On Ambiguity Function Shaping for Broadband Constant-Modulus Signals

Mandar Chitre[1], Jing Tian[2], Hari Vishnu[1]

Abstract

Constant-modulus signals such as m-sequences are known to have good autocorrelation properties, as well as good peak-to-average power ratio that allows for full utilization of the transmitter's power. However, the broadband ambiguity surface for such signals exhibit high sidelobe levels that are undesirable in applications where the signal is subject to broadband Doppler. We formulate an optimization problem to minimize the maximum sidelobe levels of such signals over a set of delay-Doppler values. This problem is non-convex and difficult to solve. We explore a convex regularization of the problem that can readily be solved using semi-definite programming, and show that optimal or near-optimal signals can be designed using this method. We further explore some heuristic methods to reduce computational and memory complexity of the solution, to enable us to design longer signals. We demonstrate the advantage of our signal design over conventional unimodular signals for target detection in strong clutter in a continuous active sonar application.

Keywords: Unimodular signal design, continuous active sonar, semi-definite programming, broadband ambiguity function, wideband ambiguity function, Doppler.

1. Introduction

Constant-modulus (a.k.a. unimodular) signals encode information by periodically changing the phase of a carrier. They are used in applications such as active sonar, radar, communications, seismology, non-destructive testing, and biomedical imaging [1, 2, 3, 4, 5]. Unimodular signals are particularly well suited to applications that impose constraints on dynamic range, maximum power rating, and maximum permitted source levels [2, 3]. They also simplify hardware design for transmitters, and allow full utilization of the transmitter's power, due to their low peak-to-average power (PAPR) ratios. Recent improvements in hardware and computational capabilities have given a boost to the design of optimal signals for such applications [1].

In the case of active sonar and radar, we are often interested in targets that are possibly moving, and therefore subject to Doppler. In such cases, the performance of the signal can be characterized by its delay-Doppler ambiguity function (AF) [1]. Sonar

[1]Acoustic Research Laboratory, National University of Singapore.
[2]Institute of Acoustics, Chinese Academy of Sciences.

and radar literature has explored the use of unimodular signals such as m-sequences, Barker, Frank, Zadoff-Chu, Golomb and P1-P4 codes for quite some time [6]. Period matched m-sequences (PMMS) with perfect autocorrelation properties are also used in active sonars [7]. Many of these signals offer good autocorrelation properties, but have high sidelobe levels in the presence of Doppler. Of the ones that offer good Doppler properties, most are designed for a narrowband scenario, and exhibit high sidelobe levels in the AF when a broadband setup with Doppler is considered. Some of these signals have restrictions on signal length and some are optimized for minimum integrated sidelobe level over the whole AF. In many applications, we are required to design signals with a particular AF structure based on application-specific constraints and requirements. For example, a sonar might be designed to detect ships at a 1–10 km range and speeds of up to 10 knots. The signal used for such a sonar only needs to consider sidelobes in a limited part of the AF, corresponding to a delay of 1–10 km and Doppler of -10–10 knots. The design of a unimodular signal required to synthesize a particular narrowband AF is known to be an NP-hard problem [8]. The design of a broadband signal is an even harder problem to solve [1].

A large body of work [9, 10, 11, 12, 13, 14] has considered the problem of signal design, but focused on the autocorrelation properties, rather than on performance in presence of Doppler. In [14], the authors presented a coordinate-descent based framework for jointly considering minimization of the peak-to-sidelobe level and integrated sidelobe level of the autocorrelation function of a unimodular code, but do not consider Doppler. Signal design for matched filter detection in known Doppler was considered in [15, 16]. This work was later extended to include detection of targets of unknown Doppler by maximizing the worst-case signal-to-noise ratio (SNR) [17] and incorporating PAPR constraints [18]. These methods were, however, developed for narrowband signals, with an objective of maximizing the SNR of the target returns. In [19], the authors discussed the design of unimodular codes via quadratic programming using convex relaxation. This method was used to design signals for detecting moving targets in the presence of clutter via the Neyman-Pearson detection framework [8]. These works deal with a slow-time coded radar model in which the clutter returns from one digit of the code are assumed to have no overlap with target returns from another digit, and they do not consider AF shaping. AF shaping for narrowband cognitive radar for targets with Doppler was considered in [20, 21]. Joint transmit signal and receiver filter design for narrowband radar considering apriori information was considered for targets with known Doppler in [22] and for unknown Doppler in [23]. Reference [3] provides a good tutorial and literature survey on joint transmit and receive signal design in the presence of Doppler. In [24], the authors have incorporated target sparsity using a compressed sensing approach. The objective of maximizing the average SNR [25], and the worst case SNR [26], in the presence of clutter, has also been explored. Bounds on achievable correlation performance for narrowband signals have been previously established [27]. However, an extension to broadband signal design is not straightforward.

In this paper, we focus on arbitrary-phase unimodular signal design for a broadband setup in the presence of clutter for detecting a target of unknown Doppler. To the best of our knowledge, this problem has not been previously tackled effectively. The broadband setup is particularly pertinent in applications such as active sonar and seismology [1]. For example, in low-frequency continuous active sonar (CAS), long duration signals are used to increase the energy of the target returns [7]. Since the transmission duration is long,

the receiver has to detect weak target returns in the presence of a strong interference from the direct blast of the signal transmission. Thus targets can be masked by the high sidelobe levels of the transmitted signal. This target-detection scenario is clutter-limited and the effect of noise is minimal in comparison to the masking effect of the transmit signal. To ensure target detectability, appropriate signal design is necessary in order to minimize the sidelobe levels caused by the transmitted signal in regions of the ambiguity surface where target returns are expected. A large signal bandwidth is often needed to achieve good range resolution, and a low carrier frequency is typically needed for longer range in underwater applications due to increased attenuation at high frequency [28].

We assume that signal design requirements are provided in terms of a region of interest in the AF where high sidelobes are undesirable. We formulate the design problem based on the desired AF. Since the original problem is non-convex and difficult to solve, we present a convex regularization which can be solved using semi-definite programming (SDP) to achieve optimal or near-optimal solutions. We dub this method unimodular signal synthesis via minimization (USSM). We further discuss heuristic methods to reduce computational and memory complexity of the solution. This enables us to design longer signals.

We now highlight the specific contributions of this paper:

- We formulate and study a broadband signal design problem where the maximum sidelobe level over a delay-Doppler region of interest is to be minimized. This problem is of value to several applications such as sonar, radar and communication, but has not been studied before.

- The signal design problem is non-convex and difficult to solve. We present an innovative solution by recasting the problem as a biconvex optimization problem that can be solved iteratively.

- We further suggest a modification to our method to allow us to reduce computational complexity while designing long signals. The modification reduces the number of terms required to be computed during each iteration of the optimization problem by roughly a factor of 100.

- Our methods have three free parameters that have to be tuned for good performance. We study the effect of these parameters on performance, and suggest values for the parameters or procedures to tune them.

- We demonstrate the efficacy of a 255-symbol signal designed using our algorithm in a CAS application for detection of a weak target in the presence of direct blast and/or clutter.

The paper is organized as follows. Section 2 formulates the optimization problem associated with designing the signal. Section 3 presents a heuristic method that utilizes SDP to iteratively solve a convex relaxation of this problem. Section 4 discusses the convergence of this method. In section 5, we explore further reduction of computational complexity using a statistical optimization approach. We discuss the results of our signal design and demonstrate some examples in section 6, and then conclude the paper in section 7.

Notation: Superscript $^\intercal$, † and * stand for transpose, Hermitian transpose, and conjugate respectively. Column vectors and matrices are denoted by boldface lowercase and uppercase letters. $\mathbf{I}$ and $\mathbf{0}$ represent the identity and zero matrix respectively. For any matrix $\mathbf{X}$, X_{ij} denotes the entry in the i^{th} row and j^{th} column. Imaginary unit is denoted by i, with $\mathrm{i}^2 = -1$. $\Re(\cdot)$ and $\Im(\cdot)$ represent the real and imaginary parts of a complex number, and $|\cdot|$ represents its magnitude. $|\mathcal{Q}|$ denotes the cardinality of a finite set $\mathcal{Q}$. Sets $\mathbb{R}$, $\mathbb{C}$ and $\mathbb{S}_+$ represent real numbers, complex numbers and positive semidefinite symmetric real matrices respectively. $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix. An inner product $\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{Tr}(\mathbf{A}^\dagger \mathbf{B})$. The matrix inequality $\mathbf{A} \preceq \mathbf{B}$ denotes that $\mathbf{B} - \mathbf{A} \in \mathbb{S}_+$.

2. Problem Formulation

2.1. Arbitrary-phase unimodular signal

Let $s_j = \mathrm{e}^{\mathrm{i}\theta_j} \ \forall\ j \in \{0 \cdots N-1\}$ be the symbols of a complex baseband *unimodular signal* of length N. We assume periodic signals, and hence for all other j, $s_j = s_{j \bmod N}$. If the symbol duration is T then the baseband signal $x(t)$, with period NT, is given by:

$$x(t) = \sum_{j=-\infty}^{\infty} s_j g(t - jT), \tag{1}$$

where $g(t)$ is the pulse shaping function. The passband analytic signal $\tilde{x}(t)$ with carrier frequency f is given by:

$$\tilde{x}(t) = x(t)\mathrm{e}^{\mathrm{i}\omega t}, \tag{2}$$

where $\omega = 2\pi f$. The real passband signal transmitted is simply $\Re[\tilde{x}(t)]$, and the passband analytic signal can be easily reconstructed on reception using a Hilbert transform.

The unimodular signal can be equivalently represented by an N-dimensional vector $\mathbf{s} = [\, s_0\, s_1 \cdots s_{N-1} \,]^\top$ or an N-dimensional vector $\theta = [\, \theta_0\, \theta_1 \cdots \theta_{N-1} \,]^\top$.

2.2. Ambiguity function

Let $\mathcal{D}_{\alpha\delta}[\cdot]$ be an operator that applies a Doppler α and delay δ to a signal, such that:

$$\mathcal{D}_{\alpha\delta}[\tilde{x}(t)] = \sqrt{1+\alpha}\tilde{x}\left((1+\alpha)(t - \delta T)\right). \tag{3}$$

The broadband *ambiguity function* $\chi(\alpha, \delta)$ is then defined as [1]:

$$\chi(\alpha, \delta) = \frac{1}{NT} \int_0^{NT} \tilde{x}(t)^* \mathcal{D}_{\alpha\delta}[\tilde{x}(t)]\, \mathrm{d}t. \tag{4}$$

The *sidelobe level* at Doppler α and delay δ is defined as $|\chi(\alpha, \delta)|$. The mainlobe level is $|\chi(0,0)| = 1$.

Expanding (4) in terms of the baseband signal, we have:

$$\chi(\alpha, \delta) = K_{\alpha\delta} \int_0^{NT} x(t)^* x(\bar{\alpha}t - \bar{\alpha}\delta T)\mathrm{e}^{\mathrm{i}\omega\alpha t}\, \mathrm{d}t,$$

where $K_{\alpha\delta} = \frac{\mathrm{e}^{-\mathrm{i}\omega\bar{\alpha}\delta T}\sqrt{\bar{\alpha}}}{NT}$, and $\bar{\alpha} = 1 + \alpha$. Using (1) and interchanging the order of integration and summations, we get:

$$\chi(\alpha, \delta) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \bar{A}_{\alpha\delta_{jk}} s_j^* s_k,$$

where

$$\bar{A}_{\alpha\delta_{jk}} = K_{\alpha\delta} \int_0^{NT} g(t - jT)^* g(\bar{\alpha}t - \bar{\alpha}\delta T - kT) \mathrm{e}^{\mathrm{i}\omega\alpha t} \, \mathrm{d}t. \tag{5}$$

The calculation of $\bar{A}_{\alpha\delta_{jk}}$ for a rectangular pulse:

$$g(t) = \begin{cases} 1 & 0 \le t < T \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

is demonstrated in Appendix A. For other pulse shapes, a similar approach can be used to evaluate $\bar{A}_{\alpha\delta_{jk}}$. Without loss of generality, we present results for a rectangular pulse in this paper.

Due to the periodicity of $\mathbf{s}$, we can rewrite $\chi(\alpha, \delta)$ as:

$$\begin{aligned}
\chi(\alpha, \delta) &= \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \bar{A}_{\alpha\delta_{jk}} s_j^* s_k \\
&= \sum_{m=-\infty}^{\infty} \sum_{j=0}^{N-1} \sum_{p=-\infty}^{\infty} \sum_{k=0}^{N-1} \bar{A}_{\alpha\delta_{j+mN,k+pN}} s_j^* s_k \\
&= \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \left(\sum_{m=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} \bar{A}_{\alpha\delta_{j+mN,k+pN}} \right) s_j^* s_k \\
&= \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} A_{\alpha\delta_{jk}} s_j^* s_k.
\end{aligned}$$

While computation of $A_{\alpha\delta_{jk}}$ requires infinite sums, we can see from (5) that for practical pulse shapes with finite support and small Doppler, all except one or two of the $\bar{A}_{\alpha\delta_{jk}}$ terms are zero. For most practical Doppler and delays values, we only need to evaluate this sum for $m, p \in \{-1 \cdots 1\}$. So:

$$A_{\alpha\delta_{jk}} = \sum_{m=-1}^{1} \sum_{p=-1}^{1} \bar{A}_{\alpha\delta_{j+mN,k+pN}}. \tag{7}$$

Defining matrix $\mathbf{A}_{\alpha\delta} = [A_{\alpha\delta_{jk}}]$, we can write $\chi(\alpha, \delta)$ as:

$$\chi(\alpha, \delta) = \mathbf{s}^\dagger \mathbf{A}_{\alpha\delta} \, \mathbf{s}. \tag{8}$$

2.3. Signal design

The signal design problem is to find a signal vector θ such that the maximum sidelobe level is minimized:

$$\underset{\theta \in \mathbb{R}^N}{\text{minimize}} \max_{(\alpha,\delta) \in \mathcal{Q}} |\chi(\alpha, \delta)|,$$

where $\mathcal{Q}$ is the set of (α, δ) values that form the delay-Doppler region of interest, over which we wish to minimize the sidelobes. The mainlobe at $(0, 0)$ and a small region around it are excluded from $\mathcal{Q}$. See Appendix Appendix B for the detailed choice of $\mathcal{Q}$.

Since a constant change in phase does not change the magnitude of the ambiguity function, we can arbitrarily set θ_0 to zero [6]. Additionally adopting the ambiguity function given in (4), we can write the design problem as:

$$\begin{aligned} & \underset{\theta \in \mathbb{R}^N}{\text{minimize}} \ \max_{(\alpha,\delta)\in\mathcal{Q}} \left|\mathbf{s}^\dagger \mathbf{A}_{\alpha\delta}\, \mathbf{s}\right| \\ \text{s.t.} \quad & \theta_0 = 0, \\ & 0 \le \theta_j < 2\pi, \ \forall j \in \{1 \cdots N-1\}, \end{aligned} \tag{9}$$

where $\mathbf{s}$ is dependent on θ, i.e., $\mathbf{s} = \mathbf{s}(\theta)$.

The design problem in (9) is similar in essence to the minimax approach in [17, 8], in the sense that the worst-case scenario is optimized for. The difference is that while the previous works focused on maximizing the minimum SNR in the presence of clutter, we focus on minimizing the maximum sidelobe level due to clutter. The ambiguity function's sidelobes are determined by $\mathbf{s}^\dagger \mathbf{A}_{\alpha\delta}\, \mathbf{s}$, which is in general complex, since $\mathbf{A}_{\alpha\delta}$ is not Hermitian. Therefore, the $|\cdot|$ operator cannot be dropped, and the problem cannot be reduced to a quadratic form, to which diagonal loading is often applied to further transform the cost function into a convex form that is readily optimized [17, 8].

3. Optimization

3.1. Near-convex reformulation

The optimization problem in (9) is non-convex in θ and generally very difficult to solve, even for relatively small values of N. However, we can reformulate it into a near-convex problem as we outline next.

First, we rewrite the problem in terms of decision variables $\mathbf{s}$, rather than θ:

$$\begin{aligned} & \underset{\mathbf{s} \in \mathbb{C}^N}{\text{minimize}} \ \max_{(\alpha,\delta)\in\mathcal{Q}} \left|\mathbf{s}^\dagger \mathbf{A}_{\alpha\delta}\, \mathbf{s}\right| \\ \text{s.t.} \quad & s_0 = 1, \\ & |s_j| = 1 \ \ \forall j \in \{0 \cdots N-1\}. \end{aligned}$$

We next expand $\mathbf{s}$ into its real and imaginary components:

$$\begin{aligned} \mathbf{x} &= [\, x_0 \; x_1 \; \cdots \; x_{2N-1} \,]^\top \\ &= [\, \Re(s_0) \; \Im(s_0) \; \cdots \; \Re(s_{N-1}) \; \Im(s_{N-1}) \,]^\top. \end{aligned} \tag{10}$$

We can decompose $\mathbf{s}^\dagger \mathbf{A}_{\alpha\delta}\, \mathbf{s}$ into its real and imaginary parts and write them in terms of $\mathbf{x}$:

$$\mathbf{s}^\dagger \mathbf{A}_{\alpha\delta}\, \mathbf{s} = \mathbf{x}^\top \hat{\mathbf{A}}_{\alpha\delta}\, \mathbf{x} + \mathrm{i}\, \mathbf{x}^\top \check{\mathbf{A}}_{\alpha\delta}\, \mathbf{x}, \tag{11}$$

where the $2N \times 2N$ symmetric matrices $\hat{\mathbf{A}}_{\alpha\delta}$ and $\check{\mathbf{A}}_{\alpha\delta}$ can be easily obtained from $\mathbf{A}_{\alpha\delta}$ by expanding in terms of the real and imaginary parts and taking their symmetric

components:

$$\hat{\mathbf{A}}_{\alpha\delta} = \begin{bmatrix} \Re(A_{00}) & \Im(A_{00}) & \Re(A_{01}) & \Im(A_{01}) & \cdots \\ -\Im(A_{00}) & \Re(A_{00}) & -\Im(A_{01}) & \Re(A_{01}) & \cdots \\ \Re(A_{10}) & \Im(A_{10}) & \Re(A_{11}) & \Im(A_{11}) & \cdots \\ -\Im(A_{10}) & \Re(A_{10}) & -\Im(A_{11}) & \Re(A_{11}) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

$$\check{\mathbf{A}}_{\alpha\delta} = \begin{bmatrix} \Im(A_{00}) & -\Re(A_{00}) & \Im(A_{01}) & -\Re(A_{01}) & \cdots \\ \Re(A_{00}) & \Im(A_{00}) & \Re(A_{01}) & \Im(A_{01}) & \cdots \\ \Im(A_{10}) & -\Re(A_{10}) & \Im(A_{11}) & -\Re(A_{11}) & \cdots \\ \Re(A_{10}) & \Im(A_{10}) & \Re(A_{11}) & \Im(A_{11}) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

So the optimization problem in terms of decision variables $\mathbf{x}$ is:

$$\begin{aligned} & \underset{\mathbf{x}\in\mathbb{R}^{2N}}{\text{minimize}} \ \max_{(\alpha,\delta)\in\mathcal{Q}} \sqrt{\left(\mathbf{x}^\top \hat{\mathbf{A}}_{\alpha\delta}\,\mathbf{x}\right)^2 + \left(\mathbf{x}^\top \check{\mathbf{A}}_{\alpha\delta}\,\mathbf{x}\right)^2} \\ \text{s.t.} \quad & x_0 = 1, \\ & x_1 = 0, \\ & x_{2j}^2 + x_{2j+1}^2 = 1 \ \ \forall j \in \{0 \cdots N-1\}. \end{aligned}$$

We define a rank-1 semidefinite matrix $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$, and observe that a term of the form $\mathbf{x}^\top\mathbf{A}\mathbf{x}$ can be written in terms of matrix $\mathbf{X}$ using the cyclic invariance property:

$$\mathbf{x}^\top\mathbf{A}\,\mathbf{x} = \mathrm{Tr}(\mathbf{x}^\top\mathbf{A}^\top\mathbf{x}) = \mathrm{Tr}(\mathbf{A}^\top\mathbf{x}\mathbf{x}^\top) = \mathrm{Tr}(\mathbf{A}^\top\mathbf{X}) = \langle\mathbf{A},\mathbf{X}\rangle.$$

Now the optimization problem can be written in terms of decision variables $\mathbf{X}$:

$$\begin{aligned} & \underset{\mathbf{X}\in\mathbb{S}_+^{2N}}{\text{minimize}} \ \max_{(\alpha,\delta)\in\mathcal{Q}} \sqrt{\left\langle\hat{\mathbf{A}}_{\alpha\delta},\mathbf{X}\right\rangle^2 + \left\langle\check{\mathbf{A}}_{\alpha\delta},\mathbf{X}\right\rangle^2} \\ \text{s.t.} \quad & X_{00} = 1, \\ & X_{11} = 0, \\ & X_{2j,2j} + X_{2j+1,2j+1} = 1 \ \ \forall j \in \{0 \cdots N-1\}, \\ & \mathrm{rank}(\mathbf{X}) = 1. \end{aligned}$$

We can simplify the objective function by defining a collection of $2|\mathcal{Q}|+1$ new optimization variables: $\hat{z}_{\alpha\delta} = \langle\hat{\mathbf{A}}_{\alpha\delta},\mathbf{X}\rangle$, $\check{z}_{\alpha\delta} = \langle\check{\mathbf{A}}_{\alpha\delta},\mathbf{X}\rangle$ and $z_{\max} = \max\sqrt{\hat{z}_{\alpha\delta}^2 + \check{z}_{\alpha\delta}^2}$. Combining them into a vector $\mathbf{z} = [\, z_{\max},\, \hat{z}_{\alpha\delta},\, \check{z}_{\alpha\delta}\,\forall\,(\alpha,\delta)\in\mathcal{Q}\,]^\top$, then the problem can be written

as:

$$\underset{\mathbf{X}\in\mathbb{S}_+^{2N},\mathbf{z}\in\mathbb{R}^{2|\mathcal{Q}|+1}}{\text{minimize}} \; z_{\max} \tag{12}$$

$$\begin{aligned}
\text{s.t.} \quad & X_{00} = 1, \\
& X_{11} = 0, \\
& X_{2j,2j} + X_{2j+1,2j+1} = 1 \;\; \forall j \in \{0 \cdots N-1\}, && (13)\\
& \hat{z}_{\alpha\delta} - \langle \hat{\mathbf{A}}_{\alpha\delta}, \mathbf{X} \rangle = 0 \;\; \forall\, (\alpha,\delta) \in \mathcal{Q}, && (14)\\
& \check{z}_{\alpha\delta} - \langle \check{\mathbf{A}}_{\alpha\delta}, \mathbf{X} \rangle = 0 \;\; \forall\, (\alpha,\delta) \in \mathcal{Q}, && (15)\\
& z_{\max} \geq \sqrt{\hat{z}_{\alpha\delta}^2 + \check{z}_{\alpha\delta}^2} \;\; \forall\, (\alpha,\delta) \in \mathcal{Q}, && (16)\\
& \text{rank}(\mathbf{X}) = 1. && (17)
\end{aligned}$$

In this optimization problem, the objective function and constraints are all linear, except for conic constraints (16) and rank constraint (17). Except for the rank constraint, this problem is convex and can be readily solved using semidefinite cone programming [29].

In going from (9) to (12), we have not made any approximations or relaxations. However, we have increased the dimensionality of the problem from N to $N(2N+1) + 2|\mathcal{Q}| + 1$.

3.2. Rank constraint regularization

Semidefinite programs (SDP) with rank constraints are commonly encountered in many problems, and are considered NP-hard because they are combinatorial in nature [30]. If we drop the rank constraint (17), (12) becomes a convex problem that can be solved within polynomial time [29]. There are many ways of reconstructing the desired signal from the non rank-one optimal $\mathbf{X}_{\text{opt}}$ that we get from SDP relaxation. For example, we can extract the eigenvector corresponding to the largest eigenvalue, based on a singular value decomposition (SVD). Although the intuition behind the eigenvector approximation is quite straightforward, the quality of the extracted solution is highly problem-dependent, and may even be infeasible (no longer satisfying the original optimization constraints). In our case, the minimax nature of the optimization problem makes it even harder to establish a guarantee on the quality of the extracted eigenvector, and indeed the method did not yield good results. Another alternative method is based on a randomized approximation that exploits the equivalence between stochastic programming and SDP [31, 18, 19, 26]. The idea of randomization-rounding algorithm first originated from the *max-cut* problem [32], and was then generalized to model the unit circle in the complex domain. For acceptable worst-case performance, the method requires positive semidefinite cost function with inequality constraints [33]. However, this is not the case in our problem since our cost function cannot be simplified to a quadratic form, and our constraints are strictly defined as the intersection of the boundaries instead of exteriors of hyper-ellipsoids.

Other than the rank relaxation approach discussed above, such problems can also be solved by replacing the rank constraint with a trace or nuclear norm penalty [34, 35, 36]. Due to constraints (13), the trace and nuclear norm in our problem are constant, and thus attempting to minimize these does not help to reduce the rank of $\mathbf{X}$. Instead, we follow a generalization of the trace heuristic for rank minimization in [37], and minimize

a weighted sum of the original cost function and an eigenvalue residual (sum of all eigenvalues except the highest):

$$\begin{aligned}
&\underset{\mathbf{X}\in\mathbb{S}_+^{2N},\,\mathbf{z}\in\mathbb{R}^{2|\mathcal{Q}|+1}}{\text{minimize}} \left[\zeta z_{\max} \;+\; \underset{\mathbf{W}\in\mathbf{\Phi}_{2N,1}}{\text{minimize}}\, \langle \mathbf{W}, \mathbf{X}\rangle\right] \qquad (18)\\
\text{s.t.}\quad & X_{00} = 1,\\
& X_{11} = 0,\\
& X_{2j,2j} + X_{2j+1,2j+1} = 1 \;\; \forall j \in \{0 \cdots N-1\},\\
& \hat{z}_{\alpha\delta} - \langle \hat{\mathbf{A}}_{\alpha\delta}, \mathbf{X}\rangle = 0 \;\; \forall\, (\alpha,\delta) \in \mathcal{Q},\\
& \check{z}_{\alpha\delta} - \langle \check{\mathbf{A}}_{\alpha\delta}, \mathbf{X}\rangle = 0 \;\; \forall\, (\alpha,\delta) \in \mathcal{Q},\\
& z_{\max} \geq \sqrt{\hat{z}_{\alpha\delta}^2 + \check{z}_{\alpha\delta}^2} \;\; \forall\, (\alpha,\delta) \in \mathcal{Q},
\end{aligned}$$

where $\zeta > 0$ is a positive scalar that controls the tradeoff between minimizing the original cost function $z_{\max}$ and the eigenvalue residual. Here:

$$\mathbf{\Phi}_{2N,1} = \{\, \mathbf{W} \in \mathbb{S}_+^{2N},\; \mathbf{0} \preceq \mathbf{W} \preceq \mathbf{I},\; \mathrm{Tr}\,(\mathbf{W}) = 2N-1 \,\}$$

is the convex hull of the rank-$(2N-1)$ projection matrices [37, 30] that can be used to compute the sum of the $(2N-1)$ smallest eigenvalues of a matrix, i.e., the sum of all eigenvalues except the largest one, by solving an optimization problem, as described in Lemma 1 below. Minimizing the eigenvalue residual leads to minimization of the rank to one, since all but the largest eigenvalue are forced to zero. When we have a rank-1 solution, constraint (17) is satisfied and the penalty term in (18) vanishes. Thus the solution for problem (18) is identical to the solution for problem (12).

Lemma 1: Consider $\mathbf{X} \in \mathbb{S}_+^{2N}$ whose eigenvalues are $\lambda_1(\mathbf{X}) \geq \lambda_2(\mathbf{X}) \geq \cdots \geq \lambda_{2N}(\mathbf{X})$, then

$$\sum_{i=2}^{2N} \lambda_i(\mathbf{X}) = \underset{\mathbf{W}\in\mathbf{\Phi}_{2N,1}}{\text{minimize}}\; \langle \mathbf{W}, \mathbf{X}\rangle. \qquad (19)$$

If $\mathbf{X}$ is fixed, the right hand side of (19) is a convex problem which has a closed form solution. Let $\mathbf{X} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ be the diagonalization of a positive semidefinite matrix $\mathbf{X}$. We have the optimal direction matrix $\mathbf{W} = \mathbf{U}\mathbf{U}^\top$, where $\mathbf{U}$ correspond to the directions of $\mathbf{Q}$ corresponding to the $(2N-1)$ smallest entries of the diagonal matrix $\mathbf{\Lambda}$.

In our problem (18), however, both $\mathbf{X}$ and $\mathbf{W}$ are varying during optimization, which makes it no longer a convex problem but a biconvex one [38]. Although there is no proof of global convergence due to the non-convexity of (18) in nature, as we will discuss later in section 4, we can effectively solve the problem by iterative alternating minimization of $\mathbf{X}$ and $\mathbf{W}$:

$$\mathbf{W}^{(k+1)} = \arg\left\{ \underset{\mathbf{W}\in\mathbf{\Phi}_{2N,1}}{\text{minimize}}\; \left\langle \mathbf{W}, \mathbf{X}^{(k)} \right\rangle \right\}, \qquad (20)$$

$$\mathbf{X}^{(k+1)} = \arg\left\{ \underset{\mathbf{X}\in\mathbb{S}_+^{2N},\mathbf{z}\in\mathbb{R}^{2|\mathcal{Q}|+1}}{\text{minimize}} \left[\zeta z_{\max} \;+\; \underbrace{\left\langle \mathbf{W}^{(k+1)}, \mathbf{X} \right\rangle}_{w^{(k+1)}}\right] \right\}, \qquad (21)$$

subject to the same constraints in (18), and $\mathbf{W}^{(k)}$ and $\mathbf{X}^{(k)}$ denote the estimates of $\mathbf{W}$ and $\mathbf{X}$ at the k^{th} iteration, respectively. Note that (20) and (21) are individually convex

Algorithm 1: The USSM algorithm.

Input : Initial complex signal $\mathbf{s}^{(0)}$,
region of delay-Doppler interest $\mathcal{Q}$,
weight ζ,
maximum iterations $k_{\max}$,
tolerance ϵ.

Output: Optimized signal $\mathbf{s}$.

$k = 0$;
Compute real $\mathbf{x}^{(0)}$ from complex $\mathbf{s}^{(0)}$ using (10);
$\mathbf{X}^{(0)} = \mathbf{x}^{(0)}\mathbf{x}^{(0)\top}$;
while $k < k_{\max}$ **do**
    Compute $\mathbf{W}^{(k+1)}$ from $\mathbf{X}^{(k)}$ using (20) – *closed form solution or convex optimization*;
    Compute $\mathbf{X}^{(k+1)}$ using (21), subject to constraints in (18) – *convex optimization*;
    if $\left|z_{\max}^{(k+1)} - z_{\max}^{(k)}\right| < \epsilon$ **then** break;
    $k = k + 1$;
end
$\mathbf{x}$ = the eigenvector corresponding to the largest eigenvalue of $\mathbf{X}^{(k)}$;
Compute complex $\mathbf{s}$ from real $\mathbf{x}$ using (10);

problems. Problem (20) is equivalent to (19), and we can use the closed form solution instead of a numerical solution, although the latter method sometimes helps to avoid non-feasible stalling points [37, 30]. Problem (21) is a mixed conic linear programming [39] that can be readily solved by standard commercial optimization softwares such as CVX, SDPT3, MOSEK, etc.

We summarize the algorithm described in this section as Algorithm 1.

4. Convergence

We next study the convergence of the iteratively alternating minimization algorithm described in Algorithm 1. As we have already noted in the previous section, there is no proof of global convergence, i.e., convergence to the global optima of (12) with $\operatorname{rank}(\mathbf{X}_{\text{opt}}) = 1$, since problem (18) itself is non-convex in nature. However, it is possible to achieve *local convergence*, which means convergence to a point of possibly infeasible rank or local minimum of (12). The concept of local convergence is extended from [37], and its proof is a particular case of *alternate convex search* for general biconvex problems [38]. The proof has been presented in [37], but we outline it here using the notation in use in this paper for clarity:

Proof. Let us denote the value of expression in (18) as $\eta^{(k)}$ after any particular k^{th} iteration of the two minimization problems (20) and (21). $\eta^{(k)}$ forms a monotonically non-increasing sequence, since at each iteration, the previous minimizers are still feasible, while the current value of $\eta^{(k)}$ is simultaneously established to be smaller than

previous ones if possible. Because the infimum of inner-product of two positive semidefinite matrices is zero, i.e., $\langle \mathbf{W}, \mathbf{X} \rangle \geq 0$, and the term $\zeta z_{\max}$ is guaranteed to be greater than zero (due to (16)), the non-increasing sequence of $\eta^{(k)}$ is bounded below by 0 hence convergent, since any bounded monotonic sequence in $\mathbb{R}$ converges. □

The non-convexity of (18) comes from minimization of the additional term:

$$\underset{\mathbf{W} \in \mathbf{\Phi}_{2N,1}}{\text{minimize}} \langle \mathbf{W}, \mathbf{X} \rangle$$

since the *min-min* of convex functions usually does not retain convexity. Although the total cost value η always goes down monotonically, it does not guarantee to converge to a rank-1 solution or ensure the maximum sidelobe $z_{\max}$ always decreases. There are two parameters that control the convergence: ζ and $\mathbf{W}$. $\mathbf{W}$ varies over iterations and changes the convex landscape along the way. However, since we can use the closed form solution of $\mathbf{W}$ which is uniquely determined, (18) can be viewed as a function of only $\mathbf{X}$ with ζ fixed. As long as we start from the same ζ and $\mathbf{X}^{(0)}$, the algorithm will converge to the same point due to the convexity of each sub-iteration (21). One natural question is how to choose ζ and $\mathbf{X}^{(0)}$ to minimize the maximum sidelobe as much as possible while keeping the rank low. Intuitively speaking, when ζ approaches 0, we effectively search for a rank-1 matrix without taking the original cost function into account, whereas when ζ is large, we minimize (12) without considering the rank. In Appendix Appendix C, we show that the regularization term $w^{(k)} = \langle \mathbf{W}^{(k)}, \mathbf{X}^{(k)} \rangle$ for any given iteration k reduces as ζ decreases, at the cost of a higher sidelobe level $z_{\max}^{(k)}$. Since $w^{(k)}$ is bounded below by 0, it will eventually converge to 0 when ζ is small enough. Note that since $\mathbf{W}^{(k)}$ is computed from $\mathbf{X}^{(k-1)}$ and not $\mathbf{X}^{(k)}$, $\langle \mathbf{W}^{(k)}, \mathbf{X}^{(k)} \rangle$ is not a rank residual for $\mathbf{X}^{(k)}$. Thus $w^{(k)} = 0$ is a sufficient, but not a necessary condition for a rank-1 solution at the k^{th} iteration, as the eigenvector of any rank-1 $\mathbf{X}^{(k)}$ does not have to lie in the null space of $\mathbf{W}^{(k)}$. In fact, there is at most one matrix $\mathbf{X}$ within the feasible set satisfying $\langle \mathbf{W}^{(k)}, \mathbf{X} \rangle = 0$ for our problem[3]. In practice, $w^{(k)} = 0$ is unachievable anyway, due to limited numerical precision. Although a near-zero $w^{(k)}$, however small it is, cannot in theory constrain the rank, it is a good indicator of $\mathbf{X}^{(k)}$ being nearly rank-1. Therefore, we regard $\mathbf{X}^{(k)}$ as rank-1 in a practical sense when $w^{(k)}$ is small enough. As we show later in section 6, there exists a critical (smallest) weight ζ for which the rank-1 constraint is met. Similar observations have been reported by others too [37]. Usually, if the rank does not converge to 1, we can decrease ζ to boost the minimization of $w^{(k)}$.

As briefly discussed earlier, there is in general no guarantee of monotonic decrease of the maximum sidelobe even when the total cost reduces. However, if we maintain the rank of $\mathbf{X}$ to be 1 throughout the optimization, the maximum sidelobe is guaranteed to monotonically decrease. Assuming the initial point of the k^{th} iteration, $\mathbf{X}^{(k-1)}$, is

[3]The null space of $\mathbf{W}^{(k)}$ is of dimensionality one, and $\mathbf{X}$ cannot be arbitrarily scaled due to constraints (13). If $\mathbf{W}^{(k)}$ is determined by some rank-1 initial point $\mathbf{X}^{(k-1)}$ of that iteration, then $\mathbf{X}^{(k-1)}$ is the only *feasible* matrix having zero inner product with $\mathbf{W}^{(k)}$. If we generate $\mathbf{W}^{(k)}$ from Φ_{2N-1} randomly, however, it is likely that its corresponding rank-1 matrix does not satisfy constraints (13) and is therefore infeasible, hence we say there is *at most* one feasible $\mathbf{X}$ satisfying $w^{(k)} = \langle \mathbf{W}^{(k)}, \mathbf{X} \rangle = 0$. Unlike other similar convex regularizations desiring rank higher than one [37], where the null space of $\mathbf{W}$ has higher dimensionality and may exist many feasible candidates giving zero inner products with $\mathbf{W}$, we do not expect to have $w^{(k)} = 0$.

rank-1, since the total cost value is non-increasing, we have:

$$\begin{aligned} & \zeta z_{\max}^{(k)} + \underbrace{\langle \mathbf{W}^{(k)}, \mathbf{X}^{(k)} \rangle}_{\geq 0} \leq \zeta z_{\max}^{(k-1)} + \underbrace{\langle \mathbf{W}^{(k)}, \mathbf{X}^{(k-1)} \rangle}_{=0}, \\ \Rightarrow \quad & \zeta z_{\max}^{(k)} \leq \zeta z_{\max}^{(k-1)} - \langle \mathbf{W}^{(k)}, \mathbf{X}^{(k)} \rangle \leq \zeta z_{\max}^{(k-1)}, \\ \Rightarrow \quad & z_{\max}^{(k)} \leq z_{\max}^{(k-1)}, \end{aligned} \tag{22}$$

In practice, even if $\langle \mathbf{W}^{(k)}, \mathbf{X}^{(k-1)} \rangle = 0$ does not strictly hold due to numerical limitations, it is generally true that $\langle \mathbf{W}^{(k)}, \mathbf{X}^{(k-1)} \rangle \leq \langle \mathbf{W}^{(k)}, \mathbf{X}^{(k)} \rangle$, as long as we keep $\mathbf{X}^{(k-1)}$ (approximately) rank-1 for all the k. Therefore, although we only need the final value of $\mathbf{X}$ to be rank-1, our method does not aim to gradually decreasing the rank over iterations; instead, by careful selection of ζ, we constrain the rank of $\mathbf{X}$ to be close to 1 through all the iterations, since this can guarantee that the cost function approximately equals to the sidelobe level and decreases monotonically, as we will show in section 6.

5. Managing Computational Complexity

The method described in Algorithm 1 provides a computationally efficient solution for the design of good arbitrary-phase unimodular signals, as long as the delay-Doppler region of interest $\mathcal{Q}$ is not very large. When $|\mathcal{Q}|$ becomes large, the number of constraints in (18) as well as the dimensionality of $\mathbf{z}$ grows. For applications requiring the design of unimodular signals with large $\mathcal{Q}$, we propose an approximate method that retains the key steps in the method described previously, but reduces the computational complexity by operating on a carefully selected subset $\tilde{\mathcal{Q}}^{(k)}$ at the k^{th} iteration. We discuss the details of this method in the rest of this section.

Denote the value of cost z for a given delay-Doppler tuple (α, δ) at a given $\mathbf{X}$ as

$$z_{\alpha\delta}(\mathbf{X}) = \sqrt{\langle \hat{\mathbf{A}}_{\alpha\delta}, \mathbf{X} \rangle^2 + \langle \check{\mathbf{A}}_{\alpha\delta}, \mathbf{X} \rangle^2}. \tag{23}$$

At the k^{th} iteration, the cost function we minimize is equivalent to:

$$J^{(k)}(\mathbf{X}) = \zeta \cdot \max_{(\alpha,\delta)\in\mathcal{Q}} z_{\alpha\delta}(\mathbf{X}) + \langle \mathbf{W}^{(k)}, \mathbf{X} \rangle. \tag{24}$$

For brevity of notation, we write $J^{(k)}(\mathbf{X})$ simply as $J^{(k)}$, and implicitly remember that it is a function of $\mathbf{X}$. Evaluating $J^{(k)}$ by searching over the whole set of $\mathcal{Q}$ is computationally very intensive, if $|\mathcal{Q}|$ is large. Instead, we consider the possibility that we can optimize over a subset $\tilde{\mathcal{Q}}$ of $\mathcal{Q}$ that is most likely to contain the largest $z_{\alpha\delta}(\mathbf{X})$ $\forall(\alpha, \delta) \in \mathcal{Q}$, and $|\tilde{\mathcal{Q}}| \ll |\mathcal{Q}|$.

Let us consider the following situation. At the start of k^{th} iteration, the initial value of $\mathbf{X}$ is the optimized output from the previous iteration, $\mathbf{X}^{(k-1)}$, that determines the current $\mathbf{W}^{(k)}$, and results in the cost:

$$\bar{J}_0^{(k)} = \zeta \cdot z_{\alpha\delta(k-1)}(\mathbf{X}^{(k-1)}) + \langle \mathbf{W}^{(k)}, \mathbf{X}^{(k-1)} \rangle, \tag{25}$$

where $\alpha\delta(k-1)$ denotes the values of α and δ that correspond to the maximum z at $\mathbf{X}^{(k-1)}$. At the end of the k^{th} iteration, we will have a new optimized $\mathbf{X}^{(k)}$ corresponding

Algorithm 2: The p-USSM algorithm.

Input : Initial complex signal $\mathbf{s}^{(0)}$,
region of delay-Doppler interest $\mathcal{Q}$,
weight ζ,
fraction p,
maximum iterations $k_{\max}$,
tolerance ϵ.

Output: Optimized signal $\mathbf{s}$.

$k = 0$;
Compute real $\mathbf{x}^{(0)}$ from complex $\mathbf{s}^{(0)}$ using (10);
$\mathbf{X}^{(0)} = \mathbf{x}^{(0)}\mathbf{x}^{(0)\top}$;
while $k < k_{\max}$ **do**
    Compute $\tilde{\mathcal{Q}}$ from $\mathbf{X}^{(k)}$ using (28);
    Compute $\mathbf{W}^{(k+1)}$ from $\mathbf{X}^{(k)}$ using (20) – *closed form solution or convex optimization*;
    Compute $\mathbf{X}^{(k+1)}$ using (21) and $\tilde{\mathcal{Q}}$ instead of $\mathcal{Q}$, subject to constraints in (18) – *convex optimization*;
    if $\left|z_{\max}^{(k+1)} - z_{\max}^{(k)}\right| < \epsilon$ **then** break;
    $k = k + 1$;
end
$\mathbf{x}$ = the eigenvector corresponding to the largest eigenvalue of $\mathbf{X}^{(k)}$;
Compute complex $\mathbf{s}$ from real $\mathbf{x}$ using (10);

to a new tuple $\alpha\delta(k)$ that results in the maximum value at $\mathbf{X}^{(k)}$. The cost then is:

$$\bar{J}_{\text{end}}^{(k)} = \zeta \cdot z_{\alpha\delta(k)}(\mathbf{X}^{(k)}) + \langle \mathbf{W}^{(k)}, \mathbf{X}^{(k)} \rangle. \tag{26}$$

$\mathbf{X}^{(k)}$ will serve as the initial point at the beginning of the $(k+1)^{\text{th}}$ iteration, and such a procedure will carry on. During the k^{th} iteration, if we use a lower complexity cost function:

$$\tilde{J}^{(k)} = \zeta \cdot \max_{(\alpha,\delta)\in\tilde{\mathcal{Q}}^{(k)}} z_{\alpha\delta}(\mathbf{X}) + \langle \mathbf{W}^{(k)}, \mathbf{X} \rangle, \tag{27}$$

where $\tilde{\mathcal{Q}}^{(k)}$ is a small subset of $\mathcal{Q}$, $\mathbf{X}^{(k)}$ will be the same as using the original cost function $J^{(k)}$ as long as $\alpha\delta(k) \in \tilde{\mathcal{Q}}^{(k)}$, since the maximum cost element $\alpha\delta(k)$ of $\mathcal{Q}$ is contained in the subset $\tilde{\mathcal{Q}}$. How do we find such a subset $\tilde{\mathcal{Q}}^{(k)}$ given ζ and $\mathbf{X}^{(k-1)}$?

We now outline a heuristic method that includes a certain largest fraction of:

$$\{z_{\alpha\delta}(\mathbf{X}^{(k-1)}) \; \forall (\alpha,\delta) \in \mathcal{Q}\}$$

at the beginning of the k^{th} iteration. When ζ is small enough, optimization iteration k converges to some point $\mathbf{X}^{(k)}$ *close* to the previous iteration's output $\mathbf{X}^{(k-1)}$, led by its direction matrix $\mathbf{W}^{(k)}$ [37]. Since the cost function is the maximum of $z_{\alpha\delta}(\mathbf{X})$ terms, its value is typically determined by one $\alpha\delta$ pair, and correspondingly one *active* $\mathbf{A}_{\alpha\delta(k)}$. The cost function $\max_{(\alpha,\delta)\in\mathcal{Q}} z_{\alpha\delta}(\mathbf{X})$ is thus continuous (but not smooth) over

$\mathbf{X}$, since the active matrix $\mathbf{A}_{\alpha\delta}$ at different values of $\mathbf{X}$ may be different. At the point where one $\mathbf{A}_{\alpha\delta}$ takes over from another as the active matrix, continuity is guaranteed (since the $\max(f, g)$ of two continuous functions $f(\cdot)$ and $g(\cdot)$ is continuous). As $\mathbf{X}^{(k)}$ only slightly deviates from $\mathbf{X}^{(k-1)}$, the corresponding active $\mathbf{A}_{\alpha\delta(k)}$ will either remain the same as $\mathbf{A}_{\alpha\delta(k-1)}$, or takes over from another as the active matrix that provides a value $z_{\alpha\delta(k)}(\mathbf{X}^{(k-1)})$ close to $z_{\alpha\delta(k-1)}(\mathbf{X}^{(k-1)})$ due to the Lipschitz continuity of $z_{\alpha\delta}(\mathbf{X})$. Intuitively speaking, $\mathbf{X}^{(k)}$ being close to $\mathbf{X}^{(k-1)}$ indicates only a small change in the phase of the unimodular signal, and correspondingly only a small change in the envelope of the ambiguity surface. Therefore, the largest few sidelobes of signal $\mathbf{s}^{(k)}$ will still be from the set of the largest ones for $\mathbf{s}^{(k-1)}$. We therefore should choose set $\tilde{\mathcal{Q}}^{(k)}$ to contain the (α, δ) values corresponding to the only the largest few sidelobes at the start of iteration k:

$$\tilde{\mathcal{Q}}^{(k)} = \{(\alpha, \delta) \; \forall \; z_{\alpha\delta}(\mathbf{X}^{(k-1)}) \geq \Gamma_p^{(k)}\}, \tag{28}$$

where $\Gamma_p^{(k)}$ is the $(1-p)^{\text{th}}$ fractile of $\{z_{\alpha\delta}(\mathbf{X}^{(k-1)}) \; \forall \; \alpha\delta \in \mathcal{Q}\}$. We dub this algorithm as p-USSM, and summarize it as Algorithm 2.

The question that remains is what value of p to use? The choice of p is a trade-off between computational complexity and algorithm performance, and can be made based on practical constraints of available computational power and time. As we show shortly, a value of 1% is large enough that it has no significant effect on the performance of the algorithm with a large reduction in computational complexity.

6. Results and discussion

We apply our design method to some scenarios to test its effectiveness. Henceforth, we assume a broadband scenario with carrier frequency $f = 500$ Hz and bandwidth $B = 500$ Hz as design parameters unless otherwise specified. For a given code length, there are three alternatives of initialization (i.e., choice of $\mathbf{s}^{(0)}$) for the optimization: PMMS, uniform randomly distributed phase (URDP) or setting $\mathbf{W}^{(0)} = \mathbf{0}$ [37]. We use the same initialization in all cases where the effect of the same factor other than initialization is compared, unless otherwise specified. When optimizing over a range of Dopplers, we minimize sidelobes in the Doppler range $\alpha \in (\frac{-0.5}{N}, \frac{0.5}{N})$. We use the MOSEK optimization toolbox [40] for convex optimization, wherever required.

6.1. Naïve non-convex code design

Although we can attempt to solve the non-convex code design problem outlined in (9) using standard optimization techniques, the time required for doing this scales up rapidly with the length of the code being designed. Moreover, we have no guarantees as to the quality of the solution. We compare the complexity of signal design using our near-convex formulation, against that of solving the non-convex formulation in (9). In Fig. 1, we analyze the scaling of the time complexity for solving (9) using MATLAB's optimization toolbox, by plotting the optimization time against the code length. The optimization time plotted is the average of four optimization runs initialized with URDP codes. It can be observed that the time required for solving the non-convex problem (9) grows very rapidly with the code length. Therefore this algorithm is infeasible for designing codes of large lengths, such as those used for CAS. However, we use it as a benchmark to compare our method against, when designing short codes.

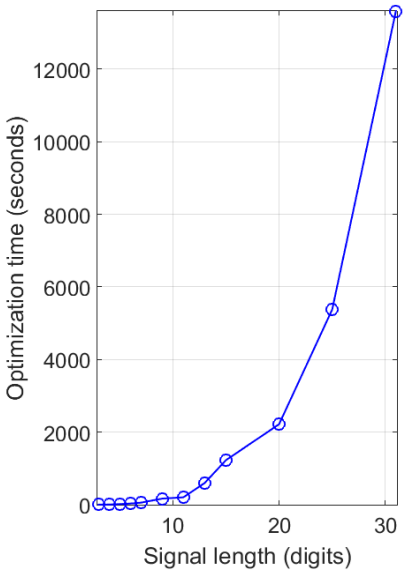


Figure 1: Average time taken on a typical Desktop computer for signal design using naïve non-convex methods (using MATLAB's optimization toolbox) increases rapidly with signal length.

6.2. Design of short codes with USSM

We first consider the design of a 7-digit code using USSM. Although this is a very short code, it helps us develop an intuition for the behavior of USSM, and is hence instructive.

We use an URDP to initialize the optimization. In Table 1, we show the normalized eigenvalue residual (the summation of all the eigenvalues except the largest one, $r^{(k)} = \sum_{i=1}^{2N-1} \lambda_i^{(k)}$) versus iterations with $\zeta = 0$. Although we may expect to see $r^{(k)} = 0$ for all the iterations since the starting point is already rank-1, as the accuracy of the algorithm is set to 10^{-6}, we get $r^{(k)} \neq 0$ (but small) for all the iterations. This suggests that we should differentiate between rank-1 in a mathematical and practical sense. Henceforth, we assume the accuracy of the algorithm to be 10^{-6} and all the matrices with $r^{(k)} < 10^{-3}$ are considered to be approximately rank-1.

The convergence of the design method in terms of cost function, normalized projection $w^{(k)}/n$, and normalized eigenvalue residual $r^{(k)}$ versus iterations are respectively plotted in Figs. 2, 3, and 4, for different values of the parameter ζ. We see that although the cost function always monotonically decreases with iteration, w and r generally increase as ζ becomes larger, since the penalty for non-rank-1 solutions reduces. Projection value $w^{(k)}$ and eigenvalue residual $r^{(k)}$ generally decrease to near zero as the optimization progresses. However, when ζ becomes too large, $w^{(k)}$ and $r^{(k)}$ become significantly greater than 0, resulting in undesirable high rank solutions. Consistent with theory, $w^{(k)} = 0$ is not a necessary condition for a rank-1 solution in mathematical or practical sense; however, it is a fair indicator for the closeness of $r^{(k)}$ to 0, based on empirical evidence.

Based on our observations, there exists a critical threshold of ζ. A value of ζ below the threshold will give us solutions with desired rank, whereas values above do not. For

Table 1: Normalized eigenvalue residual versus iterations

Iteration k	Normalized eigenvalue residual $r^{(k)}$
0	1.04×10^{-16}
1	2.65×10^{-7}
2	2.53×10^{-7}
3	2.38×10^{-7}
4	2.17×10^{-7}
5	1.90×10^{-7}

example, we see that in Fig. 4, $\zeta = 4.8$ seems to be that threshold. A higher $\zeta = 7$ also yields the desired rank after a few iterations, but not initially. An even higher $\zeta = 10$ does not converge to a solution with the desired rank. This suggests an approach to tune ζ by observing the convergence of the algorithm over a few iterations. If the algorithm exhibits erratic oscillations in normalized maximum sidelobe, the value of ζ needs to be reduced. If the algorithm exhibits monotonic, but slow convergence, the value of ζ can be increased.

The convergence in terms of the normalized maximum sidelobe level is shown in Fig. 5, for ζ within the range that yields rank-1 solutions. We also solve the optimization problem using the simulated annealing method and the interior point method in the MATLAB toolbox as a benchmark. In both cases, we obtained a normalized maximum sidelobe of 0.321, which we indicate in Fig. 5 as a benchmark. We can see that using a value of ζ that is too small slows down the convergence as the optimization focuses more on meeting the rank constraint rather than lowering the sidelobe level, whereas using a larger value of ζ (up to the threshold discussed previously) speeds up the convergence of the maximum sidelobe. Nevertheless, given sufficient iterations, all the parameters of ζ within the suitable range yield results better than the MATLAB benchmark.

We next consider the effect of the initialization on the convergence of our method. We first explore the range of suitable ζ for different initializations. Since we desire the output of every iteration to be rank-1, it is reasonable to use the maximum eigenvalue residual among all the iterations as a metric. As the eigenvalue residual generally decreases as the optimization progresses, the maximum eigenvalue residual is typically observed within the first few iterations. We plot the maximum eigenvalue residual out of the first five iterations as a function of ζ in Fig. 6, for different initializations. We observe that the threshold of ζ is sensitive to initialization. We then plot the normalized maximum sidelobe level for optimizations initialized with different URDP codes in Fig. 7, using the largest ζ allowed for each. Note that ζ here for different URDP codes are different since we want to explore the best performance of each code individually. We see that although the maximum sidelobe always decreases, it converges to different levels for different choices of initialization codes. This is not surprising, given that although our optimization sub-problems are convex, our overall problem is not convex and therefore only local convergence of the solution can be guaranteed.

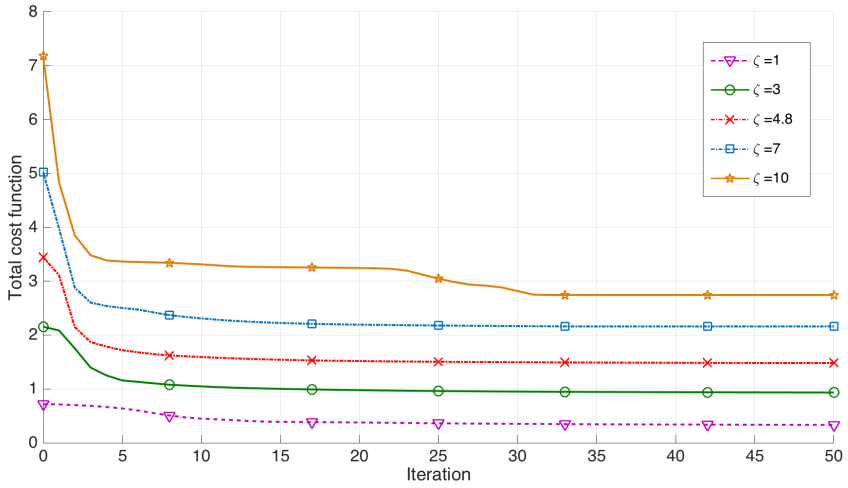


Figure 2: The total cost function monotonically decreases as the optimization progresses. While it may seem to converge to different values for different settings of ζ, this is simply due to the fact that ζ scales part of the cost function. The total cost functions for different ζ cannot be directly compared.

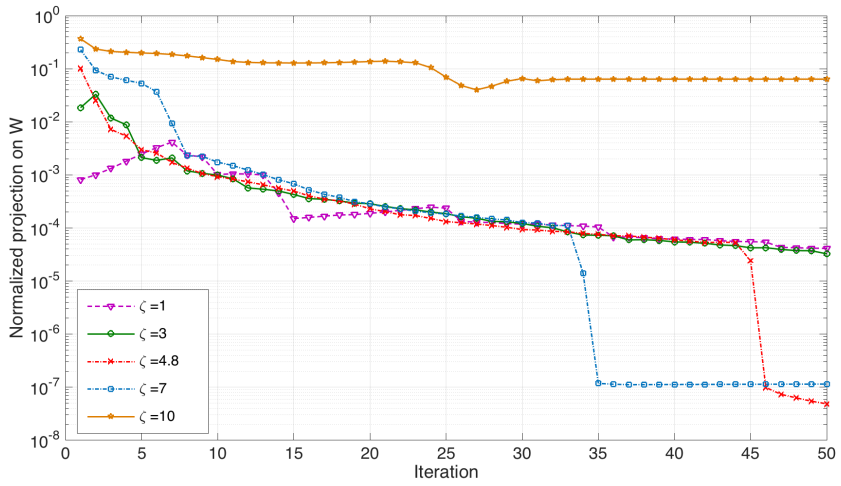


Figure 3: The normalized projection $w^{(k)}/n$ generally reduces as the optimization progresses. As long as ζ is below the threshold, the projection eventually reaches a small value that guarantees a rank-1 solution. Do bear in mind, however, that a small value of $w^{(k)}$ is a sufficient but not necessary condition for a rank-1 solution.

6.3. Choice of initialization

Solving (18) starting from different starting points leads us to different local minima, each of which yield solutions of different quality. A general method shown to be effective in semidefinite programming, with regularization terms similar to ours, is to initialize the direction matrix $\mathbf{W}^{(0)}$ to $\mathbf{0}$ [37]. The intuition behind this initialization is to first find the

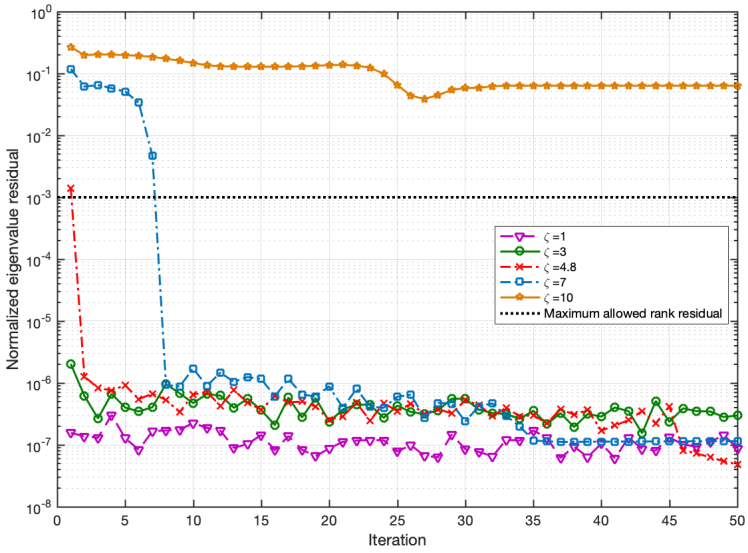


Figure 4: The normalized eigenvalue residual $r^{(k)}$ shows that we obtain rank-1 solutions within the first few iterations, as long as ζ is less than a threshold. For higher values of ζ, the eigenvalue residual r remains large, and the optimization does not converge to the desired rank-1 solution.

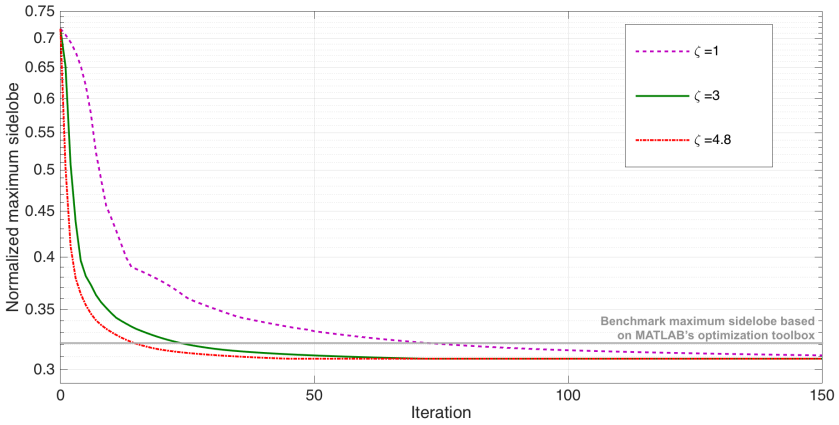


Figure 5: The normalized maximum sidelobe level improves as the optimization progresses ($N = 7$). We benchmark it against the optimized maximum sidelobe level obtained using MATLAB's optimization toolbox, and show that given sufficient number of iterations, our algorithm converges to a better solution.

global minimum of (12) setting aside the regularization term $\langle \mathbf{W}, \mathbf{X} \rangle$ and constraint (17), since the ultimate problem we aim to solve is bounded below by that global minimum. Although such a global optimizer, in general, does not have the desired rank, we hope to reach a better local minimum of (12) by searching around the neighborhood of the global minimum without rank constraint. Since the optimization landscape without rank constraint is convex and continuous in the ambient matrix space, and there is a higher

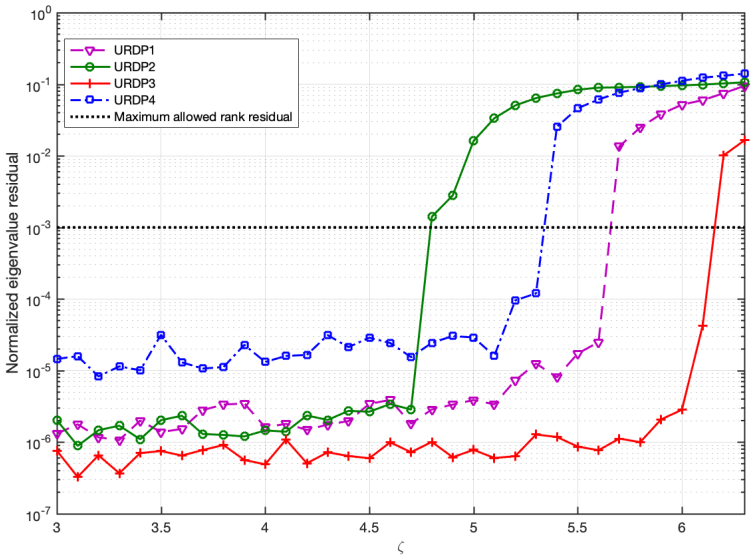


Figure 6: The normalized eigenvalue residual should be less than 10^{-3} for the solution to be considered practically rank-1. For different initializations of the optimization, we observe different thresholds for ζ, below which we consistently obtain a rank-1 solution, and above which, we do not.

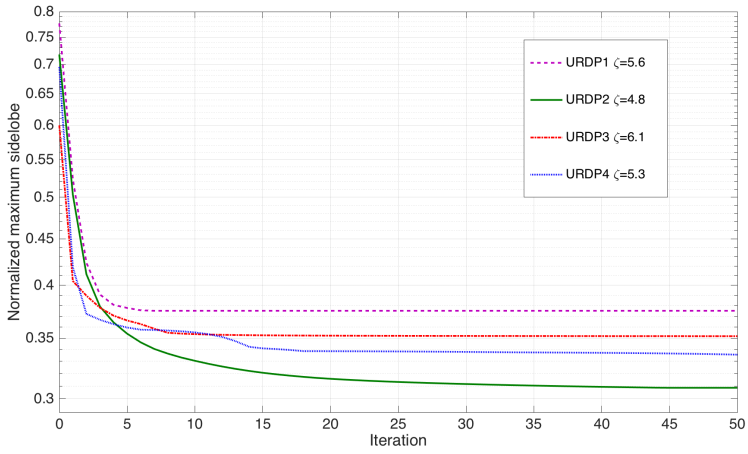


Figure 7: The quality of the final solution is measured in terms of the normalized maximum sidelobe level at the end of the optimization ($N = 7$). Here we show the convergence of the optimization for different initializations and the corresponding best value of ζ. We observe that the quality of the solution depends on the initial code used to seed the optimization.

chance for the lowest minimum of (12) to be situated close to it. Note that the largest difference as compared to initialization with any unimodular signal is that there is no $\mathbf{X}^{(k+1)}$ within the feasible set satisfying $\langle \mathbf{W}^{(k)}, \mathbf{X}^{(k+1)} \rangle = 0$ for those $\mathbf{W}^{(k)}$ that resulted from non rank-1 $\mathbf{X}^{(k)}$ during the first few iterations. In other words, we slowly reduce

Table 2: Normalized maximum sidelobe level using different optimization methods for $n = 7$ unimodular signal.

Optimization method		Normalized maximum sidelobe
Non-convex formulation	Simulated annealing	0.3204
	Interior point	0.3214
Iterative optimization with different initializations	PMMS	0.3516
	URDP1	0.3750
	URDP2	0.3086
	URDP3	0.3516
	URDP4	0.3356
	$\mathbf{W}^{(0)} = \mathbf{0}$	0.2850

the rank over the first few iterations instead of always forcing it to be 1, as is the case of the unimodular signal initialization. Therefore, we do not expect the outputs of the first few iterations to be rank-1 or the sidelobe level to monotonically decrease. The convergence of initialization with $\mathbf{W}^{(0)} = \mathbf{0}$ in terms of normalized maximum sidelobe level and normalized eigenvalue residual $r^{(k)}$ versus iterations is plotted in Fig. 8. When a certain $\mathbf{X}^{(k)}$ is not rank-1, we extract the eigenvector corresponding to its maximum eigenvalue, reconstruct the baseband signal based on that, and normalize it to calculate the normalized maximum sidelobe level. We see that the maximum sidelobe level reduces only after a few iterations, and converges to the same level for different values of ζ, as long as ζ is suitably chosen. When ζ is too large (beyond the threshold discussed earlier in this section), however, the maximum sidelobe level is poorer. Initialization with $\mathbf{W}^{(0)} = \mathbf{0}$ shows a conspicuous advantage as compared to the curves in Fig. 7. We summarize the maximum sidelobe levels for different optimization schemes in Table 2. As can be seen, our iterative optimization method (with appropriate initialization) can achieve lower sidelobe levels than interior point method or simulated annealing. Based on our observations, setting $\mathbf{W}^{(0)} = \mathbf{0}$ generally outperforms other initializations for USSM.

Conventional signals such as m-sequences, PMMS and other PN-codes of length 7 have a normalized maximum sidelobe level of about 0.7-0.8. As compared to that, a normalized maximum sidelobe level of 0.2850 is an improvement of about 4 dB. This can lead to significant improvement in detection performance in sonar, radar and communication systems. The exact improvement obtained depends on the signal length, frequency and bandwidth of operation, and the delay-Doppler region over which the optimization is performed. Generally, the smaller the region, the larger the improvement expected.

6.4. Design of longer codes with p-USSM

We next consider the design of a 31-digit code using the p-USSM method with $p = 0.01$. The convergence of the design method initialized with URDP code and PMMS are respectively plotted in Fig. 9 and Fig. 10, in terms of normalized maximum sidelobe level versus iterations for different values of the parameter ζ. We ensure that ζ is below the threshold for the rank-1 requirement as in USSM, where the sidelobe level is expected to monotonically decrease. We observe from Figs. 9 and 10 that the statistical optimization method converges noisily to a good optimum when ζ is suitably

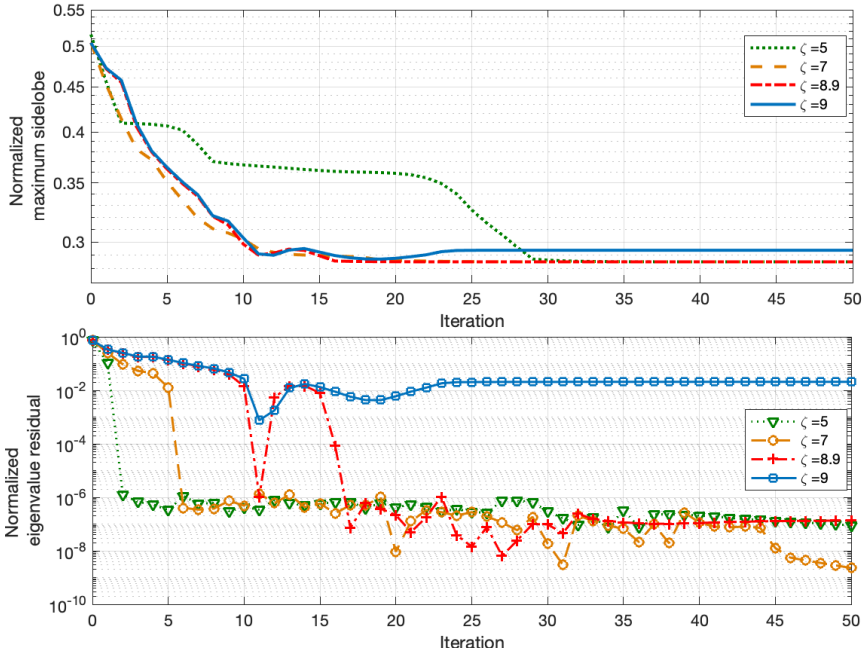


Figure 8: Convergence of USSM with $N = 7$ and $\mathbf{W}^{(0)} = \mathbf{0}$ for various values of ζ, in terms of (a) normalized maximum sidelobe level, and (b) normalized eigenvalue residual.

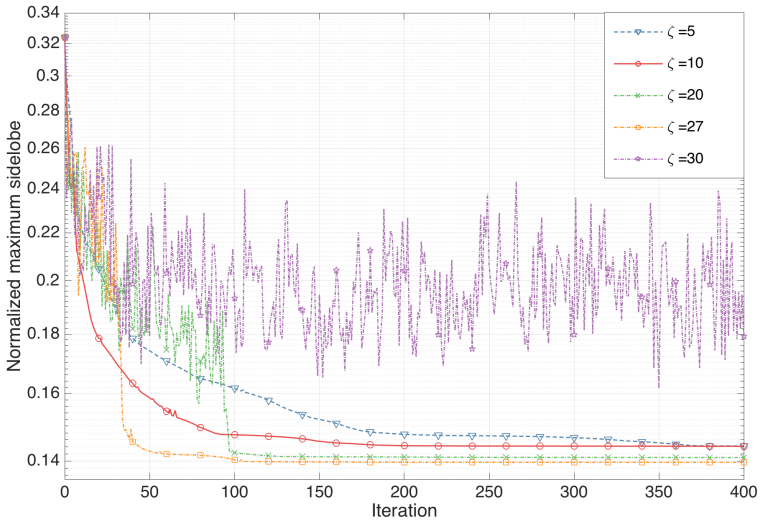


Figure 9: Convergence of p-USSM for $N = 31$ with $p = 0.01$ and URDP initialization for various values of ζ. As the value of ζ increases, the rate of convergence improves until a threshold $\zeta = 27$, after which the algorithm no longer converges.

chosen. The optimum given by the MATLAB optimization toolbox for $N = 31$ is 0.1480, whereas our statistical optimization method yields an optimum of 0.1396 (Fig. 9) and 0.1454 (Fig. 10) respectively. Using a value of ζ that is too small slows down the convergence as the optimization focuses more on meeting the rank constraint rather than lowering the sidelobe level. Nevertheless, the final sidelobe level is still comparable to that of using larger ζ, after sufficient iterations. On the other hand, using too large a value of ζ prevents convergence, as this allows the optimizer to take steps large enough to reshape the ambiguity surface sufficiently to violate the sampling assumption made in p-USSM. This is the reason why the code optimization initialized with $\zeta = 30$ and $\zeta = 50$ in Fig. 9 and Fig. 10 respectively do not converge, but instead show erratic oscillations. The threshold of parameter ζ for the p-USSM to work is therefore lower than that for USSM. For the scenario considered in Figs. 9 and 10, $\zeta = 27$ and $\zeta = 45$ seem to be the appropriate thresholds.

We briefly study the effect of p in p-USSM by plotting the sidelobe convergence for different values of p in Fig. 11. We see that the method converges for sufficiently large values of p (e.g., the recommended $p = 0.01$ from section 5), but fails to converge if p becomes too small. The threshold for p depends on ζ and other signal design parameters, and has to be determined empirically for a given application. However, $p = 0.01$ is a conservative recommendation that can serve as a guideline for most problems.

Solving each iteration of p-USSM took 3.4 seconds on an average for computation on a typical Desktop computer. For the same problem and the same computer, solving the non-convex formulation using the MATLAB optimization toolbox took $13,602$ seconds. Since our method typically converges in less than 100 iterations, we see a $40\times$ computational performance gain in the use of our method, even for small $N = 31$. For larger values of N, the MATLAB optimization toolbox is unable to solve the problem,

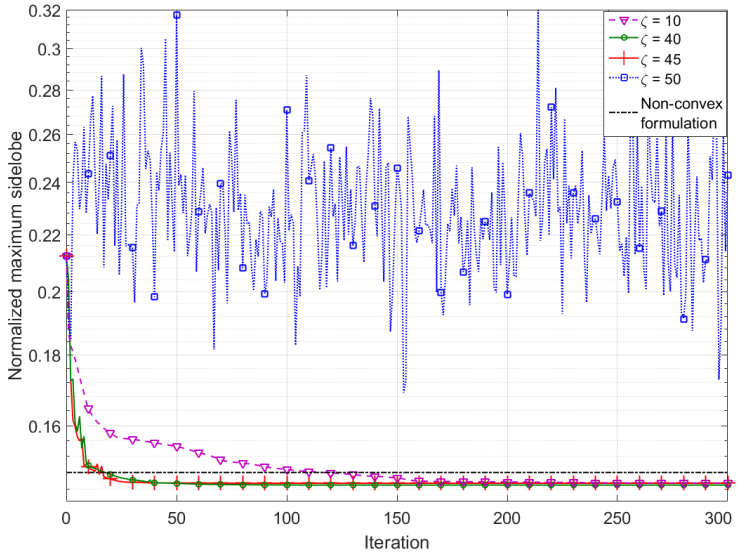


Figure 10: Convergence of p-USSM for $N = 31$ with $p = 0.01$ and PMMS initialization for various values of ζ. As the value of ζ increases, the rate of convergence improves until a threshold $\zeta = 45$, after which the algorithm no longer converges.

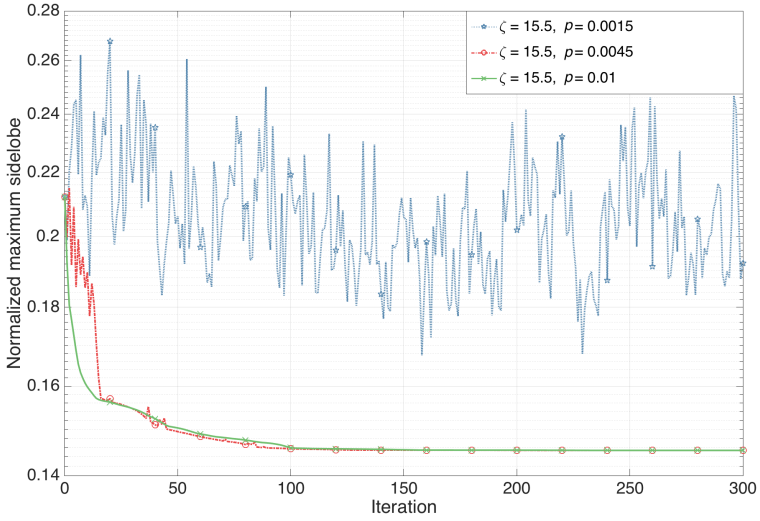


Figure 11: Convergence of p-USSM with $\zeta = 15.5$ and PMMS initialization for various values of p.

whereas p-USSM scales well. Note that some other noteworthy rank-relaxation methods for signal design, e.g., the randomization based low-rank approximation in [31], require generation of the globally optimal matrix without the rank constraint as an initial step. This may be computationally intensive when the signal length is large. With p-USSM, we can start with any unimodular signal and obtain a solution with desired sidelobe

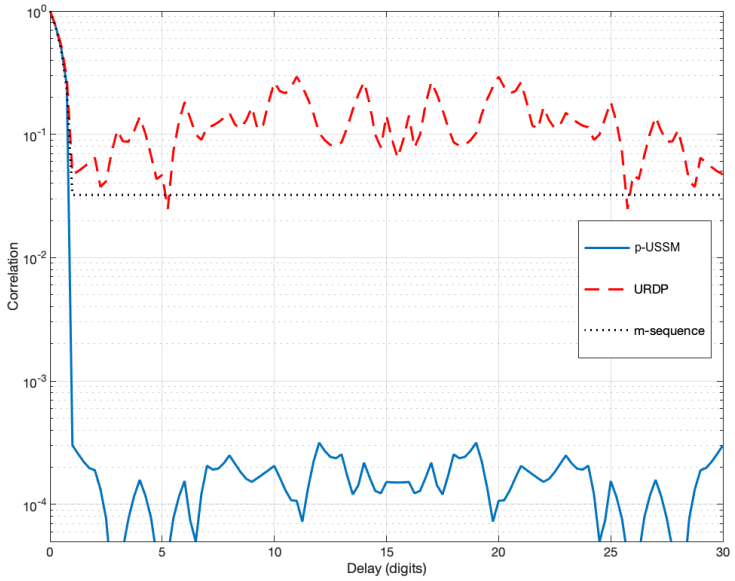


Figure 12: Comparison of autocorrelation of designed signal with $N = 31$ against other pseudorandom codes.

properties. If a globally optimal matrix is available, p-USSM can also use that as the initial point and find an optimal or near-optimal signal.In section 6.5, we present results for a signal generated with $N = 255$ using p-USSM.

We now benchmark p-USSM against other unimodular codes such as m-sequences and PMMS [7], since these are known to be well suited for applications with stationary targets (no Doppler). We do so by designing a 31-digit signal by minimizing the sidelobe levels over the zero-Doppler channel only, with $p = 0.01$ and URDP initialization. For $N = 2^n - 1$ where n is an integer, a PMMS yields ideal autocorrelation properties with zero sidelobe level, while the regular m-sequence yields a sidelobe level of $1/N$. In Fig. 12, we plot the cyclic autocorrelation of the designed signal and compare it against a regular m-sequence and URDP code. We see that our design method is able to find a unimodular code with sidelobe level of about 3.1×10^{-4}. This is lower than that of a URDP code and a regular m-sequence, but not as perfect as PMMS. With an increased Doppler range, however, our signals have significant advantage over PMMS, as we show in the next section.

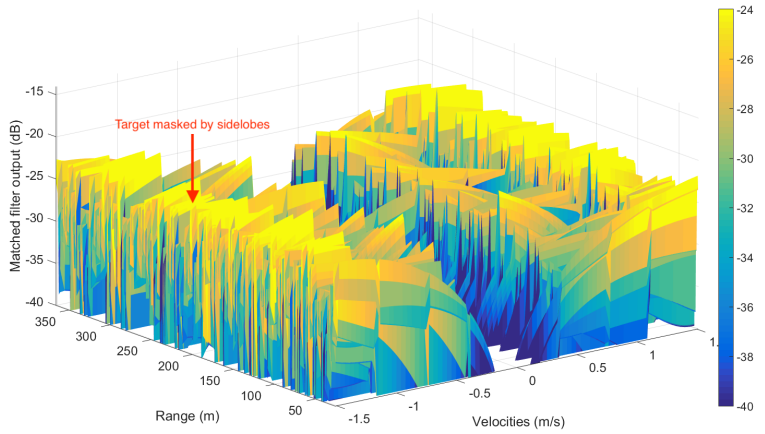
6.5. Application example: Continuous Active Sonar

We now depict an application scenario for our signal design method using a much longer signal ($N = 255$). Consider a low-frequency CAS scenario, where the arrivals of the target echoes overlap with the transmitted signal which is of much larger amplitude. In this scenario, the sidelobe floor due to the direct blast must be sufficiently low if we are to be able to detect the target in the ambiguity surface of the data. We consider a

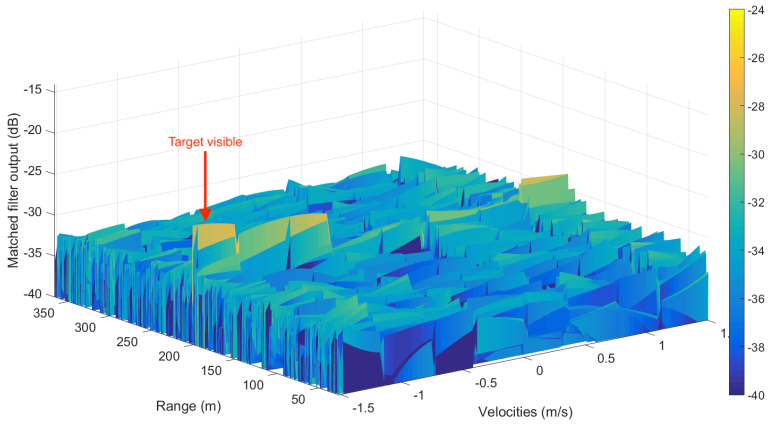
scenario with an underwater target located at 200 m moving away at a velocity of 1.5 m/s with respect to the sonar. A 255-digit unimodular signal is employed to detect the target, whose echoes are received at a level of -24 dB with respect to the transmitted signal. In Fig. 13, we plot the delay-Doppler matched-filter output of the received data, when a (a) PMMS and (b) p-USSM unimodular signal, is used in the CAS transmissions. The matched-filter output is normalized with the peak value. We see that the target at 200 m is clearly visible in the matched-filter output for the p-USSM unimodular signal, whereas it is not discernible for PMMS. The PMMS has good autocorrelation properties that translate to very favorable sidelobe levels along the zero-Doppler axis (along the zero velocity line in Fig. 13(a)), but high sidelobes at other Doppler values. These high sidelobes mask the target return. The p-USSM unimodular signal has a more uniform distribution of sidelobes, but with lower worst-case sidelobes. This enables better detection of a target masked by the direct blast.

7. Conclusions

We formulated an optimization problem to minimize the maximum sidelobe levels of unimodular signals over a set of delay-Doppler values. While this problem is non-convex and difficult to solve, we were able to demonstrate that an iterative near-optimal method (USSM) of solving the problem can yield signals with desirable properties. For longer signals, we proposed a statistical approach (p-USSM) to further reduce the computational complexity of the problem, and showed that it is able to generate good signals much faster. The proposed USSM and p-USSM algorithms are controlled by two free parameters ζ and p. We studied the effect of both parameters, and suggested ways to tune them. For applications where only delay is of interest, and not Doppler, the optimal unimodular signals are known. We benchmarked p-USSM against those signals for that application. For applications where Doppler is significant, we demonstrated an reduction in maximum sidelobe levels of up to 4 dB (depending on signal parameters such as length, frequency, bandwidth, etc) over a delay-Doppler region of interest. Finally, we demonstrated the effectiveness of p-USSM designed signals in a CAS application, where the target sonar return is buried in a strong direct blast from the transmission. The p-USSM designed 255-digit unimodular signal was able to clearly see the target, while an equivalent length PMMS failed to discern it, demonstrating the advantage of our signals as compared to traditional signals.



(a) PMMS



(b) p-USSM unimodular signal

Figure 13: Matched-filter output of simulated CAS data containing returns from a target at 200 m and moving at -1.5 m/s, when a (a) 255-digit PMMS and (b) 255-digit p-USSM unimodular signal is used in the CAS transmissions. The target is clearly visible in (b) but not in (a), demonstrating the advantage of our signals as compared to traditional signals.

Appendix A.

From (5), we have:

$$\bar{A}_{\alpha\delta_{jk}} = K_{\alpha\delta} \int_0^{NT} G_{\alpha\delta_{jk}}(t)\, \mathrm{e}^{\mathrm{i}\omega\alpha t}\, dt,$$

where:

$$\begin{aligned} G_{\alpha\delta_{jk}}(t) &= g(t - jT)g(\bar{\alpha}t - \bar{\alpha}\delta T - kT), \\ K_{\alpha\delta} &= \frac{\mathrm{e}^{-\mathrm{i}\omega\bar{\alpha}\delta T}\sqrt{\bar{\alpha}}}{NT}, \text{and} \\ \bar{\alpha} &= 1 + \alpha. \end{aligned}$$

When $g(t)$ is the rectangular pulse as in (6), $G_{\alpha\delta_{jk}}(t)$ simplifies to:

$$G_{\alpha\delta_{jk}}(t) = \begin{cases} 1 & \text{if } t_1 \le t < t_2 \\ 0 & \text{otherwise,} \end{cases}$$

where:

$$\begin{aligned} t_1 &= \max\{\, jT,\ (k/\bar{\alpha} + \delta)T,\ 0\,\}, \text{and} \\ t_2 &= \min\{\, (j+1)T,\ ((k+1)/\bar{\alpha} + \delta)T,\ NT\,\}. \end{aligned}$$

Hence,

$$\bar{A}_{\alpha\delta_{jk}} = \begin{cases} \dfrac{\mathrm{i}\,\mathrm{e}^{-\mathrm{i}\omega\bar{\alpha}\delta T}\sqrt{\bar{\alpha}}}{NT\omega\alpha}(\mathrm{e}^{\mathrm{i}\omega\alpha t_2} - \mathrm{e}^{\mathrm{i}\omega\alpha t_1}) & \text{if } t_1 < t_2 \text{ and } \alpha \neq 0 \\ \dfrac{\mathrm{e}^{-\mathrm{i}\omega\delta T}}{NT}(t_2 - t_1) & \text{if } t_1 < t_2 \text{ and } \alpha = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Appendix B.

In section 2 we formulated the optimization problem over a delay-Doppler region of interest $\mathcal{Q}$. However, since we can only optimize over a discretized set $\{(\alpha, \delta) \in \mathcal{Q}\}$, it is important to select an appropriate sampling rate so that no significant sidelobe is missed. The discrete samples calculated in (4) are in perfect agreement with χ at those grid points, hence can be written as:

$$\chi(j,k) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \chi(\alpha,\delta)\phi\left(\frac{\alpha}{T_\alpha} - j\right)\phi\left(\frac{\delta}{T_\delta} - k\right)\, \mathrm{d}\alpha\mathrm{d}\delta\,,$$

where ϕ is the Dirac delta function, (j,k) denotes the two dimensional indices of the discrete samples, and T_α, T_δ are the sampling steps in Doppler and delay domain respectively. Now if we reconstruct another two-dimensional function $\tilde{\chi}$ via linear interpolation:

$$\tilde{\chi}(\alpha,\delta) = \sum_j \sum_k \chi(j,k)\psi\left(\frac{\alpha}{T_\alpha} - j\right)\psi\left(\frac{\delta}{T_\delta} - k\right)\,,$$

where $\psi(t)$ is the tent function, $\tilde{\chi}$ is therefore piecewise linear, and the largest sidelobe $\max_{(\alpha,\delta)\in\mathcal{Q}}\{\chi(\alpha,\delta)\}$ is guaranteed to be the maximum of $\tilde{\chi}$ too. The above procedure is called quasi-interpolation that has been well studied in approximation theory. The ℓ_2 approximation error can be computed exactly by a simple integration between the Fourier transform of the function to approximate and an error kernel determined by the Fourier transforms of ϕ and ψ [41]. Although the result is derived for a single dimensional function, it can carry over directly for our multidimensional function by analyzing the time and frequency domains separately, because all filtering and interpolation operations are separable. The error kernel calculated using ϕ and ψ and its decay with ideal band-limited signals is given in [41]. In our case, we do not apply any pre-filtering to get $\chi(j,k)$. Nevertheless, the out-of-band portion will reduce as the sampling steps become smaller.

Here we give a rule of thumb for selecting the sampling rate: four times of the Nyquist bandwidth or its equivalent. This guarantees no sidelobe level significantly larger than our optimized maximum value, even if the sampled grids fail to capture the exact largest one, as the piecewise linear AF is a fairly good approximation of the actual AF.

Appendix C.

For a given ζ_q, assuming the solution to the k^{th} iteration with direction matrix $\mathbf{W}^{(k)}$ is $\mathbf{X}_q^{(k)}$, which is solely determined by ζ_q, the objective cost $z_{\max}^{(k)}$ and the regularization term can be respectively denoted as

$$\begin{aligned} z_{\max_q}^{(k)} &= \max_{(\alpha,\delta)\in\mathcal{Q}} \sqrt{\left\langle \hat{\mathbf{A}}_{\alpha\delta}, \mathbf{X}_q^{(k)} \right\rangle^2 + \left\langle \check{\mathbf{A}}_{\alpha\delta}, \mathbf{X}_q^{(k)} \right\rangle^2}\,, \\ \text{and} \quad w_q^{(k)} &= \left\langle \mathbf{W}^{(k)}, \mathbf{X}_q^{(k)} \right\rangle, \end{aligned}$$

where the subscript $\cdot_q$ implies that $z_{\max_q}^{(k)}$ and $w_q^{(k)}$ are functions with respect to ζ_q with $\mathbf{W}^{(k)}$ fixed. As ζ_q decreases, $z_{\max_q}^{(k)}$ monotonically increases whereas $w_q^{(k)}$ decreases.

Proof. Suppose $\zeta_1 > \zeta_2$ with corresponding $(z_{\max_1}^{(k)}; w_1^{(k)})$ and $(z_{\max_2}^{(k)}; w_2^{(k)})$ respectively. Since the problem is convex, any local minimum is guaranteed to be the global minimum and therefore results in a cost no greater than all other feasible solutions. Thus we have the inequalities:

$$\zeta_1 z_{\max_1}^{(k)} + w_1^{(k)} \le \zeta_1 z_{\max_2}^{(k)} + w_2^{(k)}, \tag{C.1}$$

$$\zeta_2 z_{\max_2}^{(k)} + w_2^{(k)} \le \zeta_2 z_{\max_1}^{(k)} + w_1^{(k)}. \tag{C.2}$$

Multiplying both sides of (C.2) with a minus sign and adding it up with (C.1) yields

$$\begin{aligned} (\zeta_1 - \zeta_2) z_{\max_1}^{(k)} &\le (\zeta_1 - \zeta_2) z_{\max_2}^{(k)}, \\ \Rightarrow \quad z_{\max_1}^{(k)} &\le z_{\max_2}^{(k)}. \end{aligned}$$

Therefore from (C.2),

$$w_2^{(k)} \le \zeta_2 (z_{\max_2}^{(k)} - z_{\max_1}^{(k)}) + w_1^{(k)} \le w_1^{(k)}.$$

In other words, decreasing ζ to meet the rank constraint sacrifices the superiority of the optimal cost.

[1] J. Benedetto, I. Konstantinidis, and M. Rangaswamy, "Phase-Coded Waveforms and Their Design," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 22–31, jan 2009.
[2] S. Haykin, "Cognitive radar: a way of the future," *IEEE Signal Processing Magazine*, vol. 23, no. January, pp. 30–40, 2006.
[3] P. Stoica, J. Li, and M. Xue, "Transmit codes and receive filters for radar: A look at the design process," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 94–109, 2008.
[4] J. Ling, H. He, J. Li, W. Roberts, and P. Stoica, "Covert underwater acoustic communications," *The Journal of the Acoustical Society of America*, vol. 128, no. 5, pp. 2898–2909, nov 2010.
[5] T. Dean, "The use of pseudorandom sweeps for vibroseis surveys," *Geophysical Prospecting*, vol. 62, no. 1, pp. 50–74, 2014.
[6] N. Levanon and E. Mozeson, *Radar signals*. John Wiley & Sons Ltd, 2004.
[7] H. Chang, "Detection of weak, broadband signals under Doppler-scaled, multipath propagation," in *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1992, pp. 461–464 vol.2.
[8] M. Naghsh, M. Soltanalian, P. Stoica, and M. Modarres-Hashemi, "Radar Code Design for Detection of Moving Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 2762 – 2778, 2014.
[9] M. Soltanalian and P. Stoica, "On prime root-of-unity sequences with perfect periodic correlation," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5458–5470, 2014.
[10] M. Soltanalian, P. Stoica, M. M. Naghsh, and A. De Maio, "Design of piecewise linear polyphase sequences with good correlation properties," in *22nd European Signal Processing Conference (EUSIPCO)*. IEEE, 2014, pp. 1297–1301.
[11] S. Kay, "Optimal signal design for detection of Gaussian point targets in stationary Gaussian clutter/reverberation," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 1, pp. 31–41, 2007.
[12] P. Stoica, H. He, and J. Li, "New algorithms for designing unimodular sequences with good correlation properties," *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1415–1425, 2009.
[13] M. Soltanalian, B. Tang, J. Li, and P. Stoica, "Joint design of the receive filter and transmit sequence for active sensing," *IEEE Signal Processing Letters*, vol. 20, no. 5, pp. 423–426, 2013.
[14] M. Alaee Kerahroodi, A. Aubry, A. De Maio, M. M. Naghsh, and M. Modarres-Hashemi, "A Coordinate-Descent Framework to Design Low PSL/ISL Sequences," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5942–5956, nov 2017.
[15] A. De Maio, S. De Nicola, Yongwei Huang, Shuzhong Zhang, and A. Farina, "Code Design to Optimize Radar Detection Performance Under Accuracy and Similarity Constraints," *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5618–5629, nov 2008.
[16] A. De Maio, S. De Nicola, Yongwei Huang, Zhi-Quan Luo, and Shuzhong Zhang, "Design of Phase Codes for Radar Performance Optimization With a Similarity Constraint," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 610–621, feb 2009.
[17] A. De Maio, Y. Huang, and M. Piezzo, "A Doppler robust max-min approach to radar code design," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4943–4947, 2010.
[18] A. De Maio, Y. Huang, M. Piezzo, S. Zhang, and A. Farina, "Design of optimized radar codes with a peak to average power ratio constraint," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2683–2697, 2011.
[19] M. Soltanalian and P. Stoica, "Designing unimodular codes via quadratic optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1221–1234, 2014.
[20] A. Aubry, A. De Maio, B. Jiang, and S. Zhang, "Ambiguity function shaping for cognitive radar via complex quartic optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5603–5619, 2013.
[21] H. He, P. Stoica, and J. Li, "On synthesizing cross ambiguity functions," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 3536–3539.
[22] A. Aubry, A. Demaio, A. Farina, and M. Wicks, "Knowledge-aided (potentially cognitive) transmit signal and receive filter design in signal-dependent clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, pp. 93–117, 2013.
[23] A. Aubry, A. De Maio, and M. M. Naghsh, "Optimizing Radar Waveform and Doppler Filter Bank via Generalized Fractional Programming," *IEEE Journal on Selected Topics in Signal Processing*, vol. 9, no. 8, pp. 1387–1399, 2015.
[24] H. Hu, M. Soltanalian, P. Stoica, and X. Zhu, "Locating the Few: Sparsity-Aware Waveform Design for Active Radar," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 651–662, 2017.
[25] M. Naghsh, M. Soltanalian, P. Stoica, M. Modarres-Hashemi, A. De Maio, and A. Aubry, "A

Doppler Robust Design of Transmit Sequence and Receive Filter in the Presence of Signal-Dependent Interference," *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 772–785, feb 2014.

[26] M. Naghsh, M. Soltamalian, P. Stoica, M. Modarres-Hashemi, A. De Maio, and A. Aubry, "A max-min design of transmit sequence and receive filter," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2014, pp. 71–75.

[27] M. Soltanalian, M. Naghsh, and P. Stoica, "On meeting the peak correlation bounds," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1210–1220, 2014.

[28] W. H. Thorp, "Analytic Description of the Low-Frequency Attenuation Coefficient," *The Journal of the Acoustical Society of America*, vol. 42, no. 1, p. 270, 1967. [Online]. Available: http://link.aip.org/link/JASMAN/v42/i1/p270/s1&Agg=doi

[29] L. Vandenberghe and S. Boyd, "Semidefinite Programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, mar 1996.

[30] R. A. Delgado, J. C. Agüero, and G. C. Goodwin, "A Rank-Constrained Optimization approach: Application to Factor Analysis," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 10 373–10 378, 2014.

[31] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

[32] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, no. 6, pp. 1115–1145, 1995.

[33] S. Zhang and Y. Huang, "Complex Quadratic Optimization and Semidefinite Programming," *SIAM Journal on Optimization*, vol. 16, no. 3, pp. 871–890, 2006.

[34] M. Fazel, H. Hindi, and S. Boyd, "Rank minimization and applications in system theory," *Proceedings of the American Control Conference*, vol. 4, pp. 3273–3278, 2004.

[35] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[36] V. S. Mai, D. Maity, B. Ramasubramanian, and M. C. Rotkowitz, "Convex methods for rank-constrained optimization problems," in *2015 Proceedings of the Conference on Control and its Applications*. SIAM, 2015, pp. 123–130.

[37] J. Dattoro, *Convex optimization & Euclidean distance geometry*. Lulu.com and $\mathcal{M}\epsilon\beta oo$, 2010.

[38] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: a survey and extensions," *Mathematical Methods of Operations Research*, vol. 66, no. 3, pp. 373–407, 2007.

[39] D. G. Luenberger and Y. Ye, "Conic linear programming," in *Linear and Nonlinear Programming*. Springer, 2016, pp. 149–176.

[40] MOSEK ApS, "MOSEK MATLAB toolbox Version 8.0.0.81," 2017. [Online]. Available: http://docs.mosek.com/8.0/toolbox.pdf

[41] M. Unser, "Sampling: 50 Years After Shannon," *Proc of the IEEE*, vol. 88, no. 4, pp. 569–587, 2000.