Adaptive Sensing Techniques for Dynamic Target Tracking and Detection with Applications to Synthetic Aperture Radars

by

Gregory Evan Newstadt

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2013

Doctoral Committee:

Professor Alfred O. Hero, III, Chair
Dean David C. Munson, Jr.
Assistant Professor Rajesh Rao Nadakuditi
Assistant Professor Shuheng Zhou

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Adaptive Sensing Techniques for Dynamic Target Tracking and Detection with Applications to Synthetic Aperture Radars

by
Gregory Evan Newstadt

Chair: Alfred O. Hero, III

This thesis studies adaptive allocation of a limited set of sensing or computational resources in order to maximize some criteria, such as detection probability, estimation accuracy, or throughput, with specific application to inference with synthetic aperture radars (SAR). Sparse scenarios are considered where the interesting element is embedded in a much larger signal space. For example, in wide area surveillance using synthetic aperture radars, the goal is to localize and track moving vehicles over a large scene. In this application, resources may be constrained in two ways: (a) limited dwell time of the radar in any particular location; and (b) limited computational resources in order to have a real-time detection/tracking system. Policies are examined that adaptively distribute the constrained resources by using observed measurements to inform the allocation at subsequent stages. This thesis studies adaptive allocation policies in three main directions.

First, a framework for adaptive search for sparse targets is proposed to simultaneously detect and track moving targets. Previous work is extended to include a dynamic target model that incorporates target transitions, birth/death probabilities, and varying target amplitudes. Policies are proposed that are shown empirically to have excellent asymptotic performance in estimation error, detection probability, and robustness to model mismatch. Moreover, policies are provided with low computational complexity as compared to state-of-the-art dynamic programming solutions.

Second, adaptive sensor management is studied for stable tracking of targets under different modalities. Using the guaranteed uncertainty management principle, a sensor scheduling policy is proposed that guarantees that the target spatial uncertainty remains bounded. When stability conditions are met, fundamental performance limits are derived such as the maximum number of targets that can be tracked stably, the maximum spatial uncertainty of those targets, and the system occupancy rates. The theory is extended to the case where the system may be engaged in tasks other than tracking, such as wide area search or target classification. Also, performance limits such as maximum load margin and multipurpose occupancy rates are provided.

Lastly, these developed tools are applied to a specific application, namely tracking targets using SAR imagery. A hierarchical Bayesian model is proposed for efficient estimation of the posterior distribution for the target and clutter states given observed SAR imagery. This model provides a unifying framework that combines working knowledge of the physical, kinematic, and statistical properties of SAR imagery. It is shown that this posterior estimation technique generally outperforms common algorithms for change detection. Moreover, the proposed method has the additional benefits of (a) easily incorporating additional information such as target motion models and/or correlated measurements, (b) having few tuning parameters, and (c) providing a characterization of the uncertainty in the state estimation process.

CHAPTER I

Introduction

Everyday life is full of situations where we choose how to best utilize limited resources. For example, one may consider choosing what to buy at a grocery store with a restricted monetary budget or how to plan an education course schedule within a limited time period. In both cases, the 'optimal' choice depends on the cost that we wish to optimize. In the former case, we may want to either maximize nutritional value or maximize palate acceptability by all members of the family. In the latter case, we may choose to maximize course load or job marketability. Moreover, these cost functions will likely change over time: in the former case, nutritional requirements or food tastes may change over time; in the latter case, academic interests may change (e.g., from math to engineering or vice versa).

This dissertation generally considers applications where an 'agile' sensor can be used to scan individual components of a scene. Resources are limited in the sense that there is an upper bound on the total amount of time, energy, or computation that can be used over the entire scene. Performance is then measured by our ability to detect/estimate the components of interest within the scene. Moreover, we focus on applications where we can adaptively allocate the limited resources in order to estimate and detect a 'sparse' element within a larger signal.

In particular, this thesis pursues three distinct directions: (1) the development of adaptive policies for searching for a sparse number of targets under resource constraints (Chapters II and III); (2) development of fundamental performance limits for tracking moving targets that guarantee a prescribed level of system performance as a function of a given system provisioning (Chapter IV); and (3), application of these adaptive techniques to a specific application, namely tracking moving vehicles with synthetic aperture radars (Chapter V). This chapter continues with brief introductions of these directions, my contributions to the field, and a comprehensive literature review of related work. Finally, in Chapter VI, we conclude and point to future work.

1.1 Adaptive sensing under resource constraints

The first direction of this work concerns itself with the problem of localizing and estimating targets in noise using energy-constrained measurements. In particular, the work focuses on problems where targets occupy only a small fraction of the scanned domain, which is referred to as the 'region of interest' (ROI).

This work is primarily motivated by two applications. In early cancer detection, the goal is to scan the body for tumors on the order of one cubic centimeter placed somewhere inside the torso. Moreover, the constrained resource is the maximum amount of ionizing radiation that can be safely endured by the patient. In target detection/tracking with radars, the analyst is required to scan a large field of view (FOV), where the number of radar cells containing targets is often much smaller than the size of the scene. Moreover, to satisfy real-time constraints, the total amount of radar dwell time is often limited.

In both of these applications, the common search scheme is to scan all possible locations with an equal effort allocation, which we term an ‘exhaustive policy’. In computed tomography (CT) scans, this is equivalent to using the same energy level for each CT projection. In radar tracking, an exhaustive search scans each radar cell with equal dwell time. An exhaustive search can be considered a special case of a static resource allocation policy; i.e., where the allocation efforts are predetermined before any action is taken. This work considers the development of adaptive policies (or adaptive sensing), where the allocation scheme is allowed to change over time as a function of previous observations. Indeed, by using adaptive sensing, it has been shown that one can perform significantly better compared to static polices. Bashan et al. has shown benefits including near-optimal gains in estimation error and related cost functions [11] as well as provable convergence to the true support of of the sparse signal [11]. Haupt et al. also considers adaptive sensing, demonstrating convergence rates that are significantly faster than non-adaptive policies [48], and proving that adaptive policies can reliably detect/estimate targets with significantly smaller minimum amplitudes below which signal detection is impossible [50], as compared to non-adaptive strategies.

My work is heavily influenced by the development of adaptive sensing schemes by Bashan and Hero [11] where a novel cost function was introduced, and a solution to a related minimization problem yielded an asymptotically optimal two-stage adaptive resource allocation policy, namely ARAP. In [9], Bashan proposes a multiple-scale modification (M-ARAP) that leads to significant savings in the number of measurements. The first contribution of my thesis (a) provides extended performance analysis in the multiple-scale case, (b) compares the computational complexity of M-ARAP to other competing methods, and (c) theoretically analyzes the asymptotic behavior of M-ARAP.

The second contribution of this work is to extend ARAP in two significant ways: (1) the allocation policy is broadened to $T \gg 2$ stages, and (2) targets are allowed to exhibit time-varying behavior, such as transitioning between cells, entering/exiting the scene, and/or being obscured. A novel adaptive resource allocation policy for simultaneously localizing and estimating dynamic targets, namely D-ARAP, is introduced. D-ARAP has low computational complexity as compared to other approaches in the literature, yet it can be easily generalized to a multitude of state, target, and measurement models. Moreover, empirical performance analysis of D-ARAP has shown excellent properties as either $T \to \infty$ or SNR$\to \infty$ in comparison to both exhaustive and greedy alternatives. The performance of D-ARAP is compared to oracle policies as well as online policies (which have much higher computational complexities) and the utility of this approach is demonstrated on a target tracking example using synthetic aperture radar imagery.

1.2 Sensor management and provisioning through the guaranteed uncertainty principle

In the next section of this thesis, we look at sensor management from the viewpoint of developing fundamental performance limits for stable tracking of targets with different modalities. In Chapters II and III, we considered resource management in the context of applying a limited set of resources to detect, estimate, and/or localize a sparse number of targets in an efficient manner. In practice, the signal processing algorithms used for tracking targets with radars are different than those used for detection or track initialization. Nevertheless, we are still interested in efficient methods for maintaining tracks on targets, where the constrained resources may be related to the physical sensor, such as when we have limited dwell times per radar cell, or may be abstractions, such as when we have real-time processing constraints. The objective, however, differs in the sense that we would like to maximize the number of targets that can tracked in a stable fashion (i.e., the uncertainty of the target states remain bounded).

We propose a general framework for maintaining stable track on N targets that includes (a) a system model which describes the amount of service time required to reduce target uncertainty to a nominal value, and (b) a target state model which describes the growth of state uncertainty as a function of time and system parameters. We propose using the prioritized longest queue (PLQ) policy, a variant of the 'largest weighted queue length' policy [89] proposed by Wasserman et al., to assign free resources (i.e., time required to reduce uncertainty to a nominal value) to track all N targets. We provide conditions for stable tracking of targets under the PLQ policy. Moreover, by solving a system of balance equations, we are able to provide fundamental performance limits, such as

1. The number of targets that can be tracked stably.
2. The system occupancy rates; i.e., the amount of wasted resources that the system could use more efficiently for other tasks.
3. The maximum uncertainty error; i.e., how large the entropy/uncertainty on the targets state can grow.

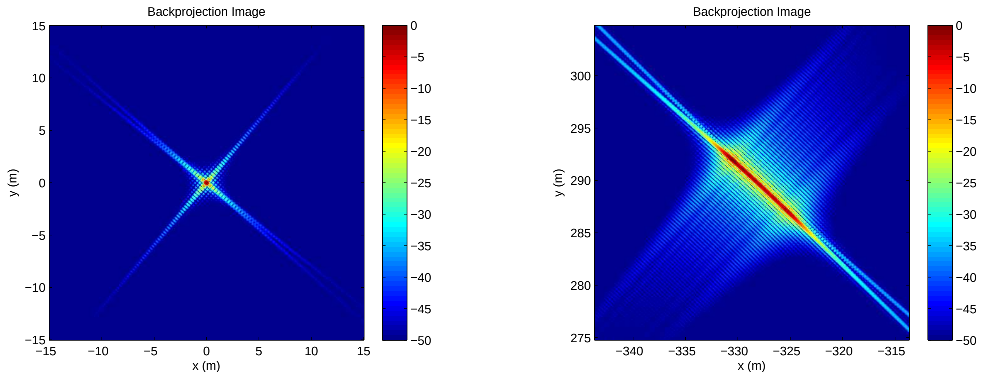
We provide several example modalities for which we can apply these performance limits, including tracking targets with both synthetic aperture radars and multistatic passive radars. It should be noted that these performance limits consider the worst-case scenario, where all targets are equally difficult to track.

1.3 Applications to synthetic aperture radar (SAR) imagery

The last direction of this thesis is concerned with developing adaptive sensing techniques for localizing and tracking vehicles using synthetic aperture radar (SAR) imagery. The ability to track moving targets with airborne radars is a problem that has drawn considerable interest from both the academic and government communities. In many cases, the opportunity now exists for continuous observation of regions of interest. However, the amount of information available often severely outpaces our ability to extract the information needed for decision-making. Indeed, Lt. Gen. David A. Deptula, the U.S. Air Force Deputy Chief of Staff for Intelligence, Surveillance, and Reconnaissance, recently remarked that we will "find ourselves in the not too distant future swimming in sensors and drowning in data." [37].

Motivated by the work in Chapters II through IV, it may be possible to efficiently localize/classify targets or detect anomalous behavior by adaptively managing the available resources. In practice, we may be interested in deciding how to adaptively collect radar pulses to optimize our performance criteria. Unfortunately, current SAR systems do not possess this capability. On the other hand, we consider the problem of allocating computational resources to efficiently use previously collected SAR samples. Applications include (a) tracking multiple targets over a sparse state space and (b) efficiently reconstructing the scene of interest only in the (sparse) locations where targets exist. Note that the latter problem is related to the simultaneous detection/estimation problem discussed with regard to adaptive sampling.

Airborne radar systems may operate in a multitude of modes depending on the application. In moving target indication (MTI), the radar focuses a narrow beam over small regions in the field of view for small integration times on the order of



(a) Stationary

(b) Moving

Figure 1.1: Here SAR images constructed through the backprojection method provided by Gorham and Moore [44] are shown for point targets. In (a) the point target is stationary at $(0, 0)$ and the majority of the energy is focused at that point. In (b) the point target has velocity $(v_x, v_y) = (30, 5)\ m/s$ and acceleration $(a_x, a_y) = (3, 1)\ m/s^2$. The target is both displaced in the image (by more than 300 meters) and smeared (with smear length of about 10 meters).

milliseconds. Since MTI radars illuminate radar cells independently, they face the tradeoff of long dwell times (that lead to improved detection/track accuracies) versus the number of targets than can be detected or tracked stably. This may prohibit efficient analysis of large fields of view (FOV).

Synthetic aperture radar (SAR) mode has traditionally been used to image stationary or slow-moving targets over a much larger FOV than other airborne radar operating modes, particularly with respect to MTI. By integrating radar pulses from spatially diverse points in the radar trajectory, SAR data can be used to form 2- or 3-dimensional images with much finer resolutions than MTI due to the ability to use long integration times. However, the situation becomes complex when considering moving targets, which can cause phase errors in the reconstruction of SAR images. This leads to well known defocusing and displacement of the target's energy (Jao [54], Fienup [42], and Newstadt et al. [70]). As an example, Figure 1.1 displays two SAR images constructed from ideal phase histories for 1 second collected from point targets located at the origin. In the left plot, the target is stationary, leading to a focused image at the origin. In the right plot, the target has velocity $(v_x, v_y) = (30, 5)$ m/s and acceleration $(a_x, a_y) = (3, 1)$ m/s^2, with respect to a radar moving with velocity $(v_x^{radar}, v_y^{radar}) = (100, 0)$ m/s. This moving target is both displaced in the image (by more than 300 m) and its energy is smeared over approximately 10 m. We refer to the target energy within a reconstructed SAR image as the 'target signature,' which is focused for stationary targets and displaced/dispersed for moving targets.

Regardless of the complexities of moving targets, target tracking with SAR imagery has been well studied in the literature. This includes methods that directly estimate the phase errors induced by moving targets such as in the work of Jao [54] and Fienup [42], as well as a multitude of algorithms for extracting moving targets from a background embedded in a low-dimensional subspace (Soumekh [81], Ender [38], Erten [39], and Ranney and Soumekh [79]). Most of these algorithms work well in some situations and poorly in others. However, they lack the ability to characterize their uncertainty (e.g., through estimation of the posterior distribution or belief state) that is required for adaptive sensing or sensor management.

This work combines our understanding of the physical, kinematic, and statistical properties of SAR imagery into a single unified Bayesian structure that simultaneously (a) estimates the nuisance parameters such as clutter distributions and antenna miscalibrations and (b) extracts a sparse component containing the target signatures required for detection and estimation of the target state. The proposed algorithm requires few tuning parameters since most quantities of interest are inferred directly from the data - this allows the algorithm to be robust to a large collection of operating conditions. The performance of the proposed approach is analyzed over both simulated and measured datasets, demonstrating competing or better performance than state-of-the-art algorithms.

One key feature of the proposed inference algorithm is its ability to easily incorporate additional prior information without greatly increasing the computational cost. For example, if the target is known to move smoothly through the scene, a Markov property can be enforced on the spatial locations of the targets within the scene. Moreover, if the target state is known with uncertainty (i.e., in a tracking scenario or in cases where the target may exhibit 'normal' behavior such as near an intersection), then this work also provides methods for (a) predicting likely locations of the target signatures and (b) using this information directly in the inference process.

The last contribution to this area is the development of performance prediction methods for detection and estimation in SAR imagery. The following are provided in this work: (a) a likelihood ratio statistic for detection in the multiple-pass, multiple-antenna SAR image model that is shown to have a well-known form from which exact hypothesis tests can be derived; and (b) a Cramer Rao Lower Bound for estimation error for position and velocity of moving targets.

1.4 Literature review

In this section, we provide a literature review of related work. The chapters related to resource allocation and sensor management (Chapters II, III and IV) draw on similar research topics, so they are grouped together in Section 1.4.1. Section 1.4.2 provides both a simple overview of SAR systems as well as related work in detection/tracking of targets using SAR imagery.

1.4.1 Adaptive sensing/sensor management under resource constraints

The work laid out in Chapters II, III and IV draws on research from many related fields, including sensor management, adaptive sampling, sampling in sparse scenarios, and dynamic programming. In sensor management, one considers how to best utilize a sensor in order to maximize performance criteria. Adaptive sampling involves estimating an underlying signal in noise by choosing where to sample the signal based on previous observations. Sparse approximation and compressed sensing look at the problem of learning the sparse support of a signal in noise by designing an intelligent sampling scheme. Dynamic programming considers the problem of choosing a policy over multiple stages that maximizes utility as a function of the (partially-observable) belief state. Finally, dynamic scheduling looks at the problem of optimally assigning multiple servers to process multiple (infinite-length) queues.

Sensor Management

Sensor management is a rich field composed of many well-studied problems and applications. Those readers interested in a detailed exposition should peruse the work by Hero in [52]. This work is primarily interested in the problem of deciding where to point and how to utilize a sensor in order to minimize some associated cost. Kastella [56] considers the problem of selecting where to a point a sensor among S radar cells in order to detect a signal target in noise. He shows that using 'discrimination gain', a quantity based on the Kullback-Leibler (KL) divergence, to select the location of the next sample can decrease the overall probability of incorrectly detecting the location of the target. Kreucher et. al. [59, 60] show that integrating sensor management with target tracking via the joint multi-target probability density (JMPD) can dramatically improve sensor efficiency for tracking multiple targets.

Similar to Kastella's work [56], they use KL divergence to select the sensing modality with the highest predicted information gain among a discrete set of choices.

Krishnamurthy [61, 62] studies variants of the multi-armed bandit (MAB) problem. In [62], he considers the problem of selecting where to point an agile sensor in order to track P targets among a finite number of cells. When the state is fully observable, the problem can be posed as Markov decision process (MDP) with well-known solutions. Krishnamurthy formulates the problem as a hidden Markov model (HMM) tracking problem in the more practical case, when the state is observed with noisy measurements. Under certain assumptions of the dynamics of the system, Krishnamurthy shows that the optimal solution can be decoupled into P independent optimizations; each of these can be solved by minimizing the 'Gittins index,' which is in turn a function only of each individual target and its associated 'information state' - the conditional density of the state given the observation history. Moreover, a suboptimal approach to estimating the Gittins index is provided to combat the prohibitive computational complexity of the optimal solution. [61] considers the related problem of tracking a single target by choosing among multiple sensors. Once again, an optimal approach is provided along with a suboptimal (yet computationally feasible) alternative.

The methods developed in this work adopt a Bayesian framework and optimize the sensing allocation as a function of posterior probabilities in place of KL divergences or Gittins indices. Moreover, our methods differ in that we choose to select the sensing modes from a continuous spectrum rather than from a discrete set of choices.

Adaptive Sampling

Adaptive sampling has been studied in many different contexts, often appearing in the literature as active learning or active sampling. Castro, Willet, and Nowak [24, 25, 91] consider the problem of estimating a function using samples that are either chosen statistically independent of measurements (i.e., 'passive sampling') or as a function of previous sample points and samples (i.e., 'active sampling'). They develop fundamental limits based on minimax lower bounds, showing that for certain classes of signals, one can achieve nearly optimal convergence rates in terms of estimation mean square error (MSE). Moreover, it is shown that for spatially homogeneous signals, active sampling has no advantage over passive sampling. In addition to performance limits, [25] provides a multiple-stage algorithm that samples the signal uniformly at the first stage, and subsequently focuses samples to the boundaries of the function. This algorithm is applied in a variety of ways to reconstruct spatially inhomogeneous signals, including estimating a Holder smooth boundary of a $(d-1)$-size manifold embedded in a d-dimensional space [24] and estimating boundaries using wireless sensor networks [91]. The work in this thesis differs from active learning in multiple ways: (1) the signals that we consider are not restricted to a class of inhomogeneous signals, (2) we expolit the sparsity of the ROI explicitly in determining sensor allocations, and (3) active learning assumes identical sampling procedure for all samples (leading to similar noise variance), while our work considers separate sampling procedures across stages and locations.

Rangarajan et. al. [76–78] considers adaptive waveform design for estimating a parameter vector under average energy constraints. They provide an solution to the N-step problem that is optimal for $N = 2$ in terms of minimizing MSE. However, since the parameter vector is not assumed to be sparse, only minimal gains are

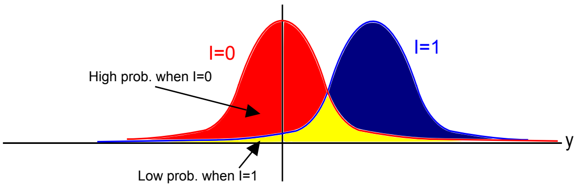


Figure 1.2: This plot shows the unequal distribution of measurements that is exploited by algorithms such as distilled sensing. The posterior probability of a target being present ($I = 1$) given a negative measurement is much smaller than the posterior probability when the target is missing ($I = 0$).

possible. In our work, it is shown that the asymptotic gains in MSE over non-adaptive approaches is inversely proportional to the sparsity of the signal.

Bashan et. al. [9, 11] developed a two-stage policy, namely ARAP, for simultaneously localizing and estimating a sparse ROI within a larger signal under fixed resource constraints. ARAP was shown to be asymptotically optimal in terms of a cost function that is a surrogate for MSE and probability of error. The framework and problem formulation for ARAP are provided in Chapter II, as they form a basis for extensions discussed in this thesis, including multiple-scale modifications in Chapter II and dynamic targets in Chapter III.

Haupt et. al. [48] provide the 'distilled sensing' adaptive sampling procedure that is formulated as a general sequential multiple hypothesis testing approach that simultaneously seeks to localize the target and to test for presence of targets in the scene under a fixed energy constraint. Like ARAP, distilled sensing performs coordinate-wise allocations of the sensing resources to each locations. At each stage, distilled sensing refines its estimate of the ROI by thresholding measurements at each stage. In particular, the method exploits the inequality between the posterior distributions of the measurements under the hypotheses that a target exists or doesn't exist at a location (see Figure 1.4.1). Taken over sequential measurements, distilled sensing is able to provide aysmptotic guarantees on perfect recovery of the ROI for arbitrarily small values of the false discovery rate. While they have similarities, ARAP and the methods developed in my thesis differ from distilled sensing in important ways: (1) ARAP adopts a Bayesian framework that generates a posterior probability of target presence at each location given the measurements; (2) the optimization procedure used by ARAP depends on these posterior probabilities; and (3), ARAP optimization is simply performed on a surrogate convex performance metric.

Sampling Sparse Signals

In recent years, there has been a great deal of work in reconstructing the sparse support of a signal, β, by intelligently choosing the measurement matrix, $\mathbf{X}$ given noisy measurements of the form:

$$y = \mathbf{X}\beta + n, \tag{1.1}$$

for a $n \times p$ matrix, $\mathbf{X}$, and $n \ll p$. Often, the vector β is either exactly k-sparse (i.e., only k non-zero entries) or approximately k-sparse (i.e., only k high amplitude elements). In sparse approximation, the goal is to recover a k-sparse vector, $\hat{\beta}$, so that the residual errors have the relationship:

$$\left\|\mathbf{X}\hat{\beta} - y\right\|_2 \approx \left\|\mathbf{X}\hat{\beta}_k - y\right\|_2, \tag{1.2}$$

where β_k is the best k-sparse approximation to β. Note that in this formulation, $\hat{\beta}$ can be thought of as a linear combination of a sparse number of columns of the sensing matrix, $\mathbf{X}$. In fact, one might consider solving the optimization problem:

$$\min_{\beta} \|y - \mathbf{X}\beta\|_2 \text{ s.t. } \|\beta\|_0 \leq k \tag{1.3}$$

However, this problem is NP-hard to solve due to the l_0 constraints on β. Davis and Mallat [30] present a greedy algorithm called orthogonal matching pursuit (OMP) for selecting the k dictionary elements (i.e., a column of $\mathbf{X}$) that best approximate β_k as a function of the residual error in Equation (1.2). OMP is fast and simple to implement, but does not necessarily identify the correct sparse solution. On the other hand, if the dictionary is sufficiently 'incoherent' (i.e., the maximum inner product between columns is small), then Tropp [86] shows OMP can recover k-sparse with high probability. Moreover, Tropp and Gilbert [88] improve these results and show the surprising results that one can recover the k-sparse signal β using only $O(k \log p)$ measurements. As an alternative to OMP for solving the NP-hard problem in Equation (1.3), Chen, Donoho and Saunders [26] propose 'basis pursuit' that solves a convex relaxation to Equation (1.3) and replaces the l_0 norm with a l_1 norm. Tropp [87] also studies this same convex relaxation, while Gorodnitsky and Rao [45] provide the FOCUSS algorithm, which replaces the l_0 norm with an l_p norm for $0 < p < 1$. Since the latter relaxation is non-convex, they present an iterative algorithm for solving the optimization problem. Aharon, Elad, and Bruckstein [4] provide a general algorithm for adapting the dictionary $\mathbf{X}$ to a given training set, which is adaptable to many of the discussed pursuit algorithms including OMP, basis pursuit and FOCUSS.

Compressed sensing (CS) looks at a very similar problem to sparse approximation, although performance of CS is often characterized by constraining the errors of the approximations themselves:

$$\left\|\hat{\beta} - \beta\right\|_p \approx \|\beta - \beta_k\|_q \tag{1.4}$$

where p, q aren't necessarily equal to 2 as in Equation (1.2). Donoho [34] provides conditions on the sensing matrix $\mathbf{X}$ and shows that it is possible to reconstruct $\hat{\beta}$ reliably using only $O(k \log p)$ measurements in the noiseless situation. Moreover, he shows that basis pursuit is a nearly optimal algorithm for reconstructing β in terms of MSE. Candes and Tao [20] provide additional properties on the sensing matrix, such as the exact reconstruction property (ERP) and the uniform uncertainty principle (UUP), that guarantee the ability to reconstruct the k-sparse signal from a small number of measurements. Baraniuk et. al. [7] prove the existence of these types of matrices. When measurements are corrupted by noise, Candes and Tao [21] provide the Dantzig selector that can reliably reconstruct sparse vectors as long as the sensing matrix is UUP. Moreover, the estimated $\hat{\beta}$ is shown to be within a logarithmic factor of the oracle estimator in terms of MSE. Haupt and Nowak [49] provide a similar result where their sensing matrix is composed of random projections.

Many applications employ sparse approximation and compressed sensing techniques in order to efficiently recover sparse signals. These include medical imaging [64] by Lustig et. al., privacy [94], source localization [65] by Malioutov et. al., and compressive radars [8] by Baraniuk and Steeghs and [74] by Potter et. al.. For an extensive listing of papers related to CS and sparse approximation, the interested reader should peruse the papers listed at http://dsp.rice.edu/cs.

There are many connections between compressed sensing, sparse approximation, and adaptive sampling. Indeed, Castro, Willet, and Nowak [23] show that for certain classes of signals, (CS) performs almost as well as adaptive sampling. Ji, Xue, and Carin [55] present compressed sensing in a Bayesian framework that allows them to create error bars on the uncertainty of measurements. Under this framework, they select the random projections that maximize expected variance (similar to selection based on discrimination gain as discussed by Kastella et. al in [56]). Haupt et. al. [50] extend distilled sensing to highly undersampled regimes (i.e., $n \ll p$) by creating a two-step procedure at each stage composed of (1) compressed sensing measurements followed by (2) refinement of the ROI. They show that by focusing measurements into the estimated ROI, the effective SNR is greatly enhanced, leading to significantly improved error bounds as compared to the Dantzig selector while still maintaining $O(k \log p)$ measurements per stage.

The notion that one can save measurements when sampling sparse signals is also studied in the adaptive sampling literature by using a multiple scale search over sequential stages. Abdel-Samad and Tewfik [1–3] propose an adaptive sampling solution for allocating N measurements to find a single target hidden in hidden in Q cells, specifically in the case when $N < Q$. A hierarchical approach recursively groups the Q cells into $q < Q$ groups in a tree like structure, under the assumption that signal to noise ratio (SNR) decreases as the group size increases. Their multiple hypothesis testing approach is computationally intense and does not scale easily to large N and Q. The proposed search strategies in this work, on the other hand, are explicitly designed to detect and localize multiple targets even when Q is high, and they have lower solution complexity than the multi-hypothesis testing approach in [1–3].

One of the first multi-scale approaches was the adaptive pooled blood sample algorithm introduced in the early 1940's. Dorfman [35] considered the problem of detecting defective members of a large population in the context of weeding out all syphilitic men called up for military service. The test was so sensitive and accurate that Dorfman suggests the following procedure: (1) draw blood from each candidate, (2) use half of each sample to create a pool containing a mixture of n individual subjects, (3) test the pool. If a pool tested positive, the other half sample of each pool member was individually tested to detect the defective member. In the case of low disease prevalence rates, Dorfman showed that one can save a great amount of time by averaging (pooling) measurements at a first coarse scale. Dorfman procedures use a binary model (B-model) and do not account for false alarms or missed detections, which in our setting is equivalent to an infinite SNR. Therefore they do not require the additional degree of freedom of resource allocation and are only concerned with minimizing the total number of samples required. An optimal group size, n, can be analytically evaluated for each disease prevalence rate by optimizing the ratio of the expected number of tests using the Dorfman procedure and prevalence rate. Dorfman procedures enjoyed great success due to their simplicity and effectiveness. Pfeifer modifies the binomial model (M-model) and considers test values that were either zero (for negative) or greater than zero (for the degree of contamination) [72]. This way, when a pool is tested positive, each sample of a subgroup from the pool reveals information regarding the other pool members; thus even greater savings are achieved. Nonetheless, the modified model still does not account for false alarms or missed detections. Dorfman procedures do not account for noise, which arises in most signal processing applications, and this constitutes a major difference.

Dynamic Programming

In general, sequential resource allocation problems can be solved by dynamic programming (DP). Bertsekas [14] discusses dynamic programming in great detail, but the topics of greatest interest to this research are examined in Chapters 1 (introduction), 5 (imperfect state information) and 6 (approximate methods). The basic problem in DP is to minimize an additive cost of the form

$$E\left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right\}, \tag{1.5}$$

where x_k is the current state at stage k, u_k is the input at stage k, w_k is random noise, and $g_k(\cdot)$ is some cost function. In general, $\{u_k\}_k$ will be dependent on the random states x_k, so DP tends to minimize the expected cost over policies, $\mu_k(x_k)$. A key principle in DP, referred to as the 'principle of optimality' by Bertsekas, states that these complex optimizations can be solved by using backwards induction.

In practice, the state is usually observed through noisy observations, which greatly complicates the optimization problem. On the other hand, if the state is replaced by the so-called 'belief state' that characterizes the posterior distribution of the observed state, then the partially observable problem can be converted back into a fully observable DP. Nevertheless, optimal solutions to DP problems often require approximate solutions in order to be computationally tractable, especially as the size of the state grows. For an in-depth discussion of approximate methods, the interested reader should see Chapter 6 of [14].

Chong et. al. [28] describe a set of related problems for formulating adaptive sensing as a partially observable Markov decision problem (POMDP). They provide an excellent survey of optimal and approximate methods for solving POMDPs. Morevoer, they show that using non-myopic policies (i.e., policies that trade off current benefits for long-term performance gains) can lead to significant gains over standard policies. These strategies rely on approximating the so-called Q- function of the belief state $\boldsymbol{B}(t)$ and possible actions $a \in \mathcal{A}$, where

$$Q_{T-t}(\boldsymbol{B}(t), a) = r(\boldsymbol{B}(t), a) + E[V^*_{T-t-1}(\boldsymbol{B}(t+1))|\boldsymbol{B}(t), a] \tag{1.6}$$

and

$$V^*_T(\boldsymbol{B}(0)) = \max_a \big(r(\boldsymbol{B}(t), a) + E[V^*_{T-1}(\boldsymbol{B}(1))|\boldsymbol{B}(0), a]\big) \tag{1.7}$$

In general, estimating the Q-function is intractable. However, there are many approximate methods, including

- Q-learning: A reinforcement learning approach based on estimating the Q-

function from multiple trajectories of the random process.

- Relaxation of the optimization problem or of the state space.
- Parametric approximations.
- Rollout policies: Assume that at stage $\tau > t$, we use a known base policy, $\lambda^{(base)}$. Then choose the action at stage t that optimizes the Q-function where the remaining $T - t$ stages are chosen according to $\lambda^{(base)}$.
- Parallel rollout polices: A simple extension of rollout policies, where there are multiple base policies from which to choose.
- Completely observable rollout policies: An approximation to rollout policies, that maps states to actions rather than belief states to actions.

The work in this thesis considers problems similar to both POMDPs and DPs. We provide a computationally simple alternative to the optimal and approximate solutions, which is additionally shown to be asymptotically optimal through empirical evidence at a fraction of the search complexity of other algorithms.

Concurrent work

Concurrent with the work provided in this thesis, Wei and Hero [90] provide an approximation to the DP solution in the static case by using the so-called open-loop feedback control (OLFC) policy which assumes that at time t, future observations $\boldsymbol{y}(t+1)$ to $\boldsymbol{y}(T)$ are unavailable. Under these conditions, the cost function at time t becomes

$$\begin{aligned} \min_{\boldsymbol{\lambda}(t),\boldsymbol{\lambda}(t+1),\ldots,\boldsymbol{\lambda}(T)} \quad & \sum_{i=1}^{Q} \frac{\Pr(I_i = 1|\boldsymbol{Y}(t-1))}{\sigma^2/\sigma_i^2(t) + \sum_{\tau=t}^{T} \lambda_i(\tau)} \\ \text{s.t.} \quad & \sum_{i=1}^{Q} \sum_{\tau=t}^{T} \lambda_i(t) = \Lambda(t) \end{aligned} \tag{1.8}$$

This optimization problem has a closed form solution for the variable $\bar{\lambda}_i(t) = \sum_{\tau=t}^{T} \lambda_i(\tau)$ similar to that given in equation (3.76). In order to derive allocations per stage, Wei and Hero propose that

$$\boldsymbol{\lambda}(t) = R^{(T)}(t)\bar{\boldsymbol{\lambda}}^{*}(t) \tag{1.9}$$

where $R^{(T)} \in [0,1]$ is the fraction of the remaining budget used at time t. Then, they use the strategy that sets $R^{(T)}(t) = R^{(T-1)}(t-1)$ for $t = 1, 2, \ldots, T$ and optimizes over $R^{(T)}(0)$. This nested optimization problem can be done offline using Monte Carlo simulations of the T-stage policy. Furthermore, it is shown that the nested optimization guarantees that their T-stage policy is at least as good as a $T-1$-stage policy. The work in this thesis extends the problem formulation to dynamic targets, though the optimization techniques in this thesis are similar to those derived in their work.

Dynamic scheduling/queuing

Dynamic scheduling/queuing has been studied in the framework of scheduling multiple servers to multiple (infinite-length) queues by Tassiulas and Ephremides [84], Wasserman et. al [89], Michailidis [67], Armony and Barbos [5], Eryilmaz and Srikant [40] and Brémaud [18]. Applications of this work include developing stable policies for time-varying wireless networks and congestion control by Neely et. al. [68], Eryilmaz and Srikant [40] and Stolyar [83]. The general framework involves time-varying decisions for choosing which queue will be processed by each server in order to maximize throughput of the system and to maintain stability; stability is defined to guarantee that the service load rates remain bounded (i.e., the queue lengths do not grow unbounded). Generally, solving for an optimal allocation of the servers is a difficult, if not intractable, problem. However, several suboptimal approaches have been proposed. This work does not propose a new approach to the dynamic scheduling research, but rather uses an existing policy as a basis for providing performance limits of a system using a reasonable scheduling policy.

1.4.2 Detection and tracking with SAR imagery

SAR images play a central role in the second direction of this thesis, and this literature review begins with a basic overview of SAR and its related phenomenology, based on the lecture notes by Cornwall [29]. The ultimate goal is to provide adaptive techniques for detection and tracking of targets using SAR images, and so related work in change detection, clutter suppression, ground moving target indication (GMTI), and Bayesian filtering techniques are discussed as well. Change detection and clutter suppression are both based on the assumption that the image background lies in a low-dimensional subspace that is not of interest to the detection/tracking goal. GMTI algorithms tend to involve multiple-stage processes that attempt to localize targets and estimate their dynamic state (velocities, accelerations, etc.). Bayesian filtering has its roots in common algorithms such as the Kalman filter, but also incorporates many algorithms that estimate the posterior distribution of the state given noisy measurements, including particle filters, Gibbs samplers, and Hierarchical Bayesian models. Finally, the inference procedures proposed in this work are closely related to the problem of decomposing a signal into low-dimensional and sparse components, which has been well-studied in the context of the so-called robust principal component analysis (RPCA) problem.

Basics of SAR (based on notes by Cornwall [29])

Conventional radar systems emit coherent electromagnetic waveforms that hit a target and reflect back toward the radar. Measuring the time it takes for the wave

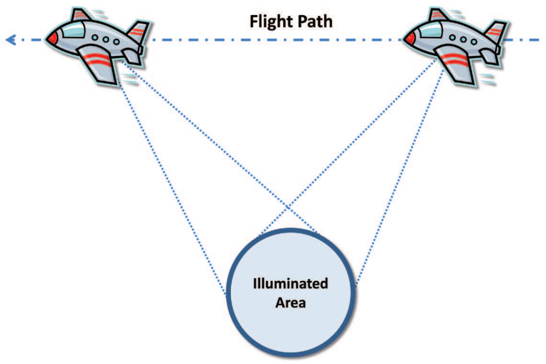


Figure 1.3: This plot shows the flight path and beam steering used in a spotlight SAR system.

to propagate from the antenna to the target (and back again) provides an estimate of the range to the target. The angular resolution of the radar is determined by the antenna beamwidth, which in turn is proportional to wavelength. Since typical radar wavelengths are orders of magnitude larger than the wavelength of light, conventional radars are not normally used to create images of the scene.

When the target is moving relative to the radar (which occurs either when the platform moves, as in airborne radar, or when the target moves), then the Doppler shift occurs, causing the returned wave to have a different frequency than the emitted one.

By using a two-dimensional FFT, one can produce an image with axes that correspond to 'range' and 'Doppler'. It should be noted that this method is an approximation to error-free SAR imaging in the xy (spatial) domain , which is discussed in more detail later in this section.

The range resolution is a function of the bandwidth of the emitted signal[1]. The

[1] A technical detail that we will overlook is that SAR signals usually emit a so-called 'chirp' signal, which causes rapid frequency changes in the signal, similar to a bird's chirp. Chirp signals (also sometimes referred to as pulse compression) allow large bandwidths needed for small range resolution without requiring excessive peak power.

second dimension of SAR images (i.e., the ‘azimith resolution’) is roughly determined by the angular resolution of the radar. To be able to get a azimuth resolution of a few cm with an X-band SAR, one would need an aperture on the order of 300 km when the radar is 1 km from the scene center. To overcome the intractability of constructing an aperture of this size, SAR constructs a ‘synthetic aperture’ by combining pulses from an antenna that moves over a large distance (such as 300 km). Often, we are interested in imaging the scene for an extended time. In this case, the antenna beam has to be steered in such a way that it illuminates the same region of interest along its flight path. This mode of operation is called ‘spotlight SAR’, and we will focus on this mode for this thesis. Figure 1.3 illustrates the flight path and beam steering used in a spotlight SAR system.

Using SAR for imaging has many practical benefits over using optical imaging. First, SAR images can be taken at day or night or through adverse weather that would block optical wavelengths. Second, SAR records phase information about the scene in addition to intensity. The phase information can be used to form interferometric images by combining SAR images formed in slightly different ways (for instance, using different locations, times, or polarizations). These images enable the measurement of very slow large scale motion (such as glacial movement), the creation of 3-dimensional images, and detection of moving targets.

The terminology used in SAR signal processing distinguishes between the returns for an individual radar pulse (fast-time) and a collection of pulses (slow-time). Indeed SAR signals are often model simply as

$$s(\tau, f) = e^{-j2\pi f\left[\frac{2R(\tau)}{c}\right]}, \tag{1.10}$$

where τ denotes slow-time, f is the radar frequency, and $R(\tau)$ is the range to the

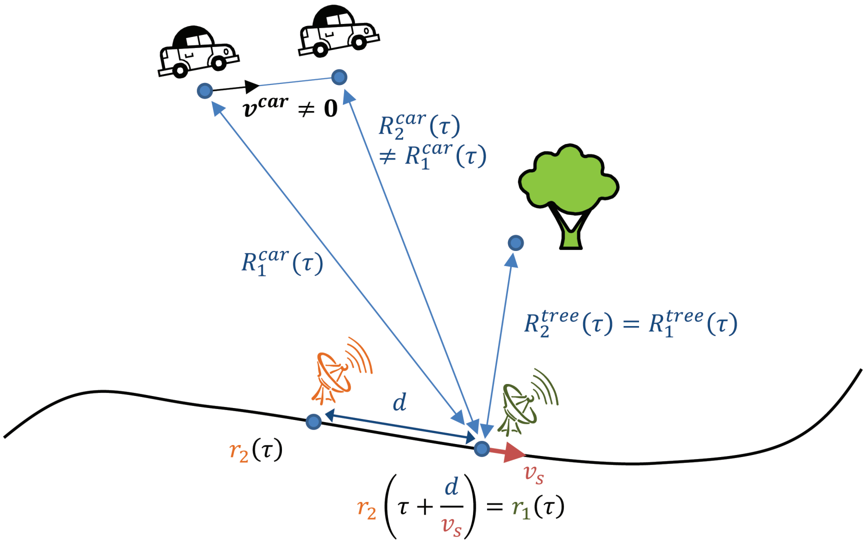


Figure 1.4: This plot shows the geometry of an along track SAR system with two antennas. After a short time lag of $\Delta\tau = d/v_s$, the second antenna occupies the same position as the first antenna. Stationary objects (such as the tree) will yield the same range and thus can be canceled by certain algorithms. On the other hand, moving targets (such as the car) will have slightly different ranges and will not be canceled.

object on the ground. In this work, we will consider along-track radar configurations, consisting of a multiple antenna phase centers that are offset in the along-track direction of the radar platform. For simplicity of discussion, suppose that two monostatic radar channels are mounted on an aircraft traveling with velocity, v_s, and separated by distance d. Let $\mathbf{r}_1(\tau)$ and $\mathbf{r}_2(\tau)$ be the locations of Channel 1 and 2 phase centers, respectively. By assumption, both channels travel along the same path, but Channe1 1 lags Channel 2 by $\Delta\tau = d/v_s$, so that $\mathbf{r}_1(\tau) = \mathbf{r}_2(\tau + d/v_s)$. Figure 1.4 illustrates this geometry. Consider a scene composed of a single unit-amplitude scatterer. From (1.10), the radar signals for both channels can be written as

$$s_1(\tau, f) = e^{-2\pi f[2R_1(\tau)/c]}, \tag{1.11}$$

$$s_2(\tau, f) = e^{-2\pi f[2R_2(\tau)/c]}, \tag{1.12}$$

where $R_i(\tau)$ is the range from Channel i to the target. Under certain assumptions, such as using sufficiently small integration times and far-field approximations, Deming [31] shows that the signal from Channel 2 can be written as

$$s_2(\tau + \frac{d}{v_s}, f) = s_1(\tau, f)e^{-j2\pi\left(\frac{2dv_r}{\lambda_0 v_s}\right)}, \tag{1.13}$$

where λ_0 is the center radar wavelength, and v_r is the range velocity of the target. Let $\mathbf{A}\left\{\cdot\right\}$ be a linear SAR imaging operator that maps the $s(\tau, f)$ to a complex-valued image $S(x, y)$. From (1.13), we find that

$$\mathbf{A}\left\{s_2(\tau + \frac{d}{v_s}, f)\right\} = \mathbf{A}\left\{s_1(\tau, f)\right\} e^{-j2\pi\left(\frac{2dv_r}{\lambda_0 v_s}\right)}. \tag{1.14}$$

This relationship is integral to two common algorithms for removing stationary background objects (aka, clutter) from moving targets. In displaced phase center array

(DPCA) processing, an image is created in the following manner:

$$\begin{aligned} S_{DPCA}(x,y) &= \mathbf{A}\left\{s_1(\tau,f)\right\} - \mathbf{A}\left\{s_2(\tau + d/v_s, f)\right\} \\ &= \mathbf{A}\left\{s_1(\tau,f)\right\}\left(1 - e^{-j2\pi\left(\frac{2dv_r}{\lambda_0 v_s}\right)}\right) \\ &= \mathbf{A}\left\{s_1(\tau,f)\right\}\alpha(v_r), \end{aligned} \tag{1.15}$$

where $\alpha(v_r) = 1-e^{-j2\pi\left(\frac{2dv_r}{\lambda_0 v_s}\right)}$. For stationary targets, $|\alpha(v_r)|$=0, so that $|S_{DPCA}(x,y)| = 0$. For most other targets, $|\alpha(v_r)| > 0$ (except for some periodicities). Thus, a detection algorithm can be simply derived by thresholding the intensity image, $|S_{DPCA}(x,y)|$.

In along-track interferometry (ATI), an image is created from the product $S_1 S_2^*$:

$$\begin{aligned} S_{ATI}(x,y) &= \mathbf{A}\left\{s_1(\tau,f)\right\}\left[\mathbf{A}\left\{s_2(\tau + d/v_s, f)\right\}\right]^* \\ &= \left|\mathbf{A}\left\{s_1(\tau,f)\right\}\right|^2 e^{j2\pi\left(\frac{2dv_r}{\lambda_0 v_s}\right)}. \end{aligned} \tag{1.16}$$

where the phase of $S_{ATI}(x,y)$ is used as the test statistic for change detection. It should be noted that both DPCA and ATI are based on idealized SAR image models and depend on properly balancing the channels, which can be difficult in practice. Moreover, modern SAR systems often have more than two channels whereas these methods are designed only for two-channel systems.

SAR Image Formation

Image formation from SAR phase histories can be done in a multitude of ways. The simplest algorithm is ‘Doppler processing’, where the data $s(\tau, f)$ undergoes a two-dimensional Fourier transform, yielding a coarse estimate of an image in Doppler and range dimensions. Unfortunately, Doppler processing creates unfocused images for larger integration lengths that lead to wider apertures (and better cross-range resolution). To avoid this tradeoff, focused SAR images usually are created by using more intricate error-free techniques, such as the ‘Polar Format’ algorithm or ‘Convolution Backprojection’. Details of these algorithms will not be provided here, but the interested reader should peruse books on the subject by Jakowatz [53], Carrara [22], and/or Soumekh [82].

For this work, SAR images are formed using the MATLAB toolbox provided by Gorham et al. [44], which provides simple implementations of many common SAR image formation algorithms.

SAR Phenomenology

SAR images often exhibit behavior much more complicated than the ideal point scatterer model discussed previously. Of particular interest to this work are two noise sources attributed to stationary targets. Borden shows that angular scintillation (aka ‘glint’ or ‘specular noise’) can cause large phase errors in the image formation process that may considerably degrade signal quality [16]. Glints, as the name suggests, can be shown to have a large angular dependence, in the sense that the intensity of the glint is only large from (few) azimuth angles. Moreover, this noise source tends to occur on man-made structures, such as edges of buildings or electrical wires. Since buildings tend to be aligned in common directions, there tends to be strong spatial correlation and angular dependence for glints.

Speckle noise as described by Raney [75] and Posner [73] is an additional multiplicative noise source arising from coherent imaging in SAR. Speckle noise tends to be spatially correlated (e.g., buildings versus vegetation). Unlike glints, the intensity of speckle noise tends to be uniform as a function of the azimuth angle.

In general, SAR images are formed by focusing the response of stationary objects to a single spatial location. Moving targets, however, will cause phase errors in the

SAR data that cause displacement and defocusing effects in the SAR image. Most methods designed to detect the moving target depend on either (a) exploiting the phase errors induced by the SAR image formation process for a single phase center system or (b) canceling the clutter background using a multiple phase center system. In this work, we provide a rich model that can combine (and exploit) both sources of information in order to improve on both methodologies.

Fienup [42] provides an analysis of SAR phase errors induced by translational motions for single-look SAR imagery. He shows that the major concerns are (a) azimuth translation errors from range-velocities, (b) azimuth smearing errors due to accelerations in range, and (c) azimuth smearing due to velocities in azimuth. Fienup also provides an algorithm for detecting targets by their induced phase errors. The algorithm is based on estimating the moving target's phase error, applying a focusing filter, and evaluating the sharpness ratio as a detection statistic. Jao [54] shows that given both the radar trajectory and the target trajectory, it is possible to geometrically determine the location of the target signature in a reconstructed SAR image. Although the radar trajectory is usually known with some accuracy, the target trajectory is unknown. On the other hand, if the target is assumed to have no accelerations, Jao provides an efficient FFT-based method for refocusing a SAR image over a selection of range velocities. Khwaja and Ma [58] provide an algorithm to exploit the sparsity of moving targets within SAR imagery; they propose a basis that is constructed from trajectories formed from all possible combinations of a set of velocities and positions. To combat the computational complexity of searching through this dictionary, the authors use compressed sensing techniques. Instead of searching over a dictionary of velocities, our work proposes to use a prior distribution on the target trajectory that can be provided a priori through road and traffic models or adaptively from previous observations of the scene.

Change Detection, Clutter Suppression, and Ground Moving Target Indication Techniques

Earlier in this chapter, DPCA and ATI were algorithms presented to detect moving targets from SAR images when multiple along-track channels were available. The process of removing the stationary background in order to detect moving targets is also known in the literature as ‘clutter suppresion’ or ’clutter cancellation’. ’Change detection’ is an algorithm that detects changes between multiple observed images of the scene. Often, clutter cancellation is a step within change detection. Moreover, change detection is sometimes used to detect changes that happen over large time differences (ex: detecting stationary tank formations). In this work, we refer to change detection as the process of determining *moving* objects between multiple images, though this is clearly a subset of all change detection algorithms in the literature.

Erten [39] provides a statistical analysis for SAR moving target change detection with multiple channels. The authors provide an algorithm based on mutual information and complex Gaussian distributed channels that outperforms classical correlation-based change detection algorithms. Gierull [43] provides a statistical analysis of the phase and magnitude of complex SAR images for two channels. He shows that SAR images cannot be modeled as spatially-invariant Gaussian in many cases of interest, such as in urban environments, where the statistics vary spatially and may be modulated by random variations. Gierull provides probability density functions based on the p-distribution, as well as adaptive techniques for estimating the parameters of this distribution to be used in detecting slow-moving targets.

Ender [38] applies space-time adaptive processing (STAP) to multiple-channel

SAR imagery. Similar to standard moving target change detection algorithms such as displaced phase center array (DPCA) and along-track interferometry (ATI), STAP models the clutter as being embedded in a one-dimensional subspace. However, STAP extends those algorithms to using $N > 2$ channels, where a single channel is used to estimate the stationary background and the remaining $(N-1)$ channels are used to estimate the moving component. However, STAP relies on estimating the complex-valued covariance matrix of the N-channel system, which in turn depends on the availability of homogeneous target-free secondary data.

There are a multitude of algorithms for change detection that are based on multi-temporal SAR images rather than multi-channel data. Bazi and Bruzzone [13] develop methods for multi-temporal change detection that choose adaptive thresholds for declaring changes based on a theoretical analysis of a generalized Gaussian model. Bovolo and Bruzonne [17] provide another algorithm for change detection that employs a wavelet-based multiple scale decomposition of multitemporal SAR images, with an adaptive scale driven fusion algorithm.

Ranney and Soumekh [79, 81] develop methods for change detection from SAR images collected at two distinct times that are robust to errors in the SAR imaging process that may be due to inaccurate position information, varying antenna gains, and autofocus errors. They propose that the stationary components of multi-temporal SAR images can be related by a spatially-varying 2-dimensional filter (for 2D images):

$$S_2(x,y) = \int\int h(u,v;x,y)S_1(x,y)dudv \tag{1.17}$$

To make the change detection algorithm numerically practical, the authors propose that the filter $h(u,v;x,y)$ can be well-approximated to be spatially invariant within small subregions about any pixel (x_i, y_j). In a discrete model, the model in the (k)-th subregion can be represented as

$$S_2^{(k)}(x_i, y_j) = \sum_{m=-n_x}^{n_x} \sum_{n=-n_y}^{n_y} h_{mn}^{(k)} S_1^{(k)}(x_i - m\Delta_x, y_j - n\Delta_y). \tag{1.18}$$

It should be pointed out that the primary error sources in an along-track system (i.e., a single transmit antenna followed by multiple receive antennas placed along the flight path) contribute to phase incoherence rather than amplitude or registration errors. In the simple case where we discard displacement errors (i.e., when $n_x = n_y = 0$), the filter coefficients $h_{0,0}^{(k)}$ can be computed easily through simple least squares. In other cases, Ranney and Soumekh provide a numerically efficient method for estimating the change detection statistic without requiring the actual estimation of the filter coefficients by use of the ‘signal subspace projection’ method that is once again based on the principle of a low-dimensional embedding of the stationary component.

Ground Moving Target Indication (GMTI) methods involve the processing of SAR imagery to detect and estimate moving targets. Often clutter cancellation and change detection play a preprocessing role in these algorithms [46, 47, 71, 95]. Perry et al. [71] provides an algorithm that introduces the keystone formatting to mitigate linear range migration, followed by phase error compensation and change detection. Zhu et al. [95] extends keystone formatting to improve focusing SAR images without a priori knowledge of the target motion parameters. Guo et al. [46, 47] provides multiple algorithms that incorporate channel calibration, clutter estimation, and refinement of target velocity estimates using iterative algorithms.

Robust Principal Component Analysis

Recently, there has been great interest by Wright et al. [93], Lin et al. [63], Candes et al. [19] and Ding et al. [33] in the so-called robust principal component analysis

(RPCA) problem that decomposes high-dimensional signals as

$$\boldsymbol{I} = \boldsymbol{L} + \boldsymbol{S} + \boldsymbol{E}, \tag{1.19}$$

where $\boldsymbol{I} \in \mathbb{R}^{N \times M}$ is an observed high dimensional signal, $\boldsymbol{L} \in \mathbb{R}^{N \times M}$ is a low-rank matrix with rank $r \ll NM$, $\boldsymbol{S} \in \mathbb{R}^{N \times M}$ is a sparse component, and $\boldsymbol{E} \in \mathbb{R}^{N \times M}$ is dense low-amplitude noise. In [19, 63, 93], inference in this model is done by optimizing a cost function of the form

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \gamma \|\boldsymbol{S}\|_1 + (2\mu)^{-1} \|\boldsymbol{I} - \boldsymbol{L} - \boldsymbol{S}\|_F \tag{1.20}$$

where the last term is sometimes replaced by the constraint $\boldsymbol{I} = \boldsymbol{L} + \boldsymbol{S}$. One major drawback of these methods involves finding the algorithm parameters (e.g., tolerance levels or choices of γ, μ), which may depend on the given signal. Moreover, it has been demonstrated that the performance of these algorithms can depend strongly on these parameters.

Ding et al. [33] provide a Bayesian formulation (as discussed below) that simultaneously learns the noise statistics and infers the low-rank and sparse components. Bayesian models can often be generalized to richer models, e.g. Markov dependencies on the target locations. Additionally, these Bayesian inferences provide a characterization of the uncertainty of the outputs through a Markov Chain Monte Carlo (MCMC) estimate of the posterior distribution.

Bayesian Filtering

It should be noted that many of the previously discussed algorithms work well in certain situations, but do not provide estimates of estimation uncertainty. Such estimates are often necessary for adaptive sensing, sensor management, or sensor fusion. The research described in this thesis aims to bridge this gap by providing a Bayesian formulation that provides uncertainty distributions for the presence of the moving targets and their positions. Under this Bayesian formulation, we can generate the posterior distribution of the target state(s) given the observations (i.e., the SAR images).

Bayesian filtering has been studied in depth for many years with roots in Kalman filtering. Both Arulampalam [6] and Doucet [36] give excellent tutorials on filters for Bayesian tracking in a multitude of scenarios, including the Kalman filter (KF), which is optimal for linear, Gaussian systems, as well as suboptimal approaches when the physical system dynamics have nonlinearity or non-Gaussianity. This includes discussion of extended Kalman filters (local linearization of KF equations), approximate grid-based methods (discretization of a continuous state space), particle filters for general systems, also known as sequential importance sampling (SIS), and bootstrap filtering.

The work by Ding et. al [33] is based on a general Bayesian framework [85] by Tipping et al. for obtaining *sparse* solutions to regression and classification problems. Their framework uses simple distributions (e.g., those belonging to the exponential class) that can be described by few parameters, known as hyperparameters. Moreover, they consider a *hierarchy* where the hyperparameters themselves are assumed to have a known 'hyperprior' distribution. Often the prior and hyperprior distributions are chosen to be conjugate, so that inference is simplified. Tipping et al. provide insight into choosing the hyperparameter distributions so as to be non-informative with respect to the prior. Non-informative priors make it possible to implement inference algorithms with few tuning parameters. Tipping relates their method to the 'relevance vector machine' (RVM), which is a Bayesian version of the support vector machine. Wipf et al. [92] provide an interpretation of the RVM as the application of a variational approximation to estimating the true posterior distribution. Wipf et al. explain the sparsity properties of the sparse Bayesian learning algorithms in a rigorous manner. Additionally, they provide connections with other popular work in sparse problems, such as the FOCUSS and basis pursuit algorithms.

CHAPTER II

Development of Resource Allocation Framework

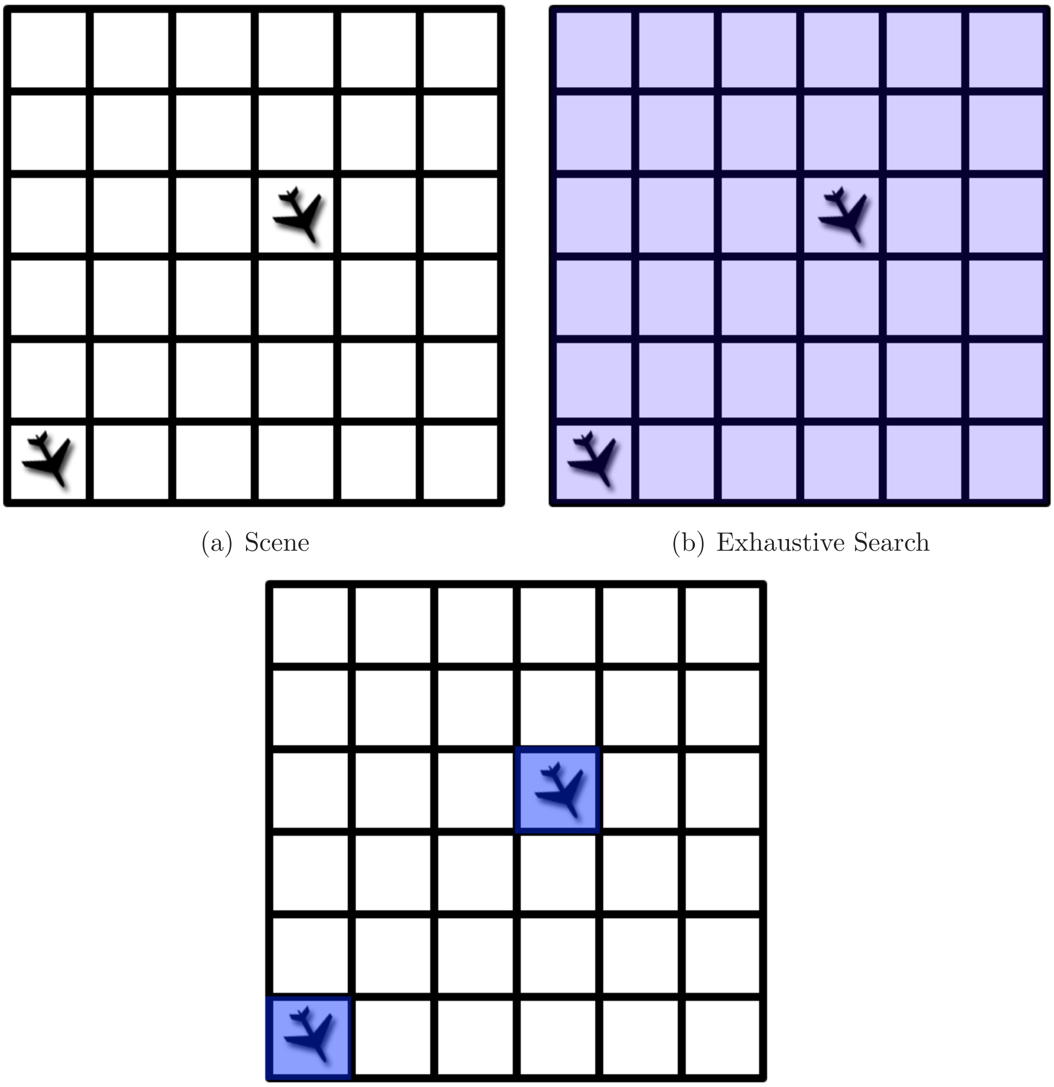
2.1 Introduction

This chapter considers the problem of localizing and estimating targets in noise. We are specifically interested in cases where targets occupy only a small fraction of the scanned domain, which we refer to as the region of interest (ROI). Related problems include detection of tumors in early cancer detection and surveillance systems using agile radars. A framework is provided that leads to descriptions of efficient algorithms for search of sparse targets under resource constraints. The problem formulation is based on the work by Bashan, Raich, and Hero [11] that proposed a two-stage policy for research allocation under total energy constraints. Moreover, this framework acts as a basis for extensions, such as (a) a multiple-scale modification that leads to fewer measurements, and (b) the additional capability to handle moving targets.

To illustrate the situation, consider the problem of estimating the location of a (static) target with a scanning radar (Figure 2.1). The standard policy (Figure 2.1(b)) is to scan all radar cells with an equal effort allocation, which is referred to as an *exhaustive search* in this work. On the other hand, if the locations of the targets were known a priori, then an optimal policy (Figure 2.1(c)) would allocate resources only to the radar cells where targets exist. The optimal policy is impossible to implement, unfortunately, since finding the target locations is precisely the problem being considered. Instead, this chapter focuses on adaptive policies that determine the ROI from measurements collected at multiple stages. At the initial stage, the scene is exhaustively searched with a fraction of the total resource budget. Subsequently, measurements from the first stage are used to estimate the ROI and focus the next stage measurements to areas likely to contain targets. Figure 2.2 illustrates an adaptive allocation effort over multiple stages. In this work, a cost function is introduced and the solution of a related minimization problem yields an asymptotically optimal allocation policy. In order to be computationally tractable, the allocation policy is limited to two stages, though this constraint is relaxed in Chapter III.

In the original formulation, the search at the first stage included measuring each cell individually. When the size of the search space is very large, this may be prohibitive to the implementation of a practical system. Moreover, if the number of targets is much smaller than the size of the scene, then many of these measurements may be unnecessary. This chapter also provides a coarse-to-fine scale modification of the original policy. Measurements are pooled at the first stage and subsequently, the scanned domain is re-sampled on a fine grid. This modification is depicted in Figure 2.3.

The rest of the chapter is organized as follows. Notation is introduced in Section 2.2. In Section 2.3, the search problem is introduced and the cost function is defined. Section 2.4 provides the adaptive research allocation policy (ARAP) and discusses its basic properties. Section 2.5 extends ARAP with a multi-scale modification (M-ARAP) and analyzes its asymptotic performance. The performanc of M-ARAP and ARAP are compared to exhaustive search in Section 2.6. In Section 2.7, M-ARAP

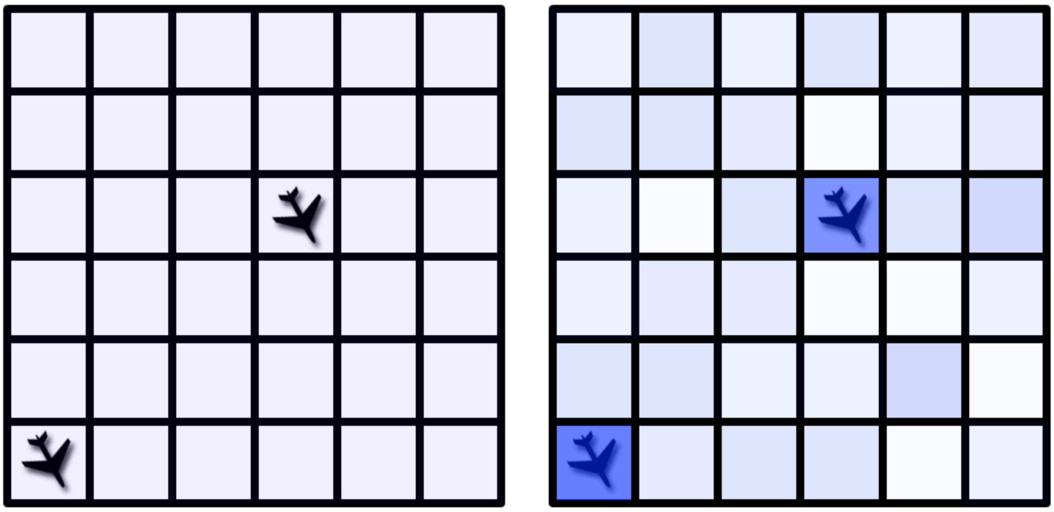


(a) Scene

(b) Exhaustive Search

(c) Optimal Search

Figure 2.1: In (a), a scene that we wish to scan is shown with two static targets. The standard policy, shown in (b) is to allocate equal effort to each cell individually. The optimal policy, shown in (c), is to allocate effort only to cells containing targets.



(a) First stage (b) Second (adaptive) stage

Figure 2.2: This figure depicts an adaptive policy for estimating the ROI over multiple stages. In the first stage, shown in (a), a fraction of the resource budget is applied to all of the cells equally. In the second stage, allocations are refined to reflect the estimated ROI. Note that the second stage allocation is a noisy version of the optimal allocation given in Figure 2.1(c).

is applied to an MTI radar simulation. Finally, a discussion and concluding remarks are provided in Section 2.8.

2.2 Notation

- Q - Number of cells in search space.

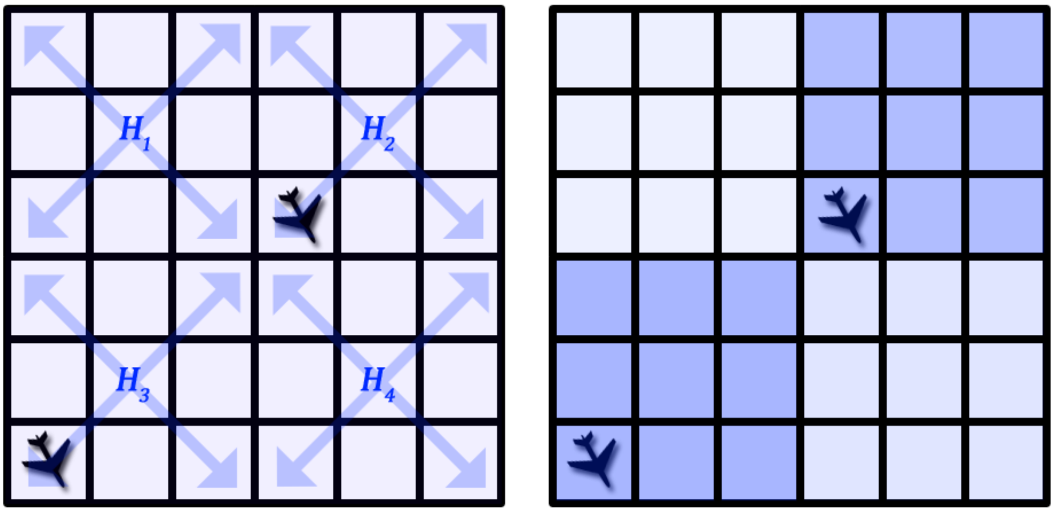
- $\mathcal{X} = \{1, 2, \ldots, Q\}$ - Discrete space of Q cells.

- $\Psi \subseteq \mathcal{X}$ - Subset of $\mathcal{X}$ referred to as the ROI.

- I_i - indicator function of the ROI such that

$$I_i = \begin{cases} 1, & i \in \Psi \\ 0, & \text{Otherwise} \end{cases} \qquad \text{for } i \in \{1, 2, \ldots, Q\}. \tag{2.1}$$

- $\widehat{\Psi} \subseteq \mathcal{X}$ - Estimated ROI, learned from measurements.



(a) First stage (with pooling)

(b) Second (adaptive) stage

Figure 2.3: This figure depicts a multi-scale adaptive policy for estimating the ROI over multiple stages. In the first stage, shown in (a), a fraction of the resource budget is applied to *pooled* measurements . In the second stage, allocations are re-sampled to a fine grid refined to reflect the estimated ROI. Note that although significantly fewer measurements were made at the first step, a significant amount of wasted resources is wasted searching cells within a support region where targets exist. This tradeoff between measurement savings and wasted resources is analyzed later in this chapter.

- $\lambda_i(t) \in [0, \lambda_T]$ - Search effort allocated to cell i at stage t.
- $\lambda_T = \sum\limits_{t=1}^{2} \sum\limits_{i=1}^{Q} \lambda_i(t)$ - Total search effort.
- $\boldsymbol{Y} = \{\boldsymbol{y}(1), \boldsymbol{y}(2)\} \in \mathbf{R}^N$ - Set of measurements.
- $p_{I_j|\boldsymbol{y}(1)}$ - Posterior probability for target existence in the j-th cell.
- ν - A selectable parameter (see Bashan et al. [11] for details).
- w_j - A quantity that is a function of the posterior probabilities, $p_{I_j|\boldsymbol{y}(1)}$, and ν, used in the allocation of resources to cells at stage 2.
- $\tau(\cdot)$ - A sorting operator for w_j's, so that

$$w_{\tau(1)} \leq w_{\tau(2)} \leq \cdots w_{\tau(Q)}.$$

- $u(\cdot)$ - The discrete unit step function.

2.2.1 For extensions to multiple-scales

- L - Scale factor between stages, with L a factor of Q.
- N_t - # of measurements at stage t.
- N - # of total measurements, with $N = \sum_{t=1}^{2} N_t$.
- H_j - Indicator function of the j-th support region.
- $\tilde{y}_j(1)$ - Measurement of the j-th support region for $t = 1$.
- $\tilde{p}_{H_j|\boldsymbol{y}(1)}$ - Posterior probability for target existence in j-th support region.
- $\tilde{\cdot}$ - Tilde notation, used to denote the energy allocation or measurement applied to the support region (as opposed to an individual cell.)

2.3 Problem formulation

Consider a space $\mathcal{X}$ containing Q cells equipped with a probability measure P, and a region of interest (ROI), $\Psi \subseteq \mathcal{X}$. In the sequel, Ψ will be a randomly selected small subset of $\mathcal{X}$. Exact definition of the ROI is application dependent. In radar target localization, the ROI is the collection of all cells containing targets and target related phenomena, e.g., target shadows. In a medical imaging application, such as early detection of breast cancer where tumor boundaries are poorly defined, the ROI may be defined as the collection of all cells containing targets (a tumor) plus some neighboring cells.

Let I_i be an indicator function defined in (2.1) and $\{p_i = \Pr(I_i = 1)\}_{i=1}^{Q}$ be an associated set of prior probabilities. In this chapter, we consider the case of non-informative priors, so that the I_i's are i.i.d. and

$$p_i = p = \mathrm{E}\left[\frac{|\Psi|}{Q}\right] \tag{2.2}$$

for all i. Let $\boldsymbol{I}_\Psi = [I_1, \ldots, I_Q]^T$ be a vector corresponding to the set of all indicators and $(\cdot)^T$ denote the transpose operator. We say that the presence of a target affects cell i if $i \in \Psi$. Targets are assumed to be static, so that I_i is constant over time. We define the random vector of N observations, $\boldsymbol{Y} : \mathcal{X} \to \mathbb{R}^N$ and consider the conditional probability $p(\boldsymbol{Y}|\boldsymbol{I}_\Psi)$.

Consider a sequential experiment where at the first stage we observe $\boldsymbol{y}(1) : \mathcal{X} \to \mathbb{R}^Q$ and at the second stage we observe a selected subset of $\boldsymbol{y}(2) : \mathcal{X} \to \mathbb{R}^Q$, i.e., we observe $y_i(2)$ for $i \in \widehat{\Psi}$, defined in (2.4). This formulation allows us to limit the total number of observations to $N = Q + |\widehat{\Psi}|$, where $|\widehat{\Psi}|$ is the number of elements of $\widehat{\Psi}$. Let $\lambda_i(t) \geq 0$ denote the search effort allocated to cell i at time t, under total energy constraint λ_T, so that

$$\lambda_T = \sum_{t=1}^{2} \sum_{i=1}^{Q} \lambda_i(t) \tag{2.3}$$

Let $\{\lambda_i(2)\}_{i=1}^{Q}$ be a mapping from past observations $\boldsymbol{y}(1)$ to the probability simplex that we call an effort allocation policy. The set $\widehat{\Psi}$ is formally defined as

$$\widehat{\Psi} = \{i \in \mathcal{X} : \lambda_i(2) > 0\}. \tag{2.4}$$

Without prior knowledge of Ψ, it is logical to let $\lambda_i(1) = \lambda_1$ for all i. Then, the combination of λ_1, $\{\lambda_i(2)\}_i$ and $\widehat{\Psi}$ is termed a search policy. We focus here on deterministic mappings λ, although a more general random mapping could also be incorporated into our framework. We assume that a sample's 'quality' is an increasing function of the allocated effort to the associated cell, e.g. measured in terms of information or inverse variance, called a precision parameter by Haupt et al. [48]. In general, effort might be computing power, complexity, cost, or energy that is allocated to acquiring a particular cell location.

Let $\boldsymbol{\Theta} = [\Theta_1, \ldots, \Theta_Q]^T$ be a random vector where $\Theta_i \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$ are i.i.d. random variables (r.v.) corresponding to the target amplitudes. Let $\boldsymbol{I}_\Psi$ be a vector of indicators marking whether or not cell i contains a target. Consider the following measurement model:

$$\boldsymbol{y}(t) = \operatorname{diag}\left\{\sqrt{\boldsymbol{\lambda}(t)}\right\} \operatorname{diag}\{\boldsymbol{\Theta}\}\, \boldsymbol{I}_\Psi + \boldsymbol{\varepsilon}(t), \tag{2.5}$$

where $\boldsymbol{\lambda}(t)$ is a vector describing resource allocation at time t, $[\sqrt{\boldsymbol{x}}]_i$ denotes $\sqrt{x_i}$, $\operatorname{diag}\{\boldsymbol{x}\}$ is a square matrix with $[\operatorname{diag}\{\boldsymbol{x}\}]_{ii} = x_i$ and $[\operatorname{diag}\{\boldsymbol{x}\}]_{ij} = 0$ for $i \neq j$, and $\boldsymbol{\varepsilon}(t) \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathrm{I}_{\mathrm{Q}\times\mathrm{Q}})$ where $\mathrm{I}_{\mathrm{Q}\times\mathrm{Q}}$ is an $(Q \times Q)$ identity matrix. Hence, an element-wise version of the first stage measurement model (2.5) is given by

$$y_i(1) = \sqrt{\lambda_1}\theta_i I_i + \varepsilon_i(1), \qquad i = 1, 2, \ldots, Q, \tag{2.6}$$

where $y_i(1)$ is the i-th element of $\boldsymbol{y}(1)$. Thus, we know that

$$\begin{aligned} y_i(1)|(I_i = 0) &\sim \mathcal{N}(0, \sigma^2), \\ y_i(1)|(I_i = 1) &\sim \mathcal{N}\left(\sqrt{\lambda_1}\mu_\theta, \sigma^2 + \lambda_1\sigma_\theta^2\right) \end{aligned} \tag{2.7}$$

The knowledge of these two distributions will be used to focus measurements at the second stage based on the received measurements at the first stage, using posterior probabilities derived from the first stage measurements. Second stage measurements are defined with the element-wise version

$$y_i(2) = \sqrt{\lambda_i(2)}\theta_i I_i + \varepsilon_i(2), \qquad i = 1, 2, \ldots, Q. \tag{2.8}$$

2.4 Search policy under total effort constraints

Bashan et al. [11] introduced the following cost function

$$J(\lambda) = \sum_{i=1}^{Q} \frac{\nu I_i + (1-\nu)(1-I_i)}{\lambda_i(1) + \lambda_i(2)}, \tag{2.9}$$

with $\nu \in [\frac{1}{2}, 1]$. Minimizing the expected value of equation (2.9) will allocate ν of the total effort to the ROI and the rest to its complement. Thus, the ν parameter can be thought of as a tuning nob for exploitation of the ROI ($\nu = 1$) versus exploration ($\nu = 0.5$, which leads to an exhaustive search). This cost function has some appealing properties, including the important distinction that it can actually be optimized in our framework. This is in stark contrast to other common functions, such as the mean square error (MSE) or probability of error. Moreover, Bashan et al. [11] shows that in some contexts, minimizing equation (2.9) is equivalent to (a) minimizing the Cramer-Rao lower bound on $E[\sum_i I_i(\hat{\theta}_i - \theta_i)^2]$ and (b) uniformly minimizing the Chernoff bound on the probability of error over the ROI. For a total allocation budget of $\lambda_T = 1$, the cost function in equation (2.9) is also shown to be lower bounded by

$$J(\lambda) \geq [\sqrt{\nu}|\Psi| + \sqrt{1-\nu}(Q - |\Psi|)]^2, \tag{2.10}$$

where the lower bound is achievable by a policy given by

$$\lambda_i^{optimal} = \begin{cases} \dfrac{\sqrt{\nu}}{\sqrt{\nu}|\Psi| + \sqrt{1-\nu}(Q - |\Psi|)}, & i \in \Psi \\ \dfrac{\sqrt{1-\nu}}{\sqrt{\nu}|\Psi| + \sqrt{1-\nu}(Q - |\Psi|)}, & i \in \Psi^c \end{cases} \tag{2.11}$$

for $i = 1, 2, \ldots, Q$. Note that when $\nu = 1$, this policy is the optimal policy defined in the introduction (i.e., all resources shared equally among cells containing targets). In contrast, consider the exhaustive search where

$$\lambda_i^{exhaustive} = 1/Q, \tag{2.12}$$

for all $i = 1, 2, \ldots, Q$. Then the cost associated with this policy is

$$J(\lambda^{exhaustive}) = Q[\nu|\Psi| + (1-\nu)(Q - |\Psi|)] = \begin{cases} \frac{Q^2}{2}, & \nu = 0.5 \\ \\ Q|\Psi|, & \nu = 1 \end{cases} \tag{2.13}$$

Since this policy is both non-informative and non-adaptive, one would expect better performance from any adaptive policy. Thus, we consider the performance gain (in dB) as

$$G(\lambda) = -10\log\frac{J(\lambda)}{J(\lambda^{exhaustive})}. \tag{2.14}$$

When $\nu = 1$, the optimal gain can be expressed as

$$G(\lambda^{optimal}) = -10\log\frac{|\Psi|}{Q}, \tag{2.15}$$

which shows that gains are proportional to the sparsity of the ROI, $p^* = |\Psi|/Q$. Since $\lambda^{optimal}$ requires oracle knowledge of the ROI, it is unfeasible. Thus, our goal is to find policies that come close to achieving the performance of this optimal policy. Moreover, since the I_i's are random variables, we consider replacing equation (2.9) with its expected value:

$$J(\lambda) = \sum_{i=1}^{Q} E\left[\frac{\nu I_i + (1-\nu)(1-I_i)}{\lambda_i(1) + \lambda_i(2)}\right]. \tag{2.16}$$

2.4.1 The Adaptive Resource Allocation Policy (ARAP)

Indeed, Bashan et al. [11] shows that minimizing (2.16) subject to a total energy constraint λ_T yielded ARAP, which is summarized in Algorithm 1.

Algorithm 1. *Two stage Adaptive Resource Allocation Policy (ARAP) λ^{ARAP}*

Step 1: Allocate $\lambda_i^{ARAP}(1) = \lambda_1^*$ to each cell and measure $\boldsymbol{y}(1)$.

Step 2: Given $\boldsymbol{y}(1)$ compute probabilities,

$$p_{I_i|\boldsymbol{y}(1)} \triangleq \Pr(I_i = 1|\boldsymbol{y}(1)), \tag{2.17}$$

$$w_i \triangleq \nu p_{I_j|\boldsymbol{y}(1)} + (1-\nu)(1 - p_{I_j|\boldsymbol{y}(1)}). \tag{2.18}$$

Step 3: Order the w_i's by rank so that $w_{\tau(1)} \leq w_{\tau(2)} \leq \cdots \leq w_{\tau(Q)}$.

Step 4: Use λ_1^* and the ordered statistics $w_{\tau(i)}$ to find an optimal threshold k_0, where $k_0 \in \{1, 2, \ldots, Q-1\}$ is the integer satisfying

$$\frac{\sum_{i=k_0+1}^{Q} \sqrt{w_{\tau(i)}}}{\sqrt{w_{\tau(k_0+1)}}} < \frac{\lambda_T}{\lambda_1} - k_0 \leq \frac{\sum_{i=k_0+1}^{Q} \sqrt{w_{\tau(i)}}}{\sqrt{w_{\tau(k_0)}}} \tag{2.19}$$

if

$$\frac{\lambda_T}{\lambda_1} \leq \frac{\sum_{i=1}^{Q} \sqrt{w_{\tau(i)}}}{\sqrt{w_{\tau(1)}}}, \tag{2.20}$$

and $k_0 = 0$ otherwise.

Step 5: Given k_0, apply $\lambda_i^{ARAP}(2)$, the energy allocation, to cell i as

$$\begin{aligned} \lambda_{\tau(i)}^{ARAP}(2) = \lambda_{\tau(i)}(2) = \\ \left(\frac{\lambda_T - k_0 \lambda_1^*}{\sum_{j=k_0+1}^{Q} \sqrt{w_{\tau(j)}}} \sqrt{w_{\tau(i)}} - \lambda_1^* \right) u(\tau(i) - k_0), \end{aligned} \tag{2.21}$$

and measure $\boldsymbol{y}(2)$.

To complete the description of ARAP, the quantity $\lambda_1 \in (0, \lambda_T/Q]$ can be selected by line search. A proof of existence and uniqueness of k_0 is given in the work by Bashan et al. [11].

2.4.2 Properties of ARAP

ARAP has many interesting properties that will be reproduced here without proof. The interested reader should peruse the work of Bashan et al. [9, 11] for details. For simplicity of notation, we will assume that $\nu = 1$. These properties include

1. (Performance compared to exhaustive search:)

   $$J(\lambda^{ARAP}) \leq J(\lambda^{exhaustive}), \tag{2.22}$$

   with equality achieved if $p_{I_i|\boldsymbol{y}(1)} = c, \ \forall i$. This property can be restated as ARAP is an optimal policy among all 2-stage policies that allocate resources uniformly at the first stage.

2. (Asymptotic properties [for large λ_T]):

   (a) (Convergence to ROI:)

   $$E\left[k_0\right] \to (1-p)Q \tag{2.23}$$

   (b) (Optimal first stage allocations:)

   $$\lambda_1^* \to 0 \tag{2.24}$$

   (c) (Convergence to optimal gains:)

   $$G(\lambda^{ARAP}) \to -10 \log \frac{|\Psi|}{Q} = G(\lambda^{optimal}) \tag{2.25}$$

The asymptotic properties depend on the consistency of the measurements, defined as

$$p_{I_i|\boldsymbol{y}(1)} \to I_i \tag{2.26}$$

in probability. The authors prove the validity of this condition in the Gaussian case and speculate that the condition holds for many other cases as well.

2.4.3 Suboptimal two-stage search policy

Since ARAP requires $\mathcal{O}(\log Q)$ operations to sort the w_i's, Bashan et al. [11] also provides a simple alternative to ARAP with $\lambda_i(1) = \lambda_1$ and

$$\lambda_i(2) = \frac{\lambda_T - Q\lambda_1}{\sum_{j=1}^{Q} \sqrt{w_j}} \sqrt{w_i}. \tag{2.27}$$

Although not stated in their work, this policy is equivalent to optimizing the cost function:

$$J^{suboptimal}(\lambda) = E\left[\sum_{j=1}^{Q} \frac{I_i}{\lambda_i(2)} |\boldsymbol{y}(1)\right], \tag{2.28}$$

given λ_1 and $\boldsymbol{y}(1)$ with total energy budget λ_T. Note that if $p_{I_i|\boldsymbol{y}(1)} \to I_i$, this suboptimal policy becomes equivalent to ARAP (and thus has asymptotically optimal behavior). However, since this policy does not threshold at the second stage, it will in general waste resources as compared to ARAP when SNR is not in the asymptotic regime.

2.4.4 Limitations of ARAP

The ARAP policy has many appealing properties compared to an exhaustive search. However, it is limited in a couple of important ways. First, ARAP requires sampling each individual cell of the ROI at the first stage. In some applications where Q is very large, this may result in a prohibitive cost. The next section discusses a relaxation to this problem. Second, the optimization method used to minimize equation (2.16) cannot easily be extended to $T > 2$ stages, because the optimization problem grows in a combinatorial fashion as a function of T. Bashan [9] provides a myopic policy for choosing $\lambda_i(t)$ in this case, but this policy may perform poorly in many situations. Chapter III extends ARAP to $T > 2$ and also allows for dynamic targets (i.e., where $I_i = I_i(t)$). A related problem involves the case where informative priors exists, so that $\Pr(I_i = 1) \neq p, \ \forall i$. A performance analysis in the case where $\Pr(I_i = 1)$ can take one of two levels is given in the work by Newstadt et al. [69].

2.5 Search policy under total effort constraints and multi-scale sampling constraints

In order to constrain the total number of measurements used to search the ROI, we propose a two-stage policy that uses a coarse-scale search at the first stage, followed by fine-scale search over locations in our estimated ROI at the second stage. In particular, consider the following measurement model:

$$\tilde{\boldsymbol{y}}(t) = \mathbf{H}(t)\mathrm{diag}\left\{\sqrt{\tilde{\boldsymbol{\lambda}}(t)}\right\}\mathrm{diag}\left\{\boldsymbol{\Theta}\right\}\boldsymbol{I}_{\Psi} + \boldsymbol{\varepsilon}(t), \tag{2.29}$$

where $\mathbf{H}(t)$ is an $(N_t \times Q)$ matrix describing the "beamforming" measurement operator, $\tilde{\boldsymbol{\lambda}}(t)$ is a vector describing resource allocation at time t, $[\sqrt{\boldsymbol{x}}]_i$ denotes $\sqrt{x_i}$, $\mathrm{diag}\left\{\boldsymbol{x}\right\}$ is a square matrix with $[\mathrm{diag}\left\{\boldsymbol{x}\right\}]_{ii} = x_i$ and $[\mathrm{diag}\left\{\boldsymbol{x}\right\}]_{ij} = 0$ for $i \neq j$, and $\boldsymbol{\varepsilon}(t) \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathrm{I}_{\mathrm{N_t}\times\mathrm{N_t}})$ where $\mathrm{I}_{\mathrm{N_t}\times\mathrm{N_t}}$ is an $(N_t \times N_t)$ identity matrix. The beamforming operator $\mathbf{H}(t)$ simply forms linear combinations of neighboring pixels and is what distinguishes M-ARAP from ARAP. In our model (2.29), both $\mathbf{H}(t)$ and $\tilde{\boldsymbol{\lambda}}(t)$ are design parameters that satisfy the user-defined constraints. We propose the following simple design. Define L to be a factor of Q, and define the column vectors

$\mathbf{g}_L, \mathbf{0}_L \in \mathbb{R}^L$:

$$\mathbf{g}_L = L \text{ times} \left\{ \begin{bmatrix} \frac{1}{L} \\ \vdots \\ \frac{1}{L} \end{bmatrix} \right. , \qquad \mathbf{0}_L = L \text{ times} \left\{ \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \right. \tag{2.30}$$

Then, we consider the beamforming matrix for the first stage measurements (defined through the transpose)

$$\mathbf{H}(1) = \underbrace{\begin{bmatrix} \mathbf{g}_L & \mathbf{0}_L & \mathbf{0}_L & \cdots & \mathbf{0}_L \\ \mathbf{0}_L & \mathbf{g}_L & \mathbf{0}_L & \cdots & \mathbf{0}_L \\ \mathbf{0}_L & \mathbf{0}_L & \mathbf{g}_L & \cdots & \mathbf{0}_L \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \mathbf{0}_L & \mathbf{0}_L & \cdots & \mathbf{g}_L \end{bmatrix}^T}_{\frac{Q}{L} \text{ times}} \in \mathbf{R}^{(Q/L)\times Q} \tag{2.31}$$

Hence, an element-wise version of the first stage measurement model (2.29) is given by

$$\tilde{y}_j(1) = \frac{\sqrt{\tilde{\lambda}_1}}{L} \sum_{i=(j-1)L+1}^{jL} \theta_i I_i + \varepsilon_j(1), \qquad j = 1, 2, \ldots, N_1, \tag{2.32}$$

where $\tilde{y}_j(1)$ is the j-th element of $\tilde{\boldsymbol{y}}(1)$. Let $\mathcal{X}_j = \{(j-1)L+1, \ldots, jL-1, jL\}$ denote the support of the j-th row of $\mathbf{H}(t)$, and note that $|\mathcal{X}_j| = L$ for all j. With small p, large Q, and $L \ll Q$ the probability that $\mathcal{X}_j$ contains more than one target is negligible. Let H_j denote the indicator function of the event, "$I_i = 1$ for some $i \in \mathcal{X}_j$." Then we know that

$$\begin{aligned} \tilde{y}_j(1)|(H_j = 0) &\sim \mathcal{N}(0, \sigma^2), \\ \tilde{y}_j(1)|(H_j = 1) &\sim \mathcal{N}\left(\frac{\sqrt{\tilde{\lambda}_1}}{L} \mu_\theta, \sigma^2 + \frac{\tilde{\lambda}_1}{L^2} \sigma_\theta^2 \right) \end{aligned} \tag{2.33}$$

As in the work by Bashan et al. [11], the knowledge of these two distributions will be used to focus measurements at the second stage based on the received measurements at the first stage, using posterior probabilities derived from the first stage measurements. Second stage measurements are defined with $\mathbf{H}(2) = \mathrm{I}_{Q\times Q}$, which gives the element-wise version

$$y_j(2) = \sqrt{\lambda_j(2)}\theta_j I_j + \varepsilon_j(2), \quad j = 1, 2, \ldots, Q. \tag{2.34}$$

We define the following resource allocation policy based on ARAP:

Algorithm 2. *Two stage Multi-scale Adaptive Resource Allocation Policy (M-ARAP)* λ^{M-ARAP}

Step 1: Allocate $\tilde{\lambda}_j^{M-ARAP}(1) = \tilde{\lambda}_1^*$ *to each support* $\mathcal{X}_j$ *and measure* $\tilde{\boldsymbol{y}}(1)$ *in (2.29).*

Step 2: Compute probabilities $\tilde{p}_{H_j|\tilde{\boldsymbol{y}}(1)} \overset{\triangle}{=} \Pr(H_j = 1|\boldsymbol{y}(1))$ *and* $\tilde{w}_j \overset{\triangle}{=} \nu\tilde{p}_{H_j|\boldsymbol{y}(1)} + (1-\nu)(1-\tilde{p}_{H_j|\boldsymbol{y}(1)})$ *over each support region.*

Step 3: Rank order the $\tilde{w}_j$*'s using (2.37), then use* $\tilde{\lambda}_1^*$ *and the ordered statistic* $\tilde{w}_{\tau(j)}$ *to find a threshold* $\tilde{k}_0$ *via (2.38) and (2.39).*

Step 4: Given $\tilde{k}_0$*, define the energy allocation to support region* j *as*

$$\tilde{\lambda}_{\tau(j)}(2) = \left(\frac{\lambda_T - \tilde{k}_0\tilde{\lambda}_1^*}{\sum_{l=\tilde{k}_0+1}^{Q}\sqrt{\tilde{w}_{\tau(l)}}}\sqrt{\tilde{w}_{\tau(j)}} - \tilde{\lambda}_1^*\right)u(\tau(j) - \tilde{k}_0), \tag{2.35}$$

Step 5: Define the energy allocation to cell i *in support region* j *as*

$$\lambda_i^{M-ARAP}(2) = \frac{\tilde{\lambda}_j(2)}{L}, \qquad j = 1, 2, \ldots, N_1 \tag{2.36}$$

and measure $\boldsymbol{y}(2)$ *using* $\mathbf{H}(2) = \mathrm{I}_{Q\times Q}$ *and* $[\boldsymbol{\lambda}(2)]_i = \lambda_i^{M-ARAP}(2)$.

To complete the definition of M-ARAP, define the permutation operator τ corresponding to the rank ordering of the $\tilde{w}_j$'s as

$$\tau(j) = \arg\min_{i=1,\ldots,N_1}\{\tilde{w}_i : \tilde{w}_i \geq \tilde{w}_{\tau(j-1)}\}, \quad j \in \{1, 2, \ldots, N_1\}, \tag{2.37}$$

with $\tilde{w}_{\tau(0)} \overset{\triangle}{=} 0$. Whenever the r.h.s. of (2.37) is not unique, we select an arbitrary i satisfying $\tilde{w}_{\tau(1)} \leq \tilde{w}_{\tau(2)} \leq \cdots \leq \tilde{w}_{\tau(N_1)}$. Then, assuming $\tilde{w}_{\tau(1)} > 0$, define $\tilde{k}_0$, the threshold parameter, as $\tilde{k}_0 = 0$ if

$$\frac{\lambda_T}{\tilde{\lambda}_1^*} > \frac{\sum_{j=1}^{N_1} \sqrt{\tilde{w}_{\tau(j)}}}{\sqrt{\tilde{w}_{\tau(1)}}}, \tag{2.38}$$

otherwise $\tilde{k}_0 \in \{1, \ldots, N_1 - 1\}$ is the integer satisfying

$$\frac{\sum_{j=\tilde{k}_0+1}^{N_1} \sqrt{\tilde{w}_{\tau(j)}}}{\sqrt{\tilde{w}_{\tau(\tilde{k}_0+1)}}} < \frac{\lambda_T}{\tilde{\lambda}_1^*} - \tilde{k}_0 \leq \frac{\sum_{j=\tilde{k}_0+1}^{N_1} \sqrt{\tilde{w}_{\tau(j)}}}{\sqrt{\tilde{w}_{\tau(\tilde{k}_0)}}} \tag{2.39}$$

A proof of the existence and uniqueness of $\tilde{k}_0$ in (2.39), as well as a discussion of its properties, is given in the work by Bashan et al. [11]. Note that the definition of $\tilde{k}_0$ is identical to ARAP for a search space of size $N_1 = Q/L$. For $N_1 = Q$, M-ARAP is completely equivalent to ARAP, provided that $\tilde{\lambda}_1^*$ is correctly defined. To define $\tilde{\lambda}_1^*$, let

$$\tilde{\lambda}_1^* = \arg \min_{\tilde{\lambda}_1 \in \left(0, \frac{\lambda_T}{N_1}\right)} \mathrm{E}\left\{ \sum_{i=1}^{Q} \frac{\nu I_i + (1-\nu)(1-I_i)}{\frac{\tilde{\lambda}_1}{L} + \lambda_i(2)} \right\}, \tag{2.40}$$

where $\lambda_i(2)$ is defined in (2.36) substituting $\tilde{\lambda}_1$ for $\tilde{\lambda}_1^*$. Note that $\lambda_i(2)$ depends on $\tilde{p}_{H_j|\tilde{\boldsymbol{y}}(1)}$, which, in turns depends on the distribution of target amplitudes, $\boldsymbol{\Theta}$. To analyze the performance of M-ARAP, we first establish properties of $\tilde{p}_{H_j|\tilde{\boldsymbol{y}}(1)}$.

2.5.1 Detectability index and asymptotic properties of $\tilde{p}_{H_j|\tilde{\boldsymbol{y}}(1)}$ when $\nu = 1$

Let $|\mathcal{X}_j| = L$ denote an observed support size for the first stage in M-ARAP. Let the true mean sparsity of the observed signal be $p = \frac{\mathrm{E}\{|\Psi|\}}{Q}$. Define the *detectability index* as

$$d = \frac{|\mathrm{E}\{\tilde{y}_j(1)|H_j = 1\} - \mathrm{E}\{\tilde{y}_j(1)|H_j = 0\}|}{\sqrt{\mathrm{var}(\tilde{y}_j(1)|H_j = 0)}}. \tag{2.41}$$

Substituting (2.7) into (2.41) yields

$$d = \frac{\mu_\theta}{L}\sqrt{\frac{\tilde{\lambda}_1}{\sigma^2}}, \tag{2.42}$$

which is proportional to μ_θ and to the square root of effective SNR $\frac{\tilde{\lambda}_1}{L\sigma^2}$. Therefore, for a specified false alarm rate, we expect the power of the likelihood ratio test (LRT) performed on $\boldsymbol{y}(1)$ to increase as either the inherent contrast μ_θ or effective SNR increase. In this section, we analyze asymptotic properties of M-ARAP, where by asymptotic we mean high SNR and large Q. We further assume that $\mathcal{X}_j$ contains at most one target, and that the target amplitude variance, $\sigma_\theta^2 = 0$. Under these assumptions, we can establish Claim 1, Claim 2, and Claim 3 given below.

Claim 1. *$\tilde{p}_{H_j|\tilde{\boldsymbol{y}}(1)} \to H_j$ in probability as SNR $\to \infty$.*

Derivation of Claim 1: Under the assumptions that $\mathcal{X}_j$ contains at most one target and $\sigma_\theta^2 = 0$, the first stage measurement reduces to

$$\tilde{y}_j(1) = \frac{\sqrt{\tilde{\lambda}_1} H_j \mu_\theta}{L} + \varepsilon_j(1) = \sqrt{\gamma_1} H_j \mu_\theta + \varepsilon_j(1) \tag{2.43}$$

where $\gamma_1 = \tilde{\lambda}_1 / L^2$. Bashan et al. [11] proved that $\tilde{p}_{H_j|\tilde{\boldsymbol{Y}}(1)} \to H_j$ in probability as $\gamma_1 \to \infty$. Thus, Claim 1 follows directly from this result, noting that SNR $\to \infty$ implies $\gamma_1 \to \infty$.

In the work by Bashan et al. [11], the asymptotic consistency property of $\tilde{p}_{H_j|\tilde{\boldsymbol{Y}}(1)}$ was used to show that the threshold parameter $\tilde{k}_0$ converges to the true number of empty search cells $(1-p)Q$ of the scanned domain (recall that $|\hat{\Psi}| = Q - \tilde{k}_0$). For λ^{M-ARAP} we can provide an asymptotic bound on $\tilde{k}_0$ that holds in probability. The logic behind the bound is that if p can be used to bound K, the number of support regions $\mathcal{X}_j$ that contain targets, then $\tilde{k}_0 \geq Q - KL$. Therefore, we have the following:

Claim 2. *The normalized number of samples N^* used by M-ARAP, defined as*

$$N^* = \frac{N_1 + |\widehat{\Psi}|}{Q}, \tag{2.44}$$

is bounded between $\frac{1}{L}$ and $\frac{1}{L} + p^*L$, where p^* is the true sparsity of the underlying domain, in the sense that

$$\lim_{\text{SNR}\to\infty} \Pr\left(\frac{1}{L} \le N^* \le \frac{1}{L} + p^*L\right) = 1, \tag{2.45}$$

for sufficiently large Q.

Derivation of Claim 2: To prove (2.45) note first that

$$N^* = \frac{N_1 + |\widehat{\Psi}|}{Q} = \frac{1}{L} + \frac{|\widehat{\Psi}|}{Q} \ge \frac{1}{L}. \tag{2.46}$$

It suffices to show

$$\lim_{\text{SNR}\to\infty} \Pr\left(\frac{|\widehat{\Psi}|}{Q} \le pL\right) = 1. \tag{2.47}$$

Bashan et al. [11] showed that if $p_{I_i|\boldsymbol{y}(1)} \to I_i$ in probability as SNR → ∞, then

$$\tilde{k}_0 \to Z \sim \text{Bin}\,(S, 1-r) \tag{2.48}$$

in probability as SNR → ∞ for a search space of size S and sparsity r. Recalling that $\tilde{k}_0$ is calculated using ARAP for $S = Q/L$ and $r = pL$, it follows that $\tilde{k}_0 \to Z \sim \text{Bin}\,(Q/L, 1-pL)$ as SNR → ∞. Using the definition of $\widehat{\Psi}$ from (2.4), we have

$$\begin{aligned}
|\widehat{\Psi}| &= |\{i \in \mathcal{X} : \lambda_i(2) > 0\}| \\
&= \left|\bigcup_{j=1}^{Q/L} \{i \in \mathcal{X}_j : H_j = 1\}\right| && \text{(as SNR} \to \infty) \\
&= \sum_{j=1}^{Q/L} |\{i \in \mathcal{X}_j : H_j = 1\}| && \text{(for disjoint } \mathcal{X}_j) \\
&= \sum_{j=1}^{Q/L} LH_j = L\sum_{j=1}^{Q/L} H_j = L\left(Q/L - \tilde{k}_0\right)
\end{aligned} \tag{2.49}$$

Since, $\tilde{k}_0 \sim \text{Bin}\,(Q/L, 1-pL)$ as SNR → ∞, we have $E\left[\frac{|\widehat{\Psi}|}{Q}\right] \to pL$ and $\text{var}\left(\frac{|\widehat{\Psi}|}{Q}\right) \to \frac{pL(1-pL)}{Q}$ as SNR → ∞. Moreover, $\text{var}\left(\frac{|\widehat{\Psi}|}{Q}\right) \to 0$ as $Q \to \infty$. Thus, $\frac{|\widehat{\Psi}|}{Q} \to pL$ in the mean square sense. This establishes Claim 2.

Claim 3. *The expected proportion of the scanned area that is empty of targets at the second stage is bounded with probability, in the sense that*

$$\Pr\left(\frac{|\Psi \triangle \widehat{\Psi}|}{Q} \leq p(L-1)\right) = 1. \tag{2.50}$$

Derivation of Claim 3: This result is established similarly to Claim 2. In particular, note that when SNR → ∞

$$\begin{aligned} |\Psi \triangle \widehat{\Psi}| &= \left|\bigcup_{j=1}^{Q/L} \{i \in \mathcal{X}_j, i \notin \Psi : H_j = 1\}\right| \\ &= \sum_{j=1}^{Q/L} |\{i \in \mathcal{X}_j : H_j = 1\} \triangle \{i \in \mathcal{X}_j, i \in \Psi : H_j = 1\}| \\ &= \sum_{j=1}^{Q/L} |\{i \in \mathcal{X}_j : H_j = 1\}| - |\{i \in \mathcal{X}_j, i \in \Psi : H_j = 1\}| \\ &= \sum_{j=1}^{Q/L} (L-1)H_j = (L-1)\left(Q/L - \tilde{k}_0\right) \end{aligned} \tag{2.51}$$

This leads to $E\left[\frac{|\Psi \triangle \widehat{\Psi}|}{Q}\right] = p(L-1)$ and $\text{var}\left(\frac{|\Psi \triangle \widehat{\Psi}|}{Q}\right) = \frac{(L-1)^2}{LQ} p(1-pL) \to 0$ as $\text{SNR}, Q \to \infty$. Thus, $\frac{|\Psi \triangle \widehat{\Psi}|}{Q} \to p(L-1)$ in the mean square sense (which implies convergence in probability as required). This establishes Claim 3.

Claim 3 provides a bound that can be used to evaluate the tradeoff between reducing the number of measurements and increasing the expected estimation gains. Specifically, use the cost function (2.9) and assume that asymptotically we learn $\widehat{\Psi}$ at almost no cost. Recall that an optimal policy $\lambda^{optimal}$, given in (2.11), allocates resources only to cells containing targets, and has gain:

$$G(\lambda^{optimal}) = 10\log\frac{Q}{|\widehat{\Psi}|} = 10\log\frac{Q}{|\Psi| + |\Psi \triangle \widehat{\Psi}|}. \tag{2.52}$$

Using *Claim 3* we obtain

$$G(\lambda_{N_o}) \geq 10\log\frac{Q}{pQ(1+L-1)} = 10\log\frac{1}{p} - 10\log L, \tag{2.53}$$

where $10\log L$ is the gain penalty that we pay due to multi-scale search.

2.5.2 Discussion of performance for clustered targets

The assumption that only a single target may appear in a support region may not hold when apparent targets clump together. This scenario arises in radar target detection where large scattering objects may occupy consecutive pixels on the radar screen and appear as a cluster of targets. Nevertheless, the overall area occupied by targets is small compared to the area a scanning radar system covers. As another example, in early detection of cancer tumors such as in breast cancer, the diameter of a typical tumor is a few millimeters to 1.5 centimeters. Hence, on a fine grid a tumor may appear as a cluster of targets, yet, its overall volume is very small compared to the volume of the entire breast.

In these cases we speculate that the performance of M-ARAP with clustered targets is lower bounded by the performance in the case of single-pixel targets. Indeed, if we let Ξ_j be the total number of targets in support j, with $E|\Xi_j| = E|\Xi| \geq 1$, and Claim 1 still holds, then it can be shown that

$$G(\lambda_{N_o}) \geq 10 \log \frac{1}{p} - 10 \log(1 + L - E|\Xi|), \tag{2.54}$$

where $10 \log(1 + L - E|\Xi|) \leq 10 \log L$ is the gain penalty that we pay due to multiscaling. In other words, having clustered targets within a support region tends to reduce the gain penalty due to multi-scale processing.

Despite the limitations of the single deterministic target-per-region assumption in Section 2.5.1 to obtain asymptotic limits, we believe that M-ARAP's predicted performance gains will hold under broader conditions. This belief is supported by our numerical results shown in the next section. Note that the gains established in Section 2.5.1 require only that the posterior probabilities, $\tilde{p}_{H_j|\boldsymbol{Y}\tilde{(1)}}$ converge to the indicator function H_j, which may not require the assumption of a single target per multi-resolution cell.

2.6 Performance comparisons

2.6.1 Estimation

Assume that cell l belongs to the ROI, i.e., $I_l = 1$. Here we introduce an estimate of target amplitude θ_l using the measurement pair $(\tilde{\boldsymbol{y}}(1), \boldsymbol{y}(2))$. Assuming these amplitudes are independent and obey a Gaussian prior distribution, $\theta_l \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$, we can derive the conditional mean estimator (CME) $\hat{\theta}_l = E[\theta_l | \tilde{\boldsymbol{y}}(1), \boldsymbol{y}(2)]$, which is the minimum mean squared error (MSE) estimator. As a baseline, we will compare this estimator to the CME $\mathrm{E}\{\theta_l | \boldsymbol{y}(0)\}$ under an exhaustive search policy, where

$$y_i(0) = \sqrt{\lambda_0}\theta_i I_i + \varepsilon_i(0), \quad \varepsilon_i(0) \sim \mathcal{N}(0, \sigma^2) \tag{2.55}$$

and $\lambda_0 = \frac{\lambda_T}{Q}$. The MSE of the CME for an exhaustive search policy is given by

$$\mathrm{var}\{\theta_l | y_l(0)\} = \sigma_\theta^2 - \frac{\lambda_0 \sigma_\theta^4}{\sigma^2 + \lambda_0 \sigma_\theta^2} = \frac{\sigma_\theta^2}{1 + \lambda_0 \frac{\sigma_\theta^2}{\sigma^2}}. \tag{2.56}$$

Recall from equation (2.7) that $\tilde{y}_j(1) | H_j = r$ is Gaussian for $r = 0, 1$. Thus, we know for $I_l = 1$ and $l \in \mathcal{X}_j$ that

$$\tilde{y}_j(1) | I_l = 1 \sim \mathcal{N}\left(\frac{\sqrt{\tilde{\lambda}_1}}{L}\mu_\theta, \sigma^2 + \frac{\tilde{\lambda}_1}{L^2}\sigma_\theta^2\right). \tag{2.57}$$

In general, we can formulate an estimate of θ_l from the past t measurements by taking advantage of the conditional Gaussian distributions given those measurements. In particular, we know from the Kalman filter equations that

$$\delta_i(t) = m_i(t) - h_i(t)\hat{\theta}_i(t|t-1) \tag{2.58}$$

$$s_i(t) = h_i(t)^2\hat{\sigma}_i^2(t|t-1) + \sigma^2 \tag{2.59}$$

$$\Gamma_i(t) = \frac{\hat{\sigma}_i^2(t|t-1)h_i(t)}{s_i(t)} \tag{2.60}$$

$$\hat{\theta}_i(t|t) = \hat{\theta}_i(t|t-1) + \Gamma_i(t)\delta_l(t), \tag{2.61}$$

$$\hat{\sigma}_i^2(t|t) = [1 - \Gamma_i(t)h_i(t)]\hat{\sigma}_i^2(t|t-1), \tag{2.62}$$

where $h_i(t)$ is the measurement gain, $m_i(t)$ is the measurement of the i-th target, $\delta_i(t)$ is the residual measurement error, $s_i(t)$ is the update measurement error, $\Gamma_i(t)$ is the Kalman gain, and $(\hat{\theta}_i(t|t), \hat{\sigma}_i^2(t|t))$ are the updated state estimates. Moreover, since we assume that the θ_i's are static, we have

$$\hat{\theta}_i(t|t-1) = \hat{\theta}_i(t-1|t-1), \tag{2.63}$$

$$\hat{\sigma}_i^2(t|t-1) = \hat{\sigma}_i^2(t-1|t-1). \tag{2.64}$$

These equations can be simplified as:

$$\hat{\theta}_i(t|t) = \frac{[h_i(t)m_i(t)]\,\hat{\sigma}_i^2(t-1|t-1) + \hat{\theta}_i(t-1|t-1)\sigma^2}{\sigma^2 + [h_i(t)]^2\hat{\sigma}_i^2(t-1|t-1)}, \tag{2.65}$$

$$\hat{\sigma}_i^2(t|t) = \frac{\hat{\sigma}_i^2(t-1|t-1)\sigma^2}{\sigma^2 + [h_i(t)]^2\hat{\sigma}_i^2(t-1|t-1)}. \tag{2.66}$$

For our model, we have:

$$m_i(t) = \begin{cases} \tilde{y}_l(1), & t = 1 \\ y_i(2), & t = 2 \end{cases} \tag{2.67}$$

and

$$h_i(t) = \begin{cases} \frac{\sqrt{\lambda_1}}{L}, & t = 1 \\ \sqrt{\lambda_i(2)}, & t = 2 \end{cases} \tag{2.68}$$

Finally, the estimator for θ_i is given by applying equations (2.65) and (2.66) after each measurement, yielding an estimate, $\hat{\theta}_i(2|2)$. Simulation results are shown in Figs. 2.4-2.5. In both figures we plot the MSE performance gain $g(\lambda)$, defined as

$$g(\lambda) = 10 \log \frac{\text{var}(\theta_l | y_l(0))}{\text{MSE}(\hat{\theta}_l)} \tag{2.69}$$

as a function of SNR (Fig. 2.4) and the detectability index given in (2.42) (Fig. 2.5). Monte-Carlo simulations were used to estimate the MSE of the estimator given in the previous section (2.65). We chose $Q = 12,000$, $p = \frac{1}{1000}$, and each point on the figure represents the mean MSE based on 500 realizations. We let signal to noise ratio, defined as $10 \log \frac{\lambda_T/Q}{\sigma^2}$, vary from 0 to 40 [dB], used contrast level $\mu_\theta \in \{2, 4, 8\}$, and set $\sigma_\theta^2 = \frac{1}{16}$. Different lengths L were simulated for the first stage, but we present here the cases of $L = 8$ and $L = 32$ since it is enough to understand the general trends. Curves with different markers and colors represent different contrast levels μ_θ.

Note that, in contrast to ARAP, we do not claim optimality of the proposed estimator that uses the M-ARAP policy. Indeed the optimal gain of 30 [dB] is not attained. Moreover, asymptotic gains decrease as L increases. This is natural since the posterior probabilities $\tilde{p}_{H_j|\tilde{\boldsymbol{y}}(1)}$ are identical within each support. Hence, if the resource allocation scheme λ^{M-ARAP} suspects that a target exists in $\mathcal{X}_j$, all cells within this support receive the same effort allocation for the second stage. As L increases, this translates to wasted resources according to *Claim 3*. Fig. 2.4 shows asymptotic gain of 21 [dB] for $L = 8$ and 15 [dB] for $L = 32$, both agreeing with (2.53).

In Fig. 2.5 we plot estimation gains vs. the detectability index since it incorporates both the contrast level and the scale in a single parameter. We see that the detectability index is a reasonable predictor of the performance of M-ARAP across contrast levels and scales. We note that when L is large and μ_θ is small, the mean MSE gains may be negatively affected by a very small number of missed detections when θ_i is small compared to μ_θ. This is a drawback of using the detectability index to predict performance, as it does not consider the variance of the amplitudes, σ_θ^2. Consider Figure 2.6, which plots gains in median MSE instead of mean MSE. In the cases where detection is hard due to variations in θ_i (i.e., large L and small μ_θ), the median MSE is more robust to a few outliers than mean MSE. This leads to fewer discrepancies as a function of the performance index. On the other hand, when $L = 8$, we see that in the transition region between no gain (i.e., index values less than 2.5) and asymptotic gain (index values greater than 10), there is a discrepancy across contrast levels.

Both of these plots indicate that the detectability index may be useful for performance prediction, though a better statistic may exist that factors in the variance of the amplitudes through σ_θ^2.

2.6.2 Normalized number of samples, N^*

The normalized number of samples N^* is lower bounded by $\frac{1}{L}$, and hence there is a tradeoff between estimation gain and reduction in number of measurements. We show here that if the detectability index is sufficiently high ($d > 5$ in our case) we can both save measurements (according to *Claim 2*) and enjoy significant estimation gain within the ROI. Fig. 2.7 shows the expected saving in measurements or N^* for the scenario depicted in Fig. 2.5. Black curves represent $L = 8$, blue curves represent $L = 16$, and purple curves represent $L = 32$. Combining the information on both figures shows measurement saving relative to estimation gain. For example,

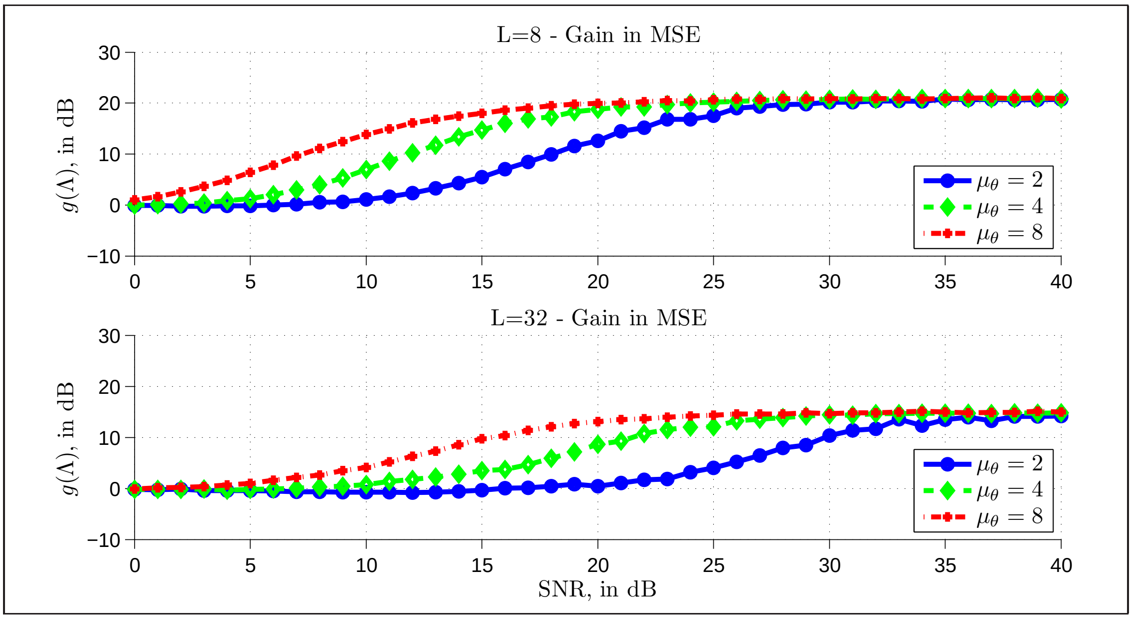


Figure 2.4: We plot estimation gains as a function of SNR for different contrast levels. The upper plot show gains for $L = 8$ while the lower plot show gains for $L = 32$. In the upper plot, significant gains of 10 [dB] are achieved for all contrasts at SNR values less than 13 [dB]. In the lower plot, 10 [dB] gains occur at high contrasts at SNR less than 20 [dB]. Note that the asymptotic lower bound on the gain (2.53) yields 21.0 [dB] and 15.0 [dB] for $L = 8$ and $L = 32$ respectively, which agree well with the gains in these plots.

for $d = 5$, M-ARAP with $L = 8$ yields about 10 [dB] performance gain in estimation while using only 18% of the number of samples used by an exhaustive search. Similar performance gain is achieved by M-ARAP for $L = 32$ with $d = 5$ and about 8% of the samples. Note that in all three cases, N^* converges to the upper bound in *Claim* 2 (0.133, 0.0785, and 0.06325 for $L = 8, 16, 32$ respectively).

Previous experience with ARAP and M-ARAP in the context of detection tasks suggests that optimizing the energy allocation between the two stages is difficult and very much application dependent. This will negatively affect performance if noise variance σ^2 is unknown and must be estimated from the data. Since M-ARAP is a coarser version of ARAP, we present detection performance only in the context of

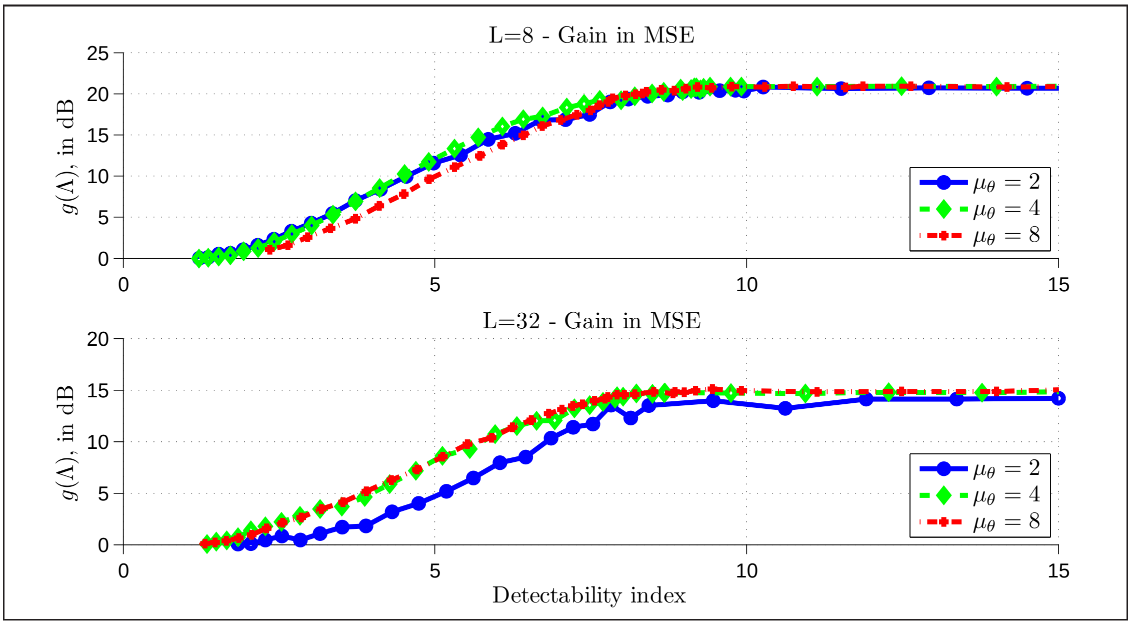


Figure 2.5: Estimation gains (in mean MSE) are plotted against detectability index for $L = 8$ and $L = 32$. Note that the detectability index can be used as a reasonable predictor of MSE gain, regardless of the actual contrast, SNR, or scale.

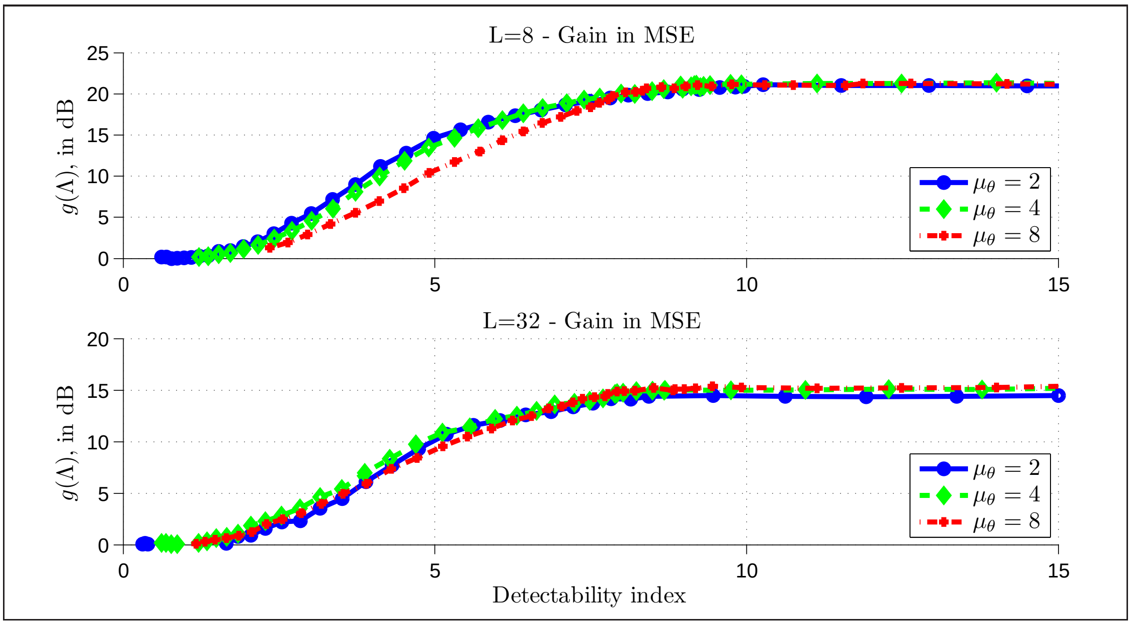


Figure 2.6: Estimation gains (in median MSE) are plotted against detectability index for $L = 8$ and $L = 32$. Note that when the median MSE is used as compared to mean MSE in Figure 2.5, we see many fewer discrepancies as a function of the detectability index for large L or small μ_θ. On the other hand, for small L, the median MSE is overly optimistic for small μ_θ causing a discrepancy across contrast levels in the transition region.

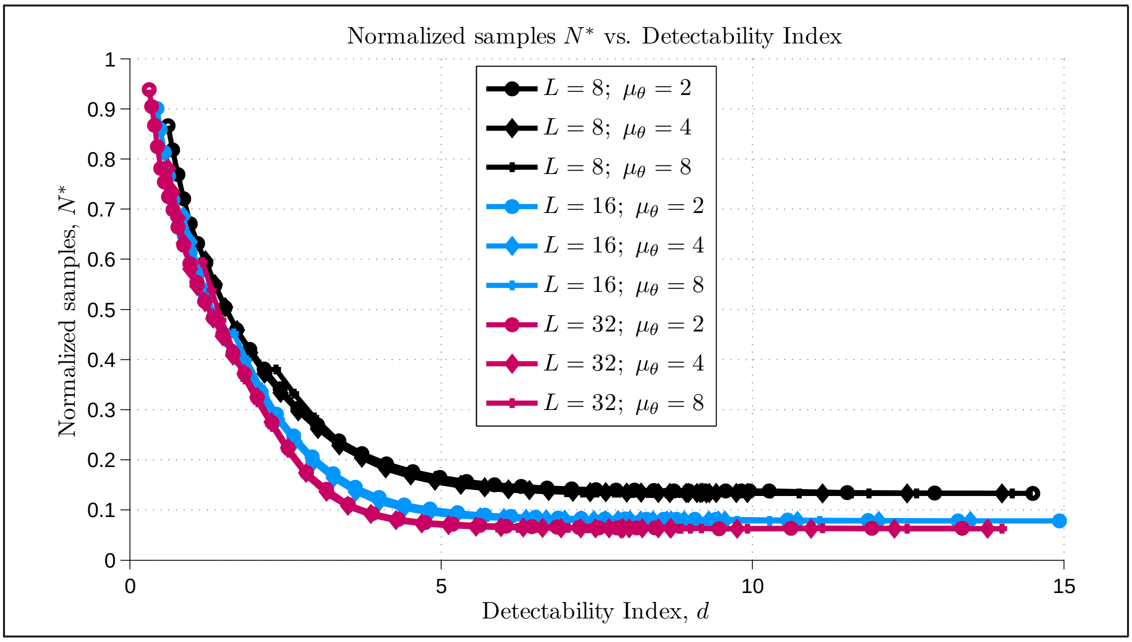


Figure 2.7: We plot the normalized number of samples N^* as a function of detectability index for $L = 8, 16, 32$, and different contrast levels $\mu_\theta \in \{2, 4, 8\}$. These N^* values are associated with estimation gains seen in Fig. 2.5. For example for a relatively low detectability index of $d = 5$ and $L = 8$, estimation performance gain of 10 [dB] is achieved with less than 18% of the sampling used by exhaustive search. Similar gains are achieved for $d = 5$, $L = 32$, and less than 8% of the samples.

the moving target indicator example given in Section 2.7.1, where we compute false discovery rate test for target detections using M-ARAP as compared to exhaustive search (see Fig. 2.12).

2.6.3 Computational complexity comparison

The computation of the search policy in M-ARAP requires computing $N_1 = Q(1/L)$ posterior probabilities ($\mathcal{O}(N_1)$), sorting the probabilities ($\mathcal{O}(\log N_1)$ operation), and computation of the second stage allocations ($\mathcal{O}(|\widehat{\Psi}|)$). Thus, the complexity of M-ARAP is

$$\mathcal{C}_{\mathrm{M-ARAP}} = \mathcal{O}(N_1) + \mathcal{O}(\log N_1) + \mathcal{O}(|\widehat{\Psi}|) = \mathcal{O}(N) \qquad (2.70)$$

where N is the number of total measurements used by M-ARAP. Assuming that the asymptotic properties of the previous section hold, $N \to Q(1/L + pL)$ so that the complexity of M-ARAP is $\mathcal{O}\left(Q(1/L + pL)\right)$. Note that ARAP is just a special case of M-ARAP for L=1. Moreover, the computational complexity of the Abdel-Samad and Tewfik algorithm (henceforth referred to as AS-T) is given by the authors as $\mathcal{O}(tN^2)$, where $t = \log_2(Q)$ is the number of stages in their hierarchical binary search [3]. Thus, the ratio of computational complexity between M-ARAP and AS-T is $\mathcal{O}(N \log_2 Q)$. Table 2.1 shows the dB gain in number of measurements between the AS-T algorithm and M-ARAP for $p = 0.01$ and various values of Q, and L. This comparison highlights the computational burden of AS-T, which requires significantly more computations than M-ARAP in all cases studied.

Distilled sensing only requires calculation of an allocation once for each stage in its implementation. Haupt et al. states that the number of stages should be chosen to be $K = 1 + \left\lceil \frac{\log_2 \log_2 Q}{\log_2(2-\Delta)} \right\rceil$ [48], where we choose $\Delta = 0.9$ for our comparison. Then the ratio of computational complexity to M-ARAP can be calculated as $\mathcal{O}(K/N)$.

Table 2.1: Computational complexity comparison between M-ARAP and AS-T for m=2 in dB

Q	ARAP (M-ARAP, L=1)	M-ARAP L=8	M-ARAP L=32
128	30.0	23.1	20.8
1024	40.5	33.6	31.3
8192	50.7	43.8	41.5

However, it should be noted that although DS has lower computational complexity, the number of measurements is generally larger than M-ARAP. Under the assumption that exactly one half of the cells with $I_i = 0$ are removed at each stage of DS, the expected number of measurements can be derived as

$$E[N_{DS}] = pQK + 2(1-p)Q(1-2^{-K}) \tag{2.71}$$

We set $Q = 12000$, $p = 0.001$ and compare DS to M-ARAP for $L = 1, 8, 32$ over a range of SNR. Figure 2.8(a) plots the loss in computational complexity between M-ARAP and DS. In the studied case, DS requires 26 dB, 17 dB, and 14 dB fewer computations asymptotically than M-ARAP for $L = 1, 8, 32$, respectively. Figure 2.8(b) plots the gain in the cost function over exhaustive search as a function of SNR for all four algorithms. It is seen that DS outperforms M-ARAP for low SNR values, but its asymptotic performance is significantly lower (which is to be expected, considering that ARAP optimizes this cost function). Figure 2.8(c) plots the gain as a function of the detectability index, and shows that for $d > 5$, M-ARAP outperforms DS at all given scales. Finally, Figure 2.8(d) plots the percentage of measurements used by M-ARAP compared to DS as a function of detectability index. In (a) and (d), yellow markers indicate the points on the curve where the performance of DS is approximately equal to that of M-ARAP. It is seen that M-ARAP saves considerably on measurements at all observed scales.

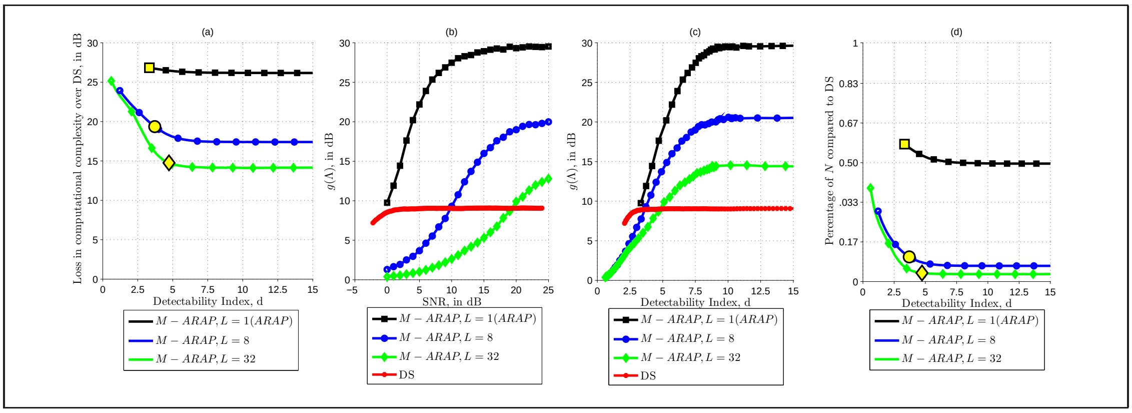


Figure 2.8: In (a), we plot the loss in computational complexity of M-ARAP ($L = 8, 32$) and ARAP ($L = 1$) vs distilled sensing (DS). We see that DS requires significantly fewer computations than M-ARAP and ARAP. In (b), we plot the gain in cost function over an exhaustive search given by (2.14) for M-ARAP ($L = 8, 32$), ARAP (L=1), and DS. For lower values of SNR, DS outperforms all versions of M-ARAP. However, the asymptotic performance of DS is lower than M-ARAP. In (c), the same gains are plotted as a function of the detectability index. In (d), the percentage of total measurements between M-ARAP and DS is plotted. In (a) and (d), yellow markers indicate the points on the curve where the performance of DS equals M-ARAP. It is seen that in all cases, M-ARAP uses significantly fewer measurements to get similar performance to DS.

2.7 Application: Moving target indication/detection

Moving target indication (MTI) radars provide the capability to detect targets reflections having differential radial motion with respect to an interfering background (called clutter) that might typically consist of reflections from terrain, sea, weather, or chaff. A typical application of such radar is surveillance, e.g., to detect low-flying aircraft moving over terrain through possible weather disturbances. Schleher shows that the function of the MTI radar is to reject returns from terrain and weather so to maintain adequate target detection [80]. In many cases MTI radar suppresses clutter by more than 20 [dB].

In the following section we use a simplified MTI simulation to illustrates potential benefits yielded by M-ARAP. A full and realistic emulation of MTI is outside the scope of this thesis. We simulate a field of view (FOV) about 66 [km^2] with pixel dimensions of 20×20 [m^2] and radar resolution cell of $100 \times 100 \times 150$ [m^3]. A sparsity level of $p = 0.0007$ was selected and $Q = 408^2$. We chose identical targets with target reflection coefficient (per pixel) representing an aircraft similar to an Airbus A-320. An operating point of 0.1 was selected for the Radar Cross Section (RCS) and this parameter was varied in simulation from 0.01 to 1. Target velocities were normally distributed with $v_t \sim \mathcal{N}(200, 49)$ centered at 200 [m/s]. Rain intensity was random between 0-6 [mm/hr] and spatial correlation on the order of 1 [km^2]. Maximal clutter velocity was 30 [m/s] and targets were randomly placed within the clutter regions. A simple second order FIR line canceler was implemented (frequency response of $\sin^2 \frac{\omega T}{2}$) with the pass band centered at 200 [m/s]. The Swerling II noise model (Exponential) was used as a measurement noise model and the total energy budget was $\lambda_T = NQ$ pulses, i.e., N pulses transmitted at each grid stop.

Note that this model violates the assumption of single-pixel targets that we have used for the performance prediction analysis. However, as hypothesized earlier, we believe that the performance with clustered targets will not be negatively impacted as compared to the single-target scenario. Indeed, for the case where target returns add constructively to the measurement, clustered targets will actually increase the detectability index.

Based on M-ARAP we suggest the following measurement scheme: (1) define a coarse grid pattern; (2) use $N_1 < N$ pulses to measure each point on the coarse grid; (3) use M-ARAP to decide where and how to rescan the domain in a restricted fine grid; and (4) rescan the searched domain according to the pulse allocation of M-ARAP. An example of a single realization is given in Fig. 2.9.

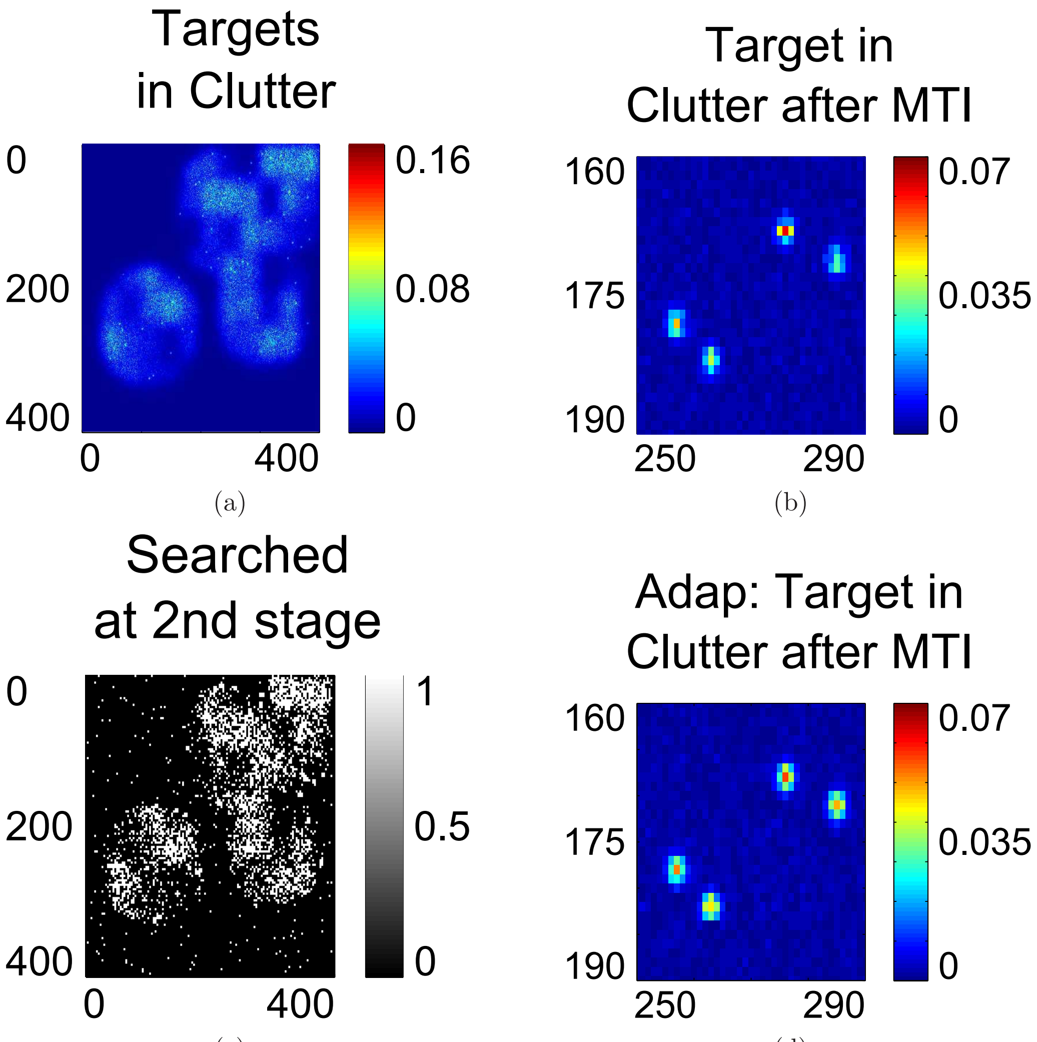


Figure 2.9: Moving target indication example. We set targets RCS to 0.1 and chose $N = 8$ and $N_1 = 5$. (a) A single realization of targets in clutter. Figures (b) and (d) zoom in on to the yellow rectangular to allow easier visualization of the improved estimation due to M-ARAP. (b) Portion of the estimated image when data was acquired using exhaustive search and MTI filtration. Figures (c) and (d) are due to M-ARAP search scheme where multi-scale was set to a coarse grid search of 3×3 pixels at the first stage. (c) Estimated ROI $\widehat{\Psi}$ that is searched on a fine resolution level on stage two. (d) Portion of the estimated image when data was acquired using M-ARAP.

2.7.1 MTI performance analysis

The first question addressed in implementing M-ARAP is how to choose N_1. One approach is to use the data shown in Fig. 5 in the work by Bashan et al. [11], where the optimal energy allocation at the first step is calculated for ARAP. In the simulations below we sweep over the range $\{4, 8, 16\}$ of N values. Performance was evaluated in terms of estimation MSE gain and area under the curve (AUC) of the false discovery rate (FDR) Q-ROC curve. The results are shown in Figure 2.10. Note that the fewer pulses used (e.g., $N = 4$) the more crucial it is to select N_1 appropriately.

We evaluate potential gains in both estimation MSE as well as performance of a false discovery rate type of target detection and localization. We used the estimator suggested in Section 2.6.1 and compared it to the CME implemented with exhaustive search, as given in (2.56). Results are seen in Fig. 2.11. Note that M-ARAP has nearly the same performance as measured by estimation gain as compared to ARAP (blue and green dotted lines), yet the number of measurements is significantly reduced (red solid line). For example, with an RCS of 1, the estimation gain differs by less than 1 dB, but M-ARAP uses only about 14% of the measurements.

We also compare false discovery rate and compare Q-ROC curves. Fig. 2.12 displays the two curves for both exhaustive and M-ARAP search. RCS value of 0.1 was selected and it is clearly seen that M-ARAP provides better detection performance for equivalent false discovery rates.

Finally, we note that although the assumption of single-pixel targets may not be validated in this application, the performance in terms of estimation gain and probability detection is still significantly better than an exhaustive search with many fewer measurements than with ARAP.

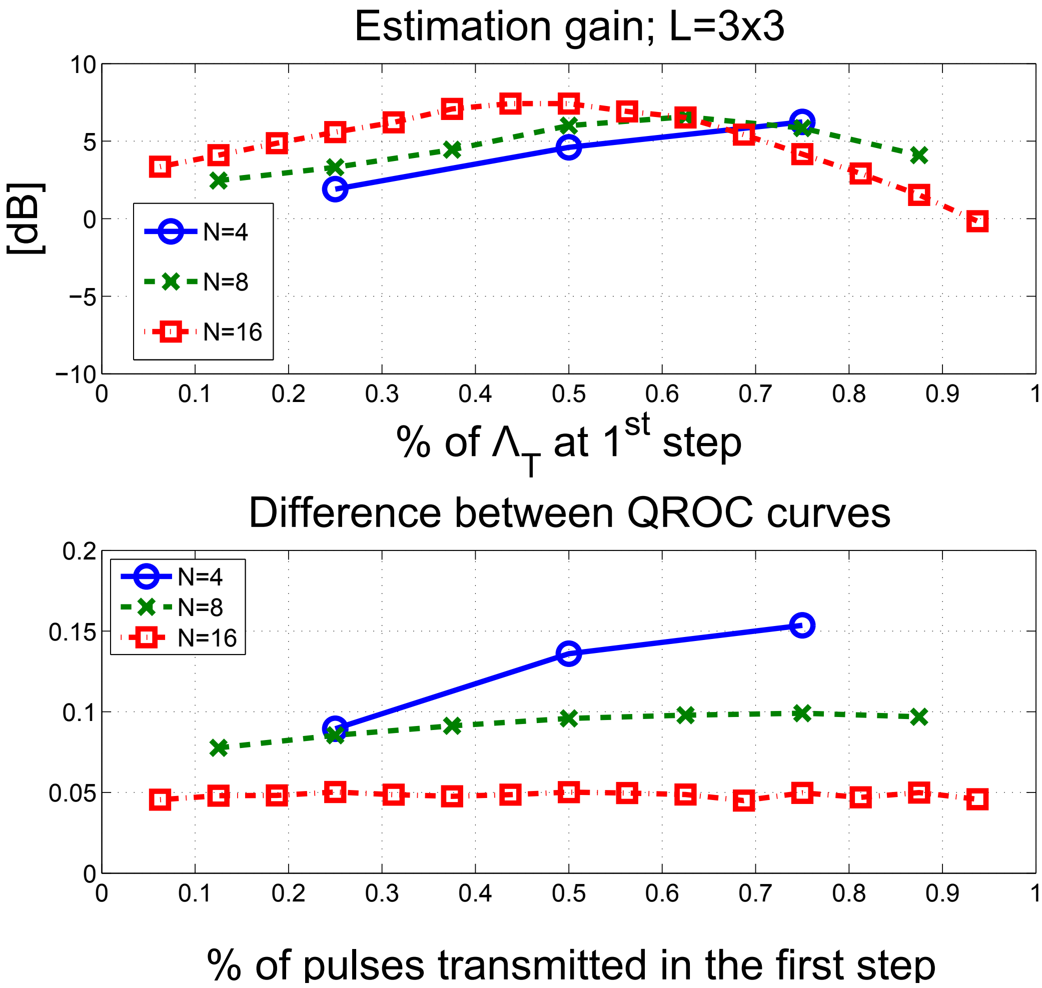


Figure 2.10: Simulated gain in estimation and detection performances as a function of N_1 the number of pulses used in the uniform search stage. The operating point of RCS=0.1 was selected. The upper plot displays gains in estimation MSE. Note that with $N = 16$ and N_1 equals 7 or 8 yields almost 8 [dB] gains in MSE. The lower plot shows difference in the area under the curve of an FDR test as a function of N_1. For $N = 8, 16$, the exhaustive search yield an almost optimal curve and there is less room for improvement

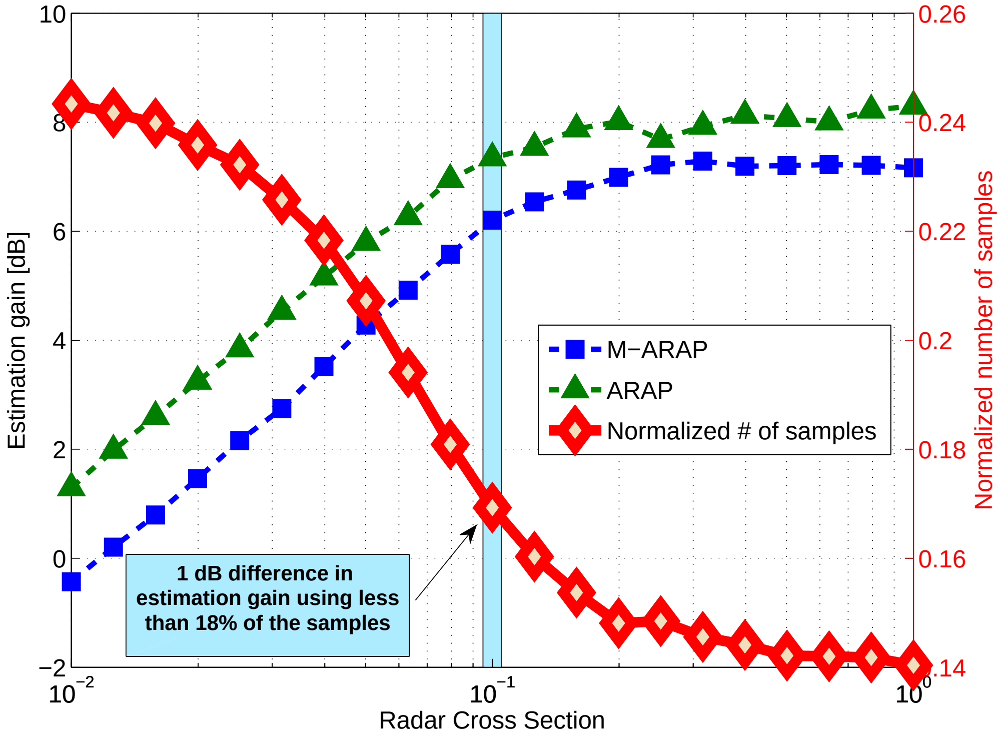


Figure 2.11: Simulated gain in estimation and the normalized number of measurement used by M-ARAP vs. targets radar cross section (RCS) coefficient. RCS is alias to signal to noise ratio or contrast since background scatter level was kept fixed. The solid curve with square markers and dashed curve with triangular markers represent estimation gains of M-ARAP and ARAP compared to an exhaustive search, respectively. The dash-dotted curve with diamond markers represent N^* the number of measurements used by M-ARAP divided by Q with the corresponding Y-axis values on the right hand side of the figure. For both M-ARAP and ARAP a total of four pulses per cell ($N = 4$) was selected as the energy budget of which three were used at the first stage ($N_1 = 3$) for all RCS values. Recall that for ARAP we have $N^* > 1$. Our results clearly illustrate that significant estimation gains can be obtained using M-ARAP with a fraction of the number of measurement required by ARAP.

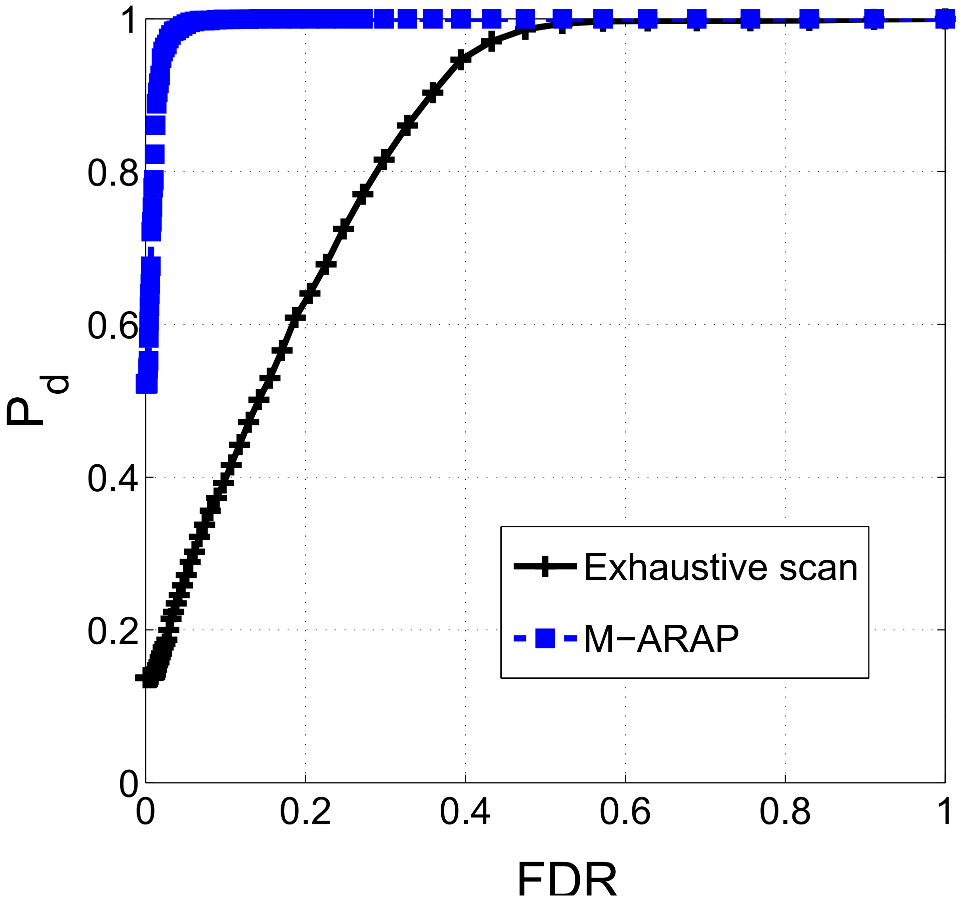


Figure 2.12: The two curves on the above figure represent an FDR detection test. One hundred runs in a Monte-Carlo simulation were used to generate each point on the curves. Radar cross section coefficient of 0.1 was selected, $N = 4$ (four pulses) was the overall energy budget, and $N_1 = 3$ was used in the first scan for M-ARAP. It is clearly evident that M-ARAP yield significantly better detection performance for equivalent false discovery rate levels.

2.8 Discussion and conclusions

In this chapter, we introduced the framework and problem formulation for adaptive resource allocation policies under total resource constraints. ARAP was shown to be an asymptotically optimal policy (in SNR) and optimal when compared to all two-stage policies that allocate uniformly at the first stage. However, ARAP is limited in several ways, including potentially prohibitive measurement requirements. In this chapter, we also proposed a multi-scale modification, called M-ARAP, that incorporating a coarse-to-fine scale search. We have established that M-ARAP reduces the number of required measurements for nearly equal gain in target search performance. Specific examples showed 9-15 [dB] gain in estimation performance, for detectability index $d = 5$, using less than 18% of the samples needed to perform an exhaustive search.

However, there are still many related directions of research that have yet to be explored. In Chapter III, we discuss the extension of ARAP to $T \gg 2$ stages and the potentially moving (dynamic) targets. In Chapter V, we introduce an application to target detection/tracking with SAR images where ARAP-like policies can be used to efficiently use computational resources.

CHAPTER III

Adaptive search for Sparse and Dynamic Targets under Resource Constraints

3.1 Introduction

Chapter II provided a framework for adaptive search for sparse targets. However, the model was restricted to the static case where targets remained stationary. In many interesting cases, targets might exhibit dynamic behavior, such as transitioning to neighboring cells, entering/leaving the scene, or being temporarily obscured. This chapter extends adaptive search to this domain by introducing

1. A time-varying target state and state dynamical model.

2. Oracle allocation policies that provide upper bounds on performance.

3. Allocation policies that can be computed efficiently and asymptotically approximate the oracle policies as either SNR or T gets large.

Consider measurements of the form

$$\boldsymbol{y} = \operatorname{diag}\left\{\sqrt{\boldsymbol{\lambda}}\right\} \operatorname{diag}\left\{\boldsymbol{\Theta}\right\} \boldsymbol{I}_{\Psi} + \boldsymbol{\varepsilon}, \tag{3.1}$$

where $\boldsymbol{\lambda}$ is a vector describing the resource allocation policy, $[\sqrt{\boldsymbol{x}}]_i$ denotes $\sqrt{x_i}$, $\operatorname{diag}\{\boldsymbol{x}\}$ is a square matrix with $[\operatorname{diag}\{\boldsymbol{x}\}]_{ii} = x_i$ and $[\operatorname{diag}\{\boldsymbol{x}\}]_{ij} = 0$ for $i \neq j$,

$\boldsymbol{\Theta}$ is a vector of target amplitudes, $\boldsymbol{I}_\Psi$ is a sparse vector of indicator variables, and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathrm{I}_{\mathrm{Q}\times\mathrm{Q}})$ where $\mathrm{I}_{\mathrm{Q}\times\mathrm{Q}}$ is an $(Q \times Q)$ identity matrix. In Chapter II, we introduced several policies for choosing $\{\lambda_i\}_{i=1}^{Q}$ in order to estimate $\boldsymbol{\Theta}$ and $\boldsymbol{I}_\Psi$ under the total resource constraint

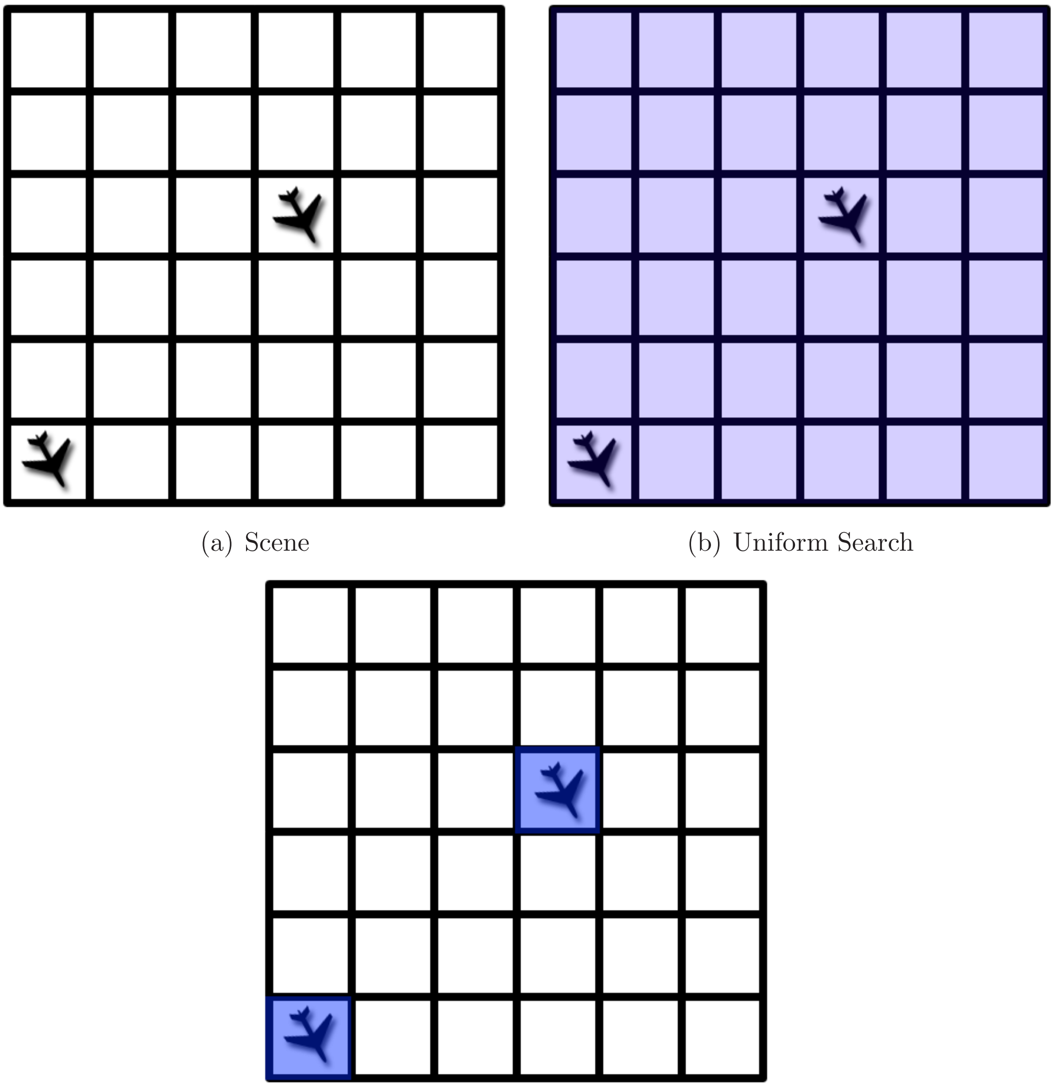
$$\sum_{t=1}^{2}\sum_{i=1}^{Q}\lambda_i(t) \leq \lambda_{total} \tag{3.2}$$

The baseline policy is to scan each location with an equal effort location, termed an 'exhaustive' or 'uniform' search. In order to avoid confusion from the dynamic programming literature, we will use the latter term (uniform search) to denote the baseline policy. If the locations of the targets were known a priori, then the oracle policy would allocate resources only to the locations containing targets. To illustrate the situation, in Figure 3.1 we have reproduced the figure from Chapter II that shows both of these policies in the context of an agile scanning radar. In Chapter II, we introduced the policies ARAP and M-ARAP in order to adaptively estimate the region of interest (ROI) using measurements taken across $T = 2$ stages of the form

$$\boldsymbol{y}(t) = \mathrm{diag}\left\{\sqrt{\boldsymbol{\lambda}(t)}\right\}\mathrm{diag}\left\{\boldsymbol{\Theta}\right\}\boldsymbol{I}_\Psi + \boldsymbol{\varepsilon}(t), \quad t = 1, 2. \tag{3.3}$$

It was shown that by using an adaptive policy, one can perform significantly better as compared to the uniform search. Benefits include near-optimal gains in estimation error and provable convergence to the true support of the ROI. Moreover, it was shown that by using a multi-scale policy, one could reduce the number of measurements with only minimal performance losses.

However, the algorithms presented in Chapter II are limited in several key ways. First, the optimization problem that used to derive these methods grows in a combinatorial fashion as a function of T and is not tractable for $T > 2$. Second, the



(a) Scene

(b) Uniform Search

(c) Oracle Search

Figure 3.1: In (a), a scene that we wish to scan is shown with two static targets. The standard policy, shown in (b) is to allocate equal effort to each cell individually. The oracle policy, shown in (c), is to allocate effort only to cells containing targets.

provable gains of ARAP and M-ARAP were restricted to the asymptotic regime, where the signal-to-noise ratio (SNR) was very large.

The work in this chapter alleviates both of these issues in the following manner. First, we consider applications that only have low to medium SNR levels, but we allow the number of stages to grow indefinitely. For example, consider two applications that repeatedly probe the scene: (a) an air traffic control (ATC) system that employs an adaptive radar system to continuously monitor a scene, and (b) a real-time tracking system that is limited by a finite amount of computational resources per iteration. Both of these applications share the property that as T gets large, the targets of interest are likely to transition to neighboring cells, enter the scene, or leave the scene. Figure 3.2 illustrates the time-varying problem. In order to compensate for these effects, we propose a simple dynamic state model and let the indicator functions of the ROI change over time. In particular, we now consider the problem:

$$\boldsymbol{y}(t) = \operatorname{diag}\left\{\sqrt{\boldsymbol{\lambda}(t)}\right\} \operatorname{diag}\left\{\boldsymbol{\Theta}(t)\right\} \boldsymbol{I}_{\Psi}(t) + \boldsymbol{\varepsilon}(t), \tag{3.4}$$

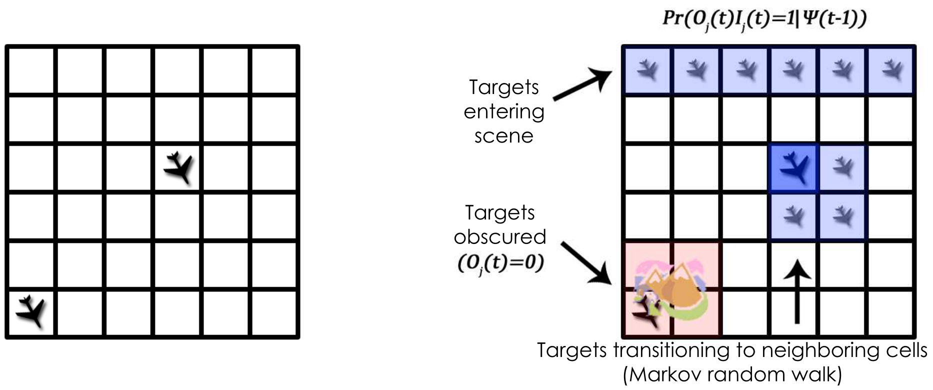
for $t = 1, 2, \ldots, T$. Resource constraints per stage are given by

$$\sum_{i=1}^{Q} \lambda_i(t) \leq \lambda_{total}(t), \quad t = 1, 2, \ldots, T. \tag{3.5}$$

In general the constraints $\{\lambda_{total}(t)\}_{t=1}^{T}$ and the number of stages T may be a design parameter. However, in this chapter, we consider the simpler situation where these constraints are known beforehand. [1] In general, sequential resource allocation problems can be solved by dynamic programming (DP.) The basic problem in DP is to minimize an additive cost of the form

$$E\left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right\}, \tag{3.6}$$

[1] It is also possible that T might be a random variable such that the number of stages of measurements may be stopped once ‘sufficient’ signal has been gathered. This thesis does not consider these issues, though they may be studied in future work.


Pr(O_j(t)I_j(t)=1|Ψ(t-1))
Targets entering scene
Targets obscured (O_j(t)=0)
Targets transitioning to neighboring cells (Markov random walk)

(a) Scene $\Psi(t-1)$ at time, $t-1$

(b) Prior probabilities at time, t

Figure 3.2: In (a), a scene that we wish to scan is shown with two dynamic targets at time, $t-1$. In (b), we show the prior probabilities for the targets. The target in the bottom-left corner is obscured at time, t. The target in the middle can transition to neighboring cells with some probability, modeled as a Markov random walk. Finally, targets may enter the scene along the top border with some small probability.

where x_k is the current state at stage k, u_k is the input at stage k, w_k is random noise, and $g_k(\cdot)$ is some cost function. In general, $\{u_k\}_k$ will be dependent on the random states x_k, so DP tends to minimize the expected cost over policies, $\mu_k(x_k)$. Bertsekas provides a key principle in DP, referred to as the 'principle of optimality,' which states that these complex optimizations can be solved by using backwards induction [14]. In practice, the state is usually observed through noisy observations, which greatly complicates the optimization problem. On the other hand, if the state is replaced by the so-called 'belief state' that characterizes the posterior distribution of the observed state, then the partially observable problem can be converted back into a fully observable DP. Nevertheless, optimal solutions to DP problems often require approximate solutions in order to be computationally tractable, especially as the size of the state grows. An in-depth discussion of approximate methods is outside the scope of this chapter, but the interested reader should see Chapter 6 in [14].

Chong et. al. [28] describe a set of related problems for formulating adaptive sensing as a partially observable Markov decision problem (POMDP.) They provide an excellent survey of optimal and approximate methods for solving POMDPs. Morevoer, they show that using non-myopic policies (i.e., policies that trade off current benefits for long-term performance gains) can lead to significant gains over standard policies. These strategies rely on approximating the so-called Q- function of the belief state $\boldsymbol{B}(t)$ and possible actions $a \in \mathcal{A}$, where

$$Q_{T-t}(\boldsymbol{B}(t), a) = r(\boldsymbol{B}(t), a) + E[V^*_{T-t-1}(\boldsymbol{B}(t))|\boldsymbol{B}(t), a] \tag{3.7}$$

and

$$V^*_T(\boldsymbol{B}(0)) = \max_a \big(r(\boldsymbol{B}(t), a) + E[V^*_{T-1}(\boldsymbol{B}(0))|\boldsymbol{B}(0), a]\big) = \max_a Q_{T-t}(\boldsymbol{B}(t), a) \tag{3.8}$$

In general, estimating the Q-function is intractable. However, there are many approximate methods, including

- Q-learning: A reinforcement learning approach based on estimating the Q-function from multiple trajectories of the random process.
- Relaxation of the optimization problem or of the state space.
- Parametric approximations.
- Rollout policies: Assume that at stage $\tau > t$, we use a known base policy, $\lambda^{(base)}$. Then choose the action at stage t that optimizes the Q-function where the remaining $T - t$ stages are chosen according to $\lambda^{(base)}$.

The work in this chapter considers problems similar to both POMDPs and DPs. A computationally simple alternative to the optimal and approximate solutions is provided, which is additionally shown to be asymptotically optimal through empirical evidence at a fraction of the search complexity of other algorithms. In Section 3.5, we also provide a comparison of the proposed policy to an approximate POMDP solution, and show that performance is similar although at much reduced computational cost.

This chapter introduces the Dynamic Adaptive Resource Allocation Policy (D-ARAP) that

- Optimizes an appropriate time-varying cost function subject to a per-stage budget. For example, in an ATC system, the allocation budget is the total time it takes for the radar to complete one complete cycle.
- Accounts for dynamic targets through a dynamic state model.
- Provides significant gains over a uniform policy that allocates the budget evenly over all potential target locations.
- Outperforms a greedy policy in terms of robustness and convergence.
- Is near-optimal as $T \to \infty$.
- Accomplishes these goals with low computational cost.

The rest of this chapter is organized as follows. We provide notation in Section 3.2. We formalize the problem in Section 3.3 and provide the adaptive sensing policy in Section 3.4. Performance analysis is given in Section 3.5. In Section 3.6, we conclude and point to future work. Moreover, Sections 3.7 and 3.8 provide appendices with extended details of the policies and approximations given in this chapter.

3.2 Notation

- Q - Number of cells in search space.
- T - Number of stages (time).
- t - Current time, $t \in 1, 2, \ldots, T$.
- $\mathcal{X} = \{1, 2, \ldots, Q\}$ - Discrete space of Q cells.
- $\Psi(t) \subseteq \mathcal{X}$ - Subset of $\mathcal{X}$ referred to as the ROI at time t.
- $I_i(t)$ - indicator function of the ROI such that

$$I_i(t) = \begin{cases} 1, & i \in \Psi(t) \\ 0, & \text{Otherwise} \end{cases} \quad \text{for } i \in \{1, 2, \ldots, Q\}. \tag{3.9}$$

- $\widehat{\Psi}(t) \subseteq \mathcal{X}$ - Estimated ROI, learned from measurements up until time t.
- $\lambda_i(t) \in [0, \lambda_{total}(t)]$ - Search effort allocated to cell i at stage t.
- $\lambda_{total}(t) = \sum_{i=1}^{Q} \lambda_i(t)$ - Total search effort allocated at time t.
- $\boldsymbol{Y}(t) = \{\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(t)\} \in \mathbf{R}^N$ - Set of measurements.
- $\kappa(t) \in [0, 1]$ - Percentage of resource budget to be used for exploration of $\mathcal{X}$ rather than exploitation of $\hat{\Psi}(t)$.

3.2.1 For dynamic target state model

- α - Target birth probability.
- β - Target death probability.
- π_0 - Probability that target remains in cell ($\pi_0 = 1$ yields static targets).

- $G(i) \subseteq \mathcal{X}$ - Set of neighbors to cell i (i.e., locations where target may transition to from cell i).
- $\tilde{N}(t)$ - Number of targets in scene at time t.
- $\hat{\theta}_i(t)$ - Estimate of $\theta_i(t)$ at time t given $\boldsymbol{Y}(t-1)$.
- $\sigma_i^2(t)$ - Estimated variance of $\theta_i(t)$ at time t given $\boldsymbol{Y}(t-1)$.
- $p_i(t) = \Pr(I_i(t) = 1|\boldsymbol{Y}(t-1))$ - Posterior probability of target existence.
- $\mathbf{B}(t) = \left\{p_i(t), \hat{\theta}_i(t), \sigma_i^2(t)\right\}_{i=1}^{Q}$ - Belief state at stage t.
- $O_i(t)$ - indicator function of the observability at time t such that

$$O_i(t) = \begin{cases} 1, & \text{Location } i \text{ is observable} \\ 0, & \text{Otherwise} \end{cases} \quad \text{for } i \in \{1, 2, \ldots, Q\}. \tag{3.10}$$

3.3 Problem formulation

Consider a space $\mathcal{X} = \{1, 2, \ldots, Q\}$ containing Q cells equipped with a probability measure P, and a region of interest (ROI), $\Psi \subset \mathcal{X}$. We assume that Ψ will be a randomly selected small subset of $\mathcal{X}$, though its definition is application specific. In radar target localization, the ROI is the collection of all cells containing targets and target related phenomena, e.g., target shadows. As described in Chapter II, in early detection of breast cancer, where tumor boundaries are poorly defined, the ROI may be defined as the collection of all cells containing targets (a tumor) plus some neighboring cells. In this chapter, we generalize the formulation from Chapter II to account for a time-varying ROI by making $\Psi = \Psi(t)$ a function of time.

3.3.1 Dynamic state model

At time $t \in \{1, 2, \ldots, T\}$, consider tracking $N(t)$ targets. For each target, $n \in N(t)$, define

$$s^{(n)}(t) \in \mathcal{X} = \{1, 2, \ldots, Q\}, \tag{3.11}$$

and

$$x^{(n)}(t) \in \mathbb{R}, \tag{3.12}$$

to be the n-th target's position and amplitude, respectively, at time t. Let

$$\boldsymbol{S}(t) = \{s^{(1)}(t), s^{(2)}(t), \ldots, s^{(n)}(t)\}, \tag{3.13}$$

and

$$\mathbf{X}(t) = \{x^{(1)}(t), x^{(2)}(t), \ldots, x^{(n)}(t)\}. \tag{3.14}$$

Then the state at time t is defined as

$$\xi(t) = \{\boldsymbol{S}(t), \mathbf{X}(t), N(t)\} \tag{3.15}$$

At time $t = 1$, the prior distribution for $\xi(t)$ is given by

$$N(1) \sim \text{Binomial}(Q, p), \tag{3.16}$$

$$s^{(n)}(1) \sim \text{Uniform}\{1, 2, \ldots, Q\}, \qquad \text{for } n = 1, 2, \ldots, N(1), \tag{3.17}$$

$$x^{(n)}(1) \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2), \qquad \text{for } n = 1, 2, \ldots, N(1). \tag{3.18}$$

We propose a simple dynamic model for $\xi(t+1)$ given the current state $\xi(t)$ that includes transition, birth, and death probabilities. This can easily be generalized to other situations, such as those given in Section 3.8.5. In words, this model assumes

- A target leaves the scene with probability α.
- A single target enters the scene with probability β.

- A target that is already in the scene at location i
  - remains in the same location with probability π_0.
  - moves to a neighboring cell with probability $(1-\pi_0)/|G(i)|$, where $G(i)$ is the set of neighboring cells.
  - has amplitude that changes with zero-mean Gaussian noise.

This model can be defined mathematically in the following manner. Define the sets

$$G(i) = \{j \in \mathcal{X} : j \text{ is a neighbor of location } i\}, \tag{3.19}$$

$$H(i) = G(i) \cup \{i\}. \tag{3.20}$$

For $n = 1, 2, \ldots, N(t)$

$$\Pr\left(s^{(n)}(t+1) = i | s^{(n)}(t) = j\right) = \begin{cases} \alpha, & i = 0, j \neq 0 \\ (1-\alpha)\pi_0, & i = j, j \neq 0 \\ \dfrac{(1-\alpha)(1-\pi_0)}{|G(j)|}, & i \in G(j), j \neq 0 \\ 0, & \text{else} \end{cases} \tag{3.21}$$

where α is the probability that a target is removed from the scene and π_0 is the probability that a target remains in the same location. Let a single target be added to the scene with probability β so that

$$\Pr(N(t+1) = n | N(t)) = \begin{cases} \beta, & n = N(t)+1 \\ 1-\beta, & n = N(t) \end{cases} \tag{3.22}$$

Given $N(t+1) = N(t)+1$, define the distribution of the added target as

$$s^{(N(t)+1)}(t+1) \sim \text{Uniform}\{1, 2, \ldots, Q\} \tag{3.23}$$

$$x^{(N(t)+1)}(t+1) \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2) \tag{3.24}$$

Let the target amplitudes for targets in the current scene[2] evolve according to

$$x^{(n)}(t+1) = x^{(n)}(t) + \mathcal{N}(0, \Delta_\theta^2) \tag{3.26}$$

for all $n = 1, 2, \ldots, N(t)$ and $s^{(n)}(t) \neq 0$. Section 3.7 provides analysis of the choice of α and β to ensure that the sparsity of the state remains the same, in the sense that the expected number of targets in the scene remains constant.

3.3.2 Observation model

For $i = 1, 2, \ldots, Q$ and $t = 1, 2, \ldots, T$, we assume a measurement model of the form

$$y_i(t) = \sqrt{\lambda_i(t)} \sum_{n:s^{(n)}(t)=i} x^{(n)}(t) + \varepsilon_i(t) \tag{3.27}$$

where $\varepsilon_i(t) \sim \mathcal{N}(0, \sigma^2)$ and $\lambda_i(t)$ is a design parameter. Define the indicator variables

$$I_i(t) = \begin{cases} 1, & \exists n : s^{(n)}(t) = i \\ 0, & \text{else} \end{cases} \tag{3.28}$$

for $i = 1, 2, \ldots, Q$. Since this chapter considers sparse scenarios, the expected number of targets $E[N(t)] = pQ$ is much smaller than the size of the scene, so that $pQ \ll Q$. When targets are equally likely to be located in any of the Q locations, the prior

[2] Note that $N(t)$ is the number of targets that have been tracked up until time t. However, it is NOT necessarily the number of targets in the scene at time t. This quantity, $\tilde{N}(t)$ is defined as

$$\tilde{N}(t) = \sum_{n=1}^{N(t)} \mathbf{1}_{\{s^{(n)}(t) \neq 0\}} \tag{3.25}$$

where $\mathbf{1}_{\{A\}}$ is the indicator variable of event A.

probability a cell is occupied by at most one target is given by

$$\begin{aligned}\Pr(|\{n : s^{(n)}(t) = i\}| \leq 1) &= \Pr(|\{n : s^{(n)}(t) = i\}| = 0) + \Pr(|\{n : s^{(n)}(t) = i\}| = 1)\\ &= \left(\frac{Q-1}{Q}\right)^{pQ} + \frac{1}{Q}\left(\frac{Q-1}{Q}\right)^{pQ-1}\\ &= \left(\frac{Q-1}{Q}\right)^{pQ-1}\left(\frac{Q-1}{Q} + \frac{1}{Q}\right) = \left(\frac{Q-1}{Q}\right)^{pQ-1}\end{aligned} \tag{3.29}$$

where it is assumed that $E[N(t)] = pQ$ is an integer. For large Q and small p, this probability is very close to one. For example, when $Q = 1,000$ and $p = 0.01$, we have $\Pr(|\{n : s^{(n)}(t) = i\}| \leq 1) = 0.991$. Thus, for policy planning, we make the simplifying assumption that

$$|\{n : s^{(n)}(t) = i\}| \leq 1, \tag{3.30}$$

for $i = 1, 2, \ldots, Q$. Then we define the cell-by-cell target amplitudes as

$$\theta_i(t) = \begin{cases} x^{(n)}(t), & \text{if } s^{(n)}(t) = i \\ 0, & \text{else} \end{cases} \tag{3.31}$$

Combining equations (3.27), (3.28) and (3.31), we have the familiar equation

$$y_i(t) = \sqrt{\lambda_i(t)} I_i(t)\theta_i(t) + \varepsilon_i(t). \tag{3.32}$$

In many cases, targets in certain locations may not be observable at certain times. For example, locations within the null of a radar beam cannot be observed until the radar platform moves. Define the modified measurement model as

$$y_i(t) = \sqrt{\lambda_i(t)} I_i(t) O_i(t)\theta_i(t) + \varepsilon_i(t), \tag{3.33}$$

where

$$O_i(t) = \begin{cases} 1, & \text{Location } i \text{ is observable} \\ 0, & \text{else} \end{cases} \qquad i = 1, 2, \ldots, Q. \tag{3.34}$$

In this chapter, it is assumed that the indicators of observability, $\{O_i(t)\}_{i,t}$, are known a priori and do not have to be estimated.

In general, it is a difficult, if not intractable, problem to exactly estimate the extremely high-dimensional posterior distribution $f(\xi(t)|\boldsymbol{Y}(t-1))$ that is needed to calculate $\Pr(I_i(t) = 1|\boldsymbol{Y}(t-1))$. On the other hand, Section 3.8 provides an appendix for efficiently approximating the posterior distribution. Moreover, other signal processing algorithms, such as the particle filter, extended Kalman filter, or the Unscented Kalman filter, could be used to approximate the distributions with varying tradeoffs between accuracy and computational burden.

3.3.3 Resource constraints in sequential experiments

Consider a sequential experiment where we observe measurements $\mathbf{y}(t) : \mathcal{X} \rightarrow \mathbf{R}^Q$ for $t = 1, 2, \ldots, T$. Let $\lambda_i(t) \geq 0$ denote the search effort allocated to cell i at time t, under total budget constraints given by (3.5) where the constraints $\{\lambda_{total}(t)\}_{t=1}^T$ are assumed to be known. We consider mappings of $\{\lambda_i(t)\}_{i=1}^Q$ from past observations $\mathbf{Y}(t-1) \triangleq \{\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(t-1)\}$. The choice of $\{\lambda_i(t)\}_{i,t}$ is called a search policy. We focus here on deterministic mappings of λ, although a more general random mapping could also be incorporated into our framework. We assume that a sample's 'quality' is an increasing function of the allocated effort to the associated cell, e.g. measured in terms of Fisher information or inverse variance. In general, effort might be computing power, complexity, cost, or energy that is allocated to probing a particular cell location.

3.4 Search policy for dynamic targets under resource constraints

3.4.1 Related work

In Chapter II and the work by Bashan et al. [11] we introduced the following cost function

$$J_{ARAP}(\lambda) = E\left[\sum_{i=1}^{Q} \frac{\nu I_i + (1-\nu)(1-I_i)}{\lambda_i(1)+\lambda_i(2)}\right]. \tag{3.35}$$

Minimizing (3.35) over $\lambda_i(t)$ under the total energy constraint

$$\sum_{i=1}^{Q}\sum_{t=1}^{T} \lambda_i(t) = \lambda_T \tag{3.36}$$

yielded the search policy ARAP, in the case where $T = 2$ and targets were static, i.e., $I_i(t) = I_i$. Chapter II discusses properties of ARAP and its multi-scale modification, M-ARAP.

This chapter wishes to generalize search policies in two major directions: (1) policies with $T \gg 2$ stages; and (2) dynamic targets where the target indicators and amplitudes change with time, so that $I_i(t) \neq I_i$ with positive probability. The most straightforward extension of the cost function in equation (3.35) to $T > 2$ is

$$J_{ARAP}(\lambda; T) = E\left[\sum_{i=1}^{Q} \frac{\nu I_i + (1-\nu)(1-I_i)}{\sum\limits_{t=1}^{T} \lambda_i(t)}\right]. \tag{3.37}$$

where $\{\lambda_i(t)\}_{i,t}$ is optimized under the total energy constraint

$$\sum_{t=1}^{T}\sum_{i=1}^{Q} \lambda_i(t) = \sum_{t=1}^{T} \lambda_{total}(t) = \lambda_T. \tag{3.38}$$

In contrast to the 2-stage problem, this cost function cannot be simply optimized for $T > 2$. Wei and Hero [90] provide an alternative way to extend ARAP-like policies to $T > 2$. Although they consider the static case where $I_i(t) = I_i$ and $\theta_i(t) = \theta_i$, their methodology provides useful insights for the work in this chapter. In particular, they show that minimizing the mean squared error (MSE) over Ψ:

$$E\left[\sum_{i=1}^{Q} I_i(\hat{\theta}_i - \theta_i)^2\right], \tag{3.39}$$

where the expectation is taken over I_i, θ_i, and $\boldsymbol{Y}(T)$, is done by letting

$$\hat{\theta}_i(T) = E[\theta_i | I_i = 1, \boldsymbol{Y}(T)]. \tag{3.40}$$

Using this estimator, the cost in equation (3.39) can be shown to be proportional to

$$E\left[\sum_{i=1}^{Q} \frac{\Pr(I_i = 1|\boldsymbol{Y}(T))}{\sigma^2/\sigma_\theta^2 + \sum\limits_{t=1}^{T} \lambda_i(t)}\right]. \tag{3.41}$$

Moreover, they show that the covariance of the CME estimator can be computed recursively as

$$\sigma_i^2(t) = \left(\frac{1}{\sigma_i^2(t-1)} + \frac{\lambda_i(t)}{\sigma^2}\right)^{-1} \tag{3.42}$$

$$= \sigma^2\left(\frac{\sigma^2}{\sigma_i^2(t-1)} + \lambda_i(t)\right)^{-1} \tag{3.43}$$

so that

$$\frac{\sigma^2}{\sigma_i^2(t)} = \frac{\sigma^2}{\sigma_i^2(t-1)} + \lambda_i(t) \tag{3.44}$$

Thus, equation (3.41) reduces to

$$E\left[\sum_{i=1}^{Q} \frac{\Pr(I_i = 1|\boldsymbol{Y}(T))}{\sigma^2/\sigma_i^2(T) + \lambda_i(T)}\right]. \tag{3.45}$$

Note that both $\Pr(I_i = 1|\boldsymbol{Y}(T))$ and $\sigma_i^2(T)$ depend on the previous allocations $\{\lambda_i(t)\}_{t<T}$. Furthermore, if the belief state is defined as

$$\mathbf{B}(t) = \left\{\Pr(I_i = 1|\boldsymbol{Y}(T)), \hat{\theta}_i(T), \sigma_i^2(T)\right\}_{i=1}^{Q}, \tag{3.46}$$

then an optimal sequential effort allocation policy can be determined by formulating the problem as a dynamic program (DP).

Table 3.1: Parameters for cost function for various target amplitude models for $I_i(t) = I_i$ and cost given by equation (3.48)

Model	Variance of CME, $\sigma_i^2(T\|T)$	$c_i(T)$	Recursive form for $c_i(T)$
$\theta_i(t) = \mu_i$ $\mu_i \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$	$\sigma^2 \left(\frac{\sigma^2}{\sigma_\theta^2} + \sum_{t=1}^{T} \lambda_i(t) \right)^{-1}$	$\frac{\sigma^2}{\sigma_\theta^2} + \sum_{t=1}^{T-1} \lambda_i(t)$	$c_i(T-1) + \lambda_i(t)$
$\theta_i(t) \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$	$\sigma^2 \left(\frac{\sigma^2}{\sigma_\theta^2} + \lambda_i(t) \right)^{-1}$	$\frac{\sigma^2}{\sigma_\theta^2}$	$c_i(T-1)$
$\theta_i(t) = \theta_i(t-1) + \delta_i(t)$ $\delta_i(t) \sim \mathcal{N}(0, \Delta^2)$ $\theta_i(0) \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$	$\sigma^2 \left(\frac{\sigma^2}{\sigma_i^2(T\|T-1)} + \lambda_i(t) \right)^{-1}$ where $\sigma_i^2(t+1\|t)$ $= \sigma_i^2(t\|t) + \Delta^2$	$\frac{\sigma^2}{\sigma_i^2(T\|T)}$	$\left(\frac{\Delta^2}{\sigma^2} + \frac{1}{c_i(T-1)} \right)^{-1}$

3.4.2 Proposed cost function

The cost function in equation (3.45) can be generalized to cases where $\theta_i(t) \neq \theta_i$ for all t. For example, consider the cost function

$$J(\lambda; T) = E\left[\sum_{i=1}^{Q} \frac{I_i(t)}{c_i(t) + \lambda_i(t)} \right], \tag{3.47}$$

where $c_i(t)$ are non-negative constants that may depend on $\boldsymbol{Y}(T-1)$ and $\{\lambda_i(t)\}_{t<T}$. When $\hat{\theta}_i(t)$ is given by the conditional mean estimator for $\theta_i(t)$, this cost function corresponds to the MSE within the ROI for many models of $\theta_i(t)$. Table 3.1 provides several of these models and their corresponding expressions for $c_i(t)$. It should be noted that in all of these cases, $c_i(t)$ is a function of the belief state $\mathbf{B}(t)$ given by equation (3.46).

Although the cost function given by equation (3.47) does not correspond exactly to the MSE within the ROI when $I_i(t) \neq I_i$ with some positive probability, it is hypothesized that the cost function will be useful in many cases. Indeed, conditioned on the ROI $\Psi(t)$, this cost function is exactly the MSE within $\Psi(t)$ in the cases described in Table 3.1. In other cases where $\Psi(t)$ is known with high confidence (i.e., as $t \to \infty$ or for high SNR), this cost function is assumeted to be close to the MSE within the ROI.

In dynamic cases, the user might be interested in estimation error for all stages (rather than just at stage T). Thus, we modify the cost function as

$$J(\lambda; T) = E\left[\sum_{t=1}^{T} \gamma(t) \sum_{i=1}^{Q} \frac{I_i(t)}{c_i(t) + \lambda_i(t)}\right], \tag{3.48}$$

where $\{\gamma(t)\}_{t=1}^{T}$ is a set of coefficients describing the relative weights of cost as a function of stage. Note that equation (3.47) is a special case of equation (3.48) with $\gamma(T) = 1$ and $\gamma(t) = 0$ for $t < T$. In this chapter, we generally consider exponential weights on the cost so that for a T-stage problem, we have

$$\gamma(t; T) = (\gamma_0)^{T-t}, \quad 0 < \gamma_0 < 1. \tag{3.49}$$

Future work will look at the sensitivity of the performance of our policies to the choice of $c_i(t)$ and $\gamma(t)$.

3.4.3 Oracle policies

Next, we provide an achievable lower bound on the cost function given by equation (3.48) in the case where $\gamma(T) = 1$ and $\gamma(t) = 0$ for $t < T$. This lower bound will specify the performance of an oracle that has access to the target state, called the omniscient policy.

Lemma III.1. *The cost function given by equation (3.48) is lower bounded by*

$$J(\lambda; t) \geq \frac{|\Psi(t)|^2}{\lambda_{total}(t) + \sum\limits_{i \in \Psi(t)} c_i(t)}. \tag{3.50}$$

This lower bound is achievable by an omniscient policy defined as

$$\lambda_i^{omn}(t) = \begin{cases} \dfrac{\lambda_{total}(t) + \sum\limits_{j \in \Psi(t)} c_j(t)}{|\Psi(t)|} - c_i(t), & i \in \Psi(t) \\ 0, & i \notin \Psi(t). \end{cases} \tag{3.51}$$

Proof. For a non-negative series, $\{I_i(t)\}$, the Cauchy-Schwartz inequality provides

$$\begin{aligned} \left(\sum_{i=1}^{Q} \frac{I_i(t)}{c_i(t) + \lambda_i(t)}\right) & \left(\sum_{i=1}^{Q} c_i(t) I_i(t) + \lambda_i(t)\right) \\ & \geq \left(\sum_{i=1}^{Q} \sqrt{\frac{I_i(t)}{c_i(t) + \lambda_i(t)}} \sqrt{c_i(t) I_i(t) + \lambda_i(t)}\right)^2 \\ & = \left(\sum_{i \in \Psi(t)} \sqrt{\frac{1}{c_i(t) + \lambda_i(t)}} \sqrt{c_i(t) + \lambda_i(t)}\right)^2 \\ & = |\Psi(t)|^2. \end{aligned} \tag{3.52}$$

Moreover, we note that

$$\sum_{i=1}^{Q} c_i(t) I_i(t) + \lambda_i(t) = \lambda_{total}(t) + \sum_{i \in \Psi(t)} c_i(t) \tag{3.53}$$

Combining the definition of the cost function in equation (3.48) with equations (3.52) and (3.53), we get

$$J(\lambda; t) \geq \frac{|\Psi(t)|^2}{\lambda_{total}(t) + \sum\limits_{i \in \Psi(t)} c_i(t)}. \tag{3.54}$$

To prove the second part of the lemma, note that

$$\begin{aligned} J(\lambda^{omn}; t) = \sum_{j=1}^{Q} \frac{I_j(t)}{c_j(t) + \lambda_j^{omn}(t)} &= \sum_{j \in \Psi(t)} \frac{1}{c_j(t) + \frac{\lambda_{total}(t) + \sum_{i \in \Psi(t)} c_i(t)}{|\Psi(t)|} - c_j(t)} \\ &= \sum_{j \in \Psi(t)} \frac{|\Psi(t)|}{\lambda_{total}(t) + \sum\limits_{i \in \Psi(t)} c_i(t)} \\ &= \frac{|\Psi(t)|^2}{\lambda_{total}(t) + \sum\limits_{i \in \Psi(t)} c_i(t)}. \end{aligned} \tag{3.55}$$

□

In comparison to the omniscient policy, we may consider the uniform search as a baseline policy, defined as

$$\lambda_i^{uni}(t) = \frac{\lambda_{total}(t)}{Q}, \tag{3.56}$$

which achieves a cost

$$J(\lambda^{uni}; t) = \sum_{i=1}^{Q} \frac{I_i(t)}{c_i(t) + \lambda_{total}(t)/Q} = \sum_{i \in \Psi(t)} \frac{Q}{Qc_i(t) + \lambda_{total}(t)} \tag{3.57}$$

Assuming symmetry within the ROI, it may be illustrative to assume that $c_i(t) = c_0$ for $i \in \Psi(t)$. Then we have

$$J(\lambda^{uni}; t) = \frac{Q|\Psi(t)|}{Qc_0 + \lambda_{total}(t)} \tag{3.58}$$

We then define the gain of a policy with respect to the uniform search as

$$\Gamma(\lambda; t) = -10 \log \frac{J(\lambda; t)}{J(\lambda^{uni}; t)} \tag{3.59}$$

In particular, the gain of the omniscient policy when $c_i(t) = c_0$ in the ROI is given by

$$\begin{aligned} \Gamma(\lambda^{omn}; t) &= -10 \log \left(\frac{|\Psi(t)|^2}{|\Psi| c_0 + \lambda_{total}(t)} \cdot \frac{Qc_0 + \lambda_{total}(t)}{Q|\Psi|} \right) \\ &= -10 \log \frac{|\Psi(t)|}{Q} - 10 \log \frac{Qc_0 + \lambda_{total}(t)}{|\Psi| c_0 + \lambda_{total}(t)} \\ &\leq -10 \log \frac{|\Psi(t)|}{Q} \end{aligned} \tag{3.60}$$

The last inequality illustrates that when $c_0 \neq 0$, we actually incur a penalty in the gain of the omniscient policy since $|\Psi(t)| \ll Q$ by assumption. For $c_i(t)$ chosen according to Table 3.1, this penalty can be interpreted as the loss incurred by noisy measurements. In asymptotic regimes where SNR → ∞, one would expect that $c_i(t) \ll \lambda_i(t)$ so that the loss is negligible. We'd also like to point out that when $c_i(t) = 0$ for all i and t, the cost function still minimizes the probability of error for a deterministic $\theta_i(t) = \mu_\theta$ as shown in the work by Bashan [9]. Thus the choice of $c_i(t)$ also influences the tradeoff between detection and estimation of targets within the ROI.

The omniscient policy is not actually attainable, even asymptotically, as long as there is a non-zero probability that targets may enter, leave, or transition from time $t-1$ to t. On the other hand, we may consider the policy that optimizes equation (3.48) given the previous state

$$\xi(t-1) = \{I_i(t-1), \theta_i(t-1)\}_{i=1}^{Q}. \tag{3.61}$$

Assuming that $c_i(t) = c_0 \ll \lambda_{total}(t)$ for all $i \in \mathcal{X}$, it can be shown that solution of this optimization problem, which we term the 'semi-omniscient policy,' is given by

$$\lambda_i^{semi}(t) = \frac{\lambda_{total}(t)\sqrt{\Pr(I_i(t) = 1|\Psi(t-1))}}{\sum\limits_{j=1}^{Q}\sqrt{\Pr(I_j(t) = 1|\Psi(t-1))}} \tag{3.62}$$

Under the assumptions given in Section 3.8, it is shown in equation (3.150) that the cost of the semi-omniscient policy is

$$J(\lambda^{semi}; t) = \frac{\left[\sqrt{Q\beta} + |\Psi(t)|\sqrt{(1-\alpha)}\left(\sqrt{\pi_0} + \sqrt{G(1-\pi_0)}\right)\right]^2}{\lambda_{total}(t)}, \tag{3.63}$$

where α is the death probability, β is the birth probability, π_0 is the probability of a target remaining in the same location, and G is the number of neighboring location that a target may transition to from time t to time $t+1$. It can be seen that in the static case when $\pi_0 = 1$, $\alpha = \beta = 0$, we get the same cost as the omniscient policy:

$$J(\lambda^{semi}; t)|_{\pi_0=1,\alpha=\beta=0} = \frac{|\Psi(t)|^2}{\lambda_{total}(t)} \tag{3.64}$$

Moreover, in cases where $\alpha, \beta \approx 0$, we have

$$J(\lambda^{semi}; t)|_{\alpha,\beta\approx 0} = \frac{|\Psi(t)|^2}{\lambda_{total}(t)}\left(\sqrt{\pi_0} + \sqrt{G(1-\pi_0)}\right)^2, \tag{3.65}$$

yielding a gain

$$\Gamma(\lambda^{semi};t)|_{\alpha,\beta\approx 0} = -10\log\frac{|\Psi(t)|}{Q} - 10\log\left(\sqrt{\pi_0}+\sqrt{G(1-\pi_0)}\right)^2, \tag{3.66}$$

where the second term is the penalty incurred due to the probability that a target transitions to a neighboring cell, which we define as

$$L^{transition}(\pi_0, G) = -10\log\left(\sqrt{\pi_0}+\sqrt{G(1-\pi_0)}\right)^2. \tag{3.67}$$

Note that if there is an equal probability that a target remains in the same location as compared to moving to one of G neighbors, then $\pi_0 = (1+G)^{-1}$ and

$$L^{transition}(\pi_0, G)|_{\pi_0=(1+G)^{-1}} = -10\log(G+1). \tag{3.68}$$

3.4.4 Optimal sequential policies

The oracle policies in the previous section provide upper bounds on the performance of any policy. In this chapter, we wish to provide adaptive algorithms that have performance that approach that of the optimal policy. In principle, it is possible to use dynamic programming (DP) to exactly obtain an optimal sequential policy that minimizes equation (3.48) using a sequence of recursive optimizations that proceed as follows:

$$J^*(\lambda;T) = \min_{\{\lambda_i(T)\}_{i=1}^{Q}} J(\lambda;T) \qquad \text{s.t} \qquad \sum_{i=1}^{Q}\lambda_i(T) = \lambda_{total}(T) \tag{3.69}$$

and define recursively

$$J^*(\lambda;t) = \min_{\{\lambda_i(t)\}_{i=1}^{Q}} E\left[J_{t+1}(\lambda)\middle|\xi(t),\{\lambda_i(t)\}_{i=1}^{Q}\right] \qquad \text{s.t} \qquad \sum_{i=1}^{Q}\lambda_i(t) = \lambda_{total}(t) \tag{3.70}$$

for $t = T-1, T-2, \ldots, 1$. Wei and Hero [90] show that this solution is only tractable for $T \leq 2$. For $T > 2$, we have to consider approximations to the optimal sequential policy. In the next sections, we provide a greedy solution that optimizes $J^*(\lambda; t)$ for $t = 1, 2, \ldots, T$ without recursion (i.e., assuming that t is the last stage) and an alternative policy that improves upon the greedy solution with very low additional computational cost.

3.4.5 Greedy sequential policy

The greedy optimization problem at time t can be stated as

$$\min_{\{\lambda_i(t)\}_{i=1}^{Q}} J(\lambda; t) \qquad \text{s.t} \qquad \sum_{i=1}^{Q} \lambda_i(t) = \lambda_{total}(t) \tag{3.71}$$

given the belief state $\mathbf{B}(t-1) = \left\{ \Pr(I_i = 1 | \boldsymbol{Y}(t-1)), \hat{\theta}_i(t-1), \sigma_i^2(t-1) \right\}_{i=1}^{Q}$. Recall that $c_i(t)$ is a function of the $\mathbf{B}(t-1)$. Then, define

$$w_i(t) = \sqrt{\Pr(I_i(t) | \mathbf{Y}(t-1))} \tag{3.72}$$

$$z_i(t) = \frac{w_i(t)}{c_i(t)} \tag{3.73}$$

and order the z_i's so that

$$z_{\tau(1)}(t) \leq z_{\tau(2)}(t) \leq \cdots \leq z_{\tau(Q)}(t). \tag{3.74}$$

Define the cutoff

$$k_0 = \begin{cases} 0, & c_{\tau(1)}(t) \sum_{i=1}^{Q} \frac{w_{\tau(i)}(t)}{w_{\tau(1)}(t)} < \lambda_{total}(t) + \sum_{i=1}^{Q} c_i(t) \\ k, & z_{\tau(k)}(t) \leq \dfrac{\sum\limits_{i=k+1}^{Q} w_{\tau(i)}(t)}{\lambda_{total}(t) + \sum_{i=k+1}^{Q} c_{\tau(i)}(t)} < z_{\tau(k+1)}(t) \end{cases} \tag{3.75}$$

where k_0 is proved to exist and be unique in the work by Bashan et al. [12]. Then the solution to (3.71) is given by

$$\lambda_i^{greedy}(t) = \begin{cases} \dfrac{\lambda_{total}(t) + \sum\limits_{k=k_0+1}^{Q} c_{\tau(k)}(t)}{\sum\limits_{k=k_0+1}^{Q} w_{\tau(k)}(t)} w_i(t) - c_i(t), & \tau(i) > k_0 \\ 0, & \tau(i) \leq k_0. \end{cases} \tag{3.76}$$

Note that when $c_i(t) = 0$ for all i, then

$$\lambda_i^{greedy}(t) = \frac{\lambda_{total}(t)\sqrt{\Pr(I_i(t)|\mathbf{Y}(t-1))}}{\sum_{j=1}^{Q}\sqrt{\Pr(I_j(t)|\mathbf{Y}(t-1))}}. \tag{3.77}$$

It should be noted that this solution is of identical form to the sub-optimal policy provided by Bashan et al. [11], which provided nearly the same performance as ARAP, although without requiring sorting of the posterior probabilities.

Either solution given by equation (3.77) or (3.76) is simple to implement, though the former does not require sorting of the posteriors. However, they are both greedy and have several drawbacks. First, these approaches will perform well when the posteriors $Pr(I_i(t) = 1|\boldsymbol{Y}(t-1)) \approx I_i(t)$ and poorly otherwise. This property causes the greedy policy to have a limited ability to recover missed targets. Second, the greedy policy is also myopic in the sense that it does not trade off short term performance gains for long-term benefits. Chong et al [28] shows that there are significant gains to be had by using non-myopic policies in cases such as

1. Target motion, where there is potential benefit for sensing the target before it becomes unresolvable.

2. Environment variability, when some locations may become unobservable at a particular epoch (e.g., $O_i(t) = 0$ for some i and t).

Finally, the greedy policy does not take into account the weights per stage, $\gamma(t)$, and therefore might allocate more resources than necessary at earlier stages.

3.4.6 Non-myopic policies

To tackle the drawbacks of the greedy algorithm, it may be reasonable to save some resources for exploring the entire space $\mathcal{X}$ rather than simply exploiting the current estimate of the ROI $\Psi(t)$. One straightforward way to do this is to allocate a percentage of the budget $\lambda_{total}(t)$ to all cells, yielding the allocation policy that we term D-ARAP:

$$\lambda_i^{DARAP}(t;\kappa) = \frac{\kappa(t)\lambda_{total}(t)}{Q} + (1-\kappa(t))\lambda_i^{greedy}(t), \qquad (3.78)$$

where $\kappa(t) \in (0,1)$ and $\lambda_i^{greedy}(t)$ is given by either equation(3.76) or (3.77). Note that there are other, possibly equivalent or better, ways to tradeoff exploitation of $\Psi(t)$ and exploration of $\mathcal{X}$. This includes

- Dispersion of the target indicator probabilities by replacing these probabilities with non-normalized pseudo-probabilities:

  $$\tilde{p}_i(t) = \left(\Pr(I_i(t)=1|\boldsymbol{Y}(t-1))\right)^m(t), \quad m(t) \in (0,1]. \qquad (3.79)$$

  There are several possible ways to choose the dispersion exponent $m(t)$ sequence for $t = 1, 2, \ldots, T$.

- Random sampling of $\mathcal{X}$, using a compressed sensing matrix, rather than a uniform allocation to all cells.

- Multiple (coarse)-scale sampling of $\mathcal{X}$, rather than a uniform allocation to all cells.

To keep the discussion focused, this chapter considers the policy given in equation (3.78), though the general framework could be applied to any of the policies described above[3].

In general, choosing $\kappa(t)$ is still a difficult problem. This work proposes two strategies for selecting $\kappa(t)$.

[3] It should be noted that we look for deterministic policies computed offline (i.e., before measurements are measured). In Section 3.4.9, we provide an approximate to the POMDP solution for finding an online allocation policy. In Section 3.5, performance of the offline and online algorithm are compared, as well as their computational costs. Note that in the online policies, computational cost is many orders higher than their offline alternatives, though the additional performance gains might be beneficial in some scenarios.

3.4.7 Nested optimization for $\kappa(t)$

First, consider the nested optimization problem for finding $\{\kappa_D(t)\}_{t=1}^T$, which we will denote as $\kappa_D^{(T)}(t)$:

Algorithm 3. *Nested policy for* $\kappa_D^{(T)}(t), \quad t = 1, 2, \ldots, T.$

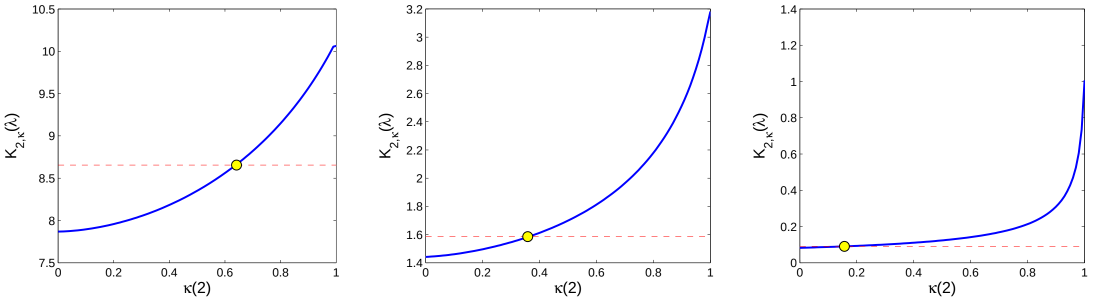
```
For each $\tau = 1, 2, \ldots, T$:
  Step 1: Set $\kappa_D^{(\tau)}(1) = 1$, since we assume a non-informative prior on the locations
          of targets at $t = 1$.
  Step 2: Set $\kappa_D^{(\tau)}(\tau) = 0$, which reflects the fact that the last stage should not save
          resources for later stages.
  Step 3: Set $\kappa_D^{(\tau)}(t) = \kappa_D^{(\tau-1)}(t-1)$ for $t = 3, 4, \ldots, \tau - 1$.
  Step 4: Optimize $J(\lambda_i^{DARAP}; \tau)$ over $\kappa_D^{(\tau)}(2)$.
```

3.4.8 Heuristic optimization of $\kappa(t)$

Note that the nested optimization procedure requires searching over a single parameter at each stage. However, it also requires simulation of a τ-stage policy for $\tau = 3, 4, \ldots, T$. Thus, we require $O(T^2)$ simulations in order to find $\kappa_D^{(T)}$ according to this strategy. Although this optimization is tractable, we may consider a heuristic approximation that reduces the computational burden to $O(T)$ simulations. To understand this heuristic, it is illustrative to look at Fig. 3.3, which plots the myopic cost

$$K_{t,\kappa}(\lambda) = J(\lambda_i^{DARAP}(t; \kappa); t) \tag{3.80}$$

for $\kappa(1) = 1$, $\kappa(2) \in (0, 1)$, $t = 2$, and $c_i(t) = 0$. We plot $K_{t,\kappa}(\lambda)$ for low, medium, and high values of $\lambda_{total}(t)$ in (a), (b), and (c), respectively. It is seen that in all



(a) SNR = 0 dB

(b) SNR = 5 dB

(c) SNR = 10 dB

Figure 3.3: We plot the myopic cost $K_{2,\kappa}(\lambda)$ given by equation (3.80) for $\kappa(1) = 1$, $\kappa(2) \in (0, 1)$, $t = 2$, and $c_i(t) = 0$. We plot $K_{t,\kappa}(\lambda)$ for low, medium, and high values of $\lambda_{total}(t)$ in (a), (b), and (c), respectively. It is seen that in all cases, the myopic cost is optimized when $\kappa(2) = 0$. However, lower SNR values can tolerate a larger value of $\kappa(2)$ and only have a small deviation in cost. The red dotted line shows a deviation of 10% from the minimum cost, while the yellow circle marks the point where κ attains this value.

cases, the myopic cost is optimized when $\kappa(2) = 0$. However, lower SNR values can tolerate a larger value of $\kappa(2)$ and only have a small increase in cost. The red dotted line shows a deviation of 10% from the minimum cost, while the yellow circle marks the point where κ attains this value. We use the following heuristic strategy:

Algorithm 4. *Heuristic policy for* $\kappa_D^{(T)}(t), \quad t = 1, 2, \dots, T.$

For each $\tau = 1, 2, \dots, T$*:*

Step 1: Set $\kappa_D^{(\tau)}(1) = 1$*, since we assume a non-informative prior on the locations of targets at* $t = 1$.

Step 2: Set $\kappa_D^{(\tau)}(t) = \kappa_D^{(\tau-1)}(t)$ *for* $t = 2, 3, \dots, \tau - 1$.

Step 3: Let $\varepsilon \geq 0$ *control the deviations that we are willing to tolerate.*

Step 4: Then define

$$\kappa_D^{(\tau)}(\tau) = \max_{\kappa} \{\kappa : K_{\tau,\kappa}(\lambda) < (1 + \varepsilon) K_{\tau,0}(\lambda)\} \tag{3.81}$$

Table 3.2: Parameters used for simulation analysis

Parameter	Variable Name	Value
Number of locations	Q	$1,000$
Prior sparsity	$p = \Pr(I_i(1) = 1)$	0.01
Expected number of targets	$E[\tilde{N}(t)]$	10
Target amplitude mean	μ_θ	1
Target amplitude standard deviation (prior)	σ_θ	1/6
Target amplitude standard deviation (update)	Δ	1/20
Noise variance	σ^2	1
Death probability	α	0
Birth probability	β	0
Number of neighbors	G	2

This strategy allocates as much energy to the floor probability as possible, under the condition that the myopic cost is within $(100\varepsilon)\%$ of the minimum myopic cost. At low SNR[4], we would expect that $\kappa_D^{(T)}(t)$ will be large and decay to zero as SNR improves or t gets large. We note that this strategy clearly does not optimize our greedy cost function, since $\kappa_D^{(T)}(T) \neq 0$ unless SNR is asymptotically high. However, this heuristic solution can be computed by simulating $O(T)$ stages for estimating $K_{T,\kappa}(\lambda)$ through Monte-Carlo simulations.

Fig. 3.4 figure shows the selection of $\kappa_D^{(T)}(T)$ according to Algorithms 3 (nested) and 4 (heuristic) for simulation parameters given by Table 3.2. In these simulations, it is assumed that the total energy budget available at each stage is equal, so that $\lambda_{total}(t) = \lambda_{total}(0)$, leading to higher total energy as t increases. Thus, we expect that the floor percentage $\kappa_D^{(T)}(t)$ should decrease as either t or SNR increase. The policies in Figure 3.4 generally reflect this trend.

[4] SNR is defined as $\mathrm{SNR}(\lambda_{total}) = 10\log_{10}\left(\frac{\lambda_{total}/Q}{\sigma^2}\right)$.

It should be noted that in both Figure 3.4(b) and (d), the heuristic policy has very smooth behavior as either t or SNR increase. In order to take advantage of this, Figure3.4(e) plots the heuristic selections against observed SNR, which is defined as

$$\lambda_{total}^{(obs)}(T) = \sum_{t=1}^{T} \lambda_{total}(t) = T\lambda_{total}(0) \tag{3.82}$$

$$\mathrm{SNR}^{(obs)}(T) = 10\log_{10}\left(\frac{\lambda_{total}^{(obs)}(T)/Q}{\sigma^2}\right) \tag{3.83}$$

It is seen that there is a high correlation between all selections as function of observed SNR, which suggests that the heuristic policy could be well-approximated by a function only of observed SNR, as shown by the solid black line in (e). This leads to the definition of a functional approximation to the heuristic policy as

$$\kappa_D^{(T)}(t) = \kappa_D^{(2)}\left(2; SNR^{(obs)}(t)\right). \tag{3.84}$$

Note that this functional approximation reduces the computational burden for finding $\left\{\kappa_D^{(T)}(t)\right\}_{t=1}^{T}$ to just $O(1)$ simulations (i.e., it is does not grow as a function of T). For large T, this computational savings may be tremendous. We provide a succinct description of D-ARAP in Algorithm 5.

Algorithm 5. *Dynamic Adaptive Resource Allocation Policy (D-ARAP), λ^{D-ARAP}*

```
For each $t = 1, 2, \ldots, T$:
  Step 1: Select $\kappa^{(T)}(t) = \kappa_D^{(T)}(t; SNR)$, according to the nested optimization policy
          (Algorithm 3) or the heuristic policy (Algorithm 4).
  Step 2: Allocate $\lambda_i^{DARAP}(t; \kappa)$ according to equation (3.78) and collect measurements
          $\boldsymbol{Y}(t)$ according to equation (3.32).
  Step 3: Update belief state $\mathbf{B}(t)$ from collected measurements.
```

3.4.9 Approximate POMDP optimization for $\kappa(t)$

In this chapter, we also consider approximations to a POMDP solution through rollout policies of various lengths (increasing length leads to higher computational costs). Recall, that POMDPs are based on maximizing the Q-function, which depends on the belief state $\boldsymbol{B}(t)$ and the possible actions $a \in \mathcal{A}$, so that

$$Q_{T-t}(\boldsymbol{B}(t), a) = r(\boldsymbol{B}(t), a) + E[V^*_{T-t-1}(\boldsymbol{B}(t))|\boldsymbol{B}(t), a] \tag{3.85}$$

and the optimal cost is given by

$$V^*_T(\boldsymbol{B}(0)) = \max_a \big(r(\boldsymbol{B}(t), a) + E[V^*_{T-1}(\boldsymbol{B}(0))|\boldsymbol{B}(0), a]\big) = \max_a Q_{T-t}(\boldsymbol{B}(t), a) \tag{3.86}$$

In rollout policies, the Q-function is approximated as

$$Q_{T-t}(\boldsymbol{B}(t), a) = r(\boldsymbol{B}(t), a) + E[V^{(base)}_{T-t-1}(\boldsymbol{B}(t+1))|\boldsymbol{B}(t), a] \tag{3.87}$$

where a base policy is defined a priori and the second term can be found through simple Monte Carlo estimation, starting from the belief state at time $t+1$ and applying the base policy for the next T_0 stages. The choice of base policy is a design parameter, but given enough Monte Carlo simulations to estimate the mean term $E[V^{(base)}_{T-t-1}(\boldsymbol{B}(t+1))|\boldsymbol{B}(t), a]$, rollout policies are guaranteed to do at least as well as the base policy.

In this chapter, we consider two types of rollout policies. First, we consider optimization over $\kappa(t)$ using D-ARAP as the base policy, $\kappa^{(base;T)}(t) = \kappa^{(T)}_D(t)$. Note that this optimization procedure is done separately for each individual realization of the target state. However, in aggregate, the solution is very similar to the nested solution which optimizes a single parameter given a base policy (i.e., the nested solution with 1 fewer stage). Thus we expect the performance of this rollout policy to be quite similar to D-ARAP.

Note that it might be desirable to use an approximate POMDP solution to find $\{\lambda_i(t)\}_{i=1}^{Q}$. Generally, this is intractable since Q is very large. However, in some cases involving symmetry, we might able to extend optimization of the allocation policy to more than a single parameter. For example, consider the situation where a small part of the scene may have significantly noisier measurements than other parts (e.g., being in the null of a radar beam). Then, a better policy could allocate differently to this region by weighting the probabilities by a coefficient $z(t)$:

$$\tilde{p}_i(t) = \begin{cases} z(t)p_i(t), & i \in \mathcal{R}_{faulty}(t) \subset \mathcal{X}, \\ p_i(t), & i \notin \mathcal{R}_{faulty}(t) \subset \mathcal{X}. \end{cases} \tag{3.88}$$

where $\mathcal{R}_{faulty}(t)$ is the region (possibly changing) where faulty measurements may occur. In such a situation, an approximate POMDP solution can optimize over $w(t)$ and $\kappa(t)$ in order to provide additional gains over D-ARAP or other alternative policies.

3.5 Performance analysis

3.5.1 Simulation set-up

In this section, we compare the performance of D-ARAP to a variety of other policies, including the oracle policies, the uniform policy, and a myopic policy that sets $\kappa_D^{(T)}(t) = 0$ for all $t = 1, 2, \ldots, T$. Simulation parameters are given by Table 3.2 unless stated otherwise.

3.5.2 Model Mismatch

In this section, a comparison done between D-ARAP (functional) to a myopic policy that sets $\kappa_D^{(T)}(t) = 0$ for all $t = 1, 2, \ldots, T$. In the simulation, we consider a static case with very large T and very small prior probability of targets ($p = 10^{-3}$).

Moreover, we consider the case of model mismatch, where the model assumed by the policy planning is given by

$$\theta_i(t) = \theta_i, \qquad \theta_i \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2), \tag{3.89}$$

$$I_i(t) = I_i. \tag{3.90}$$

However, the realizations of the amplitudes is given by a smaller parameter

$$\theta_i = \theta_0 < \mu_\theta \tag{3.91}$$

for $\theta_0 = (0.50, 0.55, 0.60, 0.65, 0.70, 0.75)$ and $\mu_\theta = 1$. For $\theta_0 \ll \mu_\theta$, noisy measurements from cells containing targets can be easily confused with the background noise. In these situations, the myopic policy will be more adversely affected by small $\Pr(I_i = 1|\boldsymbol{Y}(t))$ for targets in the ROI as compared to D-ARAP. In Figure 3.5, we compare these policies to a uniform search. For low values of θ_0, noisy measurements cause missed targets that are never recovered by the myopic policy for $\theta_0 < 0.75$. On the other hand, D-ARAP has approximately monotonically increasing gains for all $\theta_0 > 0.5$, suggesting greater robustness to noise than the myopic policy. Moreover, even when $\theta_0 = 0.75$, D-ARAP converges to the optimal gain (30 dB) in fewer stages than the myopic policy.

3.5.3 Complex dynamic behavior: faulty measurements

In the next simulation, we provide a scenario where the sensor has random (undetected) faults every 15 stages, causing highly noisy measurements. Note that in this case, the myopic policy will have catastrophic failure, in the sense that it will lose existing targets and not be able to recover them. On the other hand, D-ARAP (which has a minimum floor percentage greater than zero), will always save some resources in order to recover missed targets.

This simulation compares the performance of the myopic policy, D-ARAP, and two approximate POMDP solutions using rollout of size $T_0 = 2$ and $T_0 = 5$. The POMDP solutions use D-ARAP as the base policy, which should guarantee performance at least as good as D-ARAP. The computational burden of each of these policies is given in Table 3.5.3. Clearly, using an approximate POMDP solution is several orders of magnitude more computational time than the nested, heuristic, functional, or myopic solutions.

Figure 3.6 shows the performance of the myopic policy and D-ARAP for several values of SNR per stage and $T = 200$. It is seen that the myopic policy shown in (a) suffers from catastrophic failures for high SNR cases, in the sense that targets are lost and not recovered. Indeed, as t and SNR increase, the performance of the myopic policy trends downwards and eventually becomes worse than a uniform search. On the other hand, the D-ARAP (functional) policy shown in (b) has the ability to recover from misdetections, because it always allocates some resources to all cells.

Due to computational constraints, we only show comparisons to the approximate POMDP solutions for SNR of 10 dB. In Figure 3.7, it is seen that the performance of the rollout policies parallel the D-ARAP policy. This gives confidence in using an offline policy in order to determine the best allocation policy in this scenario.

It is not completely surprising that the approximate POMDP solution does not perform significantly better than the D-ARAP solution, since computational constraints make it difficult to optimize over $\kappa_D^{(T)}(t)$ in an online fashion.

On the other hand, we consider an additional scenario where the user knows when/where the sensor will fail. In this case, a POMDP solution can be used to weight the predictive posterior probabilities by a factor $w(t)$ in the region where

Table 3.3: Computational cost comparison

Algorithm	Offline Simulation	Online Simulation	Simulated Parameters*		
	Time (Big-$\mathcal{O}$)	Time (Big-$\mathcal{O}$)	Offline	Online	Total
Myopic, $\kappa(t) = 0$	$\mathcal{O}(1)$	$\mathcal{O}(N_{sim})$	1×10^0	1×10^2	1.01×10^2
Nested, $\kappa(t)$	$\mathcal{O}(T^2 K N_{mc})$	$\mathcal{O}(N_{sim})$	4×10^5	1×10^2	4.00×10^5
Heuristic, $\kappa(t)$	$\mathcal{O}(T K N_{mc})$	$\mathcal{O}(N_{sim})$	2×10^4	1×10^2	2.01×10^4
Functional, $\kappa(t)$	$\mathcal{O}(K N_{mc})$	$\mathcal{O}(N_{sim})$	1×10^3	1×10^2	1.10×10^3
T_0-stage Rollout, $\kappa(t)$	$\mathcal{O}(1)^{**}$	$\mathcal{O}(T T_0 K N_{mc} N_{sim})$	1×10^0	1×10^7	1.00×10^7
T_0-stage Rollout, $\kappa(t)$ and $z(t)$	$\mathcal{O}(1)^{**}$	$\mathcal{O}(T T_0 K Z N_{mc} N_{sim})$	1×10^0	1×10^8	1.00×10^8

* For parameters, $N_{sim} = 100$, $T = 20$, $K = 10$ (number of possibilities for $\kappa(t)$), $N_{mc} = 100$, $T_0 = 5$, $Z = 10$ (number of possibilities for $z(t)$).

** Using a myopic base policy. Otherwise, include the offline simulation time for nested/heuristic/functional policies.

faulty measurements occur, so that

$$\tilde{p}_i(t) = \begin{cases} z(t)p_i(t), & i \in \mathcal{R}_{faulty}(t) \subset \mathcal{X}, \\ p_i(t), & i \notin \mathcal{R}_{faulty}(t) \subset \mathcal{X}. \end{cases} \tag{3.92}$$

where $\mathcal{R}_{faulty}(t)$ is the region (possibly changing) where faulty measurements may occur. In contrast, we assume that the offline policies (myopic/D-ARAP) can only ignore these measurements (without additional planning). In this case, the offline policies will lose significant performance at times where their resources could be used more effectively.

Figure 3.8 shows the gains in our optimization function over a uniform search for the myopic policy, D-ARAP, and a POMDP solution that selects over both the probability weight $w(t)$ and the floor percentage $\kappa(t)$. It is seen that the POMDP solution has better performance during these faulty measurement periods (i.e., every 15 stages), as compared to D-ARAP and the myopic policy. Note that in the standard situation (i.e., without faulty measurements), D-ARAP performs very closely with the POMDP solution. On the other hand, the myopic policy continues to have a downward trend, even though no catastrophic events occur as in Figures 3.6 and 3.7.

3.5.4 Comparison to optimal/uniform policies

This section now considers the case where targets are completely observable with parameters given by Table 3.2. We compare performance in terms of gains in the optimization function (i.e., cost), gains in MSE and probability of detection for a small fixed false alarm rate, as compared to a uniform search. Figure 3.9 shows gains in the optimization function given by equation (3.48) for various values of SNR and varying policy lengths, T. It is seen that generally the nested policy has the highest gains in the optimization function among non-oracle policies. The differences are most apparent for higher SNR scenarios (c) and (d). Generally, the nested policy performs very similarly with the heuristic and functional policies, although those policies have much smaller computational burden. The myopic policy, on the other hand, has significantly worse performance as t or SNR increase.

Figure 3.10 shows the MSE gains for estimating $\{\theta_i(t)\}_{i\in\Psi(t)}$ for targets within the ROI with respect to a uniform search. Note that generally the nested policy has the highest gains in MSE among non-oracle policies. The differences are most apparent for higher SNR scenarios (c) and (d), with performance close to the optimal level as t gets large. Once again, there is relatively little performance differences between the nested and heuristic/functional policies.

Finally, Figure 3.11 shows the probability of detection for a fixed probability of false alarm ($P_{fa} = 10^{-4}$) as a function of t. Note that the probability of detection for D-ARAP (nested, heuristic, and functional policies) consistently approaches 1 as t and $\lambda_{total}(t)$ approach infinity. Moreover, they approach unity significantly faster than the uniform and myopic policies. Finally, note that for the larger values of SNR, D-ARAP approaches $P_d = 1$ within very few stages.

3.6 Discussion and future work

In this chapter, we developed a general framework for extending previous adaptive search policies to more than 2 stages that can account for dynamic targets. Consequently, we provided a related cost function that is adaptable to many target and state models and derived oracle allocations that provided bounds on achievable performance. We developed a non-myopic policy based on a heuristic approximation to a dynamic programming optimization problem, which we named D-ARAP. This heuristic reduces the computational burden related to searching for a policy to requiring only $O(T)$ or $O(1)$ simulations. Finally, we demonstrated excellent empirical evidence for D-ARAP, including asymptotic consistency, significant performance gains over a uniform policy, and increased robustness to noise as compared to a non-myopic policy.

3.7 Appendix: Discussion of the choice of α and β

It should be noted that the performance of the previous algorithms, ARAP and M-ARAP, depend strongly on the sparsity of the state, which is given by

$$\tilde{p}(t) = \frac{\tilde{N}(t)}{Q}. \tag{3.93}$$

In order to make valid comparisons across stages, it is assumed that α and β are chosen in order to keep

$$E[\tilde{p}(t)] = p, \tag{3.94}$$

where p is the prior sparsity level. Note that under the event that $N(t+1) = N(t)$, we know

$$\begin{aligned} Z &\triangleq E[\tilde{N}(t+1) = n | \tilde{N}(t) = N, N(t+1) = N(t)] \\ &= \sum_{j=0}^{N} (N-j)(1-\alpha)^{N-j}\alpha^{j} \end{aligned} \tag{3.95}$$

Assume that α is small so that $\alpha^j \approx 0$ for $j > 1$. Then we have

$$Z \approx N(1-\alpha)^N + (N-1)(1-\alpha)^{N-1}\alpha. \tag{3.96}$$

Including the probability that a target is added to the scene, we have

$$\begin{aligned} E[\tilde{N}(t+1) = n | \tilde{N}(t) = N] &= (1-\beta)Z + \beta(Z+1), \\ &= \beta + Z. \end{aligned} \tag{3.97}$$

Thus, the condition that the number of targets in the scene remain the same in expectation is given by

$$N = \beta + E[\tilde{N}(t+1) = n | \tilde{N}(t) = N] \tag{3.98}$$

for $N = E[\tilde{N}(1)] = Qp$. Thus, we have a simple equation for choosing β as

$$\begin{aligned} \beta &= Qp - [Qp(1-\alpha)^{Qp} + (Qp-1)(1-\alpha)^{Qp-1}\alpha] \\ &= Qp - (Qp - \alpha)(1-\alpha)^{Qp-1} \end{aligned} \tag{3.99}$$

Since $\alpha = [0, 1)$ and $Qp > 1$, we know that equation (3.99) is monotonic in α. Thus, to ensure that $\beta \in [0, 1]$, we require

$$1 \ll \beta = Qp - (Qp - \alpha)(1-\alpha)^{Qp-1}. \tag{3.100}$$

For example, with $Q = 10^4$ and $p = 10^{-3}$, we have $\alpha \ll 0.01$.

3.8 Appendix: Efficient posterior estimation for given dynamic state model

In order to use the algorithms provided in this thesis to adaptively estimate the state $\xi(t)$ given the measurements, we need to be able to calculate the posterior probabilities for the indicator variables, $\{I_i(t)\}_{i=1}^{Q}$ given the measurements up until time t. Define the measurement vectors

$$\boldsymbol{y}(t) = \{y_1(t), y_2(t), \ldots, y_Q(t)\} \tag{3.101}$$

and

$$\boldsymbol{Y}(t) = \{\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(t)\} \tag{3.102}$$

Let

$$\Pr(I_i(t) = 1 | \boldsymbol{Y}(t-1)) \qquad i = 1, 2, \ldots, Q, \quad t = 1, 2, \ldots, T \tag{3.103}$$

be the posterior probabilities that need to be calculated. For $t = 1$, we have

$$\Pr(I_i(1) = 1) = p, \tag{3.104}$$

under the assumption that at most one target exists in a single location. For $t > 1$, we have

$$\begin{aligned}\Pr(I_i(t) = 1|\boldsymbol{Y}(t-1)) &= \int \Pr(I_i(t) = 1, \boldsymbol{S}(t)|\boldsymbol{Y}(t-1))d\boldsymbol{S}(t) \\ &= \int \Pr(I_i(t) = 1|\boldsymbol{S}(t))f(\boldsymbol{S}(t)|\boldsymbol{Y}(t-1))d\boldsymbol{S}(t) \\ &= \sum_{n=1}^{N(t)} \Pr(s^{(n)}(t) = i|\boldsymbol{Y}(t-1)),\end{aligned} \tag{3.105}$$

where the last equation can be derived noting that

$$\Pr(I_i(t) = 1|\boldsymbol{S}(t)) = \begin{cases} 1, & \exists n : s^{(n)}(t) = i \\ 0, & \text{else} \end{cases} \tag{3.106}$$

Thus, in order to compute equation (3.105), we need to be able to estimate the state $\xi(t)$ given $\boldsymbol{Y}(t-1)$.

3.8.1 Recursive equations for updating $\xi(t)$

In general, we can compute the posteriors using the equations:

$$f(\xi(t)|\boldsymbol{Y}(t-1)) = \int f(\xi(t)|\xi(t-1))f(\xi(t-1)|\boldsymbol{Y}(t-1))d\xi(t-1) \tag{3.107}$$

$$f(\xi(t)|\boldsymbol{Y}(t)) = \frac{f(\boldsymbol{y}(t)|\xi(t))f(\xi(t)|\boldsymbol{Y}(t-1))}{\int f(\boldsymbol{y}(t)|\tilde{\xi}(t))f(\tilde{\xi}(t)|\boldsymbol{Y}(t-1))d\tilde{\xi}(t)} \tag{3.108}$$

Note that each target has an associated real-valued amplitude $x^{(n)}(t)$ and a location on a large discrete grid $s^{(n)}(t) \in \{1, 2, \ldots, Q\}$ for large Q. Thus, the joint densities $f(\xi(t)|\boldsymbol{Y}(t-1))$ and $f(\xi(t)|\boldsymbol{Y}(t-1))$ are in general very high-dimensional functions that may be intractable to estimate exactly. Under certain assumptions, however, it may be possible to derive exact equations for these updates.

3.8.2 Static case

In the static case when $\alpha = \beta = 0$ and $\pi_0 = 1$, we have the simple situation where

$$\boldsymbol{S}(t) = \boldsymbol{S}(t-1). \tag{3.109}$$

Since targets are fixed in position and cannot occupy the same cell by assumption in equation (3.30), we can easily show that the joint density factors into:

$$f(\xi(t)|\boldsymbol{Y}(t')) = f(\Psi(t), \Theta(t)|\boldsymbol{Y}(t')) = f(\Theta(t)|\boldsymbol{Y}(t'), \Psi(t))f(\Psi(t)|\boldsymbol{Y}(t')) \tag{3.110}$$

for $t' = t, t-1$, $\Psi(t) = \{I_i(t)\}_{i=1}^{Q}$, and $\Theta(t) = \{\theta_i(t)\}_{i=1}^{Q}$. Moreover, $\Psi(t) = \Psi$ and since the targets are independent across cells, we have:

$$f(\Theta(t)|\boldsymbol{Y}(t'), \Psi(t)) = \prod_{i=1}^{Q} f\left(\theta_i(t)|\mathbf{y}_i(t'), I_i\right) \tag{3.111}$$

$$f(\Psi(t)|\boldsymbol{Y}(t')) = \prod_{i=1}^{Q} f\left(I_i|\mathbf{y}_i(t')\right) \tag{3.112}$$

where $\mathbf{y}_i(t) = \{y_i(t_0)\}_{t0=1}^{t}$. Note that $\theta_i(t)$ is only defined if $I_i = 1$. Conditioned on this event, we furthermore note that $\theta_i(t)$ and $y_i(t)|\theta_i(t)$ are normally distributed given the allocations $\lambda_i(t)$. Thus, the posteriors $f\left(\theta_i(t)|\mathbf{y}_i(t'), I_i\right)$ for $t' = t, t-1$ can be updated exactly through the Kalman filter equations similar to those given in Chapter II:

$$\delta_i(t) = y_i(t) - \sqrt{\lambda_i(t)}\hat{\theta}_i(t|t-1) \tag{3.113}$$

$$s_i(t) = \lambda_i(t)\hat{\sigma}_i^2(t|t-1) + \sigma^2 \tag{3.114}$$

$$\Gamma_i(t) = \frac{\hat{\sigma}_i^2(t|t-1)\sqrt{\lambda_i(t)}}{s_i(t)} \tag{3.115}$$

$$\hat{\theta}_i(t|t) = \hat{\theta}_i(t|t-1) + \Gamma_i(t)\delta_l(t), \tag{3.116}$$

$$\hat{\sigma}_i^2(t|t) = [1 - \Gamma_i(t)h_i(t)]\hat{\sigma}_i^2(t|t-1), \tag{3.117}$$

where $\delta_i(t)$ is the residual measurement error, $s_i(t)$ is the update measurement error, $\Gamma_i(t)$ is the Kalman gain, and $(\hat{\theta}_i(t|t), \hat{\sigma}_i^2(t|t))$ are the updated state estimates. The predict equations are given by:

$$\hat{\theta}_i(t|t-1) = \hat{\theta}_i(t-1|t-1), \tag{3.118}$$

$$\hat{\sigma}_i^2(t|t-1) = \hat{\sigma}_i^2(t-1|t-1) + \Delta_\theta^2. \tag{3.119}$$

Moreover, the posteriors on the indicator functions can be easily computed recursively as

$$f(I_i = 1|\mathbf{y}_i(t)) = \frac{f(y_i(t)|I_i = 1, \mathbf{y}_i(t-1))f(I_i = 1|\mathbf{y}_i(t-1))}{\sum\limits_{j=0,1} f(y_i(t)|I_i = j, \mathbf{y}_i(t-1))f(I_i = j|\mathbf{y}_i(t-1))}, \tag{3.120}$$

where we note that when $I_i = 0$

$$y_i(t)|I_i = 0, \mathbf{y}_i(t-1) \sim \mathcal{N}(0, \sigma^2) \tag{3.121}$$

and when $I_i = 1$

$$\begin{aligned} & f(y_i(t)|I_i = 1, \mathbf{y}_i(t-1)) \\ & \qquad = \int f(y_i(t)|\theta_i(t), I_i = 1) f(\theta_i(t)|I_i = 1, \mathbf{y}_i(t-1)) d\theta_i(t) \\ & \qquad = \int \phi(y_i(t); \sqrt{\lambda_i(t)}\theta_i(t), \sigma^2)\phi(\theta_i(t); \hat{\theta}_i(t|t-1), \hat{\sigma}_i^2(t|t-1)) d\theta_i(t) \\ & \qquad = \phi(y_i(t); \sqrt{\lambda_i(t)}\hat{\theta}_i(t|t-1), \lambda_i(t)\hat{\sigma}_i^2(t|t-1) + \sigma^2) \end{aligned} \tag{3.122}$$

where $\phi(x; \mu, \sigma^2)$ is the Gaussian pdf with mean μ and variance σ^2 evaluated at x. From this equation we see that

$$y_i(t)|I_i = 1, \mathbf{y}_i(t-1) \sim \mathcal{N}(\sqrt{\lambda_i(t)}\hat{\theta}_i(t|t-1), \lambda_i(t)\hat{\sigma}_i^2(t|t-1) + \sigma^2) \tag{3.123}$$

In the static case, we see that updating the posteriors for I_i for all $t = 1, 2, \ldots, T$ involves (a) updating the conditional mean and variances for $\theta_i(t)$ given the measurements, and (b) updating the posterior probability for $I_i = 1$. This gives insight into an approximate method that we will use in the general case when the targets are allowed to move, enter, or leave the scene.

3.8.3 Approximations in the general case

Similar to the static case, we assume that there are no interacting targets so that we can factor our posterior density into a form that makes it tractable to estimate directly. In order to do this, we have to make the additional assumption that there is at most one target in the vicinity of a location, so that

$$|\{n : s^{(n)}(t) \in H(i)\}| \leq 1 \tag{3.124}$$

for all $i = 1, 2, \ldots, Q$. This is clearly more restrictive than the assumption in equation (3.32). Under this assumption, we have for $t' = t, t-1$

$$\begin{aligned} f(\xi(t)|\boldsymbol{Y}(t')) &= f(\Psi(t), \Theta(t)|\boldsymbol{Y}(t')) \\ &= f(\Theta(t)|\Psi(t), \boldsymbol{Y}(t'))f(\Psi(t)|\boldsymbol{Y}(t')) \\ &= \prod_{i=1}^{Q} f(\theta_i(t)|\Psi(t), \boldsymbol{Y}(t'))f(I_i(t)|\boldsymbol{Y}(t')) \end{aligned} \tag{3.125}$$

Beginning with the target amplitudes, we note that

$$\begin{aligned} &f(\theta_i(t)|\Psi(t), \boldsymbol{Y}(t-1)) \\ &\qquad = f(\theta_i(t)|I_i(t) = 1, \boldsymbol{Y}(t-1)) \\ &\qquad = \int f(\theta_i(t)|I_i(t) = 1, \Psi_{H(i)}(t-1), \boldsymbol{Y}(t-1)) \\ &\qquad \cdot f(\Psi_{H(i)}(t-1)|I_i(t) = 1, \boldsymbol{Y}(t-1))d\Psi_{H(i)(t-1)}, \end{aligned} \tag{3.126}$$

where $\Psi_{H(i)}(t) = \{I_j(t)\}_{j \in H(i)}$. Define $E_{i,j}(t)$ to be the event that assigns $\Psi_{H(i)}(t)$ as

$$E_{i,j}(t) \triangleq \begin{cases} I_k(t) = 1, & j = k, \\ I_k(t) = 0, & j \neq k, \end{cases} \qquad \forall k \in H(i) \tag{3.127}$$

The event $E_{i,0}(t)$ refers to the case where a target is added to the scene at location i at time t. Then, under the assumption that at most one target exists in the vicinity of a cell, we have

$$\begin{aligned} & f(\theta_i(t)|\Psi(t), \boldsymbol{Y}(t-1)) \\ & \quad = \sum_{j \in H(i) \cup \{0\}} f(\theta_i(t)|I_i(t) = 1, E_{i,j}(t-1), \boldsymbol{Y}(t-1)) \\ & \qquad \cdot f(E_{i,j}(t-1)|I_i(t) = 1, \boldsymbol{Y}(t-1)) \\ & \quad = \sum_{j \in H(i) \cup \{0\}} \int f(\theta_i(t)|\theta_j(t-1)) \\ & \qquad \cdot f(E_{i,j}(t-1)|I_i(t) = 1, \boldsymbol{Y}(t-1)) \\ & \qquad \cdot f(\theta_j(t-1)|I_j(t-1) = 1, \boldsymbol{Y}(t-1)) d\theta_j(t-1), \end{aligned} \tag{3.128}$$

where it is understood that in the case where a new target is added to the scene

$$f(\theta_0(t-1)|I_0(t-1) = 1, \boldsymbol{Y}(t-1)) = f(\theta_0(t-1)) \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2) \tag{3.129}$$

and

$$\theta_i(t) = \theta_0(t-1) \tag{3.130}$$

In the static case, both $f(\theta_i(t)|\theta_j(t-1))$ and $f(\theta_j(t-1)|I_j(t-1) = 1, \boldsymbol{Y}(t-1))$ are Gaussian which makes it possible to analytically integrate equation (3.128). Indeed, at time $t = 1$, it can be easily seen that $\theta_j(1) \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$. However, for $t > 1$, equation (3.128) shows that we get a Gaussian mixture model with mixing coefficients given by

$$f(E_{i,j}(t-1)|I_i(t) = 1, \boldsymbol{Y}(t-1)) \tag{3.131}$$

In order to make the estimation of the posterior distributions very simple, we make the assumption that

$$f(E_{i,j}(t-1)|I_i(t) = 1, \boldsymbol{Y}(t-1)) = \mathbf{1}_{\{j=j^*\}} \tag{3.132}$$

for a single $j^* \in H(i) \cup \{0\}$. In other words, conditioned on the event that a target exists at cell i, it is known with probability 1 that the target transitioned from either a single neighboring cell or entered the scene at time t. This assumption is restrictive except at high SNR. However, it allows us to simplify equation (3.128) as

$$\begin{aligned} f(\theta_i(t)|\Psi(t), \boldsymbol{Y}(t-1)) = \int & f(\theta_i(t)|\theta_{j^*}(t-1)) \\ & \cdot f(\theta_{j^*}(t-1)|I_{j^*}(t-1) = 1, \boldsymbol{Y}(t-1)) d\theta_{j^*}(t-1), \end{aligned} \tag{3.133}$$

which can easily seen to be Gaussian distributed as long as $f(\theta_j(t)|I_j(t) = 1, \boldsymbol{Y}(t))$ is Gaussian. Indeed, we see the recursion

$$f(\theta_i(t)|I_i(t) = 1, \boldsymbol{Y}(t)) \propto f(y_i(t)|I_i(t), \theta_i(t)) f(\theta_i(t)|I_i(t) = 1, \boldsymbol{Y}(t-1)) \tag{3.134}$$

Using equations (3.133) and (3.134), it is simple to show that a simply modified Kalman filter will give the exact recursion required to update the posterior densities. In fact, it is the same recursion given in the static case, except that we have

$$\hat{\theta}_i(t|t-1) = \hat{\theta}_{j_i^*(t-1)}(t-1|t-1), \tag{3.135}$$

$$\hat{\sigma}_i^2(t|t-1) = \hat{\sigma}^2_{j_i^*(t-1)}(t-1|t-1) + \Delta_\theta^2 \tag{3.136}$$

$$j_i^*(t-1) = \arg \max_{j \in H(i) \cup \{0\}} f(E_{i,j}(t-1)|I_i(t) = 1, \boldsymbol{Y}(t-1)). \tag{3.137}$$

Looking at the update equations for the target indicators, we get

$$f(I_i(t) = 1|\boldsymbol{Y}(t-1)) = \sum_{j \in H(i) \cup \{0\}} f(I_i(t) = 1|E_{i,j}(t-1)) f(E_{i,j}(t-1)|\boldsymbol{Y}(t-1)) \tag{3.138}$$

and

$$f(I_i(t) = 1|\boldsymbol{Y}(t)) \propto f(y_i(t)|I_i(t) = 1, \boldsymbol{Y}(t-1)) f(I_i(t) = 1|\boldsymbol{Y}(t-1)), \tag{3.139}$$

where

$$\begin{aligned} & f(y_i(t)|I_i(t) = 1, \boldsymbol{Y}(t-1)) \\ & \quad = \int f(y_i(t)|I_i(t) = 1, \theta_i(t) \boldsymbol{Y}(t-1)) \\ & \qquad \cdot f(\theta_i(t)|I_i(t) = 1, \boldsymbol{Y}(t-1)) d\theta_i(t) \end{aligned} \tag{3.140}$$

Similar to the derivation in the static case, it can easily be seen that

$$y_i(t)|I_i(t)=1,\boldsymbol{Y}(t-1)\sim\mathcal{N}(\sqrt{\lambda_i(t)}\mu_i(t|t-1),\lambda_i(t)\sigma_i^2(t|t-1)+\sigma^2) \tag{3.141}$$

and

$$y_i(t)|I_i(t)=0,\boldsymbol{Y}(t-1)\sim\mathcal{N}(0,\sigma^2) \tag{3.142}$$

3.8.4 Derivation of cost of optimal allocation

From Section 3.4 and equation 3.62, we have the semi-omniscient allocations

$$\lambda_i^{semi}(t)=\frac{\lambda_{total}(t)\sqrt{\Pr(I_i(t)=1|\Psi(t-1))}}{\sum\limits_{j=1}^{Q}\sqrt{\Pr(I_j(t)=1|\Psi(t-1))}} \tag{3.143}$$

Define $w_i^{semi}(t)\sqrt{\Pr(I_i(t)=1|\Psi(t-1))}$. Then the associated cost for $c_i(t)=0$

$$\begin{aligned}
J_t(\lambda^{semi}) &= E\left[\sum_{i=1}^{Q}\frac{I_i(t)}{\lambda_i^{semi}(t)}\middle|\Psi(t-1)\right]\\
&= \frac{1}{\lambda_{total}(t)}E\left[\sum_{i=1}^{Q}\frac{I_i(t)\sum\limits_{j=1}^{Q}w_j^{semi}(t)}{w_i^{semi}(t)}\middle|\Psi(t-1)\right]\\
&= \frac{1}{\lambda_{total}(t)}\left(\sum_{i=1}^{Q}w_i^{semi}(t)\right)E\left[\sum_{i=1}^{Q}\frac{I_i(t)}{w_i^{semi}(t)}\middle|\Psi(t-1)\right]\\
&= \frac{1}{\lambda_{total}(t)}\left(\sum_{i=1}^{Q}w_i^{semi}(t)\right)\left(\sum_{i=1}^{Q}\frac{\Pr(I_i(t)=1|\Psi(t-1))}{w_i^{semi}(t)}\right)\\
&= \frac{1}{\lambda_{total}(t)}\left(\sum_{i=1}^{Q}w_i^{semi}(t)\right)^2.
\end{aligned} \tag{3.144}$$

Given the assumptions in this section, we know that the probabilities, $\Pr(I_i(t)=1|\Psi(t-1))$ can take one of three values. In the case that $I_j(t-1)=1$, then

$$\Pr(I_i(t)=1|\Psi(t-1),I_j(t-1)=1)=\begin{cases}(1-\alpha)\pi_0, & i=j\\ \frac{(1-\alpha)(1-\pi_0)}{|G(j)|}, & i\in G(j)\end{cases} \tag{3.145}$$

Moreover, since a target is added to the scene with probability β and location uniformly at random, we have

$$\Pr(I_i(t) = 1|\Psi(t-1)) = \beta/Q, \qquad j \notin \Psi(t-1)\forall j \in H(i) \tag{3.146}$$

where the condition $j \notin \Psi(t-1)\forall j \in H(i)$ is used to enforce the assumption that there are no interacting targets. Define

$$G(\Psi(t)) = \{j : j \in G(i), i \in \Psi(t-1)\}, \tag{3.147}$$

$$H(\Psi(t)) = \{j : j \in H(i), i \in \Psi(t-1)\}. \tag{3.148}$$

Then, given the assumption that there are $|\Psi(t)| = |\Psi(t-1)|$ targets and assuming that $|G(i)| = G$ for all i we have

$$\begin{aligned}
&\sum_{i=1}^{Q} \sqrt{\Pr(I_i(t) = 1|\Psi(t-1))} \\
&\quad = \sum_{\substack{i\in\mathcal{X} \\ i\notin H(\Psi(t-1))}} \sqrt{\frac{\beta}{Q}} + \sum_{i\in\Psi(t-1)} \sqrt{(1-\alpha)\pi_0} + \sum_{i\in G(\Psi(t-1))} \sqrt{\frac{(1-\alpha)(1-\pi_0)}{G}} \\
&\quad \approx Q\sqrt{\frac{\beta}{Q}} + |\Psi(t)|\sqrt{(1-\alpha)}\left(\sqrt{\pi_0} + G\sqrt{\frac{1-\pi_0}{G}}\right) \\
&\quad = \sqrt{Q\beta} + |\Psi(t)|\sqrt{(1-\alpha)}\left(\sqrt{\pi_0} + \sqrt{G(1-\pi_0)}\right)
\end{aligned} \tag{3.149}$$

This yields a cost

$$J_t(\lambda^{semi}) = \frac{\left[\sqrt{Q\beta} + |\Psi(t)|\sqrt{(1-\alpha)}\left(\sqrt{\pi_0} + \sqrt{G(1-\pi_0)}\right)\right]^2}{\lambda_{total}(t)} \tag{3.150}$$

3.8.5 Discussion of generalizations of state model and posterior estimation methods

As mentioned earlier, it is a difficult, if not intractable, problem to exactly estimate the posterior distribution of $\xi(t)$ given $\boldsymbol{Y}(t-1)$ that is required for our adaptive algorithms. We have provided a simple algorithm that approximates the posterior distribution under some restrictive assumptions. A simple way to alleviate these restrictions is to use a particle filter implementation for $\xi(t)$ or other approximate method (e.g., the extended and unscented Kalman filters).

Moreover, we have provided a particular state model that builds on our previous work with the inclusion of transition, birth, and death probabilities. However, there are many other models for dynamic state models, including linear and nonlinear motion models, targets that may occupy multiple adjacent cells, and various noise models. In any of these cases, one would have to use a different posterior estimation algorithm to provide estimates of $\Pr(I_i(t) = 1|\boldsymbol{Y}(t-1))$.

Future work plans to compare other posterior estimation algorithms such as the JMDP particle filter by Kreucher et al. [59] to the one presented in this thesis, as well as generalizations to more interesting dynamic state models.

3.8.6 Unobservable targets

One particular generalization of the measurement model that is used in this work is the inclusion of indicator variables for observable/unobservable targets. In many applications, certain locations may be obscured for short durations, such as locations in the null of a radar beam. Define

$$O_i(t) = \begin{cases} 1, & \text{Location } i \text{ is observable} \\ 0, & \text{Otherwise} \end{cases} \quad \text{for } i \in \{1, 2, \ldots, Q\}. \tag{3.151}$$

to be an indicator variable for the observability of the i-th location. Then the measurement model becomes

$$y_i(t) = \sqrt{\lambda_i(t)} I_i(t) O_i(t) \theta_i(t) + \varepsilon_i(t), \tag{3.152}$$

It is assumed that $\boldsymbol{O} = \{O_i(t)\}_{i,t}$ is known to the user a priori. Thus, we are required to estimate the densities:

$$f(\xi(t)|\boldsymbol{Y}(t'), \boldsymbol{O}) \tag{3.153}$$

for $t' = t, t-1$. We make the simplifying assumption that if $I_i(t-1) = 1$, then

$$O_i(t) = 1 \Leftrightarrow O_j(t) = 1, \forall j \in H(i) \tag{3.154}$$

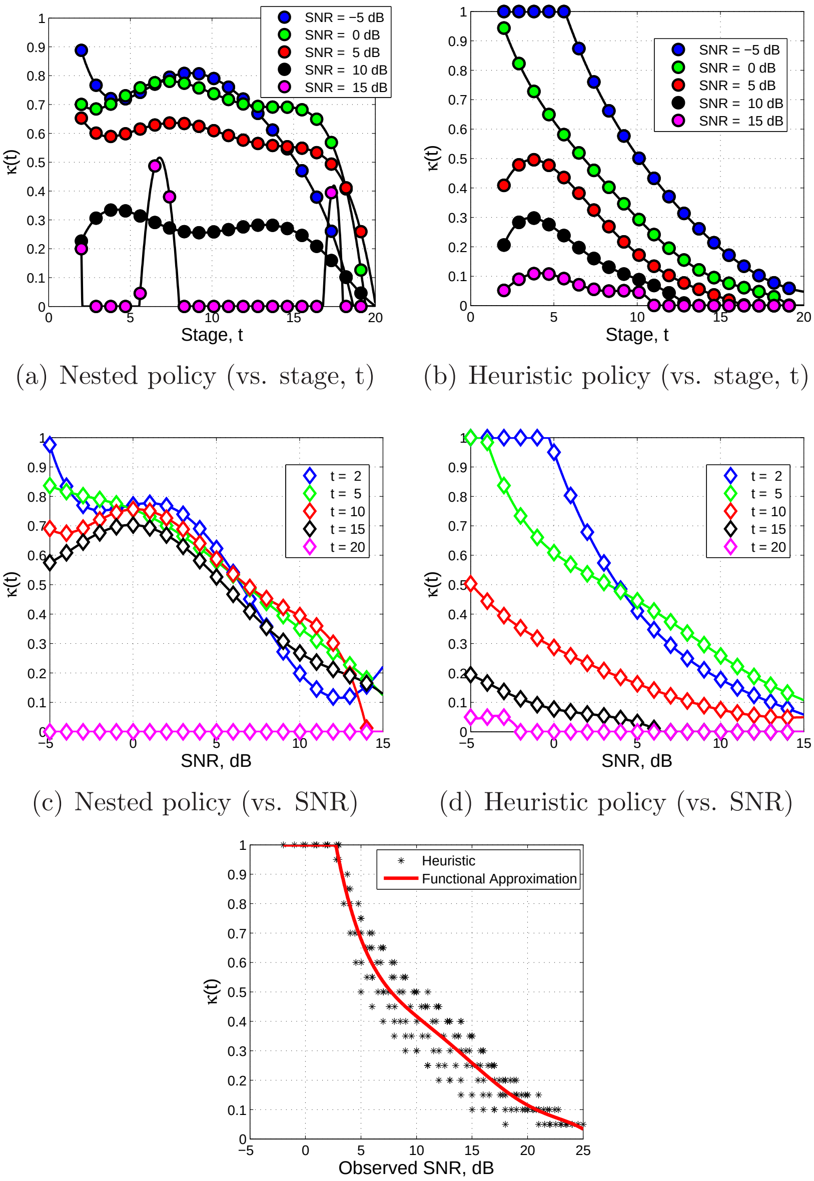
It can easily be seen that when $O_i(t) = 1$, we have the identical update equations to the fully observable case. However, when $O_i(t) = 0$, the predict equations remain the same as before, but the update equations are changed in the following manner:

$$f(I_i(t) = 1|\boldsymbol{Y}(t), O_i(t) = 0) = f(I_i(t) = 1|\boldsymbol{Y}(t-1)) \tag{3.155}$$

and the target amplitudes when $O_i(t) = 0$:

$$\hat{\theta}_i(t|t) = \hat{\theta}_i(t|t-1) \tag{3.156}$$

$$\hat{\sigma}_i^2(t|t) = \hat{\sigma}_i^2(t|t-1) \tag{3.157}$$



(a) Nested policy (vs. stage, t)

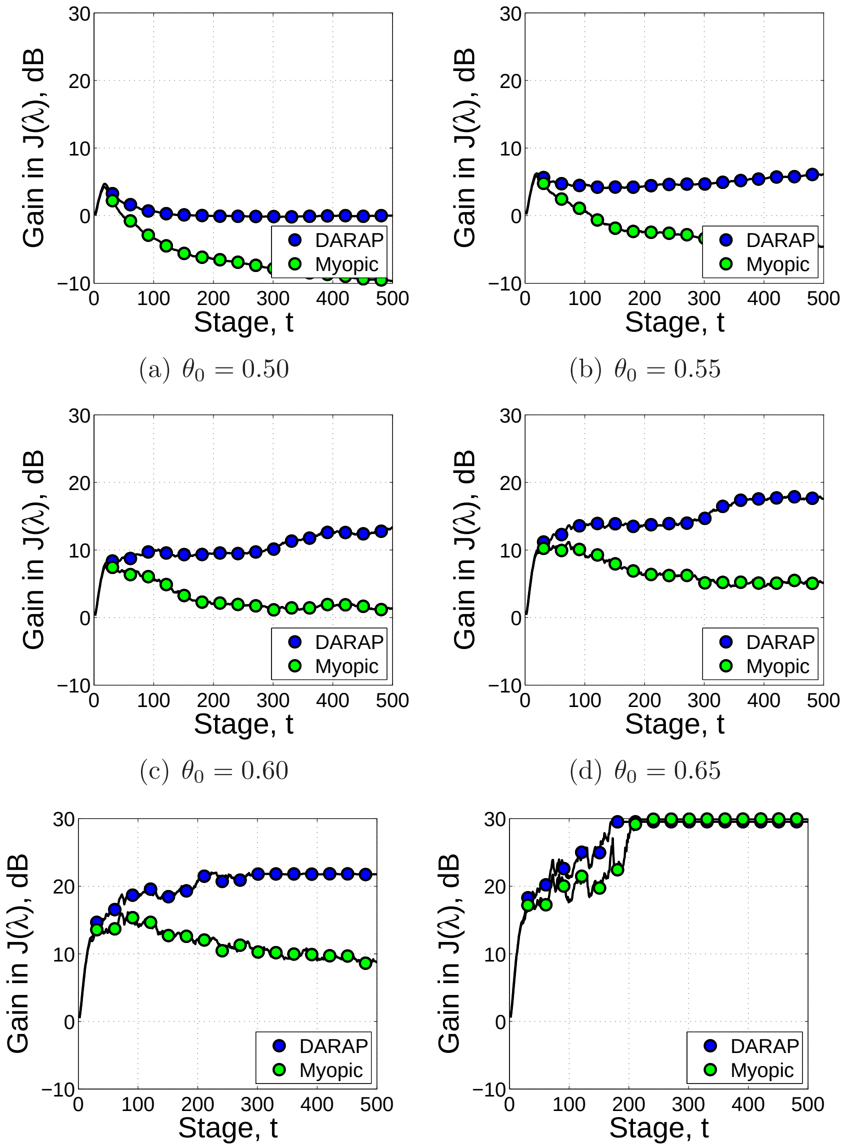
(b) Heuristic policy (vs. stage, t)

(c) Nested policy (vs. SNR)

(d) Heuristic policy (vs. SNR)

(e) Functional approximation to heuristic policy (vs. observed SNR)

Figure 3.4: This figure shows the selection of $\kappa_D^{(T)}(T)$ according to Algorithms 3 (nested) and 4 (heuristic) for policies of length $T = 20$. In (a) and (b), the selections are plotted against stage for the nested and heuristic strategies, respectively. In (c) and (d), the selections are plotted against SNR per stage for the nested and heuristic strategies, respectively. In (e), a functional approximation to the heuristic strategy is motivated by plotting the selections in (d) against observed SNR, which is defined in equation (3.83). The functional approximation is then given by the black line.



(a) $\theta_0 = 0.50$

(b) $\theta_0 = 0.55$

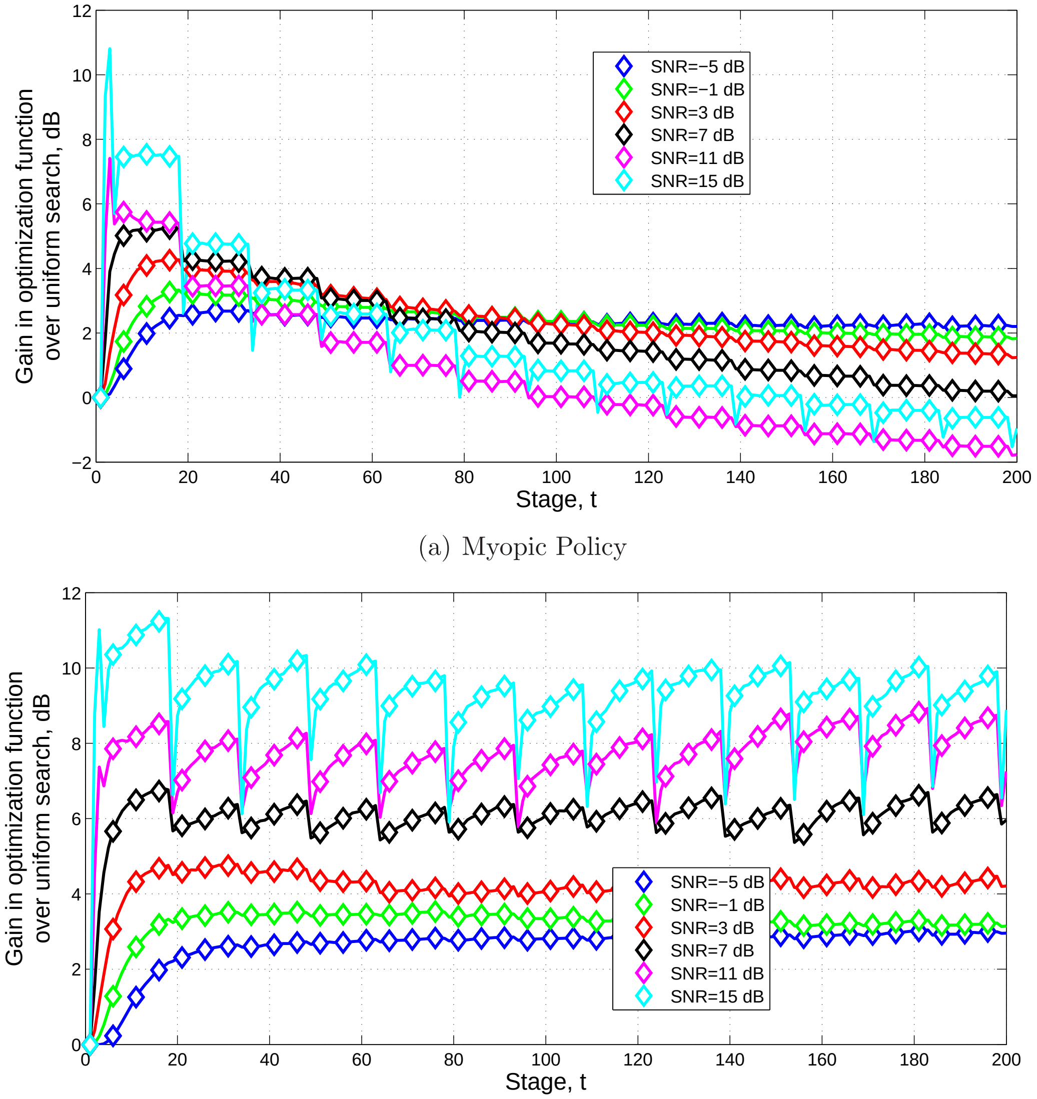
(c) $\theta_0 = 0.60$

(d) $\theta_0 = 0.65$

(e) $\theta_0 = 0.70$

(f) $\theta_0 = 0.75$

Figure 3.5: This figure shows a comparison of the proposed policy (D-ARAP, blue) with the myopic policy (green) as a function of gains in cost over a uniform search in a worst-case analysis (static, $\pi_0 = 1$), where the target returns $\theta_i(t)$ are set to various values, $\theta_0 < \mu_\theta = 1$. For low values of θ_0, noisy measurements cause missed targets that are never recovered by the myopic policy for $\theta_0 < 0.75$. On the other hand, D-ARAP has approximately monotonically increasing gains for all $\theta_0 > 0.5$, suggesting greater robustness to noise than the myopic policy. Moreover, even when $\theta_0 = 0.75$, D-ARAP converges to the optimal gain in fewer stages than the myopic policy.



(a) Myopic Policy

(b) D-ARAP Policy

Figure 3.6: This figure shows the performance gain (dB) in the expected value of the optimization function in equation (3.48) in the scenario with faulty measurements once every 15 stages, which causes the drops in performance at these stages. With the myopic policy shown in (a), this causes catastrophic failure for high SNR, in the sense that targets are lost and not recovered. Indeed, as t and SNR increase, the performance of the myopic policy trends downwards and eventually becomes worse than a uniform search. On the other hand, the D-ARAP (functional) policy shown in (b) has the ability to recover from misdetections, because it always allocates some resources to all cells.

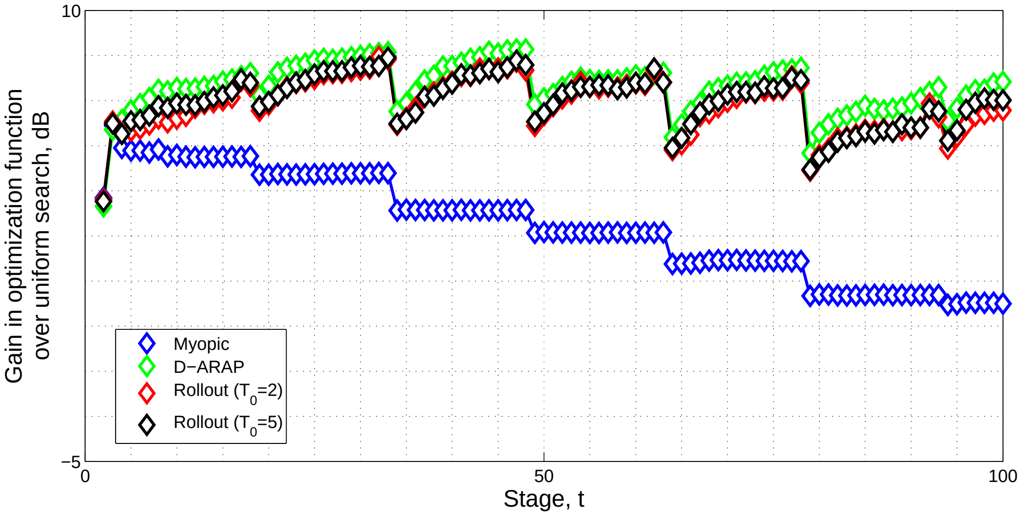


Figure 3.7: In this figure, the performance of two POMDP approximate solutions (a 2-stage rollout policy and a 5 stage-rollout policy) are compared against the myopic policy and D-ARAP for SNR = 10 dB in the case of faulty measurements once out of every 15 stages. It is seen that the POMDP solutions parallels the D-ARAP solution, which suggests that D-ARAP is close to optimal in this scenario, although at a fraction of the computational cost of the POMDP solutions.

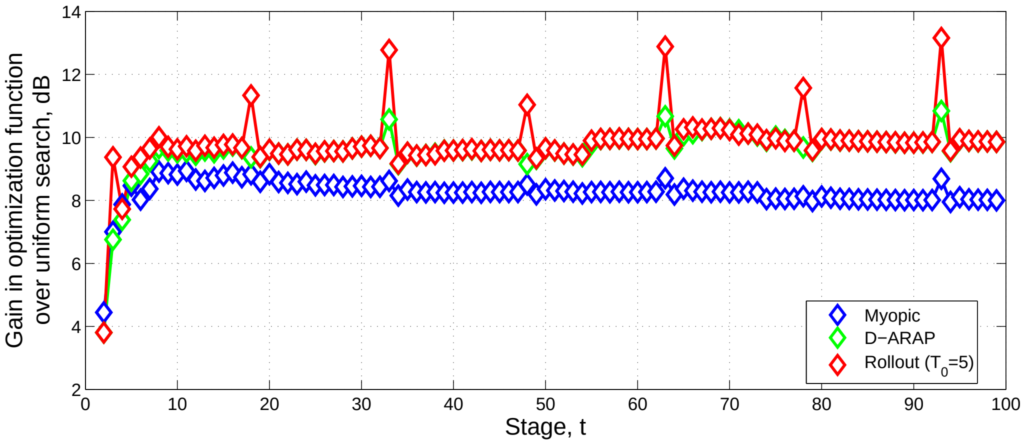
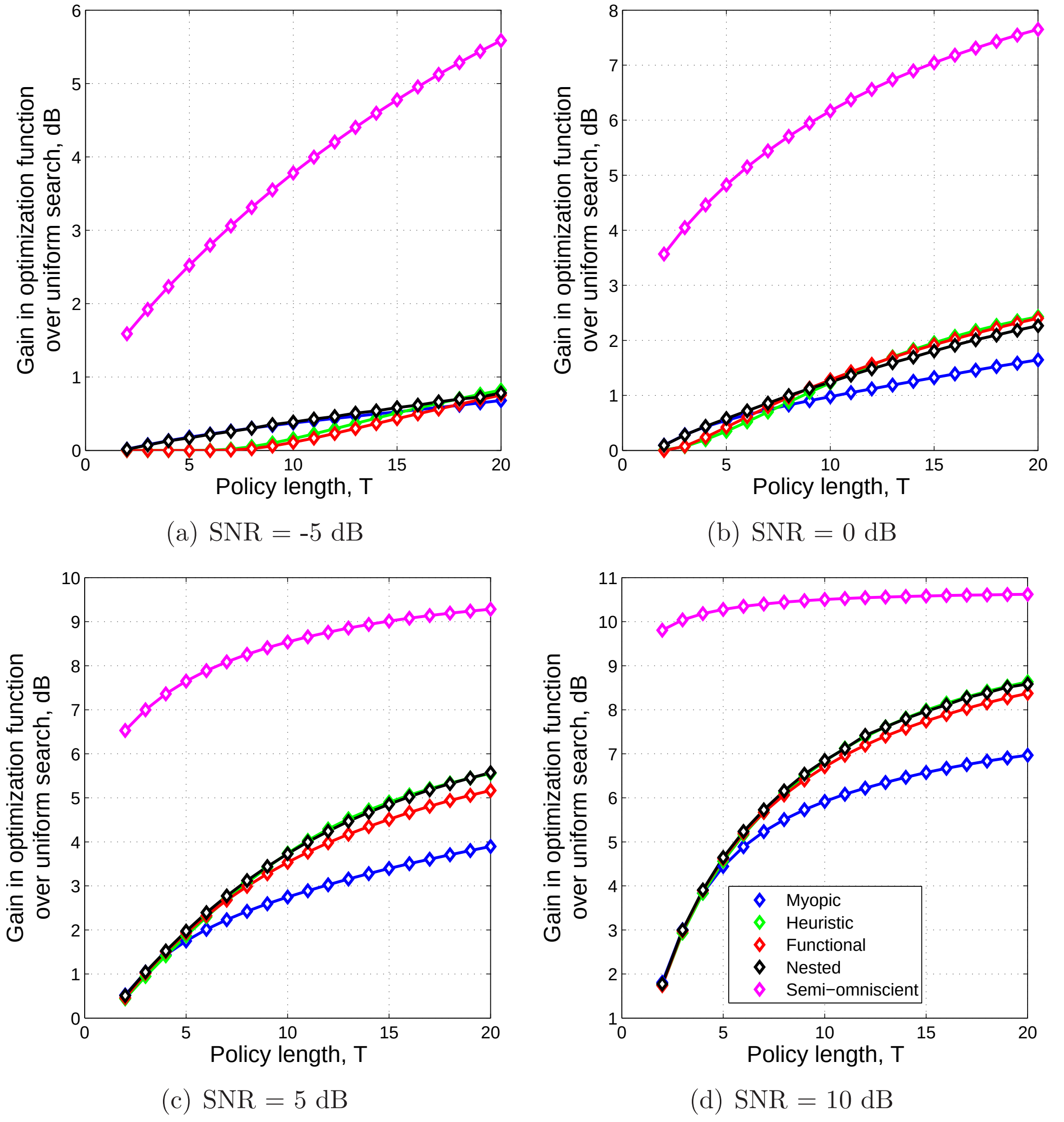


Figure 3.8: In this figure, the performance of a POMDP approximate solutions (2-stage rollout) is compared against the myopic policy and D-ARAP policy for SNR = 10 dB. In this scenario, the user has the ability to know when faulty measurements will occur and allocate resources differently. This is reflected in the fact that the POMDP solution has better performance during these faulty measurement periods (i.e., every 15 stages), as compared to D-ARAP and the myopic policy. Note that in the standard situation (i.e., without faulty measurements), D-ARAP performs very closely with the POMDP solution. On the other hand, the myopic policy continues to have a downward trend, even though no catastrophic events occur as in Figures 3.6 and 3.7.



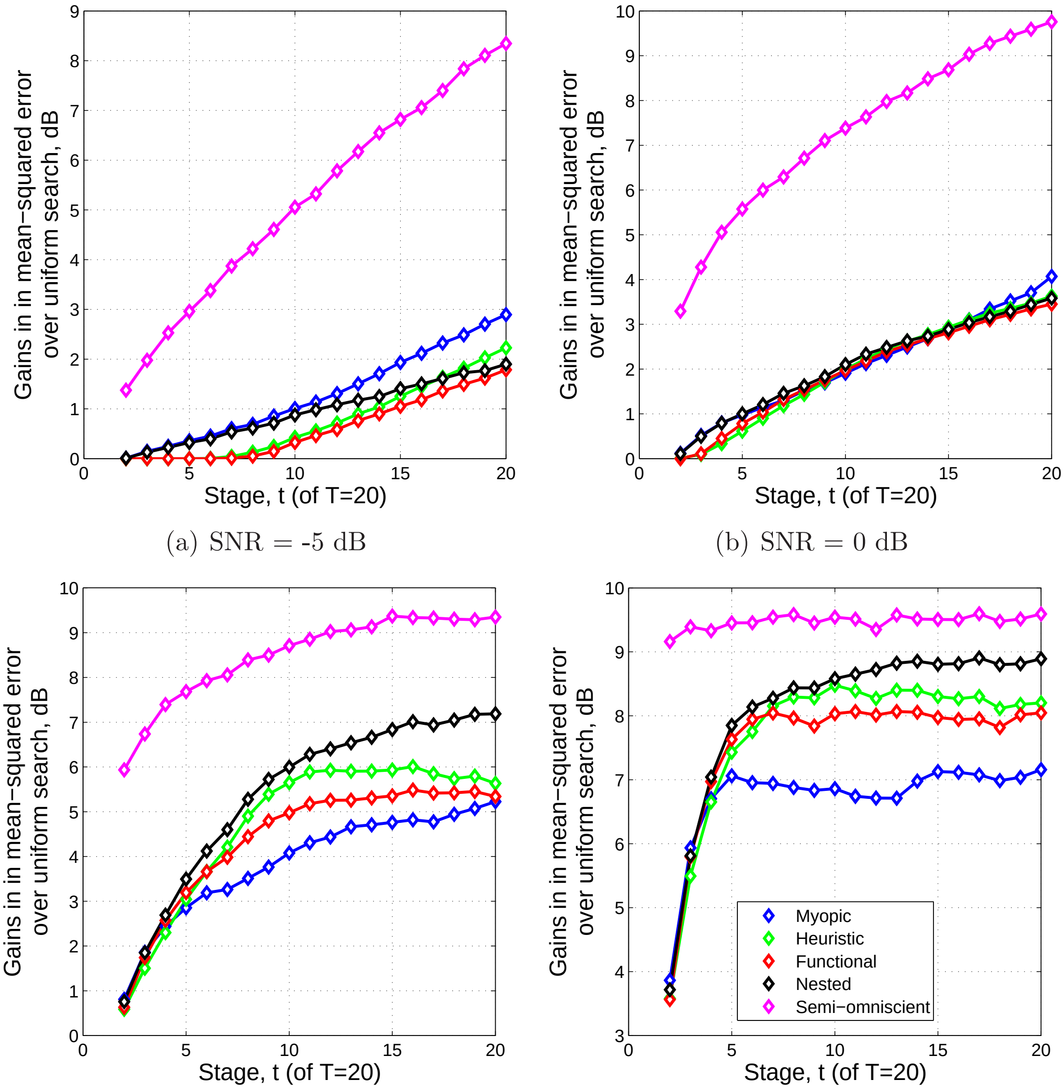
(a) SNR = -5 dB

(b) SNR = 0 dB

(c) SNR = 5 dB

(d) SNR = 10 dB

Figure 3.9: These plots compare the expected values of the cost (optimization function) given by equation (3.48) as function of the length of the policy, $T = 1, 2, \ldots, 20$. Gains over a uniform search (on a dB scale) are plotted for 5 alternative policies: a myopic policy (blue), the heuristic policy (green), the functional approximation to the heuristic policy (red), the nested policy (black), and the semi-omniscient oracle policy (magenta). Note that generally the nested policy has the highest gains in the optimization function among non-oracle policies. The differences are most apparent for higher SNR scenarios (c) and (d). Generally, the nested policy performs very similarly with the heuristic and functional policies, although those policies have much smaller computational burden. The myopic policy, on the other hand, has significantly worse performance as t or SNR increase.



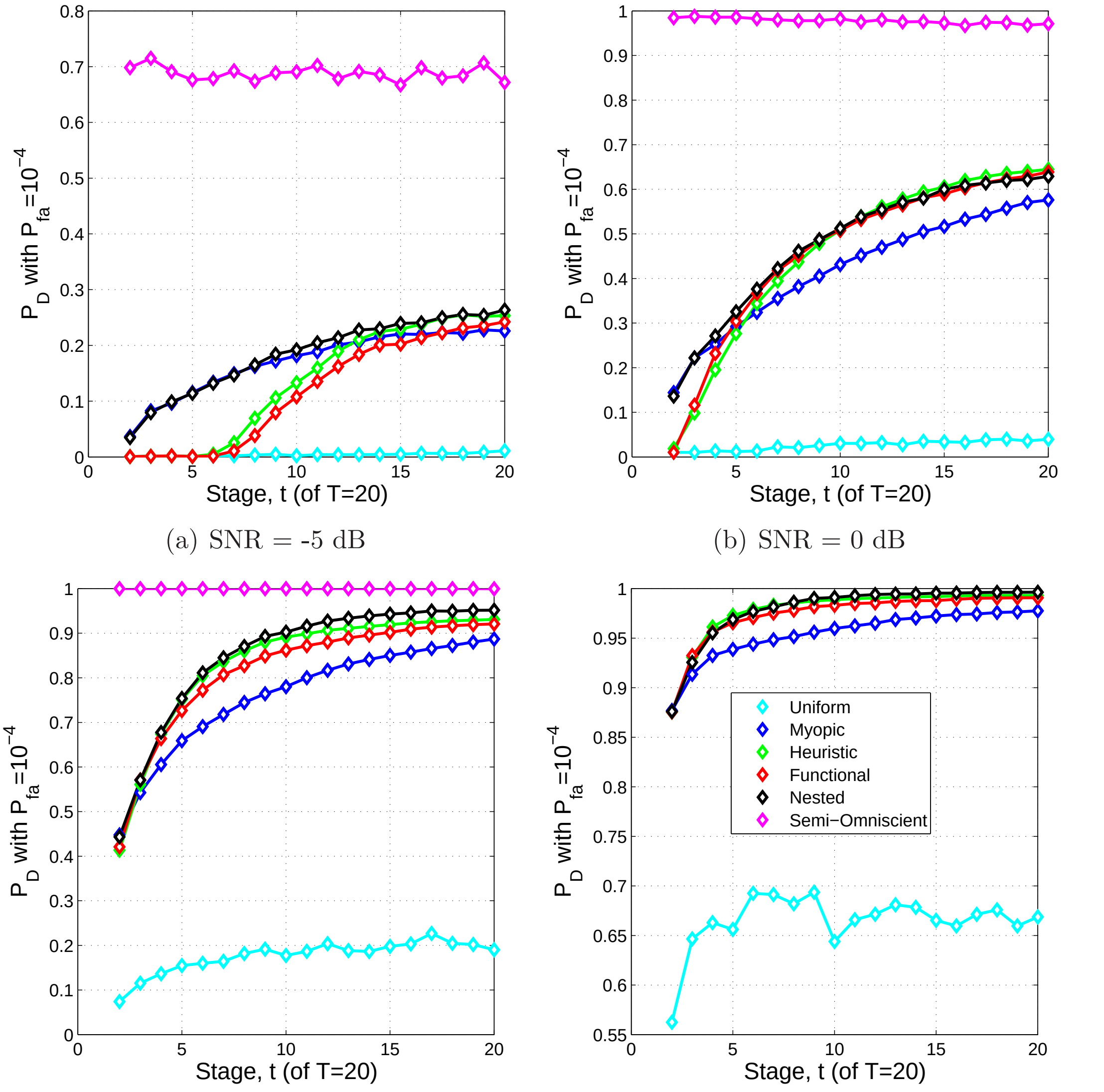
(a) SNR = -5 dB

(b) SNR = 0 dB

(c) SNR = 5 dB

(d) SNR = 10 dB

Figure 3.10: These plots compare the mean squared error of $\theta_i(t)$ within the region of interest (i.e., $I_i(t) = 1$) as function of the stage number, $t = 1, 2, \ldots, 20$. Gains over a uniform search (on a dB scale) are plotted for 5 alternative policies: a myopic policy (blue), the heuristic policy (green), the functional approximation to the heuristic policy (red), the nested policy (black), and the semi-omniscient oracle policy (magenta). Note that generally the nested policy has the highest gains in MSE among non-oracle policies. The differences are most apparent for higher SNR scenarios (c) and (d), with performance close to the optimal level as t gets large.



(a) SNR = -5 dB

(b) SNR = 0 dB

(c) SNR = 5 dB

(d) SNR = 10 dB

Figure 3.11: These plots compare the probability of detection for a fixed probability of false alarm (10^{-4}) as function of the stage number, $t = 1, 2, \ldots, 20$. The four subplots show different values of SNR per stage. Within each subplot, the blue curve represents the myopic policy, the green curve represents the heuristic policy, the red curve represents the functional policy, the black curve represents the nested policy, the magenta curve represents the semi-omniscient policy, and the cyan curve represents the uniform (or exhaustive) search. Note that generally the nested policy has the highest probability of detection among non-oracle policies, though it is barely distinguishable from the heuristic and functional policies. The myopic policy has lower probability of detections, while the uniform policy performs the worst of all alternatives.

CHAPTER IV

Sensor Management and Provisioning for Multiple Target Radar Tracking Systems

4.1 Introduction

In this chapter, we develop fundamental performance limits for system provisioning, i.e. the problem of determining the number of resources required to accomplish a complicated system level task such as tracking or discriminating between N targets. This is a central problem where the number of targets can easily exceed the available resources. Finding these performance limits is important in many environments, such as benchmarking, feasibility analysis, and performance tradeoff studies. Being able to determine the capabilities of tracking/discrimination is important in sensor management and development of new technologies.

The provided framework is generalizable to many applications, but we specialize to two specific contexts:

1. Sensor provisioning with a multistatic passive radar system. In particular, we consider the problem of opportunistic target tracking using the resources of wireless CDMA cellphone network. The available resource is the time required to collect measurements for all targets within the region of interest (ROI). In this analysis, it should be noted that the parameters of the CDMA system

are selected to be representative of a real system, but they do not necessarily represent any cellular CDMA protocol or operating conditions.

2. Computational provisioning with images formed from a synthetic aperture radar (SAR) system, as described in Chapter V The available resource is the computational time required to form the SAR images within the ROI.

This chapter describes a general approach to system provisioning for multiple 'sensor' systems that uses the guaranteed uncertainty management (GUM) philosophy. By system provisioning we mean using physical models for target detection and estimation to specify fundamental limits on performance (system stability, track entropy, occupancy rates) for a given system description, such as the number of sensors, location of sensors, and/or the desired standard errors.

The GUM approach is more conservative than standard stochastic scheduling approaches to radar provisioning. In particular, it carries strict and absolute guarantees on the probability of loss of track of the system. This is in contrast to average performance guarantees that have been previously adopted [52] by Hero for similar applications. By using this strict performance-constrained approach, the sensor management problem becomes non-stochastic and leads to strong results that could not easily be obtained in the less stringent stochastic scheduling context.

The rest of this chapter is organized as follows. In Sections 4.2 and 4.3, we present a service load model for both provisioning examples, where a service load model describes the amount of resources required to reduce uncertainty of a single tracked target to a prescribed level. In general, we look at the worst-case scenario when there is uncertainty in the target state. For example, moving targets in SAR images are likely to cause displacement and dispersion effects when there is uncertainty in their state. Thus, we include the worst-case errors when deriving the service load and system load models. However, with multistatic radars, we provide an additional evaluation tool for determining performance bounds by deriving the Fisher Information Matrix (FIM) for Doppler/range measurements in a given system configuration.

Section 4.4 proposes a simple resource allocation policy for tracking multiple targets based on the 'largest weighted queue length' policy by Wasserman et al. [89]. By utilizing load balance equations we develop stability conditions for guaranteeing bounded target uncertainties. This leads to deriving the maximum number of targets that can be tracked stably, the maximum spatial uncertainty of those targets, and the occupancy rate (i.e., system efficiency) for track-only provisioning. In Section 4.5, this theory is extended to a multi-purpose system that engages in activities other than tracking, such as discrimination and search. Finally, Section 4.6 presents a case-study analysis for SAR computational provisioning.

4.2 Target and system model: network provisioning for mulitstatic tracking

4.2.1 Target model

Assume that at time 0 a target is detected in a radar cell

$$\mathcal{C}_0 = \{z = (x, y) : -\sigma_x \leq x - \bar{x} \leq \sigma_x, -\sigma_y \leq y - \bar{y} \leq \sigma_y\} \tag{4.1}$$

where $\bar{z} = [\bar{x}, \bar{y}]$ is the center position of the cell. In this chapter, we consider targets in a 2-dimensional space, though the theory is easily extended to 3 dimensions. From a radar signal processing algorithm, an estimate of the target state and trajectory

$$\xi = \left\{\hat{z}, \dot{\hat{z}}\right\} \tag{4.2}$$

is extracted, along with a set of associated standard errors. This could be the output of a Kalman filter, sigma tracker, particle filter or other common tracking algorithm.

Table 4.1: Parameterizations for target estimates from a radar signal processing algorithm in the context of multistatic network provisioning

Context	Parameter	Description	Standard Errors
Multistatic	$\hat{x}, \hat{y}$	Position	σ_x, σ_y
	$\hat{\phi}$	Direction	σ_ϕ
	$\hat{v}$	Speed	σ_v

The particular parameterization of equation (4.2) depends on the sensor that is being used. Table 4.1 provides the parameterization in the multistatic network provisioning example.

From these estimates and standard errors a confidence region for the positions $(\hat{x}, \hat{y})$ having coverage probability of at least $1 - \varepsilon_T$ can be specified as

$$[\hat{x} - \sigma_x, \hat{x} + \sigma_x] \times [\hat{y} - \sigma_y, \hat{y} + \sigma_y] \tag{4.3}$$

With probability no less than $1 - \varepsilon_T$, after an elapsed time of τ seconds from the last revisit of the target, the above confidence region will map to the union of an uncountable number of segments. For the multistatic case, the union of confidence regions is complicated. Instead, we provide the circumscribing spatial uncertainty region:

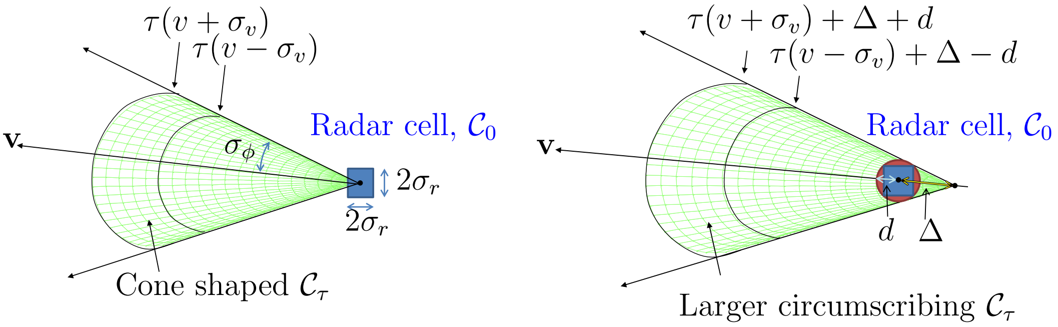
$$\mathcal{C}_\tau^{Multi} = \left\{ (r, \phi) : \begin{array}{c} -(d + \tau\sigma_v) \leq r - (\hat{r} + \hat{v}\tau + \Delta) \leq (d + \tau\sigma_v), \\ -\sigma_\phi \leq \phi - \hat{\phi} \leq \sigma_\phi \end{array} \right\}, \tag{4.4}$$

where $d = \sigma_r\sqrt{2}$ and $\Delta = d/\sin(\sigma_\phi)$.

Figure 4.1 provides an illustration of this union of uncertainty regions, as well as the circumscribing region. The areas of this uncertainty region can be described by

$$|\mathcal{C}_\tau^{Multi}| = 4\sigma_\phi[\hat{v}\sigma_v\tau^2 + (\hat{v}d + \sigma_v\Delta)\tau + \Delta d]. \tag{4.5}$$

Note that the uncertainty region grows superlinearly in τ, which reflects the underlying stability problem. When a target is not tracked sufficiently often, it becomes increasingly more difficult to find/track it as time progresses.


$\tau(v+\sigma_v)$
$\tau(v-\sigma_v)$
Radar cell, $\mathcal{C}_0$
v
σ_ϕ
$2\sigma_r$
$2\sigma_r$
Cone shaped $\mathcal{C}_\tau$
$\tau(v+\sigma_v)+\Delta+d$
$\tau(v-\sigma_v)+\Delta-d$
Radar cell, $\mathcal{C}_0$
v
d
Δ
Larger circumscribing $\mathcal{C}_\tau$

(a) Multistatic (single cell)

(b) Multistatic (all cells)

Figure 4.1: This figure shows the characterizations of the uncertainty region $\mathcal{C}_\tau$ in the multistatic network provisioning example. The blue rectangular regions show a small radar cell $\mathcal{C}_0$ that contains a target with high uncertainty immediately after revisit. The target's trajectory is given by (v, ϕ) with standard errors (σ_v, σ_ϕ). After τ seconds, a target with initial state $(x, y) \in \mathcal{C}_0$ will lie in the conical segment $\mathcal{C}_\tau$ with high probability. When the target can lie anywhere in $\mathcal{C}_0$, then we can only be confident that the target will lie in the union of all induced regions. In this situation, the union of the uncertainty regions is a difficult quantity to compute. Instead, we consider the larger circumscribing area as shown in (b).

4.2.2 Service load model

In general, we assume that the amount of time required to reduce the uncertainty of a target to a prescribed value can be described by the service load:

$$q(\tau) = \gamma(\tau) T_{GUM}, \tag{4.6}$$

where $\gamma(\tau)$ describes the growth of the uncertainty region and T_{GUM} is the amount of time required to scan a single cell in that region to guarantee some performance criterion.

It is assumed that a target's state can be resolved as long as it remains within the neighborhood of the radar cell. The size of the radar cell is a system-dependent quantity that may differ as a function of radar operating mode (SAR vs. multistatic) and the size of the radar beamwidth, among other parameters. We define T_{MAX} as an upper bound on the target revisit time that guarantees that the target can be resolved within the cell.

In general, we are interested in scenarios with $N > 1$ targets and $R \geq 1$ sensors. A sensor could be a physical quantity (such as the radar platform), a processing unit for creating images of the scene (SAR context), or a centralized processor for evaluating measurements from all sensors (multistatic context.) For R sensors and N targets, let $q_{r,n}(\tau)$ denote the load (in seconds) on the r-th sensor to revisit and update the n-th target after an elapsed time of τ:

$$q_{r,n}(\tau) = \gamma_{r,n}(\tau; \delta_d) T_{GUM}(r, n), \tag{4.7}$$

where $\gamma_{r,n}$ and $T_{GUM}(r, n)$ are analogously defined as target and sensor dependent quantities that guarantee the performance criteria.

Note that the service load function $\gamma(\tau)$ depends strongly on the application. For the multistatic passive radar system, we assume that we have $L = 1$ static transmitters (e.g., a cell tower) and M static receivers with known positions (e.g., cellular phones with GPS). The goal is to use bistatic range and Doppler measurements to estimate the target state. In particular, let the target state be given by $\{x, y, v, \phi\}$ so that

$$\mathbf{r}(0) = [x \quad y]^T \tag{4.8}$$

$$\mathbf{r}(\tau) = \mathbf{r}(0) + [(v \cos\phi)\tau \quad (v \sin\phi)\tau]^T \tag{4.9}$$

Let $d^{(m)}(\tau)$ be the bistatic range for the transmitter and m-th receiver, and let $d^{(m)}(\tau)$ be the bistatic Doppler shift for the the transmitter and m-th receiver. Table 4.2 defines many of the quantities used in this section. In general, we only observe the

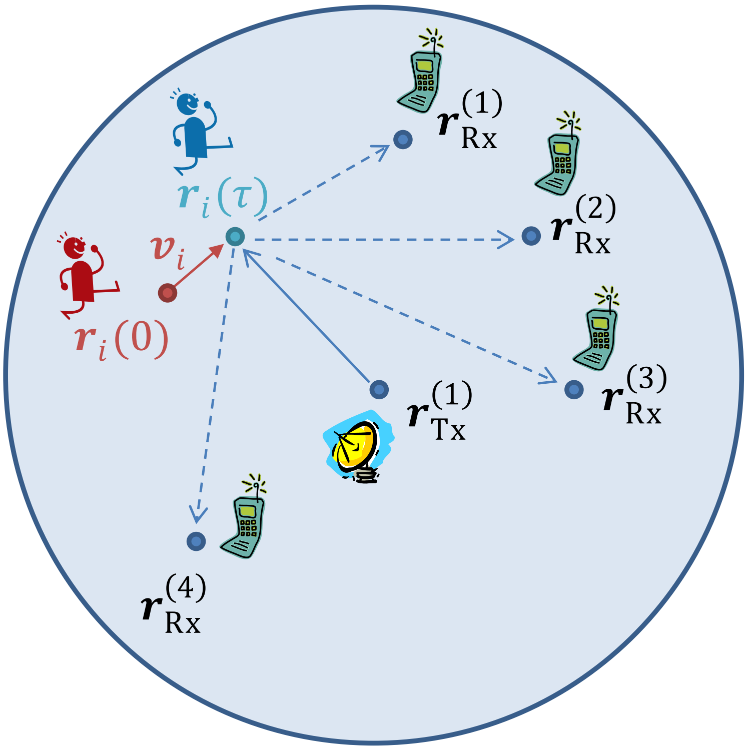

$\boldsymbol{r}_{\mathrm{Rx}}^{(1)}$
$\boldsymbol{r}_i(\tau)$
$\boldsymbol{v}_i$
$\boldsymbol{r}_{\mathrm{Rx}}^{(2)}$
$\boldsymbol{r}_i(0)$
$\boldsymbol{r}_{\mathrm{Tx}}^{(1)}$
$\boldsymbol{r}_{\mathrm{Rx}}^{(3)}$
$\boldsymbol{r}_{\mathrm{Rx}}^{(4)}$

Figure 4.2: This figure shows a possible multistatic passive radar situation with $L = 1$ static transmitters, $M = 4$ static receivers, and $N = 1$ targets of interest. While measurements are being collected, the target moves from an initial position in the direction of the shown velocity vector.

Table 4.2: Variables used for multistatic passive radar

Variable	Description	Definition
$\mathbf{r}_T^{(1)}$	Tx Position	
$\mathbf{r}_R^{(m)}$	m-th Rx Position	
$\boldsymbol{\delta}_T^{(1)}(\tau)$	Range (target-Tx)	$\delta_T^{(1)}(\tau) = \mathbf{r}(\tau) - \mathbf{r}_T^{(1)}$
$\boldsymbol{\delta}_R^{(m)}(\tau)$	Range (target-Rx m)	$\delta_R^{(m)}(\tau) = \mathbf{r}(\tau) - \mathbf{r}_R^{(m)}$
$d^{(m)}(\tau)$	Bistatic Range	$d^{(m)}(\tau) = \left\Vert\boldsymbol{\delta}_T^{(1)}(\tau)\right\Vert_2 + \left\Vert\boldsymbol{\delta}_R^{(m)}(\tau)\right\Vert_2$
$f^{(m)}(\tau)$	Bistatic Doppler shift	$d^{(m)}(\tau) = \frac{1}{\lambda}\left[\mathbf{v}\cdot\frac{\boldsymbol{\delta}_T^{(1)}(\tau)}{\left\Vert\boldsymbol{\delta}_T^{(1)}(\tau)\right\Vert_2} + \mathbf{v}\cdot\frac{\boldsymbol{\delta}_R^{(m)}(\tau)}{\left\Vert\boldsymbol{\delta}_R^{(m)}(\tau)\right\Vert_2}\right]$

bistatic range and Doppler shifts through noisy measurements:

$$\tilde{d}^{(m)}(\tau) = d^{(m)}(\tau) + \varepsilon^{(m)}(\tau) \tag{4.10}$$

$$\tilde{f}^{(m)}(\tau) = f^{(m)}(\tau) + \nu^{(m)}(\tau) \tag{4.11}$$

$$\tag{4.12}$$

for I.I.D. zero-mean Gaussian distributed $\varepsilon^{(m)}(\tau)$ and $\nu^{(m)}(\tau)$. Moreover, measurements are collected at a discrete set of times,

$$\tau_k = k\tau_{CPI}, \qquad k = 1, 2, \ldots, K \tag{4.13}$$

where τ_{CPI} is the coherent processing interval and K is chosen to satisfy a performance criterion such as probability of detection for a given SNR value. Define the set of measurements

$$\tilde{\boldsymbol{Y}} = [\tilde{D} \quad \tilde{F}]^T, \tag{4.14}$$

where

$$\tilde{D} = [\tilde{d}^{(m)}(\tau_k) : \quad m = 1, 2, \ldots, M, \quad k = 1, 2, \ldots, K], \tag{4.15}$$

$$\tilde{F} = [\tilde{f}^{(m)}(\tau_k) : \quad m = 1, 2, \ldots, M, \quad k = 1, 2, \ldots, K]. \tag{4.16}$$

In order to characterize the ‘quality’ of the measurements in a given configuration of transmitter and receivers, we derive the Fisher Information matrix (FIM) which provides a lower bound on the errors $\{\sigma_x, \sigma_y, \sigma_v, \sigma_\phi\}$ for unbiased estimators for the target state through the inverse of the FIM. Let $\Lambda_{m,n}$ be the (m, n)-th entry of the FIM. Then

$$\Lambda_{m,n} = -E\left[\frac{\partial^2}{\partial\theta_m\partial\theta_n}\log f(\boldsymbol{Y};\boldsymbol{\theta})\Big|\boldsymbol{\theta}\right], \qquad m, n = 1, 2, 3, 4 \tag{4.17}$$

where

$$\boldsymbol{\theta} = \{\sigma_x, \sigma_y, \sigma_v, \sigma_\phi\} \tag{4.18}$$

In particular, if we assume that

$$\mathrm{var}(\varepsilon^{(m)}(\tau_k)) = [\sigma^{(m)}(\tau_k)]^2, \tag{4.19}$$

$$\mathrm{var}(\nu^{(m)}(\tau_k)) = [\beta^{(m)}(\tau_k)]^2, \tag{4.20}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathrm{diag}\left\{[\sigma^{(m)}(\tau_k)]^2\right\}_m & 0 \\ 0 & \mathrm{diag}\left\{[\beta^{(m)}(\tau_k)]^2\right\}_m \end{bmatrix}, \tag{4.21}$$

then

$$\Lambda_{m,n} = \left[\frac{\partial \boldsymbol{Y}}{\partial \theta_m}\right]^T \boldsymbol{\Sigma}^{-1} \left[\frac{\partial \boldsymbol{Y}}{\partial \theta_n}\right] + \frac{1}{2}\mathrm{trace}\left(\boldsymbol{\Sigma}^{-1}\frac{\partial \boldsymbol{\Sigma}}{\partial \theta_m}\boldsymbol{\Sigma}^{-1}\frac{\partial \boldsymbol{\Sigma}}{\partial \theta_n}\right). \tag{4.22}$$

Define

$$\boldsymbol{J} = \boldsymbol{\Lambda}^{-1} \tag{4.23}$$

to be the inverse of the FIM. Then the diagonal elements of $\boldsymbol{J}$ are lower bounds on $\{\sigma_x, \sigma_y, \sigma_v, \sigma_\phi\}$. Let $\sigma_r = \max\{\sigma_x, \sigma_y\}$. Then we define the service load as

$$q(\tau) = \gamma(\tau; \sigma_r, \sigma_v, \sigma_\phi) T_{GUM}, \tag{4.24}$$

where

$$\gamma(\tau; \sigma_r, \sigma_v, \sigma_\phi) = \frac{|\mathcal{C}_\tau(\tau; \sigma_r, \sigma_v, \sigma_\phi)|}{|\mathcal{C}_0(\tau; \sigma_r, \sigma_v, \sigma_\phi)|} \tag{4.25}$$

is the growth of the uncertainty region, $\mathcal{C}_\tau$ is defined by 4.28, and

$$T_{GUM} = K\tau_{CPI}. \tag{4.26}$$

Note that this definition for $q(\tau)$ may be overly optimistic if the bounds on $\{\sigma_x, \sigma_y, \sigma_v, \sigma_\phi\}$ are unrealistic. However, we may still use this formulation in order to understand the basic performance bounds of a multistatic passive radar system.

Table 4.3: Parameterizations for target estimates from a radar signal processing algorithm in the context of SAR computational provisioning

Context	Parameter	Description	Standard Errors
SAR	$\hat{x}, \hat{y}$	Position	σ_x, σ_y
	$\hat{v_x}, \hat{v_y}$	Velocity	σ_{vx}, σ_{vy}
	$\hat{a_x}, \hat{a_y}$	Acceleration	σ_{ax}, σ_{ay}

4.3 Target and system model: SAR computational provisioning

The target and system models can be analogously defined in the context of SAR computational provisioning. In this case, we consider the parameterization of the target state given by Table 4.3. Moreover, the confidence region for the positions $(\hat{x}, \hat{y})$ after τ seconds is given by

$$\mathcal{C}_\tau^{SAR} = \left\{ (x, y) : \begin{array}{c} -\varepsilon_x(\tau) \le x - \hat{r}_x(\tau) \le \varepsilon_x(\tau), \\ -\varepsilon_y(\tau) \le y - \hat{r}_j(\tau) \le \varepsilon_y(\tau) \end{array} \right\}, \tag{4.27}$$

where $\varepsilon_j(\tau) = \sigma_j + \sigma_{vj}\tau + \sigma_{aj}\tau^2/2$ and $\hat{r}_j(\tau) = \hat{j} + \hat{v}_j\tau + \hat{a}_j\tau^2/2$ for $j = x, y$.

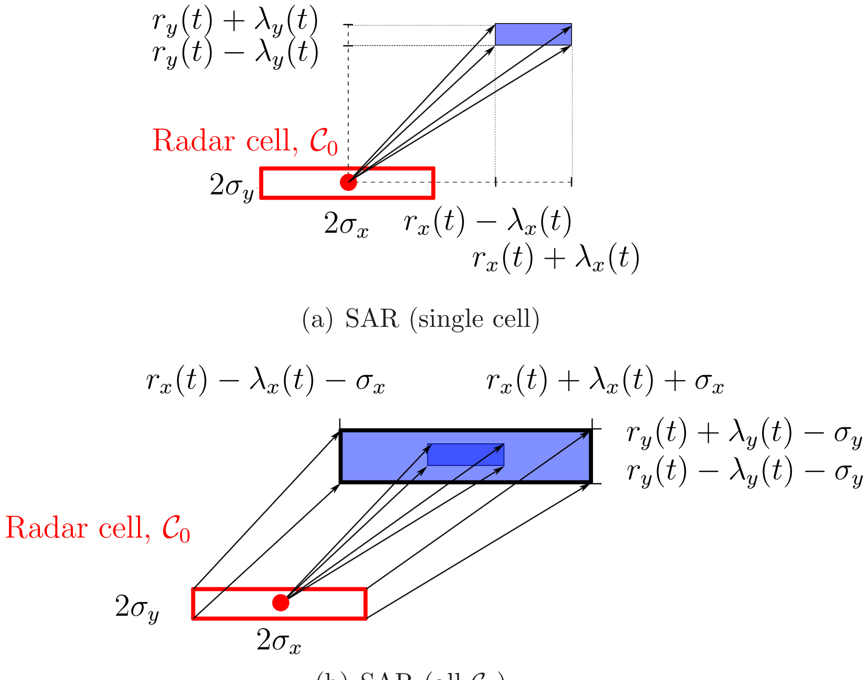
This region is illustrated in Figure 4.3, and the associated area can be computed as

$$|\mathcal{C}_\tau^{SAR}| = 4\epsilon_x(\tau)\epsilon_y(\tau), \tag{4.28}$$

$$\tag{4.29}$$

Note that similar to the multistatic case, the uncertainty region grows superlinearly in τ.

In order to derive a service load model in the SAR context, we assume that streaming samples from a single SAR sensor are available from an X-band sensor with standard parameters ($f_0 \approx 10$ GHz, $BW \approx 500$ MHz, $\tau_{PRI} \approx 10^{-3}$). Without loss of generality, we will assume that the radar platform travels in the x-direction.


$r_y(t) + \lambda_y(t)$
$r_y(t) - \lambda_y(t)$
Radar cell, $\mathcal{C}_0$
$2\sigma_y$
$2\sigma_x$
$r_x(t) - \lambda_x(t)$
$r_x(t) + \lambda_x(t)$
$r_x(t) - \lambda_x(t) - \sigma_x$
$r_x(t) + \lambda_x(t) + \sigma_x$
$r_y(t) + \lambda_y(t) - \sigma_y$
$r_y(t) - \lambda_y(t) - \sigma_y$
Radar cell, $\mathcal{C}_0$
$2\sigma_y$
$2\sigma_x$

(a) SAR (single cell)

(b) SAR (all $\mathcal{C}_0$)

Figure 4.3: This figure shows the characterizations of the uncertainty region $\mathcal{C}_\tau$ for the SAR computational provisioning example. The red rectangular regions show a small radar cell $\mathcal{C}_0$ that contains a target with high uncertainty immediately after revisit. The target's trajectory (v_x, v_y, a_x, a_y) is known with standard errors $(\sigma_{vx}, \sigma_{vy}, \sigma_{ax}, \sigma_{ay})$. Thus, we can be confident that a target at the center of the radar cell will lie in the blue rectangular region after τ seconds as in (a). When the target can lie anywhere in $\mathcal{C}_0$, then we can only be confident that the target will lie in the union of all induced rectangular regions as depicted by the blue region in (b). For this figure, the notation is defined with $\lambda_i(t) = \sigma_{vi}t + \sigma_{ai}t^2$ for $i = x, y$.

We are interested in the computational burden associated with standard image formation from the radar samples using back-projection as described by Soumekh [82]. The number of FLOPs associated with this process is proportional to the number of radar samples N_p and the number of pixels in the formed image, also proportional to N_p. The required time to detect and/or track within a target cell is then

$$T_{GUM} = \kappa N_p^2, \quad \kappa = \alpha_{radar} \tau_{CPU} \tag{4.30}$$

where τ_{CPU} is the number of seconds/FLOP associated with the CPU and α_{radar} is the number of FLOPs/N_p^2 that is dependent on the radar. For concreteness in this chapter, we assume that $\kappa \approx 3e10^{-7}$ using a 2.8 GHz CPU.

In SAR tracking non-zero velocities can cause errors in the cross-range (x-) direction as described by Fienup [42]. Note that these errors will depend only on the standard errors $(\sigma_{vx}, \sigma_{vy}, \sigma_{ax}, \sigma_{ay})$, since images can be focused to $\hat{v}_x, \hat{v}_y, \hat{a}_x, \hat{a}_y$ with no additional computational cost. For a maximum error of δ_d, this augmented region and its associated area is

$$h(\mathcal{C}_\tau, \delta_d) = \{(x, y) : (u, v) \in \mathcal{C}_\tau, x \in u + [-\delta_d, \delta_d], y = v\} \tag{4.31}$$

$$|h(\mathcal{C}_\tau, \delta_d)| = 4(\varepsilon_x(\tau) + \delta_d)\varepsilon_y(\tau) \tag{4.32}$$

We are interested in resolving targets within a cell size, denoted $|\mathcal{C}_0|$. Due to real-time constraints, we assume that only streaming data is available. Thus, after τ seconds since the last revisit, the system will be occupied by the task of forming subimages in $h(\mathcal{C}_\tau, \delta_d)$ to reduce uncertainty on the target parameters back down to a $1 - \varepsilon_T$ confidence region of size $|\mathcal{C}_0|$. Thus, the load on the CPU is given by

$$q(\tau) = \gamma(\tau; \delta_d) T_{GUM}, \tag{4.33}$$

where $\gamma(\tau;\delta_d) = \frac{|h(\mathcal{C}_\tau,\delta_d)|}{|h(\mathcal{C}_0,\delta_d)|} - 1$ is the growth of the confidence region. Note that δ_d does not depend on τ, but only on the target's trajectory (v_x, v_y, a_x, a_y), which is arbitrary for any target, and the number of pulses, N_p, which is fixed by the user.

4.4 Guaranteed uncertainty management

The problem of utilizing available sensors in an optimal fashion to detect and track targets falls in the framework of dynamic scheduling of multiple servers to multiple queues (targets) by Brémaud [18] and Wasserman et al. [89]. The sensor manager must assign sensors to queues of target-revisit jobs in queues that grow as time elapses. Each job may have different service requirements. Generally, solving for the optimal allocation of servers to queues is a difficult, if not intractable, problem. However, several sub-optimal strategies have been proposed. A suboptimal prioritized longest queue (PLQ) strategy is to assign free servers to the longest queues, where each queue is processed by the server that is best matched to the service requirements. The following implementation of this strategy is the 'largest weighted queue length' policy proposed by Wasserman et al. [89] for heterogeneous multiqueueing systems. Let $\mathcal{N} \subset \{1, 2, \ldots, N\}$ be the number of target tracks not in the process of being revisited. Then Algorithm 6 provides the Prioritized longest queue (PLQ) sensor scheduling policy.

Algorithm 6. *Prioritized longest queue sensor scheduling policy*

When a sensor r is unoccupied and available for assignment to updating a target track then either

1. idle *the sensor if all target tracks are in process of being revisited ($\mathcal{N}$ is empty).*
2. deploy *the sensor on the target $n \in \mathcal{N}$ that maximizes the weighted service time $\max_{n\in\mathcal{N}} q_{r,n}(\tau_n)$, where τ_n is the elapsed time since the last revisit of target n.*

4.4.1 Balance equations guaranteeing system stability

Balance equations for stable operation of the system are equations that guarantee that at the time of revisit of a target its service load has not grown larger than it was at the previous revisit. With a single sensor, we drop the index r from $q_{r,n}(\tau)$. Let $q^{(n)}(\tau)$ be the service load to the n-th target chosen according to the PLQ policy. For $n = 1, 2, \ldots, N$, we have

$$q^{(n)}(\tau) = \max_{j\in\mathcal{N}(n)} q_j(q^{(n-1)}(\tau) + \tau), \tag{4.34}$$

where $q^{(0)}(\tau) = 0$. To simplify notation, we assume that the targets have been ranked in decreasing order of service load, so that

$$\arg\max_{j\in\mathcal{N}(n)} q_j(q^{(n-1)}(\tau) + \tau) = n, \tag{4.35}$$

and $q^{(n)}(\tau) = q_n(q^{(n-1)}(\tau) + \tau)$. Next define the system loading function

$$Q^{(N)}(\tau) = \sum_{i=1}^{N} q^{(i)}(\tau), \tag{4.36}$$

which is stable when $Q^{(N)}(\tau) < \tau$ (critically when $Q^{(N)}(\tau) = \tau$). If a solution exists, let $\tau = \tau^*$ be the solution of the balance equation

$$Q^{(N)}(\tau) = \tau. \tag{4.37}$$

Proposition IV.1. *For a single sensor tracking N targets the PLQ policy is stable, in the sense that the system maintains bounded tracking errors, if the following conditions hold:*

1. *a solution to (4.37) exists;*
2. *the revisit rate is at least $1/\tau^*$;*
3. *The target can be resolved, so that $\tau^* \ll T_{MAX}$.*

The value τ^* can be interpreted as the steady state total time required for the sensor to cycle through a complete sequence of target revisits. The stability result of Proposition IV.1 is tight in the sense that the system becomes unstable if Conditions 1 and 2 are not satisfied. When stability of the PLQ policy is guaranteed, we have a tight bound on the associated tracking error.

Corollary IV.2. *If the system is stable in the sense of Proposition IV.1, then the track uncertainty region of the i-th target will never exceed $H^*(i) = \ln|\mathcal{C}_{\tau^*}(i)|$.*

The proof of the above proposition is straightforward but we do not provide details here. The full proof relies on the fact that $q^{(i)}(\tau)$ is monotonic increasing in τ . We then use mathematical induction to obtain equations (4.34) as the time required to service the targets, and apply standard load balancing condition of optimal scheduling theory to obtain (4.37).

4.4.2 A simple slope criterion for stability

The system load function $Q^{(N)}(\tau)$ defined in (4.36) is zero at $\tau = 0$ and is smooth, differentiable, and monotonic increasing. Thus a necessary condition for the balance equation (4.37) to have a solution is that its derivative be less than or equal to 1 at the point $\tau = 0$. By induction the derivative $[Q^{(N)}]'(0) = dQ^{(N)}(\tau)/d\tau|_{\tau=0}$ can be shown to be of the form:

$$[Q^{(N)}]'(0) = \sum_{j=1}^{N}\sum_{k=1}^{j}\prod_{i=N-k+1}^{N} q_i'(0) \leq \sum_{j=1}^{N}\sum_{k=1}^{j}(q_0')^k, \tag{4.38}$$

where we have defined $q_0' = \max_i q_i'(0)$. If $\min_i q_i'(0) > 1$, then necessarily $[Q^{(N)}]' > 1$ so that $Q^{(N)}(\tau) > \tau$ and the system is unstable. If $q_0' < 1$, then the system may be stable. To obtain closed form results we will derive sufficient conditions on N that guarantee stability by using the upper bound on the right of (4.38) instead of the exact expression in the middle of (4.38). This upper bound is attained when all service load functions are identical, $q_i(0) = q_j(0)$ in which the conditions derived below will also be necessary. Therefore, the conditions will be tight for a worst case scenario but will be more stringent than might be required for a typical scenario. As $q_0' \geq 0$, the series summation formula applied to the right hand side of (4.38) gives the following proposition:

Proposition IV.3. *A solution τ^* to the balance equation (4.37) exists if and only if*

$$[Q^{(N)}]'(0) = \frac{q_0'}{1-q_0'}\left(N - \frac{q_0'}{1-q_0'}(1-[q_0']^N)\right) < 1 \tag{4.39}$$

We can obtain q_0' by differentiating (4.28) or (4.5) plugging into (4.6), and evaluating at $\tau = 0$. For the SAR context, this yields

$$q_0' = \kappa N_p^2(\sigma_x\sigma_y)^{-1}[\sigma_{vx}\sigma_y + \sigma_{vy}(\sigma_x + \delta_d)]. \tag{4.40}$$

For the multistatic radar, we have

$$q_0' = \left(\frac{\sigma_v + v\sin\sigma_\phi}{\sigma_r\sqrt{2}}\right) K\tau_{CPI}. \tag{4.41}$$

Define N_{max} as the maximum value of N such that the inequality in Proposition IV.3 is satisfied. When the sensor is tasked to track N_{max} targets then the system will be stable (however, we must still verify that the associated τ^* is such that condition 3 of Proposition IV.1 is satisfied). In the case $N = N_{max}$ the sensor is fully utilized and operating at maximum efficiency. When q_0' is small, N_{max} can be found approximately as

$$N_{max} = (1 - q_0')/q_0' + q_0'/(1 - q_0') \tag{4.42}$$

Furthermore, since $0 \leq 1 - [q_0']^N \leq 1$, we can assert that if the number of targets N exceeds N_{max} in (4.42), then no solution to the balance equations exists and the system diverges.

4.4.3 Extension to multiple sensors

When there are $R > 1$ sensors to manage we can obtain stability conditions in a similar manner to the previous section. Define b as the ratio of targets per sensor

$$b = \text{ceil}(N/R) \tag{4.43}$$

Define $q(\tau) = \max_{n,r} q_{r,n}(\tau)$ and the service load, $q^{(b)}(\tau) = q(q^{(b-1)}(\tau) + \tau)$. In analogy to the previous section, the system loading function is defined as

$$Q^{(b)}(\tau) = \sum_{i=1}^{b} q^{(i)}(\tau). \tag{4.44}$$

Stability conditions and slope conditions can be derived in a similar fashion to the previous section by replacing N with $b = \text{ceil}(N/R)$. The details are omitted here, but can be found in the technical report by Hero et al. [51].

4.4.4 Determining track-only system occupancy

We can use the Propositions to determine the efficiency of the system in terms of its occupancy rates, defined as one minus the proportion of time a sensor in the system is idle. We assume that the sensors are scheduled under the PLQ policy. In steady state a stable system of R sensors will be at maximum utilization when the system is critically stable. This occurs when there are approximately $b^* = N/R$ targets per sensor where b^* is the solution to the equation

$$\frac{q_0'}{1-q_0'}\left(b - \frac{q_0'}{1-q_0'}(1-[q_0']^b)\right) = 1 \tag{4.45}$$

Define $N_{max} = \text{floor}(b^* R)$. At this critically stable operating point of N_{max} targets, the sensors are fully occupied performing just-in-time revisits of the targets. In this case the maximum service load that each target places on the system is $Q^{(N_{max}/R)}(\tau^*)$ where τ^* is the solution of $Q^{(N_{max}/R)}(\tau) = \tau$. When the same system is assigned to track a fewer number $N < N_{max}$ of targets, there will be idle time. We define the occupancy of the track-only system as $\rho = \tau^*/\tau_\varepsilon$, where τ_ε is the value of τ that satisfies

$$Q^{(N/R)}(\tau) = Q^{(N_{max}/R)}(\tau^*). \tag{4.46}$$

The interpretation is that τ_ε is operating point of the system that results in the same loading for the underloaded system tracking N targets as the fully loaded system tracking N_{max} targets.

4.5 Multi-purpose system provisioning

Finally we turn to scenarios when the system may be engaged in other tasks in addition to tracking. From a computational standpoint, this could be as basic as time needed for transfer of data and communication. At a more abstract level, tasks could include discrimination of targets and/or wide area search for new targets. This is handled by building in headroom into the track update stability equations. Let Δ be the additional load in seconds spent after each revisit on tasks other than tracking. Consider the case of a single sensor and N targets. For a given Δ, the stability condition is that there must exist a solution, $\tau = \tau^*$ such that

$$Q^{(N)}(\tau, \Delta) + N\Delta = \tau, \tag{4.47}$$

where

$$Q^{(N)}(\tau, \Delta) = \sum_{i=1}^{N} q^{(i)}(\tau, \Delta) \tag{4.48}$$

and

$$q^{(N)}(\tau, \Delta) = q_N(q^{(N-1)}(\tau) + \tau + \Delta). \tag{4.49}$$

Note that since the q_i's are monotonically increasing, we have the bound

$$Q^{(N)}(\tau, \Delta) \leq Q^{(N)}(\tau + \Delta), \tag{4.50}$$

where $Q^{(N)}(\tau)$ is the simpler function defined in (4.36). Therefore, for specified Δ, a sufficient condition for stability is that there exist a $\tau = \tau^*$ such that

$$Q^{(N)}(\tau + \Delta) + N\Delta = \tau. \tag{4.51}$$

Re-expressing this in terms of the variable $u = \tau + \Delta$, we have the equivalent condition that there exist a solution $u = u^*$ to

$$Q^{(N)}(u) = u - (N+1)\Delta. \tag{4.52}$$

4.5.1 Load margin, excess capacity, and occupancy

The load margin represents the maximum additional load that can be accommodated by a tracking system that must perform joint operations such as tracking, detection, etc. The load margin Δ_{max} is defined as the maximum value Δ for which a solution u to (4.52) exists. When there are N targets and the multi-purpose system spends $\Delta \leq \Delta_{max}$ seconds per update performing other tasks we define the excess capacity

$$c_{excess}(\Delta) = 1 - \Delta/\Delta_{max}. \tag{4.53}$$

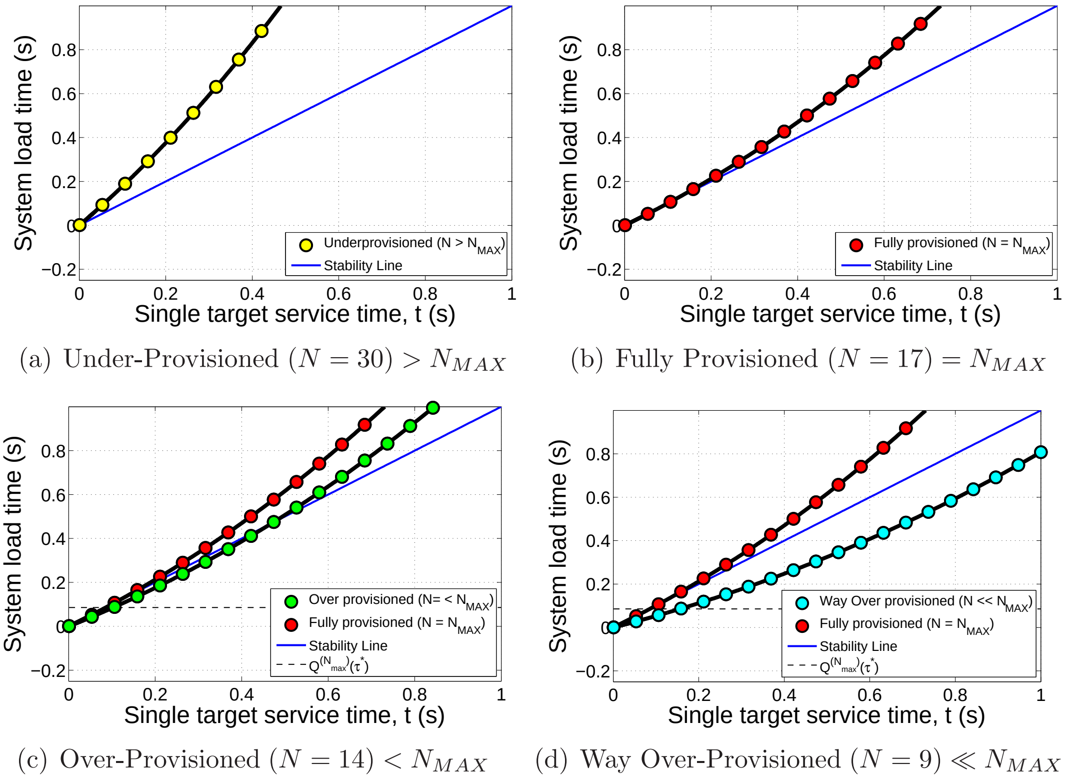
Likewise, we define the multi-purpose system occupancy as

$$\rho(\Delta) = 1 - [(\Delta_{max} - \Delta)N]/Q^{(N)}(u^*). \tag{4.54}$$

4.6 Application: SAR computational provisioning

For specified standard errors on tracking accuracy, e.g., available from Kalman tracking covariance estimates, the above results can be used to generate tables and curves on the required number of sensor, their revisit rates, and their occupancy, for tracking N targets with prescribed track error (entropy). In this section, we focus on computational provisioning within the SAR context. In particular, we assume we have

1. A radar with parameters defined in Section 4.3.
2. $N_p = 250$ corresponding to P_f=10^{-6}, $P_d > 0.99$ for detecting a Swerling II target at a SNR=0 dB [66], Fig. 12.23.
3. Target cell given by (σ_x, σ_y)=$(6, 0.3)$ m.
4. Target trajectory, (v_x, v_y)=$(5, 5)$ m/s, (a_x, a_y)=$(0, 0)$ m/s^2, with std. errors $(\sigma_{vx}, \sigma_{vy})$=(1,1) m/s $(\sigma_{ax}, \sigma_{ay})$=(1,1) m/s^2.
5. $T_{MAX} = 1$ second.



(a) Under-Provisioned $(N = 30) > N_{MAX}$

(b) Fully Provisioned $(N = 17) = N_{MAX}$

(c) Over-Provisioned $(N = 14) < N_{MAX}$

(d) Way Over-Provisioned $(N = 9) \ll N_{MAX}$

Figure 4.4: This figure demonstrates various combinations of N/R (for $R = 1$). In each plot, the blue diagonal line is the stability boundary and separates the two regions of operation. When the load curve is below the diagonal, track is maintained on all targets. Above the stability line, the system is unstable. Figure 4.4(a) shows the under-provisioned case where the system load is always above the stability line for $\tau > 0$. In this case, the system is overwhelmed and tracks are lost. Figure 4.4(b) shows the fully provisioned case ($\rho = 100\%$), where the minimal amount of resources are wasted. Figures 4.4(c) and 4.4(d) show the over-provisioned case where the system keeps all targets in track and has spare time for other tasks, as well as a dotted line showing the equivalence point compared to the fully provisioned case.

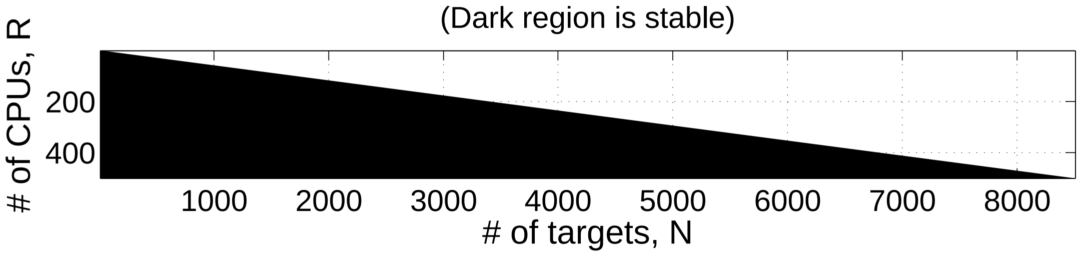


Figure 4.5: The system provisioning matrix specifies stability region (dark) as a function of the numbers of radars and the number targets for track-only radar.

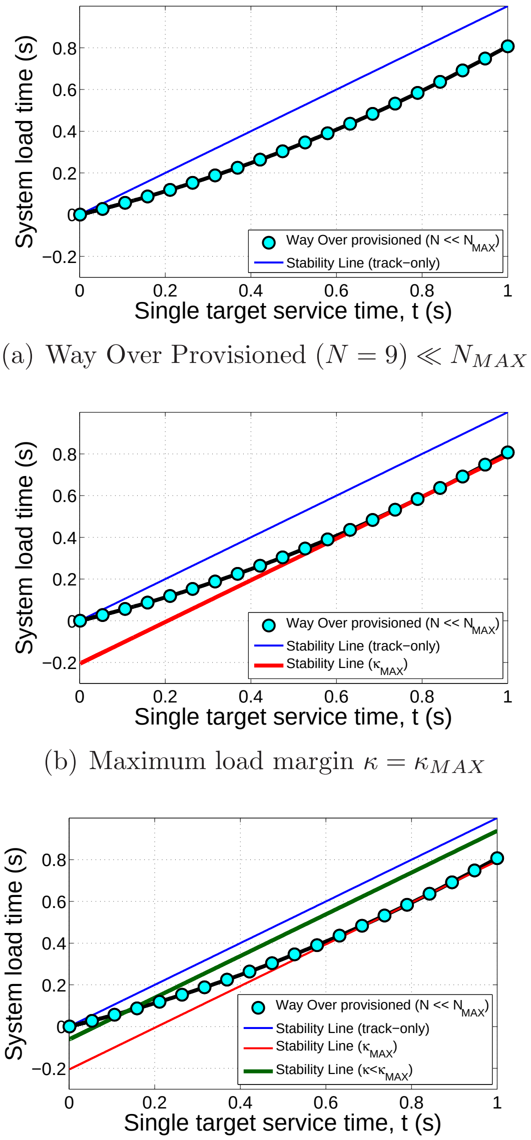
4.6.1 Loading of track-only system

Figure 4.4 shows results for various numbers of R sensors and N targets, such that $N/R = 9, 14, 17, 30$, respectively. In each plot, the blue diagonal line is the stability boundary and separates the two regions of operation. When the load curve is below the diagonal, track is maintained on all targets. Above the stability line, the system is unstable. Figure 4.4(a) shows the under-provisioned case where the system load is always above the stability line for $\tau > 0$. In this case, the system is overwhelmed and tracks are lost. Figure 4.4(b) shows the fully provisioned case ($\rho = 100\%$), where the minimal amount of resources are wasted. Figures 4.4(c) and 4.4(d) show the over-provisioned case where the system keeps all targets in track and has spare time for other tasks, as well as a dotted line showing the equivalence point compared to the fully provisioned case.

Figure 4.5 provides a graphical representation of stability in track-only provisioning as a function of the number of N targets and R sensors. The figure shows a matrix whose (i, j) entry is equal to 1 if i sensors can track j targets stably and equal to 0 otherwise. The dark areas represent the stable operating region.

4.6.2 Multi-purpose system provisioning

Figure 4.6 illustrates a computation of the excess capacity, occupancy, and load margin for the same radar as in the previous section but when it is tracking only 9 targets and can devote resources to other tasks. Unlike the case of 17 targets that only intersects the diagonal line $y(u) = u - \Delta$ when $\Delta = 0$, there is a substantial load margin for the case of 9 targets, $\Delta_{max} = 0.206/N$ secs as shown in Figure 4.6(b). At this full utilization operating point the radar devotes approximately 11% of its time to tracking and the rest of its time to other tasks. The distance between the upper



(a) Way Over Provisioned $(N = 9) \ll N_{MAX}$

(b) Maximum load margin $\kappa = \kappa_{MAX}$

(c) Maximum load margin $\kappa = \kappa_{MAX}$

Figure 4.6: System loading curves for computing occupancy and excess capacity for the multi-purpose radar tracking example. Unlike the case of 17 targets that only intersects the diagonal line $y(u) = u - \Delta$ when $\Delta = 0$, there is a substantial load margin for the case of 9 targets, $\Delta_{max} = 0.206/N$ secs as shown in Figure 4.6(b). At this full utilization operating point the radar devotes approximately 11% of its time to tracking and the rest of its time to other tasks. The distance between the upper and lower diagonal lines $y(u) = u$ and $y(u) = u - \Delta_{max}N$ is 0.206 secs. If the actual load for other tasks was set to only $\Delta = 0.06/N$ secs as in Figure 4.6(c), giving $c_{excess} = 0.70$ and an occupancy of $\rho(\Delta) = 0.76$, the system would be idle 24% of the time.

and lower diagonal lines $y(u) = u$ and $y(u) = u - \Delta_{max} N$ is 0.206 secs. If the actual load for other tasks was set to only $\Delta = 0.06/N$ secs as in Figure 4.6(c), giving c_{excess} = 0.70 and an occupancy of $\rho(\Delta) = 0.76$, the system would be idle 24% of the time.

4.7 Conclusions

This chapter has proposed a conservative approach to sensor resource management for multiple target tracking subject to typical computational resource constraints. The approach requires finding solutions to load balance equations that guarantee system stability. These solutions yield the minimal system requirements for provisioning radars. The solutions guarantee stable tracking with prescribed level of statistical confidence. The provisioning results given here are conservative and specify the system requirements, steady state occupancy, revisit times, and track entropy in terms of the PLQ sensor scheduling policy. The PLQ policy will always perform at least as well as the performance predictions we provide. One can expect considerably better performance of the system than these predictions for typical scenarios, although there exists a scenario (namely, all targets are equally difficult to track) where the predictions are exact. Less stringent provisioning requirements might be explored using a stochastic optimization.

Future work will consider and compare policies other than the PLQ policy. Of particular interest are policies using a random allocation that can be used as a baseline comparison, as well as policies that may have a class-dependent allocation. Additionally, we may consider optimizing a scheduling policy (in terms of τ^*) among multiple alternative policies. Furthermore, in this chapter we have considered tasks in the load margin separately from the target tracking task. However, if we consider multiple epochs, tasks such as Kalman filtering and/or target classification may significantly improve the capability to track targets stably over time. Future work plans to develop a framework for scheduling in the multiple-epoch scenario and analyze the tradeoffs in load margin vs. non-myopic improvements to our performance bounds.

CHAPTER V

Adaptive Target Detection/Tracking with Synthetic Aperture Radar Imagery

5.1 Introduction

In the previous chapters, tools were developed for sensor management and adaptive search for sparse moving targets. This chapter provides an application for these tools, namely target detection and tracking with synthetic aperture radar (SAR) images. The previous chapters relied on explicit characterizations of the uncertainty of target state estimation. In Chapters II and III, one needed the posterior probabilities on target location and posterior estimates of the target amplitudes. In Chapter IV, one needed estimates of the state estimation errors. This chapter proposes inference algorithms and performance prediction (in terms of likelihood ratios and Cramér Rao lower bounds) that can be used explicitly for the algorithms in Chapters II through IV.

The image formation process for SAR images is more complicated than that of standard electro-optical images. Examples of these complexities include:

- SAR images have complex-valued rather than real-valued intensities, and the SAR phase information is of great importance for detection and estimation of target states. [31, 32, 42].

- SAR images are corrupted by spatiotemporally-varying antenna gain/phase patterns that often need to be estimated from homogeneous target-free data [79, 81].
- SAR images have spatially-varying clutter that can mask the target signature unless known a priori or properly estimated [38].
- SAR images contain motion-induced displacement and diffusion of the target response [42, 54].
- SAR images include multiple error sources due to radar collection and physical properties of the reflectors, such as angular scintillation (a.k.a. glints) [16] and speckle [73, 75].

Despite these complications, a great deal of structure exists in SAR images that can be leveraged to provide stronger SAR detection and tracking performance. This includes (a) using the coherence between multiple channels of an along-track radar in order to remove the stationary background (a.k.a, 'clutter'), (b) assuming that pixels within the image can be described by one (or a mixture) of a small number of object classes (e.g., buildings, vegetation, etc.), and (c) considering kinematic models for the target motion such as Markov smoothness priors. From this structure in SAR imagery, one might consider models that assume that the clutter lies in a low-dimensional subspace that can be estimated directly from the data. Indeed, recent work Borcea et al. [15] has shown that SAR signals can be represented as a composition of a low-rank component containing the clutter, a sparse component containing the target signatures, and additive noise.

In general, SAR images are formed by focusing the response of stationary objects to a single spatial location. Moving targets, however, will cause phase errors in the standard formation of SAR images that cause displacement and defocusing effects. Most methods designed to detect the target depend on either (a) exploiting the phase errors induced by the SAR image formation process for a single phase center system or (b) canceling the clutter background using a multiple phase center system. In this chapter, we provide a rich model that can combine (and exploit) both sources of information in order to improve on both methodologies.

Fienup [42] provides an analysis of SAR phase errors induced by translational motions for single-look SAR imagery. He shows that the major concerns are (a) azimuth translation errors from range-velocities, (b) azimuth smearing errors due to accelerations in range, and (c) azimuth smearing due to velocities in azimuth. Fienup also provides an algorithm for detecting targets by their induced phase errors. The algorithm is based on estimating the moving target's phase error, applying a focusing filter, and evaluating the sharpness ratio as a detection statistic. Jao [54] shows that given both the radar trajectory and the target trajectory, it is possible to geometrically determine the location of the target signature in a reconstructed SAR image. Although the radar trajectory is usually known with some accuracy, the target trajectory is unknown. On the other hand, if the target is assumed to have no accelerations, Jao provides an efficient FFT-based method for refocusing a SAR image over a selection of range velocities. Khwaja and Ma [58] provide a algorithm to exploit the sparsity of moving targets within SAR imagery; they propose a basis that is constructed from trajectories formed from all possible combinations of a set of velocities and positions. To combat the computational complexity of searching through this dictionary, the authors use compressed sensing techniques. Instead of searching over a dictionary of velocities, our work proposes to use a prior distribution on the target trajectory that can be provided a priori through road and traffic models or adaptively through observations of the scene over time.

The process of removing the stationary background in order to detect moving targets is also known in the literature as 'change detection' or 'clutter suppresion'. Gierull [43] provides a statistical analysis of the phase and magnitude of complex SAR images for two channels. He shows that SAR images cannot be modeled as spatially-invariant Gaussian in many cases of interest, such as in urban environments, where the statistics vary spatially and may be modulated by random variations. In our work, we model the distributions of the clutter as spatially varying and model the random modulations directly.

Ender [38] applies space-time adaptive processing (STAP) to multiple-channel SAR imagery. Similar to standard change detection algorithms such as displaced phase center array (DPCA) and along-track interferometry (ATI), STAP models the clutter as being embedded in a one-dimensional subspace. However, STAP extends those algorithms to using $N > 2$ channels, where a single channel is used to estimate the stationary background and the remaining $(N - 1)$ channels are used to estimate the moving component. However, STAP relies on estimating the complex-valued covariance matrix of the N-channel system, which in turn depends on the availability of homogeneous target-free secondary data.

There are a multitude of algorithms for change detection that are based on multi-temporal SAR images rather than multi-channel data. Bazi and Bruzzone [13] develop methods for multi-temporal change detection that use adaptive thresholds for declaring changes based on a theoretical analysis of a generalized Gaussian model. Bovolo and Bruzonne [17] provide another algorithm for change detection that employs a wavelet-based multiple scale decomposition of multitemporal SAR images, with an adaptive scale driven fusion algorithm.

Ranney and Soumekh [79,81] develop methods for change detection from SAR images collected at two distinct times that are robust to errors in the SAR imaging process. They address error sources including inaccurate position information, varying antenna gains, and autofocus errors. They propose that the stationary components of multi-temporal SAR images can be related by a spatially-varying 2-dimensional filter. To make the change detection algorithm numerically practical, the authors propose that this filter can be well-approximated by a spatially invariant response within small subregions about any pixel in the image. This thesis adopts this model for the case where there are no registration errors. Under a Gaussian assumption for the measurement errors, it can be shown that the maximum likelihood estimate for the filter coefficients can be computed easily through simple least squares.

Ground Moving Target Indication (GMTI) methods involve the processing of SAR imagery to detect and estimate moving targets. Often clutter cancellation and change detection play a preprocessing role in these algorithms [46, 47, 71, 95]. This chapter aims to combine properties of many of these algorithms into a unifying framework that simultaneously estimates the target signature and the nuisance parameters, such as clutter distributions and antenna calibrations.

It should be noted that many of the previously discussed algorithms work well in certain situations, but do not provide estimates of their uncertainty that may be necessary for adaptive sensing, sensor management, or sensor fusion. This chapter aims to bridge this gap by providing a Bayesian formulation that provides uncertainty distributions for the presence of the moving targets and their positions. Under this Bayesian formulation, we can generate the posterior distribution of the target state(s) given the observations (i.e., the SAR images).

Recently, there has been great interest by Wright et al. [93], Lin et al. [63], Candes et al. [19] and Ding et al. [33] in the so-called robust principal component analysis (RPCA) problem that decomposes high-dimensional signals as

$$\boldsymbol{I} = \boldsymbol{L} + \boldsymbol{S} + \boldsymbol{E}, \tag{5.1}$$

where $\boldsymbol{I} \in \mathbb{R}^{N \times M}$ is an observed high dimensional signal, $\boldsymbol{L} \in \mathbb{R}^{N \times M}$ is a low-rank matrix with rank $r \ll NM$, $\boldsymbol{S} \in \mathbb{R}^{N \times M}$ is a sparse component, and $\boldsymbol{E} \in \mathbb{R}^{N \times M}$ is dense low-amplitude noise. In [19, 63, 93], inference in this model is done by optimizing a cost function of the form

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \gamma \|\boldsymbol{S}\|_1 + (2\mu)^{-1} \|\boldsymbol{I} - \boldsymbol{L} - \boldsymbol{S}\|_F \tag{5.2}$$

where the last term is sometimes replaced by the constraint $\boldsymbol{I} = \boldsymbol{L} + \boldsymbol{S}$. One major drawback of these methods involves finding the algorithm parameters (e.g., tolerance levels or choices of γ, μ), which may depend on the given signal. Moreover, it has been demonstrated that the performance of these algorithms can depend strongly on these parameters.

Bayesian methods by Ding et al. [33] have been proposed that simultaneously learn the noise statistics and infer the low-rank and sparse components. Moreover, they show that their method can be generalized to richer models, e.g. Markov dependencies on the target locations. Additionally, these Bayesian inferences provide a characterization of the uncertainty of the outputs through a Markov Chain Monte Carlo (MCMC) estimate of the posterior distribution.

The work by Ding et al. [33] is based on a general Bayesian framework [85] by Tipping for obtaining *sparse* solutions to regression and classification problems. Tipping's framework uses simple distributions (e.g., those belonging to the exponential class) that can be described by few parameters, known as hyperparameters. Moreover, Tipping considers a *hierarchy* where the hyperparameters themselves are assumed to have a known 'hyperprior' distribution. Often the prior and hyperprior distributions are chosen to be conjugate, so that inference is simple. Tipping provides insight into choosing the hyperparameter distributions so as to be non-informative with respect to the prior. This latter property is important in making it possible to implement inference algorithms with few tuning parameters. Finally, Tipping provides a specialization to the 'relevance vector machine' (RVM), which can be thought of as a Bayesian version of the support vector machine. Wipf et al. [92] provides an interpretation of the RVM as the application of a variational approximation to estimating the true posterior distribution. Wipf et al. explains the sparsity properties of the sparse Bayesian learning algorithms in a rigorous manner. Additionally, it also provides connections with other popular work in sparse problems, such as the FOCUSS and basis pursuit algorithms.

We adopt this hierarchical Bayesian model to SAR images. This requires the following non-trivial extensions: (a) we consider complex-valued data rather than real-valued intensity images; (b) we model correlated noise sources based on physical knowledge of SAR phase history collection and image formation; (c) we relax the assumption of a low-rank background component by assuming that the background component lies in a low-dimensional subspace; and (d), we directly model SAR phenomenology by including terms for glints, speckle contributions, antenna gain patterns, and target kinematics. Moreover, we demonstrate the performance of the proposed algorithm on both simulated and measured datasets, showing competitive or better performance in a variety of situations. Finally, we show that the output of the Bayesian model can be used for performance prediction for future passes of the radar.

The rest of the paper is organized as follows: Notation is given in Section 5.2

and the image model is provided in 5.3. Markov, spatial, and/or target kinematic extensions are discussed in Section 5.4. The inference algorithm is given in Section 5.5. Section 5.6 provides theory for performance prediction using the output of the Bayesian inference. Performance is analyzed over both simulated and measured datasets in Section 5.7. We conclude and point to future work in Section 5.8.

5.2 Notation

Available is a set of SAR images of a region formed from multiple passes of an along-track radar platform with multiple antennas (i.e., phase centers.) Moreover, images are formed over distinct azimuth angle ranges that can be indexed by the frame number, f. Table 5.1 provides the indexing scheme used throughout this chapter in order to distinguish between images from various antennas, frames, and/or passes. Table 5.2 provides a list of indexing conventions used to denote collections of variables.

Table 5.1: Index variable names used in paper

Index Description	Index Variable	Range
Antenna (channel)	k	$1, 2, \ldots, K$
Frame (azimuth range)	f	$1, 2, \ldots, F$
Pass	i	$1, 2, \ldots, N$
Pixel	p	$1, 2, \ldots, P$

We model the quadrature components of the SAR images with the complex-normal distribution, where we use the notation

$$\underline{w} \sim \mathcal{CN}\left(0, \mathbf{\Gamma}\right) \tag{5.3}$$

where $\mathcal{CN}(\mu, \mathbf{\Gamma})$ represents the complex-Normal distribution with mean μ and complex covariance matrix $\mathbf{\Gamma}$, and $\vec{w}$ is random vector of K complex-values (from each

Table 5.2: Our data indexing conventions

Variable	Convention	Description
$i^{(p)}_{k,f,i}$	Standard	Value at pixel p, antenna k, and frame f, pass i
$\underline{i}^{(p)}_{f,i}$	Underline	Values at pixel p, frame f, and pass i over all antennas
$\boldsymbol{i}^{(p)}_{f,1:N}$	Lower-case, Boldface	Values at pixel p and frame f over all antennas and passes
$\boldsymbol{I}_{f,i}$	Upper-case Boldface	Values over all pixels and antennas at frame f and pass i
$\boldsymbol{I}$	Upper-case, Boldface, No Indices	Values over all pixels, antennas, frames, and passes

of K antennas.)

5.3 SAR image model

We propose a decomposition of SAR images at each frame f and pass i as follows

$$\boldsymbol{I}_{f,i} = \boldsymbol{H}_{f,i} \circ (\boldsymbol{L}_{f,i} + \boldsymbol{S}_{f,i} + \boldsymbol{V}_{f,i}), \tag{5.4}$$

where $\boldsymbol{H}_{f,i}$ is a spatiotemporally-varying filter that accounts for antenna calibration errors, $\boldsymbol{L}_{f,i}$ is a low-dimensional representation of the background clutter, $\boldsymbol{S}_{f,i}$ is a sparse component that contains the targets of interest, $\boldsymbol{V}_{f,i}$ is zero-mean additive noise, and $\circ$ denotes the Hadamard (element-wise) product. Each of these components belongs to the space $\mathbb{C}^{P \times K}$. The remainder of this section discusses the model in detail. Figure 5.3 shows a graphical representation of the model.

5.3.1 Low-dimensional component, $\boldsymbol{L}_{f,i}$

We propose a decomposition of the low-rank component as

$$\boldsymbol{L}_{f,i} = \boldsymbol{B}_f + \boldsymbol{X}_{f,i}, \tag{5.5}$$

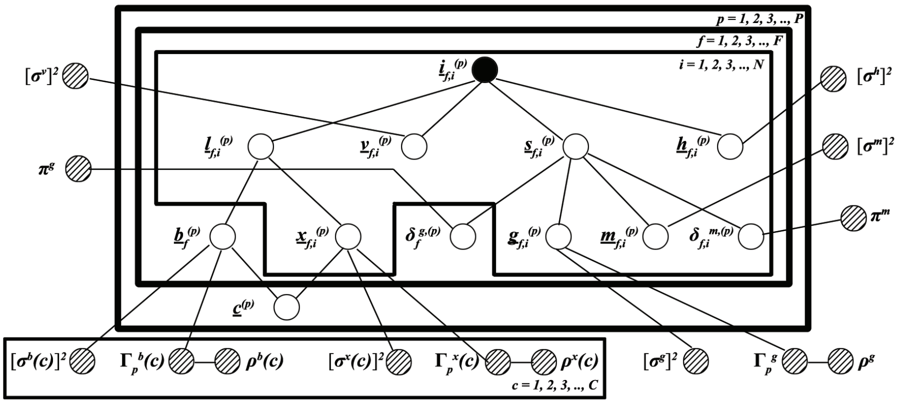


Figure 5.1: This figure provides a graphical representation of the proposed SAR image model. The dark circle represents the observed random variable. The unshaded circles represent the basic parameters of the model, while the dashed circles represent hyperparameters that are also modeled as random variables.

where $\boldsymbol{B}_f$ is the inherent background that is identical over all passes, $\boldsymbol{X}_{f,i}$ is the speckle noise component that arises from coherent imaging in SAR. Posner [73] and Raney [75] describe speckle noise, which tends to be spatially correlated depending on the texture of the surrounding pixels.

The quadrature components of radar channels are often modeled as zero-mean Gaussian processes, though Gierull [43] demonstrates that for heterogeneous clutter (such as in urban scenes), one must consider spatially varying models. To account for this spatial variation, this model assumes that each background pixel can be defined by one of J classes that may be representative of roads, vegetation, or buildings within the scene. Our model is low-dimensional since $J \ll P$, where P is the number of pixels in the measured images. We put a multinomial model on each object class

$$\underline{c}^{(p)} = \left\{ c_j^{(p)} \right\}_{j=1}^{J} \sim \text{Multinomial}(1; q_1, q_2, \ldots, q_J) \tag{5.6}$$

where q_j is the prior probability of the j-th object class. Then the class assignment $C^{(p)}$ is the single location in $\underline{c}$ with value equal to one. We use a hidden Markov model dependency that reflects that neighboring pixels are likely to have the same class. The class $C^{(p)}$ defines the distribution of the pixel p, where we specifically model the background and speckle components respectively as complex-normal distributed:

$$\underline{b}_f^{(p)} \sim \mathcal{CN}\left(0, \mathbf{\Gamma}_B^{C^{(p)}}\right), \quad \underline{x}_{f,i}^{(p)} \sim \mathcal{CN}\left(0, \mathbf{\Gamma}_X^{C^{(p)}}\right) \tag{5.7}$$

Note that the class type specifies the distribution of the pixels and each vector of K values (e.g. background $\underline{b}_f^{(p)}$ or speckle $\underline{x}_{f,i}^{(p)}$) is drawn independently from that distribution.

5.3.2 Sparse component, $\boldsymbol{S}_{f,i}$

The sparse component is modeled as

$$\boldsymbol{S}_{f,i} = \left(\mathbf{\Delta}_f^G \otimes \mathbf{1}_K^T\right) \circ \boldsymbol{G}_{f,i} + \left(\mathbf{\Delta}_{f,i}^M \otimes \mathbf{1}_K^T\right) \circ \boldsymbol{M}_{f,i}, \tag{5.8}$$

where $\boldsymbol{G}_{f,i} \in \mathbb{C}^{P\times K}$ is the specular noise (glints) component with associated indicator variables $\mathbf{\Delta}_f^G \in \{0,1\}^P$, $\boldsymbol{M}_{f,i} \in \mathbb{C}^{P\times K}$ is the (moving) target component with associated indicator variables $\mathbf{\Delta}_{f,i}^M \in \{0,1\}^P$, $\mathbf{1}_K$ is the all ones vector of size $K \times 1$, and $\otimes$ is the Kronecker product. Note that this shared sparsity model assumes that the glint/target components are present in one antenna if and only if they are present in the other antennas. Moreover, glints are known to have a large angular dependence, in the sense that the intensity of the glint dominates in only a few azimuth angles but is present from pass to pass as described by Borden [16]. Thus, the indicators for glints do not depend on the pass index i. Once again, we assume that the glints and target components are zero-mean complex-normal distributed with covariances $\mathbf{\Gamma}_G$ and $\mathbf{\Gamma}_M$, respectively.

The indicator variable $\delta^{z,(p)}$ at pixel p where z is representative of either g or m

is modeled as

$$\delta^{z,(p)} \sim \text{Bernoulli}(\pi^{z,(p)}), \tag{5.9}$$

$$\pi^{z,(p)} \sim \text{Beta}(a_\pi, b_\pi) \tag{5.10}$$

A sparseness prior is obtained by setting $a_\pi/[a_\pi + b_\pi] \ll 1$. Alternatively, we can introduce additional structure in our model by letting a_π and b_π depend on previous frames (temporally) and/or neighboring pixels (spatially). This is particularly useful for detecting multi-pixel targets that move smoothly through a scene. Section 5.4 discusses this modification in greater detail.

5.3.3 Distribution of quadrature components

Many SAR detection algorithms rely on the ability to separate the target from the background clutter by assuming that the clutter lies in a low-dimensional subspace of the data. Consider a random vector of complex variables $\underline{w} \sim \mathcal{CN}(0, \mathbf{\Gamma})$ where w is representative of b, x, g or m. Under the assumptions that (a) the quadrature components of each antenna are zero-mean normal with variance σ^2 and (b) the correlation among components w_m and w_n is given by $\rho e^{-j\phi_{mn}}$, then $\mathbf{\Gamma}$ can be shown to have the form

$$\mathbf{\Gamma} = \sigma^2 \begin{bmatrix} 1 & \rho e^{-j\phi_{12}} & \cdots & \rho e^{-j\phi_{1K}} \\ \rho e^{j\phi_{12}} & 1 & \cdots & \rho e^{-j\phi_{2K}} \\ \vdots & \vdots & \ddots & \vdots \\ \rho e^{j\phi_{1K}} & \rho e^{j\phi_{2K}} & \cdots & 1 \end{bmatrix}, \tag{5.11}$$

where σ^2 is the channel variance, ρ is the coherence between antennas, and $\{\phi_{nm}\}_{n,m}$ are the interferometric phase differences between the antennas[1]. In an idealized

[1] A more general model could account for different channel variance and coherence values, but since we use the calibration constants $\boldsymbol{H}_{f,i}$ to equalize the channels, the effect was seen to be relatively insignificant.

model with a single point target, the interferometric phases ϕ_{mn} can be shown to be proportional to the target radial velocity. In images containing only stationary targets (i.e., the background components), the covariance matrix has a simpler form:

$$\mathbf{\Gamma}_{background} = \sigma^2 \left((1-\rho)\boldsymbol{I}_{K \times K} + \rho \mathbf{1}_K \mathbf{1}_K^T \right) \tag{5.12}$$

where $\boldsymbol{I}_{K \times K}$ is the $K \times K$ identity matrix and $\mathbf{1}_K$ is the all-ones vector of length K.

It should be noted that the covariance matrix in equation (5.11) is related directly to some common methods for change detection in SAR imagery. In particular, consider the two antenna case ($K = 2$). Along-track interferometry (ATI) thresholds the phase ϕ_{12} in order to detect moving targets which have non-zero phases. Moreover, one can easily show that the eigendecomposition of $\mathbf{\Gamma}$ leads to eigenvalues λ and eigenvectors ν:

$$\lambda(\mathbf{\Gamma}) = \left\{ 2\sigma^2(1+\rho), 2\sigma^2(1-\rho) \right\} \tag{5.13}$$

$$\nu(\mathbf{\Gamma}) = \left\{ \begin{bmatrix} 1 \\ e^{-j\phi_{12}} \end{bmatrix}, \begin{bmatrix} 1 \\ -e^{-j\phi_{12}} \end{bmatrix} \right\}. \tag{5.14}$$

Displaced phase center array (DPCA) processing thresholds the difference between the two channels. Indeed, for small phases, the second eigenvector of $\mathbf{\Gamma}$ reduces to $[1;\ -1]^T$. Thus DPCA can be interpreted as a projection onto the eigenvector of $\mathbf{\Gamma}$. Deming [31] shows that ATI performs well when canceling bright clutter (i.e., high σ^2 and $\rho \approx 1$), while DPCA performs well for canceling dim clutter (i.e., small σ^2 and $\rho \approx 0$.) In our work, we combine the discriminating power of both DPCA and ATI by modeling the covariance matrices directly. Ender [38] provides space-time adaptive processing (STAP), where optimal detection schemes for moving targets are based on the estimation of $\mathbf{\Gamma}$. However, the performance of STAP depends on the availability of target-free homogeneously distributed measurements in order to estimate $\mathbf{\Gamma}$ effectively. In this chapter, we simultaneously estimate the covariance matrices as well as the target contributions. Thus, we demonstrate the capability to detect targets even in the presence of heterogeneous measurements.

In this thesis, the covariance matrix $\mathbf{\Gamma}$ is modeled as a random variable using a modified version of the Multivariate-Normal-Inverse-Wishart conjugate distributions. In particular, we let

$$\begin{aligned}
\underline{w} &\sim \mathcal{CN}\left(\mathbf{0}, \sigma^2 \mathbf{\Gamma}_\rho\right) && (5.15)\\
\mathbf{\Gamma}_\rho &\sim \text{InvWishart}\left(a_\Gamma((1-\rho)\boldsymbol{I}_{K\times K} + \rho \mathbf{1}_K \mathbf{1}_K^T), \nu_\Gamma\right) && (5.16)\\
\sigma^2 &\sim \text{InvGamma}(a_\sigma, b_\sigma) && (5.17)\\
\rho &\sim \text{Beta}(a_\rho, b_\rho) && (5.18)
\end{aligned}$$

where $a_\sigma = b_\sigma = 10^{-6}$ as suggested by Tipping [85] to promote non-informative priors, (a_ρ, b_ρ) are chosen so that $\rho \approx 1$ to ensure a high coherence among the background components, ν_Γ is a parameter that controls how strongly to weight the prior covariance matrix, and a_Γ is chosen so that $E[\mathbf{\Gamma}_\rho] = (1-\rho)\boldsymbol{I}_{K\times K} + \rho \mathbf{1}_K \mathbf{1}_K^T$. In this thesis, we choose ν_Γ to be large to reflect our belief that $\sigma^2 \mathbf{\Gamma}_\rho$ should be close to equation (5.11). Note that this model separates the learning of the channel variance σ^2, which we have no a priori knowledge about, from the learning of the correlation structure $\mathbf{\Gamma}_\rho$.

5.3.4 Calibration filter, $\boldsymbol{H}_{f,i}$

The calibration constants are assumed to be constant within small spatial regions $p \in Z_g$, though they may vary as a function of antenna, frame, or pass. In particular, we let

$$h_{k,f,i}^{(p)} = z_{k,f,i}(g), \forall p \in Z_g, \tag{5.19}$$

$$z_{k,f,i}(g) \sim \mathcal{CN}(1, (\sigma^H)^2) \tag{5.20}$$

where we note that if $(\sigma^H)^2$ is large, then maximum likelihood inference in this case yields the least-squares solution.

5.3.5 Summary of SAR Image Model

Tables 5.3 provides a summary of the distributions for the proposed decomposition of SAR images. The table also provides a characterization of spatial (across pixels) and temporal (across frames and passes) dependencies. For example, background and speckle components have distributions characterized by their class j. Thus, all pixels with class j belong to a subset $Q_j \subset \{1, 2, \ldots, P\}$. In contrast, the distribution of moving targets is assumed to be identical across all pixels, yet the distribution of their indicators varies for each pixel, frame, and pass.

Tables 5.5 and 5.6 provide a summary of the parameters of the distributions in Table 5.3. We provide the simple model for target and glint indicator probabilities that just assumes that they are sparse in the image. We can introduce additional richness in the model by allowing the parameters a_π and b_π to vary over pixels, frames, and passes as described in Section 5.4.1. Table 5.4 shows a matrix of properties of each individual component to explain how they can be separately identified from each other.

5.3.6 Discussion of SAR Image Model

This dissertation provides a unifying framework for SAR imagery that incorporates significant amounts of expert knowledge regarding the physics, kinematics, and

Table 5.3: Distributional models for each component in equations (5.4), (5.5), and (5.8). Spatial column refers to region where pixels share distribution. Temporal column refers to pixels which share values across either frame, pass, or both.

Component	Variable	Distribution	Parameters	Spatial	Temporal
Background	$\underline{b}_f^{(p)}$	$\mathcal{CN}\left(0, \mathbf{\Gamma}^B(j)\right)$	$\mathbf{\Gamma}^B(j) = [\sigma^B(j)]^2\mathbf{\Gamma}^B_\rho(j)$	$p \in Q_j$	Each f, All i
Speckle	$\underline{x}_{f,i}^{(p)}$	$\mathcal{CN}\left(0, \mathbf{\Gamma}^X(j)\right)$	$\mathbf{\Gamma}^X(j) = [\sigma^X(j)]^2\mathbf{\Gamma}^X_\rho(j)$	$p \in Q_j$	Each f, Each i
Glints	$\underline{g}_{f,i}^{(p)}$	$\mathcal{CN}\left(0, \mathbf{\Gamma}^G\right)$	$\mathbf{\Gamma}^G = [\sigma^G]^2\mathbf{\Gamma}^G_\rho$	All p	Each f, Each i
Moving targets	$\underline{m}_{f,i}^{(p)}$	$\mathcal{CN}\left(0, \mathbf{\Gamma}^M\right)$	$\mathbf{\Gamma}^M = [\sigma^M]^2\boldsymbol{I}_{K\times K}$	All p	Each f, Each i
Additive noise	$V_{k,f,i}(p)$	$\mathcal{CN}\left(0, \mathbf{\Gamma}^V\right)$	$\mathbf{\Gamma}^V = [\sigma^V]^2\boldsymbol{I}_{K\times K}$	All p	All f, All i
Glint indicator	$\delta_f^{G,(p)}$	Bernoulli $\left(\pi_f^{G,(p)}\right)$	$\pi_f^{G,(p)}$	Each p	Each f, All i
Target indicator	$\delta_{f,i}^{M,(p)}$	Bernoulli $\left(\pi_{f,i}^{M,(p)}\right)$	$\pi_{f,i}^{M,(p)}$	Each p	Each f, Each i
Class assignment	$\underline{c}(p)$	Multinomial$(1; \underline{q})$	$\underline{q}$	Each p	All f, All i
Calibration coefficients	$H_{k,f,i}(p) = z_{k,f,i}(g)$	$\mathcal{CN}\left(1, [\sigma^H]^2\right)$	$[\sigma^H]^2$	$p \in Z_g$	Each f, Each i

Table 5.4: Identifiability for components of model in equations (5.4), (5.5), and (5.8).

Component	Identical over passes	Identical in spatial regions	Strong correlations*	Described by a few distributions	Amplitude	Sparse (angle)	Sparse (pass)	Markov
Background	Yes	No	Yes	Yes	Varying	No	No	No
Speckle	No	No	Yes	Yes	Varying	No	No	No
Glints	No	No	Yes	No	High	Yes	No	No
Moving Targets	No	No	No	No**	Medium	Yes	Yes	Yes
Calibration coeff.	No	Yes	No	No	Medium	No	No	No
Additive noise	No	No	No	No	Low	No	No	No

* Strong dependencies when the background/speckle/glints have high amplitudes.

*∗ Model can be extended to include generalizations such as multiple target classes and/or template matching.

Table 5.5: Distributional models for covariance parameters of distributions in Table 5.3

Component	Variable	Distribution	Parameters	Suggested Value	Region
Background covariance	$\mathbf{\Gamma}^B(j) = [\sigma^B(j)]^2 \mathbf{\Gamma}^B_\rho(j)$				Each Q_j
Variance	$\left[\sigma^B(j)\right]^2$	Inv-Gamma	$a_\sigma,\ b_\sigma$	$10^{-6}, 10^{-6}$	
Correlation matrix	$\mathbf{\Gamma}^B_\rho(j)$	Inv-Wishart	a_Γ, ν_Γ	See note^b, $O(P)$	
Coherence	$\rho^B(j)$	Beta	$a_\rho,\ b_\rho$	$a_\rho/(a_\rho + b_\rho) \approx 1$	
Speckle covariance	$\mathbf{\Gamma}^X(j) = [\sigma^X(j)]^2 \mathbf{\Gamma}^X_\rho(j)$				Each Q_j
Variance	$\left[\sigma^X(j)\right]^2$	Inv-Gamma	$a_\sigma,\ b_\sigma$	$10^{-6}, 10^{-6}$	
Correlation matrix	$\mathbf{\Gamma}^X_\rho(j)$	Inv-Wishart	a_Γ, ν_Γ	See note^b, $O(P)$	
Coherence	$\rho^X(j)$	Beta	$a_\rho,\ b_\rho$	$a_\rho/(a_\rho + b_\rho) \approx 1$	
Glint covariance	$\mathbf{\Gamma}^G = [\sigma^G]^2 \mathbf{\Gamma}^G_\rho$				All p
Variance	$\left[\sigma^G\right]^2$	Inv-Gamma	$a_\sigma,\ b_\sigma$	$10^{-6}, 10^{-6}$	
Correlation matrix	$\mathbf{\Gamma}^G_\rho$	Inv-Wishart	a_Γ, ν_Γ	See note^b, $O(P)$	
Coherence	ρ^G	Beta	$a_\rho,\ b_\rho$	$a_\rho/(a_\rho + b_\rho) \approx 1$	
Target covariance	$\mathbf{\Gamma}^M = [\sigma^M]^2 \boldsymbol{I}_{K\times K}$				All p
Variance	$\left[\sigma^M\right]^2$	Inv-Gamma	$a_\sigma,\ b_\sigma$	$10^{-6}, 10^{-6}$	
Additive noise covariance	$\mathbf{\Gamma}^V = [\sigma^V]^2 \boldsymbol{I}_{K\times K}$				All p
Variance	$\left[\sigma^V\right]^2$	Inv-Gamma	$a_\sigma,\ b_\sigma$	$10^{-6}, 10^{-6}$	

Table 5.6: Distributional models for other parameters of distributions in Table 5.3

Component	Variable	Distribution	Parameters	Suggested Value	Region
Variance	$\left[\sigma^V\right]^2$	Inv-Gamma	$a_\sigma,\ b_\sigma$	$10^{-6}, 10^{-6}$	
Target indicator probability	$\pi^M_{f,i}(p)$	Beta	a_π, b_π	$a_\pi/(a_\pi + b_\pi) \ll 1$	Each p, f, i
Glint indicator probability	$\pi^Y_f(p)$	Beta	a_π, b_π	$a_\pi/(a_\pi + b_\pi) \ll 1$	Each p, f
Region type probabilities	$\underline{q} = \{q_1, \ldots, q_J\}$	Dirichlet	$\{e_j\}_{j=1}^J$	$e_j = 1/J$	All p
Calibration coefficient variance	$(\sigma^H)^2$	Inv-Gamma	$a_\sigma,\ b_\sigma$	$10^{-6}, 10^{-6}$	All p

statistics behind SAR imagery. However, it is important to note that the framework could be extended to increase model fidelity or to generalize to other interesting cases. In particular, this model primarily uses random variables that are distributed as either multivariate Gaussian or Bernoulli. In contrast, we could use other distributional models that are more robust to mismatches between the model and the observed data. For example, Chen et al. [27] shows that covariance estimation can be done with increased robustness to heavy-tailed distributions as compared to standard estimators that use Gaussian assumptions.

Another important generalization that may considered concerns the choice of the prior parameters and distributions as described by the tables in the previous sections. For example, the inclusion of a glint variable makes sense if the scene contains many man-made objects, but doesn't really impact SAR images of natural scenes. In order to deal with these considerations, one might consider another level of the hierarchical model that contains contextual information. It should be noted that this sort of knowledge has significantly lower dimension than the SAR images themselves, and thus might not require a large computational burden.

Both of the discussed generalizations have the property that they may create higher fidelity with the observed data. However, both of the generalizations introduce a tradeoff between this improved performance and the required computational burden. This is most apparent when using non-Gaussian distributions that do not admit simple posterior distributions (that can be sampled efficiently). Future work may look at the relative gains of using more general models as compared to their computational burden.

This framework also has the property that it encompasses many other algorithms as special cases. For example, template matching for target signatures could easily be accomplished by using templates as priors on the target indicator probabilities, but without considering the random variables for the background, speckle, and glint variables. On the other hand, we could consider a model that includes estimation of the nuisance parameters (background, glints, etc.) as well as using template matching. This is one of the key contributions of this model, in the sense that the proposed framework can combine expert knowledge in a simple way in order to gain discriminating power.

5.4 Markov/spatial/kinematic models for the sparse component

5.4.1 Indicator probability models

This model contains multiple indicator variables with prior probabilities distributed as Beta(a_π, b_π). Moreover, sparsity is obtained when $a_\pi/[a_\pi + b_\pi] \ll 1$. Alternatively, we can introduce additional structure in our model by letting a_π and b_π depend on previous frames (temporally) and/or neighboring pixels (spatially). This is especially useful for detecting multi-pixel targets that move smoothly through a scene.

Define $W^M(p, \mathbf{\Delta}^M_{f,i})$ to be a function that maps the indicator variables $\mathbf{\Delta}^M_{f,i}$ to a real number. For example, this may be the average number of non-zero indicators in the neighborhood of pixel p, or a weighted version that puts higher value on neighboring pixels. For $f = 1$, we let

$$\begin{bmatrix} a^M_{1,i}(p) \\ b^M_{1,i}(p) \end{bmatrix} = \begin{cases} [a_H \quad b_H]^T, & W^M(p, \mathbf{\Delta}^M_{1,i}) > \varepsilon^M_{spatial}, \\ [a_L \quad b_L]^T, & \text{else}, \end{cases} \tag{5.21}$$

and for $f > 1$

$$\begin{bmatrix} a^M_{f,i}(p) \\ b^M_{f,i}(p) \end{bmatrix} = \begin{cases} [a_H \quad b_H]^T, & W^M(p, \mathbf{\Delta}^M_{f,i}) > \varepsilon^M_{spatial} \text{ and} \\ & W^M(p, \mathbf{\Delta}^M_{f-1,i}) > \varepsilon^M_{temporal}, \\ [a_L \quad b_L]^T, & \text{else.} \end{cases} \tag{5.22}$$

In this chapter, we choose (a_L, b_L, a_H, b_H) so that $a_L/(a_L+b+L) \ll 1$ and $a_H/(a_H+b+H) \gg 0$. A similar model can be introduced for the probabilities of the glints.

5.4.2 Target kinematic model

In some applications, such as target tracking or sequential detection, we may have access to an estimate of the kinematic state of the target(s) of interest, such as position, velocity and acceleration. This may be useful for predicting the location of the target at sequential frames. For simplicity, consider a single target at time τ whose state $\xi(\tau) = (\boldsymbol{r}(\tau), \dot{\boldsymbol{r}}(\tau))$ is known with standard errors $\mathbf{\Sigma}_\xi(\tau)$. Note that the uncertainty model for $(\boldsymbol{r}, \dot{\boldsymbol{r}})$ may be (a) known a prior from road maps or traffic behavior patterns, or (b) learned adaptively using some signal processing algorithm such as the Kalman or particle filters.

In standard SAR image formation, moving targets tend to appear displaced and defocused as described by Fienup [42] and Jao [54]. Moreover, Jao showed that given the radar trajectory $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and the target trajectory $(\boldsymbol{r}, \dot{\boldsymbol{r}})$, one can predict the location of the target signature within the image $\boldsymbol{p}$ by solving a system of equations that equate Doppler shifts and ranges, respectively, at each pulse:

$$\frac{d}{d\tau}\left[\|\boldsymbol{p} - \boldsymbol{q}(\tau)\|_2 - \|\boldsymbol{r}(\tau) - \boldsymbol{q}(\tau)\|_2\right]_{\boldsymbol{p}=\boldsymbol{p}^*} = 0 \tag{5.23}$$

$$\|\boldsymbol{p}^* - \boldsymbol{q}(\tau)\|_2 = \|\boldsymbol{r}(\tau) - \boldsymbol{q}(\tau)\|_2\,, \tag{5.24}$$

```
procedure $\{\boldsymbol{\Theta}\}_{i=1:N_{samples}} = \text{SARGibbs}(\boldsymbol{\Theta}_0, \boldsymbol{I})$
  $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta}_0$
  for $iteration = 1$ to $N_{burnin} + N_{samples}$ do
      Sample $\sim f\left(\boldsymbol{B}, \boldsymbol{X}, \boldsymbol{G}, \boldsymbol{M}, \boldsymbol{\Delta}^G, \boldsymbol{\Delta}^M | \boldsymbol{I}, -\right)$          //Base
      Sample $\sim f\left(\boldsymbol{H} | \boldsymbol{I}, -\right)$          //Calibration filter
      Sample $\sim f\left(\boldsymbol{C} | \boldsymbol{I}, -\right)$          //Class assignment
      Sample $\sim f\left(\boldsymbol{\eta} | \boldsymbol{I}, -\right)$          //Hyper-parameters
      $\boldsymbol{\Theta}_{iteration-N_{burnin}} \leftarrow \boldsymbol{\Theta}$ if $iteration > N_{burnin}$
  end for
end procedure
```

Figure 5.2: Gibbs Sampling Pseudocode

which can be reduced to the simpler system of equations:

$$\dot{\boldsymbol{q}}(\tau) \cdot [\boldsymbol{p}^* - \boldsymbol{q}(\tau)] = [\dot{\boldsymbol{r}}(\tau) - \dot{\boldsymbol{q}}(\tau)] \cdot [\boldsymbol{r}(\tau) - \boldsymbol{q}(\tau)] \tag{5.25}$$

$$\|\boldsymbol{p}^* - \boldsymbol{q}(\tau)\|_2 = \|\boldsymbol{r}(\tau) - \boldsymbol{q}(\tau)\|_2 \tag{5.26}$$

The probable locations of the target can be predicted by one of several methods, including:

- Monte Carlo estimation of the target posterior density.
- Gaussian approximation using linearization or the unscented transformation to approximate the posterior density
- Analytical approximation.

Given an estimate of the posterior density, we can modify the function W^M described in the previous section to include dependence on this kinematic information. Details of the posterior density estimation are provided in appendix 5.9.

5.5 Inference

In the proposed hierarchical model, the distribution of hyper-parameters at the base layer are generally chosen to be conjugate to the distributions at the next layer.

This allows for efficient approximation methods for the posterior distribution in the sense that we can sample exactly from these distributions. In particular, we use a Markov Chain Monte Carlo (MCMC) algorithm in the form of a Gibbs sampler to iteratively estimate the full joint posterior. In MCMC, this distribution is approximated by drawing samples iteratively from the conditional distribution of each (random) model variable given the most recent estimate of the rest of the variables (which we denote by $-$). Let $\boldsymbol{\Theta} = \left\{\boldsymbol{B}, \boldsymbol{X}, \boldsymbol{G}, \boldsymbol{M}, \boldsymbol{\Delta}^G, \boldsymbol{\Delta}^M, \boldsymbol{H}, \boldsymbol{C}, \boldsymbol{\eta}\right\}$ represent a current estimate of all of the model variables where $\boldsymbol{\eta}$ represents the set of all hyperparameters. Given measurements $\boldsymbol{I}$, the inference algorithm is given in Figure 5.2. Note that MCMC algorithms require a burn-in period after the Markov chain has become stable, where the duration of burn-in period depends on the problem. After this point, we collect $N_{samples}$ samples that represent the full joint distribution. Full details of the sampling procedures are given in in appendix 5.10. However, we point out a couple of important features here. First, the sampling of the base model can be rewritten as

$$\begin{aligned} & f(\boldsymbol{B}, \boldsymbol{X}, \boldsymbol{G}, \boldsymbol{M}, \boldsymbol{\Delta}^G, \boldsymbol{\Delta}^M | \boldsymbol{I}, -) \\ &= \prod_{p,f} f(\underline{b}_f^{(p)}, \boldsymbol{x}_{f,1:N}^{(p)}, \boldsymbol{g}_{f,1:N}^{(p)}, \boldsymbol{m}_{f,1:N}^{(p)}, \delta_f^{G,(p)}, \boldsymbol{\delta}_{f,1:N}^{M,(p)} | \boldsymbol{I}, -) \end{aligned} \tag{5.27}$$

The conditional independence among pixels and frames given the nuisance parameters allows us to easily parallelize the sampling procedure over the largest dimensions of the state. Moreover, we can extend the parallelization to sampling independently over passes by separating the sampling of equation (5.27) into two Gibbs steps from the densities:

$$
\begin{aligned}
&f(\underline{b}_f^{(p)}, \boldsymbol{x}_{f,1:N}^{(p)}, \boldsymbol{m}_{f,1:N}^{(p)}, \boldsymbol{\delta}_{f,1:N}^{M,(p)} | \boldsymbol{I}, -) \\
&\quad = f(\underline{b}_f^{(p)} | \boldsymbol{I}, \boldsymbol{x}_{f,1:N}^{(p)}, \boldsymbol{m}_{f,1:N}^{(p)}, \boldsymbol{\delta}_{f,1:N}^{M,(p)}, -) \\
&\quad \cdot \prod_i f(\underline{x}_{f,i}^{(p)}, \underline{m}_{f,i}^{(p)} | \delta_{f,i}^{M,(p)}, \boldsymbol{I}, -) f(\delta_{f,i}^{M,(p)} | \boldsymbol{I}, -)
\end{aligned} \tag{5.28}
$$

$$
\begin{aligned}
&f(\boldsymbol{g}_{f,1:N}^{(p)}, \delta_f^{G,(p)} | \boldsymbol{I}, -) \\
&\quad = f(\delta_f^{G,(p)} | \boldsymbol{I}, -) \prod_i f(\underline{g}_{f,i}^{(p)} | \boldsymbol{I}, \delta_f^{G,(p)}, -)
\end{aligned} \tag{5.29}
$$

In both of the sampling steps in equations (5.28) and (5.29), we have an exact inference algorithm over multivariate-Gaussian distributed variables and Bernoulli distributed variables. This leads to faster convergence of the Markov chain and subsequently fewer burn-in samples. The conditional density for the nuisance parameters $\boldsymbol{\eta}$ given the remainder variables can also be re-written to allow for efficient sampling. In particular, due to conditional independence we have:

$$
\begin{aligned}
f(\boldsymbol{\eta} | \boldsymbol{I}, -) =& f(\boldsymbol{\Gamma}^M | \boldsymbol{M}, \boldsymbol{\Delta}^M) f(\boldsymbol{\pi}^M | \boldsymbol{M}, \boldsymbol{\Delta}^M, \boldsymbol{\Gamma}^M) \\
&\cdot f(\boldsymbol{\Gamma}^G | \boldsymbol{G}, \boldsymbol{\Delta}^G) f(\boldsymbol{\pi}^G | \boldsymbol{G}, \boldsymbol{\Delta}^G, \boldsymbol{\Gamma}^G) \\
&\cdot \prod_j f(\boldsymbol{\Gamma}^B(j) | \boldsymbol{B}, \boldsymbol{C}) f(\boldsymbol{\Gamma}^X(j) | \boldsymbol{X}, \boldsymbol{C})
\end{aligned} \tag{5.30}
$$

where $\boldsymbol{\Gamma}$ represents the parameters related to the covariance matrices (i.e., the variance σ^2, correlation structure $\boldsymbol{\Gamma}_\rho$, and the coherence ρ). Once again, this decomposition allows for a sampling procedure that leads to faster convergence of the Gibbs sampler. Moreover, the sampling procedures for the individual densities in equation (5.30) tend to require sufficient statistics that are of significantly smaller dimension and thus more desirable from a computational viewpoint. For example, sampling of the covariance matrix $\boldsymbol{\Gamma}^M$ depends only on a $K \times K$ sample covariance matrix. It should be noted that sampling of the covariance matrices requires additional effort in order to constrain its shape to that of equation (5.11). In particular, we use a Metropolis-Hastings step, which can be easily done by noting that the posterior density $f(\mathbf{\Gamma}^W, \rho^W, (\sigma^2)^W|\boldsymbol{W})$ is proportional to an Inverse-Wishart distribution. Details are provided in the in appendix 5.10.

5.6 Performance prediction

Among the benefits of using our SAR image model is the ability to use the inferred statistics in order to do performance prediction that can be used for planning purposes and feasibility analysis. In this chapter, we provide a likelihood ratio test (LRT) for detection and derive a Cramér-Rao Lower Bound (CRLB) for estimator mean squared error (MSE).

5.6.1 Detection

The LRT for detecting a target in frame f and pass i follows from the detection problem:

$$\begin{array}{ll} H0: & \underline{i}_{f,i}^{(p)} = \underline{h}_{f,i}^{(p)} \circ \left(\underline{c}_{f,i}^{(p)} + \underline{v}_{f,i}^{(p)}\right) \\ H1: & \underline{i}_{f,i}^{(p)} = \underline{h}_{f,i}^{(p)} \circ \left(\underline{c}_{f,i}^{(p)} + \underline{m}_{f,i}^{(p)} + \underline{v}_{f,i}^{(p)}\right) \end{array} \quad p \subset \{1, \ldots, P\}$$

where we assume $p \in \mathcal{X} \subset \{1, \ldots, P\}$ is a set of pixels containing the target signature and the background clutter is given by

$$\underline{c}_{f,i}^{(p)} = \underline{b}_{f}^{(p)} + \underline{x}_{f,i}^{(p)} + \delta_f^G \underline{g}_{f,i}^{(p)} \sim \mathcal{CN}\left(0, \mathbf{\Gamma}^C(p)\right), \tag{5.31}$$

where $\mathbf{\Gamma}^C(p) \overset{\triangle}{=} \mathbf{\Gamma}^B(p) + \mathbf{\Gamma}^X(p) + \delta_f^G \mathbf{\Gamma}^G(p)$. Assuming that the pixels are independently distributed, the LRT is given by:

$$\Lambda_{LRT} = \prod_{p \in \mathcal{X}} \frac{\psi_{\mathcal{CN}}(\underline{i}_{f,i}^{(p)}; 0, \mathbf{\Gamma}^1(p))}{\psi_{\mathcal{CN}}(\underline{i}_{f,i}^{(p)}; 0, \mathbf{\Gamma}^0(p))} \tag{5.32}$$

$$\mathbf{\Gamma}^0(p) = \mathbf{\Gamma}^M + \mathbf{\Gamma}^C(p) + \mathbf{\Gamma}^V, \quad \mathbf{\Gamma}^1(p) = \mathbf{\Gamma}^0(p) + \mathbf{\Gamma}^M \tag{5.33}$$

Table 5.7: Parameters of simulated dataset

Parameter	**Value**
Pixels in image, P	$P = 100 \times 100$
Number of frames per pass, F	$F = 1$
# of antennas, K	$K = 3$
# of passes, N	$N \in \{5, 10, 20\}$
# of target pixels/image, $N_{targets}$	$N_{targets} = 20$
Clutter of background, ρ	$\rho \in \{0.9, 0.99, 0.999, 0.9999\}$
Variance of targets, σ^2_{target}	$\sigma^2_{target} = 1$
Variance of background	Either $\sigma^2_{dim} = \sigma^2_{clutter}/100$ or $\sigma^2_{bright} = \sigma^2_{clutter}$
Signal-to-noise-plus clutter (SCNR)	$\text{SCNR} \overset{\triangle}{=} \frac{\sigma^2_{target}}{\sigma^2_{clutter}+\sigma^2_{noise}}$ $\in \{0.1, 0.5, 1, 2\}$

where $\psi_{\mathcal{CN}}(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Gamma})$ is the complex-normal distribution pdf of $\boldsymbol{y}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Gamma}$. Moreover, it can be easily shown that Λ_{LRT} is equivalent to the test statistic

$$T = \sum_{p \in \mathcal{X}} \left(\underline{i}^{(p)}_{f,i} \right)^H \left[(\boldsymbol{\Gamma}^1(p))^{-1} - (\boldsymbol{\Gamma}^0(p))^{-1} \right] \left(\underline{i}^{(p)}_{f,i} \right). \tag{5.34}$$

5.6.2 The CRLB

The derivation of the CRLB is provided in Appendix 5.11.

5.7 Performance analysis

5.7.1 Simulation

We first demonstrate the performance of the proposed algorithm, which we refer to as the Bayes SAR algorithm, on a simulated dataset. Images were created according to the model given in Section 5.3 with parameters given in Table 5.7. The low-dimensional component was divided into one of two classes (‘dim’ or ‘bright’). Pixels were deterministically assigned to one of these classes to resemble a natural

Table 5.8: Comparison of proposed method (Bayes SAR) to RPCA Methods with $N = 20$, $F = 1$, $K = 3$. Note that the Bayes SAR method performs about twice as well as either of the RPCA methods for all criteria. In particular, the Bayes SAR method produces a sparse result (last column), whereas the RPCA methods do not.

(a) Bayes SAR

SCNR	Coherence	$\frac{\left\|\boldsymbol{L}-\hat{\boldsymbol{L}}\right\|_2}{\|\boldsymbol{L}\|_2}$	$\frac{\left\|\boldsymbol{S}-\hat{\boldsymbol{S}}\right\|_2}{\|\boldsymbol{S}\|_2}$	$\frac{\left\|\boldsymbol{S}-\hat{\boldsymbol{S}}\right\|_0}{\|\boldsymbol{S}\|_0}$
10%	0.900	0.057	0.578	0.550
10%	0.9999	0.045	0.419	0.367
100%	0.900	0.057	0.155	0.150
100%	0.9999	0.052	0.122	0.096
200%	0.900	0.057	0.123	0.137
200%	0.9999	0.056	0.114	0.092

(b) Opt. RPCA

SCNR	Coherence	$\frac{\left\|\boldsymbol{L}-\hat{\boldsymbol{L}}\right\|_2}{\|\boldsymbol{L}\|_2}$	$\frac{\left\|\boldsymbol{S}-\hat{\boldsymbol{S}}\right\|_2}{\|\boldsymbol{S}\|_2}$	$\frac{\left\|\boldsymbol{S}-\hat{\boldsymbol{S}}\right\|_0}{\|\boldsymbol{S}\|_0}$
10%	0.900	0.111	3.175	111.026
10%	0.9999	0.113	3.237	109.716
100%	0.900	0.111	1.189	109.520
100%	0.9999	0.110	1.173	108.203
200%	0.900	0.112	1.058	111.120
200%	0.9999	0.110	1.035	109.583

(c) Bayes RPCA

SCNR	Coherence	$\frac{\left\|\boldsymbol{L}-\hat{\boldsymbol{L}}\right\|_2}{\|\boldsymbol{L}\|_2}$	$\frac{\left\|\boldsymbol{S}-\hat{\boldsymbol{S}}\right\|_2}{\|\boldsymbol{S}\|_2}$	$\frac{\left\|\boldsymbol{S}-\hat{\boldsymbol{S}}\right\|_0}{\|\boldsymbol{S}\|_0}$
10%	0.900	0.117	0.998	3.761
10%	0.9999	0.108	0.990	3.799
100%	0.900	0.116	0.764	3.451
100%	0.9999	0.117	0.741	3.494
200%	0.900	0.125	0.706	3.665
200%	0.9999	0.129	0.692	3.720

SAR image (see Figure 5.4). The sparse component included a randomly placed target with multiple-pixel extent. A spatiotemporally varying antenna gain filter was uniformly drawn at random on the range $[0, 2\pi)$ for groups of pixels of size 25×25. Lastly, zero-mean IID noise was added with variance σ^2_{noise}.

The Bayes SAR model is applied to infer the low-dimensional component $\boldsymbol{L}_{f,i}$ and sparse target component $\boldsymbol{S}_{f,i}$ with estimates denoted $\hat{\boldsymbol{L}}_{f,i}$ and $\hat{\boldsymbol{S}}_{f,i}$, respectively. Hyperparameters of the model are chosen according to the Section 5.5. Results are given by the mean of MCMC inference with 500 burn-in iterations followed by 100 collection samples. We consider three metrics to evaluate the reconstruction errors: $\frac{\|\boldsymbol{L}-\hat{\boldsymbol{L}}\|_2}{\|\boldsymbol{L}\|_2}$, $\frac{\|\boldsymbol{S}-\hat{\boldsymbol{S}}\|_2}{\|\boldsymbol{S}\|_2}$, $\frac{\|\boldsymbol{S}-\hat{\boldsymbol{S}}\|_2}{\|\boldsymbol{S}\|_0}$, where the norm is taken over the vectorized quantities.

In comparison to the Bayes SAR model, results are given for state-of-the-art algorithms for Robust Principal Component Analysis (RCPA): an optimization-based approach proposed by Wright et al. [93] and Candes et al. [19] and a Bayesian-based approach proposed by Ding et al. [33][2]. The optimization-based approach requires a tolerance parameter which is related to the noise level, as suggested by Ding et al. [33]. We chose this parameter in order to have the smallest reconstruction errors. The Bayesian method did not require tuning parameters, except for choosing the maximum rank of $\boldsymbol{L}_{f,i}$ which was set to 20.

Figure 5.3 compares the relative reconstruction error of the sparse (target) component, $\frac{\|\boldsymbol{S}-\hat{\boldsymbol{S}}\|_2}{\|\boldsymbol{S}\|_2}$, across all algorithms, number of passes N, coherence of antennas ρ, and SCNR. In all cases, the Bayes SAR method outperforms the RPCA algorithms with improving performance if either coherence or SCNR increases. Table 5.8 provides additional numerical results for the case $N = 20$. The RCPA algorithms

[2] For the optimization-based approach, we used the exact_alm_rpca package (MATLAB) by Lin et al. [63], downloaded from `http://watt.csl.illinois.edu/perceive/matrix-rank/home.html`. For the Bayesian-based approach, we used the Bayesian robust PCA package, downloaded from `http://www.ece.duke.edu/~lihan/brpca_code/BRPCA.zip`.

perform poorly in reconstructing the sparse component with relative errors near or greater than 1. This reflects the fact that (a) these algorithms miss significant sources of information, such as the correlations among antennas and among quadrature components, and (b) $N = 20$ may be too few samples to reliably estimate the principal components in these non-parametric models. In measured SAR imagery, it might be unreasonable to expect $N \gg 20$ passes of the radar, which suggests that these RPCA algorithms will likely perform poorly on such signals. In contrast, it is seen that the Bayes SAR method obtains low reconstruction errors for both low-dimensional and sparse components as either coherence or SCNR increase.

Since both Bayes SAR and BRPCA are hierarchical Bayesian models for separating signals into low-dimensional and sparse components, one might wonder why the performance of the two algorithms is so different in these simulations. The main underlying reason for the difference in the performance is that BRPCA models the low-dimensional signal as low-rank, so that each pixel can be described by a few linear factors, while Bayes SAR assumes that the low-dimensional background is random, but can be described by a few multivariate Gaussian distributions. This suggests that the random variations in the low-dimensional component caused by speckle and glints cannot be adequately modeled using a low-rank assumption (as in BRPCA). It should be noted that both Bayes SAR and BRPCA have very similar models for the sparse target component, using temporal/spatially Markov models for the target indicators.

5.7.2 Measured data

In this section, we compare performance of the Bayes SAR approach using a set of measured data from the 2006 Gotcha SAR sensor collection. In particular, images were formed from phase histories collected over a scene of size 375m by 1200m for $N = 3$ passes and $K = 3$ antennas. Each image was created with a coherent processing time of 0.5 seconds with the addition of a Blackman-Harris window in the azimuth direction to reduce sidelobes. Images were created at overlapping intervals spaced 0.25 seconds apart for a total of 18 seconds. Note that the ability to take advantage of correlated images (as in this case) is one of the benefits of using the proposed model/inference algorithm.

We consider three alternative approaches in comparison to the Bayes SAR approach: (1) displaced-phase center array (DPCA) processing, (2) along-track interferometry (ATI), and (3) a mixture of DPCA/ATI. Note that all variants of ATI/DPCA depend on the chosen thresholds for phase/magnitude, respectively.

Comparisons to DPCA/ATI

We begin by comparing the output of the proposed algorithm across the entire 375m by 1200m scene. Figure 5.5 shows the output of the Bayes SAR algorithm and the DPCA/ATI comparisons. It is seen that there are significant performance gains by using calibrated images as shown in (c) and (f) as compared to their original versions, (b) and (e), respectively. Furthermore, the proposed approach also provides a sparse output without choosing thresholds as required by DPCA/ATI. Note that in this figure, calibration is accomplished by using the outputs $\boldsymbol{H}_{f,i}$ from the Bayes SAR approach.

Figures 5.6 and 5.7 display the detection performance over two smaller scenes of size 125m by 125m as a function of magnitude and phase, respectively. For each scene, images are provided for sequential scenes separated by 0.5 seconds. Scene 1 contains strong clutter in the upper left region, while Scene 2 has relatively little clutter. It is seen that the proposed approach (2nd column) provides a sparse solution containing the targets of interest in each of the 4 images. Moreover, the 3rd column provides the estimated probability that a target occupies a given pixel, in comparison to the (0,1) output of DPCA. Although most estimated probabilities are near 1, there are a few cases where this is not the situation: in scene 2(d), a low-magnitude target is detected with low probability in the lower-right; in scene 1(b) a few target pixels from the clutter region are detected with low probability. In contrast, the performance of DPCA depends strongly on the threshold. In Scene 1, a 30 dB threshold provides a large number of false alarms. However, in Scene 2, the low-magnitude targets are missed for the 15 dB threshold, but detected at the 30 dB threshold.

Figure 5.7 shows the detection performance based on phase over the same 4 images. It is once again seen that the performance of the ATI/DPCA algorithms depend strongly on the thresholds, with performance that varies across thresholds from image to image. On the other hand, the proposed approach is able to detect the targets with high fidelity regardless of the scene/image and does not require tuning of thresholds for detection.

Target motion models

Figure 5.8 shows the output of the proposed approach when prior information on the location of the targets might be available. For example, in the shown scene, targets are likely to be stopped at an intersection. The performance improvement is given for a mission scene that contains target in this high probability region. On the other hand, there are no significant performance decreases in the reference scene that does not contain targets in the intersection region. This type of processing could be extended to a tracking environment, where targets are projected to likely be in a

Table 5.9: Radial velocity estimation (m/s) in 2006 Gotcha collection dataset

(a) Target 1

Algorithm	Bias	MSE	No. Missed
Raw	0.64	0.94	1
Calibrated	0.71	1.02	0
Bayes SAR	0.03	0.10	0
ATI/DPCA*	-0.04	0.20	27
ATI/DPCA**	0.10	0.20	2

(b) Target 2

Algorithm	Bias	MSE	No. Missed
Raw	0.47	0.77	6
Calibrated	0.48	0.79	0
Bayes SAR	0.19	0.22	0
ATI/DPCA*	-0.07	0.43	30
ATI/DPCA**	0.23	0.28	3

given location within the formed SAR image as discussed in Section 5.4.

Estimation of radial velocity

The dataset used in this section contained a few GPS-truthed vehicles from which we can derive (a) the ‘true’ location of the target within the formed SAR image, and (b) the target’s radial velocity which is known to be proportional to the measured interferometric phase of the target pixels in an along-track system. Figure 5.9 shows the estimated radial velocities for two targets over 18 seconds at 0.25 second increments. We compare the estimation of radial velocity from the output of the Bayes SAR algorithm, from the raw images, from the calibrated images, and from two DPCA/ATI joint algorithms with phase/magnitude thresholds of (25 deg, 15 dB) and (25 deg, 30 dB) respectively. For fair comparisons, the DPCA/ATI thresholds are applied to the calibrated imagery, though this is a non-trivial step in general. Numerical results are summarized in Table 5.9. It is seen that the Bayes SAR algorithm outperforms the others in terms of MSE for both targets. Moreover, the Bayes SAR algorithm never misses a target detection in this dataset, which is not the case for the DPCA/ATI algorithms.

Performance prediction

Since the proposed SAR image model estimates the statistics of the background component directly, we can predict performance for detection and estimation for future passes without the entire machinery of the hierarchical Bayesian model. Figure 5.10 shows an example of using the Bayes SAR model in order to derive likelihood ratio tests according to the test statistic given in equation (5.34). Similarly, Figure 5.11 provides an example for bounds on estimation errors for $x-$ and $y-$ positions given the outputs of the Bayes SAR model.

5.8 Discussion and future work

This chapter extends research in decomposing high-dimensional signals/images into low-rank and sparse components in the presence of noise [19, 63, 93] to the case of separating target signatures from a low-dimensional clutter subspace in SAR imagery. In particular, we combine our understanding of the physical, kinematic, and statistical properties of SAR imagery into a single unified Bayesian structure that simultaneously (a) estimates the nuisance parameters such as clutter distributions and antenna miscalibrations and (b) extracts a sparse component containing the target signatures required for detection and estimation of the target state. Similar to Ding et al. [33], this algorithm requires few tuning parameters since most quantities of interest are inferred directly from the data - this allows the algorithm to be robust to a large collection of operating conditions. The performance of the proposed approach is analyzed over both simulated and measured datasets, demonstrating competing or better performance than the robust PCA algorithms and ATI/DPCA. Moreover, it is shown that the outputs of the Bayesian inference can be used for future performance prediction through examples of derived likelihood ratio tests and Cramér-Rao Lower Bounds for spatial errors.

Future work will include the development of algorithms that exploit the use of a posterior distribution for improved performance in a signal processing task, e.g. detection, tracking or classification. In particular, we are interested in using algorithms for simultaneously detecting and estimating targets over a sparse scene with resource constraints as discussed in Chapters II and III, as well determining the fundamental performance limits of a SAR target tracking system. Furthermore, we would also like to consider other generalizations to the SAR image model, such as complex target maneuvers, multiple target classes, and explicit tracking of the target phase.

5.9 Appendix: Target signature prediction

In some applications, such as target tracking or sequential detection, we may have access to an estimate of the kinematic state of the target(s) of interest, such as position, velocity and acceleration. This may be useful for predicting the location of the target at sequential frames. For simplicity, consider a single target whose state $(\boldsymbol{r}(\tau), \dot{\boldsymbol{r}}(\tau))$ is known with standard errors $(\sigma_r, \sigma_{\dot{r}})$, where τ denotes the slow-time (i.e., time of the radar pulse). In standard SAR image formation, moving targets tend to appear displaced and defocused in as described in the literature by Fienup [42] and Jao [54]. Moreover, Jao shows that given the radar trajectory $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and the target trajectory $(\boldsymbol{r}, \dot{\boldsymbol{r}})$, one can predict the location of the target signature within the image $\boldsymbol{p}$ by solving a system of equations that equate Doppler shifts and ranges, respectively, at each pulse:

$$\frac{d}{d\tau}\left[\|\boldsymbol{p}-\boldsymbol{q}(\tau)\|_2-\|\boldsymbol{r}(\tau)-\boldsymbol{q}(\tau)\|_2\right]_{\boldsymbol{p}=\boldsymbol{p}^*}=0 \tag{5.35}$$

$$\|\boldsymbol{p}^*-\boldsymbol{q}(\tau)\|_2=\|\boldsymbol{r}(\tau)-\boldsymbol{q}(\tau)\|_2\,, \tag{5.36}$$

which can be reduced to the simpler system of equations:

$$\dot{\boldsymbol{q}}(\tau)\cdot[\boldsymbol{p}^*-\boldsymbol{q}(\tau)]=[\dot{\boldsymbol{r}}(\tau)-\dot{\boldsymbol{q}}(\tau)]\cdot[\boldsymbol{r}(\tau)-\boldsymbol{q}(\tau)] \tag{5.37}$$

$$\|\boldsymbol{p}^*-\boldsymbol{q}(\tau)\|_2=\|\boldsymbol{r}(\tau)-\boldsymbol{q}(\tau)\|_2 \tag{5.38}$$

In practice, the target state $(\boldsymbol{r},\dot{\boldsymbol{r}})$ is unknown or known with some uncertainty. In the latter case, we can predict the probable locations of the target signature by one of several methods, including:

- Monte Carlo estimation of the target signature locations.
- Gaussian approximation using linearization or the unscented transformation.
- Analytical approximation as proposed by Newstadt et al. [70].

5.9.1 Notation

Following the derivation of Jao [54], we will assume the following notation:

- $\mathbf{r}(\tau)=(r_x,r_y,r_z)$ is the position of a point scatterer.
- $\mathbf{q}(\tau)=(q_x,q_y,q_z)$ is the position of the radar platform.
- $\dot{\mathbf{r}}(\tau)=(\dot{r}_x,\dot{r}_y,\dot{r}_z)$ is the velocity of a point scatterer.
- $\dot{\mathbf{q}}(\tau)=(\dot{q}_x,\dot{q}_y,\dot{q}_z)$ is the true position of the platform.
- $\mathbf{p}=(p_x,p_y,p_z)$ is a pixel location within the image.
- τ represents the slow-time (i.e., pulse of the radar sample).

5.9.2 Deterministic solution

In the deterministic case, where $\mathbf{r}, \dot{\mathbf{r}}, \mathbf{q}$, and $\dot{\mathbf{q}}$ are all known, we can find the pixel $\boldsymbol{p}^*$ where the target signature will be focused at time τ by solving equations (5.25) and (5.26). In particular, if we assume that z-coordinate is given by a function

$$p_z = h(p_x, p_y), \tag{5.39}$$

then we can give explicit expressions for (p_x, p_y, p_z) in some cases of h. We will focus on the simple case where $h(p_x, p_y) = z_0$ (i.e, constant elevation), though this can be easily extended to other cases (for example, with a depth elevation map).

To solve the system of equations, let

$$\alpha(\tau) = \|\mathbf{r}(\tau) - \mathbf{q}(\tau)\|_2^2 \tag{5.40}$$

$$\beta(\tau) = \dot{\mathbf{q}}(\tau) \cdot \mathbf{r}(\tau) - \dot{\mathbf{r}}(\tau) \cdot (\mathbf{r}(\tau) - \mathbf{q}(\tau)) \tag{5.41}$$

Then we have

$$\begin{aligned} \alpha(\tau) &= \|\mathbf{p}^* - \mathbf{q}(\tau)\|_2^2 \\ &= (p_x^* - q_x(\tau))^2 + \left(p_y^* - q_y(\tau)\right)^2 + (z_0 - q_z(\tau))^2 \end{aligned} \tag{5.42}$$

and re-arranging equation (5.25) we have

$$\begin{aligned} \beta(\tau) &= \dot{\mathbf{q}}(\tau) \cdot \mathbf{p}^* \\ &= \dot{q}_x p_x^* + \dot{q}_y p_y^* + \dot{q}_z z_0 \end{aligned} \tag{5.43}$$

For this derivation, assume that $\dot{q}_x \neq 0^3$. Therefore, solving for p_x^*, we get:

$$\begin{aligned} p_x^* &= \frac{\beta(\tau) - \dot{q}_y p_y^* - \dot{q}_z z_0}{\dot{q}_x} \\ &= \left(\frac{\beta(\tau) - \dot{q}_z z_0}{\dot{q}_x}\right) + \left(-\frac{\dot{q}_y}{\dot{q}_x}\right) p_y^* \\ &= \gamma_0 + \gamma_1 p_y^* \end{aligned} \tag{5.44}$$

[3] By assumption, the radar has non-zero velocity in the xy-plane. Thus, if $\dot{q}_x = 0$, then this derivation should be valid if we switch the x and y indices.

Plugging into equation (5.42) we get:

$$\alpha(\tau) = \left(\gamma_0 + \gamma_1 p_y^* - q_x(\tau)\right)^2 + \left(p_y^* - q_y(\tau)\right)^2 + \left(z_0 - q_z(\tau)\right)^2 \tag{5.45}$$

which can be re-arranged as

$$a_y(p_y^*)^2 + b_y p_y^* + c_y(p_y^*)^2 = 0 \tag{5.46}$$

where

$$\begin{aligned} a_y &= \left(1 + \gamma_1^2\right) \\ b_y &= 2\left(\gamma_0\gamma_1 - \gamma_1 q_x(\tau) - q_y(\tau)\right) \\ c_y &= \left(\gamma_0 - q_x(\tau)\right)^2 + \left(q_y(\tau)\right)^2 + \left(z_0 - q_z(\tau)\right)^2 - \alpha(\tau) \\ &= \left(\gamma_0^2 + z_0^2\right) - 2\left(\gamma_0 q_x(\tau) + z_0 q_z(\tau)\right) + \|\mathbf{q}(\tau)\|_2^2 - \|\mathbf{r}(\tau) - \mathbf{q}(\tau)\|_2^2 \end{aligned} \tag{5.47}$$

Then p_y^* is given by the solution of the quadratic equation:

$$p_y^* = \frac{-b_y \pm \sqrt{b_y^2 - 4a_y c_y}}{2a_y} \tag{5.48}$$

and p_x^* is given by equation (5.44). This solution suggests that the target energy will generally actually appear at two locations. However, in most cases only one of these locations will be in the formed SAR image. Thus, we generally choose the solution (p_x^*, p_y^*) that is closest to the scene center $(0, 0)$.

Finally, we note that equations (5.44) and (5.48) provide the pixel location containing the target energy at a single pulse time, τ. Generally, images are formed by integrating pulses over a coherent processing interval (CPI) containing multiple times $\tau \in [T_0, T_1]$.

5.9.3 Uncertainty model

It is unlikely that we will have perfect information regarding the target state $\boldsymbol{r}(\tau)$ and $\dot{\boldsymbol{r}}(\tau)$ at all times τ. On the other hand, there are special cases where we might have some information about these states that include

- A tracking environment, where we estimate target position and velocities with standard errors, σ_r and $\sigma_{\dot{r}}$, respectively.
- A persistently monitored scene, where we have knowledge of traffic patterns or road systems. In such a case, we have prior knowledge of likely target states.

In either case, we have a characterization of likely behavior of the target kinematic state. We can represent this knowledge in many ways that could include

- A linear kinematic model, where

$$\begin{aligned} \boldsymbol{r}(\tau) &= \boldsymbol{r}_0 + \boldsymbol{v}\tau + \boldsymbol{a}\tau^2/2 \\ \boldsymbol{r}_0 &\sim N\left(\boldsymbol{\mu}_r, \sigma_r^2\boldsymbol{I}\right) \\ \boldsymbol{v} &\sim N\left(\boldsymbol{\mu}_v, \sigma_v^2\boldsymbol{I}\right) \\ \boldsymbol{a} &\sim N\left(\boldsymbol{\mu}_a, \sigma_a^2\boldsymbol{I}\right) \end{aligned} \tag{5.49}$$

- A random kinematic model, where at each time τ

$$\begin{aligned} \boldsymbol{r}(\tau) &\sim N\left(\boldsymbol{\mu}_r(\tau), [\sigma_r(\tau)]^2\boldsymbol{I}\right) \\ \dot{\boldsymbol{r}}(\tau) &\sim N\left(\boldsymbol{\mu}_{\dot{r}}(\tau), [\sigma_{\dot{r}}(\tau)]^2\boldsymbol{I}\right) \end{aligned} \tag{5.50}$$

Note that both models assume that the position and velocity vectors are Gaussian distributed. However, the first model is characterized by only 6 random variables (2 each for position, velocity, and acceleration) regardless of the number of pulses. The second model, on the other hand, assumes 4 random variables for each pulse τ. In fact, the first model can be seen as a specialization of the second model for specific structures for the mean and variance parameters.

In this document, the choice of target kinematic model depends on the inference method which we will use to derive the distribution of the target locations. In Monte Carlo sampling, the choice of model is of relatively insignificant computational burden as compared to the generation of the Monte Carlo samples. On the other hand, in the analytical approximation methods, the choice of target kinematic model is of great importance.

The goal of this section is to provide a prediction model for the locations of targets within a SAR image given a target kinematic model. In particular, we define this model through the distribution of pixel locations:

$$f(p_x, p_y) \tag{5.51}$$

which is assumed to have support on $\mathbb{R}^2$. Generally, images are formed on a discrete grid so that we should really consider a discrete distribution. However, for simplicity we consider a continuous domain in this section.

Moreover, we have to be careful how we define the distribution of pixel locations for SAR images formed by integrating multiple radar pulses. Consider a probability distribution function (PDF) for the target location at pulse τ_i given by $f(p_x(\tau_i), p_y(\tau_i))$. We define the target distribution of interest as:

$$f(p_x, p_y) = \frac{1}{T}\sum_{i=1}^{T} f(p_x(\tau_i), p_y(\tau_i)) \tag{5.52}$$

This is equivalent to the distribution of the target occupying location (p_x, p_y) at any of T integrated pulses. We could consider a richer description by solving for the joint distribution on $\{p_x(\tau_i), p_y(\tau_i)\}_{i=1}^{T}$. However, this will be generally very high-dimensional and might not provide any additional benefit over the distribution given by equation (5.52) be useful for the purposes described in the paper.

Finally, since we will solve for the distributions $f(p_x(\tau_i), p_y(\tau_i))$ independently for each τ_i, we will consider the random kinematic model only. In the naive situation where the distributions of target locations are independent over time, this will provide solutions that can be approximated analytically with just a few minor assumptions.

5.9.4 Monte Carlo prediction

The most straightforward way to approximate the distribution in equation (5.52) is to use Monte Carlo sampling from the linear/random target kinematic models, followed by projection of those target states into the image domain using equations (5.44) and (5.48). A Monte Carlo representation is subsequently given by the average number of samples occupying any pixel. Note that since the Monte Carlo representation will contain discrete samples, we will end up with a discrete probability mass function (PMF) rather than a PDF.

5.9.5 Gaussian approximation

Rather than using a potentially high-dimensional PMF or PDF representation of the target location at time τ_i, we could consider a Gaussian approximation that represents the probability distribution with just two parameters: the mean $\boldsymbol{\mu}_i \in \mathbb{R}^2$ and the covariance $\boldsymbol{\Sigma}_i \in \mathbb{R}^{2\times 2}$ for each slow-time. Then the PDF is given by a Gaussian mixture model of form:

$$f(\boldsymbol{p}) \approx \frac{1}{T}\sum_{i=1}^{T} \phi_{\mathcal{CN}}\left(\boldsymbol{p}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \tag{5.53}$$

where $\phi_{\mathcal{CN}}\left(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ is the multivariate normal distribution PDF of $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

To find the means $\boldsymbol{\mu}_i$ and covariances $\boldsymbol{\Sigma}_i$, we could consider linearization of the solution to equations (5.44) and (5.48) around a particular state $\boldsymbol{\xi}(\tau_i) \triangleq \{\boldsymbol{r}(\tau_i), \dot{\boldsymbol{r}}(\tau_i)\}$. This is akin to the approximation made by the Extended Kalman Filter, where the state is Gaussian but the observations are non-linear. Let us define

$$g(\boldsymbol{\xi}(\tau_i)) = \begin{bmatrix} p_x^*(\boldsymbol{\xi}(\tau_i)) \\ p_y^*(\boldsymbol{\xi}(\tau_i)) \end{bmatrix} \tag{5.54}$$

where p_x^* and p_y^* are given by equations (5.44) and (5.48). Then the first-order linearization around the mean $\boldsymbol{\mu}_\xi$ is given by:

$$G(\boldsymbol{\xi}) = g(\boldsymbol{\mu}_\xi) + \left(\nabla g(\boldsymbol{\xi})\big|_{\boldsymbol{\xi}=\boldsymbol{\mu}_\xi}\right)^T (\boldsymbol{\xi} - \boldsymbol{\mu}_\xi) \tag{5.55}$$

Since this is a linear function of a Gaussian distributed vector in $\boldsymbol{\xi}(\tau_i)$, we know that the pixels $\boldsymbol{p}(\tau_i) \approx G(\boldsymbol{\xi}(\tau_i))$ are distributed as

$$\boldsymbol{p}(\tau_i) \sim N\left(g(\boldsymbol{\mu}_\xi(\tau_i)), \left(\nabla g(\boldsymbol{\xi})\big|_{\boldsymbol{\xi}=\boldsymbol{\mu}_\xi(\tau_i)}\right)^T \boldsymbol{\Sigma}_{\boldsymbol{\xi}}(\tau_i) \left(\nabla g(\boldsymbol{\xi})\big|_{\boldsymbol{\xi}=\boldsymbol{\mu}_\xi(\tau_i)}\right)\right), \tag{5.56}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}(\tau_i)$ is given by

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}}(\tau_i) = \begin{bmatrix} \boldsymbol{\Sigma}_r(\tau_i) & 0 \\ 0 & \boldsymbol{\Sigma}_{\dot{r}}(\tau_i) \end{bmatrix} \tag{5.57}$$

5.9.6 Analytical approximation

It is also possible to get a closer approximation than the linearization example provided above by doing some analytical derivations. In particular, let us assume that the radar platform moves in the x-direction so that $\dot{q}_y = \dot{q}_z = 0$ and $|\dot{q}_x| > 0$. In the general case, this derivation would hold for a transformed set of coordinates (p'_x, p'_y), though we won't go into that derivation here. In the former case, equations (5.44) and (5.48) reduce to:

$$p_x^* = \frac{\beta(\tau)}{\dot{q}_x} = \frac{\dot{q}_x r_x - (r_x - q_x)v_x - (r_y - q_y)v_y}{\dot{q}_x} \tag{5.58}$$

$$p_y^* = q_y \pm \sqrt{\alpha(\tau) - (p_x - q_x)^2 - (z_0 - q_z)^2} \tag{5.59}$$

From equation (5.58), we see that

$$f(p_x^* | r_x, r_y) \sim N(\mu, \sigma^2) \tag{5.60}$$

where

$$\mu = \frac{1}{\dot{q}_x}\left([\dot{q}_x - \mu_{vx}]r_x - \mu_{vy}r_y + q_x\mu_{vx} + q_y\mu_{vy}\right) \tag{5.61}$$

$$\sigma^2 = \frac{(r_x - q_x)^2 + (r_y - q_y)^2}{(\dot{q}_x)^2}\sigma_v^2 \tag{5.62}$$

where we have assumed that $\Sigma_{\dot{r}} = \sigma_v^2 \boldsymbol{I}$. Note that σ^2 is a function of the position $\boldsymbol{r}$. However, since $\|q\| \gg \|r\|$ in general, we make a zero-th order approximation here so that

$$\sigma^2 \approx \sigma_0^2 \triangleq \frac{(\mu_{rx} - q_x)^2 + (\mu_{ry} - q_y)^2}{(\dot{q}_x)^2}\sigma_v^2 \tag{5.63}$$

In this case, we see that we can find

$$\begin{aligned} f(p_x) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(p_x|r_x, r_y) f(r_x, r_y) dr_x dr_y \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi_{\mathcal{CN}}(p_x; \mu(r_x, r_y), \sigma_0^2)\phi_{\mathcal{CN}}(r_x; \mu_{rx}, \sigma_r^2)\phi_{\mathcal{CN}}(r_y; \mu_{ry}, \sigma_r^2) dr_x dr_y \end{aligned} \tag{5.64}$$

where we have assumed that $\Sigma_r = \sigma_r^2 \boldsymbol{I}$. Since $\mu(r_x, r_y)$ is linear in r_x and r_y from equation (5.61), we can analytically solve this integral to see that

$$p_x^* \sim N(\mu_{px}, \sigma_{px}^2) \tag{5.65}$$

where

$$\mu_{px} = \frac{1}{\dot{q}_x}\left(q_x\mu_{vx} + q_y\mu_{vy} + \dot{q}_x\mu_{rx} - \mu_{vx}\mu_{rx} - \mu_{vy}\mu_y\right) \tag{5.66}$$

$$\sigma_{px}^2 = \frac{\sigma_r^2}{(\dot{q}_x)^2}\left[(\dot{q}_x - \mu_{vx})^2 + \mu_{vy}^2\right] + \sigma_0^2 \tag{5.67}$$

Since p_y^* in equation (5.59) is non-linear, we make one more assumption with a first-order linearization around $p_y^*(\boldsymbol{\mu}_r)$. Note note that given p_x^*, equation (5.59) only depends on the target state through $\boldsymbol{r}$ (and not on the velocity $\dot{\boldsymbol{r}}$). Define $s(\boldsymbol{r}, p_x)$ to be the value of equation (5.59) given state $\boldsymbol{r}$ and pixel location p_x. Then we approximate p_y^* as

$$p_y^*(\boldsymbol{r})|p_x^* \approx s(\boldsymbol{\mu}_r, p_x^*) + \left(\nabla s(\boldsymbol{\mu}_r, p_x^*)\big|_{\boldsymbol{r}=\boldsymbol{\mu}_r}\right)^T (\boldsymbol{r} - \boldsymbol{\mu}_r) \tag{5.68}$$

Finally, we note that given p_x, p_y is Gaussian distributed with distribution:

$$p_y|p_x \sim N(\mu_{py}, \sigma_{py}^2) \tag{5.69}$$

$$\mu_{py} = s(\boldsymbol{\mu}_r, p_x^*) \tag{5.70}$$

$$\sigma_{py}^2 = \sigma_r^2 \left(\nabla s(\boldsymbol{\mu}_r, p_x^*)\big|_{\boldsymbol{r}=\boldsymbol{\mu}_r}\right)^T \left(\nabla s(\boldsymbol{\mu}_r, p_x^*)\big|_{\boldsymbol{r}=\boldsymbol{\mu}_r}\right) \tag{5.71}$$

Note that both p_x^* and $p_y^* \mid p_x^*$ have Gaussian distributions that can be described by a mean and covariance term. In contrast to Section 5.9.5, the distribution is not jointly Gaussian because the mean and covariance of $p_y^* \mid p_x^*$ depend on p_x^*. Nevertheless, one can easily evaluate this PDF at any pixel (p_x, p_y). Over short CPIs, both approximations will probably lead to similar results.

5.10 Appendix: Inference Details

In the hierarchical model proposed in Section 5.3, the distribution of hyperparameters at the base layer are generally chosen to be conjugate to the distributions at the next layer. This allows for efficient approximation methods for the posterior distribution in the sense that we can sample exactly from these distributions. In particular, we use a Markov Chain Monte Carlo (MCMC) algorithm in the form of a Gibbs sampler to iteratively estimate the full joint posterior. In MCMC, this distribution is approximated by drawing samples iteratively from the conditional distribution of each (random) model variable given the most recent estimate of the rest of the variables (which we denote by $-$). Let $\boldsymbol{\Theta} = \left\{\boldsymbol{B}, \boldsymbol{X}, \boldsymbol{G}, \boldsymbol{M}, \boldsymbol{\Delta}^G, \boldsymbol{\Delta}^M, \boldsymbol{H}, \boldsymbol{C}, \boldsymbol{\eta}\right\}$ represent a current estimate of all of the model variables where $\boldsymbol{\eta}$ represents the set of all hyper-parameters. Given measurements $\boldsymbol{I}$, the inference algorithm is given in Figure 5.2. Note that MCMC algorithms require a burn-in period after the Markov chain has become stable, where the duration of burn-in period depends on the problem. After this point, we collect $N_{samples}$ samples that represent the full joint distribution.

The sampling details are provided for each of the steps in Figure 5.2 individually.

5.10.1 Basic Decomposition

Given the parameters in Tables 5.5 and 5.6, we arrive at one of the primary benefits of using Gibbs Sampling for inference: namely that we can independently sample across pixels and frames. In distributional form, we have

$$\begin{aligned} & f(\boldsymbol{B}, \boldsymbol{X}, \boldsymbol{G}, \boldsymbol{M}, \boldsymbol{\Delta}^G, \boldsymbol{\Delta}^M | \boldsymbol{I}, -) \\ & = \prod_{p,f} f(\underline{b}_f^{(p)}, \boldsymbol{x}_{f,1:N}^{(p)}, \boldsymbol{g}_{f,1:N}^{(p)}, \boldsymbol{m}_{f,1:N}^{(p)}, \delta_f^{G,(p)}, \boldsymbol{\delta}_{f,1:N}^{M,(p)} | \boldsymbol{I}, -) \end{aligned} \tag{5.72}$$

The conditional independence among pixels and frames given the nuisance parameters allows us to easily parallelize the sampling procedure over the largest dimensions of the state. Moreover, we can extend the parallelization to sampling independently over passes by separating the sampling of equation (5.72) into two Gibbs steps from the densities:

$$\begin{aligned} & f(\underline{b}_f^{(p)}, \boldsymbol{x}_{f,1:N}^{(p)}, \boldsymbol{m}_{f,1:N}^{(p)}, \boldsymbol{\delta}_{f,1:N}^{M,(p)} | \boldsymbol{I}, -) \\ & \quad = f(\underline{b}_f^{(p)} | \boldsymbol{I}, \boldsymbol{x}_{f,1:N}^{(p)}, \boldsymbol{m}_{f,1:N}^{(p)}, \boldsymbol{\delta}_{f,1:N}^{M,(p)}, -) \\ & \quad \cdot \prod_i f(\underline{x}_{f,i}^{(p)}, \underline{m}_{f,i}^{(p)} | \delta_{f,i}^{M,(p)}, \boldsymbol{I}, -) f(\delta_{f,i}^{M,(p)} | \boldsymbol{I}, -) \end{aligned} \tag{5.73}$$

$$\begin{aligned} & f(\boldsymbol{g}_{f,1:N}^{(p)}, \delta_f^{G,(p)} | \boldsymbol{I}, -) \\ & \quad = f(\delta_f^{G,(p)} | \boldsymbol{I}, -) \prod_i f(\underline{g}_{f,i}^{(p)} | \boldsymbol{I}, \delta_f^{G,(p)}, -) \end{aligned} \tag{5.74}$$

It should be noted that each of these distributions have explicit forms as either multivariate Gaussian or Bernoulli distributed. For example consider the distribution

$$f(\underline{x}_{f,i}^{(p)}, \underline{m}_{f,i}^{(p)} | \delta_{f,i}^{M,(p)}, \boldsymbol{I}, -) = f(\underline{x}_{f,i}^{(p)}, \underline{m}_{f,i}^{(p)} | \delta_{f,i}^{M,(p)}, \underline{r}_{f,i}^{(p)}, -) \tag{5.75}$$

where we define

$$\underline{r}_{f,i}^{(p)} \triangleq \left(\underline{i}_{f,i}^{(p)} ./ \underline{h}_{f,i}^{(p)}\right) - \delta_{f}^{G,(p)} \underline{g}_{f,i}^{(p)}, \tag{5.76}$$

where ./ is the point-wise division operator. In particular, we know that

$$\underline{r}_{f,i}^{(p)} = \underline{b}_{f}^{(p)} + \underline{x}_{f,i}^{(p)} + \delta_{f,i}^{M,(p)} \underline{m}_{f,i}^{(p)} + \underline{v}_{f,i}^{(p)}, \tag{5.77}$$

Thus, given $\delta_{f,i}^{M,(p)}$, we have $\underline{r}_{f,i}^{(p)}$ as a linear combination of Gaussian random variables. Thus, by standard conditional distributions of a Gaussian random vector, we know that the distribution of $\underline{b}_{f}^{(p)}$ and $\underline{x}_{f,i}^{(p)}$ must also be Gaussian. Table 5.10 gives the means and covariances of the Gaussian distributions in equations (5.73) and (5.74). Moreover, the distributions of the indicator functions are easily found by Bayes rule. For example, define the quantity

$$z_d = f(\underline{i}_{f,i}^{(p)} | \delta_{f,i}^{M,(p)} = d, \underline{h}_{f,i}^{(p)}, \delta_{f}^{G,(p)}, \underline{g}_{f,i}^{(p)}, \underline{b}_{f}^{(p)}). \tag{5.78}$$

Then it is easily seen that for the target indicators we have by Bayes rule

$$\begin{aligned} f(\delta_{f,i}^{M,(p)} = 1 | \boldsymbol{I}, -) &= \frac{z_1 f(\delta_{f,i}^{M,(p)} = 1)}{\sum\limits_{d=(0,1)} z_d f(\delta_{f,i}^{M,(p)} = d)} \\ &= \left[1 + \frac{1 - \pi_{f,i}^{M}(p)}{\pi_{f,i}^{M}(p)} \frac{z_0}{z_1} \right]^{-1} \end{aligned} \tag{5.79}$$

Note that z_d is just an evaluation of the Normal PDF so that it can be simply calculated. Table 5.11 provides explicit values of z_d for both the target and glint indicators, where $\phi_{\mathcal{CN}}(x; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the multivariate normal PDF with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

Table 5.10: Gaussian distribution parameters for distributions of base layer parameters in SAR image model equations (5.73) and (5.74)

Component	Variable	Mean	Covariance	Parameters
Background	$\underline{b}_f^{(p)}$	$\boldsymbol{\Lambda}\underline{r}_f^{(p)}$	$(\boldsymbol{I}_{K\times K} - \boldsymbol{\Lambda})\boldsymbol{\Gamma}^B(p)$	$\underline{r}_f^{(p)} = \sum_{i=1}^N (\underline{i}_{f,i}^{(p)}./\underline{h}_{f,i}^{(p)} - \underline{x}_{f,i}^{(p)} - \delta_f^{G,(p)}\underline{g}_{f,i}^{(p)} - \delta_{f,i}^{M,(p)}\underline{m}_{f,i}^{(p)})$ $\boldsymbol{\Lambda} = \boldsymbol{\Gamma}^B(p)\left(\boldsymbol{\Gamma}^B(p) + \boldsymbol{\Gamma}^V\right)^{-1}$
Speckle, Target	$\begin{bmatrix} \underline{x}_{f,i}^{(p)} \\ \underline{m}_{f,i}^{(p)} \end{bmatrix}$	$\boldsymbol{\Lambda}\underline{r}_{f,i}^{(p)}$	$\boldsymbol{\Sigma}_{11} - \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{12}$	$\boldsymbol{\Sigma}_{11} = \begin{bmatrix} \boldsymbol{\Gamma}^X(p) & 0 \\ 0 & \delta_{f,i}^{M,(p)}\boldsymbol{\Gamma}^M \end{bmatrix}, \boldsymbol{\Sigma}_{12} = \begin{bmatrix} \boldsymbol{\Gamma}^X(p) \\ \delta_{f,i}^{M,(p)}\boldsymbol{\Gamma}^M \end{bmatrix}$ $\underline{r}_{f,i}^{(p)} = \underline{i}_{f,i}^{(p)}./\underline{h}_{f,i}^{(p)} - \underline{b}_f^{(p)} - \delta_f^{G,(p)}\underline{g}_{f,i}^{(p)}$ $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}_{12}^T\left(\boldsymbol{\Gamma}^X(p) + \delta_{f,i}^{M,(p)}\boldsymbol{\Gamma}^M + \boldsymbol{\Gamma}^V\right)^{-1}$
Glints	$\underline{g}_{f,i}^{(p)}$	$\boldsymbol{\Lambda}\underline{r}_{f,i}^{(p)}$	$(\boldsymbol{I}_{K\times K} - \boldsymbol{\Lambda})\boldsymbol{\Gamma}^G$	$\underline{r}_{f,i}^{(p)} = \underline{i}_{f,i}^{(p)}./\underline{h}_{f,i}^{(p)} - \underline{x}_{f,i}^{(p)} - \delta_{f,i}^{M,(p)}\underline{m}_{f,i}^{(p)} - \underline{b}_f^{(p)}$ $\boldsymbol{\Lambda} = \delta_f^{G,(p)}\boldsymbol{\Gamma}^G\left(\delta_f^{G,(p)}\boldsymbol{\Gamma}^G + \boldsymbol{\Gamma}^V\right)^{-1}$

Table 5.11: Bernoulli distribution parameters for distributions of indicator variables in equations (5.73) and (5.74)

Component	Variable	$f(\delta = 1 \vert -)$	z_d	$\underline{r}^{(p)}$
Glints	$\delta_f^{G,(p)}$	$\left[1 + \frac{1-\pi_f^G(p)}{\pi_f^G(p)} \frac{z_0}{z_1}\right]^{-1}$	$\phi_{\mathcal{CN}}(\underline{r}^{(p)}/N; \mathbf{0}, d\mathbf{\Gamma}^G + \mathbf{\Gamma}^V)$	$\sum_{i=1}^N \left(\underline{i}_{f,i}^{(p)}./\underline{h}_{f,i}^{(p)} - \underline{b}_f^{(p)} - \underline{x}_{f,i}^{(p)} - \delta_{f,i}^{M,(p)} \underline{m}_{f,i}^{(p)}\right)$
Targets	$\delta_{f,i}^{M,(p)}$	$\left[1 + \frac{1-\pi_{f,i}^M(p)}{\pi_{f,i}^M(p)} \frac{z_0}{z_1}\right]^{-1}$	$\phi_{\mathcal{CN}}(\underline{r}^{(p)}; \mathbf{0}, d\mathbf{\Gamma}^M + \mathbf{\Gamma}^V)$	$\underline{i}_{f,i}^{(p)}./\underline{h}_{f,i}^{(p)} - \underline{b}_f^{(p)} - \underline{x}_{f,i}^{(p)} - \delta_f^{G,(p)} \underline{g}_{f,i}^{(p)}$

5.10.2 Calibration coefficients

For this thesis, we assumed that pixels within a subset $Z_g \subset \{1, 2, \ldots, P\}$ share the same calibration constant so that

$$h_{k,f,i}^{(p)} = z_{k,f,i}(g), \qquad \forall p \in Z_g. \tag{5.80}$$

with $z_{k,f,i}(g) \sim \mathcal{CN}\left(1, (\sigma^H)^2\right)$. In our formulation (and dropping the (k, f, i) indices for simplicity) we have measurements of the form

$$i^{(p)} = z(g)(l^{(p)} + s^{(p)} + v^{(p)}), \qquad \forall p \in Z_g. \tag{5.81}$$

Define $y^{(p)} = l^{(p)} + s^{(p)}$ which is a known quantity in our Gibbs sampling inference step. Moreover, we assume that $|y^{(p)}| \gg |v^{(p)}|$ so that for any $p \in Z_g$ we have

$$\begin{aligned} i^{(p)} &= z(g)\left(y^{(p)} + v^{(p)}\right) \\ &\approx z(g)y^{(p)} + E[z(g)]v^{(p)} \\ &= z(g)y^{(p)} + v^{(p)} \end{aligned} \tag{5.82}$$

Note that given $y^{(p)}$ (as in the Gibbs sampling step), we have the situation where $i^{(p)}$, $z(g)$, and $v^{(p)}$ are all Gaussian distributed random variables. Thus, the conditional distribution of $z(g)$ is also Gaussian with:

$$z(g)|\boldsymbol{y} \sim \mathcal{CN}\left(\mu_z(g), \sigma_z^2(g)\right) \tag{5.83}$$

$$\mu_z(g) = 1 + \left[\frac{(\sigma^H)^2}{(\sigma^V)^2 + \boldsymbol{y}^H\boldsymbol{y}(\sigma^H)^2}\right]\boldsymbol{y}^H\left(\boldsymbol{i} - \boldsymbol{y}\right) \tag{5.84}$$

$$\sigma_z^2(g) = \frac{(\sigma^H)^2(\sigma^V)^2}{(\sigma^V)^2 + \boldsymbol{y}^H\boldsymbol{y}(\sigma^H)^2} \tag{5.85}$$

where $\boldsymbol{y} = \left\{y^{(p)}\right\}_{p\in Z_g}$ and $\boldsymbol{i} = \left\{i^{(p)}\right\}_{p\in Z_g}$. Note that when $(\sigma^H)^2$ is large, then maximum likelihood inference in this case yields the least-squares solution for $z(g)$.

5.10.3 Object class assignment

In this model, we assume that each pixel can be assigned to one of J possible classes. We assume that the number J is known a priori and do not consider the details involved in the merging or splitting of object classes here. More detailed models (such as the so-called Indian Buffet processes) can also estimate the number of classes directly from the data.

In this model, inference on class assignment is straightforward given the distributions (i.e., covariance matrices) for each class. Define the matrices

$$\boldsymbol{b}^{(p)} = \begin{bmatrix} \underline{b}_1^{(p)} \\ \underline{b}_2^{(p)} \\ \vdots \\ \underline{b}_F^{(p)} \end{bmatrix} \in \mathbb{C}^{F \times K} \qquad \boldsymbol{x}^{(p)} = \begin{bmatrix} \underline{x}_{1,1}^{(p)} \\ \underline{x}_{1,2}^{(p)} \\ \vdots \\ \underline{x}_{F,N}^{(p)} \end{bmatrix} \in \mathbb{C}^{FN \times K} \tag{5.86}$$

Then the probability that pixel p belongs to class j is given by:

$$w_j^{(p)} \triangleq \Pr(\text{pixel } p \text{ has class } j) = \exp\{T_B + T_X + q_j\} \tag{5.87}$$

where q_j is the prior probability of class j and

$$T_B = -\text{trace}\left([\boldsymbol{\Gamma}^B(j)]^{-1}\boldsymbol{b}^{(p)}(\boldsymbol{b}^{(p)})^H\right) - \frac{F}{2}\log|\boldsymbol{\Gamma}^B(j)| - KF\log(2\pi) \tag{5.88}$$

$$T_X = -\text{trace}\left([\boldsymbol{\Gamma}^X(j)]^{-1}\boldsymbol{x}^{(p)}(\boldsymbol{x}^{(p)})^H\right) - \frac{FN}{2}\log|\boldsymbol{\Gamma}^X(j)| - KFN\log(2\pi) \tag{5.89}$$

Then the class assignment to pixel p is the single location in $\underline{c}^{(p)}$ with value equal to one, where

$$\underline{c}^{(p)} \sim \text{Multinomial}(1; \underline{w}^{(p)}) \tag{5.90}$$

Note that we can improve upon this model by allowing the probabilities for pixel p to vary spatially (i.e., pixels are likely to share the same class with neighboring pixels). One simple way to include this information is to let

$$\underline{m}^{(p)} = \underline{w}^{(p)} * \underline{g}_{HMM}^{(p)} \tag{5.91}$$

where $\underline{g}_{HMM}^{(p)}$ is some filter for averaging nearby pixels and $*$ is the convolution operator (assumed to be supported on the same set of pixels as $\underline{w}^{(p)}$). Then we draw

$$\underline{c}^{(p)} \sim \text{Multinomial}(1; \underline{m}^{(p)}) \tag{5.92}$$

5.10.4 Hyper-parameters

In this model, we have three types of hyper-parameters that need to be estimated: covariance matrices (or variances), indicator probabilities, and object class probabilities. In all cases, the distribution of these parameters depend on test statistics of much smaller dimension that P.

Covariance matrix inference

We model the covariance matrices for the Normal distributions in two ways: (1) for the stationary components (background, speckle, and glints), we model the covariance matrix as a random variable; and (2) for the other components (targets, additive noise, calibration coefficients), we assume independence among the antennas. In particular, consider a random vector of K elements, $\underline{w}$, with

$$\underline{w} \sim \mathcal{CN}\left(\mathbf{0}, \sigma^2 \mathbf{\Gamma}_\rho\right) \tag{5.93}$$

$$\sigma^2 \sim \text{InvGamma}(a_\sigma, b_\sigma) \tag{5.94}$$

Then in the stationary case, we have

$$\mathbf{\Gamma}_\rho \sim \text{InvWishart}\left(a_\Gamma((1-\rho)\boldsymbol{I}_{K\times K} + \rho \mathbf{1}_K \mathbf{1}_K^T), \nu_\Gamma\right) \tag{5.95}$$

$$\rho \sim \text{Beta}(a_\rho, b_\rho) \tag{5.96}$$

and in the independent case, we have

$$\boldsymbol{\Gamma}_\rho = \boldsymbol{I}_{K\times K} \tag{5.97}$$

First consider the case where $\boldsymbol{\Gamma}_\rho$ is a random variable. Assume that we have n independent samples of $\underline{w}$, which we refer to as $\boldsymbol{W} = \text{vec}\{\underline{w}\}$. Then, we consider a Gibbs sampling procedure:

$$\text{Sample} \sim f(\sigma^2|\boldsymbol{W}, \boldsymbol{\Gamma}_\rho, \rho) \tag{5.98}$$

$$\text{Sample} \sim f(\boldsymbol{\Gamma}_\rho, \rho|\boldsymbol{W}, \sigma^2) \tag{5.99}$$

Let $\tau = 1/\sigma^2 \sim \text{Gamma}(a_\sigma, b_\sigma)$. Then

$$\begin{aligned} f(\tau|\boldsymbol{W}, \boldsymbol{\Gamma}_\rho, \rho) &\propto f(\boldsymbol{W}|\tau, \boldsymbol{\Gamma}_\rho) f(\tau) \\ &\propto \left[\tau^{n/2} \exp\left\{-\frac{\tau}{2}\text{trace}(\boldsymbol{\Gamma}_\rho^{-1}\boldsymbol{W}\boldsymbol{W}^H)\right\}\right] \left[\frac{(b_\sigma)^{a_\sigma}}{\Gamma(a_\sigma)}\tau^{a_\sigma-1}\exp\{-b_\sigma\tau\}\right] \\ &\propto \frac{(b')^{a'}}{\Gamma(a')}\tau^{a'-1}\exp\{-b'\tau\} \end{aligned} \tag{5.100}$$

where

$$\begin{aligned} a' &= a_\sigma + \frac{n}{2} \\ b' &= b_\sigma + \frac{\text{trace}(\boldsymbol{\Gamma}_\rho^{-1}\boldsymbol{W}\boldsymbol{W}^H)}{2} \end{aligned} \tag{5.101}$$

This demonstrates that in this situation, σ^2 has an Inverse-Gamma distribution with parameters a' and b'. Note that in the case where $\boldsymbol{\Gamma}_\rho = \boldsymbol{I}_{K\times K}$, then the posterior parameters are given by

$$a' = a_\sigma + \frac{n}{2} \tag{5.102}$$

$$b' = b_\sigma + \frac{\text{trace}(\boldsymbol{W}\boldsymbol{W}^H)}{2} \tag{5.103}$$

Thus, in the Gibbs sampling procedure, the variance parameter σ^2 is Inverse-Gamma distributed whether or not $\boldsymbol{\Gamma}_\rho$ is modeled as a random variable. Table 5.12 provides the posterior Inverse Gamma distribution parameters for the variance parameters in our model, where vec $\{\cdot\}$ refers to the vectorization operator.

For the background, speckle, and glint components, we also need to sample the coherence parameter ρ and correlation matrix $\mathbf{\Gamma}_\rho$. Let $\tilde{\boldsymbol{W}} = \boldsymbol{W}/\sigma$ be our observed measurements given σ^2 as given by equation (5.99). Define $\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu}/\sigma$. Then we have

$$\begin{aligned} \tilde{\boldsymbol{W}}| \left(\mathbf{\Gamma}_\rho, \sigma^2, \rho\right) &\sim \mathcal{CN}\left(\tilde{\boldsymbol{\mu}}, \mathbf{\Gamma}_\rho\right) \\ \mathbf{\Gamma}_\rho &\sim \text{InvWishart}\left([\rho \mathbf{1}_K \mathbf{1}_K^T + (1-\rho)\boldsymbol{I}_{K\times K}](\nu - K - 1), \nu\right) \\ \rho &\sim \text{Beta}(a_\rho, b_\rho) \end{aligned} \tag{5.104}$$

Note that this is in the form of the Multivariate-Normal-Inverse-Wishart conjugate distribution given ρ. This leads to the well known posterior parameters:

$$\mathbf{\Gamma}_\rho| \left(\tilde{\boldsymbol{W}}, \sigma^2, \rho\right) \sim \text{InvWishart}(\mathbf{\Lambda}_\rho a_\Gamma + \sum_{m=1}^{n} \left(\underline{y}_m - \tilde{\boldsymbol{\mu}}\right)\left(\underline{y}_m - \tilde{\boldsymbol{\mu}}\right)^H, \nu_\Gamma + n) \tag{5.105}$$

where $\mathbf{\Lambda}_\rho = \rho \mathbf{1}_K \mathbf{1}_K^T + (1-\rho)\boldsymbol{I}_{K\times K}$. Ideally, we would like to sample both $\mathbf{\Gamma}_\rho$ and ρ jointly. Even though we can simply sample from equation (5.105), the same is not true for the density

$$\rho| \left(\tilde{\boldsymbol{W}}, \sigma^2, \rho\right) \tag{5.106}$$

which is required in order to jointly sample these parameters. Fortunately, we know that

$$f(\mathbf{\Gamma}_\rho, \rho | \tilde{\boldsymbol{W}}, \sigma^2) \propto f(\mathbf{\Gamma}_\rho | \tilde{\boldsymbol{W}}, \sigma^2, \rho) f(\rho) \tag{5.107}$$

which is easily evaluated since we have closed form functions for both of these densities. Thus, we can use Metropolis-Hastings to sample ρ and $\mathbf{\Gamma}_\rho$.

Table 5.12: Inverse Gamma distribution parameters for distributions of variances and covariance matrix estimates

Component	Variable	a'	b'	Other parameters
Background	$(\sigma^B(j))^2$	$a_\sigma + \frac{M}{2}$	$\frac{\text{trace}(\mathbf{\Gamma}_\rho^{-1}\boldsymbol{W}\boldsymbol{W}^H)}{2}$	$M = F\lvert Q_j\rvert, \boldsymbol{W} = \text{vec}\left\{\underline{b}_f^{(p)}\right\}_{f,p\in Q_j}$
Speckle	$(\sigma^X(j))^2$	$a_\sigma + \frac{M}{2}$	$\frac{\text{trace}(\mathbf{\Gamma}_\rho^{-1}\boldsymbol{W}\boldsymbol{W}^H)}{2}$	$M = NF\lvert Q_j\rvert, \boldsymbol{W} = \text{vec}\left\{\underline{x}_{f,i}^{(p)}\right\}_{f,i,p\in Q_j}$
Glints	$(\sigma^G)^2$	$a_\sigma + \frac{M}{2}$	$\frac{\text{trace}(\mathbf{\Gamma}_\rho^{-1}\boldsymbol{W}\boldsymbol{W}^H)}{2}$	$M = \sum_{f,p} \delta_f^{G,(p)}, \boldsymbol{W} = \text{vec}\left\{\underline{g}_{f,i}^{(p)}\right\}_{\left\{p,f,i:\delta_f^{G,(p)}=1\right\}}$
Target	$(\sigma^M)^2$	$a_\sigma + \frac{M}{2}$	$\frac{\text{trace}(\boldsymbol{W}\boldsymbol{W}^H)}{2}$	$M = \sum_{f,i,p} \delta_{f,i}^{M,(p)}, \boldsymbol{W} = \text{vec}\left\{\underline{m}_{f,i}^{(p)}\right\}_{\left\{p,f,i:\delta_{f,i}^{M,(p)}=1\right\}}$
Calibration	$(\sigma^H)^2$	$a_\sigma + \frac{M}{2}$	$\frac{\text{trace}(\boldsymbol{W}\boldsymbol{W}^H)}{2}$	$M = KFNP/\lvert Z_g\rvert, \quad \boldsymbol{W} = \text{vec}\left\{z_{k,f,i}(g)\right\}_{k,f,i,g}$
Additive noise	$(\sigma^V)^2$	$a_\sigma + \frac{M}{2}$	$\frac{\text{trace}(\boldsymbol{W}\boldsymbol{W}^H)}{2}$	$M = KFNP, \quad \boldsymbol{W} = \text{vec}\left\{\frac{i_{k,f,i}^{(p)}}{h_{k,f,i}^{(p)}} - l_{k,f,i}^{(p)} - s_{k,f,i}^{(p)}\right\}_{k,f,i,p}$

Note: The set Q_j is defined as $Q_j \overset{\triangle}{=} \left\{p : c^{(p)} = j\right\}$

Indicator probabilities

In the basic model where the indicator Beta distribution parameters do not depend spatially or temporally, then the posterior indicator probabilities for

$$\delta \sim \text{Bernoulli}(\pi) \tag{5.108}$$

$$\pi \sim \text{Beta}(a_\pi, b_\pi) \tag{5.109}$$

are given by

$$\pi|\delta \sim \text{Beta}(a_\pi + \delta, b_\pi + (1 - \delta)) \tag{5.110}$$

Note that we can modify a_π and b_π as in Section 5.4.1. However, the posterior inference for the probabilities is identical by replacing a_π and b_π by their spatiotemporally varying version.

Object class probabilities

We use a Multinomial-Dirichlet conjugate pair to determine object class assignments, where the class probabilities $\underline{q}$ have a prior Dirichlet distribution with $c_j = 1/J$ for $j = 1, 2, \ldots, J$. Then, after observing the class assignments, we can calculate the number of pixels in any class

$$N_j = |Q_j| = \left|\left\{p : c^{(p)} = j\right\}\right| \tag{5.111}$$

Then the posterior distribution for the class probabilities is given by

$$\underline{q}|\left\{N_j\right\}_{j=1}^{J} \sim \text{Gamma}\left(\underline{q} + \underline{N}, 1\right)/J \tag{5.112}$$

where $[\underline{N}]_j = N_j$.

5.11 Appendix: Cramér Rao Lower Bound

5.11.1 Model

Assume we have phase history measurements for k antennas at each of l frequencies and m slow-time pulses. Note that a SAR image is formed with a non-linear mapping of the phase-history to the imaging domain. Range-Doppler images can be formed using a 2-dimensional FFT of the phase-history data, while other algorithms such as backprojection or polar format algorithm transform the phase-history data to the spatial (x, y) domain.

Previous bounds for estimation error by Fasih [41] in SAR images have considered IID noise over all pulses, frequencies, and antennas. In this section, we consider correlated noise across antennas as given in our SAR image model. In particular, for the l-th frequency and m-th pulse, let us assume we have phase history measurements:

$$g_{lm}^{k} = \mu_{lm}^{k} + \varepsilon_{lm}^{k}, \quad k = 1, \ldots, K \tag{5.113}$$

for

$$\mu_{lm}^{k} = G \exp\left\{j\left(\psi_{lm}^{k} + x(m\tau)h_{lm}^{x} + y(m\tau)h_{lm}^{y}\right)\right\} \tag{5.114}$$

$$\psi_{lm}^{k} = \psi_0 + (k-1)\phi, \tag{5.115}$$

$$x(m\tau) = x_0 + (m\tau)v_x, \tag{5.116}$$

$$y(m\tau) = y_0 + (m\tau)v_y, \tag{5.117}$$

where τ is the time between slow-time pulses and ε_{lm}^{k} is zero-mean complex-valued additive noise with circular normal distribution:

$$\left\{\varepsilon_{lm}^{k}\right\}_{k=1}^{K} \sim \mathcal{CN}\left(0, \Lambda_{lm}\right). \tag{5.118}$$

Note that ε_{lm}^{k} is representative of the background variation that can possibly include clutter. In particular, we assume that the covariance matrix is given by

$$\Lambda_{lm} = \lambda_{lm}^2 \boldsymbol{I}_{K\times K} + \kappa_{lm}^2 \mathbf{1}_K \mathbf{1}_K^T. \tag{5.119}$$

where $\boldsymbol{I}_{K\times K}$ is the $K \times K$ identity matrix and $\mathbf{1}_K$ is the all-ones vector of length K. In the no-clutter case, we have $\lambda_{lm}^2 = \sigma_v^2$ and $\kappa_{lm}^2 = 0$. In the clutter-case, we have

$$\lambda_{lm}^2 = \sigma_v^2 + (\sigma_{lm}^s)^2 + (\sigma_{lm}^x)^2 + \delta^{y,m}(\sigma_{lm}^y)^2 \tag{5.120}$$

$$\kappa_{lm}^2 = \rho_{lm}^s(\sigma_{lm}^s)^2 + \rho_{lm}^x(\sigma_{lm}^x)^2 + \delta^{y,m}\rho_{lm}^y(\sigma_{lm}^y)^2 \tag{5.121}$$

Let the parameter vector be defined as

$$\Theta = \{x_0, y_0, v_x, v_y, G, \psi_0, \phi\} \tag{5.122}$$

Then the Fisher information matrix is given by the elemnent-wise definition by Kay [57]:

$$[\text{FIM}]_{uv} = 2\sum_{l,m} \Re\left\{ \left(\frac{\partial \vec{u}_{lm}}{\partial \theta_u}\right)^H \Lambda_{lm}^{-1} \left(\frac{\partial \vec{u}_{lm}}{\partial \theta_v}\right) \right\} + \sum_{l,m} \text{trace}\left(\Lambda_{lm}^{-1} \frac{\partial \Lambda_{lm}}{\partial \theta_u} \Lambda_{lm}^{-1} \frac{\partial \Lambda_{lm}}{\partial \theta_v} \right) \tag{5.123}$$

where $\mu_{lm} = \left\{\mu_{lm}^k\right\}_{k=1}^K$.

5.11.2 Mean term

Let us first focus on the FIM contribution from the mean component:

$$2\sum_{l,m} \Re\left\{ \left(\frac{\partial \vec{\mu}_{lm}}{\partial \theta_u}\right)^H \Lambda_{lm}^{-1} \left(\frac{\partial \vec{\mu}_{lm}}{\partial \theta_v}\right) \right\} \tag{5.124}$$

The following table gives the partial derivatives for each parameter in Θ for

$$\alpha_{lm} = \exp\left\{j\left(\psi_0 + x(m\tau)h_{lm}^x + y(m\tau)h_{lm}^y\right)\right\} \tag{5.125}$$

Note that the last column of Table 5.13 shows that all components at the k-antenna share the same dependence on the parameter α_{lm} and a phase component

Table 5.13: Partial derivatives for FIM derivation

Parameter, θ_u	$\frac{\partial \mu_{lm}^k}{\partial \theta_u}$	$u_k \overset{\triangle}{=} \frac{\partial \mu_{lm}^k}{\partial \theta_u} / (\alpha_{lm} e^{j\phi(k-1)})$
x_0	$Gh_{lm}^x \alpha_{lm} e^{j\phi(k-1)}$	Gh_{lm}^x
y_0	$Gh_{lm}^y \alpha_{lm} e^{j\phi(k-1)}$	Gh_{lm}^y
v_x	$Gm\tau h_{lm}^x \alpha_{lm} e^{j\phi(k-1)}$	$Gm\tau h_{lm}^x$
v_y	$Gm\tau h_{lm}^y \alpha_{lm} e^{j\phi(k-1)}$	$Gm\tau h_{lm}^y$
G	$\alpha_{lm} e^{j\phi(k-1)}$	1
ψ_0	$G\alpha_{lm} e^{j\phi(k-1)}$	G
ϕ	$G\alpha_{lm}(k-1) e^{j\phi(k-1)}$	$G(k-1)$

$e^{j\phi(k-1)}$. Define the vectors

$$\vec{u} = \begin{bmatrix} u_1 \\ u_2 e^{j\phi} \\ \vdots \\ u_K e^{j\phi(K-1)} \end{bmatrix}, \vec{v} = \begin{bmatrix} v_1 \\ v_2 e^{j\phi} \\ \vdots \\ v_K e^{j\phi(K-1)} \end{bmatrix} \tag{5.126}$$

where u_k and v_k are given by the third column in Table 5.13 for partial derivatives with respect to θ_u and θ_v respectively. Then the summand in equation (5.124) can be written as

$$\begin{aligned} \Re\left\{ \left(\frac{\partial \vec{\mu}_{lm}}{\partial \theta_u}\right)^H \Lambda_{lm}^{-1} \left(\frac{\partial \vec{\mu}_{lm}}{\partial \theta_v}\right) \right\} &= \Re\left\{ (\alpha_{lm}\vec{u})^H \Lambda_{lm}^{-1} (\alpha_{lm}\vec{v}) \right\} \\ &= \Re\left\{ \alpha_{lm}^* \alpha_{lm} \vec{u}^H \Lambda_{lm}^{-1} \vec{v} \right\} \\ &= \Re\left\{ \vec{u}^H \Lambda_{lm}^{-1} \vec{v} \right\} \end{aligned} \tag{5.127}$$

where we note that $\alpha_{lm}^* \alpha_{lm} = 1$. Note that u and v do not depend on the target state (x_0, y_0, v_x, v_y). Thus, the mean contribution to the FIM will be independent of the target state. Furthermore, if Λ_{lm} does not depend on these parameters as in the clutter-free case, then the FIM and CRB will be independent of the target parameters; i.e., we should be able to find an algorithm that can track the target equally well regardless of its actual state.

To simplify the computation of the mean contribution to the FIM, consider the fact that

$$\Lambda_{lm}^{-1} = \left[\kappa_{lm}^2 \mathbf{1}_K \mathbf{1}_K^T + \lambda_{lm}^2 \boldsymbol{I}_{K\times K}\right]^{-1} \tag{5.128}$$

$$= \lambda_{lm}^{-2} \left(\boldsymbol{I}_{K\times K} - \frac{\kappa_{lm}^2}{K\kappa_{lm}^2 + \lambda_{lm}^2} \mathbf{1}_K \mathbf{1}_K^T \right) \tag{5.129}$$

$$= L_{lm}^0 \boldsymbol{I}_{K\times K} + L_{lm}^1 \mathbf{1}_K \mathbf{1}_K^T, \tag{5.130}$$

where

$$L_{lm}^0 = \lambda_{lm}^{-2}, \tag{5.131}$$

$$L_{lm}^1 = \frac{\kappa_{lm}^2}{K\kappa_{lm}^2 \lambda_{lm}^2 + \lambda_{lm}^4}. \tag{5.132}$$

Then equation (5.127) becomes

$$\begin{aligned}
\Re\left\{\vec{u}^H \Lambda_{lm}^{-1} \vec{v}\right\} &= \Re\left\{\vec{u}^H \left(L_{lm}^0 \boldsymbol{I}_{K\times K} + L_{lm}^1 \mathbf{1}_K \mathbf{1}_K^T\right) \vec{v}\right\} \\
&= L_{lm}^0 \sum_k u_k v_k + L_{lm}^1 \Re\left\{ \left(\sum_k u_k e^{-j\phi(k-1)} \right) \left(\sum_k v_k e^{j\phi(k-1)} \right) \right\} \\
&= L_{lm}^0 \sum_k u_k v_k + L_{lm}^1 \Re\left\{ \left(\sum_k \sum_n u_k v_n e^{j\phi(n-1-k+1)} \right) \right\} \\
&= L_{lm}^0 \sum_k u_k v_k + L_{lm}^1 \sum_k \sum_n u_k v_n \Re\left\{ e^{j\phi(n-k)} \right\} \\
&= L_{lm}^0 \sum_k u_k v_k + L_{lm}^1 \sum_k \sum_n u_k v_n \cos(\phi(n-k))
\end{aligned} \tag{5.133}$$

This equation simplies further when $u_k = u$ or $v_k = v$ as is the case for all parameters except ϕ. In particular, if $u_k = u$ and $v_k = v$, then:

$$\begin{aligned}
\Re\left\{\vec{u}^H \Lambda_{lm}^{-1} \vec{v}\right\} &= (uv) \left(K L_{lm}^0 + L_{lm}^1 \sum_k \sum_n \cos(\phi(n-k)) \right) \\
&= (uv) \left(K L_{lm}^0 + \alpha_\phi L_{lm}^1 \right),
\end{aligned} \tag{5.134}$$

where $\alpha_\phi = \sum_{k,n} \cos(\phi(n-k))$. In the case where $\theta_u \neq \phi$ and $\theta_v = \phi$, we have

$$\begin{aligned}\Re\left\{\vec{u}^H \Lambda_{lm}^{-1} \vec{v}\right\} &= L_{lm}^0 \left(Gu \sum_k (k-1) \right) + L_{lm}^1 \left(Gu \sum_k \sum_n (n-1)\cos(\phi(n-k)) \right) \\ &= Gu \left(L_{lm}^0 \frac{K(K-1)}{2} + L_{lm}^1 \sum_k \sum_n (n-1)\cos(\phi(n-k)) \right) \\ &= Gu \left(L_{lm}^0 \frac{K(K-1)}{2} + L_{lm}^1 (\alpha_\phi^n - \alpha_\phi) \right),\end{aligned} \tag{5.135}$$

where

$$\alpha_\phi^n = \sum_k \sum_n n \cos(\phi(n-k)) \tag{5.136}$$

Finally, in the case where $\theta_u = \theta_v = \phi$, we have

$$\begin{aligned}\Re\left\{\vec{u}^H \Lambda_{lm}^{-1} \vec{v}\right\} &= L_{lm}^0 \left(G^2 \sum_k (k-1)^2 \right) \\ &\quad + L_{lm}^1 \left(G^2 \sum_k \sum_n (k-1)(n-1)\cos(\phi(n-k)) \right) \\ &= G^2 \left(L_{lm}^0 \frac{K(K-1)(2K-1)}{6} + L_{lm}^1 (\alpha_\phi^{kn} - 2\alpha_\phi^n + \alpha_\phi) \right),\end{aligned} \tag{5.137}$$

where

$$\alpha_\phi^{kn} = \sum_k \sum_n kn \cos(\phi(n-k)) \tag{5.138}$$

In all cases, the summand can be represented as

$$\Re\left\{\vec{u}^H \Lambda_{lm}^{-1} \vec{v}\right\} = \beta_{uv}^0 L_{lm}^0 + \beta_{uv}^0 L_{lm}^1, \tag{5.139}$$

where the parameters $\beta_{uv}^0, \beta_{uv}^1$ are given in Table 5.14.

5.11.3 Covariance term

We now consider the FIM contribution from the covariance component:

$$\operatorname{trace}\left(\Lambda_{lm}^{-1} \frac{\partial \Lambda_{lm}}{\partial \theta_u} \Lambda_{lm}^{-1} \frac{\partial \Lambda_{lm}}{\partial \theta_v} \right) \tag{5.140}$$

Table 5.14: $\beta^0_{uv}, \beta^1_{uv}$ parameters

$\theta_u \in$	$\theta_v \in$	β^0_{uv}	β^1_{uv}
$\left\{\begin{matrix} x_0, y_0, v_x, \\ v_y, G, \psi_0 \end{matrix}\right\}$	$\left\{\begin{matrix} x_0, y_0, v_x, \\ v_y, G, \psi_0 \end{matrix}\right\}$	uvK	$uv\alpha_\phi$
$\left\{\begin{matrix} x_0, y_0, v_x, \\ v_y, G, \psi_0 \end{matrix}\right\}$	$\{\phi\}$	$\frac{GuK(K-1)}{2}$	$Gu(\alpha^n_\phi - \alpha_\phi)$
$\{\phi\}$	$\left\{\begin{matrix} x_0, y_0, v_x, \\ v_y, G, \psi_0 \end{matrix}\right\}$	$\frac{GvK(K-1)}{2}$	$Gv(\alpha^n_\phi - \alpha_\phi)$
$\{\phi\}$	$\{\phi\}$	$\frac{G^2K(K-1)(2K-1)}{6}$	$G^2(\alpha^{kn}_\phi - 2\alpha^n_\phi + \alpha_\phi)$

In general, it may difficult to determine the effect of the target state parameters on the noise covariance matrix. In this section, we suppose that within the small regions where the target signature will appear over a CPI, we can approximate this dependence with a linear map from spatial coordinates to the parameters ρ, σ^2 of our covariance matrix. In particular

$$\begin{aligned} \kappa^2(x,y) &= \kappa_0^2 + x\gamma_x^\kappa + y\gamma_y^\kappa, \\ \lambda^2(x,y) &= \lambda_0^2 + x\gamma_x^\lambda + y\gamma_y^\lambda. \end{aligned} \tag{5.141}$$

for parameters $(\kappa_0^2, \lambda_0^2, \gamma_x^\kappa, \gamma_y^\kappa, \gamma_x^\lambda, \gamma_y^\lambda,)$ that can be fitted from our SAR image model. In this case, we have

$$\begin{aligned} \frac{\partial \Lambda_{lm}}{\partial \theta_u} &= \frac{\partial}{\partial \theta_u}\left\{\kappa_{lm}^2 \boldsymbol{I}_{K\times K} + \lambda_{lm}^2 \mathbf{1}_K \mathbf{1}_K^T\right\} \\ &= \frac{\partial \kappa_{lm}^2}{\partial \theta_u}\boldsymbol{I}_{K\times K} + \frac{\partial \lambda_{lm}^2}{\partial \theta_u}\mathbf{1}_K\mathbf{1}_K^T, \end{aligned} \tag{5.142}$$

For example, when $\theta_u = x_0$, we have

$$\frac{\partial \kappa_{lm}^2}{\partial x_0} = \gamma_x^\kappa,\ \ \frac{\partial \lambda_{lm}^2}{\partial x_0} = \gamma_x^\lambda. \tag{5.143}$$

Then we have

$$\begin{aligned} \Lambda_{lm}^{-1}\frac{\partial \Lambda_{lm}}{\partial \theta_u} &= \left(L_{lm}^0 \boldsymbol{I}_{K\times K} + L_{lm}^1 \mathbf{1}_K \mathbf{1}_K^T\right)\left(\frac{\partial \kappa_{lm}^2}{\partial \theta_u}\boldsymbol{I}_{K\times K} + \frac{\partial \lambda_{lm}^2}{\partial \theta_u}\mathbf{1}_K\mathbf{1}_K^T\right), \\ &= \tilde{\beta}_u^{\mathrm{I}} \boldsymbol{I}_{K\times K} + \tilde{\beta}_u^{11}\mathbf{1}_K\mathbf{1}_K^T, \end{aligned} \tag{5.144}$$

where

$$\tilde{\beta}_u^{\mathrm{I}} = L_{lm}^0 \frac{\partial \kappa_{lm}^2}{\partial \theta_u}, \tag{5.145}$$

$$\tilde{\beta}_u^{11} = L_{lm}^1 \frac{\partial \kappa_{lm}^2}{\partial \theta_u} + \left(L_{lm}^0 + K L_{lm}^1\right) \frac{\partial \lambda_{lm}^2}{\partial \theta_u}. \tag{5.146}$$

Finally, we have

$$\begin{aligned}
&\operatorname{trace}\left(\Lambda_{lm}^{-1} \frac{\partial \Lambda_{lm}}{\partial \theta_u} \Lambda_{lm}^{-1} \frac{\partial \Lambda_{lm}}{\partial \theta_v}\right) \\
&\quad = \operatorname{trace}\left\{\left(\tilde{\beta}_u^{\mathrm{I}} \boldsymbol{I}_{K\times K} + \tilde{\beta}_u^{11} \mathbf{1}_K \mathbf{1}_K^T\right)\left(\tilde{\beta}_v^{\mathrm{I}} \boldsymbol{I}_{K\times K} + \tilde{\beta}_v^{11} \mathbf{1}_K \mathbf{1}_K^T\right)\right\} \\
&\quad = \operatorname{trace}\left\{\tilde{\beta}_u^{\mathrm{I}} \tilde{\beta}_v^{\mathrm{I}} \boldsymbol{I}_{K\times K} + \left(\tilde{\beta}_u^{\mathrm{I}} \tilde{\beta}_v^{11} + \tilde{\beta}_v^{\mathrm{I}} \tilde{\beta}_u^{11} + K \tilde{\beta}_u^{11} \tilde{\beta}_v^{11}\right) \mathbf{1}_K \mathbf{1}_K^T\right\} \\
&\quad = K\left(\tilde{\beta}_u^{\mathrm{I}} \tilde{\beta}_v^{\mathrm{I}} + \tilde{\beta}_u^{\mathrm{I}} \tilde{\beta}_v^{11} + \tilde{\beta}_v^{\mathrm{I}} \tilde{\beta}_u^{11} + K \tilde{\beta}_u^{11} \tilde{\beta}_v^{11}\right)
\end{aligned} \tag{5.147}$$

This equation may simplify in some cases, though we won't give all details here. However, it should be noted that the original derivatives $\frac{\partial \sigma_{lm}^2}{\partial \theta_u}$ and $\frac{\partial \rho}{\partial \theta_u}$ are non-zero only for $\theta_u \in \{x_0, y_0, v_x, v_y\}$. Moreover, if either θ_u or θ_v does not belong to this set, then our trace will equal zero. Thus, we only have to compute this term for a subset of the elements of the FIM.

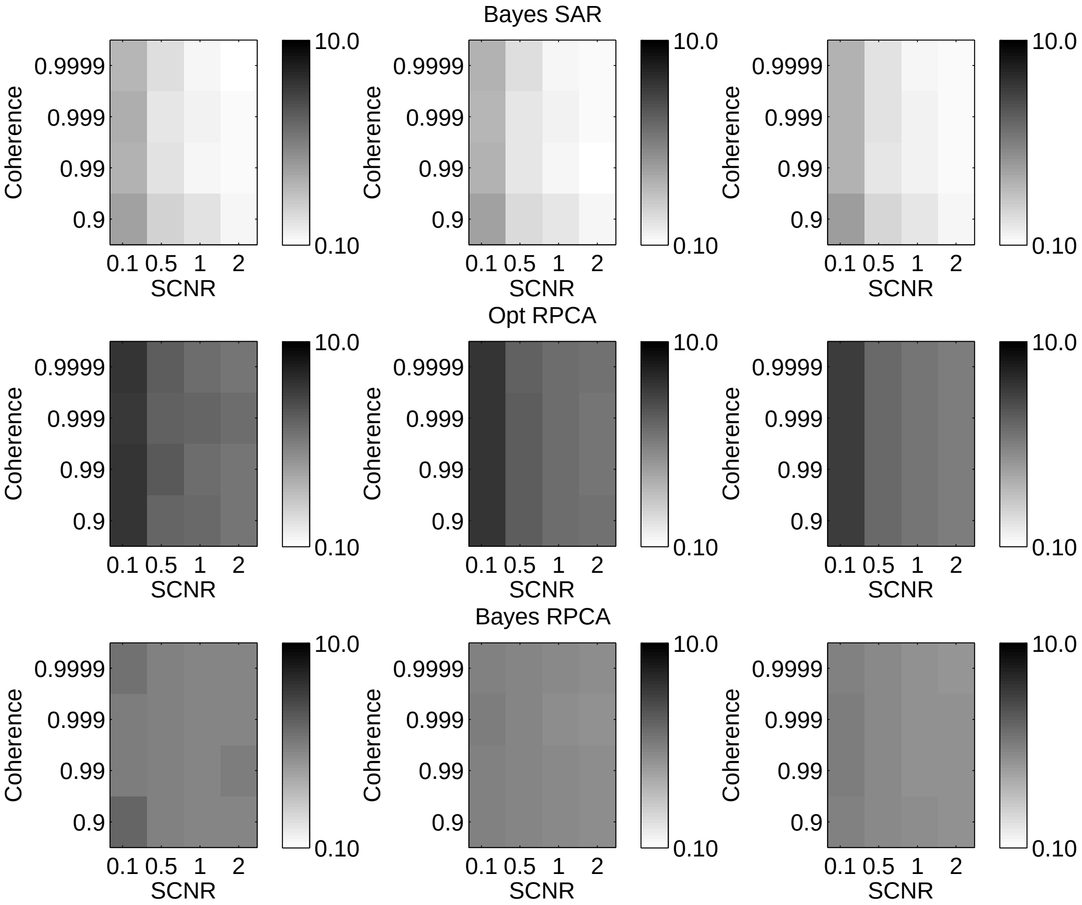
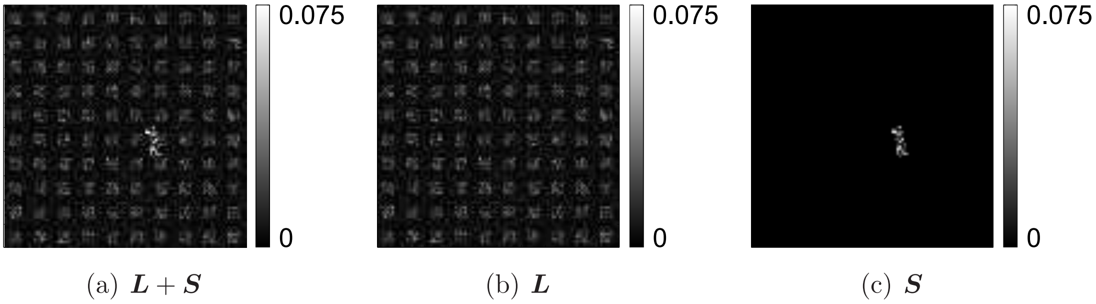
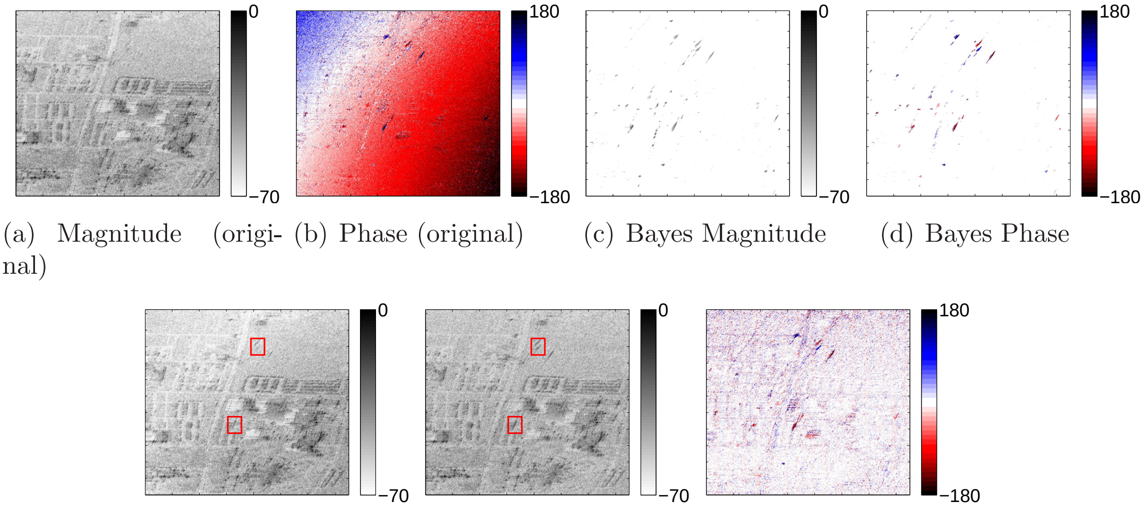


Figure 5.3: This figure compares the relative reconstruction error of the target component, $\frac{\|\boldsymbol{S}-\hat{\boldsymbol{S}}\|_2}{\|\boldsymbol{S}\|_2}$, as a function of algorithm, number of passes N, coherence of antennas ρ, and signal-to-clutter-plus-noise ratio (SCNR). From top-to-bottom, the rows contains the output of the Bayes SAR algorithm, the optimization-based RPCA algorithm, and the Bayes RPCA algorithm. From left-to-right, the columns show the output for $N = 5$, $N = 10$, and $N = 20$ passes (with $F = 1$ frames per pass). The output is given by the median error over 20 trials on a simulated dataset. It is seen that in all cases, the Bayes SAR method outperforms the RPCA algorithms. Moreover, the Bayes SAR algorithm performs better if either coherence increases (i.e., better clutter cancellation) or the SCNR increases. On the other hand, the performance of the RPCA algorithms does not improve with increased coherence, since these algorithms do not directly model this relationship.



(a) $\boldsymbol{L} + \boldsymbol{S}$ (b) $\boldsymbol{L}$ (c) $\boldsymbol{S}$

Figure 5.4: This figure provides a sample image used in the simulated dataset for comparisons to RPCA methods, as well as its decomposition into low-dimensional background and sparse target components. This low SCNR image is typical of measured SAR images. Note that the target is randomly placed within the image for each of N passes. In some of these passes, the target is placed over low-amplitude clutter and can be easily detected. In other passes, the target is placed over high-amplitude clutter, which reduces the capability to detect the target.



(a) Magnitude (original)
(b) Phase (original)
(c) Bayes Magnitude
(d) Bayes Phase
(e) DPCA (original)
(f) DPCA (calibrated)
(g) Phase (calibrated)

Figure 5.5: This figure compares the output of the proposed algorithm as a function of magnitude and phase for a scene of size 375m by 1200m and coherent processing interval of 0.5s. The Bayes SAR algorithm takes the original SAR images in (a) and (b), estimates the nuisance parameters such as antenna miscalibrations and clutter covariances, and yields a sparse output for the target component in (c) and (d). In contrast, the DPCA and ATI algorithms are very sensitive to the nuisance parameters, which make finding detection thresholds difficult. In particular, consider the original interferometric phase image shown in (b). It can be seen that without proper calibration between antennas, there is strong spatially-varying antenna gain pattern that makes cancellation of clutter difficult. Calibration is generally not a trivial process, but to make fair comparisons to the DPCA/ATI algorithms, calibration in (f) and (g) is done by using the estimated coefficients $\boldsymbol{H}_{f,i}$ from the Bayes SAR algorithm. In (e) and (f), the outputs of the DPCA algorithm are applied to the original images (all antennas) and the calibrated images (all antennas), respectively. It should be noted that even with calibration, the DPCA outputs contain a huge number of false detections in high clutter regions. Nevertheless, proper calibration enables detection of moving targets that are not easily detected without calibration, as highlighted by the red boxes. Note that the Bayes SAR algorithm provides an output that is sparse, yet does not require tuning of thresholds as required by DPCA and/or ATI.

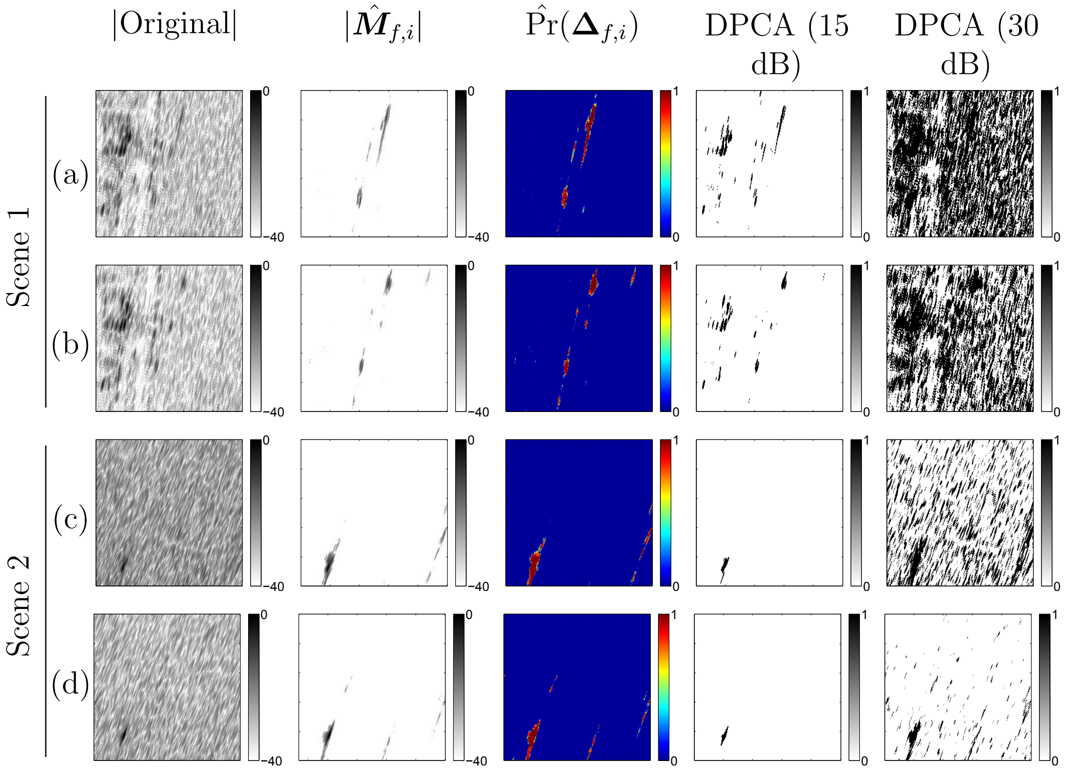


Figure 5.6: This figure shows detection performance based on the magnitude of the target response with comparisons between the proposed Bayes SAR algorithm and displaced phase center array (DPCA) processing. Note that DCPA declares a detection if the relative magnitude to the brightest pixel is greater than some threshold. Results are given for two scenes of size 125m x 125m; within each scene, images were formed for two sequential 0.5 second intervals. Scene 1 contains strong clutter in the upper left region, while Scene 2 has relatively little clutter. The columns of the figure provide from left-to-right: the magnitude of the original image, the estimated target component from the proposed algorithm, the probability of the target occupying a particular pixel, the output of DPCA with a relative threshold of 15 dB, and the output of DPCA with a relative threshold of 30 dB. It is seen that DPCA has difficulty in canceling the clutter in Scene 1 with either threshold. Moreover, in Scene 2 (c-d) DPCA misses detections of the low-magnitude target in the lower right for the 15 dB threshold. In both scenes, there are many false alarms at the 30 dB threshold. On the other hand, the proposed algorithm provides a sparse solution that detects all of these targets, while simultaneously providing a estimate of the probability of detection rather than an indicator output.

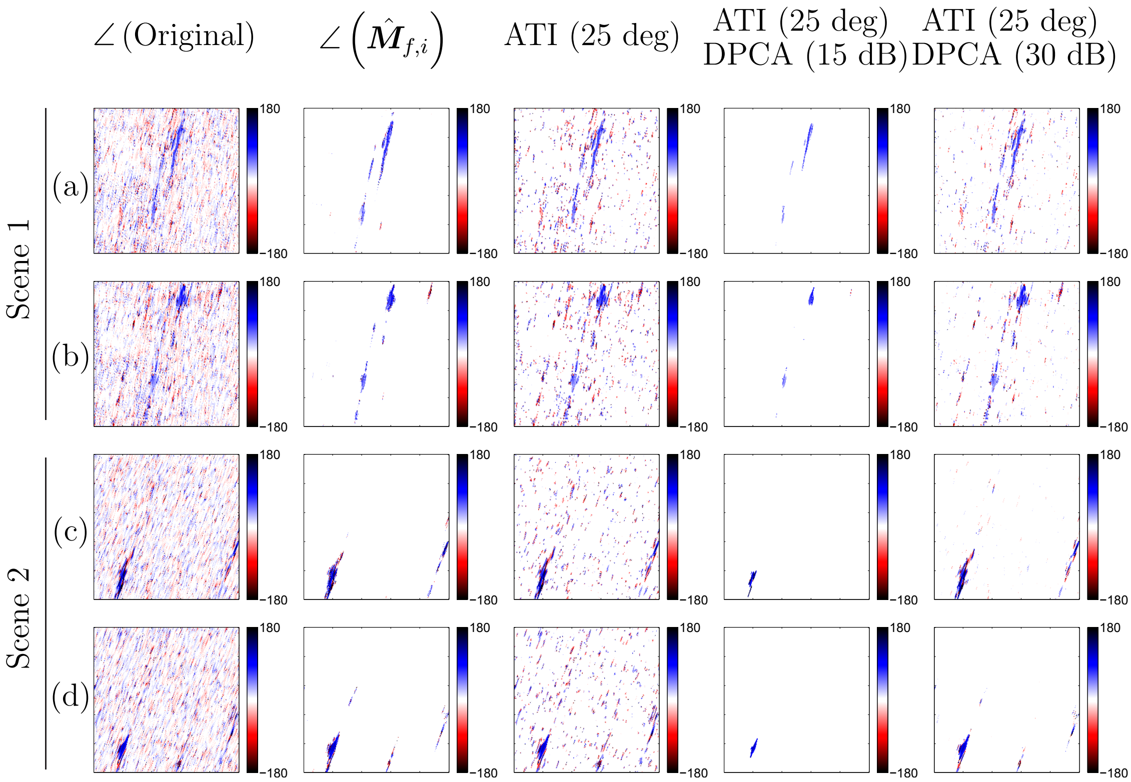
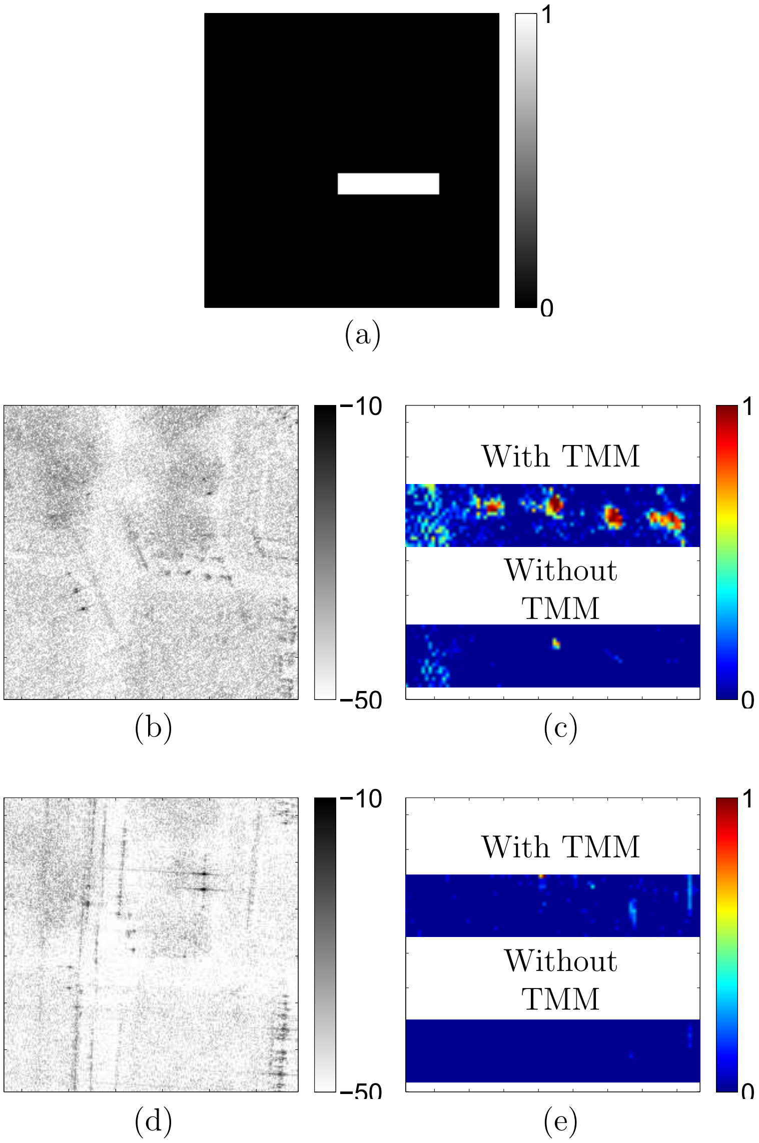


Figure 5.7: This figure shows detection performance based on the phase of the target response with comparisons between the proposed algorithm, along-track interferometry (ATI) and a mixture algorithm between ATI/DPCA. Results are given for the same two scenes in Figure 5.6. In all cases, we show results for calibrated imagery where $\boldsymbol{H}_{f,i}$ are given by the output of the Bayes SAR algorithm, though this step is not trivial. The columns of the figure provide from left-to-right: the phase of the image without thresholding, the estimated target phase component from the proposed algorithm, the output of ATI with a threshold of 25 degrees, the output of ATI/DPCA with (25 deg, 15 dB) thresholds, and the output of ATI/DPCA with (25 deg, 30 dB) thresholds. In contrast to Figure 5.6, the contributions from the strong clutter are not very strong, though there are still numerous false alarms in the ATI and ATI/DPCA outputs. It is seen that the ATI/DPCA combination with 15 dB magnitude threshold over-sparsifies the solution, missing targets in (b), (c), and (d). On the other hand, the ATI/DPCA combination with 30 dB magnitude threshold detects these targets, but also includes false alarms in (a) and (b).



(a)

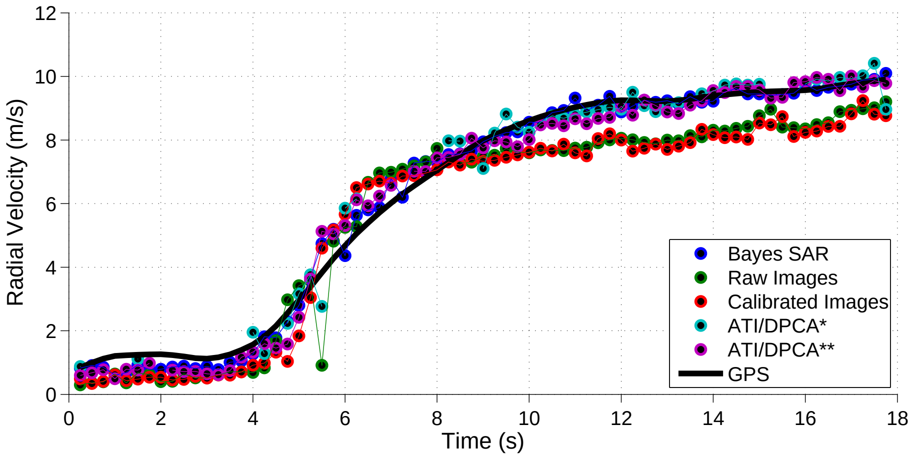
(b)

(c)

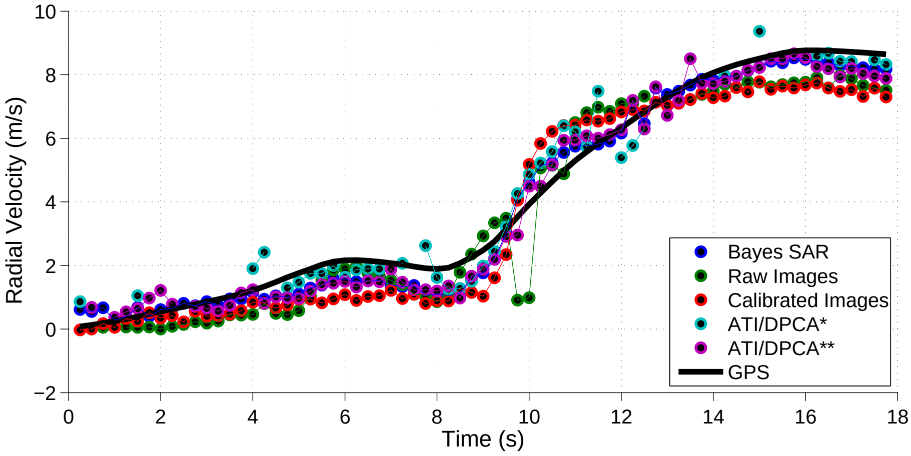
(d)

(e)

Figure 5.8: This figure compares the performance of our proposed method with and without priors on target signature locations. In this scene, targets are likely to be stopped at an intersection as shown by the region in (a). A mission image containing targets is shown in (b) and a reference image without targets is shown in (d). The estimated target probabilities are shown in (c) for the mission scene where inference was done both with/without a target motion model (TMM). It can be seen that by including the prior information, we are able to detect stationary targets that cannot be detected from standard SAR moving target indication algorithms. The estimated target probabilities in the reference scene are shown in (e), showing little performance differences when prior information is included in the inference.

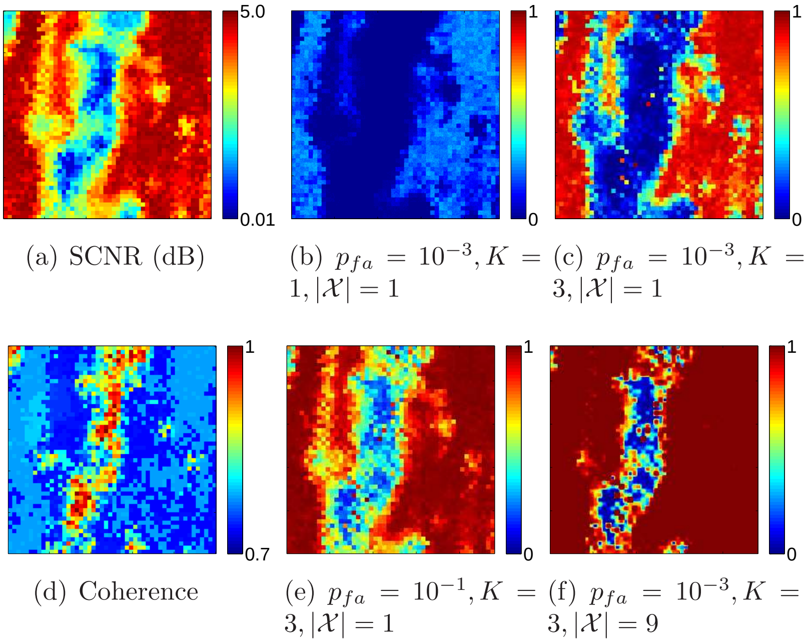


(a) Target 1



(b) Target 2

Figure 5.9: This figure plots the estimated radial velocities (m/s) for two targets from measured SAR imagery over 18 seconds at 0.25 second increments. Radial velocity, which is proportional to the interferometric phase of the pixels from multiple antennas in an along-track SAR system, is estimated by computing the average phase of pixels within a region specified by the GPS-given target state (position, velocity). We compare the estimation of radial velocity from the output of the Bayes SAR algorithm, from the raw images, from the calibrated images (i.e, using the estimated calibration coefficients), and from two DPCA/ATI joint algorithms with phase/magnitude thresholds of (25 deg, 15 dB) and (25 deg, 30 dB) respectively. For best comparisons, the DPCA/ATI thresholds are applied to the calibrated imagery, though this is a non-trivial step in general. The black line provides the GPS provided radial velocities. Numerical results are summarized in Table 5.9. It is seen that the Bayes SAR algorithm outperforms the others in terms of MSE for both targets. Moreover, the Bayes SAR algorithm never misses a target detection in this dataset, which is not the case for the DPCA/ATI algorithms.



(a) SCNR (dB)

(b) $p_{fa} = 10^{-3}, K = 1, |\mathcal{X}| = 1$

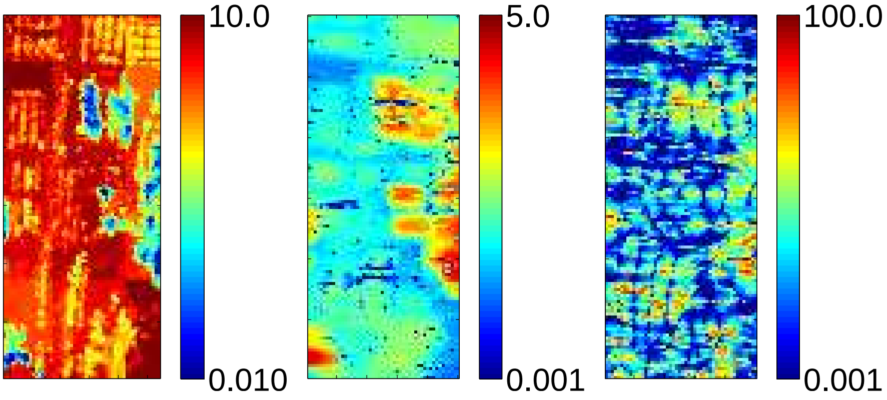
(c) $p_{fa} = 10^{-3}, K = 3, |\mathcal{X}| = 1$

(d) Coherence

(e) $p_{fa} = 10^{-1}, K = 3, |\mathcal{X}| = 1$

(f) $p_{fa} = 10^{-3}, K = 3, |\mathcal{X}| = 9$

Figure 5.10: This figure shows an example of using the output of the Bayes SAR algorithm in order to derive detection algorithms for future performance prediction. In (a) and (d), the estimated signal-to-clutter-plus-noise ratio (SCNR) and coherence are provided for a scene of size 125m by 125m. Detection probabilities are given in (b), (c), (e), and (f) for various values of false alarm probability, number of antennas K, and number of independent pixels useed in the LRT. It is seen that detection performance is improved by increasing either K or $|\mathcal{X}|$.



(a) SCNR (dB) (b) $x-$spatial error (m) (c) $y-$spatial error (m)

Figure 5.11: This figure provides an example of lower bounds on spatial errors derived from the output of the Bayes SAR algorithm. Results are shown for a scene of size 375m by 1200m and coherent processing interval (CPI) of 0.5s. In this specific scene the radar was nearly aligned with the $x-$axis. Thus, the lower bounds reflect the fact that it is easier to locate targets in the radial dimension as shown in (b), compared with the azimuthal dimension as shown in (c). Note that this would be alleviated for longer CPIs.

CHAPTER VI

Conclusions and Future Work

In this thesis, adaptive sensing and sensor management was studied in the context of detecting and estimating moving targets using limited resources. This thesis studied adaptive sensing and sensor management from three main directions: (a) development of a framework for adaptive allocation of limited resources in order to detect and estimate moving targets, (b) derivation of bounds on fundamental performance limits for stable tracking of multiple targets, and (c) application of adaptive sensor management to the specific application of detection/tracking with synthetic aperture radar (SAR) imagery.

This thesis provided a general framework for adaptive search for targets that exhibit dynamic behavior such as moving targets, target birth/death, and varying target amplitudes. A cost function was provided that generalizes well to many target and state models, and oracle allocations were derived that provide bounds on achievable performance. A non-myopic policy was provided that can be found through nested optimization that grows as $O(T^2)$, where T is the number of stages. In contrast, a heuristic policy based on the idea that resources should be saved for future exploitation was also provided with complexity $O(T)$. Finally, a functional approximation to the heuristic policy was given with complexity $O(1)$. All of these policies were examined through empirical analysis, showing excellent performance in many cases, including asymptotic consistency, significant performance gains over an exhaustive search alternative, and increased robustness to model mismatch of myopic policies.

Future work in this area includes consideration of constraints on the number of measurements, which may include coarse-scale or compressed sensing measurements. Moreover, we are interested in deriving analytical results such as convergence rates (in comparison to exhaustive search) and/or minimum detectable amplitudes. We would also like to consider online policies that are computed as measurements are observed - this could improve performance dramatically in many cases, including cases where targets will be obscured in the near future. Moreover, we are interested in studying cases in which the number of stages T is a random variable, so that no additional measurements of the scene are required once 'sufficient' signal quality has been reached, at least in the case of static targets.

In the next of the direction of this thesis, a conservative approach to sensor management was proposed for multiple target tracking subject to computational constraints. The approach requires finding solutions to load balance equations that guarantee system stability. These solutions yield the minimal system requirements for provisioning radars. The solutions guarantee stable tracking with prescribed level of statistical confidence. The provisioning results given here are conservative and specify the system requirements, steady state occupancy, revisit times, and track entropy in terms of the PLQ sensor scheduling policy. The PLQ policy will always perform at least as well as the performance predictions we provide. One can expect considerably better performance of the system than these predictions for typical scenarios, although there exists a scenario (namely, all targets are equally difficult to track) where the predictions are exact. Less stringent provisioning requirements might be explored using a stochastic optimization.

Future work will consider and compare policies other than the PLQ policy. Of particular interest are policies using a random allocation that can be used as a baseline comparison, as well as policies that may have a class-dependent allocation. Additionally, we may consider optimizing a scheduling policy (in terms of τ^*) among multiple alternative policies. Furthermore, in this thesis we have considered tasks in the load margin separately from the target tracking task. However, if we consider multiple epochs, tasks such as Kalman filtering and/or target classification may significantly improve the capability to track targets stably over time. Future work plans to develop a framework for scheduling in the multiple-epoch scenario and analyze the tradeoffs in load margin vs. non-myopic improvements to our performance bounds.

In the last direction of this thesis, research in decomposing high-dimensional signals/images into low-rank and sparse components in the presence of noise by Wright et al. [93], Lin et al. [63] and Candes et al. [19] was extended to the case of separating target signatures from a low-dimensional clutter subspace in SAR imagery. In particular, we combine our understanding of the physical, kinematic, and statistical properties of SAR imagery into a single unified Bayesian structure that simultaneously (a) estimates the nuisance parameters such as clutter distributions and antenna miscalibrations and (b) extracts a sparse component containing the target signatures required for detection and estimation of the target state. Similar to work by Ding et al. [33], this algorithm requires few tuning parameters since most quantities of interest are inferred directly from the data - this allows the algorithm to be robust to a large collection of operating conditions.

The performance of the proposed approach is analyzed over both simulated and measured datasets, demonstrating competing or better performance than the robust PCA algorithms and ATI/DPCA.

Moreover, it is shown that the outputs of the Bayesian inference can be used for future performance prediction through examples of derived likelihood ratio tests and Cramér-Rao Lower Bounds for spatial errors.

Other work will include the development of algorithms that exploit the use of a posterior distributions for improved performance in a signal processing task, e.g. detection, tracking or classification. In particular, we are interested in using algorithms for simultaneously detecting and estimating targets over a sparse scene with resource constraints, similar to work by Bashan et al. [10,11], as well as determining the fundamental performance limits of a SAR target tracking system. Furthermore, we would also like to consider other generalizations to the SAR image model, such as complex target maneuvers, multiple target classes, and explicit tracking of the target phase.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] A. A. Abdel-Samad and A. H. Tewfik. Search strategies for radar target localization. In *Proceedings of the 1999 International Conference on Image Processing*, volume 3, pages 862–866, October 1999.

[2] A. A. Abdel-Samad and A. H. Tewfik. Hierarchical radar target localization. In *Proceedings of the 2000 International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3033–3036, June 2000.

[3] A. A. Abdel-Samad and A. H. Tewfik. Sequential techniques in hierarchical radar target localization. In *Proceedings of the 2000 International Conference on Image Processing*, volume 1, pages 697–700, September 2000.

[4] M. Aharon, M. Elad, and A. Bruckstein. k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006.

[5] M. Armony and N. Bambos. Queueing dynamics and maximal throughput scheduling in switched processing systems. *Queueing systems*, 44(3):209–252, 2003.

[6] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[7] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the Restricted Isometry Property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[8] R. Baraniuk and P. Steeghs. Compressive radar imaging. In *Proceedings of the 2007 IEEE Radar Conference*, pages 128–133, 2007.

[9] E. Bashan. *Efficient resource allocation schemes for search.* PhD thesis, Univ. of Michigan, Ann Arbor, May 2008.

[10] E. Bashan, G. Newstadt, and A. O. Hero III. Two-stage multi-scale search for sparse targets. *IEEE Transactions on Signal Processing*, 59(5):2331–2341, 2011.

[11] E. Bashan, R. Raich, and A. O. Hero III. Optimal two-stage search for sparse targets using convex criteria. *IEEE Transactions on Signal Processing*, 56:5389–5402, 2008.

[12] E. Bashan, R. Raich, and A. O. Hero III. Adaptive sampling: efficient search schemes under resource constraints. Technical Report 385, University of Michigan, Communications and Signal Processing Lab., October 2007.

[13] Y. Bazi, L. Bruzzone, and F. Melgani. An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):874–887, 2005.

[14] D.P. Bertsekas. Dynamic programming and optimal control, 3rd edition, volume II. 2011.

[15] L. Borcea, T. Callaghan, and G. Papanicolaou. Synthetic aperture radar imaging and motion estimation via robust principle component analysis. *arXiv preprint arXiv:1208.3700*, 2012.

[16] B. H. Borden and M. L. Mumford. A statistical glint/radar cross section target model. *IEEE Transactions on Aerospace and Electronic Systems*, (5):781–785, 1983.

[17] F. Bovolo and L. Bruzzone. A detail-preserving scale-driven approach to change detection in multitemporal SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 43(12):2963–2972, 2005.

[18] P. Brémaud. *Point processes and queues, martingale dynamics.* Springer, 1981.

[19] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011.

[20] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[21] E. J. Candes and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, December 2007.

[22] W. G. Carrara, R. S. Goodman, and R. M. Majewski. Spotlight synthetic aperture radar- signal processing algorithms(book). *Norwood, MA: Artech House*, 1995.

[23] R. Castro, J. Haupt, and R. Nowak. Compressed sensing vs. active learning. In *Proceedings of the 2006 International Conference on Acoustics, Speech and Signal Processing*, volume 3, May 2006.

[24] R. Castro, R. Willet, and R. Nowak. Coarse-to-fine manifold learning. In *Proceedings of the 2004 International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages iii–992–5, May 2004.

[25] R. Castro, R. Willett, and R. Nowak. Faster rates in regression via active learning. In *Proceedings of the Neural Information Processing Systems Conference (NIPS) 2005*, Vancouver, Canada, December 2005.

[26] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1999.

[27] Y. Chen, A. Wiesel, and A.O. Hero. Robust shrinkage estimation of high-dimensional covariance matrices. *Signal Processing, IEEE Transactions on*, 59(9):4097–4107, 2011.

[28] E. K. P. Chong, C. M. Kreucher, and A. O. Hero III. Monte-carlo-based partially observable markov decision process approximations for adaptive sensing. In *9th International Workshop on Discrete Event Systems*, pages 173–180. IEEE, 2008.

[29] P. Cornwall. Synthetic aperture radar lecture notes, 2004.

[30] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive approximation*, 13(1):57–98, 1997.

[31] R. W. Deming. Along-track interferometry for simultaneous SAR and GMTI: application to Gotcha challenge data. In *Proceedings of SPIE*, volume 8051, page 80510P, 2011.

[32] R.W. Deming, S. MacIntosh, and M. Best. Three-channel processing for improved geo-location performance in SAR-based GMTI interferometry. In *Proceedings of SPIE*, volume 8394, page 83940F, 2012.

[33] X. Ding, L. He, and L. Carin. Bayesian robust principal component analysis. *IEEE Transactions on Image Processing*, 20(12):3419–3430, 2011.

[34] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[35] R. Dorfman. The detection of defective members of large populations. *The Annals of Mathematical Statistics*, 14(4):436–440, December 1943.

[36] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[37] C. Drew. Military is awash in data from drones. *The New York Times*, January 10, 2010.

[38] J. H. G. Ender. Space-time processing for multichannel synthetic aperture radar. *Electronics & Communication Engineering Journal*, 11(1):29–38, 1999.

[39] E. Erten, A. Reigber, and O. Hellwich. Aspects of multivariate statistical theorywith the application to change detection. In *Proceedings of the 2010 IEEE International on Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1960 –1963, 2010.

[40] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1794–1803. IEEE, 2005.

[41] A. R. Fasih, C. W. Rossler, J. N. Ash, and R. L. Moses. Analysis of motion disambiguation using multi-channel circular SAR. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7699, page 17, 2010.

[42] J. R. Fienup. Detecting moving targets in SAR imagery by focusing. *IEEE Transactions on Aerospace and Electronic Systems*, 37(3):794–809, 2001.

[43] C. H. Gierull. Statistical analysis of multilook SAR interferograms for CFAR detection of ground moving targets. *IEEE Transactions on Geoscience and Remote Sensing*, 42(4):691–701, 2004.

[44] LeRoy A. Gorham and Linda J. Moore. SAR image formation toolbox for MATLAB. volume 7699. SPIE, 2010.

[45] I.F. Gorodnitsky and B.D. Rao. Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, March 1997.

[46] B. Guo, D. Vu, L. Xu, M. Xue, and J. Li. Ground moving target indication via multichannel airborne SAR. *IEEE Transactions on Geoscience and Remote Sensing*, 49(10):3753 –3764, Oct. 2011.

[47] J. Guo, Z. F. Li, and Z. Bao. Adaptive clutter suppression and resolving of velocity ambiguities for an experimental three-channel airborne synthetic aperture radar-ground moving target indication system. *Radar, Sonar & Navigation, IET*, 5(4):426–435, 2011.

[48] J. Haupt, R. Castro, and R. Nowak. Distilled sensing: adaptive sampling for sparse detection and estimation. *Arxiv preprint arXiv:1001.5311*, 2010.

[49] J. Haupt and R. Nowak. Signal reconstruction from noisy random projections. *IEEE Transactions on Information Theory*, 52(9):4036–4048, 2006.

[50] J. D. Haupt, R. G. Baraniuk, R. M. Castro, and R. D. Nowak. Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements. In *the 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pages 1551–1555. IEEE, 2009.

[51] A. O. Hero III. Sensor management provisioning for multiple target radar tracking systems. Technical Report 407, Univ. of Michigan, CSPL, 2008.

[52] A. O. Hero III, D. A. Castañón, D. Cochran, and K. Kastella, editors. *Foundations and Applications of Sensor Management.* Springer, Boston, MA, 2007.

[53] C.V. Jakowatz, D.E. Wahl, P.H. Eichel, D.C. Ghiglia, and P.A. Thompson. *Spotlight-mode synthetic aperture radar: a signal processing approach*, volume 101. Kluwer Academic Publishers, 1996.

[54] J. K. Jao. Theory of synthetic aperture radar imaging of a moving target. *IEEE Transactions on Geoscience and Remote Sensing*, 39(9):1984–1992, 2001.

[55] S. Ji, Y. Xue, and L. Carin. Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356, June 2008.

[56] K. Kastella. Discrimination gain to optimize detection and classification. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 27(1):112–116, January 1997.

[57] S. M. Kay. *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory.* Prentice-Hall PTR, January 1998.

[58] A. S. Khwaja and J. Ma. Applications of compressed sensing for SAR moving-target velocity estimation and image compression. *IEEE Transactions on Instrumentation and Measurement*, 60(8):2848–2860, 2011.

[59] C. Kreucher, K. Kastella, and A. O. Hero III. Multitarget tracking using the joint multitarget probability density. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1396–1414, October 2005.

[60] C. Kreucher, K. Kastella, and A. O. Hero III. Sensor management using an active sensing approach. *Signal Processing*, 85(3):607–624, March 2005.

[61] V. Krishnamurthy. Algorithms for optimal scheduling and management of hidden Markov model sensors. *IEEE Transactions on Signal Processing*, 50(6):1382–1397, June 2002.

[62] V. Krishnamurthy and R. J. Evans. Hidden Markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking. *IEEE Transactions on Signal Processing*, 49(12):2893–2908, December 2001.

[63] Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[64] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[65] D. Malioutov, M. Cetin, and A. S. Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE Transactions on Signal Processing*, 53(8):3010–3022, August 2005.

[66] H. Meikle. *Modern radar systems.* Artech House Publishers, 2008.

[67] G. Michailidis. Optimal allocation in a queueing system with shared resources. In *2003 Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 3, pages 2106–2111. IEEE, 2003.

[68] M. J. Neely, E. Modiano, and C. E. Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *IEEE Journal on Selected Areas in Communications*, 23(1):89–103, 2005.

[69] G. E. Newstadt, E. Bashan, and A. O. Hero III. Adaptive search for sparse targets with informative priors. In *Proceedings of the 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 3542–3545. IEEE, 2010.

[70] Gregory E. Newstadt, Edmund Zelnio, Leroy Gorham, and Alfred O. Hero III. Detection/tracking of moving targets with synthetic aperture radars. volume 7699, page 76990I. SPIE, 2010.

[71] R. P. Perry, R. C. Dipietro, and R. L. Fante. SAR imaging of moving targets. *IEEE Transactions on Aerospace and Electronic Systems*, 35(1):188–200, 1999.

[72] C. G. Pfeifer and P. Enis. Dorfman-type group testing for a modified binomial model. *Journal of the American Statistical Association*, 73(363):588–592, September 1978.

[73] F. L. Posner. Texture and speckle in high resolution synthetic aperture radar clutter. *IEEE Transactions on Geoscience and Remote Sensing*, 31(1):192–203, 1993.

[74] L. C. Potter, P. Schniter, and J. Ziniel. Sparse reconstruction for radar. *SPIE Algorithms for Synthetic Aperture Radar Imagery XV*, 6970, 2008.

[75] R. K. Raney and G. J. Wessels. Spatial considerations in SAR speckle consideration. *IEEE Transactions on Geoscience and Remote Sensing*, 26(5):666–672, 1988.

[76] R. Rangarajan, R. Raich, and A. O. Hero III. Optimal experimental design for an inverse scattering problem. In *Proceedings. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 1117–1120, March 2005.

[77] R. Rangarajan, R. Raich, and A. O. Hero III. Sequential design of experiments for a Rayleigh inverse scattering problem. In *2005 IEEE/SP 13th Workshop on Statistical Signal Processing*, pages 625–630, July 2005.

[78] R. Rangarajan, R. Raich, and A. O. Hero III. Optimal sequential energy allocation for inverse problems. *IEEE Journal on Selected Topics in Signal Processing*, 1(1):67–78, June 2007.

[79] K. I. Ranney and M. Soumekh. Signal subspace change detection in averaged multilook SAR imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 44(1):201–213, 2006.

[80] D. C. Schleher. *MTI and pulsed Doppler radar.* Artech House, Inc., Norwood, MA, 1991.

[81] M. Soumekh. Moving target detection in foliage using along track monopulse synthetic aperture radar imaging. *IEEE Transactions on Image Processing*, 6(8):1148 –1163, August 1997.

[82] M. Soumekh. *Synthetic aperture radar signal processing with MATLAB algorithms*, volume 138. Wiley, 1999.

[83] A. L. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 50(4):401–457, 2005.

[84] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, 1992.

[85] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

[86] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[87] J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, March 2006.

[88] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[89] K. M. Wasserman, G. Michailidis, and N. Bambos. Optimal processor allocation to differentiated job flows. *Performance Evaluation*, 63(1):1–14, 2006.

[90] D. Wei and A. O. Hero III. Multistage adaptive estimation of sparse signals. In *Proceedings of 2012 IEEE Statistical Signal Processing Workshop*, August 2012.

[91] R. Willett, A. Martin, and R. Nowak. Backcasting: adaptive sampling for sensor networks. In *Third International Symposium on Information Processing in Sensor Networks*, pages 124–133, April 2004.

[92] D. P. Wipf and B. D. Rao. Sparse Bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164, August 2004.

[93] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. *submitted to Journal of the ACM*, 2009.

[94] S. Zhou, J. Lafferty, and L. Wasserman. Compressed and privacy-sensitive sparse regression. *IEEE Transactions on Information Theory*, 55(2):846–866, 2009.

[95] S. Zhu, G. Liao, Y. Qu, Z. Zhou, and X. Liu. Ground moving targets imaging algorithm for synthetic aperture radar. *IEEE Transactions on Geoscience and Remote Sensing*, 49(1):462–477, 2011.