{\mathbf{k}}_{\mathbf{shift}}|}. \tag{3.33}$$

With that, $\hat{\mathbf{k}}_{\mathbf{shift}}$ can now be weighted by the wavenumber, giving each receive beam a unique steer vector,

$$\hat{\mathbf{k}}_{\mathbf{shft,str}} = -k\hat{\mathbf{k}}_{\mathbf{shift}} \tag{3.34}$$

evenly spaced in azimuth and elevation and populating the entire scene that the spoiled transmit beam is illuminating.

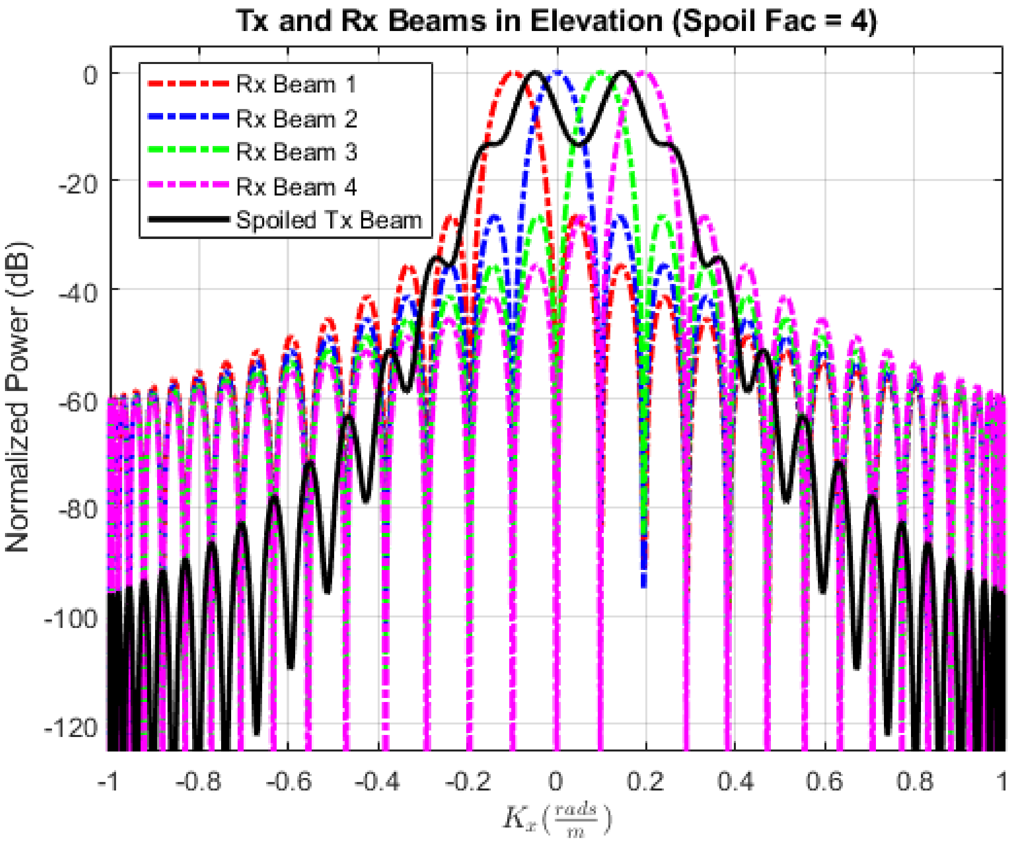


Figure 3.4: Elevation Beams

Modifying (3.7) with the newly shifted steering vectors for each receive beam produces,

$$\phi_{rx,n} = e^{-j\mathbf{P_{NED}} \bullet \hat{\mathbf{k}}_{\mathbf{shft,str}}}. \tag{3.35}$$

A transmit beam spoiled by a factor of four in elevation is displayed in figure 3.4 with four evenly spaced receive beams to cover the entire scene illuminated by the spoiled transmit beam.

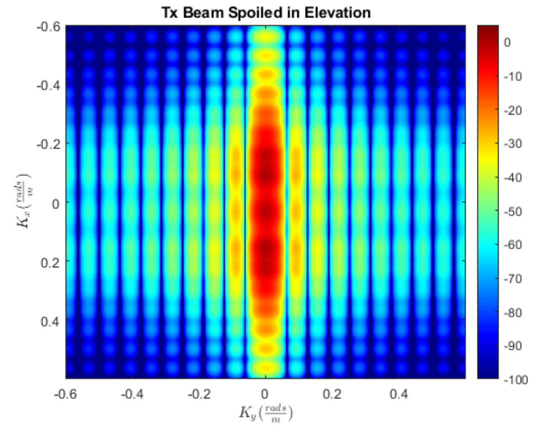


Figure 3.5: Spoiled Elevation Beam Tx

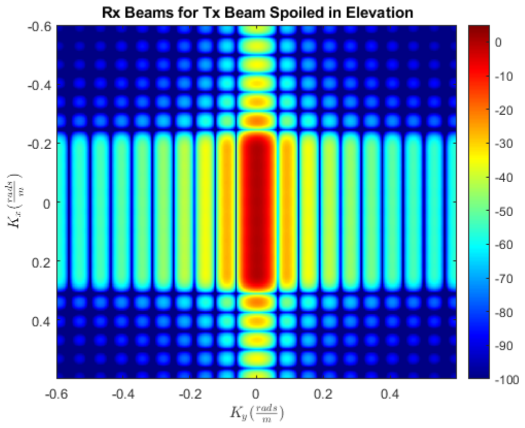


Figure 3.6: Spoiled Elevation Beam Rx

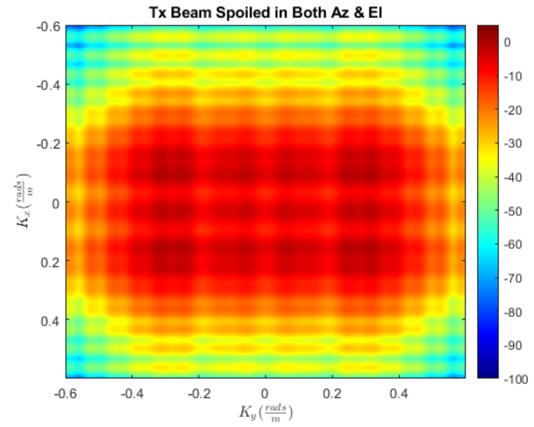


Figure 3.7: Spoiled El/Az Beam Tx

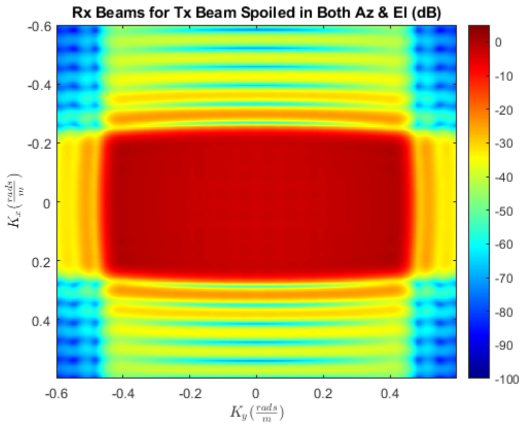


Figure 3.8: Spoiled El/Az Beam Rx

Figure 3.5 shows another normalized transmit beam spoiled in elevation by a factor of eight. The transmit beam has degraded by the spoiling weights, and the transmitted power is less concentrated and spread out. Figure 3.6 shows the the combined power patterns of the evenly spaced receive beams that capture the illuminated scene created by the transmitted beam. The power of the receive beams is much denser due to the finely spaced receive beams. The finely spaced receive beams are essential to capturing any received energy from the degraded spoiled beam.

In the final example of the QPS capability, a transmit beam is spoiled by a factor of eight in elevation and 16 in azimuth. The normalized transmit beam is shown in a decibel scale in figure 3.7. The power of the transmitted beam is spread out, creating a lower power density on transmission. Because of the low power density, a received signal is likelier to have a low SNR. Because the transmit beam is spoiled by factors eight and 16. There must be at least 128 beams to fill the volume illuminated by the spoiled beam. Figure 3.8 displays those receive beams. The 128 receive beams can capture a received signal transmitted by the degraded transmit beam.

3.4 Monopulse Beams

Because of the type of angle estimation used in this analysis, this section will cover the extra steps needed for the received beams so received data can be processed using monopulse. While there are several steps within monopulse processing, this section will concentrate on the receive beamforming part of the process. For a 2D array, there are two difference beams and one sum beam on receive [25].

An early type of angle tracking in radar was lobe switching [26]. This technique does not point a beam directly at a target but instead points two beams offset to either side of the target, switching rapidly [26]. Monopulse tracking is an offshoot of lobe switching; more accurately, the Δ beams in monopulse processing originated from lobe switching [26].

For the difference beams, the two-dimensional array is divided into four equal-sized subarrays on receive. As shown in figure 3.9, each quadrant produces one of the receive beams displayed in figure 3.10. The Quadrants are combined to create the elevation and azimuth channels for the difference beams. The channels used to build the sum and difference beams are given as,

$$Chan1_{az} = Q_1 + Q_2, \tag{3.36}$$

$$Chan2_{az} = Q_3 + Q_4, \tag{3.37}$$

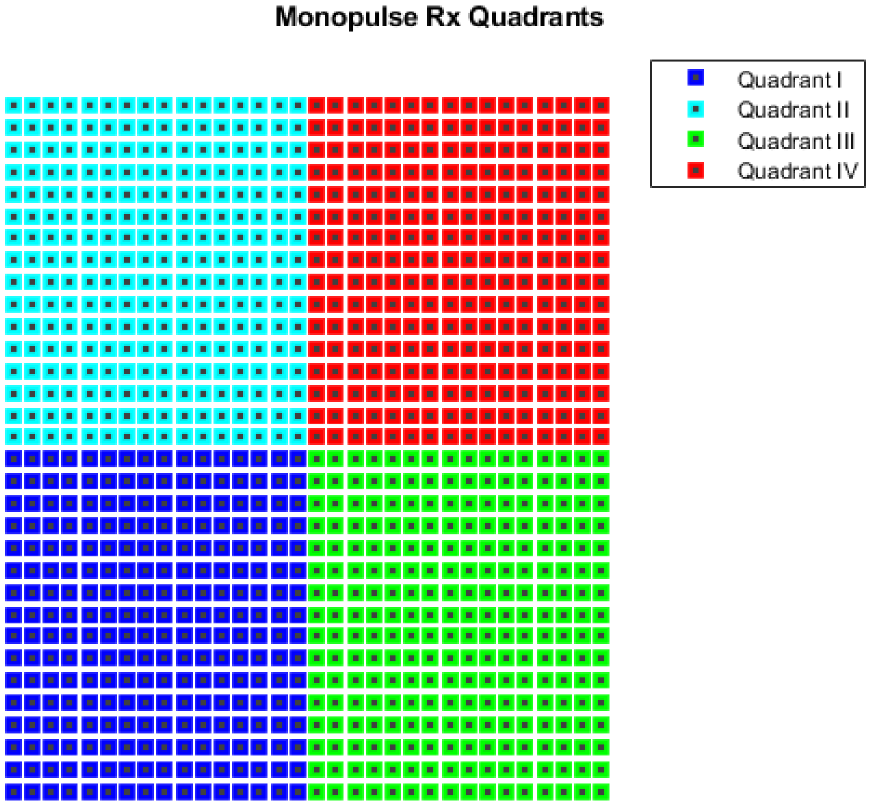


Figure 3.9: Monopulse Receive Beam Quadrants

$$Chan1_{el} = Q_2 + Q_4, \tag{3.38}$$

and,

$$Chan2_{el} = Q_1 + Q_3. \tag{3.39}$$

The difference beams for the respective dimensions are created by subtracting their respective channels,

$$\Delta_{AZ} = Chan1_{az} - Chan2_{az}, \tag{3.40}$$

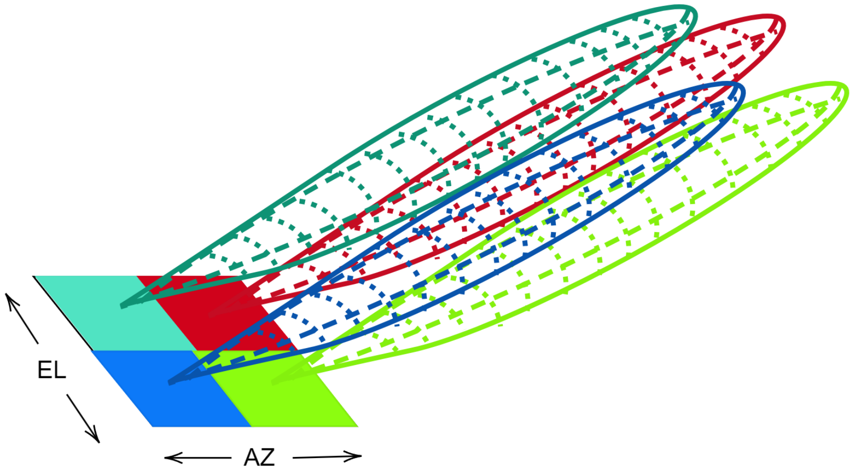


Figure 3.10: Monopulse Receive Beam Quadrants

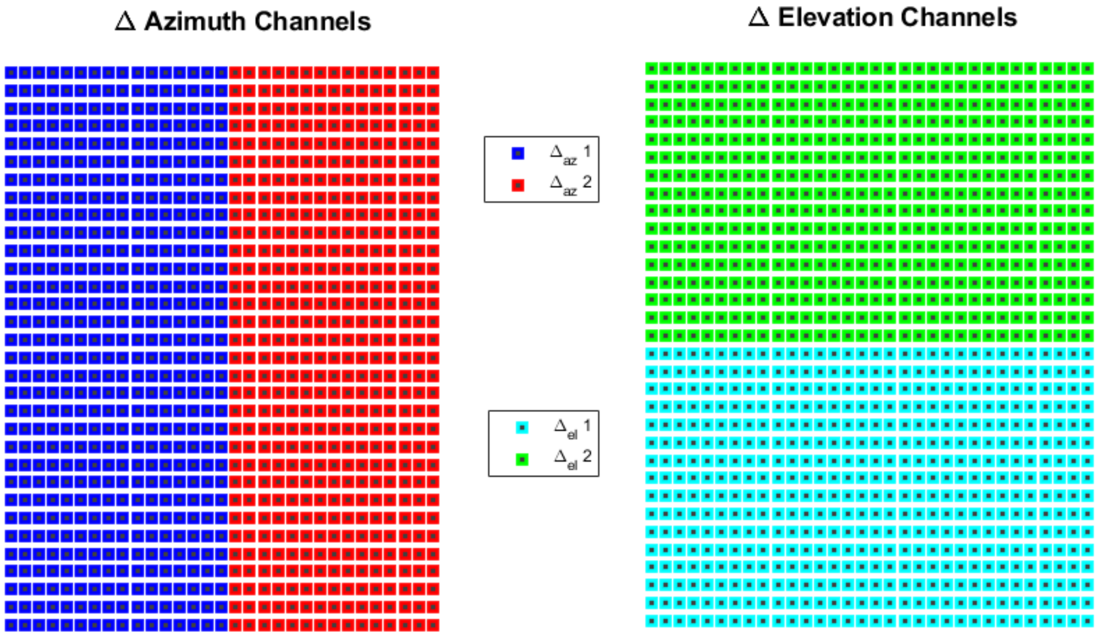


Figure 3.11: Monopulse Delta Quadrants in Elevation and Azimuth

$$\Delta_{EL} = Chan1_{el} - Chan2_{el}. \tag{3.41}$$

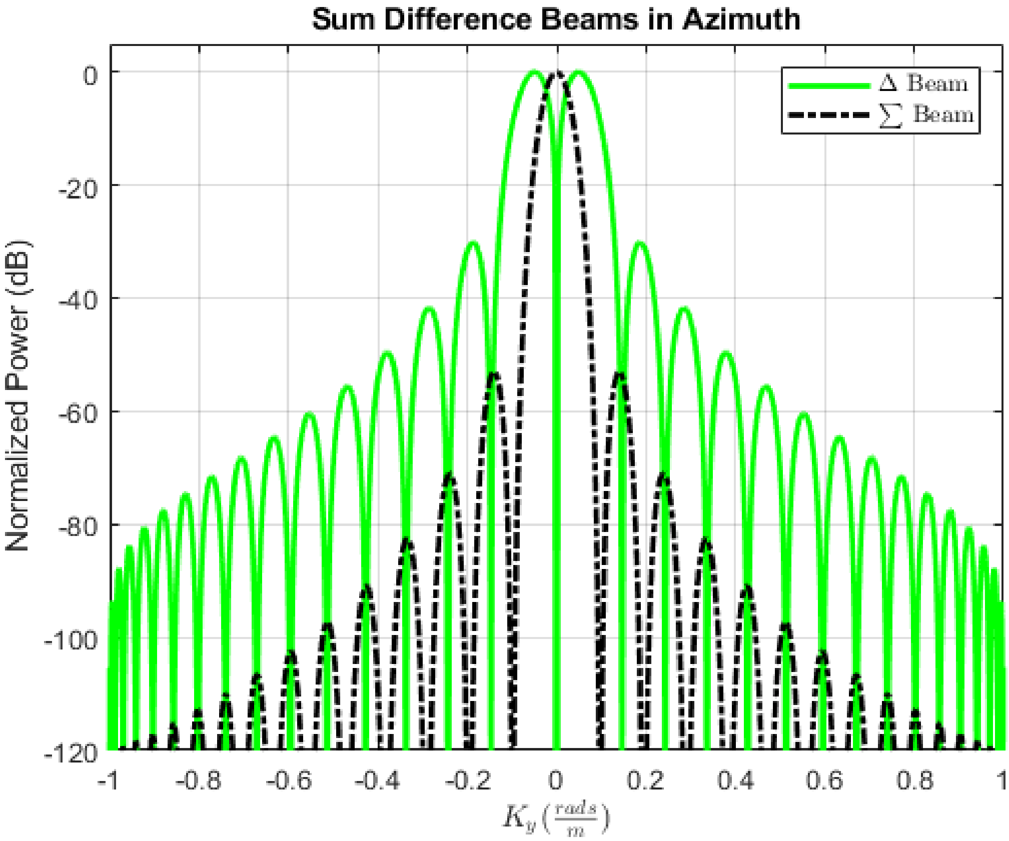


Figure 3.12: Monopulse Sum and Difference Beams in Azimuth

As for the Sum ($\sum$) beam, it is simply the sum of all four quadrants to create a pencil beam,

$$\sum = Q_1 + Q_2 + Q_3 + Q_4. \tag{3.42}$$

The $\sum$ and Δ_{AZ} beams are displayed in figure 3.12, where Δ_{AZ} has a null located at the center of the $\sum$ beam. In contrast, the $\sum$ beam is a traditional pencil beam. The post-processing that utilizes the $\sum$ and Δ_{AZ} beams will be detailed in Section 4.3 of Chapter 4.

3.5 Summary

This chapter has covered the different beamforming techniques used in this particular project. This chapter also detailed the unique QPS technique used to illuminate a broader area. Lastly, this chapter discussed the monopulse beams formed on receive to prepare the received data for monopulse processing. The next chapter will discuss the different processing techniques used for identifying a target's range, velocity, and direction, as well as a technique for distinguishing targets from noise.

Chapter 4

Radar Signal Processing and CFAR

One of the many functions of radar is target detection; target detection is accomplished through a process known as radar signal processing [27]. A radar can measure three target components based on a return signal; the distance the target is from the radar, the Doppler shift due to the target's motion, and the target's direction with respect to the antenna array. There are various signal processing techniques used to measure each of these components. The three methods used during the research are, matched filtering, the Fast Fourier Transform across the pulse dimension to acquire the Doppler shift, and monopulse processing to identify the azimuth and elevation angles of the target. After discussing the signal processing techniques, the section concludes with a discussion of the constant false alarm rate (CFAR) algorithm used to filter out the false alarms in the simulation.

4.1 Match Filtering

One of the most common techniques in analyzing a complex signal is filtering, which alters the spectrum with predetermined parameters [28]. Matched filtering is used in many signal processing sub-fields to detect a signal corrupted by noise. The matched filter is designed to maximize the signal-to-noise ratio ($\mathbf{SNR}$) of a received signal corrupted by noise to extract a target's information from the radar. In most radar signal processing algorithms, a target's range is considered the most significant measurement [29].

The traditional matched filter takes an impulse response $h(t)$ and a transfer function $H(\omega)$. The conjugated time-reversed version of the transmitted signal determines the filter response,

$$h(t) = x_p^*(-t). \tag{4.1}$$

The transfer function $H(\omega)$ takes advantage of the time-reversal and conjugation properties of the Fourier transform where

$$\mathcal{F}\{f(-t)\} = F(-\omega) \tag{4.2}$$

and,

$$\mathcal{F}\{f^*(t)\} = F^*(-\omega). \tag{4.3}$$

Therefore, the transfer function of the matched filter is given as

$$\mathcal{F}\{h(t)\} = \mathcal{F}\{x_p^*(-t)\} = X_p^*(\omega) = H(\omega). \tag{4.4}$$

We also utilize the multiplication property of the Fourier transform,

$$\mathcal{F}\{x(t) \otimes h(t)\} = X(\omega)H(\omega). \tag{4.5}$$

Utilizing the multiplication property of the Fourier transform allows us to avoid the time sync that comes with convolution. Where convolution in the time domain translates to multiplication in the frequency domain.

With AGWN, figure 4.1 shows a target's matched filter response at a distance of approximately 12 km. Figure 4.2 shows the same target at the same distance without the corruption of AWGN.

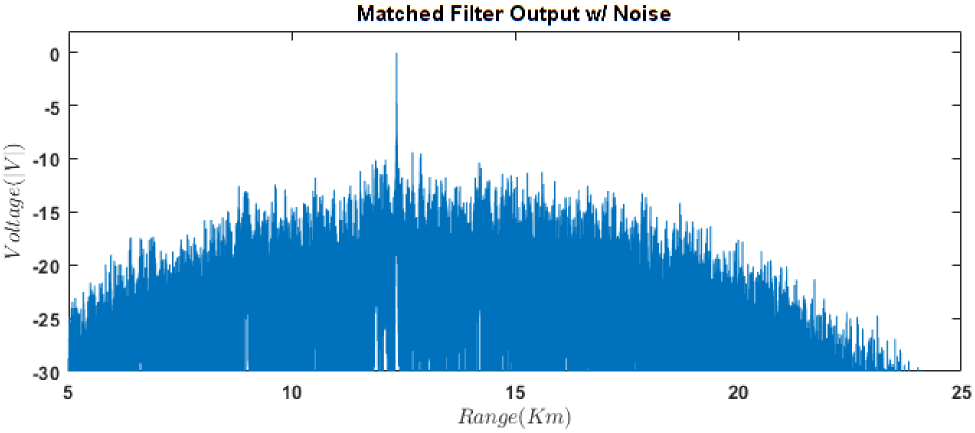


Figure 4.1: Output of Matched Filter With Noise

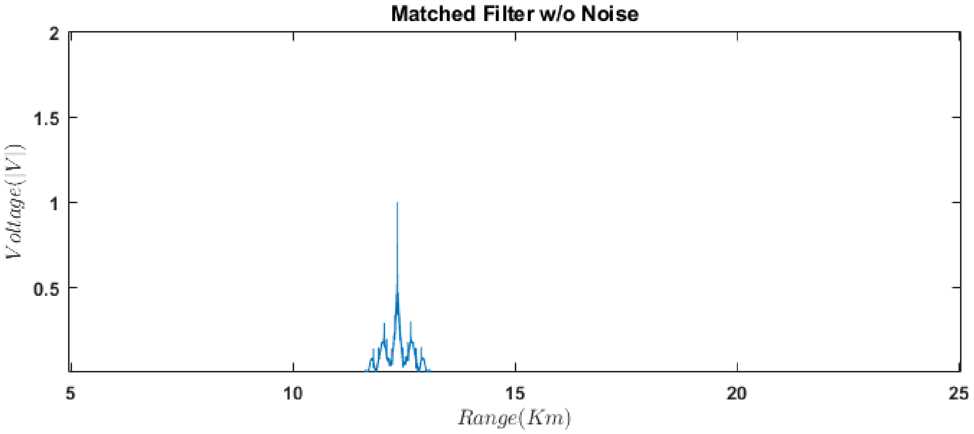


Figure 4.2: Output of Matched Filter Without Noise

4.2 Doppler Processing

There are several RSP techniques used for Doppler detection in radar systems. The received data are collected and organized into a data cube of ADC samples for each pulse along each antenna element.

A Fast Fourier Transform (FFT) is applied to the pulse dimension, converting it to the frequency shift due to the target's radial velocity (V_r). Figure 4.3 depicts the application of an FFT to a radar data cube.

Match filtering followed by Doppler processing result in a range-Doppler (RD) map.

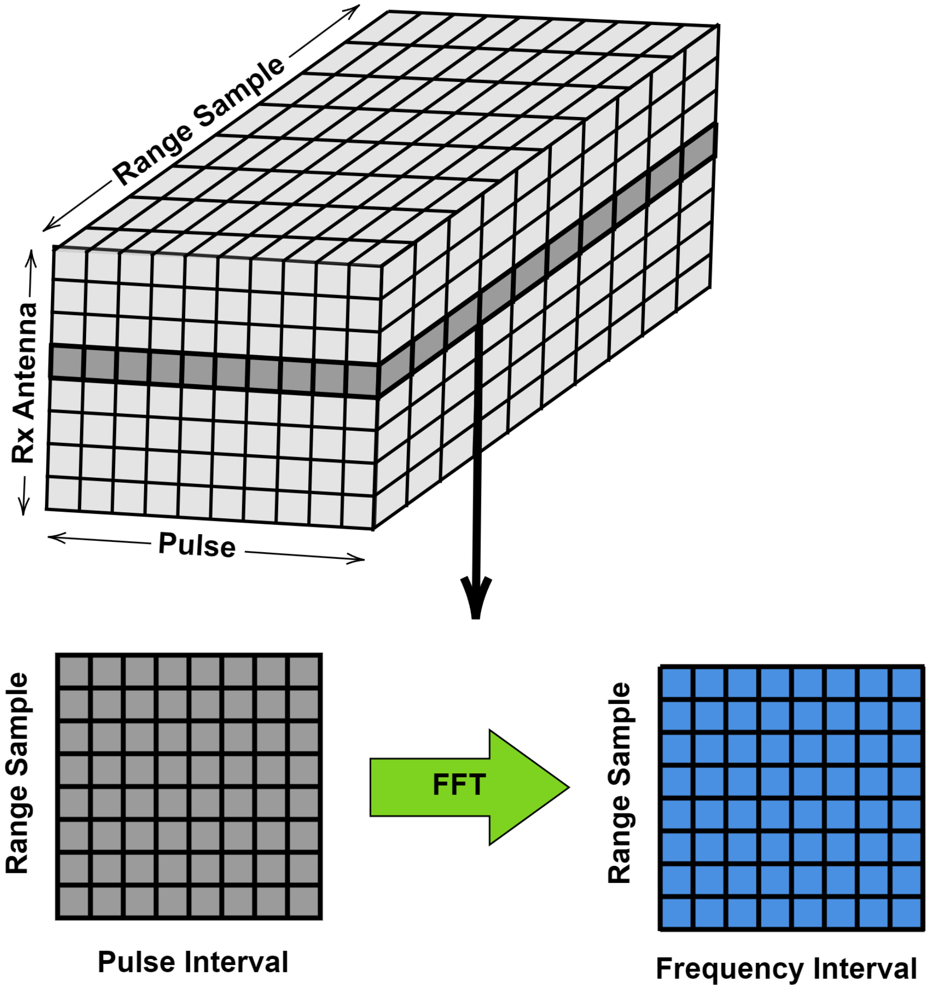


Figure 4.3: Received Data Cube

Doppler frequency (f_d) can be converted to V_r using 2.8,

$$f_d = \frac{2V_r}{\lambda}$$

In figure 4.4, AWGN distorts the received range-Doppler data. In contrast, without AWGN, the sidelobes from the range-Doppler processing are more defined in figure 4.5.

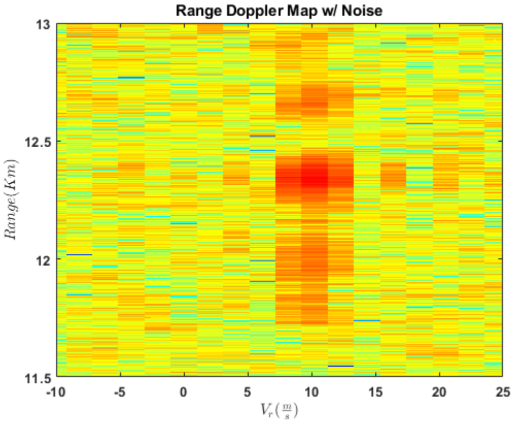


Figure 4.4: RD Map with AWGN

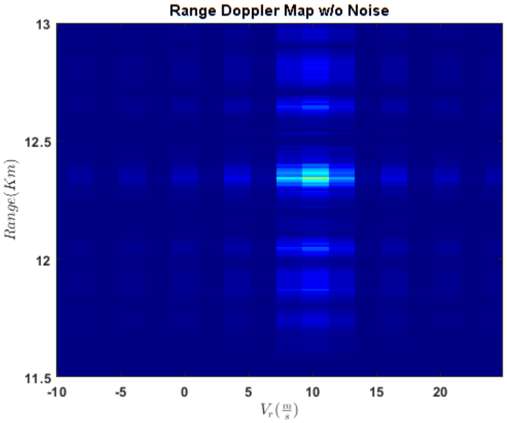


Figure 4.5: RD Map without AWGN

4.3 Monopulse AoA Estimation

Monopulse is a signal processing technique for measuring the angle of arrival (AoA) of the received signal [26]. In theory, monopulse processing estimates a target's direction angles for each pulse of a pulse train [30]. Chapter 3, Section 3.4 details the Sum ($\sum$)and Difference (Δ) beamforming used in Monopulse processing. This section will review the process of angle estimation through monopulse processing detailed by Barton [26].

The first step in monopulse processing is to create a monopulse table composed of monopulse bins that span at least the 3dB beamwidth (Bw_{3dB}) of the $\sum$ beam in both azimuth and elevation. To do so,

$$Bw_{3dB} = \frac{0.886\lambda}{L} \tag{4.6}$$

where λ is the wavelength of the carrier frequency of the transmitted signal and L is the array's length in the elevation or azimuth direction. For the array used in this simulation, the receive channels in the elevation and azimuth directions are the same. Therefore, the beam widths for both azimuth and elevation of the sum beam are the same, where the Monopulse angles span from $-\frac{Bw_{3dB}}{2}$ to $+\frac{Bw_{3dB}}{2}$, the width of the sum beam.

The next step is populating the monopulse table with the azimuth and elevation $\frac{\Delta}{\sum}$ ratios for each angle. Each table holds the angles that span the 3dB beamwidth of the $\sum$,

$$u_{az} = \begin{bmatrix} sin(\theta_{1,az}) & 0 \\ \vdots & \vdots \\ sin(\theta_{N_{ang},az}) & 0 \end{bmatrix} \tag{4.7}$$

and

$$u_{el} = \begin{bmatrix} 0 & sin(\theta_{1,el}) \\ \vdots & \vdots \\ 0 & sin(\theta_{N_{ang},el}) \end{bmatrix}. \tag{4.8}$$

Both are then converted to the K-space by scaling each matrix by the wave number of the carrier frequency,

$$k_{mono} = -ku. \tag{4.9}$$

Next, the complex phases pointing out from the array's center is created from the azimuth and elevation,

$$v_{mono} = \exp(-jAk_{mono}) \tag{4.10}$$

where A is the antenna phase center of the receive array. Summing v_{mono} for each of the azimuth and elevation complex phases creating Σ_{az} and Σ_{el},

$$\Sigma_{az} = \sum_{i=1}^{N_{ang}} v_{mono,az}, \tag{4.11}$$

$$\Sigma_{el} = \sum_{i=1}^{N_{ang}} v_{mono,el}. \tag{4.12}$$

The Δ_{az} and Δ_{el} beams are the difference in the half sums of v_{mono},

$$\Delta_{az} = \sum_{i=1}^{\frac{N_{ang}}{2}} v_{mono,az} - \sum_{i=\frac{N_{ang}}{2}+1}^{N_{ang}} v_{mono,az} \tag{4.13}$$

and,

$$\Delta_{el} = \sum_{i=1}^{\frac{N_{ang}}{2}} v_{mono,el} - \sum_{i=\frac{N_{ang}}{2}+1}^{N_{ang}} v_{mono,el}. \tag{4.14}$$

With the Σ and Δ channels built, the imaginary components for the ratios can be extracted and stored for each respective angle,

$$IDSR = Im\left(\frac{\Delta}{\Sigma}\right) \tag{4.15}$$

where $IDSR$ denotes the imaginary part of the Δ and Σ ratio. With ratio of the imaginary parts, we are able to identify the angle error from the center of the receive beam.

Once the range Doppler processing have been applied to the receive signal, the monopulse processing can be applied to any potential targets.

$$IDSR_{az} = Im\left(\frac{\Delta_{det,az}}{\sum_{det,az}}\right) \tag{4.16}$$

$$IDSR_{el} = Im\left(\frac{\Delta_{det,el}}{\sum_{det,el}}\right) \tag{4.17}$$

The $IDSR_{az}$ and $IDSR_{el}$ are compared to their respective ratios stored in the monopulse table, a null indicating the target's direction angles θ_{az} and θ_{el}. Figures 4.6 and 4.7 picture the nulls. With the receive sum beam's center being zero, the null represents the difference in a target's direction angle with respect to the center of the received sum beam. Figure 4.6 indicates that the target is not at the direct center of the beam in azimuth. Figure 4.7 shows a receive sum beam pointed at the target in elevation.

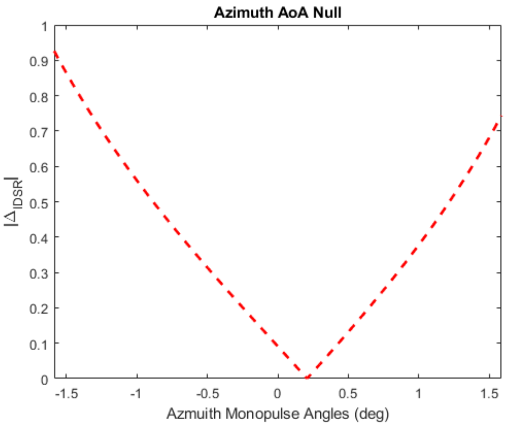


Figure 4.6: Azimuth AoA

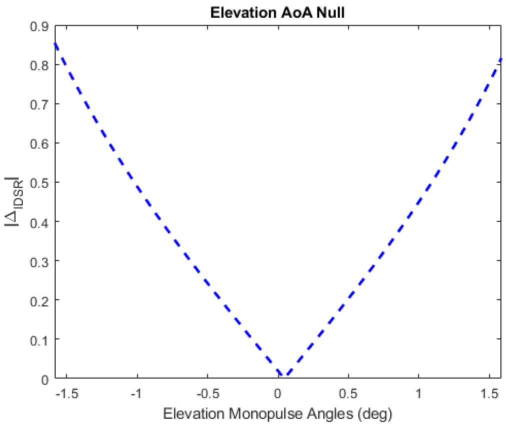


Figure 4.7: Elevation AoA

To find the estimated angle of arrival,

$$\theta_{aoa,az} = \theta_{ant,az} + \theta_{mono,az} \tag{4.18}$$

and,

$$\theta_{aoa,el} = \theta_{ant,el} + \theta_{mono,el}. \tag{4.19}$$

4.4 CFAR

While there are several CFAR algorithms, the one picked for this simulation is Cell Averaging Constant False Alarm Rate (CA-CFAR). The rationale for using the CA-CFAR is that it is not as computationally intensive as others. For example, Order Statistic Constant False Alarm Rate (OS-CFAR) can performed better than a CA-CFAR detector; however, it is more computationally intensive [31]. As for the CA-CFAR used, it sufficiently identifies the targets without sacrificing the computational processing needed for other CFAR algorithms. The most common approach to Constant False Alarm Rate (CFAR) is to assume the average noise power within a CFAR window does not contain a target and base the detection threshold on this assumption [32].

The estimated threshold $\hat{T}$,

$$\hat{T} = \sqrt{\frac{-4}{\pi}\mu_l \ln|P_{FA}|} \tag{4.20}$$

where $\ln|P_{FA}|$ is a function of the desired false alarm probability. When the test cell value is greater than $\hat{T}$, the test cell is potentially a target.

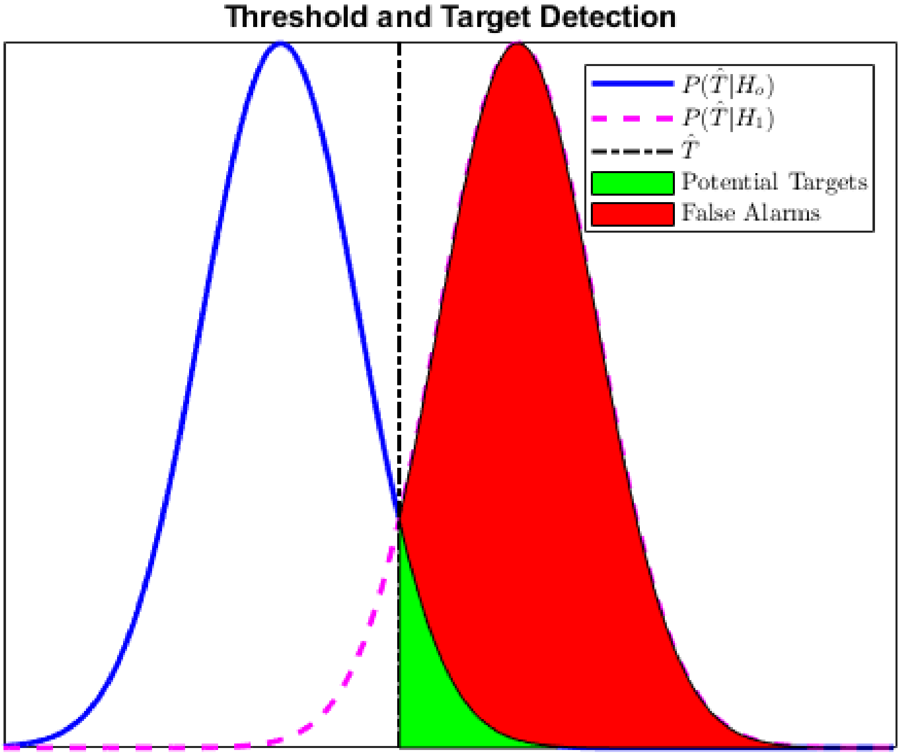


Figure 4.8: CFAR Threshold, Detection's and False Alarms

Figure 4.8 demonstrates the application of $\hat{T}$ and how it distinguishes between possible targets and false alarms using hypothesis testing with equal variances. The area under $P(\hat{T}|H_1)$ shaded in red is the probability of detection. The areas of $P(\hat{T}|H_1)$ and $P(\hat{T}|H_0)$ shaded in green is probability of false alarms (P_{FA}). The detector is designed to adjust the threshold $\hat{T}$ for an acceptable constant false alarm rate.

The CA-CFAR algorithm slides a window over the processed range-Doppler data with a predetermined window size. The CA-CFAR window has three basic types of range-Doppler cells. The first is the test cell, which is at the very center of the window. The test cell is being tested against the 2nd type of cells, the reference cells. Finally, the third type is the guard cells, these work as a buffer between the test and reference cells. The composition of the CA-CFAR window is shown in figure 4.9, labeling the three types of cells within the window. The more reference cells within a CFAR window, the more accurate the background noise estimation is [29]. However, there is a threshold to the number of reference cells used within a CFAR window before the noise is no longer homogeneous [29].

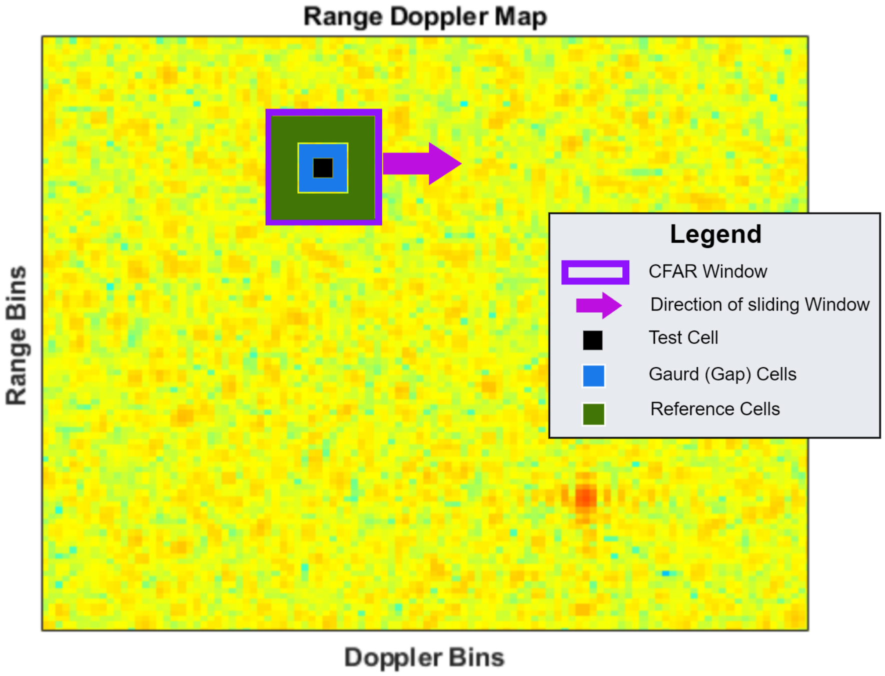


Figure 4.9: Range Doppler map with Sliding CA-CFAR Window

4.5 Summary

This chapter has covered the basics of the processing and target detection algorithms used within the simulation, from the basics of the matched filter used to identify a target's range, the extraction of the Doppler frequency to determine a target's radial velocity to the monopulse processing to determine a target's AoA. Furthermore, implementing the CA-CFAR algorithm that yields false alarms.

The next chapter will detail the theory behind the Kalman filter and the two trackers used in this simulation.

Chapter 5

Target Tracking

The ultimate goal of a target tracking algorithm is to estimate with high precision a target's attributes, such as position and velocity, continuously while the attributes are constantly changing [18]. Engineers have developed various algorithms to predict a target's next position based on previously observed information. This chapter will cover the basics of the Kalman filter, the Extended Kalman filter (EKF) implementation, and the theory and application of the Unscented Kalman filter (UKF).

5.1 The Kalman Filter and Background

The Kalman filter operates within linear stochastic systems; it allows for the mean and covariance of a random parameter to be estimated in the presence of unwanted AWGN [33]. Global Positioning Systems (GPS) utilize the Kalman filter for high-precision positioning.

There are three basic steps to a Kalman filter; the prediction step, the observation step, and the update step [34]. The process within the Kalman filter detailed throughout this section is summarized from Subhash [34].

The prediction, or priori will be the first step to be discussed. When tracking a target, an observer can predict a target's next position with previous information given. The prediction step calculates the state vector and estimated error. The state vector contains attributes associated with the target, such as position, velocity, and acceleration. Mathematically the state vector prediction is described as

$$\mathbf{\hat{x}_{n+1,n}} = \mathbf{F}\mathbf{\hat{x}_{n,n}} + \mathbf{G}\mathbf{u_n} \tag{5.1}$$

where $\mathbf{\hat{x}_{n+1,n}}$ is the predicted state vector of the next iteration, $\mathbf{F}$ is the state transition matrix $\mathbf{\hat{x}_{n,n}}$ is the previous state vector, $\mathbf{G}$ is the control matrix, and $\mathbf{u_n}$ is the input variable. $\mathbf{u_n}$ accounts for any added variations to the state vector. For example if the state vector is predicting position and velocity, $\mathbf{u_n}$ will account for any changes in the velocity with respect to an acceleration quantity. $\mathbf{G}$ will account for the time Δt since the last prediction.

$\mathbf{\hat{x}_{n+1,n}}$ is a $N_x \times 1$ vector holding the values of the the attributes of interest. for example if $\mathbf{\hat{x}_{n+1,n}}$ is a 6×1 predicting an objects position and velocity the result of 5.1 would be,

$$\mathbf{\hat{x}_{n+1,n}} = \begin{bmatrix} r_x \\ r_y \\ r_z \\ v_x \\ v_y \\ v_z \end{bmatrix}. \tag{5.2}$$

Where, $\mathbf{r}$ and $\mathbf{v}$ would be the vectors holding the position and velocity components of the object. The state transition matrix $\mathbf{F}$ can best be described as an $N_x \times N_x$ matrix where N_x is the number of variables within the state vector and accounts for the time Δt since the last prediction. For example, if the state vector predicted a target's Cartesian position and velocity, $\mathbf{F}$ would be a matrix holding the following values,

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.3}$$

The values of $\mathbf{F}$ account for the position due to the change in time for $\mathbf{\hat{x}_{n,n}}$.

Recall that the $\mathbf{\hat{x}_{n,n}}$ is the updated previous state vector. For this example, we will assume that $\mathbf{\hat{x}_{n,n}}$ predicts six variables, the position and velocity of a target in a Cartesian coordinate system according to,

$$\mathbf{\hat{x}_{n,n}} = \begin{bmatrix} \hat{x}_{n,n} \\ \hat{y}_{n,n} \\ \hat{z}_{n,n} \\ \hat{v_{x}}_{n,n} \\ \hat{v_{y}}_{n,n} \\ \hat{v_{z}}_{n,n} \end{bmatrix}. \tag{5.4}$$

The control matrix $\mathbf{G}$ maps the control input to the state variable. For example, if there was an increase or decrease in a target's velocity, $\mathbf{G}$ accounts for that acceleration.

Finally, the control input $\mathbf{u_n}$ is the last variable to account for in the predicted state vector equation, which is any change in velocity to a target, accounting for a target's change in position.

The estimated error for the prediction is represented in a covariance matrix,

$$\mathbf{P_{n+1,n}} = \mathbf{F}\mathbf{P_{n,n}}\mathbf{F^T} + \mathbf{Q} \tag{5.5}$$

where $\mathbf{P_{n+1,n}}$ is a matrix holding the predicted error for each variable in the predicted state vector $\mathbf{\hat{x}_{n+1,n}}$. $\mathbf{F^T}$ is a transposed state transition matrix, $\mathbf{P_{n,n}}$ is the previously estimated error, and $\mathbf{Q}$ is the process noise covariance matrix $\mathbf{P_{n,n}}$, and $\mathbf{P_{n+1,n}}$ are both $n_x \times n_x$ matrices holding the error variances for the state vector variables. The covariance is updated during each prediction. The updated prediction depends on the accuracy of the previous prediction.

Because the actual noise process of a system can not be precisely determined, $\mathbf{Q}$ is estimated and considered constant for the entire process of the Kalman Filter.

The observation step is relatively simple in the traditional Kalman Filter due to the assumption that $\mathbf{\hat{x}_{n,n}}$ is measured through a Linear Time-Invariant (LTI) system. The observation step takes in a vector $\mathbf{z_{n,n}}$ which is the measurement.

The Kalman filter is considered an optimal filter; therefore, during the update stage, it will seek to minimize the error variance using the Kalman gain $\mathbf{K}$. $\mathbf{K}$ will minimize the variance of the covariance $\mathbf{P_{n,n}}$ along the diagonal, which characterize the error variance in estimating these variables within the state vector $\mathbf{\hat{x}_{n,n}}$. $\mathbf{K}$ is described as

$$\mathbf{K} = \mathbf{P_{n,n-1}}\mathbf{H^T}(\mathbf{H}\mathbf{P_{n,n-1}}\mathbf{H^T} + \mathbf{R_n})^{-1} \tag{5.6}$$

where $\mathbf{P_{n,n-1}}$ is being the previous covariance, $\mathbf{H}$ is a linear mapping from the state variables to the observations, and $\mathbf{R_n}$ accounting for the measurement uncertainty. $\mathbf{R_n}$ is a $N_x \times N_x$ matrix that holds the error variances of each measurement along the diagonal. When computed, the Kalman gain is a matrix with $n_z \times n_x$ dimension.

For the state update equation ($\mathbf{\hat{x}_{n,n}}$), the main factors that it is controlled by are the Kalman gain and the error between the observation and predicted state vector. With the update to state vector being,

$$\mathbf{\hat{x}_{n,n}} = \mathbf{\hat{x}_{n,n-1}} + \mathbf{K}(\mathbf{z_n} - \mathbf{H}\mathbf{\hat{x}_{n,n-1}})). \tag{5.7}$$

Upon closer inspection, if the difference between the measurement and the predicted measurement is zero, then the updated state vector equals predicted state vector. Alternatively, if the difference between the measurement and the predicted measurement is small, the weighted $\mathbf{K}$ affects the updated state vector very little. A large Kalman gain shows low confidence in the predicted state vector and places more weight on the new measurement.

For the updated covariance, $\mathbf{K}$ again plays a role. Along with $\mathbf{R_n}$, the updated covariance is,

$$\mathbf{P_{n,n}} = (\mathbf{I} - \mathbf{K_n H})\mathbf{P_{n,n-1}}(\mathbf{I} - \mathbf{K_n H})^{\mathbf{T}} + \mathbf{K_n R_n K_n^T}. \tag{5.8}$$

If the Kalman gain is very small the covariance becomes $\mathbf{P_{n,n}} \simeq \mathbf{P_{n,n-1}}$. However, as the values of $\mathbf{K}$ get closer to one, the first term in (5.7) approaches zero and the previous covariance is ignored, such that the measurement uncertainty is weighted more heavily when previous covariance is negated, and measurement uncertainty is weighted much greater when calculating the updated covariance.

To see how the Kalman filter uses the previous operation and how the steps work off each other, reference figure 5.1. The steps continue until the observations cease to exist.

5.2 Extended Kalman Filter Theory

While the traditional Kalman filter is practical when predicting linear systems, it does not work well with non-linear such as the measurements of range, velocity, and angle. For a radar system that predicts, variations of the Kalman filters have been developed over the years. One variation of the Kalman filter is the Extended Kalman filter (EKF). The EKF approximates the local linear behavior of a non-linear function so that a Kalman filter can be applied [35].

The essential operation of a Kalman filter is to propagate a Gaussian random variable

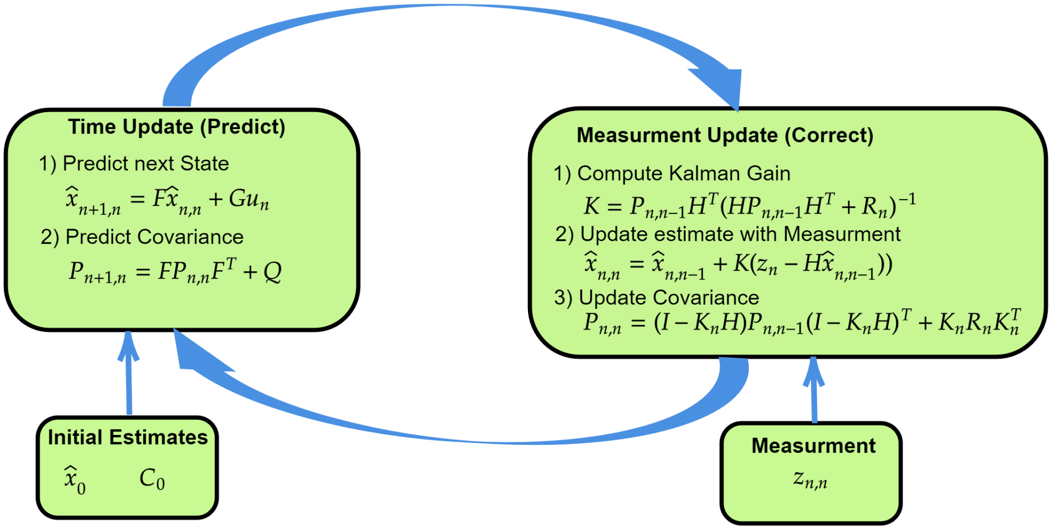

Time Update (Predict)
1) Predict next State
$\hat{x}_{n+1,n} = F\hat{x}_{n,n} + Gu_n$
2) Predict Covariance
$P_{n+1,n} = FP_{n,n}F^T + Q$
Measurment Update (Correct)
1) Compute Kalman Gain
$K = P_{n,n-1}H^T(HP_{n,n-1}H^T + R_n)^{-1}$
2) Update estimate with Measurment
$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K(z_n - H\hat{x}_{n,n-1}))$
3) Update Covariance
$P_{n,n} = (I - K_nH)P_{n,n-1}(I - K_nH)^T + K_nR_nK_n^T$
Initial Estimates
$\hat{x}_0$
C_0
Measurment
$z_{n,n}$

Figure 5.1: Flowchart of a Basic Kalman Filter

(GRV) through a system. In the EKF, the state vectors distribution is approximated as a Gaussian distribution with a GRV, which is then propagated analytically through the first-order linearization using a Jacobian of the nonlinear system to approximate a linear system [1]. Section 5.3 the details of the EKF and how to approximate the linearization. Essentially, the EKF approximates the state vector through the a linear approximation and has the ability to propagate the covariance through the system exactly.

5.3 Extended Kalman Filter Algorithm and Application

The EKF operations are similar to the classic Kalman filter, estimating the next state and error of the state due to noise, taking a measurement, and correcting the estimated state due to the error between the initial estimate and measurement. As stated in 5.2, the critical difference is that the EKF can handle nonlinear systems through a first-order linearization of the nonlinear system. This section will break down the time update and measurement update given by Kay and how those apply to the simulation [35].

After predicting an initial state and error, the state vector is updated to account for the time between measurements. Because this radar system predicts position and velocity, the predicted state is a 6×1 vector. Assuming the dynamic system model is known based on the previous velocity and position, a state transition function is applied to the previous state vector,

$$\mathbf{\bar{x}_k} = f(\mathbf{\bar{x}_{k-1}}). \tag{5.9}$$

The linearized approximated estimated error,

$$\mathbf{\bar{C}_k} = \mathbf{F}\mathbf{\bar{C}_{k-1}}\mathbf{F^T} + \mathbf{Q_{k-1}}, \tag{5.10}$$

$\mathbf{\bar{C}_{k-1}}$ is the previously updated covariance, and $\mathbf{Q_{k-1}}$ is the estimated system noise. $\mathbf{A_k}$ is a matrix of partial derivatives (the Jacobian). Being a first-order Taylor series expansion, the Jacobian is evaluated at the current predicted state to linearize the covariance at the current predicted state. The mathematical description of $\mathbf{A_k}$ is,

$$\mathbf{A_k} = \begin{bmatrix} \frac{\partial f_{1,k}}{\partial x_{1,k}} & \frac{\partial f_{1,k}}{\partial x_{2,k}} & \frac{\partial f_{1,k}}{\partial x_{3,k}} & \frac{\partial f_{1,k}}{\partial x_{4,k}} & \frac{\partial f_{1,k}}{\partial x_{5,k}} & \frac{\partial f_{1,k}}{\partial x_{6,k}} \\ \frac{\partial f_{2,k}}{\partial x_{2,k}} & \frac{\partial f_{2,k}}{\partial x_{2,k}} & \frac{\partial f_{2,k}}{\partial x_{3,k}} & \frac{\partial f_{2,k}}{\partial x_{4,k}} & \frac{\partial f_{2,k}}{\partial x_{5,k}} & \frac{\partial f_{2,k}}{\partial x_{6,k}} \\ \frac{\partial f_{3,k}}{\partial x_{2,k}} & \frac{\partial f_{3,k}}{\partial x_{2,k}} & \frac{\partial f_{3,k}}{\partial x_{3,k}} & \frac{\partial f_{3,k}}{\partial x_{4,k}} & \frac{\partial f_{3,k}}{\partial x_{5,k}} & \frac{\partial f_{3,k}}{\partial x_{6,k}} \\ \frac{\partial f_{4,k}}{\partial x_{2,k}} & \frac{\partial f_{4,k}}{\partial x_{2,k}} & \frac{\partial f_{4,k}}{\partial x_{3,k}} & \frac{\partial f_{4,k}}{\partial x_{4,k}} & \frac{\partial f_{4,k}}{\partial x_{5,k}} & \frac{\partial f_{4,k}}{\partial x_{6,k}} \\ \frac{\partial f_{5,k}}{\partial x_{2,k}} & \frac{\partial f_{5,k}}{\partial x_{2,k}} & \frac{\partial f_{5,k}}{\partial x_{3,k}} & \frac{\partial f_{5,k}}{\partial x_{4,k}} & \frac{\partial f_{5,k}}{\partial x_{5,k}} & \frac{\partial f_{5,k}}{\partial x_{6,k}} \\ \frac{\partial f_{6,k}}{\partial x_{2,k}} & \frac{\partial f_{6,k}}{\partial x_{2,k}} & \frac{\partial f_{6,k}}{\partial x_{3,k}} & \frac{\partial f_{6,k}}{\partial x_{4,k}} & \frac{\partial f_{6,k}}{\partial x_{5,k}} & \frac{\partial f_{6,k}}{\partial x_{6,k}} \end{bmatrix}. \tag{5.11}$$

The EKF linearization in figure 5.2 shows the actual sampling along with an estimated covariance and mean on the left [1]. The samples are transformed by $f(\mathbf{x})$, and a true covariance and mean are given. However, this is not possible in a nonlinear system. To the right, the EKF transforms the mean and approximates it through $h(\mathbf{x})$, and $\mathbf{A_k}\mathbf{\bar{C}_{k-1}}\mathbf{A_k^T}$ linearizing the covariance.

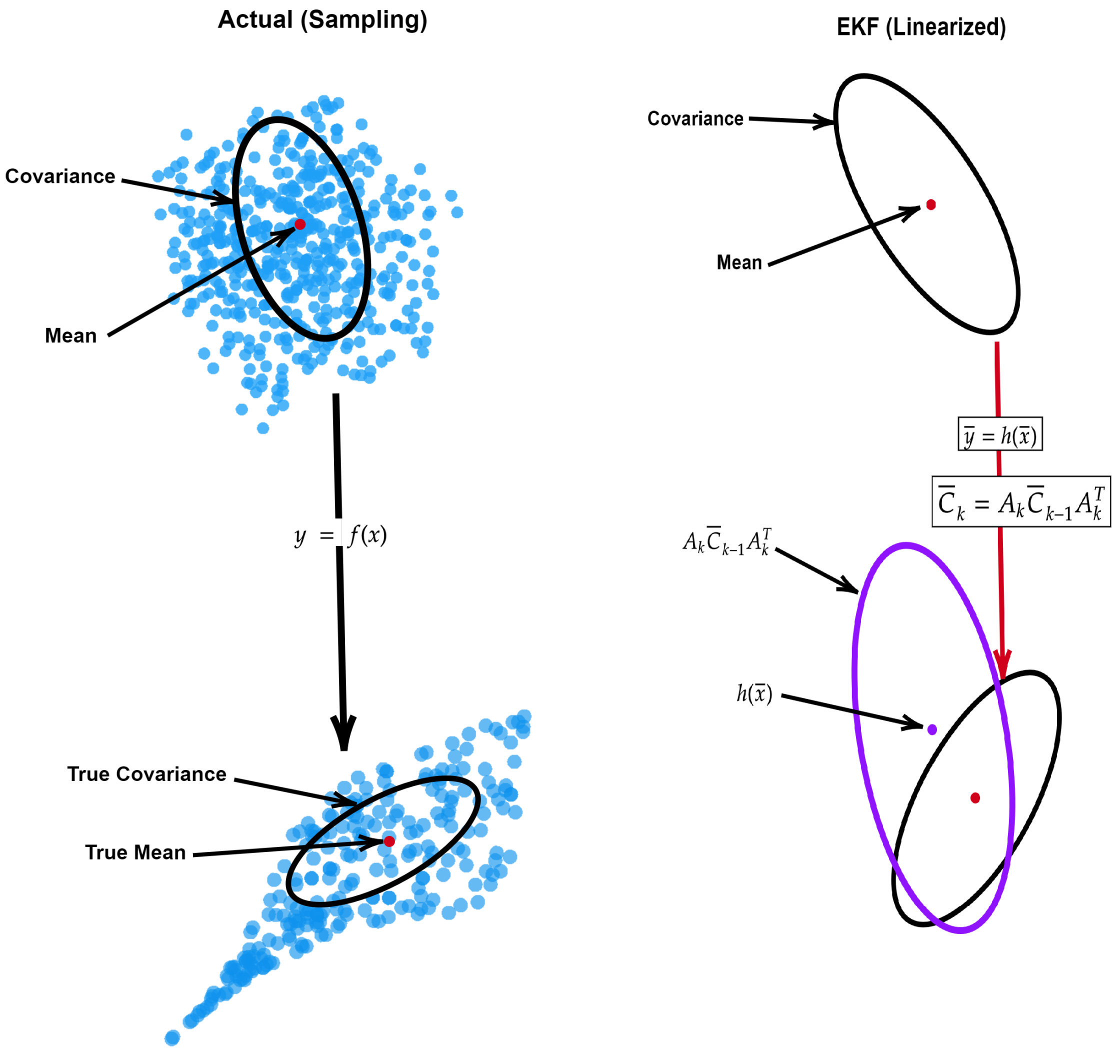


Figure 5.2: Comparison of Sampling vs EKF Linearization [1].

Measurements are needed to update (correct) the estimated state vector and covariance. A transfer function $h(\mathbf{\bar{x}_k})$ is applied to the state vector and converted to the measurement domain. Giving the following vector,

$$h(\bar{\mathbf{x}}_{\mathbf{k}}) = \begin{bmatrix} \hat{x} \\ \hat{y} \\ r_k \\ V_{r,k} \\ V_{x,k} \\ V_{y,k} \\ V_{z,k} \end{bmatrix}$$

where $\hat{x}$, $\hat{y}$, and r_k are parameters directly related to the range and angles measured by the radar. V_r is the radial velocity measured by the radar. $V_{x,k}$, $V_{y,k}$, and $V_{z,k}$ is the velocity vector components that are related to the range, angle, and radial velocity of the radar. The values of The unit vectors, range, and radial velocity are converted to the measurement domain using the following series of equations;

$$r_k = \sqrt{r_{x,k}^2 + r_{y,k}^2 + r_{z,k}^2}, \tag{5.12}$$

$$\hat{x} = \frac{r_{x,k}}{r_k}, \tag{5.13}$$

$$\hat{y} = \frac{r_{y,k}}{r_k}, \tag{5.14}$$

$$V_{r,k} = \hat{r_k} \bullet V_k, \tag{5.15}$$

where

$$\hat{\mathbf{r}}_{\mathbf{k}} = \begin{bmatrix} r_{x,k} & r_{y,k} & r_{z,k} \end{bmatrix}. \tag{5.16}$$

These particular combinations of the range, velocity, and angle measurements are used to more effectively compute the Jacobian. The Jacobian ($\mathbf{A_k}$) relates $\bar{\mathbf{x}}_{\mathbf{k}}$ to $h(\bar{\mathbf{x}}_{\mathbf{k}})$. $\mathbf{A_k}$ is computed as the first-order Taylor series expansion,

$$\mathbf{A} = \left.\frac{\partial h_k}{\partial x_k}\right|_{\mathbf{x}=\bar{\mathbf{x}}} = \begin{bmatrix} \frac{1}{r_k} - \frac{\hat{x}^2}{r_k^2} & -\frac{\hat{x}\hat{y}}{r_k^3} & -\frac{\hat{x}\hat{z}}{r_k^3} & 0 & 0 & 0 \\ -\frac{\hat{x}\hat{y}}{r_k^3} & \frac{1}{r_k} - \frac{\hat{y}^2}{r_k^3} & -\frac{\hat{y}\hat{z}}{r_k^3} & 0 & 0 & 0 \\ \frac{\hat{x}}{r_k} & \frac{\hat{y}}{r_k} & \frac{\hat{z}}{r_k} & 0 & 0 & 0 \\ \frac{V_{x,k}}{r_k} - \frac{\hat{x}V_{x,k}}{r_k^2} & \frac{V_{y,k}}{r_k} - \frac{\hat{y}V_{y,k}}{r_k^2} & \frac{V_{z,k}}{r_k} - \frac{\hat{z}V_{z,k}}{r_k^2} & \frac{\hat{x}}{r_k} & \frac{\hat{y}}{r_k} & \frac{\hat{z}}{r_k} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.17}$$

The variance of the error can be represented as a diagonal matrix $\mathbf{R_k}$ with each index along the diagonal representing the estimated variance for each element within the state vector being predicted in the measurement space since the $\mathbf{SNR}$ affects the error in the r, V_r, and unit vector measurements.

$$\mathbf{SNR} = \frac{|Rx_{sig}|}{P_{ns}} \tag{5.18}$$

The range resolution is predetermined, being dependent on the bandwidth β,

$$\delta r = c/2\beta \tag{5.19}$$

The resolutions of $\hat{x}$ and $\hat{y}$ depend on the number of antenna elements transmitting in the respective direction N_{tx} and N_{tx},

$$\delta\hat{x} = 2/N_{tx} \tag{5.20}$$

$$\delta\hat{y} = 2/N_{ty} \tag{5.21}$$

the radial velocity resolution,

$$\delta V_r = \frac{\lambda}{2N_p f_{PR}} \tag{5.22}$$

which is related to Doppler frequency, depending on the transmit wavelength λ, the number of pulses per CPI N_p, and the pulse repetition frequency f_{PR}.

The variance of each measurement is calculated with the estimated $\mathbf{SNR}$ from the measurement resolutions of the range, unit vectors, and radial velocity;

$$\sigma_r^2 = \frac{\delta R}{2\mathbf{SNR}}, \tag{5.23}$$

$$\sigma_x^2 = \frac{\delta \hat{x}}{2\mathbf{SNR}}, \tag{5.24}$$

$$\sigma_y^2 = \frac{\delta \hat{y}}{2\mathbf{SNR}}, \tag{5.25}$$

$$\sigma_{Vr}^2 = \frac{\delta V_r}{2\mathbf{SNR}}. \tag{5.26}$$

The final variances of $\mathbf{R_k}$ are estimated to be the velocity unit vector variance from $\mathbf{\bar{C}_k}$. With this information, the measurement error matrix,

$$\mathbf{R_k} = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_r^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{Vr}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{C}_{k,55} & \bar{C}_{k,56} & \bar{C}_{k,57} \\ 0 & 0 & 0 & 0 & \bar{C}_{k,65} & \bar{C}_{k,66} & \bar{C}_{k,67} \\ 0 & 0 & 0 & 0 & \bar{C}_{k,75} & \bar{C}_{k,76} & \bar{C}_{k,77} \end{bmatrix}. \tag{5.27}$$

The Kalman Gain $\mathbf{K_k}$ is evaluated on each step using the updated $\mathbf{\bar{C}_k}$ and $\mathbf{R_k}$,

$$\mathbf{K_k} = \mathbf{\bar{C}_k H_k (H_k \bar{C}_k H_k^T + R_k)}^{-1}. \tag{5.28}$$

Notice how the system error stored in $\mathbf{\bar{C}_k}$ influences $\mathbf{K_k}$. The measurement error stored in $\mathbf{R_k}$ also influences the output of $\mathbf{K_k}$. $\mathbf{\bar{C}_k}$ and $\mathbf{R_k}$ determines how much weight to give the measurement $\mathbf{z_k}$ and how much weight to distribute between the measurement $\mathbf{z_k}$ and the state vector $\mathbf{\bar{x}_k}$.

The state vector $\mathbf{\hat{x}_k}$ is updated,

$$\mathbf{\hat{x}_k} = \mathbf{\bar{x}_k} + \mathbf{K_k}\sigma_\mathbf{z} \tag{5.29}$$

where,

$$\sigma_\mathbf{z} = \mathbf{z_k} - h(\mathbf{\bar{x}_k}). \tag{5.30}$$

Accounting for the error in the prediction and $\mathbf{K_k}$ in 5.29 weights the $\sigma_\mathbf{z}$. Note that for equation 5.29,

$$\lim_{\mathbf{K_k}\to 1} \mathbf{\hat{x}_k} = \mathbf{\bar{x}_k} + \sigma_\mathbf{z} \tag{5.31}$$

and,

$$\lim_{\mathbf{K_k}\to 0} \mathbf{\hat{x}_k} = \mathbf{\bar{x}_k} \tag{5.32}$$

The updated covariance,

$$\mathbf{\hat{C}_k} = (\mathbf{I} - \mathbf{K_k}\mathbf{H_k})\mathbf{\bar{C}_k}, \tag{5.33}$$

theoretically, causing the predicted error to converge on the actual error as the iterations increase. Figure 5.3 summarizes the EKF algorithm. The prediction step takes in the initial estimates of the state vector, applies the transfer function to the previous state vector, and estimates the system noise. Then, it updates the Kalman gain based on the trust in the measurement and measurement noise and updates the state vector and covariance.

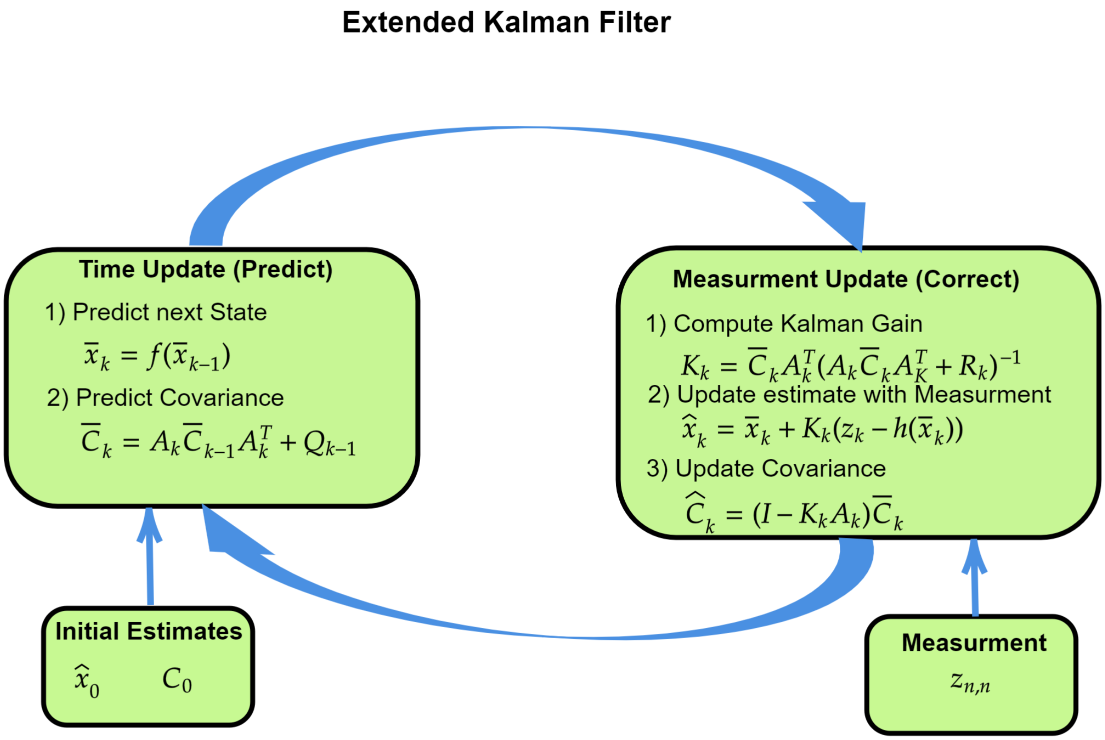

Extended Kalman Filter
Time Update (Predict)
1) Predict next State
$\overline{x}_k = f(\overline{x}_{k-1})$
2) Predict Covariance
$\overline{C}_k = A_k \overline{C}_{k-1} A_k^T + Q_{k-1}$
Measurment Update (Correct)
1) Compute Kalman Gain
$K_k = \overline{C}_k A_k^T (A_k \overline{C}_k A_K^T + R_k)^{-1}$
2) Update estimate with Measurment
$\widehat{x}_k = \overline{x}_k + K_k(z_k - h(\overline{x}_k))$
3) Update Covariance
$\widehat{C}_k = (I - K_k A_k)\overline{C}_k$
Initial Estimates
$\widehat{x}_0$ C_0
Measurment
$z_{n,n}$

Figure 5.3: Extended Kalman Filter Algorithm

While the EKF can keep the simple recursive computations of the classical Kalman filter, the linear approximations applied using the Jacobian to non-linear equations representing the system dynamics can produce inaccurate results [36]. The inaccurate results could be because the resulting Jacobian may be inaccurate or nonexistent [36], [37]. Because of the EKF limitations, the next section will introduce another filter from the Kalman filter family.

5.4 Unscented Kalman Filter Theory

With the EKF's limitations, the unscented Kalman filter (UKF) has grown in popularity recently. The UKF compensates for the deficiencies of the EKF by introducing the Unscented Transform (UT) [37]. With the UT, there is no need for a complex Jacobian calculation for the linearization process needed to transform the mean and covariance information [34].

5.5 Unscented Kalman Filter Algorithm and Application

In keeping the Kalman tradition, the UKF algorithm is similar to the EKF and classic Kalman filter. The central part of the algorithm predicts a state vector and covariance, takes a measurement, then updates the state vector and covariance. The UKF repeats these steps until the iterations are complete. The UKF algorithm used for the simulation follows the UKF algorithm documented in Challa [34].

Before the UKF algorithm begins, the weights are selected, $w_{i,m}$ denoting the mean weights and $w_{i,c}$ denoting the covariance weights:

$$w_{0,m} = \frac{\lambda}{N+\lambda}, \tag{5.34}$$

$$w_{i,m} = \frac{\lambda}{2(N+\lambda)} \qquad 1 \leq i \leq 2N, \tag{5.35}$$

$$w_{0,c} = \frac{\lambda}{N+\lambda} + 1 - \alpha^2 + \beta, \tag{5.36}$$

$$w_{i,c} = \frac{\lambda}{2(N+\lambda)} \qquad 1 \leq i \leq 2N. \tag{5.37}$$

Where,

$$\lambda = 3 - N, \tag{5.38}$$

N is the number of data points in the state vector, β is 2, the optimal value for a Gaussian distribution and α is set to a constant 1×10^{-2} [1]. The weights control the distribution of the sigma points of the UT. To remain unbiased, the weights for the mean must satisfy the condition,

$$\sum_i w_i = 1. \tag{5.39}$$

The first step in the UKF Algorithm is the calculating the sigma points ($\chi_{\mathbf{k,i}}$) for the UT,

$$\chi_{\mathbf{k,0}} = \hat{\mathbf{x}}_{\mathbf{k-1}}, \tag{5.40}$$

$$\chi_{\mathbf{k,i}} = \hat{\mathbf{x}}_{\mathbf{k-1}} + \left(\sqrt{(N+\lambda)\hat{\mathbf{C}}_{\mathbf{k-1}}}\right)_i \qquad 1 \leq i \leq N, \tag{5.41}$$

and

$$\chi_{\mathbf{k,i}} = \hat{\mathbf{x}}_{\mathbf{k-1}} - \left(\sqrt{(N+\lambda)\hat{\mathbf{C}}_{\mathbf{k-1}}}\right)_i \qquad 1 \leq i \leq N. \tag{5.42}$$

A function is then applied, transforming the sigma points,

$$\chi^{\mathbf{f}}_{\mathbf{k,i}} = f(\chi_{\mathbf{k,i}}) \tag{5.43}$$

accounting for the CPI time Δt and the estimated V for the simulation.

Once the sigma points have been transformed, $w_{i,m}$ is applied to each $\chi^{\mathbf{f}}_{\mathbf{k,i}}$ and summed together, giving the UT approximation of the mean,

$$\bar{\mathbf{x}}_{\mathbf{k}} = \sum_{i=0}^{2N} w_{i,m} \chi^{\mathbf{f}}_{\mathbf{k,i}}. \tag{5.44}$$

A weighted covariance is created by applying $w_{i,c}$ to each estimated error of the sigma points and mean calculated in (5.44),

$$\bar{\mathbf{C}}_{\mathbf{k}} = \sum_{i=0}^{2N} w_{i,c} \left[\chi^{\mathbf{f}}_{\mathbf{k,i}} - \bar{\mathbf{x}}_{\mathbf{k}} \right] \left[\chi^{\mathbf{f}}_{\mathbf{k,i}} - \bar{\mathbf{x}}_{\mathbf{k}} \right]^T + \mathbf{Q} \tag{5.45}$$

summing the weighted estimated covariance, giving a UT approximation.

The unscented transform represented in figure 5.4, demonstrates how the UT better estimates the true mean and approximated covariance when applied to the state vector and estimated covariance. Essentially, the UT allows the exact sigma points to pass through the system, while the sigma points are used to approximate the covariance.

After the predictions and measurements, the sigma points are converted from Cartesian coordinates to Spherical coordinates due to the radar measurements taking place in a Spherical coordinate system,

$$\zeta_{\mathbf{k,i}} = h(\chi^{\mathbf{f}}_{\mathbf{k,i}}) \tag{5.46}$$

where,

$$\zeta_{\mathbf{k,i}} = \begin{bmatrix} r_{k,i} \\ \theta_{az,k,i} \\ \theta_{el,k,i} \\ Vr_{k,i} \end{bmatrix}.$$

With the UKF tracking the position and velocity in a Cartesian coordinate system, these are straightforward conversions. The algorithm converts the predicted range from the predicted position vector, the angles of arrival from the angles of the position vector, and the radial velocity from the dot product of the velocity vector with the position's unit vector. The conversions are as follows,

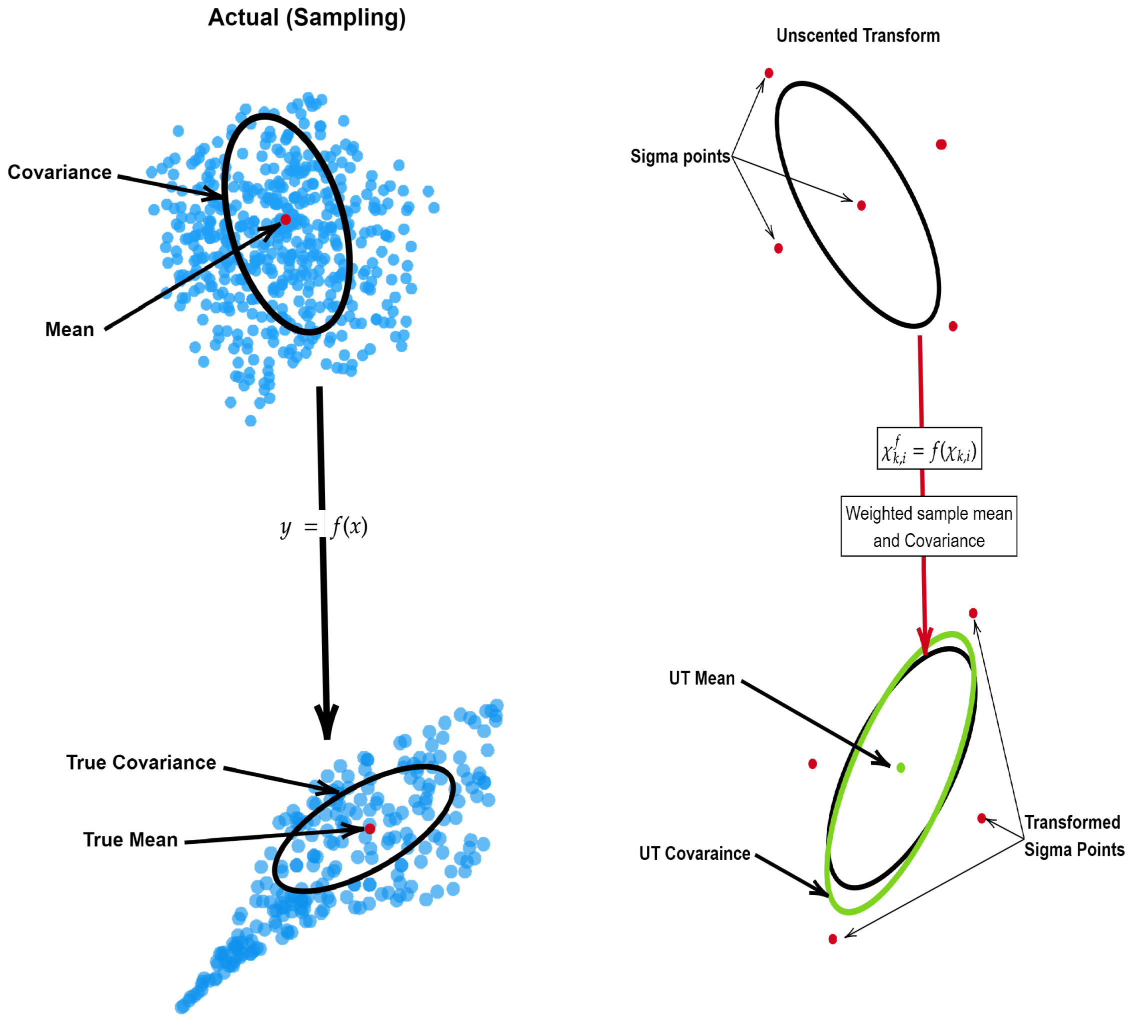


Figure 5.4: Comparison of Sampling vs Unscented Transform [1].

$$r_{k,i} = \sqrt{x_{k,i}^2 + y_{k,i}^2 + r_{k,i}^2}, \tag{5.47}$$

$$\theta_{az,k,i} = tan^{-1}(\frac{y_{k,i}}{x_{k,i}}), \tag{5.48}$$

$$\theta_{el,k,i} = tan^{-1}(\frac{sqrtx_{k,i}^2 + y_{k,i}^2}{z_{k,i}}), \tag{5.49}$$

and

$$Vr_{k,i} = V_{k,i} \bullet \hat{r}_{k,i}. \tag{5.50}$$

The mean weights are applied to the converted sigma points $\zeta_{\mathbf{k},\mathbf{i}}$ in the measurement domain.

Using a similar approach to calculating the $\mathbf{R_k}$ of the measurement space from Section 5.3. Recall (5.23) and (5.26), which are reused to calculate the estimated variance on each CPI for the range and radial velocity. As for the angles $\theta_{az,k,i}$ and $\theta_{el,k,i}$, an estimated variance is calculated before the simulation begins and re-calibrates using the $\mathbf{SNR}$ estimated on each CPI. Because the actual variance is unknown, each estimated variance is compared by taking the larger of the two.

The angle variance calculated before the simulation begin equal to the width of a single monopulse bin in an angles respective direction,

$$\sigma_{az,initial} = \frac{BW_{az}}{Nx_{bin} - 1} \tag{5.51}$$

and

$$\sigma_{el,initial} = \frac{BW_{el}}{Ny_{bin} - 1}. \tag{5.52}$$

The angles errors are recalculated every CPI with respect to the estimated $\mathbf{SNR}$,

$$\sigma_{az,SNR} = \frac{BW_{az}}{1.39\sqrt{2\mathbf{SNR}}} \tag{5.53}$$

and

$$\sigma_{el,SNR} = \frac{BW_{el}}{1.39\sqrt{2\mathbf{SNR}}}. \tag{5.54}$$

Because only four measurements were taken, $\mathbf{R_k}$ dimensions are 4×4,

$$\mathbf{R_k} = \begin{bmatrix} \sigma^2_{r,k,i} & 0 & 0 & 0 \\ 0 & \sigma^2_{az,k,i} & 0 & 0 \\ 0 & 0 & \sigma^2_{el,k,i} & 0 \\ 0 & 0 & 0 & \sigma^2_{Vr,k,i} \end{bmatrix}. \tag{5.55}$$

Next, the weighted measurement exception and covariance are calculated,

$$\mathbf{\hat{z}_k} = \sum_{i=0}^{2N} w_{i,m} \zeta_{\mathbf{k,i}} \tag{5.56}$$

and

$$\mathbf{S_k} = \sum_{i=0}^{2N} w_{i,c} \Big[\zeta_{\mathbf{k,i}} - \mathbf{\hat{z}_k}\Big] \Big[\zeta_{\mathbf{k,i}} - \mathbf{\hat{z}_k}\Big]^T + \mathbf{R_k}. \tag{5.57}$$

Applying the covariance weights to the expected and measured difference updates the measurement's cross-covariance,

$$\mathbf{\bar{C}_{k,xz}} = \sum_{i=0}^{2N} w_{i,c} \Big[\chi^{\mathbf{f}}_{\mathbf{k,i}} - \mathbf{\bar{x}_k}\Big] \Big[\zeta_{\mathbf{k,i}} - \mathbf{\hat{z}_k}\Big]^T. \tag{5.58}$$

The updated cross-covariance and measurement covariance update the Kalman gain,

$$\mathbf{K_k} = \mathbf{\bar{C}_{k,xz}} \mathbf{S_k}^{-1}. \tag{5.59}$$

$\mathbf{K_k}$ is then applied to the measured difference to update the state vector,

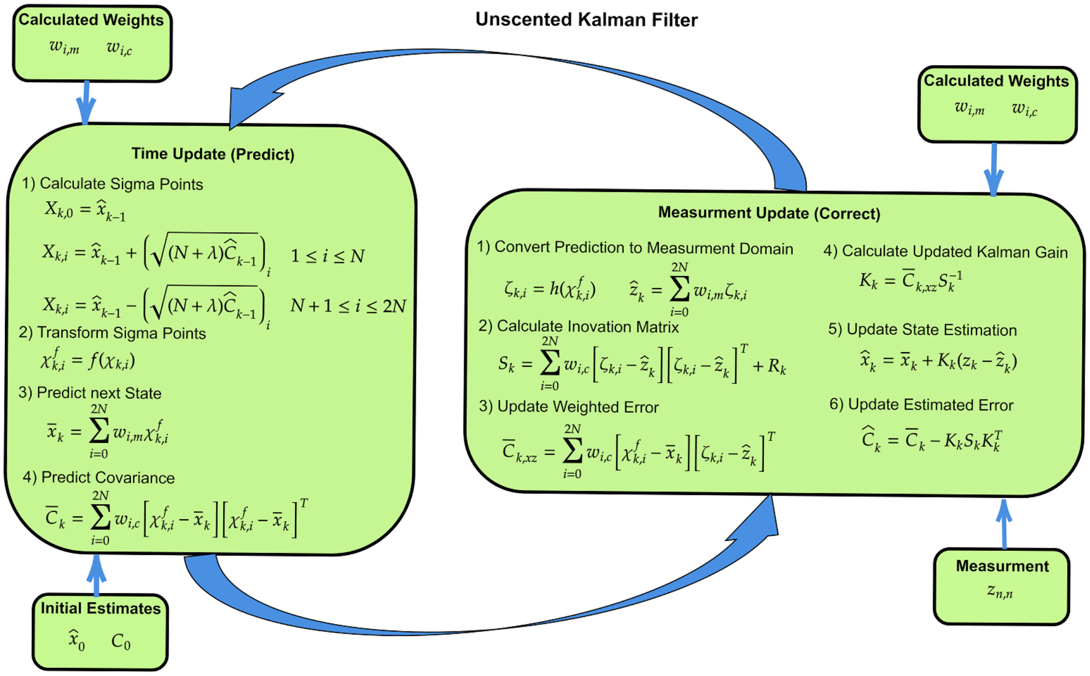
$$\mathbf{\hat{x}_k} = \mathbf{\bar{x}_k} + \mathbf{K_k}(\mathbf{z_k} - \mathbf{\hat{z}_k}) \tag{5.60}$$

following the same conditions as the Kalman filters $\mathbf{K_k}$ and the UKF's $\mathbf{K_k}$ from equations 5.31 and 5.32, the $\mathbf{K_k}$ weights the measurement difference based on the trust of the measurement.

Before the next iteration begins, the measurement covariance updates the expected covariance:

$$\mathbf{\hat{C}_k} = \mathbf{\bar{C}_k} - \mathbf{K_k S_k K_k^T}. \tag{5.61}$$

The UKF algorithm is pictured in figure 5.5, showing how the prediction and update steps work together to converge on a target through the iterative process.



Unscented Kalman Filter
Calculated Weights
$w_{i,m}$ $w_{i,c}$
Time Update (Predict)
1) Calculate Sigma Points
$X_{k,0} = \hat{x}_{k-1}$
$X_{k,i} = \hat{x}_{k-1} + \left(\sqrt{(N+\lambda)\hat{C}_{k-1}}\right)_i \quad 1 \le i \le N$
$X_{k,i} = \hat{x}_{k-1} - \left(\sqrt{(N+\lambda)\hat{C}_{k-1}}\right)_i \quad N+1 \le i \le 2N$
2) Transform Sigma Points
$\chi^f_{k,i} = f(\chi_{k,i})$
3) Predict next State
$\bar{x}_k = \sum_{i=0}^{2N} w_{i,m}\chi^f_{k,i}$
4) Predict Covariance
$\bar{C}_k = \sum_{i=0}^{2N} w_{i,c}\left[\chi^f_{k,i} - \bar{x}_k\right]\left[\chi^f_{k,i} - \bar{x}_k\right]^T$
Initial Estimates
$\hat{x}_0$ C_0
Calculated Weights
$w_{i,m}$ $w_{i,c}$
Measurment Update (Correct)
1) Convert Prediction to Measurment Domain
$\zeta_{k,i} = h(\chi^f_{k,i})$ $\hat{z}_k = \sum_{i=0}^{2N} w_{i,m}\zeta_{k,i}$
2) Calculate Inovation Matrix
$S_k = \sum_{i=0}^{2N} w_{i,c}\left[\zeta_{k,i} - \hat{z}_k\right]\left[\zeta_{k,i} - \hat{z}_k\right]^T + R_k$
3) Update Weighted Error
$\bar{C}_{k,xz} = \sum_{i=0}^{2N} w_{i,c}\left[\chi^f_{k,i} - \bar{x}_k\right]\left[\zeta_{k,i} - \hat{z}_k\right]^T$
4) Calculate Updated Kalman Gain
$K_k = \bar{C}_{k,xz}S_k^{-1}$
5) Update State Estimation
$\hat{x}_k = \bar{x}_k + K_k(z_k - \hat{z}_k)$
6) Update Estimated Error
$\hat{C}_k = \bar{C}_k - K_k S_k K_k^T$
Measurment
$z_{n,n}$

Figure 5.5: Unscented Kalman Filter Algorithm

5.6 Estimated Uncertainty

For the traditional Kalman filter, $\mathbf{P}$ measures the state estimate error. Recall C_0, the initially estimated covariance. For this section, we will assume that $\mathbf{C_0}$ will match the measured error of the first measurement. The variances within $\mathbf{P_{n,n}}$ will decrease to zero as the radar continues to take accurate measurements [38], [39]. Recall (5.6),

$$\mathbf{K} = \mathbf{P_{n,n-1}H^T(HP_{n,n-1}H^T + R_n)}^{-1}.$$

Assuming that the Kalman filter is linear, the $\mathbf{H}$ transform reduces to an identity matrix, in this case (5.6) reduces to,

$$\mathbf{K} = \frac{\mathbf{P_{n,n-1}}}{\mathbf{P_{n,n-1} + R_n}}. \tag{5.62}$$

Under these circumstances, $\mathbf{K}$ approaches $\mathbf{I}$, as the values measurement uncertainty $\mathbf{R_n}$ reduces to zero. In this ideal case (5.7) reduces to,

$$\mathbf{\hat{x}_{n,n}} = \mathbf{\hat{x}_{n,n-1}} + \mathbf{I}(\mathbf{z_n} - \mathbf{\hat{x}_{n,n-1}}) = \mathbf{z_n}, \tag{5.63}$$

indicating that the Kalman filter fully trusts the measurement. Finally, for this ideal case (5.8) reduces to

$$\mathbf{P_{n,n}} = (\mathbf{I} - \mathbf{I})\mathbf{P_{n,n-1}} = 0. \tag{5.64}$$

While this ideal scenario is not possible, it does demonstrate how $\mathbf{R_k}$ weights the validity of the measurement and affects the accuracy of the predictions.

Recall the effect $\mathbf{K}$ has on the measurements, where a low $\mathbf{K}$ trusts the measurement and a high $\mathbf{K}$ increases the uncertainty of the measurement. Reducing (5.7) to approximately $\mathbf{\hat{x}_{n,n}} = \mathbf{\hat{x}_{n,n-1}}$ and (5.8) reduces to $\mathbf{P_{n,n}} = (\mathbf{I} - 0)\mathbf{P_{n,n-1}}$ when $\mathbf{K}$ is very low. With these phenomena, we will be able the analyze the performance of the EKF and UKF in Chapters 7 and 8.

5.7 Summary

In conclusion, this chapter has discussed the foundation of the EKF and UKF, along with the implementation of the EKF and UKF to track a single target using a phased array radar. It also discussed the limitations of the EKF. Furthermore, this chapter has demonstrated why the UKF has risen in popularity with the development of the UT. The next chapter will discuss the final simulation constraints, preliminary results, and the results of the Monte Carlo.

Chapter 6

Monte Carlo Simulation

There are a variety of techniques widely used to test the limits and capabilities of a system. The one chosen for this simulation to test the beamforming along with the UKF and the EKF capabilities is the Monte Carlo. A Monte Carlo simulation essentially applies statistical analysis to random sampling to converge on a solution to the results [40]. A Monte Carlo simulation is an offshoot of random experiments [40]. This chapter will discuss the setup of the Monte Carlo and how the simulation parameters that dictate the automated QPS and a lost target.

6.1 Simulation Design and Monte Carlo Setup

Table 6.1 holds all the parameters for the DAR system are dependent on. Several parameters are tested and randomly chosen for the simulation. A single target is randomly generated for each Monte Carlo iteration which runs for 50 CPIs; the time between each CPI is simulated to last 2 seconds. The simulation randomly chooses the target's position, velocity, and maneuvers at the beginning of each Monte Carlo. The targets range, velocity, and angle limits are predetermined as constraints. To ensure the target can be adequately tracked, within the natural constraints of the radar, the target's properties are randomly generated iteratively until the target's properties converge on a solution that satisfies the constraints.

A target's position must be within the radar's minimum and maximum range. A target's range can be at least the minimum range of 1 km and no more than the maximum range of 25 km for the entire 50 CPI Monte Carlo iteration. Because of the NED coordinate, the target's z position can not be a negative quantity or zero because it is an airborne target. The azimuth and elevation constrain the target's position as well. The target cannot exceed the field of view of the radar.

The PRF limits the radar's ability to detect a target's radial velocity. The randomly generated velocity of the target is chosen using a normal distribution giving the highest weight of 50% to the x component of the velocity vector. Then randomly distributing the remaining 50% to the y and z components of the velocity vector. The max velocity that a target can reach is $\pm 262 \frac{m}{s}$ due to the constraint of the PRF.

The number of target maneuvers is randomly generated from one to five. Acceleration and the length of the acceleration define a maneuver. The randomly chosen acceleration follows a normal distribution for each acceleration vector $< x, y, z >$ component. Weighing the x (north) component at 75 to 50% of the acceleration vector and the remaining percentage randomly distributed to the y (east) and z components. The length of time that the acceleration last is a Rayleigh random variable. The Rayleigh random variable gives more weight to lower accelerations, increasing the likelihood that a low acceleration will last longer than a high acceleration. This process can be visualized in Figure 6.1, giving four different Rayleigh distributions based on the previously randomly chosen acceleration. The maneuvers are randomly chosen and generated for the 50 CPIs within a Monte Carlo iteration and added to the velocity, which is added to an updated target's position to ensure that the range, velocity, and angle constraints are satisfied.

At the beginning of each iteration the radar does not know where the target is and is actively searching for the target. One of five Radar Cross-section (RCS) values is chosen

Parameter Reference Table			
Parameter	**Quantity**	**Units**	**Description**
F_c	3	GHz	Carrier Frequency
λ	0.0999	m	Carrier Frequency Wavelength
B	100	MHz	Bandwidth
F_s	250	MHz	Sampling Frequency
T_p	5	μs	Pulse Width
PRF	10.5	KHz	Pulse Repetition Frequency
PRI	0.952	μs	Pulse Repetition Interval
γ	2×10^{13}	$\frac{cycles}{s^2}$	Chirp Rate
N_p	128	N/A	Number of Pulses per CPI
N_e	1024	N/A	Number of Antenna Elements
P_{tx}	10	$Watts$	Transmitted Power per Element
Δt_{CPI}	2	s	Time Between CPI
F_{adc}	75×10^3	$\frac{samples}{s}$	Receiver ADC Rate
N_{CPI}	50	N/A	Number of CPI's
R_{min}	1	Km	Minimum Range
R_{max}	25	Km	Maximum Range
R_{ref}	13	Km	Reference Range

Table 6.1: Radar Parameter Reference Table

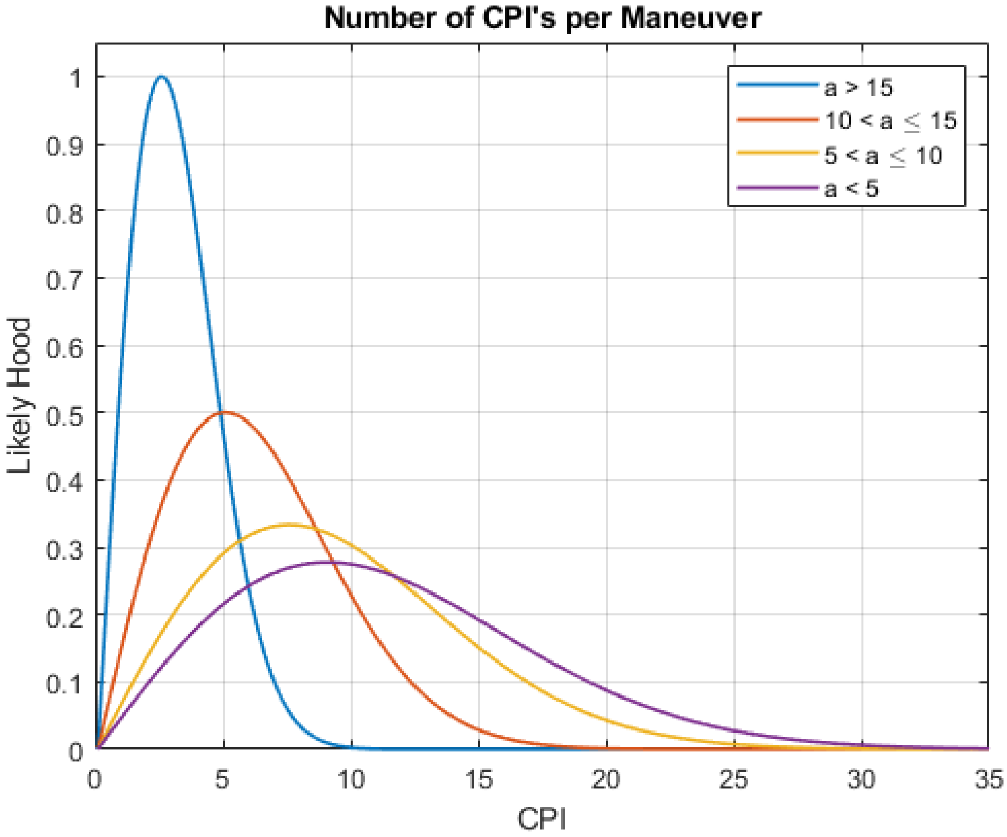


Figure 6.1: Rayleigh Distributions Based of Randomly Chosen Acceleration

for the target that would give the target between 5dB and 30dB SNR positioned at the mid-range of 13 km tracked by an unspoiled beam. Each one of the RCS values is simulated 100 times for each the UKF and the EKF. The UKF and EKF are initiated with the automated QPS enabled with a default factor of four, the automated QPS enabled with a default spoiling factor of eight, and the QPS disabled. With this, the Monte Carlo randomly generates 3000 targets, giving the possibility of up to 150,000 CPIs generated over the entirety of the simulation.

6.2 Conditions for QPS and Lost Targets

Figure 6.2 displays the process in a flow diagram that dictates the conditions for spoiling. The first rule states that if a target is not within a beam and the detected monopulse angle either being the minimum or maximum possible detection. The radar will then throws a spoiling flag for that particular direction for the upcoming CPI. The following rule states that a "fall" flag will be thrown when a target is detected $\frac{Bw}{4}$ from the center of the receive beam. If the "fall" flag is thrown for four consecutive CPIs, the target is considered to be "falling" out of the beam, triggering the beam spoiling for that particular direction. For the Monte Carlo simulation, the default QPS factors are eight and four. With these conditions set, the DAR now has a basic automated QPS.

For these QPS factors, a factor of eight illuminates $25.38°$ of the scene in the respective spoiled direction, while a QPS factor of four is half that and illuminates a scene of $12.69°$ in the respective spoiling direction. However, because of this, there is a performance trade-off. For a QPS factor of eight, the beam is spoiled in one direction the transmit beam's power contributions to the $\mathbf{SNR}$ decrease by a factor of eight. If the beam is spoiled in azimuth and elevation, the transmit beam contributions are decreased by 64. The decrease in $\mathbf{SNR}$ contributions is due to the decreased power density of the illuminated transmitted beam. With a QPS factor of four, the transmit beam's contributions to the $\mathbf{SNR}$ are decreased by four when spoiled in a single direction and by 16 when spoiled in both directions. Again, this is due to the transmitted beam's power density being decreased by the spoiled beam.

Figure 6.3 details the lost target decision process. A target is lost if the difference between the current target's detected range and the previous range is 1.5 km; the detection is invalid, throwing a flag and resetting the tracker's predicted range to the previous CPI range. Three consecutive CPIs of range flags target indicates a lost target. Suppose a measured position vector is complex due to severely inaccurate detected angles. In that case,

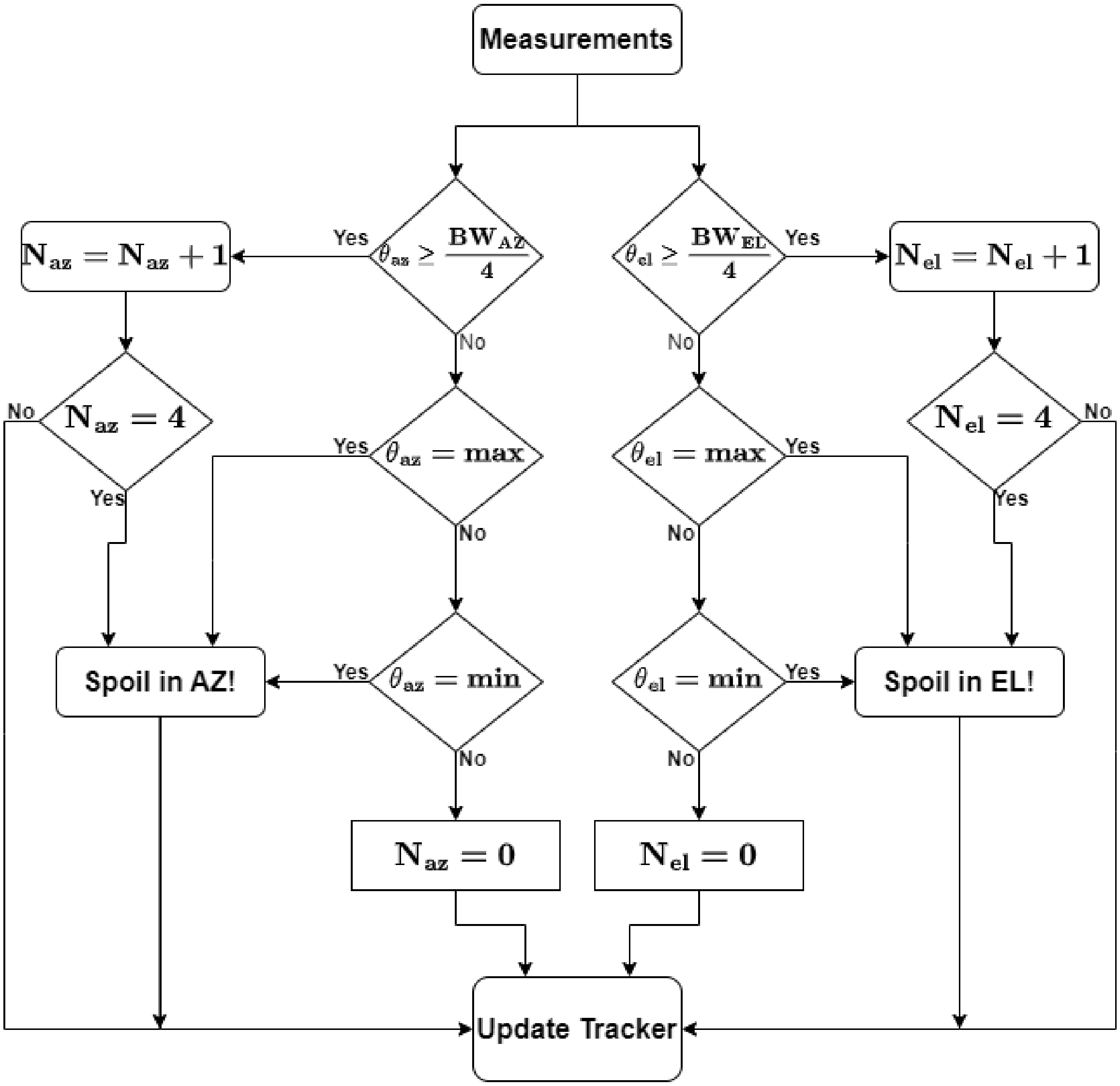


Figure 6.2: Flow Diagram of the Spoiling Decision Process

a complex flag is triggered, and the predicted position of the target is set to the previous CPI's position prediction. three consecutive CPIs of complex flags indicate a lost target. A lost target terminates the current Monte Carlo iteration, and the following Monte Carlo iteration begins randomly generating a new target.

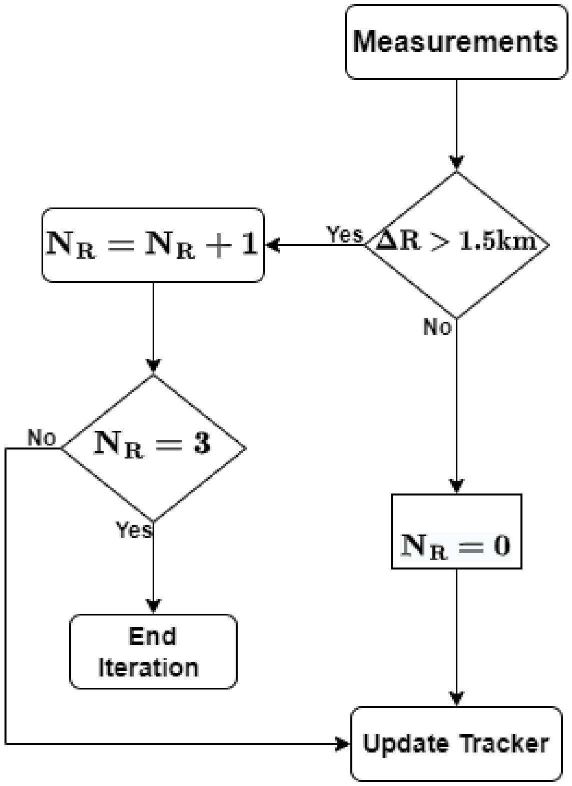


Figure 6.3: Flow Diagram of Decision Process for Lost Targets

6.3 RCS Values and SNR due to Contributions of the Radar

Recall that RCS values were chosen based on a target's range at 13 km. The following process finds the RCS values. First, the radar equation can be restructured to calculate the received power,

$$P_{Rx} = \frac{P_{Tx} G \lambda \sigma}{(4\pi)^3 R^4} \tag{6.1}$$

where P_{Tx} is the transmitted power of a single emitter, G is the gain of an antenna element, σ is the radar cross-section, and R is the range of a target. Giving the received power on a single antenna element. Several variables can increase the P_{Rx}. Modifying (6.1),

$$\hat{P}_{Rx} = P_{Rx} N_e B T_p N_p \tag{6.2}$$

given the increased $\hat{P}_{Rx}$ due to the number of antenna elements (N_e) within an array, the time bandwidth product of the bandwidth (B) and the pulse width (T_p), and the number of pulses (N_p) in a pulse train. With the modified $\hat{P}_{Rx}$ from equation 6.2, the $\mathbf{SNR}$ is represented in dB can be taken at 5.18 to achieve the $\mathbf{SNR}$ of a received signal in decibels. Calculating the $\hat{P}_{Rx}$ using an iterative process for different RCS values, the equation converges on five unique solutions for RCS values that range from 10 dB to 30 dB with the results cataloged in Table 6.2.

RCS Results	
SNR (dB)	RCS
10	0.21
15	0.67
20	2.12
25	6.72
30	20.78

Table 6.2: Target RCS Values used in the Simulation

The $\mathbf{SNR}$ contribution due to the target will vary due to several factors. First is the target's range; the closer a target is to the radar, the better the $\mathbf{SNR}$ it will have. The greater the distance a target is from the radar, the worse the $\mathbf{SNR}$ contributions are due to the target. Secondly, the spoiling factor decreases the $\mathbf{SNR}$ contributions. Modifying (5.18) with the contributions due to $\hat{P}_{Rx}$, and accounting for the spoil factor of the QPS gives,

$$\mathbf{SNR} = \frac{\hat{P}_{Rx}\frac{1}{S_f}}{P_{ns}} \tag{6.3}$$

where S_f is the spoil factor of QPS, the decrease in $\mathbf{SNR}$ due to the QPS's S_f is because of the spread of power on transmit and not a concentrated pencil beam of an unspoiled transmit beam.

6.4 Summary

For this chapter, we have discussed the type of test, the Monte Carlo, that will be performed on the system to test QPS using different Kalman filters to track randomly generated targets of chosen RCS values. The section further explained the different conditions that trigger the QPS and the conditions that test if a target is lost. The parameters the radar system used were discussed as well. With those subjects covered, the next chapter will discuss the preliminary results of the trackers.

Chapter 7

Tracker Performance

This chapter evaluates the performance of both the UKF and the EKF. Beginning with a discussion on both trackers' ability to track a target's x (north) and y (east) positions. Next, there will be a discussion on the estimated uncertainty and how the estimated uncertainty can is used to assess the performance of both trackers. The chapter will conclude with a discussion on both trackers performed with and without the automated QPS.

7.1 Two-Dimensional Tracker Performance

This section will discuss the ability of the UKF and EKF to estimate a target's position and velocity while the target performs a series of maneuvers. A maneuver is a stress test for each tracker because a maneuver causes a variant in a target's position concerning change over time. The observations made in this section will help to understand each tracker's ability to estimate a single target's attributes as the target's attributes change over time.

Figure 7.1 is a two-dimensional plot of a target's x (north) and y (east) positions tracked by the UKF. The UKF appears to track the target with estimates following the same trend as the positions. The updated estimates appear to trust the measurements since the updated estimates converge on the target as the change in position stays constant. However, when the change in position is no longer constant, the UKF estimates become inaccurate. The

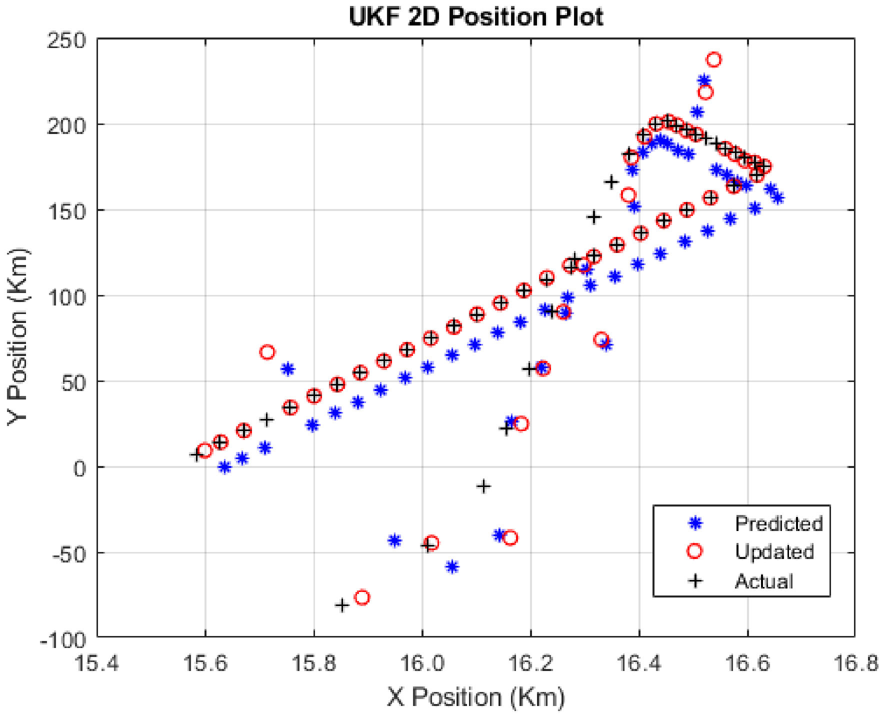


Figure 7.1: Tracked X and Y Positions Using UKF

inaccurate estimates cause the UKF to struggle to converge on the target until the change in position of the targets is constant again.

Figure 7.2 is a plot of the x and y velocities corresponding to the positions plotted in Figure 7.1. The UKF does not appear to estimate the target's velocity components accurately. The inaccurate estimates and updates could be due to the UT estimating the measurement covariance by approximating the sigma points. Another cause of the inaccurately estimated velocity components by the UKF appears to be caused by the measurement variances.

Figure 7.3 is a two-dimensional plot of a target's x (north) and y (east) positions tracked by the EKF. The EKF appears to follow the trend of the target's actual positions. The EKF's updated estimates appear to trust the predictions after the measurement step. However, after the rate at which the target's position changes, the EKF appears to estimate the change. The updated estimates again tend to trust the predictions allowing the EKF to estimate the

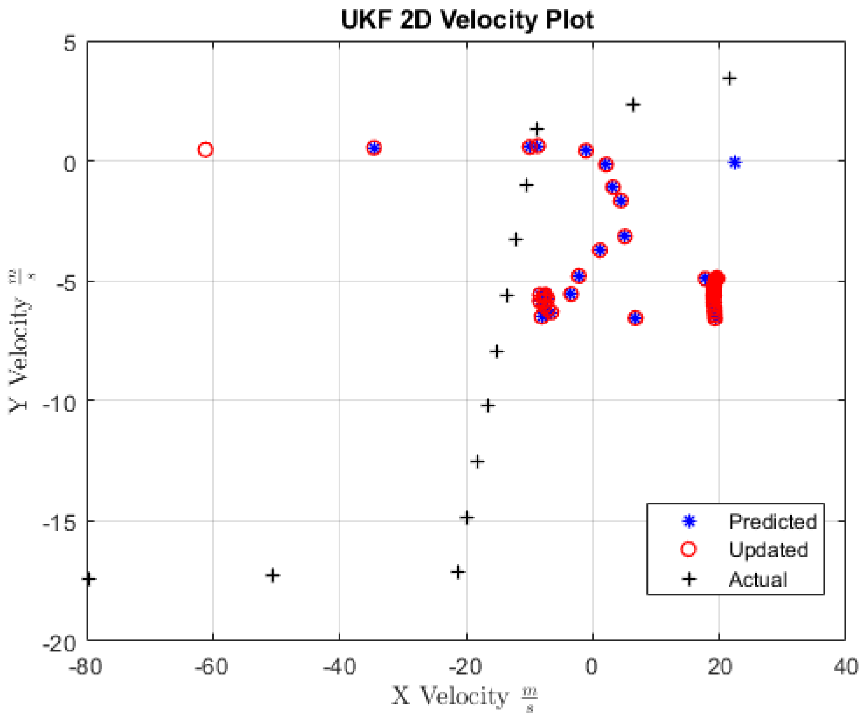


Figure 7.2: Tracked X and Y Velocity Using UKF

target's actual positions accurately.

Figure 7.4 is a plot of the x and y velocities corresponding to the positions plotted in Figure 7.3. The EKF does not converge on the target's actual velocity components. However, it appears the EKF does estimate the change in velocity. Again the EKF tends to trust the prediction after a measurement. The difference in behavior from the UKF could be due to the EKF algorithm propagating the covariance through the system. The EKF estimates the state vector using a Jacobian but propagates the true covariance from the prediction step to the update step. Another step that could have increased the accuracy of the estimates could have been decreasing the measurement variances, assuming that the radar is tracking a single target.

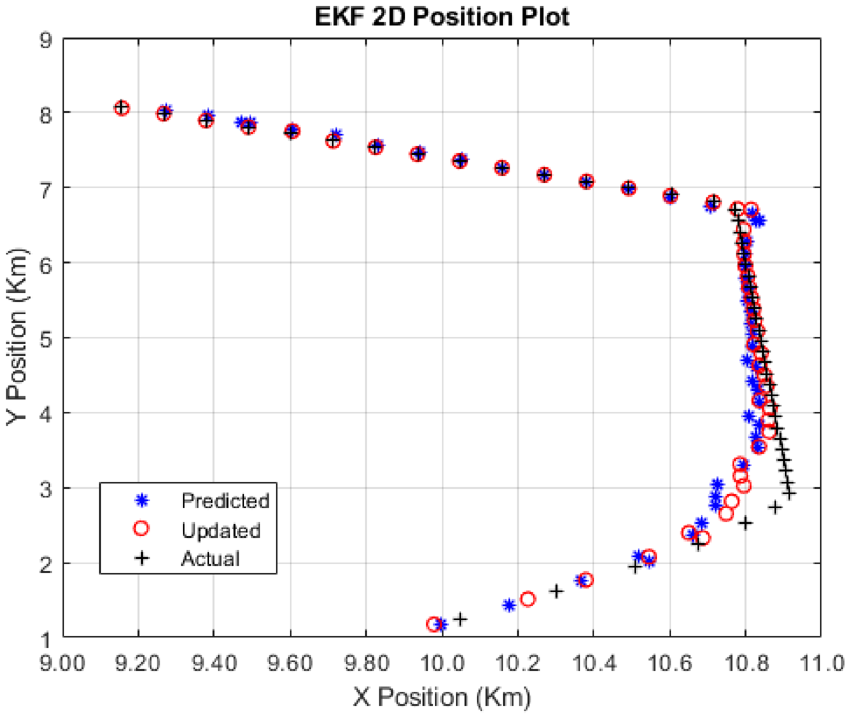


Figure 7.3: Tracked X and Y Positions Using EKF

7.2 Estimated Uncertainty

To better understand the performance of the UKF and EKF under the different conditions, the determinant ($\mathbf{det}(\bar{\mathrm{C}})$) of the estimate error covariance ($\bar{\mathrm{C}}_{\mathbf{k}}$) is calculated on each CPI. The initial error tends to be primarily due to the lack of measurements to correct the estimated error. However, with the accumulation of measurements taken, the $\mathbf{det}(\bar{\mathrm{C}})$ values fall sharply following an exponential decay.

Recall the discussion from Section 5.6 about the non-ideal circumstances of the traditional Kalman filter. The discussion concludes with the effect $\mathbf{K}$ has on the measurements, where a low $\mathbf{K}$ trusts the measurement and a high $\mathbf{K}$ increases the uncertainty of the measurement. Reducing (5.7) to approximately $\hat{\mathbf{x}}_{\mathbf{n,n}} = \hat{\mathbf{x}}_{\mathbf{n,n-1}}$ and (5.8) reduces to $\mathbf{P}_{\mathbf{n,n}} = (\mathbf{I} - 0)\mathbf{P}_{\mathbf{n,n-1}}$ when $\mathbf{K}$ is very low. The concept of $\mathbf{P}_{\mathbf{n,n}} = (\mathbf{I} - 0)\mathbf{P}_{\mathbf{n,n-1}}$ trans-

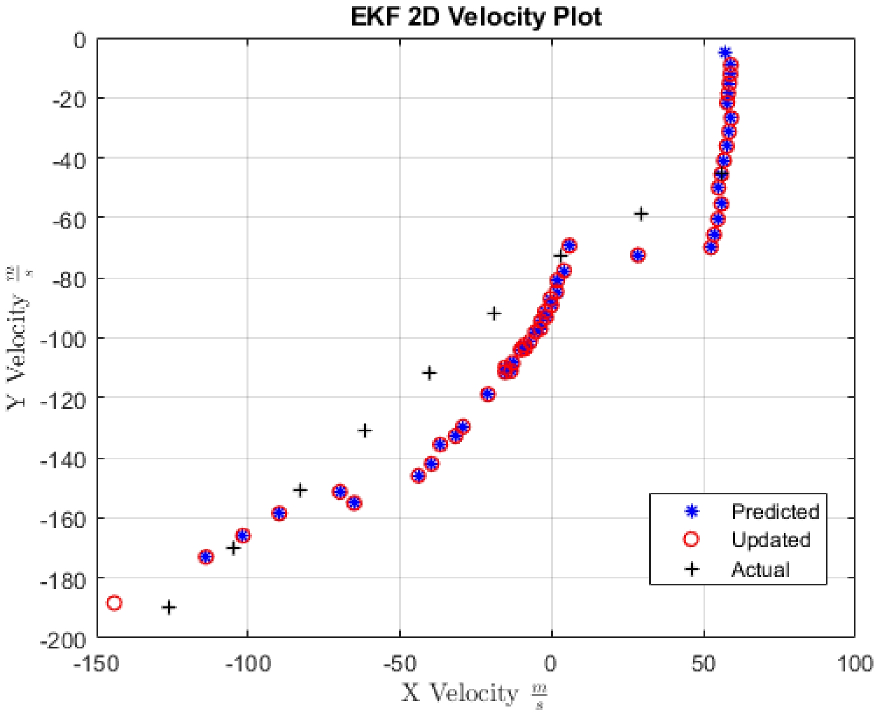


Figure 7.4: Tracked X and Y Velocity Using EKF

lates to the behavior of the EKF and UKF's covariance behavior.

Ten targets that the UKF tracked were chosen, and their **det**($\bar{\mathrm{C}}$) were plotted in Figure 7.5. The conditions of these particular targets were an RCS value of 20.78 and a QPS spoiling factor of eight. After the **det**($\bar{\mathrm{C}}$) decays to near zero, there are sudden increases to follow a saw tooth pattern. Recall (5.45),

$$\bar{\mathbf{C}}_{\mathbf{k}} = \sum_{i=0}^{2N} w_{i,c} \left[\chi^{\mathbf{f}}_{\mathbf{k,i}} - \bar{\mathbf{x}}_{\mathbf{k}}\right] \left[\chi^{\mathbf{f}}_{\mathbf{k,i}} - \bar{\mathbf{x}}_{\mathbf{k}}\right]^T + \mathbf{Q}$$

increases in **det**($\bar{\mathrm{C}}$) lead to (5.45) not reducing,

$$\bar{\mathbf{C}}_{\mathbf{k}} = \hat{\mathbf{C}}_{\mathbf{k}}$$

this indicates a missed measurement, and **det**($\bar{\mathrm{C}}$) will continue to increase due to **Q** being a non-negative matrix. Recall (5.61),

$$\mathbf{\hat{C}_k} = \mathbf{\bar{C}_k} - \mathbf{K_k S_k K_k^T}$$

where both $\mathbf{S_k}$ and $\mathbf{K_k}$ are controlled by the error between the measurement and the prediction, any discrepancy between the measurement error causes $\mathbf{\bar{C}_k} = \mathbf{\hat{C}_k}$ not to be satisfied. However, **det($\mathbf{\bar{C}}$)** quickly decays again once a sequence of valid measurements are taken satisfying $\mathbf{\bar{C}_k} = \mathbf{\hat{C}_k}$.

As for the floor of **det($\mathbf{\bar{C}}$)** the never completely reaching zero and stabilizes at 10^{-7}. This phenomenon is due to the estimated measurement variance $\mathbf{R_k}$ calculated on each iteration which controls $\mathbf{S_k}$ and $\mathbf{K_k}$. Because $\mathbf{R_k}$ will never perfectly estimate the variance for each measurement, the floor of **det($\mathbf{\bar{C}}$)** is only as good as the accuracy of $\mathbf{R_k}$. However, the more accurately the variance in $\mathbf{R_k}$ can be estimated, the lower the floor of **det($\mathbf{\bar{C}}$)**, increasing the likelihood of the UKF's ability to track a target.

The Same ten target's estimated $\mathbf{SNR}$ is plotted for each tracked CPI in Figure 7.6. The difference in $\mathbf{SNR}$ can be due to several factors. The transmit and receive beams may not be pointed directly at the target leading to a loss of $\mathbf{SNR}$. Another is that the beam has been spoiled, decreasing the transmitted power. A sudden drop in $\mathbf{SNR}$ indicates that a target has been lost, as shown by several targets.

Target 6 in Figure 7.6 $\mathbf{SNR}$ tends to fall from just below 25dB to just below 20dB until CPI 30, where the target's $\mathbf{SNR}$ steadily increases to 25dB. Corresponding to target 6's **det($\mathbf{\bar{C}}$)** in Figure 7.5, the **det($\mathbf{\bar{C}}$)** models that saw tooth behavior until CPI 30 where the **det($\mathbf{\bar{C}}$)** reaches its minimum value and stays for the remaining CPIs. Target 2's **det($\mathbf{\bar{C}}$)** in Figure 7.5 is never able to converge and satisfy $\mathbf{\bar{C}_k} = \mathbf{\hat{C}_k}$, eventually ending on the 32nd CPI. $\mathbf{\bar{C}_k} = \mathbf{\hat{C}_k}$ is confirmed by erratic $\mathbf{SNR}$ values of target 2 in Figure 7.6 being dropped on the 32nd CPI.

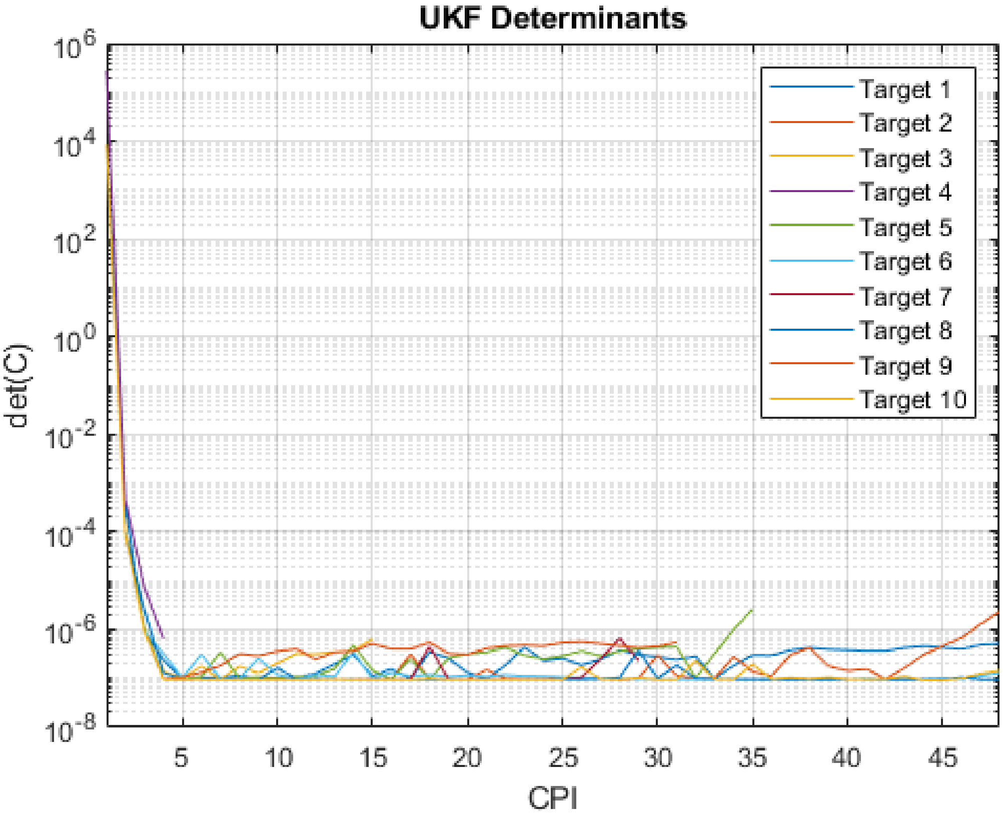


Figure 7.5: UKF **det**($\bar{\mathbf{C}}$) for 10 Targets with QPS Enabled

For the EKF, the **det**($\bar{\mathbf{C}}$) was plotted in Figure 7.7 with the same conditions as the **det**($\bar{\mathbf{C}}$) for the UKF previously discussed. Similarly to the UKF, the EKF also follows an exponential decay due to high estimated error on the first few CPIs. While some of the **det**($\bar{\mathbf{C}}$) follow the saw tooth behavior because of the brief period where $\bar{\mathbf{C}}_{\mathbf{k}} = \hat{\mathbf{C}}_{\mathbf{k}}$ is not satisfied, some are slower to decay. $\bar{\mathbf{C}}_{\mathbf{k}} = \hat{\mathbf{C}}_{\mathbf{k}}$ not satisfied is due to the variance between the prediction and target of the EKF being larger than the variance between the prediction and target of the UKF. However, the floor of **det**($\bar{\mathbf{C}}$) is lower for the EKF at 10^{-8}. The lower floor of **det**($\bar{\mathbf{C}}$) for the EKF is due to the $\mathbf{R}_{\mathbf{k}}$ being more accurately estimated.

Figure 7.8 presents the $\mathbf{SNR}$ values of the targets from Figure 7.7. While the error does not decay as sharply after a bad detection as with the UKF that was previously discussed, the EKF appears to have fewer variations in the $\mathbf{SNR}$ and is less likely to lose the target.

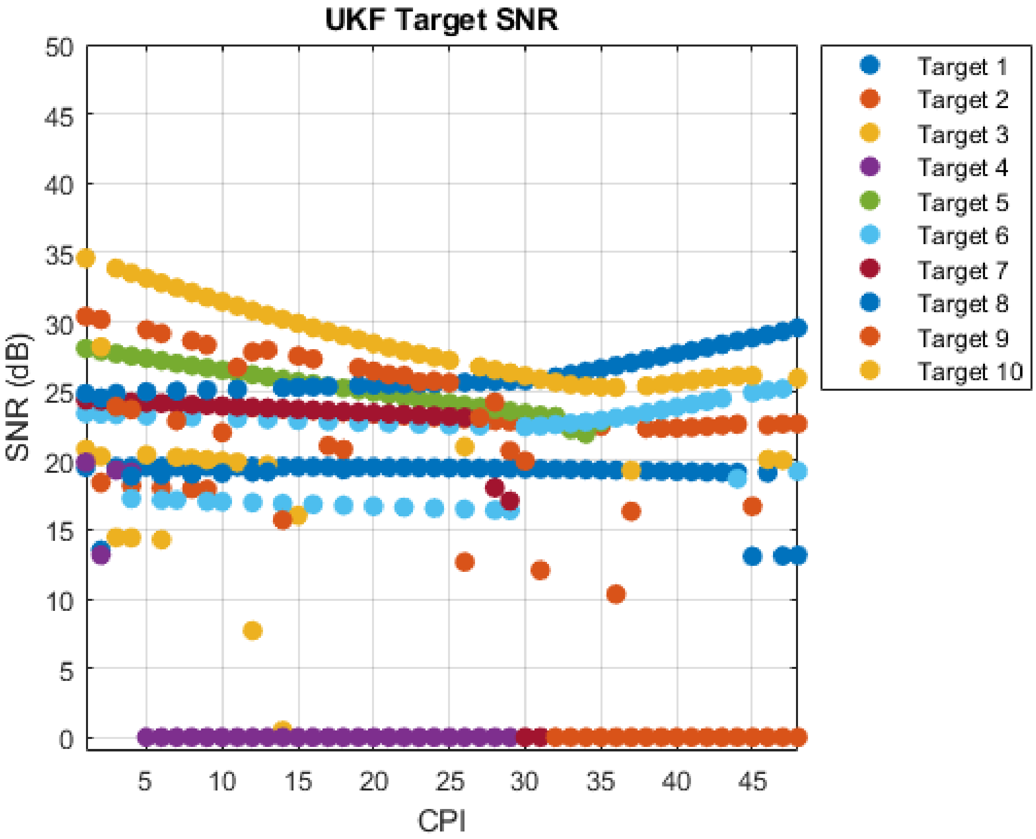


Figure 7.6: UKF SNR for 10 Targets with QPS Enabled

Target 4 in Figure 7.8 has a steady SNR of approximately 25dB until the 14th CPI, where the SNR drops to around 12dB. The SNR of the target recovers for a while, then inconsistent from CPI 38 until the end. As for target 4's $\mathbf{det}(\bar{\mathbf{C}})$ in Figure 7.7 the $\mathbf{det}(\bar{\mathbf{C}})$ never quite converges to the floor at the low SNR on the 14th CPI and the $\mathbf{det}(\bar{\mathbf{C}})$ increases again at the 39th CPI after SNR of the target begins to become erratic leading the $\mathbf{det}(\bar{\mathbf{C}})$ to never converge again. The inconsistent $\mathbf{det}(\bar{\mathbf{C}})$ is due to the large measurement variance, but the ability to continuously track the target is due to the high approximation of the measurement variance. Target 1 has the highest SNR just above 35dB in Figure 7.8 except for a lower SNR detection starting at the 41st CPI. Target 1's SNR could correlate to the relatively flat and low $\mathbf{det}(\bar{\mathbf{C}})$ of target 1 in Figure 7.7 with the saw tooth behavior at the

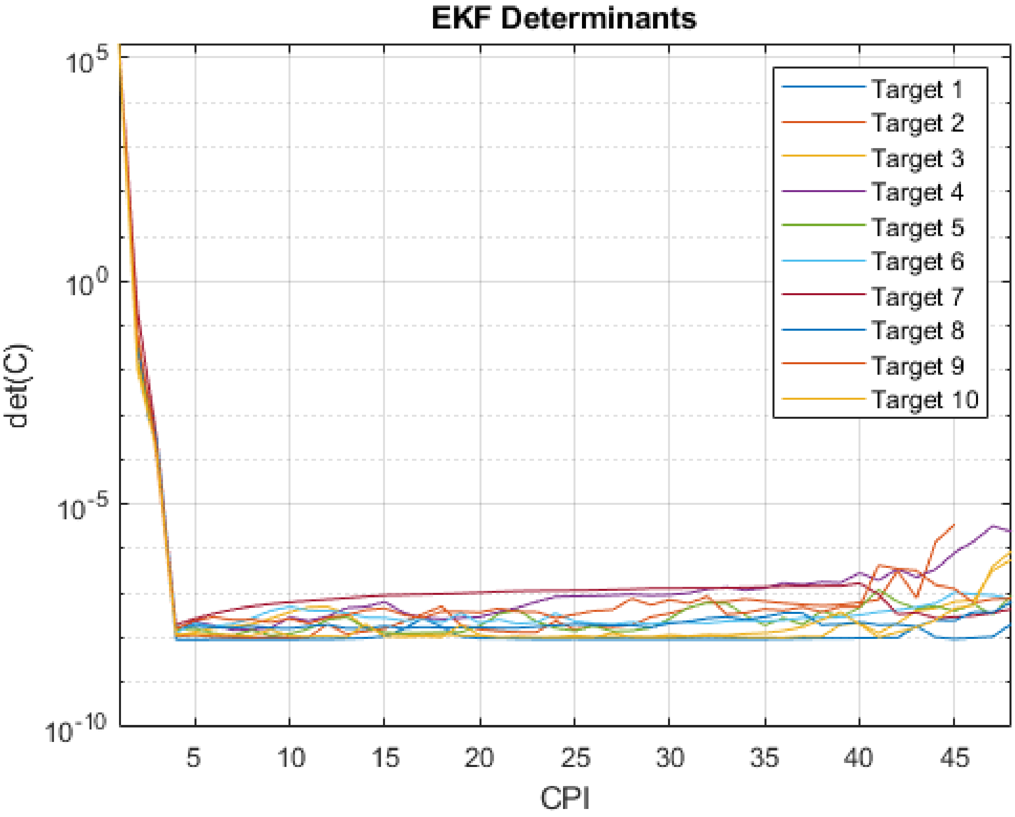


Figure 7.7: EKF **det**($\bar{\text{C}}$) for 10 Targets with QPS Enabled

end of the iteration due to the lower SNR detection's. These lower SNR detection's are most likely due to a maneuver causing the target to not be centered within the radars beams.

With the **det**($\bar{\text{C}}$) for each CPI cataloged. The **det**($\bar{\text{C}}$) analysis will help to understand the behavior of the UKF and EKF under different conditions.

7.3 Preliminary Results of Trackers and Phase Beamspoiling

Before discussing the results of the Monte Carlo simulation, a discussion of the trackers with and without the QPS enabled is presented.

Figure 7.9 demonstrates the UKF predicting a target's radial velocity and range. The initial predictions are relatively inaccurate, but the detections are accurate for the range and

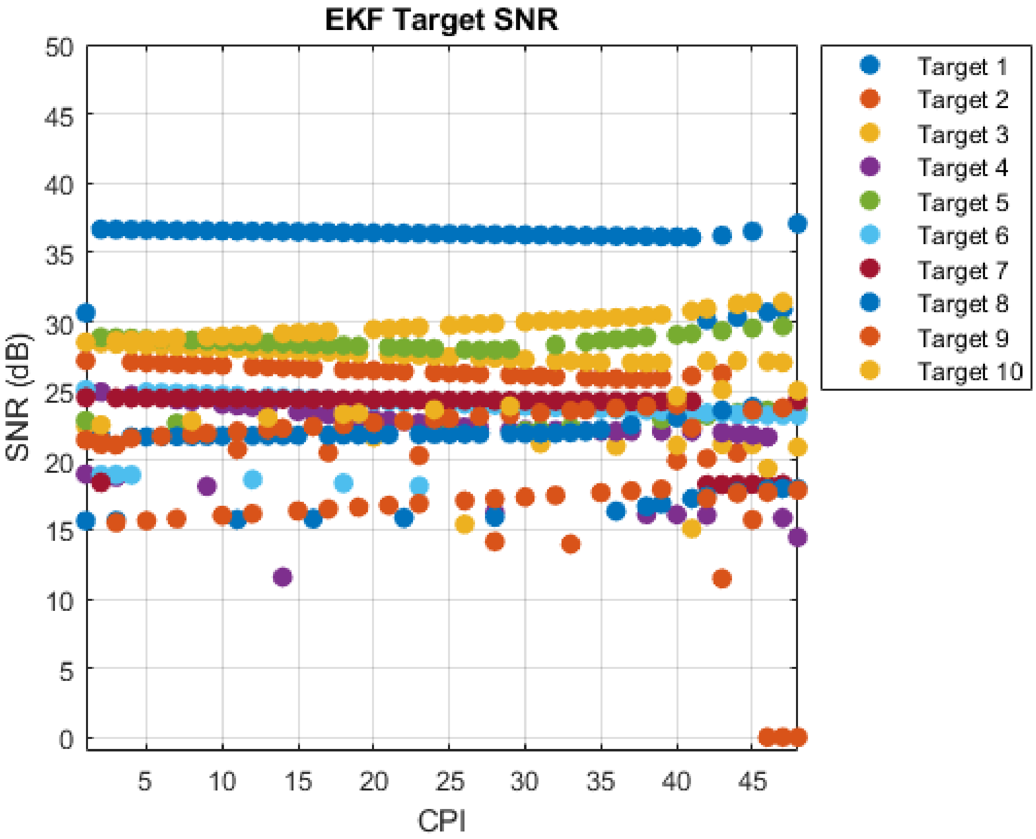


Figure 7.8: EKF **SNR** for 10 Targets with QPS Enabled

radial velocity. The updates begin to converge on the target's approximate position and radial velocity due to the decrease in error between the detection and prediction. The updated radial velocity seems to give more weight to the detection converging on the target's true radial velocity. The predicted range tends to oscillate around the target's true position until it converges on the target's true position. Nevertheless, the UKF lost the target after 23 CPIs.

Further exploring the UKF's behavior, the $\mathbf{det(\bar{C})}$ is plotted in Figure 7.10. It follows the exponential behavior due to increased measurements and the assumption that the tracker converges on the target's actual radial velocity and position. There is a slight increase causing the saw tooth behavior leading up to CPI 10, but it quickly decays. However, the

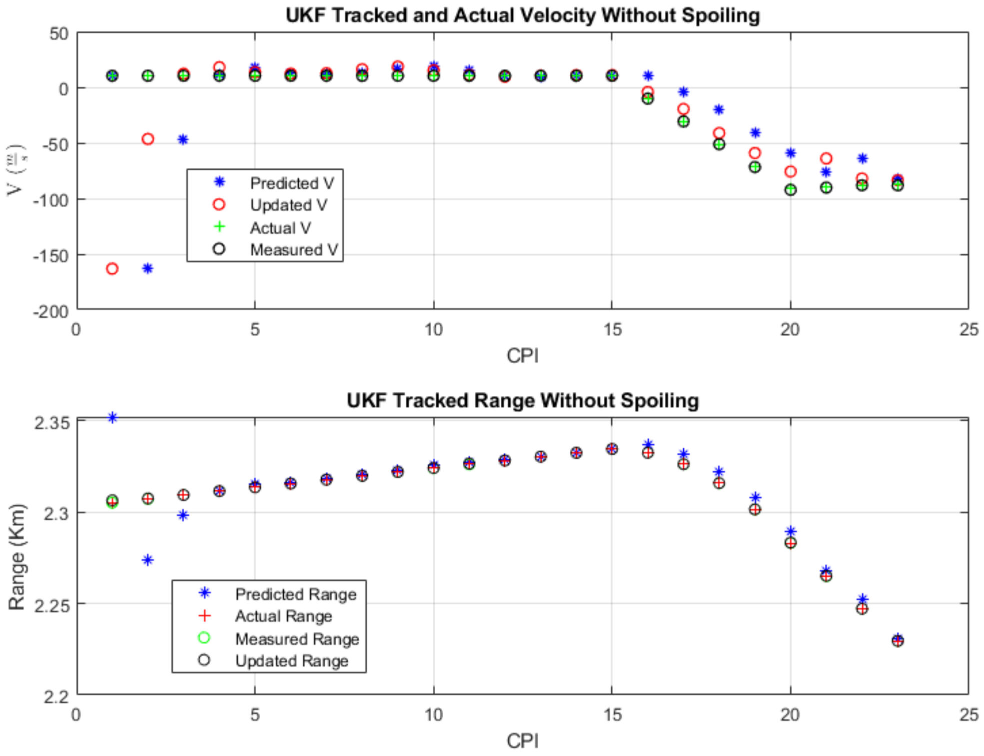


Figure 7.9: Range and Velocity Tracking With Beam Spoiling Turned Off

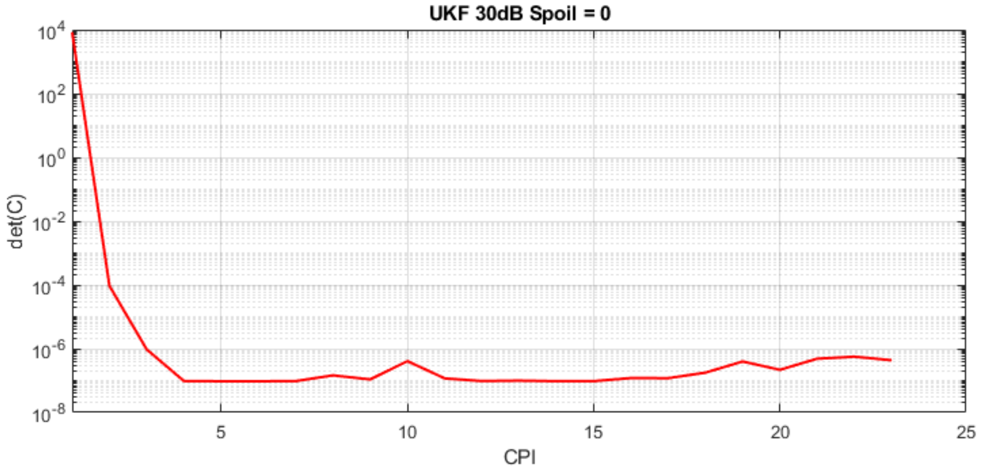


Figure 7.10: UKF Determinate of the Predicted Covariance For the Lost Target

det($\bar{\mathbf{C}}$) begins to diverge between CPI's 15 and 20, converges one more time, then diverges and is never able to satisfy $\bar{\mathrm{C}}_{\mathrm{k}} = \hat{\mathrm{C}}_{\mathrm{k}}$, leading to inaccurate predictions. Because the error between the prediction and the measurement never decreases to the point that allows $\bar{\mathrm{C}}_{\mathrm{k}} = \hat{\mathrm{C}}_{\mathrm{k}}$ to be satisfied.

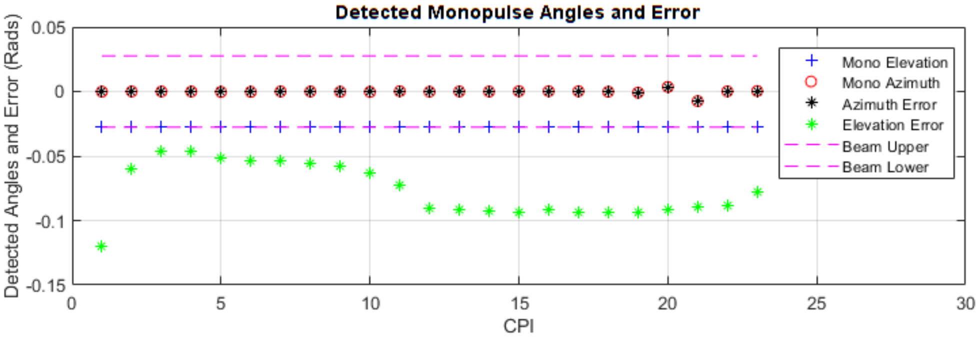


Figure 7.11: Monopuluse Detection and Error With Beam Spoiling Turned Off

To better understand the inability of the tracker to decrease the measurement error, Figure 7.11 shows the target's detected monopulse angles, actual error, and the beamwidth that limits the monopulse detentions. The azimuth monopulse angles observed show that the target's azimuth angles are centered in the receive beam. In contrast, the detected elevation monopulse angles are at the edge of the beam (-0.0277 radians). The angle of -0.0277 radians indicates that the target's actual Elevation angle is less than the radar's steered transmit and receive beams controlled by the predicted UKF coordinates of the target. With error it is clear that the radar is never able to detect the target's true angles.

Figure 7.12 shows the tracked azimuth and elevation angles along with the actual and measured angles of the target. Combining the predicted azimuth and elevation angles with the detected respective monopulse angles gives the target's measured azimuth and elevation angles. The measured azimuth angles of the target are reasonably accurate and within the threshold of the received beamwidth. However, the elevation angle between the target's x and z position is most likely the largest of the two detected angles and has the most prominent error between the predicted and actual Elevation angles. This error is more significant than half the bandwidth of the receive beam, which is the maximum angle the

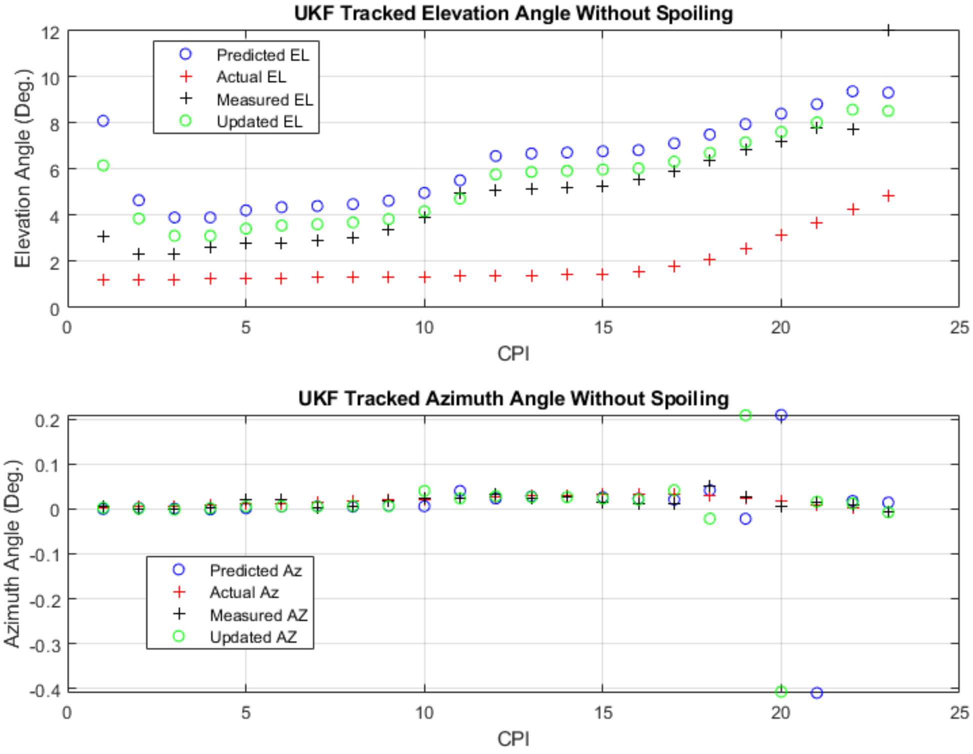


Figure 7.12: Azimuth and Elevation Angle Tracking With Beam Spoiling Turned Off

monopulse can detect. Therefore it is difficult for the UKF to converge on the target's actual elevation angle. The tracked elevation begins to converge on the target. It follows the trend of the target's change in the elevation direction but cannot converge on the target due to the limits of the max and min monopulse angles. The error tends to increase with each maneuver, the first maneuver is a slight change in velocity at the 10th CPI, the next coming at the 16th CPI, and the final coming at the 21st CPI. Figure 7.9 displays these maneuvers with the target's radial velocity on each CPI.

Figure 7.13 shows a target tracked by the UKF. The target performs three maneuvers while traveling toward the radar. The initial range and radial velocity predictions are somewhat inaccurate but converge on the target by the third CPI. The velocity prediction on the maneuvers follows the trend of the target's change in radial velocity and converges on the

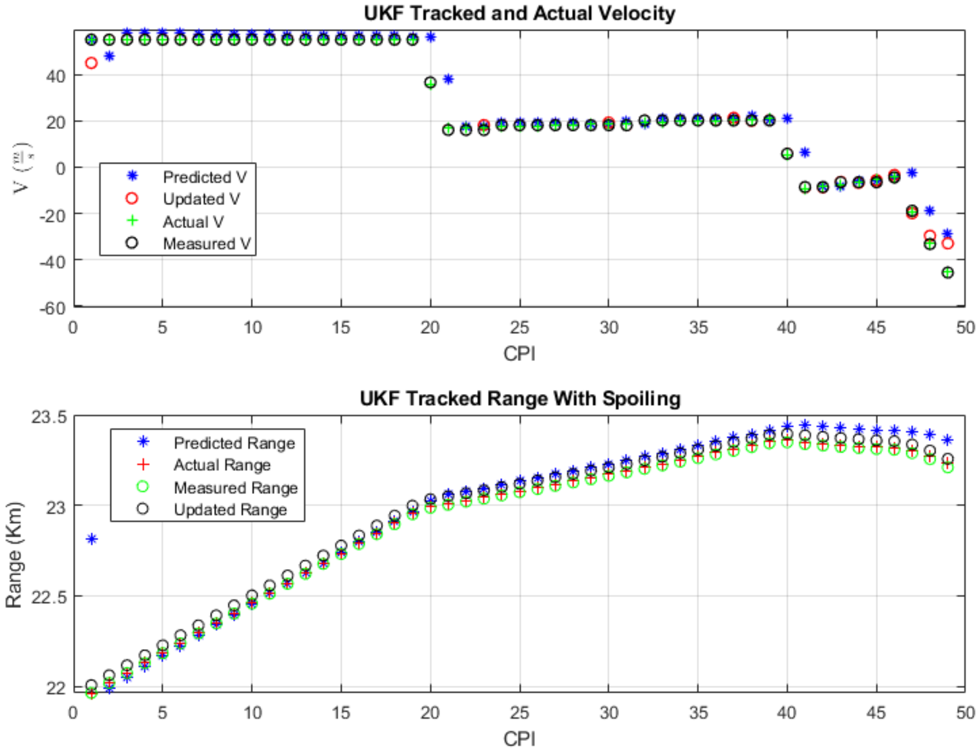


Figure 7.13: UKF Range and Velocity Tracking With Beam Spoiling Turned On

approximate actual radial velocity on the next CPI. Following the end of the maneuver on the first CPI of the third maneuver but with the Monte Carlo iteration ending, it is unknown if it would have converged in the 51st CPI. The target's range is tracked for the entire iteration and has a low error between the measured range and the predicted range relative to the target's position.

The **det($\bar{\mathrm{C}}$)** for this particular target is plotted in Figure 7.14. The saw tooth behavior happens shortly after CPI 20 due to the first maneuver and is repeated at CPI's 39-41 on the second maneuver. However, due to the last several maneuvers, the **det($\bar{\mathrm{C}}$)** struggles to converge until the acceleration is constant.

With introducing the quadratic phase beamspoiling to the simulation, the spoiling with respect to azimuth and elevation was tracked for each CPI. Figure 7.15 shows the compara-

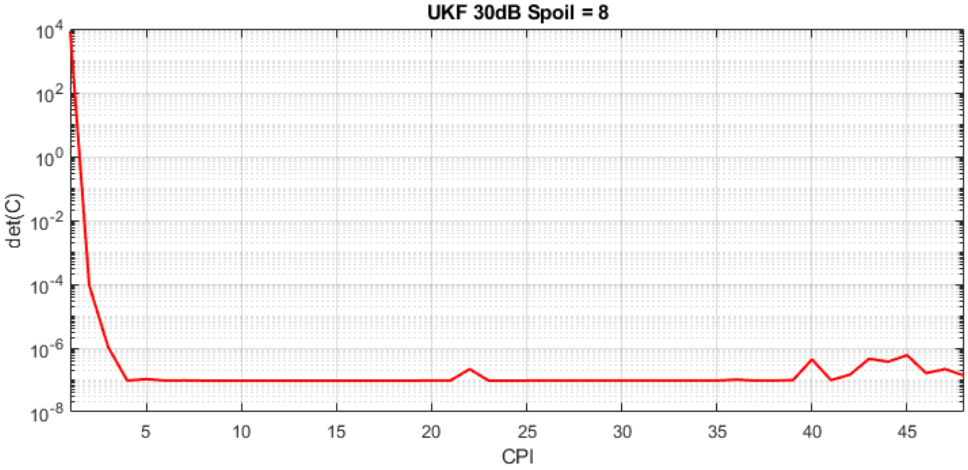


Figure 7.14: UKF Determinate of the Predicted Covariance For the Tracked Target

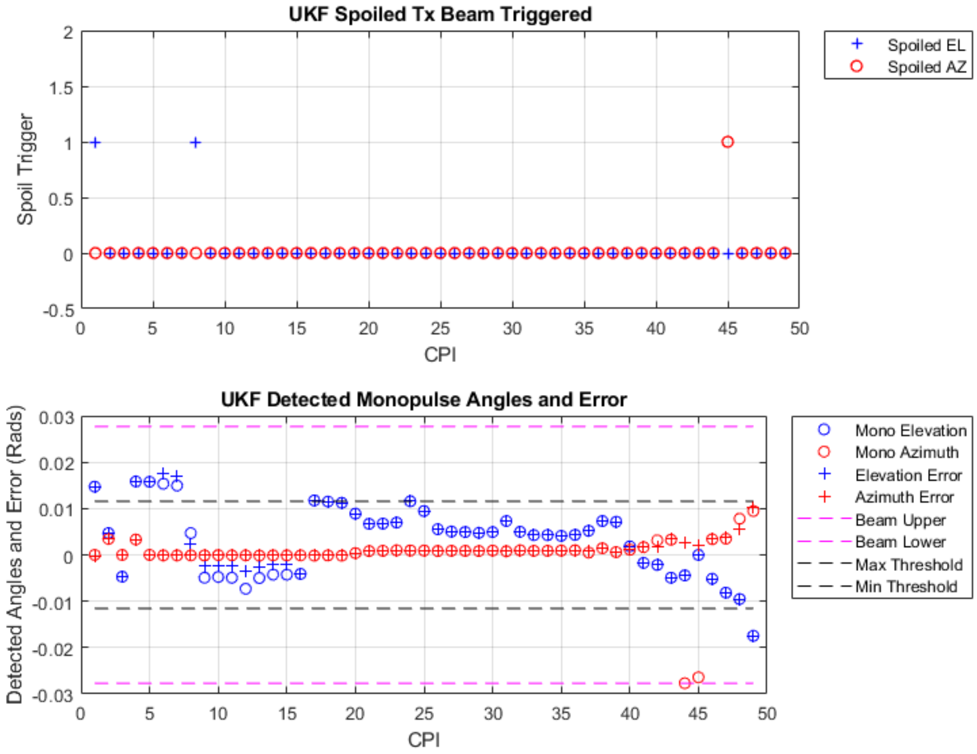


Figure 7.15: UKF Monopuluse Detection and Error With Beam Spoiling Turned On

ble data for CPI's 2 through 49 because the trackers do not start tracking until the 2nd CPI. The top plot represents the triggered spoiled beams. The detected monopulse angles, angle error, beam limits, and thresholds are in the graph below. The first initial direction in which the beam is steered in elevation is most likely in a direction in which the detected elevation monopulse angle is either the max or min detected, causing the radar to spoil the transmit beam in elevation to find the target. The next spoiling occurs during the 9th CPI due to the four previous elevation monopulse detections being consecutively out of the chosen threshold of $\frac{Bw}{4}$ radians from the center of the beam, triggering the “fall” flag. The “fall” flag indicates that the target is beginning to fall out of the beam. So the transmit beam is again spoiled in elevation. for this instance of spoiling the “fall” flag was valid due to the error also being out of the set tolerance. The “fall” flag would have been triggered again on the 22nd CPI, but the 21st CPI was detected below the threshold of $\frac{Bw}{4}$ radians, and the “fall” flag was reset, which agian was accurate due to the monopulse detections being similar to the angle error in elevation. The final beam spoiling occurs in the transmit beam’s azimuth direction due to the detected azimuth monopulse angle being the minimum monopulse angle. This instance was an invalid detection due to the error being much smaller than the false azimuth monopulse detection.

Figure 7.16 displays the angles of the target tracked by the UKF. The initial elevation angle prediction is highly inaccurate. However, it can converge on the target with the additional beam spoiling and track the target for the entirety of the iteration. Similarly, the first few azimuth detections are inaccurate but converge on the target once there are accurate detections. Towards the end of the iteration, the azimuth detections are slightly inaccurate, causing the UKF to diverge slightly but update and adjust towards the target on the last CPI.

Finally, the performance of the EKF with quadratic phase beamspoiling is analyzed.

Similar to the previous targets, this target is tracked by the EKF, whose radial velocity and range are plotted for each CPI in Figure 7.17. The target performs five maneuvers, the first and third being a slow acceleration that begins on the 17th CPI, the other two being

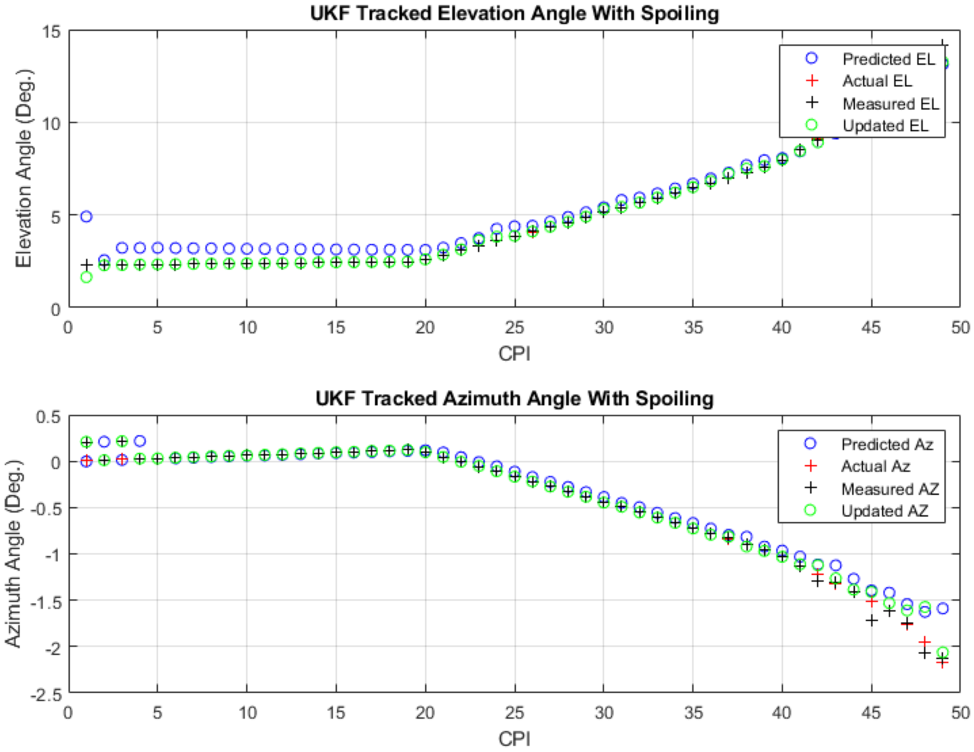


Figure 7.16: UKF Azimuth and Elevation Angle Tracking With Beam Spoiling Turned On

relatively large accelerations with respect to the first. The EKF tracks the change in velocity fairly accurately, converging close to the actual radial velocity. The range of the target is accurately tracked as the target moves away from the radar and then begins to turn back, moving towards the radar. The first prediction has more than a kilometer error from the detected range, with the EKF converging on the target's true range on the proceeding CPI.

The EKF tracked targets $\mathbf{det}(\bar{\mathbf{C}})$ is plotted in Figure 7.18. The saw tooth behavior happens after the target begins its first maneuver and converges; however, once the target begins to accelerate for a longer period, the $\mathbf{det}(\bar{\mathbf{C}})$ of the EKF does not completely converge until the third maneuver is complete.

As opposed to the UKF tracked target that included the addition of the quadratic phase beamspoiling, this iteration using the EKF had a few more spoiled beams triggered as

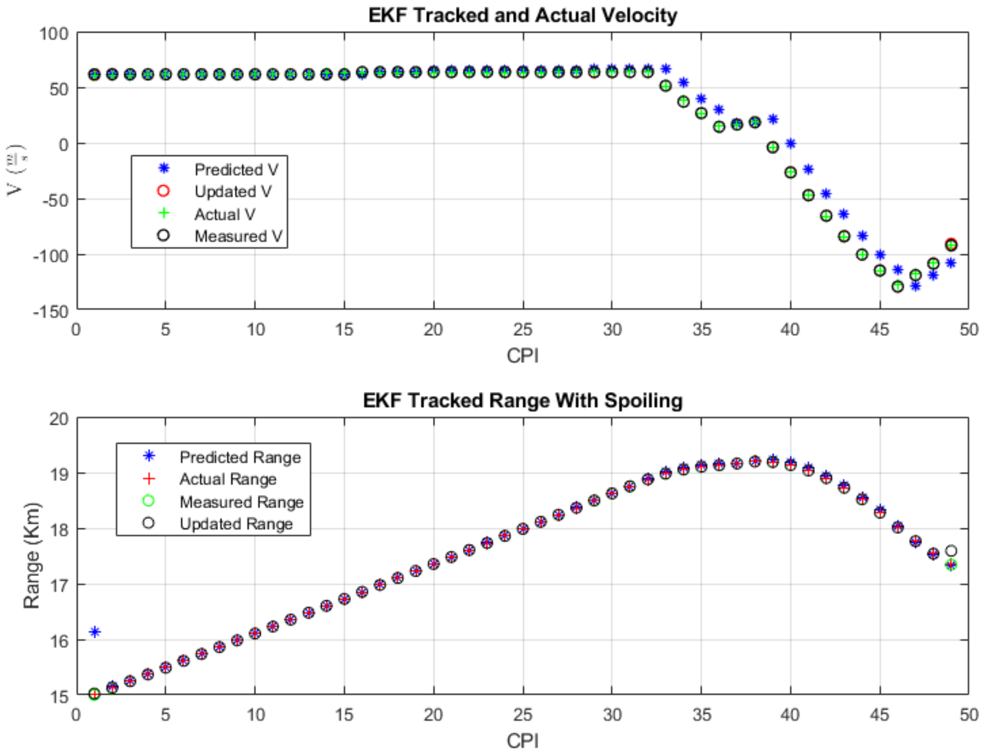


Figure 7.17: EKF Range and Velocity Tracking With Beam Spoiling Turned On

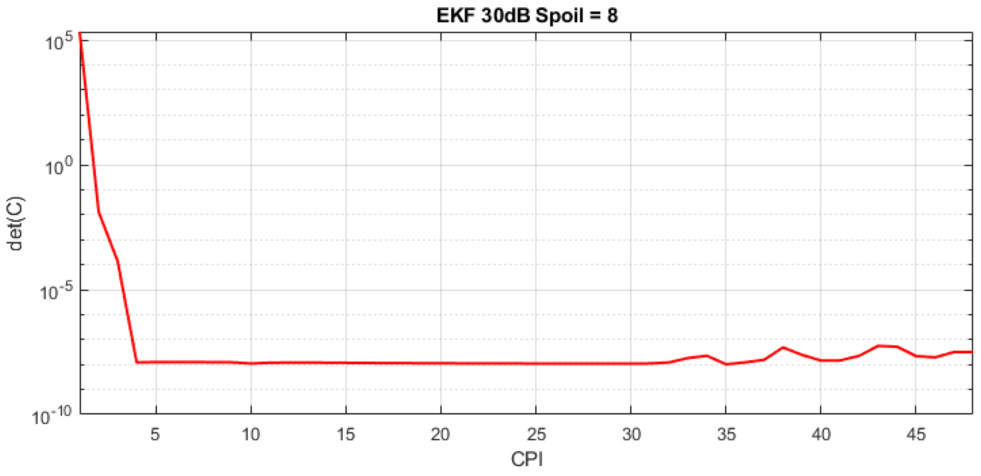


Figure 7.18: UKF Determinate of the Predicted Covariance For the Tracked Target

pictured in Figure 7.19. The transmit beam is spoiled in elevation at the beginning of the iteration due to an inaccurate elevation prediction that was outside the bounds of the receive

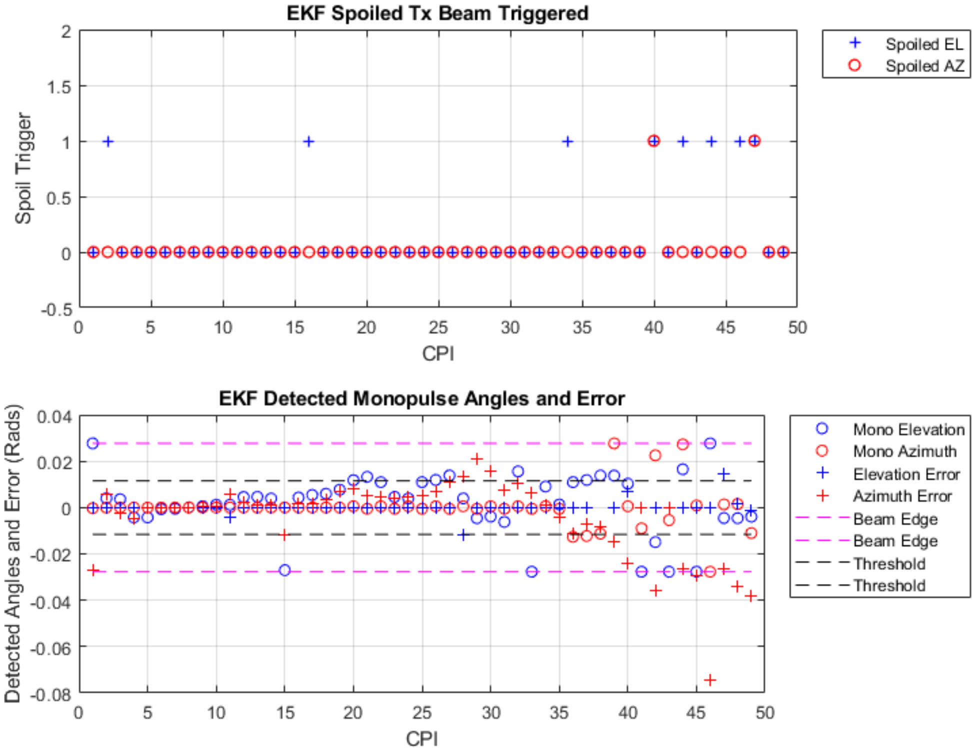


Figure 7.19: EKF Monopuluse Detection and Error With Beam Spoiling Turned On

beams monopulse table, which is a invalid detection due to the error not matching the detection. Then monopulse detection for both azimuth and elevation becomes a reasonably small indication that the target is close to the center of the receive beam tracking the target. Until the 26th CPI, the Elevation monopulse angle detected is again maxed out, causing the transmit beam to spoil on the proceeding CPI in Elevation. It does this several more times, most likely due to the significant changes in velocity toward the end of the iteration. Along with azimuth spoiling as well. The “fall” flag is triggered once for the elevation direction due to the assumption that the target is beginning to fall out of the beam, with measurements of the elevation angle from CPI’s 37 to 41.

Finally, the randomly generated target’s angles are shown to be tracked by the EKF in Figure 7.20. The beginning of the iteration shows that EKF starts to diverge in azimuth

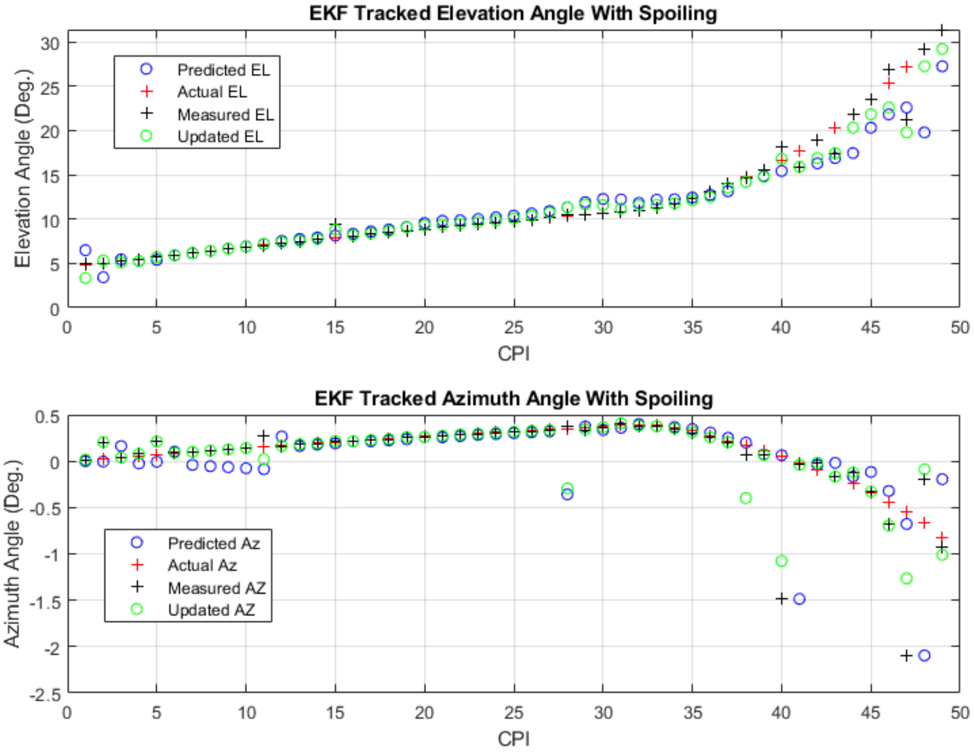


Figure 7.20: EKF Azimuth and Elevation Angle Tracking With Beam Spoiling Turned On

but quickly converges on the target's actual angle due to the detections and updates done to the EKF. In elevation, the tracker oscillates around the target's angle until it converges in the target's elevation angle. A few false detections are made during the more significant maneuvers causing the EKF to diverge from the target. However, once more accurate measurements are made, the EKF follows the trend of the target, beginning to converge on it again.

7.4 Summary

For this chapter, we have discussed performance of the UKF and EKF while tracking a single target. The section further explained the different conditions that trigger the QPS and the conditions that test if a target is lost. Discussed the preliminary results of the UKF with automated QPS disabled and what conditions led to lost targets. This chapter explained how the determinants of both trackers can be used to observe the performance of the trackers. The discussion led to early results of the UKF and EKF's ability to track a target due to the automated QPS ability to re-establish a target's position and radial velocity when a target performs a randomly generated maneuver, or the DAR shows signs of struggling to follow a targets path. With those subjects covered, the next chapter will discuss the results from the Monte Carlo and how effective or ineffective the automated QPS is.

Chapter 8

Results

This chapter will discuss the results of the Monte Carlo simulation. The chapter begins with the EKF and UKF utilizing the QPS for default transmit beam spoiling factors of eight and four and the performance of the EKF with the QPS disabled. The tracked targets with five RCS values are stated in the previous chapter for all three of these cases. Finally, presenting the quantitative results of each tracker, assessing the QPS's ability to improve a tracker's performance.

Due to the number of targets from the Monte Carlo, there was no need for legends for most figures within this chapter. This chapter is less interested in the behavior of a few targets and more in the overall behavior of the trackers under the different conditions. The following sections will focus on the overall behavior of the trackers when presented with different conditions.

8.1 EKF Performance

This section will discuss the performance of the EKF tracker with the different conditions presented. First, discussing the target's actual SNR and how it relates to the Determinate of the EKF's estimated error.

Figure 8.1 shows target SNR for the different RCS values referenced in Section 6.3

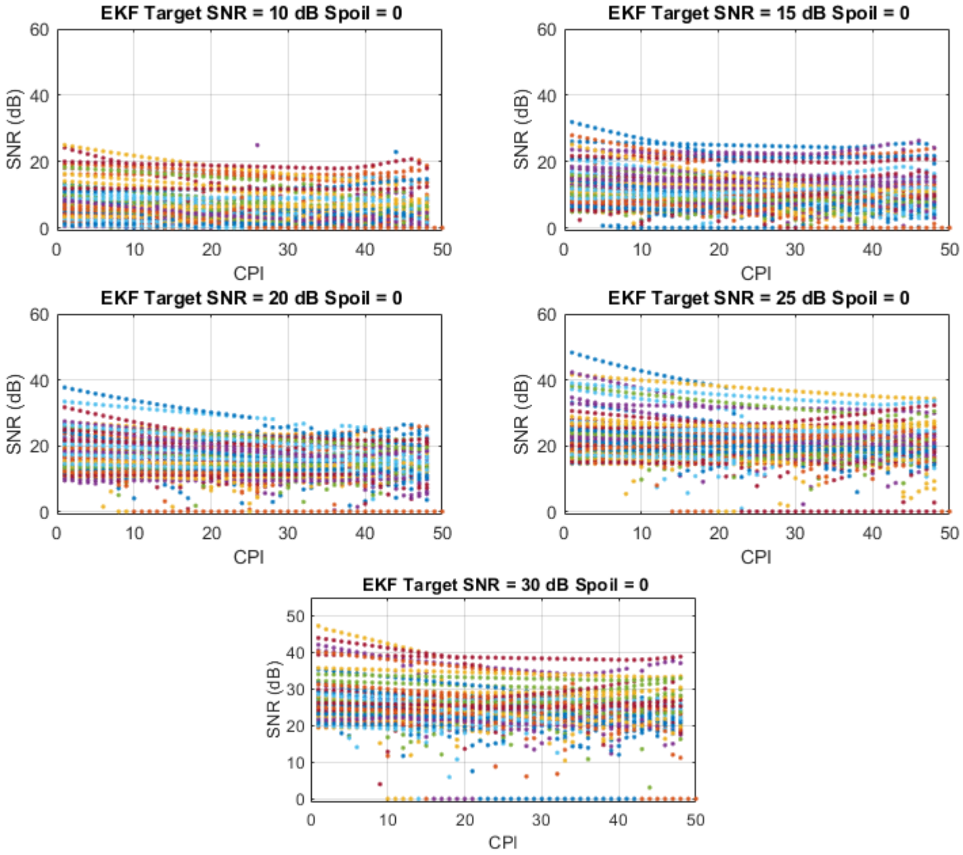


Figure 8.1: 100 Randomly Generated Targets $\mathbf{SNR}$ Tracked by the EKF With Spoiling Disabled

of Chapter 7. The QPS functionality was disabled for this EKF Monte Carlo iteration. Each plot contains 100 randomly generated targets. The $\mathbf{SNR}$ of the targets tends to trend toward the corresponding RCS value. With each increasing RCS value, the likelihood of a randomly generated target's $\mathbf{SNR}$ increased. All of the $\mathbf{SNR}$ values are within 10dB of the expected $\mathbf{SNR}$.

Figure 8.2 is a plot of all 500 randomly generated targets $\mathbf{det}(\bar{\mathbf{C}})$ for the portion of the Monte Carlo simulation where the QPS was disabled. As expected, the $\mathbf{det}(\bar{\mathbf{C}})$ decays at almost an exponential rate for the first few CPIs. However, there seem to be continuous

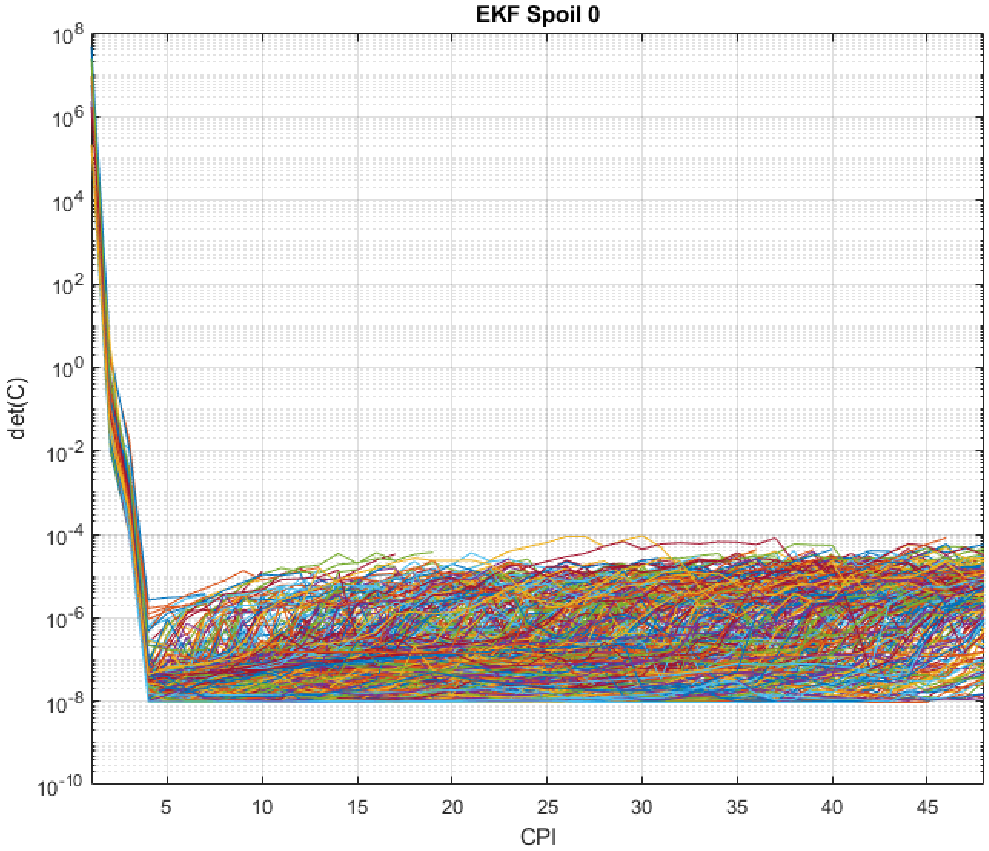


Figure 8.2: EKF Determinant Trends of 500 Randomly Generated Targets for the QPS Disabled

detections with significant errors leading to the constant divergence and sometimes convergence of $\mathbf{det(\bar{C})}$. The divergence is due to the radar's inability to steer the transmit and receive beams directly at the targets and the randomly generated maneuvers the targets perform on each iteration. Because of the inability of the radar to steer the beams to point at the target directly, the error between the prediction and a target's actual position continues to grow, causing the large spread in $\mathbf{det(\bar{C})}$ values between 10^{-8} and 10^{-4}. However, since the estimated variances of $\mathbf{R_k}$ are closer to the actual variance value in many cases, the floor of the EKF's $\mathbf{det(\bar{C})}$ is relatively low at 10^{-8}.

Figure 8.3 presents the targets $\mathbf{SNR}$ plotted for the 500 iterations of the Monte Carlo

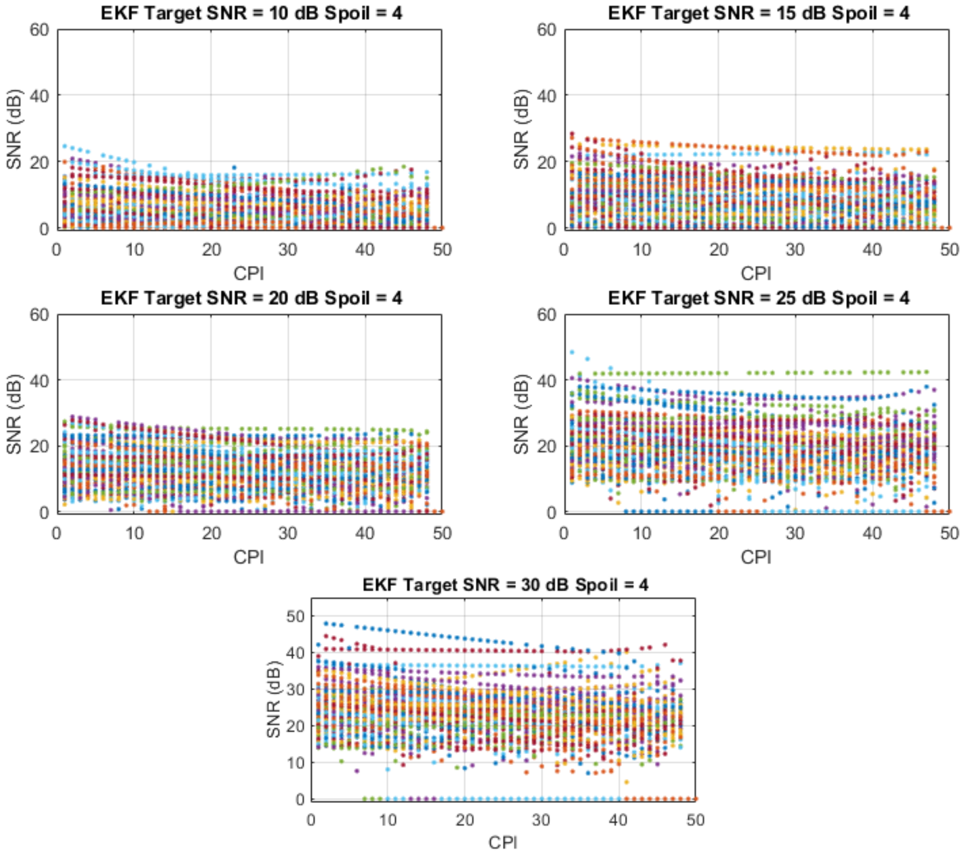


Figure 8.3: 100 Randomly Generated Targets $\mathbf{SNR}$ Tracked by the EKF With Spoiling Enabled by a Factor of 4

where the QPS spoiling factor was enabled for a spoiling factor of four. The $\mathbf{SNR}$ plots follow the same trends as the previous $\mathbf{SNR}$ plots in figure 8.2, where the overall floor of the randomly generated targets $\mathbf{SNR}$ rises with the increased RCS value. One difference is that it does not increase as much due to a decreased transmitted power density triggered by the QPS functionality.

Figure 8.4 represents the $\mathbf{det}(\bar{\mathbf{C}})$ of the 500 randomly generated targets the EKF tracked utilizing the automated QPS with a spoiling factor of four. The density of the $\mathbf{det}(\bar{\mathbf{C}})$ increases for the majority of the CPIs, the values of the $\mathbf{det}(\bar{\mathbf{C}})$ of the EKF straddles between

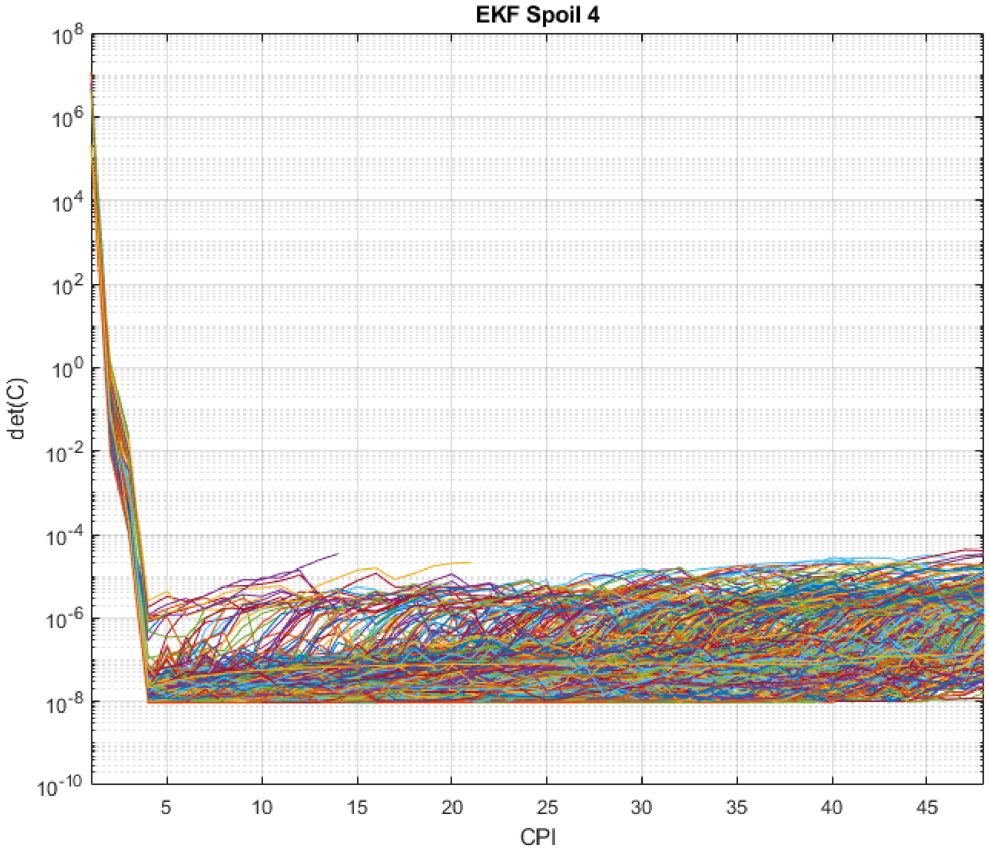


Figure 8.4: EKF Determinant Trends of 500 Randomly Generated Targets for a Spoil Factor of 4 Enabled

10^{-8} and 10^{-5}. The decrease in the ceiling of the $\mathbf{det(\bar{C})}$ values is due to the lower error between the target and the prediction. Again, since the estimated variance's of $\mathbf{R_k}$ are closer to the actual error value in many cases, the floor of the EKF's $\mathbf{det(\bar{C})}$ is relatively low at 10^{-8}.

Figure 8.5 is a plot of each of 100 randomly generated targets. The EKF tracked these targets with the automated QPS enabled and a default spoiling factor of eight. Continuing in the trends of the previous two $\mathbf{SNR}$ plots, the $\mathbf{SNR}$ of the targets trend towards increasing with a more significant RCS value. However, with the spoiling factor eight, the amount of $\mathbf{SNR}$ that increased per RCS value decreased due to the decreased power density of the

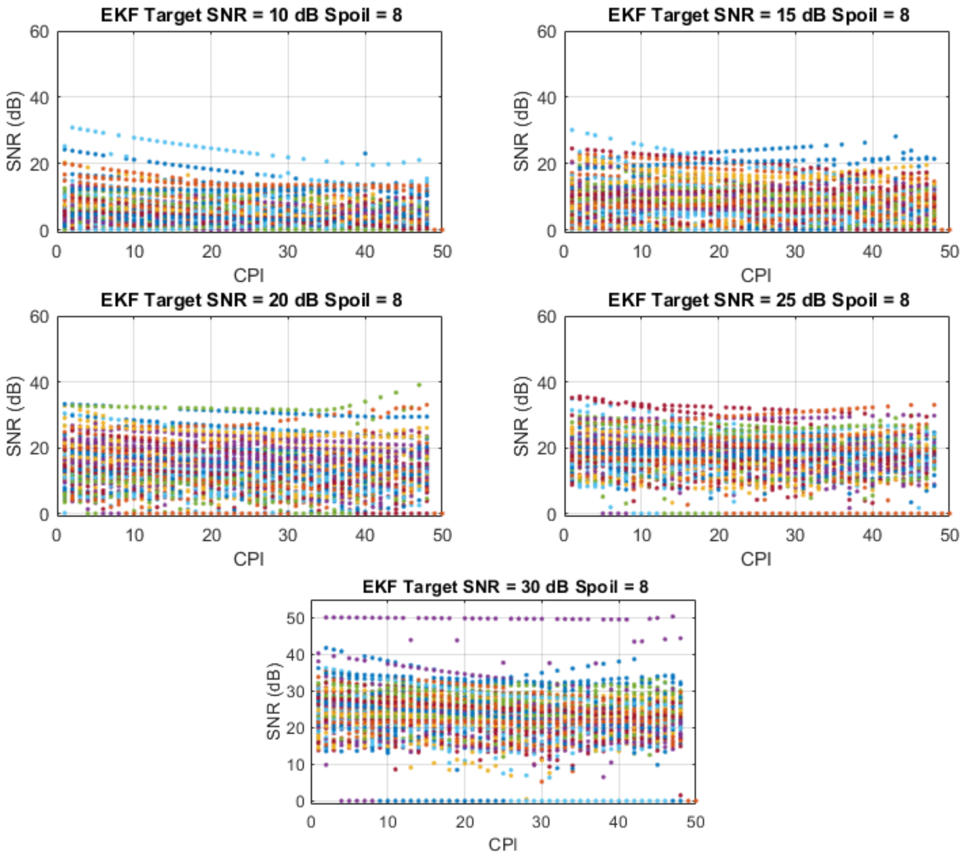


Figure 8.5: 100 Randomly Generated Targets SNR Tracked by the EKF With Spoiling Enabled by a Factor of 8

transmitted spoiled beam.

Figure 8.6 is a plot of the EKF's **det($\bar{\mathrm{C}}$)** with the automated QPS enabled with a default spoil factor of eight. Again the **det($\bar{\mathrm{C}}$)** of the 500 randomly generated targets decay quickly but have a large distribution of error due to the dependency in measurements with the predictions. However, similarly the **det($\bar{\mathrm{C}}$)** to figure 8.4, the concentration of **det($\bar{\mathrm{C}}$)** for figure 8.6 is denser and lower. However, due to the lower SNR created by, the lower power density on transmit because of the increased spoiling factor of eight, the **det($\bar{\mathrm{C}}$)** mimics the spread of values associated with figure 8.2.

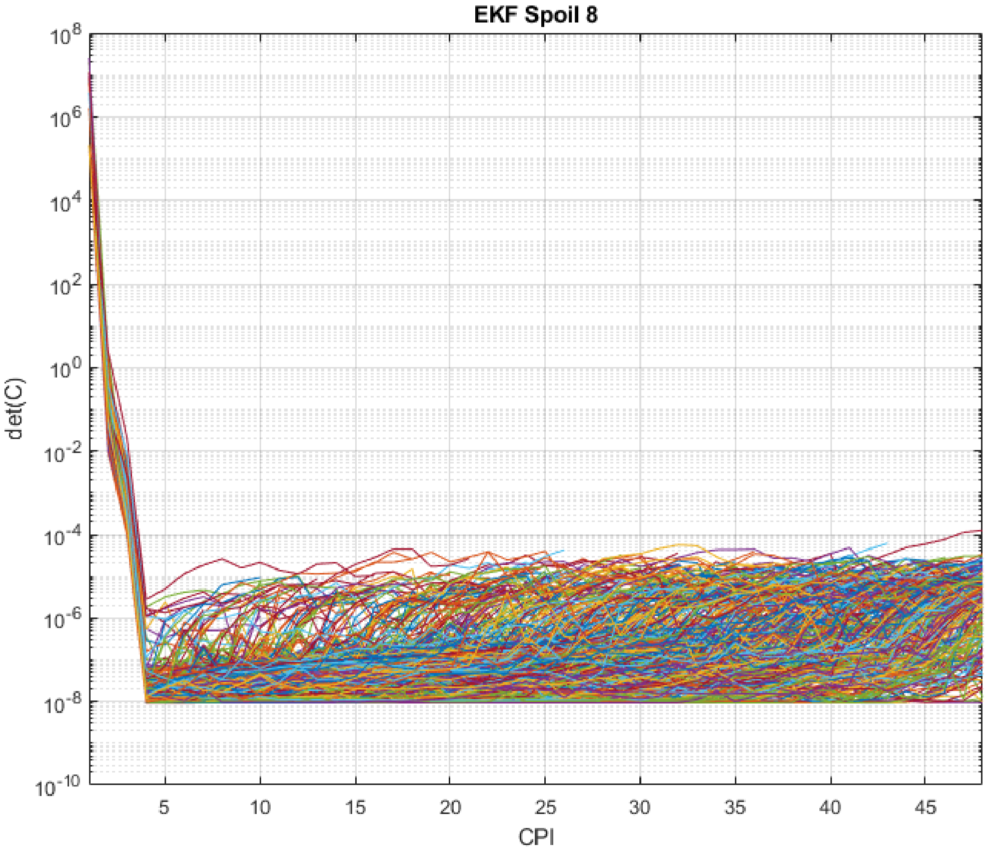


Figure 8.6: EKF Determinant Trends of 500 Randomly Generated Targets for a Spoil Factor of 8 Enabled

Figure 8.7 is the scatter plots of CPI's tracked with respect to the average $\mathbf{SNR}$ of the generated target. Each plot contains 500 data points, one for each of the 1500 randomly generated targets that the EKF tracked. Each plot is for each of the automated QPS cases, with the spoiling enabled for factors of eight and four and the automated QPS disabled. For the bottom plot, the automated QPS was disabled, resulting in a large distribution of the length of time a target is tracked with respect to the average $\mathbf{SNR}$ value. The middle plot represents the length of time the targets we tracked with the automated QPS enabled for a spoiling factor of four. The length of time a target is tracked appears to be trending towards a more extended period. Finally, in the top scatter plot, it is for a spoiling factor of

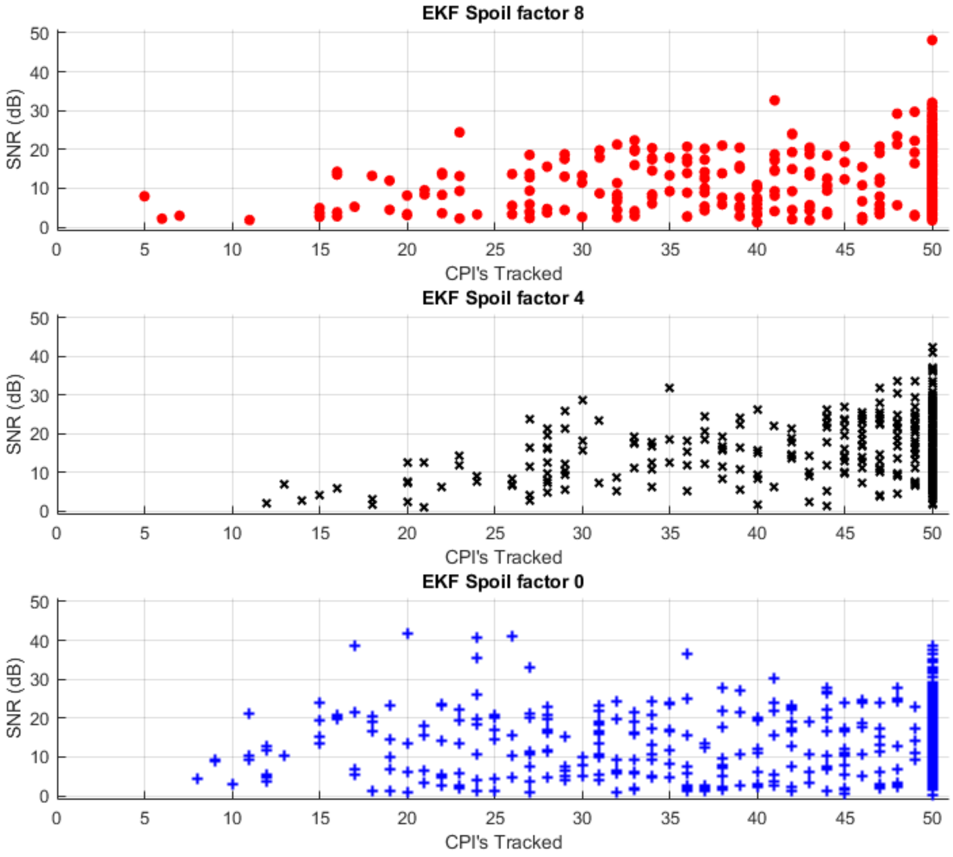


Figure 8.7: Scatter Plot of Number of CPI's Tracked per Monte Carlo For the EKF

eight. Similar to the plot with the automated QPS enabled for a spoiling factor of four, the top tracked CPI's with respect to average SNR appear to trend towards longer times than bottom scatter plot. However, points are not as dense as the plot that displays the tracked targets with the QPS enabled for a spoiling factor of four. The distribution in the length of time a target is tracked for the QPS enabled with a spoiling factor of eight is most likely due to the decreased SNR that is attributed to the increased QPS spoiling.

Figure 8.8 is the complement of the cumulative distribution function (CDF) of the scatter plots in figure 8.7. The compliment of a CDF represents a probability at a certain point. Figure 8.7 represents the probability that a target will be tracked for each of the

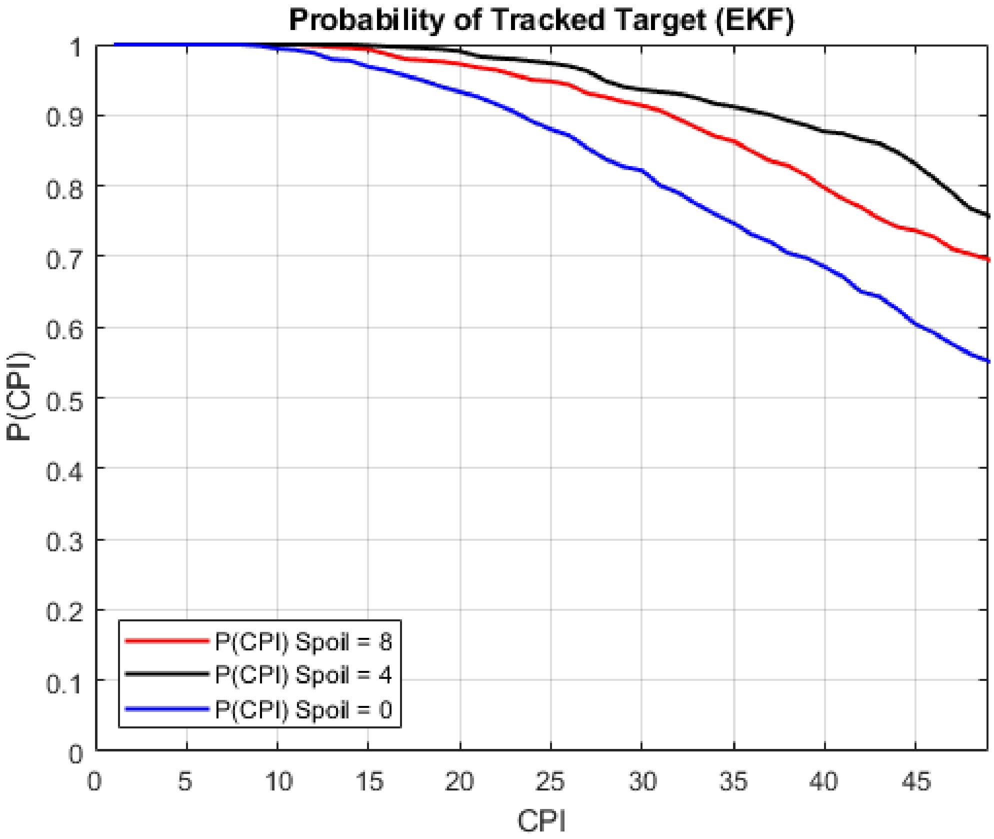


Figure 8.8: EKF Compliment of Cumulative Distribution Function (CDF) of Tracked CPI's

first 49th CPIs. The CDFs complement for the QPS disabled has the lowest probability of being tracked for an entire Monte Carlo iteration. As for the scenario for the EKF utilizing the automated QPS with a spoiling factor of four, the CDFs complement for this situation has the highest probability to be tracked ending above 0.75. The spoiling factor of four's probability indicates that this scenario had the most encouraging results. When the EKF utilized the spoiling factor of eight, the probability that a target would be tracked for a more extended time increased as opposed to when the automated QPS was disabled for the EKF.

8.2 UKF Performance

For this section, we will discuss the performance of the UKF under the same conditions as the EKF. An analysis of the UKF's performance will determine if the automated QPS impacted the UKF's ability to track a target. Following the same process as Section 8.1, we will begin discussing SNR and determinate plots of the targets and UKF, respectively. Next, the scatter plots of the CPI's tracked with respect to average SNR will be analyzed. Ending the section with a discussion on the CDF for each spoiling factor.

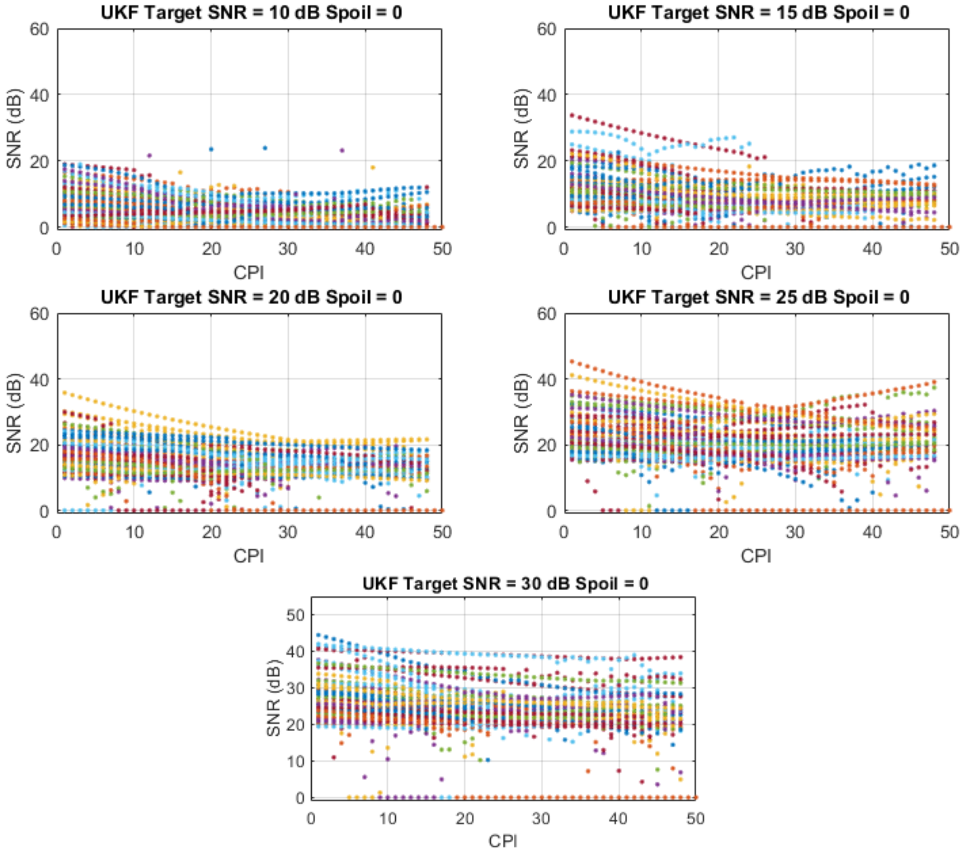


Figure 8.9: 100 Randomly Generated Targets SNR Tracked by the UKF With Spoiling Disabled

Similar to the EKF $\mathbf{SNR}$ plots in figure 8.1 from Section 8.1, the target $\mathbf{SNR}$ in figure 8.9 increase with each increased respective RCS value. One thing of note here is that the distribution's average ceiling is lower with the UKF than the EKF. The lower ceiling of the target's $\mathbf{SNR}$ tracked using the UKF is most likely due to an increase in lost targets due to the automated QPS disabled, decreasing the lost targets $\mathbf{SNR}$.

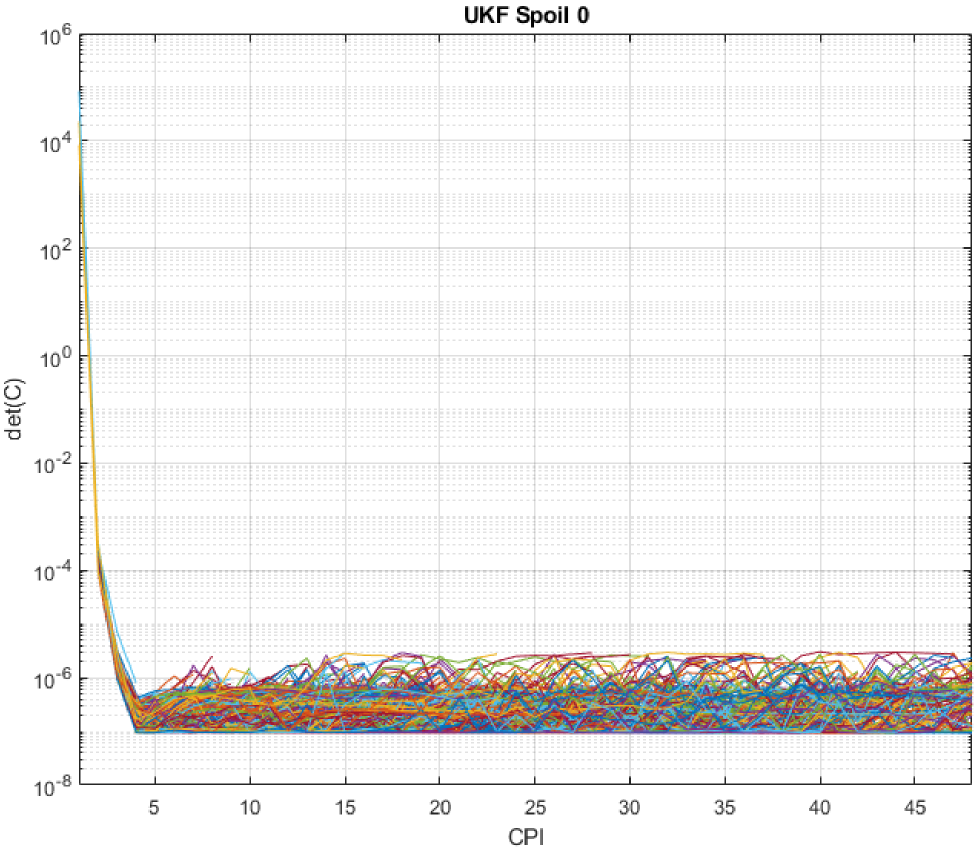


Figure 8.10: UKF Determinant Trends of 500 Randomly Generated Targets for the QPS Disabled

Figure 8.10 displays the $\mathbf{det(\bar{C})}$ for the 500 randomly generated targets in this scenario. Following a similar exponential decay trend as the EKF's $\mathbf{det(\bar{C})}$**'s**, the $\mathbf{det(\bar{C})}$ converges when the estimated error is approximately the updated estimated error. A few things of note, first, the minimum $\mathbf{det(\bar{C})}$ for the UKF increased by a factor of 10^2, indicating that the estimated variance of the UKF was not as accurate as the EKF's estimated variance. Secondly, the divergence of the $\mathbf{det(\bar{C})}$ after a measurement is lower than the EKF due to the distribution of the UKF's $\mathbf{det(\bar{C})}$ of the tracked targets. Finally, the density of $\mathbf{det(\bar{C})}$ decreases as the CPI's increase, which appears to be caused by the utilization of the UT converging a the target's.

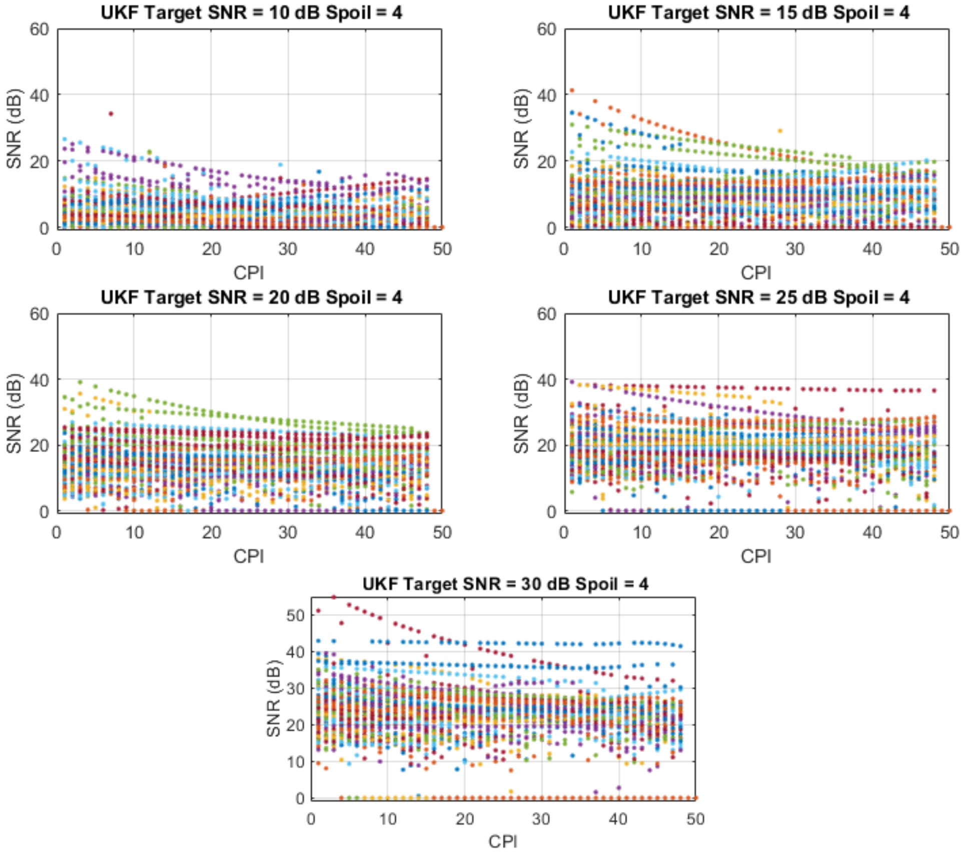


Figure 8.11: 100 Randomly Generated Targets $\mathbf{SNR}$ Tracked by the UKF With Spoiling Enabled by a Factor of four

For the 500 randomly generated targets that the UKF tracked using automated QPS with a spoiling factor of four. The targets $\mathbf{SNR}$ values are plotted with respect to CPI in figure 8.11. Interestingly, the distributions for each RCS value seem more spread out and abundant than in figure 8.9. The distribution of the targets $\mathbf{SNR}$ is most likely due to the increased detection. The floor of the $\mathbf{SNR}$ value is still lower due to the decreased $\mathbf{SNR}$ contribution of the transmit beam with the added automated QPS. The spoiling of the transmitted beam appears to cause a lower $\mathbf{SNR}$ due to the decreased power density on transmit that naturally occurs with a spoiled beam.

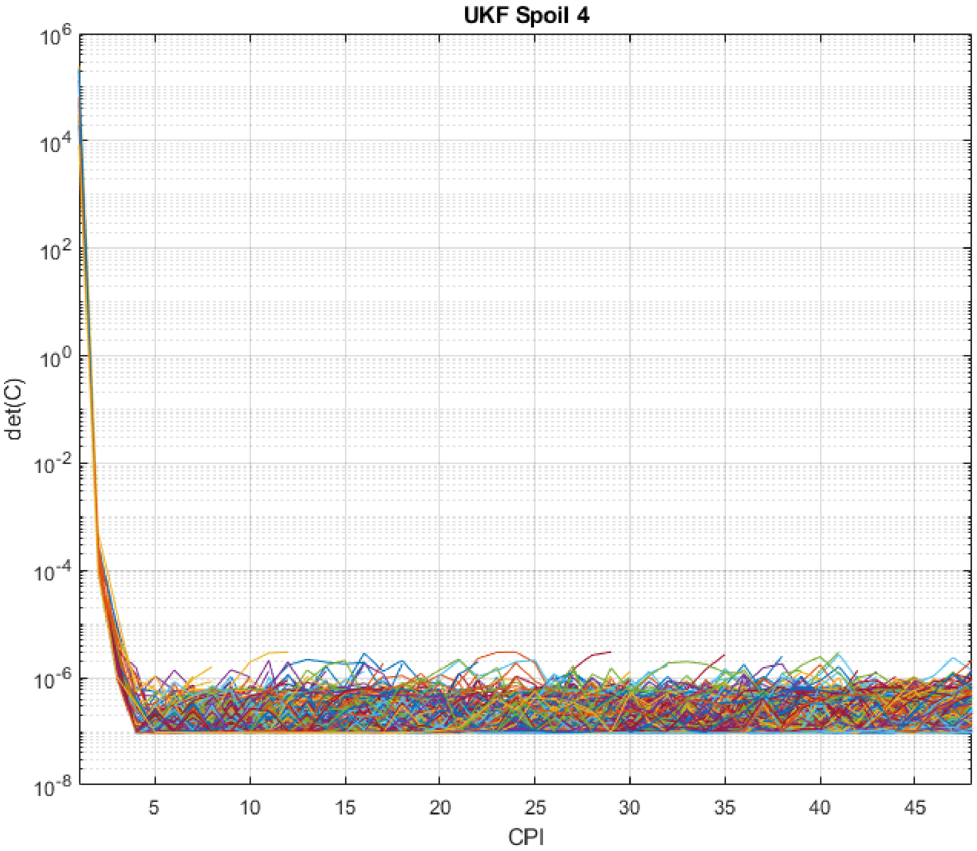


Figure 8.12: UKF Determinant Trends of 500 Randomly Generated Targets for a Spoil Factor of four Enabled

Figure 8.12 displays the **det($\bar{\mathrm{C}}$)** of the UKF that tracked the 500 randomly generated targets using the automated QPS with a spoiling factor of four. A few things of note, the first being that the **det($\bar{\mathrm{C}}$)** of the UKF with the automated QPS spoil factor four still has a higher minimum value by a factor of 10^2 than the EKF's **det($\bar{\mathrm{C}}$)**. Secondly, the concentration of the UKF **det(C̄)** is denser than the **det(C̄)** of the UKF with the automated QPS disabled, alluding to the targets tracked for a more significant time.

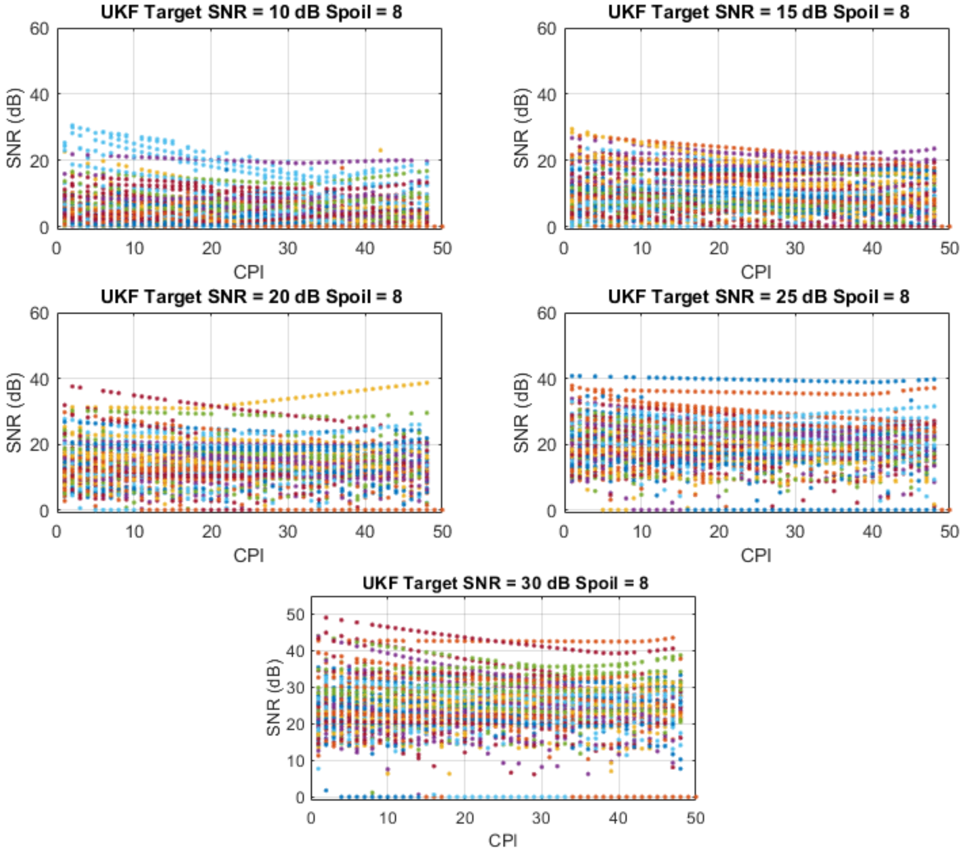


Figure 8.13: 100 Randomly Generated Targets SNR Tracked by the UKF With Spoiling Enabled by a Factor of eight

Figure 8.13 depicts SNR values of 500 randomly generated targets the UKF tracked with the automated QPS enabled with a spoiling factor of eight. Each SNR plot corresponds to its respective RCS value; the trend continues where the SNR floor increases with each increasing RCS value. Also, the density of the target SNR values is increased from figure 8.9, indicating that more targets were tracked for a more significant time with the QPS enabled as opposed to the QPS disabled. The rate at which the SNR floor rose decreased due to the decreased **SNR** contributions of the transmit beam when the QPS was enabled. This behavior is due to a more significant decrease in **SNR** contribution on transmission with a spoiling factor of eight instead of four.

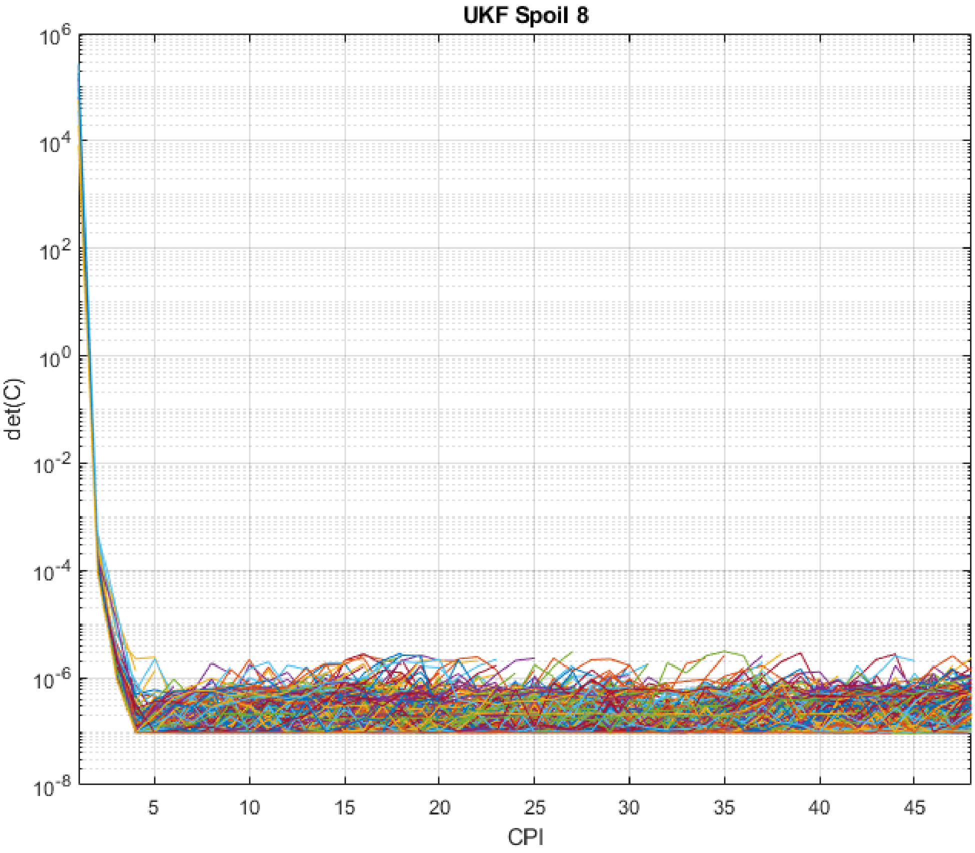


Figure 8.14: UKF Determinant Trends of 500 Randomly Generated Targets for a Spoil Factor of eight Enabled.

Figure 8.14 presents the $\mathbf{det}(\bar{\mathbf{C}})$ of the UKF with the automated QPS enabled with a spoiling factor of eight. Following the trend of the previous two $\mathbf{det}(\bar{\mathbf{C}})$ plots for the UKF, figure 8.14 shows that the UKF's $\mathbf{det}(\bar{\mathbf{C}})$ has a higher floor than the EKF. However, the concentration of $\mathbf{det}(\bar{\mathbf{C}})$ is more significant, leading to the assumption that the updated estimated error is more accurate than the EKF's updated estimated error. Finally, the spread of $\mathbf{det}(\bar{\mathbf{C}})$ for a QPS spoiling factor of eight is slightly larger than the $\mathbf{det}(\bar{\mathbf{C}})$ values when the QPS spoiling factor is four. The spread of $\mathbf{det}(\bar{\mathbf{C}})$ in figure 8.14 most likely means that there were fewer targets tracked for this Monte Carlo scenario as opposed to the previous scenario.

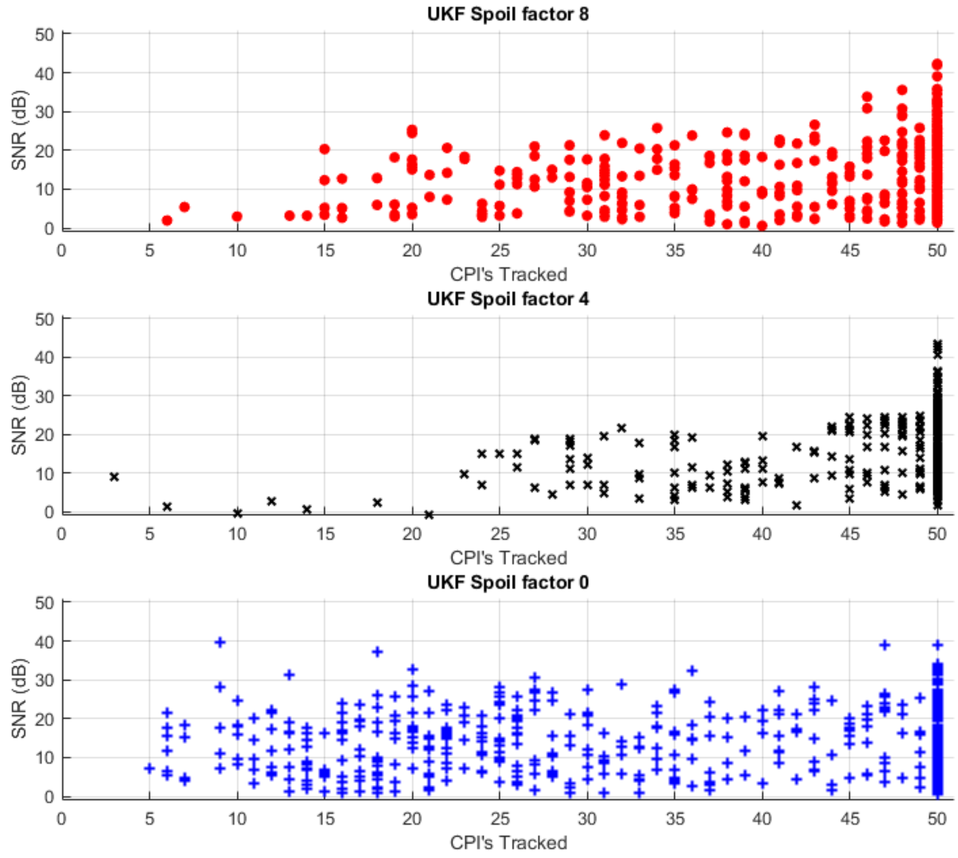


Figure 8.15: Scatter Plot of Number of CPI's Tracked per Monte Carlo For the UKF.

Figure 8.15 represents the number of CPIs a target was tracked for that Monte Carlo's iterations average $\mathbf{SNR}$. Each scatter plot represents 500 targets for each of the QPS scenarios. For the automated QPS disabled, the UKF's spread in CPI's tracked for $\mathbf{SNR}$, showing no trends except for a target being less likely tracked by the UKF without the automated QPS enabled. When the UKF utilized the automated QPS with a spoiling factor of four, there appears to be a correlation between the length of time a target is tracked and the average SNR. Similarly, when automated QPS is enabled with a spoiling factor of eight for the UKF, there appears to be a correlation between the length of time a target is tracked and the average SNR per Monte Carlo iteration. The correlation of the case when the QPS defaults to a spoiling factor of eight is not as concentrated as when the QPS defaults to a spoiling factor of four. The lower distribution of results is assumed to be due to the natural loss of SNR due to the lower power density of the transmit beam with a spoiling factor of eight instead of four.

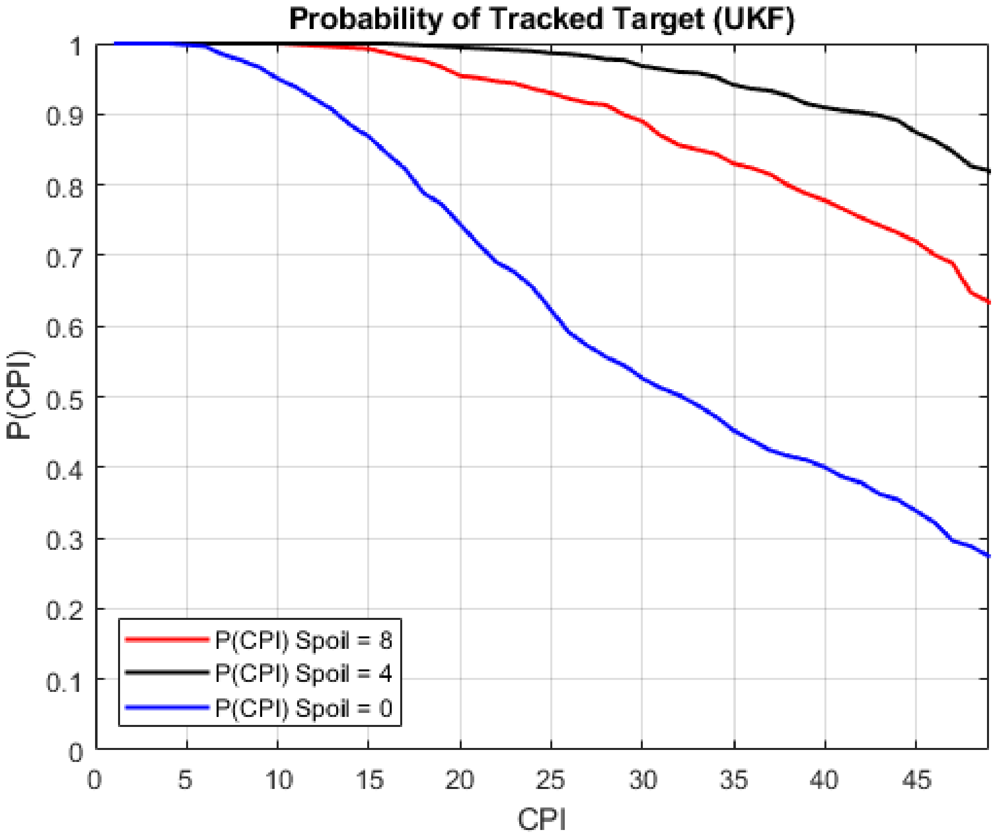


Figure 8.16: UKF Complement of the Cumulative Distribution Function (CDF) of Tracked CPI's.

Figure 8.16 is the complement of the CDF for the scatter plots in figure 8.15. The CDFs complement represents the probability that a target will be tracked for CPIs one through 49. When the QPS is disabled for the UKF, the probability of a target being tracked for all

CPIs is the lowest. The low probability indicates a higher likelihood of a lost target than any other scenario. When the UKF utilizes the automated QPS with a spoiling factor of four, the probability that a target would be tracked for the entirety of the iteration rose to above 0.80—indicating that the scenario of the UKF utilizing the spoiling factor of four produced the most promising results. When the UKF increased the spoiling factor to eight, the UKF's ability to track a target still increased from the scenario of the automated QPS being disabled. However, the scenario with a spoiling factor of four performed better than the scenario with a spoiling factor of eight. With this continued trend of the spoiling factor of four outperforming the spoiling factor of eight for both trackers, there appears to be a trend where the spoiling factor of four performed better due to the increase in SNR. The decrease in SNR is due to an increase in the spoiling factor. A decrease in SNR leads to a decrease in accurate detections. Therefore, the likelihood of a lost target increased with an increase in the spoiling factor from four to eight.

8.3 Quantitative Results

For the quantitative results, we will explore both the UKF and EKF how each tracker performed under each of the unique conditions, from the five standard RCS values given to the 30,000 randomly generated targets to the different QPS scenarios. Then determine the effectiveness of the automated QPS with the data presented.

For each case presented in table 8.1 there is a maximum of 5000 tracked CPIs. For the totals in table 8.1 there is a maximum of 25,000 tracked CPIs for each QPS scenario. The QPS scenario of "Spoil = 0" in table 8.1 corresponds to the automated QPS being disabled. From the data presented in table 8.1, it is clear that both the EKF and UKF performed better with the automated QPS enabled as opposed to it being disabled. Another observation of note is that it is clear that when the automated QPS was enabled, it performed negligibly

Cumulative Tracked CPI's Per Monte Carlo Iteration							
$\mathbf{SNR_{REF}}$ (dB)	**RCS**	**EKF**			**UKF**		
		$\mathbf{S_F}$ 8	**$\mathbf{S_F}$ 4**	**$\mathbf{S_F}$ 0**	**$\mathbf{S_F}$ 8**	**$\mathbf{S_F}$ 4**	**$\mathbf{S_F}$ 0**
10	0.21	4403	4435	4189	4102	4492	3289
15	0.67	4456	4402	4081	4340	4684	3048
20	2.21	4514	4644	4151	4352	4724	3285
25	6.72	4372	4634	4210	4113	4727	3378
30	20.78	4578	4764	3970	4601	4876	3493
	Totals	22323	22879	20601	21508	23503	16493

Table 8.1: Cumulative Tracked CPI's Per Monte Carlo Iteration

better for the EKF when the spoiling factor is four as opposed to eight. In contrast, the UKF performed noticeably better with the automated QPS spoiling factor set to four instead of eight. However, both trackers performed better for their respective Monte Carlo iterations using the automated QPS spoiling factor of four instead of the other two scenarios. The increased performance of the trackers appears to be due to that specific scenario taking advantage of the wider illuminated scene when needed and also not having the loss in $\mathbf{SNR}$ due to more significant spoiling factors such as a spoiling factor of eight.

Table 8.2 is the percentage of tracked CPIs over the entirety of the Monte Carlo simulation. There is no clear trend that correlates with the number of CPIs tracked due to the increase in the RCS value of the randomly generated target. However, the EKF performed better than the UKF, except when the automated QPS spoiled by a factor of four. The in-

Percent of Tracked Targets per Monte Carlo Scenario							
$\mathbf{SNR_{REF}}$ (dB)	**RCS**	**EKF**			**UKF**		
		$\mathbf{S_F}$ 8	**$\mathbf{S_F}$ 4**	**$\mathbf{S_F}$ 0**	**$\mathbf{S_F}$ 8**	**$\mathbf{S_F}$ 4**	**$\mathbf{S_F}$ 0**
10	0.21	88.06	88.7	83.78	82.04	89.84	65.78
15	0.67	89.12	88.04	81.62	86.8	93.68	60.96
20	2.21	90.28	92.88	83.02	87.04	94.48	65.7
25	6.72	87.44	92.68	84.2	82.26	94.54	67.56
30	20.78	91.56	95.28	79.4	92.02	97.52	69.86
	Totals	89.292	91.516	82.404	86.032	94.012	65.972

Table 8.2: Percent of Tracked Targets per Monte Carlo Scenario

crease in performance of the EKF, as opposed to the UKF, is most likely due to the initial setup of the estimated measurement variances for the UKF, which would explain why the **det(**$\bar{C}$**)** values for the UKF never decayed as far as the **det(**$\bar{C}$**)** values.

Table 8.3 displays the results of targets tracked for a complete Monte Carlo iteration. Each case has a maximum possible tracked Mote Carlo iteration of 100. Again, both trackers showed the best performance with the automated QPS enabled with a spoiling factor of four. Moreover, both trackers performed the worst with the automated QPS disabled. The UKF had the worst performance of the two trackers when the automated QPS disabled, only able to track 33.8% of the targets for a complete Monte Carlo iteration.

Table 8.4 contains the mean and median number of tracked CPIs for a target using the UKF for each condition, further solidifying that the automated QPS allowed the tracker to track a target for a more extended period. The UKF iterations averaged a slightly higher CPI track length when utilizing a spoiling factor of four instead of eight to increase the $\mathbf{SNR}$ potentially. Furthermore, there appears to be less discrepancy from the mean median CPIs tracked when the beamspoiling is enabled for a spoiling factor of 4.

Continuing the trend, table 8.5 holds the mean and median tracked CPIs for all conditions that the EKF tracked and performed better with automated QPS enabled instead of the alternative of disabling the automated QPS. Also, this further confirms that the EKF

Tracked Targets for 50 CPI's							
$\mathbf{SNR_{REF}}$ **(dB)**	**RCS**	**EKF**			**UKF**		
		$\mathbf{S_F}$ **8**	$\mathbf{S_F}$ **4**	$\mathbf{S_F}$ **0**	$\mathbf{S_F}$ **8**	$\mathbf{S_F}$ **4**	$\mathbf{S_F}$ **0**
10	0.21	76	74	53	56	73	31
15	0.67	69	70	55	68	82	31
20	2.21	67	78	59	62	80	32
25	6.72	65	73	60	57	85	32
30	20.78	71	84	49	74	90	43
	Totals	348	379	276	317	410	169

Table 8.3: Total Targets Tracked for a Full Monte Carlo Iteration

UKF Mean and Median CPI's Tracked							
SNR_{REF} (dB)	RCS	S_F 8		S_F 4		S_F 0	
		Mean	Median	Mean	Median	Mean	Median
10	0.21	41	50	45	50	33	32
15	0.67	44	50	47	50	31	27
20	2.21	44	50	47	50	33	31
25	6.72	41	50	48	50	34	36
30	20.78	47	50	49	50	35	40

Table 8.4: UKF Average and Median Tracked CPI's for Each Condition

EKF Mean and Median CPI's Tracked							
SNR_{REF} (dB)	RCS	S_F 8		S_F 4		S_F 0	
		Mean	Median	Mean	Median	Mean	Median
10	0.21	44	50	45	50	42	50
15	0.67	45	50	44	50	41	50
20	2.21	45	50	47	50	42	50
25	6.72	44	50	47	50	42	50
30	20.78	46	50	48	50	40	45

Table 8.5: EKF Average and Median Tracked CPI's for Each Condition

performed slightly better overall than the UKF.

8.4 Summary

To summarize the proceeding sections, the automated QPS increased the EKF and UKF's ability to track a target for a more extended time. By decreasing the original spoil factor of eight in half, the trackers improved even more due to the increased SNR.

Chapter 9

Conclusion and Future Work

This thesis has detailed the versatility of a Phased-Array radar and its ability to electronically manipulate a beam to illuminate a wider scene by utilizing the digitized phase amplitude weights to spoil a beam. The QPS allows the proper weights to be selected to spoil a beam accurately. With the QPS and the radar signal processing techniques outlined in this thesis, we were able to build a simulation that accurately mimics the real-world environment in which a phased array will operate. The simulation helped to verify the improved tracker performance when utilizing the automated QPS. With the automated QPS with a spoiling factor of four, the UKF preformed 28.04% better than when the QPS was disabled. When the EKF utilized the automated QPS with spoiling factor of four, the EKF preformed 9.112% better, as opposed to the EKF not utilizing the automated QPS. The automated QPS laid the foundation for optimizing target tracking with a phased-array radar allowing a spoiled beam only when needed to avoid unnecessary processing time and energy of the multiple receive beams created from the QPS functionality.

The addition of the UKF and EKF allowed for the verification of the automated QPS functionality for better performance in tracking through the Monte Carlo process. This research is just a building block for testing the abilities of Digital phased-array radar systems and automated energy distribution. This research allows for more opportunities to discover more complex functionality of Digital phased-array radars.

9.1 Future Work

The first area would be, adding Mulitple Input Multiple Output (MIMO) concepts the the phased-array radar simulation. MIMO concepts allow phase-array radar by adding the ability simultaneously transmit and receive multiple waveforms at a time [41]. The addition of the MIMO capabilities adds the concept of frequency shifting and a frequency mode to the radar's ability. Frequency shifting allows for the LFM signal to be transmitted multiple times simultaneously by shifting the carrier frequency to multiple sub-carriers, each having its sub-bandwidth within the initially allotted bandwidth of the radar [42]. The use of unique waveforms can be researched even further by combining the frequency shifting of the frequency mode with other waveforms, such as the PSRB, avoiding the complications of non-orthogonal waveforms.

The introduction of multiple waveforms allows for multiple target tracking using the tracking algorithms like the ones used in this thesis. Because of this, new concepts need to be introduced, such as greedy nearest neighbor. With the multiple predictions made with a single tracker, greedy nearest neighbor assigns the detections with the nearest prediction made by the tracker [43]. The greedy nearest neighbor algorithm's introduction allows a MIMO radar to track multiple targets while lowering the probability of each being mismatched with the other [43]. A more optimal correlation algorithm to explore would be the Hungarian algorithm. Instead of relying on a minor error between the track and prediction like the greedy nearest neighbor algorithm, the Hungarian relies on a correlation matrix to decrease the likelihood of mismatching tracks [44]. Because of the addition of the correlation matrix, the Hungarian is more computationally intensive than the Greedy Nearest Neighbor algorithm [44].

Moving away from the MIMO concepts, one limitation of this simulation was the error of the CA-CFAR. While the CA-CFAR was less computationally intensive than others, the CA-CFAR relies on isolated targets and works under the assumption of homogeneous interference [4]. When the clutter overcomes the thermal noise of a range-Doppler map, the interference becomes heterogeneous, increasing the probability of a lost target [4]. With that, it would be beneficial to explore other CFAR algorithms, such as an OS-CFAR. A drawback to the OS-CFAR is that it is computationally intensive. With the CA-CFAR a normalized mean window can be convolved with the entire range-Doppler map, and the CFAR test is performed with the entire map at once. There is no current shortcut to the OS-CFAR. The OS-CFAR must slide its OS-CFAR window one cell at a time, sort the cells and grab the kth statistic to perform the ordered statistic test. This process causes the OS-CFAR to be much slower than the CA-CFAR but also more accurate.

The final improvement would be the ability to spoil a transmit beam at different spoiling factors during an iteration. The rate at which a target's angle changes is dependent on the range of a target. When a target performs a maneuver closer to the radar, there are more significant changes in the angle, causing the possible need for a wider spoiled beam. While if a target performs a maneuver at a greater distance, a sizeable spoiling factor is not needed. With these conditions, an optimized automated QPS is an ideal research topic.

References

[1] E. Wan and R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.

[2] M. A. Zatman, “Digitization requirements for digital radar arrays,” *Proceedings of the 2001 IEEE Radar Conference (Cat. No.01CH37200)*, pp. 163–168, 2001.

[3] D. C. Gaydos, P. Nayeri, and R. L. Haupt, “Adaptive beamforming with software-defined-radio arrays,” *IEEE Access*, vol. 10, pp. 11 669–11 678, 2022.

[4] M. A. Richards, *Fundamentals of Radar Signal Processing*, 2nd ed. McGraw-Hill Education, 2014.

[5] P. Rohal and J. Ochodnicky, “Radar target tracking by kalman and particle filter,” in *2017 Communication and Information Technologies (KIT)*, 2017, pp. 1–4.

[6] T. Jeffrey, *Phased-Array Radar Design: Application of Radar Fundamentals*, ser. Electromagnetics and Radar. Institution of Engineering and Technology, 2009. [Online]. Available: https://books.google.com/books?id=UUMhCFWl5hEC

[7] M. Matsuki, Y. Yokota, M. Shibata, A. Karasawa, H. Fujihara, S. Nakamura, Y. Kankaku, T. Motohka, and S. Suzuki, “Phase spoiling technique for high power and wide beam in alos-4,” in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 8326–8327.

[8] R. Kinsey, “Phased array beam spoiling technique,” in *IEEE Antennas and Propagation Society International Symposium 1997. Digest*, vol. 2. IEEE, 1997, pp. 698–701.

[9] J. Gaffney, G. Guttrich, B. Babu, and J. Torres, “Performance comparison of fast-scan gmti/stap architectures,” in *Proceedings of the 2001 IEEE Radar Conference (Cat. No.01CH37200)*, 2001, pp. 252–257.

[10] J. Yan, J. Dai, W. Pu, H. Liu, and M. Greco, “Target capacity based resource optimization for multiple target tracking in radar network,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 2410–2421, 2021.

[11] I. Ullah, Y. Shen, X. Su, C. Esposito, and C. Choi, “A localization based on unscented kalman filter and particle filter localization algorithms,” *IEEE Access*, vol. 8, pp. 2233–2246, 2020.

[12] O. Aydogmus and M. F. Talu, “Comparison of extended-kalman- and particle-filter-based sensorless speed control,” *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 2, pp. 402–410, 2012.

[13] S. Pate, “Predictive tracking simulation and techniques for all-digital radar,” Master’s thesis, University of Oklahoma, 2021.

[14] W. R. Blanding, W. Koch, and U. Nickel, “Adaptive phased-array tracking in ecm using negative information,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 1, pp. 152–166, 2009.

[15] M. Yeary, R. Palmer, C. Fulton, J. Salazar, and H. Sigmarsson, “Recent advances on an s-band all-digital mobile phased array radar,” in *2019 IEEE International Symposium on Phased Array System and Technology (PAST)*, 2019, pp. 1–5.

[16] R. A. Monzingo and T. W. Miller, *Introduction to adaptive arrays*. Scitech publishing, 2004.

[17] S. H. Talisa, K. W. O’Haver, T. M. Comberiate, M. D. Sharp, and O. F. Somerlock, “Benefits of digital phased array radars,” *Proceedings of the IEEE*, vol. 104, no. 3, pp. 530–543, 2016.

[18] G. H. B. C. Stimson, G. and D. Adamy, *Introduction to Airborne Radar*, 2nd ed. SciTech Publishing, 1998.

[19] J. Guo, S. Yang, S.-W. Qu, J. Hu, and Z. Nie, “A study on linear frequency modulation signal transmission by 4-d antenna arrays,” *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 12, pp. 5409–5416, 2015.

[20] A. Davies, “Properties of waveforms obtained by nonrecursive digital filtering of pseudorandom binary sequences,” *IEEE Transactions on Computers*, vol. C-20, no. 3, pp. 270–281, 1971.

[21] D. F. Albanese and A. M. Klein, “Pseudorandom code waveform design for cw radar,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-15, pp. 67–75, 1979.

[22] C. A. Balanis, *Antenna theory: analysis and design*. John wiley & sons, 2015.

[23] M. C. Leifer, “An airborne radar phased array with low sidelobes and a spoiled beam,” *2018 IEEE Radar Conference (RadarConf18)*, pp. 0072–0076, 2018.

[24] ——, “Revisiting a method of beam shaping using phase weights,” in *2016 IEEE International Symposium on Phased Array Systems and Technology (PAST)*. IEEE, 2016, pp. 1–3.

[25] N. O’Donoughue, *Emitter detection and geolocation for electronic warfare*. Artech House, 2019.

[26] S. Sherman and D. Barton, *Monopulse Principles and Techniques*, ser. Artech House radar library. Artech House, 2011. [Online]. Available: https://books.google.com/books?id=I1vksFcnQngC

[27] M. I. Skolnik, “Fifty years of radar,” *Proceedings of the IEEE*, vol. 73, pp. 182–197, 1985.

[28] S. Mitra and S. Mitra, *Digital Signal Processing: A Computer-based Approach*, ser. Connect, learn, succeed. McGraw-Hill, 2011. [Online]. Available: https://books.google.com/books?id=QyuNQgAACAAJ

[29] M. Skolnik, *Introduction to Radar Systems*, ser. Electrical engineering series. McGraw-Hill, 2001. [Online]. Available: https://books.google.com/books?id=Y6-APwAACAAJ

[30] P. Peebles, *Radar Principles*. Wiley India Pvt. Limited, 2007. [Online]. Available: https://books.google.com/books?id=rnX21aAMKCIC

[31] H. Rohling, “Radar cfar thresholding in clutter and multiple target situations,” *IEEE transactions on aerospace and electronic systems*, no. 4, pp. 608–621, 1983.

[32] R. Mitchell and J. Walker, “Recursive methods for computing detection probabilities,” *IEEE Transactions on Aerospace and Electronic Systems*, no. 4, pp. 671–676, 1971.

[33] M. S. Grewal and A. P. Andrews, *Linear Optimal Filters and Predictors*, 2008, pp. 131–181.

[34] S. Challa, M. Morelande, D. Mušicki, and R. Evans, *Fundamentals of Object Tracking*, ser. Cambridge books online. Cambridge University Press, 2011. [Online]. Available: https://books.google.com/books?id=DOKJGnNh9HgC

[35] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[36] H. Khazraj, F. Faria da Silva, and C. L. Bak, “A performance comparison between extended kalman filter and unscented kalman filter in power system dynamic state estimation,” in *2016 51st International Universities Power Engineering Conference (UPEC)*, 2016, pp. 1–6.

[37] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[38] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[39] B. Friedland, "Optimum steady-state position and velocity estimation using noisy sampled position data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-9, no. 6, pp. 906–911, 1973.

[40] S. Raychaudhuri, "Introduction to monte carlo simulation," in *2008 Winter Simulation Conference*, 2008, pp. 91–100.

[41] F. Robey, S. Coutts, D. Weikle, J. McHarg, and K. Cuomo, "Mimo radar theory and experimental results," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1, 2004, pp. 300–304 Vol.1.

[42] R. Feger, C. Pfeffer, and A. Stelzer, "A frequency-division mimo fmcw radar system based on delta–sigma modulated transmitters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 12, pp. 3572–3581, 2014.

[43] X. Lyu and J. Wang, "Sequential multi-sensor jpda for target tracking in passive multistatic radar with range and doppler measurements," *IEEE Access*, vol. 7, pp. 34 488–34 498, 2019.

[44] B. Sahbani and W. Adiprawita, "Kalman filter and iterative-hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system," in *2016 6th International Conference on System Engineering and Technology (ICSET)*, 2016, pp. 109–115.