NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

KALMAN FILTER TRACKING RADAR SYSTEM WITH AN ADAPTIVE SEARCH SPACE RESOLUTION TECHNIQUE

by

Georgios Souchlas

December 2022

Thesis Advisor: Ric Romero
Co-Advisor: Roberto Cristi

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2022	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
KALMAN FILTER TRACKING RADAR SYSTEM WITH AN ADAPTIVE SEARCH SPACE RESOLUTION TECHNIQUE

5. FUNDING NUMBERS

6. AUTHOR(S) Georgios Souchlas

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

This thesis develops a radar system to improve tracking by avoiding loss of target track when signal-to-noise ratio is low or when the target leaves the main beam of the antenna. Target tracking is performed by a Kalman tracker with two modes. The default mode uses a significant number of phased array elements, thereby forming a narrow beam for target search, which is referred to as high resolution (HR) mode. If the target leaves the main beam, the radar system enlarges the beam and thus can re-capture the target. This is accomplished by lowering the number of antenna elements, which is referred to as the low-resolution (LR) or wide search mode. To maintain SNR, the radar is assumed to increase the transmit output power. Once the target is recaptured, the radar switches back to high-resolution mode. Thus, this radar is an integrated beamsteering and tracking system.

14. SUBJECT TERMS Kalman filter, tracking techniques, cognitive radar	**15. NUMBER OF PAGES** 69
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

KALMAN FILTER TRACKING RADAR SYSTEM WITH AN ADAPTIVE SEARCH SPACE RESOLUTION TECHNIQUE

Georgios Souchlas
Plotarhis, Hellenic Navy
BNS, Hellenic Naval Academy, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2022

Approved by: Ric Romero
Advisor

Roberto Cristi
Co-Advisor

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis develops a radar system to improve tracking by avoiding loss of target track when signal-to-noise ratio is low or when the target leaves the main beam of the antenna. Target tracking is performed by a Kalman tracker with two modes. The default mode uses a significant number of phased array elements, thereby forming a narrow beam for target search, which is referred to as high resolution (HR) mode. If the target leaves the main beam, the radar system enlarges the beam and thus can re-capture the target. This is accomplished by lowering the number of antenna elements, which is referred to as the low-resolution (LR) or wide search mode. To maintain SNR, the radar is assumed to increase the transmit output power. Once the target is recaptured, the radar switches back to high-resolution mode. Thus, this radar is an integrated beamsteering and tracking system.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AESA	active electronically scanned array
AWGN	additive white Gaussian noise
EW	electronic warfare
HR	high resolution
KF	Kalman filter
LR	low resolution
MMSE	minimum mean square error
NPS	Naval Postgraduate School
OpIntel	operational intelligence
PRF	pulse repetition frequency
RCS	radar cross section
SNR	signal-to-noise ratio
TWS	track while scan
URA	uniform rectangular array

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

First of all, I would like to thank the Hellenic Navy for giving me the opportunity to study at the Naval Postgraduate School.

I would like to express my gratitude to my thesis advisor, Professor Ric R. Romero and my co-advisor, Professor Roberto Cristi, for their insight, guidance and patience over the course of my research. It is only with their valuable help and advice that I was able to successfully complete this thesis. I also have to commend their efforts on ensuring the quality of the final product.

Also, I would like to thank my family, whom I missed these two years of study, and even though far away, were really supportive of my efforts at the Naval Postgraduate School.

Last but not least, I would like to express my deepest appreciation and love to my wife, Anastasia Galanou, who gives meaning to everything I do, and her support for me is insurmountable.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Motivation

In the modern era, technology is advancing at unprecedented rates. A main driving force of this advance is military applications, which are in a constant battle of cyclical innovation between offense and defense across all battlefields and operational environments. More specifically, the advancement of aircraft speeds, as well as ever increasing ranges of ballistic or cruise missiles, has ushered a significant shift in priorities of modern warfare, gravitating heavily to electronic warfare (EW).

EW pertains to the use the electromagnetic spectrum and signals transmitted therein, for the purposes of sensing, protection, and communication. The definition also includes the denial of use of said spectrum against an adversary [1]. One tool that has been developed and is ubiquitous as the main sensor of the electromagnetic spectrum is radar (radio detection and ranging). As the acronym itself states, radar uses an antenna to emit electromagnetic pulses, which are reflected off of a target of interest and then received by the same or a different antenna, thus providing the information if a target exists, as well as the distance to that target.

Tremendous progress had been made in improving and fine tuning radars, depending on the specific application. The scope of this thesis is integrated radar tracking and scene surveillance. One of the most significant milestones in radar development is the phased array antenna for sensing. Traditional radar antennas have various reflector shapes (circular, paraboloid, horn, etc.), where the main purpose is to focus the majority of the transmitted energy into a particular direction. The main beam direction is uniquely and geometrically defined for every antenna and is usually fixed relative to the antenna body. In order to focus the energy towards a specific point of interest, the whole antenna body needs to be mechanically rotated or elevated so that the main axis of the antenna coincided with the direction of interest.

This has been the case for quite some time. However, demanding military applications have found the mechanical steering of traditional radar to be a limiting factor in terms of robustness, speed, accuracy and response time. Thus the phased array radar was introduced, which dealt with this problem by taking advantage of the size reduction in microelectronics as well as digital signal processing techniques [2]. Hence, phased array radars now exist in numerous applications, both military and civilian. Two such examples are shown in the following figures: a meteorological phased array radar in Figure 1.1, and a search and track military radar installed on a U.S. Navy ship in Figure 1.2.

Figure 1.1. Meteorological phased array radar for weather observation. Source: [3].

Figure 1.2. U.S. Navy Ticonderoga class guided-missile cruiser equipped with phased array radar system. Source: [4].

Active phased array radars consist of an array of individual transmit/receive (TX/RX) elements positioned in a specific pattern. The elements are connected together via a series of phase shifters. By manipulating each individual phase shift introduced to the original generated waveform, a selected beam pattern is formed. This pertains to the magnitude of the emitted electromagnetic energy in all directions as it leaves the antenna array [2]. The direction in which most of the energy is transmitted is called the main lobe or main beam. As such, that main beam can be directed on command (with certain limitations) by inducing the appropriate phase shifts into the individual elements. The process described is called beamsteering [5], and a diagram illustrating its function is provided in Figure 1.3.

In tracking applications, the main objective is pinpointing a target position and kinematics. Thus, the more accurate the measurements are, the better are the estimated position and velocity of the target being tracked. To this end, a higher received signal power can be achieved by a more directed antenna beam. This is called a "narrow" or "pencil" main beam/lobe, which is desirable to increase received signal power. This can be achieved by the number of elements and appropriate spacing of said array elements with regards to the wavelength of the transmitted pulse [6].

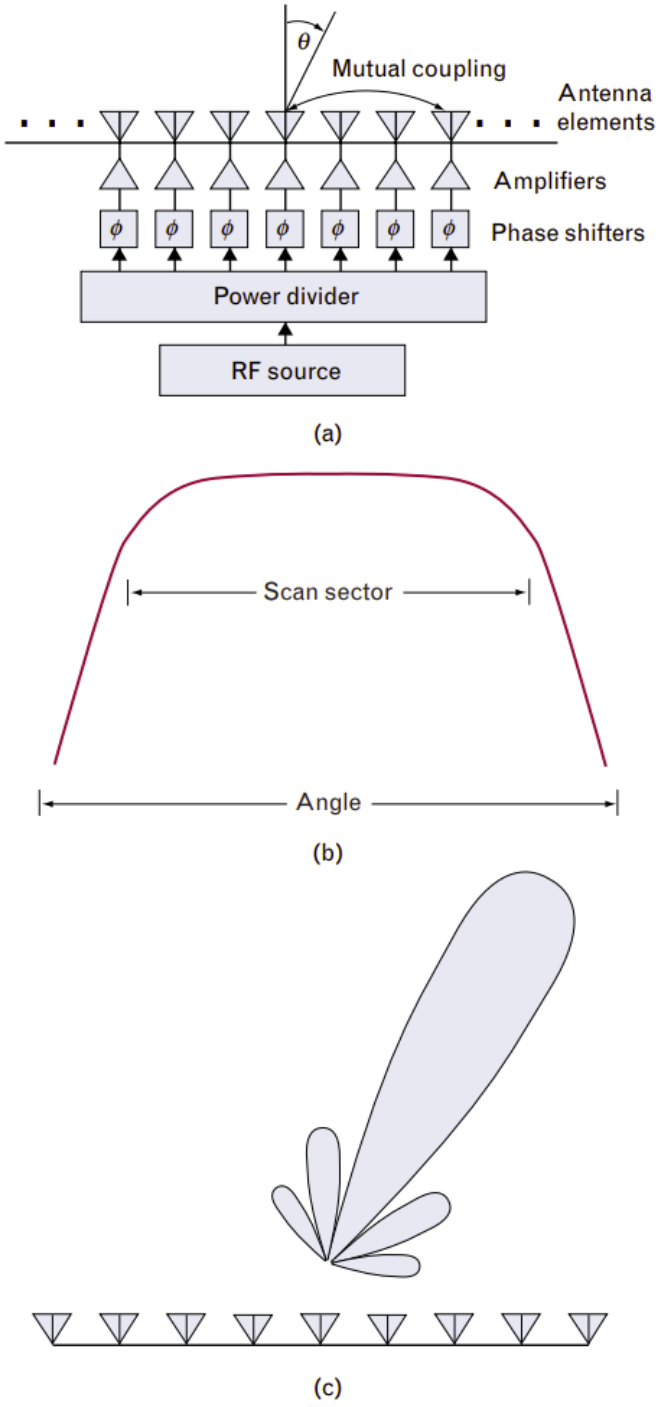

Figure 1.3. Beam forming in a phased array. Source: [5].

The aforementioned technology coincides conveniently with the other pillar this thesis is based on, which is the linear quadratic estimator, which is better known as Kalman filter. It was developed in the form that it has today around 1960 [2]. It is an algorithm based on a mathematical model that was created by Rudolph E. Kálmán. The algorithm takes sequential measurements over time as inputs, which are generally corrupted by noise or other unwanted interference, and produces estimates that are more accurate than if they were based on single measurements alone. This is achieved by evaluation of a joint probability distribution over the variables for each time frame [7], [8]. The operation of the Kalman filter will be explained extensively in Chapter 2.

This thesis proposes a combination of the two technologies mentioned above, along with algorithms simulated in a Matlab environment. The purpose is to improve the quality of tracking of a single target, while at the same time maintaining awareness of the whole scene of operation. This is performed in an adaptive manner, or on a need-to-do basis, while keeping as a main goal the efficient allocation of resources (be those time, spatial awareness or processing power).

While both aspects have been in widespread use for more than five decades [2], [7]–[10], a few practical issues need to be addressed in the application being considered in this thesis. In the simulation environment, time is considered discrete, i.e., an ongoing chain of time frames or time-steps. In practice however, those time frames may have to concur with the radar pulse repetition frequency (PRF), as it is this frequency that determines how often a measurement is received as an input to the algorithm. As such, the necessary calculations the Kalman filter and the rest of the proposed algorithm have to be executed between two consecutive radar pulses, such that the estimate for every time frame is produced as needed. Even though the set of algorithms proposed in this current form is not extremely complicated, it can be used as a basic form upon which other more complex ones can be developed. This, of course, calls for even more computing power, which in turn implies more cost. (The recent advancements in quantum computing might in the near future render the latter problem obsolete; however this is outside the scope of this thesis.)

1.2 Thesis Outline

Chapter 2 provides the theory and mathematical models designed to support the simulations. The necessary equations are presented for both linear Kalman filter estimation and uniform rectangular array beamsteering. Chapter 3 explains the problem this proposal attempts to tackle, and explains the setup of the simulations in detail. Simulations are subsequently performed, and their outcomes are presented in Chapter 4. Finally, Chapter 5 contains conclusions drawn as well as suggestions for further improvements that can be made in future work.

CHAPTER 2:
Background

2.1 Kalman Filter Tracking

Since the invention of radar, efforts have been and are constantly being made to improve its performance characteristics. Spatial resolution is one of the most important ones, which refers to how close can two objects be resolved as separate targets. Equally important is the accurate representation of the target trajectory. In numerous military or civilian applications, tracking the kinematic behavior of targets moving in the operating environment has always been an important goal. For this reason, some tracking techniques/algorithms that make use of mathematical models have been developed. The most prevalent of these is linear quadratic estimation, otherwise known as Kalman filter (KF) [7]–[10].

2.1.1 Working Principle

KF as mentioned is an estimator. It makes use of all the measurements or sensing information about the target position from the past up until the present [11], [12]. It does not rely on a single measurement to provide a reliable estimate as to where the target is located. KF prevails as a tracking technique because it is designed and proven to be the best possible linear estimator, in the sense that it always provides the estimate with the minimum possible mean-square error [7]–[10].

KF can accurately track movement by using a state-space model which includes random disturbances and model uncertainties. To make effective use of it, the motion of the target has to be represented in a dynamic model. While the dynamic model selected will never be a 100% accurate representation of the actual target kinematics, the constant velocity model is the most prevalent model used, mainly due to its simplicity and adaptability. This model assumes the velocity of the target between each time step remains constant.

The main problem addressed by this research is to track a target position and predict its trajectory in the near future, while also adaptively maintaining the capability of momentary or continuous surveillance of the whole visible region. The technique used is based on a

KF which combines observations from a sensor (in this case a radar) with a dynamic model of the target motion. Beamsteering techniques are implemented to beneficially exploit the phased array's capabilities. The particular model is a stochastic state space model of the form described in the next section.

2.1.2 Stohastic State Space Model Equations

Any linear system, like the one being discussed, can be modeled in state space with the schematic seen in Figure 2.1 and and described with (2.1).

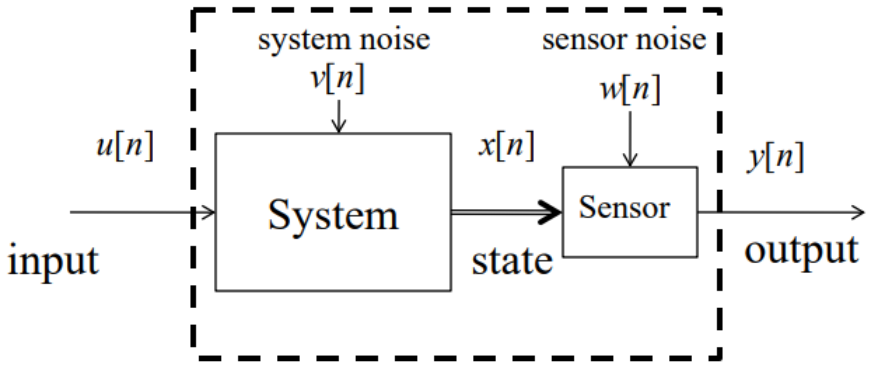

Figure 2.1. Block diagram describing the state space linear system model. Source: [9].

$$\begin{aligned} \mathbf{x}(n+1) &= \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n) + \mathbf{v}(n) \quad \textbf{System} \\ \mathbf{y}(n) &= \mathbf{C}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n) \qquad\qquad \textbf{Sensor} \end{aligned} \tag{2.1}$$

The **A**, **B**, and **C** matrices are typically used to describe the system behavior, and are responsible on how the next state and the system outputs are derived from the previous state and the system external input. KF is added in the system model as shown in Figure 2.2 below, and produces an estimated value of the state space variable $\mathbf{x}(n)$, by taking the system

input $\mathbf{u}(n)$ and the system output $\mathbf{y}(n)$ as inputs, without direct knowledge of the state space variable itself. **Q** and **R**, shown below in Equation 2.2, are the correlation matrices of the system noise ($v(n)$) and sensor noise ($w(n)$) respectively, where E is the expectation operator and the T symbol transpose operation.

$$\begin{aligned} \mathbf{R} &= E\{\mathbf{w}(n)\mathbf{w}(n)^T\} \\ \mathbf{Q} &= E\{\mathbf{v}(n)\mathbf{v}(n)^T\} \end{aligned} \tag{2.2}$$

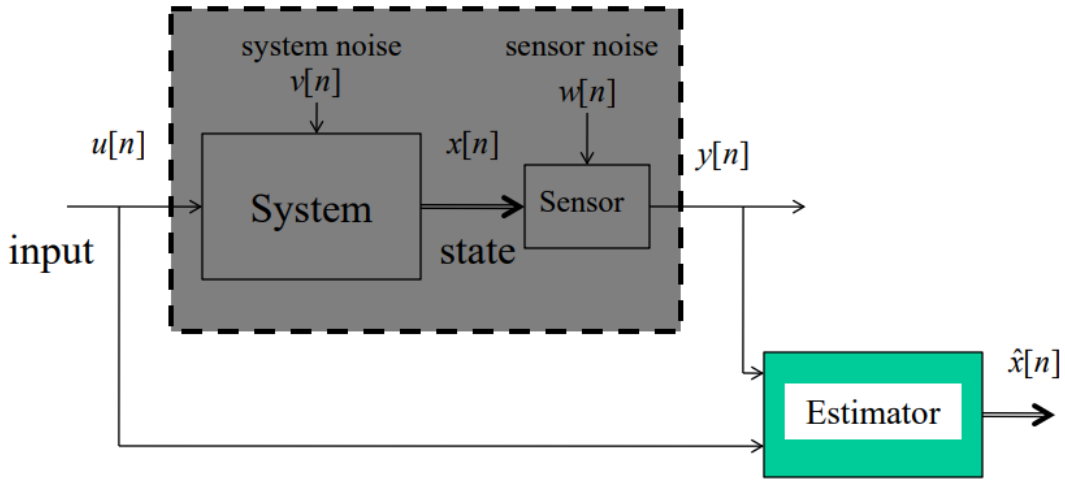

Figure 2.2. Block diagram describing the state space linear system model with the addition of an estimator. Source: [9].

For our simulation, we use the aforementioned constant velocity model. The state space equations describing the system are presented below. This model is described by the following equations, following the two-step process of correction and prediction, as described

above. The correction phase is described by the following equations [9]:

$$\begin{aligned} \mathbf{K}(n) &= \mathbf{P}_{-}(n)\mathbf{C}^{T}(\mathbf{C}\mathbf{P}_{-}(n)\mathbf{C}^{T}+\mathbf{R})^{-1} \\ \hat{\mathbf{x}}_{+}(n) &= \hat{\mathbf{x}}_{-}(n)+\mathbf{K}(n)(\mathbf{y}(n)-\mathbf{C}\hat{\mathbf{x}}_{-}(n)) \\ \mathbf{P}_{+}(n) &= \mathbf{P}_{-}(n)-\mathbf{P}_{-}(n)\mathbf{C}^{T}(\mathbf{C}\mathbf{P}_{-}(n)\mathbf{C}^{T}+\mathbf{R})^{-1}\mathbf{C}\mathbf{P}_{-}(n) \end{aligned} \tag{2.3}$$

The prediction phase follows the equations:

$$\begin{aligned} \hat{\mathbf{x}}_{-}(n+1) &= \mathbf{A}\hat{\mathbf{x}}_{+}(n)+\mathbf{B}\mathbf{u}(n) \\ \mathbf{P}_{-}(n+1) &= \mathbf{A}\mathbf{P}_{+}(n)\mathbf{A}^{T}+\mathbf{Q} \end{aligned} \tag{2.4}$$

The state variables are updated, as well as a matrix (**P**) that tracks the uncertainty of the estimation. The expected errors in the measurements and the accuracy in which the model describes the actual target kinematic behaviour are also taken into account. Thus, the estimator provides a weighted average between each new measurement and the previous movement of the target up until the present moment.

The above process happens in two steps/phases. The first one is called prediction phase (Equation 2.4), which occurs before the algorithm receives a new measurement. During this phase, the KF produces an estimate of the state variables, as well as their uncertainties (i.e., how reliable those estimates are). Once a new measurement is observed (together with any unwanted noise / disturbance), the KF goes into the correction phase (Equation 2.3), in which it produces a more accurate estimation by weighing the prediction from the previous phase and the observed measurement appropriately.

It is noteworthy to state that, although in essence all the past behaviour of the target is taken into account, this does not explicitly occur in every time step. The KF only needs the previous state of the variables and the uncertainty matrices to produce the next estimate. This is because each state effectively summarizes all past observations. Since the process is recursive, it is expected that as time goes by, the estimations are bound to get ever more accurate, as more information is compounded into the algorithm. The recursive aspect of the algorithm means that the process described is entirely repeated in an identical manner at every time step.

2.1.3 Variables Used

In order for the reader to have a better idea of the specific meaning of the above variables as implemented in this simulation, the state (**x**) is the target position and the state estimate ($\hat{\mathbf{x}}$) is the estimation of the target position calculated by the tracker (KF). The output (**y**) is the measurement provided by the sensor (in this case the URA). Sensor noise (**R / w**) here means mainly additive white Gaussian noise (AWGN) that is unavoidably inserted into our sensor (radar) from the receiver itself and/or environment. Process or system noise (**Q / v**) is something always present in any KF model, and is an adjustable quantity that can be manipulated empirically depending on how accurately our model is described in the system. The function variable dependence n means that the quantity is referring to this time step/sample, while n+1 signifies an estimation/prediction referring to the next time step/sample. Lastly, any quantities with the minus/plus subscript mean before/after we receive a measurement/observation respectively for the indicated time step/sample. A summary of all involved variables is provided in Table 2.1.

Table 2.1. Kalman filter variables.

Variable	Name	Description
$\mathbf{x}$	system state	the actual state that our system is in at the present time
$\mathbf{u}$	system input	any external input to the system
$\mathbf{y}$	system output	in this case the noise-contaminated radar measurement
$\mathbf{v}$	system noise	noise due to model-reality relationship imperfection
$\mathbf{w}$	sensor noise	noise coming into the sensor from the environment
$\hat{\mathbf{x}}$	state estimate	the best "guess" for the state that the estimator provides
$\mathbf{K}$	Kalman gain matrix	weighting coefficient between previous estimate and new measurements
$\mathbf{P}$	estimate covariance matrix	a measure of the estimated accuracy of the state estimate
$\mathbf{Q}$	system noise covariance matrix	describes relationship between system noise realization for different variables
$\mathbf{R}$	sensor noise covariance matrix	describes relationship between measurement /sensor noise realization for different variables

2.2 Uniform Rectangular Array

Having discussed our system modeling and general description, we now must describe our sensor. It is one of the most important components of the system, where the measurements come from. The type of antenna system to be utilized is called a uniform rectangular array (URA). This is a subcategory of radar antennae called phased arrays or electronically scanned arrays. Traditional radars depend on mechanical parts so that the scanning beam can be directed to the area of interest, whether that is tracking a target or surveying an area. On the contrary, phased arrays consist of an array of antenna elements (usually controlled by a digital circuit) which creates a beam of radio waves that can be electronically steered to illuminate different locations without moving the antenna. The latter described process is known as beamsteering. This is achieved by inducing different time delays to the waveforms, that are then fed into each individual array element [5], [13].

2.2.1 Working Principle

The elements of the phased array can vary in number as well as formation, depending on the application and objective to be achieved. For our simulation, we consider an active electronically scanned array (AESA). This means every array element has both a transmit and a receive module. As for the formation of the elements, we simulate a URA. In such a configuration, the array elements are positioned in a flat rectangular plane (with the antenna broadside facing the general area to be scanned), in a uniformly distanced pattern, as shown in Figure 2.3 below. Being the natural evolution of the linear array (in which all elements are placed on a single line), the URA allows for selectively steering the beam in 2 dimensions, in this case, azimuth (ϕ) and elevation (θ). Figure 2.4 illustrates a 3-dimensional representation of the beam pattern, assuming the beam is not steered (at its default broadside direction).

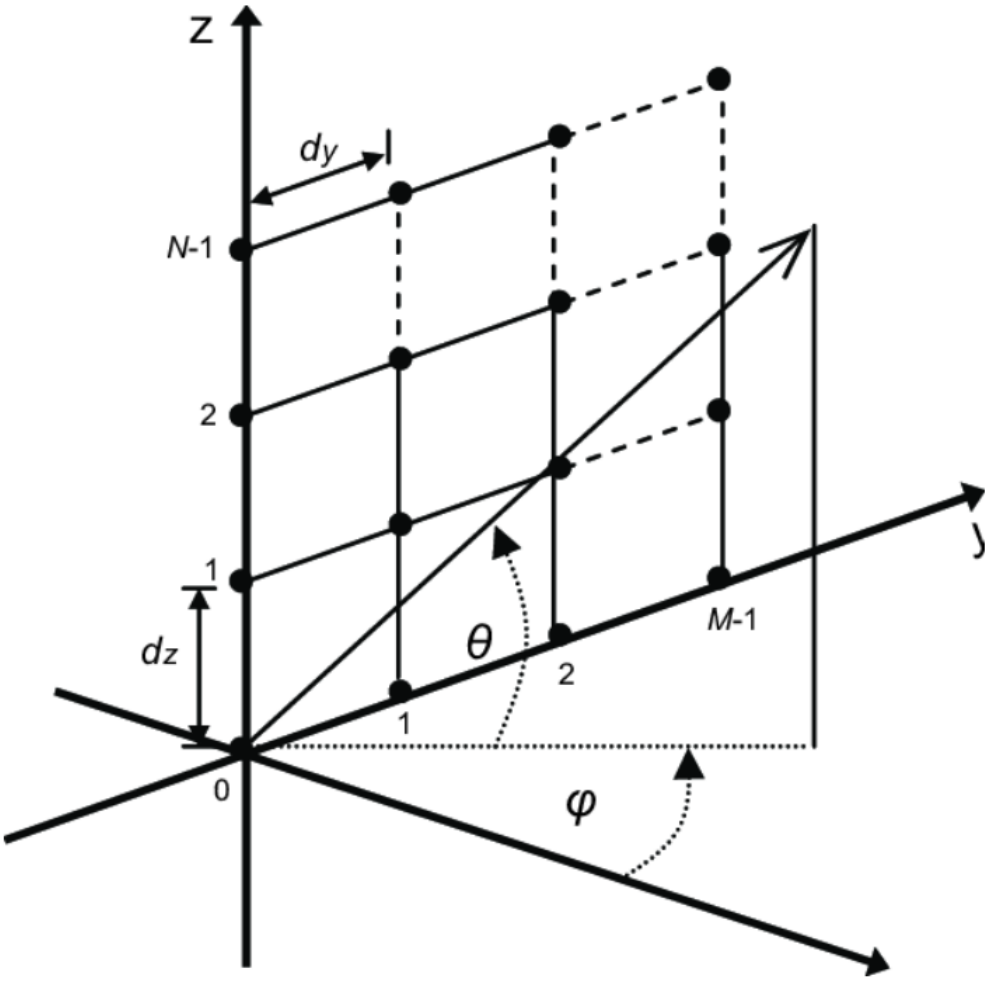

Figure 2.3. Position and formation of array elements in a Uniform rectangular array. Source: [14].

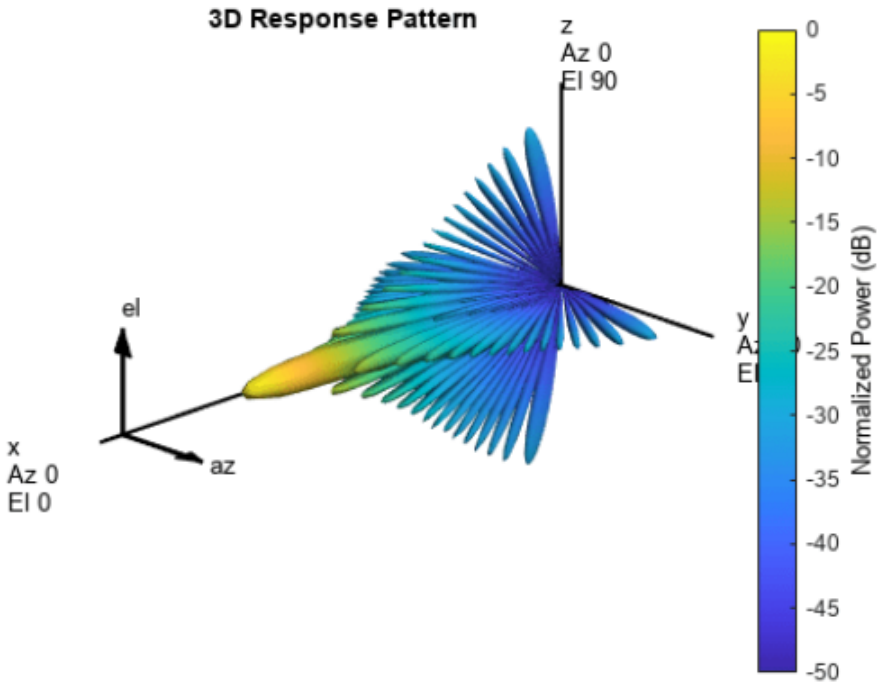

Figure 2.4. Visualization of the beam pattern with uniform feeding of the elements (no steering) with grating lobes. Source: [15].

2.2.2 Equations and Variables Used

Although not true in general, but simply for illustration, one may think of the URA in Figure 2.4 to be positioned at an altitude above ground level, so it is "facing" in a direction towards the horizon. That way, the beam being steered in two dimensions is directly equivalent to the azimuth and elevation of the target being tracked, similarly to what is shown in Figure 2.3. The distance between elements in both **x** and **y** direction is chosen to be equal to half of the wavelength of the transmitted electromagnetic wave to avoid grating lobes. Rectangular arrays with this spacing between the elements are also called standard rectangular arrays (SRA).

The equations we used that show how the energy is spread in space when steering the beam

into a certain angle in azimuth and elevation are shown and then explained below. This is otherwise known as the beam pattern [6]:

$$\mathbf{B}(\boldsymbol{\psi_x}, \boldsymbol{\psi_y}) = e^{-j(\frac{N-1}{2}\boldsymbol{\psi_x} + \frac{M-1}{2}\boldsymbol{\psi_y})} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} w_{nm}^{*} e^{j(n\boldsymbol{\psi_x} + m\boldsymbol{\psi_y})}, \tag{2.5}$$

where:

$$\begin{aligned} \psi_x &= \frac{2\pi}{\lambda} d_x \sin\boldsymbol{\theta} \cos\boldsymbol{\phi}, \\ \psi_y &= \frac{2\pi}{\lambda} d_y \sin\boldsymbol{\theta} \sin\boldsymbol{\phi}. \end{aligned} \tag{2.6}$$

Equations 2.6 can also be expressed in terms of the directional cosines :

$$\begin{aligned} u_x &= \sin\boldsymbol{\theta} \cos\boldsymbol{\phi}, \\ u_y &= \sin\boldsymbol{\theta} \sin\boldsymbol{\phi}. \end{aligned} \tag{2.7}$$

For the case in which

$$d_x = d_y = \frac{\lambda}{2}, \tag{2.8}$$

Equations 2.6 and 2.7 reduce to

$$\begin{aligned} \psi_x &= \pi u_x, \\ \psi_y &= \pi u_y. \end{aligned} \tag{2.9}$$

As seen above, the pattern can be expressed both in terms of u-space and ψ-space. These are transformations that are performed to facilitate further algebraic manipulations. However, those expressions always map to unique angles of phi and theta (elevation and azimuth).

The visible region is defined to be the area including all the possible angles the beam can be steered to. Based on the array's geometry, this area has physical limitations. For a SRA,

which is our case the limits of the visible region are as follows:

The visible region is dictated by:

$$u_r \triangleq \sqrt{u_x^2 + u_y^2} \leq 1. \tag{2.10}$$

In terms of ψ_x,ψ_y, the visible region is dictated by:

$$\sqrt{\left(\frac{\psi_x}{d_x}\right)^2 + \left(\frac{\psi_y}{d_y}\right)^2} \leq \frac{2\pi}{\lambda}.$$

For the table of variables that follows (Table 2.2), it is implied that the rectangular array is positioned along the x-y plane, and perpendicular to the z-axis.

Table 2.2. URA / SRA beam pattern equation variables.

Variable	Units	Description
$\mathbf{N}$	unitless	Total array elements in the x-direction
$\mathbf{M}$	unitless	Total array elements in the y-direction
$\mathbf{n}$	unitless	x-direction array element index
$\mathbf{m}$	unitless	y-direction array element index
$\mathbf{w}$	unitless	array element weighting coefficient
$\boldsymbol{\theta}$	radians	target azimuth (normalized)
$\boldsymbol{\phi}$	radians	target elevation (normalized)
$\boldsymbol{\psi_x}$	radians	psi-space x coordinate
$\boldsymbol{\psi_y}$	radians	psi-space y coordinate
$\mathbf{u_x}$	unitless	U-space x coordinate (directional cosine)
$\mathbf{u_y}$	unitless	U-space y coordinate (directional cosine)
$\mathbf{d_x}$	meters	array element spacing in x-direction
$\mathbf{d_y}$	meters	array element spacing in y-direction
$\boldsymbol{\lambda}$	meters	wavelength

CHAPTER 3:
Objective

3.1 Problem Description

One of the most important parameters that determines the quality of target tracking is the accuracy of it, i.e. how close the target track is compared to the actual position of the target at any given moment. A lot of factors affect the accuracy of the track. In a single target tracking radar environment, a key factor is beamwidth. The narrower the beam, the more accurate the estimate of the target can be. In a phased array application, a straightforward way of achieving narrower beam without altering the waveform characteristics is increasing the number of elements in the array.

However, in an effort to make the beam as narrow as possible, we are presented with another problem: what if the target, due to an abrupt maneuver or due to increased noise realization in the environment, is found to be outside our beam, or in general outside of the location that the beam is being steered to. For this reason, we must devise an effective way of examining (scanning) the scene surrounding the expected target location. While there are many techniques like track while scan (TWS) that achieve a similar outcome (tracking a target while surveying the whole visible region), most of the time this is performed in a predefined way. This proposed system strikes a balance between having the best target tracking accuracy possible (minimum possible error), and scanning the entirety of the visible region in an adaptive manner, that is on a need-to-do basis. The system proposed and simulated in this thesis, achieves the above by utilizing an integrated system comprised of a URA and a KF estimator as the tracker.

The general operational idea is the fast subsequent beam steering onto a grid of positions that is centered around the expected target position, in an effort to pinpoint its actual current position. For example, a 3-by-3 area contains nine grid cells. Although not necessary, a cell can be the size of a main beam and is the usual practical choice. In the event that the target escapes this tightly knit area around the track, a logical approach is to increase the search area in which to place the main beam. However, this will increase the time in which the area

will be scanned due to the increased number of cells (e.g., a cell can be a main beam size as mentioned before) in which to illuminate. To keep the same number of cells to illuminate, the system then employs less array elements (and thus a wider beam) and performs the same process that is described above, albeit in a larger area and with loss of antenna gain. With the subsequent beamsteering covering the entire URA visible region, this ensures that the target (assuming is in it) will be located. The process is described in detail later in this chapter.

Recall that there is a decrease of gain if a decreased number of elements is utilized in a phased array. Since we assume an AESA radar, with power amplifiers in the transmitter part of the TX/RX module, we compensate for loss by increasing the transmit power by the commensurate amount to retain the same SNR for this "low resolution/large search area" mode.

It is also noteworthy to mention that when the area is illuminated (narrow beam or wide beam), not only the cell or main beam is illuminated but rather a slightly larger area. With a very stringent false alarm probability given reasonable signal-to-noise ratio (SNR), it is unlikely that false detections will occur in the cells that are illuminated by the sidelobes. The natural low sidelobe gains also help reduce false alarms.

3.2 Simulation Setup

Before going into the specifics of our system, we need to provide a visualization of the simulation environment. The URA simulated contains 20-by-20 array elements, equally spaced across both dimensions, with spacing such that Equation 2.8 is satisfied. The array surface spans the x-y plane, while the z-axis is perpendicular to it. The visible region, confined by Equation 2.10, is not strictly defined, and can be altered within those bounds to fit any scenario. The tracking is performed in a 2-dimensional space, in the sense that the target range from the URA is assumed to have been estimated from the current measurements, which is a straightforward calculation. This is a basic radar function and thus it needs not be explored further. The measured quantities of interest now are the target azimuth and elevation (naturally from the URA orientation). Furthermore, those have been normalized (to span from 0 to 1) for ease of manipulation. However, once the angular dimensions of the visible region are defined, the mapping between the normalized scale and

the actual angle is unique and can be easily converted to and from.

In our simulation setup, azimuth angle θ is measured between the z-axis and the projection of the target position on the x-z plane (shown as p′ in Figure 3.1 below), while the elevation angle ϕ is measured between the z-axis and the projection of the target position on the y-z plane (shown as p″). To correspond strictly with our simulation implementation, θ is the azimuth angle here while ϕ is the elevation angle (as opposed to Chapter 2). Figure 3.2 shows a large blue area, which is a visualization of the angular visible region, while the star symbolizes the target angular position. While this region essentially has the shape of a spherical sector, it is implemented by a rectangular matrix in the simulation, with minimal distortion, due to its aforementioned size constraint.

Recall we are tracking in azimuth-elevation angular space since this is natural to our situation and eventual simulation. In practice, Cartesian coordinate is sometimes convenient or desired, then proper coordinate transformation should be implemented as appropriate to application or scenario.

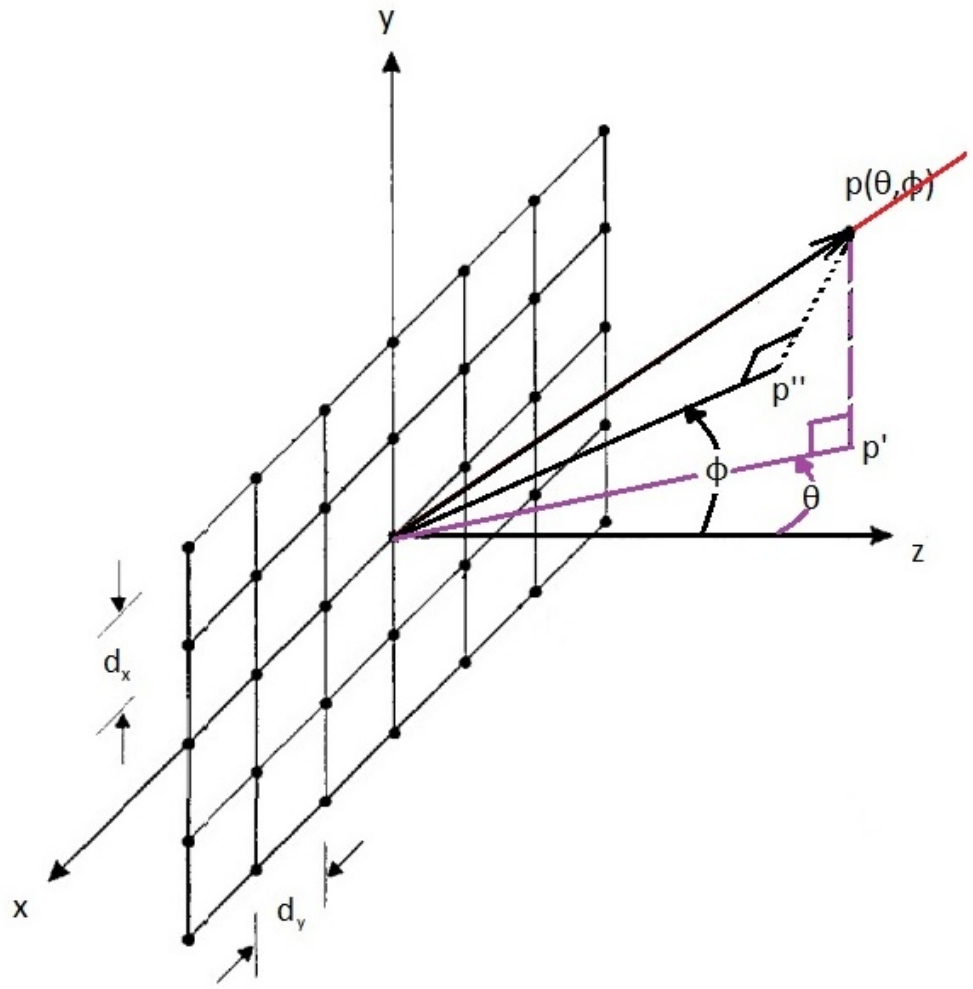

Figure 3.1. URA target angles geometry. Adapted from [16].

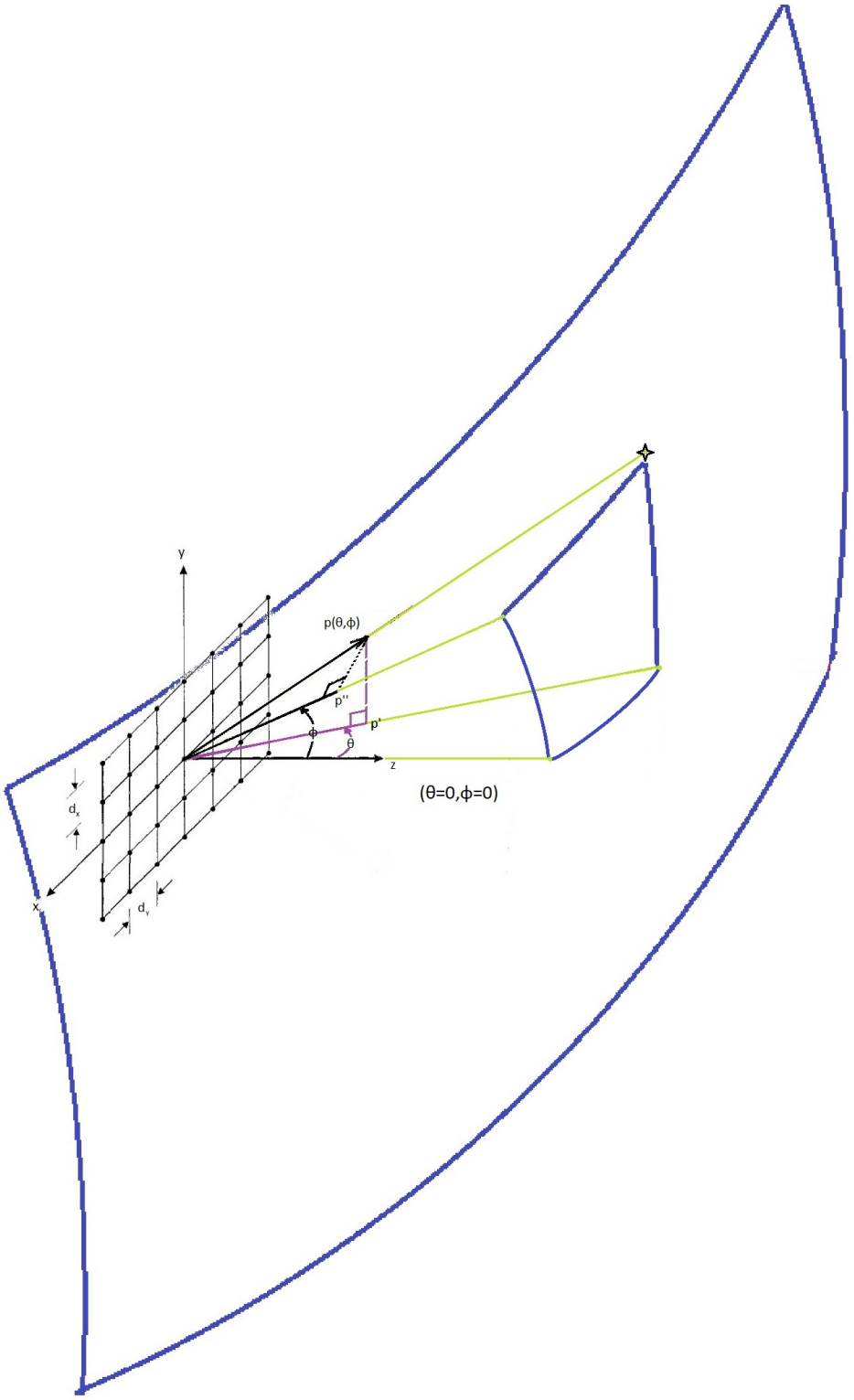

Figure 3.2. URA target angles geometry and visible region representation.

3.3 System Operation Description

The overall description of the system is briefly depicted in the following flow chart, and then further explained in detail.

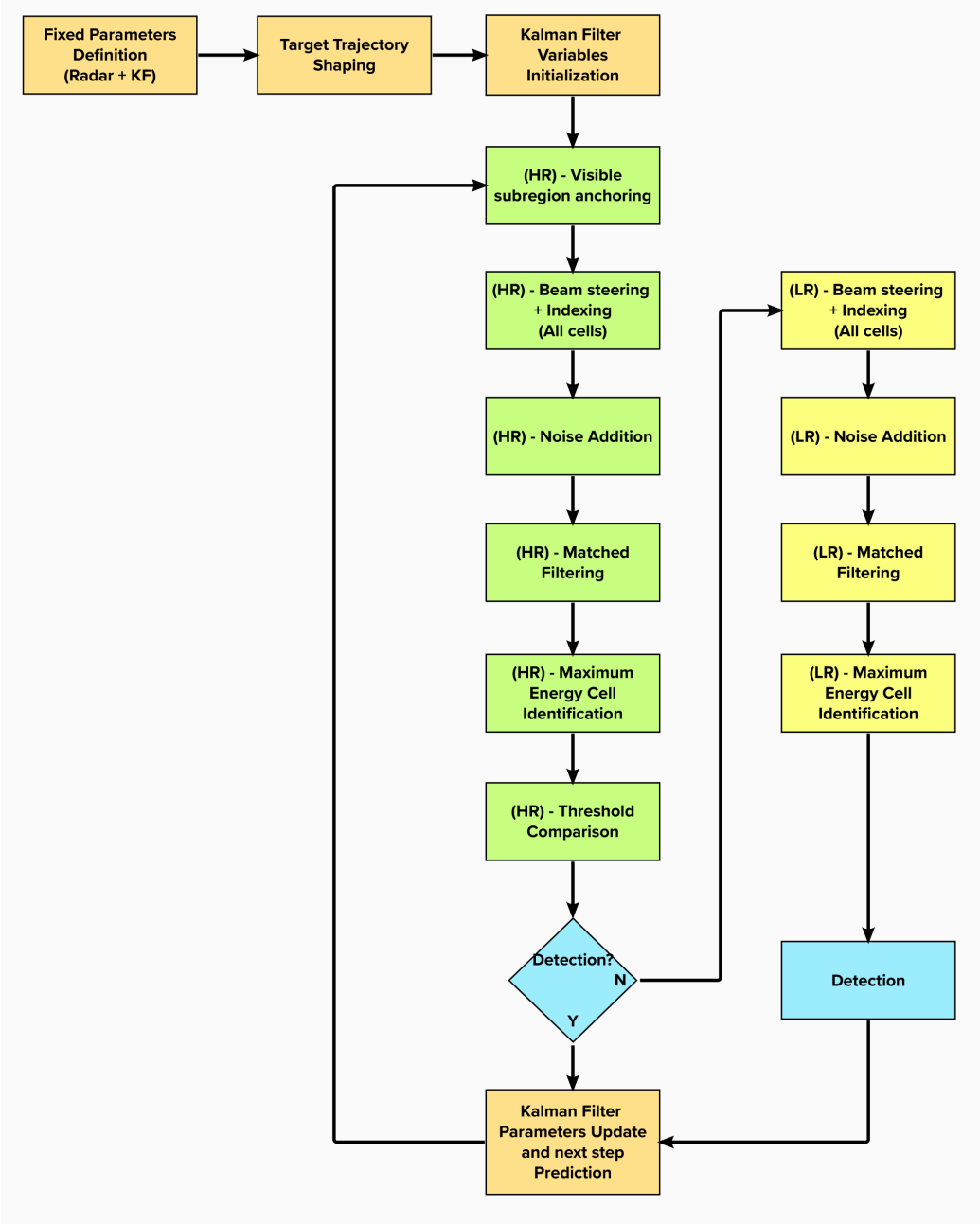

Figure 3.3. System function flow chart.

As mentioned in Section 3.1, the system features two modes of operation, one employing all of the URA's elements to provide maximum tracking accuracy, and one that uses 25% of its elements, in which the priority is to quickly illuminate the whole visible region. These two modes will from now on be referred to as high resolution (HR) and low resolution (LR) respectively. The functions that the proposed system performs when in HR mode are shown in green rectangles in the flow chart, while the LR functions are shown in yellow. A short description of every important step is provided as follows:

3.3.1 Fixed Parameters Definition

First, the fixed parameters concerning the radar system are defined. The URA simulated consists of a grid of 20-by-20 array elements. All of them are used in HR mode to achieve focusing the transmitted energy in a beam as narrow as possible. On the contrary, when the system operates in LR mode, we use a quarter of the elements, namely a 10-by-10 grid of array elements. SNR, usually called E_s/σ^2 (received signal energy over noise variance) in radar applications, is also defined here. A unity variance noise has been selected for the purposes of simulation in Matlab, such that the SNR desired depending on the scenario to be executed can be changed just by varying the simulated transmitted signal energy. Next, the visible region is divided into resolution cells. The dimensions of those cells are set in this section. The physical visible region of interest may vary from scenario to scenario depending on a priori knowledge by the system. For example, a solid angle with -30° to 30° in elevation and -30° to 30° in azimuth may actually be the area of interest. As the visible regions dimensions are normalized (0 to 1), the physical dimensions are not relevant at this point. We have opted to let the number of antennas in each respective mode to dictate the size of the cells for convenience. In both modes, the cells the radar is able to steer the main beam to and illuminate at any given time are 100 (10-by-10 grid). In LR mode they comprise the whole visible region, which means the size of each cell in LR is four times greater than the HR mode cell. An index number is assigned to each array element. The dual grid of the visible region is shown in the Figure below, with the HR grid being anchored/centered at an arbitrarily selected target estimated position (shown as a star).

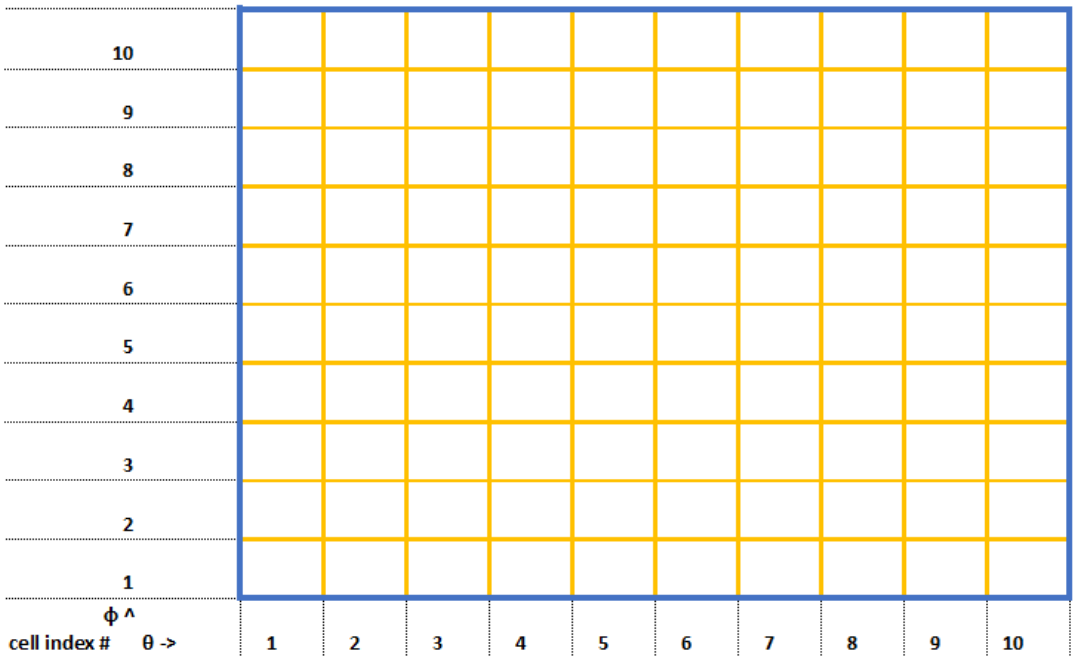

Figure 3.4. Low resolution grid construction.

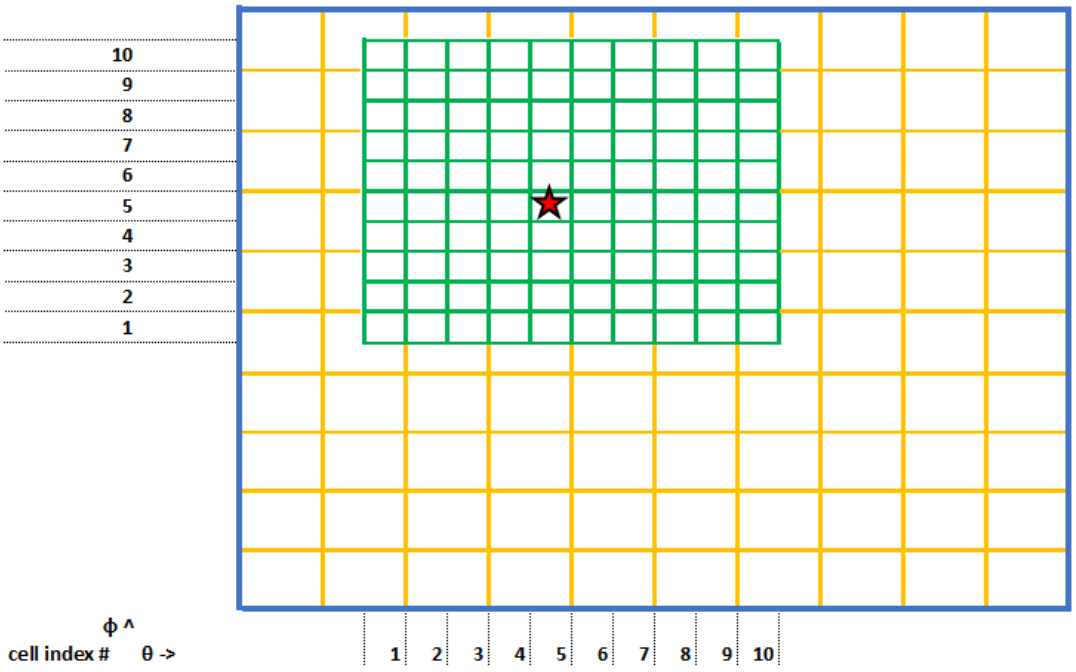

Figure 3.5. High resolution grid construction and placement.

3.3.2 Target Trajectory Shaping

This section is where the "ground truth" of the target motion is constructed. This part has no constraint other than having the target move within the bounds of the visible region. However, the more realistic the trajectory, the more applicable to a real life scenario the results and conclusions will be.

3.3.3 Kalman Filter Variables Initialization

This is the part where the variables that pertain to KF are given their initial values. This is necessary as right after this step of the dynamic simulation is where the recursion begins. As is typical in numerous KF applications, the value of **P** (estimate covariance matrix) is initially selected to be high, representing low "credibility" (or high uncertainty) in our knowledge of the target location. Its first estimate, however, is selected in the general area of the target initial position, which can come from operational intelligence (OpIntel) (i.e. the target is expected to arrive from a certain region/area). This is also the section where both process/system noise matrix (**Q**) and measurement noise matrix (**R**) are defined for HR and LR mode. The values of the latter are essentially defined by the actual resolution of our system in each mode, while the values of the **Q** matrix can and should be formed depending on the scenario (meaning depending on the movement profile we expect the target to have, as well as the "closeness/tightness" of the tracking that we want to achieve). Following this part is where the recursion/loop begins, which terminates when the target motion stops. Also note the system is initially in HR mode, which will go into LR if/when required and then will revert back into HR. In this way, the HR mode is the system default mode. This is true since we start the simulation in our study with the target already in HR track. But it is clear the LR may be the starting mode for quick detection and HR is switched for high resolution tracking (as desired or default mode).

3.3.4 Visible Sub-region Anchoring

As mentioned before, when in HR mode, the radar system is able to observe/steer the main beam to a quarter of the whole visible region. This section of the code defines where that sub-region is to be anchored. As the target moves away from the extremes of the visible region, the 10-by-10 cell grid is centered at the target's estimated position for each time-step. An example anchoring is shown in Figure 3.5. Closer to the edge, this 10-by-10 sub-region is

bounded by the actual limits of the visible region, accordingly. This is visually demonstrated in Figure 3.6. If the estimated target position is in one of the magenta colored areas, the visible region limits our main beam illumination. Therefore, in those boundary cases, the HR sub-region defaults onto the respective quadrant, shown here as separated by the double black line.

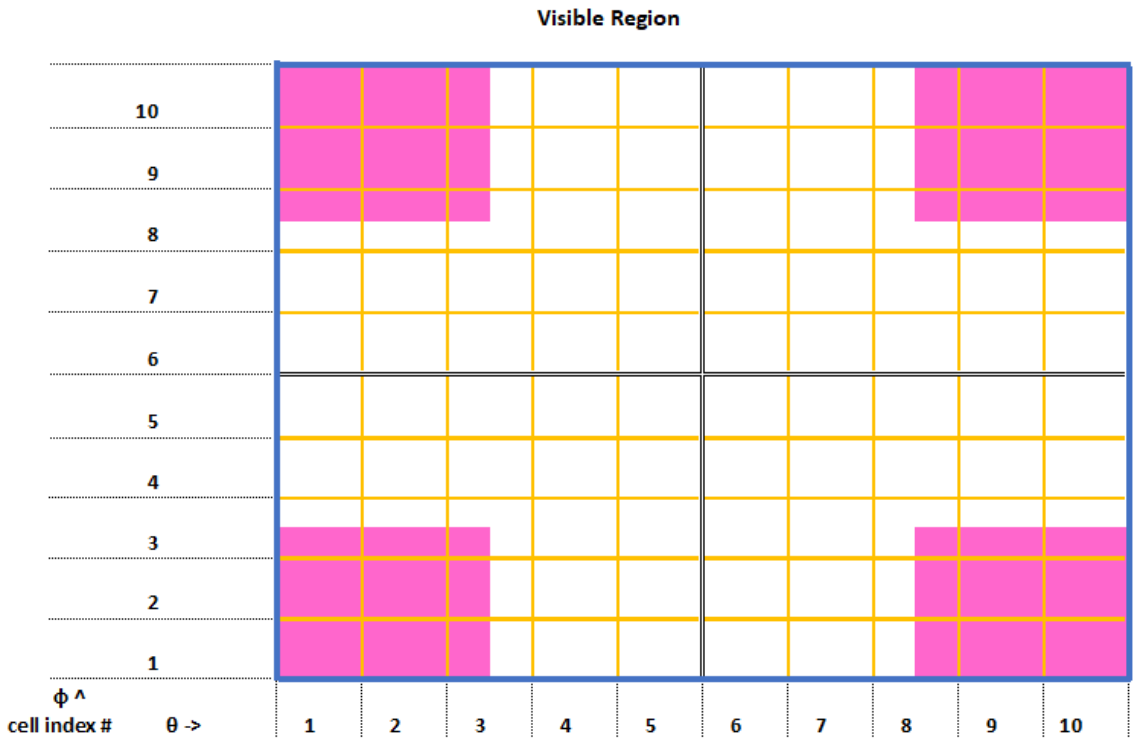

Figure 3.6. HR grid placement when estimate approaches the visible region extremes.

3.3.5 Beamsteering (HR)

This section pertains to the beam steering portion of the code. First, all the steering vectors that correspond to the illuminating beam being steered to each cell of the visible sub-region are constructed. Next, we construct the reception steering vector that corresponds to the actual/true target's position. At this point the algorithm calculates the theoretical received energy (before noise is added) to ensure it corresponds to the selected SNR. While the latter process seems like a necessary calculation for the system, in reality it is not required but a

practical moving target has to be placed in the simulation scenario with its corresponding received returned energy. An index value is assigned to each of the 10-by-10 cells for reference. Then, the received steering vector is placed on the indexed cell that corresponds to the true position of the target. Since we have made the dimensions of the cell match our URA beam width, it is assumed that no substantial energy overflows to adjacent cells. Essentially at this point, the actual received energy comes from the target's true position cell.

3.3.6 Noise Addition (HR)

In the section that follows, AWGN is added to the received signal of every cell that is being illuminated. This results in a matrix of 10-by-10 values of complex valued signals, each corresponding to a different cell.

3.3.7 Matched Filtering (HR)

Then, each of those signals goes through the process of matched filtering for detection. After this process, a 10-by-10 matrix of matched filtered values has been populated, one corresponding to each illuminated cell.

3.3.8 Maximum Energy Cell Identification (HR)

The cell index containing the maximum value of the above matrix corresponds to the cell where the target is possibly located, thus making it candidate for the corresponding angular position of the target.

3.3.9 Detection Decision

The maximum value of energy obtained is then compared to a radar detection threshold calculated based on the SNR selected for this scenario. The threshold corresponding to SNR (from fundamental radar theory) is shown in Equation 3.1. Provided the matched filter output value is higher than the threshold set, the system assumes detection and identifies that cell as the measured target position, which is the information propagated as the observation ($\mathbf{y}(n)$) to the KF. Otherwise, the system goes into LR mode as explained in 3.3.10 below. The detection threshold equation is given by:

$$Th = \sqrt{\frac{\sigma^2 E_s}{2}} Q^{-1}(P_{fa}), \tag{3.1}$$

where Th is the receiver threshold, E_s is the received signal energy, P_{fa} is our desired probability of false alarm (selected to be 10^{-4} for our simulations), and σ^2 is the AWGN variance (selected to be equal to 1 in the simulations). Q^{-1} here is the inverse Q-function, as defined by the Gaussian (normal) probability distribution.

3.3.10 Low Resolution chain

If the threshold is not exceeded by the matched filter output, the system assumes the target has escaped the visible sub-region and immediately reverts to LR mode. As mentioned, in this mode the totality of the visible region is illuminated, again in a 10-by-10 cell grid. However, the grid cells have quadrupled in size compared to the ones in HR mode. The radar system then performs exactly the same steps as described in 3.3.5 - 3.3.8 above. However, the values of **R**, **Q** matrices, as well as the size of the cells used are the ones corresponding to LR mode. The main difference here is the absence of a detection threshold comparison. The logic behind this is that, since the whole visible region has been scanned during this step, there is no other candidate location for the target since we assume only one target present in our problem formulation. Thus the cell containing the largest energy is automatically assumed to correspond to the target true location (detection), and this information is provided as the observation to the KF ($\mathbf{y}(n)$).

3.3.11 Kalman Filter Update and Estimate

In this section, the KF, utilizing the Equations in 2.3 and 2.4, updates its gain and error covariance matrices. Subsequently, using the same equations as well as the target location information provided by the detection process, it calculates/predicts the most likely (minimum mean square error (MMSE)) position of the target for the next time step ($\hat{\mathbf{x}}_{-}(n+1)$).

3.3.12 Iteration Conclusion

This concludes the recursion for this time step, as the estimated position for the next step is provided to the URA system (which is now in desired or default HR mode again), serving as a center point for the next set of illuminations.

CHAPTER 4:
Results

4.1 Results Structure and Description of Scenarios

After extensively testing the system created, the results are presented in this chapter. The resulting products of the simulation are divided into two sections. The first one pertains to certain metric comparisons that will give a mathematical sense of the appropriateness of each selected parameter for every scenario. These metrics have been produced by running Monte Carlo simulations (1000 trials for each specific set of parameters per scenario to obtain statistically reliable results). We will also observe how the MMSE calculated exclusively utilizing the KF (taking into account the system dual mode) compares to the actual error measured between the target true position and the estimated position produced with the proposed adaptive beamsteering strategy.

The second section of the resulting products in this chapter contains simulated animations produced by use of Matlab plotting capabilities. In these short videos, the viewer can obtain a first-hand sense of the effectiveness of the selected parameters for each scenario. The field of view contains the entire visible region, where the viewer can see the true target position being updated at every timeframe. This true position, of course, is unknown to the proposed system. Simultaneously, the viewer can observe the location of the estimated position, as well as a frame-by-frame update and history, of all the measurement inputs to our estimator that come from the radar system via the beamsteering technique described in Chapter 3. It is essentially an all-inclusive representation of the whole system performing the intended processes.

For the two sections of the results that are presented in this chapter, three main scenarios are considered. The word scenario in this case pertains specifically to the target trajectory. Thus, each scenario represents a specific target motion sequence, while other variables like SNR and $\mathbf{Q}$ matrix covariance are given a range of values, such that some conclusions can be drawn. The main idea is to make the target motion increasingly complex from one scenario to the next one. The three aforementioned scenarios are described below, which

for the rest of this thesis will be referred to as scenarios 1, 2 and 3 respectively.

4.1.1 Scenario 1: Linear trajectory with sinusoidal component

In this scenario, which is the simplest one, the target trajectory spans the visible region in a linear diagonal fashion (from bottom left to top right) with constant velocity. A small sinusoidal component has been added to the motion to simulate slight perturbations to the target course, which would unavoidably be the case in a practical real life situation. A representation of the trajectory described is shown in Figure 4.1.

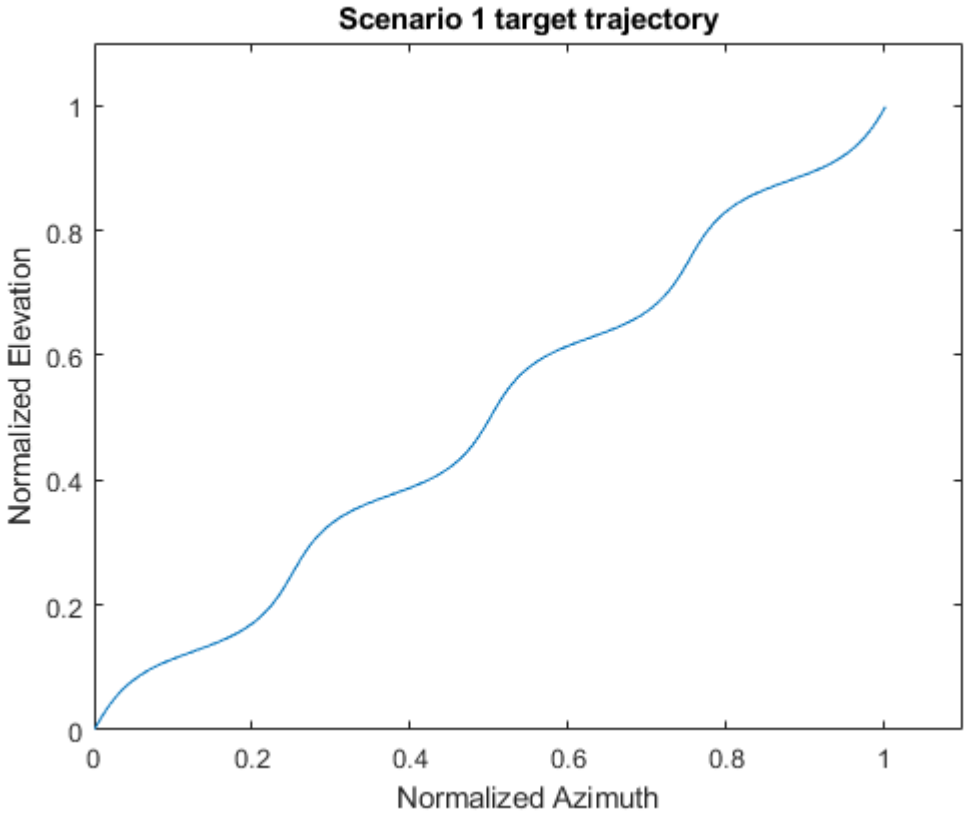

Figure 4.1. Scenario 1 target trajectory

4.1.2 Scenario 2: Linear trajectory with two smooth 90 degree turns

During this scenario, initially the target moves in a linear trajectory with a constant velocity. After a certain number of timeframes, it performs a 90 degree right turn (while maintaining constant velocity but with a turn dictated by smooth arc). Immediately after the turn, the target moves in a straight line again, before performing another turn exactly as the one described, but this time to the left. That is, after the second turn, the target has a course parallel to its initial one while maintaining the same velocity. Thus, it continues moving in a straight line until the termination of its course. A representation of the trajectory described in shown in Figure 4.2.

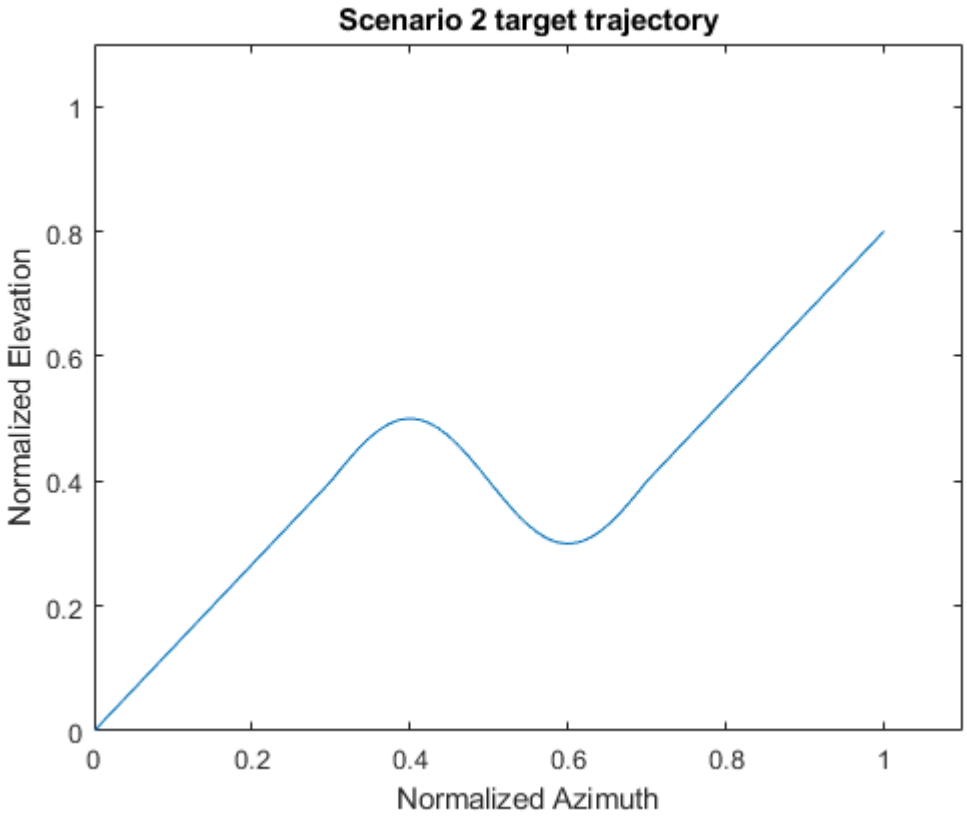

Figure 4.2. Scenario 2 target trajectory

4.1.3 Scenario 3: Quadratic / Circular trajectory

In this last scenario, the target motion is highly nonlinear. It initially follows a downward trajectory of quadratic form, while decelerating. After spanning half of the visible region extent in the azimuth dimension, it goes into a circular loop while maintaining a constant velocity. Next, immediately after exiting said loop, it mirrors its previous motion, following a quadratic trajectory again, now moving upwards (increasing elevation) and simultaneously accelerating. It eventually reaches the end of its trajectory after having spanned the full azimuth extent of the visible region. A representation of the trajectory described in shown in Figure 4.3.

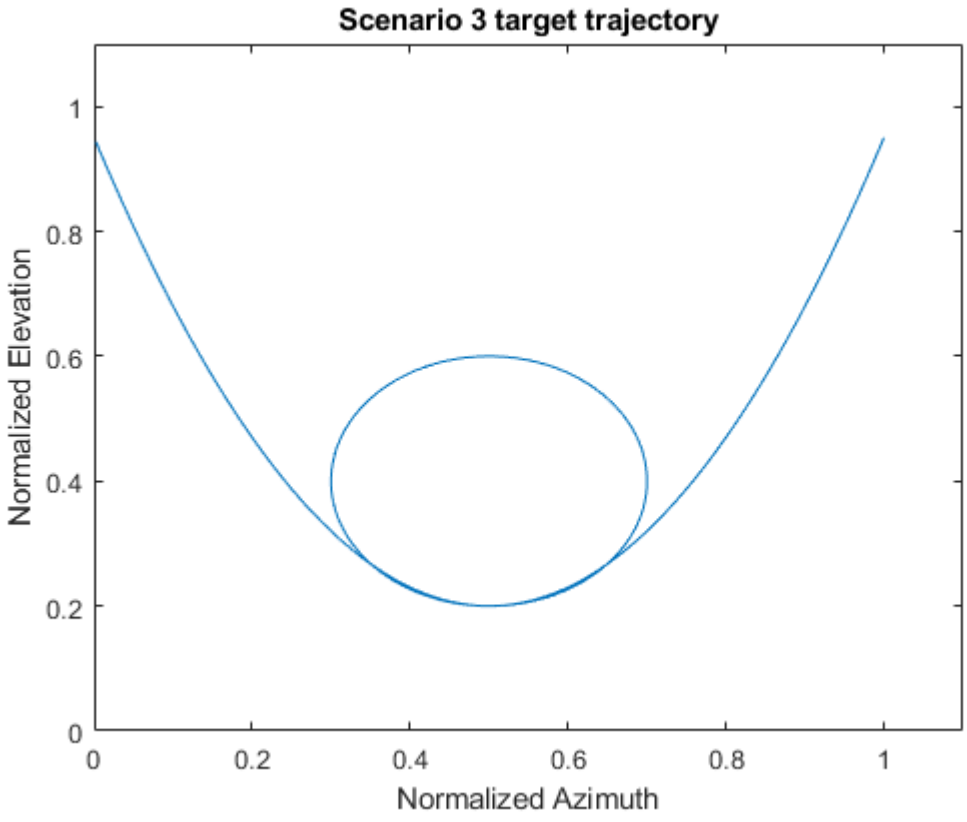

Figure 4.3. Scenario 3 target trajectory

4.2 Monte Carlo Simulation Results

In this section, after having performed Monte Carlo simulations for each of the three scenarios, the results are presented in the form of plots. The two main variables of interest are presented and explained as follows:

On one hand, we have the average angular error between the true target location and the position of the estimate produced by our system. This is the most direct way of measuring the effectiveness of the tracker, as it illustrates exactly how far the track is from the actual target location, averaged across the whole duration of its motion. This error is also expressed in a normalized fashion, as are the azimuth and elevation. Thus any value presented is essentially a fraction of the angular extent of the visible region, in azimuth or elevation.

On the other hand, we have the **P** matrix covariance, which is an indication of the uncertainty of the produced estimate. In all scenarios, during the KF initialization, this quantity has been set to be extremely high, reflecting the fact that we have essentially no or limited a priori knowledge of the target exact initial position. However, as is the case with all KF applications, the **P** matrix covariance gradually decreases as the simulation progresses, and eventually reaches a steady state value. A few details must be clarified at this point. Since **P** is a matrix and not a simple scalar, the trace of the matrix is the quantity observed. Also, while some examples of how this quantity changes over time are presented, for most of the results that follow, we have documented the final (or steady state) value of the the trace of the **P** matrix. Due to the fact that the **P** matrix contains information pertaining to both target velocity as well as position, the units of this quantity do not correspond directly to physical quantities but nevertheless have intuitive meaning. Moreover, the relationship between those values for different settings of each scenario as well as between scenarios, can provide the reader with a good sense of how certain / trustworthy the tracker is, with its produced estimate.

The above quantities have been documented and are presented in the following Subsections 4.2.1 through 4.2.3, for eight different values of **Q** matrix covariance (process noise), and values of SNR ranging from 7 to 20 dB. Recall that **Q** matrix covariance is essentially a tool that we have to "fine-tune" for our tracker, the optimal value of which cannot be mathematically derived beforehand, but rather empirically established, and depends on the scenario examined. Summarizing, the **P** matrix trace values or the MMSE are reported

while the SNR is varied. A group of curves will be in every resulting plot, each representing a different value of **Q** matrix covariance.

4.2.1 Scenario 1

Figures 4.4 and 4.5 show the steady-state error covariance and mean normalized angular error versus SNR for Scenario 1 for different values of process noise covariance (**Q**).

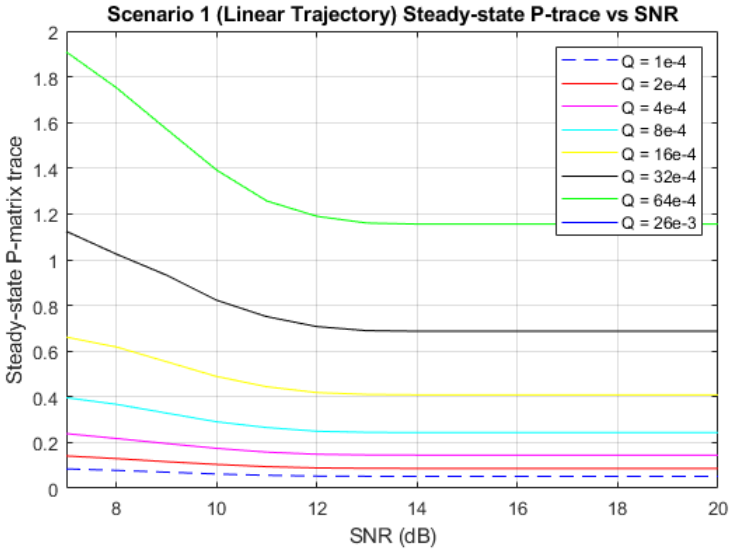

Figure 4.4. Scenario 1 P-matrix trace vs. SNR

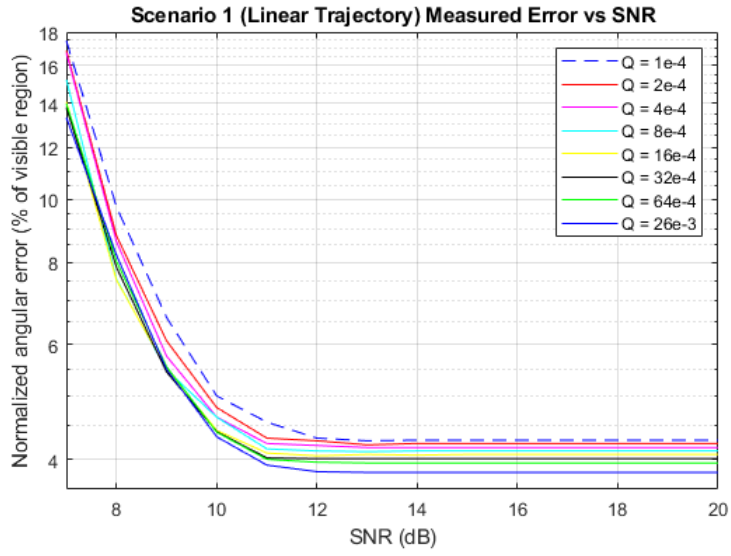

Figure 4.5. Scenario 1 normalized angular error vs. SNR

4.2.2 Scenario 2

Figures 4.6 and 4.7 show the steady-state error covariance and mean normalized angular error versus SNR for Scenario 2 for different values of process noise covariance (**Q**).

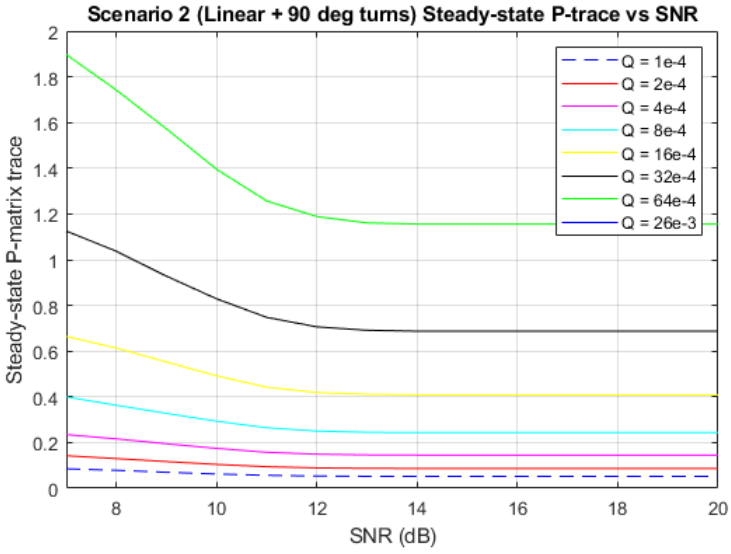

Figure 4.6. Scenario 2 P-matrix trace vs. SNR

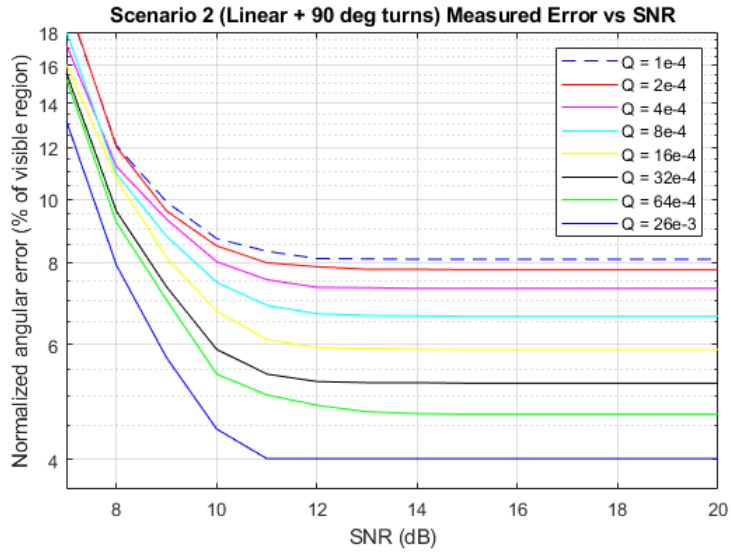

Figure 4.7. Scenario 2 normalized angular error vs. SNR

4.2.3 Scenario 3

Figures 4.8 and 4.9 show the steady-state error covariance and mean normalized angular error versus SNR for Scenario 3 for different values of process noise covariance (**Q**).

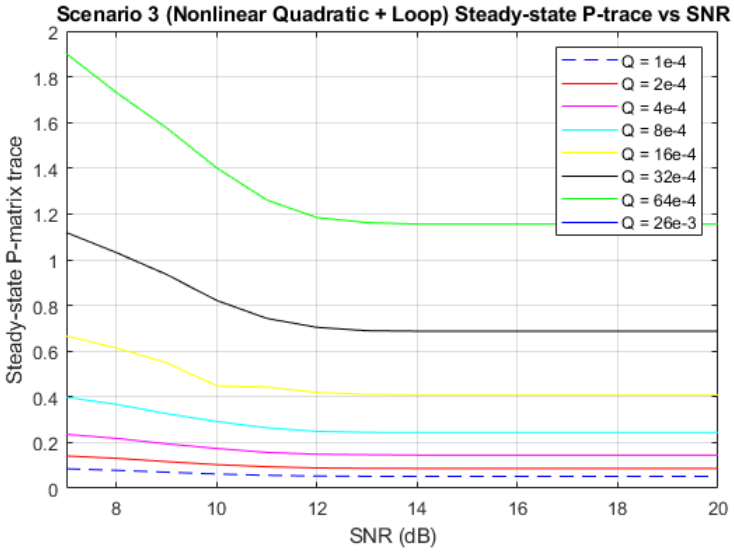

Figure 4.8. Scenario 3 P-matrix trace vs. SNR

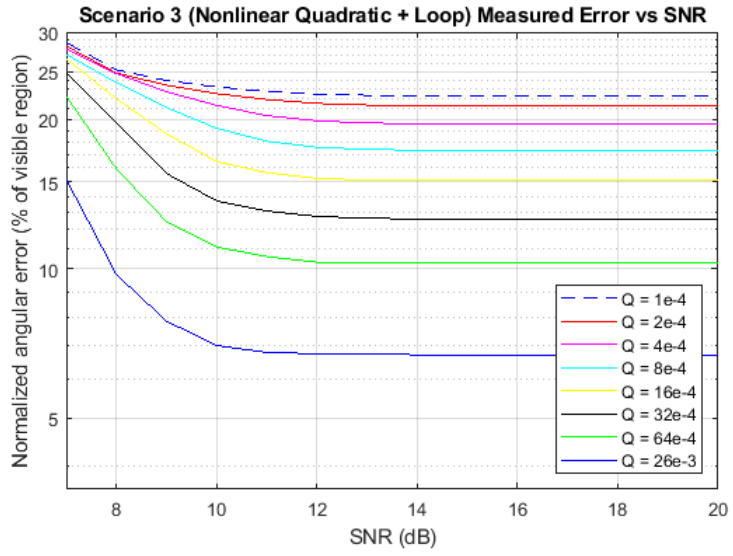

Figure 4.9. Scenario 3 normalized angular error vs. SNR

4.3 Animations

The short video animations produced by using Matlab frame capture and video creation capabilities are essential resulting products of this thesis, as they provide the most direct and intuitive sense of how well the tracker performs in each situation. The three scenarios discussed are presented in the animations, each with a selection of four different values of **Q** matrix covariance, and two SNR environments, considered as "high" and "low". Since the simulation has been extensively tested, these SNR values have been determined experimentally. They aim to illustrate the difference between the system going into LR mode occasionally (which happens in low SNR environments), and the system operating solely in HR mode in high SNR environments. In the latter case, noise realization is not enough to make detection in HR mode problematic, in a sense that the system very rarely has to switch modes. In summary we present a total of 24 videos, 8 for each scenario. The frame rate has been selected so that the animations are slow enough for the viewer to make the necessary observations, but fast enough so that they are short enough for convenience of presentation, resulting in videos of approximately 23 seconds. During each video, the target true position is shown in blue, while the system estimate (track) is shown in magenta. Green and yellow circles represent measurements / detections in HR and LR respectively. The diameter of said circles represents the actual beamwidth in each mode, which is in tune with dimensions of the grid cells in each mode. The videos, which are uploaded on the www.youtube.com platform, along with the main settings for each one are summarized and available through hyperlinks in Table 4.1 that follows:

Table 4.1. Video simulation settings and hyperlinks.

Scenario	SNR	**Q**-covariance	Hyperlink
1	High (20dB)	$1 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas
		$4 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas2
		$64 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas3
		$256 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas19
	Low (12dB)	$1 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas4
		$4 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas5
		$64 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas6
		$256 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas20
2	High (20dB)	$4 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas7
		$16 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas8
		$64 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas9
		$256 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas21
	Low (12dB)	$4 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas10
		$16 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas11
		$64 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas12
		$256 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas22
3	High (20dB)	$4 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas13
		$16 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas14
		$64 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas15
		$256 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas23
	Low (12dB)	$4 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas16
		$16 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas17
		$64 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas18
		$256 \cdot 10^{-4}$	https://tinyurl.com/GSouchlas24

4.4 Other Measurements

Finally, some other noteworthy results are presented in this section as follows:

4.4.1 Evolution of P-matrix covariance over time

Two different examples of the progression of the **P**-matrix covariance over time are shown in Figure 4.10 below. The purpose is to illustrate the effect of the LR mode in the system uncertainty. A high SNR case is presented on the left, in contrast to the low SNR case shown on the right. The smooth asymptotically declining curve in the high SNR case is typical of any KF implementation. However, the "jagged" parts of the curve in the low SNR case are a direct result of the system architecture, reflecting the increase in uncertainty whenever the system is forced to go into LR mode.

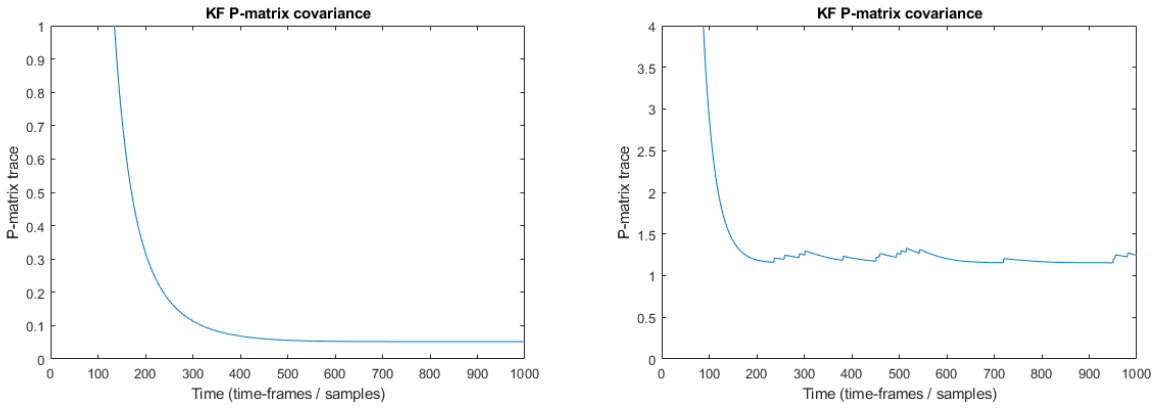

Figure 4.10. P-matrix evolution over time (High vs. Low SNR).

4.4.2 Evolution of MMSE over time

An example plot for each of the 3 scenarios simulated is presented in the following figures. This is helpful to understand how the specific maneuvers of the target impact the tracker error from the true target position. Figure 4.11 corresponds to scenario 1, Figure 4.12 to scenario 2 and Figure 4.13 to scenario 3. There is a direct correlation between the target deviating from the system-expected linear constant velocity behaviour and the increase in MMSE. Equivalently, when the target reverts back to a trajectory of the expected type, the error subsequently begins decreasing again.

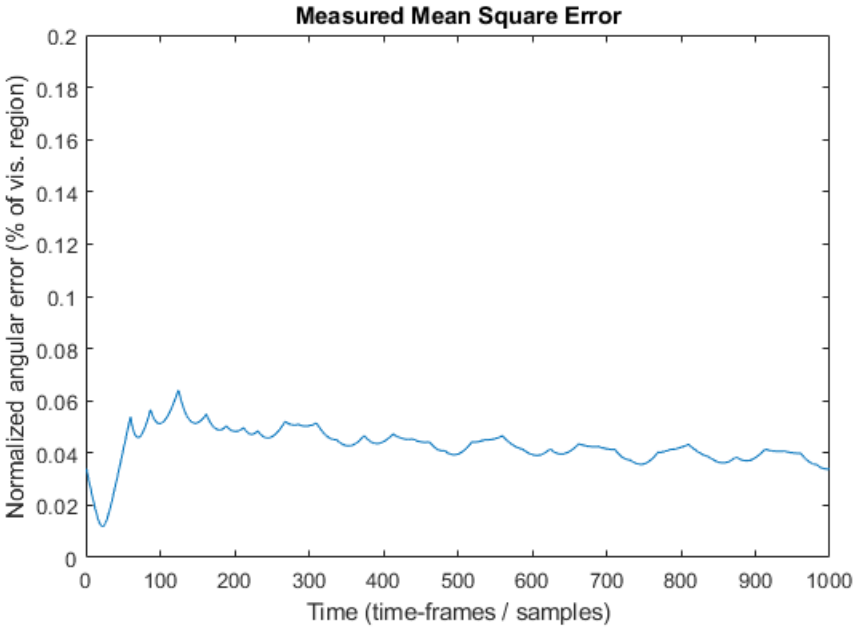

Figure 4.11. Mean square error evolution over time (Scenario 1).

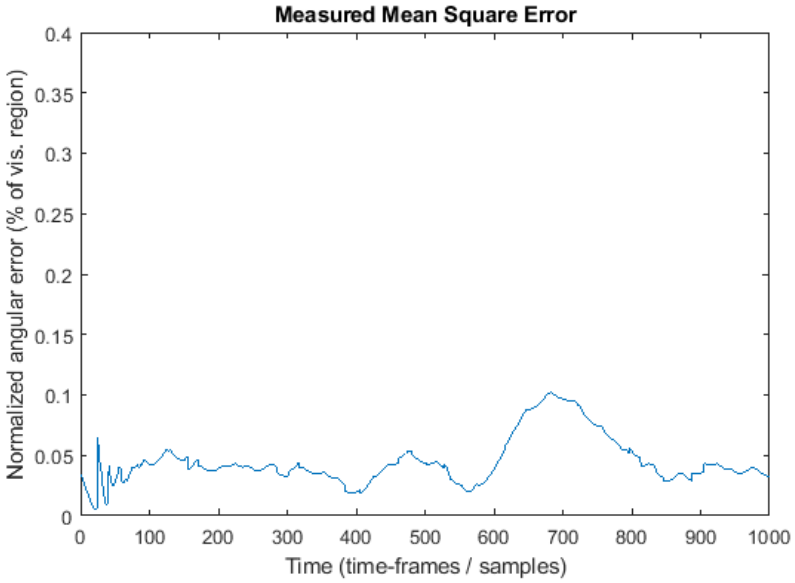

Figure 4.12. Mean square error evolution over time (Scenario 2).

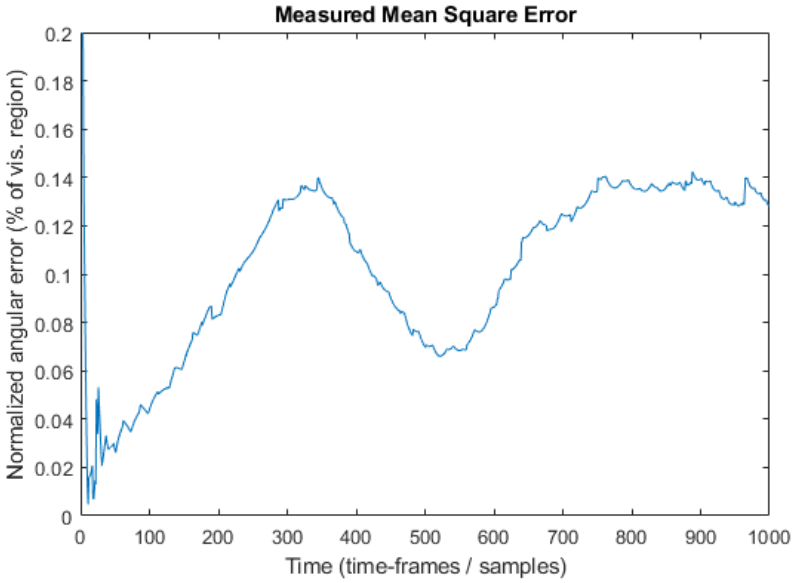

Figure 4.13. Mean square error evolution over time (Scenario 3).

Conclusions drawn based on the totality of the results presented in this chapter, are presented in Chapter 5 that follows.

CHAPTER 5:
Conclusions & Future Work

5.1 Conclusions

First, one of the main conclusions that can be drawn is that the dual mode system performs as intended. The dual set of measurement noise matrices (**R**) and process noise matrices (**Q**) coupled with the different radar beamwidth, which corresponds to the equivalent illumination cell size, provides a very realistic sense of the two distinct modes of the system (HR and LR). These work harmoniously, meaning that the transitions between them do not disrupt the correct function of the embedded KF. Additionally, the choice to have a second set of **Q**-matrix covariance, makes the transition from HR to LR reflect the desired effect on the tracker's uncertainty (**P**-matrix value increase), without requiring any additional modification to the core of the KF's code. This is illustrated in Figure 4.10 on the right side, where the "jagged" increases of the **P**-matrix represent transitions from HR to LR mode.

Another conclusion, which is typical of any radar-centric simulation or application, is the importance of SNR. This becomes obvious when looking at Figures 4.4, 4.6 and 4.8, where regardless of the scenario considered, the tracking error increases dramatically with decreased SNR. Furthermore, from the same plots we can infer that any SNR increase after 13-14 dB does not noticeably improve the performance of the system. This fact can also be deduced by observing the high SNR scenario animations/videos, where it is apparent that the noise realization does not create enough interference to force the system to go into LR mode, thus maintaining the best resolution possible and subsequently keeping error to a minimum.

However, upon further examination of the same figures, we can make concluding remarks about the role that process noise covariance (**Q**) plays in the overall track error in every scenario. It is apparent that increasing (**Q**) has minimal effect in scenario 1, but the same effect becomes significant in scenario 2 and even higher in scenario 3. This makes sense, if we consider these two things. Firstly, (**Q**) is neither mathematically derived nor corresponds directly to any physical property of the system. It is different for each application and

condition and has to be empirically determined through trial and error. It represents how much the actual movement of the target deviates from what the system "expects", based on the KF model implemented. Secondly, recall that the constant velocity model has been used in the proposed system formulation, thus rendering any other kind of target movement "unexpected". Thus, as the target movement becomes more and more nonlinear in scenario 2 and especially 3, it becomes obvious that increasing the process noise covariance lets the tracker more "freely" follow the target movement, and not be as constricted to the expected linear / constant velocity behaviour. While this is sub-optimal from a design standpoint, it is a really good way to use our tracker in situations it otherwise would be practically useless. (For example, the tracking error in scenario 3 under high SNR goes from 24% to under 7% when higher **Q** is selected, as can be seen in Figure 4.9).

Some comments and explanations are necessary to interpret the plots shown in Figures 4.11 through 4.13. The progression of the measured tracking error throughout the simulation, provides good insight as to what aspect "makes it harder" for the tracker to follow the target, which eventually comes down to the selected KF model used, in this case the constant velocity one. Going into each scenario specifically, we can observe the error have a slight fluctuation in Figure 4.11, which is a direct result of the sinusoidal component of the target motion. However, the general downward trend is typical, as the tracker compiles more measurements into the state, thus driving the error to a steady state value. Proceeding to Figure 4.12, which corresponds to scenario 2, we initially observe a steady, low error. Around timeframe 400, we see an increase in that error in the form of a "hump", and then almost immediately after, a second, bigger "hump" starting around timeframe 600. These two increasing trends of the error are the result of two consecutive 90-degree turns by the target. It is noteworthy to mention that, if it were a single turn and then the target proceeded with a constant velocity, the tracker would be able to "recover" quite quickly. However, with the second turn in the other direction being highly unanticipated (as was the first), the error is compounded. In the final part of the trajectory, where the target assumes a straight course with constant velocity again, the error has enough time to settle down almost to its steady state value. Lastly, in scenario 3, represented in the plot of Figure 4.13, we observe the error quickly rising to much higher values than in the previous cases, only to decrease briefly between timeframes 360–530, which is when the target has already decelerated and about to enter the constant velocity with circular looping trajectory. Upon entering the loop with

the velocity vector being rotational, the error increases once again. The error remains high in the final part of the target motion, which is accelerating and moving in a highly nonlinear fashion.

Closing this section, some observations are due pertaining to the realistic aspect of the simulation and the necessary adjustments for a potential practical implementation. The relationship between the target angular velocity with the system refresh rate is a serious consideration in case of a real life application. The refresh rate can be fine-tuned according to the expected angular velocity of targets in the area of interest, such that the system does not dwell on the same resolution cell for multiple measurements, while still having high enough refresh rate to correctly capture the target motion. Furthermore, this presents a limitation that also depends on computing power, which also needs to be taken account in conjunction with refresh rate / PRF, so that the system can perform the necessary calculations within the interval between each timeframe. Lastly, it is reminded that the value selected for measurement noise covariance (**R**) has to align with the beamwidth / resolution cell size, which in turn will depend on carrier frequency and array element spacing.

5.2 Future Work

The system presented in this thesis is a solid set of functioning algorithms, with a lot of flexibility for further development and improvement, in various places. Regarding the system operation, i.e. the logic of the algorithm as described in section 3.3, a few adjustments could facilitate a slightly different behaviour that could be beneficial depending on the scenario under consideration. Specifically, before going into LR mode, the system could propagate the last produced estimate to the next timestep, and then go into LR mode on the next timeframe. This would allow for less computing strain on the system in the particular interval in which the detection threshold is not tripped. Alternatively, an addition of a detection threshold in LR mode in the case of multiple targets will increase the LR system credibility. The system could remain in LR mode until a detection threshold is exceeded, and then move into HR mode on the next timeframe. Furthermore, adding a functionality for manual switch between the 2 modes as an input by the user, would improve functionality depending on the operational situation considered (resource management / spatial awareness).

As far as more general improvements and additions to the proposed system, one consider-

ation is the addition of a more generic search mode (it could be the system default mode for low resource consumption and then some condition would necessitate the transition into one of the track modes). Another is the added functionality of simultaneous tracking of two or more targets. Also, the incorporation of higher order Kalman filters, (or extended Kalman filter) could facilitate the optimal tracking of targets regardless of their trajectory, as well as to accommodate for more complex interference and noise environments. Furthermore, adding a Doppler shift component to the measurements would be a big undertaking, but one that could make tracking possible in 3 dimensions, i.e. adding the velocity dimension. Lastly, waveform design could be incorporated, along with variable radar cross section (RCS) models for the target (Swerling etc), such that the target detection in a cluttered environment would be more realistic.

List of References

[1] D. C. Schleher, *Introduction to Electronic Warfare*. Dedham, MA, USA: Artech House Publishers, Dec. 1986.

[2] A. J. Fenn, D. H. Temme, W. P. Delaney, and W. E. Courtney, "The development of phased-array radar technology," *Lincoln Laboratory Journal*, vol. 12, no. 2, p. 20, 2000.

[3] K. Kiest, "Phased-array weather radar can improve weather observations and the timeliness of warnings," July 6, 2022 [Online]. Available: https://www.meteorologicaltechnologyinternational.com/opinion/phased-array-weather-radar-can-improve-weather-observations-and-the-timeliness-of-warnings.html

[4] "Phased array | Analog Devices." Available: https://www.analog.com/en/applications/markets/aerospace-and-defense-pavilion-home/phased-array-solution.html

[5] J. Li and P. Stoica, *Robust Adaptive Beamforming*. John Wiley & Sons, Oct. 2005, google-Books-ID: a98fGhHNo_kC.

[6] H. L. Van Trees, *Detection, Estimation, and Modulation Theory. 4: Optimum Array Processing*. New York, NY, USA: Wiley, 2002.

[7] R. J. Meinhold and N. D. Singpurwalla, "Understanding the Kalman Filter," *The American Statistician*, vol. 37, no. 2, pp. 123–127, May 1983. Available: http://www.tandfonline.com/doi/abs/10.1080/00031305.1983.10482723

[8] G. Welch and G. Bishop, "An introduction to the Kalman Filter," University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, USA, Tech. Rep. 95-041, 1994.

[9] A. Gelb and A. S. Corporation, Eds., *Applied Optimal Estimation*, 19th ed. Cambridge, MA, USA: M.I.T. Press, 2006.

[10] K. Saho, "Kalman filter for moving object tracking: Performance analysis and filter design," in *Kalman Filters*, G. L. d. O. Serra, Ed. Rijeka: IntechOpen, 2017, section: 12. Available: https://doi.org/10.5772/intechopen.71731

[11] R. Cristi, "Linear systems," Lecture notes in Linear Systems, Dept. of Electrical Eng., Naval Postgraduate School, Monterey, CA, USA, fall 2021.

[12] X. Yun, "Optimal estimation," Lecture notes in Optimal Estimation, Dept. of Electrical Eng., Naval Postgraduate School, Monterey, CA, USA, winter 2022.

[13] R. Romero, "Radar fundamentals," Lecture notes in Radar Fundamentals, Dept. of Electrical Eng., Naval Postgraduate School, Monterey, CA, USA, summer 2021.

[14] M. Agatonović, Z. Stankovic, N. Doncov, L. Sit, B. Milovanović, Tomas, and T. Zwick, "Application of artificial neural networks for efficient high-resolution 2D DOA estimation," *Radioengineering*, vol. 21, pp. 1178–1186, Dec. 2012.

[15] *MATLAB version 9.10.0.1613233 (R2021a)*, The Mathworks, Inc., Natick, Massachusetts, 2021.

[16] Z. Peng and W. Li, "Two-stage 3D codebook design and fast beam search algorithm for millimeter-wave massive MIMO systems," *Electronics*, vol. 9, no. 2, p. 302, Feb. 2020, number: 2 Publisher: Multidisciplinary Digital Publishing Institute. Available: https://www.mdpi.com/2079-9292/9/2/302

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California