

Washington University in St. Louis

Washington University Open Scholarship

McKelvey School of Engineering Theses & Dissertations

McKelvey School of Engineering

Spring 5-15-2020

Cognitive Radar Detection in Nonstationary Environments and Target Tracking

Yijian Xiang

Washington University in St. Louis

Follow this and additional works at: https://openscholarship.wustl.edu/eng_etds



Part of the [Electrical and Electronics Commons](#)

Recommended Citation

Xiang, Yijian, "Cognitive Radar Detection in Nonstationary Environments and Target Tracking" (2020). *McKelvey School of Engineering Theses & Dissertations*. 556.
https://openscholarship.wustl.edu/eng_etds/556

This Dissertation is brought to you for free and open access by the McKelvey School of Engineering at Washington University Open Scholarship. It has been accepted for inclusion in McKelvey School of Engineering Theses & Dissertations by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.

WASHINGTON UNIVERSITY IN ST. LOUIS
School of Engineering and Applied Science
Department of Electrical & Systems Engineering

Dissertation Examination Committee:
Arye Nehorai, Chair
Murat Akcakaya
Ulugbek Kamilov
Neal Patwari
Xuan Zhang

Cognitive Radar Detection in Nonstationary Environments and Target Tracking

by

Yijian Xiang

A dissertation presented to
The Graduate School
of Washington University in
partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

August 2020
Saint Louis, Missouri

© 2020, Yijian Xiang

Contents

List of Figures	iv
List of Tables	vii
Acknowledgments	viii
Abstract	xi
1 Introduction	1
1.1 Target Detection	2
1.2 Target Tracking	2
1.3 Cognitive Radar	4
1.4 Contribution of This Work	7
1.5 Organization of This Dissertation	9
2 Cognitive Target Detection in Nonstationary Environments	10
2.1 Introduction	11
2.1.1 Related Work	11
2.1.2 Our Work	13
2.2 Target Detection Model	14
2.3 Change-Point Detection and the Extended CUSUM Algorithm	16
2.3.1 Sequential Hypothesis Test and CUSUM Algorithm	16
2.3.2 Extended CUSUM Algorithm	19
2.4 Clutter Distribution Identification	21
2.5 Adaptive Detection	23
2.6 Simulation Setup	24
2.6.1 Simulation Settings	24
2.6.2 Comparison Cases	30
2.7 Simulation Results	31
2.7.1 Performance of Change-point Detection Method	32
2.7.2 Performance of Clutter Distribution Identification Method	33
2.7.3 Performance of the Unified Adaptive Detection Framework: One Change-point	35
2.7.4 Performance of the Unified Adaptive Detection Framework: Multiple Change-points	41

2.8	Chapter Summary	46
3	Target Tracking in Cognitive Radar Networks	50
3.1	Introduction	50
3.1.1	Related Work	51
3.1.2	Our Work	53
3.2	Framework of Target Tracking in Network of Radars	54
3.2.1	Signal Model	57
3.2.2	Kinematic Model	58
3.2.3	Measurement Model	59
3.2.4	Recursive Bayesian Estimation	62
3.2.5	Adaptive Tracking and Recursive Bayesian Estimation	63
3.2.6	Target Tracking with Multiple Target Models	67
3.3	Simulation Results	70
3.3.1	Simulation Settings (a Scenario)	71
3.3.2	Simulation Results	76
3.4	Chapter Summary	78
4	Conclusions and Future Work	83
4.1	Summary and Conclusions	83
4.2	Future Work	84
	Bibliography	87
Appendix A	Proof of $\hat{t}_{AMF} \sim F(2, 2LN)$ Under the Null Hypothesis	95
Appendix B	Comparison of Ozturk Method and Our Proposed Method	97
B.1	Ozturk Algorithm	97
B.2	Simulation Results	99
B.2.1	Robustness to Location and Scale Parameters	101
B.2.2	Performance Comparison between Our Method and Ozturk Method	105
Appendix C	Derivation of (3.39) and (3.40)	108
C.1	Derivation of (3.39)	108
C.2	Derivation of (3.40)	110
Vita	113

List of Figures

1.1	A comparison of a traditional radar system and a cognitive radar system. The gray dashed box includes the basic modules of a traditional radar system. The red dashed box includes modules of a cognitive radar system.	5
1.2	An illustrative example of target detection in nonstationary environment. The target is moving across a sea coast, and the underlying clutter changes from land clutter to sea clutter.	6
1.3	A design of a cognitive radar network. The green dot denotes the target in the space. The radar network is formed by radars installed on unmanned aerial vehicles (UAVs). At every time step, the cognitive radar network designs radars' waveforms, plans their paths, and selects a subset of radars to accurately track the target.	7
2.1	An illustration of the cell under test and target absent cells (secondary data) at the k_{th} CPI.	15
2.2	An example of a change-point point detection problem. Blue points and green points are samples from a time series before and after the change point, respectively.	16
2.3	An illustration of the averaging procedure.	20
2.4	A flowchart of the proposed cognitive radar detection in nonstationary environments.	25
2.5	Performance of the change-point detection with initial clutter distribution as Gaussian at CPDB = -12.5dB.	33
2.6	Performance of the change-point detection with initial clutter distribution as Weibull at CPDB = -12.5dB.	34
2.7	Performance of the change-point detection with initial clutter distribution as K at CPDB = -12.5dB.	34
2.8	Performance of the adaptive detection framework. The initial clutter distribution is K with CPDB = -12.5dB. The preset $P_{\text{fa}} = 0.05$	38
2.9	Performance of the nonadaptive detection case. The initial clutter distribution is K with CPDB = -12.5dB. The preset $P_{\text{fa}} = 0.05$	38
2.10	Performance of the AMF case. The initial clutter distribution is K with CPDB = -12.5dB. The preset $P_{\text{fa}} = 0.05$	38
2.11	Performance of the adaptive detection framework. The initial clutter distribution is Weibull with CPDB = -12.5dB. The preset $P_{\text{fa}} = 0.01$	39

2.12	Performance of the nonadaptive detection case. The initial clutter distribution is Weibull with CPDB = -12.5dB. The preset $P_{fa} = 0.01$	39
2.13	Performance of the AMF case. The initial clutter distribution is Weibull with CPDB = -12.5dB. The preset $P_{fa} = 0.01$	39
2.14	Performance of the adaptive detection framework, in comparison with the non-adaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-10dB)→CW(-10dB)→CG(-12.5dB) with change-points at the 60, 120, and 180 _{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.05$	44
2.15	Performance of the adaptive detection framework, in comparison with the non-adaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-15dB)→CW(-7.5dB)→CG(-12.5dB) with change-points at the 60, 120, and 180 _{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.05$	44
2.16	Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-12.5dB)→CW(-12.5dB)→CG(-12.5dB) with change-points at the 60, 120, and 180 _{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.05$	45
2.17	Performance of the adaptive detection framework, in comparison with the non-adaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-10dB)→CW(-10dB)→CG(-12.5dB) with change-points at the 60, 120, and 180 _{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.005$	47
2.18	Performance of the adaptive detection framework, in comparison with the non-adaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-15dB)→CW(-7.5dB)→CG(-12.5dB) with change-points at the 60, 120, and 180 _{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.005$	47
2.19	Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-12.5dB)→CW(-12.5dB)→CG(-12.5dB) with change-points at the 60, 120, and 180 _{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.005$	48
2.20	Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-10dB)→CK(-7.5dB)→CW(-10dB)→CG(-7.5dB) with change-points at the 60, 120, and 180 _{th} CPIs. The predefined false alarm rate is $P_{fa} = 5 \times 10^{-4}$	48
2.21	Performance of the adaptive detection framework, in comparison with the non-adaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-10dB)→CK(-7.5dB)→CW(-12.5dB)→CG(-10dB) with change-points at the 60, 120, and 180 _{th} CPIs. The predefined false alarm rate is $P_{fa} = 5 \times 10^{-4}$	49

2.22	Performance of the adaptive detection framework, in comparison with the non-adaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-10dB)→CK(-12.5dB)→CW(-7.5dB)→CG(-10dB) with change-points at the 60, 120, and 180 _{th} CPIs. The predefined false alarm rate is $P_{fa} = 5 \times 10^{-4}$.	49
3.1	Graphical model (step k). The evolution of the state \mathbf{x}_k is dependent on the contextual evidence \mathbf{c}_k . Here $\mathcal{D}(\mathcal{U}_k)$ indicates the selected radars, the transmitted waveforms, and the radar states; \mathcal{Z}_k denotes noisy measurements of the state \mathbf{x}_k if $\mathcal{D}(\mathcal{U}_k)$ is applied.	55
3.2	Model settings of the illustrative example of the framework.	73
3.3	Tracking performance when process noise intensities $q_1 = q_2 = 0.1$, and the tracking time interval $T_{in} = 0.005$ s. (a) Comparison of the true and estimated trajectories of the target, (b) relative error of the position x , (c) relative error of the velocity \dot{x} , (d) relative error of the position y , (e) relative error of the velocity \dot{y} , (f) waveforms used by radars on the x and y .	79
3.4	Tracking performance when process noise intensities $q_1 = q_2 = 0.1$, and the tracking time interval $T_{in} = 0.05$ s. (a) Comparison of the true and estimated trajectories of the target, (b) relative error of the position x , (c) relative error of the velocity \dot{x} , (d) relative error of the position y , (e) relative error of the velocity \dot{y} , (f) waveforms used by radars on the x and y .	80
3.5	Tracking performance when process noise intensities $q_1 = q_2 = 10$, and the tracking time interval $T_{in} = 0.05$ s. (a) Comparison of the true and estimated trajectories of the target, (b) relative error of the position x , (c) relative error of the velocity \dot{x} , (d) relative error of the position y , (e) relative error of the velocity \dot{y} , (f) waveforms used by radars on the x and y .	81
B.1	Graphical dictionary generated by Weibull, Gaussian, and log-normal distributions.	101
B.2	Comparison of proposed method and Ozturk method for clutter distribution identification.	106

List of Tables

2.1	Clutter distribution identification accuracy with different initial clutter distributions	35
2.2	P -values of Wilcoxon rank-sum one-sided tests for false alarm rate and probability of detection after detected change-points with initial K-distributed clutter (CPDB = -12.5 dB) and preset $P_{fa} = 0.05$	41
2.3	P -values of Wilcoxon rank-sum one-sided tests for false alarm rate and probability of detection after detected change-points with initial Weibull-distributed clutter (CPDB = -12.5 dB) and preset $P_{fa} = 0.01$	41
B.1	Ozturk method performance on Weibull distributions	103
B.2	Proposed method performance on Weibull distributions	103
B.3	Ozturk method performance on Log-normal distributions	105
B.4	Proposed method performance on Log-normal distributions	105

Acknowledgments

First and foremost, I would like to express my sincere and deep gratitude to my research advisor, Dr. Arye Nehorai, for his continuous guidance and support for my research. Ever since I arrived at Washington University in St. Louis in 2014, he has given insightful and professional suggestions for my courses, research, life, and career. I always benefited from these suggestions and will continue learning from them. I would like to thank him for providing me with an excellent Ph.D. training experience that has equipped me with strong curiosity, independent and critical thinking, and self-confidence. I have always been very grateful for his patient communication, warm encouragement, and kind guidance when I have encountered obstacles and faced challenges in conducting my research.

I also would like to convey my gratitude to my collaborators, Dr. Murat Akcakaya and Dr. Satyabrata Sen, who are former students in Dr. Nehorai's lab. They are excellent researchers in the field of statistical signal processing, and they spent valuable time helping me develop ideas and address problems. I am very thankful for their constructive suggestions and great inspirations during our discussions, and I really enjoyed working with them.

I also would like to thank my dissertation defense committee members, Dr. Neal Patwari, Dr. Ulugbek Kamilov, Dr. Xuan Zhang, and Dr. Murat Akcakaya, who contributed their time, knowledge, and expertise to my dissertation. I wish to thank Dr. Neal Patwari and Dr. Zachary Feinstein for being committee members of my preliminary research exam and for their insightful comments.

I would like to thank my labmates, Alex, Jichuan, Mengxue, Mianzhi, Prateek, Zhen, Hesam, Zhenqi, Eric, and Uri for their help and support. We had many wonderful times together to discuss, exchange opinions, and collaborate. I also want to thank all my other friends who always supported me and encouraged me during my pursuit.

I owe my deepest thanks to my beloved parents for their belief, support, and love.

Yijian Xiang

Washington University in Saint Louis

August 2020

Dedicated to my parents

ABSTRACT OF THE DISSERTATION

Cognitive Radar Detection in Nonstationary Environments and Target Tracking

by

Yijian Xiang

Doctor of Philosophy in Electrical Engineering

Washington University in St. Louis, 2020

Professor Arye Nehorai, Chair

Target detection and tracking are the most fundamental and important problems in a wide variety of defense and civilian radar systems. In recent years, to cope with complex environments and stealthy targets, the concept of cognitive radars has been proposed to integrate intelligent modules into conventional radar systems. To achieve better performance, cognitive radars are designed to sense, learn from, and adapt to environments. In this dissertation, we introduce cognitive radars for target detection in nonstationary environments and cognitive radar networks for target tracking.

For target detection, many algorithms in the literature assume a stationary environment (clutter). However, in practical scenarios, changes in the nonstationary environment can perturb the parameters of the clutter distribution or even alter the clutter distribution family, which can greatly deteriorate the target detection capability. To avoid such potential performance degradation, cognitive radar systems are envisioned which can rapidly recognize the nonstationarity, accurately learn the new characteristics of the environment, and

adaptively update the detector. To achieve this cognition, we propose a unifying framework that integrates three functions: (i) change-point detection of clutter distributions by using a data-driven cumulative sum (CUSUM) algorithm and its extended version, (ii) learning/identification of clutter distribution by using kernel density estimation (KDE) methods and similarity measures (iii) adaptive target detection by automatically modifying the likelihood-ratio test and the corresponding detection threshold. We also conduct extensive numerical experiments to show the merits of the proposed method compared to a nonadaptive case, an adaptive matched filter (AMF) method, and the clairvoyant case.

For target tracking, with remarkable advances in sensor techniques and deployable platforms, a sensing system has freedom to select a subset of available radars, plan their trajectories, and transmit designed waveforms. Accordingly, we propose a general framework for single target tracking in cognitive networks of radars, including joint consideration of waveform design, path planning, and radar selection. We formulate the tracking procedure using the theories of dynamic graphical models (DGM) and recursive Bayesian state estimation (RBSE). This procedure includes two iterative steps: (i) solving a combinatorial optimization problem to select the optimal subset of radars, waveforms, and locations for the next tracking instant, and (ii) acquiring the recursive Bayesian state estimation to accurately track the target. Further, we use an illustrative example to introduce a specific scenario in 2-D space. Simulation results based on this scenario demonstrate that the proposed framework can accurately track the target under the management of a network of radars.

Chapter 1

Introduction

Target detection and tracking have long been the most relevant and important problems in a wide variety of civilian and military radar systems, such as air traffic control, navigation, and weather prediction. In these applications, target detection answers whether a target is present or not in a certain area, while target tracking predicts and estimates the trajectory of a moving target. For example, in aircraft navigation, the radar continuously reports the existence of potential objects, such as other aircraft, by detection, and then tracks the objects to avoid collision. A similar idea is also used in the emerging technology of autonomous vehicles, where, assisted by radars installed in the vehicle, detection and tracking algorithms provide position and velocity information about pedestrians, other cars, and the environments. In meteorology, radars detect atmospheric conditions and track them to predict the weather. Further, detection and tracking results can be input data for higher level processing, such as target classification and recognition, so that targets can be classified into different categories and hostile targets can be recognized. As the foundations of many real applications, target detection and tracking algorithms should be carefully designed to perform well and serve practical uses accurately.

1.1 Target Detection

In active radar systems, a radar transmits modulated electromagnetic waves to the surveillance space, and the receiver of the radar receives their echo [1]. The echo may contain a combination of the back-scattered signal from the target, some unwanted “signals” returned from objects that are not the target, and noise. The unwanted “signals” are called *clutter*, and are typically caused by returns from the ground, sea, rain, chaff, and atmospheric turbulence in the surveillance space. When clutter is present, it usually dominates the noise and can seriously hinder the performance of detection and tracking. Thus, it is critical to accurately model and capture the characteristics of the clutter constantly.

In general, in the literature and in practical use, the target detection problem in the presence of clutter and noise is formulated as a binary statistical hypothesis testing [2, 3]. The null hypothesis assumes that there is no target in a certain range bin or Doppler bin, which implies that only clutter and noise are collected by the radar. The alternative hypothesis assumes that the received echo contains the signal back-scattered from the target, which is corrupted by some clutter and noise. A test statistic is designed to compare with a threshold to decide whether a target is present or not. The design of the test statistic and the choice of the threshold are highly relevant to the underlying clutter distribution. If the clutter distribution changes, then the detection algorithm should be changed accordingly.

1.2 Target Tracking

Target tracking estimates the state of a target, e.g., its position and velocity, by using measurements received at multiple time steps in conjunction with one or more assumed

kinematic models [4, 5]. For example, radar detection may collect the range and radial Doppler information for a target at each processing step. Given these measurements, and assuming that the target is moving in a certain mode, e.g., turning mode, we can estimate the position and velocity of the target better than by using the range and radial Doppler information from only one step.

In most target tracking algorithms, the goal is to accurately estimate the posterior distribution of the target state given the measurement collected from the current time step and all the previous measurements. Recursive Bayesian estimation is widely used for this purpose. In general, a recursive Bayesian estimation algorithm updates the belief of the state at the current time step by using the belief of the state from the last time step and new measurement information collected from the current time step. Sequentially, a recursion is built for the beliefs of target states in adjacent time steps. The Kalman filter, one of the most famous algorithms for target tracking, is optimal under two key assumptions. First, it assumes that the kinematic model (i.e., the state transition model) of the target and the measurement model are linear; Second, it assumes that the process noise and the measurement noise are both Gaussian. However, for target tracking in radar and many other areas, these assumptions may not hold. For example, a marine X-band radar measures the range and the azimuth of the target, while the target state is usually modeled in the x - y coordinate. The non-linearity between the measurements and target state limits the usage of the Kalman filter. To loosen the first assumption for linearity, the extended Kalman filter (EKF) and unscented Kalman filter (UKF) have been proposed. The EKF approximates the nonlinear functions in the kinematic model and measurement model by using Taylor series expansion at the predicted target state, and the first order approximation is used in the framework of the Kalman filter. The UKF uses a deterministic sampling technique known as unscented transformation to pick a set of sigma points around the mean. Then the sigma points are

propagated through the nonlinear function to generate a new mean and covariance estimate. To further remove the second Gaussian assumption, the particle filter (PF) was proposed which uses random sampling and Monte-Carlo simulation techniques. The posterior distributions of the target state are represented by a set of random samples (particles), and then the recursive Bayesian estimation is manipulated by propagating the random samples and resampling. The merit of the Kalman filter family is that the posterior distribution of the target state can be represented by only the estimated mean and covariance, while the particle filter needs to store a large set of samples.

1.3 Cognitive Radar

In recent years, the concept of cognitive radar has been proposed, which requires that the radar system can effectively *sense* the environment, *learn* from its experience, and *adapt* to the environment by integrating more complex but intelligent modules [6–8]. Fig. 1.1 shows comparative views of a traditional radar system and a cognitive radar system. As we can see, a traditional radar system usually lacks a learning module, which can collect useful information from the environments and targets, and thus it fails to adjust its detection/tracking algorithms and communicate with its transmitters to adapt to the constantly changing environments and targets. In contrast, a cognitive radar system can avoid the target detection and tracking performance degradation by making use of the received and processed information.

For example, in target detection, due to spatial or temporal variations in radar operational scenarios, the statistical characteristics of clutter can vary enormously [9], which can greatly deteriorate the target detection capability if the radar system does not realize the changes

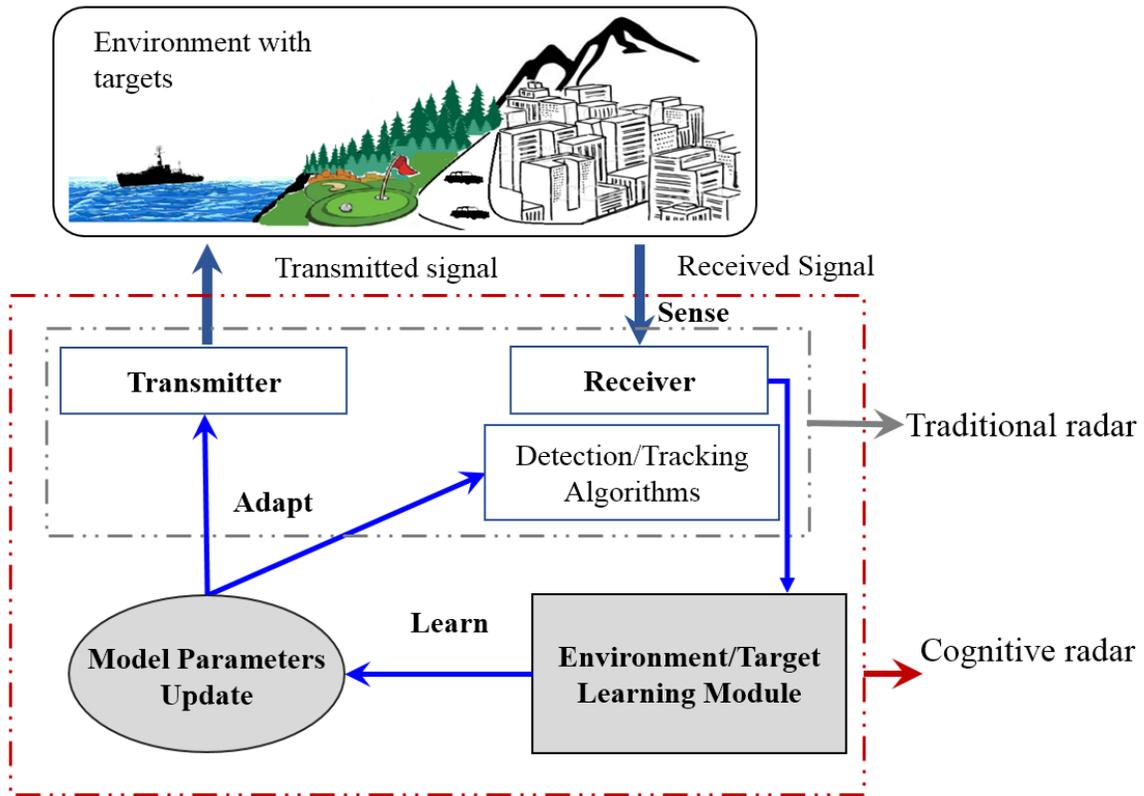


Figure 1.1: A comparison of a traditional radar system and a cognitive radar system. The gray dashed box includes the basic modules of a traditional radar system. The red dashed box includes modules of a cognitive radar system.

and maintains the non-optimized detector. An scenario of target detection in a nonstationary environment is shown in Fig.1.2. A target is moving across a sea coast, and the underlying clutter changes from land clutter to sea clutter. The characteristics of sea clutter are starkly different from those of land-clutter. Through an environment learning module, the radar system may be able to rapidly realize the changes in clutter distributions and learn the new clutter distribution. Since a detection algorithm is usually designed based on a specific clutter distribution, the radar detector can adapt its detection algorithm accordingly to maintain the optimal properties.

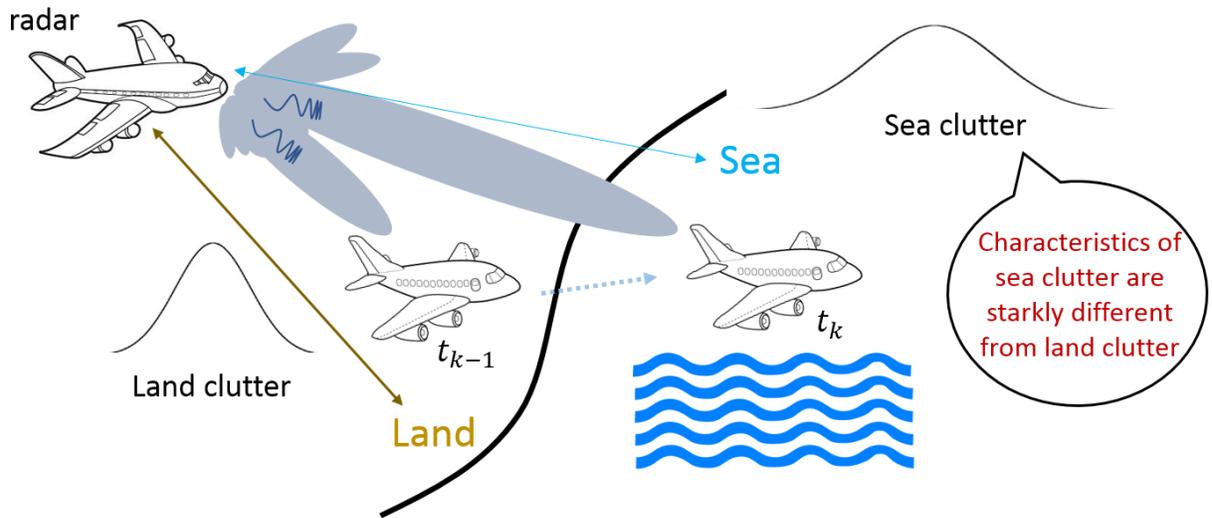


Figure 1.2: An illustrative example of target detection in nonstationary environment. The target is moving across a sea coast, and the underlying clutter changes from land clutter to sea clutter.

Another example involves target tracking. With remarkable advances in radar techniques, radars now can design various transmitted waveforms to extract more target information. Further, radar systems have evolved to include multiple static or mobile platforms that coordinate among themselves to view the target in different perspectives [10]. For example, tracking targets in urban environments using a fleet of self-controlled and self-tasked unmanned aerial vehicles (UAVs) is much more prevalent now than it was a few years ago. Thus, when a cognitive radar network receives measurements from a target, the learning model can control the transmitters to emit proper waveforms, plan the radar trajectories, and select the appropriate subset of radars, in order to accurately track the target and efficiently manage the resources on-line. An example of the cognitive radar network that includes the above functions is given in Fig.1.3, where radars are installed on mobile platforms.

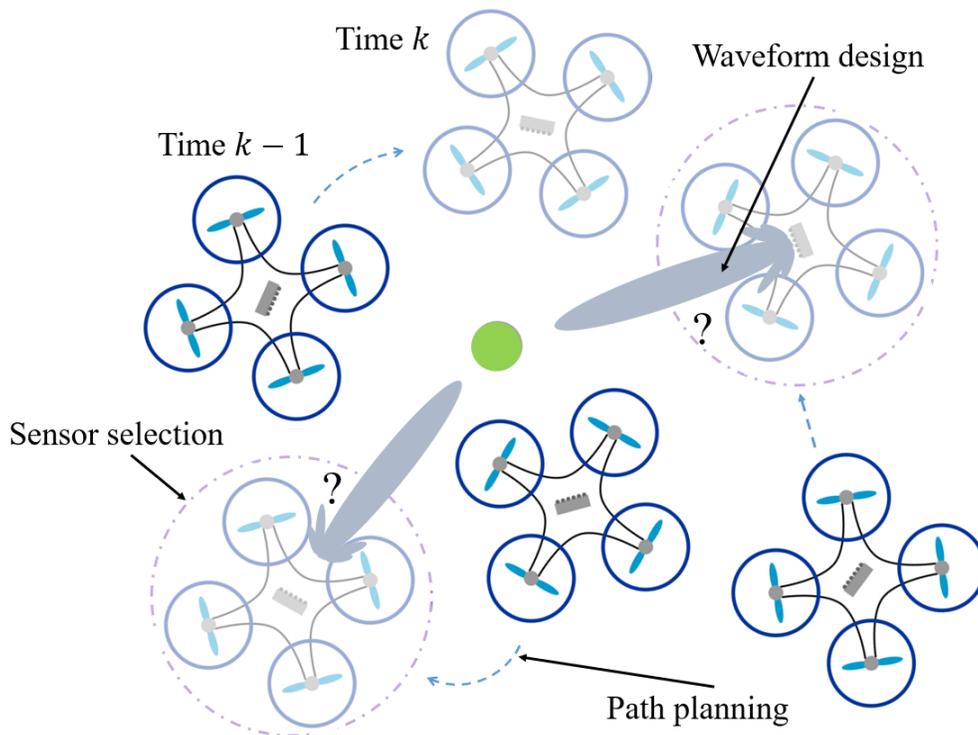


Figure 1.3: A design of a cognitive radar network. The green dot denotes the target in the space. The radar network is formed by radars installed on unmanned aerial vehicles (UAVs). At every time step, the cognitive radar network designs radars’ waveforms, plans their paths, and selects a subset of radars to accurately track the target.

1.4 Contribution of This Work

In this dissertation, we design frameworks for target detection in nonstationary environments and for target tracking in cognitive radar networks. We also conduct extensive numerical experiments to demonstrate the advantages of the proposed frameworks. Our main contributions are summarized as follows.

Cognitive radar detection in nonstationary environments: In this work, we propose a unifying framework for adaptive target detection in nonstationary environments, integrating change-point detection, clutter distribution learning, and adaptive detection. To the best

of our knowledge, this is the first unified target detection mechanism that truly implements the cognitive radar framework in detecting, learning, and adapting to changes in nonstationary environments. Specifically, we detail and apply an extended cumulative sum (CUSUM) algorithm to detect multiple change-points that arise in various common complex clutter distributions. We also apply a learning algorithm to identify the new clutter distribution after the extended CUSUM algorithm realizes the change-point, and we compare it with the classical Ozturk algorithm. We first evaluate the detection performance by focusing on one change-point, and use Wilcoxon rank-sum tests to quantitatively show the statistical significance of the improvements. We next examine the dynamic behaviors of the methods along the time axis while multiple change-points are present, which reflects the new framework’s capability for real-time applications.

Target tracking in cognitive radar networks: In this work, we propose a target tracking framework that simultaneously designs the transmit waveform, plans the radar trajectory, and selects the appropriate radar subset. Building on the established theories of dynamic graphical models (DGM) and recursive Bayesian state estimation (RBSE), we choose the expected cross-entropy as the objective function and solve a combinatorial optimization problem to select the optimal subset of radars that transmit well-designed waveforms from their best locations for receiving the most informative measurements during the next tracking instant. The received measurements associated with history information are then processed by a tracker to achieve a more accurate target state estimation. Additionally, we validate the proposed framework by constructing an illustrative case, where radars with certain movement constraints track a single target in a 2-D space. Specifically, we choose this simplified case as it is a representation of a large-scale network of cognitive radars which have certain constraints on their coverage areas and paths. These constraints reflect real-time scenarios, especially when each radar in the network is aware of the locations of the other radars in the

network and when overlapping coverage among radars along same path is avoided by using the location information, assuming that the target is in the far-field.

1.5 Organization of This Dissertation

The rest of this dissertation is organized as follows. In Chapter 2, we present our proposed methods for target detection in nonstationary environments, which involves change-point detection, clutter distribution learning, and adaptive detection. In Chapter 3, we introduce the framework for target tracking in cognitive radar networks, jointly considering waveform design, path planning, and radar selection. In Chapter 4, we summarize our work and discuss possible future work to improve the proposed methods for target detection and tracking.

Chapter 2

Cognitive Target Detection in Nonstationary Environments

In this chapter, we introduce our proposed framework for cognitive target detection in non-stationary environments.¹

We begin with a common target detection model in the presence of clutter. The target detection problem is formulated as a statistical hypothesis testing. Then we introduce the mathematical formulation of the change-point detection problem and our proposed extended CUSUM algorithm that detects changes in the clutter distributions. Next, we present the clutter distribution learning method that identifies the new underlying clutter distribution, so that we can adaptively adjust the detection algorithm. Finally, with extensive numerical experiments, we demonstrate the validity of our proposed unifying framework, taking into consideration potential effects caused by the change-point detection module and clutter distribution learning module.

¹This chapter is based on Y. Xiang, M. Akcakaya, S. Sen, and A. Nehorai, “Target detection via cognitive radars using change-point detection, learning, and adaptation,” in revision for *Circuits, Systems, and Signal Processing*.

2.1 Introduction

In this section, we briefly introduce related previous work and our work on target detection in nonstationary environments.

2.1.1 Related Work

In previous work, many radar detection algorithms that assume a stationary environment (clutter) have been proposed [2, 3]. However, in practice, due to spatial or temporal variations in radar operational scenarios, the statistical characteristics of the clutter can vary enormously [9], which can greatly deteriorate the target detection capability if the radar system does not realize the changes and continues to use a non-optimized detector. Nonstationary clutter is not rare in real applications, and it can have many causes, for example, changes of allocation positions of radar beams, weather conditions, and versatile hostile countermeasures and jamming. These factors can perturb the parameters of the clutter distributions, or even worse, can alter the underlying clutter distribution family.

The Gaussian assumption for clutter distribution is common in the literature, due to its simplicity and roots in the central limit theorem. This assumption is feasible if the range cell of interest contains a large number of scatterers. However, previous papers have pointed out that the Gaussian clutter model is not well-behaved in some scenarios [11–13], for example, target detection with sea clutter, whose distribution usually has a heavier tail than the Gaussian distribution. Further, to improve target detection in these scenarios, more sophisticated clutter models have been proposed, such as the compound-Gaussian model [14–16], spherically invariant random vector (SIRV) model [17, 18], and log-normal distributed model [19].

However, most of these proposed techniques derive the optimal detectors by relying on one specific clutter model. They imply stationary environments, or they assume that the proposed clutter model can sufficiently cope with the parameter-changes of the clutter.

To address the target detection problem in nonstationary environments and maintain optimal or nearly optimal performance, radar systems should minimize the mismatch between the assumed clutter model and the real clutter measurements. In other words, the radar systems need to adjust the threshold of the detector if the parameters of the underlying distribution of the clutter fluctuate, or even redesign the detector if the clutter distribution starts to follow a different family. In recent years, the concept of cognitive radar has been proposed, which requires that the radar system can effectively *sense* the environment, *learn* from its experience, and *adapt* to the changes by integrating more complex but intelligent modules [6–8]. In implementing this conceptual cognitive radar, several methods have been proposed to address the clutter parameter estimation and target detection problem. For example, with the help of a pre-learned dictionary that contains possible distributions of clutter, radar systems can intelligently choose the model from the dictionary by using a distribution identification method, such as the Ozturk method [20,21]. Furthermore, [22,23] also extend the Ozturk method and propose a clutter identification method and detection threshold setting method based on an artificial neural network. Other works utilize auto-regressive models and prior information of the covariance matrix to develop a knowledge-aided Bayesian covariance matrix estimation method [24–27]. In addition, still others assign exponentially decaying weights to the clutter covariance estimates and adapt the covariance matrix in space-time adaptive processing (STAP) by an adaptive weighted sum technique [28,29].

However, these methods usually do not answer the questions of whether there is a change in the sequentially received clutter measurements and when the change arises. To be considered

cognitive, radar systems need to (i) rapidly detect the change points of the characteristics of the clutter, (ii) accurately learn the new distributions of the clutter, and (iii) adaptively apply the optimal or nearly optimal detectors. Our previous works have separately addressed these issues, but with strong assumptions and limited analyses.

2.1.2 Our Work

To address the first issue of the cognitive radar, in our previous work [30], we have used a data-driven cumulative sum (CUSUM) algorithm [31,32] and its extended version for detecting changes in the clutter distribution, and we showed a potential improvement in performance if the clutter distribution after the change is accurately identified. However, the work in [30] was limited to investigating the case in which the clutter distribution changes from a Gaussian model to a particular compound Gaussian model. It provided only basic scenarios and did not consider various clutter distribution models. Further, the target detection performance of the proposed method was compared to only the conventional nonadaptive method, and perfect knowledge was assumed for the change-point, so that the effects of the change-point detection algorithm on the detection performance were not fully considered. It also did not learn the clutter distribution after the change-point, and the clutter distributions before and after the change-point were assumed to be known when investigating the target detection performance. In this dissertation, we not only provide versatile simulation scenarios and comparisons, but also remove these assumptions for evaluating adaptive detection performance.

To address the second requirement of the cognitive radar, we proposed a clutter distribution identification method based on sparse theory and a kernel density estimation method [33],

investigated the parameter selection and model selection [34], and compared our proposed method with the Ozturk method [35]. However, in these works [33–35], we did not consider the change-point detection problem; in other words, we implicitly assumed a perfect change-point detection result. In order to achieve an effective real-time cognitive radar application, the continuous examination of change-points should be highly coupled with clutter distribution learning. Further, these papers did not employ the adaptive detection module after the clutter distribution learning, rather, they addressed this learning as an independent problem. In this dissertation, we overcome these limitations and develop a unifying framework for adaptive detection in nonstationary environments.

2.2 Target Detection Model

In radar target detection, after receiving the echo for each coherent processing interval (CPI), the radar system usually passes the signal through a matched filter and tests the existence of the target in the cells of interest (COI). Accordingly, we have the following hypothesis testing problem for a cell under test (CUT) in the k_{th} CPI:

$$\begin{cases} \mathcal{H}_0 : \mathbf{y}_0^{(k)} = \mathbf{n}_c^{(k)} \\ \mathcal{H}_1 : \mathbf{y}_0^{(k)} = \mathbf{p}^{(k)} + \mathbf{n}_c^{(k)} \end{cases}, \quad (2.1)$$

where $\mathbf{y}_0^{(k)}$ is the received signal in the CUT, $\mathbf{p}^{(k)}$ denotes the signal corresponding to the target response, and $\mathbf{n}_c^{(k)}$ denotes the clutter in the CUT. If N pulses are considered in one CPI, then each of the above vectors is of dimension N . Note that the distribution of $\mathbf{n}_c^{(k)}$ is not known to the radar in advance, and usually the radar collects secondary target-absent data to learn the distribution of $\mathbf{n}_c^{(k)}$. For example, under a homogeneous assumption in

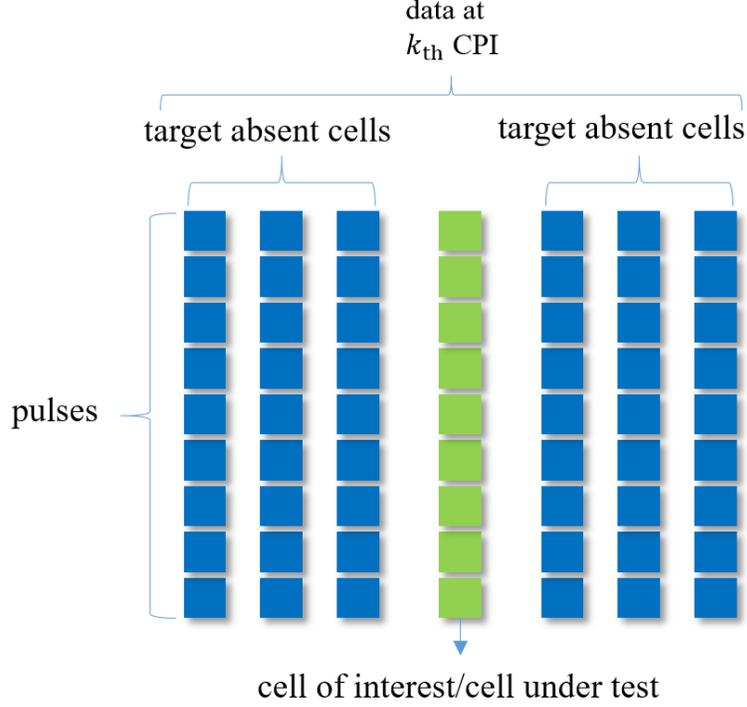


Figure 2.1: An illustration of the cell under test and target absent cells (secondary data) at the k_{th} CPI.

space-time adaptive processing (STAP), the secondary data are usually selected near the CUT and do not overlap with its guard cells. Here we assume that the secondary data are available and denote them as

$$\mathcal{Y}_c^{(k)} = \{\mathbf{y}_{c1}^{(k)}, \mathbf{y}_{c2}^{(k)}, \dots, \mathbf{y}_{cL}^{(k)}\}, \quad (2.2)$$

where elements in the set of secondary data $\mathcal{Y}_c^{(k)}$ are independent, and L is the total number of elements in $\mathcal{Y}_c^{(k)}$. Further, each element in $\mathcal{Y}_c^{(k)}$ follows the same clutter distribution as that of $\mathbf{n}_c^{(k)}$. Note that in this dissertation, $\mathbf{n}_c^{(k)}$ may follow different distributions or even different families of distributions as the time k proceeds. An illustrative example of CUT and the secondary data is given in Fig. 2.1.

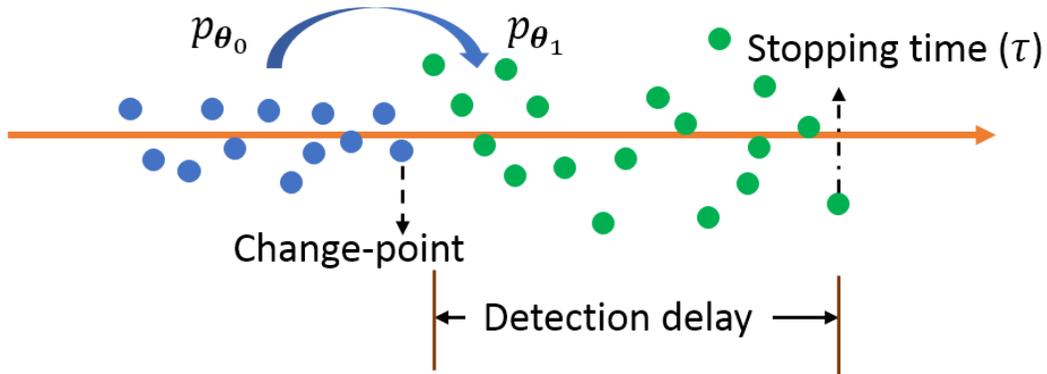


Figure 2.2: An example of a change-point point detection problem. Blue points and green points are samples from a time series before and after the change point, respectively.

2.3 Change-Point Detection and the Extended CUSUM Algorithm

2.3.1 Sequential Hypothesis Test and CUSUM Algorithm

Change-point detection aims to identify the time instants when the probability distribution of a time series changes. An illustrative example is given in Fig. 2.2. Blue points and green points are samples from a time series before and after the change point, respectively. The distribution of the samples changes from p_{θ_0} to p_{θ_1} . In this figure, we can see that the variance of the time series changes at the change point. In general, a change-point detection algorithm runs along the time axis, and at each time step, it checks whether there is a change point in the previous time step or not. If a change point is found by the algorithm, then it stops. It is a common goal to minimize the estimation delay, i.e., the time interval between the change point and the stopping time, under certain constraints.

Mathematically, the change-point detection problem for a time series $\{\mathbf{z}^{(k)} : k = 1, 2, 3, \dots\}$ is formulated as a sequential hypothesis testing problem, as below:

$$\left\{ \begin{array}{l} \mathcal{D}_0 : \mathbf{z}^{(k)} \sim p_{\theta_0} \quad k = 1, 2, \dots \\ \mathcal{D}_1 : \mathbf{z}^{(k)} \sim p_{\theta_0} \quad k = 1, 2, \dots, k_0 \\ \quad \quad \quad \mathbf{z}^{(k)} \sim p_{\theta_1} \quad k = k_0 + 1, k_0 + 2, \dots \end{array} \right. , \quad (2.3)$$

where k_0 is an unknown time instant after which the time series starts to follow a different distribution, i.e., a change point, and θ_0 and θ_1 are parameters for two distributions. In order to solve the above test, we adopt the optimal rule as follows [36, 37]:

$$\begin{aligned} \tau &= \arg \min_{\tau} \sup_{n \geq 1} \text{ess sup} \mathbb{E}_n [(\tau - n)^+ | \mathbf{z}^{(1)}, \dots, \mathbf{z}^{(n-1)}] \\ &\text{s.t. } \mathbb{E}_{\infty}[\tau] \geq \alpha, \end{aligned} \quad (2.4)$$

where τ is the stopping time, i.e., the time that the decision can be made by a rule to claim \mathcal{D}_1 using the received data. The notation $(x)^+ = \max(0, x)$. Note that $\mathbb{E}_n[\cdot]$ is the expectation operation assuming that the change-point k_0 arises at time n , and $\mathbb{E}_{\infty}[\cdot]$ is the expectation operation assuming that there is no change-point in the time series. Here α is a user-defined parameter to ensure that the decision policy cannot have a very early stop on average if there is no change-point. The essential supremum, ess sup , of a random variable X is the smallest number r such that $P(X \leq r) = 1$, thus leading to the largest average detection delay $(\tau - n)^+$ under the worst realization of $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(n-1)}\}$.

In general, the optimization problem solves a minimax problem to seek a stopping policy that minimizes change-point detection delay. A solution to the above problem is found in [31,32]:

$$\begin{aligned} \tau &= \inf \{m \geq 1 : g(\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}) \geq b_c\} \\ g(\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}) &\stackrel{\text{def}}{=} R_m - \min_{1 \leq k \leq m} R_k, \end{aligned} \quad (2.5)$$

where b_c is a predefined threshold for determining the stopping time, and $R_k = \sum_{i=1}^k \ln \frac{p_{\theta_1}(\mathbf{z}^{(i)})}{p_{\theta_0}(\mathbf{z}^{(i)})}$ is the cumulative sum of the likelihood ratio. The above method is referred as the cumulative sum (CUSUM) algorithm in the literature.

Note that the stopping time τ depends on the threshold b_c , the probability distribution before the change-point p_{θ_0} , and the probability distribution after the change-point p_{θ_1} . A data-driven method was proposed in [30] to determine the threshold b_c , by assuming the existence of the calibration/training data $\mathcal{Z}_t = \{\mathbf{z}_{t1}, \dots, \mathbf{z}_{tJ}\}$ where each element shares the same probability distribution as that before the change-point, i.e., p_{θ_0} . In real radar applications, we can collect more secondary data to form the calibration data, or generate some data if we have identified the underlying clutter distribution. Denoting $g(\mathbf{z}_{t1}, \dots, \mathbf{z}_{tj}) = R_j - \min_{1 \leq k \leq j} R_k$, we can train the threshold as $b_c = \max_{1 \leq j \leq J} g(\mathbf{z}_{t1}, \dots, \mathbf{z}_{tj})$. In the original version of the CUSUM algorithm, the distributions p_{θ_0} and p_{θ_1} are known. However, in practice, they are not always available. There are many technique to extend the CUSUM algorithm. In this dissertation, we use the asymptotically Gaussian property of the mean of i.i.d. random variables to extend the CUSUM algorithm.

2.3.2 Extended CUSUM Algorithm

First, we take the amplitude/modulus of the secondary data shown in (2.2), i.e.,

$$\mathbf{a}_{cl} = |\mathbf{y}_{cl}|, \text{ for } l = 1, 2, \dots, L, \quad (2.6)$$

where the superscript (k) is omitted here for generality, and the modulus operation $|\cdot|$ is elementwisely applied to the data vector. Then, we use $a_{cl,i}$ to denote the i th entry of \mathbf{a}_{cl} , and average every adjacent M entries of \mathbf{a}_{cl} to have

$$\bar{\mathbf{a}}_{cl} = \frac{1}{M} \sum_{d=1}^M [a_{cl,d}, a_{cl,d+M}, \dots, a_{cl,d+M*(\frac{N}{M}-1)}]^\top, \quad (2.7)$$

where M can be selected by the user so that N/M is an integer. Here $\{\bar{\mathbf{a}}_{cl} : l = 1, 2, \dots, L\}$ are approximately regarded as i.i.d. Gaussian vectors. Further, \mathbf{z} , shown in the original CUSUM algorithm at every time instant, is given by

$$\mathbf{z} = [\bar{\mathbf{a}}_{c1}^\top, \bar{\mathbf{a}}_{c2}^\top, \dots, \bar{\mathbf{a}}_{cL}^\top]^\top. \quad (2.8)$$

To be clear, both time series $\mathbf{z}^{(k)}$ and the calibration data $\{\mathbf{z}_{tj}\}$ are formulated as in (2.8).

An illustration for the above averaging is given in Fig. 2.3.

Next, we use the calibration data \mathcal{Z}_t to estimate the parameters $\boldsymbol{\theta}_0$ of the Gaussian distribution. In general, $\boldsymbol{\theta}_1$ can be determined by the extremas of the confidence interval of $\boldsymbol{\theta}_0$. To further clarify the idea, we suppose for simplicity that, after the averaging process, $\mathbf{z}_{t,j}$ follows $\mathcal{N}(\mu_0 \mathbf{1}_{N/M*L}, \sigma_0^2 \mathbf{I}_{N/M*L})$, where $\mathbf{1}_{N/M*L}$ is a (NL/M) -dimension column vector with all entries equal to 1, and $\mathbf{I}_{N/M*L}$ is an identity matrix of dimension $NL/M \times NL/M$. Thus,

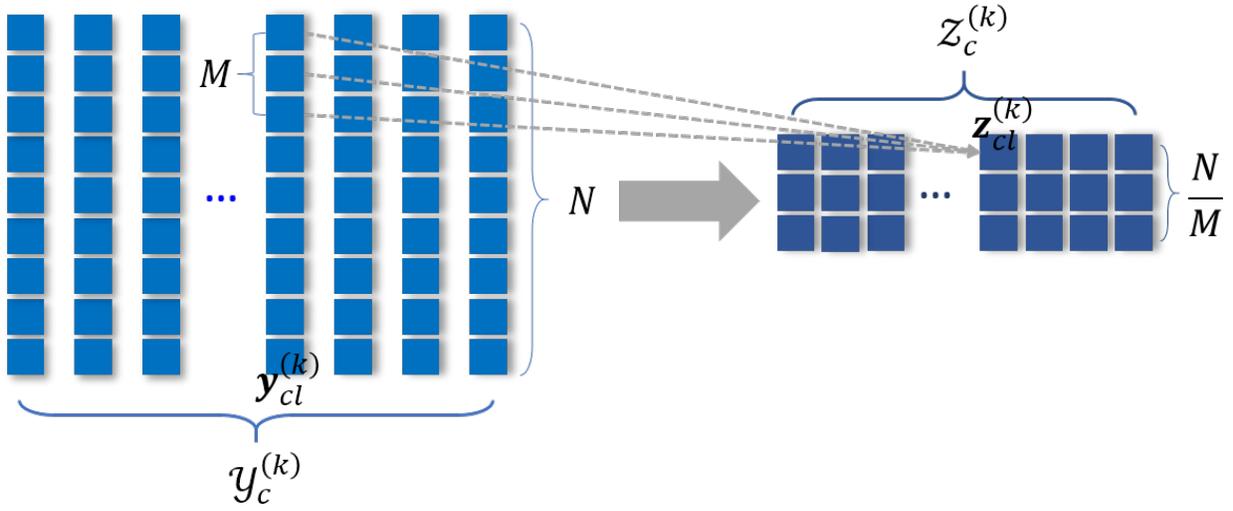


Figure 2.3: An illustration of the averaging procedure.

for $\boldsymbol{\theta}_0 = (\mu_0, \sigma_0^2)$, we have the estimation

$$\hat{\mu}_0 = \frac{1}{NLJ/M} \sum_{j=1}^J \mathbf{1}_{NL/M}^\top \mathbf{z}_{t,j}, \quad (2.9)$$

$$\hat{\sigma}_0^2 = \frac{1}{NLJ/M - 1} \times \sum_{j=1}^J (\mathbf{z}_{t,j} - \hat{\mu}_0 \mathbf{1}_{NL/M})^\top (\mathbf{z}_{t,j} - \hat{\mu}_0 \mathbf{1}_{NL/M}). \quad (2.10)$$

Now, considering the 95% confidence interval extremas of the mean and variance, we have

$$\mu_{0l} = \hat{\mu}_0 - t_{NLJ/M-1}(0.975) \frac{\hat{\sigma}_0}{\sqrt{NLJ/M}} \quad (2.11)$$

$$\mu_{0u} = \hat{\mu}_0 + t_{NLJ/M-1}(0.975) \frac{\hat{\sigma}_0}{\sqrt{NLJ/M}} \quad (2.12)$$

$$\sigma_{0l}^2 = \frac{(NLJ/M - 1) \hat{\sigma}_0^2}{\chi_{NLJ/M}^2(0.025)} \quad (2.13)$$

$$\sigma_{0u}^2 = \frac{(NLJ/M - 1) \hat{\sigma}_0^2}{\chi_{NLJ/M}^2(0.975)}. \quad (2.14)$$

Algorithm 1: Change-Point Detection

- Input** : Calibration data \mathcal{Z}_t , and secondary data at time m , $\mathcal{Y}_c^{(m)}$, for $m = 1, 2, \dots$
- 1 Use the calibration data to train the parameter $\boldsymbol{\theta}_0$, four parameters $\boldsymbol{\theta}_{1,ll}$, $\boldsymbol{\theta}_{1,ul}$, $\boldsymbol{\theta}_{1,lu}$, and $\boldsymbol{\theta}_{1,uu}$, and four corresponding thresholds b_c .
 - 2 $m = 0$;
 - 3 **repeat**
 - 4 $m = m + 1$;
 - 5 Collects the secondary data $\mathcal{Y}_c^{(m)}$ at time step m ;
 - 6 Generate $\mathbf{z}^{(m)}$ as in (2.6), (2.7), and (2.8);
 - 7 **until** $g(\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}) \geq b_c$ for any $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1)$ pair;
 - 8 A change-point is found.
-

where $t(\cdot)$ and $\chi^2(\cdot)$ are coefficients dependent on the t -distribution and χ^2 -distribution, respectively [38].

Now, we use $\boldsymbol{\theta}_0 = (\hat{\mu}_0, \hat{\sigma}_0^2)$ and four different $\boldsymbol{\theta}_1$, i.e., $\boldsymbol{\theta}_{1,ll} = (\mu_{0l}, \sigma_{0l}^2)$, $\boldsymbol{\theta}_{1,ul} = (\mu_{0u}, \sigma_{0l}^2)$, $\boldsymbol{\theta}_{1,lu} = (\mu_{0l}, \sigma_{0u}^2)$, and $\boldsymbol{\theta}_{1,uu} = (\mu_{0u}, \sigma_{0u}^2)$, to compute four $g(\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)})$ in parallel. Once the threshold of change-point detection is exceeded, i.e., one or more $g(\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}) \geq b_c$, the change-point detection algorithm stops and a clutter distribution identification module (to be introduced in Section 2.4) takes over. We summarize the above process in Algorithm 1.

2.4 Clutter Distribution Identification

In general, for clutter distribution identification, we assume that there exists a pre-trained dictionary \mathbf{D} , containing versatile clutter distributions that the radar may meet in practice. Once the radar system detects a change-point, it identifies the underlying clutter distribution as one distribution in the dictionary, based on the collected clutter data.

In [33], a method inspired from sparse recovery was proposed for clutter distribution identification, i.e., the following optimization problem is solved:

$$\hat{\boldsymbol{\gamma}} = \operatorname{argmin} \|g(\mathbf{x}) - \mathbf{D}\boldsymbol{\gamma}\|_2^2, \text{ subject to } \|\boldsymbol{\gamma}\|_0 \leq C, \quad (2.15)$$

where $\hat{\boldsymbol{\gamma}}$ is the estimated coefficient vector denoting atoms chosen from the dictionary, and C is a sparsity level defined by the users. The dictionary $\mathbf{D} = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_Z(\mathbf{x})]$, and $f_z(\mathbf{x})$ is the probability density function (pdf) pre-learned in advance for $z = 1, 2, \dots, Z$. Note that $g(\mathbf{x})$ is the pdf of the clutter distribution after the change-point, learned from the clutter secondary data by using kernel density estimation (KDE),

$$g(\mathbf{x}) = \frac{1}{S} \sum_{i=1}^S K_H(\mathbf{x} - \mathbf{y}_i), \quad (2.16)$$

where $\{\mathbf{y}_i\}$ are clutter data samples, \mathbf{x} denotes any value that a clutter sample can take, S is the total number of samples, and $K_H(\cdot)$ is a kernel function associated with a scale-relevant parameter H . Further, $g(\mathbf{x})$ and $\{f_z(\mathbf{x})\}$ are discretized and vectorized, and each of $\{f_z(\mathbf{x})\}$ is normalized to be a unit vector. If $C = 1$ or $C > 1$, $\hat{\boldsymbol{\gamma}}$ or the dominant element of $\hat{\boldsymbol{\gamma}}$ can identify one of the clutter distributions in \mathbf{D} . From another perspective, the optimization in (2.15) identifies the largest projection of $g(\mathbf{x})$ on the pdfs in \mathbf{D} . Denoting the vectorized $g(\mathbf{x})$ and $\{f_z(\mathbf{x})\}$ as \mathbf{g} and \mathbf{f}_z , we summarize the process in Algorithm 2.

In addition, it is also possible to use other similarity/distance measures to compute the similarity between $g(\mathbf{x})$ and $f_z(\mathbf{x})$. It is worth noting that we investigated and applied three $L1$ -norm type probabilistic distance measures in our work [39] for this clutter distribution identification problem.

Algorithm 2: Clutter Distribution Identification

Input : Clutter secondary data $\{\mathbf{y}_i\}$, pre-trained dictionary $\mathbf{D} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_Z]$, and kernel function $K_H(\cdot)$.

Output: I , the index of a distribution in \mathbf{D} that is the best match of the distribution of $\{\mathbf{y}_i\}$.

- 1 Use kernel density estimation to estimate the distribution of $\{\mathbf{y}_i\}$, i.e.,

$$g(\mathbf{x}) = \frac{1}{S} \sum_{i=1}^S K_H(\mathbf{x} - \mathbf{y}_i);$$

- 2 Discretize and vectorize $g(\mathbf{x})$ into \mathbf{g} by sampling the function at equal intervals;
 - 3 $I = \arg \max_{i \in \{1, 2, \dots, Z\}} \mathbf{g}^\top \mathbf{f}_i;$
-

2.5 Adaptive Detection

The likelihood-ratio test is common in radar detection, because it reaches the optimal decision that maximizes the probability of detection when the false alarm rate is fixed or less than a fixed level α . It is given by

$$\Phi(\mathbf{y}_0, p_\theta) = \frac{f(\mathbf{y}_0 | \mathcal{H}_1, p_\theta)}{f(\mathbf{y}_0 | \mathcal{H}_0, p_\theta)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} b, \quad (2.17)$$

where $f(\mathbf{y}_0 | \mathcal{H}_i, p_\theta)$ is the likelihood function of \mathbf{y}_0 under the hypothesis \mathcal{H}_i (for $i = 0, 1$), when the clutter data follow a distribution p_θ . We recall that here \mathbf{y}_0 denotes the received data in the cell of interest, \mathcal{H}_0 and \mathcal{H}_1 are hypotheses given in (2.1). In addition, b is a threshold calculated using a predefined false alarm rate α . In other word, we solve the threshold $\bar{b}(p_\theta)$ from $P_{\text{FA}} = P(\Phi(\mathbf{y}_0, p_\theta) > b | \mathcal{H}_0, p_\theta) = \alpha$. For example, by using the notations in (2.1), if the clutter sample \mathbf{n}_c follows a complex Gaussian distribution, i.e., $p_\theta \sim \mathcal{CN}(0, \sigma^2 \mathbf{I})$, then we can derive that $\Phi(\mathbf{y}_0, p_\theta) = \exp \left[\frac{1}{\sigma^2} (2 \text{Re}\{\mathbf{p}^H \mathbf{y}_0\} - \mathbf{p}^H \mathbf{p}) \right]$. To solve the threshold $\bar{b}(p_\theta)$ from a predefined level α , i.e., to solve $P_{\text{FA}} = P(\Phi(\mathbf{y}_0, p_\theta) > b | \mathcal{H}_0, p_\theta) = \alpha$, we usually need to use some numerical methods like numerical integral and interpolation, and cannot have a closed form for a general clutter distribution. If the clutter distribution p_θ changes, both the

Algorithm 3: Adaptive Target Detection

Input : An initial detector and its corresponding threshold, a pre-learned clutter distribution dictionary, and calibration data for the change-point detection.

```
1 while radar is operating do
2   | if change-point detection is triggered then
3   |   | Implement clutter distribution identification method to find the matching
4   |   | distribution in the dictionary;
5   |   | Use the new detector and threshold based on the newly learned distribution;
6   |   | Collect the calibration data for the next change-point detection;
7   | else
   |   | Continue using the detector and threshold;
```

detector $\Phi(\mathbf{y}_0, p_{\theta})$ and the threshold $\bar{b}(p_{\theta})$ need to be carefully addressed to maintain the constant false alarm rate α . In other words, once a change-point is detected and the clutter distribution after the change is identified, the detector and the corresponding threshold must be modified from $\Phi(\mathbf{y}_0, p_{\theta_0})$ and $\bar{b}(p_{\theta_0})$ to $\Phi(\mathbf{y}_0, p_{\theta_1})$ and $\bar{b}(p_{\theta_1})$.

The framework of adaptive target detection is summarized in Algorithm 3. To further illustrate the framework, we show a flowchart for the proposed framework in Fig.2.4.

2.6 Simulation Setup

2.6.1 Simulation Settings

In this section, we first introduce the clutter model, some common clutter distributions, and the target model that we used in the simulation. We further describe the simulation details for the numerical experiments.

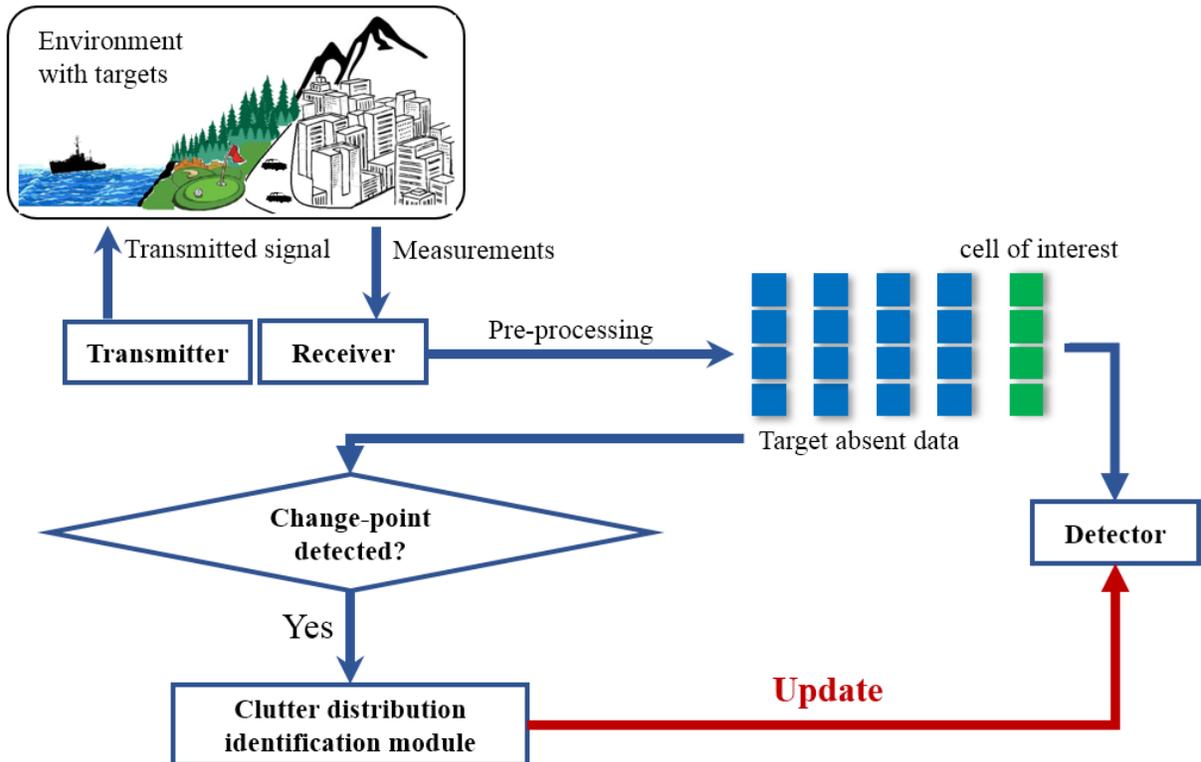


Figure 2.4: A flowchart of the proposed cognitive radar detection in nonstationary environments.

2.6.1.1 Clutter Model

In order to demonstrate the performance of the proposed adaptive target detection framework, we use a simplified but illustrative clutter model in a simulation. We assume that the entries of the clutter vector in each range cell are independent, which means that entries in $\mathbf{n}_c^{(k)}$ are independent. This assumption also applies to the secondary data, because the secondary data has the same distribution as that of $\mathbf{n}_c^{(k)}$. Further, each entry n (corresponding to one pulse in one range cell) has the following form:

$$n = \rho e^{i\psi} = c + id, \quad (2.18)$$

where ρ is the real amplitude of the clutter, and the phase $\psi \sim \mathcal{U}[0, 2\pi)$ due to the random reflections on the surfaces of the clutter-scatters. Further, ρ and ψ are independent. Readers can recognize that one-dimensional spherical invariant random vector (SIRV) is a special case of (2.18) [17, 23]. Note that the relation between c , d , ρ , and ψ is as follows:

$$c = \rho \cos \psi, \quad (2.19)$$

$$d = \rho \sin \psi. \quad (2.20)$$

Denoting that $n \sim f_n(n)$, $\rho \sim f_\rho(\rho)$, and $(c, d) \sim f_{c,d}(c, d)$, we have the relation of pdfs, i.e.,

$$\begin{aligned} f_n(n) &= f_{c,d}(c, d) = f_{\rho,\psi}(\rho(c, d), \psi(c, d)) \left| \frac{\partial(\rho, \psi)}{\partial(c, d)} \right| \\ &= \frac{1}{2\pi} f_\rho(\sqrt{c^2 + d^2}) \left| \frac{\partial(c, d)}{\partial(\rho, \psi)} \right|^{-1} \\ &= \frac{1}{2\pi} f_\rho(|n|) \frac{1}{|n|}. \end{aligned} \quad (2.21)$$

For the model in (2.18), the complex clutter distribution depends on only the distribution of its real amplitude. Thus, for Algorithm 2 in this simulation, we can generate the dictionary and the clutter distribution awaiting identification with the one-dimensional real distribution rather than the high-dimensional complex distribution, which may speed the simulation. In other words, for Algorithm 2, we use only the real amplitudes of the one-dimensional clutter samples to generate all the pdfs. In practical applications, one can flexibly modify the algorithm according to the particular structure of the clutter data.

2.6.1.2 Clutter Distributions and Parameters

The following distributions are considered in our simulation:

- Complex Gaussian (CG) distribution:

$$f_n(n; \sigma_G) = \frac{1}{\pi\sigma_G^2} e^{-|n|^2/\sigma_G^2}, \quad (2.22)$$

where the real and imaginary parts are sampled from two i.i.d. independent Gaussian distributions, respectively, i.e., $n \sim \mathcal{N}(0, \sigma_G^2/2) + j\mathcal{N}(0, \sigma_G^2/2)$. Note that

$$E|n|^2 = \sigma_G^2. \quad (2.23)$$

- Complex Weibull (CW) distribution:

$$f_n(n; \lambda_W, k_W) = \frac{1}{2\pi|n|} \frac{k_W}{\lambda_W} \left(\frac{|n|}{\lambda_W} \right)^{k_W-1} e^{-(|n|/\lambda)^{k_W}}, \quad (2.24)$$

where the λ_W is the scale parameter, and k_W is the shape parameter. Note that the amplitude $|n|$ is sampled from $\text{Weibull}(\lambda_W, k_W)$, while its phase ψ is sampled from $\mathcal{U}[0, 2\pi)$. Note that

$$E|n|^2 = \lambda_W^2 \Gamma(1 + 2/k_W). \quad (2.25)$$

- Complex K (CK) distribution:

$$f_n(n; \sigma_K, k_K, \theta_K) = \frac{2\theta_K^{-\frac{1+k_K}{2}} (|n|/\sigma_K)^{k_K}}{\pi|n|\Gamma(k_K)\sigma_K} K_{k_K-1} \left(\frac{2|n|}{\sqrt{\theta_K}\sigma_K} \right), \quad (2.26)$$

where $\Gamma(\cdot)$ is the Gamma function, and $K_\alpha(\cdot)$ is the modified Bessel function of the second kind. Let an auxiliary variable μ be sampled from $\text{Gamma}(k_K, \theta_K)$, where k_K is the shape parameter and θ_K is the scale parameter. Let another auxiliary variable h be sampled from $\mathcal{N}(0, \sigma_K^2/2) + j\mathcal{N}(0, \sigma_K^2/2)$. Then a complex-K-distributed sample can be generated by $n = \sqrt{\mu}h$ [13, 14]. Note that

$$E|n|^2 = k_K \theta_K \sigma_K^2. \quad (2.27)$$

2.6.1.3 Target Model

In our simulation, we assume that we know the target response in advance, and denote $\mathbf{p}^{(k)} = a\mathbf{1}$ without loss of generality and we fix $a = 1$ for the simulation. We represent the clutter power in dB (CPDB) level, and define

$$\text{CPDB} = 10 \log_{10} \left(\frac{1}{E|n|^2} \right), \quad (2.28)$$

In the rest of this dissertation, we use this CPDB to indicate the variance of a clutter distribution, i.e., $E|n|^2$. When the variance is larger, CPDB is lower.

2.6.1.4 Simulation details

We provide the details of our simulations in the following:

- First, we build up the pre-trained dictionary \mathbf{D} . We use KDE to generate the pdfs in the pre-trained dictionary \mathbf{D} , based on pre-collected sets of random samples. For each pdf, the set of random samples is generated from the corresponding distribution with $\text{CPDB} \in \{-5, -7.5, -10, -12.5, -15, -17.5, -20\}$ dB. Each set contains 3500 random samples. Because three families of distributions (i.e., CG, CW, and CK) are considered, there are $Z = 7 \times 3 = 21$ pdfs in the dictionary \mathbf{D} . Specifically, for the CW and CK distributions, which have multiple parameters, we fix $k_W = 4$, $\sigma_K = 1$, and $k_K = 2$. The rest of the parameters (σ_G , λ_K , and θ_K) are accordingly computed with different CPDB values based on (2.23), (2.25), (2.27), and (2.28).
- Second, we simulate the nonstationarity of the clutter distribution. We have an initial clutter distribution p_A with $\text{CPDB} = -12.5$ dB. Note that p_A can be fixed to be any of the CG, CW, or CK distributions. Then the change-point k_0 is randomly selected from $[50, 70]$, and after step k_0 , the clutter distribution starts to follow a different distribution p_B that is randomly selected from a sub-dictionary of \mathbf{D} . In order to examine the performance of the proposed method with respect to the CPDB after the change-point, we intentionally divide the dictionary \mathbf{D} into three sub-dictionaries \mathbf{D}_S , \mathbf{D}_E , and \mathbf{D}_L , which respectively collect pdfs with smaller, equal, or larger CPDB values in \mathbf{D} . For example, \mathbf{D}_S collects the pdfs with $\text{CPDB} = -5, -7.5, \text{ and } -10$ dB. It

is worth noting that \mathbf{D}_E does not contain the distribution p_A because p_B should not be equal to p_A . For each p_A and each sub-dictionary, we randomly select 50 different k_0 , and for each k_0 , we randomly select the distribution p_B in the sub-dictionary 30 times. Thus, we have 1500 Monte-Carlo runs for each p_A and each sub-dictionary.

- Third, we consider the details of the radar measurements. For each time step/CPI k , the radar transmits $N = 50$ pulses to the target. It collects the secondary data in $L = 7$ cells besides the data in CUT. Note that because we use the clutter model in Section 2.6.1.1, we actually have 350 i.i.d. samples from the underlying clutter distribution. If we use a multivariate clutter distribution model, we may need more secondary data. In the simulation, the extended CUSUM algorithm in Section 2.3 is applied to detect the change-point. We select the value of user-defined parameter $M = 10$, and the number of calibration data $J = 2000$. For the clutter distribution identification, the real amplitudes of the 350 i.i.d. samples are used to choose the most similar distribution from the dictionary \mathbf{D} . Once the change-point is detected and the clutter distribution is identified, the new detector is used and its threshold b is determined by a predefined false alarm rate P_{fa} .

2.6.2 Comparison Cases

In the numerical experiments, we will compare the proposed adaptive method, which is hereinafter called “adaptive case”, with the following cases:

- Nonadaptive case, i.e., the radar system always uses the same likelihood ratio detector and the same threshold designed for the clutter distribution before the change-point according to a preset false alarm rate;

- Clairvoyant case, i.e., the radar system has perfect knowledge of the clutter distribution and the change-point, and uses the optimal detector;
- Adaptive matched filter (AMF) case, i.e., the radar system applies a cell-averaging technique, which is widely used in constant false alarm rate (CFAR) detection techniques and STAP, to estimate the covariance/variance of the assumed Gaussian clutter, using the secondary data. Then the covariance/variance estimator is substituted into the AMF as follows:

$$t_{\text{AMF}} = \frac{|\mathbf{y}_0^{(k)H} \mathbf{R}^{-1} \mathbf{1}|^2}{\mathbf{1}^H \mathbf{R}^{-1} \mathbf{1}} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\geq}} b. \quad (2.29)$$

In this simulation, $\mathbf{R} = \sigma^2 \mathbf{I}$, where each entry of $\mathbf{y}_0^{(k)}$ follows $\mathcal{CN}(0, \sigma^2)$. The estimator of the variance is given by $\hat{\sigma}^2 = \frac{1}{LN} \sum_{l=1}^L \sum_{i=1}^N |y_{cli}^{(k)}|^2$. Note that $y_{cli}^{(k)}$ is the i_{th} entry of the secondary data $\mathbf{y}_{cl}^{(k)}$ in (2.2). From the above, (2.29) becomes

$$\hat{t}_{\text{AMF}} = \frac{|\mathbf{y}_0^{(k)H} \mathbf{1}|^2}{\hat{\sigma}^2 \mathbf{1}^H \mathbf{1}} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\geq}} b, \quad (2.30)$$

where the left-hand-side follows an F -distribution under \mathcal{H}_0 , by the Gaussian clutter assumption. More specifically, $\hat{t}_{\text{AMF}} | \mathcal{H}_0 \sim F(2, 2LN)$, which demonstrates the CFAR property. A simple proof is shown in Appendix A. We can use this F -distribution to determine the threshold b of the AMF if we want to reach a preset false alarm rate.

2.7 Simulation Results

In this section, we firstly show the performance of the extended CUSUM algorithm and clutter identification method by parts in Section 2.7.1 and Section 2.7.2, respectively. Then

we focus on one change point, and investigate the performance of our proposed adaptive detection framework in Section 2.7.3, by using plots and statistical analysis. Finally, we emulate the practical scenarios with multiple change-points, and investigate the performance along the time axis in Section 2.7.4.

2.7.1 Performance of Change-point Detection Method

In this section, we show the performance of the change-point detection algorithm in Figs. 2.5, 2.6, and 2.7. Note that τ is the stopping time, and $k_0 + 1$ is the time at which the change arises. The difference between these two time instants denotes the delay of the change-point detection. Further, the y -axis $P(\tau - k_0 - 1 \leq t)$ denotes the numerically computed probability that the change-point is found by the extended CUSUM algorithm before a given time delay t . For example, in Figs. 2.5, the initial clutter distribution before the change-point (i.e., p_A) is a Gaussian distribution with CPDB = -12.5 dB, and the clutter distribution changes to another distribution after the change-point with CPDB $\in \{-5, -7.5, -10, -12.5, -15, -17.5, -20\}$ dB.

We see that if the CPDB values of the distributions before and after the change-point remain the same, the extended CUSUM algorithm has the worst performance of all the cases, because the extended CUSUM algorithm applies an averaging pre-processing and introduces a Gaussian assumption. Thus, after the pre-processing, the outputs of the clutter data before and after the change-point may behave like those from the Gaussian distributions with close power, making the extended CUSUM does not easily detect the change-point. Hence, more time steps/CPIs are needed to trigger the change-detection. However, even in this worst case, the probability of change-point detection still reaches almost 100% within

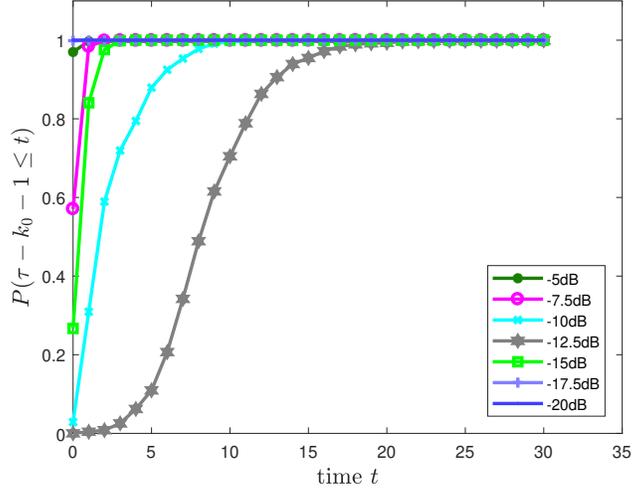


Figure 2.5: Performance of the change-point detection with initial clutter distribution as Gaussian at CPDB = -12.5dB .

20 steps. For other cases, the probabilities of the change-point detection reach 100% within fewer steps. We observe that if the difference of the CPDB values is larger, the delay of change-point detection is smaller. Fig. 2.6 and Fig. 2.7 show the cases where the initial clutter distributions are Weibull distributions and K distributions, respectively. As we see, the behaviors of these two figures are similar to that in Fig. 2.5.

2.7.2 Performance of Clutter Distribution Identification Method

We show that accuracy of the clutter distribution identification in Table 2.1. In the table, CPDB_{AC} denotes the CPDB values for the clutter distributions after the change-points. The second row I-CG Accuracy shows the clutter distribution identification accuracy when the clutter distribution before the change-point is complex Gaussian. The third row and fourth row are corresponding to the performances of complex Weibull and complex K distributions, respectively. We observe that the accuracy of the clutter distribution identification is high,

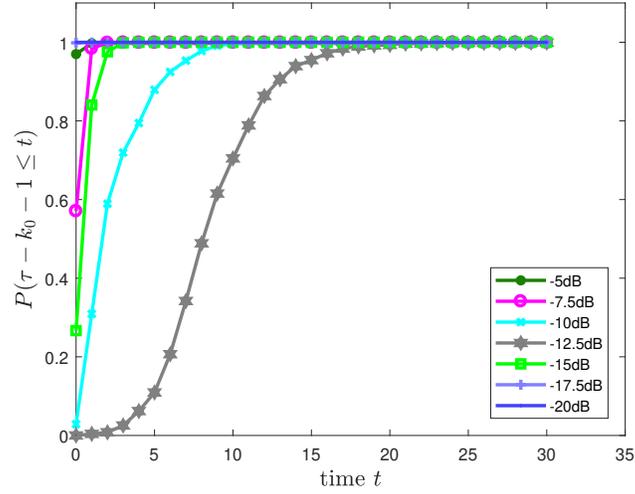


Figure 2.6: Performance of the change-point detection with initial clutter distribution as Weibull at CPDB = -12.5dB.

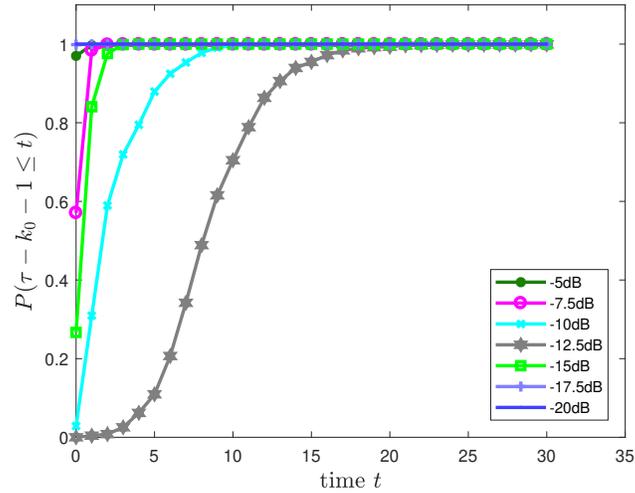


Figure 2.7: Performance of the change-point detection with initial clutter distribution as K at CPDB = -12.5dB.

Table 2.1: Clutter distribution identification accuracy with different initial clutter distributions

CPDB _{AC} (dB)	-5	-7.5	-10	-12.5	-15	-17.5	-20
I-CG Accuracy	98.9%	99.0%	98.7%	99.5%	99.0%	99.1%	99.1%
I-CW Accuracy	99.3%	98.8%	99.2%	98.9%	98.1%	99.0%	99.3%
I-CK Accuracy	99.3%	99.2%	98.5%	99.7%	99.1%	98.6%	98.7%

achieving 99% for most cases. Hence, the numerical results demonstrate the capability of the method in Section 2.4, and the high accuracy will contribute to the success of the adaptive detection framework.

To further demonstrate the merits of our method, we compare the proposed clutter distribution identification method with the classical Ozturk method. To keep the flow of this dissertation, we present the simulation settings and results in Appendix B. We show there that, in comparison to the Ozturk method, our method provides improved accuracy in identifying clutter distributions that have different parameters, but are from the same family.

2.7.3 Performance of the Unified Adaptive Detection Framework: One Change-point

In this section, focusing on one change-point, we show the performance of the proposed adaptive detection framework, and compare it with those of the nonadaptive case, AMF case, and clairvoyant case, which are defined in Section 2.6.2

2.7.3.1 Target Detection Performance Comparison

The target detection performance and the case comparisons are shown in two sets of figures, i.e., {Fig. 2.8, Fig. 2.9, Fig. 2.10} and {Fig. 2.11, Fig. 2.12, Fig. 2.13}. The first set of figures correspond to the scenario where the clutter distribution before the change-point is K with CPDB=-12.5dB, and the radar system have a preset false alarm rate $P_{fa} = 0.05$. The second set of figures correspond to the scenario where the clutter distribution before the change-point is Weibull with CPDB=-12.5dB, and the radar system have a preset false alarm rate $P_{fa} = 0.01$. Note that in each figure we show the detection performance (P_{fa} or P_d) of the clairvoyant case and one specific case indicated by the title of the figure, i.e., one of adaptive case, nonadaptive case, and AMF case. The detection performance is computed using the data after the extended CUSUM algorithm claims a change-point in each Monte-Carlo run. In each figure, there are two plot symbols. The “line and circle” symbol denotes a clairvoyant case, while the “box and two line segments” is a boxplot symbol that denotes a specific case. For the boxplot symbol, the box contains 25%-75% data, and the line in the middle denotes the median of the data. The two outermost line segments denote the maximum and the minimum of the data except the outliers. Green color indicates the preset false alarm rate (clairvoyant false alarm rate) P_{fa} that the radar system hopes to maintain. Blue denotes a case in which the clutter distribution is complex Gaussian (CG) after the change-point. Similarly, magenta corresponds to the complex K (CK) distribution, and cyan corresponds the complex Weibull (CW) distribution.

As we see, the adaptive cases and the AMF cases usually achieve the preset P_{fa} , while the nonadaptive cases usually have significant biases, by observing the first plot in each of Fig. 2.8-2.13. Further, the behaviors of P_d for the nonadaptive cases are not easy to predict. The P_d of adaptive cases can achieve that of the clairvoyant cases, as they are

almost overlapped in the figures, while the P_d of the AMF cases are usually lower than the P_d of clairvoyant cases, by observing the second plot in each of Fig. 2.8-2.13.

2.7.3.2 Statistical Significance Analysis

To quantitatively show the above conclusion, we use the clairvoyant cases as the ground truth, and compare the absolute deviations of P_{fa} or P_d computed in the adaptive cases, nonadaptive cases, and AMF cases after the detected change-points. We use the one-sided Wilcoxon rank-sum test [40,41] to show whether one group of absolute deviations are statistically less than another group. The Wilcoxon rank-sum test works as a non-parametric alternative to the well-known t -test to compare the medians of two groups of data. In our case, we have the following hypothesis testing problem:

$$\begin{cases} \mathcal{H}_0 : \text{Median}_A = \text{Median}_B \\ \mathcal{H}_1 : \text{Median}_A < \text{Median}_B \end{cases}, \quad (2.31)$$

where A and B are two groups of absolute deviations that we want to compare. We use A to denote the adaptive case, and B to denote one of nonadaptive case and AMF case. We expect that the data can support rejecting the null hypothesis, i.e., the p -value of the test is smaller than a significance level, if the proposed adaptive method is better than the other method in comparison.

By using the Wilcoxon rank-sum test, the results of Figs. 2.8, 2.9, and 2.10 are summarized into Table 2.2, while the results of Figs. 2.11, 2.12, and 2.13 are summarized into Table 2.3. These two tables show the p -values of the tests. Note that “CPDB_{AC}” denotes the CPDB values after the change-points, while “Overall” means that the absolute deviations of all the

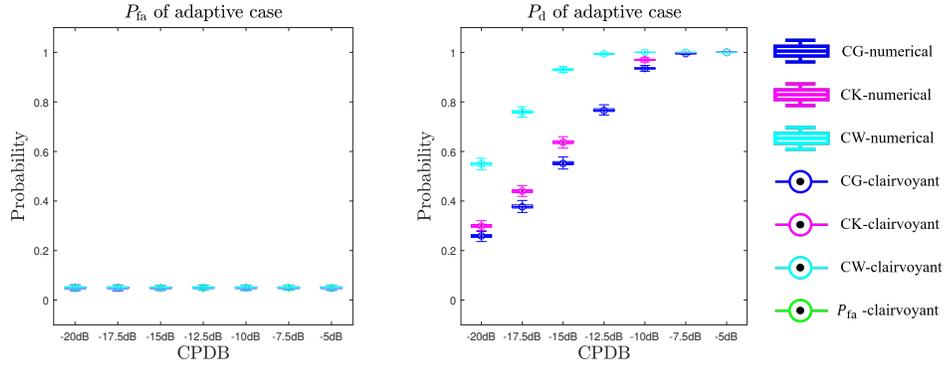


Figure 2.8: Performance of the adaptive detection framework. The initial clutter distribution is K with CPDB = -12.5dB. The preset $P_{fa} = 0.05$.

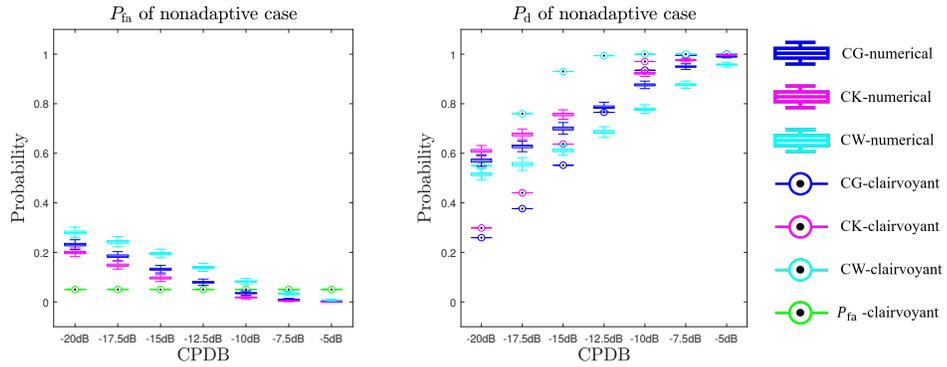


Figure 2.9: Performance of the nonadaptive detection case. The initial clutter distribution is K with CPDB = -12.5dB. The preset $P_{fa} = 0.05$.

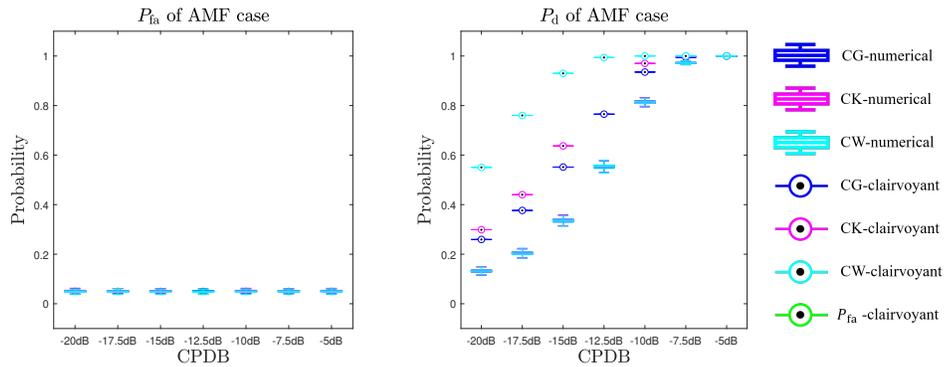


Figure 2.10: Performance of the AMF case. The initial clutter distribution is K with CPDB = -12.5dB. The preset $P_{fa} = 0.05$.

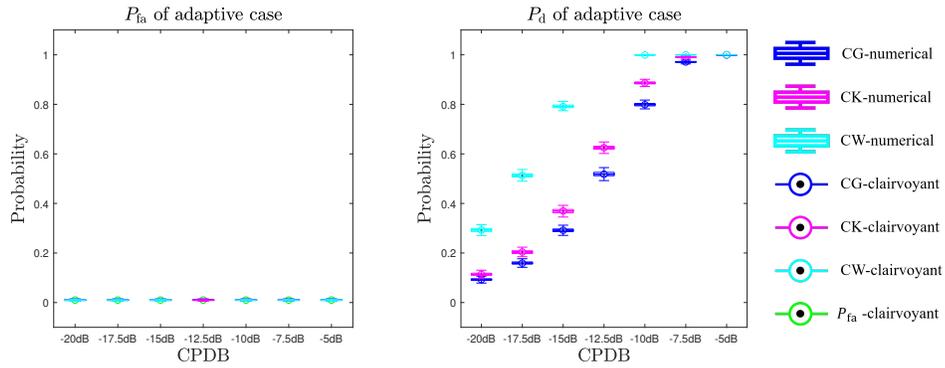


Figure 2.11: Performance of the adaptive detection framework. The initial clutter distribution is Weibull with CPDB = -12.5dB. The preset $P_{fa} = 0.01$.

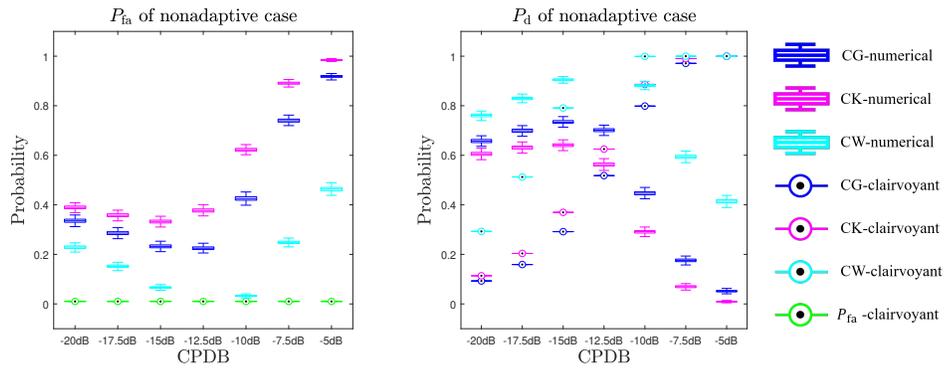


Figure 2.12: Performance of the nonadaptive detection case. The initial clutter distribution is Weibull with CPDB = -12.5dB. The preset $P_{fa} = 0.01$.

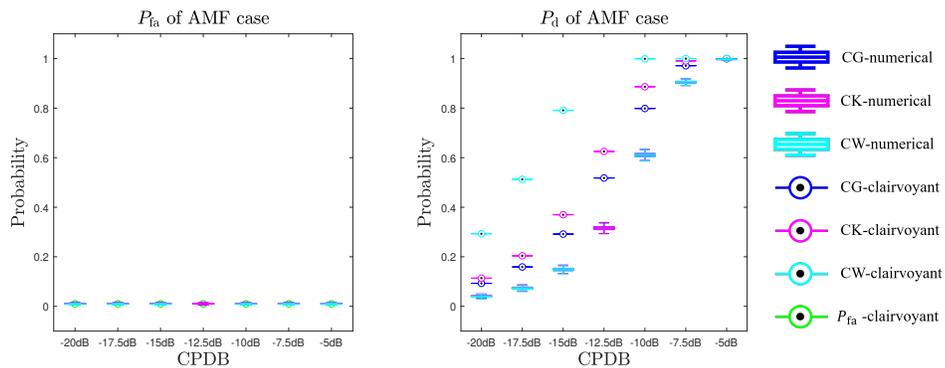


Figure 2.13: Performance of the AMF case. The initial clutter distribution is Weibull with CPDB = -12.5dB. The preset $P_{fa} = 0.01$.

CPDB cases are grouped to compare. In addition, other cells in the first row of each table indicate the groups of absolute deviations we compare in (2.31). For example, “ P_{fa} : AD v.s. NAD” means that absolute deviations of P_{fa} are compared, and in the test (2.31), group A corresponds to the adaptive case, while group B corresponds to the nonadaptive case.

In our tests, we set the significance level to 0.05. If the p -value is lower than 0.05, then we reject the null hypothesis, and we claim \mathcal{H}_1 , i.e., the absolute deviations of the proposed adaptive case are statistically less than the other case we compare. Otherwise, if the p -value is larger than 0.05, we claim that we fail to reject \mathcal{H}_0 , i.e., it is very likely that the performance of the proposed adaptive case is comparable to that of the other case we compare. From the tables, we see that:

- For false alarm rate P_{fa} , the adaptive cases and AMF cases have comparable performance, as the p -values are greater than 0.05. This observation is consistent with what is shown in Figs. 2.8, 2.10, 2.11, and 2.13, i.e., the adaptive case and the AMF case can achieve the false alarm rate of the clairvoyant case. On the contrary, by comparing the adaptive case and the nonadaptive case, the p -values are less than 0.0001, which means that \mathcal{H}_0 in (2.31) is rejected, which is consistent with that the performance of the nonadaptive cases is unpredictable, and usually has large deviations from that of the clairvoyant cases.
- For probability of detection P_{d} , the p -values for both “AD v.s. NAD” and “AD v.s. AMF” are small enough to reject \mathcal{H}_0 , and we can claim that the absolute deviations of the proposed adaptive case are statistically smaller than the others’. It is consistent with that, in Figs. 2.8-2.13, compared to two other cases, the P_{d} for the proposed adaptive case is closer to the P_{d} for the clairvoyant case.

Table 2.2: P -values of Wilcoxon rank-sum one-sided tests for false alarm rate and probability of detection after detected change-points with initial K-distributed clutter (CPDB = -12.5dB) and preset $P_{\text{fa}} = 0.05$

CPDB _{AC}	P_{fa} : AD v.s. NAD	P_{fa} : AD v.s. AMF	P_{d} : AD v.s. NAD	P_{d} : AD v.s. AMF
-20dB	<0.0001	0.0826	<0.0001	<0.0001
-17.5dB	<0.0001	0.4493	<0.0001	<0.0001
-15dB	<0.0001	0.7222	<0.0001	<0.0001
-12.5dB	<0.0001	0.4144	<0.0001	<0.0001
-10dB	<0.0001	0.4167	<0.0001	<0.0001
-7.5dB	<0.0001	0.3939	<0.0001	<0.0001
-5dB	<0.0001	0.1743	<0.0001	<0.0001
Overall	<0.0001	0.1654	<0.0001	<0.0001

Table 2.3: P -values of Wilcoxon rank-sum one-sided tests for false alarm rate and probability of detection after detected change-points with initial Weibull-distributed clutter (CPDB = -12.5dB) and preset $P_{\text{fa}} = 0.01$

CPDB _{AC}	P_{fa} : AD v.s. NAD	P_{fa} : AD v.s. AMF	P_{d} : AD v.s. NAD	P_{d} : AD v.s. AMF
-20dB	<0.0001	0.4273	<0.0001	<0.0001
-17.5dB	<0.0001	0.4656	<0.0001	<0.0001
-15dB	<0.0001	0.4697	<0.0001	<0.0001
-12.5dB	<0.0001	0.3556	<0.0001	<0.0001
-10dB	<0.0001	0.4656	<0.0001	<0.0001
-7.5dB	<0.0001	0.4733	<0.0001	<0.0001
-5dB	<0.0001	0.6556	<0.0001	<0.0001
Overall	<0.0001	0.4193	<0.0001	<0.0001

2.7.4 Performance of the Unified Adaptive Detection Framework: Multiple Change-points

In the previous sections, we focused on the effects of one change-point. In this section, we highlight the adaptive property of the proposed framework by investigating the detection performance with multiple change-points along the time axis .

2.7.4.1 Simulation Settings

For illustration, we fix three change-points, at 60, 120, and 180th CPIs, in all the following experiments. We also fix four clutter-distributions during four time intervals $[1, 60)$, $[60, 120)$, $[120, 180)$, and $[180, 240]$ when repeating Monte Carlo runs. We emphasize that although the change-points and the clutter distributions are fixed, they are unknown to the radar systems, except for the initial clutter distribution during $[1, 60)$, which is pre-learned. The dictionary setup and the extended CUSUM algorithm setup remain the same as in Section 2.6.1. For the detection, the detector thresholds are chosen to meet the a predefined false alarm rate P_{fa} .

2.7.4.2 Target Detection Performance Comparison

With a pre-defined false alarm $P_{fa} = 0.05$, we show the numerical results based on 3000 Monte Carlo runs in Figs. 2.14, 2.15, and 2.16. Then underlying clutter distributions for four intervals are indicated by the captions of the figures. The x -axis is the time axis, which is divided into CPIs. The left y -axes are either the probability of false alarm or the probability of detection, as indicated by the title of each subplot. In order to simultaneously show performance of the extended CUSUM algorithm, we use the right y -axis and the blue curve to show the behavior of change-point detection delay.

From Fig. 2.14-(a), we observe that AMF case maintains the predefined false alarm rate, i.e., P_{fa} in clairvoyant case. The nonadaptive case fails to achieve this false alarm rate during $[60, 120)$ CPIs, because it does not adaptively choose its detector and threshold to handle the change-points. The adaptive case almost maintains the predefined false alarm rate, except for some CPIs after the change points. For example, “spikes” are shown around the 60th and

120th CPIs. After several CPIs, the adaptive case returns to the predefined false alarm rate. Each of the blue curves shows the probability that the extended CUSUM algorithm finds a change-point within a t -CPI delay. We refer this probability as the “delay probability”, and here we plot only $t \in [0, 31]$. As we can observe, when this probability approaches 1, the adaptive case approaches the predefined false alarm rate.

From Fig. 2.14-(b), we observe that the AMF case now cannot reach the clairvoyant probability of detection. The nonadaptive case can reach the same performance as that of the clairvoyant case in the $[0, 60)$ and $[180, 240]$ CPI intervals, because the nonadaptive case happens to use the same detector and threshold as the clairvoyant case. However, there is a performance gap between these two cases over $[60, 180)$. For the adaptive case, the probability of detection usually does not follow that of the clairvoyant case after a change-point at the first several steps because it does not use the correct detector and threshold. However, as the extended CUSUM algorithm finds the change-point, which is illustrated by the increasing delay probability, the probability of detection for the adaptive case returns to that of the clairvoyant case.

We show two other experimental results with $P_{fa} = 0.05$ in Fig. 2.15 and Fig. 2.16, respectively. The overall behaviors among different cases are similar. It is worth noting that in Fig. 2.15-(b), the nonadaptive case produces even better P_d value than that of the clairvoyant case over $[60, 120)$ CPIs, because the wrong use of the detector and threshold produces larger/worse P_{fa} value. In Fig. 2.16, because we use clutter distributions with the same power, the false alarm rate of the nonadaptive case achieves the predefined P_{fa} . However, the P_d value of the nonadaptive case is worse than that of the clairvoyant case. For the adaptive case, the delay effects are also shown, but after several CPIs, it reaches the desired performance.

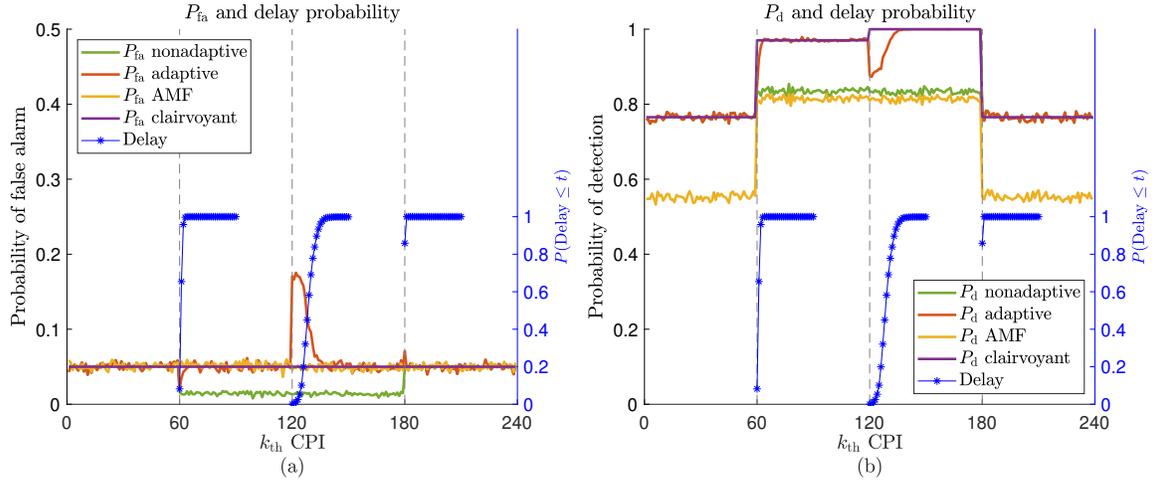


Figure 2.14: Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-10dB)→CW(-10dB)→CG(-12.5dB) with change-points at the 60, 120, and 180_{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.05$.

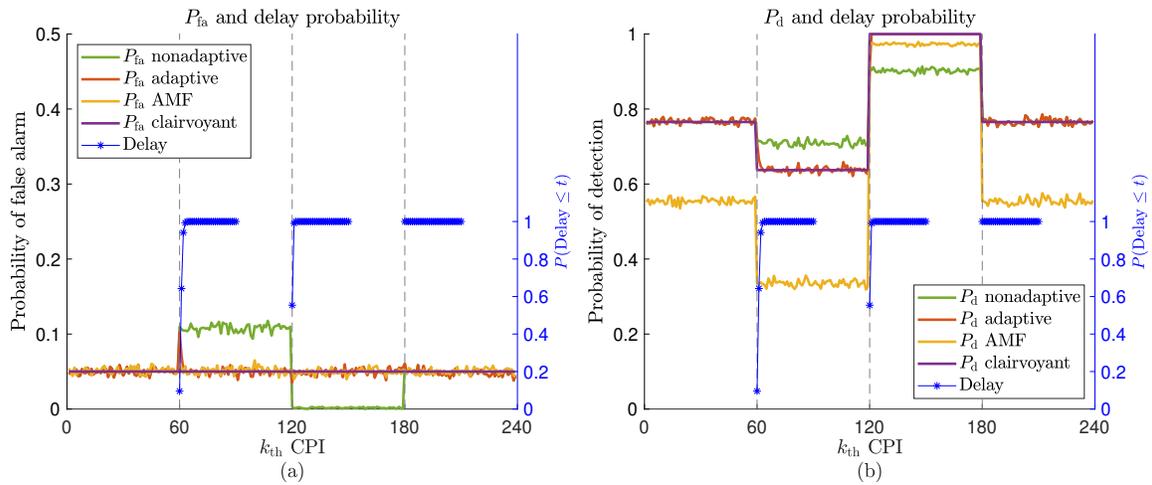


Figure 2.15: Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-15dB)→CW(-7.5dB)→CG(-12.5dB) with change-points at the 60, 120, and 180_{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.05$.

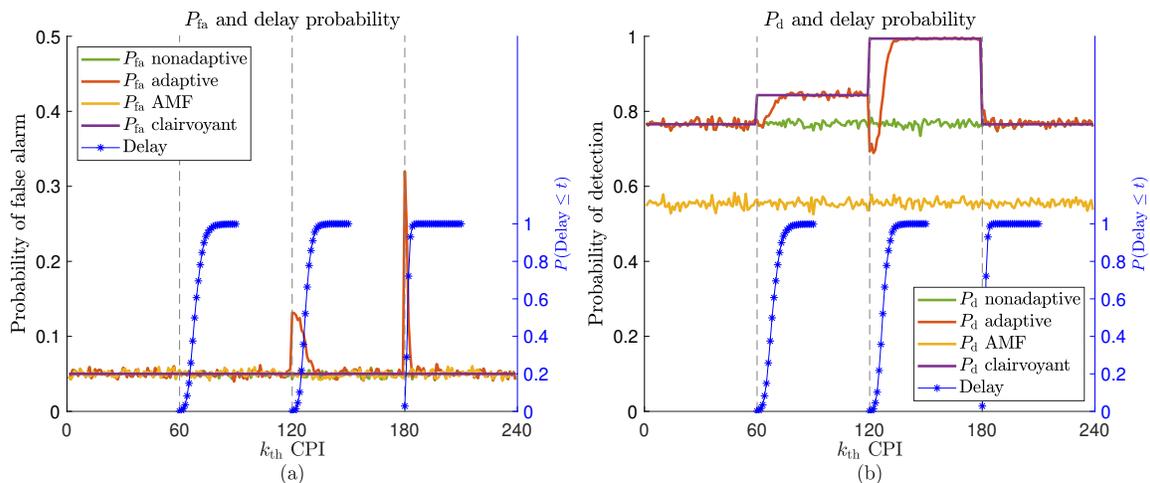


Figure 2.16: Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-12.5dB)→CW(-12.5dB)→CG(-12.5dB) with change-points at the 60, 120, and 180_{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.05$.

Two other sets of results with pre-defined false alarm rates $P_{fa} = 0.005$ and $P_{fa} = 5 \times 10^{-4}$ are given in {Figs. 2.17, 2.18, 2.19} and {Figs. 2.20, 2.21, 2.22}, respectively. Note that {Figs. 2.17, 2.18, 2.19} are based on 3000 Monte Carlo runs, while {Figs. 2.20, 2.21, 2.22} are based on 50000 Monte Carlo runs due to its smaller false alarm rate. Similarly, we can see that in each figure, the adaptive case can maintain the predefined false alarm rate and its probability of detection can follow that of the clairvoyant case, except for some CPIs after the change points. For these CPIs, the adaptive case does not use the optimal detector, thus it suffers from performance degradation. It rapidly reaches the desired performance once the change-points are detected. The behaviors of the nonadaptive case and the AMF case are also similar to those in Figs. 2.14, 2.15, and 2.16. In each figure, since the nonadaptive case always uses a detector designed for the scenario before the first change point, it cannot achieve the predefined false alarm rate when its detector is not the same as the optimal detector. This difference also causes a performance gap for P_d between the nonadaptive

case and the clairvoyant case. The AMF case can achieve the predefined false alarm rate, however, it cannot reach the clairvoyant probability of detection.

2.8 Chapter Summary

We proposed a framework for target detection in nonstationary environments, utilizing the extended CUSUM algorithm to detect the possible change-points of the clutter distributions. Once a change-point was found, we used a sparse-theory-inspired clutter distribution identification method to learn the new clutter distributions, and we adaptively adjusted the detectors based on the newly learned clutter distributions. We numerically demonstrated the performance of the change-point detection algorithm and the clutter distribution identification technique by parts, and further showed the performance of the framework, compared to the clairvoyant case, the nonadaptive detection case, and the AMF case. We showed that once the extended CUSUM algorithm found the change-point, our method could adaptively use the optimal detector and achieve the performance of the clairvoyant case.

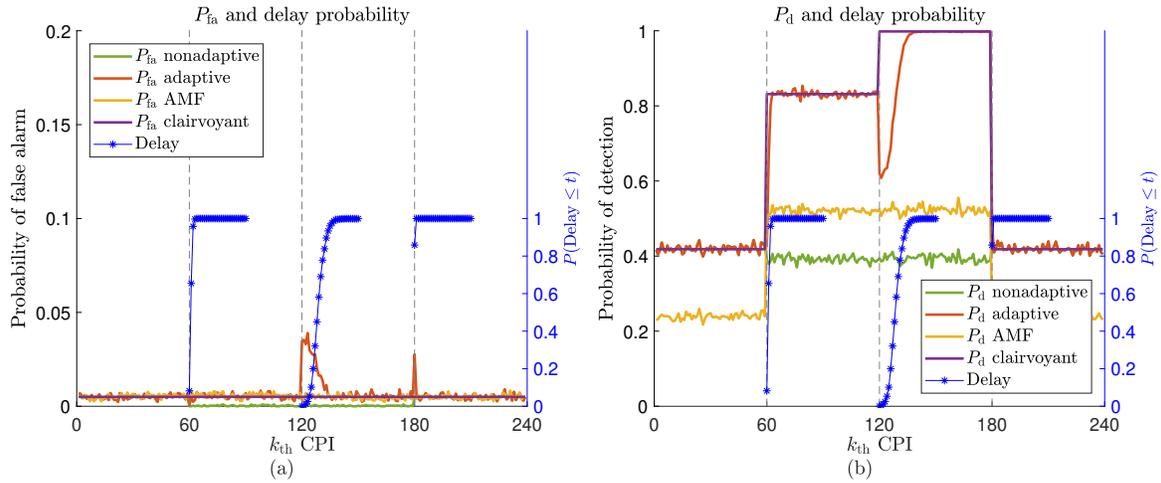


Figure 2.17: Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-10dB)→CW(-10dB)→CG(-12.5dB) with change-points at the 60, 120, and 180_{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.005$.

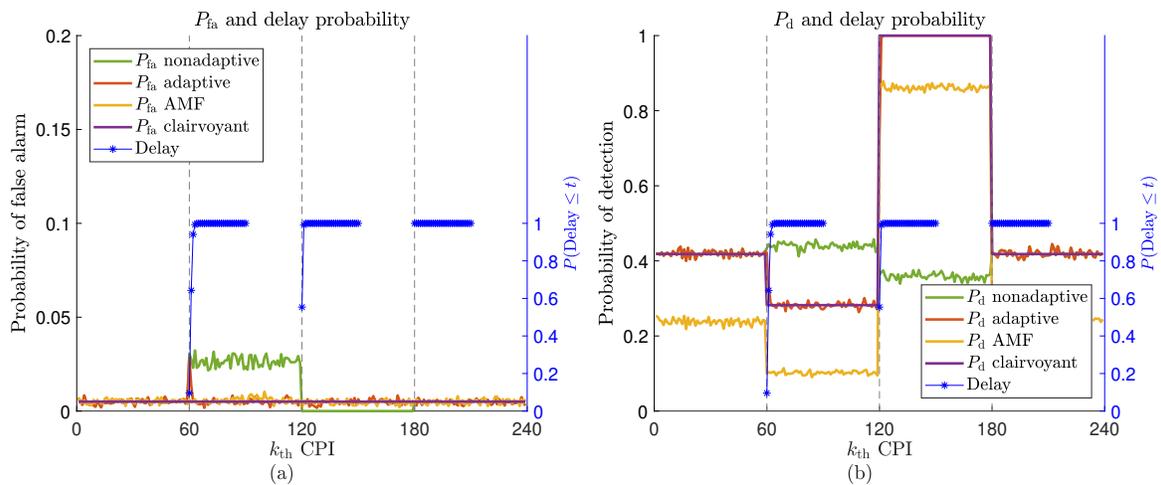


Figure 2.18: Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-15dB)→CW(-7.5dB)→CG(-12.5dB) with change-points at the 60, 120, and 180_{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.005$.

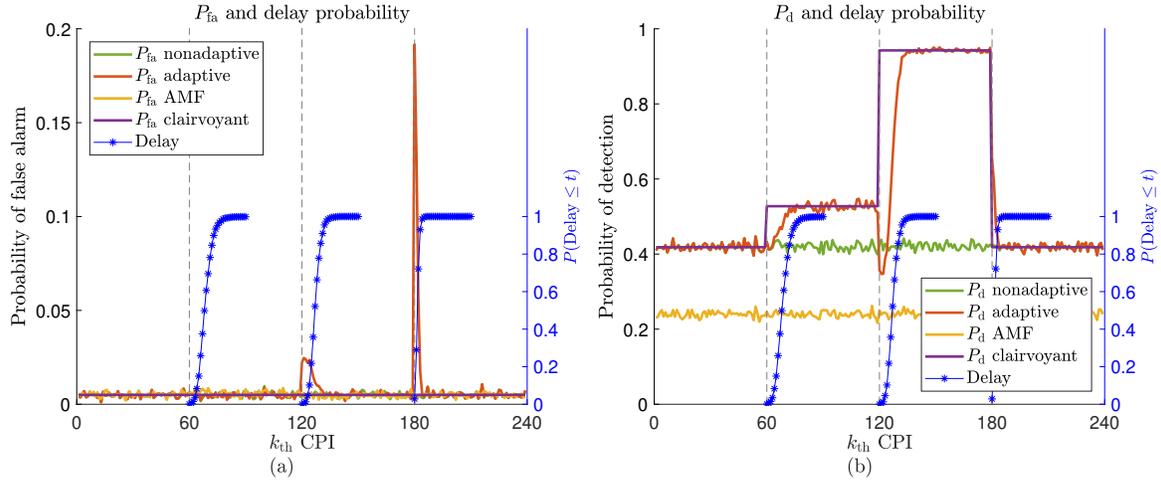


Figure 2.19: Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-12.5dB)→CK(-12.5dB)→CW(-12.5dB)→CG(-12.5dB) with change-points at the 60, 120, and 180_{th} CPIs. The predefined false alarm rate is $P_{fa} = 0.005$.

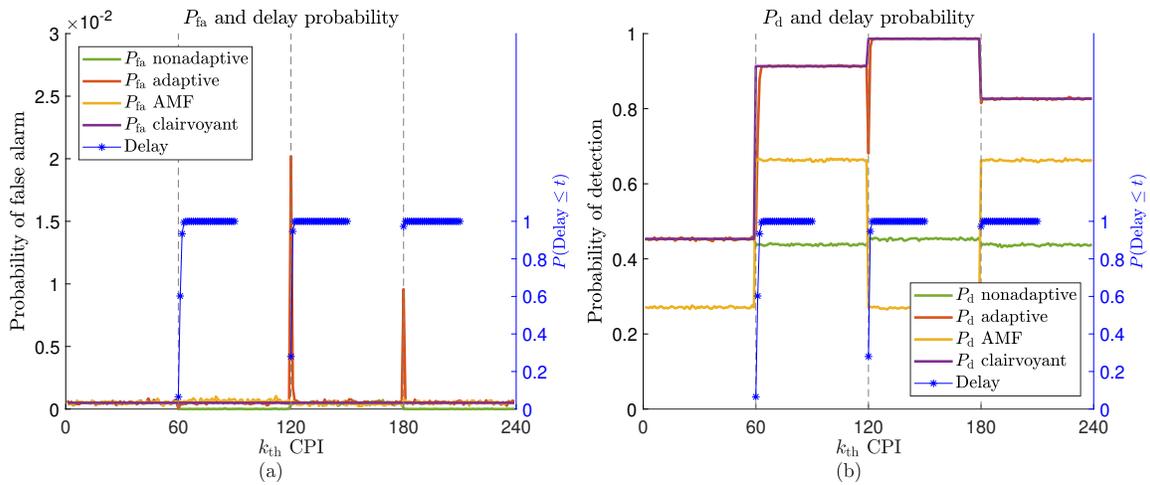


Figure 2.20: Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-10dB)→CK(-7.5dB)→CW(-10dB)→CG(-7.5dB) with change-points at the 60, 120, and 180_{th} CPIs. The predefined false alarm rate is $P_{fa} = 5 \times 10^{-4}$.

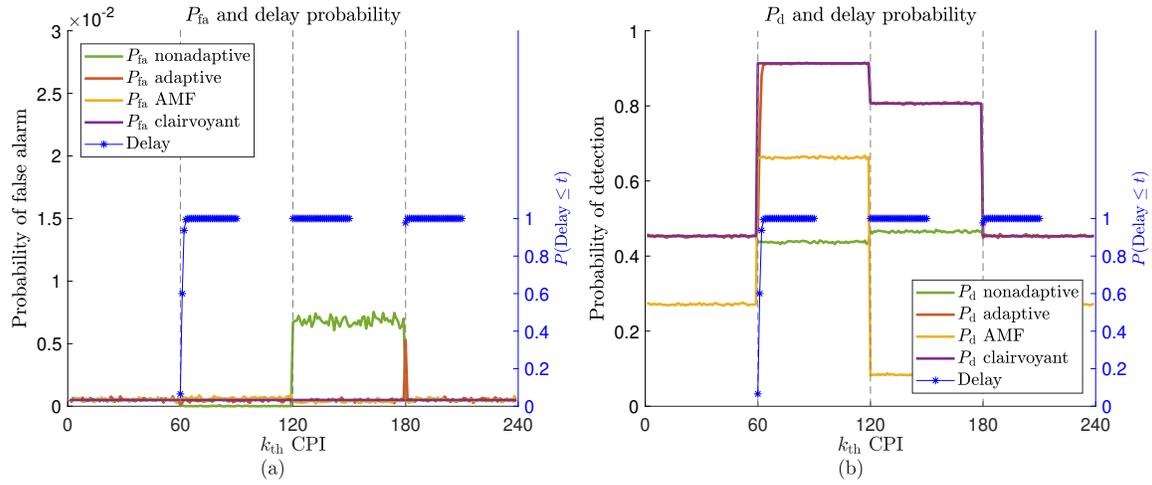


Figure 2.21: Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-10dB)→CK(-7.5dB)→CW(-12.5dB)→CG(-10dB) with change-points at the 60, 120, and 180_{th} CPIs. The predefined false alarm rate is $P_{fa} = 5 \times 10^{-4}$.

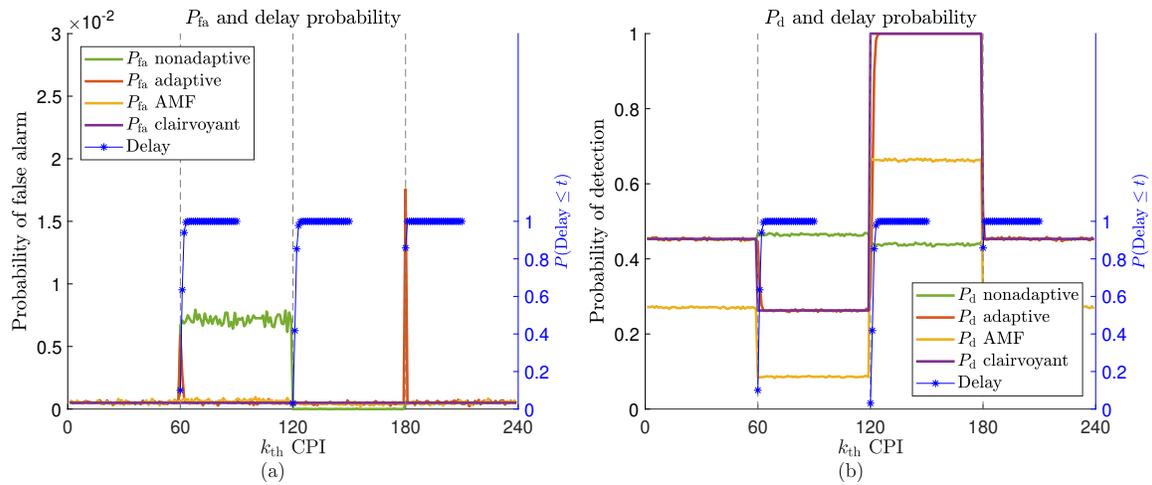


Figure 2.22: Performance of the adaptive detection framework, in comparison with the nonadaptive case, AMF case, and clairvoyant case. The clutter distributions are CG(-10dB)→CK(-12.5dB)→CW(-7.5dB)→CG(-10dB) with change-points at the 60, 120, and 180_{th} CPIs. The predefined false alarm rate is $P_{fa} = 5 \times 10^{-4}$.

Chapter 3

Target Tracking in Cognitive Radar Networks

3.1 Introduction

In this chapter, we will introduce our proposed framework for Target Tracking in Cognitive Radar Networks.²

We begin with the dynamic graphical model that describes the general idea of the proposed framework. Then, we introduce signal models for the transmitted and received signals. We also present a target kinematic model, a measurement model, and a recursive Bayesian estimation technique for target tracking. Next, we formulate the cognitive target tracking problem as two iterative steps: (i) solving a combinatorial optimization problem to select the optimal subset of radars, waveforms, and locations for the next tracking instant, and (ii) acquiring the recursive Bayesian state estimation to accurately track the target. We further

²This chapter is based on Y. Xiang, M. Akcakaya, S. Sen, D. Erdogmus, and A. Nehorai, “Target tracking via recursive Bayesian state estimation in cognitive radar networks,” *Signal Processing*, Vol. 155, pp. 157-169, Feb. 2019. DOI:10.1016/j.sigpro.2018.09.035. © Elsevier 2019.

discuss scenarios where multiple target models are considered in the target tracking. To demonstrate the proposed framework, we use an illustrative example to introduce a specific scenario in 2-D space. The results based on this scenario show that the proposed framework can accurately track the target under the management of a network of radars.

3.1.1 Related Work

A few works in the literature adopt one or combine two of the above three techniques in target tracking; however, no other work thus far has simultaneously applied waveform-agile sensing, planned the corresponding paths, and intelligently selected the subset of radars. For example, dynamic waveform adaption has been extensively investigated in target detection [42–48] and tracking [49–62] to meet a radar’s constantly changing requirements for target information. The motivation behind the adaptive waveform design techniques is to gain better tracking performance by integrating the radar transmitter and receiver in a closed-loop fashion. Previous works focus on analyzing the output of a matched filter and on improving the resolution [63] by treating the radar and tracker as independent subsystems [64]. In recent works, as the tracker recursively estimates the target state, the radar transmitter is adapted to design the next transmitted waveform according to a predefined utility function, such as the mean-square tracking error [53, 54], the trace of the posterior Cramér-Rao lower bound (PCRLB) [49, 65], and the mutual information [50, 66, 67]. In general, the utility function provides one-step ahead design (i.e., greedy); only a few articles discuss multi-step ahead waveform selection [53, 68]. Some works directly design the frequency spectrum of the waveform by using optimal frequencies [42, 43, 69], while others select the waveform from a parameterized waveform library [67], such as a linear frequency-modulated (LFM) library [49]. Additionally, waveform design for multistatic target tracking is also

investigated [70]. Although waveform design and scheduling applications show promising results, they face challenges in selecting proper design metrics, designing optimal or nearly-optimal waveform families with respect to these criteria, and formulating computationally tractable scheduling policies [71].

Path planning is another important and fundamental issue for radars, particularly for those which are installed on moving platforms, such as UAVs. To improve the performance of target localization and tracking, careful design of radar paths to actively steer them to the optimal places within the kinematic capabilities [72–78] provides another degree of freedom in utilizing the feedback information from receivers. A simple case of obtaining the optimal trajectory of a single radar by maximizing the determinant of the Fisher information matrix in a passive bearings-only case for fixed target localization is introduced in [74]. Multiple radars cases are considered in [72, 73, 75–77], and cooperative path planning strategies are derived for both passive and active radars. However, in general, it is extremely complicated to plan the paths of multiple radars simultaneously under certain kinematic constraints while appropriately adopting non-myopic policies.

Sensor selection and resource allocation schemes are generally employed in sensor network management problems to intelligently assign a subset of sensors to accomplish the task with satisfactory performance or minimized usage of resources. A sensor selection problem is usually a combinatorial optimization problem, which can be NP hard to solve for the optimal solution, and therefore approximate techniques are necessary to search for an acceptable solution. In the literature, the sensor selection problem is usually cast in one of three ways. It can be formulated as a convex optimization problem with a heuristic searching method [79], as the maximization of a submodular function problem (which can be solved by a greedy algorithm with guaranteed performance) [80], or as a linear programming and

a semi-definite programming (SDP) problem by respectively considering the measurement noises to be uncorrelated and correlated [81]. In regard to target tracking, the sensor selection and resource allocation schemes are adaptively applied from time to time [81–88].

In addition, a few works combine two of the three techniques (waveform design, path planning, and sensor selection) to further improve the tracking performance or decrease the resource usage. For example, waveform design and path planning are jointly considered in [62] and [89]. In [62], adaptive selection of the optimal pulse repetition interval (PRI) from a set of allowed values, and optimization of the radar trajectory using the trace of weighted PCRLB, are proposed, whereas [89] jointly optimizes the waveform parameters of Gaussian-LFM pulses and the guiding commands of multistatic radars using the trace of the PCRLB. Although multiple radars are taken into account, no subset selection strategy is adopted in [89], implying that the radars use all the available resources to track the target, which may lead to a waste of resources. In [90,91], waveform design and sensor selection are combined for static radars to select adaptive waveforms and a subset of sensors in accordance with the resource constraints.

3.1.2 Our Work

In our preliminary work [92], we proposed a framework for single target tracking based on the theories of dynamic graphical models (DGM) and recursive Bayesian state estimation (RBSE), jointly considering radar selection and path planning. However, the advantages of waveform-agility were not exploited to further improve the tracking performance and system intelligence.

In the work of this dissertation, we propose a target tracking framework that simultaneously designs the transmit waveform, plans the radar trajectory, and selects the appropriate radar subset. Building on DGM and RBSE [93, 94], we choose the expected cross-entropy as the objective function and solve a combinatorial optimization problem to select the optimal subset of radars that transmit well-designed waveforms from their best locations for receiving the most informative measurements during the next tracking instant. The received measurements associated with history information are then processed by a tracker to achieve a more accurate target state estimation. Additionally, we validate the proposed framework by constructing an illustrative case, where radars with certain moving constraints track a single target in a 2-D space. Specifically, we choose this simplified case because it is a representation of a large-scale network of cognitive radars which have certain constraints on their coverage area and path. Our main contribution in this dissertation is to consider the path design, waveform design and radar subset selection simultaneously. Even though the concept of cognitive radar was proposed more than a decade ago [6], until now a real-time implementation of a cognitive radar has not been presented.

3.2 Framework of Target Tracking in Network of Radars

Our proposed framework formulates the sensor (radar) network management for single target tracking as a combinatorial optimization problem. By solving the combinatorial optimization problem, we choose the optimal subset of radars to transmit well-designed waveforms in the best locations in order to receive the most informative echos in the next tracking step. The echos are then processed by the tracker associated with history information to achieve a reasonable state estimation. Fig. 3.1 depicts our proposed framework of single target tracking

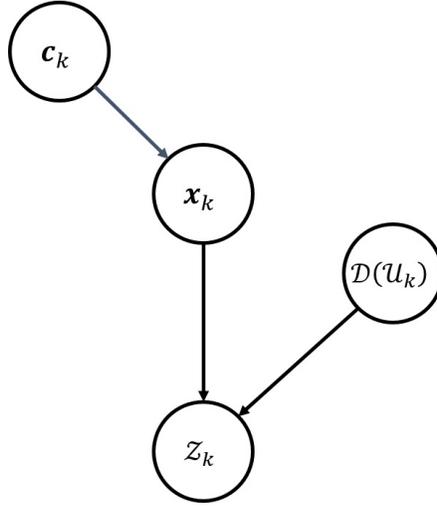


Figure 3.1: Graphical model (step k). The evolution of the state \mathbf{x}_k is dependent on the contextual evidence \mathbf{c}_k . Here $\mathcal{D}(\mathcal{U}_k)$ indicates the selected radars, the transmitted waveforms, and the radar states; \mathcal{Z}_k denotes noisy measurements of the state \mathbf{x}_k if $\mathcal{D}(\mathcal{U}_k)$ is applied.

in cognitive network of radars based on a DGM. Without loss of generality, similar to most target tracking application, order-1 Markov processes with finite-dimensional state vectors are considered. The notations of variables of interest and their dependence relationships at time k are defined as follows:

1. The state of the target is denoted as \mathbf{x}_k , which needs to be estimated from the noisy measurement and contextual information. The state of \mathbf{x}_k usually includes the target's current location and velocity. It can be augmented to contain more variables, such as the acceleration of the target and the radar cross section (RCS) of the echos.
2. The target kinematic model, the history information provided by previous measurements, and the statistical environment model are represented as probabilistic contextual evidence, \mathbf{c}_k .
3. A set (dictionary) is denoted as $\mathcal{D} = \{(\mathcal{D}^s, \mathcal{D}^w, \mathcal{D}^p)\}$, where \mathcal{D}^s contains all the possible radars that can be managed in the network, \mathcal{D}^w contains all the possible waveforms

that can be transmitted by the radars, and \mathcal{D}^p includes all the states of radars (e.g., positions and velocities) that can be achieved.

4. A set of indices is denoted as $\mathcal{U}_k = \{(\mathcal{U}_k^s, \mathcal{U}_k^w, \mathcal{U}_k^p)\}$, where \mathcal{U}_k^s denotes the set of radars that are selected to transmit the corresponding waveforms represented by the index set \mathcal{U}_k^w with their states indicated by \mathcal{U}_k^p at time step k .
5. The measurements at time k that are collected by the subset of radars with certain waveforms and states suggested by $\mathcal{D}(\mathcal{U}_k)$ are denoted as $\mathcal{Z}_k = \{z_k^{(i)} | i \in \mathcal{U}_k^s\}$, where $z_k^{(i)}$ is the measurement of i_{th} radar.

The probabilistic contextual evidence \mathbf{c}_k could include any information that might affect the estimation of the target state. It could involve several kinematic models to describe the possible motion modes of the target. A commonly used type of contextual evidence is previously collected measurements, which influence the state via estimations of previous states. Further, the contextual evidence also could be provided by a pre-trained environment model in the workspace. For instance, the sensing system has an environment model describing the distribution of obstacles, and prior knowledge that the target tends to avoid obstacles rather than strike them. In this dissertation, we use only one kinematic model together with previous measurements as the probabilistic contextual evidence \mathbf{c}_k . In order to further describe the framework in detail, we define the signal model, kinematic model, measurement model, and recursive Bayesian estimation technique in the following subsections.

3.2.1 Signal Model

The narrowband waveform of the transmitted signal of the i_{th} radar has the following form,

$$g^{(i)}(t) = s^{(i)}(t)e^{j2\pi f_c^{(i)}t}, \quad (3.1)$$

where $s^{(i)}(t)$ is the complex envelope of the signal, and $f_c^{(i)}$ is the carrier frequency. The corresponding echo from the target in the baseband is

$$r^{(i)}(t) = \alpha(f^{(i)})s^{(i)}(t - \tau^{(i)})e^{j2\pi\nu^{(i)}(t - \tau^{(i)})}e^{-j2\pi f_c^{(i)}\tau^{(i)}} + n^{(i)}(t), \quad (3.2)$$

where $\tau^{(i)}$ is the time delay, $\nu^{(i)}$ is the Doppler frequency, $n^{(i)}(t)$ is the complex Gaussian noise, $f^{(i)}$ is the frequency that i_{th} radar transmits after frequency modulation, and $\alpha(f^{(i)})$ is the complex reflectivity caused by the radar cross section (RCS) of the target and phase shift. Note that here we assume that the complex reflectivity is a function of the frequency that the radar transmits. Note that if we consider the scenario in a 2-D space, we have

$$\tau^{(i)} = \frac{2\sqrt{(x - x^{(i)})^2 + (y - y^{(i)})^2}}{c}, \quad (3.3)$$

$$\nu^{(i)} = -\frac{2f_c^{(i)}}{c} \frac{(\dot{x} - \dot{x}^{(i)})(x - x^{(i)}) + (\dot{y} - \dot{y}^{(i)})(y - y^{(i)})}{\sqrt{(x - x^{(i)})^2 + (y - y^{(i)})^2}}, \quad (3.4)$$

where c is the speed of light, (x, y) is the position of the target, $(x^{(i)}, y^{(i)})$ is the position of the i_{th} radar, (\dot{x}, \dot{y}) is the velocity of the target, and $(\dot{x}^{(i)}, \dot{y}^{(i)})$ is the velocity of the i_{th} radar.

The above formulas can be easily extended to the case of 3-D space.

Here we give a specific example of the signal models used by a network of radars. Assuming that all the radars in the network share the same carrier frequency f_c , and we let

$$s^{(i)}(t) = u(t)e^{j2\pi f_\xi^{(i)}t}, \quad (3.5)$$

where $u(t)$ can be a square pulse which is shared by all the radars, and the waveform agility is dependent on the frequency $f_\xi^{(i)}$ that is modulated by each radar separately. Note that the modulated frequency $f_\xi^{(i)}$ is chosen from a library $\{f_{\xi,1}, \dots, f_{\xi,W}\}$, which contains different discrete frequencies designed in advance. The range resolution is determined by the bandwidth of $u(t)$, thus leading to the same range resolution for each radar even though different waveforms could be transmitted. In this case, the complex reflectivity $\alpha(\cdot)$ of the echo in Eqn. 3.2 is a function of the modulated frequency $f_\xi^{(i)}$, i.e., $\alpha(f^{(i)})$ can be denoted as $\alpha(f_\xi^{(i)})$.

3.2.2 Kinematic Model

The target moving in the surveillance space follows the kinematic model,

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{v}_k, \quad (3.6)$$

where k is the time index, \mathbf{x}_k is the target state, $\mathbf{f}_k(\cdot)$ is the state transition function, and \mathbf{v}_k denotes the process noise.

A common example of the kinematic model is the noisy constant velocity (CV) model [4],

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + \mathbf{v}_k, \quad (3.7)$$

where $\mathbf{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^\top$ if a 2-D scenario is considered, (x_k, y_k) and (\dot{x}_k, \dot{y}_k) are the position and velocity of the target at time k , respectively, and the process noise \mathbf{v}_k is white Gaussian noise. In addition, the state transition matrix is

$$A_k = I_2 \otimes \bar{A}_k, \quad (3.8)$$

$$\bar{A}_k = \begin{bmatrix} 1 & T_{\text{in},k} \\ 0 & 1 \end{bmatrix}, \quad (3.9)$$

where I_2 is the 2×2 identity matrix, \otimes is the Kronecker product, and $T_{\text{in},k}$ is the time interval between two tracking steps. Furthermore, we have that

$$\mathbb{E}(\mathbf{v}_k) = \mathbf{0}, \quad (3.10)$$

$$\mathbb{E}(\mathbf{v}_k \mathbf{v}_j^\top) = V_k \delta_{kj}, \quad (3.11)$$

where δ_{kj} is the Kronecker delta function, and the noise covariance matrix is

$$V_k = \text{diag}(q_1, q_2) \otimes \begin{bmatrix} \frac{1}{3}T_{\text{in},k}^4 & \frac{1}{2}T_{\text{in},k}^3 \\ \frac{1}{2}T_{\text{in},k}^3 & T_{\text{in},k}^2 \end{bmatrix}, \quad (3.12)$$

where q_1 and q_2 are the process noise intensities of the x and y directions, respectively.

3.2.3 Measurement Model

A general measurement model is given by

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k; \mathcal{D}(\mathcal{U}_k)) + \mathbf{n}_k, \quad (3.13)$$

where $\mathbf{h}_k(\cdot)$ is a nonlinear mapping from the target state \mathbf{x}_k to the noiseless measurement, and \mathbf{n}_k is the measurement noise. Note that \mathbf{z}_k could be the time delay τ , Doppler frequency ν , or other measurements extracted from the echo via signal processing techniques [64]. In addition, it is optional to use the raw discretized received signal as \mathbf{z}_k [49]. In practical applications, the choice of \mathbf{z}_k is dependent on the channel capacity, computational resources, and requirements of the algorithms. The dimension of \mathbf{z}_k is NJ , where N is the dimension of the measurement of a single radar, and J is the total number of selected radars.

In this dissertation, we take the raw discretized received signal as an example. We assume that the i_{th} radar is selected by the sensing system to operate at time k ; i.e., $i \in \mathcal{U}_k^s$. In addition, we denote the i_{th} radar, the corresponding selected waveform, and its state as $D_k^{(i)}$. According to (3.2) and assuming that the echo from the target is collected by the i_{th} radar, we have

$$r_k^{(i)}(t) = h_k^{(i)}(t, \mathbf{x}_k; D_k^{(i)}) + n_k^{(i)}(t), \quad t = t_1, \dots, t_N, \quad (3.14)$$

where the measurement noise $n_k^{(i)}(t)$ is white circularly-symmetric complex Gaussian noise, and

$$h_k^{(i)}(t, \mathbf{x}_k; D_k^{(i)}) = \alpha(f_k^{(i)}) s_k^{(i)}(t - \tau_k^{(i)}) e^{j2\pi\nu_k^{(i)}(t - \tau_k^{(i)})} e^{-j2\pi f_c^{(i)} \tau_k^{(i)}}. \quad (3.15)$$

We lump all the samples together to obtain

$$\begin{aligned}\tilde{\mathbf{r}}_k^{(i)} &= \begin{bmatrix} h_k^{(i)}(t_1, \mathbf{x}_k; D_k^{(i)}) \\ \vdots \\ h_k^{(i)}(t_N, \mathbf{x}_k; D_k^{(i)}) \end{bmatrix} + \begin{bmatrix} n_k^{(i)}(t_1) \\ \vdots \\ n_k^{(i)}(t_N) \end{bmatrix} \\ &= \tilde{\mathbf{h}}_k^{(i)}(\mathbf{x}_k; D_k^{(i)}) + \tilde{\mathbf{n}}_k^{(i)}.\end{aligned}\quad (3.16)$$

Since many tracking filters prefer a real expression of the measurement model, we further modify (3.16) to

$$\begin{aligned}\mathbf{z}_k^{(i)} = \mathbf{r}_k^{(i)} &= \begin{bmatrix} \text{Re}\{\tilde{\mathbf{h}}_k^{(i)}(\mathbf{x}_k; D_k^{(i)})\} \\ \text{Im}\{\tilde{\mathbf{h}}_k^{(i)}(\mathbf{x}_k; D_k^{(i)})\} \end{bmatrix} + \begin{bmatrix} \text{Re}\{\tilde{\mathbf{n}}_k^{(i)}\} \\ \text{Im}\{\tilde{\mathbf{n}}_k^{(i)}\} \end{bmatrix} \\ &= \mathbf{h}_k^{(i)}(\mathbf{x}_k; D_k^{(i)}) + \mathbf{n}_k^{(i)},\end{aligned}\quad (3.17)$$

where $\text{Re}\{\cdot\}$ and $\text{Im}\{\cdot\}$ represent the real and imaginary parts of a complex number, respectively. Note that $\mathbf{n}_k^{(i)}$ remains Gaussian. By collecting all the $\mathbf{z}_k^{(i)}$ into one vector \mathbf{z}_k , such that $i \in \mathcal{U}_k^s$, we have

$$\begin{aligned}\mathbf{z}_k &= \text{vec}\mathcal{Z}_k = \text{vec}\{\mathbf{z}_k^{(i)} | i \in \mathcal{U}_k^s\} \\ &= \text{vec}\{\mathbf{h}_k^{(i)}(\mathbf{x}_k; D_k^{(i)}) + \mathbf{n}_k^{(i)} | i \in \mathcal{U}_k^s\} \\ &= \mathbf{h}_k(\mathbf{x}_k; \mathcal{D}(\mathcal{U}_k)) + \mathbf{n}_k,\end{aligned}\quad (3.18)$$

where the vec operator lumps the elements in a set into a vector. Furthermore, we define the covariance matrix as,

$$\mathbb{E}(\mathbf{n}_k \mathbf{n}_j^\top) = R_k \delta_{kj}.\quad (3.19)$$

We should note that the measurement model given by (3.18) assumes the ideal scenario, in which the radars' beams can always cover the target. However, when the beamwidths are narrow or in other practical situations, it is possible that echo received by a selected radar contains only noise when the target is not covered by its beam. For these cases, the actual measurement should be adjusted to $\mathbf{h}_k^{(i)}(\mathbf{x}_k; D_k^{(i)}) = 0$.

3.2.4 Recursive Bayesian Estimation

In target tracking, we aim to recursively estimate the posterior distribution

$$p(\mathbf{x}_k | \mathcal{Z}_{1:k}; \mathcal{D}(\mathcal{U}_k)) = \frac{p(\mathbf{x}_k | \mathcal{Z}_{1:k-1})p(\mathcal{Z}_k | \mathbf{x}_k; \mathcal{D}(\mathcal{U}_k))}{\int p(\mathbf{x}_k | \mathcal{Z}_{1:k-1})p(\mathcal{Z}_k | \mathbf{x}_k; \mathcal{D}(\mathcal{U}_k))d\mathbf{x}_k}, \quad (3.20)$$

where $\mathcal{Z}_{1:k} = \{\mathcal{Z}_1, \mathcal{Z}_2, \dots, \mathcal{Z}_k\}$ and

$$p(\mathbf{x}_k | \mathcal{Z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1})p(\mathbf{x}_{k-1} | \mathcal{Z}_{1:k-1})d\mathbf{x}_{k-1}. \quad (3.21)$$

Note that (3.20) and (3.21) formulate the recursion of the posterior distribution $p(\mathbf{x}_k | \mathcal{Z}_{1:k}; \mathcal{D}(\mathcal{U}_k))$ via the state transition probability $p(\mathbf{x}_k | \mathbf{x}_{k-1})$, the latest measurement model $p(\mathcal{Z}_k | \mathbf{x}_k; \mathcal{D}(\mathcal{U}_k))$ parametrized by $\mathcal{D}(\mathcal{U}_k)$, and the posterior distribution of the last step $p(\mathbf{x}_{k-1} | \mathcal{Z}_{1:k-1})$.

Here we give an example of the extended Kalman filter to clarify the formulations above. The extended Kalman filter gives the sufficient statistics (i.e., the posterior mean and error covariance matrix) to represent the assumed Gaussian posterior distribution $P(\mathbf{x}_k | \mathcal{Z}_{1:k}; \mathcal{D}(\mathcal{U}_k))$ by linearizing the kinematic model and measurement model [5]. The complete recursion of

the extended Kalman filter is given as,

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1|k-1}), \quad (3.22)$$

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^\top + V_{k-1}, \quad (3.23)$$

$$K_k = P_{k|k-1}H_k^\top (H_kP_{k|k-1}H_k^\top + R_k)^{-1}, \quad (3.24)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}; \mathcal{D}(\mathcal{U}_k))), \quad (3.25)$$

$$P_{k|k} = (I - K_kH_k)P_{k|k-1}, \quad (3.26)$$

where $\hat{\mathbf{x}}_{k|k-1}$ is the predicted state, $P_{k|k-1}$ is the corresponding predicted covariance matrix, $\hat{\mathbf{x}}_{k|k}$ is the updated state, and $P_{k|k}$ is the corresponding updated covariance matrix. Note that in the extended Kalman filter, $P(\mathbf{x}_k | \mathcal{Z}_{1:k-1})$ is assumed to be Gaussian with mean $\hat{\mathbf{x}}_{k|k-1}$ and covariance matrix $P_{k|k-1}$, and $P(\mathbf{x}_k | \mathcal{Z}_{1:k}; \mathcal{D}(\mathcal{U}_k))$ is assumed to be Gaussian with mean $\hat{\mathbf{x}}_{k|k}$ and covariance matrix $P_{k|k}$. Furthermore, we have

$$F_{k-1} = \left. \frac{\partial \mathbf{f}_{k-1}(\mathbf{x})}{\partial \mathbf{x}^\top} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-1}}, \quad (3.27)$$

$$H_k = \left. \frac{\partial \mathbf{h}_k(\mathbf{x}; \mathcal{D}(\mathcal{U}_k))}{\partial \mathbf{x}^\top} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}. \quad (3.28)$$

Note that if noisy CV model is used, then F_{k-1} can be replaced by A_{k-1} .

3.2.5 Adaptive Tracking and Recursive Bayesian Estimation

In order to improve the tracking performance, we solve a combinatorial optimization problem for every tracking step after the measurements are received. According to the optimal solution, the radar network adaptively modifies its geometry and assigns a subset of radars to collect measurements using the designed waveforms for the next step. In summary, the

tracking procedure is iterated by alternating between the following two steps:

$$(1) \quad \mathcal{D}(\mathcal{U}_k) = \arg \max_Q f(Q) \quad (3.29)$$

$$\text{s.t. } \mathcal{U}_k^s \in \Pi_k(\mathcal{I}_k^s),$$

$$\mathcal{U}_k^p \in \mathcal{L}_k(\mathcal{U}_k^s, \mathcal{S}_{k-1}^p),$$

$$(2) \quad \hat{\mathbf{x}}_k = \arg \max_{\mathbf{x}_k} P(\mathbf{x}_k | \mathcal{Z}_{1:k}; \mathcal{D}(\mathcal{U}_k)), \quad (3.30)$$

where $f(\cdot)$ is the utility function that evaluates the expected performance of the selected $\mathcal{D}(\mathcal{U}_k)$, \mathcal{I}_k^s is set of the indices of radars that are available for management at time k , and $\Pi_k(\mathcal{I}_k^s)$ is the collection of the subsets of \mathcal{I}_k^s according to a given policy. For example, $\Pi_k(\mathcal{I}_k^s) = \{\mathcal{U} \subset \mathcal{I}_k^s : |\mathcal{U}| \leq \tilde{c}\}$, where \tilde{c} is a given constant, is commonly used in sensor selection. In addition, \mathcal{S}_{k-1}^p denotes the states of all the radars in the sensor network at time $k-1$, and $\mathcal{L}_k(\mathcal{U}_k^s, \mathcal{S}_{k-1}^p)$ is the set of states (e.g., positions and velocities) that the selected radars can achieve at time k within the kinematic capabilities of the radars or as dictated by some predefined searching rules. For example, a radar may be responsible for exploring and monitoring a certain region, and unable to move far away from the initial position. Furthermore, the utility function $f(\cdot)$ can be one step ahead or multi-step ahead, depending on the demands and computational capacity.

As an illustrative example, an one-step ahead objective function could be the expected cross-entropy between the probability density functions (pdfs) of the predicted state and updated state, i.e.,

$$f(\mathcal{D}(\mathcal{U}_k)) = \mathbb{E}_{p(\mathbf{z}_k | \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}; \mathcal{D}(\mathcal{U}_k))} [-\mathbb{E}_{p(\mathbf{x}_k | \mathbf{z}_{1:k-1})} \log p(\mathbf{x}_k | \mathbf{z}_{1:k}; \mathcal{D}(\mathcal{U}_k))], \quad (3.31)$$

where the term in the square bracket is the cross-entropy, measuring the dissimilarity of the two pdfs. We need to take expectation of the cross-entropy with respect to \mathbf{z}_k since \mathbf{z}_k is not collected when solving the combinatorial optimization in (3.31). In order to compute the objective function, we can approximate $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k|\mathbf{z}_{1:k}; \mathcal{D}(\mathcal{U}_k))$ using the output of the extended Kalman filter as suggested by (3.22)-(3.28), regardless of whether \mathbf{z}_k includes some measurements containing only noise. For simplicity, the noisy CV model is used here. Also, note that

$$p(\mathbf{z}_k|\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}; \mathcal{D}(\mathcal{U}_k)) = \int p(\mathbf{z}_k|\mathbf{x}_k; \mathcal{D}(\mathcal{U}_k))p(\mathbf{x}_k|\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1})d\mathbf{x}_k \quad (3.32)$$

$$\approx \sum_j p(\mathbf{z}_k|\tilde{\mathbf{x}}_k^j; \mathcal{D}(\mathcal{U}_k)) \frac{q(L(\tilde{\mathbf{x}}_k^j)|\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1})}{C} \quad (3.33)$$

where an approximation is used to remove the annoying integral in (3.32) to get (3.33) by discretizing the state \mathbf{x}_k into 4-D grids with centers $\tilde{\mathbf{x}}_k^j$, and

$$q(L(\tilde{\mathbf{x}}_k^j)|\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}) = \int_{L(\tilde{\mathbf{x}}_k^j)} p(\mathbf{x}_k|\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1})d\mathbf{x}_k, \quad (3.34)$$

where $L(\tilde{\mathbf{x}}_k^j)$ is the grid that $\tilde{\mathbf{x}}_k^j$ locates, and C is the normalizing constant such that the summation of $q(L(\tilde{\mathbf{x}}_k^j)|\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1})$ is equal to 1. Furthermore, we can assume that the measurement collection processes of the selected radars are independent (i.e., $E(\mathbf{n}_k^{(i)} \mathbf{n}_k^{(j)\top}) = R_k^{(i)} \delta_{ij}$), then

$$p(\mathbf{z}_k|\tilde{\mathbf{x}}_k^j; \mathcal{D}(\mathcal{U}_k)) = \prod_{i \in \mathcal{U}_k^s} p(\mathbf{z}_k^{(i)}|\tilde{\mathbf{x}}_k^j; D_k^{(i)}). \quad (3.35)$$

Note that $p(\mathbf{z}_k^{(i)}|\tilde{\mathbf{x}}_k^j; D_k^{(i)})$ denotes the pdf of the measurement of i_{th} radar when the target state is given, and the expression is given as follows:

$$p(\mathbf{z}_k^{(i)}|\tilde{\mathbf{x}}_k^j; D_k^{(i)}) = \begin{cases} p_{\mathbf{n}_k^{(i)}}(\mathbf{z}_k^{(i)}) & \text{if scenario 1} \\ p_{\mathbf{n}_k^{(i)}}(\mathbf{z}_k^{(i)} - \mathbf{h}_k^{(i)}(\tilde{\mathbf{x}}_k^j; D_k^{(i)})) & \text{if scenario 2} \end{cases},$$

where $p_{\mathbf{n}_k^{(i)}}(\cdot)$ is the pdf of the measurement noise $\mathbf{n}_k^{(i)}$. Note that the difference between the two scenarios:

- Scenario 1: The beam of radar i does not cover $\tilde{\mathbf{x}}_k^j$, and the measurement is only noise.
- Scenario 2: The beam of radar i covers $\tilde{\mathbf{x}}_k^j$, and the measurement is the echo from the target.

It follows that, we have that $\mathbf{z}_k^{(i)}|\tilde{\mathbf{x}}_k^j \sim \mathcal{N}(\boldsymbol{\mu}^{(i)}(\tilde{\mathbf{x}}_k^j, D_k^{(i)}), R_k^{(i)})$, where

$$\boldsymbol{\mu}^{(i)}(\tilde{\mathbf{x}}_k^j, D_k^{(i)}) = \begin{cases} \mathbf{0} & \text{if scenario 1} \\ \mathbf{h}_k^{(i)}(\tilde{\mathbf{x}}_k^j, D_k^{(i)}) & \text{if scenario 2} \end{cases}. \quad (3.36)$$

Thus, we have $\mathbf{z}_k|\tilde{\mathbf{x}}_k^j \sim \mathcal{N}(\boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)), R_k)$, where

$$\boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) = \text{vec}\{\boldsymbol{\mu}^{(i)}(\tilde{\mathbf{x}}_k^j, D_k^{(i)})|i \in \mathcal{U}_k^s\}, \quad (3.37)$$

$$R_k = \text{diag}\{R_k^{(i)}|i \in \mathcal{U}_k^s\}, \quad (3.38)$$

where diag is the operator that rearranges the elements in the set into a block diagonal matrix.

$$\begin{aligned}
H_{\text{cross}} &= -\mathbb{E}_{p(\mathbf{x}_k|\mathbf{z}_{1:k-1})} \log p(\mathbf{x}_k|\mathbf{z}_{1:k}; \mathcal{D}(\mathcal{U}_k)) \\
&= \mathbb{E}_{p(\mathbf{x}_k|\mathbf{z}_{1:k-1})} \left[\frac{1}{2} \log(2\pi)^4 + \frac{1}{2} \log |P_{k|k}| + \frac{1}{2} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^\top P_{k|k}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}) \right] \\
&= \frac{1}{2} \log(2\pi)^4 + \frac{1}{2} \log |P_{k|k}| + \frac{1}{2} \text{tr}(P_{k|k}^{-1} P_{k|k-1}) + \frac{1}{2} (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1})^\top P_{k|k}^{-1} (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1})
\end{aligned} \tag{3.39}$$

$$\begin{aligned}
f(\mathcal{D}(\mathcal{U}_k)) &= \mathbb{E}_{p(\mathbf{z}_k|\mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1|k-1}; \mathcal{D}(\mathcal{U}_k))} [H_{\text{cross}}] \\
&\approx \frac{1}{2} \log(2\pi)^4 + \frac{1}{2} \log |P_{k|k}| + \frac{1}{2} \text{tr}(P_{k|k}^{-1} P_{k|k-1}) + \frac{1}{2} \text{tr}(K_k^\top P_{k|k}^{-1} K_k R_k) \\
&\quad + \frac{1}{2C} \sum_j [q(L(\tilde{\mathbf{x}}_k^j) | \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}) \times \\
&\quad (\boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}))^\top K_k^\top P_{k|k}^{-1} K_k (\boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}))] \tag{3.40}
\end{aligned}$$

Based on the above discussion and the extended Kalman filter, we can compute the cross-entropy as in (3.39) and have the explicit expression of the objective function in (3.40). A more detailed derivation is given in Appendix C.1 and C.2.

The state estimation shown by (3.30) is the maximum a posteriori (MAP) inference. Other estimators, such as the posterior mean, can also be selected if they fit better in the practical design. For the extended Kalman filter, the updated state shown by (3.25) is a good estimator.

3.2.6 Target Tracking with Multiple Target Models

Note that if the model of the complex reflectivity $\alpha(f_k^{(i)})$ with respect to $f_k^{(i)}$ is known by the sensing system, then the objective function like that in (3.40) can be directly applied. However, in many real applications, since the sensing system does not know the type of the

target to track, it is unable to know how $f_k^{(i)}$ affects the complex reflectivity $\alpha(f_k^{(i)})$. In this dissertation, we assume that there exists a target model library, which includes multiple pre-trained target models. Note that each target model is associated with a mapping from the modulated frequency $f_k^{(i)}$ to $\alpha(f_k^{(i)})$. Formally, the target model library is given by

$$\mathcal{T} = \{\alpha_1(\cdot), \alpha_2(\cdot), \dots, \alpha_M(\cdot)\}, \quad (3.41)$$

where M is total number of the target models in the library. We assume that the actual target model is contained in the target model library. Now, we replace $f(\mathcal{D}(\mathcal{U}_k))$ with $f(\mathcal{D}(\mathcal{U}_k); \alpha_m)$ in order to emphasize the dependence of the objective function on the target model m . Then we have the modified objective function which will be applied to the combinatorial optimization problem in (3.29):

$$\bar{f}(\mathcal{D}(\mathcal{U}_k)) = \frac{1}{M} \sum_{m=1}^M f(\mathcal{D}(\mathcal{U}_k); \alpha_m). \quad (3.42)$$

Note that for simplicity, here we just average $f(\mathcal{D}(\mathcal{U}_k); \alpha_m)$ with respect to all the target models in the library by implicitly assuming the all the target models have the same probability of being correct at every tracking step. A more complicated strategy could be averaging the $f(\mathcal{D}(\mathcal{U}_k); \alpha_m)$ by assigning different weights to the target models in the library \mathcal{T} , and updating the weights at every step, using the information from consecutively received measurements.

Also, if the target model $\alpha(f_k^{(i)})$ is known in advance, then we can directly apply the recursive Bayesian state estimation to compute the posterior pdf $P(\mathbf{x}_k | \mathcal{Z}_{1:k}; \mathcal{D}(\mathcal{U}_k))$. If the actual target model is not available, then we need to average the posterior pdf derived from a single target model m , denoted as $P_m(\mathbf{x}_k | \mathcal{Z}_{1:k}; \mathcal{D}(\mathcal{U}_k))$, on target models in the target model library.

Using the extended Kalman filter as an example, let the updated state and the corresponding covariance matrix with the m_{th} target model be denoted as $\mathbf{x}_{k|k,m}$ and $P_{k|k,m}$ respectively, and we have

$$\hat{\mathbf{x}}_{k|k} = \frac{1}{M} \sum_{m=1}^M \hat{\mathbf{x}}_{k|k,m}, \quad (3.43)$$

$$P_{k|k} = \frac{1}{M} \sum_{m=1}^M \left(P_{k|k,m} + \hat{P}_{k,m} \right), \quad (3.44)$$

where

$$\hat{P}_{k,m} = (\hat{\mathbf{x}}_{k|k,m} - \hat{\mathbf{x}}_{k|k})(\hat{\mathbf{x}}_{k|k,m} - \hat{\mathbf{x}}_{k|k})^\top. \quad (3.45)$$

Note that the above multiple model estimation also treats multiple target models having the same possibility as the actual model, as what we do when dealing with the objective function. More complicated multiple model estimation methods could be employed [5].

The sketch of the target tracking procedure is described as algorithm 4. In general, nested loops are needed to exhaustively search for the optimal solution of a combinatorial optimization problem at every step. In practical applications, it is realistic to select the same or nearly the same waveform parameters between consecutive dwells (steps) separated by a very short time period, such that the computational complexity can be reduced on average. Furthermore, the objective function of the optimization problem can be carefully designed with some good properties, e.g., submodularity. In this way, some greedy methods can be applied to solve the combinatorial optimization problem with guaranteed performance.

Algorithm 4: Target Tracking in a Network of Radars

Input : Initial pdf of the target state $p(\mathbf{x}_0)$, and initial radars' state S_0^p

Output: The optimal set $\hat{\mathcal{D}}(\mathcal{U}_k)$, the posterior pdf $P(\mathbf{x}_k|\mathcal{Z}_{1:k};\mathcal{D}(\mathcal{U}_k))$, and MAP inference $\hat{\mathbf{x}}$

```
1  $k \leftarrow 1$ ;  
2 while Stop condition is not met do  
3   for each target model  $m$  do  
4     for each subset of radars  $\mathcal{U}_k^s \in \Pi_k(\mathcal{I}_k^s)$  do  
5       for states of radars  $\mathcal{U}_k^p \in \mathcal{L}_k(\mathcal{U}_k^s, \mathcal{S}_{k-1}^p)$  do  
6         for waveforms of radars  $\mathcal{U}_k^w$  do  
7            $\lfloor$  Compute  $f(\mathcal{D}(\mathcal{U}_k); \alpha_m)$ ;  
8     Compute the objective function  $\bar{f}(\mathcal{D}(\mathcal{U}_k))$ ;  
9     Search the solution  $\hat{\mathcal{D}}(\mathcal{U}_k) = \arg \max \bar{f}(\mathcal{D}(\mathcal{U}_k))$ ;  
10    Get the measurement  $\mathbf{z}_k$  by the setting  $\hat{\mathcal{D}}(\mathcal{U}_k)$ ;  
11    for each target model  $m$  do  
12       $\lfloor$  Get  $P_m(\mathbf{x}_k|\mathcal{Z}_{1:k};\hat{\mathcal{D}}(\mathcal{U}_k))$  using recursive Bayesian state estimation;  
13    Average on the target model to get the posterior pdf  $P(\mathbf{x}_k|\mathcal{Z}_{1:k};\hat{\mathcal{D}}(\mathcal{U}_k))$ ;  
14    Update radars' state  $\mathcal{S}_k^p$  by a predefined rule for the radars that are not selected;  
15     $k \leftarrow k + 1$ ;
```

3.3 Simulation Results

The framework given by Section 3.2 is general, which would fit many tracking applications in the sensor network, and numerical simulations are needed to demonstrate the framework. In this section, an illustrative example is discussed in order to formulate a specific scenario as shown in Figure 3.2. The Figure 3.2 illustrates a network with small number of radars for the sake of simple illustration; however it represents a scenario with large-scale network of cognitive radars which have certain constraints about coverage area and path. Especially, assuming that each radar in the network is aware of the location of the other radars in the network and also assuming that the target is in the far-field compared to all the radars in the network, the scenario presents here a realistic case in which the overlap among the radars

along the same path is avoided. Moreover, without loss of generality, the paths are assumed to be along the x and y axes in Figure 3.2; however, in general these radar paths could be rotated. Based on this scenario, a simulation-based validation demonstrates the feasibility of the proposed method.

3.3.1 Simulation Settings (a Scenario)

As shown in Fig. 3.2, a total of J_{total} radars are moving along the x and y axes to track a single target in a 2-D plane. A central processing system fuses the measurements from all the operative radars and manages the network by dictating the waveform design, path planning, and radar selection. In order to simplify the case, we make the following assumption for this illustrative example.

- All the radars that move in the same direction transmit waveforms with the same range resolution by using the waveforms given by the specific example in Section 3.2.1. Thus, the 2-D plane can be divided into several lattices. Each lattice has the size $\delta x \times \delta y$, where δx and δy are the range resolutions of the waveforms transmitted by the radars moving on the y and x axes, respectively.
- At every tracking step, each radar is located only at the discrete coordinates on x and y axes corresponding to the centers of rows and columns formed by lattices.
- Each radar is able to emit a signal only in the direction orthogonal to its moving direction. For example, radar x_i can emit a signal only in y direction.

- The beamwidths of the radars are considered to cover only one column or one row. For example, if a radar is moving in the x direction, its beam covers only one column of lattices if it is selected to operate.
- The target is moving in the far field, which is a common scenario in practice. Since the beamwidth of a radar is comparable to its range resolution in this case, for the echo from the target, we can approximate the time delay and Doppler frequency in simpler expressions. In other words, if i_{th} radar on the x axis receives the echo from the target, we can approximate (3.3) and (3.4) as

$$\tau^{(i)} \approx \frac{2y}{c}, \quad \nu^{(i)} \approx -\frac{2f_c \dot{y}}{c}, \quad (3.46)$$

where the approximation is due to that the far field assumption results in $|x - x^{(i)}| \ll |y - y^{(i)}| = |y|$ for the i_{th} radar on the x axis. A similar approximation can be applied to radars on the y axis. We emphasize that the time delay and Doppler frequency of the echo from the target are not functions of the position or velocity of the radar if it receives the echo.

- In addition, since we have multiple radars moving on the each axis, we let each radar have its own predefined assigned surveillance space which consists of contiguous columns or rows, and the surveillance spaces of different radars are not overlapped if they are moving in the same direction. Also, the union of the assigned surveillance regions of the radars moving in the same direction covers the whole workspace of interest. In this example, we assign each radar in the network the responsibility of monitoring 3 consecutive columns or rows in the space. The centers of the assigned surveillance regions of radar x_1 and y_1 correspond to the lattice that initial state $\hat{\mathbf{x}}_{0|0}$

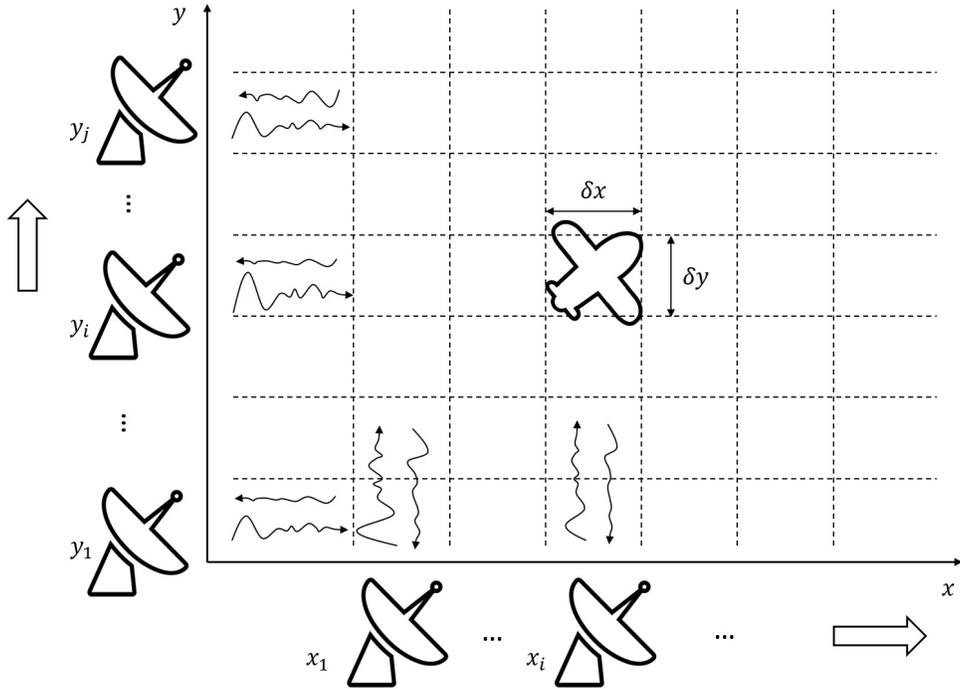


Figure 3.2: Model settings of the illustrative example of the framework.

locates. In order to reduce the searching complexity of solving the combinatorial optimization problem, at each time k , only 5 columns or rows near the predicted state $\hat{\mathbf{x}}_{k+1|k}$ and the corresponding radars are considered.

- Target remains in only one lattice during the time when the measurements of a tracking step are collected.
- Radar measurement collection procedures are independent; i.e., the measurement of one radar is not affected by signals emitted by other radars and is independent of the measurements of other radars.
- The power of the measurement noise of each radar is assumed to be known in advance, and the environment is assumed to be stationary.

In summary, each radar is responsible for a predefined surveillance region, and it transmits a signal covering one column or one row of the lattices in its region if it is selected by the central processing system to operate.

In this illustrative example, we let the target start with a state sampled from a Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_{0|0}, P_{0|0})$, where $\hat{\mathbf{x}}_{0|0} = [630 \text{ m}, 150 \text{ m/s}, 630 \text{ m}, 150 \text{ m/s}]^\top$ and $P_{0|0} = \text{diag}(1, 0.01, 1, 0.01)$. For the signal model shared by all the radars in the network, we set $u(t)$ in (3.5) as a square pulse, with pulse width $t_{\text{width}} = 10^{-7}$ s and power 1. The carrier frequency is $f_c = 3$ GHz, and the modulated frequency $f_c + f_\xi^{(i)}$ can be selected from $[f_c - (1 \text{ MHz}), f_c, f_c + (1 \text{ MHz})]^\top$ in every tracking step. In addition, the speed of light $c = 3 \times 10^8$ m/s, and the range resolutions can be computed as $\delta x = \delta y = c \times t_{\text{width}}/2 = 15$ m.

We adopt the noisy constant velocity (CV) model suggested by (3.7) as the underlying kinematic model, where the interval between two tracking steps is $T_{\text{in}} = 0.005$ s or 0.05 s, which varies in the different experiments. The process noise intensities, q_1 and q_2 , vary in the different experiments. Also, we use the raw discretized signals received from the selected radars as the measurements, and the corresponding measurement model is given by (3.18). In this model, the sampling frequency is $f_s = 20$ MHz. We take the first 334 sample points as the measurement vector in the experiments. The power of the measurement noise is $\mathbb{E}[n_k^{(i)}(t_p)n_k^{(i)}(t_q)^*] = 0.02\delta_{pq}$ for all time k and radar i , where t_p and t_q correspond to the time of two sample points. We use the extended Kalman filter to do the recursive Bayesian state estimation, as stated in Section 3.2.4.

The objective function of the combinatorial optimization problem in (3.29) could be given by the expected cross-entropy, where the definition is shown in (3.31), and the analytical expression is given by (3.40). Note that in Section 3.2.5, we approximate the integral in (3.32) with a summation by discretizing the state space of \mathbf{x}_k into 4-D grids. Here, we

discretize only the position of the state space and fix the velocity of \mathbf{x}_k as the predicted velocity of $\hat{\mathbf{x}}_{k|k-1}$ in order to reduce the computational complexity.

The feasible set for the sensor selection within the constraints of (3.29) is defined as

$$\Pi_k(\mathcal{I}_k^s) = \{\mathcal{U} \subset \mathcal{I}_k^s : |\mathcal{U}| = 2, \text{ and two radars are moving on the } x \text{ and } y \text{ axes respectively}\}, \quad (3.47)$$

which means that for each axis, we select one radar to get the measurements. This is a reasonable policy since the assigned surveillance regions are not overlapped for radars that move in the same direction. Based on this policy, \mathcal{U}_k^s can be expressed by $(x_{i,k}, y_{j,k})$, where $x_{i,k}$ and $y_{j,k}$ are two indices of radars moving on the x and y axes respectively.

For the path planning constraints $\mathcal{U}_k^p \in \mathcal{L}_k(\mathcal{U}_k^s, \mathcal{S}_{k-1}^p)$, we additionally assume that each radar is capable of transmitting the signal to any column or row in its assigned surveillance region at any step. In other words, each radar can move fast enough between any two tracking steps in its assigned range. In this example, the path planning constraint $\mathcal{U}_k^p \in \mathcal{L}_k(\mathcal{U}_k^s, \mathcal{S}_{k-1}^p)$ dictates that radars do not leave the ranges corresponding to their assigned surveillance regions. Furthermore, we do not need to take the velocities of sensors into account when solving the combinatorial optimization problem, since they do not affect the objective function and the path planning constraint.

Moreover, multiple target models, as shown in Section 3.2.6, are considered in this example. We assume that there are three target models in the target model library $\{\alpha_m(f_\xi) : m = 1, 2, 3\}$. Specifically, they are $\alpha_1(\{-1, 0, 1\}) = \{1/\sqrt{21}, 2/\sqrt{21}, 4/\sqrt{21}\}$, $\alpha_2(\{-1, 0, 1\}) = \{2/\sqrt{9}, 1/\sqrt{9}, 2/\sqrt{9}\}$, and $\alpha_3(\{-1, 0, 1\}) = \{1/\sqrt{14}, 2/\sqrt{14}, 3/\sqrt{14}\}$. In the simulation, we

Algorithm 5: Target Tracking in the illustrative example

Input : Initial State $\hat{\mathbf{x}}_{0|0}$ and Covariance Matrix $P_{0|0}$

Output: Selected radar pair $(x_{i,k}, y_{j,k})$, radar waveforms $(f_{\xi,k}^{w_x}, f_{\xi,k}^{w_y})$, and the radar positions $(r^{p_x}(x_{i,k}), r^{p_y}(y_{j,k}))$ in the ranges of the assigned surveillance regions at every step k . The updated state $\hat{\mathbf{x}}_{k|k}$ and covariance matrix $P_{k|k}$.

```
1  $k \leftarrow 1$ ;  
2 while Stop condition is not met do  
3   for each target model  $m$  do  
4     for each radar pair do  
5       for each position pair in the ranges of the assigned regions of the selected  
6         radar pair do  
7           for each waveform pair do  
8              $\lfloor$  Compute  $f(\mathcal{D}(\mathcal{U}_k); \alpha_m)$ ;  
9           Compute the objective function  $\bar{f}(\mathcal{D}(\mathcal{U}_k))$ ;  
10          Search the solution  $\hat{\mathcal{D}}(\mathcal{U}_k) = \arg \max \bar{f}(\mathcal{D}(\mathcal{U}_k))$ ;  
11          Get the measurement  $\mathbf{z}_k$  by the setting  $\hat{\mathcal{D}}(\mathcal{U}_k)$ ;  
12          for each target model  $m$  do  
13             $\lfloor$  Get  $\hat{\mathbf{x}}_{k|k,m}$  and  $P_{k|k,m}$  using the extended Kalman filter;  
14          Average on the target model to get  $\hat{\mathbf{x}}_{k|k}$  and  $P_{k|k}$ ;  
15           $k \leftarrow k + 1$ ;
```

use the first target model α_1 as the underlying target model, which is not known by the sensor system.

In summary, the sketch of the target tracking procedure in this illustrative example can be described as algorithm 5.

3.3.2 Simulation Results

All the experiments conduct 60 tracking steps, and all the relative errors are computed using 30 Monte Carlo runs in this section. The performance of the framework on the illustrative

example is illustrated in Fig. 3.3, Fig. 3.4, and Fig. 3.5. Fig. 3.3 shows the performance of the tracking procedure with process noise intensities $q_1 = q_2 = 0.1$ and the tracking time interval $T_{\text{in}} = 0.005$ s. Fig. 3.3(a) shows a representative example of the true trajectory and estimated trajectory. Fig. 3.3(b)-3.3(e) show the relative errors of the positions and velocities of the updated state. Fig. 3.3(f) shows the waveforms used by the radars on the x and y axes in the representative example. Fig. 3.4 shows the corresponding performance with $q_1 = q_2 = 0.1$ and $T_{\text{in}} = 0.05$ s. We can see from the results that the proposed framework is capable of tracking a single target in the sensor network under these two sets of parameters. From the representative examples of the trajectories, we note that the estimated states are close to the true states, which is also verified by the behaviors of the relative errors. We also notice that when the tracking time interval T_{in} (i.e., pulse repetition interval (PRI) in this example), is relatively small, further decreasing it could provide better velocity tracking performance after the algorithm converges, by comparing relative errors of velocities \dot{x} and \dot{y} in Fig. 3.3 and Fig. 3.4. Note that shorter PRI usually requires the higher power usage, and longer tracking time interval means higher uncertainty of the target state between two tracking steps, thus hindering the tracking performance. Hence, in practical use, the sensor network should utilize the freedom of choosing an appropriate PRI to balance the power usage and tracking performance, or adaptively adjust the PRI to address different scenarios.

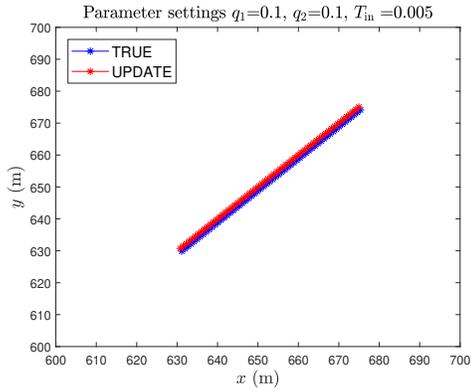
When the process noise intensities become larger, which is shown by Fig. 3.5 with $q_1 = q_2 = 10$ and $T_{\text{in}} = 0.05$ s, the relative errors increase as tracking time moves on, especially the relative errors of the velocities. This is in accordance with the intuition, since when process noise intensities are larger, the motion of the target becomes more unpredictable. In addition, the extended Kalman filter is a sub-optimal tracking filter and the errors caused by linearization accumulate, resulting in such a scenario. Note that the simulation generates the true state using this noisy CV kinematic model. In practice, the target maneuvers by

switching itself among several predefined motion modes, such as constant velocity mode, accelerating mode, and turning mode. Thus, the noisy CV kinematic model with large process intensities might not be a real case for target's motion, and multiple model-based tracking filters, such as the interacting multiple model (IMM) filter, are needed to address practical situations.

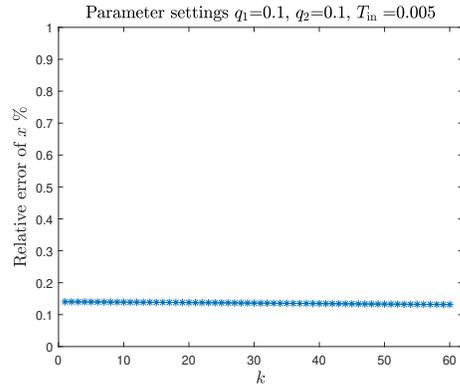
The waveforms used by the radars moving on x and y axes are shown in Fig. 3.3(f), 3.4(f), and 3.5(f) for these three sets of parameters. Note that for all the three cases, the radars prefer to use the frequency $f_\xi + f_c = f_c + (1 \text{ MHz})$. Note that the underlying true target model is $\alpha_1(\{-1, 0, 1\}) = \{1/\sqrt{21}, 2/\sqrt{21}, 4/\sqrt{21}\}$. Thus, the radars tend to use the frequencies corresponding to highest signal-to-noise ratio (SNR), such that more information can be collected from the measurements to maximize the objective function. Moreover, we need to claim that the waveform design is also affected by the target model library. In this example, we assume that all the target models in the library have the same probabilities to be correct along the whole tracking time, and we can see that all the target models in the library suggest that $f_\xi + f_c = f_c + (1 \text{ MHz})$ corresponds to highest SNR. In practice, we may need to update the probability of each target model by utilizing the received measurements.

3.4 Chapter Summary

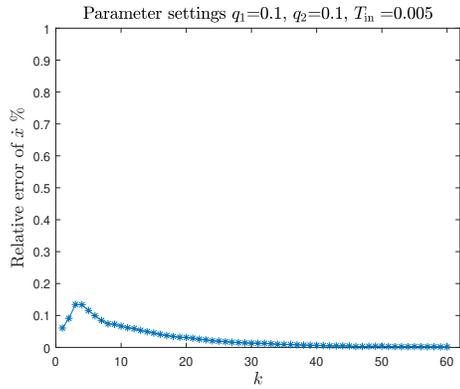
In this chapter, we proposed a framework for single target tracking in a cognitive network of radars, jointly considering waveform design, path planning, and radar selection. We applied the theories of DGM and RBSE to formulate the framework, which consists of alternately solving the combinatorial optimization problem and estimating the target state. In addition, the framework was demonstrated by an illustrative example and verified by



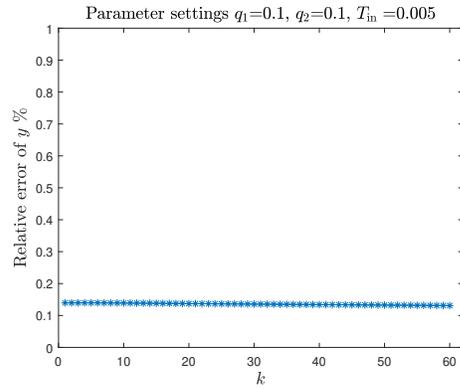
(a) The true and estimated trajectories



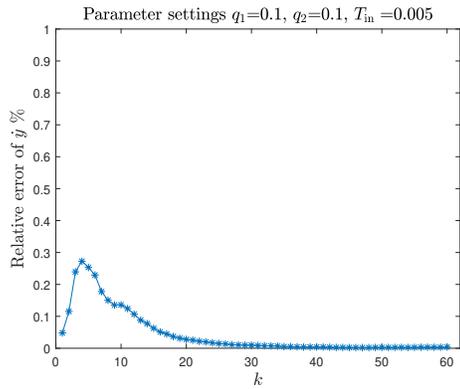
(b) Relative error of x



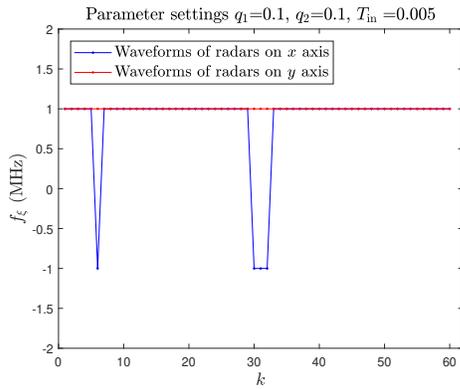
(c) Relative error of \dot{x}



(d) Relative error of y

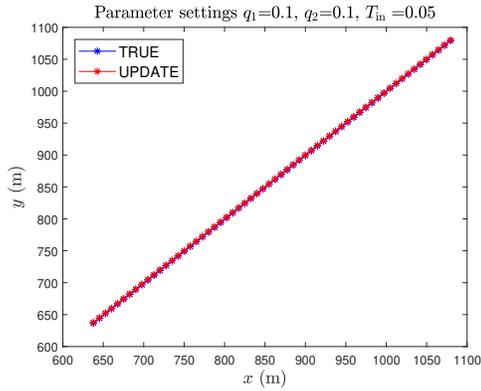


(e) Relative error of \dot{y}

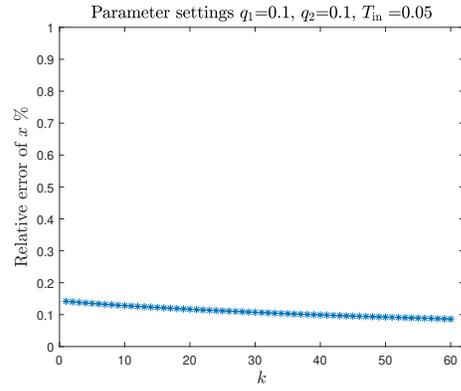


(f) Waveforms used by radars

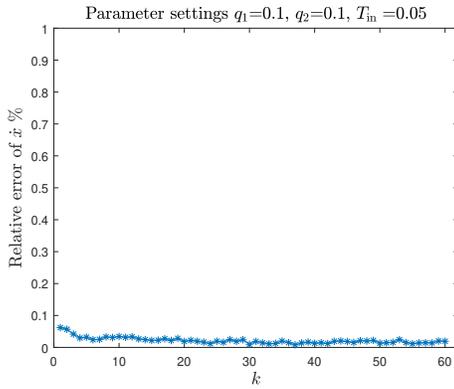
Figure 3.3: Tracking performance when process noise intensities $q_1 = q_2 = 0.1$, and the tracking time interval $T_{in} = 0.005$ s. (a) Comparison of the true and estimated trajectories of the target, (b) relative error of the position x , (c) relative error of the velocity \dot{x} , (d) relative error of the position y , (e) relative error of the velocity \dot{y} , (f) waveforms used by radars on the x and y .



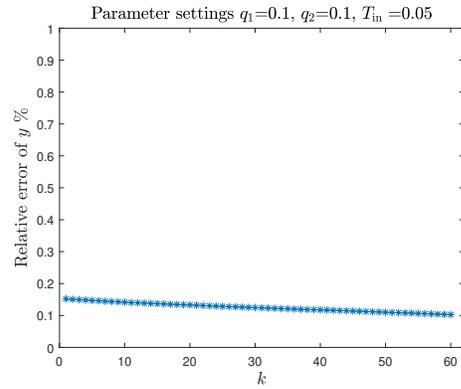
(a) The true and estimated trajectories



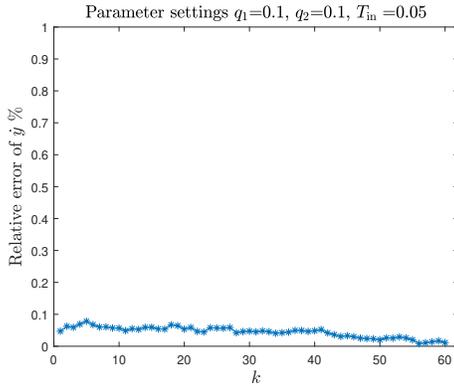
(b) Relative error of x



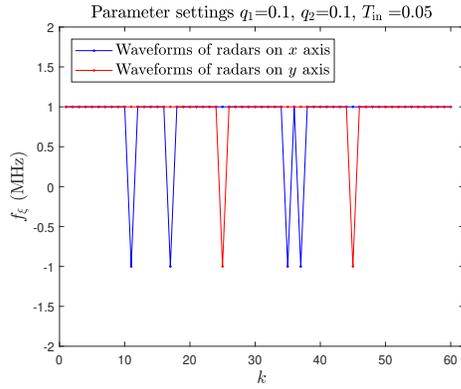
(c) Relative error of \dot{x}



(d) Relative error of y

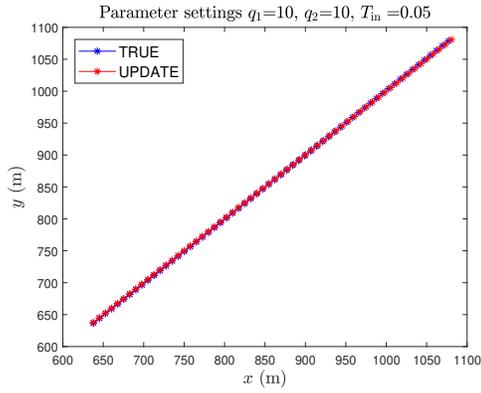


(e) Relative error of \dot{y}

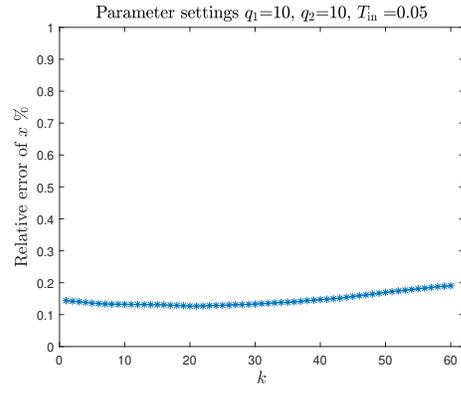


(f) Waveforms used by radars

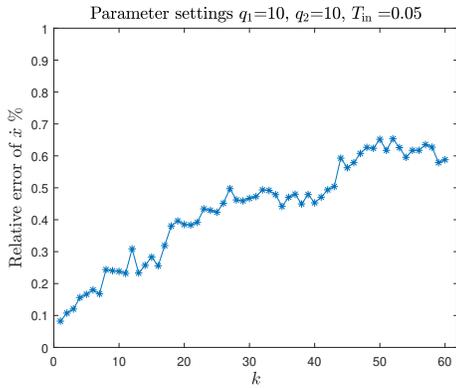
Figure 3.4: Tracking performance when process noise intensities $q_1 = q_2 = 0.1$, and the tracking time interval $T_{in} = 0.05$ s. (a) Comparison of the true and estimated trajectories of the target, (b) relative error of the position x , (c) relative error of the velocity \dot{x} , (d) relative error of the position y , (e) relative error of the velocity \dot{y} , (f) waveforms used by radars on the x and y .



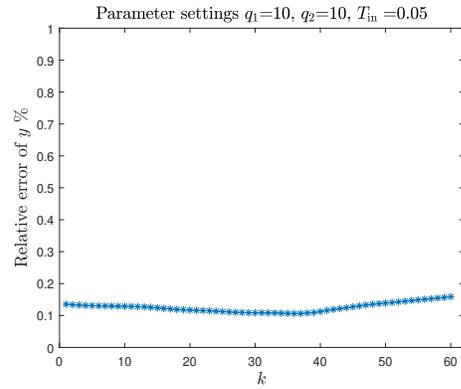
(a) The true and estimated trajectories



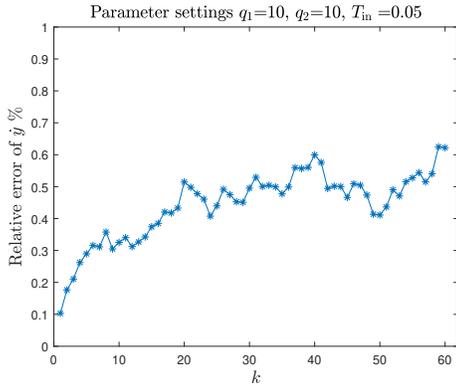
(b) Relative error of x



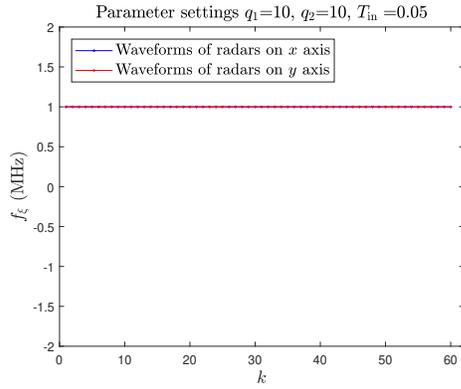
(c) Relative error of \dot{x}



(d) Relative error of y



(e) Relative error of \dot{y}



(f) Waveforms used by radars

Figure 3.5: Tracking performance when process noise intensities $q_1 = q_2 = 10$, and the tracking time interval $T_{in} = 0.05$ s. (a) Comparison of the true and estimated trajectories of the target, (b) relative error of the position x , (c) relative error of the velocity \dot{x} , (d) relative error of the position y , (e) relative error of the velocity \dot{y} , (f) waveforms used by radars on the x and y .

numerical simulations. We emphasize that the proposed framework is not limited to the illustrative example, and it can be applied to much more complicated scenarios.

Chapter 4

Conclusions and Future Work

In this chapter, we summarize the key contributions of our work in Section 4.1 and discuss future work in Section 4.2.

4.1 Summary and Conclusions

In this dissertation, we focused on developing methods for cognitive radar detection in non-stationary environments and on target tracking in cognitive radar networks.

To design a cognitive radar system for target detection in nonstationary environments, we proposed a framework that integrates change-point detection, clutter distribution learning, and adaptive detection. Specifically, we formulated the change-point detection as a sequential hypothesis test and utilized the extended CUSUM algorithm to rapidly detect the possible change-points of the clutter distributions. We used kernel density estimation to learn the new clutter distributions from random clutter samples and applied a similarity measure to identify the clutter distribution from a pre-trained dictionary. Based on the newly learned clutter distribution, we adaptively selected the corresponding likelihood ratio

test and threshold to avoid performance degradation caused by detector mismatch. In the simulation, we first demonstrated the performance of the extended CUSUM algorithm and the clutter identification method. For the whole framework, we investigated not only the adaptive detection performance when focusing on one change-point, but also the proposed framework's dynamic behavior along the time axis when handling multiple change-points, which reflects the new framework's capability for real-time applications.

To design a cognitive radar network that can intelligently track the target in its surveillance space, we proposed a framework that jointly designs the transmit waveform, plans the radar trajectory, and selects the appropriate radar subset. We built the system on the theories of dynamic graphical models (DGM) and recursive Bayesian state estimation (RBSE), and formulated the target tracking procedure as two iterative steps. In the first step, we chose the expected cross-entropy as the objective function and solved a combinatorial optimization problem to select the optimal subset of radars that transmit well-designed waveforms from their best locations for receiving the most informative measurements during the next tracking instant. In the second step, we acquired the RBSE to accurately track the target, based on measurements collected by the radar network with these optimal waveforms, radar locations, and selected radars. We also validated the proposed framework by constructing an illustrative case, in which radars with certain movement constraints track a single target in a 2-D space. We showed that the sensing system of the cognitive radar network can achieve satisfying tracking performance with relatively low process noise.

4.2 Future Work

For cognitive target detection, we can extend our work to address the following topics:

Correlated clutter model: In the current work, to illustrate the idea of the proposed framework, an i.i.d. assumption was imposed for the clutter samples between pulses. This assumption can be removed if, in the extended CUSUM algorithm, we can derive the confidence boundary for the parameters of the multivariate distribution of the sample vector in one range cell, and then use the multi-dimensional kernel density estimation (KDE) for the clutter distribution identification. However, if the dimension of the clutter sample vector is high, KDE may suffer from the curse of dimensionality. Thus, we will need to propose a change-point detection method and a clutter distribution identification method that are robust in high dimensional applications.

Off-grid effects: In this dissertation, we assumed that the clutter distribution that a radar encounters on-the-fly can be found in the dictionary. In practice, this assumption may not hold, and parameters of the true clutter distribution may be different from those of the most similar clutter distribution in the dictionary. A naive method is to refine the dictionary to include more clutter distributions, but this will require more memory and increase the computational load. Another method is select several similar clutter distributions in the dictionary and merge them to approximate the true clutter distribution, but the merging algorithm must be carefully designed for high approximation accuracy and efficiency. Further, with this merging method, we will need to consider how a detector can be designed without compromising the false alarm rate. In sum, we will need to propose an efficient algorithm that can address off-grid effects.

Dynamic dictionary size: In this dissertation, we used a pre-trained dictionary that includes versatile clutter distributions. The computational load of the clutter distribution identification algorithm depends on the size of the dictionary. As mentioned, a large dictionary can help to accurately identify clutter distribution, at the cost of increased computational load.

We could develop a dynamic mechanism that controls the size of the dictionary to balance the computational load and performance of adaptive detection.

For target tracking in cognitive radars, we can extend our work in the following directions:

Design objective functions: In this dissertation, we used only a grid search method to solve the combinatorial optimization problem. For a more efficient solver, an objective function with more particular properties can be designed. For example, if an objective is submodular, then the combinatorial optimization can be solved in a greedy manner with guaranteed performance. Further, sometimes, we will need to consider multiple goals in target tracking, in which case a multi-objective optimization technique can be applied to achieve Pareto-optimality.

Nonstationary environments: With regard to target tracking, this dissertation does not consider nonstationary environments. We could build a bridge between our work on cognitive target detection in nonstationary environments and target tracking in a network of radars, so that the two topics could benefit each other. We could also develop a statistical environment model that includes geographic information, which would make the nonstationarity of the environment more predictable.

Bibliography

- [1] M. A. Richards, *Fundamentals of Radar Signal Processing*. Tata McGraw-Hill Education, 2005.
- [2] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. II: Detection Theory*. Prentice Hall, 1998.
- [3] J. DiFranco and W. Rubin, *Radar Detection*. SciTech Publishing Inc., 2004.
- [4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [5] C.-B. Chang and K.-P. Dunn, *Applied State Estimation and Association*. MIT Press, 2016.
- [6] S. Haykin, “Cognitive radar: A way of the future,” *IEEE Signal Processing Magazine*, vol. 23, no. 1, pp. 30–40, Jan. 2006.
- [7] 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.
- [8] J. R. Guerci, R. M. Guerci, M. Rangaswamy, J. Bergin, and M. Wicks, “CoFAR: Cognitive fully adaptive radar,” in *IEEE Radar Conference*, May 2014, pp. 0984–0989.
- [9] S. Haykin and D. J. Thomson, “Signal detection in a nonstationary environment reformulated as an adaptive pattern classification problem,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2325–2344, Nov 1998.
- [10] A. O. Hero and D. Cochran, “Sensor management: Past, present, and future,” *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3064–3075, 2011.
- [11] E. Conte, A. De Maio, and C. Galdi, “Statistical analysis of real clutter at different range resolutions,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, pp. 903–918, 2004.
- [12] M. Sekine, T. Musha, Y. Tomita, T. Hagiwara, T. Irabu, and E. Kiuchi, “On Weibull-distributed weather clutter,” *IEEE Transactions on Aerospace Electronic Systems*, vol. 15, pp. 824–830, 1979.

- [13] S. Watts, "Radar detection prediction in K-distributed sea clutter and thermal noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-23, no. 1, pp. 40–45, 1987.
- [14] F. Gini, M. Greco, M. Diani, and L. Verrazzani, "Performance analysis of two adaptive radar detectors against non-Gaussian real sea clutter data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 4, pp. 1429–1439, 2000.
- [15] J. Wang, A. Dogandzic, and A. Nehorai, "Maximum likelihood estimation of compound-Gaussian clutter and target parameters," *IEEE Transactions on Signal Processing*, vol. 54, no. 10, pp. 3884–3898, 2006.
- [16] G. Sun, Z. He, and Y. Zhang, "Distributed airborne MIMO radar detection in compound-Gaussian clutter without training data," *Circuits, Systems, and Signal Processing*, vol. 37, no. 10, pp. 4617–4636, 2018.
- [17] M. Rangaswamy, D. D. Weiner, and A. Ozturk, "Non-Gaussian random vector identification using spherically invariant random processes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 1, pp. 111–124, 1993.
- [18] G. Cui, L. Kong, and X. Yang, "GLRT-based detection algorithm for polarimetric MIMO radar against sirv clutter," *Circuits, Systems, and Signal Processing*, vol. 31, no. 3, pp. 1033–1048, 2012.
- [19] A. Farina, A. Russo, and F. Studer, "Coherent radar detection in log-normal clutter," in *IEE Proceedings F-Communications, Radar and Signal Processing*, vol. 133. IET, 1986, pp. 39–53.
- [20] A. Ozturk, "An application of a distribution identification algorithm to signal detection problems," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1, 1993, pp. 248–252.
- [21] P. Chen, W. L. Melvin, and M. Wicks, "Screening among multivariate normal data," *Journal of Multivariate Analysis*, vol. 69, no. 1, pp. 10 – 29, 1999.
- [22] J. Metcalf, S. Blunt, and B. Himed, "A machine learning approach to distribution identification in non-Gaussian clutter," in *IEEE Radar Conference*, May 2014, pp. 0739–0744.
- [23] J. Metcalf, "Signal processing for non-Gaussian statistics: Clutter distribution identification and adaptive threshold estimation," Ph.D. dissertation, University of Kansas, 2015.
- [24] P. Wang, H. Li, and B. Himed, "Knowledge-aided parametric tests for multichannel adaptive signal detection," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 5970–5982, Dec 2011.

- [25] —, “Bayesian parametric GLRT for knowledge-aided space-time adaptive processing,” in *IEEE Radar Conference*, May 2011, pp. 329–332.
- [26] P. Wang, Z. Sahinoglu, M.-O. Pun, H. Li, and B. Himed, “Knowledge-aided adaptive coherence estimator in stochastic partially homogeneous environments,” *IEEE Signal Processing Letters*, vol. 18, no. 3, pp. 193–196, March 2011.
- [27] P. Wang, H. Li, Z. Wang, and B. Himed, “Knowledge-aided parametric adaptive matched filter with automatic combining for covariance estimation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6067–6071.
- [28] J. R. Guerci and E. J. Baranoski, “Knowledge-aided adaptive radar at DARPA: An overview,” *IEEE Signal Processing Magazine*, vol. 23, no. 1, pp. 41 – 50, Jan. 2006.
- [29] W. Melvin and J. Guerci, “Knowledge-aided signal processing: A new paradigm for radar and other advanced sensors,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 3, pp. 983–996, July 2006.
- [30] M. Akcakaya, S. Sen, and A. Nehorai, “A novel data-driven learning method for radar target detection in nonstationary environments,” *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 762–766, May 2016.
- [31] M. Pollak, “Optimal detection of a change in distribution,” *The Annals of Statistics*, pp. 206–227, 1985.
- [32] G. V. Moustakides *et al.*, “Optimal stopping times for detecting changes in distributions,” *The Annals of Statistics*, vol. 14, no. 4, pp. 1379–1387, 1986.
- [33] M. Kelsey, S. Sen, Y. Xiang, A. Nehorai, and M. Akcakaya, “Sparse recovery for clutter identification in radar measurements,” in *Proc. SPIE*, vol. 1021106, May 2017, pp. 1–10.
- [34] H. Wang, Y. Xiang, E. Dagois, M. Kelsey, S. Sen, A. Nehorai, and M. Akcakaya, “Clutter identification based on kernel density estimation and sparse recovery,” in *Proc. SPIE*, vol. 10658, 2018, pp. 1–10.
- [35] Y. Xiang, M. Kelsey, H. Wang, S. Sen, M. Akcakaya, and A. Nehorai, “A comparison of cognitive approaches for clutter-distribution identification in nonstationary environments,” in *IEEE Radar Conference*, April 2018, pp. 0467–0472.
- [36] G. Lorden *et al.*, “Procedures for reacting to a change in distribution,” *The Annals of Mathematical Statistics*, vol. 42, no. 6, pp. 1897–1908, 1971.
- [37] V. V. Veeravalli and T. Banerjee, “Quickest change detection,” in *Academic Press Library in Signal Processing*. Elsevier, 2014, vol. 3, pp. 209–255.

- [38] K. Krishnamoorthy, *Handbook of Statistical Distributions with Applications*. Chapman and Hall/CRC, 2006.
- [39] Y. Zhu, Y. Xiang, S. Sen, E. Dagois, A. Nehorai, and M. Akcakaya, “Clutter identification based on sparse recovery and L1-type probabilistic distance measures,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4772–4776.
- [40] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*. Chapman and Hall/CRC, 2011.
- [41] H. B. Mann and D. R. Whitney, “On a test of whether one of two random variables is stochastically larger than the other,” *The annals of mathematical statistics*, pp. 50–60, 1947.
- [42] S. Kay, “Optimal signal design for detection of Gaussian point targets in stationary Gaussian clutter/reverberation,” *IEEE journal of selected topics in signal processing*, vol. 1, no. 1, pp. 31–41, 2007.
- [43] —, “Waveform design for multistatic radar detection,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, 2009.
- [44] X.-b. Deng, C.-y. Qiu, Z. Cao, M. Morelande, and B. Moran, “Waveform design for enhanced detection of extended target in signal-dependent interference,” *IET Radar, Sonar & Navigation*, vol. 6, no. 1, pp. 30–38, 2012.
- [45] S. Sen and A. Nehorai, “Adaptive OFDM radar for target detection in multipath scenarios,” *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 78–90, 2011.
- [46] S. Sen, G. Tang, and A. Nehorai, “Multiobjective optimization of OFDM radar waveform for target detection,” *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 639–652, 2011.
- [47] M. Hurtado and A. Nehorai, “Polarimetric detection of targets in heavy inhomogeneous clutter,” *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1349–1361, 2008.
- [48] S. Sen, “Characterizations of PAPR-constrained radar waveforms for optimal target detection,” *IEEE Sensors Journal*, vol. 14, no. 5, pp. 1647–1654, 2014.
- [49] M. Hurtado, T. Zhao, and A. Nehorai, “Adaptive polarized waveform design for target tracking based on sequential Bayesian inference,” *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 1120–1133, 2008.
- [50] S. Sen and A. Nehorai, “OFDM MIMO radar with mutual-information waveform design for low-grazing angle tracking,” *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3152–3162, 2010.

- [51] S.-M. Hong, R. J. Evans, and H.-S. Shin, “Control of waveforms and detection thresholds for optimal target tracking in clutter,” in *Proceedings of the 39th IEEE Conference on Decision and Control (CDC)*, vol. 4, 2000, pp. 3906–3907.
- [52] —, “Optimization of waveform and detection threshold for range and range-rate tracking in clutter,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 17–33, 2005.
- [53] D. J. Kershaw and R. J. Evans, “Optimal waveform selection for tracking systems,” *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1536–1550, 1994.
- [54] —, “Waveform selective probabilistic data association,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1180–1188, 1997.
- [55] C. Rago, P. Willett, and Y. Bar-Shalom, “Detection-tracking performance with combined waveforms,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 2, pp. 612–624, 1998.
- [56] R. Niu, P. Willett, and Y. Bar-Shalom, “Tracking considerations in selection of radar waveform for range and range-rate measurements,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 467–487, 2002.
- [57] D. Cochran, S. Suvorova, S. D. Howard, and B. Moran, “Waveform libraries,” *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 12–21, 2009.
- [58] S. P. Sira, A. Papandreou-Suppappola, and D. Morrell, “Dynamic configuration of time-varying waveforms for agile sensing and tracking in clutter,” *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3207–3217, 2007.
- [59] J. Zhang, B. Manjunath, G. Maalouli, A. Papandreou-Suppappola, and D. Morrell, “Dynamic waveform design for target tracking using MIMO radar,” in *42nd Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2008, pp. 31–35.
- [60] C. O. Savage and B. Moran, “Waveform selection for maneuvering targets within an IMM framework,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, 2007.
- [61] M. Hurtado, J.-J. Xiao, and A. Nehorai, “Target estimation, detection, and tracking,” *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 42–52, 2009.
- [62] M. Hurtado and A. Nehorai, “Bat-inspired adaptive design of waveform and trajectory for radar,” in *42nd Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2008, pp. 36–40.

- [63] R. McAulay and J. Johnson, “Optimal mismatched filter design for radar ranging, detection, and resolution,” *IEEE Transactions on Information Theory*, vol. 17, no. 6, pp. 696–701, 1971.
- [64] S. P. Sira, Y. Li, A. Papandreou-Suppappola, D. Morrell, D. Cochran, and M. Rangaswamy, “Waveform-agile sensing for tracking,” *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 53–64, 2009.
- [65] P. Tichavsky, C. H. Muravchik, and A. Nehorai, “Posterior Cramér-Rao bounds for discrete-time nonlinear filtering,” *IEEE Transactions on signal processing*, vol. 46, no. 5, pp. 1386–1396, 1998.
- [66] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.
- [67] S. Suvorova, S. Howard, W. Moran, and R. Evans, “Waveform libraries for radar tracking applications: Maneuvering targets,” in *40th Annual Conference on Information Sciences and Systems (CISS)*, 2006, pp. 1424–1428.
- [68] S. Suvorova, D. Musicki, B. Moran, S. Howard, and B. La Scala, “Multi step ahead beam and waveform scheduling for tracking of manoeuvring targets in clutter,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, 2005, pp. v–889.
- [69] M. R. Bell, “Information theory and radar waveform design,” *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1578–1597, 1993.
- [70] N. H. Nguyen, K. Dogancay, and L. M. Davis, “Adaptive waveform selection for multi-static target tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 688–701, 2015.
- [71] D. Cochran, “Waveform-agile sensing: opportunities and challenges,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, 2005, pp. v–877.
- [72] K. Dogancay, “UAV path planning for passive emitter localization,” *IEEE Transactions on Aerospace and Electronic systems*, vol. 48, no. 2, pp. 1150–1166, 2012.
- [73] —, “Single-and multi-platform constrained sensor path optimization for angle-of-arrival target tracking,” in *18th European Signal Processing Conference (EUSIPCO)*, 2010, pp. 835–839.
- [74] Y. Oshman and P. Davidson, “Optimization of observer trajectories for bearings-only target localization,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 3, pp. 892–902, 1999.

- [75] K. Zhou and S. I. Roumeliotis, “Optimal motion strategies for range-only constrained multisensor target tracking,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1168–1185, 2008.
- [76] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte, “Information-theoretic coordinated control of multiple sensor platforms,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, 2003, pp. 1521–1526.
- [77] N. H. Nguyen and K. Doğançay, “Optimal sensor placement for Doppler shift target localization,” in *IEEE Radar Conference (RadarCon)*, 2015, pp. 1677–1682.
- [78] G. Zhang, S. Ferrari, and M. Qian, “An information roadmap method for robotic sensor path planning,” *Journal of Intelligent & Robotic Systems*, vol. 56, no. 1, pp. 69–98, 2009.
- [79] S. Joshi and S. Boyd, “Sensor selection via convex optimization,” *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.
- [80] M. Shamaiah, S. Banerjee, and H. Vikalo, “Greedy sensor selection: Leveraging submodularity,” in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 2572–2577.
- [81] X. Shen and P. K. Varshney, “Sensor selection based on generalized information gain for target tracking in large sensor networks,” *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 363–375, 2014.
- [82] A. S. Narykov, O. A. Krasnov, and A. Yarovoy, “Algorithm for resource management of multiple phased array radars for target tracking,” in *Proceedings of the 16th International Conference on Information Fusion (ISIF)*, 2013, pp. 1258–1264.
- [83] A. S. Narykov and A. Yarovoy, “Sensor selection algorithm for optimal management of the tracking capability in multisensor radar system,” in *2013 European Microwave Conference (EuMC)*, 2013, pp. 1811–1814.
- [84] M. Xie, W. Yi, and L. Kong, “Joint node selection and power allocation for multi-target tracking in decentralized radar networks,” in *19th International Conference on Information Fusion (FUSION)*, 2016, pp. 45–52.
- [85] P. Chavali and A. Nehorai, “Scheduling and power allocation in a cognitive radar network for multiple-target tracking,” *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 715–729, 2012.
- [86] H. Godrich, A. Petropulu, and H. V. Poor, “Cluster allocation schemes for target tracking in multiple radar architecture,” in *Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2011, pp. 863–867.

- [87] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola, “Scheduling multiple sensors using particle filters in target tracking,” in *IEEE Workshop on Statistical Signal Processing (SSP)*, 2003, pp. 549–552.
- [88] N. Cao, S. Choi, E. Masazade, and P. K. Varshney, “Sensor selection for target tracking in wireless sensor networks with uncertainty,” *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5191–5204, 2016.
- [89] N. H. Nguyen, K. Dogancay, and L. M. Davis, “Joint transmitter waveform and receiver path optimization for target tracking by multistatic radar system,” in *IEEE Workshop on Statistical Signal Processing (SSP)*, 2014, pp. 444–447.
- [90] B. Manjunath, J. J. Zhang, A. Papandreou-Suppappola, and D. Morrell, “Sensor scheduling with waveform design for dynamic target tracking using MIMO radar,” in *Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2009, pp. 141–145.
- [91] N. H. Nguyen, K. Doğançay, and L. M. Davis, “Adaptive waveform and Cartesian estimate selection for multistatic target tracking,” *Signal Processing*, vol. 111, pp. 13–25, 2015.
- [92] Y. Xiang, M. Akcakaya, S. Sen, D. Erdogmus, and A. Nehorai, “Target tracking via recursive bayesian state estimation in radar networks,” in *51st Asilomar Conference on Signals, Systems, and Computers*, Oct 2017, pp. 880–884.
- [93] J. Geweke and H. Tanizaki, “Bayesian estimation of state-space models using the Metropolis-Hastings algorithm within Gibbs sampling,” *Computational Statistics and Data Analysis*, vol. 37, no. 2, pp. 151 – 170, 2001.
- [94] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, April 1993.
- [95] A. Ozturk, “A general algorithm for univariate and multivariate goodness of fit tests based on graphical representation,” *Communications in Statistics - Theory and Methods*, vol. 20, no. 10, pp. 3111–3137, 1991.

Appendix A

Proof of $\hat{t}_{\text{AMF}} \sim F(2, 2LN)$ Under the Null Hypothesis

First, from (2.30) we have

$$\hat{t}_{\text{AMF}} = \frac{|\mathbf{y}_0^{(k)H} \mathbf{1}|^2}{\hat{\sigma}^2 \mathbf{1}^H \mathbf{1}}. \quad (\text{A.1})$$

where

$$\hat{\sigma}^2 = \frac{1}{LN} \sum_{l=1}^L \sum_{i=1}^N |y_{cli}^{(k)}|^2. \quad (\text{A.2})$$

Note that $y_{cli}^{(k)}$ is the i_{th} entry of the secondary data $\mathbf{y}_{cl}^{(k)}$. Since the clutter samples are assumed to be complex Gaussian distributed in AMF, we have

$$y_{cli}^{(k)} \sim \mathcal{CN}(0, \sigma^2). \quad (\text{A.3})$$

For the complex Gaussian distribution defined in (2.22), the real part and the imaginary part of $y_{cli}^{(k)}$ are independent and Gaussian. Thus, we have

$$\frac{|y_{cli}^{(k)}|^2}{\sigma^2/2} \sim \chi_2^2. \quad (\text{A.4})$$

In Section 2.6, we assume that the entries of the clutter vector in each range cell are independent. In addition, clutter vectors in different range cells are commonly assumed to be independent. Thus, we have

$$LN \frac{\hat{\sigma}^2}{\sigma^2/2} \sim \chi_{2LN}^2. \quad (\text{A.5})$$

Further, because $\mathbf{y}_0^{(k)H} \mathbf{1} \sim \mathcal{CN}(0, N\sigma^2)$, we have

$$\frac{|\mathbf{y}_0^{(k)H} \mathbf{1}|^2}{N\sigma^2/2} \sim \chi_2^2. \quad (\text{A.6})$$

Note that $\mathbf{1}^H \mathbf{1} = N$, thus, we have

$$\hat{t}_{\text{AMF}} \sim \frac{\chi_2^2/2}{\chi_{2LN}^2/2LN}, \quad (\text{A.7})$$

where the right-hand-side coincides with definition of an F -distribution, because these two χ^2 -distributions are independent due to the independence of the data in the cell under test and the secondary data.

Appendix B

Comparison of Ozturk Method and Our Proposed Method

In this section, we first introduce the Ozturk algorithm, and then we show the comparison results of our proposed method and Ozturk method.³

B.1 Ozturk Algorithm

The Ozturk algorithm provides a graphical distance measurement between the sampled data and distributions in the dictionary [17,20,95]. This algorithm could be applied for univariate and multivariate cases, by normalizing the ordered samples and then converting the ordered samples into points in the two-dimensional plane. The distance between the endpoint of the sampled data and that of a distribution in the dictionary presents the fitness of the sampled data with the specific distribution [20].

³This chapter is based on Y. Xiang, M. Kelsey, H. Wang, S. Sen, M. Akcakaya, and A. Nehorai, “A comparison of cognitive approaches for clutter-distribution identification in nonstationary environments,” *IEEE Radar Conference (RadarConf18)*, Oklahoma City, OK, April 23-27, 2018, pp.0467-0472. ©IEEE 2018.

The Ozturk algorithm is originally designed for the fitness test. Assume that X_1, X_2, \dots, X_n are randomly sampled from a distribution function $F(x)$. Then the ordered samples are written as $X_{1:n} \leq X_{2:n} \leq \dots \leq X_{n:n}$. Assume that a location-scale distribution $F_0((x-\mu)/\sigma)$ is the null distribution (reference distribution), where μ and σ are the location and scale parameters, respectively. Then we denote the expected order statistics from the standard null distribution as $m_{1:n}, m_{2:n}, \dots, m_{n:n}$. The standardized i_{th} sample order statistic could be written as $Y_{i:n} = |X_{i:n} - \bar{X}|/S$, where \bar{X} and S are the sample mean and standard deviation, respectively. Then the two-dimensional location of i_{th} point corresponding to the i_{th} sample order statistic could be defined by $Q_{i:n} = (U_{i:n}, V_{i:n}), i = 1, 2, \dots, n$, where

$$U_{i:n} = \frac{1}{n} \sum_{j=1}^i \cos(\pi F_0(m_{j:n})) Y_{i:n}, \quad (\text{B.1})$$

and

$$V_{i:n} = \frac{1}{n} \sum_{j=1}^i \sin(\pi F_0(m_{j:n})) Y_{i:n}. \quad (\text{B.2})$$

The graph starts from the origin in the two-dimensional system, and each point $(U_{i:n}, V_{i:n})$ is plotted to form linked vectors. These vectors could reveal a certain pattern under the null hypothesis [95]. Further, if samples are drawn from the hypothesized distribution then it should create a pattern uniformly close to the expected linked vector pattern. In this way, a $(100(1-\alpha))$ confidence contour for the expected endpoint $(E(U_{n:n}), E(V_{n:n}))$ can be generated, which is able to test whether samples are obtained from the hypothesized distribution [95]. This fitness test method is extended to be a distribution identification algorithm in [20] by selecting the nearest neighbour of the sample endpoint $(U_{n:n}, V_{n:n})$ from the graphical dictionary generated by the expected endpoints $(E(U_{n:n}), E(V_{n:n}))$ of various

Algorithm 6: Ozturk algorithm

- 1: Obtain the ordered sample observations $X_{i:n}$
 - 2: Calculate the standard order statistics $Y_{i:n}$
 - 3: Calculate the statistics $U_{n:n}$ and $V_{n:n}$ and plot the endpoint $Q_{n:n} = (U_{n:n}, V_{n:n})$.
 - 4: Compare the sample endpoint $Q_{n:n} = (U_{n:n}, V_{n:n})$ with the expected endpoints $(E(U_{n:n}), E(V_{n:n}))$ generated by the existing distributions in the graphical dictionary, and find the nearest neighbouring distribution.
-

predefined distributions. For the given samples X_1, X_2, \dots, X_n , the Ozturk algorithm for the distribution identification could be summarized as in Algorithm 6.

As pointed out in [20], the statistic $Q_{n:n}$ is location and scale invariant. If the expected endpoints $(E(U_{n:n}), E(V_{n:n}))$ are plotted for different distributions, then any location-scale family of distributions could be represented as a single point, while distributions having shape parameters form a curve.

B.2 Simulation Results

In this section, we compare the performances of the proposed clutter identification method introduced in Section 2 with that of the Ozturk algorithm based technique. We discuss the drawbacks of the Ozturk algorithm based method and justify the advantages of the proposed sparse recovery based technique via simulations. In the numerical examples, we consider the following clutter distributions:

- Gaussian distribution: $y_G \sim \mathcal{N}(\mu_G, \sigma_G^2)$, i.e., y_G follows a Gaussian distribution with mean μ_G and variance σ_G^2 . Note that for the Gaussian distribution, μ_G is the location parameter and σ_G is the scale parameter.

- K -distribution: $y_K = |\sqrt{\tau}n|$, i.e., y_K follows a K -distribution when $\tau \sim \mathbf{Gamma}(k, \theta)$ [where k is the shape parameter and θ is the scale parameter] and $n \sim \mathcal{CN}(0, \sigma_n^2)$, i.e., complex Gaussian distribution.
- Weibull distribution: $y_{\text{Wbl}} \sim \mathbf{Wbl}(\alpha, \beta)$, i.e., y_{Wbl} follows a Weibull distribution with the shape parameter α and the scale parameter β .
- Log-normal distribution: $y_{\text{LN}} \sim \mathbf{LogN}(\mu_{\text{LN}}, \sigma_{\text{LN}}^2)$, i.e., y_{LN} follows a log-normal distribution, implying that $(\ln y_{\text{LN}} - \mu_{\text{LN}})/\sigma_{\text{LN}} \sim \mathcal{N}(0, 1)$.
- Student-t distribution: $y_{\text{St}} = \sqrt{\tau}w$, i.e., y_{St} follows a non-standardized Student-t distribution when $1/\tau \sim \mathbf{Gamma}(v, 1/v)$ and $w \sim \mathcal{N}(0, \sigma_w^2)$.

Based on these distributions, we first construct the dictionary D with l elements, where each element is pre-learned using N i.i.d. samples from a specific clutter distribution. For example, an element of dictionary in the Ozturk algorithm based method is an expected endpoint generated by a clutter pdf. Then, considering the standard Gaussian distribution as the reference distribution, i.e., considering F_0 as a standard Gaussian cdf, the nearest neighbouring distribution in the graphical dictionary is identified as the underlying clutter distribution, as suggested by [20]. Note that the expectations of the endpoints are computed by 10,000 Monte Carlo trials in the dictionary generation. In addition, usually the expected order statistics $m_{i:n}$ do not have closed-form expressions; therefore, we use 20,000 Monte Carlo runs to approximate them. For the our proposed method, an element of dictionary D is a pre-learned discretized pdf (estimated by KDE method), which is then normalized. The underlying clutter pdf is then identified by the proposed method with sparsity level $C = 1$.

For simplicity, we define a notation $\{l : \Delta : u\}$ as a set that collects real numbers starting from l to u with increment Δ . For instance, $\{1 : 0.5 : 3\} = \{1.0, 1.5, 2.0, 2.5, 3.0\}$.

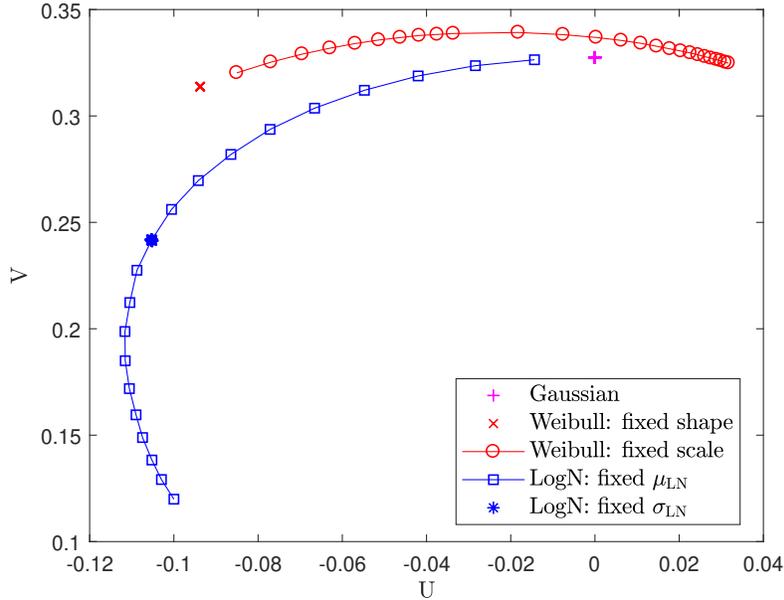


Figure B.1: Graphical dictionary generated by Weibull, Gaussian, and log-normal distributions.

B.2.1 Robustness to Location and Scale Parameters

As mentioned in [20], the endpoints generated by the Ozturk method are location and scale invariant for a family of distributions. However, it might not be a fascinating property to accurately identify the underlying clutter distributions from the same family, but having different parameters. To illustrate this issue, we plot in Fig. B.1 the graphical dictionary generated by the Gaussian, Weibull, and log-normal distributions. Specifically, in the dictionary we have:

1. Gaussian distributions with means $\mu_G \in \{-5 : 1 : 5\}$ and standard deviation parameters $\sigma_G \in \{1 : 1 : 5\}$,
2. Weibull distributions with a fixed shape parameter $\alpha = 1$, and scale parameters $\beta \in \{2 : 1 : 10\}$,

3. Weibull distributions with shape parameters $\alpha \in \{1.1 : 0.1 : 2\} \cup \{2.5 : 0.5 : 10\}$ and a scale parameter $\beta = 1$,
4. Log-normal distributions with a fixed $\mu_{\text{LN}} = 0$, and $\sigma_{\text{LN}} \in \{0.1 : 0.1 : 2\}$,
5. Log-normal distributions with $\mu_{\text{LN}} \in \{-2 : 0.1 : 2\}$ and a fixed $\sigma_{\text{LN}} = 1$. Note that $e^{\mu_{\text{LN}}}$ is the scale parameter for the log-normal distribution.

We observe from Fig. B.1 that the Gaussian distributions generate only one endpoint (irrespective of μ_{G} and σ_{G} values), since the Gaussian distribution is parameterized by only location and scale parameters. For Weibull and log-normal distribution, each of Case (3) and Case (5) (i.e., varied scale parameters with fixed other parameters) generates only one endpoint. However, varied shape parameters β for Weibull distributions (Case (2)) and varied σ_{LN} for log-normal distributions (Case (4)) can generate a curve for each one of them. Therefore, we conclude that the Ozturk algorithm is unable to directly identify the location and scale parameters for some specific types of distributions using the graphical dictionary.

We demonstrate this limitation of the Ozturk algorithm based clutter identification method based on the Weibull distributions. The dictionary is generated by Weibull distributions with shape parameters $\alpha \in \{1, 1.2\}$ and scale parameters $\beta \in \{1, 2\}$. Each endpoint in the dictionary is learned by $N = 1,000$ samples. To test the identification performance, we randomly select a test pdf from the dictionary, which generates $N_{\text{t}} = 1,000$ samples, and then apply the Ozturk algorithm based method to identify the clutter distribution. We show the overall performance of the Ozturk algorithm based method in Table B.1, where each row tabulates the percentages of the distributions in the dictionary that are selected by the identification method for a given true clutter distribution. The notation (T) indicates the true parameters, while (S) indicates the parameters that are selected.

Table B.1: Ozturk method performance on Weibull distributions

	$\alpha = 1,$ $\beta = 1$ (S)	$\alpha = 1,$ $\beta = 2$ (S)	$\alpha = 1.2,$ $\beta = 1$ (S)	$\alpha = 1.2,$ $\beta = 2$ (S)
$\alpha = 1, \beta = 1$ (T)	44.29%	49.80%	1.29%	4.63%
$\alpha = 1, \beta = 2$ (T)	46.74%	47.80%	1.23%	4.23%
$\alpha = 1.2,$ $\beta = 1$ (T)	2.12%	2.78%	45.73%	49.27%
$\alpha = 1.2,$ $\beta = 2$ (T)	2.39%	3.51%	44.71%	49.38%

Table B.2: Proposed method performance on Weibull distributions

	$\alpha = 1,$ $\beta = 1$ (S)	$\alpha = 1,$ $\beta = 2$ (S)	$\alpha = 1.2,$ $\beta = 1$ (S)	$\alpha = 1.2,$ $\beta = 2$ (S)
$\alpha = 1, \beta = 1$ (T)	96.70%	0.00%	3.30%	0.00%
$\alpha = 1, \beta = 2$ (T)	0.00%	98.50%	0.00%	1.50%
$\alpha = 1.2,$ $\beta = 1$ (T)	0.08%	0.00%	99.92%	0.00%
$\alpha = 1.2,$ $\beta = 2$ (T)	0.00%	0.24%	0.00%	99.76%

From Table B.1, we notice that the Weibull distributions with different shape parameters α are correctly identified by the Ozturk algorithm based method. For example, the distribution with $\alpha = 1, \beta = 1$ (T) is identified with high probability (i.e., 44.29% + 49.80%) to be a distribution with $\alpha = 1$, regardless of the scale parameter β . However, the Ozturk algorithm based method fails in distinguishing the Weibull distributions with different scale parameters β . For example, for $\alpha = 1, \beta = 1$ (T) case, $\alpha = 1, \beta = 1$ (S) is not selected with overwhelming probability compared to the $\alpha = 1, \beta = 2$ (S).

Then, we demonstrate that the our method is capable of identifying clutter distributions with the different scale parameters, which is beyond the capability of the Ozturk algorithm based

method. We use the same shape and scale parameters that were used in the simulations with Weibull distributions for Table B.1, along with the dictionary sample size $N = 1,000$, and test sample size $N_t = 1,000$. The performance of proposed method is given in Table B.2. We clearly observe that each distribution is correctly identified by the proposed method with high probability ($> 96\%$). Compared to Table B.1, the proposed method immediately shows a superiority in distinguishing the distributions with different scale parameters. For example, in Table B.2, the proposed method identify $\alpha = 1.2, \beta = 1$ (T) as $\alpha = 1.2, \beta = 1$ (S) with 99.92% accuracy, while in Table B.1, $\alpha = 1.2, \beta = 1$ (T) is incorrectly recognized as $\alpha = 1.2, \beta = 2$ (S) with 49.27% probability due to the scale-invariant property of the Ozturk algorithm.

Furthermore, for Ozutrck method, we present the results of a similar clutter identification case based on the log-normal distributions in Table B.3. For this simulation, we also consider the dictionary sample size $N_t = 1,000$ and test sample size $N = 1,000$. Note that $e^{\mu_{LN}}$ is the scale parameter for the log-normal distribution. We observe that, similar to the Weibull distributions, the Ozturk algorithm based method fails to accurately identify log-normal distributions with distinct scale parameter values.

We also test the performance of the proposed method on the log-normal distributions using the same simulation setup as that in Table B.3, and the resulting identification performances are shown in Table B.4. As before, by comparing Table B.4 and Table B.3, we notice the capability that the proposed method can easily distinguish the log-normal pdfs with different μ_{LN} and σ_{LN} , while Ozturk algorithm based method cannot distinctively identify pdfs with the same σ_{LN} but different μ_{LN} values.

Table B.3: Ozturk method performance on Log-normal distributions

	$\mu_{\text{LN}} = 0,$ $\sigma_{\text{LN}} = 0.4$ (S)	$\mu_{\text{LN}} = -1,$ $\sigma_{\text{LN}} = 0.4$ (S)	$\mu_{\text{LN}} = 0,$ $\sigma_{\text{LN}} = 0.5$ (S)	$\mu_{\text{LN}} = -1,$ $\sigma_{\text{LN}} = 0.5$ (S)
$\mu_{\text{LN}} = 0,$ $\sigma_{\text{LN}} = 0.4$ (T)	37.85%	49.03%	0.81%	12.31%
$\mu_{\text{LN}} = -1,$ $\sigma_{\text{LN}} = 0.4$ (T)	35.07%	51.09%	0.75%	13.09%
$\mu_{\text{LN}} = 0,$ $\sigma_{\text{LN}} = 0.5$ (T)	14.91%	0.74%	42.96%	41.40%
$\mu_{\text{LN}} = -1,$ $\sigma_{\text{LN}} = 0.5$ (T)	14.18%	0.58%	45.84%	39.39%

Table B.4: Proposed method performance on Log-normal distributions

	$\mu_{\text{LN}} = 0,$ $\sigma_{\text{LN}} = 0.4$ (S)	$\mu_{\text{LN}} = -1,$ $\sigma_{\text{LN}} = 0.4$ (S)	$\mu_{\text{LN}} = 0,$ $\sigma_{\text{LN}} = 0.5$ (S)	$\mu_{\text{LN}} = -1,$ $\sigma_{\text{LN}} = 0.5$ (S)
$\mu_{\text{LN}} = 0,$ $\sigma_{\text{LN}} = 0.4$ (T)	99.80%	0.00%	0.20%	0.00%
$\mu_{\text{LN}} = -1,$ $\sigma_{\text{LN}} = 0.4$ (T)	0.00%	99.92%	0.00%	0.08%
$\mu_{\text{LN}} = 0,$ $\sigma_{\text{LN}} = 0.5$ (T)	0.04%	0.00%	99.96%	0.00%
$\mu_{\text{LN}} = -1,$ $\sigma_{\text{LN}} = 0.5$ (T)	0.00%	0.00%	0.00%	100.00%

B.2.2 Performance Comparison between Our Method and Ozturk Method

In this section, we comprehensively compare the proposed clutter identification technique with the Ozturk algorithm based method using a dictionary that includes the K , Weibull, log-normal, and Student-t distributions. Specifically, we consider

1. K -distributions with fixed $\sigma_n = 1$, $\theta \in \{1, 10\}$, and $k \in \{0.1 : 0.5 : 3.6\} \cup \{4 : 5 : 24\}$,

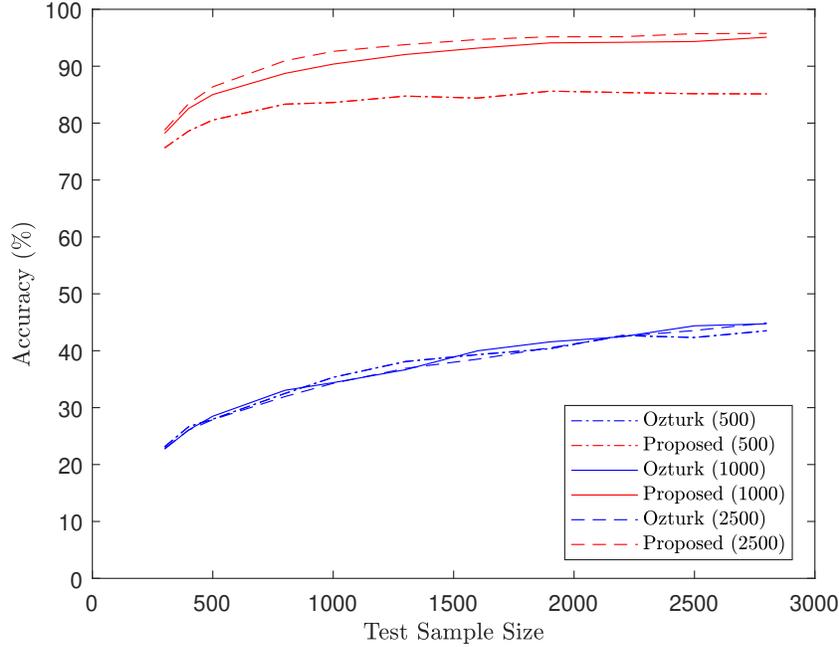


Figure B.2: Comparison of proposed method and Ozturk method for clutter distribution identification.

2. Weibull distributions with shape parameters $\alpha \in \{0.5 : 0.5 : 3\} \cup \{4 : 4 : 20\}$, and scale parameters $\beta \in \{1, 2\}$,
3. Log-normal distributions with $\mu_{LN} = 0$, and $\sigma_{LN} \in \{0.05 : 0.2 : 0.85\} \cup \{1 : 0.5 : 2\}$,
4. Student-t distributions with $\sigma_w = 1$, and $v \in \{0.5 : 0.5 : 1.5\} \cup \{2 : 3 : 8\}$.

In addition, we consider three dictionary sample sizes, $N = 500, 1000$, and 2500 . For each of them, the test sample sizes N_t vary from 300 to 2800 . To test the identification performance, we randomly select test pdfs from the dictionary, and apply the proposed method and Ozturk based method to identify them. The resulting performance comparison is shown in Fig. B.2. In general, we notice from Fig. B.2 that the proposed method for clutter identification significantly outperforms the Ozturk algorithm based method. It is due to the fact that the

deficiencies of the Ozturk algorithm, such as the location-scale invariant property, hinder its identification performance. In addition, we observe that both methods have improved performances with more test samples. Also, for proposed identification technique, the larger the dictionary sample size, the better is the overall identification accuracy. However, for the Ozturk algorithm based method, increasing the sample size in the dictionary generation does not substantially change the overall identification performance.

Appendix C

Derivation of (3.39) and (3.40)

C.1 Derivation of (3.39)

In this section, our goal is to derive

$$H_{\text{cross}} \triangleq -\mathbb{E}_{p(\mathbf{x}_k|\mathbf{z}_{1:k-1})} \log p(\mathbf{x}_k|\mathbf{z}_{1:k}; \mathcal{D}(\mathcal{U}_k)). \quad (\text{C.1})$$

From the definition of the extended Kalman filter, we have $\mathbf{x}_k|\mathbf{z}_{1:k} \sim N(\hat{\mathbf{x}}_{k|k}, P_{k|k})$. Thus,

$$-\log p(\mathbf{x}_k|\mathbf{z}_{1:k}; \mathcal{D}(\mathcal{U}_k)) = \frac{1}{2} \log(2\pi)^4 + \frac{1}{2} \log |P_{k|k}| + \frac{1}{2} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^\top P_{k|k}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}). \quad (\text{C.2})$$

Here,

$$\begin{aligned} & (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^\top P_{k|k}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}) \\ &= (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} + \hat{\mathbf{x}}_{k|k-1} - \hat{\mathbf{x}}_{k|k})^\top P_{k|k}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} + \hat{\mathbf{x}}_{k|k-1} - \hat{\mathbf{x}}_{k|k}) \\ &= (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^\top P_{k|k}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) + 2(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^\top P_{k|k}^{-1} (\hat{\mathbf{x}}_{k|k-1} - \hat{\mathbf{x}}_{k|k}) \\ & \quad + (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1})^\top P_{k|k}^{-1} (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1}). \end{aligned} \quad (\text{C.3})$$

For the first term in (C.3), we have

$$\begin{aligned}
& \mathbb{E}_{p(\mathbf{x}_k | \mathbf{z}_{1:k-1})} \left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^\top P_{k|k}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) \right] \\
&= \mathbb{E}_{p(\mathbf{x}_k | \mathbf{z}_{1:k-1})} \text{tr} \left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^\top P_{k|k}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) \right] \\
&= \mathbb{E}_{p(\mathbf{x}_k | \mathbf{z}_{1:k-1})} \text{tr} \left[P_{k|k}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^\top \right] \\
&= \text{tr} \left[P_{k|k}^{-1} P_{k|k-1} \right], \tag{C.4}
\end{aligned}$$

where $\text{tr}(AB) = \text{tr}(BA)$ holds for any two matrices A and B is applied. For the second term in (C.3), we have

$$\begin{aligned}
& \mathbb{E}_{p(\mathbf{x}_k | \mathbf{z}_{1:k-1})} \left[2(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^\top P_{k|k}^{-1} (\hat{\mathbf{x}}_{k|k-1} - \hat{\mathbf{x}}_{k|k}) \right] \\
&= 2 \underbrace{\left[\mathbb{E}_{p(\mathbf{x}_k | \mathbf{z}_{1:k-1})} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) \right]^\top}_{=0} P_{k|k}^{-1} (\hat{\mathbf{x}}_{k|k-1} - \hat{\mathbf{x}}_{k|k}) \\
&= 0. \tag{C.5}
\end{aligned}$$

Combine the above equations, we have the desired expression,

$$\begin{aligned}
H_{\text{cross}} &= - \mathbb{E}_{p(\mathbf{x}_k | \mathbf{z}_{1:k-1})} \log p(\mathbf{x}_k | \mathbf{z}_{1:k}; \mathcal{D}(\mathcal{U}_k)) \\
&= \frac{1}{2} \log(2\pi)^4 + \frac{1}{2} \log |P_{k|k}| + \frac{1}{2} \text{tr}(P_{k|k}^{-1} P_{k|k-1}) + \frac{1}{2} (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1})^\top P_{k|k}^{-1} (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1}). \tag{C.6}
\end{aligned}$$

C.2 Derivation of (3.40)

In this section, we derive (3.40), i.e.,

$$f(\mathcal{D}(\mathcal{U}_k)) \triangleq \mathbb{E}_{p(\mathbf{z}_k | \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}; \mathcal{D}(\mathcal{U}_k))} [H_{\text{cross}}], \quad (\text{C.7})$$

where H_{cross} is given in (C.6) and (3.39).

Focus on the fourth term in (C.6) and we denote it as $H_{\text{cross},4}$. Based on (3.25), we have

$$\begin{aligned} H_{\text{cross},4} &\triangleq \frac{1}{2} (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1})^\top P_{k|k}^{-1} (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1}) \\ &= \frac{1}{2} (\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}; \mathcal{D}(\mathcal{U}_k)))^\top K_k^\top P_{k|k}^{-1} K_k (\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}; \mathcal{D}(\mathcal{U}_k))). \end{aligned} \quad (\text{C.8})$$

According to (3.33), we have

$$\begin{aligned} &\mathbb{E}_{p(\mathbf{z}_k | \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}; \mathcal{D}(\mathcal{U}_k))} [H_{\text{cross},4}] \\ &\approx \sum_j \frac{q(L(\tilde{\mathbf{x}}_k^j) | \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1})}{C} \int p(\mathbf{z}_k | \tilde{\mathbf{x}}_k^j; \mathcal{D}(\mathcal{U}_k)) H_{\text{cross},4} d\mathbf{z}_k \\ &= \sum_j \frac{q(L(\tilde{\mathbf{x}}_k^j) | \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1})}{2C} \mathbb{E}_{p(\mathbf{z}_k | \tilde{\mathbf{x}}_k^j; \mathcal{D}(\mathcal{U}_k))} [2H_{\text{cross},4}]. \end{aligned} \quad (\text{C.9})$$

Reformulate $2H_{\text{cross},4}$ as follow,

$$\begin{aligned} 2H_{\text{cross},4} &= (\mathbf{z}_k - \boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)))^\top K_k^\top P_{k|k}^{-1} K_k (\mathbf{z}_k - \boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k))) \\ &\quad + 2 (\mathbf{z}_k - \boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)))^\top K_k^\top P_{k|k}^{-1} K_k (\boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1})) \\ &\quad + (\boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}))^\top K_k^\top P_{k|k}^{-1} K_k (\boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1})), \end{aligned} \quad (\text{C.10})$$

where $\boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k))$ is defined in (3.37) and $\boldsymbol{h}_k(\hat{\boldsymbol{x}}_{k|k-1}; \mathcal{D}(\mathcal{U}_k))$ is shortened to $\boldsymbol{h}_k(\hat{\boldsymbol{x}}_{k|k-1})$.

Note that according to definition in (3.37) and (3.38), we have $\boldsymbol{z}_k | \tilde{\boldsymbol{x}}_k^j \sim \mathcal{N}(\boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k)), R_k)$, thus,

$$\begin{aligned}
& \mathbb{E}_{p(\boldsymbol{z}_k | \tilde{\boldsymbol{x}}_k^j; \mathcal{D}(\mathcal{U}_k))} \left[(\boldsymbol{z}_k - \boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k)))^\top K_k^\top P_{k|k}^{-1} K_k (\boldsymbol{z}_k - \boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k))) \right] \\
&= \mathbb{E}_{p(\boldsymbol{z}_k | \tilde{\boldsymbol{x}}_k^j; \mathcal{D}(\mathcal{U}_k))} \text{tr} \left[(\boldsymbol{z}_k - \boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k)))^\top K_k^\top P_{k|k}^{-1} K_k (\boldsymbol{z}_k - \boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k))) \right] \\
&= \mathbb{E}_{p(\boldsymbol{z}_k | \tilde{\boldsymbol{x}}_k^j; \mathcal{D}(\mathcal{U}_k))} \text{tr} \left[K_k^\top P_{k|k}^{-1} K_k (\boldsymbol{z}_k - \boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k))) (\boldsymbol{z}_k - \boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k)))^\top \right] \\
&= \text{tr}(K_k^\top P_{k|k}^{-1} K_k R_k), \tag{C.11}
\end{aligned}$$

and

$$\mathbb{E}_{p(\boldsymbol{z}_k | \tilde{\boldsymbol{x}}_k^j; \mathcal{D}(\mathcal{U}_k))} (\boldsymbol{z}_k - \boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k)))^\top K_k^\top P_{k|k}^{-1} K_k (\boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) - \boldsymbol{h}_k(\hat{\boldsymbol{x}}_{k|k-1})) = 0. \tag{C.12}$$

Substitute (C.10), (C.11), and (C.12) into (C.9), we have

$$\begin{aligned}
& \mathbb{E}_{p(\boldsymbol{z}_k | \boldsymbol{x}_{k-1} = \hat{\boldsymbol{x}}_{k-1|k-1}; \mathcal{D}(\mathcal{U}_k))} [H_{\text{cross},4}] \\
& \approx \frac{1}{2} \text{tr}(K_k^\top P_{k|k}^{-1} K_k R_k) + \frac{1}{2C} \sum_j [q(L(\tilde{\boldsymbol{x}}_k^j) | \boldsymbol{x}_{k-1} = \hat{\boldsymbol{x}}_{k-1|k-1}) \times \\
& (\boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) - \boldsymbol{h}_k(\hat{\boldsymbol{x}}_{k|k-1}))^\top K_k^\top P_{k|k}^{-1} K_k (\boldsymbol{\mu}(\tilde{\boldsymbol{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) - \boldsymbol{h}_k(\hat{\boldsymbol{x}}_{k|k-1}))]. \tag{C.13}
\end{aligned}$$

Note that here we use the fact that C is the normalizing constant such that the summation of $q(L(\tilde{\boldsymbol{x}}_k^j) | \boldsymbol{x}_{k-1} = \hat{\boldsymbol{x}}_{k-1|k-1})$ is equal to 1.

Based on (C.13) and (C.6), we have the desired expression,

$$\begin{aligned}
f(\mathcal{D}(\mathcal{U}_k)) &= \mathbb{E}_{p(\mathbf{z}_k | \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}; \mathcal{D}(\mathcal{U}_k))} [H_{\text{cross}}] \\
&\approx \frac{1}{2} \log(2\pi)^4 + \frac{1}{2} \log |P_{k|k}| + \frac{1}{2} \text{tr}(P_{k|k}^{-1} P_{k|k-1}) + \frac{1}{2} \text{tr}(K_k^\top P_{k|k}^{-1} K_k R_k) \\
&\quad + \frac{1}{2C} \sum_j [q(L(\tilde{\mathbf{x}}_k^j) | \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}) \times \\
&\quad (\boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}))^\top K_k^\top P_{k|k}^{-1} K_k (\boldsymbol{\mu}(\tilde{\mathbf{x}}_k^j, \mathcal{D}(\mathcal{U}_k)) - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}))]. \quad (\text{C.14})
\end{aligned}$$

Vita

Yijian Xiang

Degrees

Ph.D., Electrical Engineering, Washington University in St. Louis, Missouri, USA, August 2020

M.S., Electrical Engineering, Washington University in St. Louis, Missouri, USA, May 2020

B.S., Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China, June 2014

Professional Memberships

The Institute of Electrical and Electronics Engineers (IEEE)
IEEE Signal Processing Society

Publications

Journal Publications:

Y. Xiang, M. Akcakaya, S. Sen, Y. Zhu, E. Dagois, and A. Nehorai, “Target detection via cognitive radars using change-point detection, learning, and adaptation,” in revision for *Circuits, Systems, and Signal Processing*.

F. Xi, **Y. Xiang**, S. Chen, and A. Nehorai, “Gridless parameter estimation for one-bit MIMO radar with time-varying thresholds,” *IEEE Trans. on Signal Processing*, Vol. 68, No. 1, pp. 1048-1063, 2020.

Y. Huang, G. Liao, **Y. Xiang**, L. Zhang, J. Li, and A. Nehorai, “Low-rank approximation via generalized reweighted iterative nuclear and Frobenius norms,” *IEEE Trans. on Image Processing*, Vol. 29, No. 1, pp. 2244-2257, Dec. 2019.

Y. Huang, G. Liao, L. Zhang, **Y. Xiang**, J. Li, and A. Nehorai, “Efficient narrowband RFI mitigation algorithms for SAR systems with reweighted tensor structures,” *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 57, No. 11, pp. 9396-9409, Nov. 2019.

Y. Huang, G. Liao, **Y. Xiang**, Z. Zhang, J. Li, and A. Nehorai, “Reweighted nuclear norm and reweighted Frobenius norm minimizations for narrowband RFI suppression on SAR system,” *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 57, No. 8, pp. 5949-5962, Aug. 2019.

Y. Xiang, M. Akcakaya, S. Sen, D. Erdogmus, and A. Nehorai, “Target tracking via recursive Bayesian state estimation in cognitive radar networks,” *Signal Processing*, Vol. 155, pp. 157-169, Feb. 2019.

Y. Huang, G. Liao, Z. Zhang, **Y. Xiang**, J. Li, and A. Nehorai, “Fast narrowband RFI suppression algorithms for SAR systems via matrix-factorization techniques,” *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 57, No. 1, pp. 250-262, Jan. 2019.

Y. Huang, G. Liao, Z. Zhang, **Y. Xiang**, J. Li, and A. Nehorai, “SAR automatic target recognition using joint low-rank and sparse multi-view denoising,” *IEEE Geoscience and Remote Sensing Letters*, Vol. 15, No. 10, pp. 1570-1574, Oct. 2018.

L. Fang, **Y. Xiang**, N. Cheung, and F. Wu, “Estimation of virtual view synthesis distortion toward virtual view position,” *IEEE Trans. on Image Processing*, Vol. 25, No. 5, pp. 1961-1976, March 2016.

Conference Publications:

Y. Zhu, **Y. Xiang**, S. Sen, E. Dagois, A. Nehorai, and M. Akcakaya, “Clutter Identification Based on Sparse Recovery and L1-Type Probabilistic Distance Measures,” *Proc. 45th IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 4772-4776.

Z. Zhang, **Y. Xiang**, L. Wu, B. Xue, and A. Nehorai, “KerGM: Kernelized graph matching,” *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, CA, Dec. 8-14, 2019.

Y. Zhu, **Y. Xiang**, S. Sen, E. Sejdic, A. Nehorai, M Akcakaya, “Clutter identification based on sparse recovery with dynamically changing

dictionary sizes for cognitive radar,” *SPIE Defense + Commercial Sensing*, Baltimore, MD, May 13, 2019.

Z. Zhang, M. Wang, **Y. Xiang**, Y. Huang, and A. Nehorai, “RetGK: Graph kernels based on return probabilities of random walks,” *Advances in Neural Information Processing Systems (NeurIPS)*, Montréal, CA, Dec. 3-8, 2018.

H. Wang, **Y. Xiang**, E. Dagois, M. Kelsey, S. Sen, A. Nehorai, and M. Akcakaya, “Clutter identification based on kernel density estimation and sparse recovery”, *SPIE Defense + Commercial Sensing*, Orlando, FL, May 14, 2018.

Y. Xiang, M. Kelsey, H. Wang, S. Sen, M. Akcakaya and A. Nehorai, “A comparison of cognitive approaches for clutter-distribution identification in nonstationary environments,” *IEEE Radar Conference (RadarConf18)*, Oklahoma City, OK, April 23-27, 2018.

J.Li, G.Liao, Y.Huang, J.Xu, **Y.Xiang**, A.Nehorai, “MIMO-stap based cognitive design of transmitted waveforms and receive filters for clutter suppression,” *IEEE Radar Conference (RadarConf18)*, Oklahoma City, OK, April 23-27, 2018.

Y. Xiang, M. Akcakaya, S. Sen, D. Erdogmus and A. Nehorai, “Target tracking via recursive Bayesian state estimation in radar networks,” *51st Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 29-Nov. 1, 2017.

M. Kelsey, S. Sen, **Y. Xiang**, A. Nehorai, and M. Akcakaya, “Sparse recovery for clutter identification in radar measurements,” *SPIE Commercial + Scientific Sensing and Imaging*, Anaheim, CA, May 5, 2017.

Z. Zhang, M. Wang, **Y. Xiang**, and A. Nehorai, “Geometry-adapted Gaussian random field regression,” *Proc. 42nd IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, New Orleans, LA, Mar. 5-9, 2017.

Y. Xiang, L. Fang, R. Li and N. Cheung, "Depth error induced virtual view synthesis distortion estimation for 3D video coding," *Data Compression Conference (DCC)*, Snowbird, UT, April 7-9, 2015.

Y. Xiang, N. Cheung, J. Zhang and L. Fang, "Analytical model for camera distance related 3D virtual view distortion estimation," *IEEE International Conference on Image Processing (ICIP)*, Paris, Oct. 27-30, 2014.

August 2020

Cognitive Radar Detection and Tracking, Xiang, Ph.D. 2020