Distributed Online Learning in Cognitive Radar Networks

William W. Howard

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

R. Michael Buehrer, Chair
Harpreet S. Dhillon
Brian D. Woerner
Eyvindur A. Palsson
Bharat Kunduri

November 29, 2023
Blacksburg, Virginia

Keywords: Cognitive Radar, Radar Networks, Reinforcement learning, Target Tracking

Distributed Online Learning in Cognitive Radar Networks

William W. Howard

(ABSTRACT)

Cognitive radar networks (CRNs) were first proposed in 2006 by Simon Haykin, shortly after the introduction of cognitive radar. In order for CRNs to benefit from many of the optimization techniques developed for cognitive radar, they must have some method of coordination and control. Both centralized and distributed architectures have been proposed, and both have drawbacks. This work addresses gaps in the literature by providing the first consideration of the problems that appear when typical cognitive radar tools are extended into networks. This work first examines the online learning techniques available to distributed CRNs, enabling optimal resource allocation without requiring a dedicated communication resource. While this problem has been addressed for single-node cognitive radar, we provide the first consideration of mutual interference in such networks. We go on to propose the first hybrid cognitive radar network structure which takes advantage of central feedback while maintaining the benefits of distributed networks. Then, we go on to investigate a novel problem of timely updating in CRNs, addressing questions of target update frequency and node updating methods. We draw from the Age of Information literature to propose Bellman-optimal solutions. Finally, we introduce the notion of mode control, and develop a way to select between active and passive target observation.

Distributed Online Learning in Cognitive Radar Networks

William W. Howard

(GENERAL AUDIENCE ABSTRACT)

Cognitive radar was inspired by biological models, where animals such as dolphins or bats use vocal pulses to form a model of their environment. As these animals seek after prey, they use information they observe to modify their vocal pulses. Cognitive radar networks are an extension of this model to a group of radar devices, which must work together cooperatively to detect and track targets. As the scene changes in time, the radar nodes in the cognitive radar network must change their operating parameters to continue performing well. This networked problem has issues not present in the single-node cognitive radar problem. In particular, as each node in the network changes operating parameters, it risks degrading the performance of the other nodes. In the contribution of this dissertation, we investigate the techniques that a cognitive radar network can use to avoid these cases of mutual performance degradation, and in particular, we investigate how this can be done without advance coordination between the nodes. In the second contribution, we go on to explore what performance improvements are available as central control is introduced. The third and fourth contributions investigate further efficiencies available to a cognitive radar network. The third contribution discusses how a resource-constrained network should communicate updates to a central aggregator. Lastly, the fourth contribution investigates additional estimation tools available to such a network, and how the network should choose between these modes.

Dedication

To my Grandpa, who was unable to complete high school due to the second world war, and who taught me the meaning of hard work, thank you for telling me to stay in school as long as I could. I think I did.

Acknowledgments

This dissertation would not be possible without the support of my family and friends First, I am forever grateful to my mother, father, brother, and extended family for their support through my life.

I am also grateful to my academic advisors for their mentorship through my undergraduate and graduate careers. I am particularly thankful for my Ph.D. advisor Dr. Mike Buehrer, who is an exceptional researcher and mentor. His guidance through my time at Virginia Tech has been invaluable. In undergrad, I received terrific mentorship from my advisor Dr. Matt Valenti and my department chair Dr. Brian Woerner, as well as from Dr. Drew Lowery. Thanks also to Dr. Tony Martone for being a frequent collaborator, and to Dr. Don Kellmel and Dr. Tim Miller for many excellent discussions which contributed to my early research. Lastly, I would like to thank the members of my committee for reviewing my work.

Next, to my friends I've had the pleasure of meeting during my time at Virginia tech. Thanks to Chris O'Lone for being a friend and mentor ever since my first week in the lab. To Charlie Thornton, Don-Roberts Emenonye, and Raghu Rao, I'm grateful for our time in the lab discussing anything and everything. Thanks also to Sam Shebert, Megan Moore, Tarun Cosik, and Alyse Jones, it was great getting to know you, and enjoy the rest of your time at VT!

I would also like to thank my friends from Morgantown and around the country. Thanks in particular to Adam, Addison, Bert, and Mandie for being there since time began. Thanks to Troy and Lenny for always being down for an adventure.

In closing, I would like to thank my patient, kind, and loving partner Riley, who has improved my life in countless ways, and to our coworkers Tycho, Dexter, and Daisy for their companionship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of cognitive radar [1] [2] [3] gained popularity recently[1], but has existed in various forms since the 1970's. Algorithms for adaptive radar [4] were developed by private companies and government researchers to improve radar function, but these techniques were not available in the open literature due to security classification and export control restrictions [5]. These approaches were not described as "cognitive" (a term which is defined later in this dissertation), but they considered the role of environmental feedback which is considered to be the hallmark of cognitive radar. Nevertheless, in the early 2000's, the open literature caught up and developed the modern concept of cognitive radar and cognitive radar networks [6] [7]. In particular, the advancements in materials science and high performance computing which enabled the implementation and huge growth of machine learning techniques (also proposed in 20th century) led directly to the re-emergence of adaptive/cognitive radar and opened the path to many exciting research thrusts [8].

This introduction first motivates our study with an overview of several fundamental concepts, followed by a survey of the literature. Then, an overview of the major thrusts of this dissertation is provided.

Researchers have primarily targeted two basic areas with cognitive radar algorithms: spectrum sharing and waveform optimization. Spectrum sharing is the process of cooperating with other devices to avoid causing harmful interference, and to avoid experiencing the same. Waveform optimization deals with modifying basic radar operating parameters (center frequency, pulse width, pulse shape, etc.) in response to changing channel conditions in order to more accurately track or detect targets. Both of these areas are well-suited to cognitive radar due to their sequential nature; as we will discuss, cognitive techniques are typically cyclic, oscillating between observing the scene and modifying parameters.

[1] It was specifically popularized in 2005/2006 by Simon Haykin.

1.1 Fundamental Concepts

1.1.1 Radar-Based Target Localization

History and Applications

At their core, radar (or RAdio Detection And Ranging) systems process the reflections from targets of emitted pulses of energy to determine the presence and position of these targets. This technology was first utilized soon after the discovery of radio waves to avoid collisions between ships at sea.

While radar has found much use in national defense, many civilian applications have been established, e.g. civil aviation, maritime navigation, earth observation, and weather prediction. Common consumer implementations of radar include automobile collision avoidance [9] and parking assistance [10], as well as in consumer cellular devices as a form of user interface [11].

Even with these civilian applications, the majority of radar systems were traditionally operated by governments. This has led to the majority of legacy spectrum allocations featuring broad dedicated radar bands. However, as civilian usage has climbed into the GHz range, dedicated radar allocations have been shrinking. This has driven a desire for *spectrum sharing* [12] technologies among regulatory agencies. Spectrum sharing techniques allow primary and secondary users to coexist in certain bands, with a goal of minimizing interference and maximizing spectrum utilization.

Radar systems have many modifiable operating parameters. Some of these are fundamental (e.g. operating frequency or bandwidth) and others are a consequence of design (e.g. pulse rate or waveform shape). Depending on the properties of the environment and target, different parameters may result in better or worse performance. Traditional radar implementations, however, often use preset fixed parameters. This makes sense in historic context; radar systems were highly complex and often needed to operate in harsh conditions, so it was beneficial to use fixed parameters that serve well enough in the expected operating conditions.

Modern radar systems are often implemented in software rather than hardware due to increasing technological complexity [13]. This development allows for *on-the-fly* parameter adjustments such as waveform selection or pulse agility. These systems are capable of adapting parameters very quickly, faster than human intervention would allow, necessitating the implementation of further software systems to control the radar. Broadly, this has inspired the development of cognitive systems to optimize radar performance. In section 1.1.1, the initial conceptualization of *cognitive radar* is described.

Cognitive radar has been shown to be a useful technique for online optimization of radar operating parameters. As the number of deployed devices (both military and civilian) has

increased, and the price of those devices has decreased, it has become more common to see networks of sensor devices be proposed in the literature. So, cognitive radar networks are a natural extension. While the primary motivation behind this concept is to improve the spatial coverage of a radar system, a network with multiple radar nodes provides many advantages as we will discuss in Section 1.1.2.

Radar Processing Techniques

Radar is performed by emitting a known waveform from an electromagnetic array and analyzing the received energy scattered from the environment. The ways in which the waveform is altered by the environment indicate the presence of targets. Broadly, the position of a target can be estimated using the delay a waveform experiences traveling from the transmitter to the target and then to the receiver, along with the angle of arrival of the reflection. Similarly, the velocity of targets can be estimated via the relative Doppler shift exhibited by the returned energy, along with the changing angle of arrival.

Radar systems are typically categorized as being either *monostatic* or *bistatic*. Monostatic systems feature a colocated transmitter and receiver (often the same device), while bistatic systems feature spatial separation between transmitter and receiver. Multistatic systems are those featuring more than one spatially separated receiver (i.e., more than one set of bistatic systems). In this work we focus on the more common monostatic systems. We refer to a single monostatic radar as a *node*. When we discuss networks of radars, we only consider networks of monostatic radars.

Radar Networks

Radar networks are increasingly of interest in the literature, due at least in part by a desire to replace large expensive devices with a collection of smaller, cheaper devices. Also, situations which benefit from one radar will almost always benefit from multiple radars, due primarily to increased spatial diversity [14, Ch. 2] [15].

Radar networks consist of multiple radar nodes [16] [17] [18]. Since we focus on monostatic radar, the radar nodes in this document are capable of functioning as independent monostatic radars. There are many reasons to use a network rather than a single radar. In the literature, networks of this type are often described as *radar sensor networks* (RSNs). A fundamental work on RSNs is found in [19] where the authors propose orthogonal center frequencies to mitigate interference between radar sensors. In this RSN, the received pulses from each node are processed at a central node which handles target tracking considerations. Central processing is necessary here to perform global fusion. This technique is shown to be effective, but assumes that the radar nodes operate in dedicated spectrum free from interference. In practice, dedicated radar bands are shrinking and future systems must be robust to outside interference.

In this type of resource optimization work, there is often an assumption that all channels are equivalent and thus can be statically assigned prior to sensing [19]. This is at odds with cognitive techniques used in cognitive radar [20] for spectrum sharing and performance optimization. In particular, in research and in the real world, channel conditions can and do vary over time, frequency, and space. It is often true for a network that the channel which provides the best performance to one node will be different from the channel which provides the best performance to another node, whether this is due to interference, target behavior, scatter from the environment, or other factors.

Other works consider how a RSN could self-organize. In this context, self-organization refers to the ad hoc assigning of functions or resources to specific parts of the network in pursuit of a goal. The works of [21] and [22] are concerned with adapting network topology, which defines the ways in which information flows in a network. These works are useful in ad hoc, low-cost, non-permanent sensor networks. In [23, Ch. 15], the authors describe the following monostatic radar network architectures:

1. **Centralized**, where all nodes transmit their measurements to a central fusion center, which fuses the measurements into tracks.
2. **Distributed**, where tracks are established within each node and subsequently fused, reducing the data transmitted over the network,
3. **Decentralized**, where no global fusion center exists.

Since we are interested in networks with a global fusion center, we will focus on centralized and distributed networks[2]. We will discuss in Section 1.1.2 the fact that while these network architectures are compatible with the definitions of cognitive radar networks, it is possible to have networks with distributed cognition which still perform information fusion in a centralized manner.

Cognitive Radar

Cognition is characterized by Haykin [6] as possession of ability to "learn from continuing interactions with the environment" to improve performance. Further, in the same work, Haykin goes on to describe three basic constituents of cognition in *radar*:

1. *Feedback*, from the radar receiver to the transmitter, facilitating intelligence.
2. *Learning*, specifically *reinforcement learning*, to adapt to novel environments.
3. *Information preservation*, to ensure efficient performance.

[2]We sometimes may use the term "decentralized" as well to refer to the distributed network architecture. We do not consider networks without global fusion.

Notably, none of this describes how these properties ought to be implemented, just that they are fundamental to a cognitive system. This is primarily because these properties are evident in biological systems, which serve as the major inspiration for cognitive radar (specifically the echolocation of bats and dolphins).

Feedback in radar systems is the process of extracting usable information from received energy to inform future actions - e.g., a target is detected, leading to future energy being directed towards it. Without feedback, in this example, the radar might fail to detect the target in the future.

Cognition in biological systems is often described in terms of the OODA loop: observe, orient, decide, act. Observations of the environment are made and then oriented in context to prior observations. From this information, decisions are made and acted upon. As the process iterates, experience is gained which (ideally) improves future performance. In short, this is the process of learning from experiences to enable adaptation to new environments.

Waveform selection [20] and design [24] are important aspects of radar system design and operation. Legacy radar systems are designed around a narrow space of possible waveforms due to filtering and DSP limitations, among others. In contrast, modern systems are often based on flexible FPGA architectures which are capable of high bandwidth sensing, faster filter switching, and other DSP pipeline improvements. These, along with other improvements (such as allowing more precise control of antenna beampatterns) helped to catalyze the development of cognitive radar. Instead of radar systems being limited to a narrow space of operational parameters, these advances in electronics and materials engineering enabled systems to perform efficiently across a more broad space. Cognitive radar is one technique which has been proposed to allow a radar system to navigate this more complex landscape.

Cognitive techniques for radar are well-established in the literature [25]. Techniques have been proposed to optimize waveform design [2] [20] [26], avoid interference [27] [28] [29], and improve target detection [3]. One common thread in all these techniques is the use of *online optimization*, most frequently reinforcement learning. Online learning is so important relative to offline optimization due to the lack of a priori knowledge of the environment. If a system is trained offline in a single instance of an environment or even in multiple instances, it may fail to generalize across a broader class. If instead a suitable online learning algorithm is selected, the system can exhibit superior performance. Section 1.1.3 of this document provides an introduction to concepts in online reinforcement learning.

1.1.2 Cognitive Radar Networks

Haykin later proposes [6] "two major cognitive contenders" to describe how cognition might be implemented in a radar network:

1. **Distributed Cognition**, where observations from individual nodes are combined at

a *fusion center*[3] but no feedback is provided to the nodes.

2. **Centralized Cognition**, where a *central coordinator*[4] is the only cognitive agent, collecting observations from each node and dictating future actions.

Here, Haykin suggests that distributed cognition is likely the superior option, due to "troublesome" delayed feedback from a central coordinator. The second major contribution of this document investigates the trade-off between these two styles, and proposes "hybrid cognition" as a balance of the two.

While the literature has contributed much towards cognitive radar, there have been relatively few considerations of cognitive radar *networks*. While many of the techniques developed for cognitive radar translate well to CRNs, there are several key points that differentiate CRNs from single-node cognitive radar or even RSNs.

1. Distributed cognitive radar has an internal contradiction. The nodes in a CRN must adaptively share resources in a distributed manner *while not degrading network performance*. Without centralized knowledge of the network, a single node may seek a new optimal action and in doing so inadvertently cause another node to suffer poorer performance. As will be discussed in Section 1.2, this challenge, while analogous to known problems with learning agents, has not been explicitly addressed in the literature. The first major contribution of this thesis draws from the multi-player multi-armed bandit literature, which has considered this distributed learning problem.

2. While Haykin claims that feedback from a central coordinator is necessarily delayed and therefore "troublesome", it is not obvious that this is the case. Further, while his proposed dichotomy in cognitive styles is useful to describe overall topologies, it is not clear that these are the *only* two options. The literature has not provided an exploration of this issue. In the second major contribution of this document, different CRN cognition styles are compared and contrasted, especially in the context of spectrum sharing problems.

3. Different nodes in the same CRN may have entirely different perspectives on the environment and may not even see the same targets. In these scenarios, it is not obvious how a distributed network should share a constrained communication resource to provide updates to the central coordinator. This is a problem that single-node cognitive radar will not encounter.

4. Previous models for cognitive radar do not consider the availability of multiple estimation tools at each node. For example, if targets in the environment are emitting signals,

[3] Fusion centers in this type of network are assumed to perform no decision functions; i.e., they simply combine measurements and provide data to operators.

[4] Central coordinators are assumed to perform the functions of a fusion center *as well as* performing some decision-making functions.

a node may be able to detect those signals and use them for target identification or even localization.

These differences result in the fact that CRNs are not exceedingly well-understood in the literature. *Aspects* of CRNs have been addressed, but as discussed above, these aspects can fail to grasp the complexity of CRNs. In Section 1.3.1, the motivations for the contributions of this document are discussed.

1.1.3 Reinforcement Learning

Reinforcement learning is the branch of machine learning that rewards learning agents for taking certain actions. In particular, reinforcement learning approaches are often used to solve problems consisting of sequential learning tasks, where an agent repetitively selects one of several actions and observes a reward for each. The goal of the agent is to maximize the *cumulative expected reward* for taking a sequence of actions. Learning problems of this variety are often analyzed by the *regret* of the agent's actions, which is the difference in cumulative reward between the actions chosen by the learning agent and the "optimal" actions, or those which provide the highest reward. The cumulative reward measures the total utility the learner achieves before a finite time horizon, and is dependent on the utility function. If the utility function is well matched to the optimization problem, then a high cumulative reward can be indicative of good performance. More technical discussion of reinforcement learning is provided in section 2.1

Reinforcement learning agents often encounter a trade-off between actions which *explore* the space of possible actions, and those which *exploit* the known rewards. Without adequately exploring the action space, an agent will be likely to miss actions which provide higher average rewards. At the same time, exploring too much may result in selecting too many actions with unfavorable rewards.

A subset of reinforcement learning techniques called *online learning* is concerned specifically with those tasks which must be optimized on-the-fly. The online learning tool primarily utilized in this document is known as the "multi-armed bandit", or MAB [30] [31]. MAB problems are those concerned with stateless processes - those that provide the same reward for a given action independently of the previously selected action. These models draw their motivation, and name, from casino slot machines. The "arms" of the bandit are the multiple actions, and the learning agent must decide which arm to "pull" in each of many time steps. Over time, the agent is motivated to select the action it believes will result in the highest payout. The MAB model is useful to cognitive radar due to its iterative, online nature.

A particular type of MAB which is applicable to the distributed radar problem is known as the multi-player multi-armed bandit, or MMAB [32]. In this relatively recently developed model, multiple agents must cooperate to draw the highest expected reward from the bandit, each selecting a single unique action in each time step. The concept of a "collision" in a

MMAB model refers to the event where two agents select the same action simultaneously. Depending on the exact framework, this can result in an equal reward to both agents, a reward to just one randomly selected agent, or no reward at all.

MMAB problems draw inspiration from problems in cognitive *radio* networks [33], where multiple devices seek to access the same spectrum resource to exchange information. While similar, it is important to note that the CRN problem has significant differences from cognitive radio networks:

1. CRNs have *cooperative* goals. This means that the network optimum state might not equal the single-node optimum state for individual nodes; rewards can only be defined in the context of the network. On the other hand, nodes in cognitive radio networks attempt to optimize link performance, which can be defined in terms of individual nodes.

2. The channel considerations in CRNs differ from cognitive radio networks. While a cognitive radio might seek to maximize the channel capacity or throughput, CRNs are better suited to maximize SINR or minimize target tracking error.

3. Nodes in a CRN operate at relatively higher power, due to the need to collect energy reflected from targets. This means that the *mutual interference* problem is much greater in CRNs, since unexpected interference within a network can cause unacceptable performance.

   The characteristics of reinforcement learning mark it as one of the best tools with which to approach cognitive radar problems. In particular, there is a harmony between the cyclic observe-act loop of cognitive radar and the iterative nature of the reinforcement learning model.

1.2 Prior Work

1.2.1 Spectrum Coordination of Radar Networks

Spectrum Sharing

The review conducted in [34] discusses three main classes of spectrum sharing problems. The first considers the case where a communication system is a secondary user, and must avoid causing interference to a primary radar system. The second class considers radar systems as a secondary user which must avoid interfering with a communication system. The third class considers spectrum sharing where both the radar and communication systems must coordinate to share spectrum. These second and third cases are of interest in this section.

A "cognitive engine" is proposed in [35] to serve as central coordinator in a joint cognitive radar and cognitive radio network. The cognitive engine uses spectrum occupancy information and the physical location of nodes to determine an optimal allocation of the constrained spectrum resources. This technique is highly centralized and relies on an abundance of sensing and communication, and is ill-suited to the general cognitive radar network problem due to the presumption of a abundant communication resource.

Radar sensing is used increasingly often in automotive systems to provide situational awareness to automated as well as human-operated vehicles. However, with increasing density of automotive radars on roadways comes a coordination problem. The work of [36] designs a distributed networking protocol to mitigate mutual interference in networked automotive radars and automotive communication systems. The technique uses a control channel to allocate time-frequency resources to potentially interfering devices. This work fits in the context of connected vehicles, where devices have dedicated spectrum and must coexist with other, similar devices. However in the wider context of radar systems, radar devices might have greater spectrum constraints and be required to interact with dissimilar, non-cooperative devices. This, coupled with a possible lack of side communication channels, motivates a more distributed coordination scheme.

Mutual Interference Mitigation

Radar nodes operating in the same region must display different signal characteristics to avoid interference to other radar devices [37]. The reasoning for this is straightforward; radars transmit known waveforms and examine the reflections for similarity. If two nearby radar nodes utilize the same waveform parameters, the signal-to-interference-plus-noise ratio (SINR) can be unacceptable. While it seems unlikely that two radars would use identical waveforms, the likelihood of two nearby cognitive radars converging to the same "optimum" waveform in the absence of coordination is considerably higher. Also, even if the waveforms are different, transmitting in the same band still causes interference.

The authors of [38] present a deinterleaving scheme to mitigate interference from multiple radars with different waveform parameters. This technique requires estimation of radar parameters, which seems reasonable when radars are non-agile and exhibit repetitive behavior over long time spans but may not apply to systems that are dynamic and adaptive.

Several works have proposed the use of *orthogonal waveforms* for mutual interference mitigation in radar networks [39] [19] [40]. The general idea is to separate the waveforms in some dimension (e.g., time, frequency or code) to improve the SINR.

In cognitive radar, the idea of waveform optimization (whether selection from a library or design) is often discussed. While assignment of orthogonal waveforms from a finite library has been shown to mitigate mutual interference in radar networks, uncoordinated radar nodes have no guarantee that adaptive waveforms will remain orthogonal.

1.2.2 Radar Network Control

In addition to spectrum sharing considerations, radar networks must control higher-level processes such as waveform optimization or power levels. These considerations are separate from spectrum coordination, since the objective tends to be oriented towards performance rather than coexistence. Importantly, CRNs must optimize performance for the entire network, which may lead to different actions than greedily optimizing the performance of each node.

The work of [41] discusses the optimal assignment of target tracks to radar nodes. Since not all targets are visible to the entire network, this is one case when centralized control can result in greater efficiency - for example, a central coordinator can decide which nodes have the best perspective on each target and allocate node time accordingly. The proposed method considers a centralized network of monostatic radar nodes and assumes that each node can only receive reflections of its own emitted signals (in other words, there is no mutual interference). Each node can track at most one target per time step, and the coordinator must determine the optimal node-target assignment.

The authors of [42] consider a centralized power-constrained target detection network. Each of the many radar nodes operates in its own frequency band and experiences no mutual interference. The central coordinator must fuse information from the nodes in order to determine an optimal transmission power and detection threshold for each node.

Control in distributed radar networks has also been considered. In distributed networks, where nodes report observations to a fusion center but receive no control feedback, data fusion can be challenging due to the lack of synchronization between nodes. In [43], the authors demonstrate a decentralized technique for establishing a synchronized clock, where each node senses the waveforms emitted by the other nodes.

Coordination in cognitive *radio* networks is a similar, though not identical, problem to coordination in CRNs. In particular, the spectrum access problem (where multiple users attempt to access the same set of channels) has been addressed in cognitive radio networks using MMAB models [44] [45]. One of several differences between CRNs and cognitive radio networks is channel occupancy: due to differences in waveform and objective, cognitive radio networks can often allow for multiple users in the same channel.

1.3 Contributions of This Dissertation

1.3.1 Motivation

As spectrum becomes increasingly scarce for radar systems, it becomes increasingly important to use spectrum efficiently. CRNs must use spectrum for many reasons: coordination,

sensing, measurement fusion, etc. As we study efficient implementations of cognition for coordination and control in CRNs, we focus on the following areas.

Distributed Spectrum Coordination

Distributed spectrum coordination for cognitive radar networks has not been investigated. As discussed above, the generally accepted framework for CRNs consists of a single fusion center (FC) and a group of several distributed radar nodes. These nodes independently conduct radar measurements in their environment, and cooperatively report target observations to the FC to aggregate information and establish higher-accuracy target estimates.

Since this framework was first described, the literature has produced several optimization schemes for *centralized* radar networks, assuming the presence of a single central coordinator, which is the only cognitive agent [46], [47], [48]. However, there is a lack of research investigating *distributed* cognition in the CRN literature. Of particular importance in CRNs is the allocation of spectrum resources. Such decisions ideally require local information (i.e., the observation quality each spectrum band provides varies from location to location). This is particularly difficult, since spatial variation in the environment may cause the optimal resource allocation to *appear different depending on radar location.* Centralized coordination has been studied, where each node reports spectrum quality to a central decision-maker, which allocates resources accordingly. However, to the best of my knowledge, techniques for online distributed spectrum allocation in CRNs is an open problem which has not been previously investigated.

The only solution proposed in the literature was [49] which *pre-allocates* resources to each radar node. This technique guarantees that the initial resource allocation is optimal. However, it does not account for any time-dependent network parameters. This shortcoming is difficult to overcome; it requires continual sensing and updating of the environment model, as well as coordination between radar nodes. The first major contribution of this thesis addresses this challenge by applying MMAB models to CRNs.

Central Feedback

Input from a central coordinator can improve the performance of CRNs due to more accurate global knowledge being used for decision-making. However, the cost (in terms of power, spectrum, etc.) of this coordination has not been well studied. While our first major contribution is concerned with completely distributed behavior in CRNs, it remains possible that some amount of centralized coordination of spectrum resource allocation could improve CRN performance, while not significantly increasing the network operation costs. As Chapter 3 shows, completely distributed techniques may incur large amounts of cumulative regret.

Work on centralized CRNs [48] [46] has established the benefits of using a central coor-

dinator. Some aspects of cognition are difficult or impossible without centralized control, e.g. coordinated beamsteering which requires tight synchronization between nodes. Further, centralized networks reduce the complexity of each radar node, enabling lower cost manufacturing.

At the same time, centralized networks have considerable drawbacks. The communication overhead in centralized CRNs has not been studied, and could be considerable when pulse-to-pulse level control is necessary. The second major contribution of this document analyzes the trade-off between feedback and performance in CRNs.

Target Update Rates

Not all targets require updating at the same frequency to obtain similar tracking performance. When communication is a constrained resource in a network, it may be necessary to restrict the rate at which each node can transmit tracking updates (estimated target parameters) to a fusion center. Further, due to differences in the expected motion of each target, the fusion center may require different update rates to maintain an error threshold. If each target were updated at an optimal rate, the use of the communication resource could be greatly reduced while maintaining the same tracking error.

Age-of-information metrics have been shown to be useful in wireless sensor networks to regulate the rate at which nodes update a fusion center on observed processes [50]. However, as discussed in Section 1.2, there exist differences in the objectives of sensor networks and CRNs. Thus, solutions for sensor networks cannot be directly applied to CRNs. As a result, there is a need for communication resource allocation in CRNs to optimize performance across the network while staying within the communication resource constraint.

The third contribution of this document address the development of track-sensitive Age-of-Information metrics to optimize the use of a shared, constrained communication channel used for target updates.

Mode Control

Use of "mode control", the selection of operation modes, can improve the spectral efficiency of CRNs. The parts of a CRN will not be the only active wireless devices in the environment, and often it is the case that targets themselves also act as interferers. In these cases, it becomes useful to consider use of the passive sensing techniques at the nodes in a CRN. Since nodes operate in a cognitive and monostatic manner, they already possess dynamic receivers. The energy emitted by targets offers the opportunity to conduct two main types of passive sensing: direction of arrival estimation and waveform classification.

The direction of arrival of a signal can inform a radar node of both the angle to the target, and the angular velocity of that target over time. Combined with sensor fusion algorithms,

direction of arrival estimation is a useful source of information that CRNs should not ignore.

The parameters of waveforms emitted by targets can also be a useful source of information, especially in the context of a relationship between types of waveforms and types of targets (e.g. drones are unlikely to emit voice-coded waveforms).

These passive tools can be even more useful if there is a need to limit the amount of active sensing for each radar node, as can be the case in adversarial environments where radar nodes may wish to avoid being identified or localized. Thus, in Chapter 6 we develop the first known algorithms for controlling the mode (active or passive) of the nodes in a CRN in an attempt to minimize active sensing while meeting a desired target tracking performance.

1.3.2 Contributions

This dissertation presents the following main contributions.

1. Motivated by problems in **distributed spectrum coordination**, and in particular problems with distributed cognition, in Chapter 3 this dissertation contributes:
    - The first system model for multi-player stochastic bandit algorithms in CRNs.
    - A novel two-level algorithm capable of converging towards optimal center frequency and waveform selection for radar operation. Our proposed online learning technique is capable of converging to an optimal solution under sub-linear cumulative regret [30], which corresponds to radar tracking estimation improving over time. We provide discussion on this self-organizing CRN can avoid both mutual and non-cooperative interference.
    - Analysis of the supporting mathematics for a MMAB algorithm used for waveform selection in cognitive radar networks. We discuss the reward scenarios and decision making structure necessary to apply the MMAB techniques to the CRN problem.
2. Building on the model of distributed decision-making, Chapter 4 then considers the role of **central feedback** in CRNs, and in particular contributes:
    - The first work studying the role of feedback in cognitive radar networks. In particular, we study the case where cognition is divided between a Central Coordinator and the individual Cognitive Radar Nodes. We do this by developing a framework for feedback, then structuring several algorithms which take advantage of different levels of feedback. We show that there is a direct correlation between feedback and target tracking performance.
    - A system model for analyzing feedback in CRNs, where a CC provides data fusion as well as cognitive functions. This is useful for future works, as such a model does not yet exist in the literature.

- A mathematical analysis of the different reward functions available to learning algorithms in such a framework. In addition, we discuss when approximations to these rewards may be merited.
- A demonstration that CRN performance can be significantly improved over short time horizons when feedback is used, and that even infrequent feedback is sufficient to improve convergence time in some scenarios.

3. With this comparison of the benefits of centralized and distributed decision-making in place, we consider how a CRN should optimize usage of limited resources to determine **target update rates**. Chapter 5 contributes:
   - A centralized "track-sensitive AoI metric," which utilizes a polling process to allow the FC to select nodes to provide updates in each update interval.
   - An adaptation of the Age of Incorrect Information metric to enable distributed decision-making, where each node implements a Bellman-optimal policy to determine when to provide updates. In contrast to the centralized solution, where the FC coordinates all interactions, the distributed solution relies on each node to decide when to send updates. Specific adaptations to the Age of Incorrect Information metric include a modified Markov model and rate limits for multiple nodes.
4. Motivated by the addition of **mode control** to such a network, Chapter 6 finally contributes:
   - A model for mode selection in multi-function cognitive radar networks.
   - An analysis of multiple target class formation based on characteristic motion and signal emission models.
   - Mathematical analysis of a clustering-based class formation technique.
   - A centralized approach which mitigates the effects of network latency.
   - Numerical simulations to support our conclusions.
   - We show that our proposed techniques outperform radar-only observation as well as outperforming a random selection algorithm which achieves the same radar observation rate, but does not consider target class formation.

The publications supporting this dissertation are listed in Section 1.3.3.

1.3.3 Publications

I have been honored to work in various research areas and with many exemplary colleagues during my time at Virginia Tech. Below are listed the journal and conference publications which directly contribute to this dissertation, followed by those publications which emphasized out-of-scope research thrusts.

Relevant Journal Articles

- [51] **W. W. Howard**, A. F. Martone and R. M. Buehrer, "Distributed Online Learning for Coexistence in Cognitive Radar Networks," in *IEEE Trans. on Aerospace and Electronic Systems*, pages 1202-1216, 2022. doi: 10.1109/TAES.2022.3198038.

- [52] **W. W. Howard** and R. M. Buehrer. "Hybrid Cognition for Target Tracking in Cognitive Radar Networks," in *IEEE Trans. on Radar Systems*, pages 118-131, 2023. doi: 10.1109/TRS.2023.3282846.

- [53] **W. W. Howard**, A. F. Martone and R. M. Buehrer, "Timely Target Tracking: Distributed Updating in Cognitive Radar Networks," submitted to *IEEE Trans. on Radar Systems (Under Revision)*, 2023.

-

- [54] **W. W. Howard**, A. F. Martone, and R. M. Buehrer, "Mode Control in Cognitive Radar Networks," submitted to *IEEE Trans. on Radar Systems (Under Review)*, 2023. Available online.

Relevant Conference Papers

- [55] **W. W. Howard**, C. E. Thornton, A. F. Martone and R. Michael Buehrer, "Multi-player Bandits for Distributed Cognitive Radar," in *2021 IEEE Radar Conference (RadarConf21)*, Atlanta, GA, USA, 2021.

- [56] **W. W. Howard**, A. F. Martone and R. M. Buehrer, "Adversarial Multi-Player Bandits for Cognitive Radar Networks," in *2022 IEEE Radar Conference (RadarConf22)*, New York City, NY, USA, 2022.

- [57] **W. W. Howard** and R. M. Buehrer, "Decentralized Bandits with Feedback for Cognitive Radar Networks," in *2022 IEEE Military Communications Conference (MILCOM)*, Rockville, MD, USA, 2022.

- [58] **W. W. Howard**, Charles E. Thornton, and R. Michael Buehrer, "Timely Target Tracking in Cognitive Radar Networks," in *2023 IEEE Radar Conference (RadarConf23)*, San Antonio, TX, USA, 2023.

- [54] **W. W. Howard**, S. R. Shebert, B. H. Kirk, and R. Michael Buehrer, "Mode Selection and Target Classification in Cognitive Radar Networks," *Submitted*, 2023. Available online.

Out-of-Scope Publications

- [59] D. Tait, J. Yu, **W. W. Howard**, and R. M. Buehrer, "Direction of Arrival Estimation of Digital Sources with Uni-Vector-Sensor ESPRIT," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Atlanta, GA, USA, 2020.

- [60] J. Yu, **W. W. Howard**, D. Tait, and R. M. Buehrer, "Direction of Arrival Estimation with a Vector Sensor using Deep Neural Networks," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, Helsinki, Finland, 2021.

- [61] **W. W. Howard** and R. M. Buehrer, "Multi-Target Localization Using Polarization Sensitive Arrays," in *2021 IEEE Military Communications Conference (MILCOM)*, San Diego, CA, USA, 2021.

- [62] J. Yu, **W. W. Howard**, Y, Xu and R. M. Buehrer, "Model Order Estimation in the Presence of Multipath Interference using Residual Convolutional Neural Networks," accepted to *IEEE Trans. on Wireless Communications*, 2023.

- [63] C. E. Thornton, **W. W. Howard** and R. M. Buehrer, "Online Learning-Based Waveform Selection for Improved Vehicle Recognition in Automotive Radar," in *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023.

- [64] R. M. Buehrer, **W. W. Howard**, and S. Ellingson, "Open and Closed-Loop Weight Selection for Pattern Control of Paraboloidal Reflector Antennas with Reconfigurable Rim Scattering," submitted to *IEEE Trans. on Aerospace and Electronic Systems (Under Review)*, 2023.

1.4 Organization

While this introductory chapter covered a high-level overview of the research areas of relevance to this dissertation, Chapter 2 provides technical details on the problems we address and the necessary modeling considerations. The remaining chapters of this dissertation are organized into journal-style contributions, with some content reproduced from prior publications.

Chapter 3 is titled "Fully Distributed Online Learning for Coexistence in Cognitive Radar Networks", and studies the ways in which a distributed nodes in a CRN can optimally select non-colliding actions from many available channels. This chapter addresses the problem of distributed spectrum coordination. In particular, this chapter contributes a study of *intentional collisions*, where the nodes in a CRN intentionally select colliding channels in

order to exchange information. To facilitate this, we derive a threshold for energy detection to enable the sensing of intentional collisions.

Chapter 4 goes on to investigate the benefit of *feedback* in CRNs, where a cognitive central coordinator (**CC**) can distribute information to radar nodes to improve performance. This is the problem of central feedback as described above. This work studies the hybrid techniques a CRN can utilize, where cognition is balanced between the edge nodes and the CC.

Chapter 5 explores the problem of distributed updating in CRNs: how should a network of independent nodes provide updates in a timely manner, while respecting resource constraints? This is the problem of target update rates. This question is answered using two tools from the Age of Information (**AoI**) literature. The first is a centralized track-inspired metric, which seeks to minimize both age and track variance by selecting the nodes which minimize an objective function. The second technique is based on the Age of Incorrect Information (**AoII**) metric, and enables the network to provide updates in a distributed manner, with each node acting independently.

Chapter 6 then introduces the trade-off between multiple modes of operation in CRNs. This is the problem of mode control. In particular, we consider the balance between active radar observation, which requires no cooperation from targets but may be "intercepted" by adversaries, and passive signal parameter estimation, which relies on targets to emit detectable signals but requires less power and is not susceptible to being intercepted. We motivate this trade-off by introducing target class formation and estimation, by which a CRN may associate the kinematics of a target with the typical signals of that target. This allows radar nodes to more accurately filter target tracks, and reduces the need for active observation. We then utilize several metrics to find an optimal trade-off between these modes.

Finally, Chapter 7 summarizes the major contributions of this dissertation, draws conclusions, and proposes future work.

Chapter 2

Technical Background

Throughout this dissertation we reference topics from mathematics, statistics, machine learning, and other areas. While it is not possible to cover all of the technical areas considered in this dissertation, this section provides a brief treatment of several critical areas to aid the reader's comprehension. First, some common notation and useful results are shown, followed by several modeling considerations utilized in the following chapters.

2.1 Stochastic Bandits

Problems of *iterative learning* involve sequential interactions between an agent and an environment [65]. The agent takes one of several available actions and observes a reward generated by the environment. This framework has been applied to many problems related to the human experience: medical trials (e.g. which treatment to prescribe), content recommendation systems (e.g. Netflix), investment strategies, etc. We choose stochastic bandit models to represent behavior in cognitive radar models due to their Markovian nature (i.e., future states only depend on the current state) and their cyclic process.

Broadly, the iterative learning framework is structured as Algorithm 1, where the agent interacts with the environment at all times $[T] = [t_1, t_2, t_3, ..., T]$ with $T < \infty$ selecting one of $K < \infty$ arms.

Algorithm 1: Action selection at time t.

1. Select arm a_t according to policy ϕ.
2. Observe reward r_t from the environment.

Problems with this structure are referred to as *Multi-Armed Bandits* (**MABs**). The goal of the agent is to select the arms $a_t,\ t = 1 : T$ such that $\sum_{t=1}^{T} r_t$ is maximized. The name is inspired by slot machines, where a gambler can choose to pull one of several slot machine "arms" and receive some cash reward from each, which the intention to maximize the payout after some finite number of pulls.

Remark 2.1 (Reward Distributions)**.** For each action a there is a reward distribution $\mathcal{D}_a$ over the positive real numbers. For simplicity, this distribution is often taken over the unit

interval $[0, 1]$. The agent only observes a sample from $\mathcal{D}_{a_t}$ when arm a_t is chosen at time t; no other reward distribution is sampled. The reward distribution can be arbitrary and need not be known to the agent; algorithms are primarily concerned with the mean rewards for each arm, since as $t \to \infty$ the average reward tends to the sample mean.

Since the problem is modeled as a collection of distributions, it is often referred to as a *stochastic bandit*. As stated above, objective of the learner is to maximize the total reward. And, as discussed in the remark on reward distributions, often a learner will only care about the expected reward for each arm.

Definition 2.2 (Policy)**.** A policy ϕ is a mapping from histories to actions.

$$\phi(a_{1:t}, r_{1:t}) \Rightarrow a_{t+1} \tag{2.1}$$

Different policies will result in different payoffs for the learner.

Example 2.3 ($\varepsilon - Greedy$)**.** The $\varepsilon - greedy$ policy with fixed probability ε selects a random action with probability ε, and selects the arm which has provided the highest average reward with probability $1 - \varepsilon$. This results in a policy which continues to explore the actions, even long after the best arm is identified.

Algorithm 2: ε-Greedy Policy.

$Q_1 = \text{ones(K)}$
for *t=1:T* **do**
  if $rand \leq 1 - \varepsilon$ **then**
    $a_t = \max_n Q_t(n)$
    Else, $a_t = \text{randi}(K)$
  end
  $Q_t = \frac{\sum_{t \text{ s.t. } a_t = n} r_t}{\sum_{t \text{ s.t. } a_t = n} 1}$
end

The example of the ε-greedy policy demonstrates that not all policies will perform the best. The performance of a policy is typically evaluated using the regret, which is the difference in cumulative reward between a policy and an optimal policy which always chooses the best action.

Definition 2.4 (Regret)**.** The regret is the difference in cumulative reward between a policy ϕ and an optimal policy ϕ^* which always selects $a^* = \max_{a_n \in A} \mu_n$ and observes $r_t = \mu^* \; \forall t$.

$$R_t^{\phi} = t\mu^* - \mathbb{E}\left[\sum_{i=1}^{t} r_i\right] \tag{2.2}$$

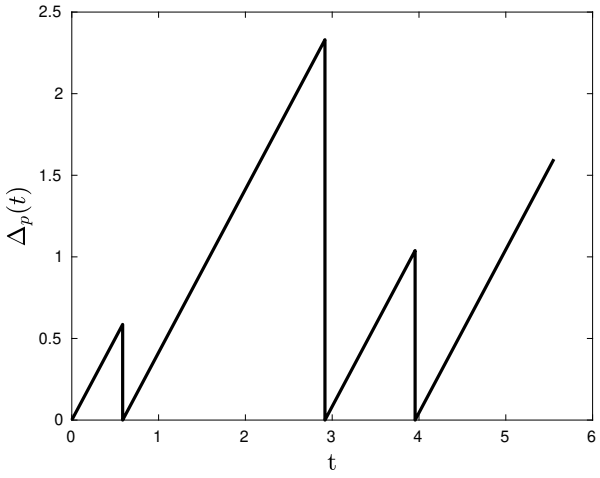


Figure 2.1: A linear age process.

Remark 2.5 (Alternative Formulations)**.** There also exist learner formulations which are concerned with the variance of reward distributions [66] as well as the worst-case reward [67]. These special cases are often important to consider in *adversarial* formulations. Adversarial bandits are aware of the policy *a priori* and can select a sequence of rewards which minimize the learner's payoff. The learner's job, in this case, is still to maximize the expected reward.

2.2 Age of Information

The age of information (**AoI**) literature [50] [68] is concerned with the freshness of information, and the latency incurred by information flow through a network. The age $\Delta_p(t)$ of a process p denotes the amount of time since knowledge of the process was updated. The process can be various time-dependent values: the status of a server, the number of items in a queue, the state of a Markov chain. The age of a process is typically assumed to be linear in time. Fig. 2.1 demonstrates the age of a process which receives updates at random times. As target information at an aggregator becomes older, it becomes less reliable, even if a tracking filter is propegated forward in time. For this reason, the study of the age of information is important in CRNs.

This quantity by itself is of limited use, but the literature has developed several analytical tools to help interpret the age of information.

Average AoI The average age of information was one of the initial AoI metrics, and captures the average age of a process. Eq. (2.3) shows the time-average age until finite time

T, while Eq. (2.4) is the limit of the time-average age as $T \to \infty$.

$$\langle \Delta_p \rangle_T = \frac{1}{T} \int_0^T \Delta_p(t) dt \tag{2.3}$$

$$\Delta_p = \lim_{T \to \infty} \langle \Delta_p \rangle_T \tag{2.4}$$

Peak AoI In contrast to the average AoI, the peak AoI is defined as the average of $\Delta_p(t)$ at the times when updates are received. Define

$$A_n = \Delta_p(t'_n)$$

to be the process age at times t'_n when updates are received. Then, the peak AoI is written as

$$\Delta_p^{(m)} = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} A_n \tag{2.5}$$

Age of Incorrect Information The age of incorrect information is defined to be the time since a data aggregator had the correct state of a Markov process being observed by a remote sensor. The sensor determines when to send updates to the aggregator.

In general, the AoII for a Markov process $X(t)$ is written as

$$\Delta_{AoII}(t) = f(t) \times g\left(\widehat{\mathbf{X}}(t), \overline{\mathbf{X}}(t)\right) \tag{2.6}$$

where $f(t)$ denotes a *time penalty* function, and $g(\widehat{\mathbf{X}}(t), \overline{\mathbf{X}}(t))$ an *information penalty* function. The penalty function takes the current state $\widehat{\mathbf{X}}(t)$ and the *last updated state* $\overline{\mathbf{X}}(t)$ as inputs, and represents the distance between a node's current state and the last known state of the FC.

2.3 Stochastic Geometry and Measure Theory for Network Modeling

In addition to the sections above which cover optimization techniques for problems of sequential decision-making, modeling considerations for networks must be discussed. We primarily utilize methods from stochastic geometry to simulate and analyze the radar networks proposed in the chapters which follow. Since we consider problems of multiple nodes and multiple targets, it is important to consider the geometry of the scenario, especially when resource constraints are introduced. The use of stochastic geometry allows questions such

as "how much of a resource should we expect to need?" and "how are targets distributed through space?".

As provided in [69, Theorem 2.9], a Poisson Point Process (**PPP**) Φ on $\mathbb{R}^d$ with intensity measure Λ and density λ per unit area can be simulated or *realized* on a compact region $B \subset \mathbb{R}^d$ with Lebesgue measure $0 < |B| < \infty$ by drawing a Poisson number N with mean $\lambda|B|$ and subsequently drawing N points uniformly at random from B. When B is a box[1] in the appropriate space, the technique is most straightforward. Depending on the context, Φ can refer to the point pattern in $\mathbb{R}^2$ and $\Phi(B)$ can refer to the measure of the Borel set B with regard to the PPP (i.e. the expected number of points in B). We have the distribution of N:

$$\Pr[N = n] = \frac{\lambda|B|^n e^{-\lambda|B|}}{n!} \tag{2.7}$$

and the measure of B:

$$m(B) = |B| = \prod_{i=1}^{d} b_i - a_i \tag{2.8}$$

where a_i and b_i are the endpoints for the interval of B in each dimension. Conveniently, the mean number of points associated with the PPP scales with the measure of B as well as the density λ. Different realizations of the PPP on B may contain a different number of points, in addition to different point patterns. However, some general results remain consistent between realizations. Of particular importance to this dissertation is the *coverage* of a PPP.

Definition 2.6 (Germ-grain model)**.** Let $\Phi = \{x_i\}$ be a point process on $\mathbb{R}^d$ representing the *germs* and let $(S_1, S_2, \dots)$ be a collection of random non-empty sets called the *grains*. Then,

$$\Xi = \bigcup_{i \in \mathbb{N}} x_i + S_i \tag{2.9}$$

is a germ-grain model. $\Xi \in \mathbb{R}^d$ is the *covered region*, and a location $y \in \mathbb{R}^d$ is said to be *covered* if $y \in \Xi$.

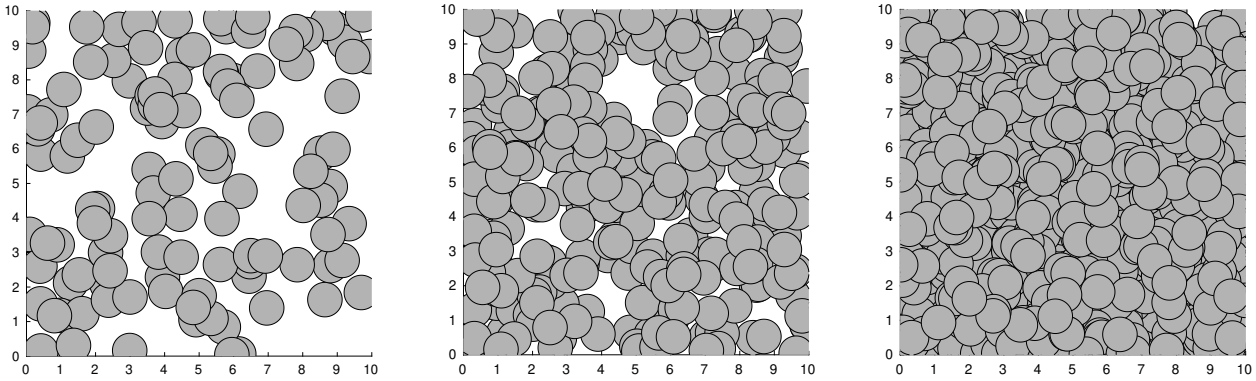
The most tractable form of germ-grain model is the Boolean model, where the germs are a uniform PPP and the grains are i.i.d, typically balls of a fixed or random radius. Figure 2.2 shows a few realizations of Boolean germ-grain models with different densities and a fixed grain radius.

The probability that any location x is covered is

$$\Pr[x \in \bigcup_{i \in \mathbb{N}} S_i] = 1 - e^{-\lambda \mathbb{E}|S} \tag{2.10}$$

and we also have that the number of times each point is covered is Poisson distributed with mean $\lambda \mathbb{E}|S|$ [69, Theorem 13.5].

[1] In the sense of [70, Def. 1.1.1].



(a) (b) (c)

Figure 2.2: Realizations of three germ-grain models with $|B| = 100$, $r = 0.5$, and (a) $\lambda = 1$, (b) $\lambda = 3$, and (c) $\lambda = 10$.

We apply this theory to our network models by modeling the distribution of radar nodes and the initial distribution of targets as PPP realizations. Say λ_n is the density of radar nodes with N the Poisson random variable describing the number of nodes. Similarly, let λ_m be the target density and M be the Poisson random variable describing the number of targets. When we consider multi-target tracking (as in Ch. 5 and Ch. 6), we take $\lambda_m > \lambda_n$.

Since each is a uniform process on B, we can discuss both the probability that a given target (i.e., random location in B) is covered, as well as the expected number of targets covered by a given node. Since a PPP can be described as a random measure on a space, the number of targets in the coverage (grain) for a given node is the target measure of the covered region. Fig. 2.3 shows the relationship between target density and coverage radius.

2.4 Target Tracking

Target tracking is a common function for radar devices. Over the course of many samples, a radar localizes a target and updates its knowledge of the target's state. We will begin by discussing single-target tracking techniques, which are useful when there is only one target of interest and the sensor has a high probability of detection and low false alarm rate. This means that in each of many time steps, a sensor returns an estimate of the target's state. Depending on the sensor, this estimate can be in various coordinate systems and measure various quantities. A radar using range-Doppler processing is able to estimate the range r to a target, the angle θ to the target, as well as the radial velocity (sometimes called the range rate) $\dot{r}$ and the angular velocity (over multiple time steps). These quantities are shown in Fig. 2.4, and range-Doppler processing is shown in section 2.5.

When a radar node observes the target at time k, it estimates the target position $\hat{\mathbf{y}}(k) =$

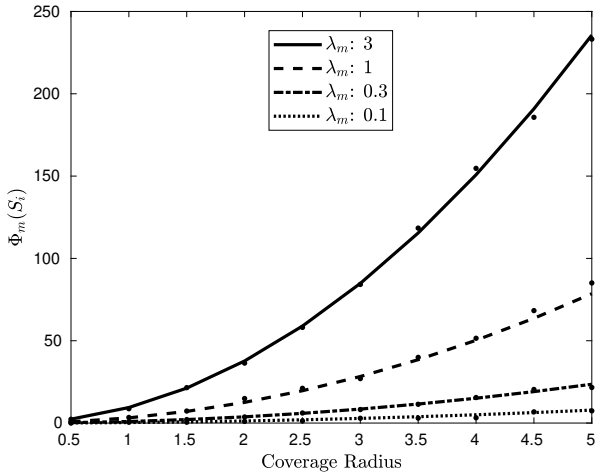


Figure 2.3: The expected and empirical number of targets in a covered region of specified radius, for various target densities.

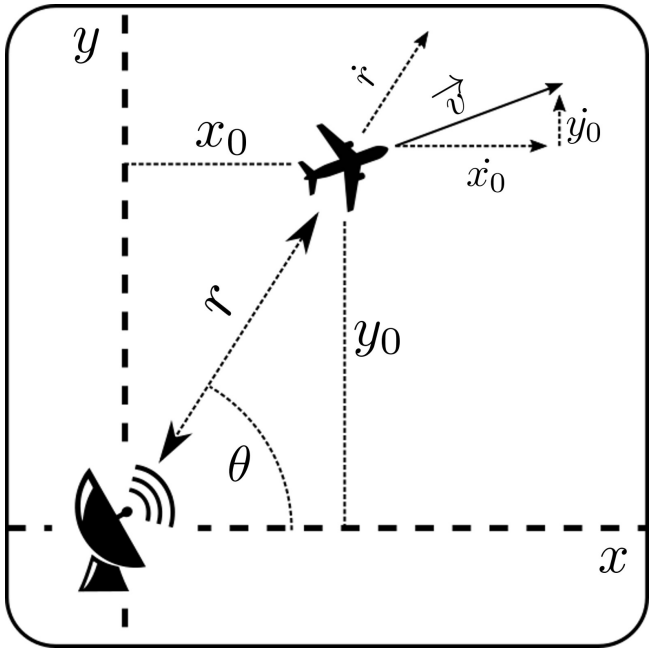


Figure 2.4: Radar measurement is in radial coordinates.

$[x_0, y_0]$ and velocity $\hat{\dot{\mathbf{y}}}(k)$. These are used to update a Kalman filter model of the target's motion. Assuming a two-dimensional motion model, the predicted state is given as Eq. (2.11) where F_k is the transition model. Note that the state $x_k = [\mathbf{y}_m(k)^T, \dot{\mathbf{y}}_m(k)^T]^T$ is composed of target position and velocity.

$$\hat{x}_{k|k-1} = F_k x_{x-1|k-1} + B_k u_k \tag{2.11}$$

The state is then updated as Eq. (2.12) where K_k is the estimated optimal Kalman gain and $\tilde{y}_k$ is the innovation.

$$x_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{2.12}$$

This formulation follows the common notation of [71].

2.4.1 Multi-Target Tracking

When multiple targets are present in the scene, the *data association* problem becomes relevant. This is the difficulty in determining which detections belong to which target tracks, as well as determining whether new targets have appeared or old targets have disappeared. The literature treats this is a *random finite set* [72]. The Probability Hypothesis Density (**PHD**) filter [73] [74] has been proposed to solve this problem.

Each target m in the observable region for node n generates a number of detections $N_{Z,mn}$:

$$\mathbf{Z}_{mn} = \left\{\mathbf{z}^{(j)}\right\}_{j=1}^{N_{Z,mn}} \tag{2.13}$$

Each target detection is missed with probability P_D. Since more than one target may be in the region covered by node n, the total number of target detections generated at time t by node n is given as Eq. (2.14).

$$\mathbf{Z}_n^t = \bigcup_{m \in \mathcal{M}_n^{(t)}} \mathbf{Z}_{mn} \tag{2.14}$$

Then, a PHD filter [72] [75] is used to associate [76] these target detections with previous tracks and generate new tracks if necessary. Since $\mathbf{Z}_n$ may contain false alarms generated at a rate of λ_{FA}, the PHD filter maintains a list of tentative and confirmed tracks. False alarms are uniformly distributed in the region.

Let $\mathbf{Z}_n^{1:t}$ denote the ordered set of observations until time t.

From [75], we have the multiple model PHD tracking filter for maneuvering targets. We reproduce the core steps of the technique here. The filter begins with an initial density Eq. (2.15).

$$\tilde{D}_{t|t-1}\left(\mathbf{X}(t-1), V(t) = i | \mathbf{Z}_m^{1:t-1}\right) \tag{2.15}$$

The motion model mixing is given by Eq. (2.16), where $V(t)$ is one of several motion model states, N_V is the number of motion model states, and $P_i j$ are the transition matrix entries,

Eq. (6.2), for transitioning from state i to state j. Note that there is a separate PHD for each possible motion model.

$$\tilde{D}_{t|t-1}\left(\mathbf{X}(t), V(t)=i|\mathbf{Z}_n^{1:t-1}\right) = \sum_{j=1}^{N_V} D_{t-1|t-1}\left(X(t-1), V(t-1)=j|\mathbf{Z}_n^{1:t-1}\right) P_{ij}, \quad i=1:N_V \tag{2.16}$$

The PHD prediction step is given as Eq. (2.17), where $\gamma_t(\cdot)$ is the target birth PHD, $e_{t|t-1}(\cdot)$ represents the probability that each target survives to the next round, $f_{t|t-1}$ and is the motion state conditioned target likelihood.

$$D_{t|t-1}\left(X(t), V(t)=i|\mathbf{Z}_n^{1:t-1}\right) = \gamma_t(X(t), V(t)=i) + \int \left[e_{t|t-1}\left(X(t)\right) f_{t|t-1}\left(X(t)|X(t-1), V(t)=i\right)\right] \times \tilde{D}_{t|t-1}\left(X(t), V(t)=i|\mathbf{Z}_n^{1:t-1}\right) dX(t-1) \tag{2.17}$$

Finally, the PHD update step is given as Eq. (2.18), where P_D is the probability of detection, λ_{FA} is the false alarm rate, C_{FA} is the false alarm spatial distribution, and $\Psi_t(\cdot)$ is the PHD likelihood function, Eq. (2.19).

$$D_{t|t}\left(X(t), V(t)=i|\mathbf{Z}_n^{1:t}\right) \cong \left[\sum_{\mathbf{z}\in\mathbf{Z}_n^t} \frac{P_D(X(t)) f_{t|t}\left(\mathbf{z}|X(t), V(t)=i\right)}{\lambda_{FA} C_{FA} + \Psi_t\left(\mathbf{z}|\mathbf{Z}_n^{1:t-1}\right)} + \left(1 - P_D(X(t))\right)\right] \times D_{t|t-1}\left(X(t), V(t)=i|\mathbf{Z}_n^{1:t-1}\right) \tag{2.18}$$

$$\Psi_t\left(\mathbf{z}|\mathbf{Z}_n^{1:t-1}\right) = \int \left[P_D(X(t)) f_{t|t}\left(\mathbf{z}|X(t), V(t)=i\right) \times D_{t|t-1}\left(X(t), V(t)=i|\mathbf{Z}_n^{1:t-1}\right)\right] dX(t) \tag{2.19}$$

2.5 Radar Signal Processing

2.5.1 Common Waveforms

Radar is performed by emitting a known waveform and processing the reflections from targets and clutter in the environment. In particular, monostatic pulse-Doppler radar is performed

by emitting a train of linear frequency modulated (**LFM**) pulses, then processing the received signal through a bank of matched filters at various delay and Doppler values. The estimated target range and radial velocity are determined by the maximum matched filter response.

Linear Frequency Modulated Pulse LFM pulses are defined from 0 until a time T. There are several parameterizations, but a common one defines a frequency sweep rate k. Eq. (2.20) shows a single time-domain pulse.

$$u_T(t) = \frac{1}{\sqrt{T}} e^{\left[j\pi k(t-\frac{1}{2}T)^2\right]}, \quad 0 \leq t \leq T \tag{2.20}$$

Since LFM pulses are typically transmitted in a phase-coherent train, an infinite sum of pulses is shown in Eq. (2.21).

$$u(t) = \sum_{i=-\infty}^{\infty} u_T(t - iT_r) \tag{2.21}$$

Here, T_r is the *pulse repetition interval*, which has implications on the processing parameters.

Radar Ambiguity Function The radar ambiguity function represents the impact of delay-Doppler processing on radar parameter estimation. The ambiguity function is entirely dependent on the waveformm, and does not depend on the presence of a target or of clutter. Eq. (2.22) shows the ambiguity function for a single LFM pulse, while Eq. (2.23) shows the ambiguity function for a train of N LFM pulses [77].

$$|\chi(\tau, \nu)| = \left| \left(t - \frac{|\tau|}{T} \right) \frac{\sin \alpha}{\alpha} \right|, \quad |\tau| \leq T \tag{2.22}$$

$$|\chi_N(\tau, \nu)| = \left| \left(1 - \frac{|\tau|}{T} \right) \frac{\sin \alpha}{\alpha} \right| \left| \frac{\sin(N\pi\nu T_r)}{N \sin(\pi\nu T_r)} \right|, \quad |\tau| \leq T \tag{2.23}$$

$$\alpha = \pi T \left(\nu \mp B\frac{\tau}{T} \right) \left(1 - \frac{|\tau|}{T} \right)$$

2.5.2 Delay-Doppler Estimation

In very broad strokes, the range of a target is estimated by the *delay* of the pulses going to the target and back to the receiver, and the radial velocity of a target is estimated by the *Doppler* shift of the received pulses.

Delay. The radar range can be simply described by the speed of light, c, in Eq. (2.24), where t is the delay of the pulse. The factor of $\frac{1}{2}$ is a result of the two-way distance; the pulse must travel out to the target and then back.

$$R = \frac{ct}{2} \tag{2.24}$$

The target range is also described by the well-known radar equation, Eq. (2.25), where P_{tx} is the transmit power, P_{rx} is the received power, G is the transmit and receive gain, λ is the pulse wavelength, and σ is the radar cross section (apparent size of the target).

$$R = \sqrt[4]{\frac{P_{tx}G^2\lambda^2\sigma}{P_{rx}(4\pi)^3}} \tag{2.25}$$

This equation describes the target range as a root of the ratio between the transmitted and received power. This makes intuitive sense; as the radar pulse propagates outwards, it spreads in all directions as a sphere. Then, the energy scattered by the target also spreads in a sphere. The radius of these spheres is the range to the target.

Doppler. The Doppler shift of the received signals relates to the target as Eq. (2.26), where v is the target radial velocity, f_{tx} is the transmitted frequency, and f_d is the Doppler shift of the frequency.

$$v = \frac{f_d}{f_{tx}}c2 \tag{2.26}$$

Most implementations contain a bank of filters matched to possible delays and Doppler shifts. The received pulses are processed with this filter bank, resulting in a grid of filter responses for each pair of delay and Doppler shift. This is called the range-Doppler image, and target detection is performed by determining a threshold value. When filter responses exceed the threshold value, a target is detected. A common method for determining the threshold is called Constant False Alarm Rate (**CFAR**) detection [78].

Example 2.7 (Pulse Doppler Processing)**.** This example presents one, relatively simple, technique for estimating the Doppler shift of a received pulse.

Let the received and demodulated signal for the m^{th} of M pulses reflected off a target moving at a velocity v such that the Doppler shift is f_d be represented as Eq. (2.27).

$$y[m] = Ae^{j2\pi f_d m T_r} \tag{2.27}$$

In order to estimate f_d, a discrete-time Fourier transform is performed as Eq. (2.28).

$$Y(F) = \sum_{m=-\infty} \infty y[m]e^{-j2\pi F T_r m} \tag{2.28}$$

Substituting in Eq. (2.27), we get Eq. (2.29).

$$Y(F) = A\frac{\sin\left(\pi(F - f_d)MT\right)}{\sin\left(\pi(F - f_d)T\right)}e^{-j\pi(M-1)(F-f_d)T} \tag{2.29}$$

The peak of Eq. (2.29) will occur at $F = f_d$, and so

$$\hat{f}_d = \max_F Y(F) \tag{2.30}$$

In other words, the estimated Doppler shift of the signal can be obtained by sampling the received signal via a DTFT, then finding the maximum value. The velocity can then be estimated by using Eq. (2.26).

For more mathematical detail on radar delay-Doppler estimation, the reader is recommended to the work of [79, Ch. 4] or [80, Ch. 7].

Chapter 3

Distributed Online Learning for Coexistence in Cognitive Radar Networks

Related Publications

The material in this chapter has been reproduced from the following publications:

- [51] **W. W. Howard**, A. F. Martone and R. M. Buehrer, "Distributed Online Learning for Coexistence in Cognitive Radar Networks," in *IEEE Trans. on Aerospace and Electronic Systems*, 2022.
- [55] **W. W. Howard**, C. E. Thornton, A. F. Martone and R. Michael Buehrer, "Multi-player Bandits for Distributed Cognitive Radar," in *2021 IEEE Radar Conference (RadarConf21)*, Atlanta, GA, USA, 2021.

3.1 Introduction

The demand for spectrum reaches new peaks daily in this era of fifth-generation wireless technologies. This increased demand for spectrum has an impact on all types of radio access devices. Radar systems in particular are being affected and must often share their spectrum with other services. To address this impact, our work is concerned with Dynamic Spectrum Access (DSA) schemes for Cognitive Radar Network (CRN) systems. DSA is an umbrella term for techniques which seek to assure spectrum access to dissimilar, and often non-cooperative, systems which attempt to access the same resource while simultaneously attempting to mitigate harmful interference between wireless devices [81]. Specifically with the increased use of spectrum by commercial communication systems in the GHz bands, there is a need for radar systems to be DSA-capable as secondary users (SU). However, while the benefits of using a network of radars are numerous (multiple observations of a target, resiliency to physical damage, etc.), they only exacerbate the coexistence problem. To address this need, this work focuses on a Multi-player Multi-Armed Bandit (MMAB) approach to DSA in CRNs.

Cognitive systems are an attractive platform for the DSA problem for several reasons. Primarily, we leverage the ability of cognitive radar systems to *monitor their environment* to inform future decisions. Of the two modes of CRN cognition outlined by Haykin in his seminal work on the topic [6], this approach falls under "distributed cognition", where each node in the network is a cognitive agent. This contrasts with the alternative "centralized cognition" where a *central coordinator* is the only intelligent agent. We do, however, describe a "fusion center" which collects observations from the network for the purpose of target tracking on the time scale of the Coherent Pulse Interval (CPI)[1]. Note that our MMAB technique does not facilitate learning in a joint or federated sense [82]; each radar node is aware of the presence of the network, but makes its own decisions. In this network, each node is aware of its own position, and the fusion center is aware of all node locations. Nodes are not aware of the placement of other nodes. This follows the assumption that communication is one-way, from node to fusion center.

We also leverage the inherent flexibility of CRN systems, assuming a network of identical, frequency agile cognitive radar nodes which are able to choose non-overlapping frequency bands and waveforms from a finite library. This agility allows each independent cognitive radar node in the network to avoid other emitters in the environment. In addition, the described CRN uses sensing and agility to avoid causing harmful interference within the network. We discuss the use of a library of orthogonal waveforms, which has been shown to result in both lower ambiguity[2] [40] and reduced mutual interference [39].

Our approach applies iterative, *online* Reinforcement Learning (RL), enabling a network of independent identical cognitive radar nodes (or just "nodes") to collectively optimize their frequency band and waveform selections over time. Each node in the network has the same goal: to optimize target tracking and detection for the entire network. This means that actions which cause low Signal to Interference plus Noise Ratios (SINR) for any node in the network are penalized. One of the ways this is accomplished is by detecting instances of "collisions", or mutual interference, which occur when more than one node selects the same waveform and center frequency in the same PRI.

This work described a method for using two simultaneous techniques to enable a cognitive radar node to select both a center frequency (i.e. frequency band) and a waveform in each time step. First to select a center frequency, a MMAB algorithm is implemented which observes rewards from the environment, and collisions between radar nodes. The goal of this first algorithm is to obtain an *optimal frequency allocation*, which is defined later, but is an assignment which maximizes tracking performance. Second, in order to select waveforms in each time step, an independent single-player bandit model is instantiated in each frequency band, for each radar node. This enables the learner to focus on interactions within the band. The single-player bandit algorithm is only concerned with rewards from the environment,

[1] For comparison's sake, one CPI contains 1024 PRIs, which is the time scale of individual decision making.

[2] This holds even though the bandwidth of the network is divided among the nodes instead of being allocated to a single node.

since over time the frequency allocation will become free of mutual interference.

Multi-armed bandit algorithms were introduced in the 1950s [83] to study the sequential decision-making problem. Specifically, the authors considered the problem of maximizing expected rewards when drawing from two or more differently distributed random sequences. The problem of *multiple players* interacting in this environment was not addressed in the literature until 2010 [84]. The multi-player problem was initially motivated by, among other things, coexistence in cognitive radio applications. We first adapted MMAB algorithms to the cognitive *radar* application in [55].

3.1.1 Problem Summary

This work considers a problem of sequential decision making. In each of many time steps, how should a group of distributed radar nodes select a waveform and center frequency? The use of *multi-player multi-armed bandit* models is discussed to address this problem. We will also investigate and mitigate the effect of non-cooperative transmitters (primary users) which can cause unacceptable levels of interference. Such a network must use some algorithm to select frequencies and waveforms. Any pre-allocation strategy places assumptions on the interference, which is not known. In addition, pre-allocating actions to avoid collisions will not necessarily maximize utility, as we will describe.

Given this, our goal is to describe the two intertwined algorithms for band and waveform selection which a distributed radar network could use to self-organize. The presence of a fusion center is still assumed, which is capable of collecting observations from each node in each CPI for the purpose of target tracking. However, this fusion center provides no support in the network's efforts to determine which actions each node should take, due to the communication and coordination issues discussed above. Measurements are sent to a fusion center and combined on the millisecond timescale, while decisions are made on microsecond intervals.

This work considers two types of interference. The first, which is referred to as *mutual interference*, is when one radar node interferes with another. As is shown later, this can have a detrimental effect on target tracking and detection due to unacceptable Signal to Interference plus Noise Ratio (SINR) levels. The second, which can be termed *outside interference*, is when another system causes interference to one or more radar nodes. Both of these are to be avoided.

We model a network of pulsed radars, where once per pulse repetition interval (PRI) each radar in the network will select a waveform and transmit it on a chosen carrier frequency. As we will describe later, the nodes must share a fixed total bandwidth of B, which is not dependent on the number of nodes. Each radar node must estimate when it encounters interference from other nodes, using a method we will describe.

Each radar is capable of steering the main beam, but due to sidelobe levels and environmental

scatter, some energy from each transmission is received at all nodes in the network. If any two nodes select interfering actions, they will experience degraded performance which we will demonstrate.

3.1.2 Contributions

This paper extends earlier work [55], [56]. We previously investigated the available MMAB algorithms, and in [55] studied reward structures which differ between radar nodes. In [56], we analyzed algorithms tailored towards *adversarial* environments, where the environment has knowledge of the algorithm being used by the radar network, and can pre-select a reward sequence in an attempt to worsen the performance of the network. Rather than focus on these different environment classes, this current work is interested in the following.

- Developing the system model for the MMAB problem in CRNs. We provide the first such model for MMAB algorithms in this context.
- Providing a novel two-level algorithm capable of converging towards optimal center frequency and waveform selection for radar operation. Our proposed online learning technique is capable of converging to an optimal solution under sub-linear cumulative regret [30], which corresponds to radar tracking estimation improving over time. We provide discussion on this self-organizing CRN can avoid both mutual and non-cooperative interference.
- Analyzing the supporting mathematics for a MMAB algorithm used for waveform selection in cognitive radar networks. We discuss the reward scenarios and decision making structure necessary to apply the MMAB techniques to the CRN problem.
- Demonstrating our proposed technique against alternatives such as SAA and fixed allocation in simulation. We then provide conclusions based on our simulations.

Combined with the two earlier publications, this paper represents the first work on the topic of MMAB algorithms for coexistence in cognitive radar networks.

3.1.3 Notation

We use the following notation. Matrices and vectors are denoted as bold upper $\mathbf{X}$ or lower $\mathbf{x}$ case letters. Functions are shown as plain letters F or f. Sets $\mathcal{A}$ are shown as script letters. The cardinality $|\mathcal{A}|$ of a set $\mathcal{A}$ refers to the number of elements in that set. The logical negation of a statement a is given by an overline $\overline{a}$. The transpose operation is $\mathbf{X}^T$. The backslash $\mathcal{A}\backslash\mathcal{B}$ represents the set difference as $\mathcal{A}\backslash\mathcal{B} = \{a \in \mathcal{A} : a \notin \mathcal{B}\}$. Indicator functions of a variable x on a set $\mathcal{A}$ are denoted as $\mathbb{1}_{\mathcal{A}}(x)$. The set of all real numbers is $\mathbb{R}$ and the

set of integers is $\mathbb{Z}$. The speed of electromagnetic radiation in a vacuum is given as c. The imaginary number is i. We use $\mathcal{U}\{\mathcal{A}\}$ to denote the uniform distribution over a space $\mathcal{A}$.

3.1.4 Organization

The rest of this paper is organized as follows. In Section II, we review previous work on radar networks and cognitive systems. Section III develops our proposed machine learning techniques to assign center frequencies and waveforms to each radar in a network. We first detail a method for frequency band selection, followed by a method for waveform selection within that frequency band. Section IV provides simulation results and discussion, and in Section V we draw conclusions and suggest future work.

3.2 Background

3.2.1 Cognitive Radar Networks

Cognitive radar, first described by Haykin [6], is described as having four main elements: the Perception-Action Cycle (PAC); memory; attention; and intelligence [85], [86]. The perception-action cycle describes the iterative interactions typical to this sort of system. Further, the IEEE has defined cognitive radar as systems which display intelligence and can modify both operating and processing parameters in response to a changing environment [87]. This could also refer to spectrum sensing, where a radar would observe spectral behavior, and adapt based on what it sees.

Cognitive radar systems have been proposed for non-cooperative coexistence many times [25, 88, 89]. Specifically of interest in this work are cognitive radar *networks*, which can combine observations from individual radars in an intelligent manner to benefit the network [6, 7].

Cognition in radar networks can be accomplished in two main "modes":

- **Distributed cognition**, where each node in a network possesses cognitive ability, and
- **Centralized cognition** where a central coordinator makes network-wide decisions.

This poses a fundamental trade-off: centralized cognition may appear to offer several benefits, most significantly a reduced problem space[3] due to the combined observations, but

[3]In a centralized scenario, a coordinator can limit the choice of actions to the space which contains no collisions, which we will define later. However in a decentralized scenario, the problem space expands to include conflicting actions.

it has significant drawbacks. Any centralized system becomes vulnerable to the loss of the central node. In addition, reporting observations on a realistic time scale may require communications at the PRI timescale while sharing target information which requires, at most, communications at the coherent pulse interval (CPI) timescale (typically two orders of magnitude slower). Since spectrum access for traditional monostatic radar is already limited, and an increased number of nodes will worsen the problem, cognitive radar networks will benefit from any reduction in wireless communication overhead.

In the context of radar, cognitive systems have been described using the PAC [25, 90, 91], a closed-loop model which implements cognition as defined above. It has been stressed that a major property of this problem is the *interaction* between the cognitive system and its environment [89]. To that end, we can visualize the cycle as shown in Fig. 3.1, where each node uses a cognitive process (multi-player bandit frequency selection paired with a "single-player" waveform selection algorithm in this work) to choose a waveform and center frequency from a library to transmit in each time step.

We use the following notation to describe the network structure:

- **Centralized network**: A CRN using a *central coordinator* to assign frequencies and waveforms, as well as to fuse network observations. The central coordinator has bidirectional communication with all of the nodes.
- **Decentralized network**: A CRN with *no feedback* with a *fusion center* to combine network observations. Information only flows from the nodes to the fusion center; there is no feedback.

3.2.2 Related Work

Radar networks have several methods to track and detect targets. Generally, radar networks fall into two classes [92]: MIMO (multi-input, multi-output) radar networks and radar sensor networks (RSNs). We focus primarily on the latter, since they are well-suited to distributed and decentralized techniques. RSNs utilize multiple radars that are independent of one another and operate in a mono-static mode. Each radar makes independent observations of the environment and forms target models. The central node, if present, can provide feedback over a longer time period and combine sensing observations from all the radars.

A fundamental work on RSN's is [19], where the authors propose a basic framework for RSN target detection. In their work, it is assumed that each radar conducts observations, then losslessly communicates the received waveforms to a clusterhead for signal processing. They show that using Continuous Wave waveforms which do not overlap in frequency, coupled with the spatial diversity inherent to RSN's, provides a significant improvement compared to the baseline single monostatic sensor. Since the waveforms do not overlap in frequency we can consider them to be orthogonal under the definition provided later.

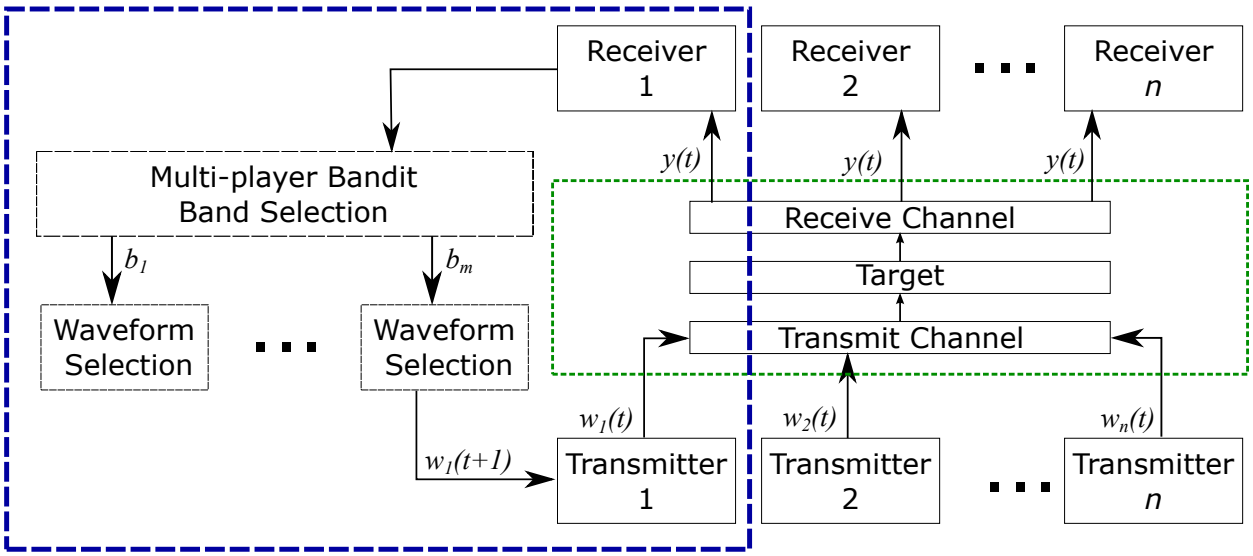


Figure 3.1: Transmit/receive cycle for the cognitive radar network. The decision process for the first transmitter/receiver pair has been shown, but is implemented at each node. Importantly, each node i independently selects a waveform $w_i(t)$, which is modulated by the environment, then returned as a waveform $y(t)$. Using the received energy, the cognitive learner selects the next transmit waveform $w_i(t+1)$.

An early work on CRNs was published by Haykin shortly after his seminal work on cognitive radar [6]. In this work, Haykin describes potential future applications for CRNs including the use of centralized cognition for data fusion and control. Another implementation, with more emphasis on distributed cognition, is described in [93]. A MIMO CRN is split into clusters, with no assumption of communication between clusters. A noncooperative game is used to formulate a power allocation algorithm between clusters to minimize mutual interference.

In this work we assume a decentralized CRN with a *fusion center* that only performs the data fusion function, i.e. it provides no feedback to the nodes. To fuse target information, each node reports target observations on some schedule, i.e. once per PRI or CPI. To avoid solving the implied communications problem, we assume that this unidirectional communication happens once per CPI, occurs on some pre-allocated communication channel separate from radar frequencies, and is error free.

Machine learning techniques are well suited to this problem due to the variable nature of Radio Frequency (RF) environments. Instead, an approach based on pre-allocation of waveforms and frequencies could be selected. This, however, makes strong assumptions on the interference environment, as well as the quality of the target measurement each radar node can make. In addition, traditional methods will place more stringent assumptions on the coordination of the network, making the distributed cognition scheme less powerful. A cognitive radar *network* is selected for this problem specifically for this reason: while a single node may be overwhelmed by an interferer or lack proximity or high Radar Cross Section

(RCS), the inherent spatial diversity of a CRN offers flexibility in many domains, ensuring better tracking accuracy and higher probability of detection [94].

In previous RL-based DSA work, the Markov assumption is often applied [95], [96]. Markov environments have the property that the next state depends *only* on the current state. The goal in this sort of problem, then, is to learn the probability distribution of next states conditioned on the current state. However, interferer behavior in realistic channels often has high temporal correlation, leading to the breakdown of the Markov assumption. This motivates study into online sequential learning, which seeks to exploit any patterning or correlation in previous observations to inform future decision-making.

Mutual interference mitigation is a common goal for machine learning systems in radar sensor networks. A common target application is automotive radar in connected vehicles, where multiple independent radars attempt to share a dynamic environment with many radars entering and leaving a region. This obviously causes problems for any technique relying on pre-allocated frequencies. It is important to note that this application has results that are very easily generalized to radar sensor networks. The work of [97] provides a good overview of the problem, current mitigation techniques, and possible future directions. One technique discussed is beamforming to steer nulls in the directions of other nodes in the network. This can be accomplished through estimating the location of the other nodes, then selecting waveforms to null specific directions. This technique limits the target tracking and detection capability of any network, since energy is necessarily not directed along the Delauny lines (i.e., those lines connecting two nodes) of a network. Any target in these regions would have a lower probability of detection due to the relatively lower power emitted in that direction.

Another common method discussed in [97] borrows Code Division Multiple Access (CDMA) waveform techniques from wireless communications. CDMA waveforms have the benefit of being orthogonal in code, which allows for separability on receive even with overlaps in other dimensions. CDMA commonly has the downside of requiring a broad bandwidth, but this issue is somewhat mitigated in the radar application since wideband channels are usually available. However, a somewhat larger issue is the requirement for synchronization in CDMA schemes which is difficult to guarantee.

In addition, the huge power disparities caused by variations in target and intra-node distances tend to cause swamped target returns, causing large problems in filtering. Proposed future work includes joint radar/communication systems, decentralized multiple access schemes, and use of future modulation techniques to mitigate interference.

In [98] the authors provide analytical bounds for network wide Signal to Interference Ratio (SIR) levels due to different waveform selections. The author of [99] shows probabilities of mutual interference, considering spatial, temporal, and frequency domains. Importantly, they conclude that regardless of temporal delay or spatial separation, mutual interference due to frequency overlap tends to dominate. Thus, in this work, we seek to avoid such interference by choosing orthogonal frequency bands.

To consider the case of non-orthogonal waveforms, [100] presents a cross-matched filtering technique to mitigate mutual interference between radars. Non-orthogonal waveforms used across a RSN can cause high-power backscatter & line of sight interference which can severely impact performance. The cross-matched filter is motivated by the fact that perfectly orthogonal transmitted signals may not remain orthogonal upon receive, so there is a need to sequentially match filters to each transmitted waveform in the network to suppress interference, which leaves only the transmitted waveform. This is predicated on the assumption that perfect orthogonality is needed for coherent processing.

In [39], the authors show that near-orthogonality is acceptable, and introduce the idea of ε-orthogonality to provide a bound on the similarity between waveforms. ε-orthogonality is defined based on the cross-ambiguity between two waveforms, which is related to cross-correlation. However, the authors do not consider the case when large power disparities exist, such as between radar reflections and line of sight interference.

Cross-ambiguity is a time (τ) and frequency (ν) analysis tool used to solve signal processing problems. Specifically, we will look at cross-ambiguity as a way to reduce mutual interference in RSN's. A thorough mathematical description of the cross-ambiguity function is provided in [101].

Mutual interference detection and mitigation in radar systems has been addressed from several directions. [102] describes several coding schemes for automotive frequency-modulated continuous wave radars which eliminate the mutual interference problem without requiring detection. For our scenario, which uses pulsed radar waveforms, this solution is incompatible. Similarly for automotive radar, [103] describes an energy detection method for mutual interference detection, enabling a higher-level process to determine a mitigation scheme. Once detected, the authors propose the use of spatial filters to mitigate the effects of mutual interference. Unfortunately, one of the limitations of this algorithm is that if the target happened to be in the same direction as an interfering node, this method would result in failed detections and poor tracking.

To summarize, one clearly important aspect of resource allocation in radar sensor networks is the consideration of multiple access schemes at each node. By selecting resources correctly at each node, mutual interference can be greatly avoided. This problem draws a lot from the similar problem of interference mitigation in wireless communication networks, but with an important distinction: wireless communication aims to maximize the information transmitted directly to some receiver, while radar networks seek to maximize target information using co-located transmitters and receivers. In this work we consider multiple access primarily through frequency and code division, due to the limitations of a radar network. We will use a library of nearly orthogonal waveforms and orthogonal frequency bands to detail a method for decentralized waveform selection at each node, requiring minimal coordination.

3.3 Optimal Waveform Assignment

Here we will describe the space of possible outcomes, when each cognitive radar node in a network must select both a *center frequency*[4] and a *waveform*[5]. Specifically, the network seeks to select the actions resulting in the highest *utility*, when there is no communication between the independent nodes.

3.3.1 Center Frequency Selection

We can begin by defining an optimal configuration based on a pre-defined set of frequency bands. We assume that the center frequencies are selected such that two radars on two different center frequencies will have non-overlapping bandwidths. In each of many time steps t less than finite time horizon T, each radar r_i selects a center frequency f_j. Let the set of radars be denoted as $\mathcal{P} := \{r_1, r_2, \ldots, r_i, \ldots, r_M\}$ with $|\mathcal{P}| = M$. Similarly, let the library of center frequencies be $\mathcal{F} := \{f_1, f_2, \ldots, f_j, \ldots, f_N\}$ with $|\mathcal{F}| = N$. Let M and N be related as $M, N \in \mathbb{Z}$ s.t. $M < N$. Note that $\mathcal{P}$ and $\mathcal{F}$ do not have a time dependence.

Then, $\mathcal{P}$ and $\mathcal{F}$ form the two disjoint sets of nodes of a fully connected bipartite graph. The matrix of choices[6] becomes the set of edges $\mathcal{E} = \mathcal{P} \times \mathcal{F}$. We specify a fully connected graph to imply that any radar $r_i \in \mathcal{P}$ can select (and observe a reward for selecting) any center frequency $f_j \in \mathcal{F}$. Further we can call the graph $(\mathcal{P}, \mathcal{F}, \mathcal{E})$. Formalizing the problem in this manner allows for the examination of the space of possible outcomes, when one node selects one frequency band. We can analyze the utility of each possible configuration, to get a sense of the objective.

An edge is any connection between the vertices r_i, f_j of the bipartite graph. In this application, edges are equivalent to a radar r_i selecting center frequency f_j. Edges can be denoted as the concatenation of the vertices they connect - $r_i f_j$. We can assign each edge $r_i f_j$ a weight $u(r_i, f_j) := \mu_{i,j}$ with

$$u(r_i, f_j) : \mathcal{P} \times \mathcal{F} \to [0, 1] \tag{3.1}$$

which is the reward that radar r_i observes for selecting center frequency f_j.

A mapping $\widehat{\pi}(t)$ at time t is any collection of edges where no vertex in $\mathcal{P}$ is repeated. Mappings may or may not contain repetitions of vertices in $\mathcal{F}$. In other words, $\widehat{\pi}(t)$ can be any function which maps $\mathcal{P}$ to $\mathcal{F}$. If there exist $a, b \in \mathbb{Z}$ such that r_a and r_b are in the same mapping ($r_a f_j \in \widehat{\pi}(t)$ and $r_b f_j \in \widehat{\pi}(t)$), then we say that radars r_a and r_b have collided. For any time step t let $\mathcal{C}_{\widehat{\pi}(t)}$ be the set of colliding radars for a mapping $\widehat{\pi}(t)$. Formally,

[4] We use both terms *center frequency* and *frequency band* interchangeably to refer to the bandwidth allocated to a channel.

[5] The term *waveform* refers to the specific waveform selected from a library to be transmitted in a particular frequency band.

[6] By "matrix of choices" we mean the space of possible radar-frequency pairings.

Definition 3.1 (Collision)**.** Let $\mathcal{C}_{\widehat{\pi}(t)}$ be the set of colliding radars: $\mathcal{C}_{\widehat{\pi}(t)} := \{r_i \mid \exists\ r_i, r_k, \textit{ with } r_i f_j, r_k f_j \in \widehat{\pi}(t)\}$.
A radar r_i is said to *collide* at a time step t if $r_i \in \mathcal{C}_{\widehat{\pi}(t)}$.

A matching $\pi(t) : \mathcal{P} \to \mathcal{F}$ is any mapping with no common vertices[7]. So, in a matching, there are no collisions. We can collect all of the possible matchings $\{\pi\}$ of a graph $(\mathcal{P}, \mathcal{F}, \mathcal{E})$ into a set $\mathcal{M}$. Note that if $\pi(t)$ is a matching, it is also a mapping, while if $\widehat{\pi}(t)$ is a mapping, it is not necessarily a matching.

Remark 3.2. If $\pi(t)$ is a matching, the set of colliding radars $\mathcal{C}_{\pi(t)}$ is empty.

Note that we'll sometimes drop the time dependence of a mapping when we discuss the space of possible mappings. We can determine which matchings are best by looking at the sum of their combined rewards. We call this value the *utility* of the matching and define it as follows.

Definition 3.3 (Utility)**.** The *utility* of a mapping π is the sum of the rewards each radar observes using that mapping:

$$U(\pi) = \sum_{r_i f_j \in \pi} \mu_{i,j} \tag{3.2}$$

Now, adding time dependence, we can find the maximum utility in a step t as

$$U^*(t) = \max_{\pi \in \mathcal{M}} U(\pi(t)) \tag{3.3}$$

This leads naturally to a definition of an optimal configuration. This represents the highest-reward center frequency selection for each radar in a network for a given time step t.

Definition 3.4 (Optimal Configuration)**.** A set of radar nodes $\mathcal{P}$ with a common center frequency library $\mathcal{F}$ is said to be in an *optimal configuration* at time step t if $\pi(t) : \mathcal{P} \to \mathcal{F}, \pi(t) \in \mathcal{M}$ has utility $U^*(t)$.

Remark 3.5. Existence of an optimal configuration is guaranteed by definition, but uniqueness of the solution is not.

3.3.2 Orthogonal Waveforms

So far, we have defined optimality based on selection from a arbitrary waveform library. We will make this more concrete by discussing the use of *orthogonal waveforms* to develop a library, and justify the resulting SINR differences. Specifically we will refer to the SINR

[7] In other words, matching functions are injective.

post matched filtering, so that orthogonal interference does not impact the observations of each radar node.

The use of orthogonal waveforms in radar networks has been shown to aid in target tracking by obtaining more information through the differences in matched filter ambiguity responses [40]. Each node might have a different view of a target, local interference, or local scattering. Each of these could lead to differing local preferences for waveforms or frequency bands. This problem is somewhat simpler in the centralized setting, when a coordinator can specify a matching of orthogonal waveforms in each time step for the network. One limitation of the centralized case is that the central controller may not have sufficient information regarding local conditions. However, in this work we examine the decentralized setting. Since we have no communication within the network on the PRI time scale, there will be no means for the network to assign a different orthogonal waveform to each node for each PRI.

Two waveforms are defined as orthogonal when their time cross-correlation is zero. In radar applications, due to target motion and noise, waveforms that are orthogonal at transmission are not necessarily orthogonal at the receiver. Therefore it has been proposed to define *near-* or ε-orthogonality. Two waveforms are nearly orthogonal when their cross-ambiguity is less than ε [39]. Formally,

$$\max_{\tau, f_s} \frac{|\chi_{w_1,w_2}(\tau, f_s)|^2}{E_{w_1} E_{w_2}} \leq \varepsilon \tag{3.4}$$

where χ_{w_1,w_2} is the cross ambiguity

$$\chi_{w_1,w_2}(\tau, f_0) = \int_{-\infty}^{\infty} w_1(t) w_2^*(t-\tau) e^{i2\pi f_0 t} dt \tag{3.5}$$

and E_{w_i} is the energy of waveform w_i. This can be calculated for either the transmitted or received energy of waveform w_i.

Intuitively, very different signals will have low cross-ambiguity. This holds for both temporally shifted (TDMA) and frequency shifted (FDMA) signals, as well as signals that are coded to be orthogonal (CDMA). However, we will not consider TDMA for the following reasons.

Consider a network using TDMA waveforms with equal PRFs. Then, every PRI, there would be a chance of receiving mutual interference at one or more time delays. Conceptually, the duration of a PRI could be increased to $M\times$PRI to allow TDMA where there are M radar nodes. This would be very inefficient and also result in limited maximum detectable range. If any of these align with target returns, then the target will not be resolvable.

Consider next a network using TDMA waveforms with coprime PRFs. Then, each radar would be unable to shift its PRF with high fidelity to optimize target tracking. This approach also assumes some sense of PRF coordination network-wide. Since PRF should be linked to target radial velocity, this places a large constraint on target tracking ability.

Due to these reasons we will not consider TDMA for waveform selection. Instead, we will

assume a library of Orthogonal Frequency Division Multiplexing (OFDM) waveforms in each frequency band. We specify two main design constraints on this library:

1. Mutual collisions must remain detectable (under criteria described later).

2. The library must contain a waveform which is orthogonal to a single narrow-band interferer in one of 2^s sub-bands[8].

We make the first constraint due to MMAB algorithms requiring collision information: if more than one node selects the same frequency band, they must detect it. Secondly, we model each frequency band as containing at most one narrow-band primary user such as a communications system. This information is not known a priori, so part of the learning problem is understanding where the primary user is since a waveform cannot be orthogonal to interference that is unknown in advance. We will detail a method for this in a following section.

Assume a set $\mathcal{H}$ of s sub-bands, enumerated $h_1, h_2, \ldots, h_s$ with s even. Further, let each sub-band h_i be contiguous with h_{i+1}. Let waveforms w_j, $j \leq s+1$ occupy sub-bands $H_j \subseteq \mathcal{H}$, where

$$H_j = \begin{cases} \text{maxB}\left(\mathcal{H} \backslash h_j\right), & j < s+1 \\ \mathcal{H}, & j = s+1 \end{cases} \tag{3.6}$$

The backslash $\mathcal{X}\backslash x$ denotes set subtraction. Let $\text{maxB}\left(x\right)$ be the function which returns the largest-bandwidth contiguous set of sub-bands.

As an example, when $s = 4$, $\mathcal{H} = \{h_1, h_2, h_3, h_4\}$. Waveform w_1 will use sub-bands $H_1 = \{h_2, h_3, h_4\}$ and w_2 will use sub-bands $H_2 = \{h_3, h_4\}$. This method provides a set of waveforms which can be orthogonal to an interferer in any given sub-band, without requiring notched waveforms.

Note that if two radar nodes were to use nearly orthogonal waveforms, *there may still be unacceptable levels of interference* due to the low power echo received from the target, relative to the direct signal from the second radar (even with low sidelobes). In addition, any non-synchronous reception will destroy the orthogonality.

So far we have had no discussion of collision detection. How are radar nodes able to sense when a collision has occurred? We will examine this in the following section.

[8]The term sub-band refers to subdivisions of the selected frequency band.

3.3.3 Collision Detection

We would like for each node i, which picks waveform w_j in PRI p to form an estimate of $\mathbb{1}_{\mathcal{C}_{\mathcal{W}_p}}(w_j)$, where

$$\mathbb{1}_{\mathcal{C}_{\mathcal{W}_p}}(w_j) = \begin{cases} 1, & w_i \in \mathcal{C}_{\mathcal{W}_p} \\ 0, & \text{else} \end{cases} \tag{3.7}$$

with $C_{\mathcal{W}_p}$ being the set of colliding waveforms. So, let I be an estimate as

$$I = \widehat{\mathbb{1}}_{\mathcal{C}_{\mathcal{W}_p}}(w_j). \tag{3.8}$$

Recall the radar equation:

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R_t^2 R_r^2} \tag{3.9}$$

where P_t is the transmission power, G_t is the transmitting array gain, G_r is the receiving array gain, λ is the wavelength corresponding to a center frequency of f_c, σ is the radar cross section (RCS) of the target, R_t is the distance from emitter to target, and R_r is the distance from target to receiver. Note that in a mono-static scenario such as the one we consider, $G_t = G_r$ and $R_t = R_r$ for a given radar node. We can compare the radar equation, which describes the power received from a target, to the power received from an interfering radar (or other) transmission at a range of R_n:

$$P_I = \frac{P_t G_t(\theta) G_r \lambda^2}{4\pi R_n^2} \tag{3.10}$$

where $G_t(\theta)$ represents the interfering radar transmit gain in the direction of the radar of interest. We can compare the received power from both types of source, assuming the same distance and gain characteristics, and an RCS of $3m^2$, which is characteristic of a medium aircraft [104]. We assume for this example that the radar node operates with an output power of 30dBw. Specifically, we will compare three levels of received power. First, the power received from a main-beam target reflection. Second, from the sidelobe (at a relative -10dB from the main beam) of a second radar node, and third from an outside interferer with an output power of 10dBw. In Fig. 3.2, we can see that a radar node will receive much higher power from another node than it would from returns of its transmitted waveform or from interferers.

The lowest-power mutual interference is still more powerful than most radar returns. If we consider main-beam interference instead, the effect would be even more evident. So, we propose the use of a received power threshold to determine when collisions occur, given some small limitations. The remainder of this paper considers distances between 1 and 3km.

If in a given PRI, an individual radar receives a return composed of a target reflection along the main beam AND high-power mutual interference at an angle θ, we can represent the

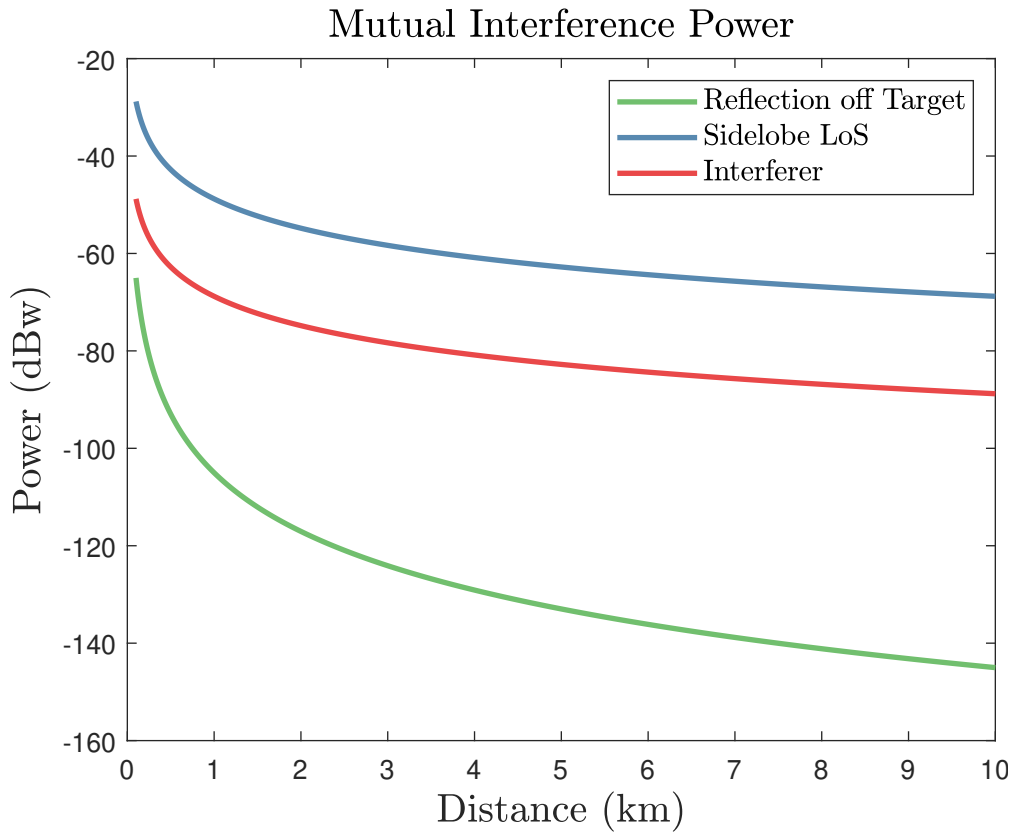


Figure 3.2: Received power over a range of distances for a target, sidelobe LoS received from another node in a network, and LoS interference.

received power as:

$$P_{r,I} = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R_t^4} + \frac{P_t G G(\theta) \lambda^2}{(4\pi)^2 R_n^2} \tag{3.11}$$

if the mutual interference is totally orthogonal to the radar return. Then, assuming that $\frac{\sigma}{4\pi R_t^4} < \frac{1}{R_n^2}$, we can see that

$$\frac{P_t \lambda^2 G G(\theta)}{(4\pi)^2 R_t^2} \leq P_{r,I} \leq \frac{2 P_t \lambda^2 G G(\theta)}{(4\pi)^2 R_t^2} \tag{3.12}$$

Note that this holds even when the mutual interference and radar return are not completely orthogonal due to the inequalities. Using this as an energy detector, we are interested in PRIs when the received energy exceeds

$$\int_t^{t+PRI} \frac{P_t G^2 \ \lambda^2}{(4\pi)^2 R_n^2} dt \tag{3.13}$$

where a PRI lasts from t to $t + PRI$.

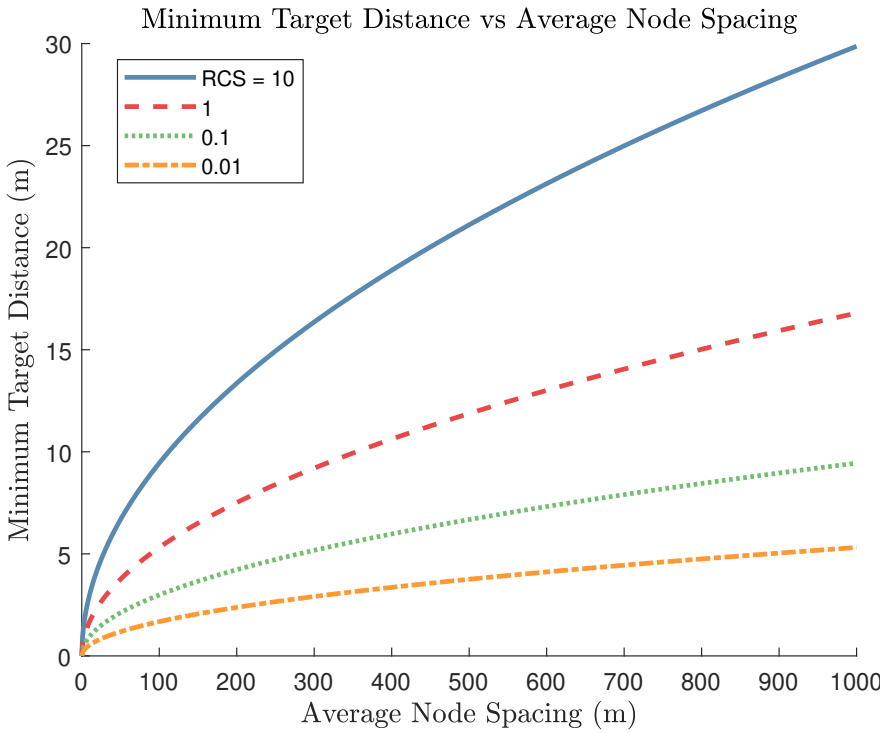


Figure 3.3: A comparison of the distances needed for the collision detection assumption to hold, given various RCS values.

Then, two issues remain before we can define a collision detector. First, how do we determine the value of R_n, the distance to the nearest node? In the scenario where each radar node is aware of the network positions, this is trivial. However, in the general case, when the network is simply distributed as a point process with some density, we can determine the average distance to the nearest node.

Specifically let the radar network be spatially distributed according to a Binomial Point Process (BPP) with intensity γ. BPPs are spatial distributions with a fixed number of points $N(t) = n$. Poisson Point Processes (PPPs), on the other hand, are spatial distributions with $N(t)$ drawn from a Poisson distribution. A BPP is equivalent to a PPP conditioned on $N(t)$. Then, the nearest neighbor distribution function is given as $g(r)$ [69]. Let the inter-node distance $R_n = \mathbb{E}[g(r)]$ be the expectation of the nearest neighbor distribution.

Secondly, we need to support the assumption that $\frac{\sigma}{4\pi R_t^4} < \frac{1}{R_n^2}$. In Fig. 3.3 we can see that for a range of RCS values and average node spacings, the minimum target distance needed for this assumption to hold is consistently low, and is sublinear with average node spacing.

We can write the final energy detector as

$$I(t) = \begin{cases} 1, & E_I(t) \geq \int_t^{t^*} \frac{P_t G^2 \lambda^2}{(4\pi)^2 \mathbb{E}[g(r)]} dt \\ 0, & else \end{cases} \tag{3.14}$$

Note that we discuss collisions only in terms of *mutual interference*. We consider instances of outside interference to primarily impact the observed reward, and do not consider these as collisions. Since outside interferers are assumed to transmit at substantially lower power than each radar node, this assumption is justified.

3.3.4 Band and Waveform Selection

We can extend the decision-making to two discrete levels by parameterizing the waveforms by an index of $\mathcal{W}$, the library of OFDM waveforms, and an index of $\mathcal{F}$ which we define to be the set of available center frequencies. Then, the action comes from the space $\mathcal{F} \times \mathcal{W}$. Specifically we will develop the use of a multi-player bandit algorithm for center frequency selection, then delegate the choice of a waveform to a single player bandit, with independent instances for each choice of center frequency.

There are two primary motivations for this dual structure. First, the outer multi-player bandit algorithm will be able to learn the average behavior of each sub-band, while simultaneously learning to select center frequencies from an optimal matching. Second, the inner single-player bandit algorithms have the ability to tailor the waveform to avoid intra-band interference, causing performance to improve over time.

We will assume a fixed maximum bandwidth for all radars, with a variable center frequency. Each radar chooses an action from the space $\mathcal{F} \times \mathcal{W}$, with $\mathcal{F}$ chosen so that there is no overlap in frequency when two different center frequencies are chosen. In other words, it selects a waveform from the library $\mathcal{W}$ and a center frequency from $\mathcal{F}$ to transmit that waveform.

3.3.5 Algorithms

Varieties of the proposed sequential decision-making problem has been studied in cognitive radio, as well as in cognitive radar [56], [20]. The approach in this work employs a multi-player multi-arm bandit framework to allow nodes to develop a model of their environment, without relying on a central decision-maker [55]. This means that each node can only access the information it has observed: rewards for each waveform selected, and an indicator for collisions. Since we are assuming a model with no dedicated communication between nodes, and only limited exchanges of information from each node to the fusion center, each node needs to execute independent algorithms that are capable of converging to the optimal utility for the entire network. There is abundant work in the literature on the topic of Multi-Arm Bandit (MAB) models for agents which make sequential decisions over a finite time horizon T [31, 65, 105]. In addition, this model is a two-layer algorithm as described above: a multi-player algorithm to determine center frequency, and a single-player algorithm to select waveforms.

Multi-Player Algorithms

MABs consider the sequential interaction between a player and an environment, which consists of multiple "arms". In each of finitely many time steps, the player selects one arm and observes a reward generated by the environment for that arm. Over time the player must balance actions which *explore* versus those that *exploit*. Exploring actions are those taken to generate a better understanding of the environment. Exploitative actions are those that take advantage of prior knowledge of the environment to maximize rewards [65]. Rewards are assumed by each player to be drawn i.i.d. from unknown distributions. Over time the player seeks to learn the mean of the distribution for each arm, which we denote as $\widehat{\mu}_f^r$ for player r's estimate of the mean of arm f. So, a naive strategy might have the player attempt each arm once then select the one with the highest reward for the rest of the game. This only works if 1) the arm variance is very low, so the arm ordering can be correctly estimated from only one sample, and 2) the arm means do not vary over time. In realistic scenarios neither of these are likely to be true. For instance, in this scenario, rewards are influenced by interferer behavior.

Multi-player MAB (or MMAB) models consider the interactions between many players which must use the same set of arms. As in the framework described above, this allows two players to select the same action and collide, since if multiple players see the same "best" arm, they will all desire to use it. So, it is clear then that MMAB models are well-suited to the problem of distributed radar: Multiple radars must select sequential actions while seeking to minimize target tracking error across a network.

We choose a structure that allows a MMAB algorithm to select the center frequency $f \in \mathcal{F}$, and a "single-player" algorithm to select the waveform $w \in \mathcal{W}$ so that the resulting action is in the space $\mathcal{F} \times \mathcal{W}$. To be clear, the goal for each node is twofold: 1) end up in a frequency band with no other nodes and which maximizes *network* SINR, and 2) select waveforms in that frequency band which maximize SINR. We will discuss the use of Sense and Avoid for both center frequency and waveform selection. This will serve as a baseline for comparison, followed by a discussion of MMAB algorithms such as Musical Chairs for center frequency selection with single-player MAB algorithms such as ϵ-greedy for waveform selection. In this structure, the goal for the network is for each radar node to learn to select a center frequency which is different from each other node, as well as waveforms which attain high SINR in the selected band. Collisions are observed via the technique described above.

While each radar node only implements a single instance of the center frequency selection algorithm, it will implement a waveform selection algorithm for *each center frequency*. This structures the algorithm such that interference near one center frequency does not influence waveforms selected near another center frequency.

In the following section when we detail a MMAB algorithm we will denote the actions as $f \in \mathcal{F}$, while if we discuss a single-player algorithm for waveform selection we will use notation $w \in \mathcal{W}_f$. If we drop the frequency band dependency on the waveform library

it is understood that we're referring to the general case of a waveform selection algorithm instantiated in some frequency band. For the particular case of Sense and Avoid in the following section, we denote the actions as waveforms $w \in \mathcal{W}$ but also discuss its use for frequency band selection.

Remark 3.6. We consider only online-learning approaches, since the described network has no prior knowledge of the environment. Due to short coherence times, variable targets, and possibly changing rewards, each radar must learn the behavior of the environment, which includes rewards, interferer behavior, and target behavior.

Sense and Avoid The simplest algorithm we will consider, called "sense and avoid" (Algorithm 3), has each radar select a random action and observe for collisions. When collisions happen, they select a new action. Otherwise they keep repeating the previous action. Assuming no outside interference, this will result in a reward matching, but it will not necessarily be optimal, since the algorithm does not consider which actions may be better for the radar. This allows SAA to use $I(t)$ to determine when to switch actions. Note that in this instance we can consider a lower threshold for collisions which includes the event where a pulse overlaps with a primary user.

Algorithm 3: Sense And Avoid

Result: w(t)
if $\overline{I(t)}$ **then**
  $w(t) = w(t-1)$;
else
  $w(t) = \mathcal{U}\{\mathcal{W} \backslash w(t-1)\}$;
end

Recall that $\mathcal{U}\{\cdot\}$ represents uniform sampling over a set, and the backslash represents the set difference. In simulation, we will use SAA as a center frequency selection algorithm *as well as* a waveform selection algorithm.

Musical Chairs The "Musical Chairs" (MC) algorithm [106], shown in Algorithm 4, is a step up in complexity from SAA. MC develops an estimate of $\mathcal{W}^*$, the set of best center frequencies, by specifying a well-defined exploration period where every player attempts to observe the reward for each action as many times as possible while avoiding collisions. Since there is no coordination the regret incurred during this exploration period is rather high, as collisions will be unavoidable. The exploration continues for T_0 time steps, where

$$T_0 = \left\lceil \max\left(\frac{16M}{\epsilon^2} \ln\left(\frac{4M^2}{\delta}\right), \frac{M^2 \log(\frac{4}{\delta})}{0.02} \right) \right\rceil$$

is chosen to guarantee a regret bound. Here, M represents the number of players, ϵ is a bound on the correctness of the estimate of $\mathcal{W}^*$, and δ represents the distance from the M^{th} best reward to the $(M+1)^{th}$ best reward.

Algorithm 4: Musical Chairs

Result: $w(t)$
Input $y_i(t), c_i(t), fixed = \text{False}, t = 0$

```
if not fixed then
    if t ≤ T_0 then
        w(t) = U{F};
    else if t > T_0 then
        if I(t-1) then
            w(t) = U{W*};
        else
            w(t) = w(t − 1);
            fixed = true;
        end
    end
else
    w(t) = w(t − 1);
end
```

Musical Chairs Top M The "Musical Chairs Top M" (or MCTopM) (Algorithm 5) [30] algorithm was designed to allow all players time to explore through the use of the Upper Confidence Bound [107] which we represent as $g(t)$.

$$g(t) = \widehat{\mu}_f^r(t) + \sqrt{\frac{\log(t)}{2T_w^r(t)}} \tag{3.15}$$

In each round, the player will attempt to pick one of the best M center frequencies (contained in the set $\mathcal{W}^*$), where there are M players. If the player successfully selects one of these actions, it is marked as a "chair" and the player will select it so long as it remains in $\mathcal{W}^*$. If a player collides on action $w \in \mathcal{W}^*$, a new action from $\mathcal{W}^*$ will be selected. If an action is marked as a "chair" but is no longer in the $\mathcal{W}^*$, the player will use the UCB values to select a new action from $\mathcal{W}^*$. One interesting feature of MCTopM is that by virtue of the UCB, the player will tend not to change actions too frequently. In a realistic radar context this will be beneficial since there is a cost associated with changing waveforms too often.

Algorithm 5: Musical Chairs Top M

Result: w(t)
if $\overline{w(t-1) \in \mathcal{W}^*}$ **then**
  $w(t) = \mathcal{U}\{\mathcal{W}^* \cap \{w_i : g_{w_i(t-1)} \leq g_{w(t-1)}(t-1)\}\}$;
else if $I(t-1)$ *&* $\overline{fixed}$ **then**
  $w(t) = \mathcal{U}\{\mathcal{W}^*\}$;
else
  $w(t) = w(t-1)$;
  fixed = true;
end

Single-Player Algorithms

Traditional bandit algorithms only consider the behavior of an individual learner, sampling from some set of actions with the goal of minimizing cumulative regret. As discussed, we utilize the MMAB algorithm to select a frequency, followed by one single player bandit algorithm *per center frequency* which learns the interferer behavior (and therefore rewards) in that band. Many algorithms exist in the literature for this problem [65], and we discuss two. In addition we can use the above SAA algorithm in the single player bandit role.

ϵ-Greedy In addition to SAA, the second algorithm we will consider for waveform selection simply selects a random action with some constant probability ϵ, and selects the highest-average-reward action with probability $1-\epsilon$ [108]. This allows the learner to explore different actions with some fixed probability over time, while attempting to ensure low regret by selecting the highest-reward action the rest of the time.

ϵ-Decaying As a variation on ϵ-Greedy, ϵ-Decaying [108] selects an ϵ value as a function of the number of trials. As a heuristic we found that setting $\epsilon = \frac{1}{t^{0.8}}$ exhibits the best performance in this setting. Fig. 3.4 shows that this value is a local minimum in a range of possible exponents.

3.3.6 Rewards

Clearly, the choice of rewards influences which matchings will have optimal utility. Rewards are typically drawn from i.i.d. Gaussian or Bernoulli distributions, although *no assumption on the type of distribution is needed*[9]. As we discussed earlier, different radar waveforms can

[9]This is because each radar only needs to consider the mean of the distribution, which can be determined with confidence given enough data.

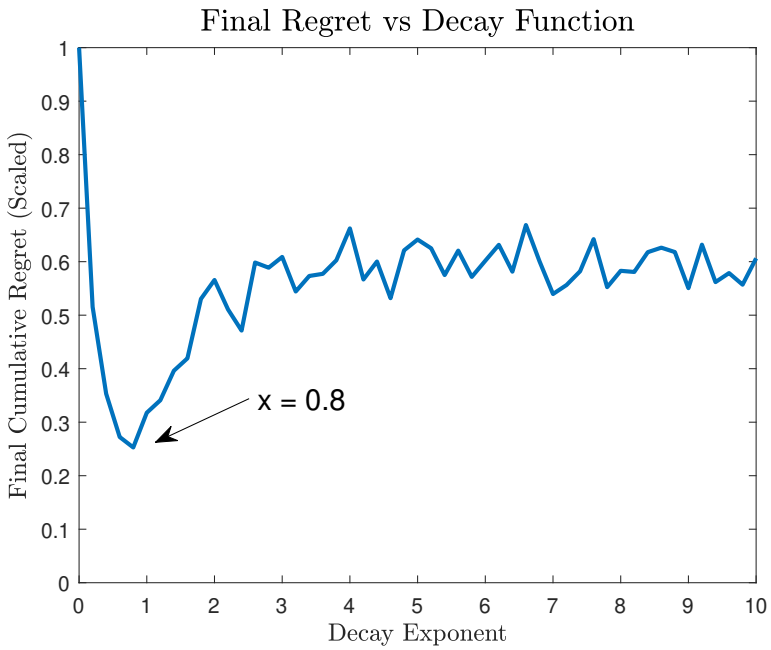


Figure 3.4: Final regret over a simulation of 100 CPIs with three radar nodes using MCTopM and ϵ-Decaying, using a decay exponent described on the x-axis.

be used to maximize different target information.

Previous work in reinforcement learning for radar [109] has shown that a reward function based on a weighted combination of SINR and bandwidth improves detection accuracy. Since each radar in this network will have a fixed bandwidth, we'll use a reward function that encourages high SINR, and discourages collisions:

$$\mu_{i,j} = \begin{cases} \alpha\left(\text{SINR}_{i,j} + \beta\right), & I_i(t) = 0 \\ 0, & I(t) = 1 \end{cases} \tag{3.16}$$

where α and β are parameters used to map the SINR observed by radar i selection action j roughly to the unit interval. $I_i(t)$ indicates whether or not radar i experienced a collision. Note that while the SINR appears dependent on both radar and action choice, it is actually only dependent on the action choice j, with the radar dependence indicating which radar selected that action.

In simulations, each node samples $\text{SINR}_{i,j}$ from a normal distribution with mean $\frac{\mu_{i,j}}{\alpha} - \beta$.

Regret

MABs have convenient theoretical guarantees on parameters such as regret, which is the difference in reward between actions an agent took and the actions that some oracle with

perfect knowledge of rewards would take. Regret comes in several flavors. Specifically, we will discuss the so-called "weak" regret, which compares actions to the best (on average) actions. Regret is defined for the single radar as Eq. (3.17), where μ^* is the reward of the optimal action.

$$R_t(r) = t\mu^* - \sum_{t_0=1}^{T} \mu(t_0) \tag{3.17}$$

Regret is also defined for a group of players. Formally, after any PRI $1 \leq t \leq T$

$$R_t = tU^* - \sum_{t_0=1}^{T} \sum_{r \in \mathcal{P}} \mu_r(t_0) \tag{3.18}$$

is the cumulative regret at the time horizon T for a group of radars $\mathcal{P}$. Since the reward function Eq. (3.16) penalizes low SINR and is maximized by high SINR, low regret will also correspond to good radar tracking performance.

Note that regret only has meaning if the number of available actions is greater than one.

Finally we can write the *average cumulative regret* in a time step $t \leq T$ as

$$\overline{R}_t = \frac{R_t}{t} \tag{3.19}$$

We use Eq. (3.19) in the results shown below.

3.3.7 Performance Analysis

Given the structure of this environment, we can make some claims. First, we need to define the *network* SINR.

Definition 3.7 (Network SINR)**.** The network SINR_π is the sum of the SINR at each node i, written as $\text{SINR}_{i,j}$ which select actions according to the policy π.

$$\text{SINR}_\pi = \sum_{r_i, f_j \in \pi} \text{SINR}_{i,j} \tag{3.20}$$

Lemma 3.8. *Maximal utility implies maximum network* SINR*.*

Proof. See Appendix A. □

Since high SINR corresponds to low estimation variance, it's clear that our reinforcement learning algorithm will result in the best-case radar tracking performance for this environment, assuming that the algorithm is capable of converging to the optimal matching.

Table 3.1: Simulation parameters, unless stated otherwise.

Parameter	Value	Parameter	Value
Number of Radars	3	Number of Targets	1
PRIs per CPI	400	Target Initial Position	[400,400] m
Total CPIs	50	Bandwidth	20MHz
Typical SNR	12 dB	Averaged Simulations	50
Frequency	2.4GHz	PRI Duration	1.024×10^{-4}

3.4 Simulations

We will investigate the performance of a radar network which tracks a moving object. Each radar is stationary and uses a combined center frequency and waveform selection algorithm (which we specify) to estimate target position, Doppler, range, and angle. The individual node locations in the network are modeled as a Binomial Point Process, with a fixed number of nodes placed randomly in the disc centered at $[500, 500]$m with a radius of 500m.

Each radar implements a two-dimensional Kalman filter, estimating position $\hat{x} = [x, y]$ and velocity $\hat{v} = [v_x, v_y]$. At the end of each CPI, each radar conducts processing on the returns it observed, updates its Kalman filter, and passes the updated track to a fusion center. At the fusion center, the position estimates are averaged using equal weights. Table 3.1 lists common simulation parameters.

Interference. We model interference through the reward function. Each frequency band is assumed to have some interference to noise ratio, with a primary user in one sub-band. Waveforms which overlap with this primary user will experience lower SINR than those which do not. Since we'd like to use as much of the available bandwidth as possible to maximize target information, we'll add a term to the reward equation which penalizes waveforms, inversely proportional to their bandwidth.

In the environment considered here, single-node full-band radars will always be outperformed by networks of multiple radar nodes, and static allocation networks will be outperformed by those using coexistence strategies. To the latter point, we first present the regret performance shown in Fig. 3.5. This is a comparison of a network using fixed allocations for frequency and waveform, and one using SAA for waveform selection. We can see that the network using sense and avoid for waveform selection is able to obtain far less regret than that using a static allocation, however, the regret is linear since SAA is not optimal. This performance corresponds to the radar tracking error shown in Fig. 3.6, where the network using SAA has far better performance.

Next, we can look at the benefit of increasing the number of nodes in the network. Each network uses the MCTopM algorithm for center frequency selection and ε-Decaying for waveform selection. Fig. 3.7 examines the difference in *average* cumulative regret Eq. (3.19).

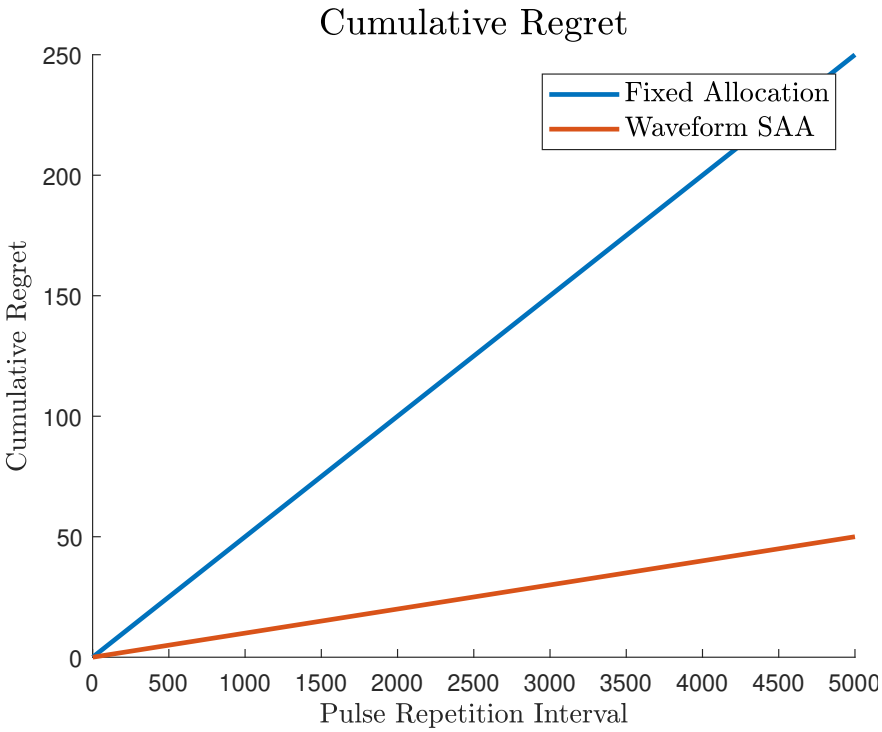


Figure 3.5: Cumulative regret for two different configurations: the first network of two radars only uses fixed allocations for center frequency and waveform selection, while the second network of two radars uses a fixed allocation for center frequency selection and SAA for waveform selection.

The networks considered consist of two, three, and four nodes. We can see that on average, each node in each of these networks attains the same amount of regret. In addition, it's clear that the average cumulative regret is asymptotically going to zero; this means that the center frequency and waveform selections are optimal by roughly the 100^{th} PRI.

Further, in Fig. 3.8, we see that the *tracking error decreases with network size*. This is because even though the actions being selected in each network approach the optimal, the averaged tracking estimates over greater network sizes improve.

Figs. 3.9 and 3.10 show the benefits of incorporating different cognitive strategies. We can see that using any strategy with MC or MCTopM outperforms the fixed-allocation or waveform SAA strategies shown in Fig. 3.6. This is because even while several radar nodes may be forced into using sub-bands or waveforms with low SINR, the network as a whole can mitigate these errors. Note also that the average cumulative regret 3.7 informs the tracking performance Fig. 3.8. Those strategies that do not asymptotically approach zero in average cumulative regret will tend to have higher tracking error than those which approach zero. The best-performing algorithm combination can be seen to be MCTopM coupled with ε-Decaying, which is as we would expect. The UCB-informed MCTopM is able to balance

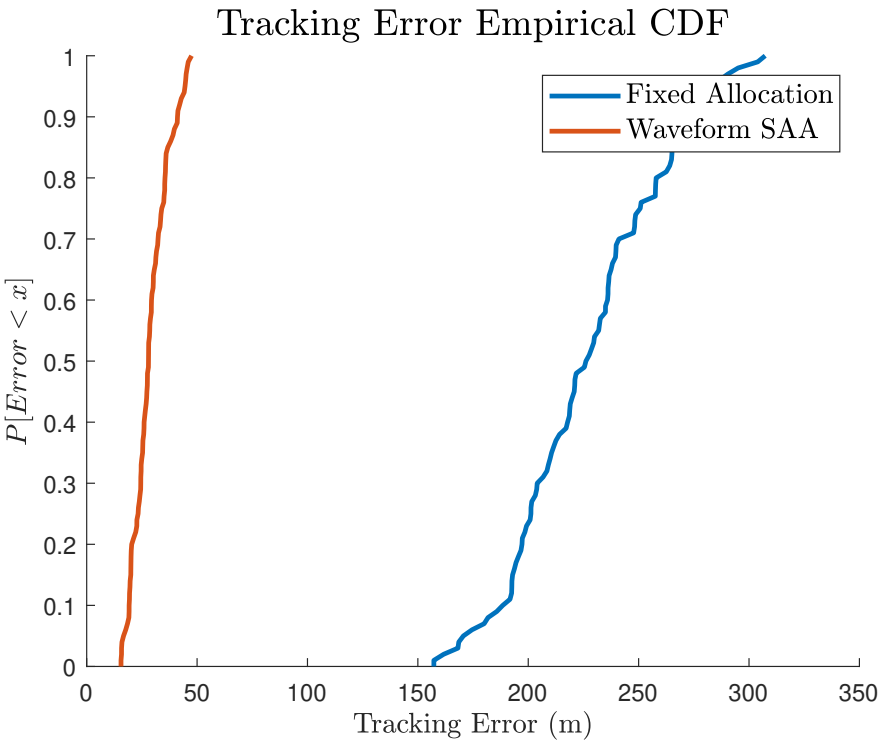


Figure 3.6: Tracking error for two networks of two radars each. The second network, using SAA for waveform selection, outperforms the network using no intelligent selection.

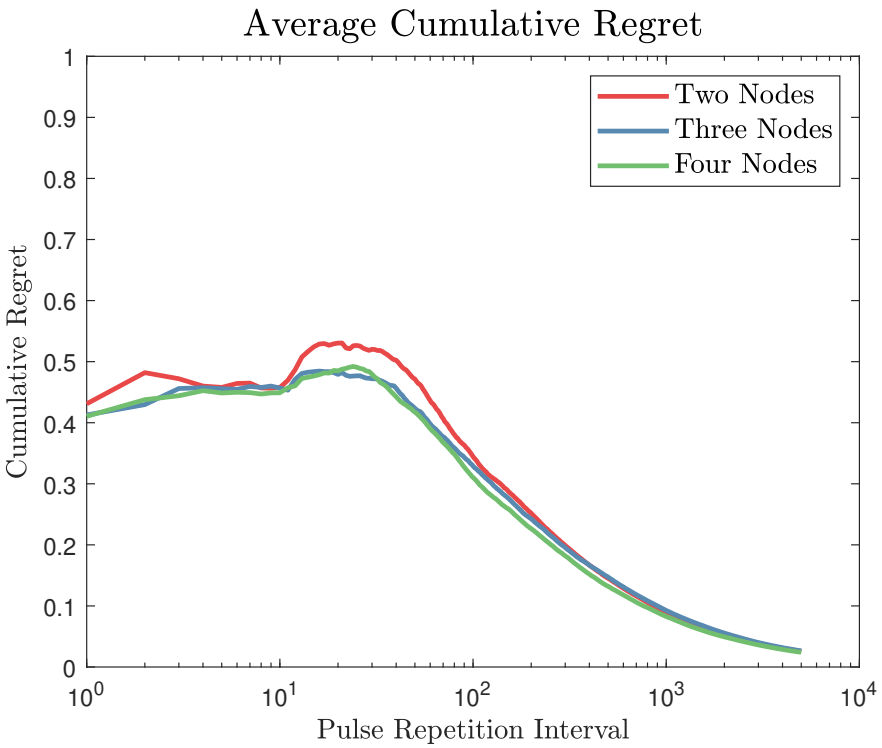


Figure 3.7: Network regret for networks of two, three, and four radars. Since the average regret *per radar* does not increase with the network size, we can see that there is no impact to the learning problem from additional radar nodes.

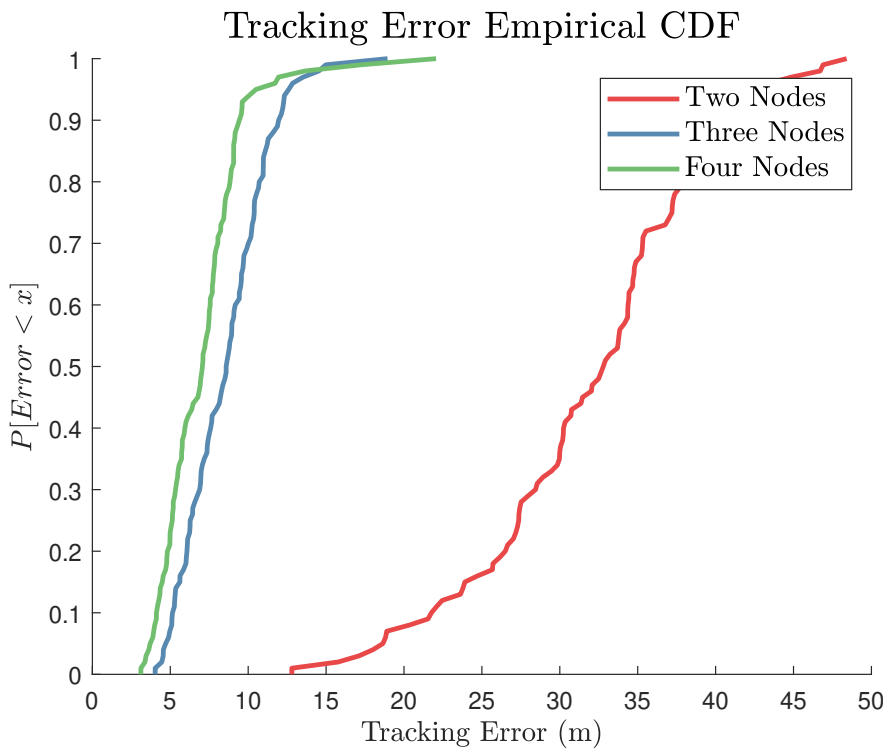


Figure 3.8: Network tracking performance for network sizes of two, three, and four radars. The improvement from three to four radars is not as pronounced as from two to three, since the observation quality of the fourth radar is less than the others due to the environment configuration.

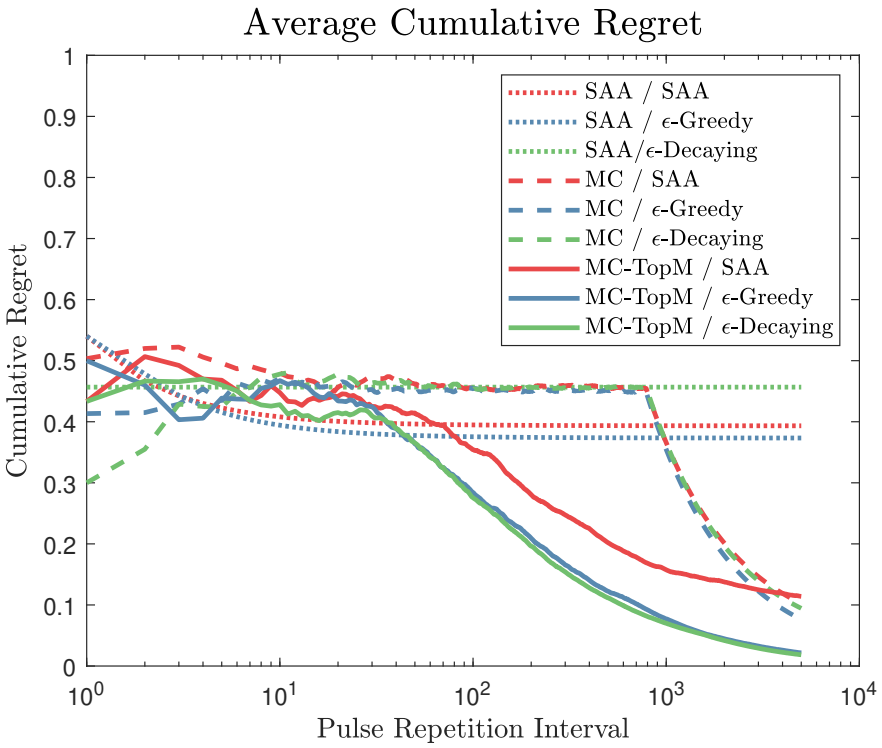


Figure 3.9: Average cumulative regret for nine different CRNs. Each network has three nodes and employs a two-step algorithm as defined above. Networks employing SAA for frequency selection tend to converge to a sub-optimal solution, while those using MCTopM tend to converge towards the optimal. Similarly, any network using SAA for waveform selection will not obtain optimal performance, and E-Decreasing will have lower regret in each time step.

exploration and exploitation while avoiding mutual interference, and ε-Decaying causes the waveform selection algorithm to converge early in the game.

3.5 Conclusions

We have seen that radar networks using any amount of cognition for center frequency and waveform selection will outperform pre-allocation techniques, even with equivalent total bandwidth and power distributed through the network. A major condition for the proper functionality of a radar sensor network is the selection of non-overlapping frequency allocations, which we define as optimal configurations. We accomplished this through the use of Multiplayer Multi-Armed Bandit algorithms, which frame this as an iterative decision-making problem. We provided a system model which we used to further define a method for the detection of *collisions*, which are instances of two cognitive radar nodes using the same frequency band.

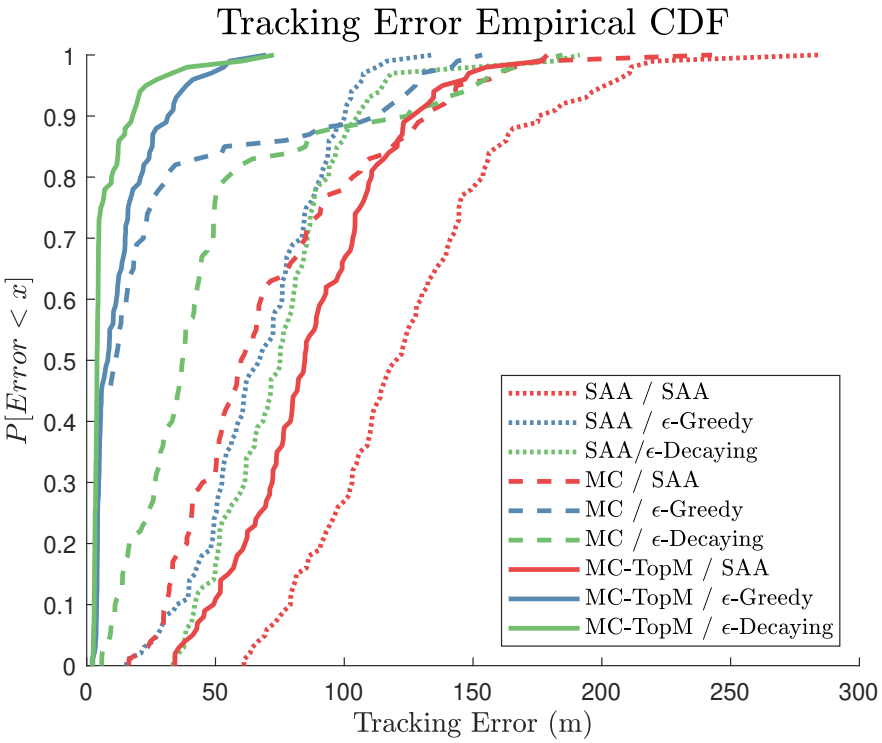


Figure 3.10: Radar tracking error for these algorithm combinations. As indicated by the regret performance, any networks using SAA for frequency or waveform selection will be outperformed by those using MMAB strategies. The regret performance and specifically the earlier convergence of MCTopM / ε-Decaying allow this combination to obtain lower error on average than other algorithms.

In addition, we described a method of using *single*-player bandit algorithms to accomplish outside interference avoidance. We did this by providing an instance of a single-player bandit algorithm to each choice of center frequency, and using the same SINR-based rewards as used for the MMAB frequency selection algorithm. The goal of the single-player bandit was to choose from a library of orthogonal waveforms to, over time, avoid any interferers in the environment and obtain the best SINR possible.

All of the algorithms described above operate in a *decentralized* manner, meaning there is no need for information from a central coordinator to make decisions. All decisions are made locally to each radar node which reduces communication overhead as well as improves the redundancy of the system to loss of the coordinator.

We demonstrated that the CRN using a combination of MMAB and single-player bandit outperforms SAA as well as the static, pre-allocated case. Specifically, MCTopM for center frequency selection paired with ε-Decaying is shown to obtain the lowest regret bound, which we prove corresponds to optimal asymptotic radar tracking performance.

Future work will include a stochastic treatment of the collision detection problem. While in this work we use the Nearest Neighbor distribution function of the BPP to estimate the power received from LoS mutual interference, a more rigorous analysis could be accomplished. Specifically we assume that the radars point in uniformly random directions in each time step while developing the collision detection algorithm, while in reality each node may be able to form a prior distribution on the pointing of other nodes given a target's estimated position. This could be used to inform a collision detection algorithm.

In addition, in this work we assume that all decision-making can be accomplished on the order of a PRI, while in many radar implementations, this and all other processing occurs on the CPI scale to preserve coherence pulse-to-pulse. While there exist methods to do coherent processing on pulses which vary inside the CPI, these remain somewhat impractical. Future work may examine the impact of reducing all decision-making to the CPI scale, and analyzing the consequences. The current algorithms would be hampered by this change, as observations of the environment are assumed to immediately follow the previous action, and immediately precede the next action. In addition more realistic radar scenarios such as clutter-limited environments and multiple targets may be considered.

Chapter 4

Hybrid Cognition: Balancing Cognition with Communication

Related Publications

The material in this chapter has been reproduced from the following publications:

- [52] **W. W. Howard** and R. M. Buehrer. "Hybrid Cognition for Target Tracking in Cognitive Radar Networks," in *IEEE Trans. on Radar Systems*, pages 118-131, 2023. doi: 10.1109/TRS.2023.3282846.
- [57] **W. W. Howard** and R. M. Buehrer, "Decentralized Bandits with Feedback for Cognitive Radar Networks," in *2022 IEEE Military Communications Conference (MILCOM)*, Rockville, MD, USA, 2022.

4.1 Introduction

This work seeks to improve the learning rate of a cognitive radar network (CRN) by introducing a central coordinator (CC) to provide limited feedback. Specifically in this work we address the role of a central coordinator within a cognitive radar network using an online learning strategy to achieve coordination as well as optimize radar tracking and spectrum sharing performance. Generally, radar networks achieve superior tracking performance than is possible for a single high-powered radar node [110]. This is due in part to the increased spatial diversity [111] and spectral agility [25]. Distributed nodes can cover a greater area to perform detection, and can exploit more spatial degrees of freedom to more accurately estimate target parameters. However, to obtain this superior performance, the individual radar nodes which comprise the radar network must coordinate with each other to efficiently use the available spectrum and avoid causing harmful interference inside or outside the network. At the root, this problem is caused by a fundamental need to both explore the available channels and simultaneously exploit the best channels (in terms of tracking performance).

Fixed, rule-based coordination has been proposed to solve this problem [112] [113]. This works well when the scenario parameters are well-known *a priori*, but can suffer poorer performance when the scenario is more unpredictable (i.e., unknown targets or interference).

Among other things, the desire for more flexible and adaptable systems motivated the initial research into cognitive radio and radar. Cognitive systems, at the core, are defined as possessing the ability to monitor the environment and modify operating parameters towards a goal [6]. Further, Haykin provides the following dichotomy of cognitive networks:

1. **Distributed Cognition**, where observations from individual nodes are combined at a *fusion center*[1] but no feedback is provided to the nodes.

2. **Centralized Cognition**, where a *central coordinator*[2] is the only cognitive agent, collecting observations from each node and dictating future actions.

Whether distributed or centralized, cognitive systems tend to be online learners due to the necessity to specialize to new, unknown environments and the difficulty of training the network ahead of time for an unknown environment.

Fully distributed cognition [51] [84] is useful when there is a desire for the parts of a CRN to be entirely disjoint and independent. Fully distributed approaches rely on consensus techniques [114] [115] to exchange information between the parts of a network and to determine optimal actions. In the radar context, these techniques can be very slow (requiring 10^4 or greater time steps to converge to optimal actions) and can cause a large amount of mutual interference. Such a convergence rate can be problematic in some settings.

Centralized cognition is not without trade-offs either. When cognition is limited to the CC, the individual nodes become over-reliant on the CC. The feedback costs can also grow immense, as we will show.

This work investigates *hybrid cognition*, seeking the minimal amount of feedback necessary in a CRN to obtain near-optimal radar tracking performance in a short time without sacrificing node-level cognition.

Our previous work [55] [51] considered *strictly* decentralized techniques, and did not assume the presence of a CC. While our approach was effective under these circumstances, the technique resulted in a relatively slow convergence rate. Our current work provides a generalization to model the CRN as containing a CC which can communicate and provide feedback to the radar nodes, with a goal of speeding up convergence.

4.1.1 Contributions

This paper makes the following contributions to the state of the art:

[1] Fusion centers in this type of network are assumed to perform no decision functions; i.e., they simply combine measurements and provide data to operators.

[2] Central coordinators are assumed to perform the functions of a fusion center *as well as* performing some decision-making functions.

- The first work studying the role of feedback in cognitive radar networks. In particular, we study the case where cognition is divided between a Central Coordinator and the individual Cognitive Radar Nodes. We do this by developing a framework for feedback, then structuring several algorithms which take advantage of different levels of feedback. We show that there is a direct correlation between feedback and target tracking performance.
- A system model for analyzing feedback in CRNs, where a CC provides data fusion as well as cognitive functions. This is useful for future works, as such a model does not yet exist in the literature.
- A mathematical analysis of the different reward functions available to learning algorithms in such a framework. In addition, we discuss when approximations to these rewards may be merited.
- We modify an existing decentralized algorithm [115] to introduce feedback.
- We supply simulations comparing our proposed model against techniques without feedback as well as an oracle which selects the actions which are best in hindsight.
- We show that CRN performance can be significantly improved over short time horizons when feedback is used, and that even infrequent feedback is sufficient to improve convergence time in some scenarios.

4.1.2 Notation

We use the following notation. Matrices and vectors are denoted as bold upper $\mathbf{X}$ or lower $\mathbf{x}$ case letters respectively. Element-wise multiplication of two matrices or vectors is shown as $X \odot Y$. Functions are shown as plain letters F or f. Sets $\mathcal{A}$ are shown as script letters. The cardinality $|\mathcal{A}|$ of a set $\mathcal{A}$ refers to the number of elements in that set. The transpose operation is $\mathbf{X}^T$. The set of all real numbers is $\mathbb{R}$ and the set of integers is $\mathbb{Z}$. The speed of electromagnetic radiation in a vacuum is given as c. The Euclidean norm of a vector $\mathbf{x}$ is written as $||\mathbf{x}||$. Estimates of a true parameter p are given as $\hat{p}$.

4.1.3 Organization

The remainder of this paper is organized as follows. Section II discusses previous work in the field of cognitive radar networks and relevant machine learning. Section III provides the network system model assumed in this work. Section IV covers the relevant learning theory and details the reward models. Our proposed algorithms are discussed in Section V and Section VI provides simulations comparing our algorithms against several baselines. We draw conclusions in Section VII.

4.2 Background

4.2.1 Related Previous Work

Cognitive Radar

Cognitive radar (CR) has been the subject of intense study in recent years. In [25], the authors survey recent work in spectrum sharing for cognitive radar. Since CR has inherent operational flexibility, it is natural to implement spectrum sharing in environments where CR nodes are secondary users. Cognitive radar, as a field, has been investigated since the early 2000s [89] [116]. Various parameters have been exposed to cognitive decision-making: target parameter estimation, resource management, RF filtering, waveform selection, etc. The authors of [117] and [118] investigate single-node cognitive radar and apply detailed machine learning techniques describing waveform selection techniques and adapting them to a broad class of target models.

Early research into cognitive systems was motivated in part by biological systems [119]. Researchers wished to enable cognitive agents to display the adaptive intelligence and decision making capabilities exhibited by biological systems. In general, this is accomplished through observation of the environment and use of statistical or machine learning algorithms to act on new information [1]. Adaptive radar systems are a good fit for cognition since they can model the echolocation abilities of bats [120].

Cogntive Radar Networks

Cognitive radar networks have also been addressed in the literature, from their proposal in 2006 [6] to more recent work. In general, the work on CRNs has been focused on time allocation (scheduling) or power allocation.

In an early work on CRNs, the authors of [48] propose a beamsteering strategy to split a search space between two radar nodes in a centralized CRN. This work showed a performance improvement in both detection and tracking over a network of two traditional radar nodes. While the problem addressed resource sharing in CRNs, it is limited in scope to two radar nodes.

Several works focus on power and dwell time allocation in CRNs [121], [46], [122]. These works consider CRNs sharing a single channel, which must allocate the limited observation time to the nodes of the networks. Instead of considering time division access schemes, our current work considers channelized spectrum. Further, many of these works consider pre-allocation schemes rather than the adaptive methods we consider here.

Scheduling has been applied to the mutual interference problem in radar networks [49], with the goal of reducing pulse collisions within a CRN. While this method was shown to

be effective and feasible, it relies on pre-allocation of resources which can fail to perform optimally in a dynamic environment where the mean SINR in each channel can vary in time.

In addition, multistatic cognitive radar networks have been studied [123], where each radar in a network is able to receive and process the pulses transmitted by the other nodes. Multistatic radar operation allows for greater tracking accuracy, at a cost of greater amounts of coordination and processing.

Machine Learning Applied to CRNs

Statistical and machine learning (ML) approaches are natural for CRNs. Since cognitive nodes are able to observe the environment over time and choose from multiple actions, reinforcement learning is particularly well-suited. Reinforcement learning is the branch of machine learning that deals with sequential learning in possibly stochastic environments [105]. Since the exact interference and target behavior cannot be known in advance, approaches that adapt and generalize to broad classes of environments will out-perform those which depend on specific target and/or interference behavior.

As mentioned above, the purpose of this work is to investigate the balance between distributed and centralized learning. As such, we must primarily consider distributed learning models, and how they can be adapted in a hybrid framework. Distributed learning spreads components of a learning structure across nodes in a network [124]. Obviously, distributed learning techniques come with several requirements. The selected algorithm must be well-suited to the environment. For example, the field of research into federated learning [125] [114] investigates isolated models, trained on independent identically distributed (iid) data. This iid assumption is not valid in all environments; particularly, since all of the nodes in a CRN sample the same environment and are tracking the same targets, the observations are not independent.

In this work we predominantly employ models from the multi-armed bandit (MAB) literature. MAB models are applicable to sequential learning problems where one or several players attempt to maximize rewards observed from action choices. The MAB model does not provide the player(s) with prior information regarding the reward for each action choice. When multiple players are included, the relevant models are called multi-player multi-armed bandits (MMAB) [32]. MMAB models are a recent development, motivated primarily by cognitive radio networks [126] [84]. Cognitive radio networks are well studied in the literature, but are a very different problem than cognitive radar: while cognitive radio considers channel capacity and optimization for multiple users, cognitive radar attempts to maximize target tracking and detection. Further, the parts of a cognitive radio network have individual goals (i.e., desired data rate), while the parts of a cognitive radar network collaborate on joint goals.

MMABs consider multiple independent players acting on a single action set. If multiple

players select the same action at the same time, they collide and receive a discounted reward. Without cooperation, this can turn into a competition between players for the highest-reward actions, causing collisions and generally reducing performance. When the players cooperate, they can instead optimize for network-optimal solutions, rather than single-node optimal solutions. Further, the presence of coordination or communication can reduce instances of collisions and improve reward payouts over time.

As with centralized and decentralized cognition models, there exist centralized and decentralized MMAB models. Decentralized models must exploit collisions to exchange information [127] [128] [115], while centralized models have the use of a side channel for communication [129] [130].

Models also exist for adversarial environments [56], [131], where interferer behavior can be chosen in advance by an adversary which knows the cognitive strategy being used by the CRN. This models the scenario where an interferer attempts to force the CRN into a poor performing configuration. Our current work considers the case where interferers are oblivious to the CRN and do not modify their behavior in response to CRN actions since they are considered primary users.

In general, decentralized CRNs have not been well addressed in the literature. Specifically, there has been no study of the relationship between feedback and CRN performance. Further, while CRN time and frequency resource allocation has been investigated, there is a lack of study into adaptive models.

4.2.2 Problem Summary

As covered above, the problem where SINR is constant in space and time and the CRN is completely distributed with no feedback has been studied in [51]. We instead consider the case where the spectrum is interference limited in every channel, and the SINR varies by node and over time due to target motion and range from each radar node location.

We study reward models which are applicable to this situation. We discuss a model where rewards are based solely on average SINR (as determined by the CC), then provide an approximation which reduces the required feedback. Since the rewards are dependent on both the interference power in the environment as well as the relative target range at each node, the nodes can estimate future rewards by separating these two effects. The goal of the CRN is to predict the SINR for each channel at each node in the following CPI, taking into account observed interference and estimated target behavior. Then, each node in the network attempts to select a channel which maximizes the total reward for the network. We will study the amount of feedback that a CC can provide in order to accelerate this learning process. In particular, we will demonstrate a trade-off between performance and feedback cost.

This is a coupled estimation problem; the nodes must simultaneously estimate the channel

and the target parameters while avoiding other radar nodes in order to learn the environment.

Collisions occur when more than one radar node transmit in the same channel at the same time since they cause unacceptably high levels of interference at the impacted nodes. Importantly, the feedback in our network and algorithm allows collisions to be largely avoided.

4.3 Network Structure

The general structure of our network is as follows. The radar network consists of a set $\mathcal{M}$ of M radar nodes. These nodes are distributed uniformly at random throughout an area 10km by 10km. While realistic scenarios should include a third spatial dimension, we consider two dimensional space to reduce the simulation complexity. Since the algorithms we will discuss only require the position and velocity of each target, we can make this assumption without loss of generality. The position of each node $R_m \in \mathcal{M}$, $(m \leq M)$, is denoted as $\mathbf{p}_m = [x_m, y_m]$. Since the the nodes can exchange information through the CC, $\mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M]$ is known to all nodes.

The environment is assumed to contain one target, and (as we show later), the channels are sufficient to reliably detect the target. The complete target tracking and detection structure is discussed in a later section. The position of the target is denoted as $\mathbf{y}_w = [x_w, y_w]$. The set $\mathcal{N}$ contains the N orthogonal channels C_n of equal bandwidth. Each radar node is able to transmit one Linear Frequency Modulated (LFM) chirp waveform in a channel C_n which it must select.

The CRN divides time into CPIs and further into Pulse Repetition Intervals (PRIs). Each CPI consists of 512 PRIs, where a single PRI lasts for 0.1 ms and a single waveform lasts 1 µs. During each CPI k, each radar node $R_m \in \mathcal{M}$ executes the following (roughly synchronized to the CC clock):

1. Select a channel $C_n \in \mathcal{N}$ using a learning algorithm, and transmit a train of 512 LFM pulses.
2. Receive the waveforms and process the returns to determine estimates[3] of:
   (a) Target range $\hat{r}_m(k)$.
   (b) Target radial velocity $\hat{\dot{r}}_m(k)$.
   (c) Target angle of arrival $\hat{\theta}_m(k)$.
3. Transmit the target parameter estimates to the CC.
4. Receive a state estimate for the target from the CC:

[3]Recall that estimates are denoted with a hat $\hat{p}$, while true parameters are denoted without.

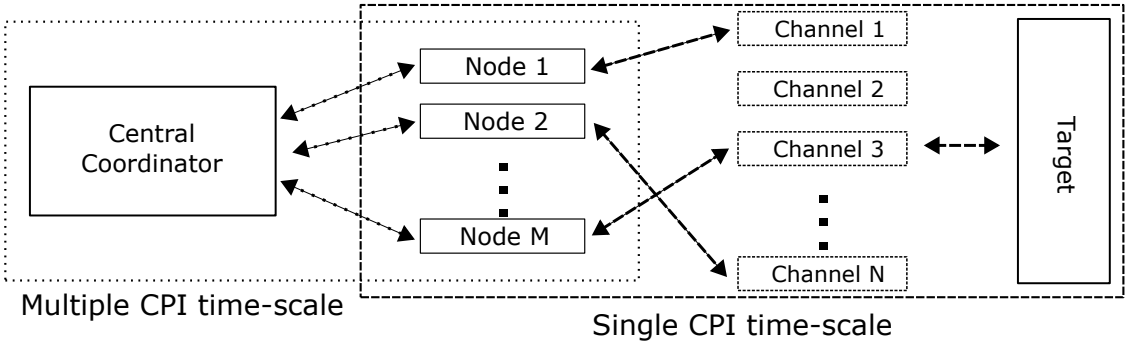


Figure 4.1: System diagram. Once per CPI, each radar node selects a single channel. Since there are more channels than radar nodes ($N > M$), some may be unused. However, every radar node will be paired. On a slower timescale (i.e., over multiple CPI's), the radar nodes communicate with and receive feedback from the central coordinator.

   (a) Target position $\hat{\mathbf{y}}_w(k)$.
   (b) Target velocity $\hat{\dot{\mathbf{y}}}_w(k)$.

5. Update target tracking filter[4].
6. Update the learning algorithm to allow better choice of channel, requesting and incorporating CC feedback as necessary.

Concurrently, the CC performs the following functions:

1. Receive target parameter estimates for all targets from each radar node.
2. Fuse these measurements to determine a target state estimate for all targets.
3. Transmit the target state estimate to the radar nodes.
4. Provide channel selection feedback as required.

We refer to target *parameter* estimates, which are the range, radial velocity, and angle of arrival for each target *from the perspective of each node*. Target *state* estimates are the fused estimates provided by the CC, which include position estimates as well as velocity estimates. Figure 4.1 shows a diagram of this network structure.

Importantly, we assume that the CRN consists of low cost, low complexity radar nodes. This has several implications:

- In order to conserve power, the radar nodes conduct signal processing only once per CPI.

[4]Filters are maintained at each node and at the CC.

- The transmit arrays of each radar node have sufficient gain to illuminate the target and are electronically steerable.
- Cognition is shared between the CC and nodes to mitigate any duplication of effort.

4.3.1 Target and Channel Modeling

We assume that the environment contains multiple sources in each channel, distributed through space, and sufficient clutter such that the interference has no strong directional components. This results in interference power with possibly strong variation by channel, but relatively little variation in space. For parts of this work, we assume that these spatial variations are sufficient to provide different interference power values at each radar node, but not so much as to cause the *rank* of these values to change.

Assumption 1 (Reward Ordering)**.** If one radar node observes a greater reward in channel C_{n_1} than in channel C_{n_2}, all other nodes will observe the same. The reward magnitudes may however differ.

Note that in some sense, this assumption represents a worst case scenario - while all nodes will observe the same "best" channel, only one of them will be able to select it. Therefore, in the absence of coordination or feedback, the network would collide frequently.

Later, we will discuss the impact of this assumption, and how it can be relaxed. Specifically, we present results with and without this assumption.

Signal Model

In each CPI, each radar node selects a channel C_n with an associated start frequency f_n and transmits a train of 2^{10} Linear Frequency Modulated pulses. Eq. (4.1) represents a single pulse.

$$s[t] = \sin\left[\phi_0 + 2\pi\left(\frac{r}{2}t^2 + f_n t\right)\right], \quad t \in [t_a, t_b] \tag{4.1}$$

Here, t is the so-called *fast time* and indexes samples of the pulse, ϕ_0 is an initial phase, and r is a constant chirp rate.

The target is modeled as an isotropic scatterer and thus has constant Radar Cross Section (RCS) as a function of angle-of-arrival θ. In addition, we assume that the target response is not frequency-selective. This means that the target will "look" the same at all frequencies.

We can write the received signal for radar node R_m as Eq. (4.2) where τ is the propagation delay, $i_n(t)$ is the interference waveform in channel C_n and $n(t)$ is noise.

$$y_m[t] = s\left[(1 - \frac{2\dot{r}_m}{c})t - \tau\right] + i_n(t) + n(t) \tag{4.2}$$

Denote the power of the transmitted signal at all nodes as $P_s = \frac{1}{N}\sum_n |s[n]|^2$ and the power of the received signal as Eq. (4.3), where $P_{y,m}$ is the power received from the target at node R_m, $P_{i,n}$ is the interference power in channel C_n, and σ^2 is the noise power.

$$P = \frac{1}{N}\sum_n |y_m[n]|^2 = P_{y,m} + P_{i,n} + \sigma^2 \tag{4.3}$$

According to the radar equation, the power $P_{y,m}$ should follow Eq. (4.4), where r_m is the target range from the m^{th} node.

$$P_{y,m} = \frac{P_x G^2 \lambda^2 \sigma}{(4\pi)^3 r_m^4} \tag{4.4}$$

Since the target RCS is constant over frequencies f_n in the bandwidth we consider and angle θ, it will be constant over radar node measurements. Each radar node can form an *estimate* of the future power received from the target as Eq. (4.5).

$$\hat{P}_m[k_0] = \frac{P_x G^2 \lambda^2}{(4\pi)^3 \hat{r}_m[t+k_0]^4} \tag{4.5}$$

This power estimate depends on an estimate $\hat{r}_m[t+k_0]$ of the range some number k_0 of time steps in the future. The quality of this estimate will be dictated by the radar observation quality in all time steps until t, and is essentially dependent on tracking performance.

Target estimation quality is directly influenced by channel SINR. Denote the SINR experienced by radar node R_m in channel C_n as Eq. (4.6).

$$\gamma_{m,n} = \frac{P_{y,m}}{P_{i,n} + \sigma^2} \tag{4.6}$$

Since radar measurement quality is influenced by SINR, we'd like to develop a metric which uses this information. So, let the metric be given as Eq. (4.7) where $\gamma_{m,n}^{(dB)} = 10\log_{10}(\gamma_{m,n})$.

$$\Gamma_{m,n}[t+k_0] = \gamma_{m,n}^{(dB)} - \hat{P}_m^{(dB)}[k_0] \tag{4.7}$$

This metric is useful because it allows each radar node m to arrive at a similar estimate of the quality of channel n. This is necessary due to the distributed nature of the problem; we'd like for the independent radar nodes to be able to avoid colliding with each other (i.e., selecting the same action simultaneously) without communication. Prior work has shown that collisions greatly reduce the performance of a radar network [55].

Note that $\hat{P}_m^{(dB)}[k_0] = 10\log_{10}(\hat{P}_m[k_0])$. Due to the assumption on interference power ordering, we can now see that if R_{m_1} experiences $P_{m_1,n_1} > P_{m_1,n_2}$ for two channels C_{n_1} and C_{n_2},

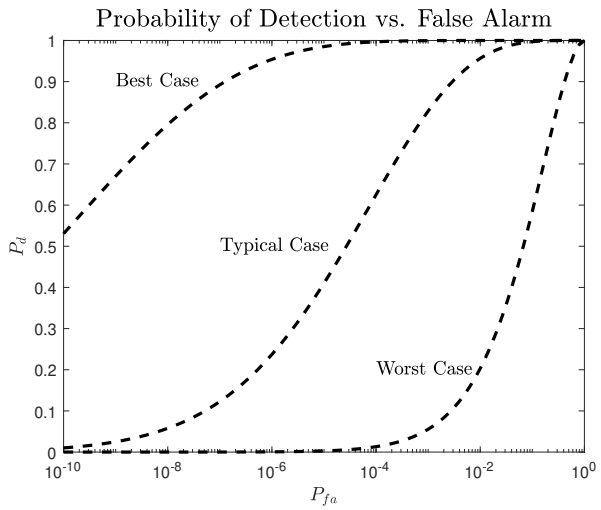


Figure 4.2: Probability of detection versus probability of false alarm for the worst, average, and best case node-channel matchings. These values are based on the assumed parameters stated in Table 4.2.

then R_{m_2} will observe the same power ordering ($P_{m_2,n_1} > P_{m_2,n_2}$) for any choice of radars R_{m_1}, R_{m_2} or channels C_{n_1}, C_{n_2}.

4.3.2 Tracking Formulation

While the spectrum is interference-limited, we assume that the best case channels have SINR high enough for consistent target detection. Figure 4.2 shows the probability of detection versus probability of false alarm for the best, typical, and worst case network average SINR based on the assumed parameters in Table 4.2.

When each radar node observes the target, it estimates the target position $\hat{\mathbf{y}}_m(k)$ and velocity $\hat{\dot{\mathbf{y}}}_m(k)$. These are used to update a Kalman filter model of the target's motion. Assuming a two-dimensional motion model, the predicted state is given as Eq. (4.8) where F_k is the transition model. Note that the state $x_k = [\mathbf{y}_m(k)^T, \dot{\mathbf{y}}_m(k)^T]^T$ is composed of target position and velocity.

$$\hat{x}_{k|k-1} = F_k x_{x-1|k-1} + B_k u_k \tag{4.8}$$

The state is then updated as Eq. (4.9) where K_k is the estimated optimal Kalman gain and $\tilde{y}_k$ is the innovation.

$$x_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{4.9}$$

This formulation follows the common notation of [71].

4.4 Learning Structure

As is common in the MAB and sequential learning literature, we will define our problem based on a series of *actions* taken by the players (i.e., radar nodes) and *rewards* provided by the environment. We are specifically considering a *sensing then collision* [30] model. This means that the players observe first the prospective reward (discounted, in case of collision) for a given action, followed by information on collisions. Collisions are instances of two radar nodes choosing identical actions at the same time. However, due to the learning framework and presence of feedback, collisions are unlikely. This is a realistic model since previous work has demonstrated a method to detect collisions [51]. In addition, we make the assumption rewards observed by node R_{m_1} are not available to any other node R_{m_2}. This follows the assumed network structure; nodes can exchange information with the CC but not directly with each other.

4.4.1 Matchings and Utility

Before we can define the learning framework, we need to better describe the objective. Let a *weight matrix* W be a matrix with M rows corresponding to the M radars and N columns corresponding to the N channels. Each index consists of the reward[5] observed by radar R_m for selecting channel C_n during the k^{th} CPI. Valid actions which can be selected by algorithms under our framework must be in the set of all *matchings*.

Definition 4.1 (Matching)**.** A *matching* $\pi : \mathcal{M} \to \mathcal{N}$ is any assignment from the set of radar nodes $\mathcal{M}$ to the set of channels $\mathcal{N}$ with the following properties:

1. Matchings are injective[6]. This means that every radar node will select a single channel per CPI, but not every channel will necessarily be used.
2. Matchings map every element of $\mathcal{M}$ to a unique element of $\mathcal{N}$.

Denote the set of all matchings as Π.

Matchings are a special case of *mappings*, which remove the injectiveness property. Obviously, since the weights vary by radar node and by channel, some matchings will be better than others. We measure the quality of a matching via its *utility*.

Definition 4.2 (Utility)**.** The *utility* of a matching π is the sum of the rewards observed under that matching.

$$U(\pi) = \sum_{\mathcal{M}} W_{m,\pi_m} \tag{4.10}$$

[5]The specific reward function is defined in the following section.

[6]While they are injective, matchings are not necessarily bijective since M does not necessarily equal N. A function $f : X \to Y$ is injective if for all a, b in X, $f(a) = f(b) \implies a = b$.

The utility of a matching represents the quality of each channel selected for radar observation. Note that W_{m,π_m} refers to reward observed by R_m due to selecting a channel C_{π_m}, where π_m is the index of matching π corresponding to node R_m. Further, there will be at least one $\pi \in \Pi$ with greatest utility. utility describes the quality of measurement obtained by a particular node for selecting a given channel.

Definition 4.3 (Optimal Matching)**.** If a matching $\pi \in \Pi$ has maximum utility U^*, it is called optimal and denoted π^*. In other words, $U(\pi) = U^* \implies \pi = \pi^*$.

$$U^* = \max_{\pi \in \Pi} U(\pi) \tag{4.11}$$

Remark 4.4. Note that while there may be many $\pi \in \Pi$ with $U(\pi) = U^*$, we slightly abuse notation and simply refer to any optimal matching as π^*. In practice, there is very rarely more than one optimal matching for a given weight matrix.

4.4.2 Learning Objective

The goal of any learning algorithm in this system is then to minimize the amount of non-optimal matchings encountered during a game. Of course, since there is no *a priori* knowledge of the environment, it is impossible to avoid selecting non-optimal matchings or even to know the value of U^*. This is particularly important in radar problems since sub-optimal mappings can result in missed targets.

It is useful to view a learning algorithm as a function $\mathfrak{f}(\mathbb{E}) = \pi^{(K)}$ which produces a sequence of matchings π^k, $k = 1 : K$ in an environment $\mathbb{E}$ where K is some finite horizon. Here, π^k is a single matching while $\pi^{(K)}$ denotes a sequence. Then, the sequence $\pi^{(K)}$ contains all of the matchings produced by the learning algorithm until CPI K. Note that $\pi^{(K)}$ is implicitly conditioned on a specific instance of an environment; if the sequence of rewards changes, then the sequence of matchings would change. Also note that $\pi_m^{(K)}$ is the slice of actions chosen by radar node R_m until CPI k.

We can measure the difference between learning algorithms by comparing the cumulative utility of a matching sequence until CPI k.

$$U^k(\pi) = \sum_{\kappa=1}^{k} U(\pi^\kappa) \tag{4.12}$$

In order to compare all learning algorithms to a universal baseline, we can refer to the utility of the sequence of *optimal* matchings $\pi^*(k)$ for a given environment. This quantity is called the *cumulative regret* of π^k.

Definition 4.5 (Cumulative Regret)**.** The *cumulative regret* of a learning algorithm $\mathfrak{f}$ which produces a sequence of matchings π^k until time k is the difference in cumulative utility between $U^k(\pi)$ and $U^k(\pi^*)$.

$$\rho_{\mathfrak{f}}^t = U^k(\pi^*) - U^k(\pi) \tag{4.13}$$

Note that cumulative regret is monotonically increasing in k, since $U(\pi^k) \leq U(\pi^{*,k}))$ by definition.

Now, the *objective* of a learning algorithm $\mathfrak{f}$ is to obtain the lowest $\rho_{\mathfrak{f}}^K$ for some finite time horizon K.

4.4.3 Rewards

The learning problem is not fully defined without specifying the reward function. Typically, sequential learning rewards are drawn from some distribution, dependent on the action selected by the learner. We will define two different reward functions that capture key aspects of the radar scenario.

The key differentiation between the two reward functions we will describe is an explicit separation between the two underlying estimation processes. The *interference estimation* process is the part of the cognitive radar scenario where each node attempts to learn some metric of the interference in each channel. The *target estimation* process, however, is the overall goal of the cognitive radar network. In the absence of interference estimation, the network may select poor channels over time and therefore sacrifice tracking performance. However, if the network attempts to optimize too quickly for radar tracking performance, again it may suffer from selecting sub-optimal actions. This is the classic trade-off in sequential learning between *exploration* and *exploitation*. The cognitive nodes must efficiently balance the exploration and exploitation in order to avoid sub-optimal long-term performance.

We will first show a reward function that attempts to separate these underlying processes, and then discuss a simpler model.

SINR **Rewards**

The first rewards we consider are based solely on the SINR observed by radar node R_m in channel C_n. Due to variability in the environment (i.e., target motion or changes in interference), the SINR may change from CPI to CPI. We define the *true* SINR observed by node R_m in channel C_n as $\gamma_{m,n}(k)$ and note that this value will vary by node and channel due to relative spatial differences in target position and differences in interference. Let the full matrix of these values be denoted as $\gamma(k)$ in a CPI k. Let each node draw an *estimate* of this SINR as Eq. (4.14).

$$\hat{\gamma}_{m,n}(k) \sim \mathcal{N}(\gamma_{m,n}(k), \sigma_\gamma^2) \tag{4.14}$$

Now we can form each element of the weight matrix under this reward function in CPI k as $W_{m,n}^{\hat{\gamma}(k)} = \hat{\gamma}_{m,n}(k)$. Also, we can write the utility of a matching π^k under SINR rewards as $U^{\hat{\gamma}}(\pi^k)$.

$$U^{\hat{\gamma}}(\pi^k) = \sum_{m \in \mathcal{M}} W_{m,\pi_m}^{\hat{\gamma}(k)} \tag{4.15}$$

Target Based Rewards

In practice, SINR-based rewards as described above would require each node to share its observed rewards with the CC, and then to rely on the CC to provide actions. This is because one node would have no other way to know the rewards being experienced by another node. This reduces the redundancy of the system, since the central coordinator is the only agent making decisions. If the radar nodes were instead able to estimate the rewards observed by each other node, then they would be able to make decisions in the absence of the coordinator.

As shown previously, the channel metric Eq. (4.7) attempts to decouple the interference behavior from the target motion. Following the assumption that interference behavior in each channel is identical as observed by each node, we can recombine the channel metric with an estimate of the target range at each node to estimate the SINR observed at each location in the network. In other words, the channel metric combined with an estimate of the target position can produce a reward estimate while requiring less information than the matrix $W^{\hat{\gamma}(k)}$.

Now, we can write the elements of this new reward function as $W^{\Gamma}(m, \pi_m) = \frac{1}{\hat{r}_m^4}\Gamma_{m,n}$ or more generally as Eq. (4.16) where $\overline{\mathbf{r}} = [\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_M]$ is a vector of the estimated distance from each node to the target and $\overline{\Gamma}_m = [\Gamma_{m,1}, \Gamma_{m,2}, \ldots, \Gamma_{m,N}]$ is a vector of channel metrics calculated by node R_m. Note that W^{Γ} is a $M \times N$ matrix, and we later denote indices with subscripts. The channel metric is divided by the estimated distance to each node in order to favor those nodes with better views of the target.

$$W^{\Gamma} = \left(\frac{1}{\overline{\mathbf{r}}^4}\right)^T \odot \overline{\Gamma} \tag{4.16}$$

Then, the estimated utility under this reward function is expressed as Eq. (4.17).

$$U^{\Gamma}(\pi^k) = \sum_{m \in \mathcal{M}} W^{\Gamma}_{m,\pi^k_m} \tag{4.17}$$

Lemma 4.6 (Reward Equivalency)**.** *The optimal matching under SINR rewards is equal to the optimal matching under target-based rewards when Assumption 1 holds.*

$$\max_{\pi \in \gamma(k)} U(\pi) = \max_{\pi \in W^{\Gamma}(k)} U(\pi) \tag{4.18}$$

Proof. See Appendix B. □

Of course, if the optimal matching provided by each reward function is the same as in Lemma 4.6, why should a CRN prefer one reward function over the other? A single node, without coordination or feedback, can not know the rewards observed by another node without feedback. This means that nodes may not be able to establish a consensus. However, if the

node is able to calculate the channel metric (which is node-independent) and target position, it can then estimate the rewards observed by all other nodes. As we will show later, this can allow a CRN to develop a near-optimal matching, while receiving feedback at semi-regular intervals can further improve this performance.

4.4.4 Feedback

In addition to measuring the performance of an algorithm through regret, we can analyze the amount of information that algorithm exchanges through the CC. In particular, we can look at the average number of floating-point values sent from the CC to each node. This will allow us to compare the benefits of varying levels of feedback in the network. Call F_k the set of values transmitted by the CC in CPI k.

Definition 4.7 (Average Feedback)**.** The *average feedback* used by a network in a CPI k is the sum of all feedback $|F_j|$ until k divided by k and the number of nodes M.

$$F_a(k) = \frac{1}{Mk} \sum_{j=1}^{k} |F_j| \tag{4.19}$$

4.5 Candidate Algorithms

We present several algorithms, starting with a fully centralized variant and moving towards minimal feedback. We do this to investigate the trade-off between feedback and performance. Ultimately, we show that a minimal amount of feedback is sufficient to provide dramatic benefits over the prior art, and increasing feedback provides diminishing returns.

These algorithms are based on Explore Then Commit [115]. This was initially developed for fully decentralized action selection. We make a slight modification to allow for a central coordinator to eliminate a lengthy phase where nodes exchange information.

We will describe an oracle for this problem, which is aware of the true SINR in each time step, and perfectly selects the optimal matching. In addition we compare against a naive algorithm which selects a new random matching each CPI as well as a previously proposed decentralized algorithm [51].

The algorithms we discuss are summarized in Table 4.1.

4.5.1 Oracle

An oracle for this problem knows the true SINR and target position, and thus can perfectly estimate the rewards. Therefore, in each CPI k, the each node R_m in the network will select

$\pi^*_m(k)$. This ensures that the cumulative regret for the oracle is always 0.

4.5.2 Centralized Explore-Then-Commit (C-ETC)

C-ETC [115] implements the Upper Confidence Bound (UCB) [107] as a threshold to refine a sequence of sets of matchings Π_0, Π_1, etc., with $\Pi = \Pi_0 \supseteq \Pi_1 \supseteq \Pi_2 \supseteq$ etc. Each set of matchings Π_j can be said to contain p_j matchings and is viewed as an ordered list. Each radar node R_m selects channel $\pi_{l,m} \in \pi_l$, and $\pi_l \in \Pi_j$, with l, j specified by C-ETC. Since each Π_j is a refinement of Π_{j-1}, Π_0 can be known to all of the nodes since it is the initial condition. Then for each subsequent $\Pi_j, j > 0$, the CC can indicate which matchings remain and which are discarded. Thus, all nodes know the channels used by other nodes.

This allows each node to observe rewards in the environment sequentially and possibly multiple times, to allow the observed rewards to average towards the true rewards. Over time, the algorithm identifies which matchings $\pi_l \in \Pi_j$ have $U(\pi_l)$ below a threshold, and removes them from Π_{j+1}. This is a fully-centralized algorithm, since decisions are only made at the CC. A sketch of our implementation of C-ETC is shown in Algorithm 6.

Algorithm 6: Sketch of Explore-Then-Commit for node R_m

```
% CPI k;
Transmit radar waveform in channel C(k) = π_{l,m} ∈ π_l ∈ Π_j;
Estimate target parameters r̂_m(k), ṙ̂_m(k), θ̂_m(k);
if |Π_k| = 1 then
    Set Π_{j+1} = Π_j;
else
    if l = |Π_j| then
        Transmit target estimates and channel estimates to CC;
        l = 0;
        Receive Π_{j+1} from CC;
        j = j + 1;
    l = l + 1;
end
k = k + 1;
```

In this way, the action selection is delegated to the CC, while the nodes conduct the radar processing. The CC also combines radar observations from the network to determine a final target state estimate.

Table 4.1: Candidate Algorithms

Algorithm	Acronym	Rewards	Feedback
Oracle	N/A	N/A	None.
Explore-Then-Commit	C-ETC	SINR	Matchings prescribed by CC.
Centralized ETP	C-ETP	SINR	Entire reward matrix.
Hybrid ETP	H-ETP	Target-based	Current target state.
Explore-Then-Predict	ETP	Target-based	Initial matchings.
Musical Chairs [51]	MC	SINR	None.
Random Matchings	N/A	N/A	None.

4.5.3 Centralized Explore-Then-Predict (C-ETP)

C-ETP modifies C-ETC in one major way, while remaining a centralized algorithm. Rather than committing to a single matching after exploration, C-ETP continues to evaluate the weight matrix until the end of the game. From a given matrix, C-ETP calculates the maximum matching at each node and selects it. The weight matrix is formed in each CPI k at the CC as $W^{\hat{\gamma}(k)}$ by using the SINR measurements made at each node. This results in an estimator that is able to modify action selections over time in order to adapt to the changing environment. Since each node is acting on the same information, they can know which actions the other nodes will select. This has the relatively large downside of requiring feedback from the CC in every time step, greatly *increasing* the feedback rate after convergence. C-ETC does not suffer from this problem since it does not evaluate the reward matrix after convergence.

The C-ETC-like exploration period is still required for C-ETP due to the noisy observations; without a well-defined exploration structure, the algorithm would fail to converge at all. However, as the nodes continue to utilize the best performing channels, the weights may cause two nodes to switch channels due to target position.

A sketch of the node-side algorithm for C-ETP is shown in Algorithm 7. Note that if more than one matching in $W^{\hat{\gamma}(k)}$ is optimal, the algorithm selects one at random.

4.5.4 Hybrid Explore-Then-Predict (H-ETP)

H-ETP alters C-ETP to use the target-based rewards W^{Γ} Eq. (4.16) to make action decisions after convergence. This only requires knowledge of the target range from each node. So, rather than transmit all of the estimated rewards from each node, H-ETP only requires the current estimated target state to be transmitted from the CC to the nodes. This greatly reduces the feedback rate, thus, this algorithm uses hybrid cognition rather than centralized.

Algorithm 7: Sketch of Centralized Explore-Then-Predict for node R_m

```
% CPI k;
Transmit C(k) = π_{l,m} ∈ π_l ∈ Π_j;
Estimate r̂_m(k), ṙ̂_m(k), θ̂_m(k);
if |Π_k| = 1 then
    Receive W_{Ŝ(k)} from CC;
    Π_{j+1} = max_{π∈Π(W_{Ŝ(k)})} U(π);
else
    if l = |Π_j| then
        l = 0;
        Receive Π_{j+1} from CC;
        j = j + 1;
    l = l + 1;
end
k = k + 1;
```

In addition, due to the structure of the target-based rewards and using Lemma 4.6, the performance of H-ETP will at least match that of C-ETC with a greatly reduced feedback rate.

4.5.5 Explore-Then-Predict (ETP)

ETP extends this line of thinking (reducing feedback) even further; instead of relying on the network's estimated target state, ETP simply uses the target parameters estimated at each node to estimate the range to each node. Due to compounding estimation errors, ETP should have reduced tracking accuracy and higher regret than either H-ETP or C-ETP, but require the smallest feedback rate. Specifically, the only information ETP requires is in the initial part of the game. The CC must determine initial matchings to explore in order to prevent collisions.

4.5.6 Random Matchings

We finally consider a naive algorithm which simply selects from a pre-determined random sequence of matchings. Before the game begins, let the CC specify an appropriately sized set Π^R of matchings. In each CPI k, let node R_m select channel $\Pi^R_m(k)$. This ensures that two nodes never select the same channel in a CPI. However, since the matching sequence is

Algorithm 8: Sketch of Hybrid Explore-Then-Predict for node R_m

% CPI k;
Transmit $C(k) = \pi_{l,m} \in \pi_l \in \Pi_j$;
Estimate $\hat{r}_m(k), \dot{\hat{r}}_m(k), \hat{\theta}_m(k)$;
if $|\Pi_k| = 1$ **then**
  Receive $\hat{\mathbf{y}}_w(k)$ from CC;
  $\hat{\mathbf{r}} = |\mathbf{p} - \hat{\mathbf{y}}_w(k)|$;
  $W_{P(k)} = \frac{1}{\hat{\mathbf{r}}} * \overline{\mathbf{P}}$;
  $\Pi_{j+1} = \max_{\pi \in \Pi(W_{P(k)})} U(\pi)$;
else
  if $l = |\Pi_j|$ **then**
    $l = 0$;
    Receive Π_{j+1} from CC;
    $j = j + 1$;
  $l = l + 1$;
end
$k = k + 1$;

Algorithm 9: Explore-Then-Predict for node R_m

% CPI k;
Transmit $C(k) = \pi_{l,m} \in \pi_l \in \Pi_j$;
Estimate $\hat{r}_m(k), \dot{\hat{r}}_m(k), \hat{\theta}_m(k)$;
if $|\Pi_k| = 1$ **then**
  $\hat{\mathbf{y}}_{w,m}(k)$;
  $\hat{\mathbf{r}} = |\mathbf{p} - \hat{\mathbf{y}}_{w,m}(k)|$;
  $W_{P(k)} = \frac{1}{\hat{\mathbf{r}}} * \overline{\mathbf{P}}$;
  $\Pi_{j+1} = \max_{\pi \in \Pi(W_{P(k)})} U(\pi)$;
else
  if $l = |\Pi_j|$ **then**
    $l = 0$;
    Receive Π_{j+1} from CC;
    $j = j + 1$;
  $l = l + 1$;
end
$k = k + 1$;

Table 4.2: Simulation parameters, unless stated otherwise.

Parameter	Value	Parameter	Value
Number of Radar Nodes M	5	Number of Targets	1
PRIs per CPI	500	Target Initial Position	[0,0] m
Total CPIs	700	Bandwidth	20 MHz
Typical SINR	12 dB	Averaged Simulations	30
Frequency	2.4 GHz	PRI Duration	1.024×10^{-4} s
Transmit power	20 dBw	RCS	100 m^2
Antenna gain	30 dB		

pre-determined, the network is not able to learn anything about the environment and simply experiences an average of the possible performance. In addition, since the matching sequence is random and not necessarily optimal in any CPI k, we should expect roughly linear regret in time.

Remark 4.8. Since $\gamma(k)$ (the full matrix of true SINR values) varies in time depending on the target location, algorithms which continue to evaluate their reward matrix will attain lower regret and superior tracking performance than those which retain a fixed allocation post convergence.

4.5.7 Musical Chairs (MC)

Musical Chairs is an algorithm developed in [30] and applied to CRNs in [51] for the completely decentralized case. It relies on a system of implicit collisions through which the network establishes the best-case matching available. This causes a large amount of regret prior to convergence, which can take up to 10^3 time steps (much longer than other algorithms considered here).

4.6 Simulations

We simulate a CRN with five radar nodes and a CC which collaboratively track a single target. The radar nodes are distributed randomly through an area 10 kilometers by 10 kilometers. The single target is initially located at the origin, and moves with a velocity of 200 m s^{-1} headed northeast. The target has a uniform radar cross section of 100 m^2. These values are consistent with a typical commercial aircraft [104].

The radar nodes have access to eight equally spaced channels of 20 MHz each from 2.34-2.5 GHz. Each transmitter outputs pulses at 20 dBw, and the arrays have a main beam gain of 30 dB. These and other simulation parameters are available in Table 4.2. Fig. 4.3 shows

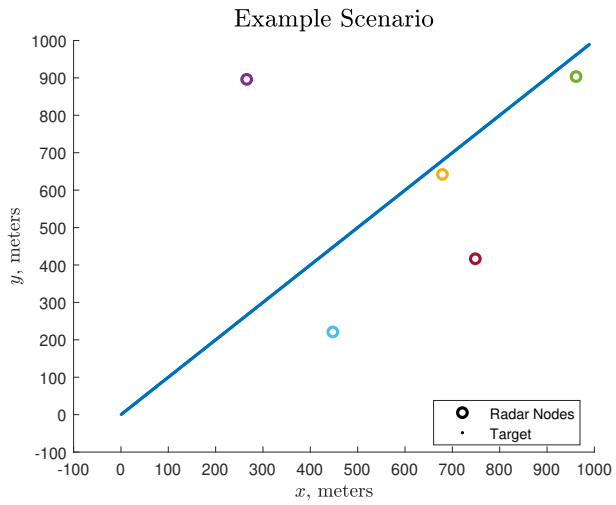


Figure 4.3: The spatial distribution of the radar nodes and the path of the target. Radar positions are drawn from a uniform distribution in each simulation instance. As the target moves through the scene, different radars benefit from selecting different actions.

a single instance of this scenario.

Generally, we should expect to see algorithms with higher feedback exhibit greater performance.

As discussed above, each algorithm will incur regret due to sub-optimal channel selection. In general, we should expect lower regret for algorithms which more closely model the environment. In Fig. 4.4, we see that the cumulative average regret of each learning algorithm goes to zero over time. However, the regret for the random matching algorithm remains constant in time. The learning algorithms take roughly 100 CPIs to converge. The decentralized algorithm MC takes much longer than the simulation time to converge (on the order of 10^3 time steps) and thus exhibits performance roughly equal to random matchings over this short time horizon.

It is also important to consider the feedback rate for each algorithm. As shown in Eq. (4.19), we can measure the amount of feedback by counting the number of floating-point numbers transferred from the CC to the nodes in each time step. Fig. 4.5 shows this. Since all of the learning algorithms incorporate an C-ETC-like exploration phase, they all require a similar amount of information in the early part of the game. However, once the algorithms converge, this feedback rate begins to change.

After convergence (around 100 CPIs), C-ETC and ETP do not use any feedback, as explained above. Before convergence, feedback is only required after the network has explored each list of matchings. Therefore, the average feedback rate trends towards zero. MC uses no feedback at all and thus is not plotted. C-ETP, on the other hand, uses a substantial amount of feedback in each CPI to maintain an updated reward matrix. Therefore, the feedback rate

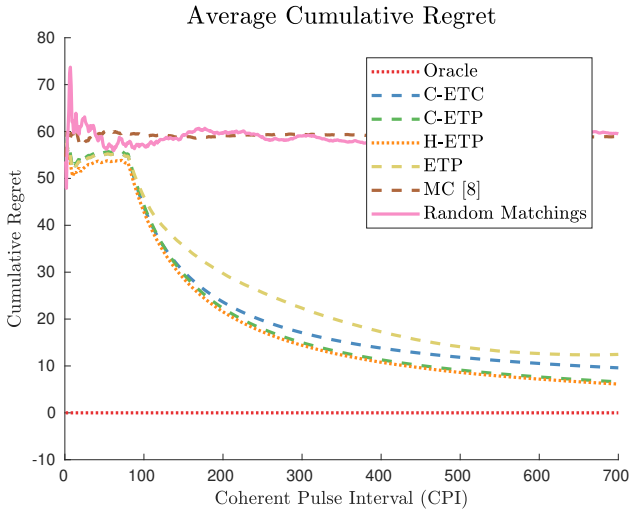


Figure 4.4: After convergence (around CPI 100), the regret of each algorithm goes towards zero. This is because each algorithm is able to identify the best channels to select. However, this identical regret performance does not translate to identical radar tracking performance.

increases after convergence. Finally, H-ETP uses a small amount of feedback every CPI to maintain knowledge of the target position, using this information to estimate the reward matrix. Instead of transferring the entire weight matrix in every CPI, the use of the channel metric allows the H-ETP CC to only transmit the predicted target location in each CPI. This reduction in feedback balances cognition between the CC and the nodes.

ETP uses no feedback after initialization since it relies on node-specific target location estimates to predict the reward matrix. Since these internal estimates are less precise, the predicted reward matrix will be less accurate (possibly leading to collisions). While not impossible, collisions are very rare events.

The radar tracking performance of each algorithm is also important in our application. Since the learning algorithms must explore the environment in the early parts of each simulation, we should expect higher tracking error. In addition, since we use a Kalman tracking filter, we should see lower error once the filter converges. In Fig. 4.6, we see that the average error for the learning algorithms is consistently below 10 m. Occasionally, the environment will shift and the error will spike. Each CRN uses the observation from all radar nodes to establish a localization estimate once per CPI, which we compare against the target's true location in the middle of the CPI. We can see that the oracle exhibits the best performance for the entire simulation, while Random Matchings has quite variable performance. Also note the increased error at the beginning and end of the track. This is due to the target being relatively further away from the nodes during these periods, as well as the poor geometry.

In addition to the average error, we can look at the distribution of the error. In Fig. 4.7, we see an empirical CDF for the error of each algorithm. This represents the probability

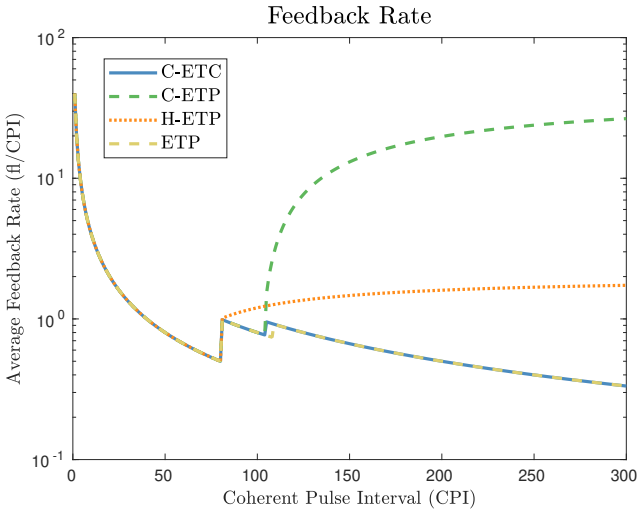


Figure 4.5: The average feedback per node in each of the different networks we examine. ETP is shown to not use any feedback after convergence, while C-ETP uses a great deal of feedback.

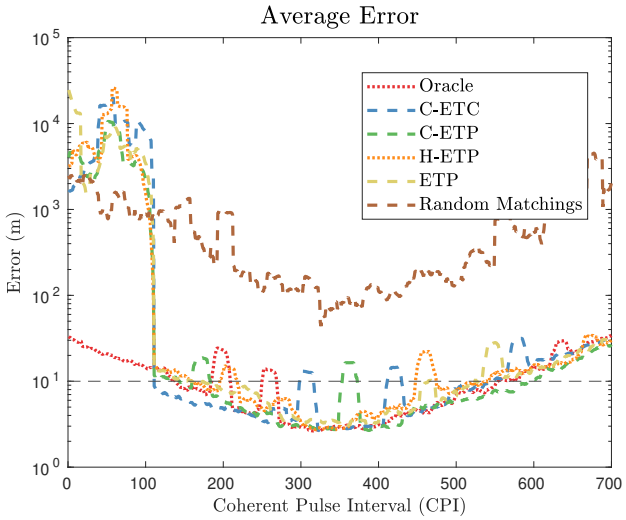


Figure 4.6: Radar localization error, averaged over 30 simulations. The performance of H-ETP is shown to be slightly reduced from C-ETP, but still superior to the other techniques. The increased error in the beginning of the simulation is due to convergence time, both of the machine learning algorithm and of the Kalman tracking filter.

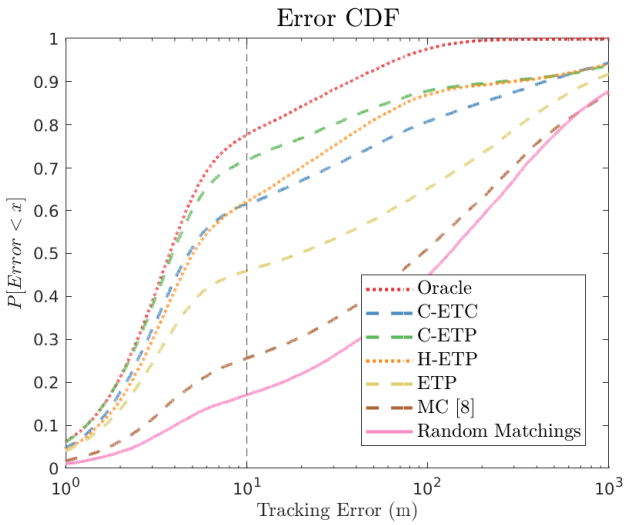


Figure 4.7: The error distribution for each algorithm for the entire simulation.

that the error will be less than a given value. In general, we would like there to be a high probability that the error is small. So, curves which are further to the left are better. From this, we can see that the performance of Random Matching is far below the learning algorithms. In addition, the performance of C-ETP nearly reaches that of the oracle. This is because C-ETP has a constantly-updated reward matrix containing information from the entire network. Of course, as we saw above, this has a high feedback cost.

ETP has reduced performance due to inaccurate prediction of the reward matrix caused by node-specific target state estimates. We can see that H-ETP, while requiring minimal updating, is able to nearly obtain the performance of the centralized variant.

MC does not demonstrate good performance because, due to the lack of feedback, it must conduct a lengthy exploration phase before a consensus is established. This phase is not concluded by the end of the simulation.

Fig. 4.8 goes on to show the performance of the proposed algorithms *after* they have converged. This shows that the performance post convergence is improved over the early game performance. In particular, the performance gap between C-ETC and C-ETP is reduced. Note that H-ETP still obtains performance between these two.

In Fig. 4.9 we see that over a much longer time frame, the MC algorithm converges to an optimal matching. Due to the lack of feedback in MC, this process takes much longer (greater than 2500 CPIs).

When the assumption on reward ordering is removed, we should expect that ETP will underperform, due to the lack of updates and estimation of rewards. H-ETP, however, would continue to receive updates as the environment evolves. Figure 4.10 demonstrates that when the ordering assumption is removed, ETP will slightly underperform. Importantly, the only

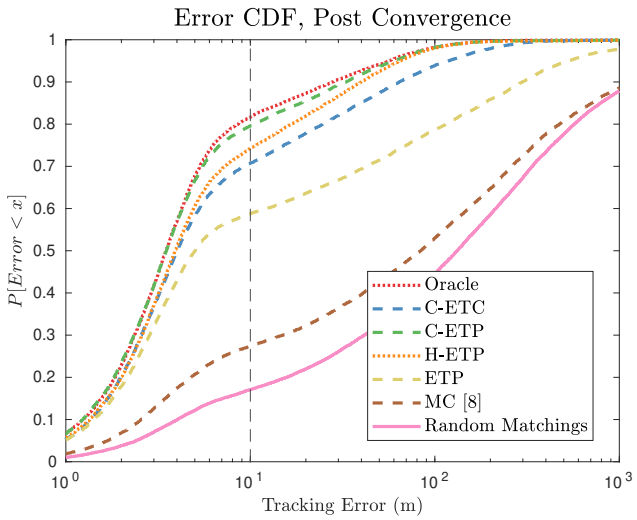


Figure 4.8: The error distribution for each algorithm after convergence.

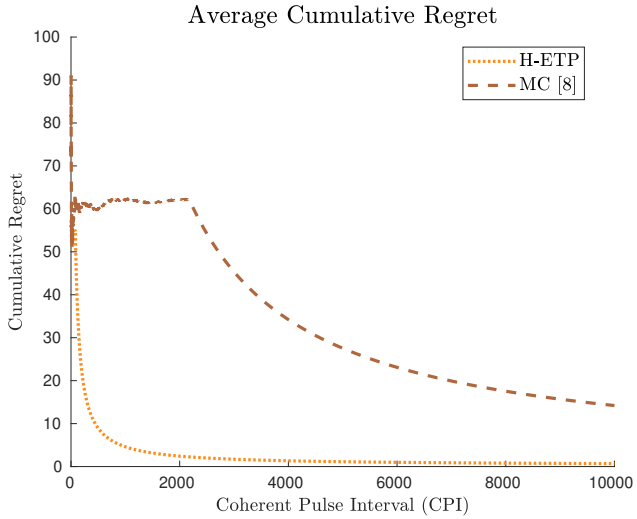


Figure 4.9: Cumulative regret for a much longer simulation.

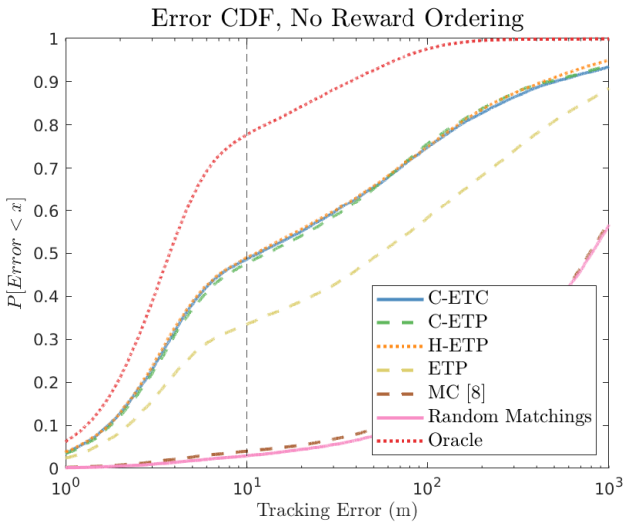


Figure 4.10: Error distributions without an assumption on reward ordering. We see that H-ETP and ETP both underperform due to the structure of W^{Γ}.

reason that H-ETP continues to obtain near-optimal performance is the feedback rate. ETP is still able to learn throughout the game, but without continued feedback is unable to obtain optimal performance. This clearly demonstrates the tradeoff we study in this work: greater feedback leads to greater performance.

4.7 Conclusions and Future Work

In this work we have examined a hybrid cognitive radar network which has both cognitive nodes and a central coordinator. The network attempts to optimize channel selections for each radar over time in order to optimize radar tracking performance of a single target. This problem requires online machine learning techniques due to the desire for low convergence times and applicability across environment instances.

The network uses the observed SINR in each channel to inform rewards for the learning algorithms. Due to this, the nodes in the network must simultaneously learn both the interference behavior in each channel and estimate the target position, since both of these impact the SINR observed for transmitting in a given channel. This results in a coupled structure, which we use to form *estimates* of the SINR at each node. This is useful because use of this estimate can reduce the amount of feedback needed from a central coordinator.

We proposed and examined several online machine learning algorithms that this network can use to effectively balance this coupled learning problem. While previous work has investigated the general CRN learning problem, this work investigated how a network could converge faster, and thus experience better performance for a greater part of the game,

through the use of limited feedback from the central coordinator.

We show that algorithms using this estimated SINR can perform almost as well as algorithms receiving full feedback, while algorithms with no feedback under-perform due to extended convergence times. This represents the trade-off between feedback and performance: generally, more feedback yields greater performance. However, the performance exhibits diminishing returns as feedback increases. Importantly, shortened convergence times only provide benefit under finite time horizons, since each technique will ultimately converge to the optimal solution.

Our simulations show that out of the algorithms studied, the centralized ETP algorithm provides the best performance at the highest feedback cost. The C-ETC algorithm, which does not adapt to the dynamic environment, is unable to match the performance of algorithms which do adapt to the environment. This performance gap is emphasized when analyzing the performance post convergence, once the algorithm attempts to exploit the information it has learned. This is due to the fact that the ETP algorithm is able to utilize information about the target to predict future range and SINR.

Our results show that H-ETP demonstrates the best trade-off between performance and feedback. For a moderate level of feedback (less than two floating-point values per CPI), H-ETP exhibits tracking performance almost as good as the best-performing algorithm C-ETP, which uses over 30 floating-point values per CPI.

In addition, these centralized algorithms show convergence times on the order of 100 CPIs, while prior work [55] required up to 2500 CPIs to meet the same target tracking performance without feedback. This improved convergence time translates directly to improved performance (i.e., track error in early CPIs) in a dynamic environment, where new optimal matchings may need to be identified rapidly. Further, improved convergence times cause less cumulative track error, since the target is accurately tracked much sooner. These improvements come at a cost of communication between the nodes and CC, as well as an implicit reliance on the CC. We demonstrated that there is a trade-off between the amount of feedback and the performance of the network, and that a minimal amount of feedback is sufficient to cause dramatic gains over the prior art.

While this work focused on a single-target scenario, a multi-target extension is straightforward, as the SINR will still be dependent on target location and environmental interference. However, as the number of targets increases, the amount of target parameter estimates sent from each node to the CC will also increase. For this reason, future work could examine the impact of CC-level node selection for target state updating. Further, the multi-target problem could present opportunities for each node to conduct both active radar tracking and passive observation. Future work could also examine the possibilities of CRN mode control.

Chapter 5

Timely Target Tracking: Distributed Updating in Cognitive Radar Networks

Related Publications

The material in this chapter has been reproduced from the following publications:

- [53] **W. W. Howard**, A. F. Martone and R. M. Buehrer, "Timely Target Tracking: Distributed Updating in Cognitive Radar Networks," submitted to *IEEE Trans. on Radar Systems (Under Revision)*, 2023.
- [58] **W. W. Howard**, Charles E. Thornton, and R. Michael Buehrer, "Timely Target Tracking in Cognitive Radar Networks," in *2023 IEEE Radar Conference (RadarConf23)*, San Antonio, TX, USA, 2023.

5.1 Introduction

5.1.1 Cognitive Radar Networks

Cognitive radar networks (**CRNs**) consist of many cognitive radar devices ("nodes") which observe targets and report to a fusion center (**FC**). The literature has provided many cognition models for CRNs including completely distributed cognition [55] [51] [56] and centralized cognition [52] [57]. A topic of study for nearly two decades, the primary benefits of CRNs over traditional networked radar include spatial diversity, flexibility in dynamic environments, efficient usage of spectrum, and increased resilience to outages [6].

CRNs exist to gather information from targets in a possibly geographically diverse region. Cognitive radio was first proposed to address the need for communication systems to be adaptive in congested and possibly contested spectrum. The term "cognitive" appeals to the biological decision-making cycle as described variously [25] [85] [86]. In particular, cognitive radios employ the "perception-action cycle," observing their environment and taking actions

which are predicted by some internal model to positively impact performance. The current work employs cognitive processing to evaluate dynamic target behavior, reduce utilization of a shared communication resource, and improve the quality of FC estimation when using a network of radars.

Other works have conducted research in the area of radar network *resource allocation*. Several works consider the application of the Bayesian Cramér-Rao Lower Bound **(BCRLB)** [132], which is a bound on the variance of any unbiased estimator. The BCRLB is a very useful tool in the radar context is at allows a device to estimate the expected error for proposed parameters. A specific implementation of the BCRLB as an optimization criteria has been in the field of *power allocation*. Yan et al. consider the use of the BCRLB and develop a power allocation scheme for a radar sensor network [133] [134]. They consider constrained optimization of network power subject to a minimum tracking error, determined by the BCRLB. They also provide a survey of resource allocation techniques for radar sensor networks [135]. Rather than seeking to allocate resources between heterogeneous radars in a network, which each may have different operating parameters and performance, we assume that our network is composed of similar cognitive radar nodes. This means that problems such as power allocation will not have the same trade-offs as in a heterogeneous network. We study the allocation of communication resources in order to optimize the timeliness of updates. Further, the prior work does not consider the impact of heterogeneous *targets* (in terms of motion model) on the resource optimization nor communication overhead required to achieve these gains. We seek to fill that void through this work.

Sensor collaboration [136] is the process of collecting target information from low-power ad-hoc distributed sensor networks, often using graph theory to determine the optimal path for sensor information or for sensor querying. This style of target tracking differs from our problem. Instead of selecting the sensors (radar nodes) which should make target measurements, we are optimizing the time steps at which radar nodes send updates to a central fusion center.

5.1.2 Single Node Techniques

Single-node cognitive radar (i.e., not networked) has been investigated even more in the literature. Examples of proposed problems include waveform design optimization [2] [20] [26] [118], dynamic spectrum access (**DSA**) [27] [28] [29], and improved target detection [3] [63]. Commonly, cognitive radar problems include a reliance on *online optimization* - sequential processes through which performance is improved. Online learning is more effective because offline learning is ineffective without high quality prior knowledge.

Another branch of the single-node cognitive radar investigates the performance of co-located multiple-input multiple-output (**MIMO**) radar systems using multiple beams for target observation [137] [138]. The authors use convex optimization to separate the nonconvex allocation problem into several convex problems which may be solved optimally, and show

that the worst case single-node tracking accuracy can be significantly improved.

In particular, there has been a study of meta-cognitive techniques in single-node cognitive radar [90], [139], where a higher-level process analyzes the performance of different cognitive agents and adaptive selects those agents which exhibit superior performance. The work of [140] analyzes the specific trade-off between a learning-based approach to DSA in cognitive radar and a fixed rule-based approach, concluding that there are regimes in which either technique is dominant.

Multi-target tracking is the segment of the literature which addresses problems of data association and track maintenance [141]. Data association [73] becomes necessary when a sensor receives measurements of multiple targets (with some probability of missed detections and some probability of false alarms) and must *associate* each measurement with a target track. The probability hypothesis density (**PHD**) filter [142] [143] [76] has been proposed as a method of interpreting multi-target sensor data and forming target tracks. The PHD filter frames detections as random finite sets and uses the first-order moment to determine an intensity function over the target state space. A tractable PHD filter implementation was demonstrated in [72], which uses Gaussian mixture (**GM-PHD**) models in the first closed-form solution to PHD filtering. Since in our work target association is performed at the individual radar nodes and is not dependent on update frequency or update times, target association performance will have limited impact on the performance of node selection policies. Similarly, since we model target birth and death as isotropic processes (i.e., having constant probability throughout space and time), the probability of track initiation has no impact on the node selection algorithms.

Our work focuses on downstream systems from multi-target tracking. In particular, we consider questions such as "How often should the FC receive updates on each target?" and "How do nodes decide when to send updates?".

5.1.3 Age of Information

First proposed in [144] and gaining considerable traction recently, Age of Information (**AoI**) tools are popular when information freshness is desired. The survey by Yates et al. [145] covers recent contributions and applications, characterizing AoI as "performance metrics that describe the timeliness of a monitor's knowledge of an entity or process."

Information freshness can be quantified via an "age process." Typically, an age process $\Delta(t)$ increases linearly in time (i.e., aging at a rate of one second per second). In the literature from which we draw our approaches, age processes relate to the time since an aggregator received updates from an observer on a process. So, when the observer provides an update to the aggregator, the age of the process is reset to 0.

AoI has been implemented particularly often to solve problems in the field of *federated learning*, where a central parameter server attempts to train a large machine learning (**ML**) model

using numerous independent clients. Federated learning is most important in those domains where *data privacy*[1] is essential, e.g. medicine or other personal identifiable information. Information freshness is utilized in this field to ensure the global model is updated by the most recent data. Our current work does not have the same purpose. Information freshness remains critical in CRNs to provide the most recent and accurate data to operators or downstream systems, and data security remains important, but not data privacy.

Another field frequently finding use for AoI is distributed sensor networking, where multiple devices observe one or many processes. In [146], the authors describe a network of UAVs which assist an IoT-enabled network utilizing an AoI metric to maximize information freshness. In this and several similar works [147], [148], a scheduler must assign resources to each of several nodes. The first part of our work is similar to these, where we develop a centralized decision metric. However, the second part of our work derives a distributed technique which differs from these.

The Age of Incorrect Information (**AoII**) [149] is a recent metric which proposes to "extend the notion of fresh updates to that of fresh *informative* updates." Rather than measuring the time from the last update, AoII considers the information content of updates and in particular those which bring *new and correct* information to the aggregator. This recognizes the fact that Markovian sources may not have new states at all times, so an observer might not need to continuously update an aggregator. A new update which contains identical information to the previous update is not necessarily useful. AoII has been utilized several times for centralized tracking of remote sources [150], [151], but to the best of our knowledge it has not been adapted to the distributed, multi-process tracking problem.

5.1.4 Problem Summary

Due to the possibly large and diverse geographic region under observation by the CRN, it follows that there may be nodes which see no targets and other nodes which see many targets. Similarly, there may be targets which are seen by many nodes and targets which are seen by no nodes. The quality of each node's observations of a given target (*target observability*) will be influenced by large-scale channel conditions such as path length and small-scale channel conditions such as target frequency selectivity and angle. These effects are greatly impacted by the network geometry - the physical distances between radar nodes and their interactions with the environment and targets. We utilize a model driven by *stochastic geometry* to describe these effects.

As the scene evolves in time, targets may exhibit dynamic motion, moving unpredictably. Similarly, the number of targets existing in the region need not stay fixed; targets are able to enter and exit the region. We model the targets as uncrewed aerial vehicles (**UAVs**) with

[1]The distinction between data privacy and security is that while secure systems may share data, data privacy systems may not share data.

independent and identically distributed (**i.i.d.**) motion models, which may take off or land from anywhere in the considered region. The motion models we consider are discrete-time Markov chains, where the probability of a transition to a different state of motion (e.g. linear or turning) depends only on the current state. We'll discuss a centralized polling process by which a node can communicate to the FC whether a target in their region has exhibited "interesting" behavior, such as a change in motion model state. Then, we will present a distributed process through which a node can decide whether or not to provide an update to the FC without any centralized intervention, utilizing information from target motion modeling. We refer to this stochastic target motion as *maneuverability*.

The AoI inspired approaches we discuss in this and in our previous work [58] are intended to minimize the amount of time that the aggregator (FC) has out-of-date information on target processes (subject to constraints), and thereby minimize the error of the target state estimate which is provided to an operator or downstream system. If there are sufficient resources then the obvious and optimal solution is to send updates in every possible instance. However, as spectrum is inherently scarce, it is impractical and inefficient for the nodes to update the FC state in every possible instance. So, we impose an *update rate constraint* on each node in order to keep the average utilized communication rate below a limit. The update rate is constrained due to congested spectrum; as we will describe, the network must share spectrum resources with other devices and is limited to some fraction of the available resources.

We discuss how the consideration of target observability, maneuverability, and this update rate constraint impact the development of our AoI metric. The primary *tool* we'll use is the AoII [149], which approaches this type of problem using a Markov chain model, providing a Bellman-optimal update policy. The primary *modifications* we must make to AoII are consequences of scale; AoII was developed for single-process, single-observer systems and we consider the case where multiple nodes observe multiple targets. We measure the effectiveness of our proposed techniques using tools from the AoI literature as well as by calculating the tracking error achieved by the FC. Since this problem has not been addressed in the literature, we contextualize our results by comparison to likely candidates - namely, an approach based on iterative reinforcement learning (multi-armed bandits) and a random selection algorithm.

5.1.5 Contributions

This problem, which has not yet been addressed in the literature save for our initial work [58], resembles scheduling problems where a central server must collect information from distributed nodes. Consequently, we borrow from the AoI literature and develop two decision metrics. To the best of our knowledge, this work represents the first consideration of AoI metrics in CRNs. Specifically, we contribute the following:

- A centralized "track-sensitive AoI metric," which utilizes a polling process to allow the FC to select nodes to provide updates in each update interval.

- An adaptation of AoII to enable distributed decision-making, where each node implements a Bellman-optimal policy to determine when to provide updates. In contrast to the centralized solution, where the FC coordinates all interactions, the distributed solution relies on each node to decide when to send updates. Specific adaptations to the AoII algorithm include a modified Markov model and rate limits for multiple nodes.
- Mathematical analysis of our proposed techniques.
- Numerical simulations to support our conclusions.

5.1.6 Notation

We use the following notation. Matrices and vectors are denoted as bold upper $\mathbf{X}$ or lower $\mathbf{x}$ case letters respectively. Element-wise multiplication of two matrices or vectors is shown as $\mathbf{X} \odot \mathbf{Y}$. Functions are shown as plain letters F or f. Sets $\mathcal{A}$ are shown as script letters. Denote the Lebesgue measure of a set $\mathcal{A}$ as $|\mathcal{A}|$. When we wish to show the number of elements in a (finite) set $\mathcal{A}$ rather than its measure, we use the cardinality $\#(\mathcal{A})$. The transpose operation is $\mathbf{X}^T$. The backslash $\mathcal{A} \backslash \mathcal{B}$ represents the set difference. Boxes (intervals) in $\mathbb{R}^d$ are written as $[a, b]^d$ and when the elements of a set are denoted, they are given as $\mathcal{A} = \{a, b, c, \dots\}$. Random variables are written as upper-case letters X, and their distributions will be specified as $X \sim \mathcal{D}(\cdot)$. The letter $U[a, b]$ is used to denote the uniform distribution on an interval $[a, b]$. The set of all real numbers is $\mathbb{R}$ and the set of integers is $\mathbb{Z}$. The speed of electromagnetic radiation in a vacuum is given as c. The Euclidean (ℓ^2) norm of a vector $\mathbf{x}$ is written as $||\mathbf{x}||_2$. Estimates of a true parameter p are given as $\hat{p}$.

5.1.7 Organization

The remainder of this work is organized as follows. In Section 5.2, we discuss prior work in this area and develop relevant models. In Section 5.3 we derive a technique for centralized decision-making. Then, in Section 5.4 we discuss our AoII-based method for distributed updating. Section 5.5 provides numerical simulations and in Section 5.6 we draw conclusions and suggest future work.

5.2 System Model

In this section we will describe a general framework which will inform the development of our algorithms. We'll also cover mathematical preliminaries which will be required later. While these models make some specific assumptions (e.g., on the motion of targets), we strive to avoid any assumptions that would limit the applicability of our work.

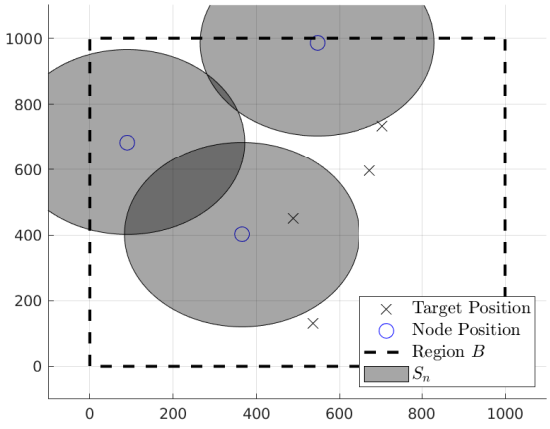


Figure 5.1: Tracking scenario with node density $\lambda_n = 3$ and target density $\lambda_m = 7$.

5.2.1 Spatial Modeling

Sensor networks (and more broadly wireless networks) are often analyzed using tools from *stochastic geometry*. This is the field of study which considers the mathematical and statistical relationships between spatial processes. Poisson point processes (**PPPs**) are of particular interest, due to their generalizabililty and mathematical tractability. A point process is a random collection of points in space. As provided in [69], a PPP on $\mathbb{R}^d$ with *intensity measure* Λ and *density* λ per unit area can be simulated on a compact region[2] $B \subset \mathbb{R}^2$ with measure $|B|$ by drawing a Poisson number N with mean $\lambda|B|$, and place N points uniformly at random in B. Formally,

$$Pr(N = n) = \frac{\lambda|B|^n e^{-\lambda|B|}}{n!} \tag{5.1}$$

In other words the number of points inside a compact region is distributed according to the measure (or size) of that region.

For numerous reasons (e.g., power limits, geographic obstructions and pulse range gating), the practical range of any radar node is limited. So, we model each radar node n as covering a specific region S_n with measure $|S_n|$. We will discuss later the fact that the FC does not need to know the shape of this region if $|S_n|$ is known.

Consider a compact region $B \subset \mathbb{R}^2$. Inside this region, targets are positioned according to a PPP with density λ_m. The target positions evolve in time according to the motion model discussed below. Further, inside the compact region B, we place a number of radar nodes according to a PPP with density $\lambda_n < \lambda_m$. Let M be the Poisson random variable describing the number of targets, and note that $\overline{M} = \lambda_m|B|$ is the mean of this distribution.

[2]Compactness is the only required condition, and the region can be disjoint. However, the procedure to simulate a PPP is most straightforward in rectangular regions

For a given instance of the model, we collect all of the targets into the set $\mathcal{M}$ and index them as $m \in \mathcal{M}$. When we wish to refer to the specific coordinates of a target m, it is denoted as X_m. Similarly, we collect each radar node n into the set $\mathcal{N}$, letting N with mean $\overline{N}$ be the random variable which describes the number of radar nodes. X_n represents the coordinates of target n. When we conduct numerical simulations of stochastic processes, a single simulation consists of a single realization of each distribution. The resulting set of points is referred to as a point pattern.

The *coverage* of a wireless network is the union of the coverage of each node. In general, each node $n \in \mathcal{N}$ covers a compact region S_n, which is a subset of B. The covered regions are drawn i.i.d. from a spatial distribution[3]. In the typical *Boolean model* [69], it is particularly assumed that each node in a network covers a disk of fixed radius r. The probability that a location x is covered by the network is then

$$Pr(x \in \bigcup_{n \in \mathcal{N}} S_n) = 1 - e^{-\lambda_n \mathbb{E}|S_n|} \tag{5.2}$$

$$= 1 - e^{-\lambda_n \pi r^2} \tag{5.3}$$

This means that the probability any target is covered is given as Eq. (5.2). Further, for a single instance of this model, the number of radar nodes which cover a given target is Poisson distributed with mean $\lambda_n \pi r^2$. Note that unlike the target spatial distribution, the coverage is constant in time since this is a general result for any location x. Since each target has a less than unity probability of being observed, some number of targets will be unobserved by the network in each time step. That probability can be found as Eq. (5.4).

$$\#(m \notin \bigcup_{n \in \mathcal{N}} S_n) \sim \text{Poisson}\left(\lambda_m e^{-\lambda_n \pi r^2}\right) \tag{5.4}$$

Figure 5.1 shows an instance of such a network, with a target density $\lambda_m = 20$ and a node density $\lambda_n = 10$. The coverage of each node is a circle centered on the node with an area of $|S_n| = 0.2|B|$.

5.2.2 Motion Modeling

We model the motion of the target UAVs according to a multi-state Markov chain [152]. While in general many types of motion could be considered, in this work we focus on constant-velocity and constant-turn motion. When initialized, each target $m \in \mathcal{M}$ draws probabilities $P_{m,1} \sim U[0.7, 0.9]$ and $P_{m,2} \sim U[0.5, 0.7]$ to form a transition matrix:

$$T_m = \begin{bmatrix} P_{m,1} & 1 - P_{m,1} \\ 1 - P_{m,2} & P_{m,2} \end{bmatrix} \tag{5.5}$$

[3]Commonly balls of random radii are used for i.i.d. coverage but in general S_n need only be compact with $|S_n| > 0$.

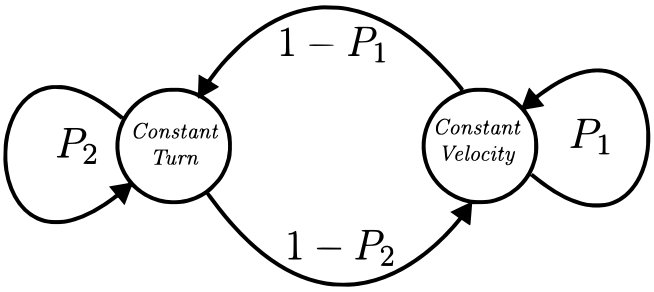


Figure 5.2: The Markov chain motion model exhibited by UAV targets.

Figure 5.2 demonstrates this Markov chain. The distributions over transition probability can be tailored such that one state is more common than others - specifically, such that constant velocity motion is much more common than constant turn motion.

Assumption 2. The motion for each target evolves according to a Markov chain with constant transition probabilities, drawn i.i.d. for each node. The radar nodes must estimate these probabilities.

5.2.3 Network Modeling

As discussed, the network consists of the set of radar nodes $\mathcal{N}$ which attempt to track the set of targets $\mathcal{M}$, feeding the information to the FC. Figure 5.3 shows a version of this network: several targets are in the scene, with some observed multiple times and some not observed at all. The shortest time interval considered is the Coherent Pulse Interval (**CPI**), during which each radar node emits several pulses, coherently integrates them, and performs signal processing to extract estimated target parameters (position and velocity). The CPI duration for each node n is constant, but they need not start simultaneously. The nodes report their observations to the FC as updates. Importantly, since the nodes use monostatic radar and process the received signals locally, the updates only contain position estimates for each target. This lessens the need for synchronization in the network; the FC combines detections rather than pulses, so does not require tight timing. Further, it is not necessary for the node CPIs to start at the same time and be of the same duration.

Measurement Model

The FC processes new updates at the same rate that CPIs occur (i.e., 1/CPI). All updates are assumed to arrive at the end of the update interval. In the distributed approach, the FC processes all updates received during an update interval at the end of the interval.

An update at time t from node n consists of that node's current estimate for each target m in the region S_n. Based on the target range and aspect, node n observes each target m with

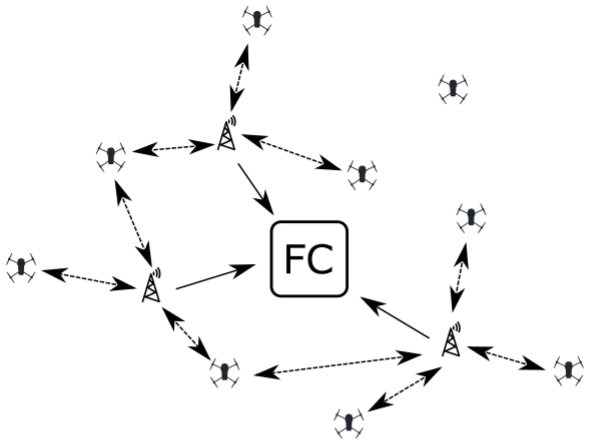


Figure 5.3: Example network showing which nodes can observe which UAVs. Some targets are observed by multiple nodes and some targets are not observed at all.

a variance $\sigma_{n,m}$ and a detection probability P_D. False alarms are generated randomly with fixed probability P_{FA}. Detections are associated with target tracks, and new target tracks are formed for new detections. The node n can gather these targets into the set $\hat{\mathcal{X}}_n^{(t)}$. Nodes estimate three main quantities for a target m:

- The target position X_m.
- The target velocity $\dot{X}_m$.
- The target motion model state γ_m.

Taken together, these quantities form a vector

$$\mathbf{X}_m^{(t)} = \left[X_m^{(t)}, \dot{X}_m^{(t)}, \gamma_m^{(t)}\right] \tag{5.6}$$

which node n estimates, forming

$$\hat{\mathbf{X}}_m^{(t)} = \left[\hat{X}_m^{(t)}, \hat{\dot{X}}_m^{(t)}, \hat{\gamma}_m^{(t)}\right] \tag{5.7}$$

for all $m \in \hat{\mathcal{M}}_n^{(t)}$. We use the set $\hat{\mathcal{X}}_n^{(t)}$ to refer to the update provided by node n at time t. Denote as $\mathcal{T}_{m,n}^*$ (with $\#(\mathcal{T}^*) \leq T$) the time steps $t < T$ where node n detected target m. Only targets m such that $\#(\mathcal{T}_{m,n}^*) > 2$ are included in the update to avoid false tracks.

The FC forms tracks of all targets it has observed, using Kalman filtering to improve position estimation. Denote as

$$\overline{\mathbf{X}}_m^{(t)} = \left[\overline{X}_m^{(t)}, \overline{\dot{X}}_m^{(t)}, \overline{\gamma}_m^{(t)}\right] \tag{5.8}$$

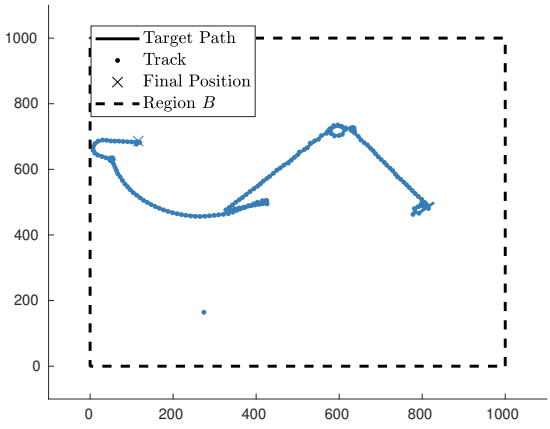


Figure 5.4: A sample fused track from the perspective of the FC.

the FC's estimate of target m. For each $m \in \mathcal{M}_n^*$, node n forms an Interacting Multiple Model (IMM) filter [153], a type of tracking Kalman filter which evaluates the probabilities of multiple motion models over time. This filter estimates the motion model transition probabilities for target m and forms state estimates based on the current estimated motion model. Figure 5.4 shows the fused FC estimate for a single target's track. Note that if missed detections occur, the node propagates the Kalman filter for the target and provides the predicted values in its update. False tracks are avoided by requiring several measurements before a track is confirmed. Only target detections are used to update motion model probabilities, not Kalman filter estimates.

Let $\mathcal{N}^{(t)}$ denote the nodes providing updates in the update interval ending at time t. Then, the FC receives updates on all targets $m \in \mathcal{M}^{(t)}$, where

$$\mathcal{X}^{(t)} = \bigcup_{n \in \mathcal{N}^{(t)}} \hat{\mathcal{X}}_n^{(t)} \tag{5.9}$$

Finally, we have set $\mathcal{X}_{FC}^{(t)}$ which is the set of all target tracks formed at the FC. Note that $\mathcal{X}_{FC}^{(t)} \supseteq \mathcal{X}^{(t)}$ since each target may not be updated at all t. The FC maintains a record of the *age* of each target track. Let v_m denote the most recent time for which $m \in \mathcal{X}^{(t)}$. Then, the age of the FC track for target m can be written as Eq. (5.10). Accordingly, when $m \in \mathcal{X}^{(t)}$, $\Delta_m(t) = 0$, which is the smallest value Δ can take.

$$\Delta_m(t) = t - v_m \tag{5.10}$$

Update Policies

The network communicates over a shared communications resource, which is divided into R resource blocks per CPI. Other devices are assumed to be present, occupying some amount of

the shared resource. In order to maintain performance and avoid unnecessary interference, the target tracking network is limited to some fraction $f < 1$ of the available spectrum resources. Let $C + 1 = fR$ be the tracking network's update capacity per CPI. Then, C resource blocks are used by the nodes to transmit updates, and one resource block is used by the FC to transmit any feedback. Since the problem is trivial if $C > N$ (there is sufficient capacity for each node to provide updates in each CPI), we will assume that $C < N$. Prior work [51] has established techniques for spectrum sharing in CRNs. One resource block is sufficient to allow a node to transmit updates on the expected number of targets in each region, $\lambda_m|S_n|$. Since the resource blocks must be assigned on-the-fly, we will assume that one resource block is required for a single node to provide any number of target updates.

Another way of phrasing this, more relevant to Section 5.4, relates this network rate limit to a limit on the rate of each node. If the network update rate is C and there are $\mathbb{E}(N) = \lambda_n|B|$ nodes, each node is allocated a rate of $\frac{C}{\lambda_n|B|}$ updates per CPI. Since $C < N$, let

$$\alpha = \frac{C}{\lambda_n|B|} \tag{5.11}$$

with $0 < \alpha < 1$ be the (average) number of updates each node can provide per CPI. Let ϕ be a *policy* which determines the actions that node n takes. Then, define an action $\Psi_n^\phi(t)$

$$\Psi_n^\phi(t) = \begin{cases} 1, & n \in \mathcal{N}^{(t)} \\ 0, & n \notin \mathcal{N}^{(t)} \end{cases} \tag{5.12}$$

which indicates whether node n sends an update during the CPI ending at time t.

Definition 5.1 (Constrained Policy)**.** A policy ϕ is said to be constrained if Eq. (5.13) holds.

$$\limsup_{T\to\infty} \frac{1}{T}\mathbb{E}^\phi\left(\sum_{t=0}^{T-1}\Psi_n^\phi(t)\right) \leq \alpha \tag{5.13}$$

Definition 5.2 (Fixed Policy)**.** A policy ϕ is said to be *fixed* if $\#(\mathcal{N}_\phi(t)) = C\ \forall t$.

Lemma 5.3 (Fixed Policies are Constrained)**.** *Let policy ϕ be fixed, so that $\#(\mathcal{N}_\phi(t)) = C\ \forall t$. Then, ϕ is constrained.*

Proof.

$$\begin{aligned}
\limsup_{T\to\infty} \frac{1}{T}\mathbb{E}^\phi\left(\sum_{t=0}^{T-1}\Psi_n^\phi(t)\right) &= \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}^\phi\left(\frac{TC}{N}\right) \\
&= \limsup_{T\to\infty} \frac{1}{T}\left(\frac{TC}{\mathbb{E}(N)}\right) \\
&= \frac{C}{\lambda_n|B|} \\
&= \alpha
\end{aligned}$$

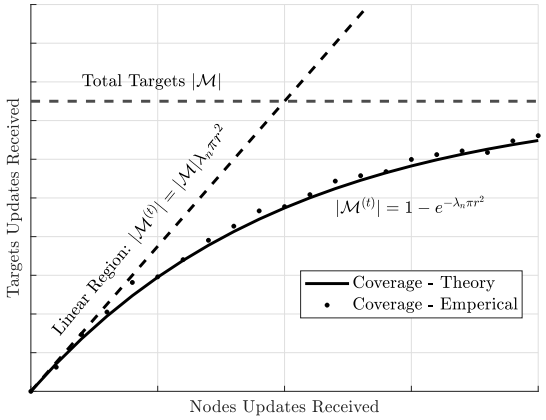


Figure 5.5: As the update rate for each node increases, the number of targets updated increases more slowly.

so ϕ is constrained. □

The next two sections focus on forming the set $\mathcal{N}^{(t)}$, first in a centralized manner and then allowing each node to decide when to send an update.

5.3 Centralized Policy

Using a polling process, the FC can determine the set $\mathcal{N}^{(t)}$ using the optimization process detailed in this section. Specifically, this polling process uses the feedback channel to first poll each node, and then to command that node to provide an update. The polling process allows each node to inform the FC if it has observed an "interesting" update, which we define as a target entering the environment, exiting the environment, or changing motion model states. When a node has an interesting update it is added to the set $\mathcal{A}^{(t)}$.

While the FC wishes to maintain low-age, high-accuracy tracks for all targets, it can only receive updates on these targets by utilizing a set of nodes $\mathcal{N}^{(t)}$ which may provide duplicate updates. Generally, due to the spatial coverage model discussed in Section 5.2.1 and as shown in Figure 5.5, the number of updated targets asymptotically approaches the total number of targets as the number of updated nodes increases. Each target may be observed by multiple nodes so as more nodes provide updates, some targets may be updated more than once. Each target may be updated more than once; as discussed in the coverage model, the number of times any point in the region B is covered is Poisson distributed. So, *more than one node may cover any given target* and *each node may cover more than one target*. The natural conclusion is that some nodes will provide updates more frequently than others. We impose a capacity limit on the centralized network, where C nodes can provide updates

per second. The centralized approach is a *fixed policy*: the number of nodes selected for updates is constant.

From the perspective of the FC, we have the set of radar nodes $\mathcal{N}$, where each node n can observe each target $m \in \mathcal{M}_{FC}^{(t)}$ with a variance $\sigma_{n,m}$. When a node is unable to observe a target, the variance is set to a sufficiently high value. This is the target *observability*. From the polling process, the FC also has $\mathcal{A}^{(t)}$, the set of nodes with interesting updates. This represents the target's *maneuverability*.

The FC also records the age of each target track for each time t as $\Delta_m(t)$. As the FC is only aware of targets $m \in \mathcal{M}_{FC}^{(t)}$, these are the only targets for which age information is maintained. Since a target may leave the scene and no longer be observed, we set a maximum age value Δ^{max} and remove any target with greater age from consideration. Formally,

$$\mathcal{M}_{FC}^{(t)} = \mathcal{M}_{FC}^{(t-\tau)} \backslash m \quad \forall m \text{ s.t. } \Delta_m(t) \geq \Delta^{max} \tag{5.14}$$

Since the FC is not aware of missed detections or false tracks, they are not considered here.

A naive age- or quality-greedy approach may select the nodes which observe the most maximum-age or minimum-variance targets respectively. The centralized objective function we use can be formed as Eq. (5.15). The product of the target age and node tracking variance forms the base of the objective, with an additional factor of $\alpha \in [0, 1]$ when the node in question isn't "available." There is an additional discount of $\gamma \in [0, 1]$ in the case where target m is not observable by node n ($m \notin S_n$).

$$\begin{aligned}
\mathcal{N}_{cent}^{(t)} &= \max_{\mathcal{N}^{(t)} \in \mathcal{N}} \sum_{n \in \mathcal{N}^{(t)}} Q(n) \\
\text{s.t. } & \#(\mathcal{N}_{cent}^{(t)}) = C \\
& Q(n) = \sum_{m \in \mathcal{M}^{(t)}} \tilde{\alpha}_n F_{n,m} \\
& F_{n,m} = \begin{cases} \Delta_m(t)\sigma_{n,m}^{-1}, & m \in \hat{\mathcal{X}}_n^{(t)} \\ \gamma, & m \notin \hat{\mathcal{X}}_n^{(t)} \end{cases} \\
& \tilde{\alpha_n} = \begin{cases} 1, & n \in \mathcal{A}(t) \\ \alpha, & \text{else} \end{cases}
\end{aligned} \tag{5.15}$$

This objective function optimizes the number of target tracks updated, the FC age of those tracks, and the variance with which each node measures that target. In other words, if a target is observed at high variance by one node and low variance by another node, the FC will be more likely to select the low variance observation for updating. Also, if one node observes more targets than another, it will be more likely to be selected by the FC. The FC can only pick C nodes per update period, so this policy cannot exceed the resource constraint. The

term $\tilde{\alpha}_n$ indicates whether or not node n has an "interesting" (targets entering, exiting, or changing motion model state) update. If there is no interesting update at a given node there will be a penalty, but it may still be selected.

A straightforward technique to solve this objective is to form a bipartite matching problem. The FC can form a matrix of edges $\mathcal{E}$ between the two vertex sets $\mathcal{N}$ and $\mathcal{M}_{FC}^{(t)}$. Each index $\mathcal{E}_{n,m}$ is set to $\tilde{\alpha}_n F_{n,m}$. The FC then fills the set $\mathcal{N}_{cent}^{(t)}$ of C nodes by selecting $n = \max_{n \in \mathcal{N}} Q(n)$ until the set is full. As each node is selected, the edges corresponding to the selected targets are set to 0 to maximize the number of target updates received. Algorithm 10 shows all of the relevant steps.

Since the optimization is subject to $\#(\mathcal{N}_{cent}^{(t)}) = C$, the centralized policy is fixed and therefore by Lemma 5.3 is constrained.

Algorithm 10: Track-Sensitive AoI Node Selection

Poll each node to form $\mathcal{A}(t)$

Form $\mathcal{E}(t)$, where

$$\mathcal{E}(t)_{n,m} = \tilde{\alpha}_n F_{n,m} \tag{5.16}$$

for *c=1:C* **do**

$$\mathcal{N}_{cent}^{(t)} = \mathcal{N}_{cent}^{(t)} \cup \max_{n \notin \mathcal{N}_{cent}^{(t)}} Q(n)$$

$$\mathcal{E}(t)_{:,m \in \mathcal{M}_n^{(t)}} = 0$$

Return $\mathcal{N}_{cent}^{(t)}$

5.4 Distributed Age of Incorrect Information Policy

In the distributed approach, each node is added to the set $\mathcal{N}^{(t)}$ if it provides an update at time t, determined by a metric which is internal to each node. For several reasons, distributed updates involve greater complexity than centralized selection. Since communication is limited in the network, each node is not privy to the observations of other nodes. In addition, each node is not aware of the times at which other nodes provide updates. This implies that since a target *may* be observed by more than one node, it is not possible to know with certainty the state of the FC. In other words, the FC may have more up-to-date information than a single node believes, leading that node to overuse the communication resource and provide inefficient redundant updates.

A distributed approach is desirable for several reasons. Primarily, the FC does not necessarily have all relevant information. Since nodes are able to directly observe the target states evolving in time, moving the decision process towards the edge of the network allows more

accurate decisions to be made. As a consequence, more useful updates can be provided to the FC meaning that the spectrum resources can be more efficiently used. In this section, we will discuss how a policy can maintain a limit on the average update rate while offloading the updating decision to each node.

5.4.1 Preliminaries

Rather than rely on the AoI as described above, we adopt the Age of Incorrect Information. AoII provides an analytically tractable method for a node to provide updates to a FC without direct control from the FC. We make several modifications to extend single-observer AoII to the multi-node scenario.

In general, the AoII for a Markov process $X(t)$ is written as

$$\Delta_{AoII}(t) = f(t) \times g\left(\widehat{\mathbf{X}}(t), \overline{\mathbf{X}}(t)\right) \tag{5.17}$$

where $f(t)$ denotes a *time penalty* function, and $g(\widehat{\mathbf{X}}(t), \overline{\mathbf{X}}(t))$ an *information penalty* function. The penalty function takes the current state $\widehat{\mathbf{X}}(t)$ and the *last updated state* $\overline{\mathbf{X}}(t)$ as inputs, and represents the distance between a node's current state and the last known state of the FC. When these two states are equal, the FC's tracking error will be limited by its Kalman filtering, even when it receives new data. However, when the target motion model changes, the FC's tracking error will increase. For the *distributed* timely target tracking problem, we must develop the idea of a *target state* and *target age* somewhat further. We discuss the time and information penalty functions more later.

Due to Assumption 2, we know that the true motion of a target m evolves in time according to a two-state Markov model with fixed transition probabilities $P_{m,1}$ and $P_{m,2}$. A general model could encompass many more states, but a two-state model is sufficient to describe the dynamics. When the transition probabilities are both small, the target's motion will update less frequently. We'd like for the motion model to update much less frequently than the duration of a single update period, so that the track quality remains high.

The update frequency is better expressed by the *entropy rate* of the Markov chain. The entropy rate describes the rate at which a Markov chain changes states. More formally the entropy rate is given as Eq. (5.18) for a Markov chain with transition matrix T and stationary distribution μ. The stationary distribution describes the asymptotic probability that a Markov chain takes a certain state - the portion of time that the Markov chain spends in that state.

$$H(T) = -\sum_{i,j} \mu_i T_{i,j} \log_2 T_{i,j} \tag{5.18}$$

This is shown in Figure 5.6. When the entropy rate is low, and once the IMM Kalman filter receives enough samples, it can estimate the target state with high accuracy. However, when the entropy rate increases, the best-case error also increases.

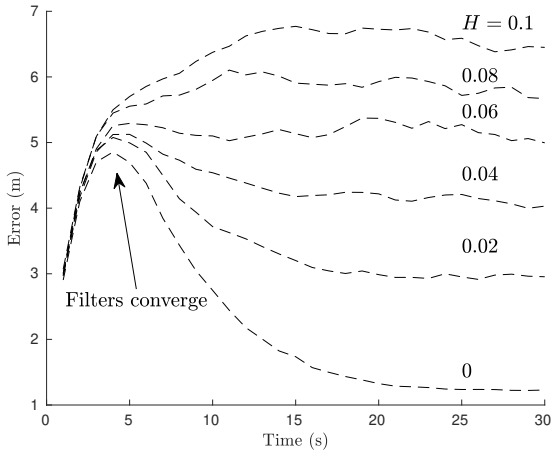


Figure 5.6: As the entropy rate of a target increases, tracking error decreases if the update rate is held constant.

5.4.2 Distributed Rate Limits

Another naive (but fixed) policy would cause every node to update on a round-robin style schedule (i.e., if the FC update capacity per second is C, each node $n \in \mathcal{N}$ sends an update every $\frac{C}{N}$ seconds). This would evenly distribute the capacity through the network. However, as discussed in the previous section, some targets (and therefore some nodes) will require more frequent updates than others. The question then becomes: How should the capacity be distributed through the network?

The Age of Incorrect Information [149] gives us some direction. AoII provides a policy ϕ which determines when an observer should send a target update to an aggregator, assuming one of each object.

The AoII formulation assumes that information from a single process must be collected by a single observer for transmission to an aggregator. However, in our problem, information from multiple processes are collected by multiple observers and efficiently relayed to the FC. So, there are two fundamental differences between our problem and AoII that must be accounted for. Firstly, each node observes possibly multiple targets. This can be addressed by extending the one-target Markov motion model to a multi-target model. Secondly, more than one node might make observations of a single target. Therefore, each node will be unaware of the true state of the FC and so might provide more updates than necessary. We address this by using the FC channel to communicate to each node the targets for which it is "responsible".

Target Assignment

After time t, the FC has received updates on targets $\mathcal{M}^{(t)}$, Eq. (5.9). The FC can determine $\mathcal{N}_m = \{n \text{ s.t. } m \in \hat{\mathcal{X}}_n^{(t)}\}$ for each target $m \in \mathcal{M}^{(t)}$. This is the set of nodes which can track target m. Then,

$$n_m^* = \underset{n \in \mathcal{N}_m}{\operatorname{argmin}} \left(||X_m - X_n||_2 \right) \tag{5.19}$$

represents the closest of the nodes which can see target m. Finally, the FC can use the feedback channel to inform node n of all targets m with $n_m^* = n$. Define $\mathcal{M}_n^* = \{m \text{ s.t. } n_m^* = n\}$ to be the set of all targets m for which node n is the closest node able to observe m.

Let a node n observe targets $\hat{\mathcal{M}}_n$ with $\#(\hat{\mathcal{M}}_n) = M_n$. Let s be a Markov motion model state (i.e., constant-velocity, constant-turn, etc.). Clearly, $\mathcal{M}_n^* \subseteq \hat{\mathcal{M}}_n$. Also, note that a Markov chain with states $\{s_1^1, s_2^1, \ldots, s_w^1\}$ and a Markov chain with states $\{s_1^2, s_2^2, \ldots, s_z^2\}$ can be combined to form a Markov chain with $w \cdot z$ states:

$$\{(s_1^1, s_1^2), (s_1^1, s_2^2), \ldots, (s_1^1, s_z^2), (s_2^1, s_1^2), \ldots, (s_w^1, s_z^2)\}$$

Also note that the probability of transitioning from state (s_1^1, s_1^2) to state (s_2^1, s_5^2) is $P_{s_1^1, s_2^1} P_{s_1^2, s_5^2}$.

Distributed Policy

Now, node n can determine 1) the target for which node n is responsible and 2) the Markov chain which models those targets. Recall the estimated motion state for target m, $\hat{\gamma}_m(t)$. Node n can estimate the probability that target m transitions from state i to state j as

$$P_{i,j}^m = \frac{\sum_{t \in \mathcal{T}_{m,n}^*} \delta_{i,j}^m(t)}{\sum_{t \in \mathcal{T}_{m,n}^*} \mathbb{1}_{\hat{\gamma}_m(t)=i}} \tag{5.20}$$

$$\delta_{i,j}^m(t) = \begin{cases} 1, & \hat{\gamma}_m(\max\left(\mathcal{T}_{m,n}^* < t\right)) = i \ \& \ \hat{\gamma}_m(t) = j \\ 0, & \text{else} \end{cases} \tag{5.21}$$

where $\mathcal{T}_{m,n}^*$ are time steps where node n detects target m. Then, node n forms a Markov chain $\hat{\Gamma}_n$ of $2^{\#(\mathcal{M}_n^*)}$ states and can estimate each transition probability as Eq. (5.22).

$$P_{i,j} = \prod_{i=1}^{M_n} P_{i,j}^m \tag{5.22}$$

This estimate improves as $t \to \infty$.

Let the penalty function Eq. (5.17) be defined by

$$f(t) = t - t_n^* \tag{5.23}$$

$$g(\widehat{\mathbf{X}}(t), \overline{\mathbf{X}}(t)) = \begin{cases} 0, & \hat{\Gamma}_n(t) = \hat{\Gamma}_n(v_n) \\ 1, & \text{else} \end{cases} \tag{5.24}$$

recalling that v_n is the most recent time for which $n \in \mathcal{N}^{(t)}$. Since the estimated motion model is not updated when detections are missed, the penalty function Eq. (5.17) does not increment and an update is not likely. The FC's estimate of the motion model of each target observed by node n can be estimated by $\hat{\Gamma}_n(v_n)$, the estimated motion model at the last time that node n provided an update.

The procedure outlined in [149] provides a threshold policy which is Bellman-optimal for this situation. In particular, a threshold p_0 is determined[4] as a function of the Markov chain transition probabilities and the update rate constraint. When the penalty function Eq. (5.17) is equal to p_0, an update occurs with a probability[5] ρ_a, and when the penalty is $p_0 + 1$, an update occurs with a probability ρ_b.

$$\rho_a = \frac{\alpha - A(p_0 + 1)}{A(p_0) - A(p_0 + 1)} \tag{5.25}$$

$$\rho_b = \frac{A(p_0) - \alpha}{A(p_0) - A(p_0 + 1)} \tag{5.26}$$

Note that while only the targets in $\mathcal{M}_n^*$ are used to determine update times, updates consist of all targets observed by node n, since resource blocks are allocated such that all targets observed by a node may be updated simultaneously. This means that the FC may, in fact, have more recent information on a given target than node n is aware. However, since there is a one-to-one mapping from each target to its closest node, each node must assume that it is the only one providing updates on each target and that it provides the most accurate updates.

Now, to determine the appropriate update constraint, first note that an AoII policy will not be a fixed rate policy since the FC does not have control over the size of $\mathcal{N}^{(t)}$. The best that can be hoped for is a mean of C. An AoII policy is however constrained, under Definition 5.1.

Lemma 5.4 (Constraint Equivalence)**.** *If the AoII constraint is*

$$\delta = \frac{\alpha}{M_n} \tag{5.27}$$

then the average number of updates will be C.

Proof. See Appendix C □

[4]The optimal threshold is determined by [149], Algorithm 1: "Optimal Threshold Finder. "

[5]$A(n)$ is given as [149], Eq. (32).

5.5 Simulations and Analysis

5.5.1 Baseline Approaches

In addition to the Age of Information and Age of Incorrect Information policies above, we also provide baseline results for comparison purposes using a multi-armed bandit algorithm, random selection, and a round-robin style approach. Multi-armed bandits are often implemented in this style of iterative, online problem solving. We select the Upper Confidence Bound (**UCB**) due to its general-purpose use [107]. UCB considers the variance of each target track, and selects nodes accordingly in an attempt to balance exploration of nodes and exploitation of low-variance tracks. This is an example of a track-greedy selection algorithm: there are *many* targets in the environment, and an algorithm which is over-fitted to track variance may be susceptible to poor performance due to neglecting high-age tracks or rarely sampled nodes.

At each update time t, the FC chooses nodes according to Eq. (5.28), where $N_t(n)$ is the number of times n has been selected before time t.

$$\mathcal{N}_{UCB}^{(t)} = \underset{\mathcal{N}^{(t)} \in \mathcal{N}}{\operatorname{argmin}} \left[\sum_{m \in \mathcal{M}} F_{n,m} + \sqrt{\frac{\log t}{N_t(n)}} \right] \tag{5.28}$$

The second alternative approach we demonstrate is random node selection. In each time step, the FC samples the appropriate number of nodes at random and without replacement. This represents a strategy that seeks to evenly spread the network capacity over the nodes; each node will be selected equally often. Unfortunately, due to the stochastic nature of the network, some nodes will may require more frequent updating and some less.

Lastly, the round-robin style approach selects the C nodes which have the oldest updates. All of these approaches are fixed policies and therefore are constrained, meeting the condition of Definition 5.1.

5.5.2 Results

First, we should note that since each policy is constrained (Def. 5.1), they all meet the resource constraint. Figure 5.7 demonstrates the average utilized capacity over many averaged simulations. Since the proposed distributed algorithm is not fixed (Def. 5.2), sometimes more or less than C nodes provide updates simultaneously. This is acceptable since the network can use on average C resource blocks for updates. Note that since $C = 2$ and $\lambda_n = 0.2$,

$$\alpha = \frac{C}{\lambda_n |B|} = 0.1 \tag{5.29}$$

is the update rate per node. Simulation parameters are listed in Table 5.1.

Table 5.1: Simulation Parameters

Variable	Description	Value
λ_n	Node Density per km^2	0.2
λ_m	Target Density per km^2	0.3
$\lvert B \rvert$	Simulated Region Area	100 km^2
$\lvert S_n \rvert$	Observable Region Area	10 km^2
C	Update Capacity	2
N/A	Averaged Simulations	120

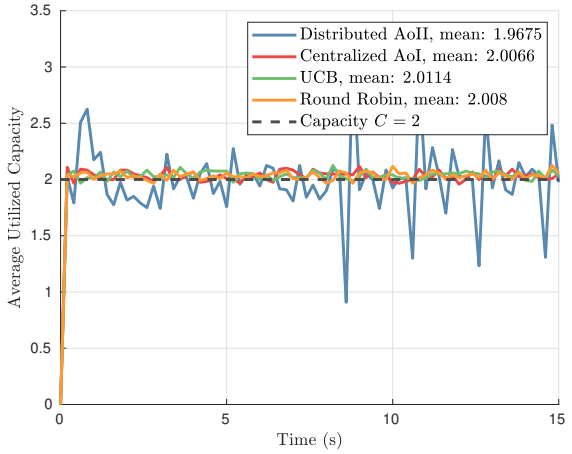


Figure 5.7: Each algorithm is constrained and therefore meets the average capacity $C = 2$.

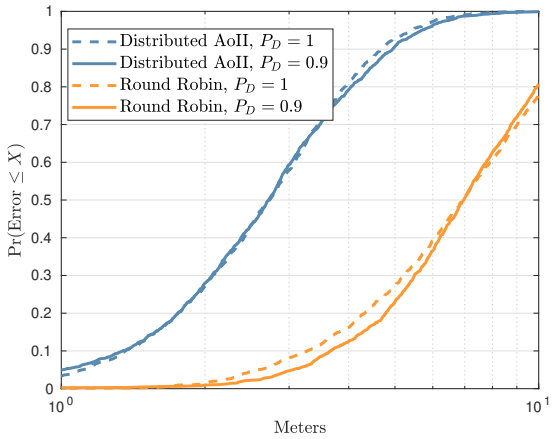


Figure 5.8: Tracking performance when the probability of detection is reduced to 0.9 with a capacity of 1. The performance of Round Robin is somewhat degraded, but the performance of the distributed AoII policy does not suffer much. This is because when a node misses a detection, it is much less likely to provide an update (as its own information is less fresh).

Missed Detections and False Alarms Radar detection and tracking problems contain an implicit trade-off between missed detections and false alarms. Missed detections occur when a target response falls below the detection threshold, and false alarms occur when the clutter or noise response is above the detection threshold. Of these, missed detections are more problematic, so detection threshold values which provide $P_{FA} < 10^{-4}$ are reasonable [154]. We show in Figure 5.8 that the performance of the distributed AoII policy is only mildly degraded when $P_D = 0.9$ and $P_{FA} = 10^{-3}$. This is because, under the AoII policy, nodes which miss detections at a time step t do not have fresh information and are unlikely to provide an update. Since the impact of missed detections and false alarms is minimal, we set $P_D = 1$ and $P_{FA} = 0$ for the remainder of the results.

Each policy is constrained, so the average network utilization rate is constant between policies. This does not mean, however, that all targets are updated at the same rate. In fact, as Fig. 5.9 shows, the distributed AoII policy updates targets which have higher entropy rates more often, while the round robin policy updates targets at the same rate independent of the motion model entropy rate. This figure shows several properties. First, the AoII policy is able to provide a higher update rate for all targets by allocating more resources to those nodes which observe more targets. Second, those targets with higher entropy rate receive more updates than those with lower entropy rates. These properties interact to somewhat suppress the effect of higher entropy rate: each node sees multiple targets, entropy rate is uniformly distributed among targets, and nodes seeing more targets get more resources. An update rate of "1" means that a target is updated by one node per update period.

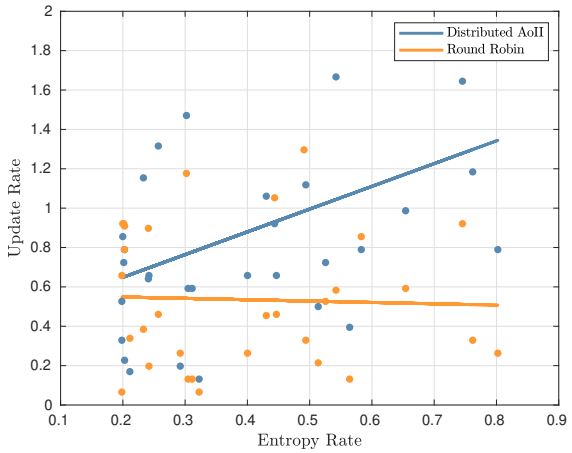


Figure 5.9: A scatter plot and best-fit line relating the entropy rate of target motion models to the rate at which the FC receives target updates. Targets are uniformly distributed entropy rates in $[0.2, 0.8]$.

Since each algorithm uses the same number of resource blocks on average, it might be assumed that each algorithm also exhibits the same tracking performance. This is not the case, since some algorithms will make more efficient use of the resource blocks, as well as choosing more carefully which nodes provide updates at which times. Figure 5.10 demonstrates that the UCB and Round Robin algorithms both perform poorly. This is due to the reward-greedy behavior exhibited by UCB and the uniform selection probability of Round Robin. UCB tends to preferentially select low-variance tracks, and avoid high-variance tracks, due to the reward function formulation. This leads to overall *higher* tracking error, since high-variance tracks will age, causing the FC estimate to become even worse.

The AoI-inspired timely algorithm performs better, since it takes into account the age of the tracks reported by each node, and is able to allocate more updates to those nodes which provide more fresh, low-variance information. The AoII algorithm performs even better since it removes the timely algorithm's need to explore nodes to discover new tracks; nodes are able to decide when to send their own updates given the update constraint.

As demonstrated by the above results, efficient use of the available resources results in lower tracking error. With this in mind, we next consider how tracking performance varies with changing constraints. Fig. 5.11 shows us that increasing the capacity C results in increased tracking performance for both the AoII algorithm and Round Robin. However, the performance gap between these two algorithms also varies. When the resource constraint is high, the timing of updates matters less since more updates can be transmitted. When the resource constraint is low, however, the update timing becomes more important. The difference in AoII and Round Robin performance under a low resource constraint is nearly an order of magnitude in tracking error. But, when the resource bound is high, they perform

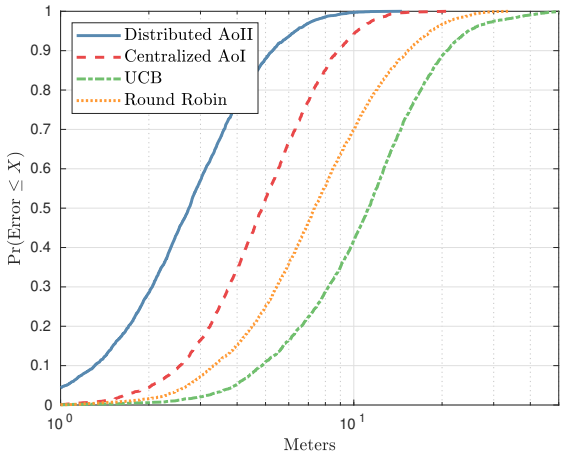


Figure 5.10: Error distributions for different selection algorithms. Since Round Robin and Random each select nodes with a constant frequency, they exhibit similar performance. Since the centralized AoI and distributed AoII metrics incorporate track age, they outperform the other techniques. Since the AoII approach further allows each node to decide when to provide updates, it achieves nearly double the probability of less than 100 meter accuracy while meeting the same spectrum usage.

nearly identically.

A critical value for this type of tracking problem is the *peak age of information* (**PAoI**). As opposed to the AoII, this is not a value we used in decision-making, but is a tool for analysis of policies. It denotes the average maximum age of a process upon updating - how long the FC must estimate the location of a target before an update is provided. We can denote the PAoI as

$$\Delta^P(t) = \lim_{\tau\to\infty} \frac{1}{U(\tau)} \sum_{n=1}^{U(\tau)} A_u - \mathbb{E}(A_u) \tag{5.30}$$

where there are $U(\tau)$ updates before $t = \tau$, and A_u is the age of the process at the n^{th} update.

We can expect that a policy which is more egalitarian towards targets will provide a moderate PAoI - each target would get updated evenly often, and so no targets will go extremely long without updating. An age-greedy policy, one which selects nodes such that the maximum-age targets are always updated, might provide a lower PAoI. However, a policy which takes into account the differing needs of each target may (somewhat paradoxically) provide a *higher* PAoI, if not all targets need similar update rates. In Figure 5.12 we see that the AoII policy outperforms the others in terms of peak age. This is because this technique relies on each node to determine when an update is needed rather than rely on the FC. The timely policy also performs well, since it optimizes for track age as well as variance.

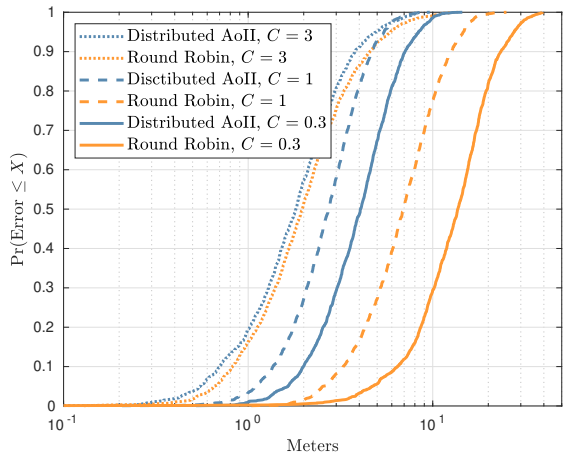


Figure 5.11: The frequency with which the FC receives updates directly impacts the track error. In addition, the performance gap between AoII and random selection increases with decreasing capacity. When less capacity is available, the selection algorithm quality becomes more important. The capacity constraint C varies from 0.3 to 3.

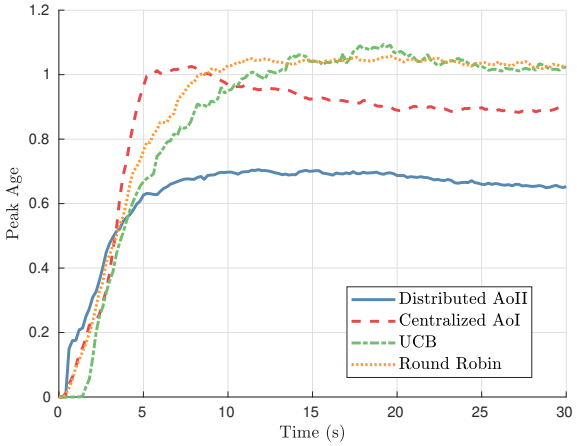


Figure 5.12: Peak age averaged over all active tracks. The distributed AoII policy exhibits the best performance, followed by the centralized AoI policy. Since the centralized approach first explores nodes which observe many targets, it does not revisit targets before all nodes are updated.

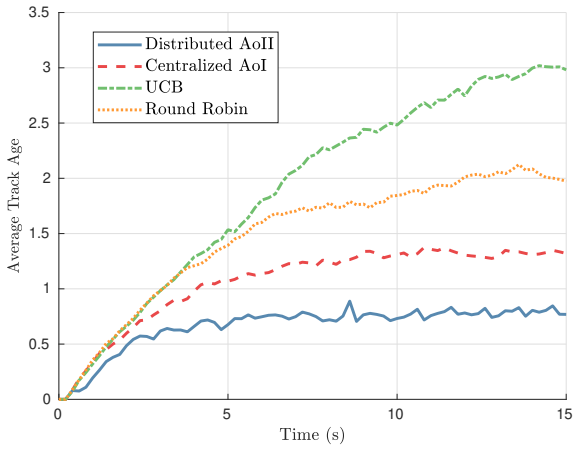


Figure 5.13: Mean age of each active target track at the FC. Since AoII updates nodes according to track age, it most effectively minimizes average age at the FC.

Another way to understand the impact of the capacity constraint is by examining the mean age of active tracks at the FC. In Fig. 5.13, we see that higher-capacity networks are able to maintain a lower mean age. This is simply because of the quantity of targets the FC can update.

We can inspect the number of targets which are observable to the network (m s.t. $m \in \cup_{n\in\mathcal{N}} S_n$) but which are not tracked by the FC, shown in Fig. 5.14. This is caused when nodes do not provide updates in a timely manner when new targets appear. As the plot demonstrates, the number of missed targets is low for all considered algorithms. This is because even those algorithms which underperform in tracking error will still tend to receive updates from all nodes over enough time. Notably, the UCB algorithm has the lowest number of untracked targets. This is due to UCB rewarding exploration early in the game - it is more likely to select nodes that have not been selected before, so it will initiate a track for all observed targets early in the simulation. One key take-away, however, is that UCB's tracking error performance is worse than the AoI based algorithms. So, it is not necessarily beneficial to maintain target tracks which receive infrequent updates.

Lastly from this figure we can see the number of targets which exist in the environment but are out of range of the network (i.e., no radar is able to track them). This is a consequence of the random nature of the network. Eq. (5.4) shows the probability distribution of the number of targets which are out of range of the network. Evaluated on the densities of the network shown, the mean number of unobservable targets is close to 4, which matches the difference between the total targets and the covered targets.

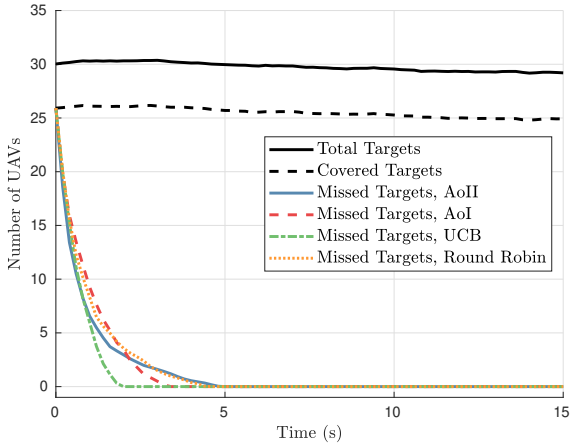


Figure 5.14: Number of missed targets for each algorithm. Due to the stochastic nature of each network, the number of unobservable targets will be distributed according to Eq. (5.4) which with $\lambda_m = 0.3$, $\lambda_n = 0.2$ has a mean close to 4.

5.6 Conclusions

In this work, we demonstrated centralized and distributed node selection algorithms for a cognitive radar network which efficiently use the limited communication resources available to the network. Each of several radar nodes must observe many targets and efficiently provide updates to the FC. The UAV targets exhibit Markov chain motion model behavior, which is exploited to provide more efficient updates.

We presented a centralized, track-sensitive AoI-inspired node selection algorithm, which utilizes a polling procedure to determine which nodes have observed a change in motion model. This method was shown to be superior to alternative techniques such as the UCB multi-armed bandit and random node selection.

Then, we developed a distributed technique based on the Age of Incorrect Information through which each node can independently determine when to provide updates to the FC. This required modification of the AoII criteria developed in [149] to accommodate the multi-target, multi-node scenario. We showed a resource constraint that is identical in both distributed and centralized formulations. This technique was also shown to be superior to alternative approaches.

In real target tracking systems, this type of optimization could result in simpler track management and lower communication requirements. Due to the lower track age demonstrated by our proposed techniques, tracking performance in realistic systems should be expected to increase, especially for large numbers of targets.

A possible extension to this work would include "mode control," by which the FC or the

nodes could determine how to allocate radar observation time and whether to perform passive sensing, such as direction of arrival estimation or signal classification. Such an extension would possibly enable a tracking network to be more power-efficient as well as to reduce the radiated power, allowing the nodes in the network to attain a lower probability of interception (i.e., lower chance of being detected by other systems).

Chapter 6

Mode Selection

Related Publications

The material in this chapter has been reproduced from the following publications:

- **W. W. Howard**, A. F. Martone, and R. M. Buehrer, "Mode Control in Cognitive Radar Networks," submitted to *IEEE Trans. on Radar Systems (Under Review)*, 2023. Available online.e
- [54] **W. W. Howard**, S. R. Shebert, B. H. Kirk, and R. Michael Buehrer, "Mode Selection and Target Classification in Cognitive Radar Networks," *Submitted*, 2023. Available online.

6.1 Introduction

There is a desire in the literature and within DoD for devices with low size weight and power (**SWaP**). One current method by which this is achieved is through the use of flexible high-speed computing platforms such as FPGAs. Modern devices are capable of very fast digital signal processing with very high bandwidths. We examine an application of flexible computing in networks of low-power, multi-mode cognitive radar devices (**CRNs**) [51] [52] [155]. These radar devices (**nodes**) [118] have the ability to observe targets via active radar observation or passive signal detection and classification (Electronic Support Measures, **ESM**) and quickly alternating between the two modes, selecting one mode in each of many time steps. In addition to the radar nodes, a fusion center (**FC**) is present to combine target information as well as provide feedback to the radar nodes. Over the course of many target tracks, the CRN groups targets by behavior similarity (**classes**), assuming that the observed targets are generated by a finite number of target distributions. We show that if targets of the same class are identifiable by their signal or physical behavior, then more accurate target class estimation will result in 1) a reduction of effective radiated power from each node and 2) an improvement in target tracking performance.

This "mode selection" is motivated by capabilities a CRN may already possess. Prior work [109] has considered the use of spectrum sensing to determine the utility of a specific radar action in the presence of noise and interference. The presence of other devices in a given

channel can indicate the quality of radar measurement using that channel. Other work [55] has considered the use of spectrum sensing to detect and avoid mutual interference, instances of nodes in a CRN interfering with each other.

The targets are not only grouped by motion model similarity, but by signal emission similarity as well. We leverage the fact that most modern targets, civilian or military, tend to have characteristic radio emissions (e.g., FM voice communication and Automatic Dependent Surveillance Broadcast (**ADS-B**) in general aviation; control, telemetry, and data downlink in consumer unmanned aerial vehicles (**UAVs**); two-way voice communication for hot-air balloons). By using passive ESM techniques rather than active radar, the CRN nodes are able to take advantage of additional target information while reducing their power usage and more importantly their radiated power. This allows the CRN to perform additional target modeling, associating unique signal characteristics with unique physical characteristics. To continue with the previous examples, general aviation aircraft tend to exhibit different kinematics than consumer UAVs, which are in turn physically different from lighter-than-air balloons. We show that leveraging these associations can result in the same or better tracking error while requiring less power consumption at the CRN nodes.

The different types of targets are formed into "classes" over many independent tracks. Then, the CRN nodes can choose a mode of operation that depends on the targets it is currently tracking - choosing radar when it is necessary, and passive ESM when it provides additional information. We propose two main categories of decision process: a centralized approach, which considers the entire network and all current tracks for decision-making at the cost of higher communication overhead; and a distributed approach, which allows CRN nodes to select a mode based on the targets they are currently tracking.

6.1.1 Organization

In Section 6.2, we review recent work in the related fields of multi-target tracking, cognitive radar, and radar networks. Section 6.3 discusses the model for the CRN considered in this work, as well as specifics on target modeling and tracking. Section 6.5 develops our centralized and distributed mode selection algorithms. In Section 6.6 we present analysis and numerical simulations of our proposed techniques, comparing them against alternatives. In Section 6.7 we discuss our results, draw conclusions, and recommend future work in distributed and centralized control of cognitive radar networks.

6.1.2 Contributions

We build on previous contributions in the areas of radar network control, cognitive radar, and multi-target tracking. Portions of this work were presented previously [54]. We contribute the following to the state of the art:

- A model for mode selection in multi-function cognitive radar networks.
- An analysis of multiple target class formation based on characteristic motion and signal emission models.
- Mathematical analysis of a clustering-based class formation technique.
- A centralized approach which mitigates the effects of network latency.
- Numerical simulations to support our conclusions.
- We show that our proposed techniques outperform radar-only observation as well as outperforming a random selection algorithm which achieves the same radar observation rate, but does not consider target class formation.

6.1.3 Notation

We use the following notation. Matrices and vectors are denoted as bold upper $\mathbf{X}$ or lower $\mathbf{x}$ case letters respectively. Element-wise multiplication of two matrices or vectors is shown as $\mathbf{X} \odot \mathbf{Y}$. Functions are shown as plain letters F or f. Sets $\mathcal{A}$ are shown as script letters. Denote the Lebesgue measure of a set $\mathcal{A}$ as $|\mathcal{A}|$. When we wish to show the number of elements in a (finite) set $\mathcal{A}$ rather than its measure, we use the cardinality $\#(\mathcal{A})$. The transpose operation is $\mathbf{X}^T$. The backslash $\mathcal{A} \backslash \mathcal{B}$ represents the set difference. Boxes (intervals) in $\mathbb{R}^d$ are written as $[a, b]^d$ and when the elements of a set are denoted, they are given as $\mathcal{A} = \{a, b, c, \dots\}$. Random variables are written as upper-case letters X, and their distributions will be specified. The set of all real numbers is $\mathbb{R}$ and the set of integers is $\mathbb{Z}$. The speed of electromagnetic radiation in a vacuum is given as c. The Euclidean norm of a vector $\mathbf{x}$ is written as $||\mathbf{x}||$. Estimates of a true parameter p are given as $\hat{p}$. The operator $x \xleftarrow{\text{R}} \mathcal{A}$ denotes assigning x as a random sample of the set $\mathcal{A}$.

6.2 Background

CRNs and general cognitive radar are two areas of much recent study. Recent contributions in the area of cognitive radar networks include [53] which investigates Age of Information (**AoI**) approaches to the problem of timely updating in CRNs, [51] which investigates cooperative spectrum allocation in CRNs, and [52] which investigates the relationship and trade-offs between centralized and distributed spectrum allocation techniques. Recent contributions in the area of cognitive radar include [20] which investigates waveform selection, [118] which discusses a universal learning technique for multi-track optimization, and [25] which discussed "meta-cognition", the process of choosing between different cognitive strategies.

There has also been recent work in the area of network-based passive target localization. The authors of [156] present a centralized passive estimation network, where the nodes act as amplify-and-forward units, and the total amplification of the network is power-limited. All of the decision-making in the network is located in the fusion center, and is confined to allocating the limited amplification power to the nodes. The targets are modeled as transmitting one of several complex-valued signals. The FC fuses the node measurements with a goal of estimating the true target signal. This work is useful because it provides a framework for signal estimation, which could support classification. The goal of [156] is simply to reconstruct (with high accuracy) the signals emitted by the target. Our work focuses (among other goals) on a slightly different aspect of signal analysis - we're concerned with identifying the *type* of signal emitted by a target, rather than reconstructing it.

Multi-target tracking is also a rich field, containing several important contributions. In particular, the study of *probability hypothesis density* (**PHD**) filters [72] [76] aims to solve the problem of *target association*. The PHD is the first moment of the target state space, and is akin to the expected value of a random variable. It allows for the estimation of the number of targets in a scene as well as the propagation of the state of those targets through time. As target detections are received (from point or extended object models, among others), the PHD filter is updated. In [75] a multiple model PHD filter is developed, where several PHD filters are used in parallel to estimate the motion model of targets.

6.3 Target Modeling

6.3.1 Discrete Time Markov Chains

We consider targets in a region B which are spatially distributed according to a stochastic Poisson point process (**PPP**) A Poisson number of targets with mean λ_M is drawn, and each target is assigned an initial state uniformly at random in the region B. Generally, each target in the observable region B can be partially or wholly described by several parameters. Let the parameters describing a target m be collected into an ordered set $\mathcal{X}_m$. In other words, a given element $E_m \in \mathcal{X}_m$ describes some quality of the target m. We limit our consideration of target m to those parameters which can be described as time-homogeneous ergodic Markov chains in a finite state space. So, a parameter E_m is a time-varying quantity that can take on finitely many values. Since E_m is a discrete-time Markov chain (**DTMC**), it has the Markov property given in Eq. (6.1). This means that the probability of transitioning to any given state is only dependent on the current state and not the history; i.e. there is no memory length.

$$\begin{aligned} \Pr[E_m^{(t+1)}] = e_m \mid E_m^{(1)} = e_m^{(1)}, E_m^{(2)} = e_m^{(2)}, \ldots, E_m^{(t)} = e_m^{(t)}] \\ = \Pr[E_m^{(t+1)} = e_m | E_m^{(t)} = e_m^{(t)}] \end{aligned} \tag{6.1}$$

We say that there are N_E states which form a finite set $\mathcal{E}$ called the state space of the DTMC. The transition matrix P_m^E consists of entries

$$p_{ij} = \Pr[E_m^{(t+1)} = j | E_m^{(t)} = i] \tag{6.2}$$

which describe the probability of transitioning states at any time step t. Note that all rows sum to 1 and all elements are non-negative. Since the process is time-homogeneous, the transition matrix is not time dependent. Lastly, the stationary distribution of parameter E_m can be defined as Eq. (6.3). The stationary distribution π_m^E[1] can be seen to be a normalized left eigenvector of the transition matrix [157].

$$\pi_m^E = \pi_m^E P_m^E \tag{6.3}$$

Since the process is ergodic, the stationary distribution is also the limiting distribution of any starting distribution [158]. In other words, there is only one eigenvector of P_m^E with an eigenvalue of 1. This means that as the number of samples n trends towards infinity, the empirical stationary distribution is not dependent on the initial state distribution.

6.3.2 Class Definitions

Definition 6.1 (Equal in State Distribution)**.** Two random variables X and Y are said to be *equal in state distribution* if the following properties hold:

1. Both X and Y can be described as DTMCs.
2. If N_X is the size of the state space for X and N_Y is similarly defined for Y, then $N_X = N_Y$.
3. If π^X is the stationary distribution for X and similarly for Y, then $\pi^X = \pi^Y$.

Definition 6.2 (Target Class)**.** Let the parameters describing target m_0 be collected into $\mathcal{X}_{m_0}$. If there exists a target m_1 with the following property, then it is said to be of the same *class* as m_0. Denote the class as C.

- Each element of $\mathcal{X}_{m_0}$ maps to a unique element of $\mathcal{X}_{m_1}$. Corresponding elements are defined over the same state space and are equal in state distribution.

For a given target class C with a given parameter E, we say that the set $\mathcal{C}$ contains the state space ($\mathcal{E}$) of E as well as the stationary distribution π^E over $\mathcal{E}$ which defines the class.

Definition 6.3 (Target Family)**.** A *target family* is a group of target classes $\{C_1, C_2, \dots\}$ with the following properties. Call the family F.

[1] In the context of DTMCs, we slightly abuse notation to allow π_m^X to denote the parameter described by π, rather than exponentiation.

1. If state space $\mathcal{E}$ is contained in $\mathcal{C}_1$, then it is also contained in $\mathcal{C}_i$ for all $\mathcal{C}_i \in \{C_1, C_2, \dots\}$.
2. If a stationary distribution π^E is contained in $\mathcal{C}_i$, then it is *not* in $\mathcal{C}_j$ for any $\mathcal{C}_j \in \{C_1, C_2, \dots\} \backslash C_i$.

In other words, all classes within a family are defined by the same set of parameters, although they do not have the same parameter values. We discuss in Section 6.4 how a CRN node forms an estimate of target parameters.

Proposition 1 (Unique Class)**.** Let node n draw an estimate $\hat{\pi}_m^E$ of the stationary distribution of parameter E_m for target m. The node estimates the corresponding class, $\hat{C}$. Further, let target m be drawn from class C in a family F.
If $\hat{\pi}_m^E = \pi_m^E$, then $\hat{C} = C$.

Proof of Prop. 1. Trivial by Def. 6.3 □

So, targets m within a family F have the useful property that if one parameter's stationary distribution π_m^E can be estimated, then the class can also be estimated.

6.3.3 Motion Modeling

As target m moves through space, its state at a time t can be described by Eq. (6.4) which is comprised of position and velocity. We summarize the higher-order derivatives by defining *motion models* [159], which describe how the target's state propagates in time. Let $V_m(t)$ be the motion model for target m, and say that it takes on one of N_V states for each time step. In other words, $V_m(t)$ is a time-homogeneous ergodic discrete-time Markov chain in a finite state space. Let π_m^V be the stationary distribution of the states in $\mathcal{V}$, the space of possible motion models. Lastly let P_m^V be the motion model state transition probability matrix for target m.

$$\mathbf{X}_m(t) = [x_m(t), \dot{x}_m(t), y_m(t), \dot{y}_m(t), z_m(t), \dot{z}_m(t)] \tag{6.4}$$

In Section 6.4.2, we discuss how the motion model for target m is estimated.

6.3.4 Signal Modeling

Each target m also operates radio equipment (i.e., radar, communications, telemetry, etc.) resulting in electromagnetic emissions that can be detected via passive ESM. We say that $S_m(t)$ is the state of the emissions from target m at time t, and that $S_m(t)$ takes on one of the N_S values in the space of possible emissions $\mathcal{S}$. So, $S_m(t)$ is a time-homogeneous ergodic discrete-time Markov chain in a finite state space. Let π_m^S be the stationary distribution for target m, and say P_m^S is the emission state transition probability matrix. We consider

the absence of a signal to be a signal state. So, we say that one of the states in the space is consists of no signals. In Section 6.4.3 we define the *probability of interception* of target signals.

6.3.5 Summary

We model a target m using two Markov processes. The set $\mathcal{X}_m$ contains the motion model $V_m(t)$ which takes on one of N_V states with transition probabilities P_m^V and stationary distribution π_m^V. Further, it contains $S_m(t)$, the emission state which takes on one of N_S states with transition probabilities P_m^S and stationary distribution π_m^S. Both processes are ergodic time-homogeneous discrete-time Markov chains in finite state spaces, which means that the transition probabilities are not time dependent and there is a nonzero probability of transitioning from any state i to any state j.

There exist *classes* of targets which are composed of targets whose motion and emission states are equally distributed[2]. Finally, the target classes form a *family* which provides uniqueness to each class.

6.4 Target Estimation

We consider a network which is capable of performing either active radar observation or passive signal parameter estimation (ESM). Due to target class estimation, ESM estimation provides the ability to classify a target and reduce subsequent tracking error. In addition, selecting the passive ESM action lowers the observabilty of the radar network to adversaries. This is quantified by the *maximum intercept range* $R_{I,\max}$, Eq. (6.5), where P_{radar} is the transmit power, G_t is the transmit gain in the direction of the intercept receiver, G_I is the intercept receiver gain, L_1 is one-way atmospheric loss, λ is the wavelength of the center frequency, L is system loss, δ_I is the intercept receiver sensitivity, F_1 is the intercept receiver noise figure, B_1 is the intercept receiver bandwidth, SNR_{Ii} is the intercept receiver SNR, k is Boltzmann's constant, and T_0 is the intercept receiver noise temperature.

$$R_{I,\max} = \sqrt{\frac{P_{\text{radar}} G_t G_I L_1 \lambda^2}{(4\pi)^2 L \delta_I}} \tag{6.5}$$

$$\delta_I = kT_0 F_1 B_1 (SNR_{Ii}) \tag{6.6}$$

This is the maximum range at which an interceptor can detect the radar. When the number of radar transmissions is reduced, this reduces the average distance at which a transmission

[2]Note that this does *not* imply that the motion or emission state of two targets in the same class are identical, just that they are identically distributed over the respective state spaces.

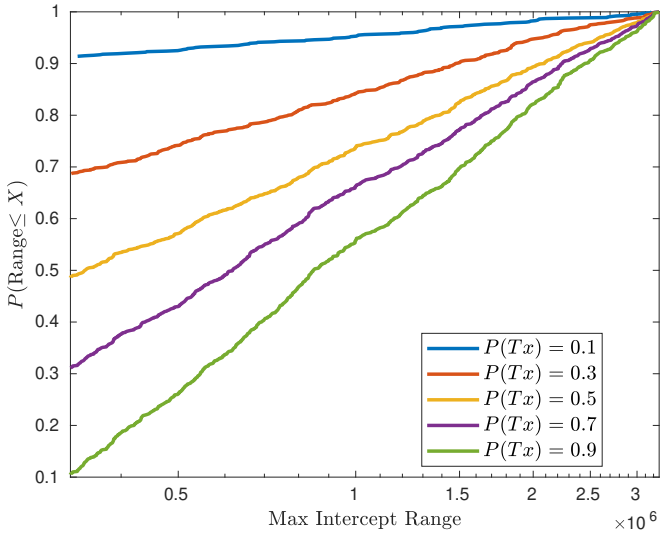


Figure 6.1: Maximum intercept range for a single radar node with a variable transmit probability.

may be intercepted. The max intercept range for a single radar node with a variable transmit probability (i.e., portion of time spent selecting active radar) is shown in Fig. 6.1.

6.4.1 Network Structure

Following a similar structure to those presented in [53] [58], we use tools from stochastic geometry [69] and consider a compact region of $\mathbb{R}^3$, assuming that the network consists of a random set $\mathcal{N}$ of multi-function radar/ESM nodes generated by a Poisson Point Process [160] with density λ_N and positions X_n. Each node n "covers" a region C_n with measure $|C_n|$, which accounts for practical range limit for radar devices. Let N with mean $\overline{N}$ be the random variable describing the number of such nodes. The distribution of N is given as Eq. (6.7), with the mean shown in Eq. (6.8).

$$\Pr[N = n] = \frac{\lambda_N |B|^n e^{-\lambda_N |B|}}{n!} \tag{6.7}$$

$$\overline{N} = \lambda_N |B| \tag{6.8}$$

Similarly, there is a random set $\mathcal{M}$ containing M targets at positions X_m, generated by a PPP with density λ_M and mean $\overline{M} = \lambda_m |B|$. When we consider numerical simulations, a single simulation consists of a single realization of this model.

Targets continue to exist in the next time step according to a fixed probability, and a Poisson number of new targets are born in each time step [161]. These two events cancel out such that the target density is maintained.

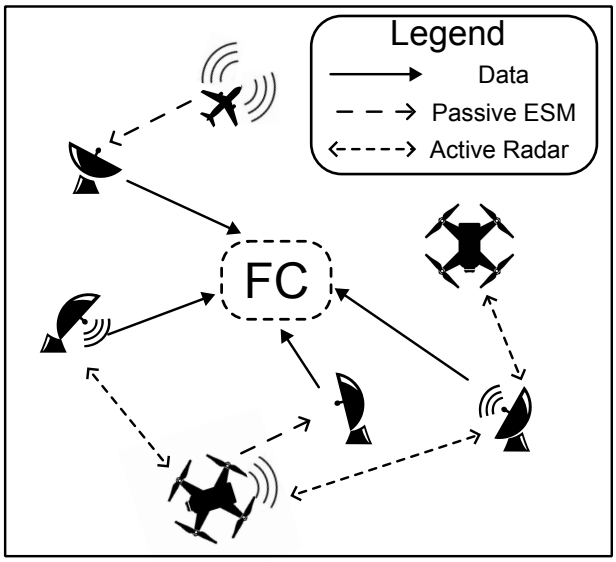


Figure 6.2: A model of the type of network we consider, where each node can choose between active and passive observation of several types of targets. The nodes send observations to the fusion center.

The coverage of the network is the union of the covered region for each node, Eq. (6.9).

$$\mathbf{C} = \bigcup_{n \in \mathcal{N}} C_n \tag{6.9}$$

As a result, the probability that a target is "covered" is given by Eq. (6.10). This is also the probability that any random point in B is covered.

$$\Pr[X_m \in \mathbf{C}] = 1 - e^{-\lambda_N \mathbb{E}[|C_n|]} \tag{6.10}$$

An example of this network geometry is shown in Fig. 6.2.

6.4.2 Active Radar

Radar estimation allows for remote target position and velocity observation. We consider a monostatic radar configuration, where each node is capable of transmitting a train of coherent pulses, receiving scatter from clutter and target responses, then coherently combining the pulses and performing range-Doppler estimation. Constant false alarm rate (**CFAR**) detection is performed, by which the radar node estimates the presence or absence of targets in the range-Doppler map. We assume that when the radar mode is used, the radar uses a multi-function detection and tracking mode. In other words, it scans for new targets and attempts to obtain further measurements of previous tracks.

Measurement

Following the extended object model [162], we assume that each target visible to the radar may occupy several resolution cells and therefore generate multiple detections per time step. Let node n cover $\mathcal{M}_n^{(t)}$ targets at time t. Since the targets can move around, spawn, or retire [74], this quantity is time-varying. Note that

$$\mathbb{E}[\#\mathcal{M}_n^{(t)}] = \lambda_M |S_n| \tag{6.11}$$

is the expected number of targets covered by any given node. So, each target m in the observable region for node n generates a number of detections $N_{Z,mn}$:

$$\mathbf{Z}_{mn} = \left\{\mathbf{z}^{(j)}\right\}_{j=1}^{N_{Z,mn}} \tag{6.12}$$

Each target detection is missed with probability P_D. Since more than one target may be in the region covered by node n, the total number of target detections generated at time t by node n is given as Eq. (6.13).

$$\mathbf{Z}_n^t = \bigcup_{m \in \mathcal{M}_n^{(t)}} \mathbf{Z}_{mn} \tag{6.13}$$

Then, a PHD filter [72] [75] is used to associate [76] these target detections with previous tracks and generate new tracks if necessary. Since $\mathbf{Z}_n$ may contain false alarms generated at a rate of λ_{FA}, the PHD filter maintains a list of tentative and confirmed tracks. False alarms are uniformly distributed in the region.

At a time step t, the output from the PHD filter is used as the current state of the estimated target track. Let $\hat{\mathcal{M}}_n^{(t)}$ be the set of active tracks at node n. Then, for a track $\hat{m} \in \hat{\mathcal{M}}_n^{(t)}$, $\mathbf{X}_{\hat{m}}$ represents the current state. Further tracking details are provided in Section 6.4.4.

6.4.3 Passive Electronic Support Measures

Passive ESM estimation allows the node to make observations of targets without the need for high-power radio emissions. Unlike active radar sensing, ESM performance is dependent on the characteristics of the transmitting equipment. The primary values which impact the performance of the ESM receiver are the *maximum detectable range* and the *probability of intercept*.

Maximum Detectable Range

The maximum detectable range of targets[3] is dependent on the received SNR[4], Eq. (6.14).

$$\text{SNR}_{mn}^{ESM} = \frac{P_t G_t G_r \lambda^2}{(4\pi R)^2 P_n L} \tag{6.14}$$

Note that the SNR from a target m received at the n^{th} node is dependent on several aspects of the target equipment: the transmit power of the target equipment (P_t), the transmit antenna gain (G_t), and the wavelength (λ). In addition, there are a few other parameters which depend on the n^{th} receiving ESM: the receive antenna gain (G_r), the range to the target (R_{mn}), the receiver noise power (P_n) and any other losses (L). The receiver noise power is given as Eq. (6.15), where T_0 is 290 K, F is the receiver noise figure (10 dB), B is the receiver bandwidth (1 MHz), and k is Boltzmann's constant.

$$P_n = kT_0FB \tag{6.15}$$

Figure 6.3 shows the received SNR for targets at various ranges and with various transmit powers. Of course, the SNR required for high probability detection with a low probability of false alarm will vary based on the detection algorithm, the signal capture duration, and the signal of interest. For cyclostationary detectors with a signal duration greater than a few milliseconds, it is possible to detect signals at a rate approaching 100% at or below 0 dB SNR with a false alarm rate less than 1/100 [163] [164]. Thus, we make Assumption 3 below.

Definition 6.4 (Maximum Detectable ESM Range)**.** The maximum detectable range R_{mn} for electronic support measures between node n and target m is the maximum range for which Eq. (6.16) holds, where the SNR is given by Eq. (6.14).

$$\text{SNR}_{mn} \geq 0 \tag{6.16}$$

Denote R_{mn} s.t. $\text{SNR}_{mn} \geq 0$ as R_{mn}^{ESM}

Note that the *probability of correct detection* $P_{d,mn}^{ESM}$ is the probability that a signal is correctly identified, and the *probability of false alarm* $P_{fa,mn}^{ESM}$ is the probability that a detected signal is misidentified or mistakenly detected (i.e. not present).

Assumption 3 (In-Range Targets are Detectable)**.** Targets for which $R_{mn} < R_{mn}^{ESM}$ have $P_{d,mn}^{ESM} = 1$ and $P_{fa,mn}^{ESM} = 0$.

While detectors with high P_d and low P_{fa} exist in the literature, the signal must still be present. We account for this by also defining the probability of intercept.

[3]Note that this is the maximum range at which a node in the CRN can passively detect a target, which is different than Eq. (6.5) which is the maximum range at which an intercept receiver could detect a node in the CRN.

[4]Specifically, this is the instantaneous or non-integrated SNR.

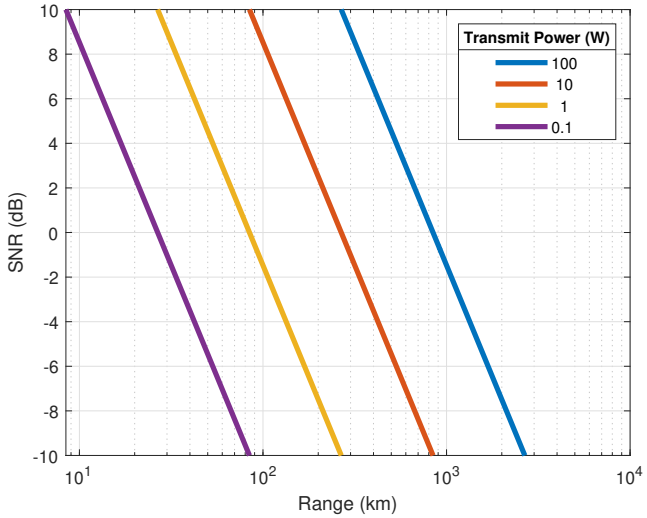


Figure 6.3: ESM receiver SNR for a center frequency of 1 GHz, omnidirectional receive antennas (gain of 1 dB), and 1 dB of losses. Given an SNR requirement of 0 dB for detection, targets can reasonably be detected at ranges of up to ~ 100 km, depending on the transmitter power.

Probability of Intercept

The probability of intercept $P_{I,mn}^{ESM}$ is a term used to characterize the uncertainties in look angle, terrain, transmission rate, etc, which decrease the rate at which an in-range target is detected[5]. In general, the probability of intercept depends on the ESM receiver's measurements in time, frequency, and space overlapping with the target transmitter's emissions in time, frequency, and direction [165, Ch. 14]. To reduce the complexity of simulating this effect, we consider omnidirectional transmit and receive antennas (i.e., $G_t = G_r = 1$) and a sufficiently wide receiver scanning bandwidth such that all signals of interest are covered. So, the probability of intercept is dominated by the fraction of time containing target transmissions. In other words, the receiver knows where to look in space and frequency, but does not know *a priori* when a target will transmit. If an ESM receiver m chooses a time t for signal detection, and no target n with $R_{nm} < R_{mn}^{ESM}$ transmits at time t, then no signals will be detected. In this way we account for the "null" transmission state discussed in Section 6.3.4. Without loss of generality, say that $\pi_m^S[1]$ denotes the stationary distribution weight that target m chooses not to transmit. Since the stationary distribution is also the limiting distribution, this reflects the fraction of time in which the target is not transmitting. Thus,

$$P_{I,mn}^{ESM} = \pi_m^S[1] \tag{6.17}$$

[5]Note that while $P_{d,mn}^{ESM}$ may be high, a low probability of intercept will still reduce the probability that a target signal can be identified.

which says that if target m spends more time transmitting, then any node has a higher probability of intercept for that target.

Measurement

When (at time t) an ESM receiver n collects a sample $\hat{s}_n(t)$ from the environment, it will detect signals $s_n(t)$, Eq. (6.18). With the set of targets $\mathcal{M}$, denote $\mathcal{M}_s(t)$ as the set of transmitting targets at time t. Then, say that $\mathcal{R}_n(t)$ targets m such that $R_{mn} < R_{mn}^{ESM}$.

$$\hat{s}_n(t) = \{S_m(t) \text{ s.t. } m \in \mathcal{M}_s(t), R_{mn} < R_{mn}^{ESM}\} \tag{6.18}$$

$$= \overline{S}(t) \odot [\mathbb{1}_{m \in \mathcal{M}_s(t)} \odot \mathbb{1}_{m \in \mathcal{R}_n}] \tag{6.19}$$

Then, the probability that the emission state of target m is observed by ESM node n can be written as Eq. (6.21).

$$\begin{aligned} &\Pr[S_m(t) \in \hat{s}_n(t) \mid m \in \mathcal{R}_n(t)] \\ &\quad = \Pr[S_m(t) \in \hat{s}_n(t) \mid m \in \mathcal{S}_m(t)] \\ &\qquad + \Pr[S_m(t) \in \hat{s}_n(t) \mid m \notin \mathcal{S}_m(t)] \qquad (6.20) \\ &\quad = P_{d,mn}^{ESM} * P_{I,mn}^{ESM} + 0 \\ &\quad = \pi_m^s[1] \qquad (6.21) \end{aligned}$$

Eq. (6.20) states that the probability of observation (given target m is in ESM range) is the sum of the probability of observation given the target actually transmits plus the probability of observation given the target does not transmit. By Assumption 3, the probability of false alarm is zero for in-range ESM targets, so $\Pr[S_m(t) \in \hat{s}_n(t) \mid m \notin \mathcal{S}_m(t)] = 0$. Then, since by the same assumption $P_{d,mn}^{ESM} = 1$ for in-range ESM targets, the probability of observation is equal to the probability of interception.

6.4.4 Tracking Formulation and Fusion

In each time step t, every node selects one of the two available actions (active radar or passive signal classification) and receives a vector of detections. These detections are then filtered and associated to target tracks. We follow the development of [75] and use the multiple model PHD filter to estimate state of each target, including motion model. From Eq. (6.13) we have

$$\mathbf{Z}_n^t = \bigcup_{m \in \mathcal{M}_n^{(t)}} \mathbf{Z}_{mn}$$

Let $\mathbf{Z}_n^{1:t}$ denote the ordered set of observations until time t.

From [75], we have the multiple model PHD tracking filter for maneuvering targets. We reproduce the core steps of the technique here. The filter begins with an initial density Eq.

(6.22).

$$\tilde{D}_{t|t-1}\left(\mathbf{X}(t-1), V(t)=i|\mathbf{Z}_m^{1:t-1}\right) \tag{6.22}$$

The motion model mixing is given by Eq. (6.23), where $V(t)$ is one of several motion model states, N_V is the number of motion model states, and P_ij are the transition matrix entries, Eq. (6.2), for transitioning from state i to state j. Note that there is a separate PHD for each possible motion model.

$$\tilde{D}_{t|t-1}\left(\mathbf{X}(t), V(t)=i|\mathbf{Z}_n^{1:t-1}\right) = \sum_{j=1}^{N_V} D_{t-1|t-1}\left(X(t-1), V(t-1)=j|\mathbf{Z}_n^{1:t-1}\right) P_{ij}, \quad i=1:N_V \tag{6.23}$$

The PHD prediction step is given as Eq. (6.24), where $\gamma_t(\cdot)$ is the target birth PHD, $e_{t|t-1}(\cdot)$ represents the probability that each target survives to the next round, $f_{t|t-1}$ and is the motion state conditioned target likelihood.

$$D_{t|t-1}\left(X(t), V(t)=i|\mathbf{Z}_n^{1:t-1}\right) = \gamma_t(X(t), V(t)=i) + \int \left[e_{t|t-1}\left(X(t)\right) f_{t|t-1}\left(X(t)|X(t-1), V(t)=i\right)\right] \times \tilde{D}_{t|t-1}\left(X(t), V(t)=i|\mathbf{Z}_n^{1:t-1}\right) dX(t-1) \tag{6.24}$$

Finally, the PHD update step is given as Eq. (6.25), where P_D is the probability of detection, λ_{FA} is the false alarm rate, C_{FA} is the false alarm spatial distribution, and $\Psi_t(\cdot)$ is the PHD likelihood function, Eq. (6.26).

$$D_{t|t}\left(X(t), V(t)=i|\mathbf{Z}_n^{1:t}\right) \cong \left[\sum_{\mathbf{z}\in\mathbf{Z}_n^t} \frac{P_D(X(t)) f_{t|t}\left(\mathbf{z}|X(t), V(t)=i\right)}{\lambda_{FA} C_{FA} + \Psi_t\left(\mathbf{z}|\mathbf{Z}_n^{1:t-1}\right)} + \left(1 - P_D(X(t))\right)\right] \times D_{t|t-1}\left(X(t), V(t)=i|\mathbf{Z}_n^{1:t-1}\right) \tag{6.25}$$

$$\Psi_t\left(\mathbf{z}|\mathbf{Z}_n^{1:t-1}\right) = \int \left[P_D(X(t)) f_{t|t}\left(\mathbf{z}|X(t), V(t)=i\right) \times D_{t|t-1}\left(X(t), V(t)=i|\mathbf{Z}_n^{1:t-1}\right)\right] dX(t) \tag{6.26}$$

The motion model at time t for target m is estimated by integrating the PHD filters in the vicinity of target m and taking the maximum. When class motion model probabilities are substituted for estimated motion model probabilities, this is done via P_{ij}.

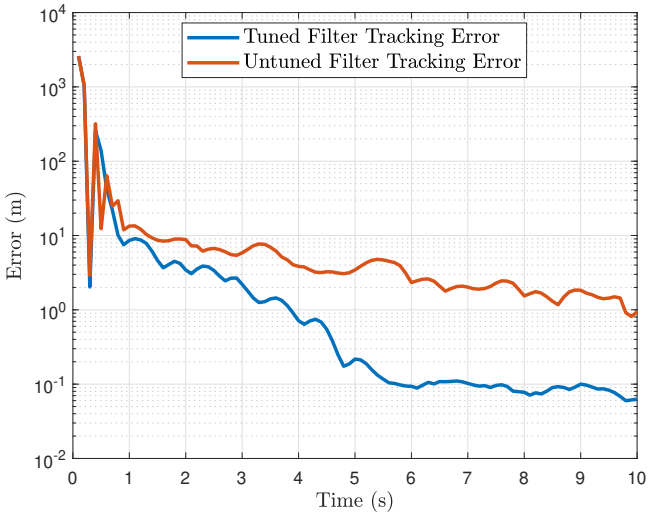


Figure 6.4: Kalman filters which are tuned to the process noise and motion model probabilities for the class will provide lower tracking error than equivalent filters which are "untuned".

Let $\tau^V_{m,n}$ denote the times when node n detects target m using active radar, and similarly let $\tau^S_{m,n}$ denote the times when node n detects target m using passive signal classification.

6.4.5 Class Formation

Since each node can estimate the motion model $\hat{V}_m(t)$ and signal state $\hat{S}_m(t)$ for each target $m \in \mathcal{M}_n^{(t)}$ (the set of targets observable by node n at time t), given multiple observations the node can also form estimates of the Markov processes which generate the motion model and signal states. Recall that $\tau^V_{m,n}$ denotes the set of times when node n detected target m using active radar, and $\tau^S_{m,n}$ are similarly the times when node n detected target m using passive signal classification. Given t_1 active and t_2 passive observations of target m, the motion model stationary distribution π^V_m and the signal model stationary distribution π^S_m can be estimated as Eq. (6.27) and Eq. (6.29). Note that this holds due to the ergodicity of the Markov chains: the stationary distribution is the limiting distribution for any initial condition.

$$\hat{\pi}^V_{m,n} = \left[\frac{\sum_{t \in \tau^V_{m,n}} I^V_{m,n}(t,v)}{\#\tau^V_{m,n}} \text{ for } v \in \mathcal{V} \right] \tag{6.27}$$

$$I^V_{m,n}(t,v) = \begin{cases} 1, & \hat{V}_{m,n}(t) = v \\ 0, & \text{else} \end{cases} \tag{6.28}$$

$$\hat{\pi}^S_{m,n} = \left[\frac{\sum_{t \in \tau^S_{m,n}} I^S_{m,n}(t,s)}{\# \tau^S_{m,n}} \text{ for } s \in \mathcal{S} \right] \tag{6.29}$$

$$I^S_{m,n}(t,s) = \begin{cases} 1, & \hat{S}_{m,n}(t) = s \\ 0, & \text{else} \end{cases} \tag{6.30}$$

Similarly, the motion model state transition probability matrix $\hat{P}^V_{m,n}$ and the signal state transition probability matrix $\hat{P}^S_{m,n}$ can be estimated by node n for each target m by Eq. (6.31) and Eq. (6.34).

$$\hat{P}^V_{m,n} = \left[\frac{\sum_{t \in \tau^V_{m,n}} \delta^V_{m,n}(i,j,t)}{\sum_{t \in \tau^V_{m,n}} I^V_{m,n}(j,t)} \text{ for } i,j \in \mathcal{V} \right] \tag{6.31}$$

$$\delta^V_{m,n}(i,j,t) = \begin{cases} 1, & \hat{V}_{m,n}(t_0) = i \ \ \& \ \ \hat{V}_{m,n}(t) = j \\ 0, & \text{else} \end{cases} \tag{6.32}$$

$$t_0 = \sup_{t_0 \in \tau^V_{m,n}} \{ t_0 \text{ s.t. } t_0 < t \} \tag{6.33}$$

$$\hat{P}^S_{m,n} = \left[\frac{\sum_{t \in \tau^S_{m,n}} \delta^S_{m,n}(i,j,t)}{\sum_{t \in \tau^S_{m,n}} I^S_{m,n}(j,t)} \text{ for } i,j \in \mathcal{S} \right] \tag{6.34}$$

$$\delta^S_{m,n}(i,j,t) = \begin{cases} 1, & \hat{S}_{m,n}(t_0) = 1 \ \ \& \ \ \hat{V}_{m,n}(t) = j \\ 0, & \text{else} \end{cases} \tag{6.35}$$

$$t_0 = \sup_{t_0 \in \tau^V_{m,n}} \{ t_0 \text{ s.t. } t_0 < t \} \tag{6.36}$$

So, in each time step, node n is able to form estimated stationary distributions $\hat{S}_{m,n}$ and $\hat{V}_{m,n}$ for parameters S_m and V_m.

At the end of an epoch, the FC performs a similar process and estimates the distributions using the entire track for each target, as well as the increased number of measurements available to the FC. Eq. (6.37) represents the times at which any node chooses the active observation method and Eq. (6.38) represents the set of times at which any node chooses the passive observation method. Call the FC estimate for the motion model stationary distribution of target m $\hat{\pi}^V_m$ and call the FC estimate for the signal model stationary distribution of target m $\hat{\pi}^S_m$.

$$\tau^V_m = \bigcup_{n \in \mathcal{N}} \tau^V_{m,n} \tag{6.37}$$

$$\tau^S_m = \bigcup_{n \in \mathcal{N}} \tau^S_{m,n} \tag{6.38}$$

These are used to determine the portion of time each target spends in each state. Then, we use a modified k-means algorithm to cluster these distributions by similarity. The system must also perform model order estimation, as the number of true classes C is not known *a priori*. A few different modifications to k-means for distributions have been proposed in the literature, all on the metric used to determine distance between elements: [166] proposes Bregman divergences, [167] uses the Wasserstein metric (earthmover distance), [168] suggests the α-divergence. We select the p-Wasserstein metric Eq. (6.39) for its simplicity.

$$W_p(\hat{\pi}^V_{m_1}, \hat{\pi}^V_{m_2}) = \left(\frac{1}{V} \sum_{v=1}^{V} ||\hat{\pi}^V_{m_1}(v) - \hat{\pi}^V_{m_2}(v)||^p \right)^{1/p} \tag{6.39}$$

This results in $\hat{C}$ different classes. Construct sets $\mathcal{M}_{\hat{C}_i}$ containing the target tracks clustered to each class. Then, for each estimated class, we call Eq. (6.40) the class motion model stationary distribution, and similarly for Eq. (6.41) the class signal model stationary distribution.

$$\overline{\pi}^V_{\hat{C}_i} = \left[\sum_{m \in \mathcal{M}_{\hat{C}_i}} \left(\frac{\sum_{t \in \tau^V_m} I^V_m(t, v)}{\#\tau^V_m} \right) \text{ for } v \in \mathcal{V} \right] \tag{6.40}$$

$$\overline{\pi}^S_{\hat{C}_i} = \left[\sum_{m \in \mathcal{M}_{\hat{C}_i}} \left(\frac{\sum_{t \in \tau^S_m} I^S_m(t, s)}{\#\tau^S_m} \right) \text{ for } s \in \mathcal{S} \right] \tag{6.41}$$

After classes are formed they are distributed to each node. Then, new targets may be associated to a class. Then, by the following assumption, we can associate targets to classes using either their estimated motion model or signal model stationary distributions.

Assumption 4 (Single Family)**.** The targets present in the environment belong to a single target family.

Assumption 4 states that all classes may be represented by the same parametres. Targets are associated to a class by Eq. (6.42), at which point the corresponding tracking filter can be updated to use the motion model probabilities from the class.

$$C_{m,n} = \min_{c \in \mathcal{C}} \left(W_2(\hat{\pi}^V_{m,n}, \overline{\pi}^V_c) + W_2(\hat{\pi}^S_{m,n}, \overline{\pi}^S_c) \right) \tag{6.42}$$

6.5 Methods

6.5.1 Centralized Bandit

We use the common Upper Confidence Bound (**UCB**) [31] [107] formulation, where a single player selects from finitely many actions ("arms") and observes a corresponding reward. Over many iterations, the goal of the player is to maximize the total expected reward. To reduce the complexity[6], we pose the problem with one bandit algorithm per node, which are all evaluated by the FC. The algorithms could possibly be implemented by each node, but since the reward function (shown below) requires global information, this approach would require more communication.

Rewards The reward for each action is generated by the normalized Shannon entropy of the motion model distribution. This value is used since it is constrained to the unit interval and reflects the information content of the motion model distribution: as the distribution of states becomes more flat, the Shannon entropy will increase. Other works have often considered target tracking error (via Kalman filter covariance or Bayesian Cramér-Rao Lower Bound (**BCRLB**) as the error for related problems. This value has the problem of being very path-dependent; the history of observations can bias the estimated reward substantially. Instead, the proposed Shannon entropy reward reflects the estimated class of the target, and is less biased. This is particularly useful because as targets become more maneuverable, Kalman filters become less accurate and therefore benefit from more frequent updating [169].

$$u_n(t) = \frac{1}{M_n} \sum_{j=1}^{M_n} [\eta(V_j(t)), \eta(S_j(t))] \tag{6.43}$$

$$\eta(X(t)) = \sum_{i=1}^{n_X} \frac{x_i \log_2(x_i)}{\log_2(n_X)} \tag{6.44}$$

Eq. (6.43) shows the reward for selecting either action at node n, where $\eta(\cdot)$ represents the normalized Shannon entropy. Eq. (6.44) shows the Shannon entropy for a distribution X with n_X states x_i. The reward for selecting the radar action is dependent on the distribution of motion states of covered targets, and the reward for selecting the passive action is dependent on the distribution of the signal states of covered targets. So, the reward formulation is dependent on the all of the targets viewed by a particular node.

[6] An alternative approach might assign a single bandit algorithm with one arm per combination of node actions, which would total 2^N arms. Our approach covers the same action space, while reducing the number of arms per bandit algorithm to two.

Table 6.1: CRN Modes

Index	Mode
1	Active Radar
2	Passive ESM

Mode Selection Then, in each time step t, Eq. (6.45) is used to select the mode with index i for node n where $N_t(n)$ is the number of times each mode has been selected before time t. Table 6.1 shows the mode for each index.

$$\text{Mode}(t) = \underset{i \in u_n}{\text{argmin}} \left[u_n + \sqrt{\frac{\log t}{N_t(n)}} \right] \tag{6.45}$$

6.5.2 Distributed Approach

While a centralized technique has the benefit of more well-informed decision-making, it can suffer from latency. In other words, the time cost of moving information from the network's edge (the nodes) to the central decision-maker may cause extra error. Moving the decision to the edge can mitigate this effect.

We can begin by noting that the *distribution of target age is not stationary in time*. When the network is "switched on", all targets will be young. Since the target model allows for targets to enter and exit the scene, there is a fixed rate at which new targets appear after the beginning of the game. Since this is a Poisson process, i.e. the time between new target appearances is exponentially distributed, it is independent of the start time. Similarly, target death is exponentially distributed. Due to these, the discrepancy in target age distribution is most pronounced at early time steps.

The target age distribution is relevant to tracking performance. For a given target track, we should expect less error if the target is classified early in the track than if the target is classified later in the track.

Lastly we must also note that passive signal detection is a more reliable indicator of target class than target motion model, due to the more frequent updating of the signal Markov chain.

Due to this relationship between target age and tracking performance, we should expect that policies which prioritize passive observation early in each target track will result in lower tracking error. So, we present a distributed mode selection technique which is informed by target age.

Eq. (6.46) shows the utility function calculated by each node in each time step. The age of each track (Δ_m) is used to weight the Shannon entropy η Eq. (6.44) of the class stationary

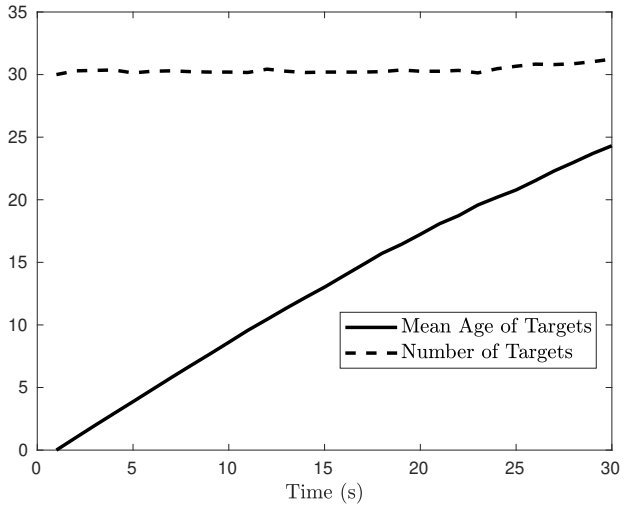


Figure 6.5: As the scenario ages, the mean target age increases.

distribution for target m, if it exists. If the target has not been associated with a class, the term γ is used to emphasize passive estimation so that the target may be classified. Older tracks are weighted less, as the importance of identifying new targets is greater.

$$\mathcal{U}(t) = \frac{1}{\#\mathcal{M}_n^{(t)}} \sum_{m \in \mathcal{M}_n^{(t)}} \frac{1}{\Delta_m} \left[f(m), 1 - f(m) \right] \tag{6.46}$$

$$f(m) = \begin{cases} \eta\left(\overline{\pi}_{C_i}^V\right), & \exists i \text{ s.t. } m \in \mathcal{M}_{C_i} \\ \gamma, & \text{else} \end{cases} \tag{6.47}$$

Then, the selected action at time t is given by Eq. (6.48), as randomly sampling the active or passive mode according to the weights in $\mathcal{U}$.

$$\text{Mode}(t) \overset{\text{R}}{\leftarrow} \mathcal{U}(t) \tag{6.48}$$

6.6 Numerical Simulations

We simulate a CRN with the parameters listed in Table 6.2. In particular, we simulate fifteen epochs. After each epoch (i.e., scenario) the CC updates the list of target classes. When the game begins, there are no classes, and when it ends the classes should have high accuracy. Figure 6.6 shows that this is the case; class accuracy increases in each epoch. Further, Figure 6.6 also shows that the accuracy with which targets are associated to a class increases in each epoch. This, coupled with the result shown in Figure 6.4 which shows that tracking accuracy improves when a tuned filter is used, implies that the observed tracking accuracy in the entire network should improve.

Table 6.2: Simulation Parameters

Description	Value	Variable
Node Density per km^2	0.2	λ_n
Target Density per km^2	0.3	λ_m
Simulated Region	100 km^2	$\lvert B \rvert$
Number of Classes	3	
Averaged Simulations	30	
Number of Epochs	15	
Epoch Duration	25s	

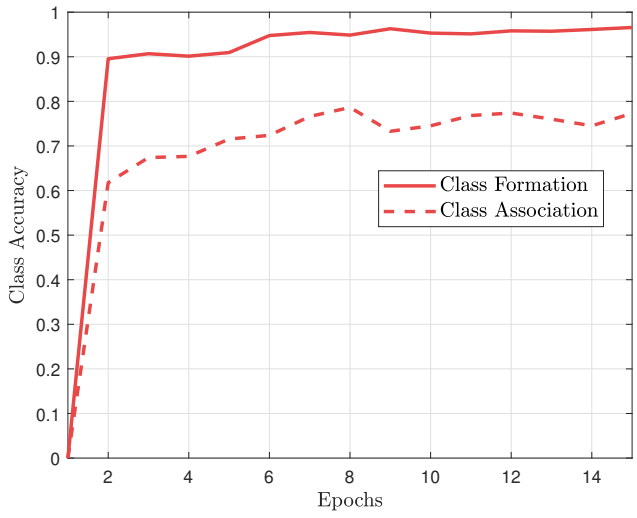


Figure 6.6: Class formation accuracy is higher than class association accuracy. This is due to the greater number of observations available to the FC during class formation; the nodes must rely on their own measurements to associate targets to classes.

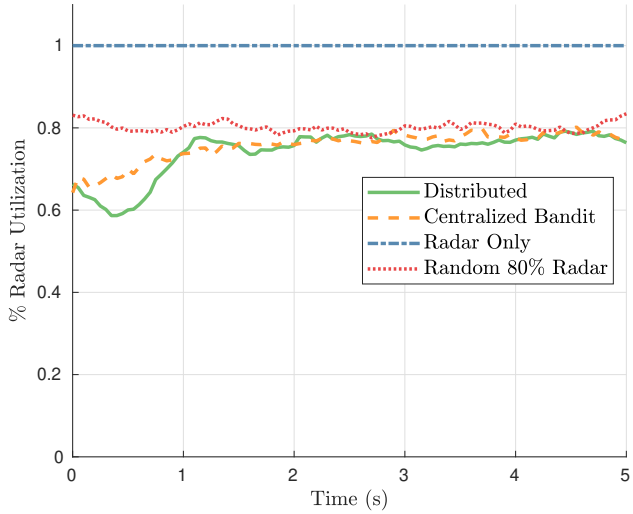


Figure 6.7: Radar is selected the indicated portion of time, with passive ESM being selected in the complementary portion of time. The centralized and distributed policies are compared against a policy which uses no mode control and only selects radar, and a policy which selects radar randomly 80% of the time.

We compare the distributed and centralized approaches against two baselines: the first selects radar all the time ("Radar Only"), and the second selects radar 80% of the time ("Random 80% Radar"). We choose these to compare against no mode control, and against a policy which provides equivalent max intercept range and power utilization. Figure 6.7 shows the percent of time in which radar is selected. Passive ESM is selected the other portion of time.

The maximum radar intercept range is the maximum range at which an intercept receiver could detect the radar emissions from a single node. Figure 6.8 shows the distribution of maximum intercept range generated by the four different policies. Since the centralized, distributed, and 80% radar policies all use radar approximately 80% of the time, they all have similar maximum intercept range distributions. Since the policy which only selects radar uses much more radar power, it has a much higher maximum intercept range.

Lastly, we present the tracking error generated by these policies. Since the active radar estimation only uses radar and does not utilize target classification, we should expect its performance to indicate the baseline. Since the 80% radar policy alternates between active radar and passive ESM, but does not use target classification, we should expect its performance to be worse than the only radar policy. Finally, since the centralized and distributed policies both use target classes and passive ESM estimation, we should expect their performance to be very good.

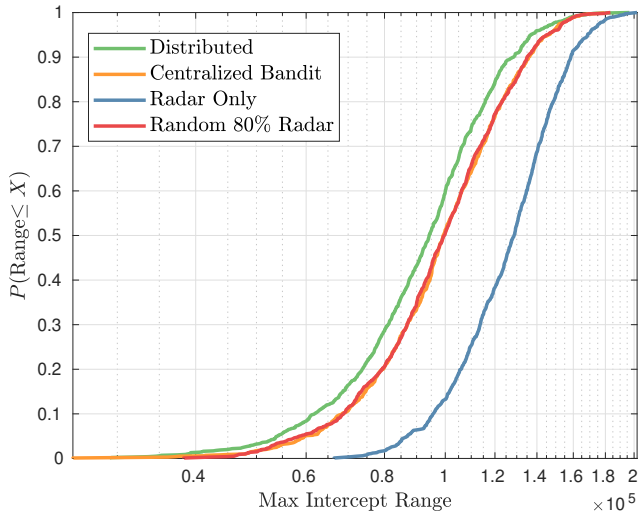


Figure 6.8: Distribution of the maximum intercept range for the four different policies.

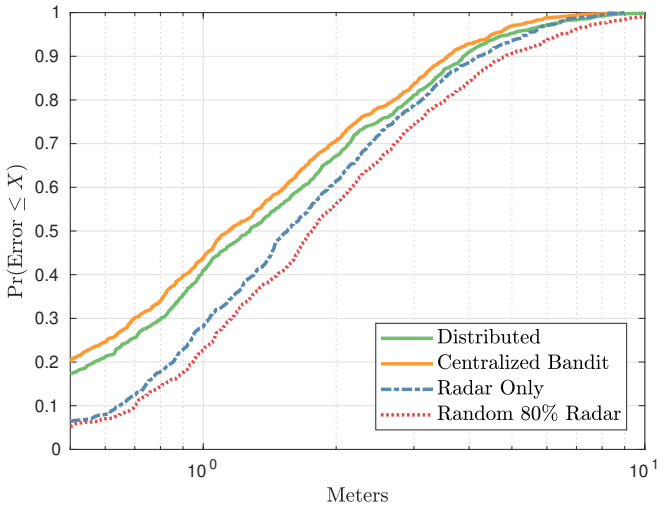


Figure 6.9: Tracking error distributions for all four policies.

6.7 Conclusions

In this work we investigated the capabilities available to a network when multiple modes of operation are present. This, along with our previously presented work [54], represents the first contribution towards the field of Cognitive Radar Network (CRN) mode control. In particular, we examine the case where the nodes of a CRN have, in addition to active radar, the ability to conduct passive signal parameter estimation. In each of many time steps, every node in the CRN can operate in one of these two modes. When conducting radar observation, a node provides to the Central Coordinator (CC) an estimate of the position and velocity of all targets within its range. When conducting passive signal parameter estimation, a node provides to the CC an estimate of the signal emissions from all targets within its range. The passive measurements are associated with targets via direction of arrival estimation.

In addition to these direct observations, the CC maintains records of the motion model (i.e., constant velocity, constant turn, etc.) and the history of signal emissions of each target. Modeling both of these as Markov processes, the CC estimates the transition probabilities for each of these parameters over time. On fixed intervals ("epochs''), the CC then clusters these targets into "classes'' which contain targets with similar behavior. Finally, using these constructed target classes, the CC is able to estimate the class of future targets in order to determine their likely behavior. In this way, the motion model of targets is able to be estimated using *passive* observation, as the class of a target is dependent on both signal and motion characteristics. So, the CC is able to trade between active radar observation and passive signal parameter estimation over time.

We show that the use of this estimation technique can 1) reduce the effective radiated power of the network by decreasing the proportion of time radar is performed and 2) increase the target tracking accuracy of the network by leveraging "prior" information on the targets. We demonstrate that passive signal parameter estimation is constrained by the maximum detectable range and probability of intercept of the targets emissions. In addition, we contribute mathematical analysis of the class formation technique.

We demonstrated that the distributed estimation technique can mitigate the effects of network latency by moving the decision-making to the edge of the network. The centralized technique, however, benefits from more complete knowledge of the targets.

Chapter 7

Discussion and Conclusion

This dissertation studies the applications and challenges associated with Cognitive Radar Networks (CRNs). While adaptive radar has existed for ~ 50 years, modern advances in semiconductors, electromagnetics, and computing speeds have enabled a new generation of researchers to study the role of environmental feedback in radar systems. While many contributions have been made towards single-node cognitive radar, the offerings in the literature regarding CRNs have not considered the greater challenges associated with networked cognitive radars, nor the abundant opportunities available to such a network.

7.1 Review of Contributions

This dissertation provides four contributions to the literature on Cognitive Radar Networks. In Chapter 3, we discussed a reinforcement learning framework in which many radar nodes may obtain near-optimal performance when selecting from several shared resources in the absence of dedicated communications. The example we studied involved each node selecting one channel per pulse transmission, where each available channel provides different radar performance generated by the environment. When multiple nodes occupied the same channel, the performance of those nodes was degraded due to mutual interference. Using multi-player multi-armed bandit reinforcement learning, we showed that such a network is reliably able to detect these instances of mutual interference and use them to exchange information between nodes, allowing the network to collaboratively converge to the optimal channel allocation.

Chapter 4 investigated the trade-off between distributed algorithms such as those in the previous chapter and centralized algorithms, paying attention to the minimum amount of information necessary to exchange in order to obtain near-optimal performance. We discussed a variety of algorithms and presented a data reduction technique, representing rewards in a target-independent formulation.

Chapter 5 focused on multi-target tracking in the regime where there are more targets than radar nodes. Here, we investigated *timely* updating of targets, where nodes must use a limited communication resource to transmit target updates to a central processor. Leveraging the fact that targets which are more maneuverable require more frequent updating (and that those which are less maneuverable require less frequent updating), we presented both a centralized and a distributed updating algorithms. First, we discussed a centralized technique

which had the advantage of knowledge of the targets observed by all nodes. However, this technique was limited by the network latency. The distributed technique was adapted from the age of incorrect information, a metric used in sensor systems. Using this distributed technique, we derived a Bellman-optimal updating policy which meets the same communication resource bound while requiring no input from a central node.

In Chapter 6 we present the final contribution of this dissertation. We investigate the trade-off between different operation modes within a CRN. In particular, we investigated the benefits of passive signal parameter estimation in addition to active radar observation. Since radar observation provides position and velocity information and requires no cooperation from the target, it tends to be the favored mode. For this reason, we investigated the utility provided by motion model estimation, and in particular, the utility provided by target class formation and estimation. Since targets of the same class exhibit similar kinematic behavior and similar signal emissions, we showed that passive signal parameter estimation can be used to estimate the class of a target and therefore its kinematics. We then provided an analysis of the performance of such a system.

7.2 Future Work

The problem of *power control*, which is the allocation of limited a limited power resource to the distributed nodes, has similarities to the distributed updating and mode control work investigated in this dissertation. A study of extended mode control, incorporating a default mode which performs no estimation, could draw inspiration from the power control literature to further reduce the observability of a radar network. This could be particularly useful for those nodes in the network which see very few targets, and may waste resources.

Further, there is opportunity to utilize more information available from signals emitted by targets; particularly direction of arrival estimates. This information could be used to update the position of the target in addition to updating the target signal model. Further, the use of networked mode control could allow a CRN to select modes in order to optimize passive localization in the sense of time delay of arrival estimation. Since this requires multiple anchors, it would require more coordination.

Appendices

Appendix A

Supplementary Materials for Chapter 3

A.1 Proof of Lemma 3.8

Lemma. Maximal utility implies maximum network SINR.

Proof. First, note that $\alpha\left(\text{SINR}+\beta\right) \geq 0$. Trivially, if this quantity is always equal to zero, utility is always maximized. So, without loss of generality, assume that $\alpha\left(\text{SINR}+\beta\right) > 0$.

Since $U(\pi) = U^*$ implies $I(t) = 0$ for all radars r_i in the set $\mathcal{P}$ and the matching π, we can simplify the reward equation to

$$\mu_{i,j} = \alpha\left(\text{SINR}_{i,j}+\beta\right).$$

Further,

$$\begin{aligned} U^* &= \max_{\pi\in\mathcal{M}} U(\pi(t)) \\ &= \max_{\pi\in\mathcal{M}} \sum_{r_i,f_j\in\pi} \mu_{i,j} \\ &= \max_{\pi\in\mathcal{M}} \sum_{r_i,f_j\in\pi} \alpha\left(\text{SINR}_{i,j}+\beta\right). \end{aligned}$$

Since α and β are constants and $\alpha\left(\text{SINR}_{i,j}+\beta\right)$, this implies that for $U(\pi) = U^*$,

$$\text{SINR}_{\pi} = \max_{\pi\in\mathcal{M}} \sum_{r_i,f_j\in\pi} \text{SINR}_{i,j}$$

which is the maximum network SINR. □

Appendix B

Supplementary Materials for Chapter 4

B.1 Proof of Lemma 4.6

Lemma. The optimal matching under SINR rewards is equal to the optimal matching under target-based rewards when Assumption 1 holds.

$$\max_{\pi \in \gamma(k)} U(\pi) = \max_{\pi \in W^{\Gamma}(k)} U(\pi) \tag{B.1}$$

Proof. For an element $x_{m,n} \in \gamma(k)$, $y_{m,n} \in W^{\Gamma}(k)$ can be written as Eq. (B.2).

$$\begin{aligned} y_{m,n} &= \left(\frac{1}{\hat{r}_m^4(k)}\right) \Gamma_{m,n} \\ &= \left(\frac{1}{\hat{r}_m^4(k)}\right) \frac{\gamma_{m,n}(k)}{\hat{P}_m(k)} \\ &= \left(\frac{1}{\hat{r}_m^4(k)}\right) \gamma_{m,n}(k) \frac{(4\pi)^3 \hat{r}_m^4(k)}{P_x G^2 \lambda^2} \\ &= \frac{(4\pi)^3}{P_x G^2 \lambda^2} \gamma_{m,n}(k) \end{aligned} \tag{B.2}$$

Now, since each element of W^{Γ} is equal to a constant times $\gamma(k)$, the optimal matching function will not change.

$$\max_{\pi \in \gamma(k)} U(\pi) = \max_{\pi \in W^{\Gamma}(k)} U(\pi) \tag{B.3}$$

□

Appendix C

Supplementary Materials for Chapter 5

C.1 Proof of Lemma 5.4

Lemma C.1 (Constraint Equivalence)**.** *If the AoII constraint is*

$$\delta = \frac{\alpha}{M_n} \tag{C.1}$$

then the average number of updates will be C.

Proof. Recall that the target density is λ_m, the node density is λ_n, and the capacity $C = \alpha\lambda_n|B|$ is the desired number of updates per CPI. Let R be the variable describing the distance from a point $x \in B$. Let M_n be the random variable describing the number of targets closest to any given node. For any given point in B, the probability that no nodes are within a radius r is

$$P(N = 0) = e^{-\lambda_n \pi r^2} \tag{C.2}$$

and therefore that the probability that at least one node is with a radius r is

$$P(N \geq 1) = 1 - e^{-\lambda_n \pi r^2} \tag{C.3}$$

Then, the PDF of the distance to the nearest node is

$$f(r) = \frac{d}{dr}\left(1 - e^{-\lambda_n \pi r^2}\right) \tag{C.4}$$

$$= 2\lambda_n \pi r e^{\lambda_n \pi r^2} \tag{C.5}$$

The expected value of this distribution can be found as

$$\mathbb{E}[R] = \int_0^\infty r f(r) dr \tag{C.6}$$

$$= \int_0^\infty \lambda_n \pi r^2 e^{-\lambda_n \pi r^2} \tag{C.7}$$

which is a Gaussian integral with solution

$$\mathbb{E}[R] = \frac{1}{2\sqrt{\lambda_n}} \tag{C.8}$$

This, then, is the expected distance from any point to the nearest node.

Now, the expected number of targets for which a given node is closest is given as the number of targets inside the disc of half this radius.

$$\mathbb{E}[M_n] = \lambda_m \pi \left(\frac{1}{4\sqrt{\lambda_n}}\right)^2 \tag{C.9}$$

$$= \frac{\pi}{16}\frac{\lambda_m}{\lambda_n} \tag{C.10}$$

Finally, the probability node n provides an update in the CPI ending at time $t = \tau$ is δ. We have:

$$\mathbb{E}[P(n \in \mathcal{N}^{(\tau)})] = \mathbb{E}[\prod_{i=1}^{M_n} \delta] \tag{C.11}$$

$$= \mathbb{E}[M_n \delta] \tag{C.12}$$

$$= \alpha \tag{C.13}$$

□

Bibliography

[1] S. Haykin, "Cognitive radar: a way of the future," *IEEE Signal Processing Magazine*, vol. 23, no. 1, pp. 30–40, 2006.

[2] S. Haykin, Y. Xue, and T. N. Davidson, "Optimal waveform design for cognitive radar," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pp. 3–7, 2008.

[3] A. Martone, K. Gallagher, K. Sherbondy, A. Hedden, and C. Dietlein, "Adaptable waveform design for enhanced detection of moving targets," *IET Radar, Sonar & Navigation*, vol. 11, no. 10, pp. 1567–1573, 2017.

[4] L. Brennan and L. Reed, "Theory of adaptive radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-9, no. 2, pp. 237–252, 1973.

[5] F. Daum, "Letter to the editor," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 6, pp. 78–78, 2020.

[6] S. Haykin, "Cognitive radar networks," in *Fourth IEEE Workshop on Sensor Array and Multichannel Processing, 2006.*, pp. 1–24, 2006.

[7] S. Haykin, "Cognitive networks: Radar, radio, and control for new generation of engineered complex networks," in *2013 IEEE Radar Conference (RadarCon13)*, pp. 1–4, 2013.

[8] A. Huizing, A. Charlish, and S. Brüggenwirth, "Response to the letter by fred daum," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 6, pp. 79–79, 2020.

[9] T. Kato, Y. Ninomiya, and I. Masaki, "An obstacle detection method by fusion of radar and motion stereo," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 182–188, 2002.

[10] W. Wijesoma, K. Kodagoda, and A. Balasuriya, "Road-boundary detection and tracking using ladar sensing," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 456–464, 2004.

[11] J. Dolcourt, "What's the deal with radar on a phone anyway? - cnet." https://www.cnet.com/tech/mobile/whats-the-deal-with-radar-on-a-phone-anyway/, 10 2019. (Accessed on 02/07/2023).

[12] NIST, "Spectrum sharing | nist." https://www.nist.gov/advanced-communications/spectrum-sharing, 2 2019. (Accessed on 02/07/2023).

[13] B. H. Kirk, A. F. Martone, K. D. Sherbondy, and R. M. Narayanan, "Performance analysis of pulse-agile sdradar with hardware accelerated processing," in *2020 IEEE International Radar Conference (RADAR)*, pp. 117–122, 2020.

[14] J. Li and P. Stoica, *MIMO Radar Signal Processing*. IEEE Press, Wiley, 2008.

[15] E. Fishler, A. Haimovich, R. Blum, L. Cimini, D. Chizhik, and R. Valenzuela, "Spatial diversity in radars—models and detection performance," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 823–838, 2006.

[16] H. D. Ly and Q. Liang, "Diversity in radar sensor networks: theoretical analysis and application to target detection," *International Journal of Wireless Information Networks*, vol. 16, pp. 209–216, 2009.

[17] Q. Liang, "Radar sensor networks: algorithms for waveform design and diversity with application to atr with delay-doppler uncertainty," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, pp. 1–9, 2007.

[18] J. Liang and Q. Liang, "Orthogonal waveform design and performance analysis in radar sensor networks," in *MILCOM 2006 - 2006 IEEE Military Communications conference*, pp. 1–6, 2006.

[19] J. Liang and Q. Liang, "Design and analysis of distributed radar sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 11, pp. 1926–1933, 2011.

[20] C. E. Thornton, R. M. Buehrer, and A. F. Martone, "Efficient online learning for cognitive radar-cellular coexistence via contextual thompson sampling," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, 2020.

[21] Q. Liang, "Clusterhead election for mobile ad hoc wireless network," in *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003.*, vol. 2, pp. 1623–1628 vol.2, 2003.

[22] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," in *IEEE INFOCOM 2004*, vol. 1, p. 640, 2004.

[23] R. Klemm, ed., *Novel Radar Techniques and Applications Volume 2: Waveform Diversity and Cognitive Radar, and Target Tracking and Data Fusion*. Radar, Sonar and Navigation, Institution of Engineering and Technology, 2017.

[24] M. Bell, "Information theory and radar waveform design," *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1578–1597, 1993.

[25] A. F. Martone, K. D. Sherbondy, J. A. Kovarskiy, B. H. Kirk, R. M. Narayanan, C. E. Thornton, R. M. Buehrer, J. W. Owen, B. Ravenscroft, S. Blunt, A. Egbert, A. Goad, and C. Baylis, "Closing the loop on cognitive radar for spectrum sharing," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 9, pp. 44–55, 2021.

[26] B. H. Kirk, J. W. Owen, R. M. Narayanan, S. D. Blunt, A. F. Martone, and K. D. Sherbondy, "Cognitive software defined radar: waveform design for clutter and interference suppression," in *Radar Sensor Technology XXI*, vol. 10188, pp. 446–461, SPIE, 2017.

[27] B. H. Kirk, R. M. Narayanan, K. A. Gallagher, A. F. Martone, and K. D. Sherbondy, "Avoidance of time-varying radio frequency interference with software-defined cognitive radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 3, pp. 1090–1107, 2018.

[28] B. Ravenscroft, J. W. Owen, J. Jakabosky, S. D. Blunt, A. F. Martone, and K. D. Sherbondy, "Experimental demonstration and analysis of cognitive spectrum sensing and notching for radar," *IET Radar, Sonar & Navigation*, vol. 12, no. 12, pp. 1466–1475, 2018.

[29] J. A. Kovarskiy, J. W. Owen, R. M. Narayanan, S. D. Blunt, A. F. Martone, and K. D. Sherbondy, "Spectral prediction and notching of rf emitters for cognitive radar coexistence," in *2020 IEEE International Radar Conference (RADAR)*, pp. 61–66, IEEE, 2020.

[30] L. Besson and E. Kaufmann, "Multi-player bandits revisited," in *Algorithmic Learning Theory*, pp. 56–92, PMLR, 2018.

[31] T. Lattimore and C. Szepesvari, *Bandit Algorithms*. Cambridge University Press, 2020.

[32] E. Boursier and V. Perchet, "A survey on multi-player bandits," 2022.

[33] E. Boursier and V. Perchet, "Sic-mmab: Synchronisation involves communication in multiplayer multi-armed bandits," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 12071–12080, Curran Associates, Inc., 2019.

[34] H. T. Hayvaci and B. Tavli, "Spectrum sharing in radar and wireless communication systems: A review," in *2014 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, pp. 810–813, 2014.

[35] Y. Nijsure, Y. Chen, C. Yuen, and Y. H. Chew, "Location-aware spectrum and power allocation in joint cognitive communication-radar networks," in *2011 6th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pp. 171–175, 2011.

[36] C. Aydogdu, M. F. Keskin, N. Garcia, H. Wymeersch, and D. W. Bliss, "Radchat: Spectrum sharing for automotive radar interference mitigation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 416–429, 2021.

[37] F. Sanders, U. S. N. Telecommunications, I. Administration, R. Sole, and B. Bedford, *Effects of RF Interference on Radar Receivers*. NTIA report, National Telecommunications and Information Administration, 2006.

[38] F. Paisana, N. J. Kaminski, N. Marchetti, and L. A. DaSilva, "Signal processing for temporal spectrum sharing in a multi-radar environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 2, pp. 123–137, 2017.

[39] U. Majumder, *Nearly orthogonal, doppler tolerant waveforms and signal processing for multi-mode radar applications*. PhD thesis, Purdue University, 2014.

[40] J-C Guey and M. R. Bell, "Diversity waveform sets for delay-doppler imaging," *IEEE Transactions on Information Theory*, vol. 44, no. 4, pp. 1504–1522, 1998.

[41] C. Shi, L. Ding, F. Wang, S. Salous, and J. Zhou, "Joint target assignment and resource optimization framework for multitarget tracking in phased array radar network," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4379–4390, 2021.

[42] J. Yan, H. Liu, W. Pu, H. Liu, Z. Liu, and Z. Bao, "Joint threshold adjustment and power allocation for cognitive target tracking in asynchronous radar network," *IEEE Transactions on Signal Processing*, vol. 65, no. 12, pp. 3094–3106, 2017.

[43] J. Sandenbergh, M. Weiss, F. V. Crespi, D. O'Hagan, and P. Knott, "An adaptive distributed clock for radar networks," in *2019 International Radar Conference (RADAR)*, pp. 1–5, 2019.

[44] M. Bande and V. V. Veeravalli, "Multi-user multi-armed bandits for uncoordinated spectrum access," 2018.

[45] S. Kang and C. Joo, "Combinatorial multi-armed bandits in cognitive radio networks: A brief overview," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1086–1088, 2017.

[46] 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.

[47] A. E. Mitchell, G. E. Smith, K. L. Bell, and M. Rangaswamy, "Single target tracking with distributed cognitive radar," in *2017 IEEE Radar Conference (RadarConf)*, pp. 0285–0288, 2017.

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

[49] H. Wang, K. Liao, S. Ouyang, and Z. Bai, "Advance stagger scheduling of pulses in cognitive radar networks," in *IET International Radar Conference (IET IRC 2020)*, vol. 2020, pp. 455–458, 2020.

[50] N. Pappas, M. A. Abd-Elmagid, B. Zhou, W. Saad, and H. S. Dhillon, *Age of Information: Foundations and Applications*. Cambridge University Press, 2023.

[51] W. W. Howard, A. F. Martone, and R. M. Buehrer, "Distributed online learning for coexistence in cognitive radar networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 2, pp. 1202–1216, 2023.

[52] W. W. Howard and R. M. Buehrer, "Hybrid cognition for target tracking in cognitive radar networks," 2023.

[53] W. W. Howard, A. F. Martone, and R. M. Buehrer, "Timely target tracking: Distributed updating in cognitive radar networks," 2023.

[54] W. W. Howard, S. R. Shebert, B. H. Kirk, and R. M. Buehrer, "Mode selection and target classification in cognitive radar networks," *arXiv preprint arXiv:2310.17006*, 2023.

[55] W. W. Howard, C. E. Thornton, A. F. Martone, and R. M. Buehrer, "Multi-player bandits for distributed cognitive radar," in *2021 IEEE Radar Conference (RadarConf21)*, pp. 1–6, IEEE, 2021.

[56] W. W. Howard, A. F. Martone, and R. M. Buehrer, "Adversarial multi-player bandits for cognitive radar networks," in *2022 IEEE Radar Conference (RadarConf22)*, pp. 1–6, IEEE, 2022.

[57] W. W. Howard and R. M. Buehrer, "Decentralized bandits with feedback for cognitive radar networks," in *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*, pp. 717–722, 2022.

[58] W. W. Howard, C. E. Thornton, and R. M. Buehrer, "Timely target tracking in cognitive radar networks," 2023.

[59] D. Tait, J. Yu, W. Howard, and R. M. Buehrer, "Direction of arrival estimation of digital sources with uni-vector-sensor esprit," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, 2020.

[60] J. Yu, W. W. Howard, D. Tait, and R. M. Buehrer, "Direction-of-arrival estimation with a vector sensor using deep neural networks," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pp. 1–7, 2021.

[61] W. W. Howard and R. M. Buehrer, "Multi-target localization using polarization sensitive arrays," in *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, pp. 612–617, 2021.

[62] J. Yu, W. W. Howard, Y. Xu, and R. M. Buehrer, "Model order estimation in the presence of multipath interference using residual convolutional neural networks," 2022.

[63] C. E. Thornton, W. W. Howard, and R. M. Buehrer, "Online learning-based waveform selection for improved vehicle recognition in automotive radar," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.

[64] R. M. Buehrer, W. W. Howard, and S. Ellingson, "Open and closed-loop weight selection for pattern control of paraboloidal reflector antennas with reconfigurable rim scattering," 2023.

[65] A. Slivkins, "Introduction to multi-armed bandits," *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.

[66] S. Ito, T. Tsuchiya, and J. Honda, "Adversarially robust multi-armed bandit algorithm with variance-dependent regret bounds," in *Conference on Learning Theory*, pp. 1421–1422, PMLR, 2022.

[67] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[68] Y. Sun, I. Kadota, R. Talak, E. Modiano, and R. Srikant, *Age of Information: A New Metric for Information Freshness*. Synthesis Lectures on Communication Networks, Morgan & Claypool Publishers, 2019.

[69] M. Haenggi, *Stochastic Geometry for Wireless Networks*. Cambridge University Press, 2012.

[70] T. Tao, *An Introduction to Measure Theory*. Graduate Studies in Mathematics, American Mathematical Society, 2021.

[71] J. Hamilton, *Time Series Analysis*. Princeton University Press, 2020.

[72] B.-N. Vo and W.-K. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[73] R. Mahler, "Phd filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, 2007.

[74] B. Ristic, D. Clark, B.-N. Vo, and B.-T. Vo, "Adaptive target birth intensity for phd and cphd filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1656–1668, 2012.

[75] K. Punithakumar, T. Kirubarajan, and A. Sinha, "Multiple-model probability hypothesis density filter for tracking maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 87–98, 2008.

[76] K. Panta, D. E. Clark, and B.-N. Vo, "Data association and track management for the gaussian mixture probability hypothesis density filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, pp. 1003–1016, 2009.

[77] N. Levanon, E. Mozeson, R. Signals, and C. Levanon, *Radar Signals*. John Wiley & Sons, Ltd, 2013.

[78] L. Scharf and C. Demeure, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley series in electrical and computer engineering, Addison-Wesley Publishing Company, 1991.

[79] M. Richards, *Fundamentals of Radar Signal Processing, Third Edition*. McGraw Hill LLC, 2022.

[80] B. Mahafza, *Radar Systems Analysis and Design Using MATLAB Second Edition*. Taylor & Francis, 2005.

[81] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, 2007.

[82] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[83] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527 – 535, 1952.

[84] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5667–5681, 2010.

[85] S. Haykin, "Cognitive dynamic systems: Radar, control, and radio [point of view]," *Proceedings of the IEEE*, vol. 100, no. 7, pp. 2095–2103, 2012.

[86] A. F. Martone, "Cognitive radar demystified," *URSI Radio Science Bulletin*, vol. 2014, no. 350, pp. 10–22, 2014.

[87] IEEE, "IEEE Standard for Radar Definitions," *IEEE Std 686-2017 (Revision of IEEE Std 686-2008)*, pp. 1–54, 2017.

[88] A. F. Martone, K. I. Ranney, K. Sherbondy, K. A. Gallagher, and S. D. Blunt, "Spectrum allocation for noncooperative radar coexistence," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 90–105, 2018.

[89] S. Z. Gurbuz, H. D. Griffiths, A. Charlish, M. Rangaswamy, M. S. Greco, and K. Bell, "An overview of cognitive radar: Past, present, and future," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 12, pp. 6–18, 2019.

[90] A. F. Martone, K. D. Sherbondy, J. A. Kovarskiy, B. H. Kirk, C. E. Thornton, J. W. Owen, B. Ravenscroft, A. Egbert, A. Goad, A. Dockendorf, R. M. Buehrer, R. M. Narayanan, S. D. Blunt, and C. Baylis, "Metacognition for radar coexistence," in *2020 IEEE International Radar Conference (RADAR)*, pp. 55–60, 2020.

[91] C. E. Thornton, R. M. Buehrer, and A. F. Martone, "Constrained contextual bandit learning for adaptive radar waveform selection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 2, pp. 1133–1148, 2022.

[92] Jian Li and Petre Stoica, *MIMO RADAR SIGNAL PROCESSING*. London: Bantam, 1988.

[93] A. Deligiannis, A. Panoui, S. Lambotharan, and J. A. Chambers, "Game-theoretic power allocation and the nash equilibrium analysis for a multistatic mimo radar network," *IEEE Transactions on Signal Processing*, vol. 65, no. 24, pp. 6397–6408, 2017.

[94] E. Fishler, A. Haimovich, R. S. Blum, L. J. Cimini, D. Chizhik, and R. A. Valenzuela, "Spatial diversity in radars—models and detection performance," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 823–838, 2006.

[95] W. Zhang, X. Yin, X. Cao, Y. Xie, and W. Nie, "Radar emitter identification using hidden markov model," in *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1997–2000, 2019.

[96] E. Selvi, R. M. Buehrer, A. Martone, and K. Sherbondy, "Reinforcement learning for adaptable bandwidth tracking radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 5, pp. 3904–3921, 2020.

[97] S. Alland, W. Stark, M. Ali, and M. Hegde, "Interference in automotive radar systems: Characteristics, mitigation techniques, and current and future research," *IEEE Signal Processing Magazine*, vol. 36, no. 5, pp. 45–59, 2019.

[98] G. Kim, J. Mun, and J. Lee, "A peer-to-peer interference analysis for automotive chirp sequence radars," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8110–8117, 2018.

[99] G. M. Brooker, "Mutual interference of millimeter-wave radar systems," *IEEE Transactions on Electromagnetic Compatibility*, vol. 49, no. 1, pp. 170–181, 2007.

[100] W. Wang and H. Shao, "Radar-to-radar interference suppresion for distributed radar sensor networks," *Remote Sens.*, vol. 6, pp. 740–755, 2014.

[101] D. Vandenberg, "Mathematical survey and application of the cross-ambiguity function," Master's thesis, Indiana University South Bend, 2012.

[102] B. Tang, W. Huang, and J. Li, "Slow-time coding for mutual interference mitigation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6508–6512, 2018.

[103] C. Fischer, H. L. Blöcher, J. Dickmann, and W. Menzel, "Robust detection and mitigation of mutual interference in automotive radar," in *2015 16th International Radar Symposium (IRS)*, pp. 143–148, 2015.

[104] E. Knott, *Radar Cross Section Measurements*. SciTech Pub., 2006.

[105] Y. Seldin, C. Szepexvari, P. Auer, and Y. Abbasi-Yadkori, "Evaluation and analysis of the performance of the EXP3 algorithm in stochastic environments," in *European Workshop on Reinforcement Learning*, pp. 55–60, 2012.

[106] J. Rosenski, O. Shamir, and L. Szlak, "Multi-player bandits – a musical chairs approach," in *Proceedings of Machine Learning Research* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48, (New York, New York, USA), pp. 155–163, PMLR, 20–22 Jun 2016.

[107] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, pp. 235–256, 05 2002.

[108] R. Sutton and A. Barto, *Reinforcement Learning, second edition: An Introduction*. Adaptive Computation and Machine Learning series, MIT Press, 2018.

[109] C. E. Thornton, M. A. Kozy, R. M. Buehrer, A. F. Martone, and K. D. Sherbondy, "Deep reinforcement learning control for radar detection and tracking in congested spectral environments," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2020.

[110] M. Liggins, C.-Y. Chong, I. Kadar, M. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 95–107, 1997.

[111] R. R. Tenney and N. R. Sandell, "Detection with distributed sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-17, no. 4, pp. 501–510, 1981.

[112] S. Choi, D. Crouse, P. Willett, and S. Zhou, "Multistatic target tracking for passive radar in a dab/dvb network: initiation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 2460–2469, 2015.

[113] T. A. Severson and D. A. Paley, "Distributed multitarget search and track assignment with consensus-based coordination," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 864–875, 2015.

[114] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive iot networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4641–4654, 2020.

[115] A. Mehrabian, E. Boursier, E. Kaufmann, and V. Perchet, "A practical algorithm for multiplayer bandits when arm means vary among players," in *Proceedings of Machine Learning Research* (S. Chiappa and R. Calandra, eds.), vol. 108, (Online), pp. 1211–1221, PMLR, 26–28 Aug 2020.

[116] M. S. Greco, F. Gini, P. Stinco, and K. Bell, "Cognitive radars: On the road to reality: Progress thus far and possibilities for the future," *IEEE Signal Processing Magazine*, vol. 35, no. 4, pp. 112–125, 2018.

[117] C. E. Thornton, R. Michael Buehrer, and A. F. Martone, "Constrained online learning to mitigate distortion effects in pulse-agile cognitive radar," in *2021 IEEE Radar Conference (RadarConf21)*, pp. 1–6, 2021.

[118] C. E. Thornton, R. M. Buehrer, H. S. Dhillon, and A. F. Martone, "Universal learning waveform selection strategies for adaptive target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–17, 2022.

[119] D. Vernon, G. Metta, and G. Sandini, "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 151–180, 2007.

[120] J. A. Simmons, "The resolution of target range by echolocating bats," *The Journal of the Acoustical Society of America*, vol. 54, no. 1, pp. 157–173, 1973.

[121] X. Liu, Z.-H. Xu, L. Wang, W. Dong, and S. Xiao, "Cognitive dwell time allocation for distributed radar sensor networks tracking via cone programming," *IEEE Sensors Journal*, vol. 20, no. 10, pp. 5092–5101, 2020.

[122] H. Zhang, W. Liu, J. Xie, Z. Zhang, and W. Lu, "Joint subarray selection and power allocation for cognitive target tracking in large-scale mimo radar networks," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2569–2580, 2020.

[123] M. Jahangir, C. J. Baker, M. Antoniou, B. Griffin, A. Balleri, D. Money, and S. Harman, "Advanced cognitive networked radar surveillance," in *2021 IEEE Radar Conference (RadarConf21)*, pp. 1–6, 2021.

[124] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

[125] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[126] W. Jouini, D. Ernst, C. Moy, and J. Palicot, "Multi-armed bandit based policies for cognitive radio's decision making issues," in *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*, pp. 1–6, 2009.

[127] A. Anandkumar, N. Michael, and A. Tang, "Opportunistic spectrum access with multiple users: Learning under competition," in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, 2010.

[128] O. Avner and S. Mannor, "Multi-user communication networks: A coordinated multi-armed bandit approach," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2192–2207, 2019.

[129] J. Komiyama, J. Honda, and H. Nakagawa, "Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays," in *International Conference on Machine Learning*, pp. 1152–1161, PMLR, 2015.

[130] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 151–159, PMLR, 17–19 Jun 2013.

[131] P. Alatur, K. Y. Levy, and A. Krause, "Multi-player bandits: The adversarial case," *Journal of Machine Learning Research*, 2020.

[132] H. Van Trees, K. Bell, and Z. Tian, *Detection Estimation and Modulation Theory, Part I: Detection, Estimation, and Filtering Theory*. No. pt. 1 in Detection Estimation and Modulation Theory, Wiley, 1968.

[133] J. Yan, W. Pu, S. Zhou, H. Liu, and Z. Bao, "Collaborative detection and power allocation framework for target tracking in multiple radar system," *Information Fusion*, vol. 55, pp. 173–183, 2020.

[134] J. Yan, W. Pu, S. Zhou, H. Liu, and M. S. Greco, "Optimal resource allocation for asynchronous multiple targets tracking in heterogeneous radar networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4055–4068, 2020.

[135] J. Yan, H. Jiao, W. Pu, C. Shi, J. Dai, and H. Liu, "Radar sensor network resource allocation for fused target tracking: A brief review," *Information Fusion*, vol. 86-87, pp. 104–115, 2022.

[136] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[137] J. Yan, B. Jiu, H. Liu, B. Chen, and Z. Bao, "Prior knowledge-based simultaneous multibeam power allocation algorithm for cognitive multiple targets tracking in clutter," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 512–527, 2015.

[138] J. Yan, H. Liu, B. Jiu, B. Chen, Z. Liu, and Z. Bao, "Simultaneous multibeam resource allocation scheme for multiple target tracking," *IEEE Transactions on Signal Processing*, vol. 63, no. 12, pp. 3110–3122, 2015.

[139] T. D. Ridder, A. F. Martone, B. H. Kirk, and R. M. Narayanan, "Multiple criteria operational reliability performance metric of a metacognitive tracking radar," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–10, 2023.

[140] C. Thornton and R. Buehrer, "When is cognitive radar beneficial? Insights from dynamic spectrum access," in *2023 IEEE Radar Conference (RadarConf23)*, pp. 1–6, 2023.

[141] B. Ristic, B.-N. Vo, D. Clark, and B.-T. Vo, "A metric for performance evaluation of multi-target tracking algorithms," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3452–3457, 2011.

[142] R. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[143] R. P. S. Mahler, B.-T. Vo, and B.-N. Vo, "Cphd filtering with unknown clutter rate and detection profile," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3497–3513, 2011.

[144] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?," in *2012 Proceedings IEEE INFOCOM*, pp. 2731–2735, 2012.

[145] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.

[146] M. A. Abd-Elmagid and H. S. Dhillon, "Average peak age-of-information minimization in uav-assisted iot networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 2003–2008, 2019.

[147] I. Krikidis, "Average age of information in wireless powered sensor networks," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 628–631, 2019.

[148] R. D. Yates and S. Kaul, "Real-time status updating: Multiple sources," in *2012 IEEE International Symposium on Information Theory Proceedings*, pp. 2666–2670, 2012.

[149] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The age of incorrect information: A new performance metric for status updates," *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2215–2228, 2020.

[150] S. Kriouile and M. Assaad, "Minimizing the age of incorrect information for real-time tracking of markov remote sources," in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2978–2983, 2021.

[151] C. Kam, S. Kompella, and A. Ephremides, "Age of incorrect information for remote estimation of a binary markov source," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2020.

[152] B. Li, Z. Gan, D. Chen, and D. Sergey Aleksandrovich, "UAV maneuvering target tracking in uncertain environments based on deep reinforcement learning and meta-learning," *Remote Sensing*, vol. 12, no. 22, 2020.

[153] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[154] M. Richards, J. Scheer, and W. Holm, *Principles of Modern Radar: Basic principles*. Tes Dee Publishing Pvt. Limited, (Published by arrangement), 2012.

[155] S. Haykin, "Cognitive radar networks," in *1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2005.*, pp. 1–3, 2005.

[156] G. Alirezaei, M. Reyer, and R. Mathar, "Optimum power allocation in sensor networks for passive radar applications," *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3222–3231, 2014.

[157] R. Howard, *Dynamic Probabilistic Systems, Volume I: Markov Models*. Dover Books on Mathematics, Dover Publications, 2007.

[158] J. Norris, *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 1998.

[159] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2003.

[160] A. Munari, L. Simić, and M. Petrova, "Stochastic geometry interference analysis of radar network performance," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2362–2365, 2018.

[161] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3460–3475, 2013.

[162] K. Granstrom, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications," 2017.

[163] W. A. Jerjawi, Y. A. Eldemerdash, and O. A. Dobre, "Second-order cyclostationarity-based detection of lte sc-fdma signals for cognitive radio systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 3, pp. 823–833, 2015.

[164] A. Al-Habashna, O. A. Dobre, R. Venkatesan, and D. C. Popescu, "Joint signal detection and classification of mobile WiMAX and LTE OFDM signals for cognitive radio," in *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, pp. 160–164, 2010.

[165] R. Wiley, *ELINT: The Interception and Analysis of Radar Signals*. Artech House, 2006.

[166] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with bregman divergences," *Journal of Machine Learning Research*, vol. 6, no. 58, pp. 1705–1749, 2005.

[167] C. Villani, *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften, Springer Berlin Heidelberg, 2008.

[168] F. Nielsen, R. Nock, and S.-i. Amari, "On clustering histograms with k-means by using mixed α-divergences," *Entropy*, vol. 16, no. 6, pp. 3273–3301, 2014.

[169] M. Silbert, S. Sarkani, and T. Mazzuchi, "Comparing the state estimates of a Kalman filter to a perfect IMM against a maneuvering target," in *14th International Conference on Information Fusion*, pp. 1–5, 2011.