DISSERTATION

# COMPRESSIVE MEASUREMENT DESIGN FOR DETECTION AND ESTIMATION OF SPARSE SIGNALS

Submitted by

Ramin Zahedi

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2013

Doctoral Committee:

Advisor: Edwin K. P. Chong
Co-Advisor: Ali Pezeshki

Donald Estep
Peter M. Young

ABSTRACT

COMPRESSIVE MEASUREMENT DESIGN FOR DETECTION AND ESTIMATION OF SPARSE SIGNALS

We study the problem of designing compressive measurement matrices for two sets of problems. In the first set, we consider the problem of adaptively designing compressive measurement matrices for estimating time-varying sparse signals. We formulate this problem as a *Partially Observable Markov Decision Process (POMDP)*. This formulation allows us to use Bellman's principle of optimality in the implementation of multi-step lookahead designs of compressive measurements. We introduce two variations of the compressive measurement design problem. In the first variation, we consider the problem of selecting a prespecified number of measurement vectors from a predefined library as entries of the compressive measurement matrix at each time step. In the second variation, the number of compressive measurements, i.e., the number of rows of the measurement matrix, is adaptively chosen. Once the number of measurements is determined, the matrix entries are chosen according to a prespecified adaptive scheme. Each of these two problems is judged by a separate performance criterion. The gauge of efficiency in the first problem is the conditional mutual information between the sparse signal support and the measurements. The second problem applies a linear combination of the number of measurements and the conditional mutual information as the performance measure. We present several simulations in which the primary focus is the application of a method known as *rollout*. The significant computational load for using rollout has also inspired us to adapt two data association heuristics in our simulations to the compressive sensing paradigm. These heuristics show promising decreases in the amount of computation for propagating distributions and searching for optimal solutions.

In the second set of problems, we consider the problem of testing for the presence (or

detection) of an unknown static sparse signal in additive white noise. Given a fixed measurement budget, much smaller than the dimension of the signal, we consider the general problem of designing compressive measurements to maximize the measurement signal-to-noise ratio (SNR), as increasing SNR improves the detection performance in a large class of detectors. We use a *lexicographic optimization* approach, where the optimal measurement design for sparsity level $k$ is sought only among the set of measurement matrices that satisfy the optimality conditions for sparsity level $k-1$. We consider optimizing two different SNR criteria, namely a worst-case SNR measure, over all possible realizations of a $k$-sparse signal, and an average SNR measure with respect to a uniform distribution on the locations of the up to $k$ nonzero entries in the signal. We establish connections between these two criteria and certain classes of tight frames. We constrain our measurement matrices to the class of tight frames to avoid coloring the noise covariance matrix. For the worst-case problem, we show that the optimal measurement matrix is a Grassmannian line packing for most—and a uniform tight frame for all—sparse signals. For the average SNR problem, we prove that the optimal measurement matrix is a uniform tight frame with minimum *sum-coherence* for most—and a tight frame for all—sparse signals.

# ACKNOWLEDGEMENTS

My experience at CSU has truly been an amazing part of my life. Over the last five years, I have been fortunate enough to meet a lot of intellectual individuals and learn from them in different aspects of life. These individuals, along with what CSU and Fort Collins had to offer, have changed my life for good and forever. I can only say thank you all.

I would like to express my deepest gratitude to my adviser, Professor Edwin K. P. Chong, who has continuously and patiently guided and encouraged me in this stage of my life. His immense knowledge, insightful comments, and most of all, his kind and pleasant personality, have made my experience to be very exceptional and joyful.

My sincere thanks go to my co-adviser, Professor Ali Pezeshki, who has spent an equivalent amount of time as Professor Chong to give me insightful advice and enhance my academic skills. As well as that, Professor Pezeshki has been a true friend inside and outside of the academic environment and has helped me to overcome several challenges in my life. I am very thankful for that.

I would also like to thank my Ph.D. committee members, Professors Don Estep and Peter Young, for their valuable time and useful guidelines that have enriched the quality of my work and dissertation.

Without my colleagues, this period of my life would have been more difficult and less tasteful. First, I like to thank my dearest friends, Yang Zhang and Vidarshana Bandara, who started this journey with me and have been there for me all along. I deeply appreciate their steady friendship and support throughout this experience. My regards also go to the rest of the team and colleagues: Shankar Ragi, Zhenliang Zhang, Lucas Krakow, Yajing Liu, Mark Ritschard, and Wenbing Dang. In particular, I would like to thank Lucas Krakow who collaborated with me on the work presented in the first part of this dissertation.

Living in Fort Collins over the last few years helped me meet a lot of good friends. Even though listing the names of all these friends would be impossible, I do want them to

know that I am very grateful for their friendship and support as well as the fun times we had together. I also like to thank my friends in Iran, in particular Majid Haghollahi and Mahdieh Sharifi, who have still kept a close relationship with me and have been encouraging me over the last few years. I do think of them all the time and hope we could all reunite someday.

My deepest appreciations go to my partner in life, Mahsa Ghafarzadeh Alipour, who kindly stood by me and supported me all along. I am very thankful for her presence and effort in spicing up my life and more importantly, for her patience.

Last but not least, words are incapable of describing of how grateful I am to have such great parents and sister, Mary Catherine Sheedy, Morteza Zahedi, and Lida Zahedi. They have been so understanding of my decision to continue working on my Ph.D. even when my presence was needed most back home. I like to thank them for their unconditional love and support and I only hope to be able to be there for them in the future.

The work reported in this dissertation was supported in part by DARPA/DSO contract N66001-11-C-4023, ONR contract N00014-08-1-110, AFOSR contract FA9550-09-1-0518, and NSF grants ECCS-0700559, CCF-0916314, and CCF-1018472.

TABLE OF CONTENTS

LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

Over the past few years, considerable progress has been made towards developing a mathematical framework for *reconstructing* sparse or compressible signals.[1] The most notable result is the development of the compressed sensing theory (see, e.g., [2]–[7]), which shows that an unknown signal can be recovered from a small (relative to its dimension) number of linear measurements provided that the signal is sparse. Thus, compressed sensing and related sparse recovery methods have become topics of great interest, leading to many exciting developments in sparse representation theory, measurement design, and sparse recovery algorithms (see, e.g, [8]–[14]).

Early results in compressive sensing show that the exact recovery of a sparse signal or computing a good estimate of a compressible signal can be achieved with a high probability if compressive measurement matrices with random Gaussian or Bernoulli entries are used for measuring the signal. Since then, numerous methods for designing the compressive measurement matrix have been proposed (see, e.g., [15]–[20]). For example, the work in [19] provides a deterministic structure for the measurement matrix which requires more measurements to recover the sparse signal but the overall computational cost is much lower than the random design. Recently, more realistic settings for the sparse signal recovery problem have been considered (see, e.g., [21] and [22]), opening new horizons for alternative compressive measurement designs.

The traditional compressive sensing methods are non-adaptive in the sense that the

[1] A vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ is *sparse* when the cardinality of its support $S = \{k : x_k \neq 0\}$ is much smaller than its dimension $N$. A vector $\mathbf{x}$ is *compressible* if its entries obey a power law, i.e., the $k$th largest entry in absolute value, denoted by $|x|_{(k)}$, satisfies $|x|_{(k)} \leq C_r \cdot k^{-r}$, $r > 1$ and $C_r$ is a constant depending only on $r$ (see, e.g., [2]). Then $\|\mathbf{x} - \mathbf{x}_k\|_1 \leq \sqrt{k} C_r' \cdot k^{-r+1}$, where $\mathbf{x}_k$ is the best $k$-term approximation of $\mathbf{x}$.

measurement matrix is determined in advance, prior to receiving any measurements of the sparse signal. The recent theoretical result in [23] shows that, in a certain asymptotic sense, adaptive designs have limited performance advantage over traditional non-adaptive schemes. On the other hand, several works (see, e.g., [24]–[28]) have proposed adaptive designs for certain scenarios, showing (non-asymptotic) improvements resulting from adaptation under these scenarios.

The above works all assume a *static* sparse signal. This means that over time, the locations and values of the nonzero entries of the sparse signal stay constant. Although this assumption is valid in some applications, in many practical scenarios, considering a static signal seems unrealistic. Examples of time-varying sparse signals arise naturally in target tracking, high-speed video capturing, time-varying multi-path in communication, etc. However, little attention has so far been paid to the time-varying case (notable exceptions being [29]–[33]), where one would expect adaptation to play a more significant role.

In the first part of this work, presented in Chapter 2, we study the problem of adaptively designing compressive measurement matrices for estimating time-varying sparse signals. We take a unique perspective that enables melding compressive sensing and data association techniques for multi-target tracking: We view the time-varying $s$-sparse signal of dimension $N$ as a representation of the combination of target locations (the vector support) and target values (the non-zero values located at the support indices) of $s$ targets moving over $N$ possible locations according to a discrete motion-model. We assume that only the target locations and not the target values vary over time.

We present two problems in which our main goal behind solving both of them is to estimate the support of the time-varying sparse signal at different time steps. These two problems present two variations of the support identification problem of sparse signals. Also, both of these problems lend themselves to adaptive measurement solutions, successively selecting action inputs as observations are collected. We formulate each problem as a finite-horizon partially observable Markov decision process (POMDP) (see [34] and [35]). This

formulation enables us to bring to bear Bellman's principle of optimality.

The POMDP framework accommodates two categories of adaptive solution methods, allowing their implementation and comparison. These two categories are known as "1-step lookahead" or "myopic" and "multi-step lookahead" methods in the literature, respectively. Throughout several simulations in Section 2.4, we show that our adaptive designs outperform the non-adaptive schemes in all of the presented scenarios. However, when comparing the performance of myopic vs. multi-step lookahead designs, several factors such as the performance criterion of the problem or the POMDP action space influence the performance difference between these two designs. This is because the multi-step lookahead methods can perform better than myopic solutions only when there is a benefit in accounting for "long term" effects. We verify this claim throughout our simulations.

The work presented in Chapter 2 can be used for estimating any time-varying sparse signal with any number of nonzero entries. Note that once we assume more than one nonzero moving entry in the signal, we must consider the problem of data association (see, e.g., [36]–[38]) that naturally arises in our problem. Therefore, we have to modify our POMDP formulation (Section 2.2) accordingly to take this problem into account.

It is well known that solving a POMDP problem exactly is typically computationally prohibitive (see, e.g., [39] and [40]). In our simulations for this work, we apply a few approximation techniques that decrease the existing computation volume. We will introduce these techniques in Section 2.2.2.

As mentioned earlier, the major part of the effort in the compressive sensing literature has been focused on *estimating* sparse signals. Hypothesis testing (detection and classification) involving sparse signal models, on the other hand, has been scarcely addressed, notable exceptions being [41]–[45]. Detecting a sparse signal in noise is fundamentally different from reconstructing a sparse signal, as the objective in detection often is to maximize the probability of detection or to minimize a Bayes' risk, rather than to find the sparsest signal that satisfies a linear observation equation.

We note that in the compressed sensing literature the term "sparse signal detection" often means identifying the support of a sparse signal. However, in the second part of this work, presented in Chapter 3, we use this term to refer to a binary hypothesis test for the presence or absence of a sparse signal in noise. The problem is to decide whether a measurement vector is a realization from a hypothesized noise only model or from a hypothesized signal-plus-noise model, where in the latter model the signal is sparse in a known basis but the indices and values of its nonzero coordinates are unknown.

Existing work (e.g., see [41]–[43]) is mainly focused on understanding how the performance of well-known detectors (e.g., the Neyman-Pearson detector) are affected by measurement matrices that have the so-called restricted isometry property (RIP). The RIP condition for the measurement matrix is sufficient for the minimum $\ell_1$-norm solution to be exact (or near-exact when the measurements are noisy) (e.g., see [5]). A fundamental result of compressed sensing has been to establish that random matrices, with independently and identically distributed (i.i.d.) Gaussian or i.i.d. Bernoulli entries, satisfy the RIP condition with high probability. The analysis presented in [41] and [42] provides theoretical bounds on the performance of a Neyman-Pearson detector—quantified by the maximum probability of detection achieved at a pre-specified false alarm rate—when matrices with i.i.d. Gaussian entries are used for collecting measurements. In [43], the authors derive bounds on the total error probability for detection, involving both false alarm and miss detection probabilities, but again for measurement matrices with i.i.d. entries. Finally, in [44] and [45], the authors develop compressive matched subspace detectors that also use random matrices for collecting measurements for detecting sparse signals in known subspaces.

The body of work reported in [41]–[45] provides a valuable analysis of the performance of different detectors, but leaves the question of how to design measurement matrices to optimize a measure of detection performance open. As in the case of reconstruction, random matrices have been studied in these papers in the context of signal detection primarily

because of the tractability of the associated performance analysis. But what are the necessary and sufficient conditions a compressive measurement matrix must have to optimize a desired measure of detection performance? How can matrices that satisfy such conditions be constructed? Our aim in Chapter 3 is to take initial but significant steps towards answering these questions.

# CHAPTER 2

# ADAPTIVE ESTIMATION OF TIME-VARYING SPARSE SIGNALS

## 2.1 Introduction

In this chapter, we study the problem of adaptively designing compressive measurement matrices for estimating time-varying sparse signals. As mentioned in Chapter 1, we view the time-varying $s$-sparse signal of dimension $N$ as a representation of the combination of target locations (the vector support) and target values (the non-zero values located at the support indices) of $s$ targets moving over $N$ possible locations according to a discrete motion-model.

At each time step $k$, the locations and strength values of these targets can be represented by an $s$-sparse vector $\mathbf{x}_k$ in $\mathbb{R}^N$. In this chapter, we call an $N$-dimensional signal *$s$-sparse* if it has at most $s \ll N$ nonzero entries. Moreover, we assume that only the target locations, or equivalently the *support* of the sparse signal $\mathbf{x}_k$, change over time and not the target values. Our goal is to estimate the support of the sparse signal at each time step from compressive measurements.

During each time step $k$, we collect $1 \leq l_k \leq l_{\max}$ measurements ($l_{\max}$ is a prespecified integer), represented by the $l_k$-vector $\mathbf{y}_k = [y_1, y_2, \ldots, y_{l_k}]^T$, according to the linear model $\mathbf{y}_k = \mathbf{A}_k\mathbf{x}_k + \mathbf{w}_k$. In this model, $\mathbf{A}_k \in \mathbb{R}^{l_k \times N}$ is a compressive measurement matrix with rows $\mathbf{A}_k(i)$, $i = 1, \ldots, l_k$, and $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}_{l_k}, \sigma_w^2\mathbf{I}_{l_k})$, where $\mathcal{N}(\mathbf{0}_{l_k}, \sigma_w^2\mathbf{I}_{l_k})$ denotes the $l_k$-variate normal distribution with the zero mean vector $\mathbf{0}_{l_k}$ and the covariance matrix $\sigma_w^2\mathbf{I}_{l_k}$, and $\mathbf{I}_{l_k}$ is the $(l_k \times l_k)$ identity matrix. We assume that $\|\mathbf{A}_k(i)\|_2 = 1$ for $i = 1, \ldots, l_k$ and $k = 1, \ldots, m$. Also, we assume that at each time step, the movement of the targets is independent of the measurement matrix $\mathbf{A}_k$ or the number of measurements $l_k$. Thus, at any time step, multiple

targets can move to the same location, effectively reducing the number of nonzero entries in the signal to less than $s$.

We now introduce two problems of interest. These problems both contain the essence of compressive sensing as described above. At the same time, each problem presents unique goals and constraints. The differences in these problems ultimately define the type of their favored solution methods.

**Problem 1.** Fix $l_k = l$ for $k = 1, 2, \ldots, m$, where $1 \leq l \leq l_{\max}$ is a fixed integer. Our goal is to sequentially select measurement matrices $\mathbf{A}_k, k = 1, 2, \ldots, m$, from a prespecified library to maximize a measure of performance for estimating locations of the nonzero entries (support) of the sparse signal $\mathbf{x}_k$ at each time step $k$. The performance measure for this problem is the conditional mutual information between the measurement vector $\mathbf{y}_k$ and the support of the signal $\mathbf{x}_k$ given the previous measurements.

**Problem 2.** Here, $l_k$ is not fixed and is, in fact, the action input chosen sequentially at each time step, $k = 1, 2, \ldots, m,$. Once the value $l_k$ is chosen, the measurement matrix $\mathbf{A}_k$ is constructed using a prespecified scheme, explained in Section 2.4. The performance criterion in this problem is defined as a weighted sum of the number of measurements $l_k$ and the conditional mutual information between the measurements and the support of the sparse signal at each time step.

The problems introduced above present two variations of support identification for sparse signals. We formulate each problem as a finite-horizon partially observable Markov decision process (POMDP) (see [34] and [35]) in Section 2.2, which enables us to bring to bear Bellman's principle of optimality. When variations of Problem 1 have been considered in the compressive sensing literature, the common solution approach is categorized as a "1-step lookahead" or "myopic" method. In other words, only the performance at the current time step is considered when selecting the compressive measurement matrices. Although Section 2.4 *includes* results from 1-step lookahead methods, our primary focus in this work is the application of "multi-step lookahead" solutions.

It is important to realize that multi-step lookahead methods can perform better than myopic solutions only when there is a benefit in accounting for long-term effects. Several factors play a role in providing such benefits. The performance criterion of the problem and the action space restrictions are examples of such factors. In this work, throughout several simulations, we show that although adaptive designs outperform the non-adaptive schemes, the performance criterion and the action space of Problem 1 preclude the multi-step lookahead solution from surpassing the myopic solutions. The alterations in Problem 2, namely the action space and the performance measure, exemplify long-term considerations where the performance of multi-step lookahead solutions exceeds that of myopic schemes. We verify this claim throughout several simulations in Section 2.4.

It is well known that solving a POMDP problem is typically computationally prohibitive. The following approximation techniques, presented in Section 2.2.2, decrease the computation volume, alleviating some of the conventional concerns:

1) We introduce two heuristics that are motivated from the well known techniques *joint probabilistic data association* (JPDA) and *multiple hypothesis tracking* (MHT) in the target tracking literature (see, e.g., [46]–[48]) to simplify the update of the belief state in the POMDP.

2) In the multi-step lookahead variation of our method, we use an approximation method, known as *rollout* (see [49]), to estimate an optimal solution for the POMDP.

## 2.2 POMDP Formulation

First, we formulate the problems introduced above as POMDPs. Since most of the POMDP elements for these two problems are similar, we present one POMDP formulation, but clarify the differences in the formulation of each problem where it is necessary.

**States and Transition Law**: The $s$-sparse vector $\mathbf{x}_k$ is fully characterized by its support (the locations of the targets) and the strength values (the values of the nonzero entries). Therefore, we take the state of the POMDP at time $k$ to be $\mathbf{s}_k = (\mathbf{d}_k, \mathbf{v}_k)$, where $\mathbf{d}_k$ and $\mathbf{v}_k$

are $(s \times 1)$ vectors. The $i$th entry of these vectors, $\mathbf{d}_k(i)$ and $\mathbf{v}_k(i)$, represent the location and the strength value of the $i$th target at time $k$, respectively. Let the set $\Omega$ be defined as $\Omega = \{1, \ldots, N\}$. Also, let $\Omega_s$ be the set containing all the $(s \times 1)$ vectors $\mathbf{d}$ such that $\mathbf{d}(i) \in \Omega$ for $i = 1, \ldots, s$. Note that by using this definition for $\Omega_s$, we are allowing any group of targets to be in the same location at the same time. Therefore, the cardinality of the set $\Omega_s$ is equal to $|\Omega_s| = N^s$. The POMDP state space is then defined as the Cartesian product $\mathcal{S} = (\Omega_s \times \mathbb{R}^s)$.

The state transition law of the POMDP, defined by the movement of the targets, is independent of our actions. Although the above definition for the POMDP state allows for the strength values of the targets to change over time, we make the simplifying assumption that these values stay constant over time. Accordingly, the state transition law of the POMDP can be described as:

$$P\{\mathbf{s}_{k+1} = (\mathbf{d}, \mathbf{v}) | \mathbf{s}_k = (\mathbf{e}, \mathbf{z})\} = \begin{cases} p_{\mathbf{ed}}, & \mathbf{v} = \mathbf{z}, \\ 0, & \text{otherwise}, \end{cases}$$

where $p_{\mathbf{ed}}$ is the probability that targets in locations represented by the vector $\mathbf{e}$ transition to locations represented by the vector $\mathbf{d}$. Depending on the movement of the targets, the probabilities $p_{\mathbf{ed}}$ could have different values. For example, if we consider the case where targets stay in the same locations at all time steps, then $p_{\mathbf{ed}} = 1$ if $\mathbf{e} = \mathbf{d}$, and $p_{\mathbf{ed}} = 0$ if otherwise.

**Actions**: In Problem 1, the action at each time step $k$ is the selection of the measurement matrix $\mathbf{A}_k$. Therefore, the action space $\mathcal{A}$ is a prespecified subset of all matrices $\mathbf{A} \in \mathbb{R}^{l \times N}$ such that, for each row $\mathbf{A}(i), i = 1, \ldots, l$, $\|\mathbf{A}(i)\|_2 = 1$. In Problem 2, choosing the number of measurements $l_k$ at each time step $k$ is the POMDP action. In this case, the action space $\mathcal{A}$ is the set of natural numbers. For simplicity, we use the notation $\mathbf{u}_k$ for the POMDP action at time $k$ for both problems.

**Observations and Observation Law**: The POMDP observations are the measurements $\mathbf{y}_k$ at each time step $k$. The observation law can be described in the following way:

Given $\mathbf{s}_k = (\mathbf{d}, \mathbf{v})$ and the matrix $\mathbf{A}_k = \mathbf{A}$ at time step $k$, it can be easily shown that $\mathbf{y}_k|(\mathbf{s}_k = (\mathbf{d}, \mathbf{v}), \mathbf{A}_k = \mathbf{A}) \sim \mathcal{N}(\mathbf{A_d v}, \sigma_w^2 \mathbf{I}_{l_k})$, where $\mathbf{A_d}$ is a matrix whose $i$th column is the $\mathbf{d}(i)$th column of the matrix $\mathbf{A}$.

**Cost**: Let $I(\mathbf{y}_k; \mathbf{d}_k|H_k)$ be the conditional mutual information between the signal support $\mathbf{d}_k$ and the measurements $\mathbf{y}_k$ given the history $H_k = \{\mathbf{u}_1, \mathbf{y}_1, \ldots, \mathbf{u}_k, \mathbf{y}_k\}$. The per-time-step cost $c_k(\mathbf{s}_k, \mathbf{u}_k)$ for Problem 1 is then simply defined as $c_k(\mathbf{s}_k, \mathbf{u}_k) = -I(\mathbf{y}_k; \mathbf{d}_k|H_k)$. For Problem 2, the cost is a combination of the number of measurements $l_k$ and the conditional mutual information $I(\mathbf{y}_k; \mathbf{d}_k|H_k)$ at each time step $k$. More specifically, the per-time-step cost is defined as

$$c_k(\mathbf{s}_k, \mathbf{u}_k) = l_k - \gamma I(\mathbf{y}_k; \mathbf{d}_k|H_k), \tag{2.1}$$

where $\gamma \geq 0$ is a weighting parameter chosen based on the priority of either $l_k$ or $I(\mathbf{y}_k; \mathbf{d}_k|H_k)$. For example, if we are not particularly concerned with the number of measurements, then we choose a large value for $\gamma$. On the contrary, if we have to be careful with the total number of measurements used in $m$ steps, we use a small value for $\gamma$.

**Belief State**: The belief state $\mathbf{b}_k$ at time $k$ is defined as the posterior (conditional) distribution of the state $\mathbf{s}_k$ given the history $H_k$. Using the belief state definition, our POMDP can be presented as an equivalent Markov Decision Process (MDP) (see [50]). This MDP, which is also referred to as the *belief MDP*, has the following components:

1. A state space consisting of all the possible belief states $\mathbf{b}_k$ for the POMDP.

2. A state transition law that is computed using the state transition law and the observation law of the POMDP. This computation is referred to as the *belief state update*, shown in detail in the next section.

3. An action space which is the same as that of the POMDP.

4. A cost $C_k(\mathbf{b}_k, \mathbf{u}_k)$, referred to as the *belief cost*, which is defined as:

$$C_k(\mathbf{b}_k, \mathbf{u}_k) = \mathbf{E}\left[c_k(\mathbf{s}_k, \mathbf{u}_k)|H_k, \mathbf{b}_k\right], \tag{2.2}$$

where the expectation is with respect to the posterior distribution of the state $\mathbf{s}_k$ given the history $H_k$ at time step $k$, i.e., the belief state $\mathbf{b}_k$.

Using the belief MDP model, we can now apply the approximation methods available in the literature, specifically rollout [51].

### 2.2.1 Belief State Update

Let $P_{\mathbf{d}_k|H_k}$ be the conditional probability mass function of $\mathbf{d}_k$ given $H_k$, and $f_{\mathbf{v}_k|\mathbf{d}_k,H_k}$ be the conditional density function of $\mathbf{v}_k$ given $\mathbf{d}_k$ and $H_k$ at time step $k$. Consider a simple example, shown in Fig. 2.1, where there are three *nodes*, $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$, representing the possible signal supports at any time step. Fig. 2.1 also displays the *links* between nodes, representing allowed support transitions. For example, there are two possible transitions arriving at $\mathbf{e}_1$, namely, those originating from nodes $\mathbf{e}_1$ and $\mathbf{e}_2$. Attached to the initial node of each link is a weight value, computed from a prior conditional probability mass function, and also a prior conditional density function. Upon choosing an action and receiving measurements, we can compute a posterior weight value as well as a posterior density function for the terminal node of the link.

To clarify, consider the selected *route*, comprised of successive links and outlined in red in Fig. 2.1, that begins at node $\mathbf{e}_3$ at time $k = 1$ and after passing through node $\mathbf{e}_2$ at time $k = 2$, it finally arrives at node $\mathbf{e}_1$ at $k = 3$. At time $k = 1$, once measurement $\mathbf{y}_1$ is received, we can compute the posterior weight value $P_{\mathbf{d}_1|H_1}(\mathbf{e}_3|H_1)$ and the posterior density function $f_{\mathbf{v}_1|\mathbf{d}_1,H_1}(\cdot|\mathbf{e}_3, H_1)$ using the initial prior functions $P_{\mathbf{d}_0}$ and $f_{\mathbf{v}_0|\mathbf{d}_0}$ (this calculation is shown later in this section). Then, examining the link from $\mathbf{e}_3$ to $\mathbf{e}_2$, the prior weight and density function attached to node $\mathbf{e}_3$ for the next time step are $P_{\mathbf{d}_1|H_1}(\mathbf{e}_3|H_1)$ and $f_{\mathbf{v}_1|\mathbf{d}_1,H_1}(\cdot|\mathbf{e}_3, H_1)$, respectively. The prior values are updated using the received observations, resulting in the posterior weight and density function $P_{\mathbf{d}_1|\mathbf{d}_0,H_1}(\mathbf{e}_2|\mathbf{e}_3, H_1)$ and $f_{\mathbf{v}_1|\mathbf{d}_1,\mathbf{d}_0,H_1}(\cdot|\mathbf{e}_2, \mathbf{e}_3, H_1)$, respectively. For the next link in the route from $\mathbf{e}_2$ to $\mathbf{e}_1$, the prior weight and density function used for $\mathbf{e}_2$ will be the posterior weight and density function of this node at $k = 2$,

i.e., $P_{\mathbf{d}_1|\mathbf{d}_0,H_1}(\mathbf{e}_2|\mathbf{e}_3,H_1)$ and $f_{\mathbf{v}_1|\mathbf{d}_1,\mathbf{d}_0,H_1}(\cdot|\mathbf{e}_2,\mathbf{e}_3,H_1)$, respectively. Again, after choosing an action and receiving measurements, the posterior weight $P_{\mathbf{d}_2|\mathbf{d}_1,H_2}(\mathbf{e}_1|\mathbf{e}_2,H_2)$ and posterior density function $f_{\mathbf{v}_2|\mathbf{d}_2,\mathbf{d}_1,H_2}(\cdot|\mathbf{e}_1,\mathbf{e}_2,H_2)$ for node $\mathbf{e}_1$ are computed.

It is important to realize that after the first time step, the posterior weight and density function for the ending node of any link are always conditioned on the initial node of that link. Thus, knowing the positions of the targets and their prior weight and density at the previous time step is required. At any time step $k$, given each possible link, we define an ordered tuple $\boldsymbol{\tau}$ consisting of four elements: the terminal node, the initial node, the prior weight value, and the prior density function attached to the initial node and let $\boldsymbol{\tau}(i)$ be the $i$th element of the tuple. For the same route used in the above example, the tuple defined for the link from $\mathbf{e}_3$ to $\mathbf{e}_2$ is $\boldsymbol{\tau} = \big(\mathbf{e}_2, \mathbf{e}_3, P_{\mathbf{d}_1|H_1}(\mathbf{e}_3|H_1), f_{\mathbf{v}_1|\mathbf{d}_1,H_1}(\cdot|\mathbf{e}_3,H_1)\big)$. Analogously, the tuple for the last link in this route, from $\mathbf{e}_2$ to $\mathbf{e}_1$, is $\boldsymbol{\tau} = \big(\mathbf{e}_1, \mathbf{e}_2, P_{\mathbf{d}_2|\mathbf{d}_1,H_2}(\mathbf{e}_1|\mathbf{e}_2,H_2), f_{\mathbf{v}_2|\mathbf{d}_2,\mathbf{d}_1,H_2}(\cdot|\mathbf{e}_1,\mathbf{e}_2,H_2)\big)$.

At any time step, multiple tuples with the same initial and terminal nodes may exist. For example, in Fig. 2.1, from time $k = 2$ to time $k = 3$, there are two links starting at $\mathbf{e}_2$ and ending at $\mathbf{e}_1$. These two links can be distinguished by the unique priors attached to the starting nodes of either link. Let $\mathcal{T}_k$ be the set containing all the possible tuples at time step $k$. Also, define $\mathcal{T}_{k,\mathbf{d}}$ to be $\mathcal{T}_{k,\mathbf{d}} = \{\boldsymbol{\tau} : \boldsymbol{\tau} \in \mathcal{T}_k, \boldsymbol{\tau}(1) = \mathbf{d}\}$, the set of all tuples representing links ending with support $\mathbf{d}$ at time $k$. At each time step $k$, the number of possible links, which depends on the number of initial nodes and the state transition law, determines the cardinality $|\mathcal{T}_k|$ of the set $\mathcal{T}_k$, which increases exponentially as time evolves. We will discuss

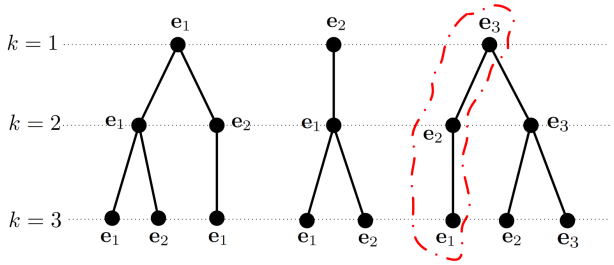



**Figure 2.1:** A simple example showing the construction of various paths from only three support possibilities $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ over time.

this issue later.

Recall that the state of the POMDP at time step $k$ is $\mathbf{s}_k = (\mathbf{d}_k, \mathbf{v}_k)$. The belief state $\mathbf{b}_k$ is the posterior joint distribution of $\mathbf{d}_k$ and $\mathbf{v}_k$ given $H_k$, or equivalently, the functions $P_{\mathbf{d}_k|H_k}$ and $f_{\mathbf{v}_k|\mathbf{d}_k,H_k}$. Therefore, updating the belief state $\mathbf{b}_k$ is equivalent to updating these functions. Moreover, given $\mathbf{y}_k = \mathbf{y}$ and $\mathbf{u}_k = \mathbf{u}$, the functions $f_{\mathbf{v}_k|\mathbf{d}_k,H_k}$ and $P_{\mathbf{d}_k|H_k}$ can be computed from $P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}$ and $f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}$ using Bayes' rule and the law of total probability in the following way:

$$f_{\mathbf{v}_k|\mathbf{d}_k,H_k}(\mathbf{v}|\mathbf{d},H_k) = \frac{\sum_{\boldsymbol{\tau}\in\mathcal{T}_{k,\mathbf{d}}} g(\mathbf{v},\mathbf{d},\boldsymbol{\tau}) P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}(\mathbf{d}|\boldsymbol{\tau}(2),H_k)\boldsymbol{\tau}(3)}{\sum_{\boldsymbol{\tau}\in\mathcal{T}_{k,\mathbf{d}}} P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}(\mathbf{d}|\boldsymbol{\tau}(2),H_k)\boldsymbol{\tau}(3)}, \tag{2.3}$$

and,

$$P_{\mathbf{d}_k|H_k}(\mathbf{d}|H_k) = \frac{\sum_{\boldsymbol{\tau}\in\mathcal{T}_{k,\mathbf{d}}} P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}(\mathbf{d}|\boldsymbol{\tau}(2),H_k)\boldsymbol{\tau}(3)}{\sum_{\mathbf{e}\in\Omega_s}\sum_{\boldsymbol{\tau}\in\mathcal{T}_{k,\mathbf{e}}} P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}(\mathbf{e}|\boldsymbol{\tau}(2),H_k)\boldsymbol{\tau}(3)}, \tag{2.4}$$

where $g(\mathbf{v},\mathbf{d},\boldsymbol{\tau}) = f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}(\mathbf{v}|\mathbf{d},\boldsymbol{\tau}(2),H_k)$. Note that the value $\boldsymbol{\tau}(3)$ is the prior weight value of the starting node $\boldsymbol{\tau}(2)$ for the link represented by the tuple $\boldsymbol{\tau}$. This means that to update the belief state $\mathbf{b}_k$ at time step $k$, it is sufficient to update functions $P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}$ and $f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}$ for all the possible links that the state transition law allows at each time step.

In this work, we assume a dynamic linear model and a initial Gaussian distribution for $\mathbf{v}_0|\mathbf{d}_0$ given each $\mathbf{d}_0 \in \Omega_s$. Each distribution can be characterized by a density function $f_{\mathbf{v}_0|\mathbf{d}_0}$ with the mean vector $\boldsymbol{\mu}_{\mathbf{d_0}}$ and the covariance matrix $\mathbf{C}_{\mathbf{d}_0}$. Consequently, the density function $f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}$, $k \geq 1$, will also be Gaussian, and we use $\boldsymbol{\mu}_{\mathbf{d}_k|\mathbf{d}_{k-1}}$ for the mean vector and $\mathbf{C}_{\mathbf{d}_k|\mathbf{d}_{k-1}}$ for the covariance matrix.

Given the action $\mathbf{u}_k = \mathbf{u}$, the measurements $\mathbf{y}_k = \mathbf{y}$, and the history $H_{k-1}$, the functions $P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}$ and $f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}$ can be computed using the following two steps:

1. For a given any link, represented by a tuple $\boldsymbol{\tau}$ in $\mathcal{T}_k$, the following system of linear equations describes the dynamics of the moving targets along that link:

$$\begin{cases} \mathbf{v}_k = \mathbf{v}_{k-1}, \\ \mathbf{y}_k = \mathbf{A}_{\boldsymbol{\tau}(1)}\mathbf{v}_k + \mathbf{w}_k. \end{cases}$$

In this system, $\mathbf{v}_k \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\tau}(1)|\boldsymbol{\tau}(2)}, \mathbf{C}_{\boldsymbol{\tau}(1)|\boldsymbol{\tau}(2)})$. The vector $\mathbf{v}_{k-1}$ also has a Gaussian distribution, which is represented by the density function $\boldsymbol{\tau}(4)$ in the tuple $\boldsymbol{\tau}$. Therefore, the values $\boldsymbol{\mu}_{\boldsymbol{\tau}(1)|\boldsymbol{\tau}(2)}$ and $\mathbf{C}_{\boldsymbol{\tau}(1)|\boldsymbol{\tau}(2)}$ for the given support pair $\mathbf{d}_k = \boldsymbol{\tau}(1)$ and $\mathbf{d}_{k-1} = \boldsymbol{\tau}(2)$ can be computed by using a Kalman filter. Note that for each possible tuple $\boldsymbol{\tau} \in \mathcal{T}_k$, we have to use a separate Kalman filter to update the corresponding prior distribution function $\boldsymbol{\tau}(4)$ of the related link. In the next section, we will discuss the computational issues that usually arise. Once this update is completed, the value of the function $f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}(\mathbf{v}|\boldsymbol{\tau}(1),\boldsymbol{\tau}(2),H_k)$ for any $\mathbf{v} \in \mathbb{R}^s$ can then be computed.

2. For a given tuple $\boldsymbol{\tau} \in \mathcal{T}_k$, the posterior weight value is computed in the following way:

$$P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}(\boldsymbol{\tau}(1)|\boldsymbol{\tau}(2),H_k) = \frac{f_1(\mathbf{y},\mathbf{u},\boldsymbol{\tau},\boldsymbol{\tau})f_2(\boldsymbol{\tau},\boldsymbol{\tau})}{\sum_{\boldsymbol{\nu}\in\mathcal{T}_k} f_1(\mathbf{y},\mathbf{u},\boldsymbol{\nu},\boldsymbol{\tau})f_2(\boldsymbol{\nu},\boldsymbol{\tau})}, \tag{2.5}$$

where functions $f_1(\mathbf{y},\mathbf{u},\boldsymbol{\nu},\boldsymbol{\tau})$ and $f_2(\boldsymbol{\nu},\boldsymbol{\tau})$ are defined as:

$$f_1(\mathbf{y},\mathbf{u},\boldsymbol{\nu},\boldsymbol{\tau}) = f_{\mathbf{y}_k|\mathbf{d}_k,\mathbf{d}_{k-1},\mathbf{u}_k,H_{k-1}}(\mathbf{y}|\boldsymbol{\nu}(1),\boldsymbol{\tau}(2),\mathbf{u},H_{k-1}),$$

and,

$$f_2(\boldsymbol{\nu},\boldsymbol{\tau}) = P_{\mathbf{d}_k|\mathbf{d}_{k-1}}(\boldsymbol{\nu}(1)|\boldsymbol{\tau}(2))\boldsymbol{\tau}(3) = p_{\boldsymbol{\tau}(2)\boldsymbol{\nu}(1)}\boldsymbol{\tau}(3),$$

respectively. Note that the value $p_{\boldsymbol{\tau}(2)\boldsymbol{\nu}(1)}$, that can be computed from the state transition law, is equal to zero for the non-existing links from $\boldsymbol{\tau}(2)$ to $\boldsymbol{\nu}(1)$. Moreover, for the function $f_1$, it can be easily shown that

$$\mathbf{y}_k|\mathbf{d}_k,\mathbf{d}_{k-1},\mathbf{u}_k,H_{k-1} \sim \mathcal{N}(\mathbf{A}_{\mathbf{d}_k}\boldsymbol{\mu}_{\mathbf{d}_k|\mathbf{d}_{k-1}}, \mathbf{A}_{\mathbf{d}_k}\mathbf{C}_{\mathbf{d}_k|\mathbf{d}_{k-1}}\mathbf{A}'_{\mathbf{d}_k} + \sigma_w^2\mathbf{I}_{l_k}),$$

where $\mathbf{A}'_{\mathbf{d}_k}$ is the transpose of the matrix $\mathbf{A}_{\mathbf{d}_k}$. Note that the values $\boldsymbol{\mu}_{\mathbf{d}_k|\mathbf{d}_{k-1}}$ and $\mathbf{C}_{\mathbf{d}_k|\mathbf{d}_{k-1}}$ have already been computed in the previous step.

As mentioned before, after going through the above two steps for all the tuples in $\mathcal{T}_k$ and also computing the functions $f_{\mathbf{v}_k|\mathbf{d}_k,H_k}$ and $P_{\mathbf{d}_k|H_k}$ given any $\mathbf{d}_k \in \Omega_s$ using (2.3) and (2.4), the update of the belief state $\mathbf{b}_k$ is completed.

### 2.2.2 Computational Issues in the Belief State Update

The belief state update procedure described in the previous section shows the computations required once measurements are received at each time step. To highlight the computational issues, consider the extreme case where the transition law $P_{\mathbf{d}_k|\mathbf{d}_{k-1}}$ allows the targets to move freely to any position. Also, assume a uniform initial support density function $P_{\mathbf{d}_0}$ over $\Omega_s$. Knowing that $|\Omega_s| = N^s$, then for each support candidate $\mathbf{d}_0$ in $\Omega_s$, there would be $N^s$ possibilities for the support $\mathbf{d}_1$. Therefore, $N^{2s}$ Kalman filters must be employed to update the belief state at time $k = 1$. Extending this line of thought, at any time $k$, $N^{(k+1)s}$ Kalman filters are needed which is an exponential increase in computations over time.

Of course in reality, the assumed transition law $P_{\mathbf{d}_k|\mathbf{d}_{k-1}}$ does not provide such freedom for the moving target(s). Although this reduces the number of possible transitions, the exponential increase in computation remains. This is seen in the following example. Again assume, at time $k = 1$, a uniform prior distribution $P_{\mathbf{d}_0}$ over $\Omega_s$. However, now assume the transition law restricts the number of possible transitions for each support candidate to $q \leq N^s$. Following the same argument as above, there are now $N^s q$ links involved in the belief state update at time $k = 1$ and for time $k = 2$, $N^s q^2$ links are present. Similarly, $N^s q^k$ links (equivalently Kalman filters) are required for the belief state update at any time step $k$.

Obviously, updating the belief state according to the above procedure requires a vast amount of time and also computational power. To deal with the computational volume we introduce two techniques explained below. This allows us to examine the proposed solution methods to Problems 1 and 2 and avoid some of the related computational load.

As we have shown, propagating the entire distribution quickly becomes unmanageable due to its exponential growth-rate in terms of the number of possible links. This issue is also a major problem in the context of data association in the multi-target tracking problem. In this problem, the main question asked is: At each time step, which target does of the collected measurements at that time step represent? If all the observations-to-target associations were

considered valid, a similar expansion of the distribution representing the current target states would occur.

We can consider the procedure in Section 2.2.1 as one of associating the measurements $\mathbf{y}_k$ to the *possible* support-value pairs, which are characterized by their corresponding belief state components $P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}$ and $f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}$. The similarity to the data association problem motivated the adaptation of available solutions in the target tracking context. We have implemented two separate methods based on popular data association algorithms to combat the expansion of our belief state distributions, namely: multi-hypothesis tracking (MHT) (see [1]) and joint probabilistic data association (JPDA) (see [46] and [47]).

**Algorithm 1**: The essential question to ask is how to choose or approximate the proper Gaussian distributions and weights to represent different possible support candidates along different possible routes. There are several approaches to accomplish this, e.g., MMSE, MLE, etc., and in the target tracking literature, there are also numerous implementations applying these approximations. We model our first method after MHT.

The basis of MHT lies in multi-hypothesis testing wherein a choice must be made between several concurrent hypotheses based on a sufficient statistic. Given a set of $J$ hypotheses $\mathcal{H} = \{h_i\}_{i=1}^{J}$ and the measurements $\mathbf{y}_k = \mathbf{y}$, the simplest version of multi-hypothesis testing chooses one hypothesis $h^*$ based on finding an optimal solution for the following optimization problem:

$$h^* = \arg \max_{1 \le i \le J} \left\{ \mathcal{L}\left(h_i|\mathbf{y}\right) P(h_i) \right\},$$

where $\mathcal{L}\left(h_i|\mathbf{y}\right)$ and $P(h_i)$ denote the likelihood given the measurement $\mathbf{y}$ and the prior probability of the hypothesis $h_i$, respectively. The function $\mathcal{L}\left(h_i|\mathbf{y}\right)$ is equivalent to $f_{\mathbf{y}_k|h}(\mathbf{y}|h_i)$, the conditional probability density function of the measurement vector $\mathbf{y}_k$ conditioned on the hypothesis $h$. Therefore, the above optimization problem will be equal to

$$h^* = \arg \max_{1 \le i \le J} \left\{ f_{\mathbf{y}_k,h}(\mathbf{y}, h_i) \right\}.$$

In our setting, similar to the target tracking setting, more measurements accrue as time progresses, providing more information about the underlying state. Obviously, more information can help increase the accuracy of the association process. Even in dynamic systems, observations of future states provide valuable information about the past system states. This is the basic motivation behind the MHT method.

Consider a time-varying signal estimation problem with a finite horizon of length $\kappa$. At each time step, a measurement is acquired through the observation law. Then, the most accurate selection of a hypothesis is made by collecting all of the measurements and maintaining each possible association over the entire horizon. This results in a hypothesis tree, $\kappa$ levels deep. Each level consists of nodes representing the possible support-value pairs (supports may be repeated) and the corresponding weights $P_{\mathbf{d}_k|H_k}$ for each time step $k$. At the final time step $k = \kappa$, applying a multi-hypothesis test to the set of leaf nodes selects the leaf terminating the most probable route in the tree (most probable with respect the posterior distribution). This selection not only yields a current estimate of the support and values that belong to the leaf at the end of the route, but this route, itself, describes the most probable sequence of support-value pairs. Note, in our setting, the value component $\mathbf{v}_k$ of the state has a static transition law, and thus the final estimate of $\mathbf{v}_\kappa$ would be the more accurate that the estimates of $\mathbf{v}_k$, $k < \kappa$ because they would not have benefited from the later observations.

Let $\boldsymbol{\rho}_k^{(i)}$ be an ordered $k$-tuple representing the $i$th possible route at time $k$. The first component $\boldsymbol{\rho}_k^{(i)}(1)$ of this tuple is the current node of this route and the last component $\boldsymbol{\rho}_k^{(i)}(k)$ is the initial node of this route. For example, for the selected route in Fig. 2.1, which we refer to as the $i$th route, we have the following tuples for time steps $k = 1, 2$, and $k = 3$, respectively: $\boldsymbol{\rho}_1^{(i)} = \mathbf{e}_3$, $\boldsymbol{\rho}_2^{(i)} = (\mathbf{e}_2, \mathbf{e}_3)$, and $\boldsymbol{\rho}_3^{(i)} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$. Note that at time step $k$, the total number of possible routes is equal to $|\mathcal{T}_k|$. Let $h_i^{(k)}$ be the hypothesis that at time step $k$, the $i$th route is the true route. If this hypothesis is true, then this means that $\mathbf{d}_k = \boldsymbol{\rho}_k^{(i)}(1), \mathbf{d}_{k-1} = \boldsymbol{\rho}_k^{(i)}(2), \ldots, \mathbf{d}_1 = \boldsymbol{\rho}_k^{(i)}(k)$. Additionally, let $h^{(k)}$ be the random

variable representing the true hypothesis; $h^{(k)}$ takes values on set of all hypotheses $h_i^{(k)}$, $i = 1, \ldots, |\mathcal{T}_k|$, at time step $k$.

Determining the terminal node of the probable route is done by ranking the possible routes with their *track scores*. Let $\boldsymbol{o}_k$ be a tuple defined as $\boldsymbol{o}_k = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$. Then, given the tuple $\boldsymbol{o}_\kappa = \boldsymbol{o}$, where $\boldsymbol{o} = (\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(\kappa)})$, and the history $H_\kappa = \{\mathbf{u}^{(1)}, \mathbf{y}^{(1)}, \ldots, \mathbf{u}^{(\kappa)}, \mathbf{y}^{(\kappa)}\}$ at time step $\kappa$, the track score $S_{\boldsymbol{\rho}_\kappa^{(i)}}$ of the route $i$, represented by the tuple $\boldsymbol{\rho}_\kappa^{(i)}$, is defined as

$$S_{\boldsymbol{\rho}_\kappa^{(i)}} = \mathcal{L}(h_i^{(\kappa)}|\boldsymbol{o})P(h_i^{(\kappa)}) = f_{\boldsymbol{o}_\kappa, h^{(\kappa)}}(\boldsymbol{o}, h_i^{(\kappa)}),$$

where $f_{\boldsymbol{o}_\kappa, h^{(\kappa)}}$ is the joint probability density function of $\boldsymbol{o}_\kappa$ and $h^{(\kappa)}$ at time step $\kappa$. By expanding the function $f_{\boldsymbol{o}_\kappa, h^{(\kappa)}}$, the value of $S_{\boldsymbol{\rho}_\kappa^{(i)}}$ is equal to

$$\begin{aligned} &f_{\mathbf{y}_\kappa|\mathbf{d}_\kappa, \mathbf{d}_{\kappa-1}, \mathbf{u}_\kappa, H_{\kappa-1}}(\boldsymbol{o}(\kappa)|\boldsymbol{\rho}_\kappa^{(i)}(1), \boldsymbol{\rho}_\kappa^{(i)}(2), \mathbf{u}^{(\kappa)}, H_{\kappa-1}) \\ &\quad \times p_{\boldsymbol{\rho}_\kappa^{(i)}(2)\boldsymbol{\rho}_\kappa^{(i)}(1)} P_{\mathbf{d}_{\kappa-1}|\mathbf{d}_{\kappa-2}, H_{\kappa-1}}(\boldsymbol{\rho}_\kappa^{(i)}(2)|\boldsymbol{\rho}_\kappa^{(i)}(3), H_{\kappa-1}). \end{aligned}$$

Note that all of the components in this expression can be computed using the equations presented in Section 2.2.1. Looking at (2.5), one can conclude that $S_{\boldsymbol{\rho}_\kappa^{(i)}}$ is proportional to $S^{\text{eq.}}_{\boldsymbol{\rho}_\kappa^{(i)}}$ where

$$S^{\text{eq.}}_{\boldsymbol{\rho}_\kappa^{(i)}} = \prod_{k=1}^{\kappa} t(\boldsymbol{o}, \boldsymbol{\rho}_k^{(i)}, H_k, k),$$

and the function $t$ is defined as

$$t(\boldsymbol{o}, \boldsymbol{\rho}_k^{(i)}, H_k, k) = \begin{cases} f_{\mathbf{y}_k|\mathbf{d}_k, \mathbf{d}_{k-1}, \mathbf{u}_k, H_{k-1}}(\boldsymbol{o}(k)|\boldsymbol{\rho}_k^{(i)}(1), \boldsymbol{\rho}_k^{(i)}(2), \mathbf{u}^{(k)}, H_{k-1}) & \\ \qquad\qquad \times p_{\boldsymbol{\rho}_k^{(i)}(2)\boldsymbol{\rho}_k^{(i)}(1)}, & k \geq 2, \\ P_{\mathbf{d}_1|H_1}(\boldsymbol{\rho}_1^{(i)}(1)|H_1), & k = 1. \end{cases}$$

Therefore, we can use $S^{\text{eq.}}_{\boldsymbol{\rho}_\kappa^{(i)}}$ equivalently to find most probable route at time step $\kappa$. Practical concerns such as numerical round-off errors motivate a conversion of $S^{\text{eq.}}_{\boldsymbol{\rho}_\kappa^{(i)}}$ to a logarithmic representation, $S^{\log}_{\boldsymbol{\rho}_\kappa^{(i)}}$, in the following way

$$S^{\log}_{\boldsymbol{\rho}_\kappa^{(i)}} = \sum_{k=1}^{\kappa} \log\left(t(\boldsymbol{o}, \boldsymbol{\rho}_k^{(i)}, H_k, k)\right).$$

This alternative representation also provides a simple recursive calculation,

$$S^{\log}_{\boldsymbol{\rho}_k^{(i)}} = S^{\log}_{\boldsymbol{\rho}_{k-1}^{(i)}} + \log\left(t(\boldsymbol{o}, \boldsymbol{\rho}_k^{(i)}, H_k, k)\right), \qquad k = 2, \ldots, \kappa.$$

As mentioned earlier, when no approximation techniques are used, the number of possible routes increases exponentially as time evolves. To deal with this problem, most MHT implementations use a sliding window to designate when the decision making process begins. The sliding window heuristic limits the size of the hypothesis tree that must be maintained. Consider a sliding window of size $w$. After the first $w$ time steps, a hypothesis test is performed, selecting the current leaf with the maximum track score. The route terminating with the selected leaf is then traced back $w$ levels in the tree. The node at the base of this branch and all of its descendants are retained, while all other nodes and branches from time-step $(k-w)$ onward are discarded. At every time step $k > w$, a similar pruning of the tree is performed. The larger the sliding window the better the chances of a correct association due to the increased information accumulated. Fig. 2.2 illustrates this process for an arbitrary time step.

The sliding window size must be balanced with the computational time and memory constraints. The maximum reduction in computation from our MHT based algorithm would be a sliding window of size one. Thus, a hypothesis test would be performed at every time step. One data-association technique that uses a sliding window of size one is JPDA (see [46] and [47]). This is the basis for Algorithm 2, described next.

**Algorithm 2**: Consider the set $\mathcal{T}_k$ defined in Section 2.2.1. Looking at the tuples in this set, each of which is associated with a possible link, we can find multiple tuples that all share a similar ending node. For example, in Fig. 2.1, there are 4 instances for node $\mathbf{e}_1$ at time step $k = 3$. To cope with the computational volume we combine the posterior weights and distributions attached to different instances of an ending node. More specifically, at each time step $k$, instead of keeping all instances of a support candidate $\mathbf{d} \in \Omega_s$, we represent the candidate with only one *approximate* posterior weight value $\hat{P}_{\mathbf{d}_k|H_k}(\mathbf{d}|H_k)$ and one *approximate* posterior distribution function $\hat{f}_{\mathbf{v}_k|\mathbf{d}_k,H_k}(\cdot|\mathbf{d}, H_k)$. This will force each set

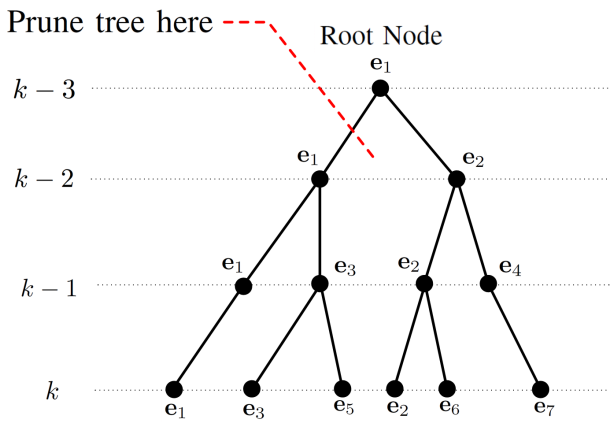



**Figure 2.2:** This illustration (adapted from [1]) represents a possible hypothesis tree formed from a time-varying signal scenario. Like numbered nodes indicate that the support has remained the same over the transition. A sliding window of size $w = 2$ is depicted and the pruning action has indicated that the maximum track score belongs to the route represented by the tuple $(\mathbf{e}_6, \mathbf{e}_2, \mathbf{e}_2, \mathbf{e}_1)$. Thus, the pruning is done as shown. This has reduced the computation for the time step $k+1$ by 50%.

$\mathcal{T}_k$ for $k = 1, 2, \ldots$, to have a fixed cardinality equal to the cardinality of the set $\Omega_s$, i.e., $|\mathcal{T}_k| = N^s$.

To pursue this, consider (2.3). Define $\alpha_{\boldsymbol{\tau},\mathbf{d},H_k}$ to be

$$\alpha_{\boldsymbol{\tau},\mathbf{d},H_k} = \frac{P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}(\mathbf{d}|\boldsymbol{\tau}(2),H_k)\boldsymbol{\tau}(3)}{\sum_{\boldsymbol{\tau}\in\mathcal{T}_{k,\mathbf{d}}} P_{\mathbf{d}_k|\mathbf{d}_{k-1},H_k}(\mathbf{d}|\boldsymbol{\tau}(2),H_k)\boldsymbol{\tau}(3)}.$$

Then, equation (2.3) simplifies to

$$f_{\mathbf{v}_k|\mathbf{d}_k,H_k}(\mathbf{v}|\mathbf{d},H_k) = \sum_{\boldsymbol{\tau}\in\mathcal{T}_{k,\mathbf{d}}} \alpha_{\boldsymbol{\tau},\mathbf{d},H_k} f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}(\mathbf{v}|\mathbf{d},\boldsymbol{\tau}(2),H_k).$$

This is the representation of a Gaussian mixture with weight values $\alpha_{\boldsymbol{\tau},\mathbf{d},H_k}$ and Gaussian components $f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}$. Given $\mathbf{d}_k = \mathbf{d}$ at time step $k$, the mean $\boldsymbol{\mu}_{\mathbf{d}}$ of this mixture will be equal to

$$\boldsymbol{\mu}_{\mathbf{d}} = \sum_{\boldsymbol{\tau}\in\mathcal{T}_{k,\mathbf{d}}} \alpha_{\boldsymbol{\tau},\mathbf{d},H_k} \int_{\mathbf{v}} \mathbf{v} f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}(\mathbf{v}|\mathbf{d},\boldsymbol{\tau}(2),H_k)d\mathbf{v} = \sum_{\boldsymbol{\tau}\in\mathcal{T}_{k,\mathbf{d}}} \alpha_{\boldsymbol{\tau},\mathbf{d},H_k}\boldsymbol{\mu}_{\mathbf{d}|\boldsymbol{\tau}(2)}.$$

Also, the covariance matrix $\mathbf{C_d}$ of this mixture can be computed in the following way:

$$\begin{aligned}\mathbf{C_d} &= \mathbf{E}\left[(\mathbf{v}_k|\mathbf{d}, H_k)(\mathbf{v}_k|\mathbf{d}, H_k)'\right] - \boldsymbol{\mu}_\mathbf{d}\boldsymbol{\mu}_\mathbf{d}^{'} \\ &= \sum_{\boldsymbol{\tau}\in\mathcal{T}_{k,\mathbf{d}}} \alpha_{\boldsymbol{\tau},\mathbf{d},H_k} \int_{\mathbf{v}} \mathbf{v}\mathbf{v}' f_{\mathbf{v}_k|\mathbf{d}_k,\mathbf{d}_{k-1},H_k}(\mathbf{v}|\mathbf{d}, \boldsymbol{\tau}(2), H_k) d\mathbf{v} - \boldsymbol{\mu}_\mathbf{d}\boldsymbol{\mu}_\mathbf{d}^{'} \\ &= \sum_{\boldsymbol{\tau}\in\mathcal{T}_{k,\mathbf{d}}} \alpha_{\boldsymbol{\tau},\mathbf{d},H_k} \left(\mathbf{C}_{\mathbf{d}|\boldsymbol{\tau}(2)} + \boldsymbol{\mu}_{\mathbf{d}|\boldsymbol{\tau}(2)}\boldsymbol{\mu}_{\mathbf{d}|\boldsymbol{\tau}(2)}^{'}\right) - \boldsymbol{\mu}_\mathbf{d}\boldsymbol{\mu}_\mathbf{d}^{'}.\end{aligned}$$

Now, we approximate the Gaussian mixture density function with a regular Gaussian density function $\hat{f}_{\mathbf{v}_k|\mathbf{d}_k,H_k}(\cdot|\mathbf{d}, H_k)$ that has the mean $\boldsymbol{\mu}_\mathbf{d}$ and the covariance matrix $\mathbf{C_d}$. We apply this approximation to all the support candidates in $\Omega_s$.

To further decrease the computation volume, we first compute $P_{\mathbf{d}_k|H_k}(\mathbf{d}|H_k)$ for all possible support values $\mathbf{d} \in \Omega_s$ using (2.4). We then prune this probability function by keeping a certain number $h$ of the weights with the highest probability values in $P_{\mathbf{d}_k|H_k}$ and forcing the rest to be zero. Adjusting the nonzero values to sum to one will result in having an approximate probability function $\hat{P}_{\mathbf{d}_k|H_k}$ to use as the prior at time step $k+1$.

This means that at each time step, we will only have $h$ initial nodes instead of $N^s$ nodes. Therefore, at time step $k$, instead of using $N^s q^k$ Kalman filters for the scenario explained above, only $hq$ Kalman filters are required to update the belief state. This will decrease the amount of computation significantly.

As mentioned earlier, the ideas used in Algorithm 2 are motivated by a target tracking heuristic algorithm known as JPDA. We refer the interested reader to [46] and [47] to learn more about this heuristic.

## 2.3 Rollout

In both problems introduced in Section 1, we want to make optimal decisions at each time step (based on appropriate criteria) by either selecting the measurement matrix $\mathbf{A}_k$ or choosing the number of measurements $l_k$, given the history $H_k$. In Section 2.2, we formulated these problems as POMDPs. In this section, we describe our solution approach based on

an approximation method called *rollout*. Our description relies heavily on well established ideas and terminology from POMDP theory, which are readily available in [34] and [35].

In principle, the solution to a POMDP problem is, at each time $k$, a mapping that takes the history $H_k$ and gives an optimal action from $\mathcal{A}$. However, POMDP theory establishes that is suffices to find an optimal mapping $\pi_k^* : \mathcal{B} \to \mathcal{A}$ for $k = 1, \ldots, m$, where $\mathcal{B}$ is the set of distributions over the state space $\mathcal{S}$, and $\mathcal{A}$ is the actions space. If all actions are chosen using these optimal mappings, then at time $k = m$, the predefined objective function is optimized and the resulting *optimal policy* $\pi^* = \{\pi_1^*, \ldots, \pi_m^*\}$ has been generated.

We refer to the objective function as the *expected cumulative cost*. Given a policy $\pi = \{\pi_1, \ldots, \pi_m\}$ the expected cumulative cost is defined as

$$V_m^{\pi}(\mathbf{b}_1) = \mathbf{E}\left[\sum_{k=1}^{m} C_k(\mathbf{b}_k, \pi_k(\mathbf{b}_k)) \middle| \mathbf{b}_1\right]. \tag{2.6}$$

Subsequently, the optimal objective function value $V_m^{\pi^*}$ is achieved when $\pi = \pi^*$ in (2.6).

By applying *Bellman's principle* to (2.6), the optimal objective function $V_m^{\pi^*}$ and the optimal policy $\pi^*$ can be characterized by the following two equations:

$$V_m^{\pi^*}(\mathbf{b}_1) = \min_{\mathbf{u}}(C_1(\mathbf{b}_1, \mathbf{u}) + \mathbf{E}[V_{m-1}^{\pi^*}(\mathbf{b}_2)|\mathbf{b}_1, \mathbf{u}]), \tag{2.7}$$

and

$$\pi_1^*(\mathbf{b}_1) = \arg\min_{\mathbf{u}}(C_1(\mathbf{b}_1, \mathbf{u}) + \mathbf{E}[V_{m-1}^{\pi^*}(\mathbf{b}_2)|\mathbf{b}_1, \mathbf{u}]). \tag{2.8}$$

Let

$$Q_{m-k}(\mathbf{b}_k, \mathbf{u}) = C_k(\mathbf{b}_k, \mathbf{u}) + \mathbf{E}[V_{m-k}^{\pi^*}(\mathbf{b}_{k+1})|\mathbf{b}_k, \mathbf{u}] \tag{2.9}$$

be the *Q-value* of taking action $\mathbf{u}$ at belief state $\mathbf{b}_k$. Combining (2.8) and (2.9), we can now view the optimal action at time $k$ as

$$\pi_k^*(\mathbf{b}_k) = \arg\min_{\mathbf{u}} Q_{m-k}(\mathbf{b}_k, \mathbf{u}).$$

In other words, the optimal action can be found at any time step $k$ by identifying the action with the minimum $Q$-value at belief state $\mathbf{b}_k$.

The two summands in (2.9), $C_k(\mathbf{b}_k, \mathbf{u})$ and $\mathbf{E}[V^{\pi^*}_{m-k}(\mathbf{b}_{k+1})|\mathbf{b}_k, \mathbf{u}]$, are the immediate cost incurred (by $\mathbf{u}$) and the *expected cost-to-go* (ECTG) at the belief state $\mathbf{b}_k$, respectively. This breakdown inspires a natural separation in action selection approaches, namely *myopic* and *non-myopic*. Myopic action selection only considers the immediate impact (cost) of the action $\mathbf{u}$ at belief state $\mathbf{b}_k$. These approximations yield a *myopic* policy $\hat{\pi}$ which selects actions by

$$\hat{\pi}_k(\mathbf{b}_k) = \arg\min_{\mathbf{u}} C_k(\mathbf{b}_k, \mathbf{u}).$$

In relation to the $Q$-values, the myopic policy can be viewed as replacing the ECTG with a fixed constant, which can be assumed to be zero without loss of generality. In contrast, non-myopic methods attempt to approximate the ECTG term of the $Q$-value. If the estimation of the ECTG is suitable, it will evince the impact of the current action on the future costs to be incurred. There are several approaches for $Q$-value approximation [49]. We choose to implement the *rollout* method.

Rollout replaces the optimal policy used in the ECTG with a so called *base policy* $\pi^{\text{base}}$, creating the following $Q$-value approximation:

$$\tilde{Q}_{m-k}(\mathbf{b}_k, \mathbf{u}) = C_k(\mathbf{b}_k, \mathbf{u}) + \mathbf{E}[V^{\pi^{\text{base}}}_{m-k}(\mathbf{b}_{k+1})|\mathbf{b}_k, \mathbf{u}].$$

This approximation differentiates the effects of the actions not only at the current time step but over the future horizon, providing *multi-step lookahead* as opposed to the *1-step lookahead* of the myopic policy. It should also be noted that the simulation results presented in the next section produced by rollout exploit *receding horizon control*, replacing the remaining horizon $m-k$ with a fixed value $\vartheta$. This "artificial" horizon is regarded as the *rollout horizon*; rollout defines a $\vartheta$-lookahead policy.

The technical details justifying both receding horizon control and rollout can be found in [49], but we provide the reader with a meaningful result which motivates the use of rollout. By ranking actions over time with their approximate $Q$-values, rollout produces a

policy $\tilde{\pi} = \{\tilde{\pi}_1, \tilde{\pi}_2, \ldots, \tilde{\pi}_m\}$, where

$$\tilde{\pi}_k(\mathbf{b}_k) = \arg\min_{\mathbf{u}} \tilde{Q}_\vartheta(\mathbf{b}_k, \mathbf{u}).$$

The policy $\tilde{\pi}$ is guaranteed to perform at least as well as the base policy $\pi^{\text{base}}$ with respect to the objective function. In fact, it is common for rollout to out-perform its base policy. This is related to the fundamental relation of rollout to policy improvement (see [52]).

There are obvious trade-offs between myopic and non-myopic policies. The clear benefit of myopic policies is the relatively simple computations of $C_k(\mathbf{b}_k, \mathbf{u})$ required for their *greedy* action choices, based solely on the immediate cost of an action at the current belief state. Often, this alone motivates the implementation of the myopic policy. Conversely, non-myopic approaches must contend with the computational complexity originating from the inclusion of the ECTG term. In particular, rollout must compute the expected cumulative cost of the chosen base policy over $\vartheta - 1$ time steps, and this computation is typically done using Monte Carlo sampling. However, given a scenario where the current action choice substantially impacts the future outlook, rollout provides a distinct advantage due to its lookahead property. The results in the following section empirically compare and contrast the greedy-type policies with rollout.

## 2.4 Simulation Results

In this section, we present numerical examples for the problems introduced in Section 1. To better demonstrate the value of adaptivity and in particular, multi-step lookahead policies, we consider several settings in our simulations. Since each simulation contains a significant amount of details about its settings, we briefly explain the goals and the results of running each simulation.

Our first simulation, Simulation 1, considers a very simple case of a steady state 1-sparse signal with a highly informative prior distribution of the location of the nonzero entry. We find solutions for Problem 1 and evaluate the performance of adaptive methods compared to non-adaptive methods. The results of this simulation indicate, even in such a simple

scenario, the proposed adaptive methods provide a better performance than the traditional non-adaptive methods. However, the various adaptive methods, i.e., 1-step and multi-step lookahead, perform similarly. In the remaining simulations, we primarily provide results for rollout compared to alternative adaptive methods.

In Simulation 2, an instance of Problem 1, there is a single moving target, i.e., a dynamic 1-sparse signal, that changes its location in an environment with occlusions. When the target occupies a position with an occlusion, then the measurements are just noise samples. We assume a uniform prior distribution on the supports, which is less informative than the prior in Simulation 1. These results again show little-to-no difference between the performance of 1-step and multi-step lookahead policies. Even with the augmentation of the problem with occlusions and dynamics, multi-step lookahead does not amass extra *useful* information compared to the 1-step lookahead solution. This relates to the POMDP cost and action space definitions. The simulations based on Problem 2 explicate this further.

In Simulation 3, we use the same settings of Simulation 2 but we consider solving Problem 2 instead, using a different POMDP action space and cost. By using this cost, each method must decide between using more measurements or a reduction in support identification accuracy at each time step. The results of this simulation show better performance for multi-step lookahead compared to the 1-step lookahead.

Simulation 4 is an extension to the general case of having multiple dynamic targets. This simulation considers two moving targets in a space with occlusions and assumes semi-informative prior distribution on the supports. Algorithms 1 and 2 from Section 2.2.2 are applied to maintain reduced computations. The conclusions from Simulation 3 are still valid, i.e., depending on the performance criteria and the action space of the problem, the multi-step lookahead has better performance than the 1-step lookahead.

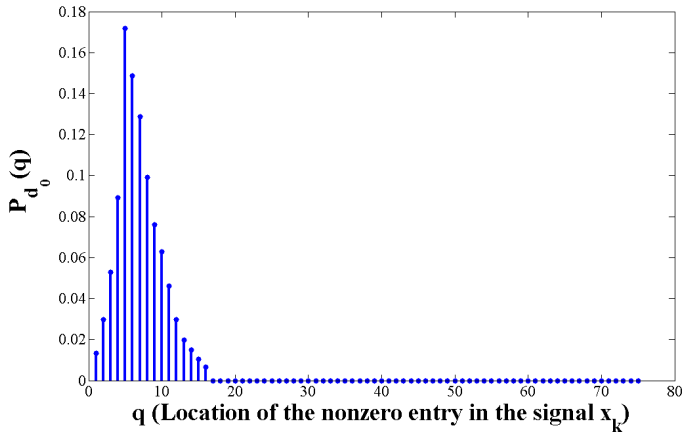



**Figure 2.3:** Prior structure used for $P_{\mathbf{d}_0}$ in Simulation 1.

### 2.4.1 Simulation 1: 1-Sparse Static Signal (Problem 1)

In this simulation, we consider a simple static scenario. We assume that the signal $\mathbf{x}_k$ is 1-sparse in $\mathbb{R}^{75}$ and that its nonzero entry stays in the same location at all times. We further assume a specific prior distribution on the support of $\mathbf{x}_k$. The probability mass function (pmf) $P_{\mathbf{d}_0}$ corresponding to this prior is shown in Fig. 2.3. The prior indicates the nonzero entry of the signal is located somewhere in the first 16 indices of the signal. Our simulations utilize 100 signal samples and rerun the experiment 50 times for each sample. Each signal sample is created by drawing a sample from the prior pmf $P_{\mathbf{d}_0}$ for the locations of the target and then drawing samples from a $\mathcal{N}(0, \sigma^2)$ distribution for the amplitudes.

We consider an instance of Problem 1 in this simulation, where the number $l_k$ of measurements per step is set to one, and the total time steps (total number of measurements) is $m = 8$. We compare five different (three non-adaptive and two adaptive) methods:

**1. Random:** In this method, the matrix $\mathbf{A}_k$ is a matrix where all its $(l_k \times N)$ entries are i.i.d. samples from a Gaussian distribution $\mathcal{N}(0, 1/N)$. We also normalize the rows of each matrix $\mathbf{A}_k$.

**2. Limited Random:** Knowing the prior $P_{\mathbf{d}_0}$, when assembling rows of the measurement matrix $\mathbf{A}_k$, we divide each measurement row $\mathbf{a}_k(i)$ into two parts: the first part is a 16-dimensional vector, and the second part is a 59-dimensional vector. We fill the first 16

entries with i.i.d. random Gaussian samples and normalize the resulting vector to have norm one. We fill the second part with zeros.

**3. Random from Library:** Similar to Limited Random, each row of the measurement matrix $\mathbf{A}_k$ is broken into two parts, where the second part is a 59-dimensional vector of zeros. However, the first part of the rows are chosen randomly from a static library of 50 measurement vectors that together, constitute a Grassmannian line packing (see [53] and [54]) in $\mathbb{R}^{16}$.

**4. Greedy:** In this approach, we also break the measurements rows into two parts like methods 2 and 3 and we determine the first 16-dimensional part of the vector. As a variation of our POMDP, we set the decision-making horizon to 1, i.e., at each time step $k$, we choose the best vector from the Grassmannian library that minimizes the one-step ahead belief cost $C_k(\mathbf{b}_k, \mathbf{u}_k)$. In other words, the actions are chosen in a *greedy* manner.

**5. Rollout:** This solution method is also based on the POMDP, in which at each time step and for each action candidate from the Grassmannian library, the $Q$-value is approximated over a four step horizon. The estimated $Q$-value, for each candidate action, is the average 50 (four step) $Q$-value samples. The action with the minimum $Q$-value approximation is selected. The base policy for Rollout is the Random from Library method.

Fig. 2.4 shows the performance of the five methods introduced above. The metric used in this figure for comparing these methods is the posterior probability of the true support after $m = 8$ measurements, i.e., the value of $P_{\mathbf{d}_8|H_8}(\mathbf{d}_T|H_8)$, where $\mathbf{d}_T$ is the true location of the nonzero entry of the signal. We have shown the performance of these methods for different values of signal-to-noise-ratio (SNR), which is defined as $\text{SNR} = \sigma^2/\sigma_w^2$. This figure shows that variations of POMDP, i.e., Greedy and Rollout, perform similar to each other in this simple static scenario, but both perform better than the three non-adaptive methods at moderate and high SNR.

A second experiment is run to see, on average, how many more measurements Random and Limited Random require in order to reach the performance of Greedy when the same

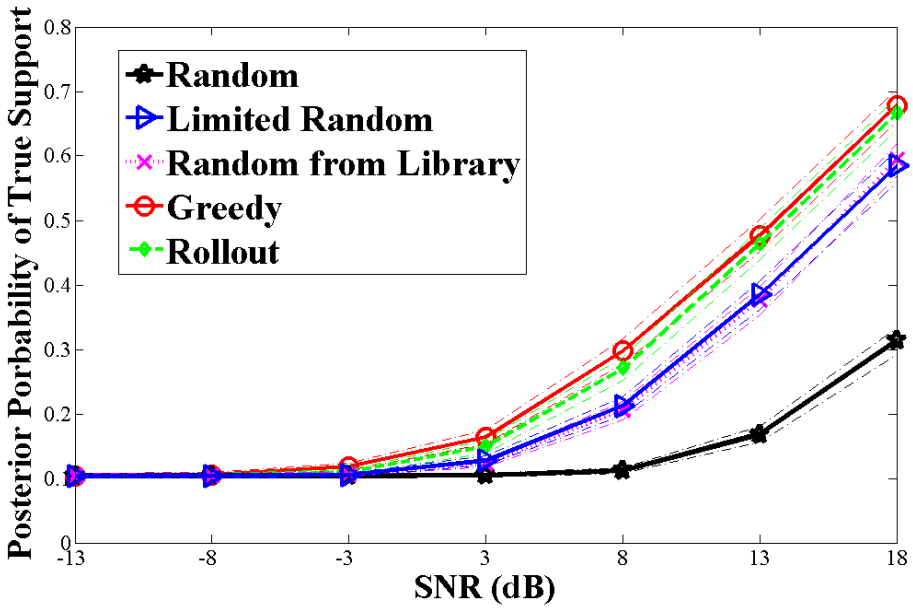



**Figure 2.4:** Performance comparison of all methods in Simulation 1. The dashed lines indicate the 95% confidence intervals.

metric, $P_{\mathbf{d}_8|H_8}(\mathbf{d}_T|H_8)$, is used for comparison. Fig. 2.5 shows the results for different values of SNR. The number of measurements for Greedy is set to $m = 8$ for all SNRs. The plots suggest that in this simple static scenario with the particular prior used, simply knowing that the true support lies within the first 16 indices provides significant performance gains over the random scheme. Moreover, adaptation only provides marginal gains over methods that exploit the highly informative prior.

We anticipate similar conclusions in the context of Problem 2. That is, due to simple and static nature of the problem, multi-step lookahead does not offer much advantage over the single-step lookahead. Also, we anticipate that adaptivity would again offer minimal gain over the Limited Random method, which exploits the highly informative prior used in this scenario.

### 2.4.2 Simulation 2: 1-Sparse Time-Varying Signal (Problem 1 with Occlusions)

We consider a 1-sparse dynamic signal in $\mathbb{R}^{20}$ for this simulation. Differing from Simulation 1, we assume a uniform prior for $P_{\mathbf{d}_0}$ and the presence of occlusions in certain areas.

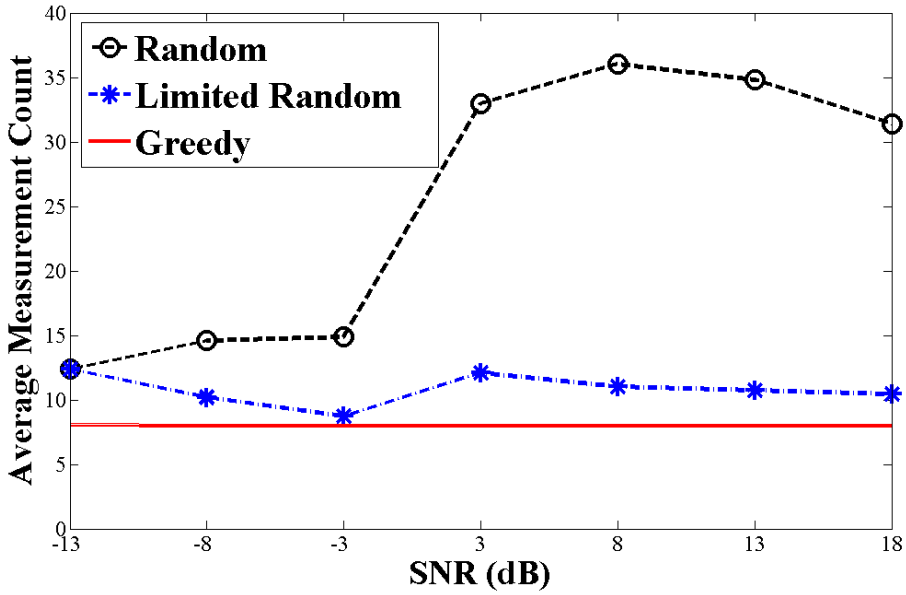



**Figure 2.5:** Average number of measurements required for Random and Limited Random to reach the performance of Greedy in Simulation 1.

When the signal support corresponds to occluded locations, the received observations are simply noise samples, i.e., $\mathbf{y}_k = \mathbf{w}_k$. The occlusions for Simulation 2 are located at indices 5, 6, 11 and 12. The movement of the target is modeled by the transition law shown in Fig. 2.6. While this transition law is used as the model, the initial position of the true target is located at the second index and the target moves one position to the right until reaching index 20. At this point, the target begins shifting in the reverse direction back to position one. Finally, the support would revert to its original transition procedure. The horizon of Simulation 2 is 40 time steps so the boundary condition will be met twice, reversing the targets motion. Furthermore, the target will be occluded at times k={4, 5, 10, 11, 27, 28, 33, 34}. Finally, we set SNR = 14 (dB) and run the experiment 36 times for both the Greedy and Rollout and 300 times for the both the Random from Library and the Bhattacharyya Distance algorithms, which we define below.

Similar to Simulation 1, the scenario of interest is based on Problem 1 where $l_k = 5$. The policies compared in this set of simulations consists of one non-adaptive and three adaptive schemes. The action selection is again among a set of measurement matrices. To build this

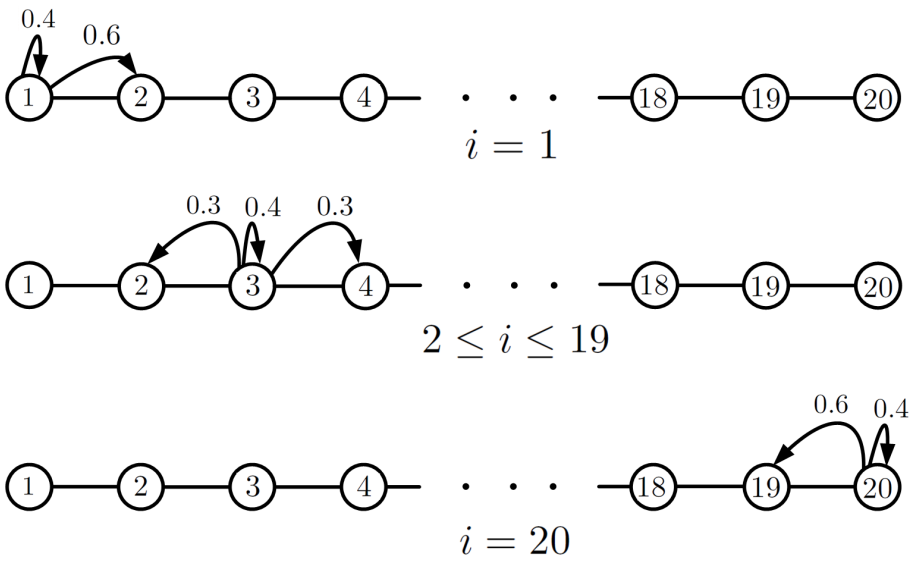



**Figure 2.6:** State transition law for the moving target.

set, we first import a Grassmannian packing consisting of 16 vectors in $\mathbb{R}^{16}$. The 16 vectors that build this Grassmannian packing are then combined into different sets consisting of five vectors. Note that we are considering every possible combination (without repeats) of 5 vectors among 16 vectors, resulting 4368 sets of vectors. Using the vectors in each set, we build a $(5 \times 16)$ matrix. These matrices are then augmented by inserting four columns of zeros in the respective occlusion positions. These 4368 matrices now comprise the library of compressive measurement matrices.

Each matrix in the library is accompanied by a power-vector which will help us "classify" the matrix. The power-vector $\boldsymbol{\phi}_{\mathbf{A}}$ of a matrix $\mathbf{A}$ is a $(20 \times 1)$ vector where its $j$th component $\boldsymbol{\phi}_{\mathbf{A}}(j)$ is defined as

$$\boldsymbol{\phi}_{\mathbf{A}}(j) = \frac{\sum_{i=1}^{5} |\mathbf{A}(i)(j)|}{\sum_{j=1}^{20} \sum_{i=1}^{5} |\mathbf{A}(i)(j)|}.$$

Note that $\mathbf{A}(i)(j)$ is an entry of the matrix $\mathbf{A}$ located at its $i$th row and its $j$th column. These power-vectors portray how the power of the matrix is "distributed" over its different columns and resemble probability mass functions. The matrix library is then sorted into 400 clusters using the K-medoids algorithm (see [55]). In this clustering algorithm the Bhattacharyya distance is used to classify the measurement matrices via their corresponding

power-vectors. Once the power-vectors, and thus the matrices, are classified, each cluster contains a *medoid* as a representative member. Then the available actions at time step $k$ are identified as the cluster ${\mathcal{A}_{\mathcal{G}}}^{(k)}$, whose medoid achieves the minimum Bhattacharyya distance to $\mathcal{P} = \mathbf{T} \times P_{\mathbf{d}_{k-1}|H_{k-1}}$, where $\mathbf{T}$ is the transition probability matrix implied in Fig. 2.6. These action restrictions are implemented for the adaptive methods described below.

**1. Random from Library:** This is the only non-adaptive scheme considered in this set of simulations. It does not abide by the action restrictions of available measurement matrices. Simply, a single measurement matrix is chosen from the library of 4368 matrices at random.

**2. Bhattacharyya Distance:** This is an adaptive heuristic inspired by the findings in [26] where the measurement should resemble the prior. This is really an extension of the sorting procedure used for clustering measurement matrix library ${\mathcal{A}_{\mathcal{G}}}^{(k)}$. Once the closest medoid of the 400 groups is identified, the Bhattacharyya distance is again used to rank the measurement matrices in the minimum medoid's cluster. The action selected is the measurement matrix with the minimum distance from its power-vector to the predicted support distribution $\mathcal{P}$.

**3. Greedy:** This action selection process is based on the POMDP with a *single* step decision-making horizon. Greedy, as with the Bhattacharyya distance, only considers the available actions based on the medoid selection. At each time step $k$, the best matrix is chosen from the restriced measurement matrix library ${\mathcal{A}_{\mathcal{G}}}^{(k)}$ minimizing the one-step ahead belief cost.

**4. Rollout:** This solution method, also based on the POMDP, selects an action, $u_k \in {\mathcal{A}_{\mathcal{G}}}^{(k)}$, which minimizes the 3-step lookahead $Q$-value approximation. The base policy used in this approximation is the Bhattacharyya Distance (method 2).

Fig. 2.7 shows the performance of each method described above over 40 time steps. As expected, the three adaptive methods out-perform the non-adaptive method (except in select occlusion areas). However, when comparing Greedy and Rollout, we see similar

performances. This simulation shows an example of a case where the multi-step lookahead approach is not gaining extra information compared to the myopic method.

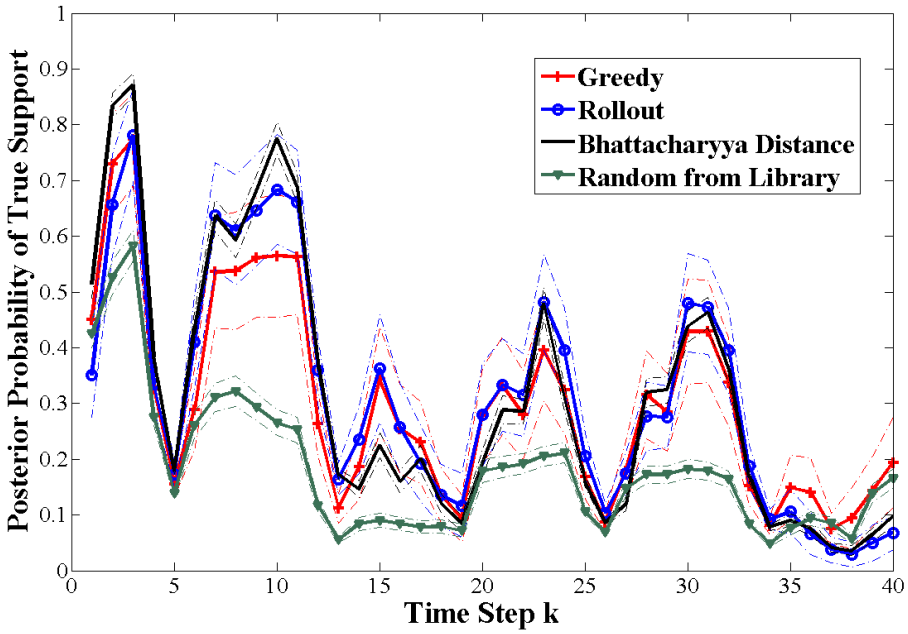



**Figure 2.7:** Performance comparison of the four policies described in Simulation 2. The dashed lines indicate the 85% confidence intervals.

### 2.4.3 Simulation 3: 1-Sparse Time-Varying Signal (Problem 2 with Occlusions)

We consider another scenario involving a moving target with occlusions. Here, a single target moves among $N = 40$ possible locations. The transition law that describes the movement of this target is shown in Fig. 2.6. Like Simulation 2, we assume a uniform distribution as the prior $P_{\mathbf{d}_0}$, but the occlusions are now located at indices 6 to 10, 16 to 20, and 36 to 40. For simplicity, we consider the case where the target starts in location 2 and moves one location to the right at each time step. This means that over the duration of $m = 10$ time steps, the target will be behind occlusion points from time step 5 to 9. We set $\gamma$ used in (2.1) to $\gamma = 300$ and we set SNR $= 2.5$ (dB). Finally, in this simulation, we use one signal sample and repeat the experiment 50 times.

We consider this scenario in the context of Problem 2. Here, we adaptively select the

number of measurements taken at each time step. We set $l_{\max} = 65$, meaning at most 65 measurements are requested per time step. After the number of measurements $l_k$ is determined, we use a fixed scheme to design the measurement matrix $\mathbf{A}_k$. The following describes the scheme used in this simulation: At each time $k$ the $j$th entry of the measurement row $\mathbf{A}_k(i)$, i.e., $\mathbf{A}_k(i)(j)$, is generated in the following way:

$$\mathbf{A}_k(i)(j) = \begin{cases} 0, & j \text{ is an occlusion point,} \\ x_{P_{\mathbf{d}_k|H_k}(j|H_k)}, & \text{otherwise,} \end{cases}$$

where the value $x_{P_{\mathbf{d}_k|H_k}(j|H_k)}$ is a sample drawn from the normal distribution $\mathcal{N}(0, P_{\mathbf{d}_k|H_k}(j|H_k))$. Each row vector $\mathbf{A}_k(i)$ is then normalized. The idea of using the function $P_{\mathbf{d}_k|H_k}$ to create row entries in the measurement matrix $\mathbf{A}_k$ comes from [26], one of the early works in adaptive design (1-step) of compressive measurement matrices.

In this simulation, we consider three methods of action selection:

**1. Greedy:** The best number of measurements which minimizes the 1-step lookahead POMDP belief cost.

**2. Smart:** A specific number of measurements $l_k \leq l_{\max}$ is selected using the following rule:

$$l_k = \begin{cases} 0, & 0.95 < \alpha, \\ 5, & 0.8 < \alpha \leq 0.95, \\ 15, & 0.65 < \alpha \leq 0.8, \\ 45, & 0.25 < \alpha \leq 0.65, \\ 35, & 0.15 < \alpha \leq 0.25, \\ 25, & \alpha \leq 0.15, \end{cases} \tag{2.10}$$

where $\alpha$ is the fraction of $P_{\mathbf{d}_k|H_k}$ at time step $k$ that belongs to occlusion points. This policy suggests that when $\alpha$ implies the target is somewhere far from the occlusion points with high probability, a fair (but not exorbitant) number of measurements is used. When the target approaches the occlusion points, the number of measurements is increased to achieve an accurate estimate of the target location before it "disappears".Once the target is occluded,

there is no point in making any more measurements until the target becomes visible again.

**3. Rollout:** This is a 3-step lookahead solution method for the POMDP. Rollout chooses the action which minimizes the total (approximate) cost incurred over three consecutive time steps. For this method, the Smart policy is the base policy of Rollout. In each run, 150 Rollout trajectories are averaged to estimate the $Q$-value of each action candidate.

Fig. 2.8 shows the expected cumulative cost of the above three methods. This figure shows that the Rollout method outperforms the Greedy method, and demonstrates the value of multi-step lookahead. Moreover, it is interesting to see that both Greedy and Smart methods perform similarly while the amount of computation required for the Smart is much less that for the Greedy method.

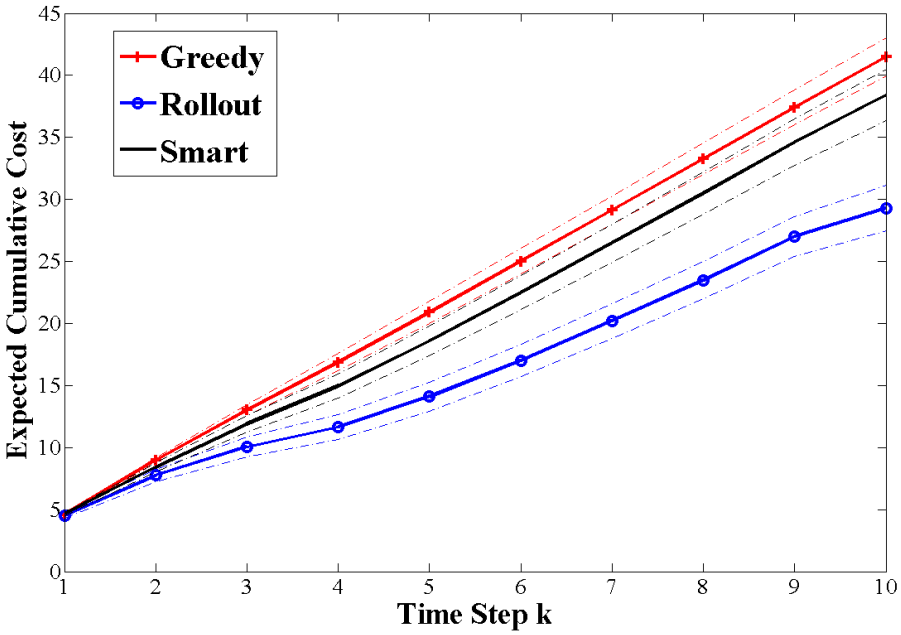



**Figure 2.8:** Performance comparison of all methods in Simulation 3 based on the expected cumulative cost at different time steps. The dashed lines indicate the 90% confidence intervals.

We have also plotted the posterior probability of true support, i.e., $P_{\mathbf{d}_k|H_k}(\mathbf{d}_T|H_k)$, at each time step $k$ in Fig. 2.9. This figure clearly captures the ability of each method tracking the moving target. When the target is outside the occlusion area, Rollout improves the

ability to detect the target location better than the other two methods over time.

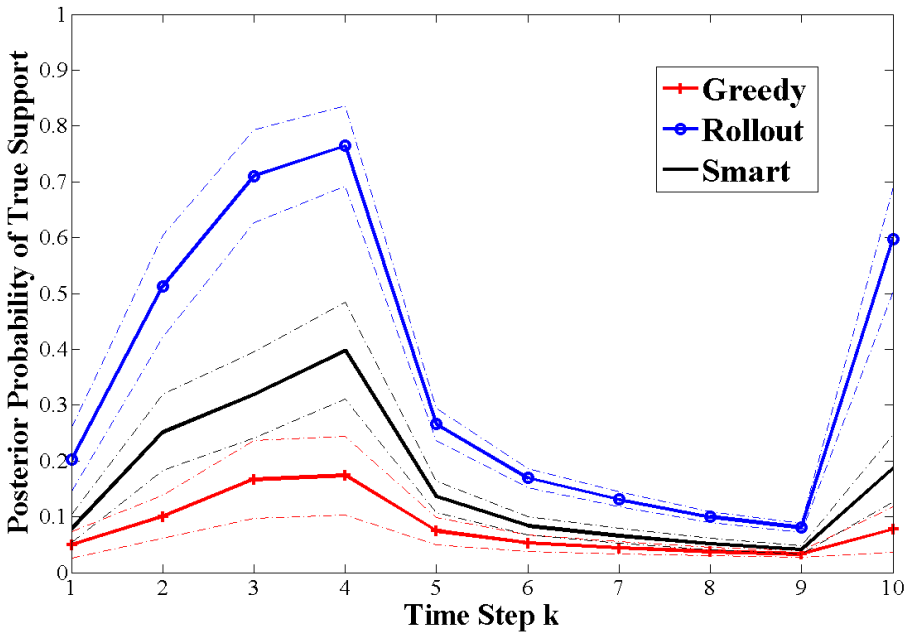



**Figure 2.9:** Performance comparison of all methods in Simulation 3 based on the posterior probability of the true support at different time steps. The dashed lines indicate the 90% confidence intervals.

### 2.4.4 Simulation 4: 2-Sparse Time-Varying Signal (Problem 2 with Occlusions)

Now, we consider a more general setting in which there is more than one moving target, i.e., a dynamic $s$-sparse signal. To keep the amount of computation low, we make several simplifying assumptions. First, we only consider $s = 2$ targets that are moving in $\mathbb{R}^{20}$. Note that even for this case, there are $N^s = 400$ possibilities for the support $\mathbf{d}_k$ at each time step. Second, we consider an initial distribution that only 25% of its entries are nonzero with equal values. This prior distribution suggests that the first target is located at one of the locations 1, 2, or 3 and the second target is located either at location 19 or 20. We let $\sigma_w^2 = 1$, and consider the strength values 1.7 and -1.5 for the first and second target, respectively, which means that SNR = 4 dB.

Targets 1 and 2 are initially located at positions 1 and 19 and as time evolves, they move towards each other one index at a time. The transition law that describes the movement of

each of these targets is the same transition law that was used in Simulation 3 and shown in Fig. 2.6. Note that these targets move independently from each other. Finally, we consider five occlusion points located at positions $4, 5, 8, 12$, and 13.

Our simulation runs for 10 time steps. During this time, the two targets pass through each other as well as the occluded locations. Table 2.1 summarizes the position of each target at each time step. This table indicates those time steps in which an occlusion occurs by a "$*$" sign next to the position of the target. Based on this table, we expect to see a drop in the performance of all the methods at time steps $3, 4, 6$ and in particular 7 (where both targets are occluded). Moreover, at time step 9, when the two targets reach the location 10, the strength value is equal to the sum of the strength values of the two targets. This means that at time $k = 9$, the measurements collected are from a 1-sparse signal with the strength value $1.7 - 1.5 = 0.2$. Thus, the SNR value drops to $-14$ dB and dip in performance is expected for all methods.

Similar to the previous simulations, the performance is compared across four approaches in this simulation:

**1. Fixed:** This is a non-adaptive method, where a fixed value of 30 measurements are taken at each time step.

**2. Smart:** This method is similar to Smart method in Simulation 3 but with two differences: a) The method has been modified to work for 2 targets, and b) The number of measurements for the different cases in (2.10) changes from 0, 5, 15, 45, 35, and 25 to 5, 10, 20, 55, 40, and 30, respectively.

**3. Greedy:** Similar to the myopic methods in previous simulations, this is the 1-step lookahead solution.

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{d}_k(1)$ | 2 | 3 | 4* | 5* | 6 | 7 | 8* | 9 | 10 | 11 |
| $\mathbf{d}_k(2)$ | 18 | 17 | 16 | 15 | 14 | 13* | 12* | 11 | 10 | 9 |

**Table 2.1:** Target positions over 10 decision epochs in Simulation 4.

**4. Rollout:** This method is the 3-step lookahead method and Smart, introduced here, is the base policy.

To estimate the expected cost in both Greedy and Rollout, we generate 25 samples from the belief state at each time step and we repeat the experiment for each trajectory 20 times.

In all of the techniques introduced, once the number of measurements $l_k$ is selected at each time step, we build the measurement matrix the same as in Simulation 3. Moreover, the maximum number of measurements allowed at each time step is $l_{\max} = 65$ and we set $\gamma$ to be equal to 293. The values shown in the results here are the average values taken over 36 rounds of simulation.

As mentioned in Section 2.2.2, we implement two heuristics, Algorithms 1 and 2, to decrease the amount of computation. In one experiment, Algorithm 1 limits the belief state distribution expansion by maintaining a hypothesis tree with a sliding window of size $w = 4$, resulting in approximately 6000 Kalman filters per time step, after the time step $k = 4$ (and significantly more before this first pruning action). This is clearly more Kalman filter computations than Algorithm 2, but yields in a higher probability of support identification. In a separate simulation, Algorithm 2 approximates the conditional posterior distribution of the support $P_{\mathbf{d}_k|H_k}$ with a distribution $\hat{P}_{\mathbf{d}_k|H_k}$ that contains only 90 nonzero entries for $k > 1$. Therefore, at any point in our simulation (after the first time step), the belief state update requires only 90 Kalman filters.

Figs. 2.10 and 2.11 show the results from running this simulation when Algorithm 1 is used. Similarly, Figs. 2.12 and 2.13 show the results from running this simulation when Algorithm 2 is used. In both set of results, similar to Simulation 3, Rollout has the best performance among all the methods. Also, notice the performance drop of all methods, as expected, in the occlusion and the low SNR areas shown in Table 2.1.

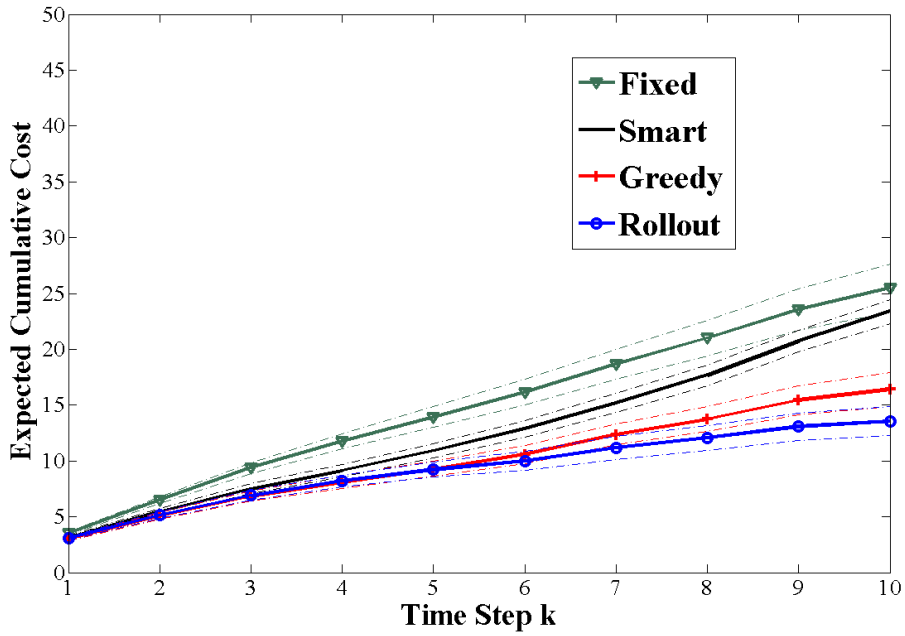



**Figure 2.10:** Performance comparison of all methods in Simulation 4 based on the expected cumulative cost at different time steps when Algorithm 1 is used. The dashed lines indicate the 85% confidence intervals.

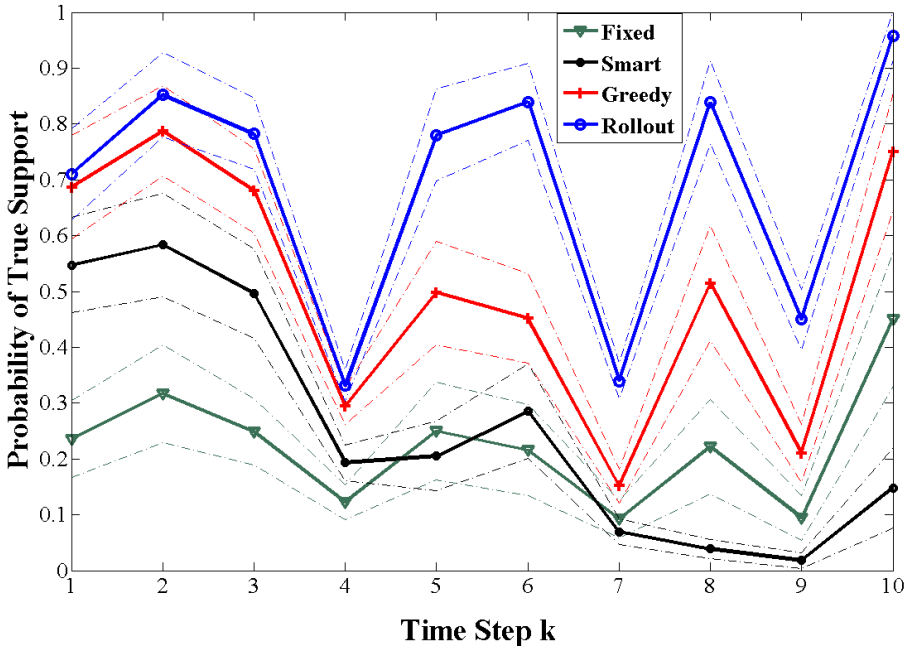



**Figure 2.11:** Performance comparison of all methods in Simulation 4 based on the posterior probability of the true support at different time steps when Algorithm 1 is used. The dashed lines indicate the 85% confidence intervals.

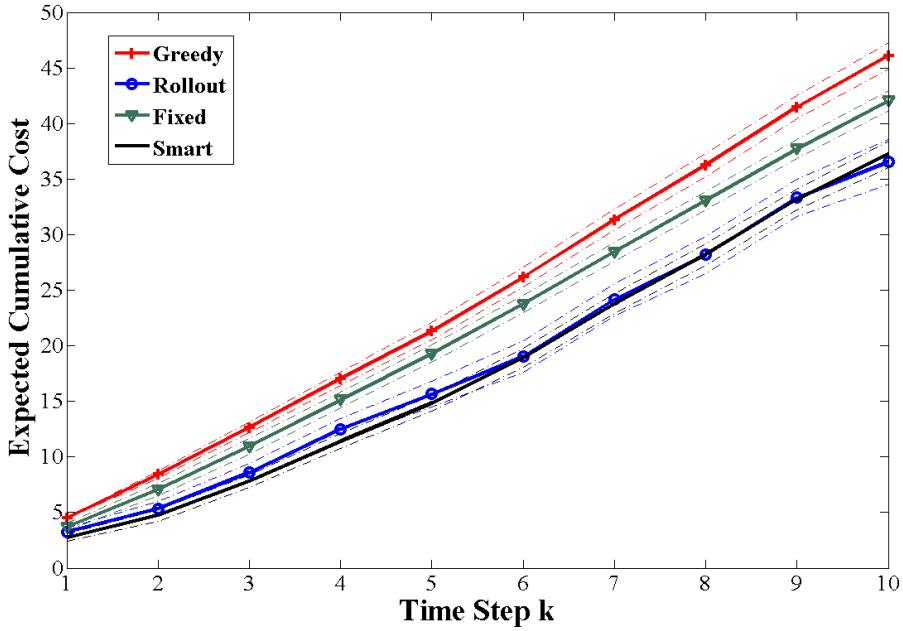



**Figure 2.12:** Performance comparison of all methods in Simulation 4 based on the expected cumulative cost at different time steps when Algorithm 2 is used. The dashed lines indicate the 85% confidence intervals.

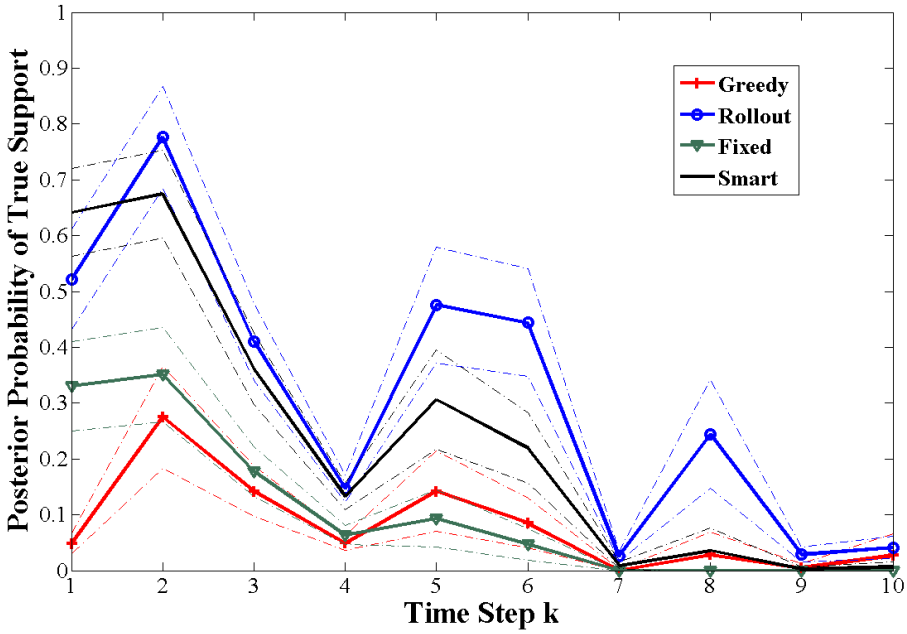



**Figure 2.13:** Performance comparison of all methods in Simulation 4 based on the posterior probability of the true support at different time steps when Algorithm 2 is used. The dashed lines indicate the 85% confidence intervals.

# CHAPTER 3

# MEASUREMENT DESIGN FOR DETECTING SPARSE SIGNALS

## 3.1 Introduction

We consider the design of low dimensional (compressive) measurement matrices, with a pre-specified number of measurements, for detecting sparse signals in additive white Gaussian noise. More specifically, we consider the following binary hypothesis test:

$$\begin{cases} \mathcal{H}_0 : \mathbf{x} = \mathbf{n}, \\ \mathcal{H}_1 : \mathbf{x} = \mathbf{s} + \mathbf{n}, \end{cases} \tag{3.1}$$

where $\mathbf{x}$ is an $(N \times 1)$ vector that describes the state of a physical phenomenon. Under the null hypothesis $\mathcal{H}_0$, $\mathbf{x}$ is a white Gaussian noise vector with covariance matrix $E[\mathbf{n}\mathbf{n}^H] = (\sigma_n^2/N)\mathbf{I}$. Under the alternative hypothesis $\mathcal{H}_1$, $\mathbf{x} = \mathbf{s} + \mathbf{n}$ consists of a deterministic signal $\mathbf{s}$ distorted by additive white Gaussian noise $\mathbf{n}$.

We assume $\mathbf{s}$ is $k$-sparse in a known basis $\mathbf{\Psi}$. That is, to say, $\mathbf{s}$ is composed as

$$\mathbf{s} = \mathbf{\Psi}\boldsymbol{\theta}, \tag{3.2}$$

where $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$ is a known matrix, whose columns form an orthonormal basis for $\mathbb{R}^N$, and $\boldsymbol{\theta} \in \mathbb{R}^N$ is a $k$-sparse vector, i.e., it has between 1 to $k \ll N$ nonzero entries. We may refer to $\mathbf{s}$ as simply $k$-sparse for brevity.

We wish to decide between the two hypotheses based on a given number $m \leq N$ of linear measurements $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$ from $\mathbf{x}$, where $\mathbf{\Phi} \in \mathbb{R}^{m \times N}$ is a compressive measurement matrix that we will design. The observation vector $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$ belongs to one of the following hypothesized

models:

$$\begin{cases} \mathcal{H}_0 : \mathbf{y} = \mathbf{\Phi n} \sim \mathcal{N}(\mathbf{0}, (\sigma_n^2/N)\mathbf{\Phi\Phi}^H), \\ \mathcal{H}_1 : \mathbf{y} = \mathbf{\Phi}(\mathbf{s}+\mathbf{n}) \sim \mathcal{N}(\mathbf{\Phi s}, (\sigma_n^2/N)\mathbf{\Phi\Phi}^H), \end{cases} \tag{3.3}$$

where the superscript $H$ is the Hermitian transpose. To avoid coloring the noise vector $\mathbf{n}$, we constraint the compressive measurement matrix $\mathbf{\Phi}$ to be right orthogonal, that is we force $\mathbf{\Phi\Phi}^H = \mathbf{I}$.

Rather than limiting ourselves to a particular detector, we look at the general problem of designing compressive measurements to maximize the measurement signal-to-noise ratio (SNR), under $\mathcal{H}_1$, which is given by

$$\text{SNR} = (\mathbf{s}^H\mathbf{\Phi}^H\mathbf{\Phi s})/(\sigma_n^2/N). \tag{3.4}$$

This is motivated by the fact that for the class of linear log-likelihood ratio detectors, where the log-likelihood ratio is a linear function of the data, the detection performance is improved by increasing SNR. In particular, for a Neyman-Pearson detector, with false alarm rate $\gamma$, the probability of detection $P_d = Q(Q^{-1}(\gamma) - \sqrt{\text{SNR}})$ is monotonically increasing in SNR, where $Q(\cdot)$ is the $Q$-function. In addition, maximizing SNR leads to maximum detection probability, at a pre-specified false alarm rate, when an energy detector is used. Without loss of generality, throughout this work we assume that $\sigma_n^2 = 1$ and $\|\mathbf{s}\|^2 = \|\boldsymbol{\theta}\|^2 = 1$, and so we design $\mathbf{\Phi}$ to maximize the measured signal energy $\|\mathbf{\Phi s}\|^2$.

In solving the problem, one approach is to assume a value for the sparsity level $k$ and design the measurement matrix $\mathbf{\Phi}$ based on this assumption. This approach, however, runs the risk that the true sparsity level might be different. An alternative approach is not to assume any specific sparsity level. Instead, when designing the measurement matrix $\mathbf{\Phi}$, we prioritize the level of importance of different values of sparsity $k$. In other words, we first find a set of solutions that are optimal for a $k_1$-sparse signal. Then, within this set, we find a subset of solutions that are also optimal for $k_2$-sparse signals. We follow this procedure until we find a subset that contains a family of optimal solutions for sparsity levels $k_1$, $k_2$, $k_3$, $\cdots$. This approach is known as a *lexicographic optimization* method (see, e.g., [56]–[58]).

Replacing (3.2) in (3.4) yields

$$\text{SNR} = \frac{\|\mathbf{\Phi\Psi\theta}\|^2}{(\sigma_n^2/N)}.$$

The basis matrix $\mathbf{\Psi}$ is known, but the $k$-sparse representation vector $\boldsymbol{\theta}$ is unknown. That is, the exact number of the nonzero entries in $\boldsymbol{\theta}$, their locations, and their values are unknown. The measurement design naturally depends on one's assumptions about the unknown vector $\boldsymbol{\theta}$. We consider two different design problems, namely a worst-case SNR design and an average SNR design, as explained below.

*Worst-case SNR design.* In the first case, we assume the vector $\boldsymbol{\theta}$ is deterministic but unknown. Then, among all possible deterministic $k$-sparse vectors $\boldsymbol{\theta}$, we consider the vector that minimizes the SNR and design the matrix $\mathbf{\Phi}$ that maximizes this minimum SNR. Of course, when minimizing the SNR with respect to $\boldsymbol{\theta}$, we have to find the minimum SNR with respect to locations and values of the nonzero entries in the vector $\boldsymbol{\theta}$. To combine this with the lexicographic approach, we design the matrix $\mathbf{\Phi}$ to maximize the *worst-case* detection SNR, where the worst-case is taken over all subsets of size $k_i$ of elements of $\boldsymbol{\theta}$, where $k_i$ is the sparsity level considered at the $i$th level of lexicographic optimization. This is a design for robustness with respect to the worst sparse signal that can be produced in the basis $\mathbf{\Psi}$. The reader is referred to Section 3.2 for a complete statement of the problem.

We show (see Section 3.3) that the worst-case detection SNR is maximized when the columns of the product $\mathbf{\Phi\Psi}$ between the compressive measurement matrix $\mathbf{\Phi}$ and the sparsity basis $\mathbf{\Psi}$ form a *uniform tight frame*. A uniform tight frame is a frame system in which the frame operator is a scalar multiple of the identity operator and every frame element has the same norm (see, e.g., [59]). We also show that when the signal is 2-sparse, the optimal frame is a *Grassmannian line packing* (see, e.g., [53]). For the case where the sparsity level of the signal is greater than two, we provide a lower bound on the worst-case performance. If the number $m$ of measurements allowed is greater than or equal to $\sqrt{N}$, then the Grassmannian line packing frame will be an equiangular uniform tight frame (see, e.g., [60]–[67]) and the maximal worst-case SNR can be expressed in terms of the Welch bound. Numerical examples

presented in Section 3.6 show that Grassmannian line packing frames provide better worst-case performance than matrices with i.i.d. Gaussian entries, which are typically used in sparse signal reconstruction.

*Average SNR design.* In the second case, we assume that the locations of nonzero entries of $\boldsymbol{\theta}$ are random but their values are deterministic and unknown. We find the matrix $\boldsymbol{\Phi}$ that maximizes the expected value of the minimum SNR. The expectation is taken with respect to a random index set with uniform distribution over the set of all possible subsets of size $k_i$ of the index set $\{1, 2, \ldots, N\}$ of elements of $\boldsymbol{\theta}$. The minimum SNR, whose expected value we wish to maximize, is calculated with respect to the values of the entries of the vector $\boldsymbol{\theta}$ for each realization of the random index set. The reader is referred to Section 3.4 for a complete statement of the problem.

We show (see Section 3.5) that for 1-sparse signals, any right orthogonal measurement matrix $\boldsymbol{\Phi}$, i.e., any tight frame, is optimal for maximizing the average minimum SNR. For signals with sparsity levels higher than one, we constrain ourselves to the class of uniform tight frames and show that optimal measurement matrix is a uniform tight frame that has minimal *sum-coherence*, as described in Section 3.5. However, to the best of our knowledge constructing such frames remains an open problem in frame theory. Therefore, we limit ourselves to providing performance bounds in the average-case problem.

## 3.2 The Worst-case Problem Statement

Since all sparse signals share the fact that they have at least one nonzero entry, it seems natural to first find an optimal measurement matrix for 1-sparse signals. Next, among the set of optimal solutions for this case, we find matrices that are optimal for 2-sparse signals. This procedure is continued for signals with higher sparsity levels. This is a lexicographic optimization approach to maximizing the worst-case SNR.

Consider the $k$th step of the lexicographic approach. In this step, the vector $\boldsymbol{\theta}$ has up to $k$ nonzero entries. We do not impose any prior constraints on the locations and the values

of the nonzero entries of $\boldsymbol{\theta}$. As mentioned earlier, we assume that $\|\mathbf{s}\|^2 = \|\boldsymbol{\theta}\|^2 = 1$ and $\sigma_n^2 = 1$. We wish to maximize the minimum (worst-case) SNR, produced by assigning the worst possible locations and values to the nonzero entries of the $k$-sparse vector $\boldsymbol{\theta}$. Referring to (3.4), this is a worst-case design for maximizing the signal energy $\mathbf{s}^H\boldsymbol{\Phi}^H\boldsymbol{\Phi}\mathbf{s}$ inside the subspace $\langle\boldsymbol{\Phi}^H\rangle$ spanned by the columns of $\boldsymbol{\Phi}^H$, since $\boldsymbol{\Phi}^H\boldsymbol{\Phi}$ is the orthogonal projection operator onto $\langle\boldsymbol{\Phi}^H\rangle$.

To define the $k$th step of the optimization procedure more precisely, we need some additional notation. Let $\mathcal{A}_0$ be the set containing all $(m \times N)$ right orthogonal matrices $\boldsymbol{\Phi}$. Then, we recursively define the set $\mathcal{A}_k$, $k = 1, 2, \ldots$, as the set of solutions to the following optimization problem:

$$\begin{aligned} \max_{\boldsymbol{\Phi}} \min_{\mathbf{s}} \quad & \|\boldsymbol{\Phi}\mathbf{s}\|^2, \\ \text{s.t.} \quad & \boldsymbol{\Phi} \in \mathcal{A}_{k-1}, \\ & \|\mathbf{s}\| = 1, \\ & \mathbf{s} \text{ is } k\text{-sparse.} \end{aligned} \tag{3.5}$$

In our lexicographic formulation, the optimization problem for the $k$th problem (3.5) involves a worst-case objective restricted to the set of solutions $\mathcal{A}_{k-1}$ from the $(k-1)$th problem. So, $\mathcal{A}_k \subset \mathcal{A}_{k-1} \subset \cdots \subset \mathcal{A}_0$.

Before we present a complete solution to these problems, we first simplify them in three steps. First, since the matrix $\boldsymbol{\Psi}$ is known, the matrix $\boldsymbol{\Phi}$ can be written as $\boldsymbol{\Phi} = \mathbf{C}\boldsymbol{\Psi}^H$, where $\mathbf{C}$ is an $(m \times N)$ matrix. Then, $\boldsymbol{\Phi}\boldsymbol{\Psi} = \mathbf{C}\boldsymbol{\Psi}^H\boldsymbol{\Psi} = \mathbf{C}$, and also $\boldsymbol{\Phi}\boldsymbol{\Phi}^H = \mathbf{C}\boldsymbol{\Psi}^H\boldsymbol{\Psi}\mathbf{C}^H = \mathbf{C}\mathbf{C}^H = \mathbf{I}$. Using (3.2), the max-min problems (3.5) become

$$\begin{aligned} \max_{\mathbf{C}} \min_{\boldsymbol{\theta}} \quad & \|\mathbf{C}\boldsymbol{\theta}\|^2, \\ \text{s.t.} \quad & \mathbf{C} \in \mathcal{B}_{k-1}, \\ & \|\boldsymbol{\theta}\| = 1, \\ & \boldsymbol{\theta} \text{ is } k\text{-sparse,} \end{aligned} \tag{3.6}$$

where $\mathcal{B}_0 = \mathcal{A}_0$, and similar to the sets $\mathcal{A}_k$, the sets $\mathcal{B}_k$ $(k = 1, 2, \ldots)$ are recursively defined

to contain all the optimal solutions of (3.6). It is easy to see that $\mathcal{B}_k = \{\mathbf{C} : \mathbf{C}\boldsymbol{\Psi}^H \in \mathcal{A}_k\}$.

Let $\Omega = \{1, 2, \ldots, N\}$ and define $\Omega_k$ to be $\Omega_k = \{E \subset \Omega : |E| = k\}$. For any $T \in \Omega_k$, let $\boldsymbol{\theta}_T$ be the subvector of size $(k \times 1)$ that contains all the components of $\boldsymbol{\theta}$ corresponding to indices in $T$. Similarly, given a matrix $\mathbf{C}$, let $\mathbf{C}_T$ be the $(m \times k)$ submatrix consisting of all columns of $\mathbf{C}$ whose indices are in $T$. Note that the vector $\boldsymbol{\theta}_T$ may have zero entries. Indeed, for cases where the $k$-sparse vector $\boldsymbol{\theta}$ has fewer than $k$, e.g., $l < k$, nonzero entries, the $(k \times 1)$ vector $\boldsymbol{\theta}_T$ has $k - l$ zero entries. This is important because our definition for $T$ and $\boldsymbol{\theta}_T$ is slightly different than the common definitions used in the compressed sensing literature, where $T$ and $\boldsymbol{\theta}_T$ only contain indices and values related to the nonzero entries of the vector $\boldsymbol{\theta}$, often called the support of $T$. We refer to a member $T$ of $\Omega_k$ as a "$k$-platform". Thus, a $k$-platform $T$ includes, but is not limited to, the support of the sparse vector $\boldsymbol{\theta}$.

Given $T \in \Omega_k$, the product $\mathbf{C}\boldsymbol{\theta}$ can be replaced by $\mathbf{C}_T\boldsymbol{\theta}_T$ instead. Now, to consider the worst-case scenario for the SNR, as well as considering the worst $\boldsymbol{\theta}_T$ that minimizes $\|\mathbf{C}_T\boldsymbol{\theta}_T\|^2$, we also have to consider the worst $T \in \Omega_k$. Thus, the max-min problem becomes

$$\begin{aligned} \max_{\mathbf{C}} \min_{T} \min_{\boldsymbol{\theta}_T} \quad & \|\mathbf{C}_T\boldsymbol{\theta}_T\|^2, \\ \text{s.t.} \quad & \mathbf{C} \in \mathcal{B}_{k-1}, \\ & \|\boldsymbol{\theta}_T\| = 1, T \in \Omega_k. \end{aligned} \tag{3.7}$$

The solution to (3.7) is the most robust design with respect to the locations and values of the nonzero entries of the parameter vector $\boldsymbol{\theta}$.

The solution to the minimization subproblem

$$\begin{aligned} \min_{\boldsymbol{\theta}_T} \quad & \|\mathbf{C}_T\boldsymbol{\theta}_T\|^2, \\ \text{s.t.} \quad & \|\boldsymbol{\theta}_T\| = 1, \end{aligned}$$

is well known; see, e.g., [68]. The optimal objective function is $\lambda_{\min}(\mathbf{C}_T^H\mathbf{C}_T)$, the smallest eigenvalue of the matrix $\mathbf{C}_T^H\mathbf{C}_T$. Therefore, the max-min-min problem (3.7) simplifies to

$$(\mathrm{P}_k) \quad \begin{cases} \max_{\mathbf{C}} \min_{T} \quad \lambda_{\min}(\mathbf{C}_T^H\mathbf{C}_T), \\ \text{s.t.} \quad \mathbf{C} \in \mathcal{B}_{k-1}, \\ \qquad\quad T \in \Omega_k. \end{cases} \tag{3.8}$$

At each step $k$, the optimal compressive measurement matrix, denoted by $\mathbf{\Phi}^*$, is determined from the optimizer $\mathbf{C}^*$ of (3.8) as $\mathbf{\Phi}^* = \mathbf{C}^*\mathbf{\Psi}^H$.

Next, we describe how to solve the max-min problem ($\mathrm{P}_k$) in (3.8).

## 3.3 Solution to the Worst-case Problem

Let $\mathbf{c}_i$ be the $i$th column of the matrix $\mathbf{C}$. We first find the solution set $\mathcal{A}_1$ for problem ($\mathrm{P}_1$). Then, we find a subset $\mathcal{A}_2 \subset \mathcal{A}_1$ as the solution for ($\mathrm{P}_2$). We continue this procedure for general sparsity level $k$.

### 3.3.1 Sparsity Level $k = 1$

If $k = 1$, then any $T$ such that $|T| = 1$ can be written as $T = \{i\}$ with $i \in \Omega$, and $\mathbf{C}_T = \mathbf{c}_i$ consists of only the $i$th column of $\mathbf{C}$. Therefore, $\mathbf{C}_T^H\mathbf{C}_T = \mathbf{c}_i^H\mathbf{c}_i = \|\mathbf{c}_i\|^2$, and the max-min problem becomes

$$\begin{aligned} \max_{\mathbf{C}} \min_{i} \quad & \|\mathbf{c}_i\|^2, \\ \text{s.t.} \quad & \mathbf{C} \in \mathcal{B}_0, \\ & i \in \Omega. \end{aligned} \tag{3.9}$$

**Theorem 1** *The optimal value of the objective function of the max-min problem (3.9) is $m/N$. A necessary and sufficient condition for a matrix $\mathbf{C}^*$ to be in the solution set $\mathcal{B}_1$ is that the columns $\{\mathbf{c}_i^*\}_{i=1}^N$ of $\mathbf{C}$ form a* uniform tight frame *with norm values equal to $\sqrt{m/N}$.*

*Proof:* We first prove the claim about the optimal value. Assume false, i.e., assume there exists an optimal matrix $\mathbf{C}^* \in \mathcal{B}_1$ for which the value of the cost function is either less than or greater than $m/N$. Suppose the former is true. Let $\mathbf{C}_1$ be an $(m \times N)$ matrix, satisfying $\mathbf{C}_1\mathbf{C}_1^H = \mathbf{I}$, whose columns have equal norm $\sqrt{m/N}$. Then, the value of the objective function in (3.9) for $\mathbf{C} = \mathbf{C}_1$ is $m/N$. This means that our proposed matrix $\mathbf{C}_1$ achieves a higher SNR than $\mathbf{C}^*$ which is a contradiction. Now, assume the latter is correct, that is the value of the objective function for $\mathbf{C}^*$ is greater than $m/N$. This means

$\min_{i\in\Omega} \|\mathbf{c}_i^*\|^2 = \|\mathbf{c}_j^*\|^2 > m/N$. Knowing this, we write

$$\mathrm{tr}\left(\mathbf{C}^*\mathbf{C}^{*H}\right) = \mathrm{tr}\left(\mathbf{C}^{*H}\mathbf{C}^*\right) = \sum_{i=1}^{N} \|\mathbf{c}_i^*\|^2 > \sum_{i=1}^{N} m/N = m.$$

However, from the constraint in (3.9) we know that $\mathbf{C}^*\mathbf{C}^{*H} = \mathbf{I}$, and $\mathrm{tr}(\mathbf{C}^*\mathbf{C}^{*H}) = m$. This is also a contradiction. Thus, the assumption is false and the optimal value for the objective function of (3.9) is $m/N$.

We now prove the claim about the optimizer $\mathbf{C}^*$. From the preceding part of the proof, it is easy to see that all columns of $\mathbf{C}^*$ must have equal norm $\sqrt{m/N}$. If not, since none of them can be less than $\sqrt{m/N}$, then the sum of all column norms will be greater than $m$, which is a contradiction. Moreover, we write

$$\mathbf{C}^*\mathbf{C}^{*H} = \sum_{i=1}^{N} \mathbf{c}_i^*\mathbf{c}_i^{*H} = \mathbf{I}. \tag{3.10}$$

Multiplying both sides of (3.10) by an arbitrary $(m \times 1)$ vector $\mathbf{x}$ from the right side and $\mathbf{x}^H$ from the left side, we get $\sum_{i=1}^{N} \|\mathbf{c}_i^{*H}\mathbf{x}\|^2 = \|\mathbf{x}\|^2$. This equation represents a tight frame with frame elements $\{\mathbf{c}_i^*\}$ and frame bound 1. In other words, it represents a Parseval frame. Since the frame elements have equal norms, the frame is also uniform. Therefore, for a matrix $\mathbf{C}^*$ to be in $\mathcal{B}_1$, the columns of $\mathbf{C}^*$ must form a *uniform tight frame*. ■

*Remark 1:* The reader is referred to [59], [63], [69], [70], and the references therein, for examples of constructions of uniform tight frames.

### 3.3.2 Sparsity Level $k = 2$

The next step is to solve $(\mathrm{P}_2)$. Since our solution for this case should lie among the family of optimal solutions for $k = 1$, results concluded in the previous part should also be taken into account, i.e., the columns of the optimal matrix $\mathbf{C}^*$ must form a uniform tight frame, where the frame elements $\mathbf{c}_i^*$ have norm $\sqrt{m/N}$.

Given $T \in \Omega_2$, the matrix $\mathbf{C}_T$ consists of two columns, e.g., $\mathbf{c}_i$ and $\mathbf{c}_j$. So, the matrix

$\mathbf{C}_T^H \mathbf{C}_T$ in the max-min problem (3.8) is a $(2 \times 2)$ matrix:

$$\mathbf{C}_T^H \mathbf{C}_T = \begin{bmatrix} \langle \mathbf{c}_i, \mathbf{c}_i \rangle & \langle \mathbf{c}_i, \mathbf{c}_j \rangle \\ \langle \mathbf{c}_i, \mathbf{c}_j \rangle & \langle \mathbf{c}_j, \mathbf{c}_j \rangle \end{bmatrix}.$$

From the $k = 1$ case, we have $\|\mathbf{c}_i\|^2 = \|\mathbf{c}_j\|^2 = m/N$. Therefore,

$$\mathbf{C}_T^H \mathbf{C}_T = (m/N) \begin{bmatrix} 1 & \cos \alpha_{ij} \\ \cos \alpha_{ij} & 1 \end{bmatrix},$$

where $\alpha_{ij}$ is the angle between vectors $\mathbf{c}_i$ and $\mathbf{c}_j$. The minimum eigenvalue of this matrix is

$$\lambda_{\min}(\mathbf{C}_T^H \mathbf{C}_T) = (m/N)(1 - |\cos \alpha_{ij}|). \tag{3.11}$$

Given any matrix $\mathbf{C} \in \mathcal{B}_1$, define *coherence* $\mu_{\mathbf{C}}$ as

$$\mu_{\mathbf{C}} = \max_{\mathbf{c}_i, \mathbf{c}_j: \text{ columns of } \mathbf{C}} \frac{|\langle \mathbf{c}_i, \mathbf{c}_j \rangle|}{\|\mathbf{c}_i\| \|\mathbf{c}_j\|}. \tag{3.12}$$

Also, let $\mu^*$ be

$$\mu^* = \min_{\mathbf{C} \in \mathcal{B}_1} \mu_{\mathbf{C}}. \tag{3.13}$$

The following theorem holds.

**Theorem 2** *The optimal value of the objective function of the max-min problem* $(P_2)$ *is* $(m/N)(1 - \mu^*)$. *A matrix* $\mathbf{C}^*$ *is in* $\mathcal{B}_2$ *if and only if the columns of* $\mathbf{C}^*$ *form a* uniform tight frame *with norm values* $\sqrt{m/N}$ *and* $\mu_{\mathbf{C}^*} = \mu^*$.

*Proof:* Since our solution must be chosen from the family of uniform tight frames with frame elements of equal norm $\sqrt{m/N}$, the objective function of $(\mathrm{P}_2)$ is only a function of the angle $\alpha_{ij}$. Using (3.11) and (3.12), it is easy to see that the minimum $\lambda_{\min}(\mathbf{C}_T^H \mathbf{C}_T)$ is $(m/N)(1 - \mu_{\mathbf{C}})$. Using (3.13), we conclude that the largest possible value of the objective function of $(\mathrm{P}_2)$ is $(m/N)(1 - \mu^*)$. ■

*Remark 2:* Methods for constructing uniform tight frames with frame elements that have a coherence $\mu^*$ is equivalent to optimal *Grassmannian packings* of one-dimensional subspaces,

or *Grassmannian line packings* (see, e.g., [53]–[67]). We will say more about this point later in this work.

*Remark 3:* In the case where $k = 2$, the matrix $\mathbf{C}_T^H \mathbf{C}_T$ (where $\mathbf{C} \in \mathcal{B}_1$), for any choice of $T \in \Omega_2$, is a $(2 \times 2)$ matrix with minimum and maximum eigenvalues equal to $(m/N)(1 \pm |\cos \alpha_{ij}|)$. Therefore, the matrix $\mathbf{C}_T^H \mathbf{C}_T$ with eigenvalues equal to $(m/N)(1 \pm \mu_{\mathbf{C}})$ has the smallest minimum eigenvalue and the largest maximum eigenvalue among eigenvalues of all matrices of the form $\mathbf{C}_T^H \mathbf{C}_T$ (for a fixed $\mathbf{C}$ and a varying $T$). Moreover, among all $\mathbf{C} \in B_1$, when comparing the resulting submatrices $\mathbf{C}_T^H \mathbf{C}_T$ for $T \in \Omega_2$, the matrix $\mathbf{C}^*$ with coherence $\mu^*$ has the largest minimum eigenvalue $(m/N)(1-\mu^*)$ and the smallest maximum eigenvalue $(m/N)(1 + \mu^*)$. This means that given any vector $\mathbf{s} \in \mathbb{R}^2$ and $T \in \Omega_2$, the following inequalities hold:

$$(1 - \mu^*)\|\mathbf{s}\|^2 \leq \|\mathbf{C}_T^* \mathbf{s}\|^2 \leq (1 + \mu^*)\|\mathbf{s}\|^2. \tag{3.14}$$

Recall the definition of Restricted Isometry Property (RIP) (see, e.g., [7]): Let $\mathbf{A}$ be a $(p \times q)$ matrix and let $l \leq q$ be an integer. Suppose $\delta_l \geq 0$ is the smallest constant such that, for every $(p \times l)$ submatrix $\mathbf{A}_l$ of $\mathbf{A}$ and every $(l \times 1)$ vector $\mathbf{s}$,

$$(1 - \delta_l)\|\mathbf{s}\|^2 \leq \|\mathbf{A}_l \mathbf{s}\|^2 \leq (1 + \delta_l)\|\mathbf{s}\|^2.$$

Then, the matrix $\mathbf{A}$ is said to satisfy the $l$-restricted isometry property ($l$-RIP) with the restricted isometry constant (RIC) $\delta_l$.

By comparing the 2-RIP definition with (3.14), we can conclude that the optimal matrix $\mathbf{C}^*$ not only satisfies the 2-RIP with RIC $\mu^*$, but also among all matrices that satisfy 2-RIP and have uniform column norms equal to $\sqrt{m/N}$, it provides the best RIC. Thus, our solution for optimizing the worst-case SNR for 2-sparse signals is also the ideal matrix for recovering 2-sparse signals based on methods that rely on the RIP condition for their performance guarantees.

### 3.3.3 Sparsity Level $k > 2$

We now consider the case where $k > 2$. In this case, $T \in \Omega_k$ can be written as $T = \{i_1, i_2, \cdots, i_k\} \subset \Omega$. From the previous results, we know that an optimal matrix $\mathbf{C}^* \in \mathcal{B}_k$ must already satisfy two properties, in addition to $\mathbf{C}^*\mathbf{C}^{*H} = \mathbf{I}$:

- Columns of $\mathbf{C}^*$ must build a uniform tight frame with equal norm $\sqrt{m/N}$ (to be in the set $\mathcal{B}_1$),
- The coherence $\mu_{\mathbf{C}^*}$ should be equal to $\mu^*$ (to be in the set $\mathcal{B}_2$).

Taking the above properties into account for $\mathbf{C}^*$, the matrix $\mathbf{C}_T^{*H}\mathbf{C}_T^*$ will be a $(k \times k)$ symmetric matrix that can be written as $\mathbf{C}_T^{*H}\mathbf{C}_T^* = (m/N)[\mathbf{I} + \mathbf{A}_T]$ where $\mathbf{A}_T$ is

$$\mathbf{A}_T = \begin{bmatrix} 0 & \cos\alpha^*_{i_1 i_2} & \dots & \cos\alpha^*_{i_1 i_k} \\ \cos\alpha^*_{i_1 i_2} & 0 & \dots & \cos\alpha^*_{i_2 i_k} \\ \vdots & \vdots & \ddots & \vdots \\ \cos\alpha^*_{i_1 i_k} & \cos\alpha^*_{i_2 i_k} & \dots & 0 \end{bmatrix}, \tag{3.15}$$

where $i_h \neq i_f \in T$ for the entry $\cos\alpha^*_{i_h i_f}$ in the $(i_h, i_f)$th location. Then,

$$\lambda_{\min}(\mathbf{C}_T^{*H}\mathbf{C}_T^*) = (m/N)(1 + \lambda_{\min}(\mathbf{A}_T)). \tag{3.16}$$

So, the problem simplifies to

$$(\mathrm{P}_k) \quad \begin{cases} \max\limits_{\mathbf{C}} \min\limits_{T} & \lambda_{\min}(\mathbf{A}_T), \\ \text{s.t.} & \mathbf{C} \in \mathcal{B}_{k-1}, \\ & T \in \Omega_k. \end{cases} \tag{3.17}$$

Solving the above problem is not trivial. It is worth mentioning that, as we will discuss later, the family of frames lying in the set $\mathcal{B}_2$ are known to be Grassmannian line packings. Building such frames is known to be very hard and in fact, for a lot of values of $m$ and $N$, no solution has been found so far (see, e.g., [53]). This means that building solutions for problems $(\mathrm{P}_k)$ is even a harder task. Nevertheless, we provide bounds on the value of the optimal objective function.

Given $T \in \Omega_k$, let $\delta^*_{i_h i_f}$ be

$$\delta^*_{i_h i_f} = \mu^* - |\cos \alpha^*_{i_h i_f}|, \quad i_h \neq i_f \in T. \tag{3.18}$$

Also, define $\Delta^*$ in the following way:

$$\Delta^* = \min_{T \in \Omega_k} \sum_{i_h \neq i_f \in T} \delta^*_{i_h i_f}.$$

The following theorem holds.

**Theorem 3** *The optimal value of the objective function of the max-min problem* $(P_k)$ *for* $k > 2$ *lies between* $(m/N)(1 - \binom{k}{2}\mu^* + \Delta^*)$ *and* $(m/N)(1 - \mu^*)$.

*Proof:* Let $\mathbf{x}_{ij}$ and $\mathbf{y}_{ij}$ be two $(k \times 1)$ vectors such that $\mathbf{x}_{ij}$ contains values $(1/\sqrt{2})$ and $(-1/\sqrt{2})$ and $\mathbf{y}_{ij}$ contains values $(1/\sqrt{2})$ and $(1/\sqrt{2})$ in the $i$th and $j$th locations $(i \neq j)$ and zeros elsewhere. Then, by using Rayleigh's inequality, i.e.,

$$\lambda_{\min}(\mathbf{A}_T) \leq \frac{\mathbf{x}^H \mathbf{A}_T \mathbf{x}}{\mathbf{x}^H \mathbf{x}},$$

for the matrix $\mathbf{A}_T$ defined above and the family of vectors $\{\mathbf{x}_{ij}\}$ and $\{\mathbf{y}_{ij}\}$ defined by $i$ and $j$ (chosen from the set $\{1, 2, \ldots, k\}$), we conclude that $\lambda_{\min}(\mathbf{A}_T) \leq -|\cos \alpha^*_{i_h i_f}|,\ i_h \neq i_f \in T$. Thus,

$$\min_{T \in \Omega_k} \lambda_{\min}(\mathbf{A}_T) \leq \min_{\substack{i_h \neq i_f \in T \\ T \in \Omega_k}} (-|\cos \alpha^*_{i_h i_f}|) = -\mu^*. \tag{3.19}$$

Given $T \in \Omega_k$, the matrix $\mathbf{A}_T$ can be written as summation of $\binom{k}{2}$ matrices $\mathbf{F}_{i_h i_f}$ $(i_h \neq i_f \in T)$ where each matrix $\mathbf{F}_{i_h i_f}$ has the entry $\cos \alpha^*_{i_h i_f}$ in the $(i_h, i_f)$th and $(i_f, i_h)$th locations and zeros elsewhere. Using matrix properties (see, e.g., [72]), we can write

$$\begin{aligned}
\lambda_{\min}(\mathbf{A}_T) &\geq \sum_{\substack{i_h \neq i_f \in T \\ T \in \Omega_k}} \lambda_{\min}(\mathbf{F}_{i_h i_f}) = \sum_{\substack{i_h \neq i_f \in T \\ T \in \Omega_k}} -|\cos \alpha^*_{i_h i_f}| \\
&= \sum_{\substack{i_h \neq i_f \in T \\ T \in \Omega_k}} -\mu^* + \delta^*_{i_h i_f} = -\binom{k}{2}\mu^* + \sum_{\substack{i_h \neq i_f \in T \\ T \in \Omega_k}} \delta^*_{i_h i_f}.
\end{aligned}$$

Therefore,

$$\min_{T\in\Omega_k} \lambda_{\min}(\mathbf{A}_T) \geq -\binom{k}{2}\mu^* + \Delta^*. \tag{3.20}$$

Using (3.16), (3.19), and (3.20) we get

$$\begin{aligned}(m/N)(1-\mu^*) &\geq \min_{T\in\Omega_k} \lambda_{\min}(\mathbf{C}_T^{*H}\mathbf{C}_T^*) \\ &\geq (m/N)(1-\binom{k}{2}\mu^* + \Delta^*).\end{aligned} \tag{3.21}$$

This completes the proof. ■

### 3.3.4 Equiangular Uniform Tight Frames and Grassmannian Packings

The inequality (3.21) in Theorem (3) suggests that if all angles between column pairs are equal, then the optimal value of the objective function of ($\mathrm{P}_k$) for $k > 2$ will reach its upper bound. In this case, the columns of $\mathbf{C}^* \in \mathcal{B}_k$ in fact form an *equiangular uniform tight frame*.

Equiangular uniform tight frames are Grassmannian packings, where a collection of $N$ one-dimensional subspaces are packed in $\mathbb{R}^m$ such that the chordal distance between each pair of subspaces is the same (see, e.g., [53], [61], and [62]). Each one-dimensional subspace is the span of one of the frame element vectors $\mathbf{c}_i$. The chordal distance between the $i$th subspace $\langle \mathbf{c}_i \rangle$ and the $j$th subspace $\langle \mathbf{c}_j \rangle$ is given by

$$d_c(i,j) = \sqrt{\sin^2 \alpha_{ij}}, \tag{3.22}$$

where $\alpha_{ij}$ is the angle between $\mathbf{c}_i$ and $\mathbf{c}_j$. When all the $\alpha_{ij}$, $i \neq j$, are equal and the frame is tight, the chordal distances between all pairs of subspaces become equal, i.e., $d_c(i,j) = d_c$ for all $i \neq j$, and they take their maximum value. This maximum value is the simplex bound given by

$$d_c = \sqrt{(N(m-1))/(m(N-1))}. \tag{3.23}$$

Alternatively, the largest absolute value of the cosine of the angle between any two frame elements is bounded as

$$\max_{i\neq j} |\cos \alpha_{ij}| \geq \sqrt{(N-m)/(m(N-1))}.$$

The derivation of this lower bound is originally due to Welch [73]. The Welch bound, or alternatively the simplex bound, are reached if and only if the vectors $\{\mathbf{c}_i\}_{i=1}^N$ form an equiangular uniform tight frame. This is possible only for some values of $m$ and $N$. It is shown in [71] that this is possible only when $1 < m < N-1$ and

$$N \leq \min\{m(m+1)/2, (N-m)(N-m+1)/2\} \tag{3.24}$$

for frames with real elements, and

$$N \leq \min\{m^2, (N-m)^2\} \tag{3.25}$$

for frames with complex elements. If the above conditions hold, then the optimal solution for $(\mathrm{P}_k)$ for $k > 2$ is a matrix $\mathbf{C}^*$ such that its columns form an equiangular uniform tight frame with frame elements of equal norm $\sqrt{m/N}$ and angle $\alpha$ defined as

$$\alpha = \arcsin\left(\sqrt{\left(\frac{m-1}{m}\right)\left(\frac{N}{N-1}\right)}\right). \tag{3.26}$$

The optimal value of the objective function of $(\mathrm{P}_k)$ in this case is $(m/N)(1-\mu^*)$, where $\mu^* = |\cos\alpha| = \sqrt{(N-m)/(m(N-1))}$.

In other cases where $N$ and $m$ do not satisfy the condition (3.24) or (3.25), the following inequality provides a tighter bound than the simplex bound for $\mu^*$ for some values of $N$ and $m$ (see [74]):

$$\mu^* \geq \cos\left[\pi\left(\frac{(m-1)}{N\sqrt{\pi}}\frac{\Gamma(\frac{m+1}{2})}{\Gamma(\frac{m}{2})}\right)^{1/(m-1)}\right].$$

Applying the above inequalities to (3.21), we conclude that by using a Grassmannian line packing where the $k$ largest angles among angles between column pairs of the matrix $\mathbf{C}^*$ are as close as possible to the angle $\alpha$ related to $\mu^*$, the value of the SNR is guaranteed to be higher than the computed lower bound. This is, however, a very difficult problem since even finding Grassmannian line packings for different values of $N$ and $m$ is still an open problem. The reader is referred to [53] and [62] for more details.

We have thus considered a worst-case design criterion in which we assume nothing about the vector $\boldsymbol{\theta}$, and our design is robust against arbitrary possibilities of this unknown.

## 3.4 The Average-case Problem Statement

In the worst-case problem, an optimal $k$-platform $T$ for problem ($\mathrm{P}_k$) is a member of $\Omega_k$ that minimizes $\|\mathbf{C}_T\boldsymbol{\theta}_T\|^2$. In this section, instead of finding the worst-case $T$, we consider an average-case problem with a random $T$. Let $T_k$ to be a random variable taking values in $\Omega_k$, uniformly distributed over $\Omega_k$. In other words, if we let $p_k(t)$ be the probability that $T_k = t$ where $t \in \Omega_k$, then

$$p_k(t) = \binom{N}{k}^{-1}, \qquad \forall t \in \Omega_k.$$

Our goal is to find a measurement matrix $\boldsymbol{\Phi}$ that maximizes the expected value of the minimum SNR, where the expectation is with respect to the random $k$-platform $T_k$, and the minimum is with respect to the entries of the vector $\boldsymbol{\theta}$ on $T_k$. Taking into account the simplifying steps used earlier for the worst-case problem in Section 3.2 and also adopting the lexicographic approach, the problem of maximizing the average SNR can then be formulated in the following way: Let $\mathcal{N}_0$ be the set containing all $(m \times N)$ right orthogonal matrices. Then for $k = 1, 2, \ldots,$ recursively define the set $\mathcal{N}_k$ as the solution set to the following optimization problem:

$$\begin{cases} \max\limits_{\mathbf{C}} \mathbf{E}_{T_k} \min\limits_{\boldsymbol{\theta}_k} & \|\mathbf{C}_{T_k}\boldsymbol{\theta}_k\|^2, \\ \text{s.t.} & \mathbf{C} \in \mathcal{N}_{k-1}, \\ & \|\boldsymbol{\theta}_k\| = 1, \end{cases} \tag{3.27}$$

where $\mathbf{E}_{T_k}$ is the expectation with respect to $T_k$. As before, the $(m \times k)$ matrix $\mathbf{C}_{T_k}$ are the columns of $\mathbf{C}$ whose indices are in $T_k$. The above can be simplified to the following:

$$(\mathrm{F}_k) \quad \begin{cases} \max\limits_{\mathbf{C}} & \mathbf{E}_{T_k}\lambda_{\min}(\mathbf{C}_{T_k}^H\mathbf{C}_{T_k}), \\ \text{s.t.} & \mathbf{C} \in \mathcal{N}_{k-1}. \end{cases} \tag{3.28}$$

## 3.5 Solution to the Average-case Problem

To solve the lexicographic problems ($\mathrm{F}_k$), we follow the same method we used earlier for the worst-case problem, i.e., we begin by solving problem ($\mathrm{F}_1$). Then, from the solution set

$\mathcal{N}_1$, we find optimal solutions for the problem ($\mathrm{F}_2$), and so on.

### 3.5.1 Sparsity Level $k = 1$

Assume that the signal $\mathbf{s}$ is 1-sparse. So, there are $\binom{N}{1} = N$ different possibilities to build the matrix $\mathbf{C}_{T_1}$ from the matrix $\mathbf{C}$. The expectation in problem ($\mathrm{F}_1$) can be written as:

$$\mathbf{E}_{T_1}\lambda_{\min}(\mathbf{C}_{T_1}^H\mathbf{C}_{T_1}) = \sum_{t\in\Omega_1} p_1(t)\lambda_{\min}(\mathbf{C}_t^H\mathbf{C}_t) = \sum_{i=1}^{N} p_1(\{i\})\|\mathbf{c}_i\|^2 = \frac{m}{N}. \tag{3.29}$$

The following result holds.

**Theorem 4** *The optimal value of the objective function of problem ($F_1$) is $m/N$. This value is obtained by using any right orthogonal matrix $\mathbf{C} \in \mathcal{N}_0$, i.e., any* tight frame.

*Proof:* The first part is already proved. The proof for optimality is very similar to the proof given in Theorem 1. Thus, $\mathcal{N}_1 = \mathcal{N}_0$. ■

Theorem 4 shows that unlike the worst-case problem, any tight frame is an optimal solution for the problem ($\mathrm{F}_1$).

Next, we study the case where the signal $\mathbf{s}$ is 2-sparse.

### 3.5.2 Sparsity Level $k = 2$

For problem ($\mathrm{F}_2$), the expected value term $\mathbf{E}_{T_2}\lambda_{\min}(\mathbf{C}_{T_2}^H\mathbf{C}_{T_2})$ is equal to

$$\sum_{t\in\Omega_2} p_2(t)\lambda_{\min}(\mathbf{C}_t^H\mathbf{C}_t) = \quad \sum_{j=2}^{N}\sum_{i=1}^{j-1} p_2(\{i,j\})\lambda_{\min}(\mathbf{C}_{\{i,j\}}^H\mathbf{C}_{\{i,j\}}).$$

Now, since $p_2(t) = 1/\binom{N}{2} = 2/(N(N-1)), \forall t \in \Omega_2$, we can go further and write $\mathbf{E}_{T_2}\lambda_{\min}(\mathbf{C}_{T_2}^H\mathbf{C}_{T_2})$ as

$$\frac{2}{N(N-1)}\sum_{j=2}^{N}\sum_{i=1}^{j-1}\lambda_{\min}(\mathbf{C}_{\{i,j\}}^H\mathbf{C}_{\{i,j\}}). \tag{3.30}$$

Solving problem ($\mathrm{F}_2$) with this objective function is not trivial in general. In fact, claiming anything about solutions of the family of problems ($\mathrm{F}_k$), $k = 2, 3, \ldots$, is hard. However, if we constrain ourselves to the class of *uniform* tight frames, which also arise in solving

the worst-case problem, we can establish necessary and sufficient conditions for optimality. Nonetheless, these conditions are different from those for the worst-case problem and as we will show next the optimal solution here is a uniform tight frame for which a cumulative measure of coherence is minimal.

Let $\mathcal{M}_1$ be defined as $\mathcal{M}_1 = \{\mathbf{C} : \mathbf{C} \in \mathcal{N}_1, \|\mathbf{c}_i\| = \sqrt{m/N}, \forall i \in \Omega\}$. Also, for $k = 2, 3, \ldots,$ recursively define the set $\mathcal{M}_k$ as the solution set to the following optimization problem:

$$(\mathrm{F}_k^{'}) \quad \begin{cases} \max\limits_{\mathbf{C}} & \mathbf{E}_{T_k} \lambda_{\min}(\mathbf{C}_{T_k}^H \mathbf{C}_{T_k}), \\ \text{s.t.} & \mathbf{C} \in \mathcal{M}_{k-1}. \end{cases} \tag{3.31}$$

We will concentrate on solving the above problems instead of the family of problems $(\mathrm{F}_k)$, $k = 2, 3, \ldots$. For $k = 2$, we have the following result.

**Theorem 5** *The matrix* $\mathbf{C}$ *is in* $\mathcal{M}_2$ *if and only if the frame sum-coherence* $\sum_{j=2}^{N} \sum_{i=1}^{j-1} |\langle \mathbf{c}_i, \mathbf{c}_j \rangle|$ *is minimized.*

*Proof:* For $k = 2$, the value of $\lambda_{\min}(\mathbf{C}_t^H \mathbf{C}_t)$ for $t = \{i, j\} \in \Omega_2$ is equal to

$$\lambda_{\min}(\mathbf{C}_{\{i,j\}}^H \mathbf{C}_{\{i,j\}}) = (1/2)(\|\mathbf{c}_i\|^2 + \|\mathbf{c}_j\|^2 - f(i,j)),$$

where $f(i,j)$ is defined as $f(i,j) = \sqrt{(\|\mathbf{c}_i\|^2 - \|\mathbf{c}_j\|^2)^2 + 4\langle \mathbf{c}_i, \mathbf{c}_j \rangle^2}$. Now, if we replace this in (3.30), we get

$$\begin{aligned} &\frac{1}{N(N-1)} \left( \sum_{j=2}^{N} \sum_{i=1}^{j-1} \|\mathbf{c}_i\|^2 + \|\mathbf{c}_j\|^2 - f(i,j) \right) \\ &= \frac{1}{N(N-1)} \left( (N-1) \sum_{i=1}^{N} \|\mathbf{c}_i\|^2 - \sum_{j=2}^{N} \sum_{i=1}^{j-1} f(i,j) \right) \\ &= \frac{(N-1)m}{N(N-1)} - \frac{1}{N(N-1)} \sum_{j=2}^{N} \sum_{i=1}^{j-1} f(i,j) = \frac{m}{N} - \frac{1}{N(N-1)} \sum_{j=2}^{N} \sum_{i=1}^{j-1} f(i,j). \end{aligned}$$

Since $\mathbf{C} \in \mathcal{M}_1$, then using the fact that $\|\mathbf{c}_i\| = \sqrt{m/N}$, $\forall i \in \Omega$, we can go one step further and write the above objective function as

$$\frac{m}{N} - \frac{2}{N(N-1)} \sum_{j=2}^{N} \sum_{i=1}^{j-1} |\langle \mathbf{c}_i, \mathbf{c}_j \rangle|.$$

Therefore, solving problem ($\mathrm{F}_2^{'}$) becomes equivalent to solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{C}} \quad & \textstyle\sum_{j=2}^{N} \sum_{i=1}^{j-1} |\langle \mathbf{c}_i, \mathbf{c}_j \rangle|, \\ \text{s.t.} \quad & \mathbf{C} \in \mathcal{M}_1. \end{aligned} \tag{3.32}$$

■

Theorem 5 shows that for problem ($\mathrm{F}_2^{'}$), angles between column pairs of the uniform tight frame $\mathbf{C}$ should be designed in a different way than for the worst-case problem. Several articles (though not many) discuss such frames. In [75], the authors introduce a similar concept where instead of finding the minimum of the above summation, they are looking for the maximum, and call it the "cluster coherence" of the frame. In [63], where the authors use frames in coding theory applications, it is proved that the solution to one of the problems discussed in the paper is found by solving (3.32). However, to the best of our knowledge, finding such a frame system is still an open problem—there is no known general solution for problem (3.32). We call the value of the optimal objective function of (3.32) the *minimum sum-coherence*. The following lemma provides bounds for the objective function of this optimization problem.

**Lemma 1** *For a uniform tight frame* $\mathbf{C}$ *with column norms equal to* $\sqrt{m/N}$*, the following inequalities hold:*

$$ab|(N/m-1) - 2(N-1)\mu_{\mathbf{C}}^2| \leq \sum_{j=2}^{N} \sum_{i=1}^{j-1} |\langle \mathbf{c}_i, \mathbf{c}_j \rangle| \leq ab(N-1)\mu_{\mathbf{C}}^2,$$

*where*

$$a = \left( \frac{(m/N)^2}{1-2(m/N)} \right), \quad b = \left( \frac{N(N-2)}{2} \right).$$

*Proof.* See Section 3.7.

### 3.5.3 Sparsity Level $k > 2$

Similar to the worst-case problem, solving problems ($\mathrm{F}_k^{'}$) for $k > 2$ is not only a hard task but also it is not known how to construct frames with the required properties in practice.

This is because the solution sets for these problems all lie in $\mathcal{M}_2$ and the problem $(\mathrm{F}_2^{'})$ is still an open problem. The following lemma provides a lower bound for the optimal objective function of problem $(\mathrm{F}_k^{'})$.

**Lemma 2** *The optimal value of the objective function for problem* $(F_k^{'})$ *is bounded below by* $(m/N)(1-(k(k-1)/2)\mu_C)$.

*Proof.* See Section 3.7.

## 3.6 Simulation Results

As mentioned earlier, constructing uniform tight frames with coherence $\mu^*$ is an open problem for arbitrary $(m, N)$ pairs. However, examples of such frames are available for modest values of $m$ and $N$, mostly for $1 \leq m \leq 16$ and $1 \leq N \leq 50$ (see [76]). To be more precise, the examples in [76] are the best uniform tight frames (in terms of coherence) that the site publisher is aware of. In some cases, these frames in fact have coherence $\mu^*$. In other cases, their coherence is larger than $\mu^*$. For the minimum sum-coherence problem, the examples are even more scarce, and in fact we are not aware of any examples for $(m, M)$ dimensions large enough to be of interest to our study. Therefore, we limit our numerical study to the worst-case problem, where we evaluate the performance of several uniform tight frames from [76].

In all simulations, we assume $\sigma_n^2 = 1$ and $\|\boldsymbol{\theta}_T\| = 1$. We present plots of the worst-case SNR/$N$, where the worst-case SNR is given by

$$\mathrm{SNR} = \min_{T} \min_{\theta_T} \|\boldsymbol{\Phi}\boldsymbol{\Psi}_T\boldsymbol{\theta}_T\|^2/(\sigma_n^2/N) = \min_{T} \lambda_{min}(\mathbf{C}_T^{*H}\mathbf{C}_T^*),$$

by fixing two of the three variables $m$, $N$, and $k$ and changing the third one.

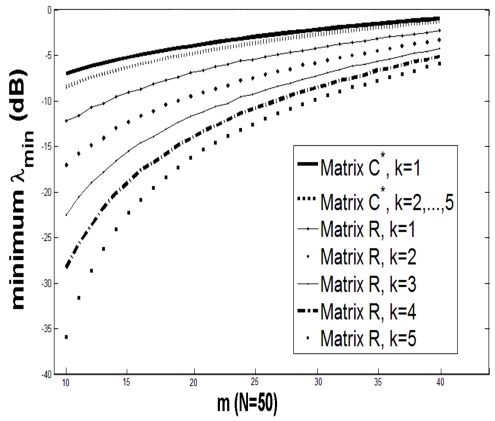
We compare the performance of our robust (worst-case) design $\mathbf{C}^*$ with that of a matrix $\mathbf{R}$ with i.i.d. Gaussian $\mathcal{N}(0,(1/m))$ entries, which is typically used for signal recovery. To satisfy the constraint in problem (3.8), we make $\mathbf{R}$ to be right orthogonal. The value of the objective function in (3.8) is averaged over 100 realizations of the matrix $\mathbf{R}$.

Fig. 3.1(a) shows the worst-case SNR performance for a case where the signal dimension is $N = 50$ and the measurement budget $m$ is varied from 10 to 40. In this case, the condition (3.24) is satisfied and the columns of the optimal matrix $\mathbf{C}^*$ form an equiangular uniform tight frame. We can therefore derive an exact expression for the optimal objective function value based on the Welch bound. For $k = 1$, this value is equal to $m/N$, and for $k \geq 2$, it is equal to $(m/N)(1 - \mu^*)$ where $\mu^* = \sqrt{(N-m)/(m(N-1))}$.

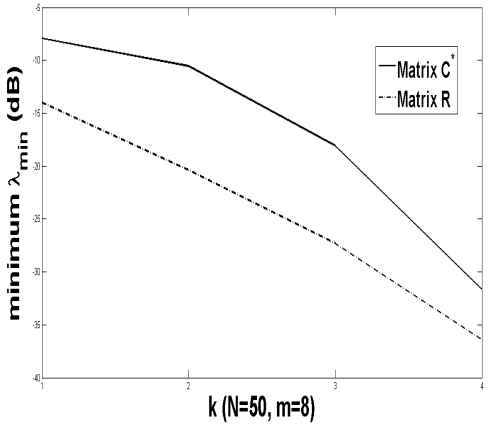
We also consider cases where the condition (3.24) is not satisfied, due to a relatively small measurement budget. Here we use Grassmannian line packings to form measurement matrices. For $(N, m)$ pairs that Grassmannian line packings are not known, we use the best available packings reported in [76] for those dimensions. Figs. 3.1(b)-(f) show the performance of such solutions versus the random matrix $\mathbb{R}$ for different case. In each case, we have fixed two of the variables $N$, $m$, and $k$ and have varied the third one. The values of the objective functions in all these plots are in dB. In all scenarios, the wort-case SNR performance corresponding to the the optimal design $\mathbf{C}^*$ is better than the average taken over 100 realization the random matrix $\mathbb{R}$.

Note that our simulations are only for cases where $m$, $k$ and $N$ are not very large. As mentioned above, one of the reasons is that the available uniform tight frames in [76] are mostly for cases where $1 \leq m \leq 16$ and $1 \leq N \leq 50$. Also, for values of $N$ bigger than 25 and $k$ bigger than 5, finding the smallest minimum eigenvalue of all ${\mathbf{C}_T^*}^H \mathbf{C}_T^*$ for different values of $T$ is computationally intractable.

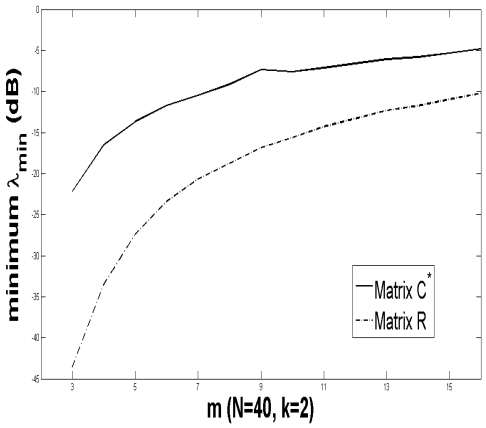
It is important to realize that for most values of $m$ and $N$, the uniform tight frames used in our simulations have a coherence $\mu$ that is bigger than $\mu^*$. In other words, for most values of $m$ and $N$, we are actually comparing the performance of a suboptimal solution matrix instead of the optimal solution with the performance of the random matrix $\mathbb{R}$ and interestingly, the suboptimal solution still has a better performance than the random matrix $\mathbb{R}$ in most, but not all, cases. For example, we notice that in Fig. 1(f), the gap between two curves decreases as $N$ increases. This does not contradict with our theoretical results,



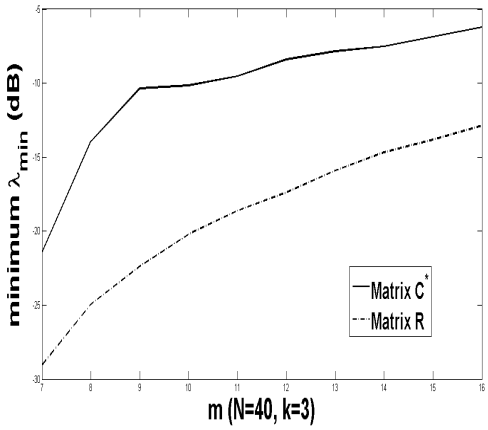

(a) Equiangular uniform tight frames.



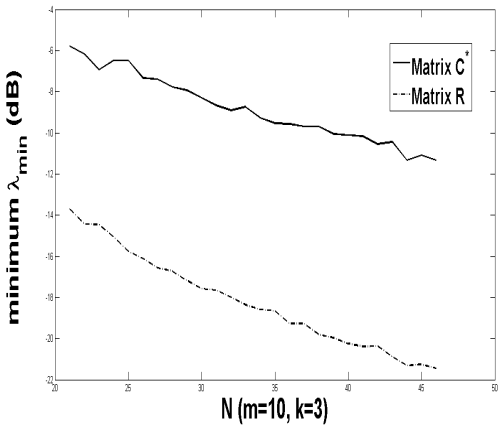

(b) $m = 8$, $N = 50$, and $k$ is varied.



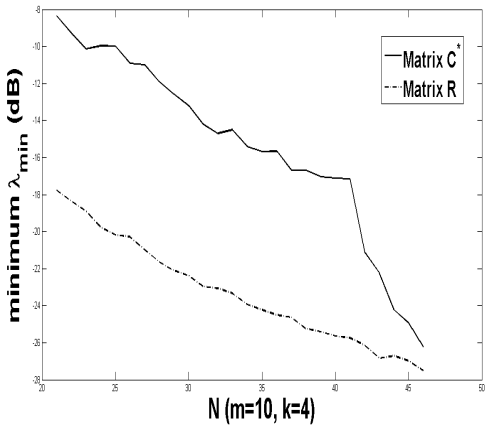

(c) $k = 2$, $N = 40$, and $m$ is varied.




(d) $k = 3$, $N = 40$, and $m$ is varied




(e) $m = 10$, $k = 3$, and $N$ is varied.




(f) $m = 10$, $k = 4$, and $N$ is varies.

**Figure 3.1:** Performance comparison between matrices $\mathbf{C}^*$ and $\mathbf{R}$.

as the plots in Fig. 3.1 do not show the performance of the optimal solution for most values of $m$ and $N$ after all. Rather they show the performance of the best available uniform tight frame example for the corresponding $(m, N)$ values.

## 3.7 Appendix: Proofs of Lemma 1 and Lemma 2

### 3.7.1 Proof of Lemma 1.

Multiply both sides of $\mathbf{C}\mathbf{C}^H = \mathbf{I}$ from the left by $\mathbf{C}^H$ and from the right side by $\mathbf{C}$ to get

$$(\mathbf{C}^H\mathbf{C})^2 = \mathbf{C}^H\mathbf{C}. \tag{3.33}$$

The matrix $\mathbf{C}^H\mathbf{C} = \mathbf{I}$ is an $(N \times N)$ Hermitian matrix, with $(i, j)$th element $(m/N)\cos\alpha_{ij}$ and diagonal elements $m/N$. Using these values, it is easy to see that the matrix $(\mathbf{C}^H\mathbf{C})^2$ is also a Hermitian matrix with the entry $(m/N)^2(\sum_{i=1}^{N}\cos^2\alpha_{ji})$ on the $j$th diagonal location and the entry

$$(m/N)^2(2\cos\alpha_{ij} + \sum_{\substack{l=1,\\ l\neq i,j}}^{N}\cos\alpha_{il}\cos\alpha_{lj})$$

located in the $i$th row and the $j$th column. By comparing the diagonal entries on each side of equation (3.33), we will get the following family of equations:

$$\left(\frac{m}{N}\right)^2\left(\sum_{i=1}^{N}\cos^2\alpha_{ji}\right) = \left(\frac{m}{N}\right), \quad j = 1, \ldots, N.$$

If we sum up all the above equations, after simplifying, we get[1]

$$\sum_{i,j=1}^{N}\cos^2\alpha_{ji} = \frac{N^2}{m}. \tag{3.34}$$

[1]The relation (3.34) is the well-known frame potential condition (see [77])

$$\mathrm{FP} = \sum_{i,j=1}^{N}|\langle\mathbf{c}_i, \mathbf{c}_j\rangle|^2 = m$$

for tight frames, after it has been simplified by enforcing the equal norm assumption.

If we compare the off-diagonal entries of matrices on each side of equation (3.33), then for $i, j = 1, \ldots, N$ and $i \neq j$, we get

$$\left(\frac{m}{N}\right)^2 \left( 2\cos\alpha_{ij} + \sum_{\substack{l=1,\\ l\neq i,j}}^{N} \cos\alpha_{il}\cos\alpha_{lj} \right) = \left(\frac{m}{N}\right)\cos\alpha_{ij},$$

which simplifies to

$$\cos\alpha_{ij} = \left(\frac{(m/N)}{1-2(m/N)}\right)\left( \sum_{\substack{l=1,\\ l\neq i,j}}^{N} \cos\alpha_{il}\cos\alpha_{lj} \right).$$

Using the triangle inequality, we write

$$\begin{aligned}
\sum_{j=2}^{N}\sum_{i=1}^{j-1} |\langle \mathbf{c}_i, \mathbf{c}_j\rangle| &= \frac{m}{N}\sum_{j=2}^{N}\sum_{i=1}^{j-1} |\cos\alpha_{ij}| \\
&\geq \frac{m}{N}\left|\sum_{j=2}^{N}\sum_{i=1}^{j-1}\cos\alpha_{ij}\right| \\
&= a\left|\sum_{j=2}^{N}\sum_{i=1}^{j-1}\sum_{\substack{l=1,\\ l\neq i,j}}^{N}\cos\alpha_{il}\cos\alpha_{lj}\right|.
\end{aligned}$$

We replace $\cos\alpha_{il}\cos\alpha_{lj}$ with $(1/2)(\cos^2\alpha_{il}+\cos^2\alpha_{lj}-(\cos\alpha_{il}-\cos\alpha_{lj})^2)$. The term $\cos^2\alpha_{il}$ is repeated $2(N-2)$ times in the above summation;

- Once $i$ and $l$ are fixed, there are $N-2$ choices left for $j$ to choose the angle $\alpha_{lj}$ in the product term $\cos\alpha_{il}\cos\alpha_{lj}$.
- There are also $N-2$ times that the term $\cos\alpha_{jl}$ is repeated, which is equal to $\cos\alpha_{lj}$.

Therefore,

$$\begin{aligned}
\sum_{j=2}^{N}\sum_{i=1}^{j-1}\sum_{\substack{l=1,\\ l\neq i,j}}^{N}\cos^2\alpha_{il}+\cos^2\alpha_{lj} &= 2(N-2)\sum_{j=2}^{N}\sum_{i=1}^{j-1}\cos^2\alpha_{ij} \\
&= 2(N-2)\frac{(N^2/m)-N}{2}.
\end{aligned}$$

The right hand side of the above inequality simplifies to

$$(a/2)\left|N(N-2)(N/m-1)-\sum_{j=2}^{N}\sum_{i=1}^{j-1}\sum_{\substack{l=1,\\ l\neq i,j}}^{N}(\cos\alpha_{il}-\cos\alpha_{lj})^2\right|.$$

It is easy to show that $|\cos\alpha_{il}-\cos\alpha_{lj}|\leq 2\mu_{\mathbf{C}}$ for any $i\neq j\neq l=1,\ldots,N$. So,

$$-\sum_{j=2}^{N}\sum_{i=1}^{j-1}\sum_{\substack{l=1,\\ l\neq i,j}}^{N}(\cos\alpha_{il}-\cos\alpha_{lj})^2\geq -4\sum_{j=2}^{N}\sum_{i=1}^{j-1}\sum_{\substack{l=1,\\ l\neq i,j}}^{N}\mu_{\mathbf{C}}^2.$$

Similarly, for a fixed $i$ and $j$, there are $N-2$ possibilities for $l$. Also, there are $\binom{N}{2}$ ways to choose $i$ and $j$ from $N$ options. Therefore, the lower bound will be larger than

$$\begin{aligned}&(a/2)\left|N(N-2)(N/m-1)-2N(N-1)(N-2)\mu_{\mathbf{C}}^2\right|\\&=ab|(N/m-1)-2(N-1)\mu_{\mathbf{C}}^2|.\end{aligned}$$

This is the claimed lower bound.

To find the upper bound, we write

$$\begin{aligned}\sum_{j=2}^{N}\sum_{i=1}^{j-1}|\langle\mathbf{c}_i,\mathbf{c}_j\rangle|&=\frac{m}{N}\sum_{j=2}^{N}\sum_{i=1}^{j-1}|\cos\alpha_{ij}|\\&=a\sum_{j=2}^{N}\sum_{i=1}^{j-1}\left|\sum_{\substack{l=1,\\ l\neq i,j}}^{N}\cos\alpha_{il}\cos\alpha_{lj}\right|\\&\leq a\sum_{j=2}^{N}\sum_{i=1}^{j-1}\sum_{\substack{l=1,\\ l\neq i,j}}^{N}|\cos\alpha_{il}\cos\alpha_{lj}|\\&\leq a\sum_{j=2}^{N}\sum_{i=1}^{j-1}\sum_{\substack{l=1,\\ l\neq i,j}}^{N}\mu_{\mathbf{C}}^2\\&=ab(N-1)\mu_{\mathbf{C}}^2.\end{aligned}$$

□

### 3.7.2 Proof of Lemma 2.

Similar to the 2-sparse signals case in Section 3.3, we can write the objective function of problem ($\mathrm{F}_k^{'}$) in the following way:

$$\mathbf{E}_{T_k}\lambda_{\min}(\mathbf{C}_{T_k}^H\mathbf{C}_{T_k}) = \sum_{t\in\Omega_k} p_k(t)\lambda_{\min}(\mathbf{C}_t^H\mathbf{C}_t) = \binom{N}{k}^{-1}\sum_{t\in\Omega_k}\lambda_{\min}(\mathbf{C}_t^H\mathbf{C}_t).$$

Since the matrix $\mathbf{C}$ is a uniform tight frame, for any $t\in\Omega_k$, the matrix $\mathbf{C}_t^H\mathbf{C}_t$ can be written as $(m/N)[\mathbf{I}+\mathbf{A}_t]$, where the $(k\times k)$ matrix $\mathbf{A}_t$ is defined in (3.15). Similar to the worst-case design, we can derive the following inequality:

$$\begin{aligned}\lambda_{\min}(\mathbf{C}_t^H\mathbf{C}_t) &\geq (\frac{m}{N})(1-\sum_{\substack{i_l\neq i_h\in t,\\ t\in\Omega_k}}|\cos\alpha_{i_li_h}|)\\ &\geq (\frac{m}{N})(1-\binom{k}{2}\mu_{\mathbf{C}})\\ &= (\frac{m}{N})(1-(k(k-1)/2)\mu_{\mathbf{C}}).\end{aligned}$$

Taking the expectation, we get

$$\begin{aligned}\mathbf{E}_{T_k}\lambda_{\min}(\mathbf{C}_{T_k}^H\mathbf{C}_{T_k}) &\geq p\sum_{t\in\Omega_k}(\frac{m}{N})(1-(k(k-1)/2)\mu_{\mathbf{C}})\\ &= p\binom{N}{k}(\frac{m}{N})(1-(k(k-1)/2)\mu_{\mathbf{C}})\\ &= (\frac{m}{N})(1-(k(k-1)/2)\mu_{\mathbf{C}}).\end{aligned}$$

This completes the proof. □

# CHAPTER 4

# SUMMARY

## 4.1 Conclusions

In this work, we have studied the problem of designing the compressive measurement matrix for estimating and detecting sparse signals. In the first portion of this work (Chapter 2), our concentration was on adaptively designing compressive measurements for estimating supports of time-varying sparse signals. We approached this problem from a unique perspective and studied the problem in the context of multi-target tracking, which enabled us to apply data association techniques that are available in the literature. In addition, we formulated this problem as a POMDP, which accommodates two categories of adaptive methods, namely, myopic and multi-step lookahead methods. This formulation allowed us to compare the performance of these two categories of adaptive methods not only with each other but also with other adaptive and nonadaptive methods.

We have provided several simulations that consider different scenarios for both single moving target (1-sparse) and multiple moving targets ($s$-sparse). Throughout our simulations, we have applied an approximation technique known as *rollout* to decrease the computation volume that is present when solving a POMDP. Moreover, for the simulations in which there are multiple moving targets, we have applied two approximation heuristics, that are motivated from the well known techniques MHT and JPDA, to overcome the problem of data association that naturally arises once the number of targets is more than one.

The simulation results for this work indicate that the adaptive techniques have a better performance than the non-adaptive traditional designs. But, when comparing the myopic vs. non-myopic designs, the performance criterion, the POMDP cost, and other factors affect the performance of each variation with respect to the other. For example, in Simulations 3

and 4, where we considered occlusion areas in the setting and the goal was to minimize the total number of measurements, the non-myopic design has the best performance. On the contrary, in cases where neither of these conditions hold, e.g., in Simulation 2, there is no difference between the performance of myopic and non-myopic designs and using the latter design only increases the computation volume.

In the second portion of this work (Chapter 3), we have considered the design of low-dimensional (compressive) measurement matrices, for a given number of measurements, for maximizing the worst-case SNR and the average minimum SNR. We have shown an interesting connection between maximizing the two SNR criteria for sparse signal detection and certain classes of frames. In the worst-case SNR problem, we have shown that the optimal measurement matrix is a Grassmannian line packing for most—and a uniform tight frame for all—sparse signals. In the average SNR problem, we have looked for the solution among the class of uniform tight frames and have shown that the optimal measurement matrix is a uniform tight frame that has minimum sum-coherence. Our solutions for both problems provide lower bounds for the performance of the detectors.

## 4.2 Future Work

### 4.2.1 Extensions of Adaptive Sparse Signal Estimation

1) In Simulation 2 of Section 2.4, we did not see any difference between the performance of the myopic design and the multi-step lookahead design. As discussed earlier, the main reason behind this performance similarity is because of the choice of the action space and the POMDP cost. We already showed in Simulations 3 and 4 that how modifying the cost enhances the performance of the multi-step lookahead compared to that of the myopic design. One possibility to extend the current work is to consider a different action space for Problem 1 and see how it affects the performances of myopic and multi-step lookahead designs. For example, the current action space of Problem 1 allows a measurement vector from the prespecified library to be selected repeatedly. An alternative option to this action

space would allow the elimination of the selected measurement vector from the library once it is used. By setting such a constraint, we believe that we are providing an opportunity for the multi-step lookahead design to over perform the myopic design.

2) The present work only provides numerical results for a limited number of scenarios. One direction for future work is to develop analytical results that hold for a variety of settings. For example, currently, we only have an analytical result for rollout in the literature, which is the evaluation of its performance compared to that of its base policy. Alternatively, we could establish a lower bound for the performance of the myopic version of our design.

3) The metric that we have commonly considered in Problems 1 and 2 in the performance criterion is the conditional mutual information $I(\mathbf{y}_k; \mathbf{d}_k | H_k)$. This metric is also used in most of the available work on adaptive compressive sensing in the literature (see, e.g., [23], [24], and [26]). One possibility for future work is to use alternative metrics as the performance criterion and evaluate the performance of those designs with the design presented in this work. For example, since we are currently looking at this problem of time-varying sparse signal support identification in the context of multi-target tracking, one choice for the performance criterion is to consider the distance between the true locations of the nonzero entries of the signal, $\mathbf{d}_{\text{true}}$, and their predicted locations, $\mathbf{d}_{\text{pred.}}$. The metric can be then defined as the error term $e(\mathbf{d}_{\text{true}}, \mathbf{d}_{\text{pred.}}) = \|\mathbf{d}_{\text{true}} - \mathbf{d}_{\text{pred.}}\|^2$. Solving Problems 1 and 2 using this metric instead of $I(\mathbf{y}_k; \mathbf{d}_k | H_k)$ will provide us with a minimum mean squared solution.

4) In this work, we have used the rollout method to find a solution for the POMDP primarily because it decreases the POMDP computation volume and it is also guaranteed to perform at least as well as its base policy. Although by using rollout we were able to take advantage of these benefits, our simulations for finding solutions for the multi-step lookahead designs still took a great amount of time to run. One possible path for future work is to consider and possibly replace rollout with alternative approximation techniques such as *certainty equivalence control* (see [51]) to decrease the POMDP computation volume even more.

5) In the current work, we have concentrated on the problem of measurement selection from a prespecified library. We can alternatively consider the problem of designing the measurement matrix, which would be similar to the goal of the work presented in Chapter 3. To do so, we modify the POMDP action space to be the set of all matrices $\mathbf{A} \in \mathbb{R}^{l \times N}$ with the property that the norm of each row of the matrix $\mathbf{A}$ is equal to one, i.e., $\|\mathbf{A}(i)\|^2 = 1$.

### 4.2.2 Extensions of Sparse Signal Detection

1) The current work in Chapter 3 provides a solution for designing compressive measurement matrices that maximize the SNR. Therefore, by using this design, we are guaranteed to have good performance for any linear detector that we use. A possible extension to the current work is to concentrate on designing compressive measurement matrices for nonlinear signal detectors. Note that by considering any non-linear detector, the performance criterion would probably be different from SNR.

2) The studied problem in Chapter 3 only considers the presence of measurement noise. Alternatively, we can consider the presence of signal interference as well as noise in our setting. This extension would probably relate our work to the matched subspace detectors, which one can find a great amount of available work about them in the literature (see, e.g., [78] and [79]).

3) In this work, we have designed a detector that can be considered to be non-adaptive. A possible extension to the current work is to consider designing an adaptive detector and evaluate the performance of this detector compared to our current design. Moreover, we can also consider alternative variations for the adaptive design in which the sparse signal could be either static or vary over time.

4) When the problem of adaptively designing the compressive measurement matrix for a static sparse signal is considered, we can study the characteristics of the performance criterion used in that problem to see if it satisfies the *submodularity* property (see [80]). If so, then we can rely on only designing myopic detectors since it is shown in [81] that by using

such detectors, we are guaranteed to have a performance that is within the 27% interval of the performance of an optimal detector. This means that it would be very unlikely that a multi-step lookahead detector could over perform the myopic detector.

# REFERENCES


[1] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, pp. 5–18, January 2004.

[2] E. J. Candés and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, pp. 4203–4215, December 2005.

[3] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, April 2006.

[4] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, pp. 118–121, July 2007.

[5] E. Candés, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 1207–1223, August 2006.

[6] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, pp. 14–20, March 2008.

[7] E. Candés, "Compressive sampling," in *Proceedings oh the International Congress of Mathematicians*, vol. 3, (Madrid, Spain), pp. 1433–1452, August 2006.

[8] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[9] H. Rauhut, K. Schnass, and P. Vandergheynst, "Compressed sensing and redundant dictionaries," *IEEE Transactions on Information Theory*, vol. 54, pp. 2210–2219, May 2008.

[10] D. Needell and J. A. Tropp, "CoSamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, pp. 301–321, August 2008.

[11] L. Applebaum, S. D. Howard, S. Searle, and R. Calderbank, "Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery," *Applied and Computational Harmonic Analysis*, vol. 26, no. 2, pp. 283–290, 2009.

[12] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[13] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.

[14] S. Sarvotham, D. Baron, and R. G. Baraniuk, "Sudocodes–fast measurement and reconstruction of sparse signals," in *IEEE International Symposium on Information Theory*, pp. 2804–2808, IEEE, 2006.

[15] R. A. DeVore, "Deterministic constructions of compressed sensing matrices," *Journal of Complexity*, vol. 23, pp. 918–925, December 2007.

[16] P. Indyk, "Explicit constructions for compressed sensing of sparse signals," in *Proceedings of the 19th annual ACM-SIAM symposium on Discrete algorithms*, pp. 30–33, Society for Industrial and Applied Mathematics (SIAM), January 20-22, 2008.

[17] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structured compressed sensing matrices," in *Proceedings of the 14th IEEE/SP Workshop on Statistical Signal Processing (SSP)*, (Madison, WI), pp. 294–298, August 26-29, 2007.

[18] X. W. Xu and B. Hassibi, "Compressed sensing over the Grassmann manifold: A unified analytical framework," in *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, (Monticello, IL), pp. 562–567, September 23-26, 2008.

[19] R. Calderbank, S. Howard, and S. Jafarpour, "Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 358–374, April 2010.

[20] S. D. Howard, A. R. Calderbank, and S. J. Searle, "A fast reconstruction algorithm for deterministic compressive sensing using second order reed-muller codes," in *Proceedings of 42nd Annual Conference on Information Sciences and Systems (CISS)*, (Princeton, NJ), pp. 11–15, March 19-21, 2008.

[21] Z. Ben-Haim and Y. C. Eldar, "The Cramér-Rao bound for estimating a sparse parameter vector," *IEEE Transactions on Signal Processing*, vol. 58, pp. 3384–3389, June 2010.

[22] A. Eftekhari, J. Romberg, and M. Wakin, "Matched filtering from limited frequency samples," *IEEE Transactions on Information Theory*, to appear.

[23] E. Arias-Castro, E. J. Candès, and M. A. Davenport, "On the fundamental limits of adaptive sensing," *IEEE Transactions on Information Theory*, vol. 59, pp. 472–481, January 2013.

[24] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, pp. 2346–2356, June 2008.

[25] E. Bashan, R. Raich, and A. O. Hero, "Optimal two-stage search for sparse targets using convex criteria," *IEEE Transactions on Signal Processing*, vol. 56, pp. 5389–5402, November 2008.

[26] R. M. Castro, J. Haupt, R. Nowak, and G. M. Raz, "Finding needles in noisy haystacks," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Las Vegas, NV), pp. 5133–5136, March 30-April 4, 2008.

[27] D. Wei and A. O. Hero, "Multistage adaptive estimation of sparse signals," *IEEE Journal of Selected Topics in Signal Processing*, to appear.

[28] J. D. Haupt, R. G. Baraniuk, R. Castro, and R. D. Nowak, "Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements," in *Conference Records of the 43rd IEEE Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA), pp. 1551–1555, November 1-4, 2009.

[29] D. Sejdinovic, C. Andrieu, and R. Piechocki, "Bayesian sequential compressed sensing in sparse dynamical systems," in *Proceedings of the 48th IEEE Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, (Monticello, IL), pp. 1730–1736, September 29-October 1, 2010.

[30] C. Qiu, W. Lu, and N. Vaswani, "Real-time dynamic MR image reconstruction using Kalman filtered compressed sensing," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Taipei, Taiwan), pp. 393–396, April 19-24, 2009.

[31] W. Dai, D. Sejdinovic, and O. Milenkovic, "Gaussian dynamic compressive sensing," in *Proceedings of International Conference on Sampling Theory and Applications (SampTA)*, (Singapore), May 2-6, 2011.

[32] M. S. Asif, D. Reddy, P. T. Boufounos, and A. Veeraraghavan, "Streaming compressive sensing for high-speed periodic videos," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP)*, (Hong Kong), pp. 3373–3376, September 26-29, 2010.

[33] W. Li and J. C. Preisig, "Estimation of rapidly time-varying sparse channels," *IEEE Journal of Oceanic Engineering*, vol. 32, pp. 927–939, October 2007.

[34] M. L. Littman, "A tutorial on partially observable Markov decision processes," *Journal of Mathematical Psychology*, vol. 53, pp. 119–125, June 2009.

[35] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[36] J. Vermaak, S. J. Godsill, and P. Perez, "Monte Carlo filtering for multi-target tracking and data association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 309–332, January 2005.

[37] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC)*, vol. 1, (Atlantis, Paradise Island, Bahamas), pp. 735–742, December 14-17, 2004.

[38] R. Karlsson and F. Gustafsson, "Monte Carlo data association for multiple target tracking," in *Proceedings of Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE*, vol. 1, pp. 13/1–13/5, October 16-17, 2001.

[39] M. Mundhenk, J. Goldsmith, C. Lusena, and E. Allender, "Complexity of finite-horizon Markov decision process problems," *Journal of the ACM*, vol. 47, pp. 681–720, July 2000.

[40] V. D. Blondel and J. N. Tsitsiklis, "A survey of computational complexity results in systems and control," *Automatica*, vol. 36, pp. 1249–1274, September 2000.

[41] M. A. Davenport, M. B. Wakin, and R. G. Baraniuk, "Detection and estimation with compressive measurements," Tech. Rep. TREE 0610, ECE Department, Rice University, 2006.

[42] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 445–460, April 2010.

[43] J. Haupt and R. Nowak, "Compressive sampling for signal detection," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, (Honolulu, Hawaii), pp. III–1509–III–1512, April 15-20, 2007.

[44] Z. Wang, G. R. Arce, and B. M. Sadler, "Subspace compressive detection for sparse signals," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Las Vegas, NV), pp. 3873–3876, March 30-April 4, 2008.

[45] J. L. Paredes, Z. Wang, G. R. Arce, and B. M. Sadler, "Compressive matched subspace detection," in *Proceeding of 17th European Signal Processing Conference*, (Glasgow, Scotland), pp. 120–124, August 2009.

[46] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *Proceedings of the 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes (CDC)*, vol. 19, (Albuquerque, NM), pp. 807–812, December 1980.

[47] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *Control Systems, IEEE*, vol. 29, pp. 82–100, December 2009.

[48] G. W. Pulford, "Taxonomy of multiple target tracking methods," *IEE Proceedings Radar, Sonar and Navigation*, vol. 152, pp. 291–304, October 2005.

[49] E. K. P. Chong, C. Kreucher, and A. O. Hero, "Partially observable Markov decision process approximations for adaptive sensing," *Discrete Event Dynamic Systems*, vol. 19, pp. 377–422, September 2009.

[50] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, pp. 99–134, May 1998.

[51] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 3rd ed., January 2007.

[52] D. P. Bertsekas, "Dynamic programming and suboptimal control: A survey from ADP to MPC," *European Journal of Control*, vol. 11, no. 4-5, pp. 310–334, 2005.

[53] J. H. Conway, R. H. Hardin, and N. J. A. Sloane, "Packing lines, planes, etc.: Packings in Grassmannian spaces," *Experimental Mathematics*, vol. 5, pp. 139–159, April 1996.

[54] R. Zahedi, A. Pezeshki, and E. K. P. Chong, "Measurement design for detecting sparse signals," *Physical Communication*, vol. 5, pp. 64–75, June 2012.

[55] H. Park and C. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert Systems with Applications*, vol. 36, pp. 3336–3341, March 2009.

[56] H. Isermann, "Linear lexicographic optimization," *OR Spectrum*, vol. 4, no. 4, pp. 223–228, 1982.

[57] B. Hajek and P. Seri, "Lex-optimal online multiclass scheduling with hard deadlines," *Mathematics of Operations Research*, vol. 30, no. 3, pp. 562–596, 2005.

[58] M. Ehrgott, *Multicriteria Optimization*. Springer, 2nd ed., June 2005.

[59] P. Casazza and M. Leon, "Existence and construction of finite tight frames," *Journal of Concrete and Applied Mathematics*, vol. 4, no. 3, pp. 277–289, 2006.

[60] T. Strohmer, "A note on equiangular tight frames," *Linear Algebra and its Applications*, vol. 429, no. 1, pp. 326–330, 2008.

[61] G. Kutyniok, A. Pezeshki, R. Calderbank, and T. Liu, "Robust dimension reduction, fusion frames, and Grassmannian packings," *Applied and Computational Harmonic Analysis*, vol. 26, no. 1, pp. 64–76, 2009.

[62] T. Strohmer and R. W. Heath Jr., "Grassmannian frames with applications to coding and communication," *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, 2003.

[63] B. G. Bodmann and V. I. Paulsen, "Frames, graphs and erasures," *Linear Algebra and its Applications*, vol. 404, pp. 118–146, 2005.

[64] P. G. Casazza and J. Kovačević, "Equal-norm tight frames with erasures," *Applied and Computational Harmonic Analysis*, vol. 18, no. 2–4, pp. 387–430, 2003.

[65] J. Renes, "Equiangular tight frames from Paley tournaments," *Linear Algebra and its Applications*, vol. 426, no. 2–3, pp. 497–501, 2007.

[66] M. Sustik, J. A. Tropp, I. S. Dhillon, and R. W. Heath Jr., "On the existence of equiangular tight frames," *Linear Algebra and its Applications*, vol. 426, no. 2–3, pp. 619–635, 2007.

[67] V. N. Malozemov and A. B. Pevnyi, "Equiangular tight frames," *Journal of Mathematical Sciences*, vol. 157, no. 6, pp. 789–815, 2009.

[68] E. K. P. Chong and S. H. Żak, *An Introduction to Optimization*. New York, NY: John Wiley and Sons, Inc., 3rd ed., Feburary 2008.

[69] P. G. Casazza, M. Fickus, D. G. Mixon, Y. Wang, and Z. Zhou, "Constructing tight fusion frames," *Applied and Computational Harmonic Analysis*, vol. 30, pp. 175–187, 2011.

[70] R. Calderbank, P. G. Casazza, A. Heinecke, G. Kutyniok, and A. Pezeshki, "Sparse fusion frames: existence and construction," *Advances in Computational Mathematics*, vol. 35, pp. 1–31, 2011.

[71] M. A. Sustik, J. A. Tropp, I. S. Dhillon, and J. R. W. Heath, "On the existence of equiangular tight frames," *Linear Algebra and its Applications*, vol. 426, no. 2–3, pp. 619–635, 2007.

[72] H. Lütkepohl, *Handbook of Matrices*. John Wiley and Sons, Inc., 1st ed., February 1997.

[73] L. R. Welch, "Lower bounds on the maximum cross-correlation of signals," *IEEE Transactions on Information Theory*, vol. 20, pp. 397–399, 1974.

[74] D. G. Mixon, "Tight frames and Grassmannian packings: How they are related and why it matters?." 2010, preprint.

[75] D. L. Donoho and G. Kutyniok, "Geometric separation using a wavelet-shearlet dictionary," in *SampTA09*, 2009.

[76] N. J. A. Sloane, "How to pack lines, planes, 3-spaces, etc.." http://www2.research.att.com/~njas/grass/index.html.

[77] J. J. Benedetto and M. Fickus, "Finite normalized tight frames," *Advances in Computational Mathematics*, vol. 18, no. 2, pp. 357–385, 2003.

[78] L. L. Scharf and B. Friedlander, "Matched subspace detectors," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 2146–2157, 1994.

[79] L. L. Scharf and L. T. McWhorter, "Adaptive matched subspace detectors and adaptive coherence estimators," in *Conference Records of the 30th IEEE Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA), pp. 1114–1117, November 3-6, 1996.

[80] D. M. Topkis, *Supermodularity and complementarity*. Princeton University Press, 2001.

[81] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions–I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.