2020-06

# ADAPTIVE BEAMSTEERING COGNITIVE RADAR WITH INTEGRATED SEARCH-AND-TRACK OF SWARM TARGETS

Johnson, Zachary W.

Monterey, California. Naval Postgraduate School


# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**ADAPTIVE BEAMSTEERING COGNITIVE RADAR WITH INTEGRATED SEARCH-AND-TRACK OF SWARM TARGETS**

by

Zachary W. Johnson

June 2020

Thesis Advisor: Ric Romero
Second Reader: David C. Jenn

THIS PAGE INTENTIONALLY LEFT BLANK

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



| **1. AGENCY USE ONLY** *(Leave blank)* | **2. REPORT DATE** June 2020 | **3. REPORT TYPE AND DATES COVERED** Master's thesis |
|---|---|---|

| **4. TITLE AND SUBTITLE** ADAPTIVE BEAMSTEERING COGNITIVE RADAR WITH INTEGRATED SEARCH-AND-TRACK OF SWARM TARGETS | **5. FUNDING NUMBERS** |
|---|---|
| **6. AUTHOR(S)** Zachary W. Johnson | |
| **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)**

Unlike traditional radar systems, cognitive radars are designed to employ a perception-action cycle to continuously adapt to their environment. Adaptive beamsteering cognitive radar (AB-CRr) systems seek to improve detection and tracking performance by formulating a beam placement strategy adapted to their environment. Rather than employing traditional raster scanning in a search-scene, AB-CRr builds a probabilistic model of the target environment that enables it to more efficiently employ its limited resources to locate and track targets. In this thesis, we investigate methods for adapting the AB-CRr framework to detect and track large target swarms. This is achieved by integrating the properties of correlated-motion swarms into both the radar tracking model and AB-CRr's underlying dynamic probability model. The results demonstrate that AB-CRr is capable of adapting its beamsteering strategy to efficiently perform resource balancing between search and tracking applications, while taking advantage of group structure and intra-swarm target correlation to resist large swarms overloading available resources.

| **14. SUBJECT TERMS** cognitive radar, adaptive beamsteering | **15. NUMBER OF PAGES** 85 |
|---|---|
| | **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.**

**ADAPTIVE BEAMSTEERING COGNITIVE RADAR WITH INTEGRATED SEARCH-AND-TRACK OF SWARM TARGETS**

Zachary W. Johnson
Ensign, United States Navy
BS, U.S. Naval Academy, 2019

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**June 2020**

Approved by: Ric Romero
Advisor

David C. Jenn
Second Reader

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

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Unlike traditional radar systems, cognitive radars are designed to employ a perception-action cycle to continuously adapt to their environment. Adaptive beamsteering cognitive radar (AB-CRr) systems seek to improve detection and tracking performance by formulating a beam placement strategy adapted to their environment. Rather than employing traditional raster scanning in a search-scene, AB-CRr builds a probabilistic model of the target environment that enables it to more efficiently employ its limited resources to locate and track targets. In this thesis, we investigate methods for adapting the AB-CRr framework to detect and track large target swarms. This is achieved by integrating the properties of correlated-motion swarms into both the radar tracking model and AB-CRr's underlying dynamic probability model. The results demonstrate that AB-CRr is capable of adapting its beamsteering strategy to efficiently perform resource balancing between search and tracking applications, while taking advantage of group structure and intra-swarm target correlation to resist large swarms overloading available resources.

THIS PAGE INTENTIONALLY LEFT BLANK

# Table of Contents

# List of Figures

# List of Acronyms and Abbreviations

| | |
|---|---|
| **AB-CRr** | adaptive beamsteering cognitive radar |
| **BAH** | beam accumulation history |
| **CRr** | cognitive radar |
| **DoD** | Department of Defense |
| **NPS** | Naval Postgraduate School |
| **PDF** | probability distribution function |
| **PSD** | power spectral density |
| **SNR** | signal to noise power ratio |
| **USG** | United States government |
| **USN** | U.S. Navy |

THIS PAGE INTENTIONALLY LEFT BLANK

# Acknowledgments

I would like to thank my thesis advisor, Dr. Romero, for his support and guidance in completing this work. This thesis would not have been possible without his investment of time and energy to provide me with the knowledge and resources necessary.

Thanks as well Dr. Fouts, Dr. Jenn, and CDR Herring for their support in making this thesis possible.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 1:
# Introduction

Unlike a traditional radar system, cognitive radar (CRr) employs a real-time transmit-receive feedback loop to adapt its resource allocation strategy to a dynamic target environment. One application of cognitive radar is adaptive beamsteering, in which a radar system develops a beam placement strategy based on prior measurements of the scene.

In applications where a radar system cannot illuminate the entire search scene with a single beam, beam illumination of each region in the scene becomes a constrained resource. Traditional beam rasterization schemes allocate an equal number of beam illuminations to each region in the search space, but depending on the application, this methodology can be far from optimal. Adaptive beamsteering cognitive radar can significantly improve both target search and tracking performance in scenarios where some prior information of the target environment is known [1].

Adaptive beamsteering cognitive radar (AB-CRr) forms a probabilistic model of the radar channel [2]. This probabilistic model is leveraged to form a dynamic beamsteering strategy. As successive measurements of the target environment provide revision of the target probability model, the beamsteering strategy changes to match the new conditions. Both works in [1] and [3] present a framework for networked radar systems employing a cooperative beamsteering strategy. In comparison, this work investigates a single AB-CRr radar system with incomplete target state measurements (i.e. the single radar cannot directly measure target motion perpendicular to the receiver). A methodology for selecting the next radar beam shape based on the underlying probability model is presented in [4]. This methodology relies on the information-theoretic binary entropy function to assign priority to regions of the probability model. In comparison, [5] introduces uncertainty functions as a generalized approach to translating the target probability model to a beamsteering strategy. This thesis work extends upon [5] by investigating the impact of uncertainty functions on target tracking performance when integrated with Kalman filter tracking.

AB-CRr systems are well suited for integrated target search and track applications, where beamforming resources must be balanced between detection and tracking requirements [6].

Prior art [1] introduces a weighted sum of new target uncertainty in the scene and active track uncertainty, both defined via binary entropy. The work in [7] develops a parameterized cost function that similarly performs resource balancing between the resource competing applications. The work in [5] introduces a third resource balancing methodology that modulates search versus tracking application resource allocation via uncertainty function design. This work uses the uncertainty function method as its primary means of balancing resources between its integrated search and tracking applications.

One promising use case for adaptive beamsteering cognitive radar (AB-CRr) is swarm target detection and tracking. While a large number of targets can overwhelm traditional radar systems, a CRr can leverage probabilistic information about the search scene to improve performance in a saturated target environment.

## 1.1 Objective

The objective of this thesis work is to expand on prior art in adaptive beamsteering cognitive radar systems and extend its application to swarm tracking scenarios. We introduce track uncertainty "hinting" feedback as a method of altering adaptive beamsteering behavior in a multi-target scenario. Further, we investigate how uncertainty function design, track uncertainty feedback, and swarm tracking methodologies impact the behavior of an AB-CRr system in the presence of large target swarms. The result of this work is a framework for cognitive radar integrated search-and-track capability compatible with large target swarms.

## 1.2 Scenario of Operation

Adaptive beamsteering cognitive radar can potentially be adapted to any radar system that employs steerable directional radar beams to probe a scene. However, for the purpose of this thesis, the scenario of operation is constrained to a single configuration. In this thesis, the radar illuminator is assumed to be airborne, scanning the ground plane below it, as depicted in Fig. 1.1, or alternatively a stationary radar illuminating a large swath of ground.

In this airborne ground-search radar scenario, it is assumed that ground clutter is the dominant noise source in the radar return signal. The radar system is also assumed to have an electronically steerable radar beam that can be controlled along two degrees of freedom:

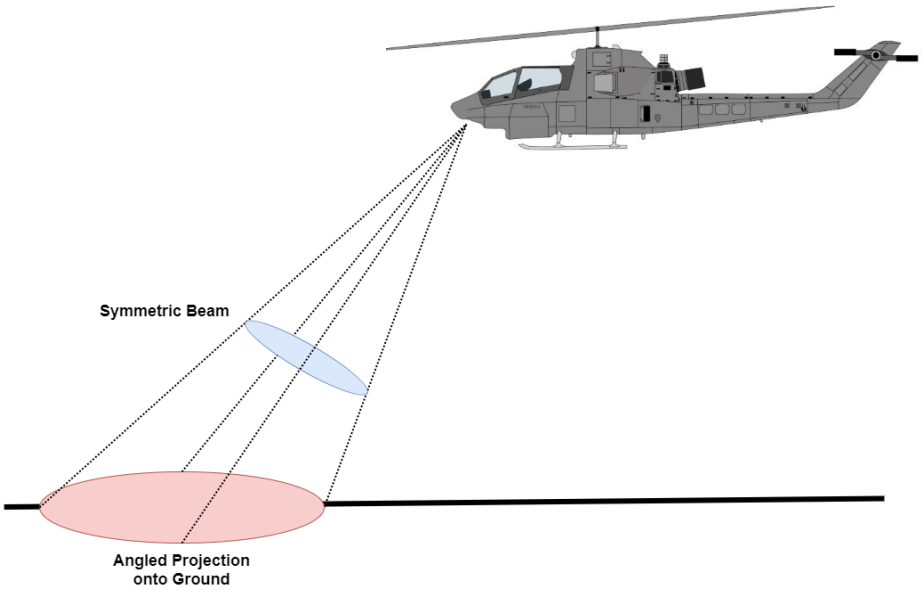


Figure 1.1. Scenario of operation: Airborne radar illuminating ground targets

elevation and azimuth. When the radar beam is projected along the ground plane, this corresponds to range-bearing coordinates relative to the ground position of the airborne radar. Additionally, the radar system is assumed to measure Doppler frequency shift information in the return signal, and therefore measures target velocity information relative to its own motion.

Further, it is assumed that there are multiple ground targets in the scene of interest moving in swarm formations. In this thesis, a swarm formation is defined as a collection of targets with highly correlated motion and arbitrary, potentially dynamic, relative spatial orientation.

## 1.3 Organization of Thesis

This thesis is organized into 7 chapters and 1 appendix. The environment model, including the discretized search space and target probability model, used by the cognitive radar system is presented in Chapter 2. A framework for cognitive adaptive beamsteering is presented in Chapter 3. Chapter 3 presents uncertainty functions and Kalman track feedback as two mechanisms for modulating beamsteering strategies for the cognitive radar system. Single target search-and-track performance is presented in Chapter 4, including the performance impacts of uncertainty function design and track feedback. In Chapter 5, the target tracking

methodology is extended to swarms in which the motion of targets is correlated and track feedback is extended to a multiple target scenario. The performance of these motion-correlated swarm tracking techniques are investigated in Chapter 6 using a range of swarms sizes and configurations. Finally, Chapter 7 draws conclusions from experimental results and provides recommendations for future research. Appendix 1 addresses efficient computation of the target probability model.

# CHAPTER 2:
# Channel Model

Cognitive radar systems seek to improve performance over traditional radar systems by exploiting information about their search environment to more efficiently employ their limited resources. The adaptive beamsteering cognitive radar (AB-CRr) architecture investigated in this thesis employs a discrete search-space probabilistic model to describe its environment. The channel model and its representation in the cognitive radar system described in this section serves as the basis of subsequent simulations and performance results.

## 2.1 Discrete Target Space Model

The geometry considered in this thesis work can be described as a 2-dimensional spatial region upon which the radar beam can be trained. Beam location in this 2-dimensional region is most readily represented via a polar coordinate system (range and azimuth) corresponding to the position of targets in the search plane relative to the radar illuminator. This polar coordinate system is easily translated to a discrete Cartesian grid representation. In this representation, the search space plane is divided into rectangular cells that each correspond to a subset of range-azimuth measurements. Each discrete cell in the grid has the Boolean attribute of "containing a target" or "not containing a target" within the continuous region of space it encloses (Fig. 2.1).

The spatial search grid is composed of $M_x$ cells and $M_y$ cells corresponding to the azimuth and elevation angular cells of the search space. The radar system measures Doppler shift in the return signal of an illuminated region, which is represented in the discrete grid by $N_y$ Doppler cells that are mapped to the continuous range of Doppler frequency shifts measured by the receiver. The resultant discrete search space representation is a Boolean matrix of size $\{M_x, M_y, N_y\}$, corresponding to a discrete representation of x-axis position, y-axis position, and target velocity (Fig. 2.2).

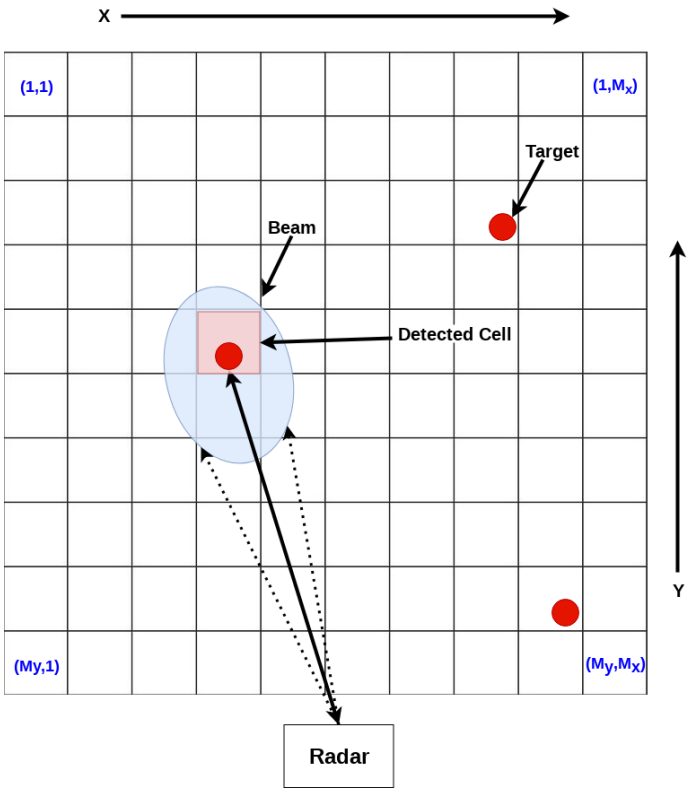


Figure 2.1. Azimuth-elevation angular measurements of a radar system projected onto a Cartesian plane

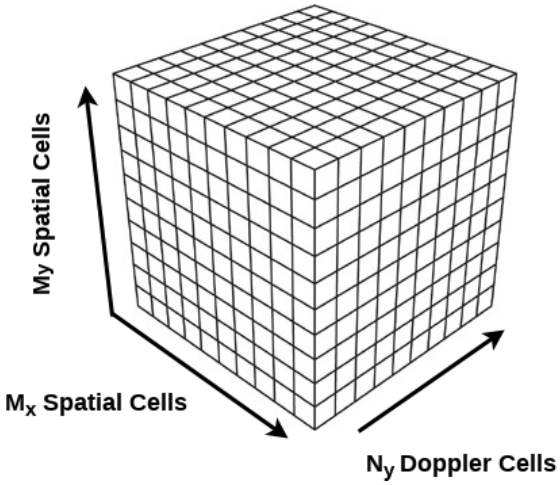


Figure 2.2. The target probability model represented as a 3D matrix of Boolean values

## 2.2 Radar Channel Model

The modelled cognitive radar system investigates its environment by radiating electromagnetic energy from a single position and measuring return energy. The radar system is assumed to employ a phased array illuminator, generating narrow beams that can be steered along an arbitrary azimuth and elevation. This work assumes a search geometry that allows the phased array system to orient its beam onto the search space at an arbitrary azimuth (which may be translated to cross-range) and elevation (which may be translated to range). The region of the search space covered by the phased array radar beam at a given time is considered to be illuminated and the region not covered by the beam is considered to be unilluminated. The radar system is assumed to measure both the magnitude of the return energy from the illuminated region as well as its Doppler shift. Therefore, information about target presence as well as the relative velocity of any targets to the radar illuminator is measured with each illumination.

### 2.2.1 Phased Array Beamsteering

**Ideal Beamsteering**

Phased array beamforming can be represented via a Kronecker product to produce a beamsteering matrix [1]

$$\mathbf{V} = \mathbf{a} \otimes \mathbf{b}_y \otimes \mathbf{c}, \tag{2.1}$$

where $\mathbf{a}$ and $\mathbf{c}$ are normalized spatial manifolds of the illuminated region and $\mathbf{b}_y$ is the temporal manifold that captures the discrete Doppler states of each spatial cell and are given by:

$$\begin{aligned} \mathbf{a} &= \frac{1}{\sqrt{M_x}} \exp\left(j2\pi k_x[0...M_x - 1]^T\right) \\ \mathbf{b}_y &= \frac{1}{\sqrt{N_y}} \exp\left(j2\pi d_y[0...N_y - 1]^T\right) \\ \mathbf{c} &= \frac{1}{\sqrt{M_y}} \exp\left(j2\pi k_y[0...M_y - 1]^T\right), \end{aligned} \tag{2.2}$$

where $k_x$ and $k_y$ are wave numbers, which are normalized (-0.5, 0.5) to correspond to the spatial region of the radar beam and $d_y$ is the normalized range of Doppler frequencies [3].

Assuming that the complex reflection of targets illuminated by the beam are zero phase and that the power of beam energy is a constant value $P_s$ along its aperture, the signal received from a beam illumination is represented by

$$\mathbf{s} = \sqrt{P_s}\mathbf{V}\mathbf{x} + \mathbf{n}, \tag{2.3}$$

where **s** is the received signal, **x** is a Boolean vector that represents the ground truth of true target locations in the cells illuminated by the beam, and **n** is noise. The vector **x** has a number of elements equal to the number of spatial cells illuminated by the beam multiplied by the number of discrete Doppler cells modelled in the discrete search space.

**Incorporating Gain of the Beam Pattern**

The beamsteering model can be further generalized to the case where beam energy is not uniform across the beam aperture. This more closely resembles a physically realizable antenna beam pattern for which beam gain is greater at the center of a formed beam than at any other portion of the beam. This non-uniform gain pattern can be represented in the discrete search grid via

$$\mathbf{P}_s = \begin{bmatrix} P_{11} & \cdots & P_{1M_x} \\ \vdots & \ddots & \\ P_{1M_y} & & P_{M_xM_y} \end{bmatrix}. \tag{2.4}$$

Observing the Kronecker matrix product presented in Eq. (2.1), **V** is equivalent to

$$\mathbf{V} = \begin{bmatrix} \mathbf{a}_{11}\mathbf{b}_{11}\mathbf{c}_{11} & \mathbf{a}_{11}\mathbf{b}_{11}\mathbf{c}_{12} & \dots & \mathbf{a}_{12}\mathbf{b}_{11}\mathbf{c}_{11} & \mathbf{a}_{12}\mathbf{b}_{11}\mathbf{c}_{22} & \dots \\ \mathbf{a}_{11}\mathbf{b}_{11}\mathbf{c}_{21} & \mathbf{a}_{11}\mathbf{b}_{11}\mathbf{c}_{21} & & \mathbf{a}_{12}\mathbf{b}_{11}\mathbf{c}_{11} & \mathbf{a}_{12}\mathbf{b}_{11}\mathbf{c}_{22} & \\ & \vdots & & \vdots & & \\ \mathbf{a}_{21}\mathbf{b}_{11}\mathbf{c}_{11} & \mathbf{a}_{21}\mathbf{b}_{11}\mathbf{c}_{12} & \dots & \mathbf{a}_{22}\mathbf{b}_{11}\mathbf{c}_{11} & \mathbf{a}_{22}\mathbf{b}_{11}\mathbf{c}_{22} & \dots \\ \mathbf{a}_{21}\mathbf{b}_{11}\mathbf{c}_{21} & \mathbf{a}_{21}\mathbf{b}_{11}\mathbf{c}_{21} & & \mathbf{a}_{22}\mathbf{b}_{11}\mathbf{c}_{11} & \mathbf{a}_{22}\mathbf{b}_{11}\mathbf{c}_{22} & \\ & \vdots & & \vdots & & \end{bmatrix}. \tag{2.5}$$

Noting that the columns of $\mathbf{a}$ and $\mathbf{c}$ represent spatial cells in the illuminated region, we see that each column of $\mathbf{V}$ belong to the same spatial cell, while the receive signal power of each column of $\mathbf{V}$ corresponds to the beam receive power

$$\mathbf{g} = \left[ [P_{11} \dots P_{1M_x)}]_1 \dots [P_{11} \dots P_{1M_x)}]_{N_y} ] \dots [P_{M_y1} \dots P_{M_yM_x)}]_1 \dots [P_{M_y1} \dots P_{M_yM_x)}]_{N_y} ] \right]. \tag{2.6}$$

It follows that the beamsteering vector $\mathbf{V}$ can be weighted by the spatial gain pattern of $\mathbf{P}_s$ by $\mathbf{G} = \text{diag}(\mathbf{g})$. The modelled target reflection then becomes

$$\mathbf{s} = (\mathbf{GV})\mathbf{x} + \mathbf{n}. \tag{2.7}$$

For computational efficiency, subsequent simulations and discussion in this work will assume a constant beam gain pattern.

### 2.2.2 Environment Noise Model

Return measurement noise is the sum of both environmental noise $\boldsymbol{e}$ and receiver noise $\boldsymbol{\omega}$,

$$\mathbf{n} = \boldsymbol{e} + \boldsymbol{\omega}, \tag{2.8}$$

where $\boldsymbol{e}$ is modelled as zero-mean Gaussian with covariance $\mathbf{C}_e$ and $\boldsymbol{\omega}$ is zero-mean Gaussian with covariance matrix $\mathbf{C}_\omega$.

Environmental noise is assumed to be dominated by return clutter and receiver noise is assumed to be dominated by internal thermal noise. Environmental clutter is initially modelled in the frequency domain [8]. In simulations, the power spectral density (PSD) of environmental clutter is modelled by a hamming window centered at zero frequency and spanning 0.33 of the normalized frequency range (Fig. 2.3).

Receiver noise is assumed to be thermal noise, so the covariance matrix of receiver noise is simply $\mathbf{C}_\omega = P_\omega \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The covariance matrix of return signal noise is then the sum of the two noise components

$$\mathbf{C}_n = \mathbf{C}_e + P_\omega \mathbf{I}. \tag{2.9}$$

In simulations, it is assumed that environment clutter noise dominates receiver thermal noise, so the return signal covariance matrix has a structure displayed in Fig. 2.4.

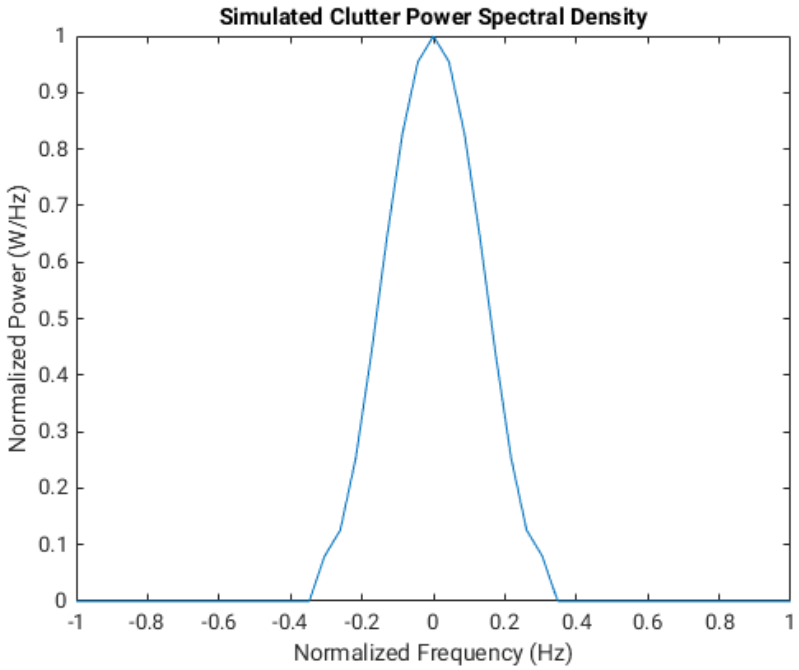


Figure 2.3. Modelled noise clutter power spectral density (PSD)

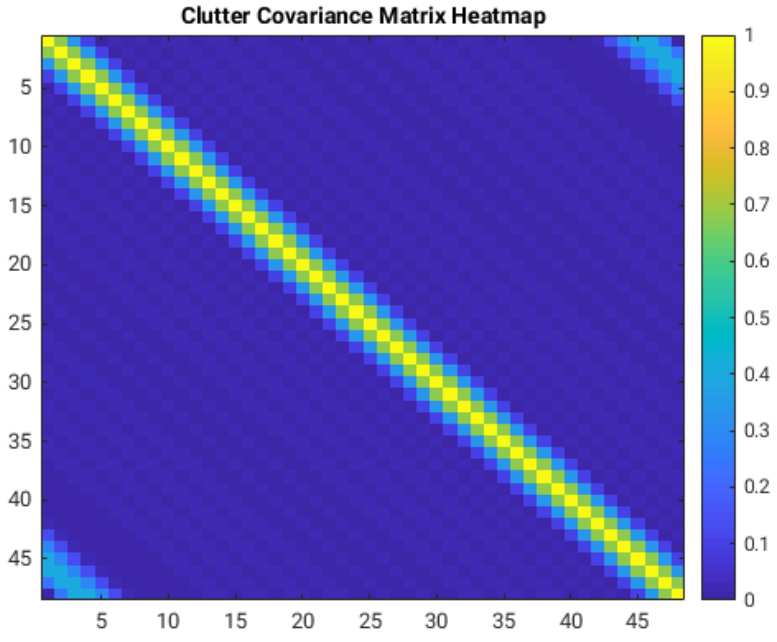


Figure 2.4. Heat map plot of simulated clutter covariance matrix

## 2.3 Target Space Probability Model

Given a discrete representation of the search space and a model for discrete space representation of beamsteering and target illumination, a discrete environment model can be formed for cognitive beamsteering. A probabilistic model is employed here to describe the properties of the search environment [1]. Each discrete cell in the search space is assigned a probability of target presence, and each iteration of the beamsteering algorithm updates the probability of each cell according to both the return signal and underlying assumptions about the target environment.

### 2.3.1 Probability Update Methodology

**Illuminated Region Update Method**

This section summarizes the recursive Bayesian probability update methodology described in [5] and implemented in this thesis' AB-CRr framework. Current and past radar measurements of a discrete target state are used to recursively define the probability of target presence via Bayes Theorem [4],

$$P(H_i|\mathbf{z}_k) = \frac{P(\mathbf{z}_{k-1}|H_i)P(H_i|\mathbf{z}_k)}{P(\mathbf{z}_k)}. \tag{2.10}$$

The sensor measurement $\mathbf{z}_k$ is for iteration $k$. Hypothesis $H_i$ is a member of the set of possible permutations of targets in the cells under illumination. Assuming that a maximum of one target is present in each cell, then there are $2^M$ such hypothesis permutations, where $M$ is the total number of cells illuminated by the radar beam. $M$ is equivalent to the number of spatial cells encompassed by the radar beam multiplied by the number of Doppler frequency cells in the scene.

Thus each return measurement hypothesis can be formed as

$$
\begin{aligned}
H_0 &: \mathbf{z} = \mathbf{n} \\
H_1 &: \mathbf{z} = \mathbf{s}_1 + \mathbf{n} \\
H_2 &: \mathbf{z} = \mathbf{s}_2 + \mathbf{n} \\
H_3 &: \mathbf{z} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n} \\
&\vdots \\
H_{2^M-1} &: \mathbf{z} = \mathbf{s}_1 + \mathbf{s}_2 + \ldots + \mathbf{s}_M + \mathbf{n},
\end{aligned}
\tag{2.11}
$$

where $\mathbf{n}$ is channel noise and $\mathbf{s}_i$ is the return signal anticipated from the $i^{th}$ cell illuminated by the beam. The number of permutations grows exponentially with the number of target state cells encompassed by the beam. If the number of targets that can appear in the illuminated region is limited to $r$, however, then the number of hypotheses that must be computed is reduced to [2]

$$
N = \sum_{k=0}^{r} \binom{M}{k}. \tag{2.12}
$$

Note that when the maximum number of targets $r$ is equal to $M$ (the total number of cells illuminated by the beam), Eq. (2.12) becomes a special case of the binomial series [9],

$$
N = \sum_{k=0}^{M} \binom{M}{k} = \sum_{k=0}^{M} \binom{M}{k} 1^k = 2^M, \tag{2.13}
$$

and the total number of possible permutations matches Eq. (2.11).

Considering that received signal $\mathbf{z}_k$ is a jointly Gaussian noisy signal with covariance matrix $\mathbf{C}_\mathbf{z}$, the conditional joint probability distribution is given by

$$
P(\mathbf{z_k}|H_i) = \frac{1}{\pi^k |\mathbf{C_z}|} \exp\left(\mathbf{s}_i - \mathbf{z}_k\right)^H \mathbf{C}_\mathbf{z}^{-1} (\mathbf{s}_i - \mathbf{z}_k). \tag{2.14}
$$

The prior probability of a target hypothesis can be computed from cell target probabilities via

$$P(\mathbf{z}_{k-1}|H_i) = \prod_{c=1}^{M} (P_{c,k-1})^{b_c} (1 - P_{c,k-1})^{1-b_c}, \tag{2.15}$$

where $b_c$ is either 1 or 0 corresponding to whether hypothesis $H_i$ includes a target present in cell $c$. $P(\mathbf{z}_k|H_i)$ and $P(\mathbf{z}_k)$ do not have to be directly computed, rather, the ratio of the two values can be computed via

$$\frac{P(\mathbf{z}_k|H_i)}{P(\mathbf{z}_k)} = K^{-1} \exp{(\mathbf{s}_i - \mathbf{z}_k)^H \mathbf{C}_\mathbf{z}^{-1} (\mathbf{s}_i - \mathbf{z}_k)}, \tag{2.16}$$

where $K$ is a normalization factor equal to the sum of each hypothesis joint probability,

$$K = P(\mathbf{z}_k) = \sum_{i=0}^{2^M - 1} P(\mathbf{z}_k|H_i). \tag{2.17}$$

The updated cell probabilities can be computed by summing each hypothesis that includes a target present in the cell $c$,

$$P = \sum_{c=1}^{M} b_c P(\mathbf{z}_k|H_i). \tag{2.18}$$

**Unilluminated Region Update Method**

With each illumination of the radar system, time elapses and the majority of cells in the search grid are not illuminated. Therefore, the uncertainty of these unilluminated cells grows during the period of non-observation. In the discrete case, with each iteration of beam selection, the uncertainty of cells that are not illuminated by the beam will be incremented by some value. As the system progresses through iterations, the uncertainty of unilluminated cells grows until the adaptive beamsteering algorithm illuminates that cell and updates its corresponding uncertainty value.

One of the greatest advantages of a cognitive radar approach towards search optimization is that prior knowledge of the target environment can be readily integrated into the scenario, impacting the beamsteering behavior of the radar accordingly. In a simple forward searching radar scenario, for example, it may be reasonable to assume that targets are most likely to enter into the search space from the edges of the grid. Translated into terms of search grid uncertainty: with each successive time step, the chance that a target appears or disappears from the edges of the map is greater than anywhere else. This assumption about target behavior can be integrated into the cognitive radar probability model by implementing non-uniform uncertainty growth across the probability cells of the search grid.

For example, the uncertainty increment value used on a given cell can be defined to be the Euclidean distance of spatial position of a cell to the center of the search grid [1].

One problem with this methodology is that there is no uncertainty growth at the center of the search grid. To overcome this limitation, a generalized probability update increment is introduced:

$$\Delta U_{i,j,k} = \alpha(1 + \sqrt{2}\beta|| < \mathbf{k}_{x_i}, \mathbf{k}_{y_j} > ||_2), \tag{2.19}$$

where $\Delta U_{i,j,k}$ is the uncertainty increment for a given cell, $\alpha$ is a coefficient that defines the overall rate of uncertainty growth, and $\beta$ relates the relative significance of Euclidean distance from the center of the search space to the baseline growth rate. Because the normalized spatial ranges of the search scene are (-0.5, 0.5), the maximum normalized distance from the center is $0.5\sqrt{2}$. When $\beta = 0$, uncertainty growth is uniform across all cells. When $\beta >> 1$, uncertainty growth behaves as described in Fig. 2.5.

### 2.3.2 Probability Map Initialization

There are multiple possible models for how to initialize the 3-dimensional probability map of the cognitive radar system. One implementation is to assume that no targets exist in the search space at the onset of operation. This assumption, however, initializes the scene with zero uncertainty and leaves the cognitive radar beam selection algorithm to employ an arbitrary beam illumination pattern until an uncertainty model for unilluminated cells begins to dominate steady state behavior.

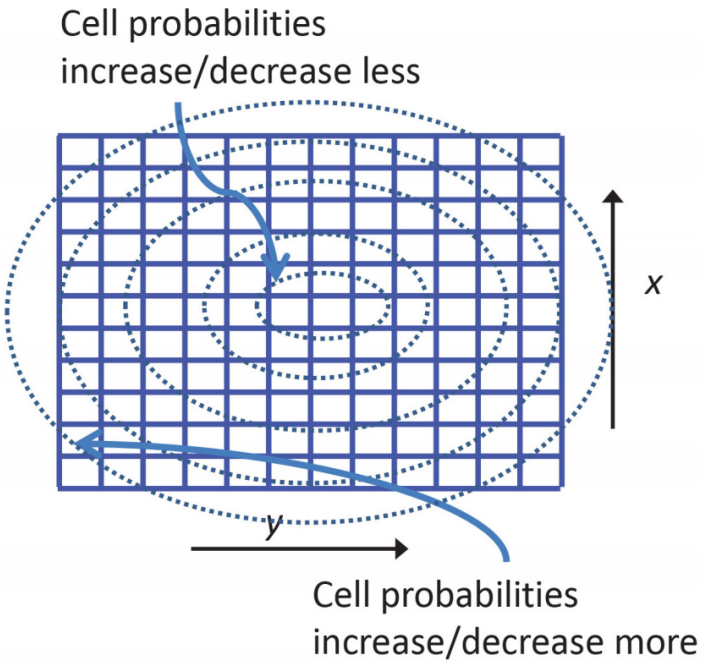


Figure 2.5. Non-uniform uncertainty growth model. Source: [1].

A second option is to assume that a finite number of targets already exist in the search space, and distribute the cell probabilities uniformly based on this assumption. For example, if one target is assumed to be present in the search area and there are 100 cells, each cell might be assigned an initial probability of 0.01, such that the sum of all the cells equals 1. If some prior knowledge of the environment is assumed, then it can be applied to provide greater initial probability to cells that are known to be more likely to initially contain a target (such as the edges of a map).

A third option is to arbitrarily initialize the probability of each cell to 0.5. As a result, at the start of operation, the cognitive radar assigns a high uncertainty value to each unilluminated cell. As such, the initial behavior of the system is biased towards illuminating each cell at least once before settling into some kind of steady state search behavior. In this implementation, the third option is selected.

# CHAPTER 3:
# Adaptive Beamsteering Cognitive Radar

The cognitive radar system presented in this thesis employs the probability model presented in the previous section to iteratively select its next beam orientation. The pattern of beam placement as the system operates over time impacts the underlying probability model, as regions that are illuminated more often will have more extreme target probabilities associated with greater confidence of their presence or absence. In this section the cognitive radar perception-action feedback loop is described and design considerations are discussed.

## 3.1 Uncertainty Mapping of Probability Model

In order for the 3-dimensional probability model to be employed in a beam selection strategy, probability values are mapped to a less rigidly defined metric of "uncertainty" [5]. In this context, uncertainty is an arbitrary cost metric which describes the significance of a given cell probability to the cognitive radar overall assessment of performance. The adaptive beamsteering algorithm seeks to minimize the cumulative uncertainty metric of the entire scene model, so the selection of an uncertainty metric drives the overarching behavior of the cognitive radar beamsteering strategy.

### 3.1.1 Uncertainty Metric

The discrete scene probability model is encapsulated in a 3-dimensional matrix composed of probabilities ranging from 0 to 1, where 1 is a 100% probability of target presence in the associated cell. An uncertainty function maps probability values to uncertainty values between 0 and 1 [5]. An uncertainty value of 0 contributes no cost to the cumulative uncertainty of the system, while an uncertainty value of 1 is the maximum contribution a cell can contribute. A well known example of an uncertainty function is the information-theoretic binary entropy function. The binary entropy function maps probability values to values of information entropy, with maximum entropy at a probability of 0.5.

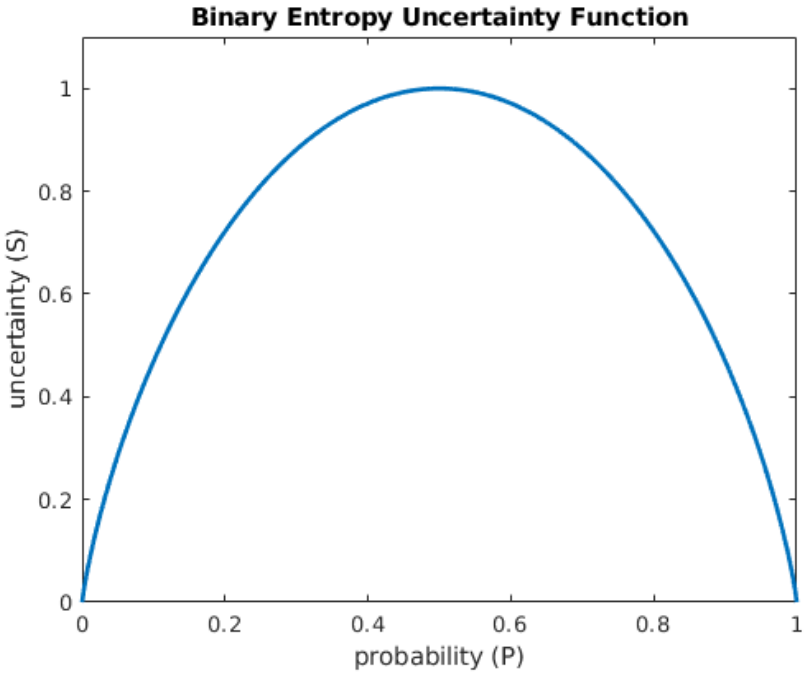


Figure 3.1. Information-theoretic binary entropy function

### 3.1.2 Uncertainty Function Design

Provided that a AB-CRr beamsteering strategy will seek to minimize the cumulative uncertainty of the search scene over time, a series of inferences can be made about the impact of an uncertainty function on the resultant beamsteering strategy.

First, the beam selection algorithm will orient its next beam at the spatial region associated with the highest contribution of uncertainty. The assumption here is that illuminating a cell with high uncertainty will eventually reduce its uncertainty. For this assumption to be true, a well formed uncertainty function should output an uncertainty of 0 at input probabilities 0 and 1. As a consequence, minimum uncertainty should exist at the deterministic extremes of probability. This is satisfied by the binary entropy function.

Second, if illuminating a cell results in an increase in the corresponding uncertainty value for the cell, the beam selection algorithm will continue to illuminate that cell until the uncertainty begins to drop. This observation assumes that the cell in question is illuminated because its region contributed the most uncertainty in the scene and that the unilluminated region uncertainty growth is much slower than the illuminated region uncertainty change.

As a consequence, the beam selection algorithm can be expected to continue to illuminate a cell as long as the derivative of the uncertainty function at its instantaneous probability value is positive. Therefore, if the desired beam selection behavior is to continue to illuminate an uncertain true-target cell until a threshold probability for decision is reached, then the global maximum of the uncertainty function should be placed at or near the desired or required probability of detection (i.e. the probability threshold for detection).

Given these design constraints, three uncertainty functions [5] are investigated alongside the binary entropy function:

$$S_1(P) = \begin{cases} 1 + \alpha^{-1}(P - \alpha) & P \leq \alpha \\ 1 + (1 - \alpha)^{-1}(P - \alpha) & P > \alpha \end{cases}, \tag{3.1}$$

$$S_2(P) = \begin{cases} 1 + \alpha^{-2}(P - \alpha)^2 & P \leq \alpha \\ 1 + (1 - \alpha)^{-2}(P - \alpha)^2 & P > \alpha \end{cases}, \tag{3.2}$$

$$S_3(P) = \begin{cases} 0 & P = 0 \\ \frac{1-P}{1-\alpha}\exp\left(\frac{P-\alpha}{1-\alpha},\right) & P \neq 0 \end{cases}. \tag{3.3}$$

The variable $\alpha$ indicates the probability value at which the uncertainty function has its local maximum. $S_1(P)$ is a triangular piece-wise function, referred to here as the 1st order Pyramid function (Fig. 3.2). $S_2(P)$ is a quadratic innovation of $S_1(P)$ and will be referred to as the 2nd order Pyramid function (Fig. 3.3). $S_3(P)$ is a reversed Chi-squared distributions with 4 degrees of freedom scaled to a maximum value of one (Fig. 3.4). These three functions were selected for their suitability based on the inferences made above.

### 3.1.3 Target Scene to Beamsteering Scene Mapping

Upon transforming the target probability matrix into uncertainty values, the next beam location is selected by locating the beam position that encompasses the greatest cumulative amount of cell uncertainty. Note that the uncertainty matrix $\boldsymbol{U}$ has dimensions $M_x \times M_y \times N_y$ while the beamsteering search plane has dimensions $M_x \times M_y$. The uncertainty matrix can

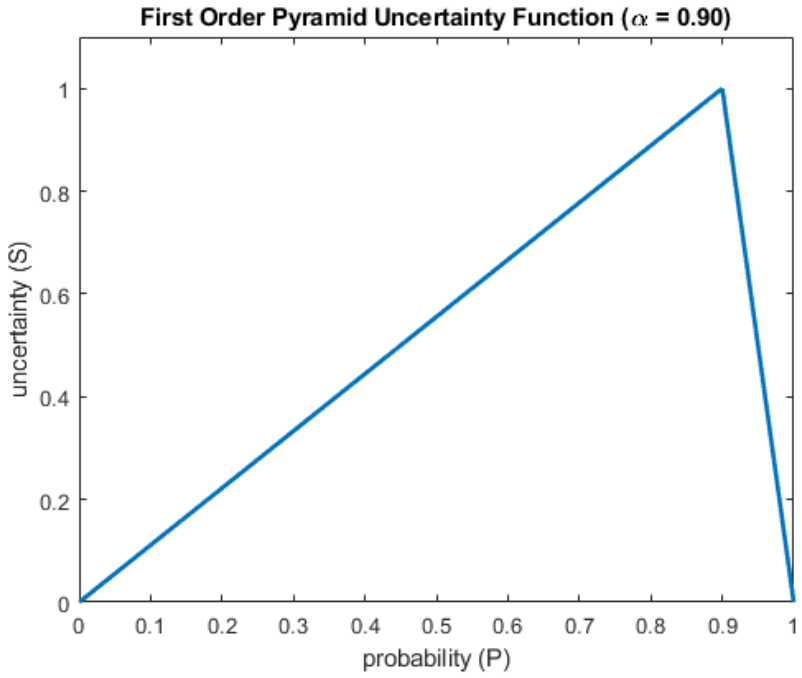


Figure 3.2. First order pyramid function

be projected to $M_x \times M_y$ by computing the mean uncertainty value for each vector of Doppler cells at each spatial index of $\mathbf{U}$. Thus the 2-dimensional uncertainty matrix is given by,

$$\mathbf{Q}_{j,k} = \frac{1}{M_d} \sum_{l=0}^{M_d-1} \mathbf{U}_{j,k,l}, \tag{3.4}$$

where $\mathbf{Q}$ is the $M_x \times M_y$ projection of $\mathbf{U}$ and $l$ is the Doppler cell index.

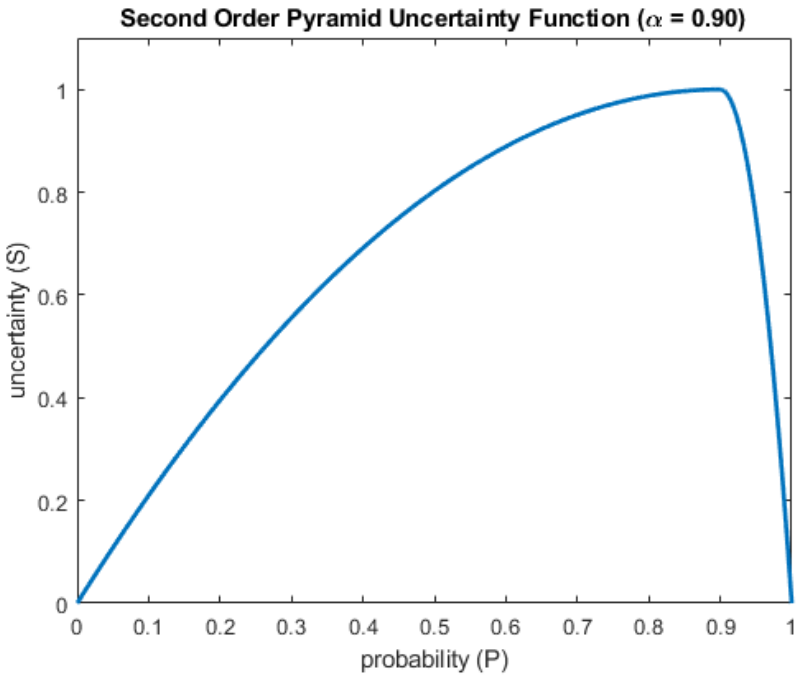


Figure 3.3. Second order pyramid function

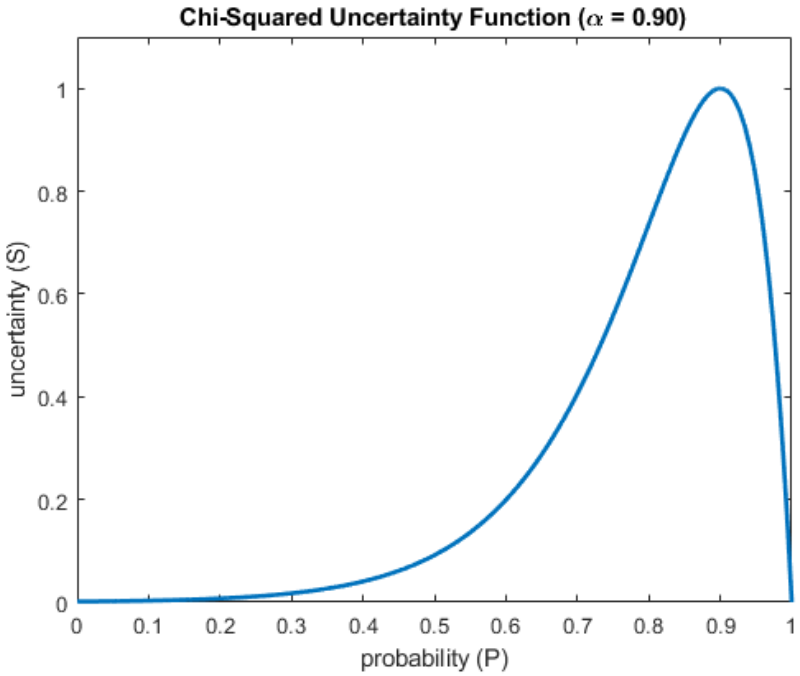


Figure 3.4. Chi-squared uncertainty function

## 3.2 Adaptive Beamsteering

A cognitive radar beamsteering strategy will seek to minimize the cumulative value of the matrix $\mathbf{Q}$. A simple methodology for minimizing the cumulative value of $\mathbf{Q}$ is to select a beam at each iteration of radar illumination that is anticipated to reduce the cumulative uncertainty more than any other beam configuration. It is assumed that the cognitive radar operates at a reasonable SNR such that the uncertainty of any cell illuminated by the radar beam becomes small. Then, the condition for the optimal next beam is

$$(\mathbf{U}_{k+1}|\mathbf{B}_k^*) \geq (\mathbf{U}_{k+1}|\mathbf{B}_k), \tag{3.5}$$

where $\mathbf{B}^*$ is the optimal beam selection and $\mathbf{B}$ is the set of all possible beam configurations in the search scene.

If the shape of the beam is assumed to be rectangular and smaller than the search scene, then the optimal next beam can be easily found via a 2-dimensional convolution:

$$< j, k >_{\mathbf{B}^*} = \arg\max_{j,k} \tilde{\mathbf{B}} *_{(2)} \mathbf{Q}, \tag{3.6}$$

where $< j, k >$ are the x and y discrete cell indices of the lower corner of the optimal beam, $\tilde{\mathbf{B}}$ is a $N \times M$ matrix of ones with the dimensions of the radar beam, and $\mathbf{Q}$ is the spatial uncertainty matrix.

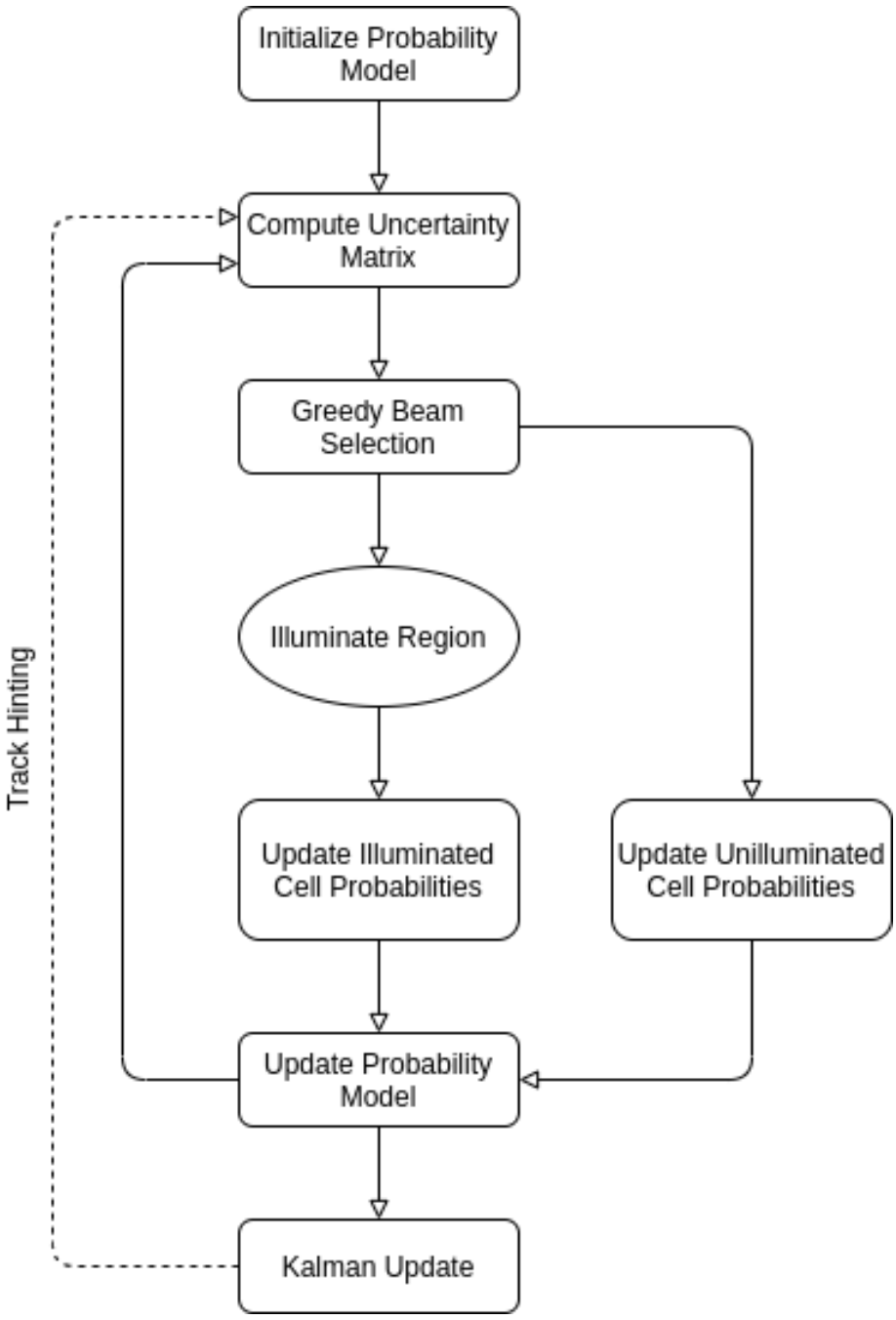


Figure 3.5. Adaptive beamsteering cognitive radar algorithm

## 3.3 Kalman Tracking Integration

The AB-CRr framework presented in this thesis has the dual functions of target search/detection and target tracking. A Kalman filter is used to track a detected target through the search scene. The Kalman filter estimates the constant motion state of a target, represented by the state vector

$$\hat{\mathbf{x}} = \begin{bmatrix} x \\ v_x \\ y \\ v_y \end{bmatrix}. \tag{3.7}$$

Target measurements by the cognitive radar are discrete space, while the Kalman filter estimates the target space in continuous space. This is accounted for in the design of Kalman filter parameters.

### 3.3.1 Initializing Kalman Model

The Kalman filter is initialized by the first measurement of a detected cell. Estimate covariance and measurement covariance are approximated to account for the error incurred by discrete measurements of the target state.

**Initial State**

A target is said to be "detected" when a cell of the search space exceeds a threshold probability of target presence. In the single target case, the first cell to reach this target probability immediately after illumination by a beam is used to initialize the target state estimate. Each cell in the search scene corresponds to a measurement $\hat{\mathbf{z}}$, which represents the measured target state at the center of the discrete region. As the search scene has three dimensions, the detected target cell map measurement of continuous space-related parameters is given by

$$\hat{\mathbf{z}} = \begin{bmatrix} x \\ y \\ v_y \end{bmatrix}. \tag{3.8}$$

As the initial measurement contains no information about the x-axis velocity of the detected target, it is initially assumed to be zero and the target state estimate is initialized as

$$\hat{\mathbf{x}}_i = \begin{bmatrix} x_i \\ 0 \\ y_i \\ v_{y_i} \end{bmatrix}. \tag{3.9}$$

**Measurement Covariance**

The dominant factor in measurement error is assumed to be quantization error of the search space. If the x-axis of the search space has a normalized range of $(-0.5, 0.5)$ and there are $N$ discrete values that are evenly spaced, then each discrete cell encloses a continuous range of normalized values $\frac{1}{N}$ wide. Given that a target exists within a detected cell, the probability of the true target parameter in this range is uniformly distributed. However, the Kalman filter assumes that the measurement noise is normally distributed. As an approximation, the standard deviation of each measurement axis is said to be equal to the quantization error of the discrete model. Each axis is said to be independent of the other axes, yielding a measurement covariance matrix

$$\mathbf{R} = \begin{bmatrix} \frac{1}{M_x^2} & 0 & 0 \\ 0 & \frac{1}{M_y^2} & 0 \\ 0 & 0 & \frac{1}{N_y^2} \end{bmatrix}. \tag{3.10}$$

**Target Covariance**

The target is assumed to be maneuvering with a stochastic motion model. Only constant motion is modelled in the target state, but acceleration is modelled with white Gaussian variance in the target velocity. This is represented with the target state covariance matrix

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma_a^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_a^2 \end{bmatrix}. \tag{3.11}$$

**State Estimate Covariance**

The initial state estimate covariance is initialized as

$$\mathbf{P} = \begin{bmatrix} \frac{1}{M_x^2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{M_y^2} & 0 \\ 0 & 0 & 0 & \frac{1}{N_y^2} \end{bmatrix}. \tag{3.12}$$

Given that x-axis velocity can range from -0.5 to 0.5 in the normalized range, the normal approximation variance of x-axis measurement is therefore initialized with a value of 1.

### 3.3.2 Track Update

Using the probabilistic model, a target detection is defined as any cell whose target probability crosses above the detection threshold in a given illumination iteration. In the single target case, it is assumed that every target detection associates with the active target track. Therefore, for every iteration of beam selection with a valid target detection, the standard Kalman update procedure is applied with a constant observation matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.13}$$

In the case where there is no target detection over one radar illumination cycle, the observation matrix is null, and only state vector and estimate covariance prediction are performed.

## 3.4 Track Feedback Hinting

The target track state estimate can be used as feedback into the adaptive beamsteering uncertainty model in order to improve tracking performance. After updating the probability map (3-D matrix) following a beam illumination and computing its associated uncertainty matrix, a small uncertainty value $\gamma$ is incremented to the cell corresponding to the target state estimate. Over numerous iterations, the uncertainty value of the cell associated with the estimated state of the target ($\hat{\mathbf{x}}$) grows faster than the rest of the map and causes the adaptive beamsteering algorithm to illuminate the estimated position, effectively verifying the estimate of the Kalman filter with an observation. In this way, track hinting can introduce a bias to target tracking into the adaptive beamsteering algorithm. Fig. 3.5 illustrates the integration of track feedback into the AB-CRr framework.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 4:
# Single Target Search-and-Track Performance

In this chapter, the integrated search-and-track performance of an AB-CRr system is modelled in simulation for single target scenarios. The impacts of uncertainty function design and track feedback hinting are investigated across a range of radar channel conditions. Observations and conclusions drawn from single target scenarios will influence design choices for swarm target cases, while providing a simplified model for characterizing adaptive beamsteering behavior.

## 4.1 Performance Metrics

Single target search-and-track performance can be quantified or visualized in multiple ways. This section presents performance metrics used to assess the effectiveness of various AB-CRr configurations.

### 4.1.1 Track Performance

Target tracking performance can be readily quantified as the mean Euclidean distance between the true and estimated target state, given as

$$\epsilon_t = \frac{1}{N_i - k_0} \sum_{k=k_0}^{N_i-1} ||\mathbf{x}_k - \hat{\mathbf{x}}_k||_2, \tag{4.1}$$

where $N_i$ is the total number of beam iterations by the radar system and $k_0$ is the first beam iteration after a target track file has been initialized.

### 4.1.2 Search Performance

Search performance is more difficult to quantify in the single target scenario. In general terms, the more beam iterations that the cognitive radar system allocates to regions of the search space that do not contain an active target track, the more it is biased towards search/detection of potential new targets. Therefore, search/detection performance can be

expected to be inversely related to the proportion of beam iterations directed on target tracks. If $N_i$ is the total number of beam iterations and $N^+$ is the number of beam iterations that were directed on an active target track, then this relationship can be expressed as

$$R_d = \frac{N^+}{N_i}. \tag{4.2}$$

The metric $R_d$ is referred to in this thesis as the target track dwell ratio, or simply dwell ratio.

Search/detection versus target tracking resource allocation can also be visualized by a beam accumulation history plot. A beam accumulation history (BAH) plot counts the total number of illuminations of each spatial cell in the search scene and displays this cumulative count as a heat map graph. When the adaptive beamsteering cognitive radar is biased to prioritize target tracking, beam illuminations will be concentrated along the target path of motion. Conversely, when the AB-CRr is biased towards target search/detection, the illumination pattern will approximate the unilluminated cell probability growth pattern described in Chapter 2.

### 4.1.3 Composite Performance Metric

The mean track error performance metric has the limitation that it does not account for the duration of time that the cognitive radar searches the scene prior to detecting the target and establishing a track. A more holistic performance metric adds a cost for each beam iteration prior to initialization of a target track. The relative significance of the target detection latency versus tracking error should be application specific and reflect the relative importance of search/detection versus tracking performance. This modified metric is given by

$$\epsilon_c = \frac{1}{N_i}\Big(\rho k_0 + \sum_{k=k_0}^{N_i-1} ||\mathbf{x}_k - \hat{\mathbf{x}}_k||_2\Big), \tag{4.3}$$

where $\rho$ is a user-defined coefficient that weights the relative contribution of detection latency compared to mean track error. In this thesis work, $\rho$ is assumed to be equal to 2.

## 4.2 Comparison of Uncertainty Functions

In Chapter 3, uncertainty functions were introduced as the primary user defined characteristic that impacts the beamsteering strategy of an adaptive beamsteering system. In this section, we consider the relative performance of four uncertainty functions introduced in the previous section. The mean track error, dwell rate, and cumulative error of a single simulated target are presented herein. The mean performance metrics of a Monte Carlo simulation with 1000 iterations for each configuration are recorded for each of the four uncertainty functions under test for a range of SNR values.

In each Monte Carlo trial, a single target enters the search scene from the top left corner and moves south-east through the scene for 1000 beam iterations. The discrete measurement model for each trial has the dimension $M_x = M_y = 20$ and $N_y = 7$ for a total of 2800 measurement states. For the three parameterized uncertainty functions, $\alpha = 0.88$ is used. The velocity of each target varies normally with each beam iteration to model unknown maneuvering acceleration, while maintaining constant mean velocity.

In Fig. 4.1, mean track error is reported for all four uncertainty functions. Observe that the 1st order pyramid $S_1(P)$, 2nd order pyramid $S_2(P)$, and binary entropy functions all have approximately the same mean track error, while the Chi-squared uncertainty function $S_3(P)$ maintains a much lower mean track error at all SNR conditions. In low SNR, the Chi-squared uncertainty function performs significantly better than the other three uncertainty functions. In high SNR, the performance discrepancy is less pronounced, but the Chi-squared uncertainty function still maintains the lowest mean track error.

In Fig. 4.2, mean dwell ratio for all four uncertainty functions is reported for a range of SNR conditions. Note that the Chi-squared uncertainty function maintains a much higher track dwell ratio than the other three uncertainty functions, meaning that it allocates much more beam time to the target track than the other three uncertainty functions. This corresponds to the observation that the Chi-squared function maintains a much lower track error.

Finally, the composite error, defined in Eq. (4.3), of all four uncertainty functions is reported in Fig. 4.3. As with Fig. 4.1, the Chi-squared uncertainty function outperforms the other three uncertainty functions in terms of composite error.

The principal observation from this performance comparison of uncertainty functions is

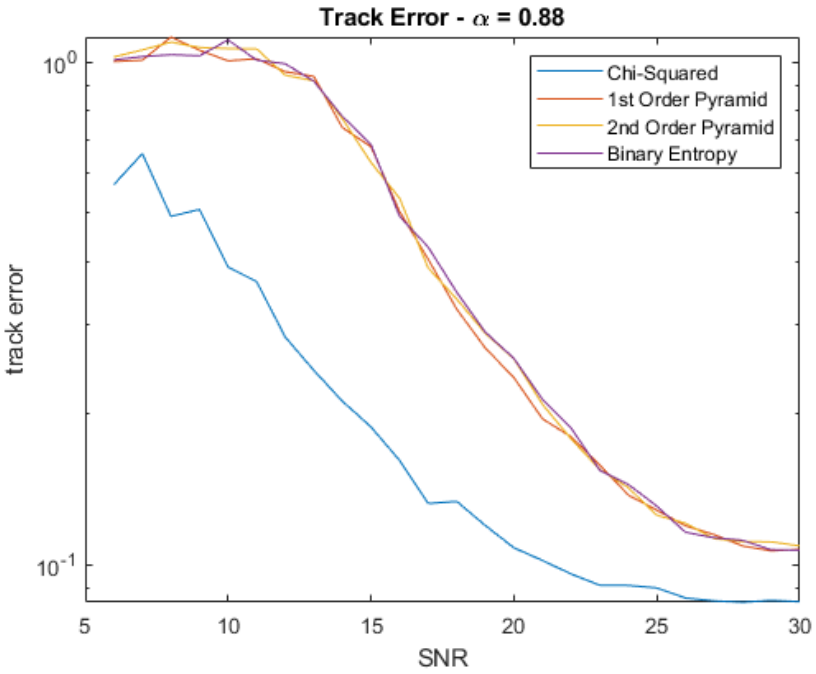


Figure 4.1. Target track error for four proposed uncertainty functions

that different uncertainty functions result in differences in beamsteering behavior. The binary entropy, 1st order pyramid, and 2nd order pyramid uncertainty functions maintain very constant track dwell ratios across a range of SNRs, while the Chi-squared uncertainty function dwell ratio is sensitive to SNR. As a consequence, with all other parameters held constant, the Chi-squared uncertainty function is a more effective uncertainty function for target tracking as it tends to "chase" and dwell on regions of high target probability, following tracks across the scene, with correspondingly lower track error.

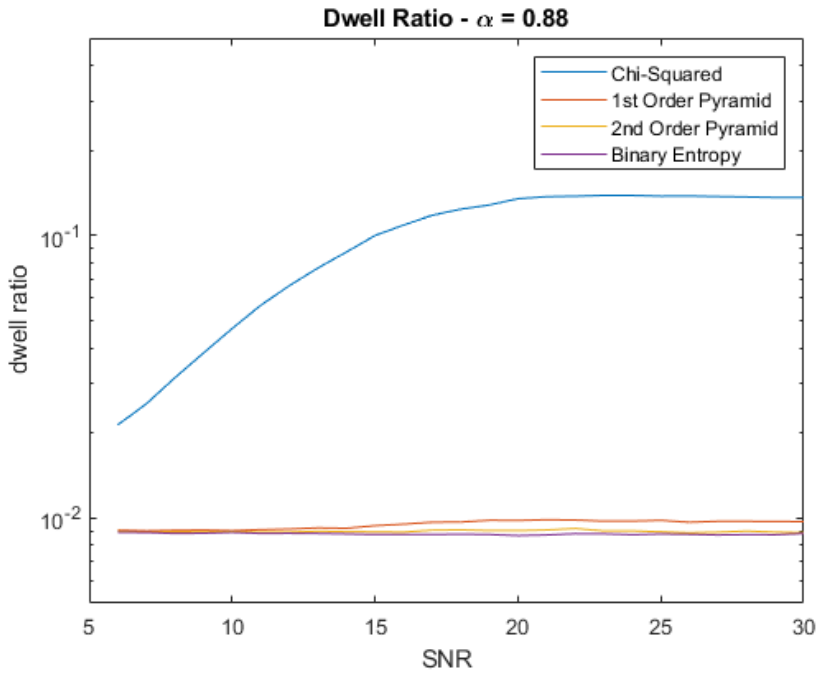


Figure 4.2. Target track dwell ratios for four proposed uncertainty functions

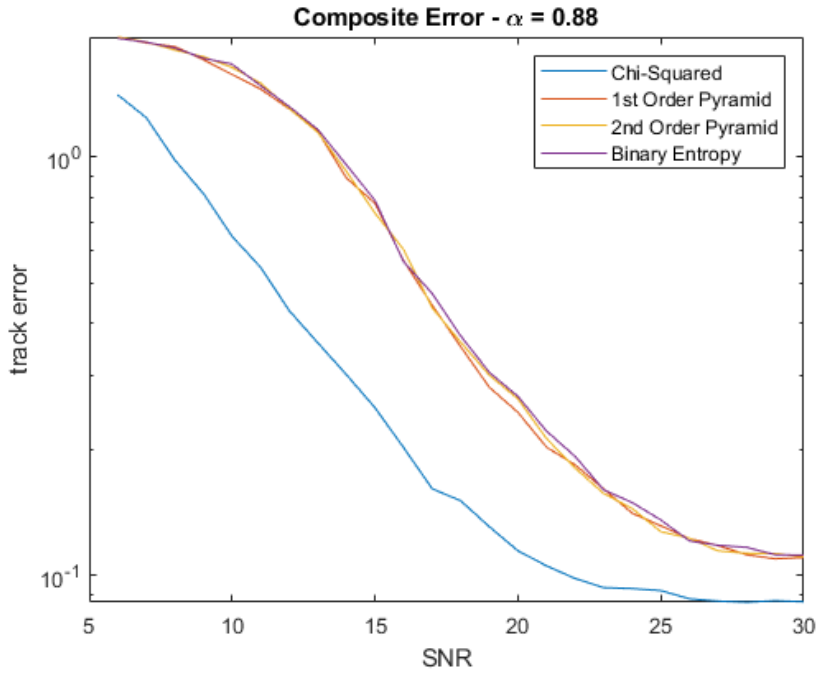


Figure 4.3. Target composite error for four proposed uncertainty functions

## 4.3 Parameterization of Chi-squared Function

To further explore the behavior of the Chi-squared uncertainty function defined in Eq. (3.3), its performance metrics are generated via Monte Carlo simulation across a range of $\alpha$ values. Fig. 4.4 shows the average composite error of single target moving through the scene at a relative heading of 135° for 1000 iterations at a given SNR and $\alpha$ value. Each simulation configuration ($\alpha$ value and SNR condition) is performed 500 times in a Monte Carlo fashion to compute average error values. The target detection threshold in this simulation is set to $P = 0.88$ and track hinting is set to $\gamma = 0.05$.

Observing Fig. 4.4, note that the composite error given a particular SNR is relatively constant across $\alpha$ values between 0.5 and 0.85. Similarly, values of $\alpha$ between 0.90 (the target detection threshold) and 0.99 result in relatively constant performance above approximately 18 dB SNR. This region approaches the maximum performance of the tracking system, where any addition track biasing ($\alpha$) or SNR improvement will not reduce the error of the track. $\alpha$ parameter values between 0.85 and 0.90 are a transition region between the two steady state error regions at high SNR. At lower SNRs, the mean composite error improves rapidly with $\alpha$ values above 0.85.

Observing Fig. 4.5, note that the dwell ratio increases rapidly in the transition region $\alpha = [0.85, 0.90]$. This corresponds to the improvement in composite track error observed in Fig. 4.4. Also note that the dwell ratio remains relatively constant for $\alpha$ parameter values across a range of SNR values. Therefore, the $\alpha$ parameter of the Chi-squared uncertainty function controls the dwell ratio of the cognitive radar beamsteering strategy.

As the dwell ratio increases, composite track error (Fig. 4.4) tends to decrease. This effect is most significant at low SNR values. At low SNR values, there is greater observation error associated with each illumination, which results in less definitive estimates of target probability. When this integrates with the recursive Bayesian cell update methodology (Section 2.3.1), the result is much slower "transient response" to a change in target state (discussed in the appendix). In the low SNR case, target detection often requires multiple beam illuminations of the same target cell to reach the target detection probability threshold. A cognitive radar system biased towards a high target dwell ratio will perform significantly better in this environment.

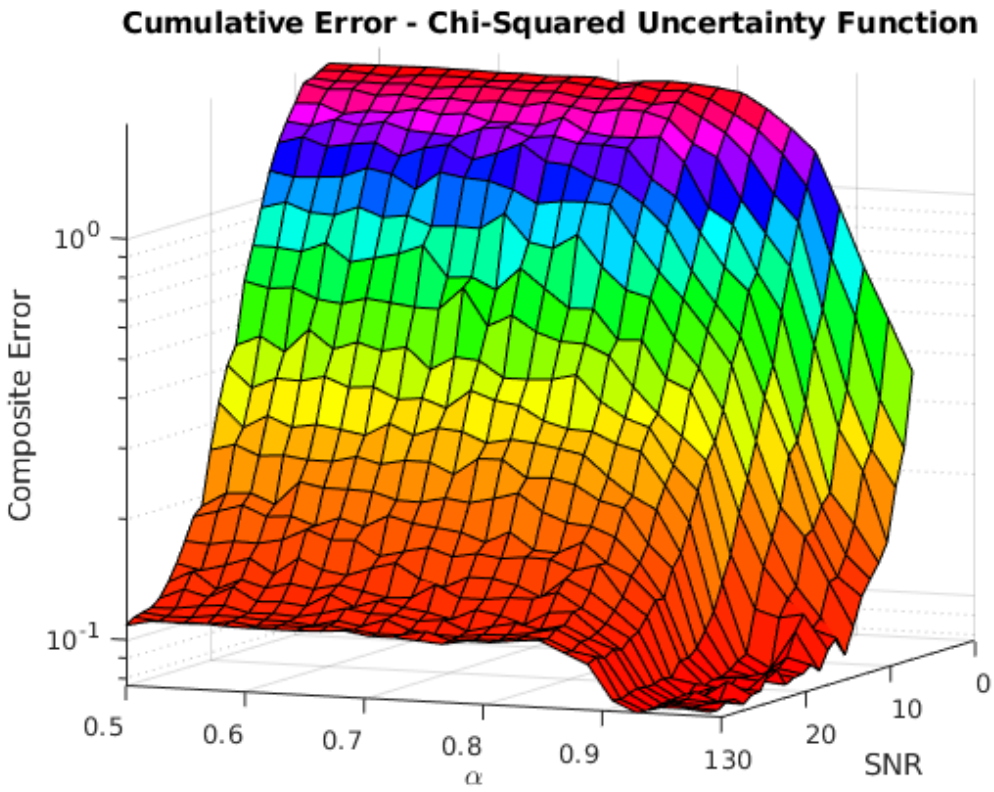


Figure 4.4. Composite error for Chi-squared uncertainty function across a range of $\alpha$ values

However, there is a cost associated with prioritizing a cognitive radar system for high target dwell ratios. The BAHs of two target simulations are shown in Fig. 4.6 and Fig. 4.7. Both simulations model a single target moving across the scene in a southeast direction with a channel model of 30 dB SNR. The first simulation uses a Chi-squared uncertainty function with $\alpha = 0.95$, the second at $\alpha = 0.80$. In the first simulation (Fig. 4.6), beam placement is heavily concentrated along the path of the target track, while in the second simulation (Fig. 4.7), the beamsteering strategy prioritizes for the environmental probability model provided to it. In this thesis, the probability of a new target appearing in the scene increases with Euclidean distance from its center (Section 2.3.1).

Because beam history is concentrated on the track path in the first simulation, composite track error is lower than in the second simulation. However, regions with a high probability of new targets appearing receive fewer beam illuminations in the first simulation than in the

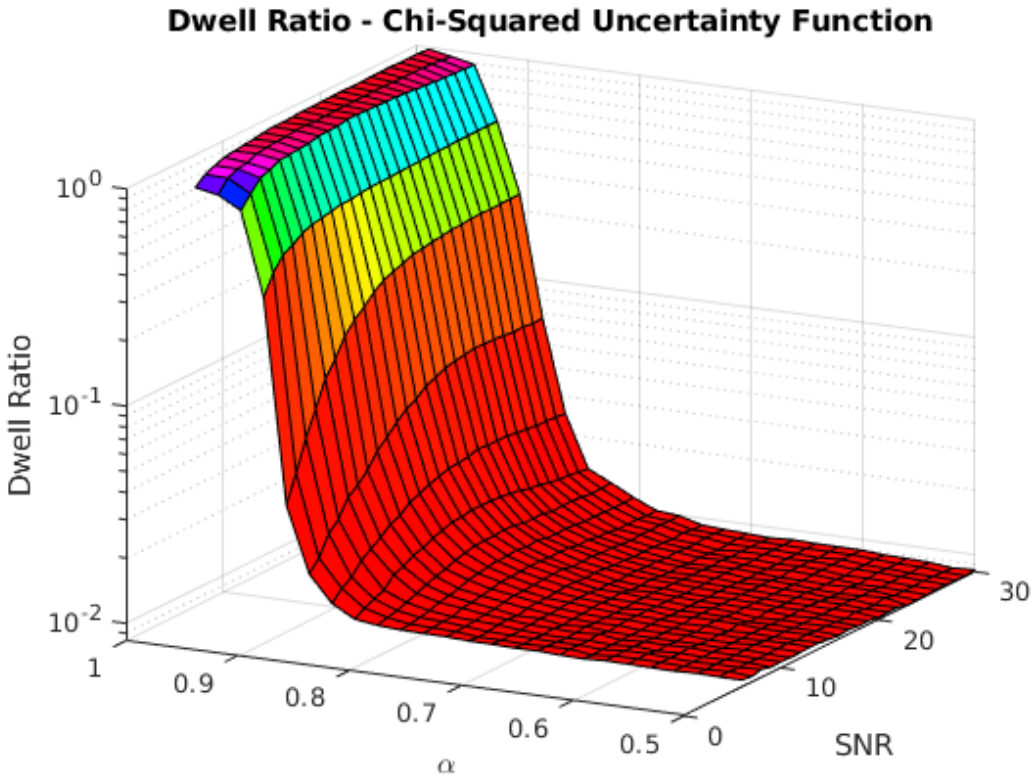


Figure 4.5. Track dwell ratio for Chi-squared uncertainty function across a range of $\alpha$ values

second, which corresponds to reduced target search/detection performance. Therefore, the $\alpha$ parameter of the Chi-squared uncertainty function can be used to tune resource allocation of a cognitive radar beamsteering strategy between its integrated search and tracking functions. For most applications, the $\alpha$ parameter should be set near the detection probability threshold of the AB-CRr system. In this configuration, the full range of search/track resource balancing can be achieved with $\alpha$ values between 0.85 and 1.0.

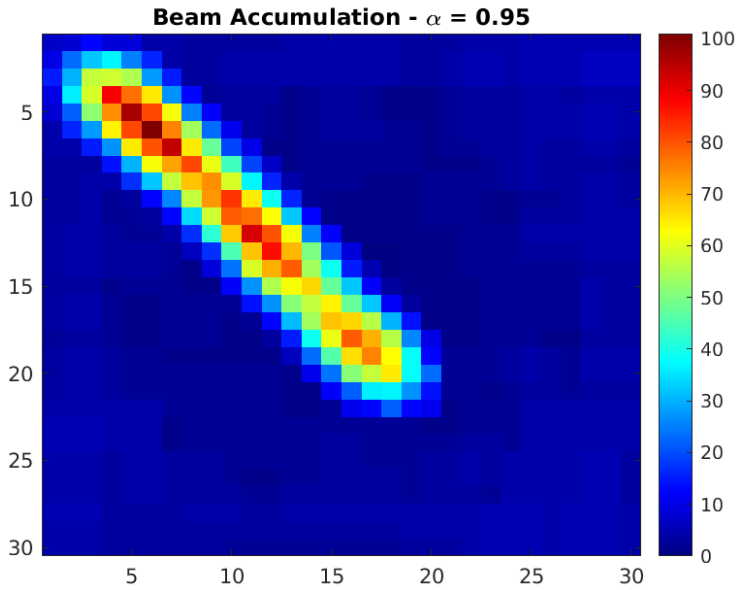


Figure 4.6. Beam accumulation history for Chi-squared AB-CRr with $\alpha = 0.95$

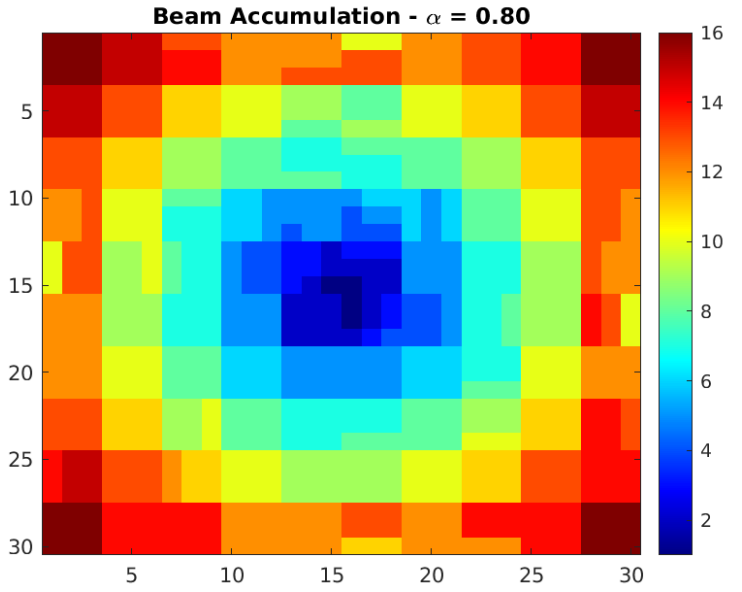


Figure 4.7. Beam accumulation history for Chi-squared AB-CRr with $\alpha = 0.80$

## 4.4 Track Feedback Hinting Performance

Track feedback hinting also impacts the search/track resource tradeoff. Increasing the uncertainty increment value, referred to in Section 3.4 as "track hinting", increases the bias of the adaptive beamsteering algorithm to revisit active target tracks as it evolves it beamsteering strategy. As a consequence, higher hinting feedback values ($\gamma$) result in higher target track dwell ratios. In Fig. 4.8, an AB-CRr system is simulated with a Chi-squared uncertainty function using $\alpha = 0.85$. Referring to Fig. 4.4, this AB-CRr configuration is biased for target search behavior (and correspondingly low dwell ratios). When track feedback hinting $\gamma = 0$, the dwell ratio maintains a steady state 0.018 at higher SNR values. However, as the feedback hinting value is increased, the target track dwell ratio also increases, indicating that adaptive beamsteering strategy is allocating more beams onto known target tracks. When the uncertainty feedback value is $\gamma = 0.15$, the high SNR dwell ratio increases to 0.026, a 44% increase in target track beam allocation compared the baseline case. Track state feedback to the adaptive beamsteering system is another means of modulating the search versus track balancing of resources in an AB-CRr system, complementing the uncertainty function design to generate the desired beamsteering behavior.

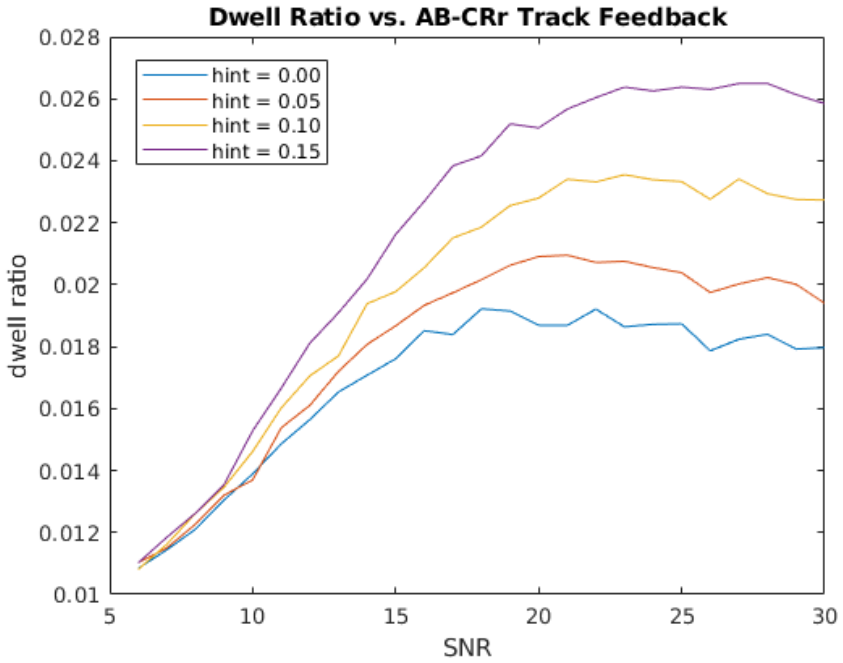


Figure 4.8. Target track dwell ratio with target track feedback

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 5:
# Swarm Dynamics Modelling

Extending AB-CRr target detection and tracking to multiple targets can be achieved by introducing a Kalman track file for each newly detected target. However, as the number of targets in the scene grows large, the tracking resources of a cognitive radar, like a traditional radar system, will become overwhelmed, lacking the resources to maintain each individual target track. Consolidated state vector Kalman tracking promises better performance with large target swarms, provided that the swarm targets have highly correlated velocity.

## 5.1 Kalman Tracking of Swarm Targets

The most direct approach to multiple target tracking is to assign each target in the scene its own Kalman track file. To accomplish this, detected target measurements must be associated with their corresponding tracks at each measurement interval.

### 5.1.1 Nearest Neighbors Measurement Association

When a target is detected, it is first tested against each existing target track in the swarm model via a chi-squared test. The Mahalanobis distance between each track in the beam scene and each target detection is computed via:

$$d_i = (\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k)^T (\mathbf{R} + \mathbf{H}_i \mathbf{P}_{k|k-1} \mathbf{H}_i^T)^{-1} (\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k) \tag{5.1}$$

where $d_i$ is the Mahalanobis distance between detected target and track $i$, $\mathbf{H}_i$ is the observation matrix that corresponds to that association, and $\mathbf{z}_k$ is the measurement vector.

For improved computational efficiency given that measurement covariance $\mathbf{R}$ is constant across iterations, the Woodbury matrix identity is preferable, yielding:

$$\begin{aligned} \mathbf{y}_k &= \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k \\ d_i &= \mathbf{y}_k^T \mathbf{R}^{-1} \mathbf{y}_k - \mathbf{y}_k^T \mathbf{R}^{-1} \mathbf{H}_i (\mathbf{P}_{k|k-1} + \mathbf{H}_i^T \mathbf{R}^{-1} \mathbf{H}_i)^{-1} \mathbf{H}_i \mathbf{R}^{-1} \mathbf{y}_k . \end{aligned} \tag{5.2}$$

The detected target is then associated with the target track with the smallest Mahalanobis distance from itself. If each track exceeds a threshold Chi-squared association value (not to be confused with and unrelated to the Chi-squared inspired uncertainty function previously introduced), then the detected target is assumed to be a new (previously unobserved) target and is added to the swarm model.

### 5.1.2 Multiple Measurements Nearest Neighbors Association

In the event that multiple targets are detected in a single beam illumination, the nearest neighbors model is easily extended to manage track associations. First, an association table is generated for all measurements and target tracks (Fig. 5.1). Next, each target detection is compared against candidate target tracks. If one or more target tracks meet the Chi-squared criterion for association, the nearest target track is updated with that target measurement. If none of the tracks associate with a detection, this detection is considered a new target. If a target track is illuminated by the beam but does not associate with any measurements, it is assumed to be a deprecated track and is removed from the swarm model. This target association and management algorithm is summarized in Fig. 5.2.

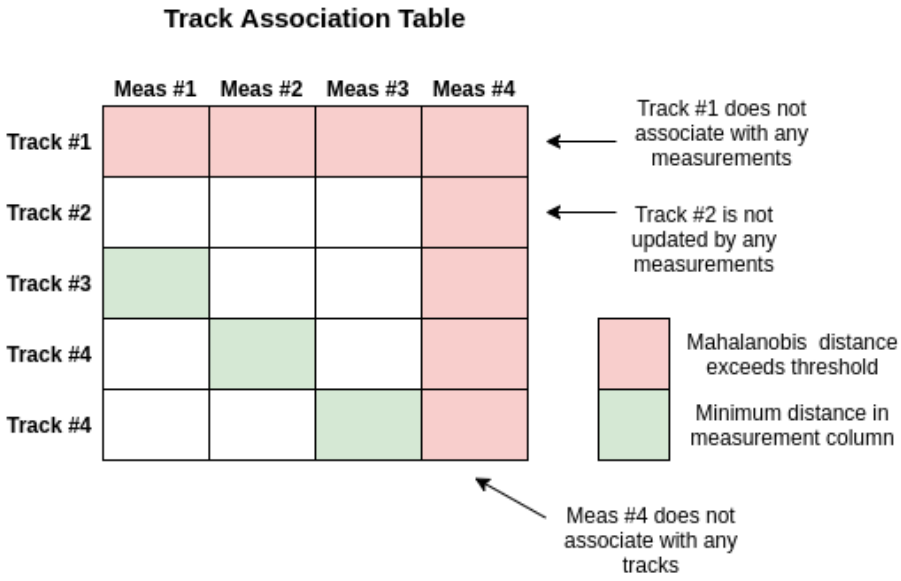


Figure 5.1. Track-to-measurement association table (Mahalanobis nearest neighbors method)

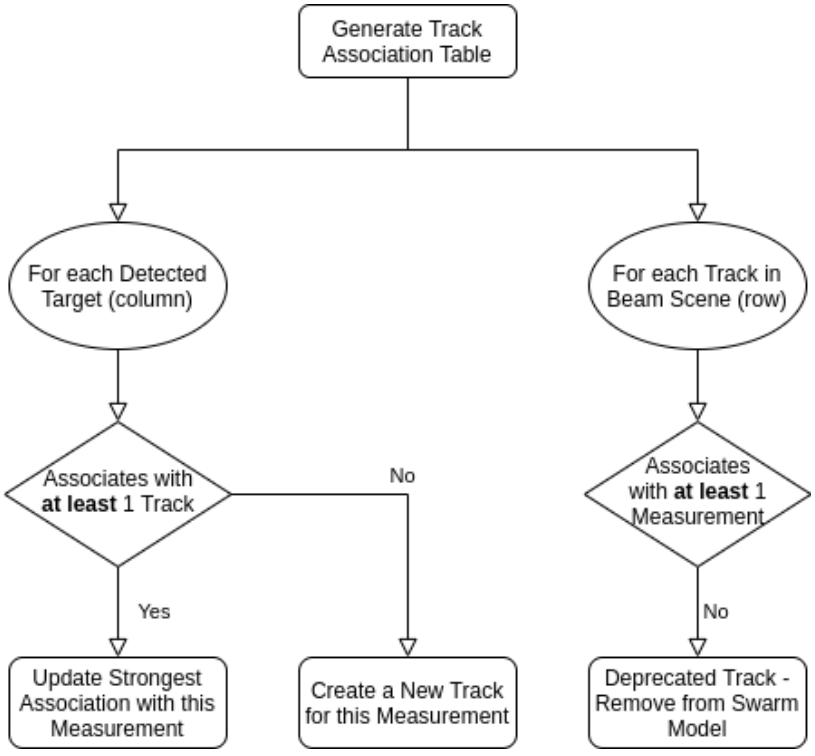


Figure 5.2. Track-to-measurement association flow diagram

## 5.2 Consolidated Swarm Kalman Tracking

The Kalman tracking problem for correlated velocity swarm targets can be simplified by assuming that all the members of a swarm have the same mean velocity. When this is the case, each target track can be represented in a single state vector. Each observation of a target in the swarm updates the velocity state estimate of the entire swarm, increasing the number of velocity measurements for the Kalman tracker. The new state model for a two target swarm can be expressed as:

$$\hat{\mathbf{x}} = \begin{bmatrix} \bar{v}_x \\ \bar{v}_y \\ x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix} \tag{5.3}$$

Note that instead of tracking each target separately, all of the member targets of a swarm are represented in a single state vector. The transition matrix for the consolidated swarm state vector is given as:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \Delta & 0 & 1 & 0 & 0 & 0 \\ 0 & \Delta & 0 & 1 & 0 & 0 \\ \Delta & 0 & 0 & 0 & 1 & 0 \\ 0 & \Delta & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.4}$$

where $\Delta$ is the time step between iterations. Future values of the state estimate vector can be computed at discrete times $n\Delta$ via:

$$\hat{\mathbf{x}}[n+N] = \mathbf{F}^n \hat{\mathbf{x}}[N] \tag{5.5}$$

Because the AB-CRr system is limited to only measuring targets enclosed within its beam

for a given iteration, the system will not always measure each member of a swarm in a single iteration. Therefore, the observation matrix changes with each iteration of measurement. If no targets are measured by a beam illumination, the observation matrix is null and no Kalman update occurs. If the first target member of a swarm is detected in a beam illumination, then the observation matrix is formed:

$$\mathbf{H}_{1,0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5.6}$$

Similarly, if the second member of a swarm track is detected, the observation matrix is formed:

$$\mathbf{H}_{0,1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.7}$$

If multiple targets are observed in a single beam, the update procedure is performed multiple times, once for each associated track in the swarm.

The subsequent Kalman update procedure for each iteration updates the state estimate of each target in the swarm even if only one of the targets is directly observed via

$$\begin{aligned} \mathbf{K} &= \mathbf{P}_{k+1|k}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k+1|k}\mathbf{H}_k^T + \mathbf{R}_k) \\ \mathbf{P}_{k+1|k+1} &= (\mathbf{I} - \mathbf{K}\mathbf{H}_k)\mathbf{P}_{k+1|k} \\ \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k) \end{aligned} \tag{5.8}$$

where $\mathbf{K}$ is the Kalman gain matrix and $\mathbf{P}$ is the state estimate covariance matrix.

### 5.2.1 Associating Targets to Consolidated Track

Measurement-to-track association for a consolidated swarm track follows the same procedure as the multiple track file method presented in Section 5.1. The only modification for the consolidated tracking system is that the observation matrix $\mathbf{H}_i$ in Eq. (5.2) becomes the consolidated state vector observation matrix corresponding to its associated track ($\mathbf{H}_{0,1}$, $\mathbf{H}_{1,0}$, etc.).

### 5.2.2 Initializing and Removing Targets from the Swarm Model

**Initializing Newly Detected Targets**

In this thesis work, if the Mahalanobis distance between a measurement and each active track exceeds the 95% Chi squared association test, it is assumed to be a newly observed target. New target tracks are appended to the existing consolidated state estimate and estimate covariance matrix. The state vector is updated with the measured spatial coordinates of the new target:

$$\hat{\mathbf{x}}_{k+1|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \mathbf{z}_{x,k} \\ \mathbf{z}_{y,k} \end{bmatrix} \tag{5.9}$$

and the estimate covariance matrix is updated with the spatial covariance sub-matrix of the associated measurement:

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \mathbf{P}_{k|k} & 0 & 0 \\ 0 & \frac{1}{M_x^2} & 0 \\ 0 & 0 & \frac{1}{M_y^2} \end{bmatrix}. \tag{5.10}$$

**Removing Dead Tracks**

A mechanism for removing old or inaccurate target tracks is also necessary for dense swarm formations. In this work, if the position of an active target track is illuminated by a beam and fails to associate at a 99% level with any of the subsequent detection measurements, it

is presumed to be a bad track and is marked for removal. The procedure for removing the first target from a two-target consolidated track begins by forming a selection matrix for the spatial sub-matrix of the first target:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{I}_2 & 0 & 0 \\ 0 & 0 & \mathbf{I}_2 \end{bmatrix} \tag{5.11}$$

.

where $\mathbf{I}_2$ is the $2x2$ identity matrix. The state matrix is then updated via:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{Y}\hat{\mathbf{x}}_{k|k} \tag{5.12}$$

and the estimate covariance matrix is updated with:

$$\mathbf{P}_{k+1|k} = \mathbf{Y}^T\mathbf{P}_{k|k}\mathbf{Y}. \tag{5.13}$$

Note that the swarm velocity estimate is unaffected by the removal of a target track. The swarm velocity estimate is assumed to converge to the mean velocity of all of the targets in the consolidated tracking model.

## 5.3 Swarm State Estimation

When considering targets as members of a swarm, it is often more useful to describe characteristics of the swarm in aggregate, rather than the characteristics of each of its component parts. In this section, methods of describing the state and behavior of swarms are discussed. These metrics will serve as the basis of performance estimation for swarm detection and tracking in the following chapter.

### 5.3.1 Swarm Centroid Estimate

The centroid of a point cloud is defined as the mean value along each coordinate axis. For the individual target track methodology (Section 5.1), the centroid of the swarm is defined as

$$\hat{\mathbf{x}}_c = \frac{1}{T}\sum_{k=1}^{T}\hat{\mathbf{x}}_k, \tag{5.14}$$

where $\hat{\mathbf{x}}_c$ is the state vector of the swarm centroid estimate, $\hat{\mathbf{x}}_k$ is the state vector of the $k^{th}$ track, and $T$ is the total number of target tracks in the swarm model. For the consolidated swarm tracking model, the centroid estimate is similarly computed as

$$\hat{\mathbf{x}}_c = \begin{bmatrix} 0 & 0 & \frac{1}{T} & 0 & \frac{1}{T} & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \frac{1}{T} & 0 & \frac{1}{T} & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & \dots \end{bmatrix} \hat{\mathbf{x}} \tag{5.15}$$

where $\hat{\mathbf{x}}$ is the consolidated swarm state vector.

### 5.3.2 Swarm Spatial Region Estimate

In some cases, it may be useful to estimate the region of space that a target swarm resides in. This can be achieved by forming a polygon that encapsulates each target track in the swarm model. First, regarding the swarm as a point cloud of spatial coordinates, we solve the convex hull problem to generate the boundary points of a polygon. This polygon is subsequently discretized back into the probability model grid to form a subset of probability cells in which the swarm is said to reside. Fig. 5.3 is a 2-dimensional visualization of this process, where black points are the continuous state estimates of swarm targets, the dark green polygon is the convex hull of the target point cloud, and yellow cells are the discrete cells of the probability model that are said to be members of the swarm region. The red star corresponds to the spatial centroid of the target swarm.

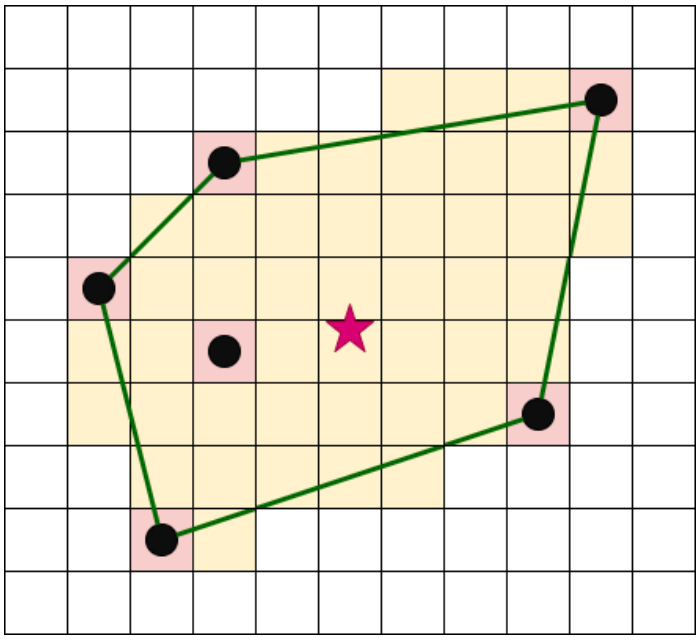

Figure 5.3. Swarm region estimate in discrete space

## 5.4 Adaptive Beamsteering Feedback

As in the single target case, multi-target track information can be used as feedback into the uncertainty model that drives adaptive beamsteering. There are numerous possible feedback models that can be employed, but in this thesis work, the uncertainty of the swarm estimate region is incremented with each iteration. This induces adaptive beamsteering behavior that not only increases dwell time on known swarm targets, but allocates more beam time to detection and tracking of potential interior members of the target swarm.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 6:
# Swarm Target Search-and-Track Performance

In this chapter, the integrated search-and-track performance of an AB-CRr system is modelled in simulation for swarm target scenarios. The impact of two proposed swarm tracking models are compared via Monte Carlo simulation. Design considerations, including uncertainty function design and track feedback hinting that were weighed in single target scenarios are integrated into the swarm tracking case. Each swarm scenario modelled in this chapter employs a Chi-squared uncertainty function and target track uncertainy feedback as described in Chapter 4.

## 6.1 Performance Metrics

As in the single target case, target swarm search-and-track performance can be quantified in numerous ways. This section presents the performance metrics used to assess the effectiveness of AB-CRr against various swarm configurations.

### 6.1.1 Swarm Velocity Estimation

One of the core capabilities of swarm target tracking is swarm velocity estimation. In this thesis, mean swarm velocity estimation is complicated by the radar geometry under investigation. Radar return measurements provide target velocity information relative to the radar receiver (y-axis of the spatial grid), but target motion perpendicular to (x-axis of the spatial grid) the receiver must be estimated from target position history. The responsiveness of swarm tracking models to individual target measurements can be visualized by plotting the mean swarm velocity estimates over time.

### 6.1.2 Swarm Centroid Cost Function

In the swarm tracking case, a tracking cost function can be defined based on the parameters of the target swarm we wish to estimate. Our cost function will seek to minimize the Euclidean distance between the true and estimated swarm mean velocity and spatial centroid, defined in Section 5.3.1 by the state vector:

$$\hat{\mathbf{x}}_c = \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_{v_x} \\ \mu_{v_y} \end{bmatrix} \tag{6.1}$$

The swarm cost function, or composite centroid error, can then be defined analogously with the single target case:

$$\epsilon_s = \frac{1}{N}\Big(\beta k_0 + \sum_{k=k_0}^{N-1} ||\bar{x}_{c,k} - \hat{x}_{c,k}||_2\Big), \tag{6.2}$$

where $N$ is the total number of beam iterations by the radar system, $k_0$ is the first beam iteration after a target track file has been initialized, and $\beta$ is a user defined coefficient that weights the relative contribution of detection latency compared to mean track error. In this thesis work, $\beta$ is assumed to be equal to 2.

## 6.2 Swarm Detection and Tracking Performance

Two swarm tracking methodologies are compared in this thesis: using separate individual target track files and a consolidated swarm state vector approach. In this section, we investigate the relative performance of the two methodologies in estimating the mean position and velocity of a target swarm (swarm centroid). All simulations and trials in this chapter use discrete scene dimensions $M_x = M_y = 30$ and $N_y = 9$, for a total of 8100 discrete measurement states.

### 6.2.1 Two Target Swarm Performance

First, we investigate behavior in a trivial swarm composed of two targets. The mean swarm velocity estimate over the course of a simulation is shown in Fig. 6.1. In this scenario, a swarm of 2 targets move across the radar search scene in a loose formation with constant mean velocity. The two targets are tracked using separate track files as described in Chapter 5. The swarm y-axis velocity measurement error is low from the onset of detecting the first target in the swarm. This is expected as y-axis velocity is directly measured by the radar

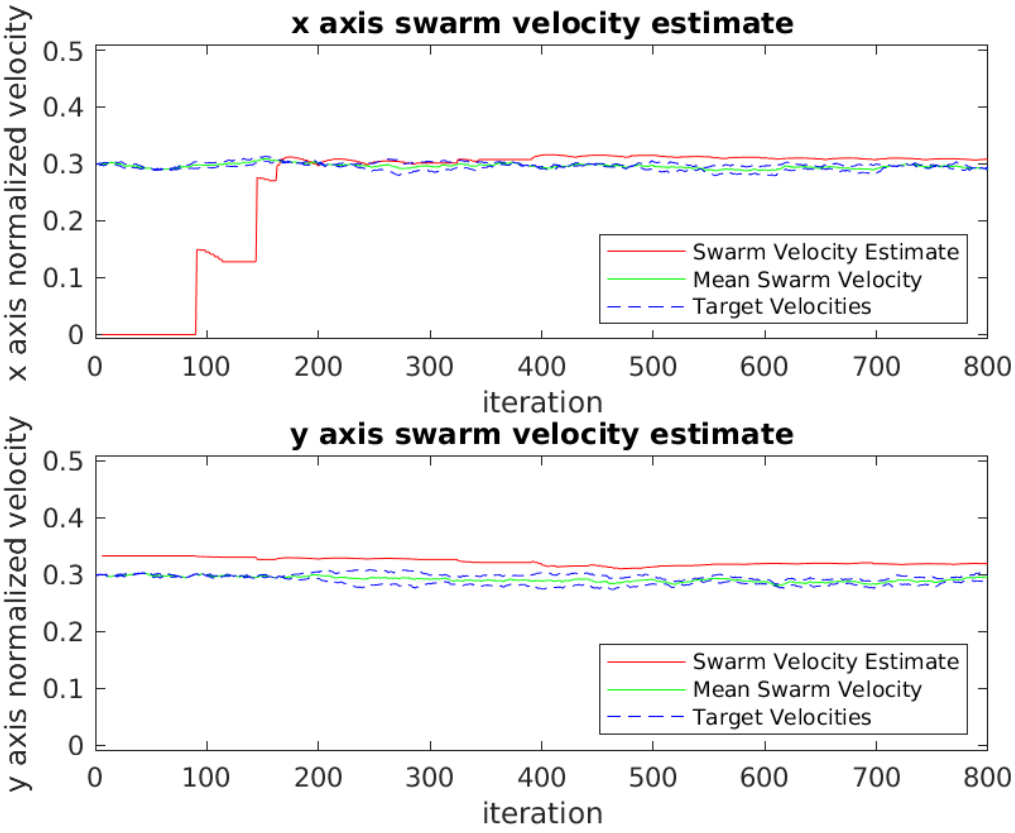


Figure 6.1. Two target swarm mean velocity estimates using separate target track files

system. The swarm x-axis velocity estimate converges on the swarm true mean velocity more slowly, as the radar system requires multiple track associations to estimate target motion perpendicular to the observer. In the simulation presented, the velocity estimate using the separate track file approach converged to true swarm x-axis velocity in approximately 180 iterations of beam selection (this count is the total number of beam iterations, including iterations that do not illuminate any targets).

Simulating the same scenario with a consolidated swarm track file approach (Fig. 6.2), we observe a reduction in the number of iterations required for the x-axis velocity estimate to converge on true swarm motion. The consolidated tracking methodology intercepts true swarm x-axis velocity in approximately 100 iterations, compared to 180 iterations with the separate target tracking method. The consolidated track file method outperforms the

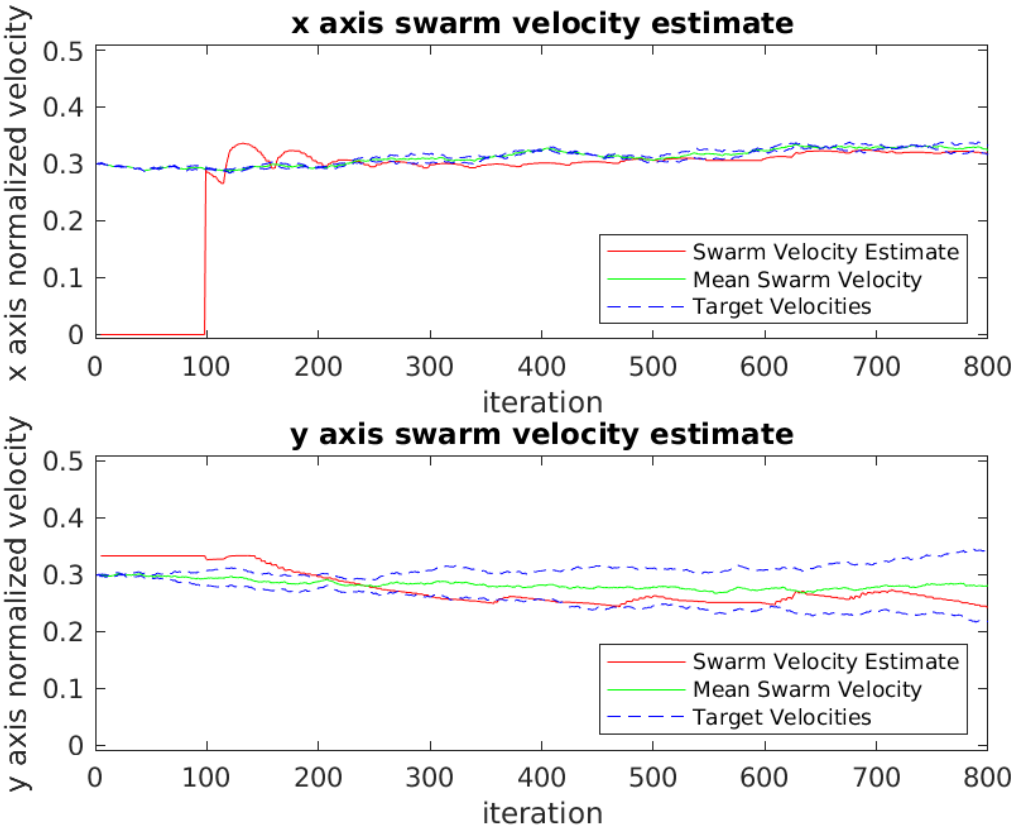


Figure 6.2. Two target swarm mean velocity estimates using a consolidated swarm state vector

separate track method in this case because each target association directly updates the velocity estimate, while with the separate target tracking methodology, each individual target track must reach a steady state velocity estimate before the swarm velocity estimate can reach steady state. Additionally, the consolidated swarm tracking model encapsulates targets' spatial correlation in its state estimate covariance matrix **P**.

Generalizing these observations, the composite swarm tracking error for a two target swarm is estimated via Monte Carlo simulation for both the separate and consolidated tracking models over a range of SNR values. The mean swarm tracking error of 500 iterations of a two target constant velocity swarm moving through the search scene is presented in Fig. 6.3. Note that composite track error for both a consolidated and separate track methodologies are comparable up to about 16 dB SNR, after which consolidated swarm tracking outperforms

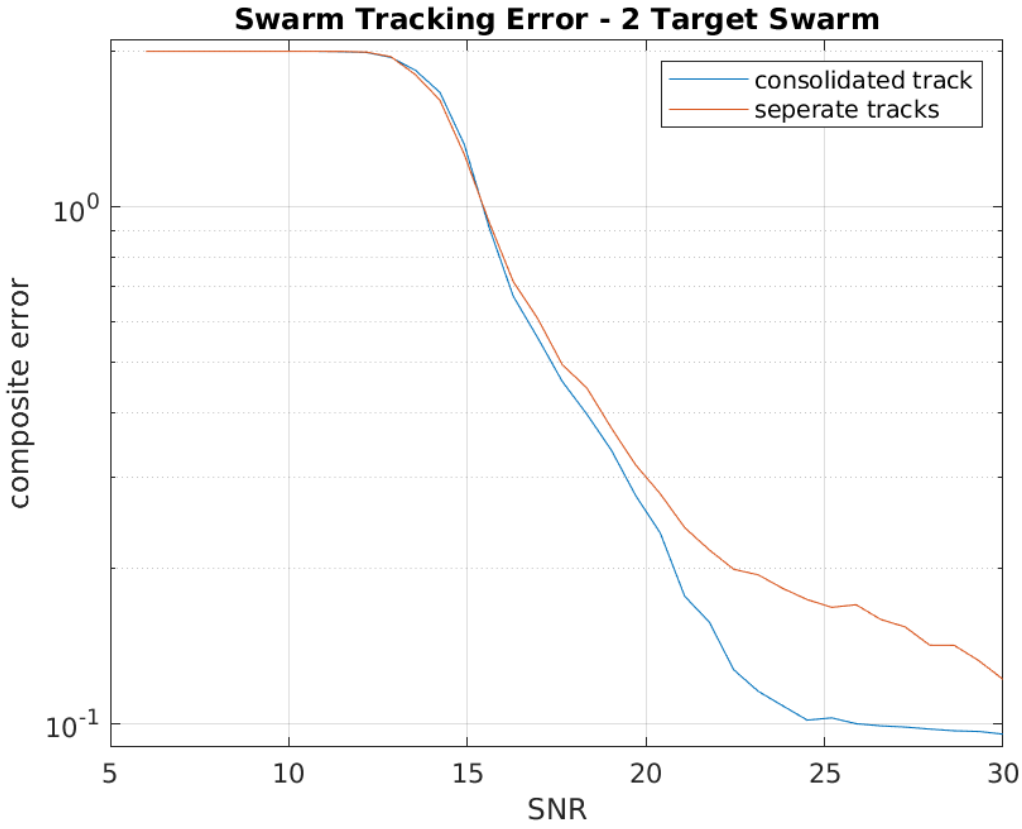


Figure 6.3. Monte Carlo simulation of swarm tracking error for a two target swarm

individual target track files (we simulated up to 30 dB SNR). At 25 dB SNR, the consolidated tracking method exhibits 38% lower tracking error than the separate tracking counterpart. Therefore, in the trivial two target swarm scenario, consolidated tracking outperforms the individual target tracking method.

### 6.2.2 Seven Target Swarm Performance

Next, we investigate how both swarm tracking methodologies perform when the number of targets in the swarm becomes large. Large swarms present a unique tracking challenge, as beamforming resources are divided among an increasingly large number of target tracks. As a consequence, we expect the swarm tracking error to be higher for a given SNR than in the 2 target case, as beamsteering resources allocated to target tracking will be strained by

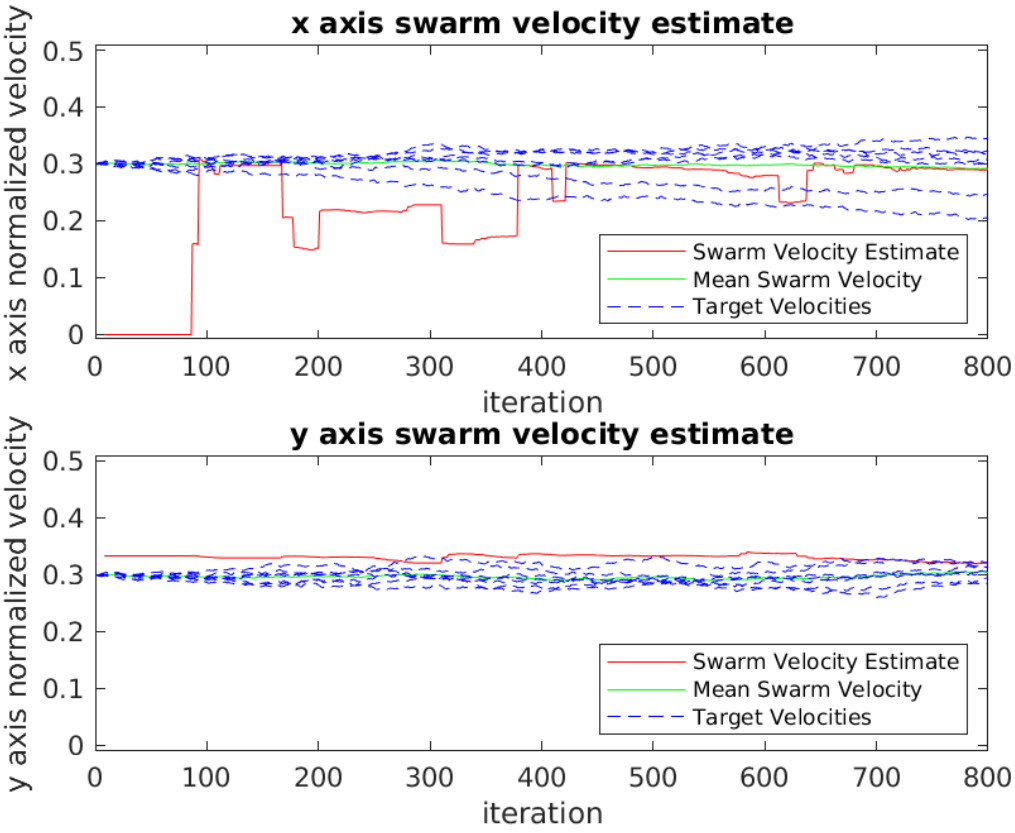


Figure 6.4. Seven target swarm mean velocity estimates using separate target track files

the large number of targets in the swarm.

Repeating the simulation scenario for Fig. 6.1 with a seven target swarm and separate target tracking, we see that the swarm x-axis velocity estimate converges slowly towards a steady-state velocity estimate and has a tendency to deviate from the steady state (i.e. an inconsistent estimate). Compared to the two target scenario, which took 100 beam iterations to converge on an x-axis velocity estimate, the seven target scenario estimate (Fig. 6.4) incrementally reaches true swarm motion but does not maintain that estimate consistently. The AB-CRr system in this scenario is overwhelmed by the number of target tracks to maintain, and poor individual track estimates result in an inconsistent mean swarm velocity estimate.

In comparison, the same seven target swarm scenario tracked with a consolidated swarm

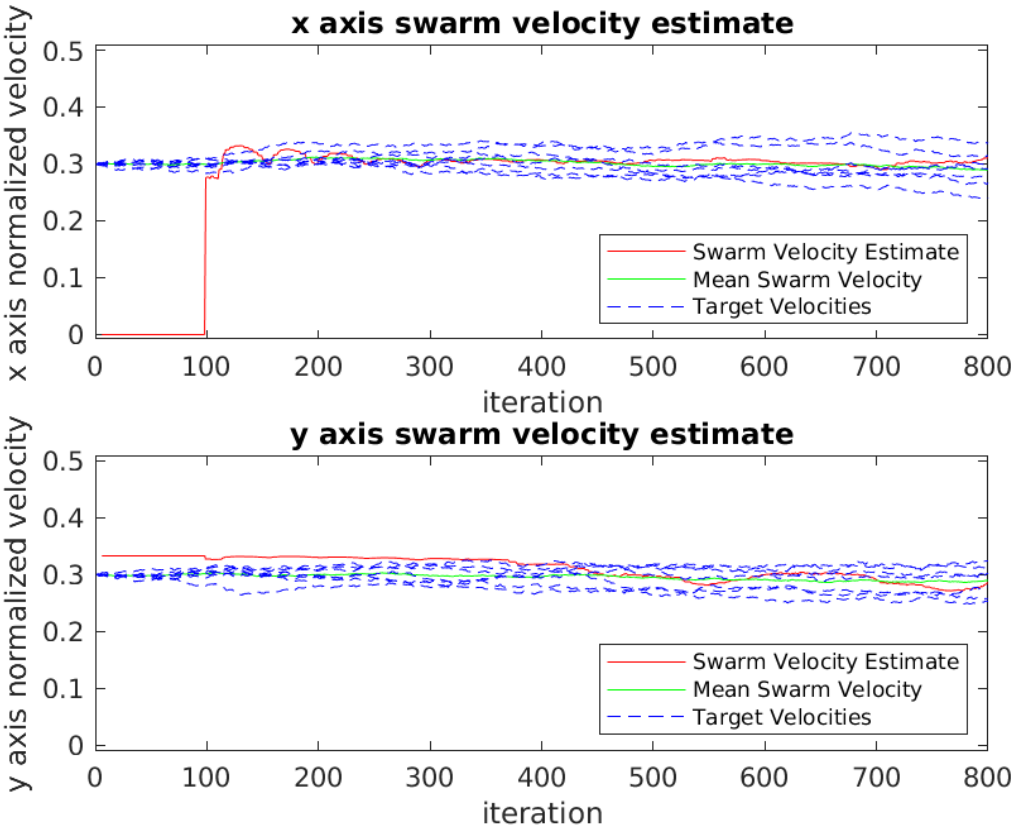


Figure 6.5. Seven target swarm mean velocity estimates using a consolidated swarm state vector

state vector achieves much more stable velocity estimates (Fig. 6.5). This simulated scenario converges to the swarm true x-axis velocity estimate in approximately 100 beam selection iterations. This is similar to the delay of 2 target swarm with consolidated state tracking. As a result, we would expect consolidated swarm tracking to be more robust to large target swarms than the separate target tracking methodology.

The composite swarm tracking error for the seven target swarm is again estimated via Monte Carlo simulation for both the separate and consolidated tracking approaches over a range of SNR, as with the two target scenario. Our results are based on 500 Monte Carlo trials for each environmental noise and tracking method case and are summarized in Fig. 6.6. Consolidated swarm tracking and individual target tracking have comparable tracking error up to about 18 dB SNR, after which the consolidated tracking model outperforms

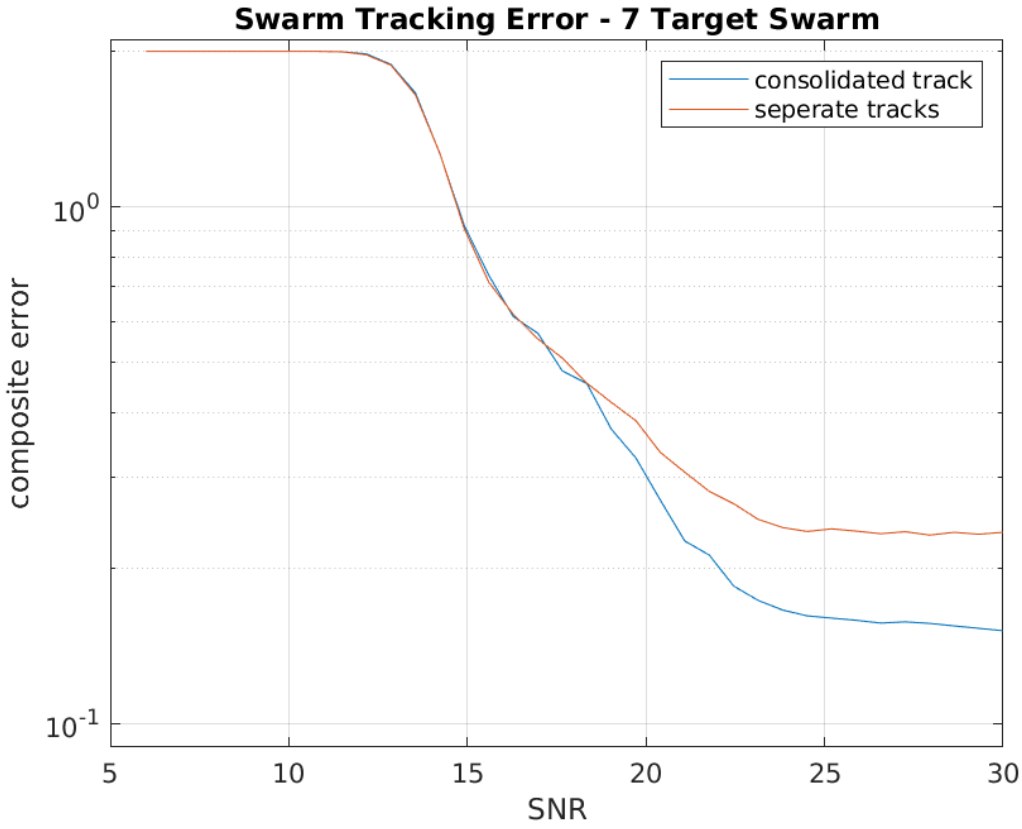


Figure 6.6. Monte Carlo simulation of swarm tracking error for a seven target swarm

the separate target tracks. Both tracking methodologies approach an asymptotic track error between 25 and 30 dB SNR, where composite tracking maintains approximately 34% lower track error than the individual target tracking method.

## 6.3 Swarm Target Search/Tracking Resource Allocation

A comparison of swarm target tracking dwell ratios for both the consolidated and separate track approaches indicates that the consolidated swarm tracking method does not compromise search/detection performance for its relative improvement in target tracking. Fig. 6.7 plots the track dwell ratio across a range of SNR for both the 2 target and 7 target swarm scenarios. The consolidated swarm tracking and separate target tracking approaches maintain similar dwell ratios for a given swarm size, indicating that the two methods allocate

approximately the same degree of beamsteering resources to swarm tracking.

Therefore, we conclude that consolidated swarm tracking is a more efficient methodology for tracking correlated velocity target swarms than an individual target tracking approach in terms of beamsteering resources.

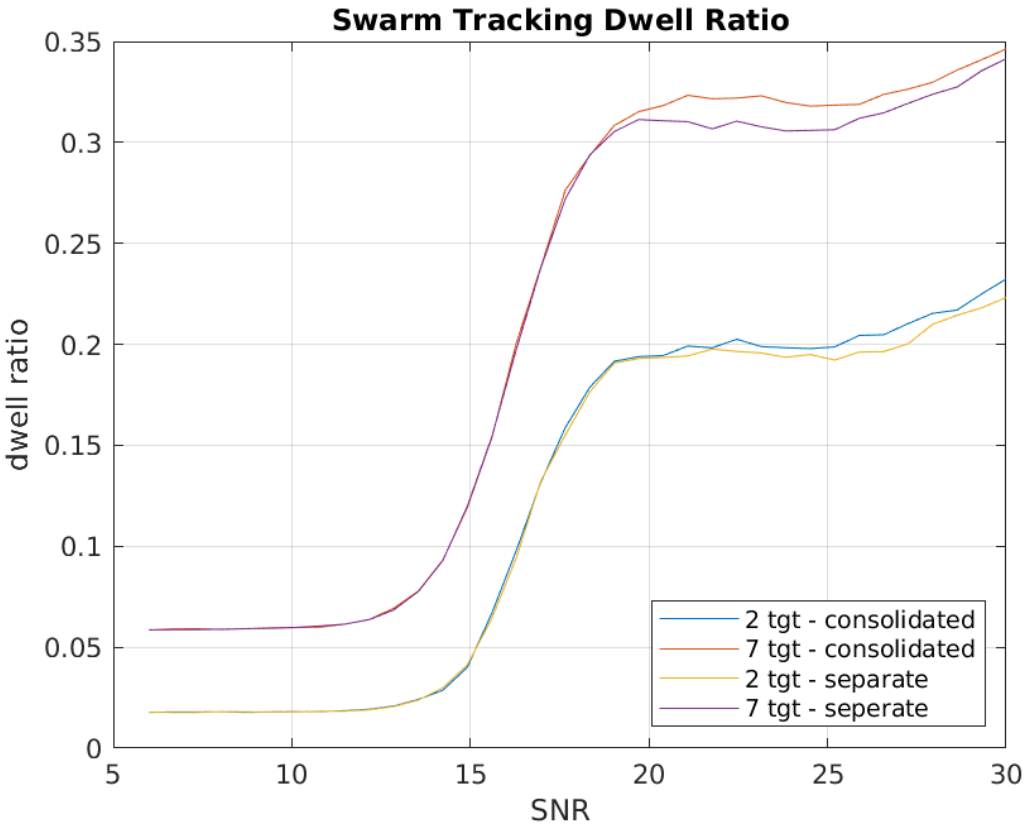


Figure 6.7. Monte Carlo simulation of swarm tracking dwell ratio for various swarm configurations

# CHAPTER 7:
# Summary

In this work, we investigated the suitability of adaptive beamsteering cogntive radar (AB-CRr) for large target swarms. We examined design considerations of a cognitive radar system and how they pertain to beamsteering behavior against groups of targets with correlated motion. Following from prior art, we implemented beamsteering resource allocation between integrated search and tracking applications via a parameterized uncertainty function, settling on a modified Chi-squared uncertainty function for swarm tracking. Additionally, this thesis introduced target track uncertainty feedback into the AB-CRr framework to complement uncertainty function design in modulating beam allocation behavior. As a result, we demonstrated improved beamsteering resource efficiency for search and tracking applications compared to prior AB-CRr configurations.

These AB-CRr methodology was then extended to multiple target swarm scenarios. Target swarms were defined as groups of targets with correlated motion. Mahalanobis distance nearest neighbors track association was employed in the swarm target scenario to integrate with Kalman tracking information. Two methodologies were proposed for swarm tracking applications. The first method establishes an individual Kalman track file for each detected member target of the swarm in the search scene. The second method assumes that each swarm member has the same mean velocity and tracks each member target within the same consolidated swarm state vector. The second methodology takes advantage of state correlation between each member target of the swarm, and was expected to have better performance.

The performance of both the separate target tracking and consolidated swarm tracking methodologies were evaluated via Monte Carlo simulation. The consolidated swarm tracking methodology outperformed the separate track methodology across a range of SNR and swarm sizes. When target swarms were large, the separate target tracking method exhibited signs of target saturation and resource overloading, while the consolidated swarm tracking method maintained up to 34% lower composite swarm tracking error in high SNR conditions.

Overall, AB-CRr presents a promising approach to improving beamsteering resource ef-

ficiency in the presence of large swarm targets. An AB-CRr framework tailored to robust tracking of a large number of swarm targets is presented in this work, employing a parameterized uncertainty function, swarm target feedback to the beamsteering uncertainty model, and a consolidated swarm state vector approach to Kalman filter target tracking.

Future work on the AB-CRr framework presented in this thesis can proceed down several avenues. First novel uncertainty functions should be investigated that correspond to beamsteering behavior more favorable for target swarm search-and-track beamsteering strategies. Second, this thesis work only considers target swarms with highly correlated target motion. Future work should investigate the impact of weak intra-swarm target motion correlation on tracking performance. Third, future work with AB-CRr should generalize the probability update and beam-selection model for a range of radar and scene geometries, rather than the single geometry considered in this work.

# APPENDIX: Efficient Computation of Probability Model

In simulation, the most expensive computation that must be performed is the receive signal joint probability density function computation. Most simulations assume that the radar beam encompasses a 3x3 matrix of spatial cells, each with 9 corresponding Doppler frequency shift cells. Referring to Eq. (2.12) and assuming that no more than 2 targets can appear in a single beam illumination, this yields 3322 hypotheses for which the joint probability distribution function must be computed on each beam iteration. Efficient computation of these hypotheses becomes the limiting factor in the adaptive beamsteering methodology.

## A.1 Joint PDF Value Scaling

The joint probability distribution value for a single hypothesis is computed via the complex multivariate Gaussian distribution

$$P(\mathbf{z}_k|H_i) = \frac{1}{\pi^k|\mathbf{C_z}|}\exp\left(\mathbf{s}_i - \mathbf{z}_k\right)^H \mathbf{C}_\mathbf{z}^{-1}(\mathbf{s}_i - \mathbf{z}_k). \tag{2.14}$$

In the case where the radar receiver has many elements, joint probability density function values become very small, as probability values are distributed across a high dimensional probability space. This can result in poor behavior with floating point numbers, as many non-zero probability values are treated as zero. To mitigate this effect, probability values are scaled to a range more appropriate for floating point computations. In this thesis, the maximum hypothesis joint PDF value is scaled to be equal to $\frac{0.001}{\pi^k|\mathbf{C_z}|}$. This is achieved via the scaling transformation:

$$K = \frac{\ln 0.001}{\text{argmax}_i\{(\mathbf{s}_i - \mathbf{z}_k)^H \mathbf{C}_\mathbf{z}^{-1}(\mathbf{z} - \mathbf{s}_i)\}} \quad \text{such that}$$
$$(\mathbf{s}_i - \mathbf{z}_k)^H \mathbf{C}_\mathbf{z}^{-1}(\mathbf{z} - \mathbf{s}_i) \rightarrow K(\mathbf{s}_i - \mathbf{z}_k)^H \mathbf{C}_\mathbf{z}^{-1}(\mathbf{z} - \mathbf{s}_i) \tag{A.1}$$

This simple scaling algorithm ensures that the joint probability of each cell is constrained to $[0, \frac{0.001}{\pi^k |\mathbf{C_z}|}]$. Referring to 2.10, $P(\mathbf{z}_k|H_i)$ is scaled by $P(\mathbf{z}_k)$, which is given by 2.17. Therefore, the sum of all $\frac{P(\mathbf{z}_k|H_i)}{P(\mathbf{z}_k)}$ remains 1, and the relative magnitude of the joint probabilities is retained through the transformation.

## A.2 Prior Probability Dynamic Target Breakout

Observing Eq. (2.10), note that if the prior probability of a target being located in a cell is equal to zero, the posterior probability of target presence will also be zero regardless of the joint probability of the received signal. In our dynamic model, targets can enter and leave the search space and are free to move from cell to cell arbitrarily. Through the course of a simulation, the radar beam will illuminate any given cell repeatedly. Each time that the cell does not contain a target, its associated probability will asymptotically approach zero. If a target were to suddenly appear in such a cell, the probability of the cell would remain near zero even if there is a high probability that the cell now contains a target. Fig. A.1 plots the probability growth of a cell over successive iterations of beam illumination given that the initial probability of the cell is $P = 10^{-6}$ when a target first appears in the cell. At high SNR, the unmodified Bayes methodology produces such a high measured probability of target presence in the cell that the small initial prior probability of the cell is overcome, allowing the cell probability to grow near $P = 1$ in a few iterations. However, at smaller SNRs, there is greater uncertainty that a given measurement contains a target, and $P(\mathbf{z}_k|H_i)$ is not large enough to allow the cell to "grow" from $P \approx 0$ to $P \approx 1$ quickly. In the case of SNR = 3 for Fig. A.1, it takes on average 13 iterations of the radar beam illuminating a target before the cell probability of target presence grows past $P = 0.5$.

Because the cognitive radar is designed to operate in a dynamic target environment, it should be flexible to the introduction of new targets to cells that previously did not contain targets, while simultaneously employing prior probability knowledge of the environment to inform its current measurements. One simple solution to this "low cell probability breakout" problem is to place a lower limit on the prior probability of the cell. Fig. A.2 shows the same simulation as Fig. A.1, except that $P(\mathbf{z}_{k-1}|H_i)$ is lower bounded to a minimum of $\alpha = 0.001$. Therefore, the modified Bayesian prior probability expression becomes:

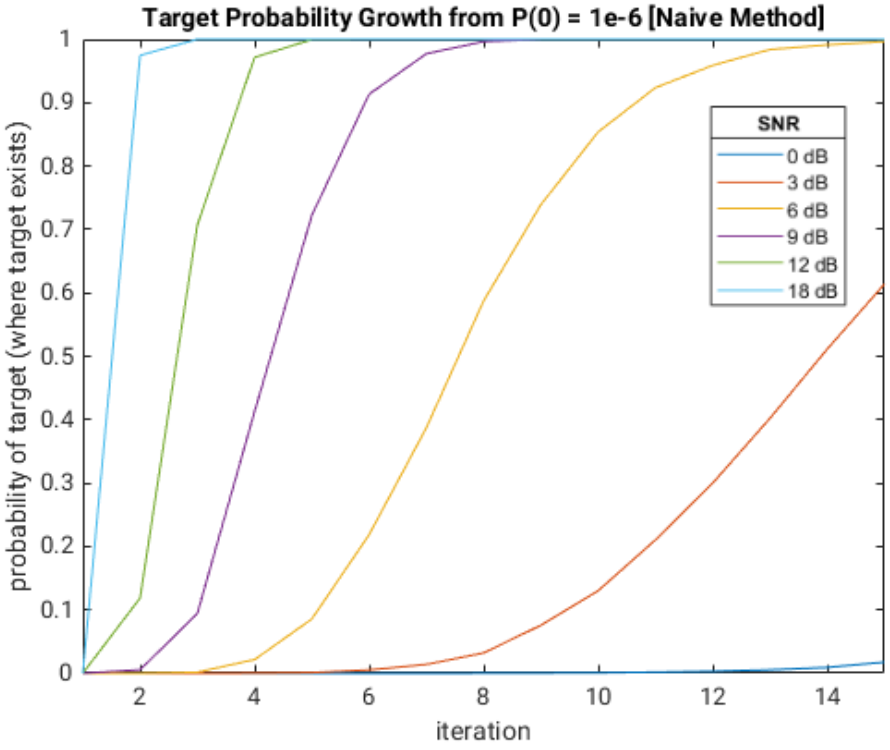


Figure A.1. Monte Carlo simulation of cell probability growth over successive illuminations in the presence of a target

$$P(H_i|\mathbf{z}_k) = \frac{\beta P(H_i|\mathbf{z}_k)}{P(\mathbf{z}_k)}$$
$$\beta = \begin{cases} \alpha & P(H_i|\mathbf{z}_k) \leq \alpha \\ P(H_i|\mathbf{z}_k) & \alpha < P(H_i|\mathbf{z}_k) \leq 1 \end{cases} \tag{A.2}$$

Observing the behavior of the system in Fig. A.2 versus Fig. A.1, note that the performance of the two update methods are fairly similar at high SNR, but that cell probability grows significantly faster for the modified method (Eq. A.2) at lower SNR.

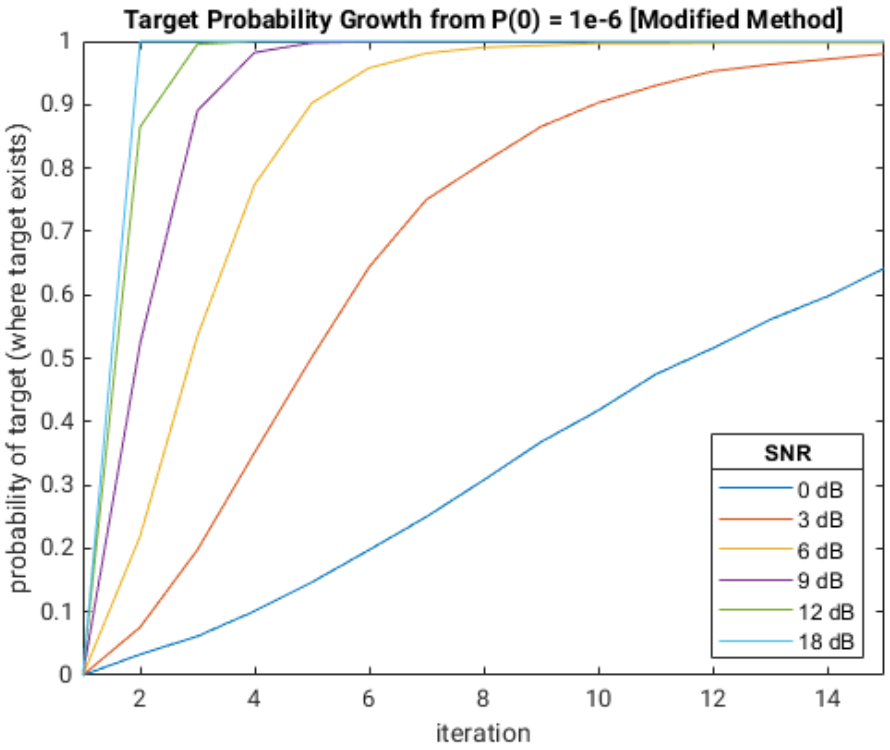


Figure A.2. Monte Carlo simulation of cell probability growth over successive illuminations in the presence of a target

# List of References


[1] R. A. Romero and N. A. Goodman, "Cognitive radar network: Cooperative adaptive beamsteering for integrated search-and-track application," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 915–931, April 2013.

[2] R. A. Romero, C. M. Kenyon, and N. A. Goodman, "Channel probability ensemble update for multiplatform radar systems," in *2010 International Waveform Diversity and Design Conference*, Aug. 2010, pp. 000 182–000 187.

[3] R. A. Romero and N. A. Goodman, "Adaptive beamsteering for search-and-track application with cognitive radar network," in *2011 IEEE RadarCon (RADAR)*, May 2011, pp. 1091–1095.

[4] P. Nielsen and N. A. Goodman, "Integrated detection and tracking via closed-loop radar with spatial-domain matched illumination," in *2008 International Conference on Radar*, Sep. 2008, pp. 546–551.

[5] Z. W. Johnson and R. A. Romero, "Uncertainty function design for adaptive beamsteering cognitive radar," in *2020 IEEE International Radar Conference*, April 2020.

[6] K. L. Bell, C. J. Baker, G. E. Smith, J. T. Johnson, and M. Rangaswamy, "Cognitive radar framework for target detection and tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 8, pp. 1427–1439, Dec. 2015.

[7] A. E. Mitchell, G. E. Smith, K. L. Bell, A. Duly, and M. Rangaswamy, "Fully adaptive radar cost function design," in *2018 IEEE Radar Conference (RadarConf18)*, April 2018, pp. 1301–1306.

[8] B. M. Bey, "Clutter covariance modelling matlab code," August 2019, unpublished.

[9] W. MathWorld. Binomial Series. *from Wolfram MathWorld*. [Online]. Available: http://mathworld.wolfram.com/BinomialSeries.html

THIS PAGE INTENTIONALLY LEFT BLANK

# Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California