UC Irvine
UC Irvine Previously Published Works

Title

Relaxed Biquadratic Optimization for Joint Filter-Signal Design in Signal-Dependent STAP

Permalink

https://escholarship.org/uc/item/02213090

Journal

IEEE Transactions on Signal Processing, 66(5)

ISSN

1053-587X

Authors

O'Rourke, Sean M
Setlur, Pawan
Rangaswamy, Muralidhar
et al.

Publication Date

2018

DOI

10.1109/tsp.2017.2775592

Peer reviewed

Relaxed Bi-quadratic Optimization for Joint Filter-Signal Design in Signal-Dependent STAP

Sean M. O'Rourke, *Member, IEEE,* Pawan Setlur, *Member, IEEE,*

Muralidhar Rangaswamy, *Fellow, IEEE,* A. Lee Swindlehurst, *Fellow, IEEE*

Abstract

We investigate an alterative solution method to the joint signal-beamformer optimization problem considered by Setlur and Rangaswamy [1]. First, we directly demonstrate that the problem, which minimizes the recieved noise, interference, and clutter power under a minimum variance distortionless response (MVDR) constraint, is generally non-convex and provide concrete insight into the nature of the nonconvexity. Second, we employ the theory of biquadratic optimization and semidefinite relaxations to produce a relaxed version of the problem, which we show to be *convex*. The optimality conditions of this relaxed problem are examined and a variety of potential solutions are found, both analytically and numerically.

I. Introduction

In 1972, at a NATO conference in the East Midlands of England, two luminaries of signal processing agreed that an adaptive system was incapable of being simultaneously spatially and temporally optimal. Agreeing with M. Mermoz's proposition, the Naval Underwater System Center's Norman Owsley asserted that since "the spectrum of the signal must be known *a priori*" for temporal optimality, "the post beamformer temporal processor cannot be fully adaptive and fully optimum (sic) simultaneously" [2]. In many ways, this fundamental philosophy has carried forward in multichannel signal processing and control theory unabated in the last 40-plus years since that conference in Loughborough. Under the fully-adaptive radar paradigm, however, it is necessary to find transmit and receive resources that *are* simultaneously optimal, or at least optimal under prior knowledge assumptions. With these additional degrees of freedom, we can in fact develop a result that defies this wisdom. The purpose of this report is to document such an attempt, which is a significant extension of the work of [1] in joint waveform-filter design for radar space-time adaptive processing (STAP), that challenges the pre-existing status quo.

S. M. O'Rourke is with the Sensors Directorate, US AFRL, WPAFB, OH and the University of California, Irvine. Email: sean.orourke.3@us.af.mil

M. Rangaswamy is with the Sensors Directorate, US AFRL, WPAFB, OH. Email: muralidhar.rangaswamy@us.af.mil

P. Setlur is affiliated with the Wright State Research Inst. as a research contractor for the US AFRL, WPAFB, OH. Email: pawan.setlur@wright.edu

A. L. Swindlehurst is with the University of California, Irvine. Email: swindle@uci.edu

Approved for Public Release, No.: 88ABW-2016-5820

In the general scenario, we assume that an airborne radar equipped with an array of sensing elements observes a moving target on the ground. Furthermore, this radar can change its transmitted waveform every coherent processing interval (CPI), instead of on a per-pulse basis. In order to develop a strategy for waveform design, we consider a STAP model that includes fast-time (range) samples as well as the usual slow-time and spatial samples. This is a departure from traditional STAP, which operates on spatiodoppler responses from the radar after matched filtering [3], [4], but recent advances in the literature have considered incorporating fast-time data for more accurate clutter modeling, both transmitter- and jammer-induced [5], [6].

An obvious result of including the fast-time data in the model is that the clutter representation is now signal-dependent, since in airborne STAP, the dominant clutter source is non-target ground reflections that persist over all range bins. We assume that, if modeled as a random process, the clutter is uncorrelated with any interference or noise (which, unlike the clutter, are assumed to have no signal dependence), and the related clutter correlation matrix is also signal dependent. If we formulate our waveform-filter design in the typical minimum variance distortionless-response (MVDR) framework [7], this dependence in the correlation structure leads to what multiple authors ([8]–[10], among others) have empirically or intuitively identified as a non-convex optimization problem. However, the authors in [1] correctly identified that for a fixed transmit waveform, the problem is convex in the receive filter, and vice versa. This led to a collection of design algorithms based on alternating minimization, a reasonable heuristic.

While the alternating minimization method is useful, it has no claims of optimality, nor do other methods in the literature that dealt with signal-dependent interference in other contexts, like the single sensor radar and reverberant channel of [11]. Furthermore, none of these authors directly proved or cited anything that demonstrated the non-convexity of the problem or its engineering consequences. We will ameliorate some of these concerns in this paper, first by directly proving the non-convexity of the objective, then relating it to the existing literature on *biquadratic programming* (BQP). While the biquadratic program is demonstrably non-convex, it is possible to relax it to a convex quadratic program using semidefinite programming (SDP) (see [12] for details on SDP and [13], [14] on relaxing the BQP). This relaxation will permit us to efficiently solve the problem computationally and analytically (up to a matrix completion), as well as reveal important structural information about the solution that matches with engineering intuition.

Before continuing, we outline some mathematical notation that will be used throughout the paper. The symbol $\mathbf{x} \in \mathbb{R}^{N \times 1}(\mathbf{x} \in \mathbb{C}^{N \times 1})$ indicates a column vector of real (complex) values, while $\mathbf{X} \in \mathbb{R}^{M \times N}(\mathbf{X} \in \mathbb{C}^{M \times N})$ indicates an N row, M column matrix of real (complex) values. The superscripts T, $*$, and H indicate the transpose, conjugate, and Hermitian (conjugate) transpose of a matrix or vector. The symbol $\otimes$ indicates a Kronecker product. The operator vec turns a matrix into a vector with the matrix stacked columnwise – that is, for $\mathbf{A} \in \mathbb{R}^{p \times q}, \mathrm{vec}(\mathbf{A}) \in \mathbb{R}^{pq \times 1}$. Special matrices that will recur often include:

- the $n \times n$ identity matrix $\mathbf{I}_n$;
- the $n \times 1$ vector of all ones $\mathbf{1}_n$;
- the $n \times m$ all-zero matrix $\mathbf{0}_{n \times m}$;
- and, via Magnus & Neudecker [15], the (p, q) *commutation matrix* $\mathbf{K}_{p,q} \in \mathbb{R}^{pq \times pq}$ that is non-uniquely

determined by the relation $\mathrm{vec}(\mathbf{A}^T) = \mathbf{K}_{p,q} \, \mathrm{vec}(\mathbf{A})$ for $\mathbf{A}$ as above. $\mathbf{K}_{p,q}$ is an orthogonal permutation matrix (*i.e.*, $\mathbf{K}_{p,q}^T = \mathbf{K}_{p,q}^{-1}$). Transposing the matrix swaps the indices as well ($\mathbf{K}_{p,q}^T = \mathbf{K}_{q,p}$), and clearly then $\mathbf{K}_{p,q}\mathbf{K}_{q,p} = \mathbf{I}_{pq}$. Additionally, $\mathbf{K}_{1,q} = \mathbf{I}_q$ and $\mathbf{K}_{p,1} = \mathbf{I}_p$. We will also frequently use the one-index form $\mathbf{K}_p = \mathbf{K}_{p,p}$ for the square commutation matrix.

We will often use calligraphic letters like $\mathcal{C}$ to indicate tensors or multilinear operators (like the Hessian matrix). Additionally, the vectorizations of various matrices will be indicated by the equivalent lowercase Greek symbol in bold – for example, a matrix $\mathbf{B}$ will have the vectorization $\boldsymbol{\beta} = \mathrm{vec}(\mathbf{B})$.

The rest of the paper continues as follows: First, we will outline the general signal model and relate it to the work on transfer functions by [16]. Next, we will define the joint design problem and demonstrate, after some transformation, that the problem is non-convex unless all clutter patches are known to be nulled a priori. Section IV will describe the relationship with the biquadratic problem and methods to relax the joint design into a convex quadratic semidefinite program. We then attempt to find analytic solutions and insights into this relaxed problem in Section V. These solutions are generally verified in Section VI through simulation, as well as demonstrate that the resulting rank-one solution performs similarly to the alternating minimization. Finally, we summarize our conclusions and highlight future research paths in Section VII.

II. STAP Model

Consider a radar that consists of a calibrated airborne linear array of M identical sensor elements, where the first element is the phase center and also acts as the transmitter. During the transmission period, the radar probes the environment with a number of pulses $s(t)$ of width T seconds and bandwidth B hertz at a carrier frequency f_o. Within each burst, we transmit L pulses at a rate of f_p (i.e. the pulses are transmitted every $T_p = 1/f_p$ seconds) and collect them in a coherent processing interval (CPI). We also assume that the phase center is located at $\mathbf{x}_r$ and the platform moves at a rate that is relatively constant over the CPI.

Assume the probed environment contains a target that lies at an azimuth θ_t and an elevation ϕ_t relative to the array phase center, moving with a relative velocity vector $\boldsymbol{\delta v} = \begin{bmatrix} \delta v_x & \delta v_y & \delta v_z \end{bmatrix}^T$. If we assume that the array interelement spacing d is small relative to the distance between the platform and the target, then the target's doppler shift is independent of the element index and given by

$$f_d = 2f_o \frac{\boldsymbol{\delta v}^T [\sin(\phi_t)\sin(\theta_t) \ \sin(\phi_t)\cos(\theta_t) \ \cos(\phi_t)]}{c} \tag{1}$$

where c is the usual speed of light.

Let us now assume that we discretely sample the pulse $s(t)$ into N samples, resulting in the sample vector $\mathbf{s} = [s_1, s_2, \cdots, s_N] \in \mathbb{C}^N$. Assuming the data is aligned to a common reference and given other assumptions from [27], we can say that at the target range gate τ_t, the combined target response over the entire CPI can be represented by a vector $\mathbf{y}_t \in \mathbb{C}^{NML}$ given by

$$\mathbf{y_t} = \rho_t \mathbf{v}_t(f_d) \otimes \mathbf{s} \otimes \mathbf{a}_t(\theta_t, \phi_t) \tag{2}$$

where ρ_t is the complex backscattering coefficent from the target, the vector $\mathbf{v}_t(f_d) \in \mathbb{C}^L$ is the doppler steering vector whose ith element is given by $e^{-\jmath 2\pi f_d(i-1)T_p}$, and the vector $\mathbf{a}_t(\theta_t, \phi_t) \in \mathbb{C}^M$ is the spatial steering vector

whose ith element is given by $e^{-j2\pi(i-1)\vartheta}$ where $\vartheta = d\sin(\theta_t)\sin(\phi_t)f_o/c$. In this form, our departure from traditional STAP is clear, given the dependence on the waveform $\mathbf{s}$.

Since no radar operates in an ideal environment, the target return $\mathbf{y}_t$ is corrupted by a variety of undesired returns from the environments – noise, signal-independent interference, and clutter. We can consider the overall return in the target range gate as an additive model:

$$\tilde{\mathbf{y}} = \mathbf{y_t} + \mathbf{y_n} + \mathbf{y_i} + \mathbf{y_c} = \mathbf{y_t} + \mathbf{y_u}, \tag{3}$$

where the subscripts n, i, and c stand for noise, interference, and clutter, respectively, and these are all collected in the Undesired signal term. We assume that these undesired energy sources are statistically uncorrelated from each other, with unique distributions. We shall subsequently describe each of these sources, starting with the noise.

We assume the noise is zero mean and identically distributed across all sensors, pulses, and fast time samples. The covariance matrix of $\mathbf{y}_n$ is given by $\mathbf{R_n} \in \mathbb{C}^{NML \times NML}$.

The interference term consists of jammers and other spurious emitters that are intentional or unintentional and are ground-based, airborne, or both. We assume that there are K known interference sources, but otherwise we have no knowledge of what they transmit into the surveillance region, and (we hope) they are not dependent on our transmitted signal. Thus, we model their contributions as a zero-mean random process spread over all pulses and fast time samples. Assume that the kth interferer is located at the azimuth-elevation pair (θ_k, ϕ_k). In the lth PRI, we assume that the waveform is a complex continuous-time signal $\alpha_{kl}(t)$. Under the same sampling scheme, this transforms into a vector $\boldsymbol{\alpha}_{kl} \in \mathbb{C}^N$, similar in form to $\mathbf{s}$. Stacked across all PRIs, we obtain another random vector $\boldsymbol{\alpha}_k = [\boldsymbol{\alpha}_{k0}^T \, \boldsymbol{\alpha}_{k1}^T \, \cdots \, \boldsymbol{\alpha}_{k(L-1)}^T]^T \in \mathbb{C}^{NL}$, whose covariance matrix we define as $\mathbf{E}\{\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^H\} = \mathbf{R}_{\boldsymbol{\alpha},k}$. Then, the response from the kth interferer can be modeled as

$$\mathbf{y_{i},k} = \boldsymbol{\alpha}_k \otimes \mathbf{a_i}(\theta_k, \phi_k) \tag{4}$$

where, as with the target, $\mathbf{a}_i(\theta_k, \phi_k)$ is the array response to the interferer. The covariance of $\mathbf{y}$ is then $\mathbf{R}_{\boldsymbol{\alpha},k} \otimes \mathbf{a_i}(\theta_k, \phi_k)\mathbf{a_i}(\theta_k, \phi_k)^H$, since expectations follow through Kronecker products that don't have a random dependence. If we assume that each of the interferers is uncorrelated with each other, then the overall covariance of the combined signal-independent interference $\mathbf{y_i}$ is

$$\mathbf{R_i} = \sum_{k=1}^{K} \mathbf{R}_{\boldsymbol{\alpha},k} \otimes \mathbf{a_i}(\theta_k, \phi_k)\mathbf{a_i}(\theta_k, \phi_k)^H. \tag{5}$$

In future sections, we will use the notation $\mathbf{R_{ni}} = \mathbf{R_n} + \mathbf{R_i}$ to denote the combined noise and interference covariance matrix, describing the second order effects of the signal-independent corruption.

Finally, we come to the clutter. In airborne radar applications, the most significant clutter source is the ground, which produces returns persistent throughout all range gates up to the horizon. Though other clutter sources exist, like large discrete objects, vegetation, and targets not currently being surveilled, the specific stochastic model we apply here only concerns ground clutter. However, we note that a later formulation in this paper *may* be amenable to considering those other sources in a manner that recalls efforts in the literature on channel estimation (see, for example, [16]). For now, let us assume we have a number of clutter patches (say, Q) each comprising P

distinct scatterers. As with the target, the return from the pth scatterer in the qth patch, located spatially at the azimuth-elevation pair (θ_{pq}, ϕ_{pq}), maintains a Kronecker structure given by

$$\gamma_{pq}\mathbf{v}(f_{\mathsf{c},pq}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_{pq}, \phi_{pq})$$

where the returned complex reflectivity is γ_{pq} and the Doppler shift observed by the platform is $f_{\mathsf{c},pq}$. This Doppler shift is solely induced by the platform motion, characterized by the aforementioned velocity vector $\dot{\mathbf{x}}_{\mathsf{r}}$, and is given by

$$f_{\mathsf{c},pq} = 2f_o \frac{\dot{\mathbf{x}}_{\mathsf{r}}^T [\sin(\phi_{pq}) \sin(\theta_{pq}) \ \sin(\phi_{pq}) \cos(\theta_{pq}) \ \cos(\phi_{pq})]}{c}. \tag{6}$$

Thus, the overall response from the qth clutter patch is

$$\mathbf{y}_{\mathsf{c},q} = \sum_{p=1}^{P} \gamma_{pq}\mathbf{v}(f_{\mathsf{c},pq}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_{pq}, \phi_{pq}). \tag{7}$$

In order to define the covariance matrix of this response, we require the qth combining matrix $\mathbf{B}_q \in \mathbb{C}^{NML \times P}$ as

$$\mathbf{B_q} = [\mathbf{v}(f_{C,1q}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_{1q}, \phi_{1q}) \ \cdots \ \mathbf{v}(f_{C,Pq}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_{Pq}, \phi_{Pq})].$$

and the covariance of the reflectivity vector $[\gamma_{1q} \ \gamma_{2q} \ \cdots \ \gamma_{Pq}]^T$ given by $\mathbf{R}_{\gamma}^{pq} \in \mathbb{C}^{P \times P}$. Then, the overall covariance matrix for the patch is

$$\mathbf{R}_{\gamma}^{q} = \mathbf{B}_q \mathbf{R}_{\gamma}^{pq} \mathbf{B}_q^{H} \tag{8}$$

If we assume that the scatterers in one patch are uncorrelated with the scatterers in any other patch, then the total clutter response is $\mathbf{y}_{\mathsf{c}} = \sum_{q=1}^{Q} \mathbf{y}_{\mathsf{c},q}$ and its covariance is given by

$$\mathbf{R}_{\mathsf{c}} = \sum_{q=1}^{Q} \mathbf{R}_{\gamma}^{q} \tag{9}$$

As above, we will denote the overall undesired response covariance matrix as $\mathbf{R}_{\mathbf{u}} = \mathbf{R}_{\mathbf{ni}} + \mathbf{R}_{\mathsf{c}}(\mathbf{s})$.

Since this is a rather cumbersome model, we can simplify our description of the clutter as follows. Let us assume that our range resolution is large enough that we cannot resolve individual scatterers in each patch – as mentioned in [1], this is typical in STAP applications. Thus, we can regard each scatterer in the patch as lying within the same range gate and having approximately equal Doppler shifts, hence $f_{\mathsf{c},pq} \approx f_{\mathsf{c},q}$. Similarly, if we assume far-field operation, the scatterers will lie in approximately the same angular resolution cell centered at (θ_q, ϕ_q), which means $\theta_{pq} \approx \theta_q, \phi_{pq} \approx \phi_q$. Given this simplification, we can modify our representation of the patch response and its covariance.

Under this assumption, the per-patch clutter response is

$$\mathbf{y}_{\mathsf{c},q} = \gamma_q \mathbf{v}(f_{\mathsf{c},q}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_q, \phi_q)$$

where $\gamma_q = \sum_{p=1}^{P} \gamma_{pq}$ is the combined reflectivity of all scatterers within the patch. For the covariance, the combining matrix for the q-th clutter patch is given by $\mathbf{B_q} = [\mathbf{v}(f_{\mathsf{c},q}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_q, \phi_q), \ldots, \mathbf{v}(f_{\mathsf{c},q}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_q, \phi_q)]$. Since the deterministic patch response is repeated P times, via standard Kronecker product properties, this is equivalent to

$$\mathbf{B_q} = \mathbf{1}_P^T \otimes \mathbf{v}(f_{\mathsf{c},q}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_q, \phi_q). \tag{10}$$

More importantly, since $\mathbf{R}_\gamma^q = \mathbf{B}_q \mathbf{R}_\gamma^{pq} \mathbf{B}_q^H$, we have

$$\mathbf{R}_\gamma^q = (\mathbf{1}_P^T \otimes \mathbf{v}(f_{\mathsf{c},q}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_q, \phi_q)) \mathbf{R}_\gamma^{pq} (\mathbf{1}_P^T \otimes \mathbf{v}(f_{\mathsf{c},q}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_q, \phi_q))^H \tag{11a}$$

$$= \overline{R}_\gamma^{pq} (\mathbf{v}_q \otimes \mathbf{s} \otimes \mathbf{a}_q)(\mathbf{v}_q \otimes \mathbf{s} \otimes \mathbf{a}_q)^H \tag{11b}$$

where $\overline{R}_\gamma^q = \mathbf{1}_P^T \mathbf{R}_\gamma^{pq} \mathbf{1}_P$.

Relationship to the Channel Model construct: In [16], the authors considered a transfer function/matrix approach for simultaneous transmit and receive resource design in MIMO radar, similar to the typical literature on control theory and digital communications. Using a rather obvious mathematical fact, we can immediately reframe our model in this form. Observe the following Kronecker mixed product property: for conformable matrices/vectors/scalars $\mathbf{A}$ through $\mathbf{F}$,

$$(\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C})(\mathbf{D} \otimes \mathbf{E} \otimes \mathbf{F}) = \mathbf{AD} \otimes \mathbf{BE} \otimes \mathbf{CF}.$$

Clearly, we can apply this to a deterministic space-time-doppler response vector. For example, the per-patch clutter response under the simplification is $\mathbf{v}(f_{\mathsf{c},q}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_q, \phi_q)$. Since we can regard the scalar 1 as a conformable matrix, we have

$$\mathbf{v}(f_{\mathsf{c},q}) \otimes \mathbf{s} \otimes \mathbf{a}(\theta_q, \phi_q) = (\mathbf{v}(f_{\mathsf{c},q}) \otimes \mathbf{I}_N \otimes \mathbf{a}(\theta_q, \phi_q))\mathbf{s}$$

$$= \mathbf{\Gamma}_q \mathbf{s}$$

where $\mathbf{\Gamma}_q$ is obviously defined. By the same token, the target response is equivalent to $\mathbf{Ts}$ where $\mathbf{T} = \mathbf{v}_t(f_d) \otimes \mathbf{I}_N \otimes \mathbf{a}_t(\theta_t, \phi_t)$. With this form, the overall received vector in the range gate of interest is

$$\tilde{\mathbf{y}} = \rho_t \mathbf{Ts} + \sum_{q=1}^{Q} \gamma_q \mathbf{\Gamma}_q \mathbf{s} + \mathbf{y}_{\mathsf{n}} + \mathbf{y}_{\mathsf{i}}.$$

This is something considerably easier on the eyes and more comprehensibly relates to the transfer function approach. We will continue to use this notation in subsequent analysis, as it reveals the structure of the design problem in a much more direct fashion.

III. JOINT WAVEFORM-FILTER DESIGN

With the signal model in place, we now turn to the initial purpose of [1]. Our goal is to find a STAP beamformer vector $\mathbf{w} \in \mathbb{C}^{NML}$ and a transmit signal $\mathbf{s} \in \mathbb{C}^N$ that minimizes the combined effect of the noise and interference represented by the covariance matrix $\mathbf{R_{ni}} \in \mathbb{C}^{NML \times NML}$) and the signal-dependent clutter. However, we also want to ensure reasonable radar operation. We describe this process below.

At the range gate we interrogate (which we assume contains the target), the return $\tilde{\mathbf{y}}$ is processed by a filter characterized by the weight vector $\mathbf{w}$, forming the output return $\mathbf{w}^H \tilde{\mathbf{y}}$. As mentioned above, we want to design this vector and the signal to minimize the expected undesired power $\mathbf{E}\{|\mathbf{w}^H \mathbf{y_u}|^2\} = \mathbf{w}^H \mathbf{R_u}(\mathbf{s})\mathbf{w}$. Additionally, we would like to constrain this minimization to ensure reasonable radar operation. First, for a given target space-time-doppler bin, we want a particular filter output, say, $\kappa \in \mathbb{C}$. This filter output can be represented by the Capon

beamformer equation $\mathbf{w}^H\mathbf{T}\mathbf{s}$, where $\mathbf{T} \in \mathbb{C}^{NML \times N}$ is the target channel response above. Second, we place an upper bound on the total signal power, say, P_o. Mathematically, we can represent this optimization problem as

$$\min_{\mathbf{w},\mathbf{s}} \quad \mathbf{w}^H\mathbf{R}_{\mathrm{u}}(\mathbf{s})\mathbf{w}$$
$$\text{s.t.} \quad \mathbf{w}^H\mathbf{T}\mathbf{s} = \kappa \cdot \tag{12}$$
$$\mathbf{s}^H\mathbf{s} \leq P_o$$

In [1], this problem was computationally shown to be non-convex, and therefore used a variety of alternating minimization algorithms to solve it. Our purpose will first be to analytically prove that the problem as formulated is non-convex from an engineering standpoint. However,determining convexity, or lack thereof, will be difficult in the current configuration, so we will make a notational change. Let $\mathbf{b}$ be the *combined beamformer-signal* vector defined by $\mathbf{b} = [\mathbf{w}^T \ \mathbf{s}^T]^T$.

In order to recover the individual elements $\mathbf{w}, \mathbf{s}$ from $\mathbf{b}$, we define two matrices:

$$\mathbf{w} = \boldsymbol{\Psi}_W\mathbf{b}, \ \boldsymbol{\Psi}_W = \begin{bmatrix} \mathbf{I}_{NML} & \mathbf{0}_{NML \times N} \end{bmatrix}$$
$$\mathbf{s} = \boldsymbol{\Psi}_S\mathbf{b}, \ \boldsymbol{\Psi}_S = \begin{bmatrix} \mathbf{0}_{N \times NML} & \mathbf{I}_N \end{bmatrix}. \tag{13}$$

For further notational simplicity, let us also define the complete objective function f_o as a sum of the noise-interference cost f_{NI} & the total clutter cost f_C (which is itself a sum of the Q per-patch clutter costs):

$$f_o = f_{NI} + f_C = \mathbf{w}^H\mathbf{R}_{\mathrm{u}}(\mathbf{s})\mathbf{w}$$
$$f_{NI} = \mathbf{w}^H(\mathbf{R_n} + \mathbf{R_i})\mathbf{w} = \mathbf{w}^H\mathbf{R_{ni}}\mathbf{w}$$
$$= \mathbf{b}^H\boldsymbol{\Psi}_W^T\mathbf{R_{ni}}\boldsymbol{\Psi}_W\mathbf{b} = \mathbf{b}^H\widetilde{\mathbf{R}_{\mathbf{ni}}}\mathbf{b}$$
$$f_C = \sum_{q=1}^{Q}\mathbf{w}^H\mathbf{R}_{\gamma}^q(\mathbf{s})\mathbf{w} = \sum_{q=1}^{Q}f_q$$

A. Massaging the clutter objective

We start with the per-patch clutter objective function. Using the channel representation above, $\mathbf{R}_{\gamma}^q$ can be distilled to $\mathbf{R}_{\gamma}^q = \overline{R}_{\gamma}^q\boldsymbol{\Gamma}_q\mathbf{s}\mathbf{s}^H\boldsymbol{\Gamma}_q^H$. Therefore, the cost functional for the q-th clutter patch is

$$f_q(\mathbf{w},\mathbf{s}) = \overline{R}_{\gamma}^{pq}\mathbf{w}^H\boldsymbol{\Gamma}_q\mathbf{s}\mathbf{s}^H\boldsymbol{\Gamma}_q^H\mathbf{w} \tag{14a}$$
$$= \overline{R}_{\gamma}^q|\mathbf{w}^H\boldsymbol{\Gamma}_q\mathbf{s}|^2 \tag{14b}$$

Substituting $\mathbf{w} = \boldsymbol{\Psi}_W\mathbf{b}$ and $\mathbf{s} = \boldsymbol{\Psi}_S\mathbf{b}$ into (14b) above, the final form of this cost function in the joint vector is

$$f_q(\mathbf{w},\mathbf{s}) = \overline{R}_{\gamma}^q|\mathbf{b}^H\boldsymbol{\Psi}_W^T\boldsymbol{\Gamma}_q\boldsymbol{\Psi}_S\mathbf{b}|^2 = \overline{R}_{\gamma}^q|\mathbf{b}^H\overline{\boldsymbol{\Gamma}}_q\mathbf{b}|^2 \tag{15}$$

where $\overline{\boldsymbol{\Gamma}}_q = \boldsymbol{\Psi}_W^T\boldsymbol{\Gamma}_q\boldsymbol{\Psi}_S$.

B. An alternative form of the per-patch clutter cost

Although the form derived above is quite compact & informative, we develop another equivalent formation of the per-patch clutter cost that will allow us to quickly identify the Hessian of each of the cost functions considered and, ultimately, determine the convexity of the problem.

An alternative form of the per-patch clutter cost function comes from Equation (14b). First, for any complex number z, its squared magnitude is $|z|^2 = \text{Re}(z)^2 + \text{Im}(z)^2$. Applying this to Equation (14b), we obtain:

$$\mathsf{f}_q(\mathbf{w}, \mathbf{s}) = \overline{R}_\gamma^q |\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s}|^2 \tag{16a}$$

$$= \overline{R}_\gamma^q \text{Re}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s})^2 + \overline{R}_\gamma^q \text{Im}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s})^2 \tag{16b}$$

Using elementary algebra, the components of Equation (16b) are given by:

$$\text{Re}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s}) = \frac{1}{2}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s} + \mathbf{s}^H \mathbf{\Gamma}_q^H \mathbf{w})$$

$$= \frac{1}{2}\mathbf{b}^H \begin{bmatrix} \mathbf{0}_{NML \times NML} & \mathbf{\Gamma}_q \\ \mathbf{\Gamma}_q^H & \mathbf{0}_{N \times N} \end{bmatrix} \mathbf{b}$$

$$\text{Im}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s}) = \frac{1}{2j}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s} - \mathbf{s}^H \mathbf{\Gamma}_q^H \mathbf{w})$$

$$= \frac{1}{2}\mathbf{b}^H \begin{bmatrix} \mathbf{0}_{NML \times NML} & \frac{1}{j}\mathbf{\Gamma}_q \\ -\frac{1}{j}\mathbf{\Gamma}_q^H & \mathbf{0}_{N \times N} \end{bmatrix} \mathbf{b}$$

Before we continue, we require a particular property of complex matrices. Any complex matrix $\mathbf{M}$ can be decomposed into the matrix sum $\mathbf{M} = \mathbf{M}_H + \mathbf{M}_{AH}$ where $\mathbf{M}_H = \frac{1}{2}(\mathbf{M} + \mathbf{M}^H), \mathbf{M}_{AH} = \frac{1}{2}(\mathbf{M} - \mathbf{M}^H)$ are the *H*ermitian and *A*nti-Hermitian parts of $\mathbf{M}$. We can see by inspection that the block matrices in the real & imaginary parts listed above contain the Hermitian and anti-Hermitian parts of $\overline{\mathbf{\Gamma}}_q$. Therefore, these forms are

$$\text{Re}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s}) = \frac{1}{2}\mathbf{b}^H \begin{bmatrix} \mathbf{0}_{NML \times NML} & \mathbf{\Gamma}_q \\ \mathbf{\Gamma}_q^H & \mathbf{0}_{N \times N} \end{bmatrix} \mathbf{b} = \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,H} \mathbf{b}$$

$$\text{Im}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s}) = \frac{1}{2}\mathbf{b}^H \begin{bmatrix} \mathbf{0}_{NML \times NML} & \frac{1}{j}\mathbf{\Gamma}_q \\ -\frac{1}{j}\mathbf{\Gamma}_q^H & \mathbf{0}_{N \times N} \end{bmatrix} \mathbf{b} = \frac{1}{j}\mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,A} \mathbf{b}$$

Substituting these into the forms above and simplifying results in

$$\mathsf{f}_q(\mathbf{w}, \mathbf{s}) = \overline{R}_\gamma^q \text{Re}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s})^2 + \overline{R}_\gamma^q \text{Im}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s})^2$$

$$= \overline{R}_\gamma^q \left((\mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,H} \mathbf{b})^2 - (\mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,A} \mathbf{b})^2 \right)$$

$$= \overline{R}_\gamma^q \begin{bmatrix} \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,H} \mathbf{b} \\ \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,A} \mathbf{b} \end{bmatrix}^H \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,H} \mathbf{b} \\ \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,A} \mathbf{b} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,H} \mathbf{b} \\ \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,A} \mathbf{b} \end{bmatrix}^H \begin{bmatrix} \overline{R}_\gamma^q & 0 \\ 0 & \overline{R}_\gamma^q \end{bmatrix} \begin{bmatrix} \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,H} \mathbf{b} \\ \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,A} \mathbf{b} \end{bmatrix}$$

$$= \mathbf{g}_q^H(\mathbf{b}) \mathbf{M}_q \mathbf{g}_q(\mathbf{b}) \tag{17}$$

This is an even more compact form of f_q that, as we will see in the next section, permits us to quickly and easily derive the Hessian of the per-patch clutter cost, the *total* clutter cost, and the overall objective function.

C. Summary of Forms

Finally, we arrive at multiple equivalent final forms of the clutter function f_C, which is a sum of the per-patch clutter forms:

$$\mathsf{f}_C = \sum_{q=1}^{Q} \overline{R}_\gamma^q |\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s}|^2 = \sum_{q=1}^{Q} \overline{R}_\gamma^q |\mathbf{b}^H \overline{\mathbf{\Gamma}}_q \mathbf{b}|^2$$

$$= \sum_{q=1}^{Q} \overline{R}_\gamma^q (\mathrm{Re}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s})^2 + \mathrm{Im}(\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s})^2) = \sum_{q=1}^{Q} \mathbf{g}_q^H(\mathbf{b}) \mathbf{M}_q \mathbf{g}_q(\mathbf{b})$$

In the joint variable $\mathbf{b}$, the overall optimization problem becomes

$$
\begin{aligned}
\min_{\mathbf{b}} \quad & \mathbf{b}^H \widetilde{\mathbf{R}}_{\mathbf{ni}} \mathbf{b} + \sum_{q=1}^{Q} \mathbf{g}_q^H(\mathbf{b}) \mathbf{M}_q \mathbf{g}_q(\mathbf{b}) \\
\text{s.t.} \quad & \mathbf{b}^H \mathbf{\Psi}_W^T \mathbf{T} \mathbf{\Psi}_S \mathbf{b} = \kappa \\
& \mathbf{b}^H \mathbf{\Psi}_S^T \mathbf{\Psi}_S \mathbf{b} \le P_o
\end{aligned}
\tag{18}
$$

Before we prove *joint* non-convexity, the form in Equation (14a) provides an immediate proof of convexity of the per-patch cost in $\mathbf{w}$ for fixed $\mathbf{s}$ and vice versa. Since the cost function f is real and the matrix $\mathbf{R}_\gamma^q$ is positive-semidefinite (by virtue of being a correlation/covariance matrix) for fixed $\mathbf{s}$, then for any non-zero $\mathbf{w}$, $f \ge 0$ and is thus convex in $\mathbf{w}$.

D. Proving Joint (Non-)Convexity

1) Methods of verifying convexity: The traditional definition of convexity is technically only defined for real-valued functions of *real* arguments (be they scalar, vector, or matrix). However, newly developed theory in [17] and the associated journal literature permits us to make similar claims for real-valued functions of *complex* arguments.

Let us define a scalar function f of the complex variable $\mathbf{b}$ as

$$
\begin{aligned}
f \colon \mathbb{C}^{J \times 1} &\longrightarrow \mathbb{R} \\
\mathbf{b} &\longmapsto \mathsf{f}(\mathbf{b}, \mathbf{b}^*)
\end{aligned}
$$

Under the traditional definition (see, for example, in [12]), this function is convex if for any $\mathbf{b}_1, \mathbf{b}_2 \in \Omega$ (where Ω is some convex subset of $\mathbb{C}^{J \times 1}$) and $\alpha \in [0, 1] \subset \mathbb{R}$,

$$f(\alpha \mathbf{b}_1 + (1 - \alpha) \mathbf{b}_2) \le \alpha f(\mathbf{b}_1) + (1 - \alpha) f(\mathbf{b}_2). \tag{19}$$

This is true because the function of interest is *real-valued*, despite the fact that its arguments are complex-valued. Equivalently, the Hessian matrix $\widetilde{\mathcal{H}}_{\mathbf{v}}[f] = \nabla_{\mathbf{v}}^2 f(\mathbf{v})$ must be positive semidefinite when evaluated at any stationary point $\mathbf{v} = \mathbf{b}_0 \in \Omega$. We will use the Hessian method, because it reveals the interesting structure of the problem in addition to showing non-convexity.

2) Determining complex Hessians: Since our function is a real-valued function of complex-valued variables, we require the following from Hjørungnes' work on complex matrix derivatives [17]. From [17, Theorem 3.2], the stationary points of a real-valued function of complex variables are the points where

$$\mathcal{D}_{\mathbf{b}} f(\mathbf{b}, \mathbf{b}^*) = \mathbf{0}_{1 \times N(ML+1)} \tag{20}$$

or,

$$\mathcal{D}_{\mathbf{b}^*} f(\mathbf{b}, \mathbf{b}^*) = \mathbf{0}_{1 \times N(ML+1)} \tag{21}$$

where, for example, the form $\mathcal{D}_{\mathbf{b}} f(\mathbf{b}, \mathbf{b}^*)$ is the complex derivative with respect to $\mathbf{b}$. As is the standard, we treat the vector $\mathbf{b}$ and its conjugate $\mathbf{b}^*$ as separate variables for the purposes of differentiation. This fact derives from the Wirtinger calculus and is directly proven in the reference above.

a) A Taylor series argument for Hessians of real-valued functions: It is well known that for a twice-differentiable real-valued scalar function, positive semi-definiteness of its Hessian matrix over a convex set is an equivalent statement of convexity. This is true even if the arguments are complex-valued. See, for example, [17, Lemma 5.2] and the consequences thereof. Namely, if the Hessian matrix in [17, Equation 5.44] is positive semidefinite at all stationary points $\mathbf{b}_o$ within the convex set of interest, then the zeroth-order convexity requirement in Equation 19 is automatically satisfied if we set $\mathbf{b}_1 = \mathbf{b}_o + \mathrm{d}\mathbf{b}, \mathbf{b}_2 = \mathbf{b}_o$. Let us rearrange Equation 19 to directly prove this assertion from the reference. First, observe that

$$f(\alpha \mathbf{b}_1 + (1-\alpha)\mathbf{b}_2) = f(\mathbf{b}_2 + \alpha(\mathbf{b}_1 - \mathbf{b}_2))$$

$$\alpha f(\mathbf{b}_1) + (1-\alpha)f(\mathbf{b}_2) = f(\mathbf{b}_2) + \alpha(f(\mathbf{b}_1) - f(\mathbf{b}_2))$$

Then, the convexity requirement becomes

$$f(\mathbf{b}_2 + \alpha(\mathbf{b}_1 - \mathbf{b}_2)) \leq f(\mathbf{b}_2) + \alpha(f(\mathbf{b}_1) - f(\mathbf{b}_2))$$

$$f(\mathbf{b}_2 + \alpha(\mathbf{b}_1 - \mathbf{b}_2)) - f(\mathbf{b}_2) \leq \alpha(f(\mathbf{b}_1) - f(\mathbf{b}_2))$$

$$\frac{f(\mathbf{b}_2 + \alpha(\mathbf{b}_1 - \mathbf{b}_2)) - f(\mathbf{b}_2)}{\alpha} \leq f(\mathbf{b}_1) - f(\mathbf{b}_2)$$

$$\frac{f(\mathbf{b}_2 + \alpha(\mathbf{b}_1 - \mathbf{b}_2)) - f(\mathbf{b}_2)}{\alpha} + f(\mathbf{b}_2) \leq f(\mathbf{b}_1)$$

Let us set $\mathbf{b}_1 = \mathbf{b}_o + \mathrm{d}\mathbf{b}, \mathbf{b}_2 = \mathbf{b}_o$, where $\mathbf{b}_o$ is a stationary point and $\mathrm{d}\mathbf{b}$ is a vector of infintesimally small perturbations. Then, the convexity requirement is

$$\frac{f(\mathbf{b}_o + \alpha \, \mathrm{d}\mathbf{b}) - f(\mathbf{b}_o)}{\alpha} + f(\mathbf{b}_o) \leq f(\mathbf{b}_o + \mathrm{d}\mathbf{b}).$$

At a stationary point, the first term on the left hand side is zero. Why? We can clearly see that, as $\alpha \to 0$, it is essentially a derivative, which must vanish at a stationary point by definition. Thus, the condition again changes to:

$$f(\mathbf{b}_o) \leq f(\mathbf{b}_o + \mathrm{d}\mathbf{b}) \tag{22}$$

Using the Taylor series expansion from [17, Lemma 5.2], we expand $f(\mathbf{b} + \mathrm{d}\mathbf{b})$,

$$f(\mathbf{b} + \mathrm{d}\mathbf{b}) = f(\mathbf{b}) + \begin{bmatrix} \mathcal{D}_{\mathbf{b}} f(\mathbf{b}, \mathbf{b}^*)^T \\ \mathcal{D}_{\mathbf{b}^*} f(\mathbf{b}, \mathbf{b}^*)^T \end{bmatrix}^T \begin{bmatrix} \mathrm{d}\mathbf{b} \\ \mathrm{d}\mathbf{b}^* \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathrm{d}\mathbf{b} \\ \mathrm{d}\mathbf{b}^* \end{bmatrix}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \begin{bmatrix} \mathrm{d}\mathbf{b} \\ \mathrm{d}\mathbf{b}^* \end{bmatrix} + HOT(\mathbf{b})$$

where the higher order terms function HOT converges to zero in the sense provided in the reference. Certainly then, regardless of the value of $\mathbf{b}$,

$$f(\mathbf{b} + \mathrm{d}\mathbf{b}) \geq f(\mathbf{b}) + \begin{bmatrix} \mathcal{D}_{\mathbf{b}} f(\mathbf{b}, \mathbf{b}^*)^T \\ \mathcal{D}_{\mathbf{b}^*} f(\mathbf{b}, \mathbf{b}^*)^T \end{bmatrix}^T \begin{bmatrix} \mathrm{d}\mathbf{b} \\ \mathrm{d}\mathbf{b}^* \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathrm{d}\mathbf{b} \\ \mathrm{d}\mathbf{b}^* \end{bmatrix}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \begin{bmatrix} \mathrm{d}\mathbf{b} \\ \mathrm{d}\mathbf{b}^* \end{bmatrix}$$

If we evaluate this expression at $\mathbf{b} = \mathbf{b}_o$, which is our aforementioned stationary point, the first order derivatives in the second term vanish by definition and therefore

$$f(\mathbf{b}_o + d\mathbf{b}) \geq f(\mathbf{b}_o) + \frac{1}{2} \begin{bmatrix} d\mathbf{b} \\ d\mathbf{b}^* \end{bmatrix}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \begin{bmatrix} d\mathbf{b} \\ d\mathbf{b}^* \end{bmatrix}.$$

In order for (22) to be satisfied, we then have

$$f(\mathbf{b}_o) + \frac{1}{2} \begin{bmatrix} d\mathbf{b} \\ d\mathbf{b}^* \end{bmatrix}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \begin{bmatrix} d\mathbf{b} \\ d\mathbf{b}^* \end{bmatrix} \geq f(\mathbf{b}_o)$$

$$\frac{1}{2} \begin{bmatrix} d\mathbf{b} \\ d\mathbf{b}^* \end{bmatrix}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \begin{bmatrix} d\mathbf{b} \\ d\mathbf{b}^* \end{bmatrix} \geq 0,$$

which is clearly the definition of positive semidefiniteness for a Hermitian complex matrix. Therefore, in order for the function f to be convex, the matrix $\begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix}$ *must* be positive semidefinite at all possible $\mathbf{b}_o$ inside the convex domain of f.

Remark 1. *Traditionally, convexity is only defined for real functions of* real *variables, which may make the analysis above seem incomplete. However, as we will show below, the definition of convexity for a complex vector variable is equivalent to the definition of joint convexity for its real and imaginary parts. To show this, we will prove that the definiteness condition remains if one replaces the differentials with respect to the complex variables with those for the real variables.*

First, let us define the real and imaginary parts of the complex vector as $\mathbf{x} = \mathrm{Re}(\mathbf{b}), \mathbf{y} = \mathrm{Im}(\mathbf{b})$ (which we assume are linearly independent real vectors and will treat as such). Clearly, then, we have $\mathbf{b} = \mathbf{x} + j\mathbf{y}$ and $\mathbf{b}^* = \mathbf{x} - j\mathbf{y}$. Applying the differential rules of [17] to these expressions, we obtain:

$$d\mathbf{b} = d\mathbf{x} + j\,d\mathbf{y} = \begin{bmatrix} \mathbf{I}_J & j\mathbf{I}_J \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix}$$

$$d\mathbf{b}^* = d\mathbf{x} - j\,d\mathbf{y} = \begin{bmatrix} \mathbf{I}_J & -j\mathbf{I}_J \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix}$$

which clearly means that we can replace the vector of complex differentials we see throughout the above derivation with

$$\begin{bmatrix} d\mathbf{b} \\ d\mathbf{b}^* \end{bmatrix} = \begin{bmatrix} \mathbf{I}_J & j\mathbf{I}_J \\ \mathbf{I}_J & -j\mathbf{I}_J \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix} = \mathbf{T}_{RC} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix},$$

where $\mathbf{T}_{RC}$ is a transformation matrix that maps the real components to the complex vectors. Making this substitution in the Taylor series expansion from above results in

$$f(\mathbf{x} + d\mathbf{x}, \mathbf{y} + d\mathbf{y}) = f(\mathbf{x}, \mathbf{y}) + \begin{bmatrix} \mathcal{D}_{\mathbf{b}} f(\mathbf{b}, \mathbf{b}^*)^T \\ \mathcal{D}_{\mathbf{b}^*} f(\mathbf{b}, \mathbf{b}^*)^T \end{bmatrix}^T \mathbf{T}_{RC} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix}$$

$$+ \frac{1}{2} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix}^T \mathbf{T}_{RC}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \mathbf{T}_{RC} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix} + HOT(\mathbf{x}, \mathbf{y})$$

where $HOT(\mathbf{x}, \mathbf{y})$ must still be positive & converge to zero in the same sense as before. As an aside, we note that the first derivatives are clearly those w.r.t. the components $\mathbf{x}, \mathbf{y}$.

Continuing on, it is still true that the following inequality holds, even if we substitute for the real and imaginary components:

$$f(\mathbf{x} + d\mathbf{x}, \mathbf{y} + d\mathbf{y}) \geq f(\mathbf{x}, \mathbf{y}) + \begin{bmatrix} \mathcal{D}_{\mathbf{b}} f(\mathbf{b}, \mathbf{b}^*)^T \\ \mathcal{D}_{\mathbf{b}^*} f(\mathbf{b}, \mathbf{b}^*)^T \end{bmatrix}^T \mathbf{T}_{RC} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix}^T \mathbf{T}_{RC}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \mathbf{T}_{RC} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix}.$$

Evaluating the inequality at the new stationary points $\mathbf{x}_o, \mathbf{y}_o$ eliminates the first derivatives, yielding

$$f(\mathbf{x}_o + d\mathbf{x}, \mathbf{y}_o + d\mathbf{y}) \geq f(\mathbf{x}_o, \mathbf{y}_o) + \frac{1}{2} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix}^T \mathbf{T}_{RC}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \mathbf{T}_{RC} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix}.$$

Again, (22) is satisfied only if the right-hand side of the above expression is greater than the value of the function at the stationary point, ergo

$$f(\mathbf{x}_o, \mathbf{y}_o) + \frac{1}{2} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix}^T \mathbf{T}_{RC}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \mathbf{T}_{RC} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix} \quad \geq \quad f(\mathbf{x}_o, \mathbf{y}_o)$$

$$\frac{1}{2} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix}^T \mathbf{T}_{RC}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \mathbf{T}_{RC} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{y} \end{bmatrix} \quad \geq \quad 0.$$

Now, the convexity requirement is that $\mathbf{T}_{RC}^H \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*} \\ \mathcal{H}_{\mathbf{bb}} & \mathcal{H}_{\mathbf{b}^*\mathbf{b}} \end{bmatrix} \mathbf{T}_{RC}$ *must be positive semidefinite at all possible* $\mathbf{x}_o, \mathbf{y}_o$ *inside the convex domain of f. However, this is* exactly identical *to the complex condition above, since according to [18, Lemma 7.63(b)], these are biconditional statements! For real-valued functions, therefore, convexity in the complex variable is identical to joint convexity in its real and imaginary parts.*

For notational simplicity, let us define this matrix – which we will call the *Hessian* of the function f with respect to the Taylor series argument above – as:

$$\widetilde{\mathcal{H}}[f] = \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*}(f) & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*}(f) \\ \mathcal{H}_{\mathbf{bb}}(f) & \mathcal{H}_{\mathbf{b}^*\mathbf{b}}(f) \end{bmatrix}.$$

b) The augmented Hessian: While the Hessian $\widetilde{\mathcal{H}}[f]$ is useful for determining convexity, it can be quite difficult to find expressions in a simple and systematic way for a given function f. If we use a change of variables, however, we can quickly find a *related* Hessian using a chain rule. As in [17], we first define the augmented variable $\mathcal{Z} = \begin{bmatrix} \mathbf{b} & \mathbf{b}^* \end{bmatrix}$. Derivatives taken with respect to $\mathcal{Z}$ are often just stackings of the derivatives with respect to $\mathbf{b}$ and $\mathbf{b}^*$. The matrix of second derivatives with respect to the augmented variable $\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(f)$, henceforth called the *augmented Hessian* of the function f, is given by:

$$\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(f) = \begin{bmatrix} \mathcal{H}_{\mathbf{bb}}(f) & \mathcal{H}_{\mathbf{b}^*\mathbf{b}}(f) \\ \mathcal{H}_{\mathbf{bb}^*}(f) & \mathcal{H}_{\mathbf{b}^*\mathbf{b}^*}(f) \end{bmatrix}.$$

c) The Chain Rule for augmented Hessians: A benefit of the augmented Hessian is that a *true* chain rule in complex arguments exists for them, as opposed to the desired Hessian form. While there is a more general form of the chain rule listed in the reference [17], we present below a stripped down version more appropriate for our purposes.

If $f = h \circ \mathbf{g}$ is a scalar function of a $J \times 1$ vector $\mathbf{b}$ (a composition of the scalar function h and the vector function $\mathbf{g}$), then the augmented Hessian of f is given by

$$\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(f) = (\mathcal{D}_{\mathbf{g}}(h) \otimes \mathbf{I}_{2T})\mathcal{H}_{\mathcal{Z}\mathcal{Z}}h + (\mathcal{D}_{\mathcal{Z}}\mathbf{g})^T \mathcal{H}_{\mathbf{gg}}(h)(\mathcal{D}_{\mathcal{Z}}\mathbf{g}).$$

As usual, the operator $\mathcal{D}_{\mathbf{Z}}(\mathbf{F}) = \dfrac{\partial \, \mathrm{vec}(\mathbf{F})}{\partial \, \mathrm{vec}^T(\mathbf{Z})}$ can be considered the derivative of the matrix function $\mathbf{F}$ with respect to the complex matrix variable $\mathbf{Z}$, and similarly for vector & scalar functions.

d) Mapping between augmented Hessian and the actual Hessian: While the augmented Hessian is excellent for calculation purposes, it is the actual Hessian that is needed for determining convexity. Thankfully, there is a straightforward relationship between the two:

$$\widetilde{\mathcal{H}}[f] = \begin{bmatrix} \mathbf{0}_{J \times J} & \mathbf{I}_J \\ \mathbf{I}_J & \mathbf{0}_{J \times J} \end{bmatrix} \mathcal{H}_{\mathcal{Z}\mathcal{Z}}(f) \tag{23}$$

and vice versa. Functionally, this means that we exchange the (1,1) block with the (2,1) block and the (1,2) block with the (2,2) block.

E. STAP Step 0: Preliminaries

1) Procedural outline: Given the various functional forms and preliminary proofs, we will now outline the rest of the procedure to determine convexity of the objective. First, we will determine the derivatives and Hessians of each component of the per-patch clutter cost f_q and use the chain rule to establish its Hessian. Using elementary techniques, we will demonstrate that this matrix is always indefinite unless a certain condition is satisfied, in which case it is always zero. Next, we will use the additivity of the combined clutter cost function to compute its Hessian and prove a similar definiteness result. Finally, we will compute the Hessian of the overall objective and show its indefiniteness by construction.

2) Functions of interest: Recall that in (17), we defined the per-patch clutter function is $\mathsf{f}_q = \mathbf{g}_q^H(\mathbf{b})\mathbf{M}_q\mathbf{g}_q(\mathbf{b})$, where

$$\mathbf{M}_q = \overline{R}_\gamma^q \mathbf{I}_2 \qquad\qquad \mathbf{g}_q(\mathbf{b}) = \begin{bmatrix} \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,H} \mathbf{b} \\ \mathbf{b}^H \overline{\mathbf{\Gamma}}_{q,A} \mathbf{b} \end{bmatrix}.$$

This form of the per-patch clutter function is quite compact and simplifies finding its augmented and actual Hessians, and therefore the augmented and actual Hessians of the total clutter and overall objective functions. Additionally, from this point forward, let $J = N(ML + 1)$ denote the joint number of parameters (and thus the length of the combined beamformer $\mathbf{b}$).

F. STAP Step 1: The differentials of $\mathbf{g}_q$

1) First derivative: Using the rules of complex matrix derivatives/differentials outlined above and in Hjørungnes [17], we can find the first derivatives of the inner vector function to be

$$\mathrm{d}\mathbf{g}_q = \begin{bmatrix} \mathrm{d}(\mathbf{b}^H\overline{\mathbf{\Gamma}}_{q,H}\mathbf{b}) \\ \mathrm{d}(\mathbf{b}^H\overline{\mathbf{\Gamma}}_{q,A}\mathbf{b}) \end{bmatrix} = \begin{bmatrix} \mathbf{b}^H\overline{\mathbf{\Gamma}}_{q,H}\,\mathrm{d}\mathbf{b} + \mathbf{b}^T\overline{\mathbf{\Gamma}}_{q,H}^T\,\mathrm{d}\mathbf{b}^* \\ \mathbf{b}^H\overline{\mathbf{\Gamma}}_{q,A}\,\mathrm{d}\mathbf{b} + \mathbf{b}^T\overline{\mathbf{\Gamma}}_{q,A}^T\,\mathrm{d}\mathbf{b}^* \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{b}^H\overline{\mathbf{\Gamma}}_{q,H} & \mathbf{b}^T\overline{\mathbf{\Gamma}}_{q,H}^T \\ \mathbf{b}^H\overline{\mathbf{\Gamma}}_{q,A} & \mathbf{b}^T\overline{\mathbf{\Gamma}}_{q,A}^T \end{bmatrix} \begin{bmatrix} \mathrm{d}\mathbf{b} \\ \mathrm{d}\mathbf{b}^* \end{bmatrix} = \mathcal{D}_{\mathcal{Z}}\mathbf{g}\,\mathrm{d}\,\mathrm{vec}(\mathcal{Z}).$$

2) Hessian: Let us first observe that $\mathbf{g}_q$ is a stacking of scalar functions of $\mathbf{b}$:

$$\mathbf{g}_q(\mathbf{b}) = \begin{bmatrix} \mathbf{b}^H\overline{\mathbf{\Gamma}}_{q,H}\mathbf{b} \\ \mathbf{b}^H\overline{\mathbf{\Gamma}}_{q,A}\mathbf{b} \end{bmatrix} = \begin{bmatrix} g_{q,0} \\ g_{q,1} \end{bmatrix}$$

Using [17, Eq. 5.69, 5.70] , we see that, as a result of the above form, the augmented Hessian of $\mathbf{g}_q$ is just a stacking of the per-element augmented Hessians:

$$\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(\mathbf{g}_q) = \begin{bmatrix} \mathcal{H}_{\mathcal{Z}\mathcal{Z}}(g_{q,0}) \\ \mathcal{H}_{\mathcal{Z}\mathcal{Z}}(g_{q,1}) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \mathcal{H}_{\mathcal{Z}\mathcal{Z}}(g_{q,0}) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \mathcal{H}_{\mathcal{Z}\mathcal{Z}}(g_{q,1})$$

In order to find the per-element augmented Hessians, we use the following fact. From [17, Example 5.2], for any compatible matrix $\mathbf{\Phi}$,

$$\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(\mathbf{b}^H\mathbf{\Phi}\mathbf{b}) = \begin{bmatrix} \mathbf{0}_{J\times J} & \mathbf{\Phi}^T \\ \mathbf{\Phi} & \mathbf{0}_{J\times J} \end{bmatrix}.$$

Continuing with this in mind, we find the per-element augmented Hessians to be

$$\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(g_{q,0}) = \begin{bmatrix} \mathbf{0}_{J\times J} & \overline{\mathbf{\Gamma}}_{q,H}^T \\ \overline{\mathbf{\Gamma}}_{q,H} & \mathbf{0}_{J\times J} \end{bmatrix} \quad \mathcal{H}_{\mathcal{Z}\mathcal{Z}}(g_{q,1}) = \begin{bmatrix} \mathbf{0}_{J\times J} & \overline{\mathbf{\Gamma}}_{q,A}^T \\ \overline{\mathbf{\Gamma}}_{q,A} & \mathbf{0}_{J\times J} \end{bmatrix}$$

Therefore, the augmented Hessian of the vector function $\mathbf{g}_q$ is

$$\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(\mathbf{g}_q) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{0}_{J\times J} & \overline{\mathbf{\Gamma}}_{q,H}^T \\ \overline{\mathbf{\Gamma}}_{q,H} & \mathbf{0}_{J\times J} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{0}_{J\times J} & \overline{\mathbf{\Gamma}}_{q,A}^T \\ \overline{\mathbf{\Gamma}}_{q,A} & \mathbf{0}_{J\times J} \end{bmatrix}.$$

G. STAP Step 2: The differentials of h *w.r.t.* $\mathbf{g}_q$

1) First derivative: Recall $h = \mathbf{g}_q^H\mathbf{M}_q\mathbf{g}_q$. Thus,

$$\mathrm{d}h = \mathrm{d}\left(\mathbf{g}_q^H\mathbf{M}_q\mathbf{g}_q\right) = \mathbf{g}_q^H\mathbf{M}_q\,\mathrm{d}\mathbf{g}_q + \mathbf{g}_q^T\mathbf{M}_q^T\,\mathrm{d}\mathbf{g}_q^*$$

This means that

$$\mathcal{D}_{\mathbf{g}_q}(h) = \mathbf{g}_q^H\mathbf{M}_q$$

$$\mathcal{D}_{\mathbf{g}_q^*}(h) = \mathbf{g}_q^T\mathbf{M}_q^T$$

2) Hessian: From [17, Example 5.2] , we know that

$$\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(\mathbf{b}^H\mathbf{\Phi}\mathbf{b}) = \begin{bmatrix} \mathbf{0}_{J\times J} & \mathbf{\Phi}^T \\ \mathbf{\Phi} & \mathbf{0}_{J\times J} \end{bmatrix}$$

for any compatible matrix $\mathbf{\Phi}$. That means that, by inspection, $\mathcal{H}_{\mathbf{g}\mathbf{g}}(h) = \mathbf{0}_{J\times J}$!

H. STAP Step 3: The per-patch clutter Hessian

Applying the chain-rule for the augmented Hessian and the derivatives found above, the augmented Hessian of the per-patch clutter cost function is

$$
\begin{aligned}
\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(\mathfrak{f}_q) &= (\mathcal{D}_{\mathbf{g}_q}(h) \otimes \mathbf{I}_{2P})\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(\mathbf{g}_q) + (\mathcal{D}_{\mathcal{Z}}(\mathbf{g}_q))^T \mathcal{H}_{\mathbf{g}_q\mathbf{g}_q}(h)(\mathcal{D}_{\mathcal{Z}}(\mathbf{g}_q)) \\
&= (\mathcal{D}_{\mathbf{g}_q}(h) \otimes \mathbf{I}_{2P})\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(\mathbf{g}_q) \\
&= (\mathbf{g}_q^H \mathbf{M}_q \otimes \mathbf{I}_{2P}) \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{0}_{J\times J} & \overline{\boldsymbol{\Gamma}}_{q,H}^T \\ \overline{\boldsymbol{\Gamma}}_{q,H} & \mathbf{0}_{J\times J} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{0}_{J\times J} & \overline{\boldsymbol{\Gamma}}_{q,A}^T \\ \overline{\boldsymbol{\Gamma}}_{q,A} & \mathbf{0}_{J\times J} \end{bmatrix} \right) \\
&= \mathbf{g}_q^H \mathbf{M}_q \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} \mathbf{0}_{J\times J} & \overline{\boldsymbol{\Gamma}}_{q,H}^T \\ \overline{\boldsymbol{\Gamma}}_{q,H} & \mathbf{0}_{J\times J} \end{bmatrix} + \mathbf{g}_q^H \mathbf{M}_q \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{0}_{J\times J} & \overline{\boldsymbol{\Gamma}}_{q,A}^T \\ \overline{\boldsymbol{\Gamma}}_{q,A} & \mathbf{0}_{J\times J} \end{bmatrix}
\end{aligned}
$$

In order to reduce this to a simpler form, we need to know what the scalars in each term are:

$$
\begin{aligned}
\mathbf{g}_q^H \mathbf{M}_q \begin{bmatrix} 1 \\ 0 \end{bmatrix} &= \overline{R}_\gamma^q \begin{bmatrix} \mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,H} \mathbf{b} \\ \mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,A} \mathbf{b} \end{bmatrix}^H \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,H} \mathbf{b} \\
\mathbf{g}_q^H \mathbf{M}_q \begin{bmatrix} 0 \\ 1 \end{bmatrix} &= \overline{R}_\gamma^q \begin{bmatrix} \mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,H} \mathbf{b} \\ \mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,A} \mathbf{b} \end{bmatrix}^H \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -\mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,A} \mathbf{b}
\end{aligned}
$$

Thus, the per-patch augmented Hessian reduces to

$$
\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(\mathfrak{f}_q) = \overline{R}_\gamma^q \left(\mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,H} \mathbf{b} \begin{bmatrix} \mathbf{0}_{J\times J} & \overline{\boldsymbol{\Gamma}}_{q,H}^T \\ \overline{\boldsymbol{\Gamma}}_{q,H} & \mathbf{0}_{J\times J} \end{bmatrix} - \mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,A} \mathbf{b} \begin{bmatrix} \mathbf{0}_{J\times J} & \overline{\boldsymbol{\Gamma}}_{q,A}^T \\ \overline{\boldsymbol{\Gamma}}_{q,A} & \mathbf{0}_{J\times J} \end{bmatrix} \right)
$$

Based on the form, we observe that the only non-zero unique block of this matrix is the (2,1) block (since the (1,2) block is its transpose). What is the (2,1) block? First, recall that

$$
\begin{aligned}
\mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,H} \mathbf{b} &= \mathrm{Re}(\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}) \quad \mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,H} \mathbf{b} = \jmath \mathrm{Im}(\mathbf{w}^H \boldsymbol{\Gamma} \mathbf{s}) \\
\overline{\boldsymbol{\Gamma}}_{q,H} &= \frac{1}{2}(\overline{\boldsymbol{\Gamma}}_q + \overline{\boldsymbol{\Gamma}}_q^H) \quad \overline{\boldsymbol{\Gamma}}_{q,A} = \frac{1}{2}(\overline{\boldsymbol{\Gamma}}_q - \overline{\boldsymbol{\Gamma}}_q^H).
\end{aligned}
$$

Temporarily omitting the $\overline{R}_\gamma^q$ term, the (2,1) block is

$$
\begin{aligned}
(\mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,H} \mathbf{b})\overline{\boldsymbol{\Gamma}}_{q,H} - (\mathbf{b}^H \overline{\boldsymbol{\Gamma}}_{q,A} \mathbf{b})\overline{\boldsymbol{\Gamma}}_{q,A} &= \frac{1}{2}\left(\mathrm{Re}(\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})(\overline{\boldsymbol{\Gamma}}_q + \overline{\boldsymbol{\Gamma}}_q^H) - \jmath \mathrm{Im}(\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})(\overline{\boldsymbol{\Gamma}}_q - \overline{\boldsymbol{\Gamma}}_q^H) \right) \\
&= \frac{1}{2}\left(\left(\mathrm{Re}(\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}) - \jmath \mathrm{Im}(\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})\right) \overline{\boldsymbol{\Gamma}}_q \right. \\
&\quad \left. + \left(\mathrm{Re}(\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}) + \jmath \mathrm{Im}(\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})\right) \overline{\boldsymbol{\Gamma}}_q^H \right) \\
&= \frac{1}{2}\left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q + (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})\overline{\boldsymbol{\Gamma}}_q^H \right) \\
&= \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H
\end{aligned}
$$

Therefore, the final form of the augmented per-patch Hessian is

$$
\mathcal{H}_{\mathcal{Z}\mathcal{Z}}(\mathfrak{f}_q) = \overline{R}_\gamma^q \begin{bmatrix} \mathbf{0}_{J\times J} & \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H^T \\ \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H & \mathbf{0}_{J\times J} \end{bmatrix}
$$

Using the second fact in Section 1, the final form of the *true* Hessian is

$$\widetilde{\mathcal{H}}[\mathsf{f}_q] = \overline{R}_\gamma^q \begin{bmatrix} \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H^T \end{bmatrix} = \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_q) & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_q)^T \end{bmatrix}$$

where we have defined $\mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_q) = \overline{R}_\gamma^q \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H$.

I. STAP Step 4: Definiteness of $\widetilde{\mathcal{H}}[\mathsf{f}_q]$

In order for f_q to be convex in $\mathbf{b}$, then $\widetilde{\mathcal{H}}[\mathsf{f}_q]$ must be positive-semidefinite for all valid $\mathbf{b}_o$. Since this Hessian is block diagonal, it is PSD if and only if each of the diagonal blocks are PSD. Furthermore, since the (2,2) block is the transpose of the (1,1) block, the (2,2) block's definiteness is the same as the (1,1) block's definiteness. Therefore, $\widetilde{\mathcal{H}}[\mathsf{f}_q]$ is PSD iff $\mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_q) = \overline{R}_\gamma^q \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H$ is PSD. We will show definitively that $\mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_q)$ *cannot be* PSD solely by its form.

First, $\mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_q)$ is given by

$$\begin{aligned} \mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_q) &= \overline{R}_\gamma^q \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H \\ &= \frac{\overline{R}_\gamma^q}{2} \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q + (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}) \overline{\boldsymbol{\Gamma}}_q^H \right) \\ &= \begin{bmatrix} \mathbf{0}_{NML \times NML} & \frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q \\ \frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}) \boldsymbol{\Gamma}_q^H & \mathbf{0}_{N \times N} \end{bmatrix} \end{aligned}$$

It is well known (see [18, Lemma 6.32, p. 124]) that a complex matrix with this form is indefinite. Why? Due to the structure, its eigenvalues are:

- the N singular values of $\frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q$ (the upper right corner matrix), which are positive;
- the *negatives* of the singular values of $\frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q$, which are negative;
- and zeroes for the rest.

What are these singular values of $\frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q$? First,

$$\begin{aligned} \left(\frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q \right)^H \left(\frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q \right) &= \left(\frac{\overline{R}_\gamma^q}{2} \right)^2 (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})(\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q^H \boldsymbol{\Gamma}_q \\ &= \left(\frac{\overline{R}_\gamma^q}{2} \right)^2 |\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}|^2 \boldsymbol{\Gamma}_q^H \boldsymbol{\Gamma}_q \\ &= \left(\frac{\overline{R}_\gamma^q}{2} \right)^2 |\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}|^2 (\mathbf{c}_q^H \otimes \mathbf{I}_N) \mathbf{K}_{LN,M}^T \mathbf{K}_{LN,M} (\mathbf{c}_q \otimes \mathbf{I}_N) \\ &= \left(\frac{\overline{R}_\gamma^q}{2} \right)^2 |\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}|^2 \|\mathbf{c}_q\|^2 \mathbf{I}_N \end{aligned}$$

It is obvious that the eigenvalues of this matrix are $\left(\frac{\overline{R}_\gamma^q}{2} \right)^2 |\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}|^2 \|\mathbf{c}_q\|^2$ with multiplicity N. The singular values of $\frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q$ are then the positive square roots of this eigenvalue, or $\frac{\overline{R}_\gamma^q}{2} |\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}| \|\mathbf{c}_q\|$.

If a matrix has negative eigenvalues, it cannot be PSD; similarly, if it has positive eigenvalues, it cannot be negative semidefinite. The only way this matrix has no negative eigenvalues is if they are all *zero*. Since two of the terms

in the singular value expression are always positive, this eigenvalue is zero only if the optimal signal/beamformer pair nulls the qth clutter patch. Therefore, the per-patch clutter Hessian is indefinite and the per-patch clutter cost is nonconvex. This is true for any choice of signal or beamformer on any choice of set.

J. STAP Step 5: Definiteness of $\widetilde{\mathcal{H}}[f_C]$

Since the overall clutter cost is a sum of the per-patch costs, its Hessian is the sum of the per-patch clutter Hessians. In other words,

$$
\begin{aligned}
\widetilde{\mathcal{H}}[f_C] &= \sum_{q=1}^{Q} \widetilde{\mathcal{H}}[f_q] \\
&= \sum_{q=1}^{Q} \overline{R}_\gamma^q \begin{bmatrix} \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H^T \end{bmatrix} \\
&= \begin{bmatrix} \sum_{q=1}^{Q} \overline{R}_\gamma^q \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \sum_{q=1}^{Q} \overline{R}_\gamma^q \left((\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \overline{\boldsymbol{\Gamma}}_q \right)_H^T \end{bmatrix} \\
&= \begin{bmatrix} \sum_{q=1}^{Q} \mathcal{H}_{\mathbf{b}\mathbf{b}^*}(f_q) & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \sum_{q=1}^{Q} \mathcal{H}_{\mathbf{b}\mathbf{b}^*}(f_q)^T \end{bmatrix} = \begin{bmatrix} \mathcal{H}_{\mathbf{b}\mathbf{b}^*}(f_C) & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \mathcal{H}_{\mathbf{b}\mathbf{b}^*}(f_C)^T \end{bmatrix}
\end{aligned}
$$

Similar to the per-patch clutter Hessian, the definiteness of the overall clutter Hessian is determined by the definiteness of $\mathcal{H}_{\mathbf{b}\mathbf{b}^*}(f_C)$. Therefore, let us take a closer look at $\mathcal{H}_{\mathbf{b}\mathbf{b}^*}(f_C)$:

$$
\begin{aligned}
\mathcal{H}_{\mathbf{b}\mathbf{b}^*}(f_C) &= \sum_{q=1}^{Q} \begin{bmatrix} \mathbf{0}_{NML \times NML} & \frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q \\ \frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}) \boldsymbol{\Gamma}_q^H & \mathbf{0}_{N \times N} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{0}_{NML \times NML} & \sum_{q=1}^{Q} \frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q \\ \sum_{q=1}^{Q} \frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}) \boldsymbol{\Gamma}_q^H & \mathbf{0}_{N \times N} \end{bmatrix}.
\end{aligned}
$$

Define the $NML \times N$ complex matrix $\boldsymbol{\Pi}_Q = \sum_{q=1}^{Q} \frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s})^* \boldsymbol{\Gamma}_q$. Hence,

$$
\mathcal{H}_{\mathbf{b}\mathbf{b}^*}(f_C) = \begin{bmatrix} \mathbf{0}_{NML \times NML} & \boldsymbol{\Pi}_Q \\ \boldsymbol{\Pi}_Q^H & \mathbf{0}_{N \times N} \end{bmatrix}
$$

Once again, the previous structure occurs. By the lemma, the eigenvalues of $\mathcal{H}_{\mathbf{b}\mathbf{b}^*}(f_C)$ are the N singular values of $\boldsymbol{\Pi}_Q$ (i.e., the square roots of the eigenvalues of $\boldsymbol{\Pi}_Q^H \boldsymbol{\Pi}_Q$), the negatives of those singular values, and $N(ML - 1)$ zeros.

What are these singular values? First, let us examine $\boldsymbol{\Pi}_Q^H \boldsymbol{\Pi}_Q$:

$$
\begin{aligned}
\boldsymbol{\Pi}_Q^H \boldsymbol{\Pi}_Q &= \left(\sum_{q=1}^{Q} \frac{\overline{R}_\gamma^q}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}) \boldsymbol{\Gamma}_q^H \right) \left(\sum_{l=1}^{Q} \frac{\overline{R}_\gamma^l}{2} (\mathbf{w}^H \boldsymbol{\Gamma}_l \mathbf{s})^* \boldsymbol{\Gamma}_l \right) \\
&= \frac{1}{4} \sum_{q=1}^{Q} \sum_{l=1}^{Q} \overline{R}_\gamma^q \overline{R}_\gamma^l (\mathbf{w}^H \boldsymbol{\Gamma}_q \mathbf{s}) (\mathbf{w}^H \boldsymbol{\Gamma}_l \mathbf{s})^* \boldsymbol{\Gamma}_q^H \boldsymbol{\Gamma}_l
\end{aligned}
$$

To simplify things slightly, we can find a different form of $\boldsymbol{\Gamma}_q^H \boldsymbol{\Gamma}_l$:

$$
\begin{aligned}
\boldsymbol{\Gamma}_q^H \boldsymbol{\Gamma}_l &= (\mathbf{c}_q^H \otimes \mathbf{I}_N) \mathbf{K}_{LN,M}^T \mathbf{K}_{LN,M} (\mathbf{c}_l \otimes \mathbf{I}_N) \\
&= \left(\mathbf{c}_q^H \mathbf{c}_l \right) \mathbf{I}_N
\end{aligned}
$$

Returning to the original matrix $\mathbf{\Pi}_Q^H \mathbf{\Pi}_Q$, we find

$$
\begin{aligned}
\mathbf{\Pi}_Q^H \mathbf{\Pi}_Q &= \frac{1}{4} \sum_{q=1}^{Q} \sum_{l=1}^{Q} \overline{R}_\gamma^q \overline{R}_\gamma^l (\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s})(\mathbf{w}^H \mathbf{\Gamma}_l \mathbf{s})^* \left(\mathbf{c}_q^H \mathbf{c}_l \right) \mathbf{I}_N \\
&= \left(\frac{1}{4} \sum_{q=1}^{Q} \sum_{l=1}^{Q} \overline{R}_\gamma^q \overline{R}_\gamma^l (\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s})(\mathbf{w}^H \mathbf{\Gamma}_l \mathbf{s})^* \left(\mathbf{c}_q^H \mathbf{c}_l \right) \right) \mathbf{I}_N
\end{aligned}
$$

If we define the matrix $\mathbf{D} = \left[(\mathbf{w}^H \mathbf{\Gamma}_1 \mathbf{s})^* \mathbf{c}_1 \cdots (\mathbf{w}^H \mathbf{\Gamma}_Q \mathbf{s})^* \mathbf{c}_Q \right]$, then the scalar inside the double sum above is the (q,l)th element of the matrix $\mathbf{D}^H \mathbf{D}$, which is a Gramian matrix and is thus positive semidefinite. The double sum, then, is the scalar $\mathbf{1}_Q^T \mathbf{D}^H \mathbf{D} \mathbf{1}_Q$, which is clearly positive. Therefore, a final form of $\mathbf{\Pi}_Q^H \mathbf{\Pi}_Q$ is

$$
\mathbf{\Pi}_Q^H \mathbf{\Pi}_Q = \frac{\mathbf{1}_Q^T \mathbf{D}^H \mathbf{D} \mathbf{1}_Q}{4} \mathbf{I}_N
$$

It is immediately obvious that the matrix has only one eigenvalue – $\frac{\mathbf{1}_Q^T \mathbf{D}^H \mathbf{D} \mathbf{1}_Q}{4}$ – with multiplicity N. This eigenvalue is either always positive or zero. From its form, we can surmise that it is zero only in the case when $\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s} = 0$ for all clutter patches q. Why is this so?

First, if $\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s} \neq 0$ for all q, then $\mathbf{1}_Q^T \mathbf{D}^H \mathbf{D} \mathbf{1}_Q = 0$ only if the inner products $\mathbf{c}_q^H \mathbf{c}_l = 0$ for all q, l. However, if this is true, then $\mathbf{c}_q^H \mathbf{c}_q = \|\mathbf{c}_q\|^2 = 0$ for all q; in other words, *there is no clutter*, which is a ridiculous requirement. What if we relax the inner product condition to $\mathbf{c}_q^H \mathbf{c}_l = 0$ for $q \neq l$? This doesn't change things much, because then $\mathbf{1}_Q^T \mathbf{D}^H \mathbf{D} \mathbf{1}_Q = \sum_{q=1}^{Q} \frac{(\overline{R}_\gamma^q)^2}{2} |\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s}|^2 \|\mathbf{c}_q\|^2$. This is absolutely positive unless $\mathbf{w}^H \mathbf{\Gamma}_q \mathbf{s} = 0$ for all q, which is the condition we were trying to avoid.

The singular values of $\mathbf{\Pi}_Q$ are therefore clearly only positive, implying $\mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_C)$ has positive, negative, and zero eigenvalues. Hence, the clutter Hessian is indefinite, and the clutter cost function is non-convex in $\mathbf{b}$.

K. STAP Step 6: Definiteness of $\widetilde{\mathcal{H}}[\mathsf{f}_o]$

At this point, one might think adding a full rank PSD matrix (in this case, the combined noise-and-interference correlation matrix) somewhere will break up the useful structure from the previous sections. However, this is not the case, and a similarly useful structure appears in the Hessian of the complete cost function. We begin by noting that, since f_o is a sum of f_{NI} and f_C, the Hessian is also a sum: $\widetilde{\mathcal{H}}[\mathsf{f}_o] = \widetilde{\mathcal{H}}[\mathsf{f}_{NI}] + \widetilde{\mathcal{H}}[\mathsf{f}_C]$. Since

$$
\mathsf{f}_{NI} = \mathbf{b}^H \begin{bmatrix} \mathbf{R_{ni}} & \mathbf{0}_{NML \times N} \\ \mathbf{0}_{N \times NML} & \mathbf{0}_{N \times N} \end{bmatrix} \mathbf{b} = \mathbf{b}^H \widetilde{\mathbf{R}}_{\mathbf{ni}} \mathbf{b} \text{ is a real quadratic form, we once again turn to Hjørungnes'}
$$

Example 5.2 as cited above, which means the Hessian $\widetilde{\mathcal{H}}[\mathsf{f}_{NI}]$ is

$$
\widetilde{\mathcal{H}}[\mathsf{f}_{NI}] = \begin{bmatrix} \widetilde{\mathbf{R}}_{\mathbf{ni}} & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \widetilde{\mathbf{R}}_{\mathbf{ni}}^T \end{bmatrix}
$$

or, more compactly, $\mathcal{H}_{\mathbf{bb}^*}(f_{NI}) = \widetilde{\mathbf{R}}_{\mathbf{ni}}$.

Hence, the total Hessian of the cost function $\widetilde{\mathcal{H}}[\mathsf{f}_o]$ is

$$
\begin{aligned}
\widetilde{\mathcal{H}}[\mathsf{f}_o] &= \begin{bmatrix} \widetilde{\mathbf{R}}_{\mathbf{ni}} & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \widetilde{\mathbf{R}}_{\mathbf{ni}}^T \end{bmatrix} + \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_C) & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_C)^T \end{bmatrix} \\
&= \begin{bmatrix} \mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_o) & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_o)^T \end{bmatrix}
\end{aligned}
$$

where $\mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_o) = \widetilde{\mathbf{R}}_{\mathbf{ni}} + \mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_C)$. Again, due to the structure, the definiteness of $\mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_o)$ determines the definiteness of the complete Hessian. In order to ascertain the definiteness of $\mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_o)$, we need to observe its structure:

$$
\begin{aligned}
\mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_o) &= \widetilde{\mathbf{R}}_{\mathbf{ni}} + \mathcal{H}_{\mathbf{bb}^*}(\mathsf{f}_C) \\
&= \begin{bmatrix} \mathbf{R}_{\mathbf{ni}} & \mathbf{\Pi}_Q \\ \mathbf{\Pi}_Q^H & \mathbf{0}_{N \times N} \end{bmatrix} .
\end{aligned}
$$

Before continuing, we introduce two essential, yet equivalent, theorems from Kreindler & Jameson [19] for the definiteness of a partitioned matrix.

Theorem 1 (via [19],Theorems $I_{a''}$ & $I_{b''}$). *A matrix* $\mathbf{M}$ *partitioned as* $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^H & \mathbf{C} \end{bmatrix}$ *is non-negative definite if and only if the following conditions are satisfied:*

A1 $\mathbf{C} \succeq 0$

A2 $\mathbf{B} = \mathbf{B}\mathbf{C}^\dagger\mathbf{C}$

A3 $\mathbf{A} - \mathbf{B}\mathbf{C}^\dagger\mathbf{B}^H \succeq 0$

or

B1 $\mathbf{A} \succeq 0$

B2 $\mathbf{B} = \mathbf{A}\mathbf{A}^\dagger\mathbf{B}$

B3 $\mathbf{C} - \mathbf{B}^H\mathbf{A}^\dagger\mathbf{B} \succeq 0$

where $\mathbf{A}^\dagger$ *indicates the Moore-Penrose pseudoinverse of the matrix* $\mathbf{A}$ *and* $\mathbf{A} \succeq 0$ *means that* $\mathbf{A}$ *is non-negative definite.*

In the case of the matrix $\mathcal{H}_{\mathbf{bb}^*}(f_o)$, $\mathbf{A} = \mathbf{R}_{\mathbf{ni}}, \mathbf{B} = \mathbf{\Pi}_Q$, and $\mathbf{C} = \mathbf{0}_{N \times N}$. The proof of definiteness follows by checking necessary conditions & observing how and if they are violated. We propose that this matrix is not PSD unless we know, a priori, that all stationary points null every clutter patch. We present two equivalent proofs of our proposition from the sets of A and B conditions in the above theorem.

Proof A. Any quadratic form involving the all-zero matrix $\mathbf{0}_{N \times N}$ is zero, implying the zero matrix is simultaneously positive semidefinite, negative semidefinite, and indefinite. Thus, Condition A1 is immediately satisfied.

Conditions A2 & A3 require the Moore-Penrose pseudoinverse of the all-zero matrix, which is itself: $\mathbf{0}_{N \times N}^\dagger = \mathbf{0}_{N \times N}$. Then, we can immediately restate condition A2 as $\mathbf{\Pi}_Q = \mathbf{0}_{NML \times N}$. As has been previously established, this only occurs if $\mathbf{w}_o^H \mathbf{\Gamma}_q \mathbf{s}_o = 0$ for all clutter patches q and all stationary points $\mathbf{w}_o, \mathbf{s}_o$ in the feasible set.

After simplification, Condition A3 is equivalent to $\mathbf{R}_{\mathbf{ni}} \succeq 0$. Since $\mathbf{R}_{\mathbf{ni}}$ is a covariance matrix and all covariance matrices are positive semidefinite by construction, Condition A3 is immediately satisfied. □

Proof B. Since $\mathbf{R}_{\mathbf{ni}}$ is a covariance matrix and all covariance matrices are positive semidefinite by construction, Condition B1 is immediately satisfied.

As a consequence of this result, we can use [18, Lemma 7.63(b)] to immediately state that because $\mathbf{R_{ni}}$ is positive semidefinite and Hermitian, its Moore-Penrose pseudoinverse $\mathbf{R_{ni}}^{\dagger}$ is also positive semidefinite.

Furthermore, using [18, Lemma 10.46(b)(i)], we can say that for *any* matrix $\mathbf{X} \in \mathbb{C}^{N \times NML}$, $\mathbf{X}^{H}\mathbf{R_{ni}}^{\dagger}\mathbf{X} \succeq 0$. Let us set $\mathbf{X} = \mathbf{\Pi}_{Q}$. Then, this lemma allows us to state $\mathbf{\Pi}_{Q}^{H}\mathbf{R_{ni}}^{\dagger}\mathbf{\Pi}_{Q} \succeq 0$, regardless of the choice of $\mathbf{b}$ that composes $\mathbf{\Pi}_{Q}$.

However, this statement is, on its face, an immediate contradiction to Condition B3, unless $\mathbf{\Pi}_{Q}^{H}\mathbf{R_{ni}}^{\dagger}\mathbf{\Pi}_{Q} = \mathbf{0}_{N \times N}$. By [18, Lemma 10.12(b)], this is only true if $\mathbf{\Pi}_{Q}^{H}\mathbf{R_{ni}}^{\dagger} = \mathbf{0}_{N \times NML}$ or, similarly, $\mathbf{R_{ni}}^{\dagger}\mathbf{\Pi}_{Q} = \mathbf{0}_{NML \times N}$

If, indeed, $\mathbf{R_{ni}}^{\dagger}\mathbf{\Pi}_{Q} = \mathbf{0}_{N \times NML}$, then that would mean Condition B2 would read

$$\mathbf{\Pi}_{Q} = \mathbf{R_{ni}}\mathbf{R_{ni}}^{\dagger}\mathbf{\Pi}_{Q}$$

$$= \mathbf{R_{ni}}\mathbf{0}_{NML \times N} = \mathbf{0}_{NML \times N}.$$

But, as we established before, this is only possible if, for each stationary point $\mathbf{w}_{o}, \mathbf{s}_{o}$ in the feasible set, $\mathbf{w}_{o}^{H}\mathbf{\Gamma}_{q}\mathbf{s}_{o} = 0$ for all clutter patches q. $\qquad\qquad\square$

A third proof relies on a similar construction of Theorem 1 from [20], which we paraphrase in Theorem 2 below:

Theorem 2 (via [20], Theorem 7.7.9(a, b)). *Let the matrix $\mathbf{M}$ be partitioned as in Theorem 1. The following statements are equivalent:*

1) $\mathbf{M}$ *is positive semidefinite*

2) $\mathbf{A}$ *and* $\mathbf{C}$ *are positive semidefinite and there is a contraction* $\mathbf{X}$ *shaped identically to* $\mathbf{B}$ *such that* $\mathbf{B} = \mathbf{A}^{1/2}\mathbf{X}\mathbf{C}^{1/2}$.

In the above theorem, a contraction is defined any matrix whose largest singular value is less than or equal to one, and the superscript $1/2$ denotes the unique matrix square root defined for all PSD matrices. Observe also that this theorem implies that $\mathcal{R}(\mathbf{B}) \subseteq \mathcal{R}(\mathbf{A}), \mathcal{R}(\mathbf{B}^{H}) \subseteq \mathcal{R}(\mathbf{C})$. Our proof continues as follows:

Proof C. As established previously, $\mathbf{R_{ni}} \succeq 0$ and $\mathbf{0}_{N \times N} \succeq 0$. Clearly, the matrix square root of the all zeros matrix is itself. Thus, for $\mathcal{H}_{\mathbf{bb}^{*}}(f_{o})$ to be PSD, $\mathbf{\Pi}_{Q} = \mathbf{R_{ni}}^{1/2}\mathbf{X}\mathbf{0}_{N \times N}^{1/2} = \mathbf{0}_{NML \times N}$, which will be true for any contraction $\mathbf{X}$. As established previously, $\mathbf{\Pi}_{Q} = \mathbf{0}_{NML \times N}$ only if, for each stationary point $\mathbf{w}_{o}, \mathbf{s}_{o}$ in the feasible set, $\mathbf{w}_{o}^{H}\mathbf{\Gamma}_{q}\mathbf{s}_{o} = 0$ for all clutter patches q. Therefore, this is the necessary and sufficient condition for $\mathcal{H}_{\mathbf{bb}^{*}}(f_{o})$ to be PSD. $\qquad\qquad\square$

In any case, $\mathbf{w}_{o}^{H}\mathbf{\Gamma}_{q}\mathbf{s}_{o} = 0$ for all clutter patches q and stationary points $\mathbf{w}_{o}, \mathbf{s}_{o}$ in order for the Hessian to be positive-semidefinite and the objective to be convex. This is a clearly illogical condition – if such a beamformer-signal pair existed *a priori*, there is effectively no clutter in the region of the desired target response, the signal design is arbitrary and we would only design a beamformer! Additionally, this condition also requires $\mathbf{R_{ni}}\mathbf{w}_{o} = \mathbf{0}_{NML \times 1}$, via the stationary point definitions. If $\mathbf{R_{ni}}$ is full rank, then $\mathbf{w}_{o} = \mathbf{0}_{NML \times 1}$, and we do no processing whatsoever.

Since the only possible ways to not obtain a contradiction are themselves contradictions, we must conclude that, regardless of the beamformer-signal pair, the overall cost function Hessian is indefinite (since the same contradiction would be reached if we desired a negative definite $\mathcal{H}_{\mathbf{bb}^{*}}(f_{o})$ instead) and the problem is not jointly convex.

IV. THE BIQUADRATIC PROGRAM & RELAXATIONS

In this section, we discuss the relatively unexplored area of biquadratic programming – that is, joint optimization of two multidimensional variables over a cost function that is quadratic in each variable – and its application to fully adaptive radar, which arises from joint signal-beamformer design schemes.

First, we will review the existing literature on biquadratic programming (henceforth, BQP) and related works on tensor approximations, in order to provide a basic mathematical framework. We will then recall the work of Setlur, et al. in order to construct a relevant BQP and its potentially solvable relaxations, with some additional insights into the other unique challenges this problem presents that are unaddressed by the literature.

Interestingly, coverage of the BQP in the literature, from the optimization community or otherwise, can be charitably described as sparse; in fact, there are only seven papers directly addressing the BQP produced by a total of ten authors, all of which have been released within the last six years. This is somewhat puzzling given the depth and breadth of related problems both cited in the works mentioned below and conceivable by an ordinary engineer – applications range from economics to quantum physics to the fully-adaptive radar concept we wish to pursue here. Nevertheless, this limitation allows a reasonable overview of the state-of-the-art without too much difficulty. Other research exists on similar problems, mostly in the areas of nearest tensor approximation, but this is extensive and highly unspecialized. Therefore, we refer the reader to the multiple overlapping sources within the papers mentioned below on these matters.

A. The General BQP

We begin by describing the general biquadratic program analyzed by Ling, et al. in [13]. Consider the vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$. The homogenous biquadratic optimization problem is

$$\min_{\mathbf{x},\mathbf{y}} \quad b(\mathbf{x},\mathbf{y}) = \sum_{1 \leq i,k \leq n, 1 \leq j,l \leq m} b_{ijkl} x_i y_j x_k y_l$$

$$\text{subject to} \quad \|\mathbf{x}\| = 1, \|\mathbf{y}\| = 1,$$

(24)

where b_{ijkl} is the (i,j,k,l)th element of the 4th-order tensor $\mathcal{B} \in \mathbb{R}^{n \times m \times n \times m}$, the subscripted x and y are the appropriate element of the corresponding vector, and $\|\cdot\|$ is the standard Euclidean 2-norm. As the authors note, for fixed $\mathbf{x}$ (resp. $\mathbf{y}$), this problem is quadratic in $\mathbf{y}$ (resp. $\mathbf{x}$), can be solved quite easily and, in fact, is convex if the appropriate inner matrix is positive semidefinite. This property also leads to the name of this class of problems (cf. *bilinear* optimization). Before we continue, it is worth noting that the objective function $b(\mathbf{x},\mathbf{y})$ can be rewritten using a tensor operator and the matrix inner product, *viz.*

$$b(\mathbf{x},\mathbf{y}) = (\mathcal{B}\mathbf{x}\mathbf{x}^T) \bullet (\mathbf{y}\mathbf{y}^T) = (\mathbf{y}\mathbf{y}^T\mathcal{B}) \bullet (\mathbf{x}\mathbf{x}^T)$$

where $\mathbf{X} \bullet \mathbf{Y} = \text{Tr}(\mathbf{X}^T\mathbf{Y})$, and the tensor-operator matrices are $\mathcal{B}\mathbf{x}\mathbf{x}^T = \sum_{i,k=1}^n b_{ijkl} x_i x_k$ and $\mathbf{y}\mathbf{y}^T\mathcal{B} = \sum_{j,l=1}^m b_{ijkl} y_j y_l$.

The above problem has been modified elsewhere in the literature, mostly in terms of the constraints. Bomze, et al. [21] constrain the objective in Equation 24 to lie on the simplex for each variable. Ling, Zhang, and Qi have described strategies when the problem is quadratically constrained in [14], [22].

Regardless of the constraints, [13] proves that the BQP is nonconvex, that both the BQP and its naive semidefinite relaxation (which we will more explicitly discuss in the next section) are NP-hard problems, and that the initial general problem does not even admit a polynomial time *approximation* algorithm. This would seem to bode ill for any hope of reasonable solutions, but the authors demonstrate that polynomial-time approximation algorithms exist (with bounds approximately inversely proportional to the dimension of the variables) when the objective is square-free or has squared terms in only one of the variables. Finally, they demonstrate the creation of such algorithms, showing a tradeoff between speed and accuracy for sum-of-squares based solvers (better accuracy) versus convex semidefinite relaxation solvers (faster). Additionally, if the structure is sufficiently sparse, then existing solvers can attack the semidefinite relaxation even more efficiently.

B. The Problem at Hand

Before continuing, we recall the basics of our problem. We operate on a radar datacube collected over L pulses with N fast time samples per pulse in M spatial bins. Our goal is to find a STAP beamformer vector $\mathbf{w} \in \mathbb{C}^{NML}$ and a transmit signal $\mathbf{s} \in \mathbb{C}^N$ that minimizes the combined effect of the noise and interference represented by the covariance matrix $\mathbf{R_{ni}} \in \mathbb{C}^{NML \times NML}$) and the signal-dependent clutter. The signal-dependent clutter is modeled as Q independent clutter patches. Each patch has an individual spatiodoppler response matrix $\mathbf{\Gamma}_q \in \mathbb{C}^{NML \times N}$, which ties into the per-patch clutter covariance $\mathbf{R}_\gamma^q(\mathbf{s}) = \mathbf{\Gamma}_q \mathbf{s}\mathbf{s}^H \mathbf{\Gamma}_q^H$. The overall clutter covariance is then $\sum_{q=1}^{Q} \mathbf{\Gamma}_q \mathbf{s}\mathbf{s}^H \mathbf{\Gamma}_q^H$.

We constrain this minimization in two ways. First, for a given target space-time-doppler bin, we want a particular filter output, say, $\kappa \in \mathbb{C}$. This filter output can be represented by the Capon beamformer equation $\mathbf{w}^H \mathbf{T}\mathbf{s}$, where $\mathbf{T} \in \mathbb{C}^{NML \times N}$ is the spatiodoppler response of the target bin. Second, we place an upper bound on the total signal power, say, P_o. With these constraints in place, the overall STAP problem is given by

$$\begin{aligned} \min_{\mathbf{w},\mathbf{s}} \quad & \mathbf{w}^H \mathbf{R_{ni}} \mathbf{w} + \mathbf{w}^H \left(\sum_{q=1}^{Q} \mathbf{\Gamma}_q \mathbf{s}\mathbf{s}^H \mathbf{\Gamma}_q^H \right) \mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}^H \mathbf{T}\mathbf{s} = \kappa \\ & \mathbf{s}^H \mathbf{s} \le P_o \end{aligned} \qquad \text{(BQP 1)}$$

If we treat the signal and beamformer as a single stacked variable, say $\mathbf{b} = \begin{bmatrix} \mathbf{w}^T & \mathbf{s}^T \end{bmatrix}^T \in \mathbb{C}^{N(ML+1)}$, we can find an equivalent form of the above optimization problem in the new variable. Define $\mathbf{\Psi}_W, \mathbf{\Psi}_S$ as the matrices that recover $\mathbf{w}, \mathbf{s}$ from $\mathbf{b}$, i.e. $\mathbf{w} = \mathbf{\Psi}_W \mathbf{b}$ and $\mathbf{s} = \mathbf{\Psi}_S \mathbf{b}$. Then, we can define the optimization problem as

$$\begin{aligned} \min_{\mathbf{b}} \quad & \mathbf{b}^H \widetilde{\mathbf{R}}_{\mathbf{ni}} \mathbf{b} + \mathbf{b}^H \left(\sum_{q=1}^{Q} \overline{\mathbf{\Gamma}}_q \mathbf{b}\mathbf{b}^H \overline{\mathbf{\Gamma}}_q^H \right) \mathbf{b} \\ \text{s.t.} \quad & \mathbf{b}^H \widetilde{\mathbf{T}} \mathbf{b} = \kappa \\ & \mathbf{b}^H \mathbf{\Psi}_S^T \mathbf{\Psi}_S \mathbf{b} \le P_o \end{aligned} \qquad \text{(BQP 2)}$$

where $\widetilde{\mathbf{R}}_{\mathbf{ni}} = \mathbf{\Psi}_W^T \mathbf{R_{ni}} \mathbf{\Psi}_W$, $\overline{\mathbf{\Gamma}}_q = \mathbf{\Psi}_W^T \mathbf{\Gamma}_q \mathbf{\Psi}_S$, and $\widetilde{\mathbf{T}} = \mathbf{\Psi}_W^T \mathbf{T} \mathbf{\Psi}_S$ are just "expanded" versions of the matrices seen in the previous problem.

In either case, these are clearly biquadratic programs, which we've shown are non-convex. The question now is: can we find a solvable representation, relaxed or otherwise?

C. Getting to the SDR

Our first goal is using the notation/mechanisms of the literature to find the semidefinite relaxation of the equivalent programs BQP 1 and BQP 2. First, assume we have defined a tensor $\mathcal{C}$ such that the following operator relation holds: $\mathcal{C}\mathbf{ss}^H = \sum_{q=1}^{Q} \mathbf{R}_{\tilde{\gamma}}^q(\mathbf{s})$. Similiarly, assume an expanded form of this tensor exists, say $\widetilde{\mathcal{C}}$, such that $\widetilde{\mathcal{C}}\mathbf{bb}^H = \sum_{q=1}^{Q} \overline{\mathbf{\Gamma}}_q \mathbf{bb}^H \overline{\mathbf{\Gamma}}_q^H$. What this tensor looks like, exactly, will be seen in a later section.

On complex matrices, the inner product becomes $\mathbf{X} \bullet \mathbf{Y} = \mathrm{Re}\{\mathrm{Tr}(\mathbf{X}^H\mathbf{Y})\}$. We'd like to get everything into a real form so we can use this inner product somehow, as in the literature.

Note the first constraint (since κ is complex) implies its conjugate must also be constrained, i.e.

$$\left(\mathbf{w}^H\mathbf{Ts}\right)^* = \kappa^*$$

$$\mathbf{w}^T\mathbf{T}^*\mathbf{s}^* = \kappa^*$$

$$\mathbf{s}^H\mathbf{T}^H\mathbf{w} = \kappa^*$$

Hence, the target-based constraints are

$$\mathbf{w}^H\mathbf{Ts} = \kappa$$

$$\mathbf{s}^H\mathbf{T}^H\mathbf{w} = \kappa^*$$

We can recast this constraint as two real constraints. First, recall the obvious things about complex numbers:

$$\kappa + \kappa^* = 2\,\mathrm{Re}\{\kappa\} = 2\kappa_R \Rightarrow \frac{1}{2}\left(\kappa + \kappa^*\right) = \kappa_R$$

$$\kappa - \kappa^* = \jmath 2\,\mathrm{Im}\{\kappa\} = \jmath 2\kappa_I \Rightarrow \frac{1}{2\jmath}\left(\kappa - \kappa^*\right) = \kappa_I.$$

If we substitute in the Capon constraints, we then obtain

$$\frac{1}{2}\left(\kappa + \kappa^*\right) = \frac{1}{2}\left(\mathbf{w}^H\mathbf{Ts} + \mathbf{s}^H\mathbf{T}^H\mathbf{w}\right) = \mathrm{Re}\{\mathbf{w}^H\mathbf{Ts}\}$$

$$= \frac{1}{2}\begin{bmatrix}\mathbf{w}\\\mathbf{s}\end{bmatrix}^H \begin{bmatrix}\mathbf{0}_{NML\times NML} & \mathbf{T}\\ \mathbf{T}^H & \mathbf{0}_{N\times N}\end{bmatrix}\begin{bmatrix}\mathbf{w}\\\mathbf{s}\end{bmatrix} = \mathbf{b}^H\widetilde{\mathbf{T}}_H\mathbf{b} = \kappa_R$$

and

$$\frac{1}{2\jmath}\left(\kappa - \kappa^*\right) = \frac{1}{2\jmath}\left(\mathbf{w}^H\mathbf{Ts} - \mathbf{s}^H\mathbf{T}^H\mathbf{w}\right) = \mathrm{Im}\{\mathbf{w}^H\mathbf{Ts}\}$$

$$= \frac{1}{2\jmath}\begin{bmatrix}\mathbf{w}\\\mathbf{s}\end{bmatrix}^H \begin{bmatrix}\mathbf{0}_{NML\times NML} & \mathbf{T}\\ -\mathbf{T}^H & \mathbf{0}_{N\times N}\end{bmatrix}\begin{bmatrix}\mathbf{w}\\\mathbf{s}\end{bmatrix} = \mathbf{b}^H\widetilde{\mathbf{T}}_A\mathbf{b} = \kappa_I$$

where the subscripts H and A indicate the hermitian and anti-hermitian part of the expanded matrix $\widetilde{\mathbf{T}}$, respectively. (As an aside, recall that $\widetilde{\mathbf{T}}_H^H = \widetilde{\mathbf{T}}_H$ and $\widetilde{\mathbf{T}}_A^H = -\widetilde{\mathbf{T}}_A$ by definition) This leads us to a new version of the second form of the optimization problem, as a function of the combined beamformer-signal vector:

$$\min_{\mathbf{b}} \quad \mathbf{b}^H \widetilde{\mathbf{R}}_{\mathbf{ni}} \mathbf{b} + \mathbf{b}^H \left(\sum_{q=1}^{Q} \overline{\mathbf{\Gamma}}_q \mathbf{b}\mathbf{b}^H \overline{\mathbf{\Gamma}}_q^H \right) \mathbf{b}$$

$$\text{s.t.} \quad \mathbf{b}^H \widetilde{\mathbf{T}}_H \mathbf{b} = \kappa_R$$

$$\mathbf{b}^H \widetilde{\mathbf{T}}_A \mathbf{b} = \kappa_I$$

$$\mathbf{b}^H \mathbf{\Psi}_S^T \mathbf{\Psi}_S \mathbf{b} \leq P_o$$

or, using the tensor form,

$$\min_{\mathbf{b}} \quad \mathbf{b}^H \widetilde{\mathbf{R}}_{\mathbf{ni}} \mathbf{b} + \mathbf{b}^H \left(\widetilde{\mathcal{C}}\mathbf{b}\mathbf{b}^H \right) \mathbf{b}$$

$$\text{s.t.} \quad \mathbf{b}^H \widetilde{\mathbf{T}}_H \mathbf{b} = \kappa_R$$

$$\mathbf{b}^H \widetilde{\mathbf{T}}_A \mathbf{b} = \kappa_I$$

$$\mathbf{b}^H \mathbf{\Psi}_S^T \mathbf{\Psi}_S \mathbf{b} \leq P_o$$

Using the properties of trace and the fact that all quantities in this optimization problem are now real, we can recast it using the inner product from above:

$$\min_{\mathbf{b}} \quad \widetilde{\mathbf{R}}_{\mathbf{ni}} \bullet \mathbf{b}\mathbf{b}^H + \left(\widetilde{\mathcal{C}}\mathbf{b}\mathbf{b}^H \right) \bullet \mathbf{b}\mathbf{b}^H$$

$$\text{s.t.} \quad \widetilde{\mathbf{T}}_H \bullet \mathbf{b}\mathbf{b}^H = \kappa_R$$

$$\widetilde{\mathbf{T}}_A \bullet \mathbf{b}\mathbf{b}^H = -\kappa_I$$

$$\mathbf{\Psi}_S^T \mathbf{\Psi}_S \bullet \mathbf{b}\mathbf{b}^H \leq P_o$$

Note that we've used the Hermitian (resp. anti-Hermitian) nature of $\widetilde{\mathbf{R}}_{\mathbf{ni}}$ and $\widetilde{\mathbf{T}}_H$ (resp. $\widetilde{\mathbf{T}}_A$) to avoid the Hermitian superscript everywhere.

A common path to semidefinite relaxations defines a matrix variable, say, $\mathbf{B} = \mathbf{b}\mathbf{b}^H$. It is clear that $\mathbf{Z}$ is Hermitian and positive semidefinite, and $\text{rank}(\mathbf{B}) = 1$. This means our optimization problem above is equivalent to the matrix quadratic program

$$\min_{\mathbf{B} \in \{\mathbf{B} = \mathbf{B}^H, \mathbf{B} \succeq 0\}} \quad \widetilde{\mathbf{R}}_{\mathbf{ni}} \bullet \mathbf{B} + \left(\widetilde{\mathcal{C}}\mathbf{B} \right) \bullet \mathbf{B}$$

$$\text{s.t.} \quad \widetilde{\mathbf{T}}_H \bullet \mathbf{B} = \kappa_R \qquad \widetilde{\mathbf{T}}_A \bullet \mathbf{B} = -\kappa_I$$

$$\mathbf{\Psi}_S^T \mathbf{\Psi}_S \bullet \mathbf{Z} \leq P_o \qquad \text{rank}(\mathbf{Z}) = 1$$

Depending on the properties of $\widetilde{\mathcal{C}}$, this could be a convex problem in $\mathbf{Z}$ if not for the rank condition, which is the primary hurdle. This is where the "relaxation" in semidefinite relaxation comes in. If we permit $\mathbf{Z}$ to take any

rank, we have relaxed the problem to the (possibly) convex one:

$$\min_{\mathbf{Z}\in\{\mathbf{Z}=\mathbf{Z}^H,\mathbf{Z}\succeq 0\}} \quad \widetilde{\mathbf{R}}_{\mathbf{ni}} \bullet \mathbf{Z} + \left(\widetilde{\mathcal{C}}\mathbf{Z}\right) \bullet \mathbf{Z}$$

$$\text{s.t.} \qquad \widetilde{\mathbf{T}}_H \bullet \mathbf{Z} = \kappa_R$$

$$\widetilde{\mathbf{T}}_A \bullet \mathbf{Z} = -\kappa_I \tag{SDR}$$

$$\mathbf{\Psi}_S^T \mathbf{\Psi}_S \bullet \mathbf{Z} \leq P_o.$$

Notice we did not, at any point, progress through a matrix bilinear form in finding this semidefinite relaxation, as seen in [13], etc. This is because the constraint set in this problem prevents such a representation – namely, the Capon constraint, which is bilinear in the vector arguments already.

If we can obtain a solution to this problem, a natural question is how to apply this solution to the original problem. If we are lucky, the rank of the relaxed solution will be one, and there will be nothing left to do. If the rank of the solution is greater than one, then we can use the nearest rank-one approximation that still satisfies the constraints. Some authors have We will see in the simulations below that for this particular scenario, we can obtain solutions that are nearly rank-one in a numerical sense.

V. Solutions of the Relaxed BQP

Recall that, when standardized, the semidefinite relaxation of the BQP is:

$$\min_{\mathbf{B}\in\{\mathbf{X}\in\mathbb{C}^{J\times J}|\mathbf{X}=\mathbf{X}^H,\mathbf{X}\succeq 0\}} \quad \widetilde{\mathbf{R}}_{\mathbf{ni}} \bullet \mathbf{B} + \left(\widetilde{\mathcal{C}}\mathbf{B}\right) \bullet \mathbf{B}$$

$$\text{s.t.} \qquad \widetilde{\mathbf{T}}_R \bullet \mathbf{B} - \kappa_R = 0$$

$$\widetilde{\mathbf{T}}_I \bullet \mathbf{B} - \kappa_I = 0 \tag{SSDR}$$

$$\mathbf{\Psi}_S^T \mathbf{\Psi}_S \bullet \mathbf{B} - P_o \leq 0.$$

Before we begin, we reformulate the problem to use the variable $\text{vec}(\mathbf{B})$. First, note that $\text{Tr}(\mathbf{AB}) = \text{vec}^T(\mathbf{A}^T)\,\text{vec}(\mathbf{B})$. For Hermitian $\mathbf{A}$, this becomes $\text{Tr}(\mathbf{AB}) = \text{vec}^H(\mathbf{A})\,\text{vec}(\mathbf{B})$ (since $\mathbf{A}^T = \mathbf{A}^*$ and the conjugation operator is linear). This allows us to say that, for Hermitian $\mathbf{A}$, the relevant inner product on $\mathbb{C}$ is $\mathbf{A} \bullet \mathbf{B} = \text{vec}^H(\mathbf{A})\,\text{vec}(\mathbf{B})$.

Next, recall that the tensor $\widetilde{\mathcal{C}}$ can be unfolded into the vector matrix $\widetilde{\mathbf{C}}_V = \sum_{q=1}^{Q} \text{vec}(\overline{\mathbf{\Gamma}}_q)\,\text{vec}(\overline{\mathbf{\Gamma}}_q)^H$. Using these, we recast SSDR into a vectorized form:

$$\min_{\mathbf{B}\in\{\mathbf{X}\in\mathbb{C}^{J\times J}|\mathbf{X}=\mathbf{X}^H,\mathbf{X}\succeq 0\}} \quad \text{vec}^H(\widetilde{\mathbf{R}}_{\mathbf{ni}})\,\text{vec}(\mathbf{B}) + \text{vec}^H(\mathbf{B})\widetilde{\mathbf{C}}_V\,\text{vec}(\mathbf{B})$$

$$\text{s.t.} \qquad \text{vec}^H(\widetilde{\mathbf{T}}_R)\,\text{vec}(\mathbf{B}) - \kappa_R = 0$$

$$\text{vec}^H(\widetilde{\mathbf{T}}_I)\,\text{vec}(\mathbf{B}) - \kappa_I = 0 \tag{VSSDR}$$

$$\text{vec}^H(\mathbf{\Psi}_S^T \mathbf{\Psi}_S)\,\text{vec}(\mathbf{B}) - P_o \leq 0.$$

In a further "development", let us explicitly put the equality constraint back as a complex number and break out the positive semi-definiteness as a separate constraint. This makes VSSDR turn into

$$\min_{\mathbf{B} \in \{\mathbf{X} \in \mathbb{C}^{J \times J} | \mathbf{X} = \mathbf{X}^H\}} \quad \mathrm{vec}^H(\widetilde{\mathbf{R}}_{\mathbf{ni}}) \, \mathrm{vec}(\mathbf{B}) + \mathrm{vec}^H(\mathbf{B}) \widetilde{\mathbf{C}}_V \, \mathrm{vec}(\mathbf{B})$$

$$\text{s.t.} \quad \mathrm{vec}^H(\mathbf{B}) \, \mathrm{vec}(\widetilde{\mathbf{T}}) - \kappa = 0$$

$$\mathrm{vec}^H(\mathbf{\Psi}_S^T \mathbf{\Psi}_S) \, \mathrm{vec}(\mathbf{B}) - P_o \leq 0$$

$$\mathbf{B} \succeq 0$$

According to Hjørungnes, there are certain methods we need to account for the Hermitian constraint in the following analysis. For notational ease, let us define the following vectors:

$$\boldsymbol{\psi}_S = \mathrm{vec}(\mathbf{\Psi}_S^T \mathbf{\Psi}_S) \qquad\qquad \boldsymbol{\rho} = \mathrm{vec}(\widetilde{\mathbf{R}}_{\mathbf{ni}}).$$

Furthermore, if we define the following matrices,

$$\mathbf{P}_{SS} = \mathbf{\Psi}_S \otimes \mathbf{\Psi}_S \qquad\qquad \mathbf{P}_{WW} = \mathbf{\Psi}_W \otimes \mathbf{\Psi}_W$$

$$\mathbf{P}_{SW} = \mathbf{\Psi}_S \otimes \mathbf{\Psi}_W \qquad\qquad \mathbf{P}_{WS} = \mathbf{\Psi}_W \otimes \mathbf{\Psi}_S,$$

then these can be rewritten as

$$\boldsymbol{\psi}_S = \mathbf{P}_{SS}^T \, \mathrm{vec}(\mathbf{I}_N) \qquad\qquad \boldsymbol{\rho} = \mathbf{P}_{WW}^T \, \mathrm{vec}(\mathbf{R}_{\mathbf{ni}}).$$

A. Proof of Convexity

Before continuing, we demonstrate directly that this relaxed optimization problem is indeed convex. Recall that the relaxed problem is

$$\min_{\mathbf{B} \in \{\mathbf{X} \in \mathbb{C}^{J \times J} | \mathbf{X} = \mathbf{X}^H\}} \quad \mathrm{vec}^H(\widetilde{\mathbf{R}}_{\mathbf{ni}}) \, \mathrm{vec}(\mathbf{B}) + \mathrm{vec}^H(\mathbf{B}) \widetilde{\mathbf{C}}_V \, \mathrm{vec}(\mathbf{B})$$

$$\text{s.t.} \quad \mathrm{vec}^H(\mathbf{B}) \, \mathrm{vec}(\widetilde{\mathbf{T}}) - \kappa = 0$$

$$\mathrm{vec}^H(\mathbf{\Psi}_S^T \mathbf{\Psi}_S) \, \mathrm{vec}(\mathbf{B}) - P_o \leq 0$$

$$\mathbf{B} \succeq 0.$$

A convex optimization problem minimizes a convex objective over a convex set. A well-known restriction of this concept is given in [12, pp. 136-137] as follows: for a given optimization problem, if

1) the objective function is convex,

2) the inequality constraint functions are convex, and,

3) the equality constraint functions are affine (i.e. it looks like $\mathbf{a}^H \mathbf{x} - b$.)

then the problem is convex. This mostly stems from the fact that affine equality constraints define a polytope, which is convex, and so its intersection with the existing convex set is also convex.

We start with the objective. A quadratic function is convex if the associated matrix is positive semidefinite. Thus, since $\tilde{\mathbf{C}}_V \succeq 0$ by definition (as a sum of rank one Hermitian matrices), then $\mathrm{vec}^H(\mathbf{B}) \widetilde{\mathbf{C}}_V \, \mathrm{vec}(\mathbf{B})$ is a convex function. Adding an affine function to it does not change the convexity, so the overall objective is convex. Continuing

to the constraints, both the power constraint and the semidefiniteness constraints are convex – the former because it is affine, the latter because it describes a convex cone. Finally, the equality constraint is convex, because it can be broken up into two real affine constraints on the real and imaginary parts. Therefore, the relaxed problem is convex. However, since $\tilde{\mathbf{C}}_V$ is typically *not* full rank (indeed, the rank is bounded above by ML in this case), it is not usually strictly convex, and thus there will exist a multitude of solutions.

B. Slater's Condition

Recall that *strong duality* is said to hold for a given optimization problem is the primal problem is convex and it satisfies Slater's condition (aka the problem is *strictly feasible*). A consequence of this property is that there is no duality gap between solutions of the primal and dual problems. We will show that our problem satisfies Slater's condition given a relationship between the constraint values.

Given an optimization problem

$$\min_{\mathbf{x} \in \Omega} \quad f_o(\mathbf{x})$$
$$\text{s.t.} \quad g_i(\mathbf{x}) \leq 0 \quad i = 1, \dots, m$$
$$h_j(\mathbf{x}) = 0 \quad j = 1, \dots, n$$

(where Ω is a convex set), let us assume $f_o(\mathbf{x})$ is convex and define the set $\mathcal{D} = \Omega \cap \text{dom}(f_o) \cap (\cap_{i=1}^m \text{dom}(g_i))$ to be the total feasible domain of the problem. Slater's condition is satisfied if there exists at least one point in the relative interior (i.e. not on the boundary) of the problem's feasibility set that satisfy all of the equality constraints and strictly satsifies the inequality constraints. Mathematically, we can represent the condition as follows:

Lemma 1 (Slater's Condition). *For the standard optimization problem above, if each g_i is convex and there exists a point $\mathbf{x}_o \in \text{relint}(\mathcal{D})$ such that $g_i(\mathbf{x}_o) < 0 \ \forall i$ and $h_j(\mathbf{x}_o) = 0 \ \forall j$, then strong duality holds and there is zero duality gap.*

For our problem, the feasibility set is positive semidefinite matrices, which means its relative interior is positive-definite matrices. Our constraint functions are

$$g(\mathbf{B}) = \boldsymbol{\psi}_S^T \boldsymbol{\beta} - P_o$$
$$h_1(\mathbf{B}) = \boldsymbol{\beta}^H \tilde{\boldsymbol{\tau}} - \kappa \ = \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \boldsymbol{\beta} - \kappa$$
$$h_2(\mathbf{B}) = \tilde{\boldsymbol{\tau}}^H \boldsymbol{\beta} - \kappa^*.$$

where $\boldsymbol{\beta} = \text{vec}(\mathbf{B})$ for a given Hermitian matrix $\mathbf{B}$. Thus, we need to find a matrix $\mathbf{B} \succ 0$ that satisfies the above equations.

We start with solving for $\boldsymbol{\beta}$ in our equality constraints, then ensure that the resulting matrix is both positive definite and strictly satisfies the power inequality. As a combined matrix-vector equation, we have:

$$\begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \\ \tilde{\boldsymbol{\tau}}^H \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} \kappa \\ \kappa^* \end{bmatrix}$$

If solutions of this equation exist (and they should exist), then they are given by

$$\boldsymbol{\beta} = \begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \\ \tilde{\boldsymbol{\tau}}^H \end{bmatrix}^\dagger \begin{bmatrix} \kappa \\ \kappa^* \end{bmatrix} + \left(\mathbf{I}_{N^2(ML+1)^2} - \begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \\ \tilde{\boldsymbol{\tau}}^H \end{bmatrix}^\dagger \begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \\ \tilde{\boldsymbol{\tau}}^H \end{bmatrix} \right) \mathbf{z}$$

where $\dagger$ indicates the pseudoinverse and $\mathbf{z}$ is the vectorization of a matrix $\mathbf{Z}$ similarly shaped to $\mathbf{B}$.

A logical question is, then, what is that pseudoinverse (and consequently, what is the existence condition)? Since the matrix $\begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \\ \tilde{\boldsymbol{\tau}}^H \end{bmatrix}$ is fat, we might have some hope in the form $\mathbf{C}^\dagger = \mathbf{C}^H (\mathbf{C}\mathbf{C}^H)^{-1}$. First, we can find the matrix we hope to invert:

$$\begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \\ \tilde{\boldsymbol{\tau}}^H \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \\ \tilde{\boldsymbol{\tau}}^H \end{bmatrix}^H = \begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \mathbf{K}_{J,J} \tilde{\boldsymbol{\tau}}^* & \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \tilde{\boldsymbol{\tau}} \\ \tilde{\boldsymbol{\tau}}^H \mathbf{K}_{J,J} \tilde{\boldsymbol{\tau}}^* & \tilde{\boldsymbol{\tau}}^H \tilde{\boldsymbol{\tau}} \end{bmatrix}$$

$$= \begin{bmatrix} \|\mathbf{T}\|_F^2 & 0 \\ 0 & \|\mathbf{T}\|_F^2 \end{bmatrix} = \|\mathbf{T}\|_F^2 \mathbf{I}_2.$$

So long as we have an actual target, this is always invertible (and hence, a solution always exists). This means that the pseudoinverse of interest is

$$\begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \\ \tilde{\boldsymbol{\tau}}^H \end{bmatrix}^\dagger = \frac{1}{\|\mathbf{T}\|_F^2} \begin{bmatrix} \mathbf{K}_{J,J} \tilde{\boldsymbol{\tau}}^* & \tilde{\boldsymbol{\tau}} \end{bmatrix}.$$

Additionally, the projection matrix in the solution is

$$\begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \\ \tilde{\boldsymbol{\tau}}^H \end{bmatrix}^\dagger \begin{bmatrix} \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} \\ \tilde{\boldsymbol{\tau}}^H \end{bmatrix} = \frac{1}{\|\mathbf{T}\|_F^2} \left(\mathbf{K}_{J,J} \tilde{\boldsymbol{\tau}}^* \tilde{\boldsymbol{\tau}}^T \mathbf{K}_{J,J} + \tilde{\boldsymbol{\tau}} \tilde{\boldsymbol{\tau}}^H \right)$$

$$= \frac{1}{\|\mathbf{T}\|_F^2} \left(\mathbf{P}_{WS}^T \boldsymbol{\tau}_H \boldsymbol{\tau}_H^H \mathbf{P}_{WS} + \mathbf{P}_{SW}^T \boldsymbol{\tau} \boldsymbol{\tau}^H \mathbf{P}_{SW} \right)$$

where $\boldsymbol{\tau}_H = \text{vec}(\mathbf{T}^H)$.

We can break our solution into the "min-norm" and "nullspace" parts. Clearly, the min-norm part is

$$\mathbf{B}_{min} = \frac{1}{\|\mathbf{T}\|_F^2} \begin{bmatrix} \mathbf{0}_{NML \times NML} & \kappa^* \mathbf{T} \\ \kappa \mathbf{T}^H & \mathbf{0}_{N \times N} \end{bmatrix}.$$

It is clear from the structure that this is not a positive definite matrix (or even a positive semidefinite one), so we require the nullspace component to place us into the relative interior. If we break the vector $\mathbf{z}$ into components as $\mathbf{z} = \mathbf{P}_{WW}^T \mathbf{z}_1 + \mathbf{P}_{SW}^T \mathbf{z}_2 + \mathbf{P}_{WS}^T \mathbf{z}_{2H} + \mathbf{P}_{SS}^T \mathbf{z}_3$, then the nullspace solution is:

$$\mathbf{B}_{null} = \mathbf{Z} - \frac{1}{\|\mathbf{T}\|_F^2} \begin{bmatrix} \mathbf{0}_{NML \times NML} & \text{tr}(\mathbf{T}^H \mathbf{Z}_2) \mathbf{T} \\ \text{tr}(\mathbf{Z}_2^H \mathbf{T}) \mathbf{T}^H & \mathbf{0}_{N \times N} \end{bmatrix}.$$

Hence, the overall matrix that satisfies the equality constraints is

$$\mathbf{B} = \mathbf{Z} + \frac{1}{\|\mathbf{T}\|_F^2} \begin{bmatrix} \mathbf{0}_{NML \times NML} & (\kappa^* - \text{tr}(\mathbf{T}^H \mathbf{Z}_2)) \mathbf{T} \\ (\kappa - \text{tr}(\mathbf{Z}_2^H \mathbf{T})) \mathbf{T}^H & \mathbf{0}_{N \times N} \end{bmatrix}.$$

Our problem now becomes finding a matrix $\mathbf{Z}$ such that $\mathbf{B} \succ 0$ and $\text{tr}(\mathbf{Z}_3) < P_o$. One method of attack is setting $\mathbf{Z}_2 = \mathbf{0}_{NML \times N}$ and attempting to reach a valid result this way. Under this assumption, our solution matrix is

$$\mathbf{B} = \begin{bmatrix} \mathbf{Z}_1 & \frac{\kappa^*}{\|\mathbf{T}\|_F^2}\mathbf{T} \\ \frac{\kappa}{\|\mathbf{T}\|_F^2}\mathbf{T}^H & \mathbf{Z}_3 \end{bmatrix}.$$

The positive definiteness requirement can then be expressed in one of two ways:

$$\mathbf{Z}_1 \succ 0, \quad \mathbf{Z}_3 - \frac{|\kappa|^2}{\|\mathbf{T}\|_F^4}\mathbf{T}^H\mathbf{Z}_1^{-1}\mathbf{T} \succ 0$$

$$\mathbf{Z}_3 \succ 0, \quad \mathbf{Z}_1 - \frac{|\kappa|^2}{\|\mathbf{T}\|_F^4}\mathbf{T}\mathbf{Z}_3^{-1}\mathbf{T}^H \succ 0.$$

Here, we rely on another judicious guess, setting $\mathbf{Z}_1 = \mathbf{I}_{NML}$, which is clearly positive definite.[1]Then, we only need to find a $\mathbf{Z}_3$ such that $\mathbf{Z}_3 \succ \frac{|\kappa|^2}{\|\mathbf{T}\|_F^4}\mathbf{T}^H\mathbf{T}$ *and* $\text{tr}(\mathbf{Z}_3) < P_o$. Let us assume that such a matrix exists. Following [20, Corollary 7.7.4(d)], since $\mathbf{Z}_3 \succ \frac{|\kappa|^2}{\|\mathbf{T}\|_F^4}\mathbf{T}^H\mathbf{T}$, then $\text{tr}(\mathbf{Z}_3) > \frac{|\kappa|^2}{\|\mathbf{T}\|_F^4}\text{tr}(\mathbf{T}^H\mathbf{T}) = \frac{|\kappa|^2}{\|\mathbf{T}\|_F^2}$. But we already know that $\text{tr}(\mathbf{Z}_3) < P_o$. Hence, we have the chained inequality

$$P_o > \text{tr}(\mathbf{Z}_3) > \frac{|\kappa|^2}{\|\mathbf{T}\|_F^2},$$

which obviously requires that $P_o > \frac{|\kappa|^2}{\|\mathbf{T}\|_F^2}$ or perhaps $\|\mathbf{T}\|_F^2 > \frac{|\kappa|^2}{P_o}$.

With this in hand, we can now construct our matrix. Since $NML > N$ and (generally) $\text{rank}(\mathbf{T}) = N$, then $\mathbf{T}^H\mathbf{T} \succ 0$.[2] For any matrix $\mathbf{A} \succ 0$ and real numbers α, β, $\alpha\mathbf{A} \succ \beta\mathbf{A}$ if and only if $\alpha > \beta$. Let $\mathbf{Z}_3 = \frac{P_o - \varepsilon}{\|\mathbf{T}\|_F^2}\mathbf{T}^H\mathbf{T}$, where $\varepsilon > 0$ is some small real number. This guarantees that the trace feasibility is strictly satisfied, since $\text{tr}(\mathbf{Z}_3) = P_o - \varepsilon < P_o$. It also satisfies the chained inequality above, so long as $\epsilon < P_o - \frac{|\kappa|^2}{\|\mathbf{T}\|_F^2}$. Thus, the chain inequality is the *only* controlling element, and thus there is no duality gap so long as $\|\mathbf{T}\|_F^2 > \frac{|\kappa|^2}{P_o}$.

In the side-looking STAP case, the target matrix has a Kronecker structure – that is, $\mathbf{T} = \mathbf{v}_t \otimes \mathbf{I}_N \otimes \mathbf{a}_t$. This means that $\mathbf{T}^H\mathbf{T} = \|\mathbf{v}_t\|^2\|\mathbf{a}_t\|^2\mathbf{I}_N$. If we assume that the spatial response comes from a uniform linear array and that the Doppler response is similarly uniform, then $\|\mathbf{a}_t\|^2 = M$ and $\|\mathbf{v}_t\|^2 = L$. This further implies that $\mathbf{T}^H\mathbf{T} = ML\mathbf{I}_N$ and $\|\mathbf{T}\|_F^2 = NML$. If we apply this knowledge to the positive definiteness condition, we need to find a $\mathbf{Z}_3$ such that $\mathbf{Z}_3 \succ \frac{|\kappa|^2}{N^2ML}\mathbf{I}_N$. This clearly exists if we set $\mathbf{Z}_3 = \alpha\mathbf{I}_N$ where $\alpha > \frac{|\kappa|^2}{NML}$. Following the path from the general case, this means that the Slater condition is satisfied so long as $NML > \frac{|\kappa|^2}{P_o}$.

C. Obtaining the KKTs

First, let us adopt the convention following conventions: each inequality constraint (multiplier) is given by $g_j(\cdot)$ (λ_j), each equality constraint (multiplier) is given by $h_i(\cdot)$ (μ_i), and an optimal value of a variable is denoted by a superscript o (*i.e.*, $\mathbf{B}^o$). The KKT conditions can then be written as

1) $\nabla_{\mathbf{B}}\mathcal{L}(\mathbf{B}^o, \boldsymbol{\mu}^o, \boldsymbol{\lambda}^o) = \mathbf{0}_{J \times J}$

---

[1]We can make this guess because $NML > N$. If, for whatever reason, the number of transmit resources were greater than the number of receive resources, then we could start with $\mathbf{Z}_3$ and continue from there.

[2]Again, if our available resources are swapped, we can use the outer product instead and get a similar answer.

2) $\lambda_j^o \geq 0 \; \forall j$

3) $\lambda_j^o g_j(\mathbf{B}^o) = 0 \; \forall j$

4) $g_j(\mathbf{B}^o) \leq 0 \; \forall j$

5) $h_i(\mathbf{B}^o) = 0 \; \forall i$

6) $\mathcal{H}[\mathcal{L}(\mathbf{B}^o, \boldsymbol{\mu}^o, \boldsymbol{\lambda}^o)] \succeq 0$

That is, for a matrix $\mathbf{B}^o$ to be a regular minimizer of the related optimization problem, it is necessary that it satisfies these conditions. If the problem is convex, these are necessary and sufficient conditions for the optimal minimizer and associated Lagrange multipliers. For future notational simplicity, we will drop the superscript o until absolutely necessary (i.e., a final statement of the optimal solution). According to [17], the gradient in the first KKT condition is $\nabla_{\mathbf{B}} = \frac{\partial}{\partial B^*}$ or, in other words, $\mathcal{D}_{\mathbf{B}^*}\mathcal{L} = \text{vec}^T(\frac{\partial \mathcal{L}}{\partial B^*})$. Thus, this vectorized form can take the place of the gradient.

1) KKT Condition 1: Let $\boldsymbol{\Sigma}$ be the slackness variable associated with the PSD condition on $\mathbf{B}$, and $\boldsymbol{\sigma}$ be its vectorization. The Lagrangian is

$$\mathcal{L}(\mathbf{B}, \boldsymbol{\Sigma}, \mu, \lambda) = \boldsymbol{\beta}^H(\widetilde{\mathbf{C}}_V\boldsymbol{\beta} + \boldsymbol{\rho} - \tilde{\mu}^*\tilde{\boldsymbol{\tau}} + \lambda\boldsymbol{\psi}_S - \boldsymbol{\sigma}) + \tilde{\mu}^*\kappa - \lambda P_o. \tag{25}$$

Condition 1 is, after taking the derivatives,

$$(\widetilde{\mathbf{C}}_V + \mathbf{K}_{J,J}\widetilde{\mathbf{C}}_V^*\mathbf{K}_{J,J})\boldsymbol{\beta} + \boldsymbol{\rho} - \tilde{\mu}^*\tilde{\boldsymbol{\tau}} - \tilde{\mu}\tilde{\boldsymbol{\tau}}_H + \lambda\boldsymbol{\psi}_S - \boldsymbol{\sigma} = \mathbf{0}_{J\times 1}$$

$$(\widetilde{\mathbf{C}}_V + \mathbf{K}_{J,J}\widetilde{\mathbf{C}}_V^*\mathbf{K}_{J,J})\boldsymbol{\beta} = \boldsymbol{\sigma} - (\boldsymbol{\rho} + \tilde{\mu}^*\tilde{\boldsymbol{\tau}} + \tilde{\mu}\tilde{\boldsymbol{\tau}}_H + \lambda\boldsymbol{\psi}_S). \tag{26}$$

In general, the full set of solutions to the second form of Condition 1 is given by

$$\boldsymbol{\beta} = (\widetilde{\mathbf{C}}_V + \mathbf{K}_{J,J}\widetilde{\mathbf{C}}_V^*\mathbf{K}_{J,J})^\dagger(\boldsymbol{\sigma} - (\boldsymbol{\rho} + \tilde{\mu}^*\tilde{\boldsymbol{\tau}} + \tilde{\mu}\tilde{\boldsymbol{\tau}}_H + \lambda\boldsymbol{\psi}_S)$$

$$+ (\mathbf{I}_J - (\widetilde{\mathbf{C}}_V + \mathbf{K}_{J,J}\widetilde{\mathbf{C}}_V^*\mathbf{K}_{J,J})(\widetilde{\mathbf{C}}_V + \mathbf{K}_{J,J}\widetilde{\mathbf{C}}_V^*\mathbf{K}_{J,J})^\dagger)\mathbf{z} \tag{27}$$

where $\mathbf{z}$ is an arbitrary vector isomorphic to a Hermitian matrix $\mathbf{Z}$, partitioned identically to $\mathbf{B}$ and $\boldsymbol{\Sigma}$. This solution exists provided the following existence condition holds:

$$(\widetilde{\mathbf{C}}_V + \mathbf{K}_{J,J}\widetilde{\mathbf{C}}_V^*\mathbf{K}_{J,J})(\widetilde{\mathbf{C}}_V + \mathbf{K}_{J,J}\widetilde{\mathbf{C}}_V^*\mathbf{K}_{J,J})^\dagger(\boldsymbol{\sigma} - (\boldsymbol{\rho} + \tilde{\mu}^*\tilde{\boldsymbol{\tau}} + \tilde{\mu}\tilde{\boldsymbol{\tau}}_H + \lambda\boldsymbol{\psi}_S))$$

$$= \boldsymbol{\sigma} - (\boldsymbol{\rho} + \tilde{\mu}^*\tilde{\boldsymbol{\tau}} + \tilde{\mu}\tilde{\boldsymbol{\tau}}_H + \lambda\boldsymbol{\psi}_S) \tag{28}$$

This may seem daunting, but we can unpack parts of this quite easily. Recall that the vectors $\boldsymbol{\beta}, \boldsymbol{\sigma}$ can be partitioned into the sums:

$$\boldsymbol{\beta} = \mathbf{P}_{WW}^T\boldsymbol{\beta}_1 + \mathbf{P}_{SW}^T\boldsymbol{\beta}_2 + \mathbf{P}_{WS}^T\boldsymbol{\beta}_{2,H} + \mathbf{P}_{SS}^T\boldsymbol{\beta}_3$$

$$\boldsymbol{\sigma} = \mathbf{P}_{WW}^T\boldsymbol{\sigma}_1 + \mathbf{P}_{SW}^T\boldsymbol{\sigma}_2 + \mathbf{P}_{WS}^T\boldsymbol{\sigma}_{2,H} + \mathbf{P}_{SS}^T\boldsymbol{\sigma}_3$$

and that these partitions are disjoint. Due to the disjointness, a quick examination of the second form of Condition 1 gives us a few results:

$$\boldsymbol{\sigma}_1 - \mathbf{r}_{ni} = \mathbf{0}_{(NML)^2\times 1} \tag{29a}$$

$$\boldsymbol{\sigma}_3 - \lambda\,\text{vec}(\mathbf{I}_N) = \mathbf{0}_{N^2\times 1} \tag{29b}$$

$$\mathbf{C}_V\boldsymbol{\beta}_2 = \boldsymbol{\sigma}_2 + \tilde{\mu}^*\boldsymbol{\tau}. \tag{29c}$$

The first two equations result from the clutter matrix on the left hand side annihilating the $\mathbf{P}_{WW}^T$ and $\mathbf{P}_{SS}^T$ components of $\boldsymbol{\beta}$, and imply $\boldsymbol{\Sigma}_1 = \mathbf{R_{ni}}$ & $\boldsymbol{\Sigma}_3 = \lambda \mathbf{I}_N$. The third equation covers both the $\mathbf{P}_{SW}^T$ and $\mathbf{P}_{WS}^T$ components, because the $\mathbf{P}_{WS}^T$ component is a conjugate permutation of the $\mathbf{P}_{SW}^T$ component. The solution (if it exists) of that third component, generally, is

$$\boldsymbol{\beta}_2 = \mathbf{C}_V^\dagger(\boldsymbol{\sigma}_2 + \tilde{\mu}^*\boldsymbol{\tau}) + (\mathbf{I}_{N^2ML} - \mathbf{C}_V^\dagger\mathbf{C}_V)\mathbf{z}_2 \tag{30}$$

where $\mathbf{z}_2$ is the $\mathbf{P}_{SW}^T$ component of the arbitrary vector $\mathbf{z}$ above. The existence condition for this solution is, unsurprisingly,

$$\mathbf{C}_V\mathbf{C}_V^\dagger(\boldsymbol{\sigma}_2 + \tilde{\mu}^*\boldsymbol{\tau}) = \boldsymbol{\sigma}_2 + \tilde{\mu}^*\boldsymbol{\tau},$$

or, after rearrangement,

$$(\mathbf{I}_{N^2ML} - \mathbf{C}_V\mathbf{C}_V^\dagger)\boldsymbol{\sigma}_2 = -\tilde{\mu}^*(\mathbf{I}_{N^2ML} - \mathbf{C}_V\mathbf{C}_V^\dagger)\boldsymbol{\tau}, \tag{31}$$

With some algebraic fiddling, one can verify that Equations 29a, 29b, 30, and 31 are equivalent to most of the much uglier solution above. The remaining necessary mathematical spackle is setting $\mathbf{B}_1 = \mathbf{Z}_1$ and $\mathbf{B}_3 = \mathbf{Z}_3$, again due to the annihilating nature of the clutter matrix on the block-diagonal terms. This more or less says the diagonal blocks of the matrix are arbitrary, subject to satisfying the other KKT conditions.

We conclude this section by repeating the necessary and sufficient conditions for the first KKT condition to be satisfied, as found above:

$$\boldsymbol{\Sigma}_1 = \mathbf{R_{ni}} \qquad\qquad \boldsymbol{\Sigma}_3 = \lambda\mathbf{I}_N$$

$$\mathbf{B}_1 = \mathbf{Z}_1 \qquad\qquad \mathbf{B}_3 = \mathbf{Z}_3$$

$$\boldsymbol{\beta}_2 = \mathbf{C}_V^\dagger(\boldsymbol{\sigma}_2 + \tilde{\mu}^*\boldsymbol{\tau}) + \mathbf{P}_{\mathbf{C}_V}^\perp\mathbf{z}_2 \qquad\qquad \mathbf{P}_{\mathbf{C}_V}^\perp\boldsymbol{\sigma}_2 = -\tilde{\mu}^*\mathbf{P}_{\mathbf{C}_V}^\perp\boldsymbol{\tau}$$

where $\mathbf{P}_{\mathbf{C}_V}^\perp = \mathbf{I}_{N^2ML} - \mathbf{C}_V\mathbf{C}_V^\dagger = \mathbf{I}_{N^2ML} - \mathbf{C}_V^\dagger\mathbf{C}_V$ is the orthogonal projector onto the nullspace of $\mathbf{C}_V$.

2) KKT Conditions 2-4: The Inequality Constraints: With the gradient condition exhausted, we turn to the inequality constraints (power bound, positive-semidefiniteness of the solution) & their related conditions. For convenience, we shall attack these somewhat independently in separate subsections, though they will interact.

a) The power constraint: The KKT conditions related to the power constraint are as follows:

$$\lambda \geq 0 \qquad\qquad \text{(Scalar Positivity)}$$

$$\boldsymbol{\psi}_S^T\boldsymbol{\beta} - P_o \leq 0 \qquad\qquad \text{(Scalar Feasibility)}$$

$$\lambda(\boldsymbol{\psi}_S^T\boldsymbol{\beta} - P_o) = 0 \qquad\qquad \text{(Scalar comp. slackness)}$$

Rearranging with our partitioned variables, we get

$$\lambda \geq 0 \qquad\qquad \text{(Scalar Positivity)}$$

$$\text{Tr}(\mathbf{B}_3) - P_o \leq 0 \qquad\qquad \text{(Scalar Feasibility)}$$

$$\lambda(\text{Tr}(\mathbf{B}_3) - P_o) = 0. \qquad\qquad \text{(Scalar comp. slackness)}$$

A nice result of the slackness condition is $\lambda \operatorname{Tr}(\mathbf{B}_3) = \lambda P_o$, which we will use later. The other "result" is that $\lambda = 0$ when the solution does not reach the power bound, and $\lambda > 0$ when it does. This will inform interpretations of the matrix situation below.

b) Positive semidefiniteness of $\mathbf{B}$*:* The conditions for semidefiniteness of the relaxed beamformer-signal basis are slightly more complex, but reveal a significant amount of structure to the solution. First, the direct form of these conditions are

$$\mathbf{\Sigma} \succeq 0, \quad \mathbf{B} \succeq 0, \quad \mathbf{\Sigma}\mathbf{B} = \mathbf{0}_{J \times J}.$$

Of course, in this form, they are not especially useful. However, recall that from the first KKT condition, we know $\mathbf{\Sigma}$ to some extent:

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{R_{ni}} & \mathbf{\Sigma}_2 \\ \mathbf{\Sigma}_2^H & \lambda \mathbf{I}_N \end{bmatrix}. \tag{32}$$

Via Theorem 1, we have conditions for the positive-semidefiniteness of the basis matrix and its slackness variable. To wit, the basis matrix $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{B}_2^H & \mathbf{B}_3 \end{bmatrix}$ is PSD if and only if

$$\mathbf{B}_1 \succeq 0 \tag{BP1}$$

$$\mathbf{B}_2 = \mathbf{B}_1 \mathbf{B}_1^\dagger \mathbf{B}_2 \tag{BP2}$$

$$\mathbf{B}_3 - \mathbf{B}_2^H \mathbf{B}_1^\dagger \mathbf{B}_2 \succeq 0, \tag{BP3}$$

and the slackness matrix $\mathbf{\Sigma}$ is PSD if and only if

$$\lambda \mathbf{I}_N \succeq 0 \tag{ΣP1}$$

$$\mathbf{\Sigma}_2 = \lambda \lambda^\dagger \mathbf{\Sigma}_2 \tag{ΣP2}$$

$$\mathbf{R_{ni}} - \lambda^\dagger \mathbf{\Sigma}_2 \mathbf{\Sigma}_2 \succeq 0, \tag{ΣP3}$$

where $\lambda^\dagger$ is a scalar pseudoinverse, equaling λ^{-1} when $\lambda > 0$ and zero otherwise. Using Theorem 2, we can also say that Conditions ΣP1–ΣP3 are satisfied if a contraction $\mathbf{X} \in \mathbb{C}^{NML \times N}$ exists such that $\mathbf{\Sigma}_2 = \sqrt{\lambda} \mathbf{R_{ni}}^{1/2} \mathbf{X}$.

The complementary slackness condition for the matrix case reduces to 4 equalities, given below

$$\mathbf{B}_1 \mathbf{R_{ni}} = -\mathbf{B}_2 \mathbf{\Sigma}_2^H \tag{CS1}$$

$$\mathbf{B}_1 \mathbf{\Sigma}_2 = -\lambda \mathbf{B}_2 \tag{CS2}$$

$$\mathbf{B}_2^H \mathbf{R_{ni}} = -\mathbf{B}_3 \mathbf{\Sigma}_2^H \tag{CS3}$$

$$\mathbf{B}_2^H \mathbf{\Sigma}_2 = -\lambda \mathbf{B}_3 \tag{CS4}$$

We can use these conditions to find equivalent forms for λ. First, taking the trace of Condition CS4, we have

$$\lambda \operatorname{tr}(\mathbf{B}_3) = -\operatorname{tr}(\mathbf{B}_2^H \mathbf{\Sigma}_2).$$

But since $\lambda \operatorname{tr}(\mathbf{B}_3) = \lambda P_o$ from scalar complimentary slackness,

$$\lambda = -\frac{\operatorname{tr}(\mathbf{B}_2^H \boldsymbol{\Sigma}_2)}{P_o}.$$

Since λ is both real and nonnegative, this means that $\operatorname{tr}(\mathbf{B}_2^H \boldsymbol{\Sigma}_2)$ is real and nonpositive! We also have $\operatorname{tr}(\mathbf{B}_2^H \boldsymbol{\Sigma}_2) = \operatorname{tr}(\boldsymbol{\Sigma}_2^H \mathbf{B}_2)$, which we can apply to the trace of Condition CS1 to obtain another form of λ:

$$\lambda = \frac{\operatorname{tr}(\mathbf{B}_1 \mathbf{R_{ni}})}{P_o}.$$

We can also apply this logic to Equation (29c) to reveal an interesting consequence about the cost function. When vectorized, $\operatorname{tr}(\mathbf{B}_2^H \boldsymbol{\Sigma}_2) = \boldsymbol{\beta}_2^H \boldsymbol{\sigma}_2$. Then, we have

$$\boldsymbol{\beta}_2^H \boldsymbol{\sigma}_2 = \boldsymbol{\beta}_2^H \mathbf{C}_V \boldsymbol{\beta}_2 - \tilde{\mu}^* \boldsymbol{\beta}_2^H \boldsymbol{\tau}$$

Applying the equality constraint, this is equivalent to

$$\boldsymbol{\beta}_2^H \boldsymbol{\sigma}_2 = \boldsymbol{\beta}_2^H \mathbf{C}_V \boldsymbol{\beta}_2 - \tilde{\mu}^* \kappa.$$

(Incidentally, this means that $\tilde{\mu}^* \kappa$ is real, and hence the optimal phase of $\tilde{\mu}$ is that of κ.) From the above, however, we can see that $-\boldsymbol{\beta}_2^H \boldsymbol{\sigma}_2 = \lambda P_o$, and so

$$\tilde{\mu}^* \kappa - \lambda P_o = \boldsymbol{\beta}_2^H \mathbf{C}_V \boldsymbol{\beta}_2$$

$$\tilde{\mu}^* \kappa = \boldsymbol{\beta}_2^H \mathbf{C}_V \boldsymbol{\beta}_2 + \lambda P_o.$$

But, $\lambda P_o = \operatorname{tr}(\mathbf{B}_1 \mathbf{R_{ni}})$ as well, and thus

$$\tilde{\mu}^* \kappa = \boldsymbol{\beta}_2^H \mathbf{C}_V \boldsymbol{\beta}_2 + \operatorname{tr}(\mathbf{B}_1 \mathbf{R_{ni}}).$$

The right hand side of the final equation is immediately recognizable as our objective function, which in some sense implies that this would be the only remaining aspect of the dual.

D. KKT Condition 5: Equality Constraints

This is the final major KKT condition left to examine, because KKT Condition 6 is trivially satisfied by $\mathbf{C}_V$ being positive semidefinite. Here, our primary concern is the equality constraint $\boldsymbol{\beta}_2^H \boldsymbol{\tau} = \kappa$. According to the first KKT condition, we know that

$$\boldsymbol{\beta}_2 = \mathbf{C}_V^\dagger (\boldsymbol{\sigma}_2 + \tilde{\mu}^* \boldsymbol{\tau}) + \mathbf{P}_{\mathbf{C}_V}^\perp \mathbf{z}_2. \tag{33}$$

Substituting this into the equality constraint gives us

$$\boldsymbol{\sigma}_2^H \mathbf{C}_V^\dagger \boldsymbol{\tau} + \tilde{\mu} \boldsymbol{\tau}^H \mathbf{C}_V^\dagger \boldsymbol{\tau} + \mathbf{z}_2^H \mathbf{P}_{\mathbf{C}_V}^\perp \boldsymbol{\tau} = \kappa. \tag{EquC}$$

Additionally, recall that, for this solution to exist, the following condition on $\boldsymbol{\sigma}_2$ must hold:

$$\mathbf{P}_{\mathbf{C}_V}^\perp \boldsymbol{\sigma}_2 = -\tilde{\mu}^* \mathbf{P}_{\mathbf{C}_V}^\perp \boldsymbol{\tau}. \tag{XC}$$

VI. CONSEQUENCES OF THE KKTS

The optimality conditions shown above are, at first blush, a complicated set of matrix equations to solve. However, we can derive some insight into the nature of the relaxed solution by attacking small pieces of it. In this section, we will first demonstrate generic properties of every solution to the KKTs, then show a potential power-bounded solution under certain conditions. Finally, we will provide a roadmap for non-power-bounded solutions and their feasibility.

A. General properties of the relaxed solution

First, we show that the solution does not reach the power bound P_o iff the slackness matrix $\boldsymbol{\Sigma}_2 = \mathbf{0}_{NML \times N}$.

Lemma 2. $\lambda = 0 \iff \boldsymbol{\Sigma}_2 = \mathbf{0}_{NML \times N}$

Proof: First, we proceed in the forward direction. If $\lambda = 0$, then the slackness matrix becomes

$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathbf{R_{ni}} & \boldsymbol{\Sigma}_2 \\ \boldsymbol{\Sigma}_2^H & \mathbf{0}_{N \times N} \end{bmatrix}.$$

To be part of a feasible solution, this must be positive semidefinite, which is only possible if $\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_2(\mathbf{0}_{N \times N})^{\dagger}(\mathbf{0}_{N \times N}) = \mathbf{0}_{NML \times N}$ (see Theorem 1). Hence, the forward direction is proved.

In the reverse direction, we prove through contradiction. Assume that $\boldsymbol{\Sigma}_2 = \mathbf{0}_{NML \times N}$ and $\lambda > 0$. The matrix slackness condition CS2 dictates that $\lambda \mathbf{B}_2 = -\mathbf{B}_1 \boldsymbol{\Sigma}_2$. Under our first assumption, this becomes $\lambda \mathbf{B}_2 = \mathbf{0}_{NML \times N}$, which simplifies to $\mathbf{B}_2 = \mathbf{0}_{NML \times N}$ under the second assumption. However, any feasible solution must also satisfy the Capon constraint $\operatorname{tr}(\mathbf{B}_2^H \mathbf{T}) = \kappa \neq 0$. Clearly, $\mathbf{B}_2 = \mathbf{0}_{NML \times N}$ violates this constraint, which leads to our contradiction and completes the proof. ∎

Next, we show that every feasible solution reaches the power bound if and only if the noise-and-interference correlation matrix $\mathbf{R_{ni}}$ is full rank. This is a considerably longer and more complex proof.

Proposition 1. $\lambda > 0 \iff \operatorname{rank}(\mathbf{R_{ni}}) = NML$

Proof of $\operatorname{rank}(\mathbf{R_{ni}}) = NML \Rightarrow \lambda > 0$*:* We will prove this by contradiction as well. Assume that $\mathbf{R_{ni}}$ is full rank and $\lambda = 0$. Given the second condition, Lemma 2 requires that $\boldsymbol{\Sigma}_2 = \mathbf{0}_{NML \times N}$. If we apply this to the slackness condition CS1, then $\mathbf{B}_1 \mathbf{R_{ni}} = \mathbf{0}_{NML \times NML}$. However, since $\mathbf{R_{ni}}$ is full rank, this implies $\mathbf{B}_1 = \mathbf{0}_{NML \times NML}$. This violates our non-triviality, but we will continue with the proof to show we reach a further contradiction. Since the overall solution matrix must be PSD, $\mathbf{B}_2 = \mathbf{0}_{NML \times N}$ as well. As in the proof of Lemma 2, we have reached a contradiction because the Capon constraint is violated, which completes the proof. ∎

Proof of $\lambda > 0 \Rightarrow \operatorname{rank}(\mathbf{R_{ni}}) = NML$*:* First, observe that for a non-trivial PSD solution matrix, $\mathcal{R}(\mathbf{B}_2^H) \subseteq \mathcal{R}(\mathbf{B}_3)$ by Theorem 2. Next, we turn to slackness condition CS4, which states $\lambda \mathbf{B}_3 = -\mathbf{B}_2^H \boldsymbol{\Sigma}_2$. If $\lambda > 0$, then clearly $\mathcal{R}(\mathbf{B}_3) \subseteq \mathcal{R}(\mathbf{B}_2^H)$. By the standard rules of subset inclusion, then, $\mathcal{R}(\mathbf{B}_2^H) = \mathcal{R}(\mathbf{B}_3)$ and $\operatorname{rank}(\mathbf{B}_2^H) = \operatorname{rank}(\mathbf{B}_3)$. This fact will become useful later.

We will now use the results of [23] on another slackness condition, CS3, and use a substitution of CS4 to get to our destination. Using [23, Theorem 2.2] on CS3 to solve for $\mathbf{R_{ni}}$, we have the following requirements for $\mathbf{R_{ni}}$ to be at least PSD (which it is):

1) $-\mathbf{B}_3 \mathbf{\Sigma}_2^H \mathbf{B}_2 \succeq 0$: If we substitute CS4 into this, we obtain $\lambda \mathbf{B}_3^2 \succeq 0$, which is satisfied because $\lambda > 0$ and $\mathbf{B}_3$ is PSD.

2) $\mathcal{R}(\mathbf{B}_3 \mathbf{\Sigma}_2^H) \subseteq \mathcal{R}(\mathbf{B}_2^H)$: This is satisfied because $\mathbf{B}_3 \mathbf{\Sigma}_2^H = -\frac{1}{\lambda} \mathbf{B}_2^H \mathbf{\Sigma}_2 \mathbf{\Sigma}_2^H$ via CS4, and the range inclusion follows directly.

3) $\mathrm{rank}(-\mathbf{B}_3 \mathbf{\Sigma}_2^H \mathbf{B}_2) = \mathrm{rank}(\mathbf{B}_3 \mathbf{\Sigma}_2^H)$: This is not immediately satisfied by the other conditions, but it does imply that $\mathrm{rank}(\mathbf{B}_3) = \mathrm{rank}(\mathbf{B}_3 \mathbf{\Sigma}_2^H)$.

The more interesting requirement comes from [24], which adds the following: $\mathbf{R_{ni}}$ is positive definite (and thus full rank) if and only if $\mathrm{rank}(-\mathbf{B}_3 \mathbf{\Sigma}_2^H \mathbf{B}_2) = \mathrm{rank}(\mathbf{B}_2^H)$. We know from the third PSD requirement above that $\mathrm{rank}(-\mathbf{B}_3 \mathbf{\Sigma}_2^H \mathbf{B}_2) = \mathrm{rank}(\mathbf{B}_3)$, and thus $\mathbf{R_{ni}}$ is full rank if and only if $\mathrm{rank}(\mathbf{B}_3) = \mathrm{rank}(\mathbf{B}_2^H)$. However, as seen from above, this condition is already satisfied if $\lambda > 0$, and so our proof is complete. ∎

Since the proof of Proposition 1 provides us with the evidence to show that power-bounded solutions occur only in non-singular noise-and-interference environments, we can also demonstrate an additional property of any power-bounded solution: namely, the rank of the overall solution matrix. Recall that for $\mathbf{B}$ to be PSD, $\mathcal{R}(\mathbf{B}_2) \subseteq \mathcal{R}(\mathbf{B}_1)$. However, slackness condition CS1 provides that $\mathbf{B}_1 \mathbf{R_{ni}} = -\mathbf{B}_2 \mathbf{\Sigma}_2^H$. Since $\mathbf{R_{ni}}$ is full rank, CS1 becomes $\mathbf{B}_1 = -\mathbf{B}_2 \mathbf{\Sigma}_2^H \mathbf{R_{ni}}^{-1}$, which implies $\mathcal{R}(\mathbf{B}_1) \subseteq \mathcal{R}(\mathbf{B}_2)$. Thus, $\mathcal{R}(\mathbf{B}_1) = \mathcal{R}(\mathbf{B}_2)$, and $\mathrm{rank}(\mathbf{B}_1) = \mathrm{rank}(\mathbf{B}_2) = \mathrm{rank}(\mathbf{B}_3) \leq N$, where the last equality is implied by Proposition 1's proof and the inequality is obvious.

In the sidelooking STAP case, the noise & interference covariance is *always* full-rank, so we conclude that most practical solutions will achieve the power bound. However, for numerical purposes, these are worthwhile observations, because computation might result in a rank-deficient estimate of $\mathbf{R_{ni}}$. We will explore this in more detail in Section VI-D below.

Lemma 3. *If $\lambda > 0$, $\tilde{\mu} \neq 0$.*

Proof: We prove by contradiction. Assume $\tilde{\mu} = 0$. When applied to (29c), we then have $\boldsymbol{\sigma}_2 = \mathbf{C}_V \boldsymbol{\beta}_2$. Premultiplying with $\boldsymbol{\beta}_2^H$, we have $\boldsymbol{\beta}_2^H \boldsymbol{\sigma}_2 = \boldsymbol{\beta}_2^H \mathbf{C}_V \boldsymbol{\beta}_2 \geq 0$ as the quadratic form of a PSD matrix. But we've already established that for $\lambda > 0$, $\boldsymbol{\beta}_2^H \boldsymbol{\sigma}_2 < 0$, which is a contradiction and our proof is complete. ∎

Lemma 4. *In a power-bounded solution,* $\mathrm{rank}(\mathbf{B}) = N - \mathrm{rank}(\lambda \mathbf{I}_N - \mathbf{\Sigma}_2^H \mathbf{R_{ni}}^{-1} \mathbf{\Sigma}_2)$.

Proof: Our above note on Proposition 1 is our starting point. Since the matrix product $\mathbf{B}\mathbf{\Sigma}$ is zero, so is its rank. This further implies $\mathrm{rank}(\mathbf{B}) = J - \mathrm{rank}(\mathbf{\Sigma})$. In general, since $\mathbf{\Sigma}$ is PSD, $\mathrm{rank}(\mathbf{\Sigma}) = \mathrm{rank}(\mathbf{R_{ni}}) + \mathrm{rank}(\lambda \mathbf{I}_N - \mathbf{\Sigma}_2^H \mathbf{R_{ni}}^\dagger \mathbf{\Sigma}_2)$. Therefore, generally, $\mathrm{rank}(\mathbf{B}) = (NML - \mathrm{rank}(\mathbf{R_{ni}})) + (N - \mathrm{rank}(\lambda \mathbf{I}_N - \mathbf{\Sigma}_2^H \mathbf{R_{ni}}^\dagger \mathbf{\Sigma}_2))$. In a power bounded solution, $\mathrm{rank}(\mathbf{R_{ni}}) = NML$ and the inverse exists, thus $\mathrm{rank}(\mathbf{B}) = N - \mathrm{rank}(\lambda \mathbf{I}_N - \mathbf{\Sigma}_2^H \mathbf{R_{ni}}^{-1} \mathbf{\Sigma}_2)$. ∎

B. A connection with waterfilling

Using some of these general results from above, we can show that the optimal solution to the KKTs at the power bound satisfies equations that resemble the well-known "waterfilling" concept.

First, we begin with a decomposition of the clutter matrix $\mathbf{C}_V$. Recall that the rank of the clutter "subspace" (i.e., the rank of $\mathbf{R}_c(\mathbf{s})$ and thus $\mathbf{C}_V$)is limited by both the physical extent and nature of non-target scatterers and our overall ability to observe this state of nature. For side-looking airborne arrays, the well-known Brennan rule [4], [25] is a reasonable approximation if certain conditions hold, but as [26] showed, a more robust result is obtained by applying the Landau-Pollak theorem. In any case, let us assume that the "true" rank of the clutter is $Q_{\text{eff}} \leq Q \leq N^2 ML$ (the subscript denoting the *effective* number of clutter patches). Then, we can use the economy eigendecomposition to find two equivalent representations of $\mathbf{C}_V$, namely:

$$\mathbf{C}_V = \breve{\mathbf{U}}_C \mathbf{D}_C \breve{\mathbf{U}}_C^H = \sum_{i=1}^{Q_{\text{eff}}} \nu_i \breve{\mathbf{u}}_i \breve{\mathbf{u}}_i^H.$$

Here, $\breve{\mathbf{U}}_C \in \mathbb{C}^{N^2 ML \times Q_{\text{eff}}}$ is the matrix that forms the basis for the Q_{eff}-dimensional vectorized clutter subspace, whose ith column is the eigenvector $\breve{\mathbf{u}}_i \in \mathbb{C}^{N^2 ML}$, $i \in \{1, \ldots, Q_{\text{eff}}\}$. $\mathbf{D}_C \in \mathbb{C}^{Q_{\text{eff}} \times Q_{\text{eff}}}$ is a diagonal matrix whose (i,i)th element is the nonzero eigenvalue ν_i. We can also consider the full eigendecomposition $\mathbf{C}_V = \breve{\mathbf{U}} \mathbf{D} \breve{\mathbf{U}}$. Here, the unitary matrix $\breve{\mathbf{U}} = [\breve{\mathbf{U}}_C \; \breve{\mathbf{U}}_N]$, where $\breve{\mathbf{U}}_C$ is as above and $\breve{\mathbf{U}}_N$ collects the $N^2 ML - Q_{\text{eff}}$ eigenvectors in the nullspace of $\mathbf{C}_V$, which we can regard as the vectors $\breve{\mathbf{u}}_i, i \in \{Q_{\text{eff}} + 1, \ldots, N^2 ML\}$. $\mathbf{D}$ is just the direct sum of $\mathbf{D}_C$ and a $N^2 ML - Q_{\text{eff}} \times N^2 ML - Q_{\text{eff}}$ all-zeros matrix. This decomposition also provides us with an alternative representation of the signal-dependent clutter covariance matrix $\mathbf{R}_c(\mathbf{s})$. Let us assume that there is a set of $N^2 ML$ matrices $\mathbf{U}_i \in \mathbb{C}^{NML \times N}$ whose vectorizations are $\text{vec}(\mathbf{U}_i) = \breve{\mathbf{u}}_i$ – that is, they correspond to the eigenvectors of $\mathbf{C}_V$. Then, the signal-dependent clutter covariance can also be given by

$$\mathbf{R}_c(\mathbf{s}) = \sum_{i=1}^{Q_{\text{eff}}} \nu_i \mathbf{U}_i \mathbf{s}\mathbf{s}^H \mathbf{U}_i^H. \tag{34}$$

We note that $\mathbf{U}_i^H \mathbf{U}_i = \frac{1}{N} \mathbf{I}_N$ for all i, not just those in the clutter indices. Thus, $\sqrt{N} \mathbf{U}_i$ is a rank-N partial isometry.

Now, given this formulation, we can show a waterfilling-like effect by combining the matrix slackness condition (CS3) and one of the Lagrangian conditions. Recall that the Lagrangian requires $\boldsymbol{\sigma}_2 = \mathbf{C}_V \boldsymbol{\beta}_2 - \tilde{\mu}^* \boldsymbol{\tau}$. In matrix form, and using the set of partial isometries, this expands to

$$\boldsymbol{\Sigma}_2 = \sum_{i=1}^{Q_{\text{eff}}} \nu_i \text{tr}(\mathbf{U}_i^H \mathbf{B}_2) \mathbf{U}_i - \tilde{\mu}^* \mathbf{T}. \tag{35}$$

Plugging this into (CS3) gives us

$$\mathbf{R}_{\text{ni}} \mathbf{B}_2 = -\boldsymbol{\Sigma}_2 \mathbf{B}_3 \tag{36}$$

$$= \tilde{\mu}^* \mathbf{T} \mathbf{B}_3 - \sum_{i=1}^{Q_{\text{eff}}} \nu_i \text{tr}(\mathbf{U}_i^H \mathbf{B}_2) \mathbf{U}_i \mathbf{B}_3. \tag{37}$$

Let us assume that we are power-bounded and everything that implies from the lemmas above. Thus, we can "directly" find $\mathbf{B}_2$ by applying $\mathbf{R}_{\mathbf{ni}}^{-1}$ above to obtain:

$$\mathbf{B}_2 = \tilde{\mu}^* \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3 - \sum_{i=1}^{Q_{\text{eff}}} \nu_i \, \text{tr}(\mathbf{U}_i^H \mathbf{B}_2) \, \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_i \mathbf{B}_3 \tag{38}$$

From here, we make a judicious guess. Premultiplying both sides with another normalized channel matrix $\mathbf{U}_j^H$ and taking the trace gives us

$$\text{tr}(\mathbf{U}_j^H \mathbf{B}_2) = \tilde{\mu}^* \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3) - \sum_{i=1}^{Q_{\text{eff}}} \nu_i \, \text{tr}(\mathbf{U}_i^H \mathbf{B}_2) \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_i \mathbf{B}_3). \tag{39}$$

We can extract the jth term from the sum and collect it on the left hand side to obtain

$$\text{tr}(\mathbf{U}_j^H \mathbf{B}_2)(1 + \nu_j \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_j \mathbf{B}_3))$$
$$= \tilde{\mu}^* \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3) - \sum_{\substack{i=1 \\ i \neq j}}^{Q_{\text{eff}}} \nu_i \, \text{tr}(\mathbf{U}_i^H \mathbf{B}_2) \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_i \mathbf{B}_3). \tag{40}$$

Now, the second term on the left-hand side of the above equation could be divided out as long as we had a guarantee it was always positive. First, if $\nu_j = 0$, this term is 1, which is positive. Otherwise, $\nu_j > 0$ since they are the non-zero eigenvalues of a positive semidefinite matrix. Next, we need examine the trace statement. $\mathbf{B}_3$ is positive semidefinite by construction, and $\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_j$ is positive definite because each $\mathbf{U}_j$ is a full rank scaled partial isometry and $\mathbf{R}_{\mathbf{ni}}$ is positive definite (see [20, Theorem 7.7.2]). Thus, the trace of this matrix product is always positive and real. With this in hand, we can now say

$$\text{tr}(\mathbf{U}_j^H \mathbf{B}_2) = \tilde{\mu}^* \frac{\text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3)}{1 + \nu_j \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_j \mathbf{B}_3)} - \sum_{\substack{i=1 \\ i \neq j}}^{Q_{\text{eff}}} \frac{\nu_i \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_i \mathbf{B}_3)}{1 + \nu_j \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_j \mathbf{B}_3)} \, \text{tr}(\mathbf{U}_i^H \mathbf{B}_2). \tag{41}$$

Since $\mathbf{B}_3$ is a relaxed version of $\mathbf{s}\mathbf{s}^H$, we can regard the expression $\nu_j \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_j \mathbf{B}_3)$ to be a relaxed form of $\nu_j \, \mathbf{s}^H \mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_j \mathbf{s}$, which is (effectively) a clutter-to-noise-and-interference ratio for the jth basis matrix. Similarly, $\text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3)$ is a joint target-and-clutter to noise-and-interference ratio, and the cross term $\nu_i \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_i \mathbf{B}_3)$ captures the ratio of patch-to-patch interaction and the noise-and-interference level. Thus, the first term on the right hand side is effectively a normalized measure of the clutter-matched target spectrum given a signal basis $\mathbf{B}_3$, and the right hand side is a normalized measure of the intraclutter spectrum given that same basis.

Traditional waterfilling dictates that power is preferentially injected to bands where the overall signal-to-noise ratio is high, proceeding to "worse"-off bands until the available power is exhausted. This is somewhat *inverted* in (41) because the left-hand side is an unscaled version of the subspace alignment between the solution $\mathbf{B}_2$ and the jth canonical clutter transfer matrix $\mathbf{U}_j$. Naively, we would like to *minimize* this alignment for $j \in \{1, \ldots, Q_{\text{eff}}\}$, since aligning with the clutter would nominally degrade our matching of the target. However, the coupled nature of (41) requires more nuance than that. First, observe that for the non-clutter directions, i.e., $j \in \{Q_{\text{eff}} + 1, ..., N^2 ML\}$, $\nu_j = 0$ and (41) becomes

$$\text{tr}(\mathbf{U}_j^H \mathbf{B}_2) = \tilde{\mu}^* \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3) - \sum_{i=1}^{Q_{\text{eff}}} \nu_i \, \text{tr}(\mathbf{U}_j^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_i \mathbf{B}_3) \, \text{tr}(\mathbf{U}_i^H \mathbf{B}_2). \tag{42}$$

This means that the solution's alignment in the non-clutter directions follows that of the whitened target's, less the combined crossover between non-clutter and clutter in the whitened spectrum. If the target is strong in these directions, we are all set, because the solution should align towards them. Otherwise, the solution must match the target in-spectrum as near as possible in directions both where the whitened clutter power is the lowest and the alignment with other directions is minimized. This corresponds to the findings in [27], who showed a similar behavior in two-step mutual information waveform design over consecutive transmit epochs. In this case, the "two steps" can be regarded as the trans-receive pair instead of sequential temporal designs.

But what of the value $\tilde{\mu}$? This clearly relates to the total available resources – in this case, κ and P_o – and provides us with the "water" in waterfilling. We can find a form of $\tilde{\mu}$ by reexamining (38) as follows. If we premultiply by the target $\mathbf{T}^H$, we obtain

$$\mathbf{T}^H \mathbf{B}_2 = \tilde{\mu}^* \mathbf{T}^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3 - \sum_{i=1}^{Q_{\text{eff}}} \nu_i \, \text{tr}(\mathbf{U}_i^H \mathbf{B}_2) \, \mathbf{T}^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_i \mathbf{B}_3 \tag{43}$$

Taking the trace of both sides, and recognizing that a feasible solution satisfies $\text{tr}(\mathbf{T}^H \mathbf{B}_2) = \kappa^*$, we now have

$$\kappa^* = \tilde{\mu}^* \, \text{tr}(\mathbf{T}^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3) - \sum_{i=1}^{Q_{\text{eff}}} \nu_i \, \text{tr}(\mathbf{U}_i^H \mathbf{B}_2) \, \text{tr}(\mathbf{T}^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_i \mathbf{B}_3). \tag{44}$$

This is where the traditional waterfilling appears, since we are saying that the gain across the target (which we know in this case to be κ^*) is the upper bound on the available resources (given by the first term) minus the overall impact of the clutter weighted by its alignment (the second term). Continuing on, we can rearrange this form to be

$$\tilde{\mu}^* \, \text{tr}(\mathbf{T}^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3) = \kappa^* + \sum_{i=1}^{Q_{\text{eff}}} \nu_i \, \text{tr}(\mathbf{U}_i^H \mathbf{B}_2) \, \text{tr}(\mathbf{T}^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_i \mathbf{B}_3). \tag{45}$$

Since according to the above argument, $\text{tr}(\mathbf{T}^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3)$ is never zero, $\tilde{\mu}^*$ is therefore:

$$\tilde{\mu}^* = \frac{\kappa^* + \sum_{i=1}^{Q_{\text{eff}}} \nu_i \, \text{tr}(\mathbf{U}_i^H \mathbf{B}_2) \, \text{tr}(\mathbf{T}^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{U}_i \mathbf{B}_3)}{\text{tr}(\mathbf{T}^H \mathbf{R}_{\mathbf{ni}}^{-1} \mathbf{T} \mathbf{B}_3)} \tag{46}$$

We can insert this into (41), collect terms, and solve for a matrix equation that we will leave for later study. Thus, the optimal solution to the relaxed problem describes a generalized whiten-and-match trans-receive filter process that exhibits waterfilling behavior, shaping the transmit process to match the target's response in clutter as much as possible.

C. A Potential Power-Bounded Solution

If $\mathbf{C}_V$ is less than full rank (which, in all practical scenarios, is always true), there is a potential path to a solution. First, assume that $\mathbf{R}_{\mathbf{ni}} \succ 0$, so $\lambda > 0$. Let us propose that the optimal vectorized corner matrix $\boldsymbol{\beta}_2$ is completely orthogonal to the clutter, i.e. $\mathbf{C}_V \boldsymbol{\beta}_2 = \mathbf{0}_{N^2 ML \times 1}$. A natural requirement for this to be a feasible solution is that $\boldsymbol{\tau}$ is not completely embedded in the clutter spectrum, i.e. $\boldsymbol{\tau} \notin \mathcal{R}(\mathbf{C}_V)$. First, this means that $\boldsymbol{\sigma}_2 = -\tilde{\mu}^* \boldsymbol{\tau}$, or $\boldsymbol{\Sigma}_2 = -\tilde{\mu}^* \mathbf{T}$. Since we know that $\boldsymbol{\beta}_2^H \mathbf{C}_V \boldsymbol{\beta}_2 = \tilde{\mu}^* \kappa - \lambda P_o$, a nullspace solution implies $\lambda = \frac{\tilde{\mu}^* \kappa}{P_o}$. Solving directly for $\tilde{\mu}$, we have $\tilde{\mu} = \frac{\text{tr}(\mathbf{B}_1 \mathbf{R}_{\mathbf{ni}})}{\kappa^*}$.

Applied to Condition CS2, we then obtain $\mathbf{B}_2 = \frac{\tilde{\mu}^*}{\lambda}\mathbf{B}_1\mathbf{T}$, which, given the substitution above implies $\mathbf{B}_2 = \frac{P_o}{\kappa}\mathbf{B}_1\mathbf{T}$. In order for this to be completely orthogonal to the clutter, $\mathbf{C}_V\operatorname{vec}(\mathbf{B}_1\mathbf{T}) = \mathbf{C}_V(\mathbf{T}^T \otimes \mathbf{I}_{NML})\operatorname{vec}(\mathbf{B}_1) = \mathbf{0}_{N^2ML\times 1}$.

Following on to Condition CS4, we also have $\mathbf{B}_3 = \frac{P_o^2}{|\kappa|^2}\mathbf{T}^H\mathbf{B}_1\mathbf{T}$. Since scalar feasibility requires that $\operatorname{tr}(\mathbf{B}_3) = P_o$, this implies that $\operatorname{tr}(\mathbf{T}^H\mathbf{B}_1\mathbf{T}) = \frac{|\kappa|^2}{P_o}$, which also satisfies the equality constraints. These results can also be shown to satisfy Conditions CS1 and CS3, as well as all of the positive semidefiniteness conditions, so long as $\operatorname{tr}(\mathbf{B}_1\mathbf{R}_{\mathbf{ni}}) \leq \frac{|\kappa|^2}{\|\mathbf{T}\|_F^2}\frac{\operatorname{tr}(\mathbf{R}_{\mathbf{ni}})}{P_o}$. If the problem satisfies Slater's condition, we also have, directly, $\operatorname{tr}(\mathbf{B}_1\mathbf{R}_{\mathbf{ni}}) < \operatorname{tr}(\mathbf{R}_{\mathbf{ni}})$, so the previous necessary condition is immediate. Hence, we have as a possible incomplete solution

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \frac{P_o}{\kappa}\mathbf{B}_1\mathbf{T} \\ (\frac{P_o}{\kappa}\mathbf{B}_1\mathbf{T})^H & \frac{P_o^2}{|\kappa|^2}\mathbf{T}^H\mathbf{B}_1\mathbf{T} \end{bmatrix} \tag{47}$$

where $\mathbf{B}_1$ minimizes $\operatorname{tr}(\mathbf{B}_1\mathbf{R}_{\mathbf{ni}})$ subject to the requirements

$$\mathbf{C}_V(\mathbf{T}^T \otimes \mathbf{I}_{NML})\operatorname{vec}(\mathbf{B}_1) = \mathbf{0}_{N^2ML\times 1}$$

$$\operatorname{tr}(\mathbf{T}\mathbf{T}^H\mathbf{B}_1) = \frac{|\kappa|^2}{P_o}$$

$$\mathbf{B}_1 \succeq 0.$$

The eagle-eyed reader will note correctly that this sequence of equations resembles another optimization problem. Indeed, we can reframe the implicit matrix completion in (47) as the solution to

$$\min_{\mathbf{B}_1} \quad \operatorname{tr}(\mathbf{B}_1\mathbf{R}_{\mathbf{ni}})$$

$$\text{s.t.} \quad \mathbf{C}_V(\mathbf{T}^T \otimes \mathbf{I}_{NML})\operatorname{vec}(\mathbf{B}_1) = \mathbf{0}_{N^2ML\times 1}$$

$$\operatorname{tr}(\mathbf{T}\mathbf{T}^H\mathbf{B}_1) = \frac{|\kappa|^2}{P_o}$$

$$\mathbf{B}_1 \succeq 0.$$

or, perhaps, after collapsing the nullspace requirement into its necessary constituent parts,

$$\min_{\mathbf{B}_1} \quad \operatorname{tr}(\mathbf{B}_1\mathbf{R}_{\mathbf{ni}})$$

$$\text{s.t.} \quad \operatorname{tr}(\mathbf{T}\mathbf{\Gamma}_q^H\mathbf{B}_1) = 0 \quad \forall q \in \{1,\ldots,Q\}$$

$$\operatorname{tr}(\mathbf{T}\mathbf{T}^H\mathbf{B}_1) = \frac{|\kappa|^2}{P_o}$$

$$\mathbf{B}_1 \succeq 0.$$

This is a standard semidefinite program so long as we use the Hermitian part of $\mathbf{T}\mathbf{\Gamma}_q^H$, which should be a semidefinite matrix, for all q. For now, we will not directly solve this subproblem, but we will note that some preliminary numerical analysis has shown that this is a reasonable path forward for the future.

D. Non-Power-Bounded Solutions

As mentioned above, there may be situations when $r_{\mathbf{NI}} = \operatorname{rank}(\mathbf{R}_{\mathbf{ni}}) < NML$ – for example, if we actually use an estimate $\hat{\mathbf{R}}_{\mathbf{ni}}$ of the noise-and-interference covariance, or if we consider a noise-free case for analysis. In

these cases, we know from Proposition 1 that $\lambda = 0$. This implies that the solution is not power bounded, and $\text{tr}(\mathbf{B}_3) < P_o$. Furthermore, due to Lemma 2, $\mathbf{\Sigma}_2 = \mathbf{0}_{NML \times N}$. With these in mind, we can produce, at the very least, a flowchart of solution properties in the non-power-bounded case.

As a simple beginning, we note that the matrix complementary slackness conditions reduce to two:

$$\mathbf{R_{ni}}\mathbf{B}_1 = \mathbf{0}_{NML \times NML} \tag{48}$$

$$\mathbf{R_{ni}}\mathbf{B}_2 = \mathbf{0}_{NML \times N} \tag{49}$$

This implies that if the solution is not power-bounded, it must null the entire noise-and-interference spectrum. Hence, we know that the matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ have the general form:

$$\mathbf{B}_1 = (\mathbf{I}_{NML} - \mathbf{R_{ni}}\mathbf{R_{ni}}^\dagger)\mathbf{V}_1(\mathbf{I}_{NML} - \mathbf{R_{ni}}\mathbf{R_{ni}}^\dagger) = \mathbf{P}^\perp_{\mathbf{R_{ni}}}\mathbf{V}_1\mathbf{P}^\perp_{\mathbf{R_{ni}}} \tag{50}$$

$$\mathbf{B}_2 = \mathbf{P}^\perp_{\mathbf{R_{ni}}}\mathbf{V}_2 \tag{51}$$

where $\mathbf{V}_1$ is an arbitrary $NML \times NML$ PSD matrix, $\mathbf{V}_2$ is an arbitrary $NML \times N$ matrix, and $\mathbf{P}^\perp_{\mathbf{R_{ni}}} = \mathbf{I}_{NML} - \mathbf{R_{ni}}\mathbf{R_{ni}}^\dagger$ is the orthogonal projection matrix onto the nullspace of $\mathbf{R_{ni}}$. Additionally, because of the first slackness condition, we know that $\text{rank}(\mathbf{B}_1) \leq NML - r_{\text{NI}}$.

This form of $\mathbf{B}_2$ has a feasibility consequence. Recall that a solution is feasible only if the Capon constraint $\text{tr}(\mathbf{B}_2^H\mathbf{T}) = \kappa \neq 0$ is satisfied. If we substitute our new form of $\mathbf{B}_2$ into the constraint, it becomes

$$\text{tr}(\mathbf{B}_2^H\mathbf{T}) = \text{tr}(\mathbf{V}_2^H\mathbf{P}^\perp_{\mathbf{R_{ni}}}\mathbf{T}) = \kappa. \tag{52}$$

Since $\mathbf{V}_2$ is arbitrary (but not trivial), feasibility is violated when $\mathbf{P}^\perp_{\mathbf{R_{ni}}}\mathbf{T} = \mathbf{0}_{NML \times N}$, which only occurs when $\mathcal{R}(\mathbf{T}) \subseteq \mathcal{R}(\mathbf{R_{ni}})$ (or, when vectorized, $\boldsymbol{\tau} \notin \mathcal{R}(\check{\mathbf{R}}_{\mathbf{ni}})$). Hence, there is no feasible solution if the target is embedded in the noise & interference spectrum. We state this directly in the following Lemma.

Lemma 5. *If $\lambda = 0$ and $\mathcal{R}(\mathbf{T}) \subseteq \mathcal{R}(\mathbf{R_{ni}})$, there is no feasible solution to (VSSDR) or its equivalent problems.*

Continuing on, our Lagrangian minimization becomes

$$\mathbf{C}_V\boldsymbol{\beta}_2 = \tilde{\mu}^*\boldsymbol{\tau} \tag{53}$$

or, given the form of $\mathbf{B}_2$ above,

$$\mathbf{C}_V\mathbf{P}^\perp_{\check{\mathbf{R}}_{\mathbf{ni}}}\mathbf{v}_2 = \tilde{\mu}^*\boldsymbol{\tau} \tag{54}$$

where $\mathbf{P}^\perp_{\check{\mathbf{R}}_{\mathbf{ni}}} = \mathbf{I}_N \otimes \mathbf{P}^\perp_{\mathbf{R_{ni}}}$ and $\mathbf{v}_2 = \text{vec}(\mathbf{V}_2)$.

Let us momentarily consider the generally unrealistic case when $\mathbf{C}_V$ is full rank (i.e. $Q_{eff} = N^2ML$). Under the $\lambda = 0$ hypothesis, we can directly solve for $\boldsymbol{\beta}_2$:

$$\boldsymbol{\beta}_2 = \tilde{\mu}^*\mathbf{C}_V^{-1}\boldsymbol{\tau}. \tag{55}$$

However, since $\boldsymbol{\beta}_2 \in \mathcal{N}(\check{\mathbf{R}}_{\mathbf{ni}})$, $\mathbf{P}_{\check{\mathbf{R}}_{\mathbf{ni}}}\boldsymbol{\beta}_2 = \mathbf{0}_{N^2ML \times 1}$, so we can also say that

$$\tilde{\mu}^*\mathbf{P}_{\check{\mathbf{R}}_{\mathbf{ni}}}\mathbf{C}_V^{-1}\boldsymbol{\tau} = \mathbf{0}_{N^2ML \times 1} \tag{56}$$

If we want this to be a feasible result (i.e. $\tilde{\mu} \neq 0$), then the whitened target $\mathbf{C}_V^{-1}\boldsymbol{\tau} \in \mathcal{N}(\check{\mathbf{R}}_{\mathbf{ni}})$ (and by extension, $\boldsymbol{\tau} \in \mathcal{N}(\check{\mathbf{R}}_{\mathbf{ni}})$). Hence, if the clutter is full rank, the target must be clear of the noise and interference (and the dimensionality of the available resources must be such that this is possible) for a feasible solution to exist. This is effectively a further restriction of Lemma 5.

Assuming $\mathbf{C}_V$ is less than full rank, we can find another condition on the target response for the Lagrange multipliers by premultiplying the above equation by the clutter nullspace projection matrix $\mathbf{P}_{\mathbf{C}_V}^{\perp}$, which becomes:

$$\tilde{\mu}^* \mathbf{P}_{\mathbf{C}_V}^{\perp} \boldsymbol{\tau} = \mathbf{0}_{N^2ML \times 1}$$

This means that either $\tilde{\mu} = 0$ or $\boldsymbol{\tau} \in \mathcal{R}(\mathbf{C}_V)$. Thus, if $\boldsymbol{\tau} \notin \mathcal{R}(\mathbf{C}_V)$ (that is, it has a non-zero component outside of the clutter), then $\tilde{\mu} = 0$.

The latter scenario, which we treat first, simplifies things considerably, because now:

$$\mathbf{C}_V \mathbf{P}_{\check{\mathbf{R}}_{\mathbf{ni}}}^{\perp} \mathbf{v}_2 = \mathbf{0}_{N^2ML \times 1} \tag{57}$$

Hence, either $\mathbf{v}_2$ (and thus $\boldsymbol{\beta}_2$) lies in $\mathcal{N}(\mathbf{C}_V \mathbf{P}_{\check{\mathbf{R}}_{\mathbf{ni}}}^{\perp})$ or $\mathbf{v}_2 = \boldsymbol{\beta}_2 = \mathbf{0}_{N^2ML \times 1}$. The second possibility violates the Capon constraint, therefore $\mathbf{v}_2$, $\boldsymbol{\beta}_2 \in \mathcal{N}(\mathbf{C}_V \mathbf{P}_{\check{\mathbf{R}}_{\mathbf{ni}}}^{\perp})$. However, if (for whatever reason), the target does not lie in this space, then we have reached another infeasiblity result via the Capon constraint.

If $\boldsymbol{\tau}$ *does* lie entirely within the clutter, then there are a few complex scenarios. First, consider a situation where $\mathcal{N}(\check{\mathbf{R}}_{\mathbf{ni}}) \subseteq \mathcal{N}(\mathbf{C}_V)$, which means the clutter is subsumed entirely into interference and noise (and thus $\mathcal{R}(\mathbf{C}_V) \subseteq \mathcal{R}(\check{\mathbf{R}}_{\mathbf{ni}})$ & $Q_{eff} \leq Nr_{\mathtt{NI}}$). In this case, $\mathbf{C}_V \mathbf{P}_{\check{\mathbf{R}}_{\mathbf{ni}}}^{\perp} = \mathbf{0}_{N^2ML \times N^2ML}$. However, this also means that the *target* is now also embedded in the noise and interference. By Lemma 5, there is no feasible solution! In any other case, a solution exists if and only if $\tilde{\mu}^*\boldsymbol{\tau} \in \mathcal{R}(\mathbf{C}_V \mathbf{P}_{\check{\mathbf{R}}_{\mathbf{ni}}}^{\perp})$.

VII. SIMULATIONS AND RESULTS

A. The Solvers

While we have preliminarily reduced solving (SSDR) to a matrix completion problem, it is possible to solve (in some sense) the problem numerically with commercial solvers. Most of the analysis presented here is enabled by the modeling package CVX [28], [29], which permitted us to construct two equivalent representations of the problem with rather different results. In all cases, we considered only the solver *package* SDPT3 [30], [31].

The major difference occurs in how the quadratic form in the problem is presented to CVX. The first method (hereinafter the "QuadSolver") uses the `quad_form()` function, which preserves the form $\text{vec}(\mathbf{B})^H \tilde{\mathbf{C}}_V \text{vec}(\mathbf{B})$. The second method (hereinafter the "NormSolver") uses a composition of the `pow_pos()` and `norm()` functions, which implements the equivalent form $\left\| \tilde{\mathbf{C}}_V^{1/2} \text{vec}(\mathbf{B}) \right\|^2$. Though these forms are symbolically equivalent, the latter is supposedly more amenable to optimization by a conic representation which is more efficiently processed by the default solvers available to CVX. We will demonstrate that while this might be true, the solvers behave very differently under certain conditions.

A note on dimensionality: Regardless of the solver particulars, it is important to note that all presented scenarios and solutions are necessarily constrained by available computing power and memory constraints. While CVX and MATLAB combine to make a powerful tool, they are limited in the size of problems that can be solved on a regular workstation. To wit, we have found that when $NML > 400$, our current workstation (AMD Athlon^{TM} II X2 B24 processor at 2.3GHz with 8GB memory) throws out-of-memory errors during CVX's setup phase. We anticipate performing this analysis on more robust systems in the future, which will allow us to extend it to more realistic radar scenarios.

B. Computational Analysis of the KKTs

In the following simulations, we used certain common parameters for ease of comparison. As noted above, computational limitations constrain our overall problem size; therefore, in this case, we assume $N = 5$ fast time samples, $M = 5$ array elements, and $L = 16$ pulses. The radar operates on a carrier frequency of 1 GHz and transmits pulses with a bandwidth of 50 MHz. The receive array has elements spaced at half a wavelength, i.e. $d = \lambda_o/2$. We recognize that these are not particularly realistic parameters (at least for the dimensionality), but as stated above, computational limitations currently conspire against a more representative simulation.

Unless otherwise stated, the scenarios presented to the solver were as follows: the noise covariance matrix was a scaled correlation matrix with correlation function $\exp(-0.05|n|)$ for $n \in \{1, \ldots, NML\}$. Interferers were placed at the azimuth-elevation pairs (0.3941, 0.3) radians and (-0.4951,0.3), with correlation matrices given by the Toeplitz matrix associated with the correlation function $\exp(0.2|l|)$ for $l \in \{1, \ldots, NL\}$. The clutter was simulated by placing $Q = 25$ patches of $P = 5$ scatterers each, equally spaced in azimuth over the interval $(-\pi/2, \pi/2)$ at an elevation angle of $\frac{\pi}{4}$ radians. Adjacent scatterers within each patch are correlated with coefficient -0.2. In all cases, we have made the Capon constraint real.

To obtain signals and beamformers from the relaxed problem, we generated the best "rank-1" approximation by summing the first B singular vectors of the solution matrix $\mathbf{B}$ weighted by their associated singular value. In the Frobenius sense, only the largest singular vector/value pair is the best approximation, but we have found that this approximation does, in fact, sketch out the nature of the unrelaxed problem. In the results below, we will often sweep the value B from 1 to J, capturing, in some sense, the entire space. We recognize, however, that this is merely a heuristic that in this case happens to work well.

1) Variation in κ: First, we examine varying the Capon constraint with a fixed power constraint. In these scenarios, κ ranges from 10^{-2} to $NML \times 10^4$ while $P_o = 10^7$. Before continuing, we note that Slater's condition dictates that the duality gap should be zero so long as $|\kappa|^2 < NMLP_o = 4 \times 10^9$, which means that our parameter sweep should lie within this region. For reasonable comparison, we assumed no interference in this scenario. Plotted scenarios exclude solver results where either the solver failed or the solution violated the KKTs significantly despite the solver claiming success. In practice, this means we excluded "solutions" where $\lambda < 0$.

We begin with the eigenspread of the output solution matrix $\mathbf{B}$ for each solver and each value of κ. Figure 1 shows the eigenspreads produced by Quad Solver, with Figure 2 showing finer detail of the first $2N$ eigenvalues. Clearly, all scenarios produce functionally low-rank solutions – the "true" rank never exceeds N. Indeed, the *effective*

numeric rank only exceeds one in the $\kappa = 1$ scenario. Since the power is constant, the peak eigenvalue is nearly same in all scenarios, but increasing κ drives the non-peak eigenvalues lower.
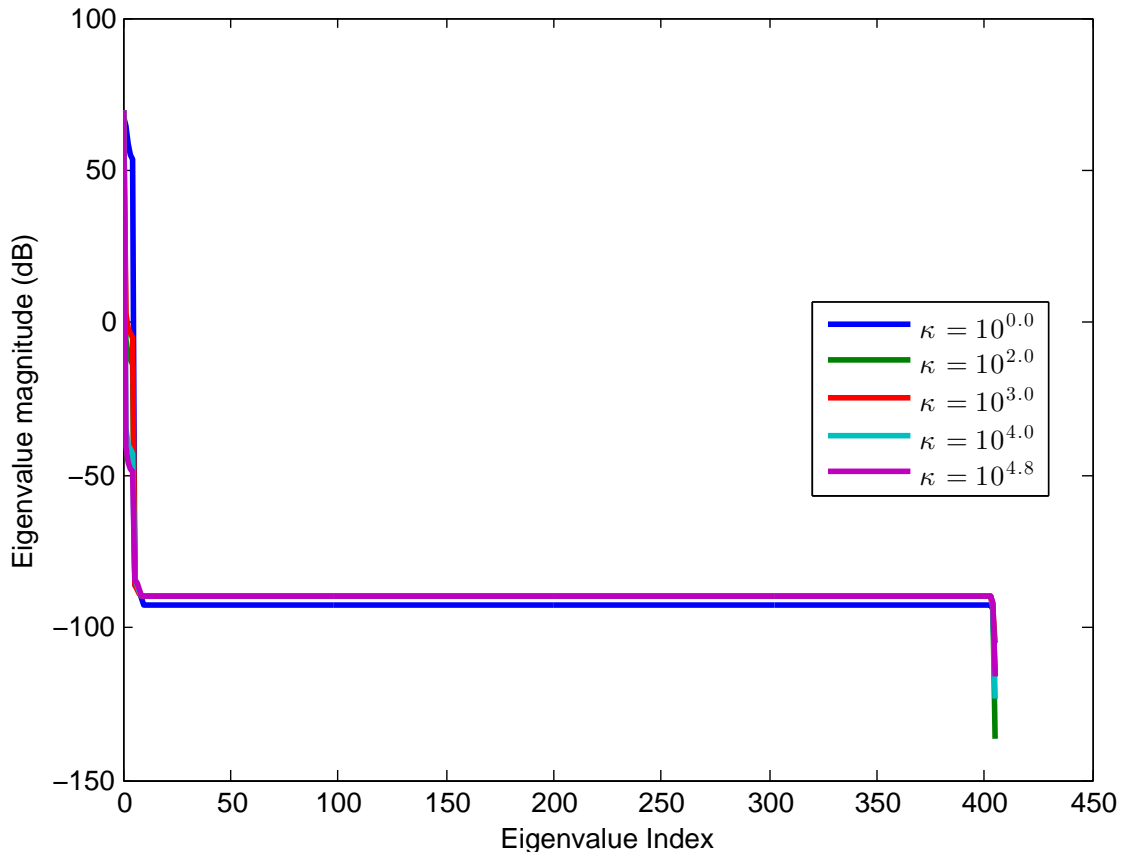
Fig. 1. Eigenvalue spread of $\mathbf{B}$ solution matrix using QuadSolver

This dependence on κ essentially repeats for the NormSolver-produced solutions, as can be seen in Figure 3 (the overall eigenspread) and Figure 4 (the first $2N$ eigenvalues). The primary difference here is that all valid solutions produced have an effective rank of one, and the overall dropoff from the Nth eigenvalue to the $N-1$th eigenvalue is not nearly as steep. This can potentially be attributed to less accumulated numerical error in NormSolver's process, leading to a cleaner solution.

As predicted by the KKTs, increasing κ increases the Lagrange multiplier λ in equal measures for a fixed power level. This effect is demonstrated directly in Figures 5 and 6. We see that the returned λ values appear to be similar for both solvers across the entire κ parameter space which indicates they converge to similar solutions, at least in terms of noise-and-interference suppression.

Next, we examine the effectiveness of our low-rank approximation by analyzing the cost function of (12)/(BQP 1). In the following figures, we use the term "basis length" to indicate how many of the J singular vectors of $\mathbf{B}_{opt}$ are combined to form our estimate of $\mathbf{b}$. That is, if $\sigma_{\mathbf{B},i}$ is the ith largest singular value of the solution matrix $\mathbf{B}$ and $\mathbf{u}_{\mathbf{B},i}$
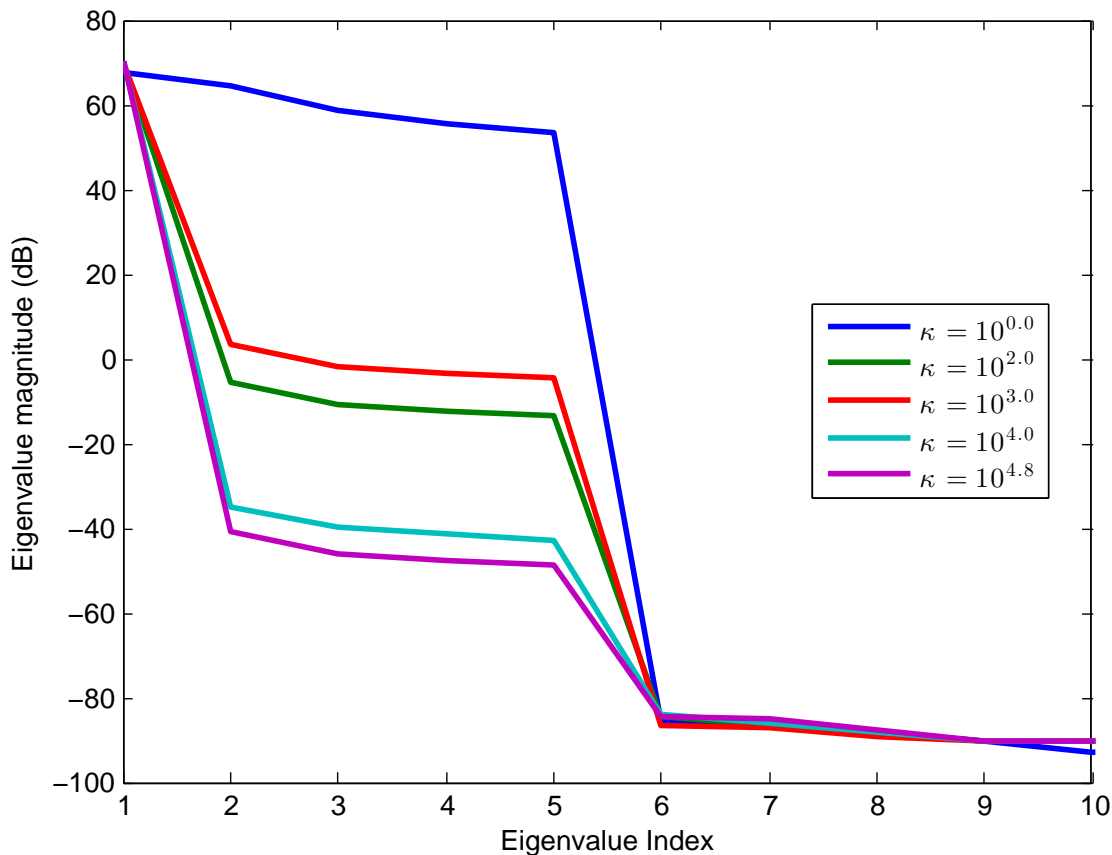
Fig. 2. Closeup of the first $2N$ eigenvals. of $\mathbf{B}$ using QuadSolver

is the associated singular vector, a basis length of B produces an approximate vector of $\mathbf{b}_{\mathrm{appx},B} = \sum_{i=1}^{B} \sigma_{\mathbf{B},i}\mathbf{u}_{\mathbf{B},i}$, with the associated beamformer $\mathbf{w}_{\mathrm{appx},B}$ and signal $\mathbf{s}_{\mathrm{appx},B}$ recovered from the stacked vector as in (13).

Figure 7 shows the overall sweep of basis length for the approximate solutions recovered from QuadSolver as a function of κ, with Figure 8 focusing only on the first $2N$ values. With the notable exception of $\kappa = 1$, as κ increases, the minimum original cost increases and the difference over the first N basis vectors becomes less pronounced. We can regard the result at $\kappa = 1$ to be a threshhold scenario where, since the effective numerical rank of the relaxed solution is closer to N, more of the basis is needed for better approximation.

In contrast, NormSolver produces solutions that generally perform poorly in the original cost function. As seen in Figures 9 and 10, only the $\kappa = 1$ scenario replicates the general behavior seen from QuadSolver, with the NormSolver-produced minimum value exceeding its QuadSolver counterpart by nearly 20 dB. For higher values of κ, the previously observed flattening effect is in full force, indicating that this solver produces identically structured solutions that merely scale with κ.

Finally, our investigation of the effect of κ concludes by examining the alignment between part of the relaxed solution and the clutter patches. Since we established in Proposition 1 that the submatrix $\mathbf{B}_2$'s left eigenspace spans
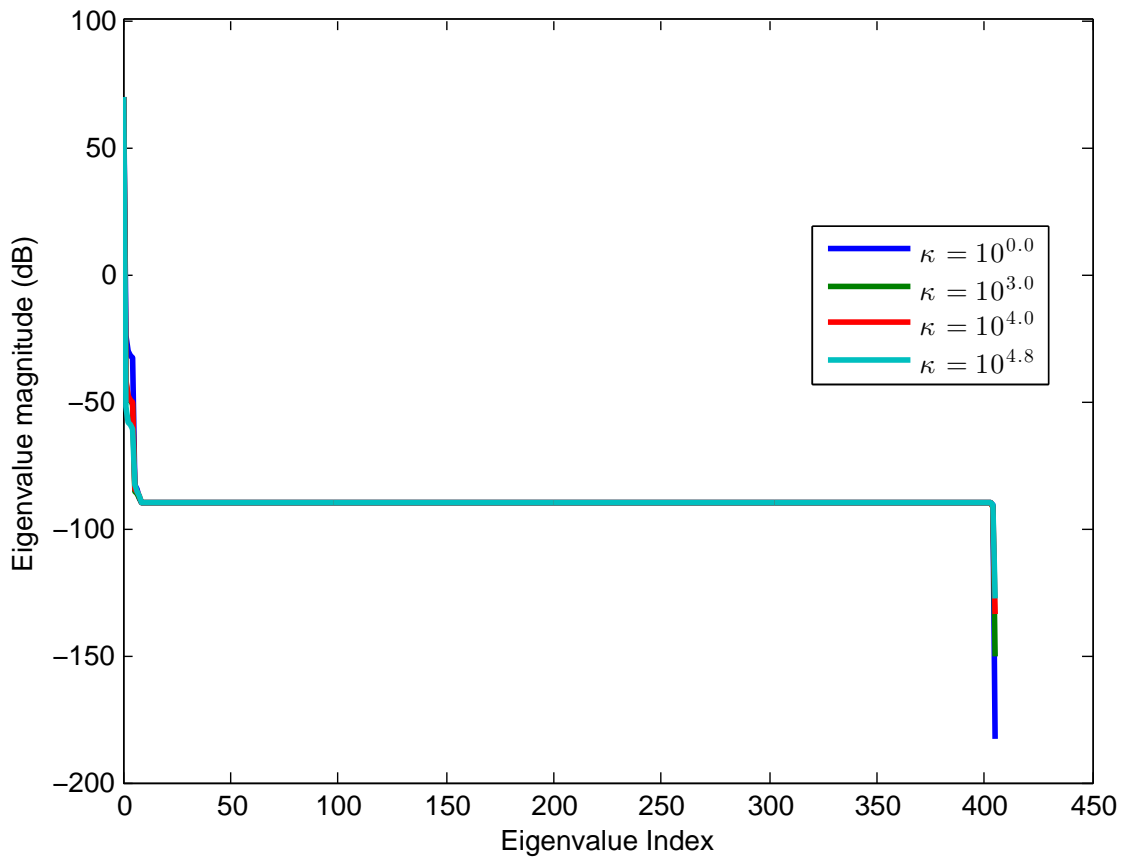
Fig. 3. Eigenvalue spread of $\mathbf{B}$ solution matrix using NormSolver

the columns of $\mathbf{B}_1$ and its right eigenspace spans those of $\mathbf{B}_3$, this effectively means that the subspace alignment between $\mathbf{B}_2$ and each clutter transfer matrix $\mathbf{\Gamma}_q$ is a proxy for the combined ability of the trans-recieve pair to suppress clutter. We can measure subspace alignment through the cosine of their *principal angles* as follows. For given matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{C}^{p \times q}$, the cosine of the principal angle $\psi_{\mathbf{X},\mathbf{Y}}$ between the subspaces $\mathrm{span}\{\mathbf{X}\}$ and $\mathrm{span}\{\mathbf{Y}\}$ is

$$\cos(\psi_{\mathbf{X},\mathbf{Y}}) = \frac{|\operatorname{tr}(\mathbf{X}^H \mathbf{Y})|}{\|\mathbf{X}\|_F \|\mathbf{Y}\|_F}. \tag{58}$$

The closer $\cos(\psi_{\mathbf{X},\mathbf{Y}})$ is to 1, the more aligned the subspaces. In Figures 11 and 12, the resulting subspace alignments for both target and clutter channel matrices are plotted for the solution from QuadSolver and NormSolver, respectively. For reference, we denote the existing alignment between the target channel matrix T and each clutter channel matrix. For both solvers, variation in κ seems to have little effect on the overall alignment. However, QuadSolver's solution for $\kappa = 1$, the solution's subspace is slightly better aligned with the target at the cost of marginally higher alignment to the clutter. In any case, we see a significant difference between the two solvers: QuadSolver attempts to avoid any alignment with the clutter, while NormSolver attempts to match the target's
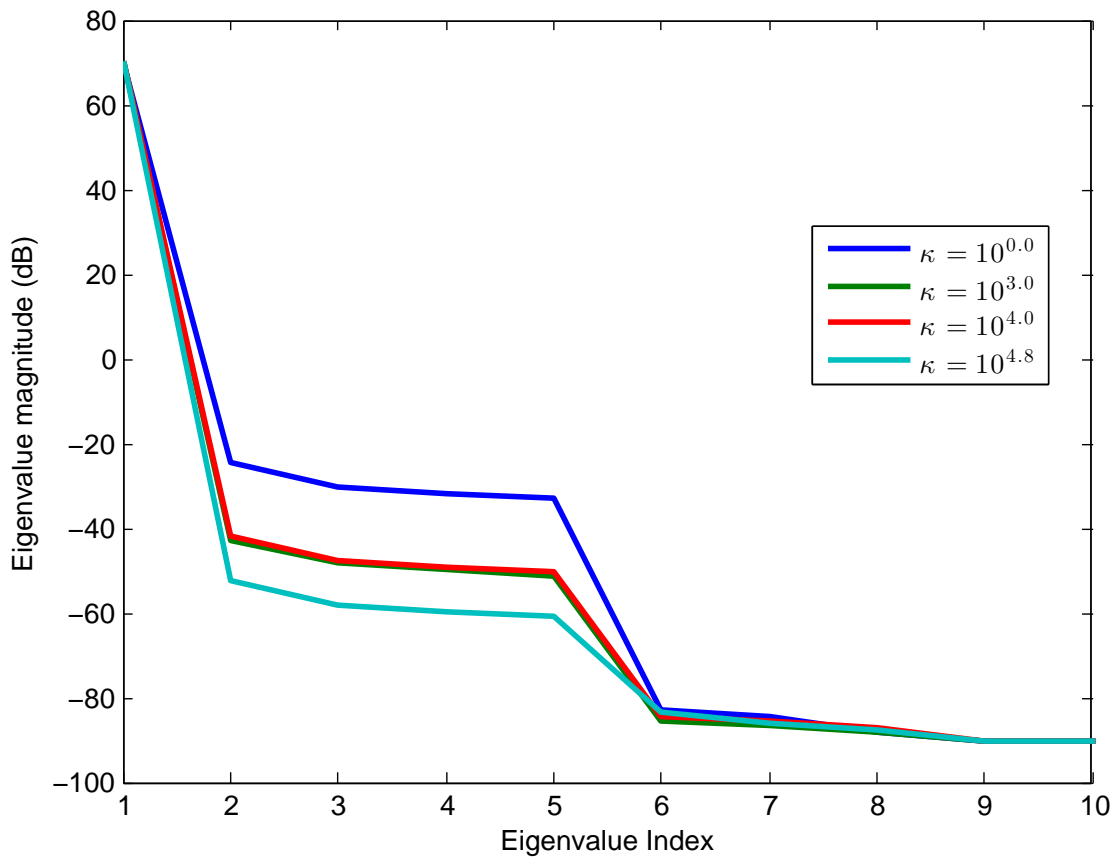
Fig. 4. Closeup of the first $2N$ eigenvals. of $\mathbf{B}$ using NormSolver

alignment spectrum (in shape, if not magnitude). This behavior in NormSolver appears to come at a cost to the overall solution-target alignment.

2) Variation in P_o: Next, we examine the scenario opposite the above: varying the power constraint while keeping the Capon constraint fixed. In these scenarios, P_o ranges from 10^3 to 10^9 while $\kappa = 100$. The Slater condition in all cases is easily satisfied; however, we do note that the solver failed to satisfy the KKTs for the final two values. This is generally attributable to scaling issues. For reasonable comparison, we assumed no interference in this scenario.

First, we return to the eigenspread of the solution matrix $\mathbf{B}$. Figures 13 & 14 show the total eigenspread and its first $2N$ eigenvalues for QuadSolver. As P_o increases, we see that the peak eigenvalue increases to maintain equality at the power bound; this also corresponds with an overall "lifting" of the eigenspectrum, which is most likely due to numerical precision limitations. In any case, the effective numerical rank of the solution is still one, while the overall effective rank of the solution is, at most, N.

Similarly, the total eigenspread and first $2N$ eigenvalues for the NormSolver solution are provided in Figures 15 and 16, respectively. We see similar behavior to the QuadSolver, but note the tighter spread in effectively non-zero
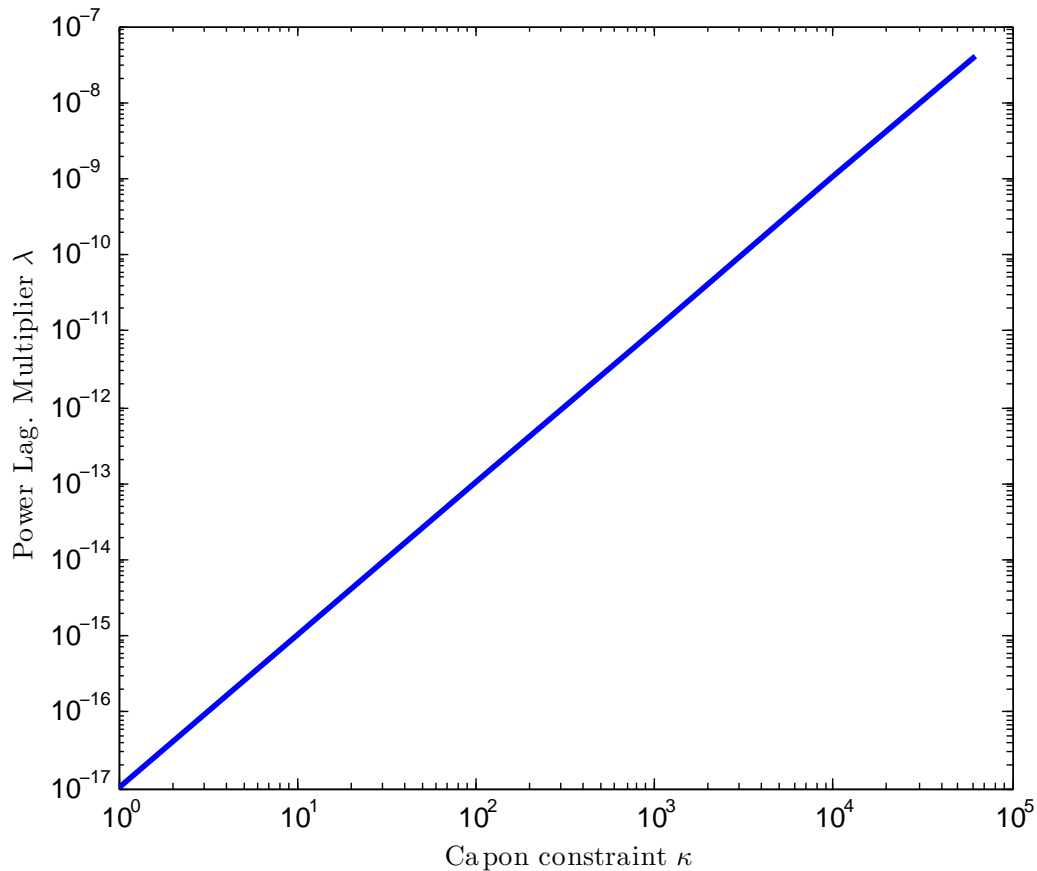
Fig. 5. Optimal λ as a function of the Capon constraint, QuadSolver

eigenvalues with increasing P_o. This indicates that NormSolver is converging to the same solution at each P_o, scaled to match the power constraint.

As above, we also explore the effect of the constraint on the Lagrange multiplier λ, shown in Figures **??** (for QuadSolver) and 18 (for NormSolver). In both cases, the multiplier sharply declines as the constraint becomes looser, indicating that the overall filtered noise power remains nearly the same and is mostly a structural issue.

We also reexamine the approximate rank-1 cost function described in the above section. For QuadSolver, we can see from Figures 19 (the entire sweep of J basis vectors) and 20 (the first $2N$ basis vectors) that increasing the power lowers the overall cost when approximating using the first N basis vectors. Outside these vectors, the cost dramatically increases for higher values of P_o. This indicates that low-rank approximation under weak power constraints is effectively unnecessary, while basis selection must be more careful in high-power scenarios.

In contrast, NormSolver's solution remains functionally fixed in cost over the sweep (as seen in Figures 21 and 22), which agrees with the behavior seen in 16 – namely, the overall solution is minimally sensitive to the power constraint. This is most obvious when $P_o = 10^6$, as even its "dramatic" increase is within a tenth of a decibel of its minimum.
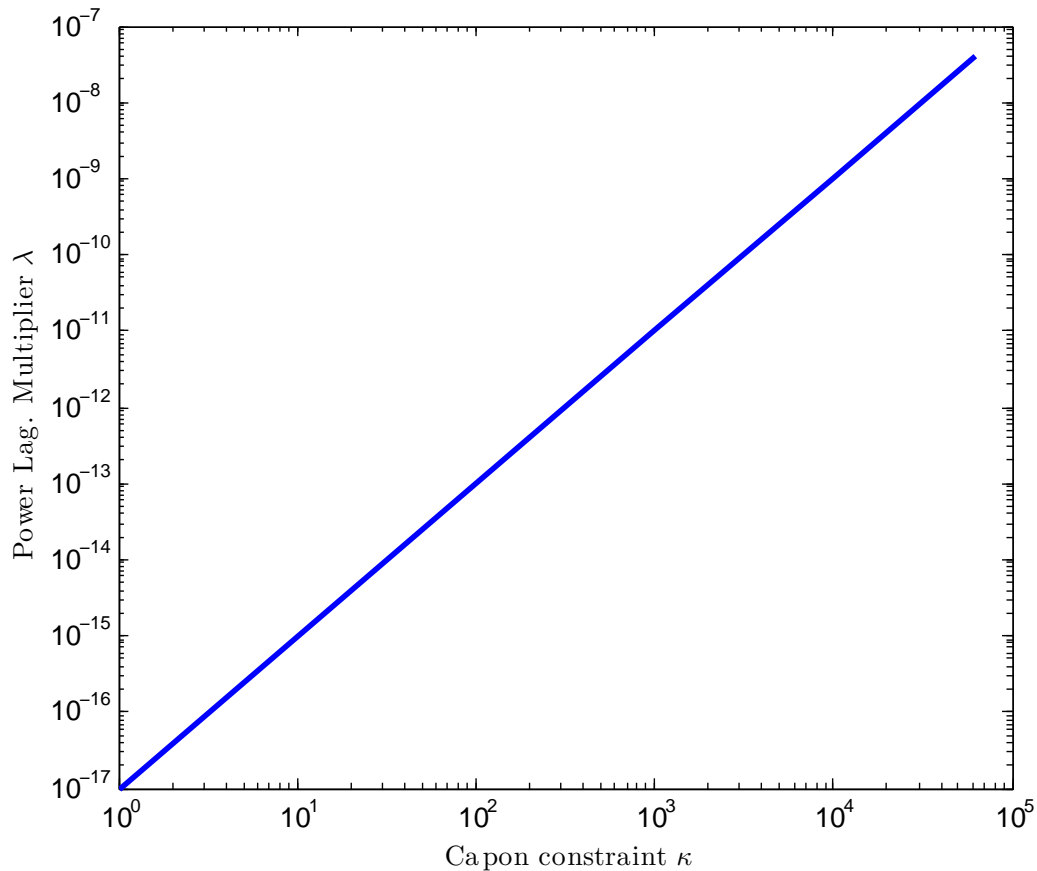
Fig. 6.   Optimal λ as a function of the Capon constraint, NormSolver

Finally, we investigate the subspace alignment as a function of the power constraint. As shown in Figure 23, QuadSolver's solution attempts to null the clutter spectrum entirely, with higher power corresponding to better nulling. That said, all solutions provide alignment figures less than 10^{-6}, which is well below the target's alignment and any reasonable interpretation of "aligned". Power does not seem to affect the solution-target alignment to a significant degree.

Again, in contrast, NormSolver's solution, seen in Figure 24, attempts to align with the target's representation in the clutter and remains insensitive to the power constraint. This provides a final confirmation of the waterfilling interpretation above.

C. Interference Effects

In this scenario, we consider the impact of interference on the relaxed solution and the effect each solver type has on this impact. Using the general scenario from above with both no interference and an interferer located at $(\theta_I, \phi_I) = (0.3941, 0.3)$ radians, we examine the impact on both subspace alignment and the overall adapted pattern.
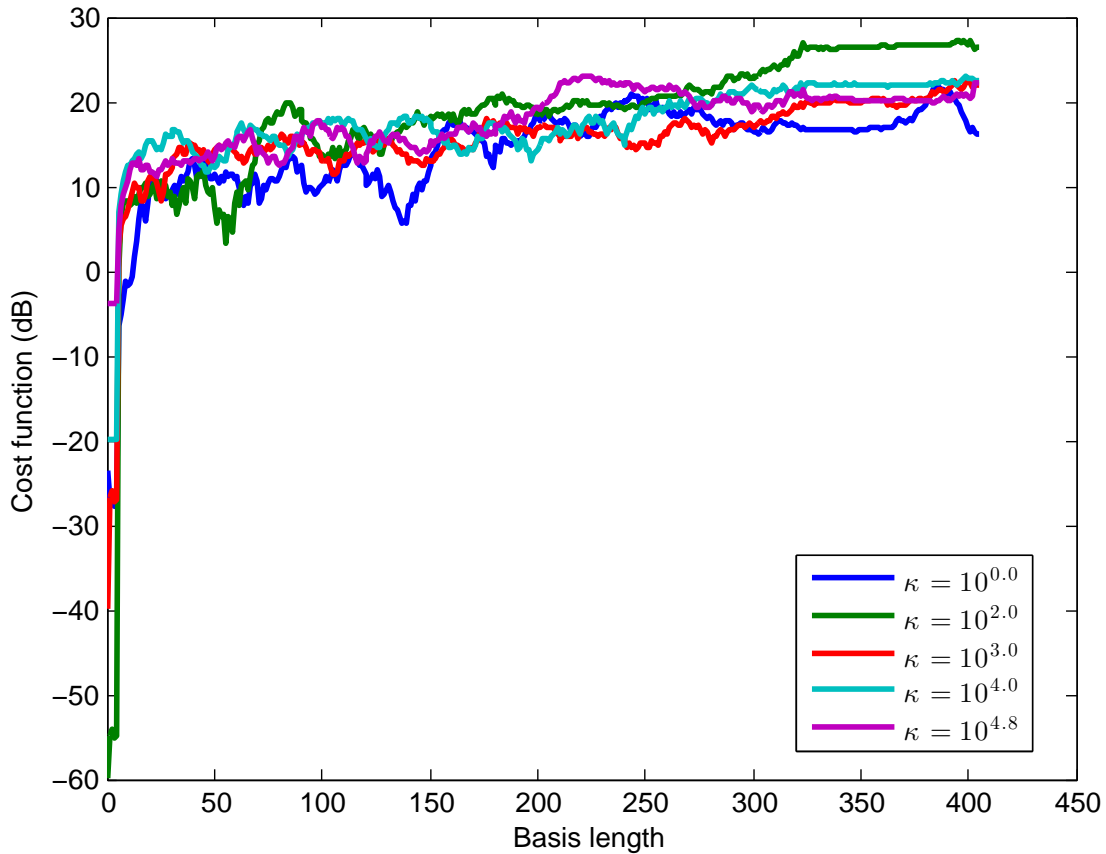
Fig. 7. Original cost as function of basis length, QuadSolver

First, we consider the subspace alignment as in Section VI. As above, the upper right corner of the solution matrix, $\mathbf{B}_2$, is compared against each clutter patch and the target. The subspace alignment cosine, $\cos(\psi)$, ranges from 0 to 1, with higher values indicating that the spans of the two matrices are more aligned. Figure 25 shows the alignment with no interference, while Figure 26 shows the alignment under the interference. In both cases, as in the previous sections, QuadSolver produces a solution that is effectively unaligned with the clutter, whereas NormSolver's solution attempts to follow the preexisting target-clutter alignment. The major difference under this metric is a clear broadening and shifting of the "mainlobe" of NormSolver's solution-clutter alignment spectrum near the direction of the interferer and beyond. QuadSolver's solution-clutter alignment appears generally unaffected, but we note that this does not mean the interferer has no impact, as we will soon see.

Next, we consider the traditional adapted pattern for STAP, which plots the following function

$$\mathcal{P}(f_d, \theta) = |\mathbf{w}_o^H(\mathbf{v}(f_d) \otimes \mathbf{s}_o \mathbf{a}(\theta, \phi)|^2. \tag{59}$$

That is, the adapted pattern for a given beamformer-signal pair $\mathbf{w}_o, \mathbf{s}_o$ is a function of the Doppler frequency f_d and the azimuth θ at a given elevation ϕ. We consider the same scenarios as above, with and without interference,
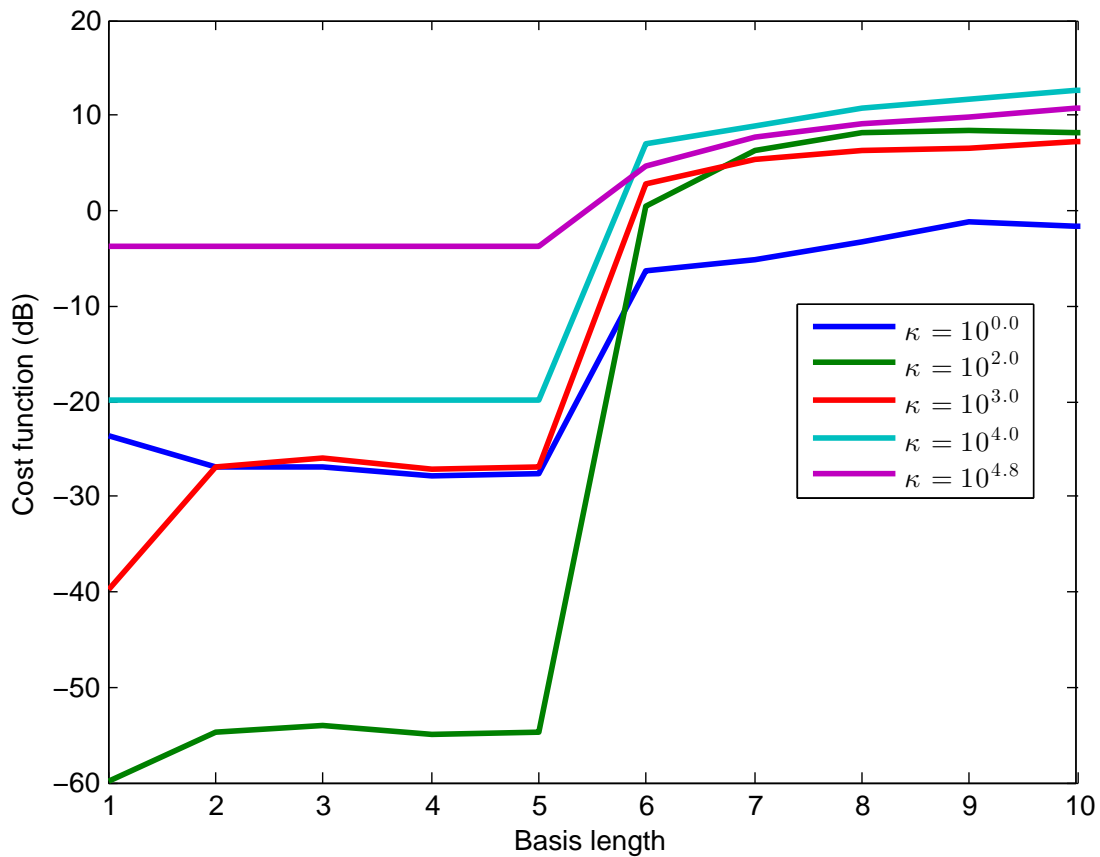
Fig. 8. Original cost as function of basis length, first $2N$, QuadSolver

and provide the overall adapted pattern as well as Doppler cuts at the target azimuth and azimuth cuts at the target Doppler frequency.

With no interference, the naive implementation performs relatively well, as illustrated in Figures 27 (for Quad-Solver) and 28 (for NormSolver). The aforementioned clutter nulling peculiarities specific to each solver reappear, but the target is well localized in each case, with the peak of the pattern at its location. This behavior is confirmed by the cuts along the target Doppler (Figures 29 and 30) and target azimuth (Figures 31 and 32), with the target location in each denoted by the dashed line. While the azimuth plots have large sidelobes, this is mostly attributable to the limited number of antennas in this simulation ($M = 5$) and not the optimization process.

Now suppose we inject a broadband interferer "close" to the target – as a reminder, the target is at $(\theta_t, \phi_t) = (0.3, \frac{\pi}{3})$ radians and the interferer is at $(\theta_I, \phi_I) = (0.3941, 0.3)$ radians. Figures 33 and 34 (for QuadSolver and NormSolver, respectively) show the adapted pattern under these conditions. As predicted by the subspace alignment in Figure 26, the peak in NormSolver's pattern is shifted away from the target quite significantly. What may be surprising, however, is that QuadSolver's adapted pattern shows the *exact same error*, despite the subspace alignment appearing to suggest otherwise. There are a variety of factors that contribute to this, but the most salient
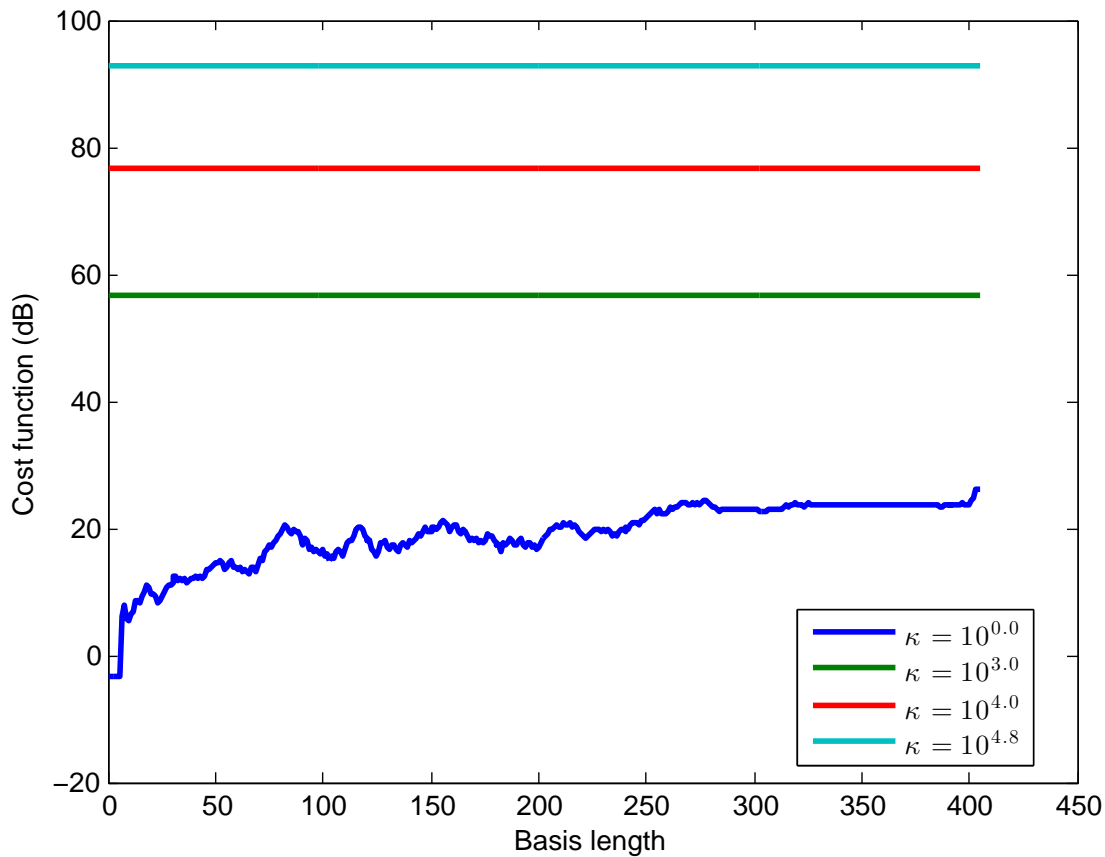
Fig. 9. Original cost as function of basis length, NormSolver

is that the elevation angles of clutter patches are not identical to that of the interferer. Hence, the subspaces being nulled by QuadSolver do project into the adapted pattern as nulls, but are not identical to the subspace spanned by the interference. This lends credence to our proposition that the relaxed optimization process is a trans-recieve generalized whiten-and-match filter. In both cases, the resources available permit us to only *whiten* the interference, especially because it is not signal-dependent.

For completeness, we also show the adapted pattern cuts under the interference scenario. Again, in all figures, the dashed line represents the target location. Figures 35 and 36 show the cut along the target Doppler. The significant bias seen above is more clear here, which would result in a rather large angle estimation error. The cuts along the target azimuth, shown in Figures 37 and 38, still appropriately localize the target in some sense. However, note that in both cases, the peak gain at the target Doppler has dropped about 5 dB from the gain in Figures 31 and 32. Thus, there is a clear loss in both dimensions due to the interferer.

We note that this interference impact (in all cases) is lessened by increasing M in our simulations. This is not surprising, since more antenna elements over the same aperture increases the degrees of freedom available to null interference and localize the target.
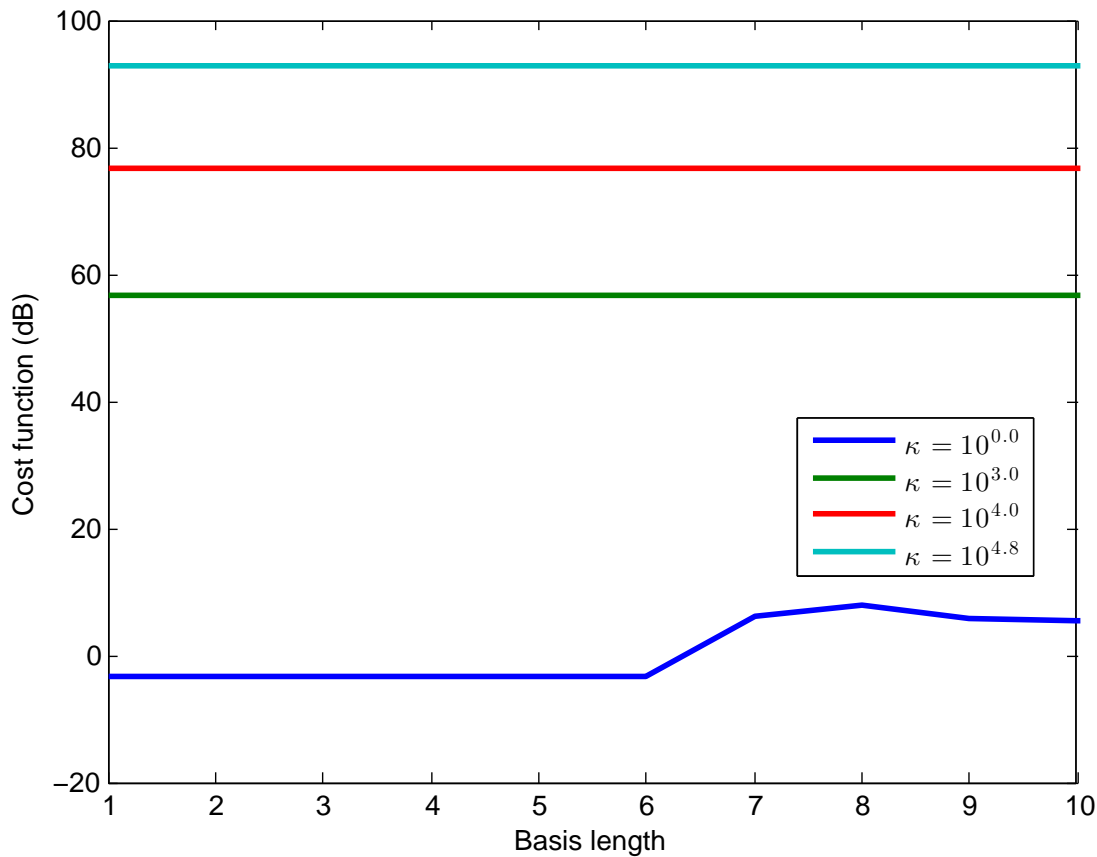
Fig. 10. Original cost as function of basis length, first $2N$, NormSolver

VIII. Conclusions

In this report, we reconsidered the problem of [1] under the relaxed biquadratic program framework. We demonstrated definitively that the problem is non-convex, then showed the relaxation process. We then showed that the KKTs require power-bounded solutions when the noise-and-interference matrix is full rank, and that such a solution admits a waterfilling interpretation. Simulations demonstrated that numerical solvers can provide divergent solution paths, depending on scaling – one closer to the traditional clutter-nulling process, the other relying on a matched target-in-clutter response. Future work will attempt to generalize the findings made in this report and demonstrate its utility for other radar system models that admit a channel response representation.

References

[1] P. Setlur and M. Rangaswamy, "Waveform design for radar STAP in signal dependent interference," *IEEE Transactions on Signal Processing*, vol. 64, no. 1, pp. 19–34, January 2016.

[2] N. L. Owsley, "A recent trend in adaptive spatial processing for sensor arrays: Constrained adaptation," in *Signal Processing*, J. W. R. Griffiths, P. L. Stocklin, and C. van Schoonfeld, Eds. Academic Press, 1972, pp. 591–604.

[3] R. Klemm, *Principles of Space-Time Adaptive Processing*. Institution of Electrical Engineers, 2002.
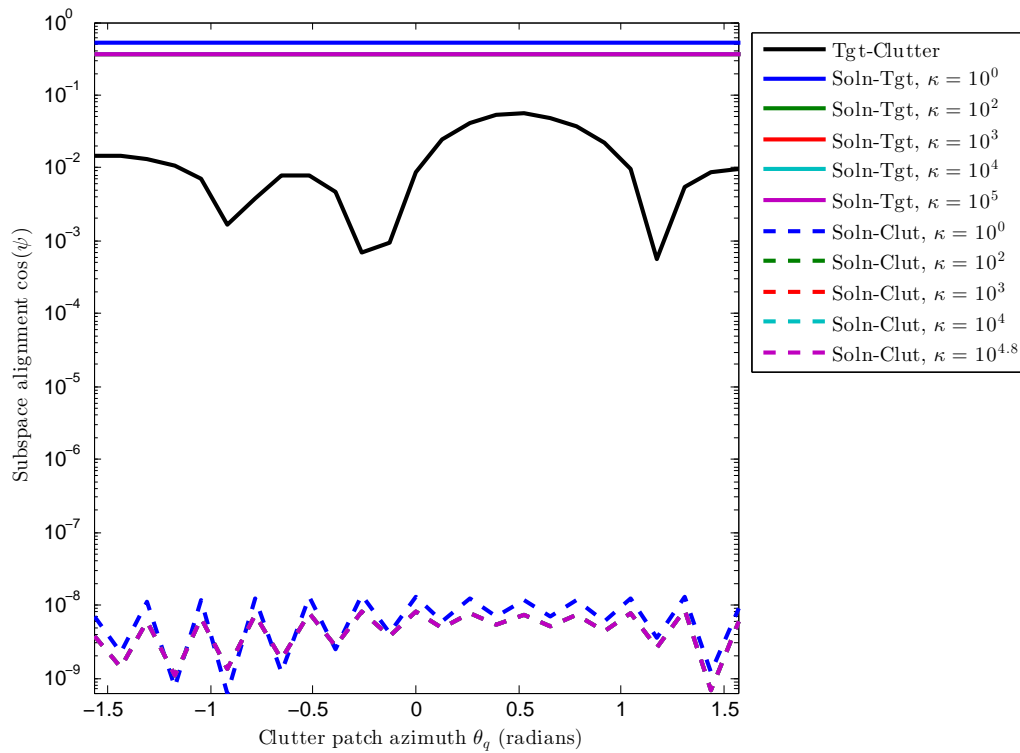
Fig. 11. Subspace Cosines, QuadSolver

[4] J. Ward, "Space-time adaptive processing for airborne radar," Massachusetts Institute of Technology, Lincoln Laboratory, Tech. Rep., 1994.

[5] D. Madurasinghe and A. P. Shaw, "Mainlobe jammer nulling via TSI finders: a space-time adaptive processor," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–13, 2006.

[6] Y. Seliktar, D. B. Williams, and E. J. Holder, "A space/fast-time adaptive monopulse technique," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–11, 2006.

[7] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, January 1969.

[8] A. Aubry, A. DeMaio, M. Piezzo, A. Farina, and M. Wicks, "Cognitive design of the receive filter and transmitted phase code in reverberating environment," *IET Radar, Sonar and Navigation*, vol. 6, no. 9, pp. 822–833, December 2012, barf 1.

[9] 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, January 2013.

[10] C. Y. Chen and P. P. Vaidyanathan, "MIMO radar waveform optimization with prior information of the extended target and clutter," *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3533–3544, 2009.

[11] P. Stoica, H. He, and J. Li, "Optimization of the receive filter and transmit sequence for active sensing," *IEEE Trans. Signal Processing*, vol. 60, no. 4, pp. 1730–1740, April 2012.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*.   New York, NY, USA: Cambridge University Press, 2004.

[13] C. Ling, J. Nie, L. Qi, and Y. Ye, "Biquadratic optimization over unit spheres and semidefinite programming relaxations," *SIAM Journal of Optimization*, vol. 20, pp. 1286–1310, 2009.

[14] C. Ling, X. Zhang, and L. Qi, "Semidefinite relaxation approximation for multivariate bi-quadratic optimization with quadratic constraints," *Numer. Linear Algebra Appl.*, vol. 19, no. 1, pp. 113–131, apr 2011. [Online]. Available: http://dx.doi.org/10.1002/nla.781

[15] J. R. Magnus and H. Neudecker, "The Commutation Matrix: Some Properties and Applications," *The Annals of Statistics*, vol. 7, no. 2,
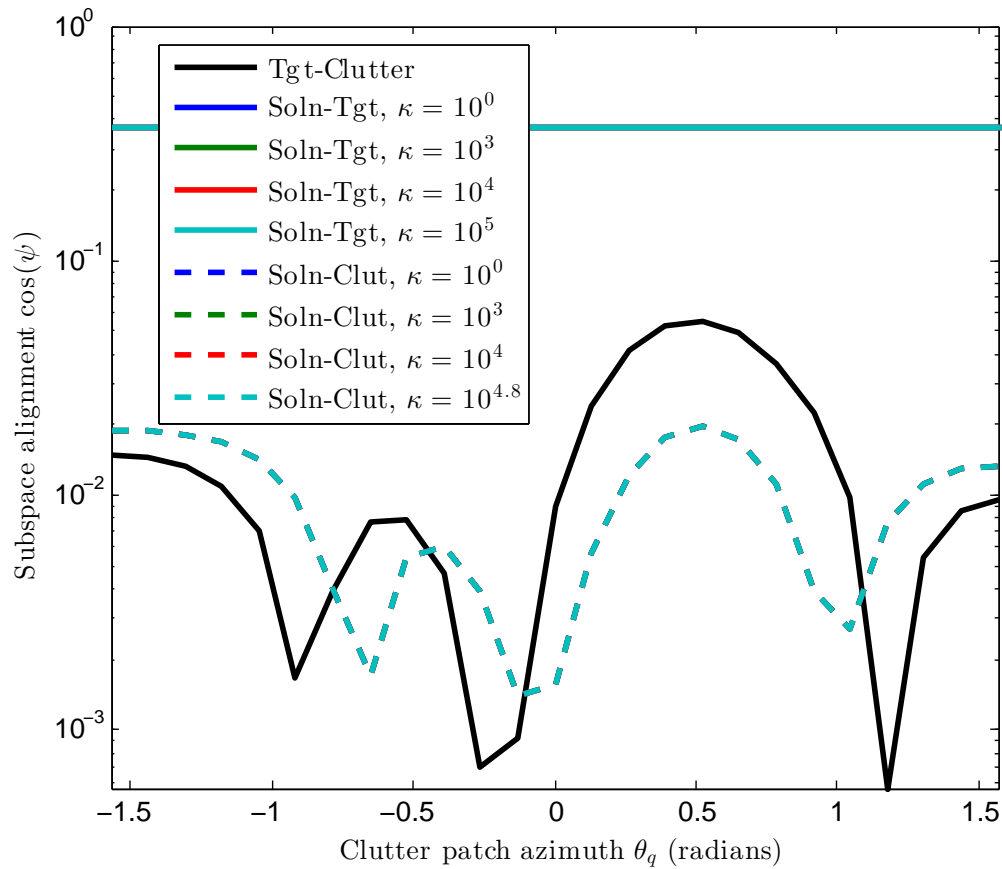
Fig. 12. Subspace Cosines, NormSolver

pp. 381–394, Mar. 1979.

[16] J. R. Guerci, J. S. Bergin, R. J. Guerci, M. Khanin, and M. Rangaswamy, "A new mimo clutter model for cognitive radar," in *2016 IEEE Radar Conference (RadarConf)*, May 2016, pp. 1–6.

[17] A. Hjørungnes, *Complex-Valued Matrix Derivatives*, 1st ed.   Cambridge, UK: Cambridge University Press, 2011.

[18] G. A. F. Seber, *A Matrix Handbook for Statisticians*, 1st ed.   Hoboken, NJ, USA: John Wiley & Sons, Inc., Nov. 2007.

[19] E. Kreindler and A. Jameson, "Conditions for nonnegativeness of partitioned matrices," *IEEE Trans. Automat. Contr.*, vol. 17, no. 1, pp. 147–148, Feb. 1972.

[20] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed.   New York, NY: Cambridge University Press, 2013.

[21] I. M. Bomze, C. Ling, L. Qi, and X. Zhang, "Standard bi-quadratic optimization problems and unconstrained polynomial reformulations," *J. Glob. Optim.*, vol. 52, no. 4, pp. 663–687, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10898-011-9710-5

[22] X. Zhang, C. Ling, and L. Qi, "Semidefinite relaxation bounds for bi-quadratic optimization problems with quadratic constraints," *J. Glob. Optim.*, vol. 49, pp. 293–311, 2011.

[23] C. G. Khatri and S. K. Mitra, "Hermitian and nonnegative definite solutions of linear matrix equations," *SIAM Journal on Applied Mathematics*, vol. 31, no. 4, pp. 579–7, 12 1976, copyright - Copyright] 1976 Society for Industrial and Applied Mathematics; Last updated - 2012-01-24. [Online]. Available: http://search.proquest.com.wrs.idm.oclc.org/docview/917326738?accountid=26189

[24] J. K. Baksalary, "Nonnegative definite and positive definite solutions to the matrix equation axa* = b," *Linear and Multilinear Algebra*, vol. 16, no. 1-4, pp. 133–139, 1984. [Online]. Available: http://dx.doi.org/10.1080/03081088408817616

[25] L. E. Brennan and F. M. Staudaher, "Subclutter visibility demonstration," Adaptive Sensors Incorporated, Tech. Rep. RL-TR-92-21, March
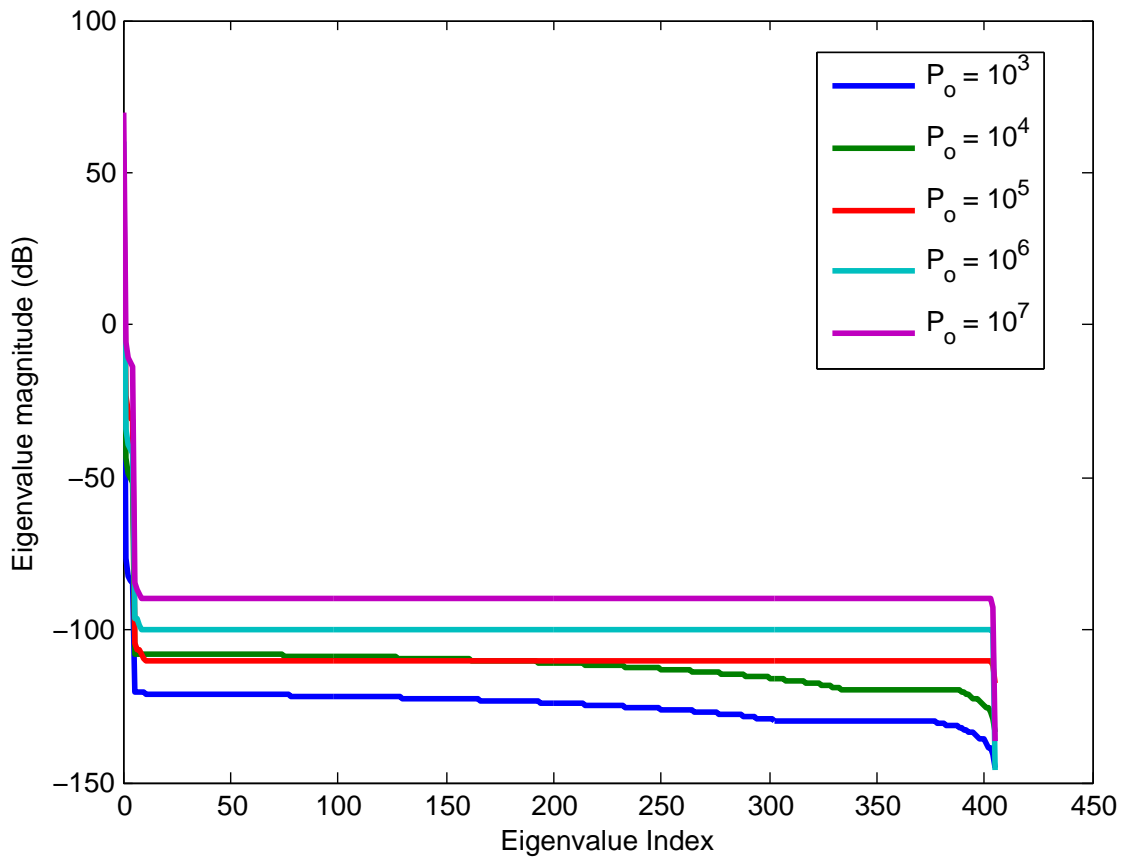
Fig. 13. Eigenvalue spread of **B** solution matrix using QuadSolver

1992.

[26] N. A. Goodman and J. M. Stiles, "On clutter rank observed by arbitrary arrays," *IEEE Trans. Signal Processing*, vol. 55, no. 1, pp. 178 – 186, January 2007.

[27] P. Setlur, N. Devroye, and M. Rangaswamy, "Radar waveform design with the two step mutual information," in *2014 IEEE Radar Conference*, May 2014, pp. 1317–1322.

[28] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, June 2015.

[29] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds.   Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

[30] K. Toh, M. Todd, and R. Tutuncu, "SDPT3 — a matlab software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, pp. 545–581, 1999.

[31] R. Tutuncu, K. Toh, and M. Todd, "Solving semidefinite-linear-quadratic programs using SDPT3," *Mathematical Programming Ser. B*, vol. 95, pp. 189–217, 2003.
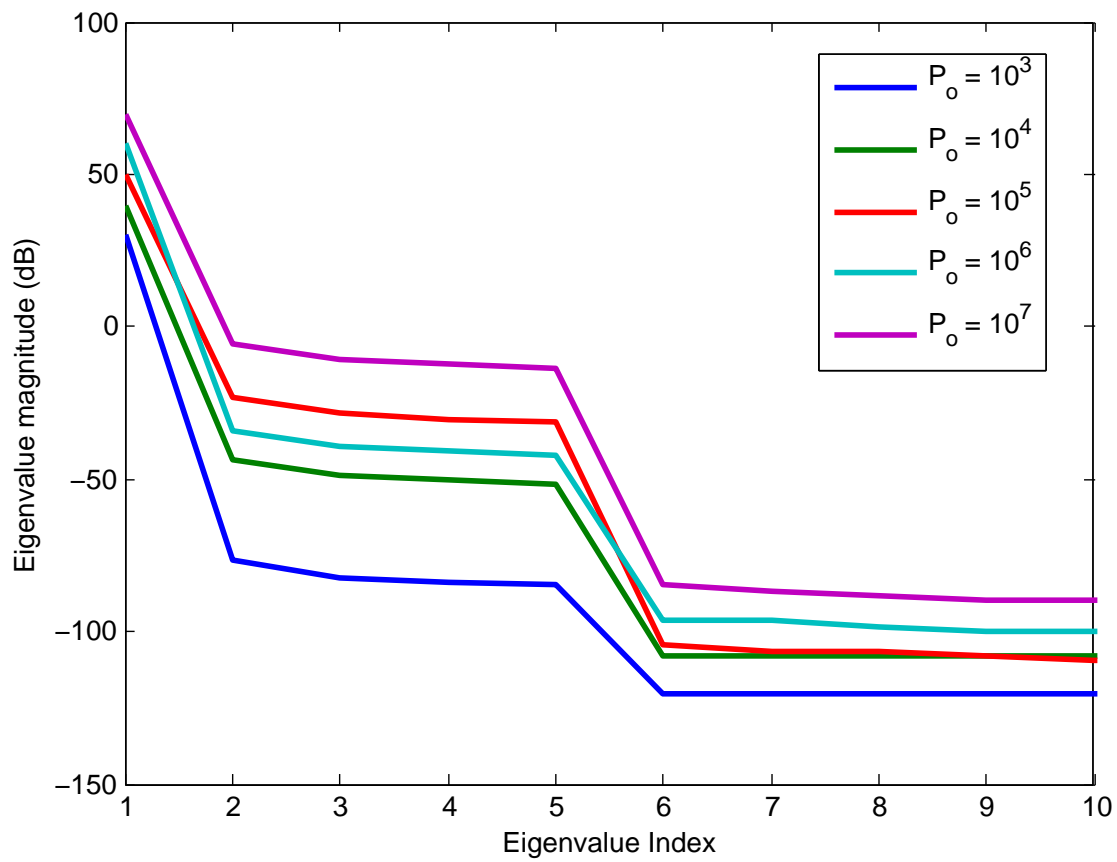
Fig. 14. Closeup of the first $2N$ eigenvals. of $\mathbf{B}$ using QuadSolver
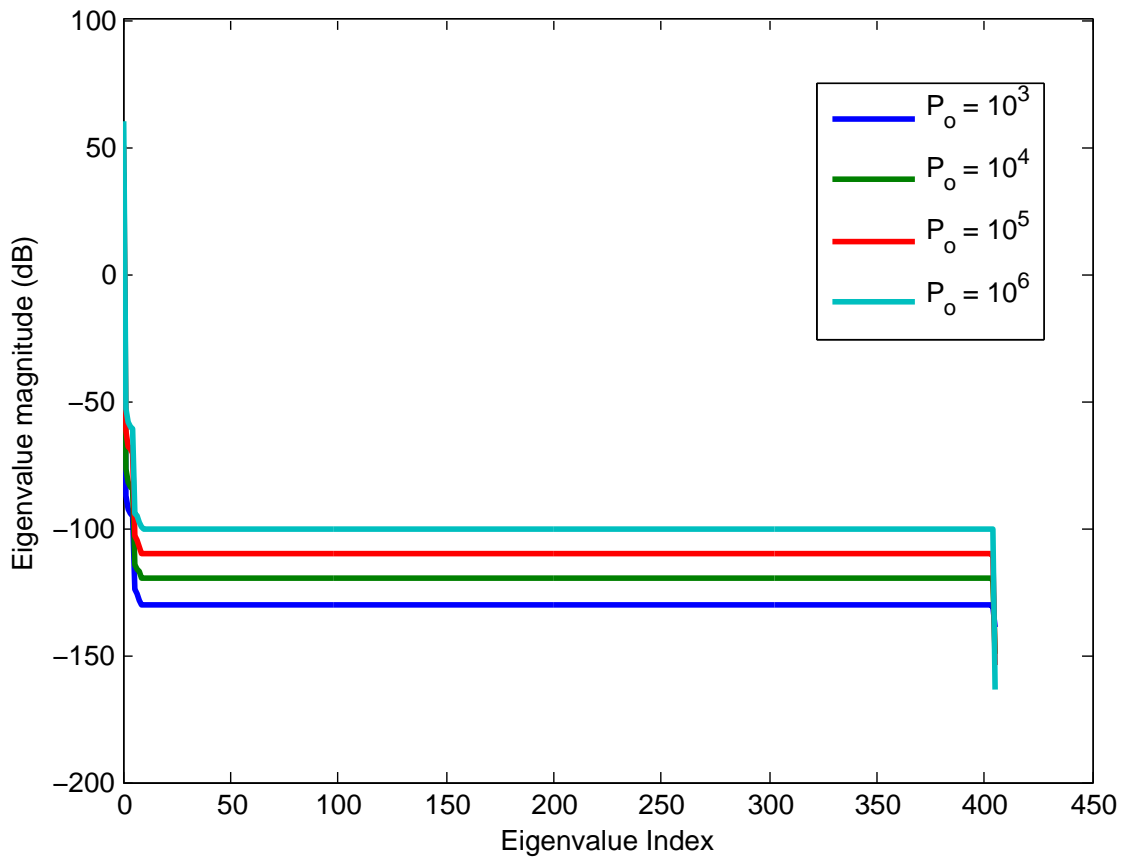
Fig. 15. Eigenvalue spread of **B** solution matrix using NormSolver
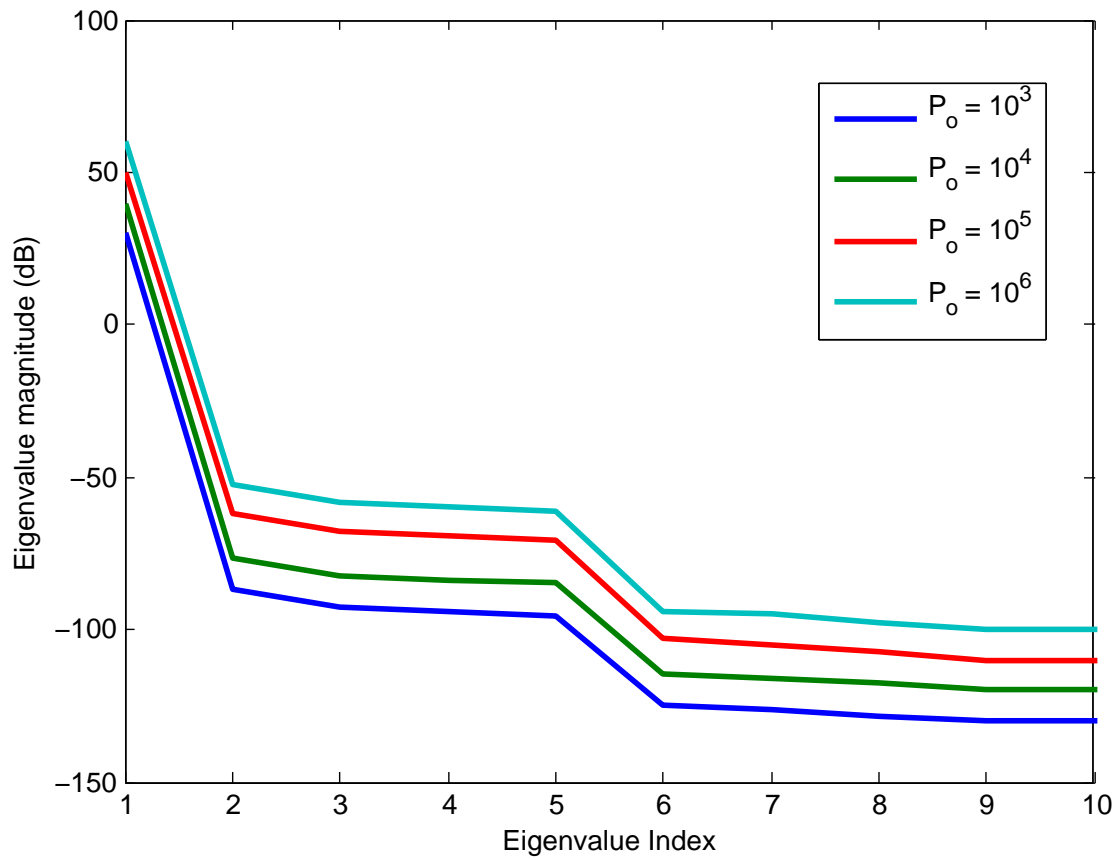
Fig. 16. Closeup of the first $2N$ eigenvals. of $\mathbf{B}$ using NormSolver
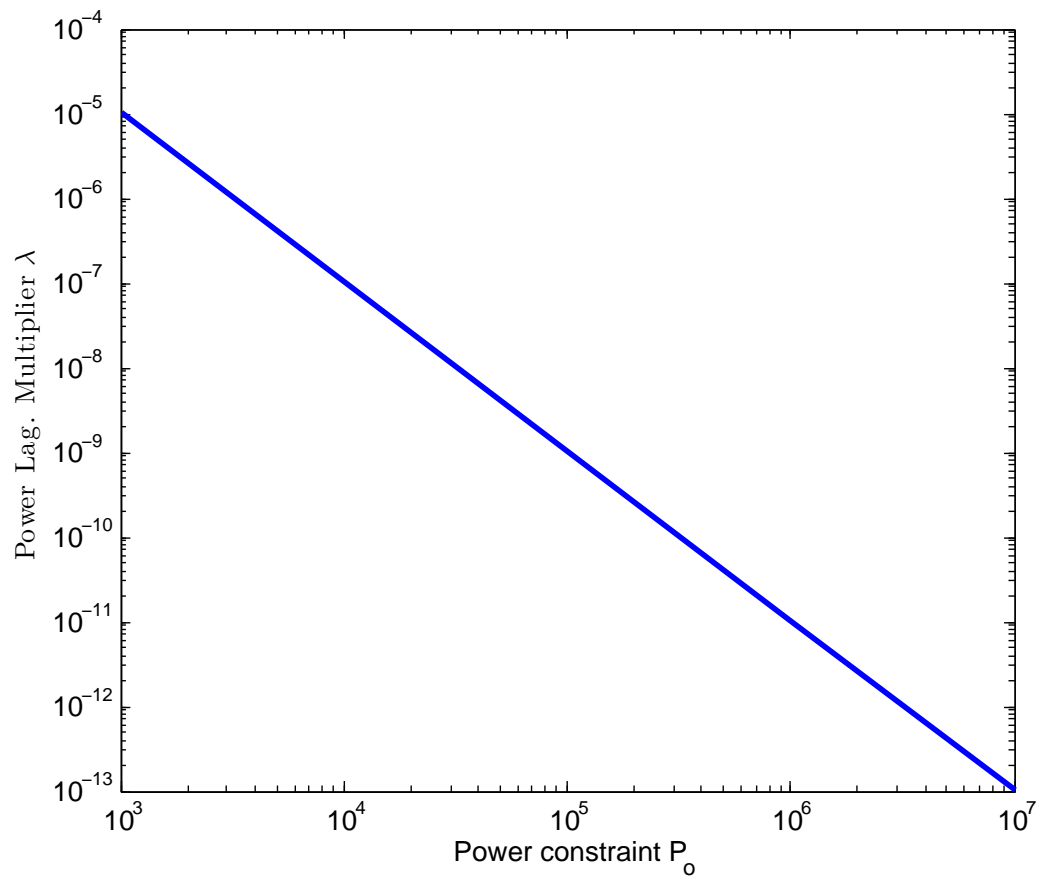
Fig. 17. Optimal λ as a function of the Capon constraint, QuadSolver
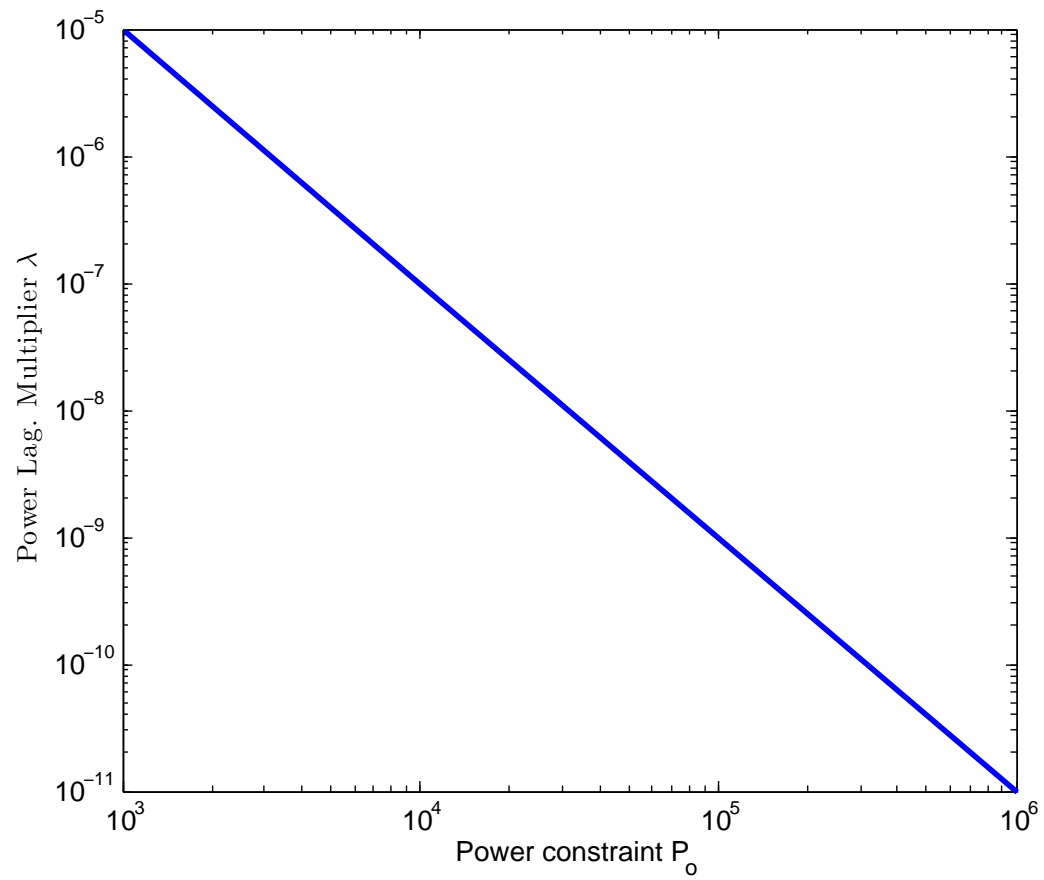
Fig. 18. Optimal λ as a function of the Capon constraint, NormSolver
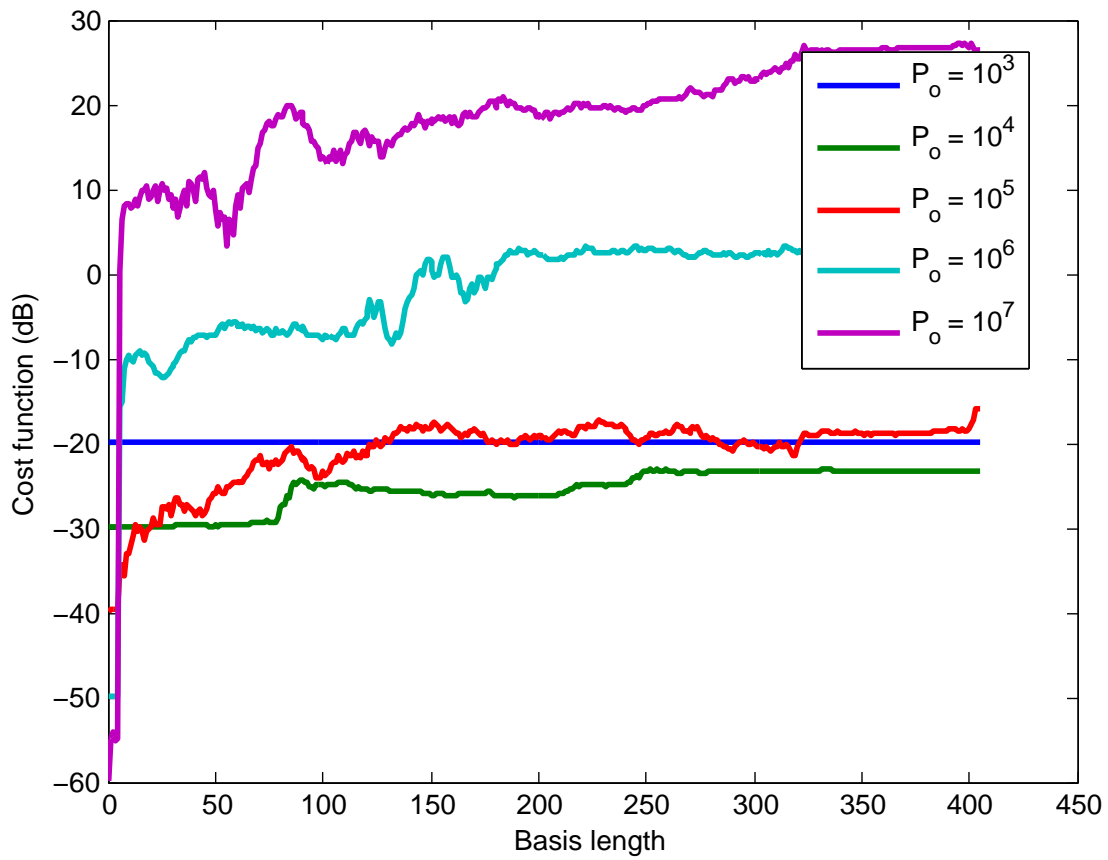
Fig. 19. Original cost as function of basis length, QuadSolver
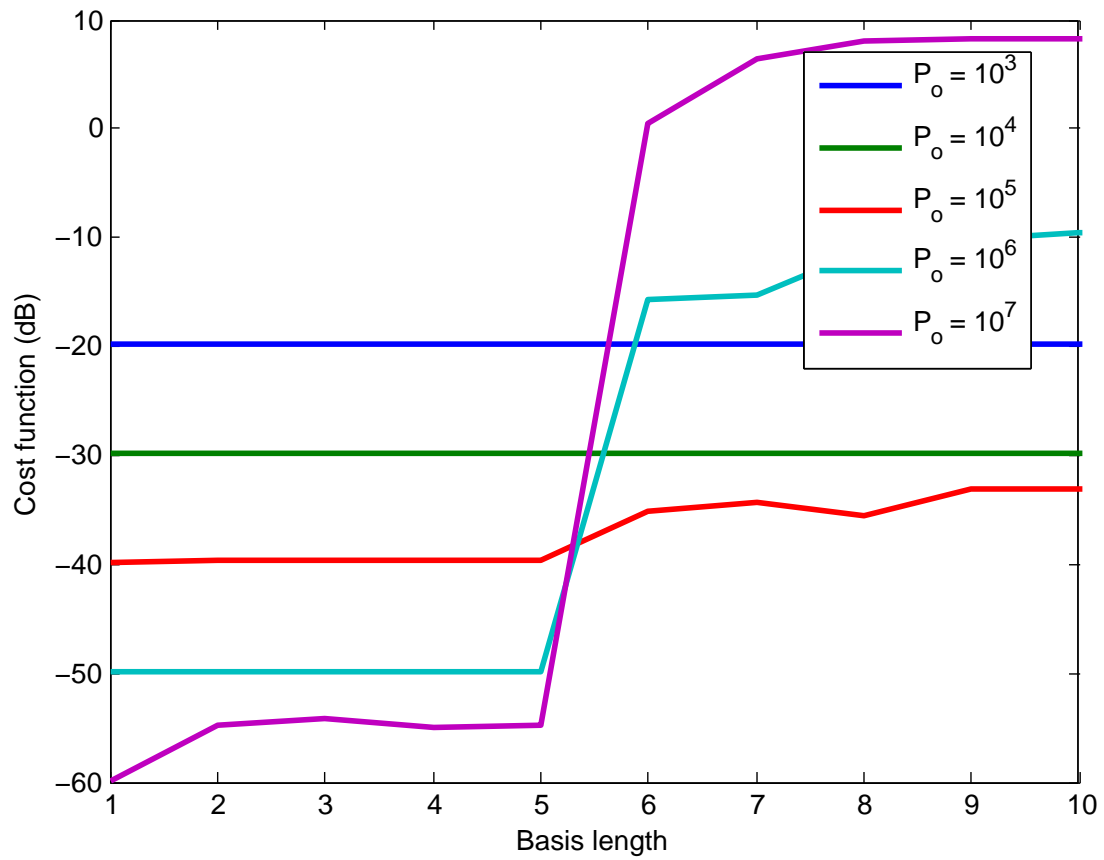
Fig. 20. Original cost as function of basis length, first $2N$, QuadSolver
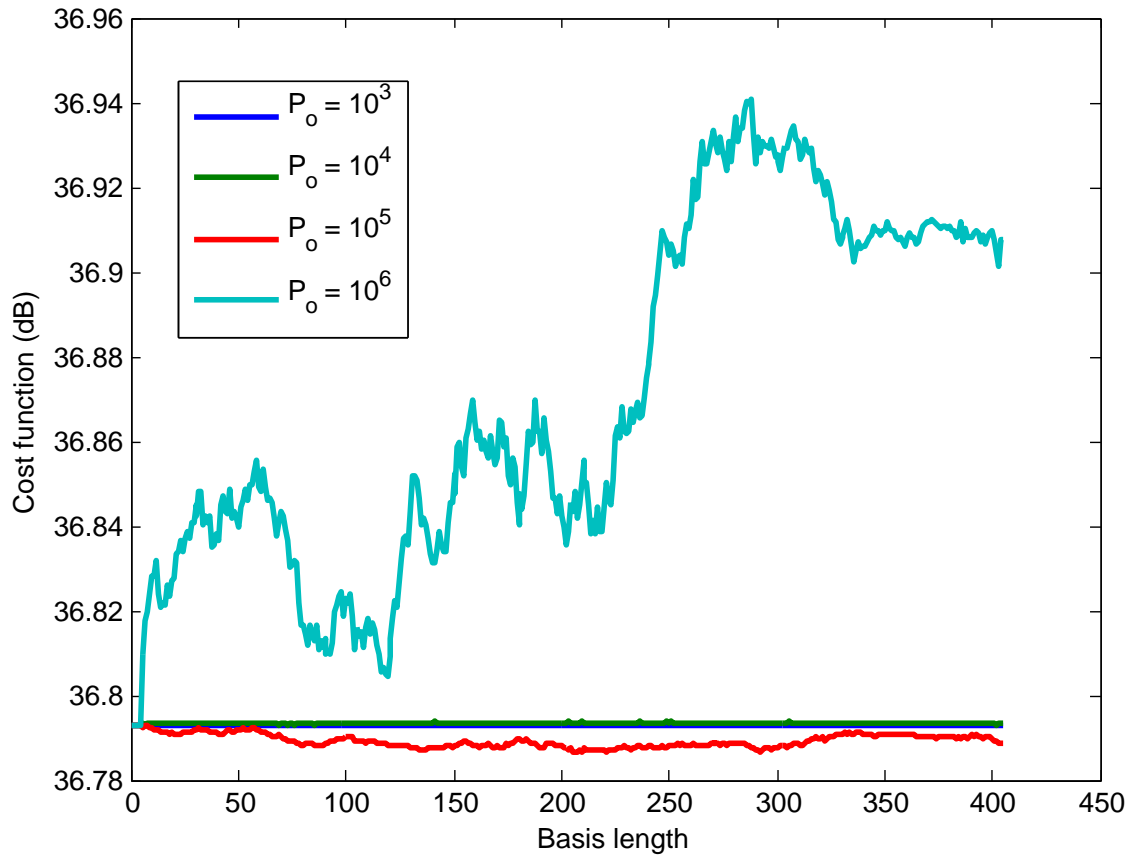
Fig. 21. Original cost as function of basis length, NormSolver
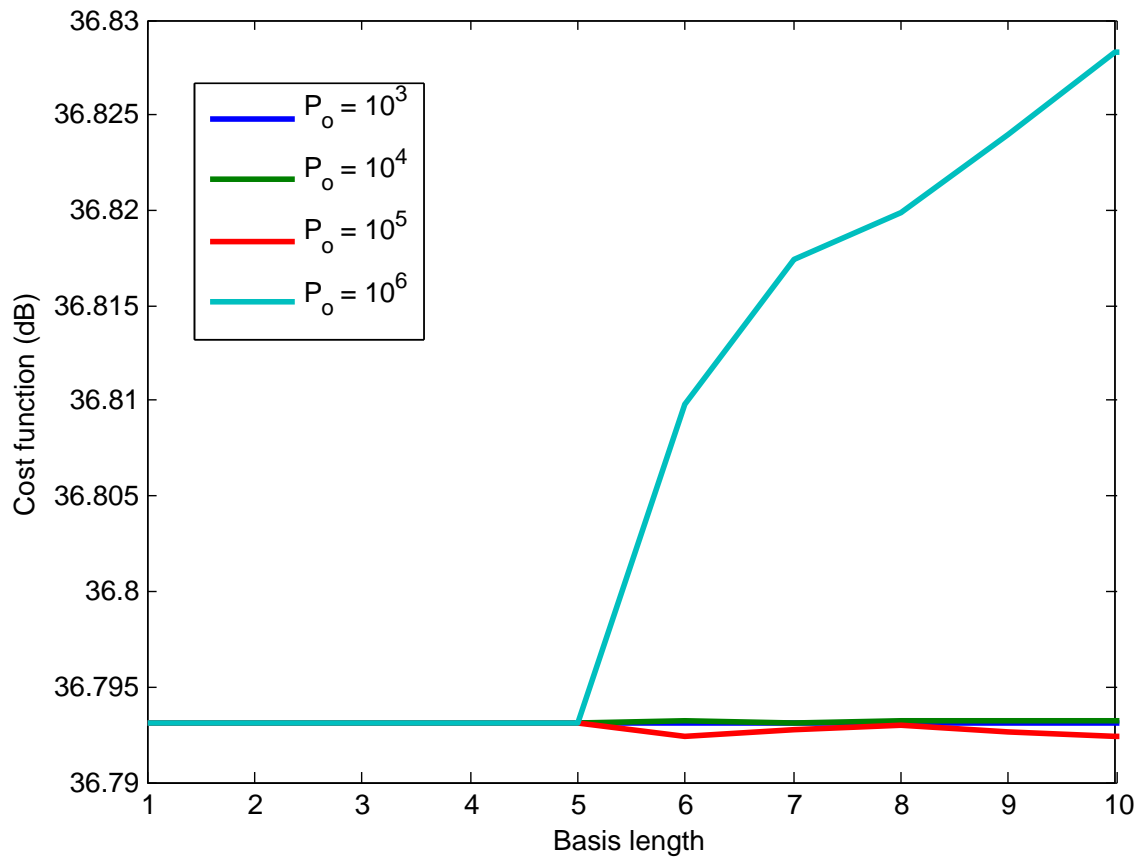
Fig. 22. Original cost as function of basis length, first $2N$, NormSolver
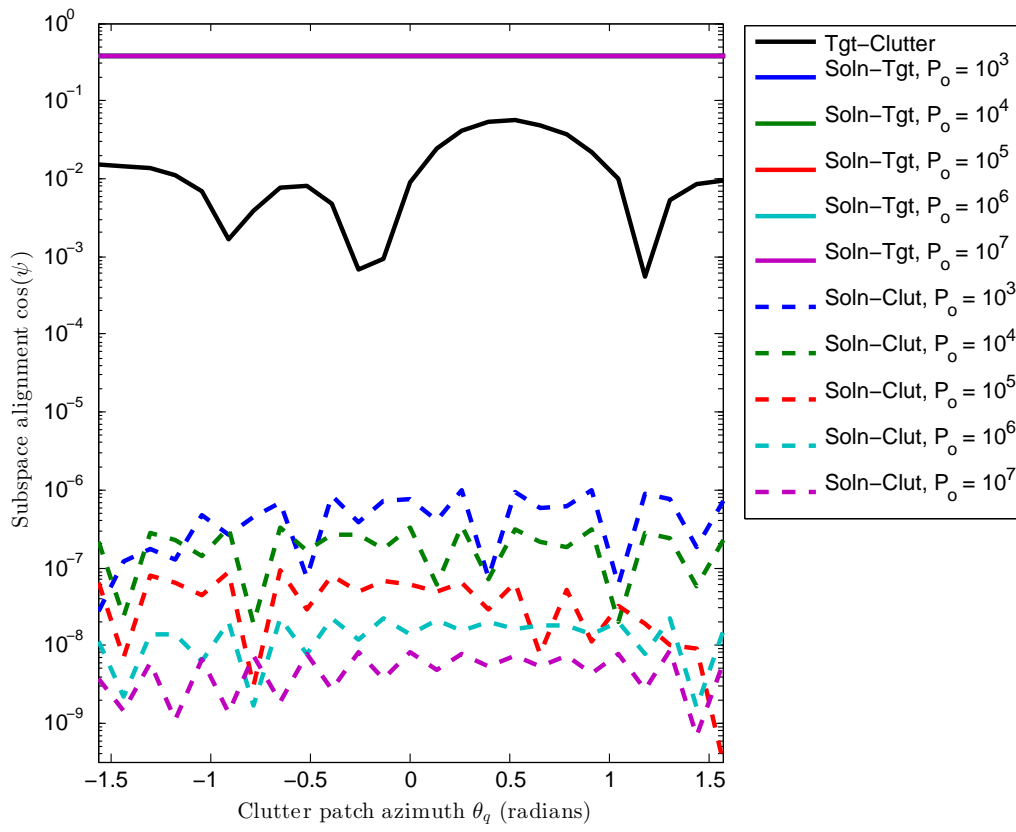
Fig. 23. Subspace Cosines, QuadSolver
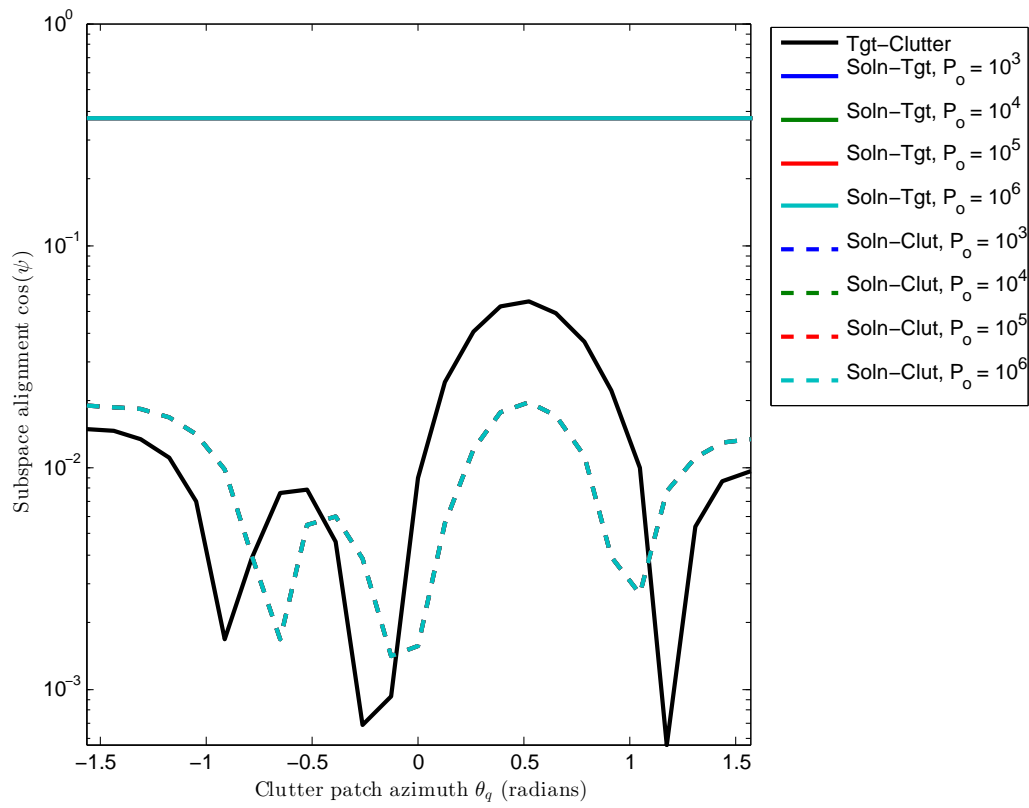
Fig. 24.　Subspace Cosines, NormSolver
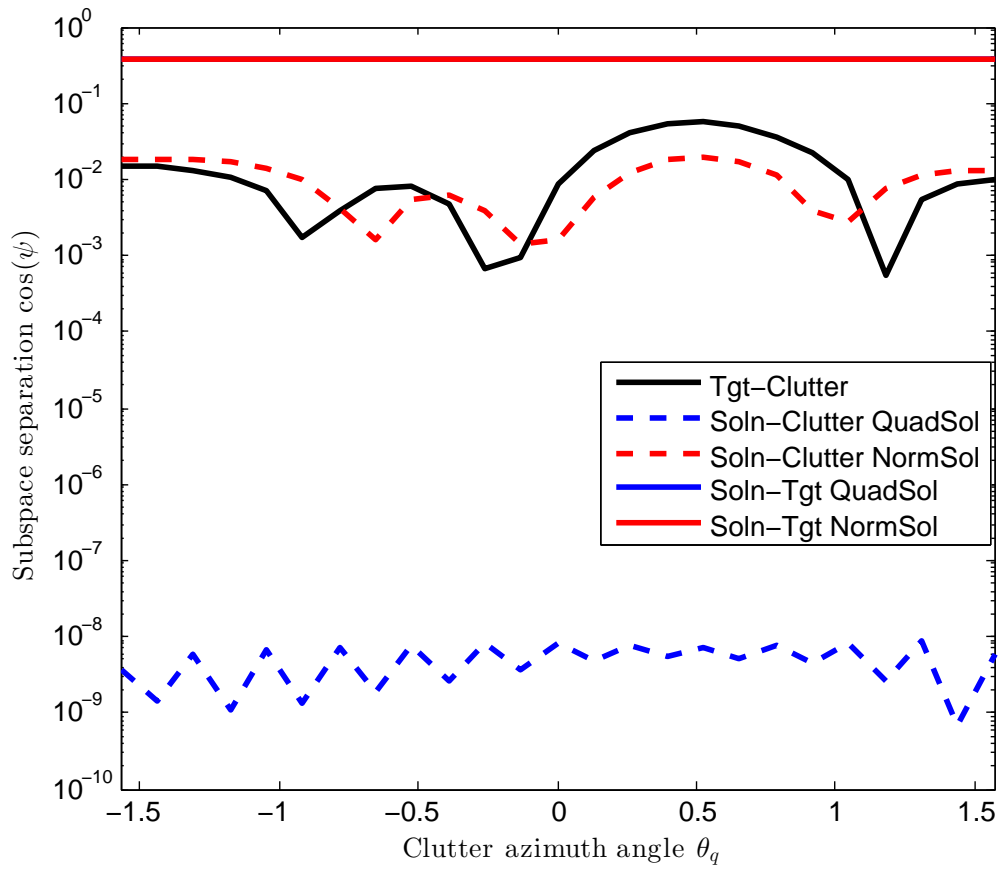
Fig. 25. Subspace Alignment, No Interference
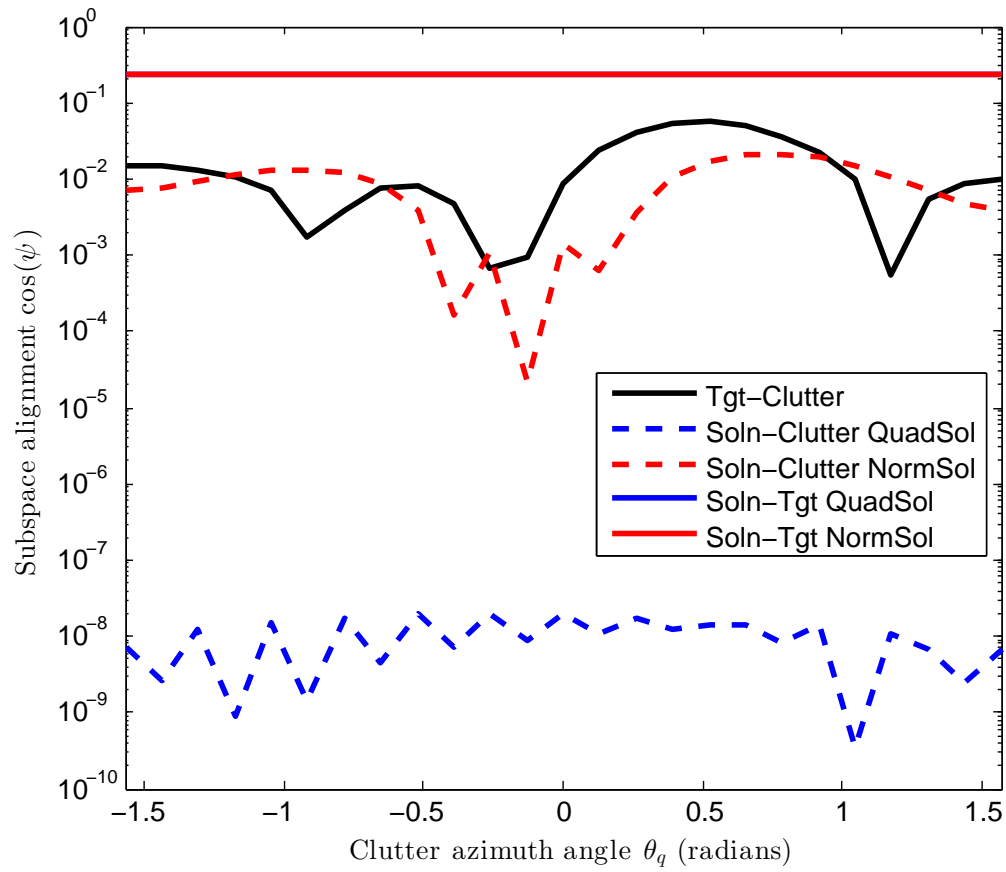
Fig. 26. Subspace Alignment, Interferer at $(\theta, \phi) = (0.3941, 0.3)$
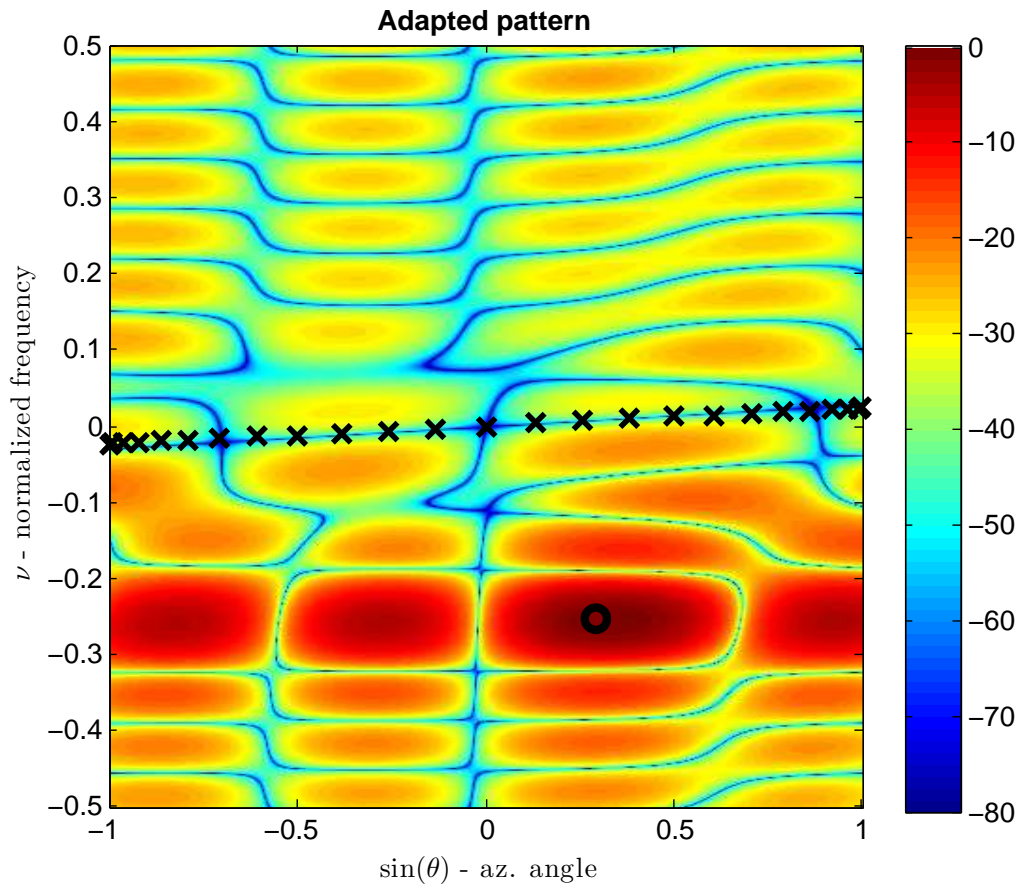
Fig. 27.   Adapted Pattern (dB scale), QuadSolver, No Interference. Target at ◯, Clutter phase centers at ×. No Interference.
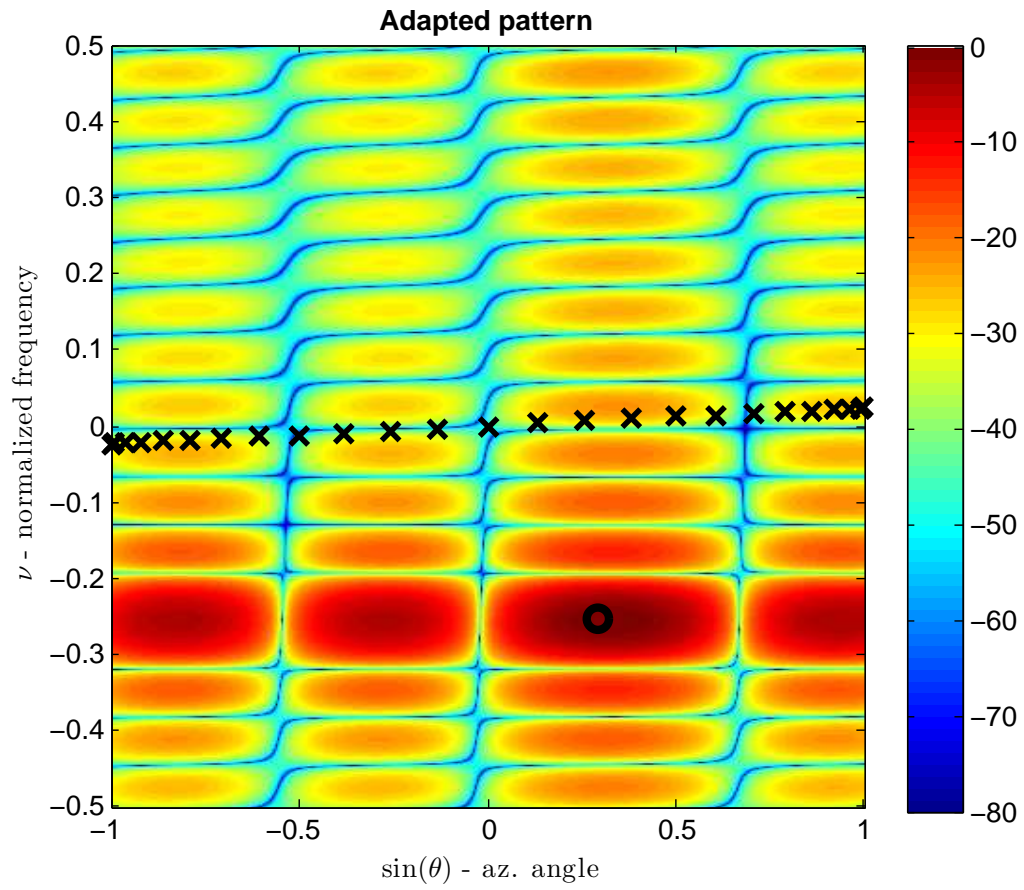
Fig. 28. Adapted Pattern (dB scale), NormSolver, No Interference. Target at ◯, Clutter phase centers at ×.
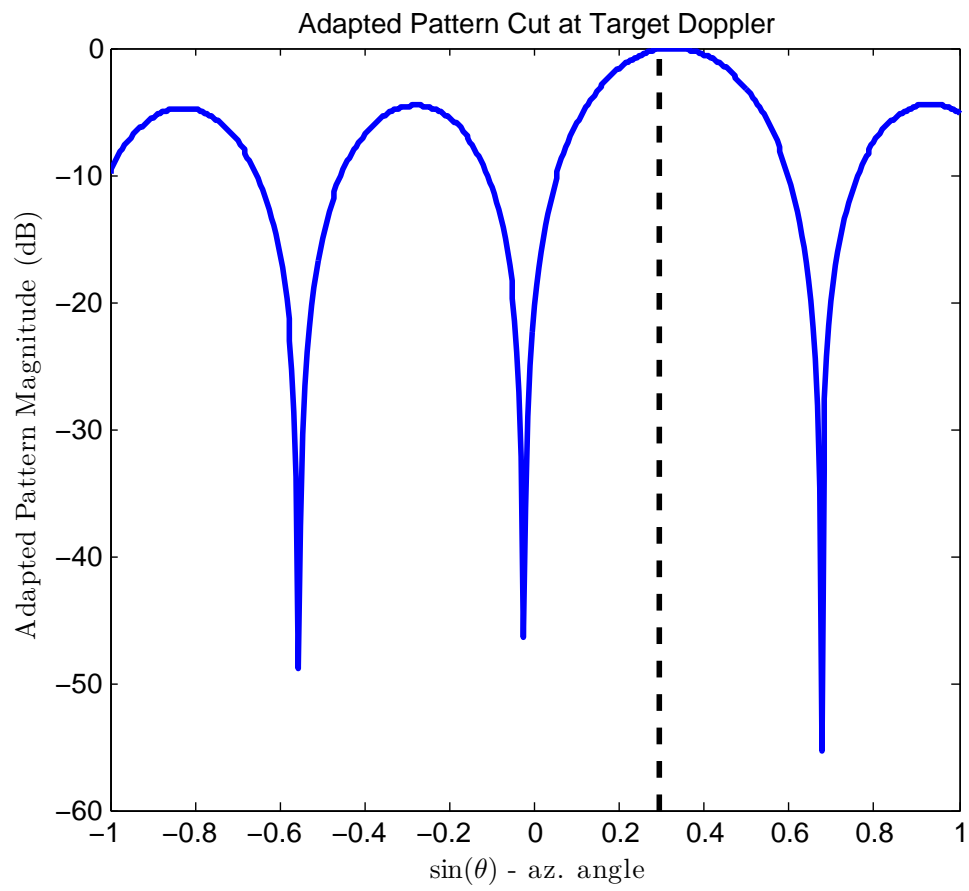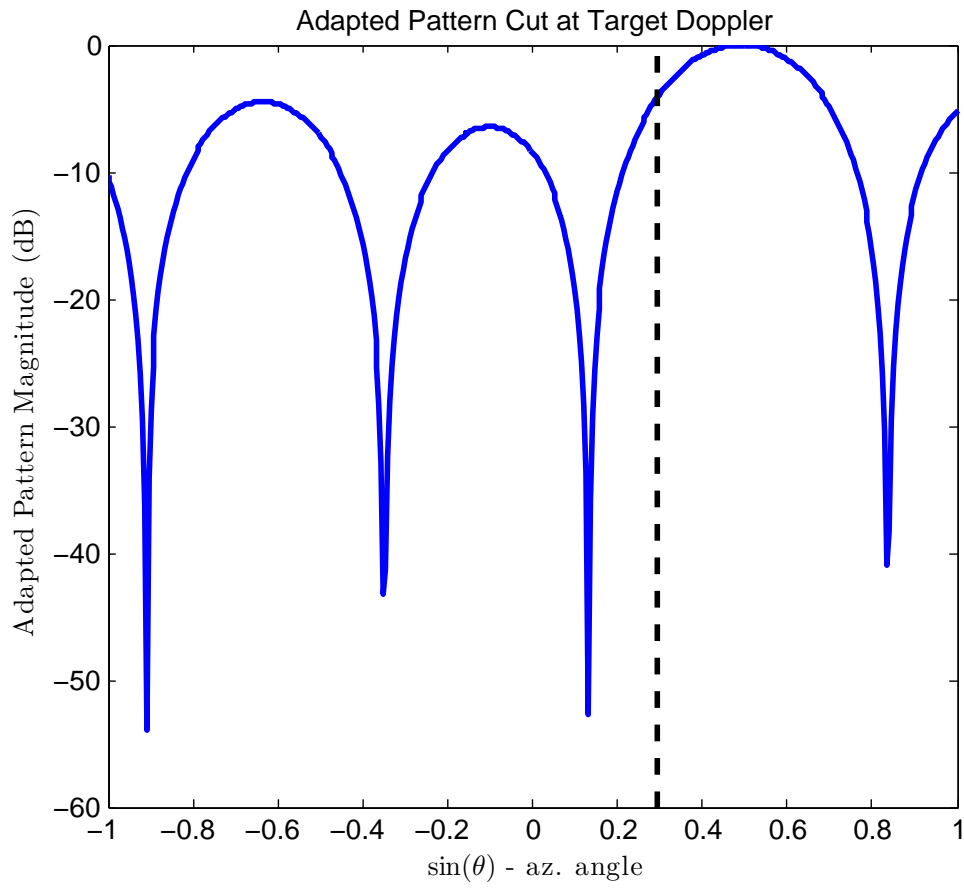
Fig. 29. Adapted Pattern (dB scale), cut at target Doppler, QuadSolver.
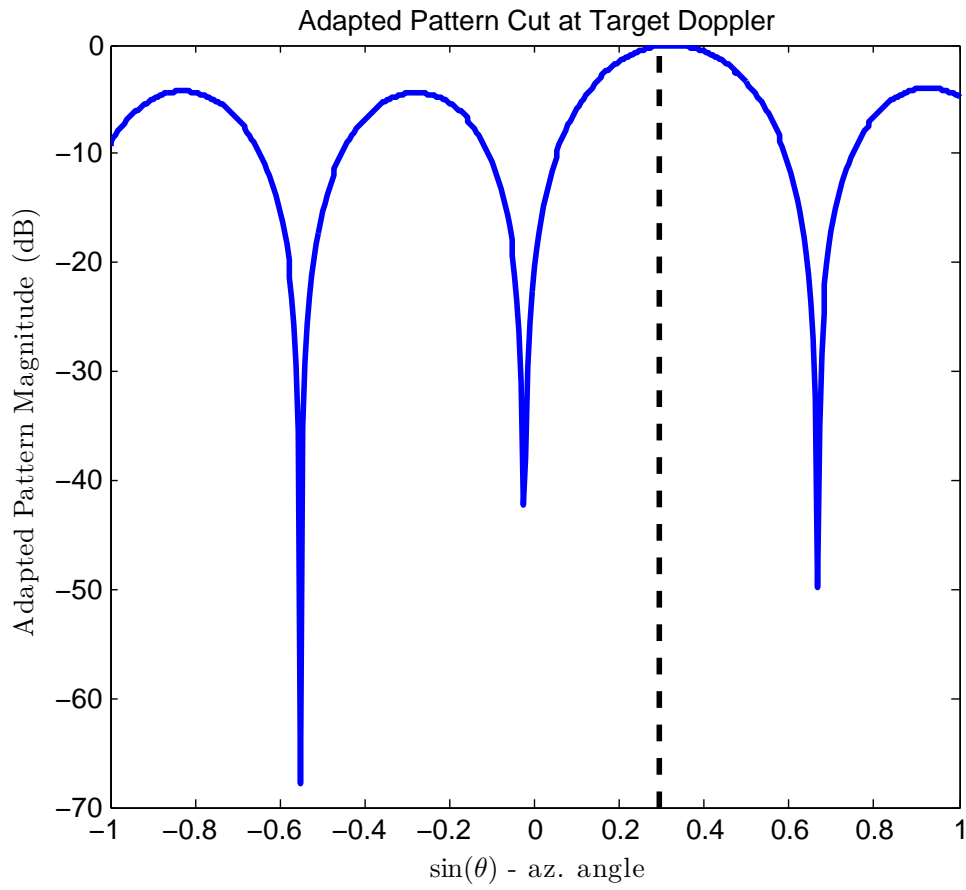
Fig. 30. Adapted Pattern (dB scale), cut at target Doppler, NormSolver.
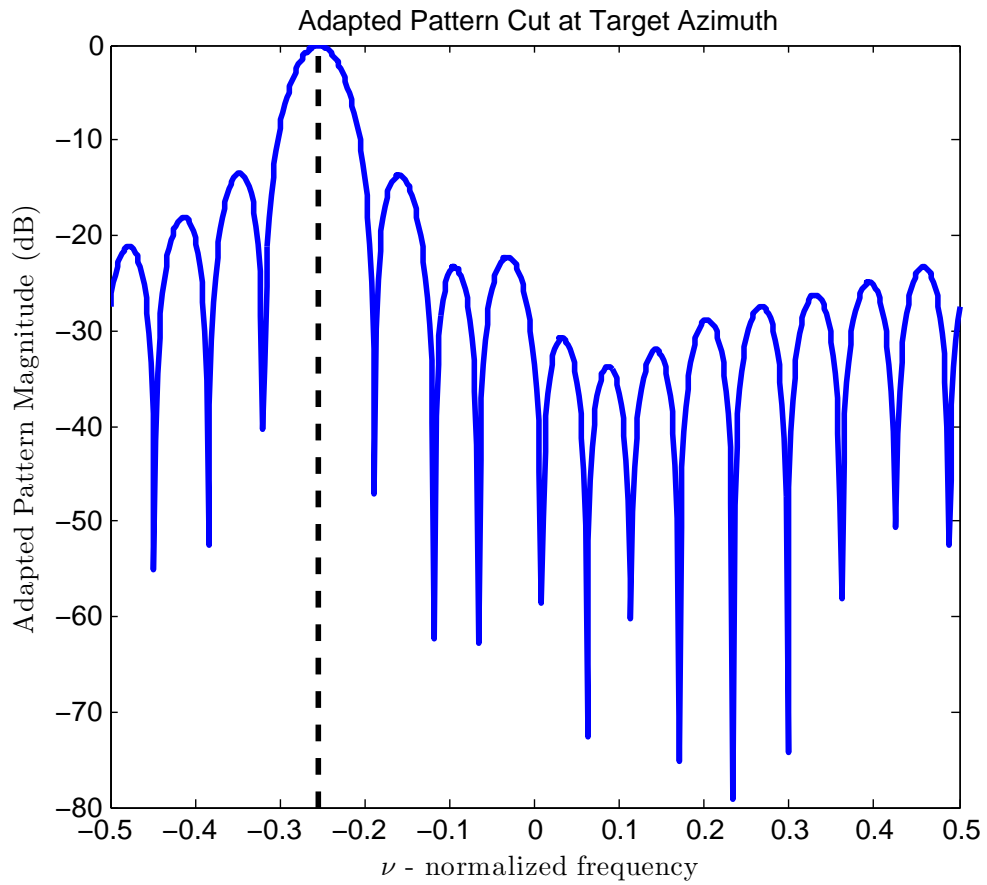
Fig. 31. Adapted Pattern (dB scale), cut at target azimuth, QuadSolver.
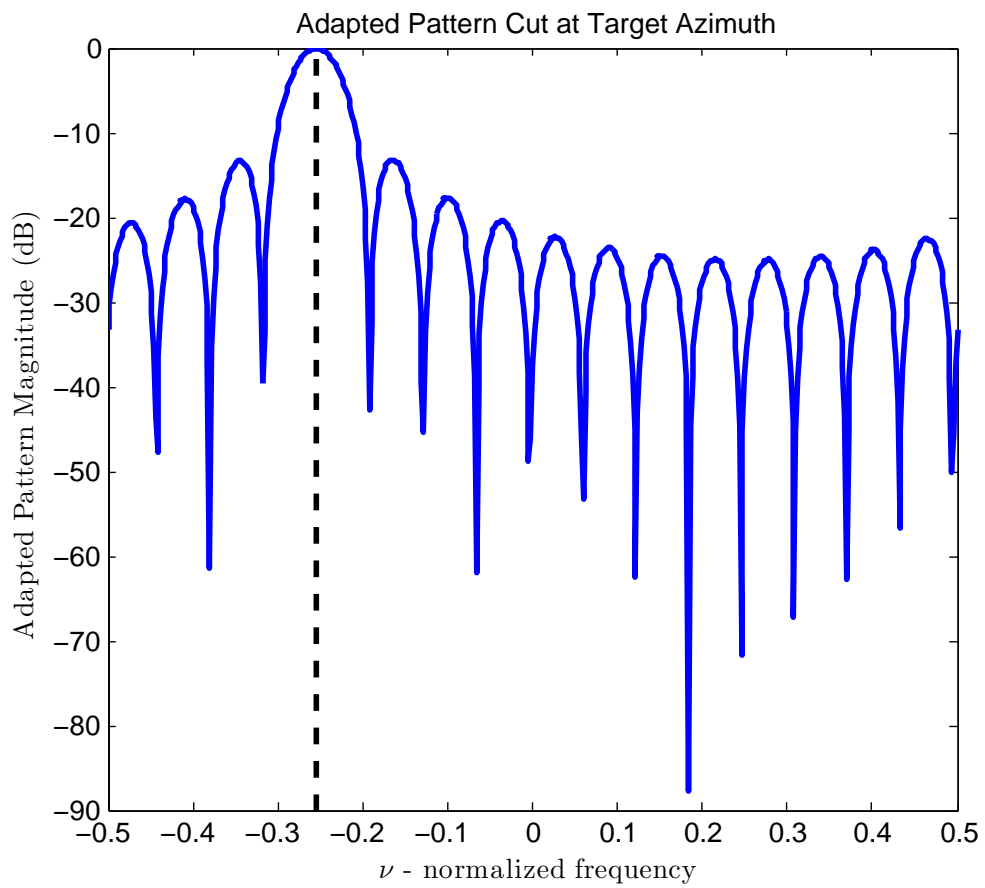
Fig. 32. Adapted Pattern (dB scale), cut at target azimuth, NormSolver.
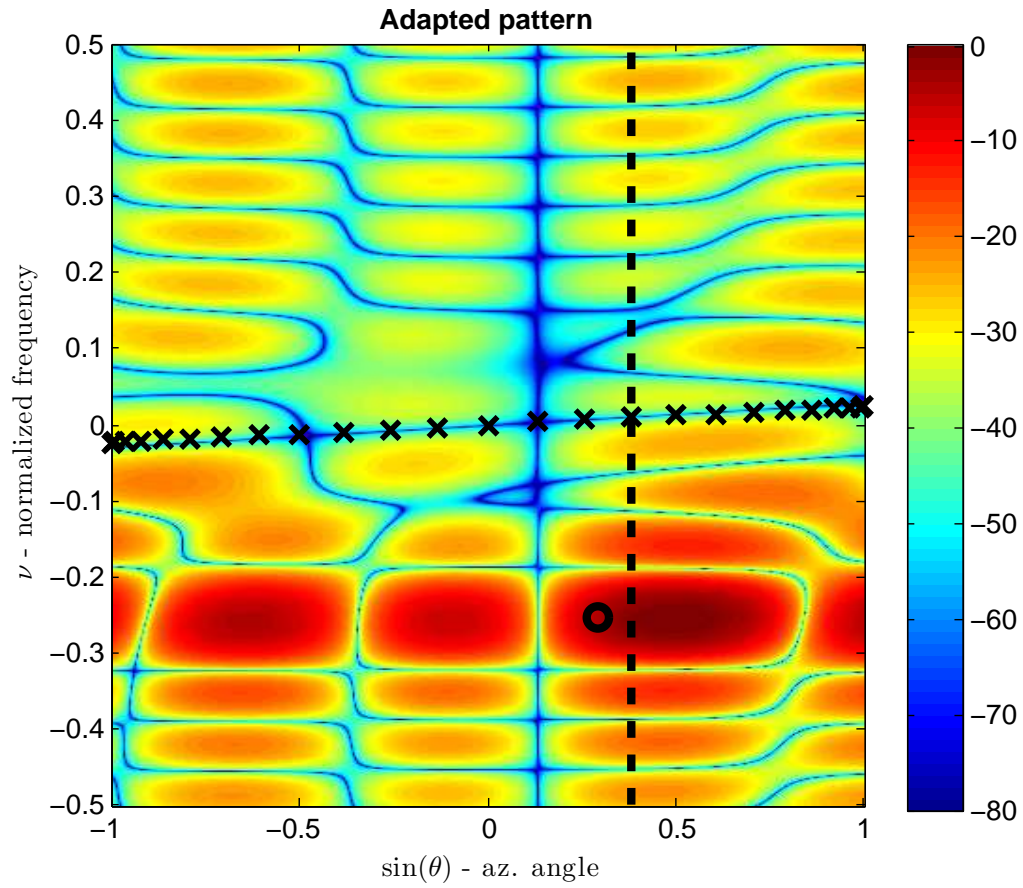
Fig. 33. Adapted Pattern (dB relative to peak), QuadSolver. Target at ◯, Clutter phase centers at ×, Interferer (dashed line) at $(\theta, \phi) = (0.3941, 0.3)$ radians
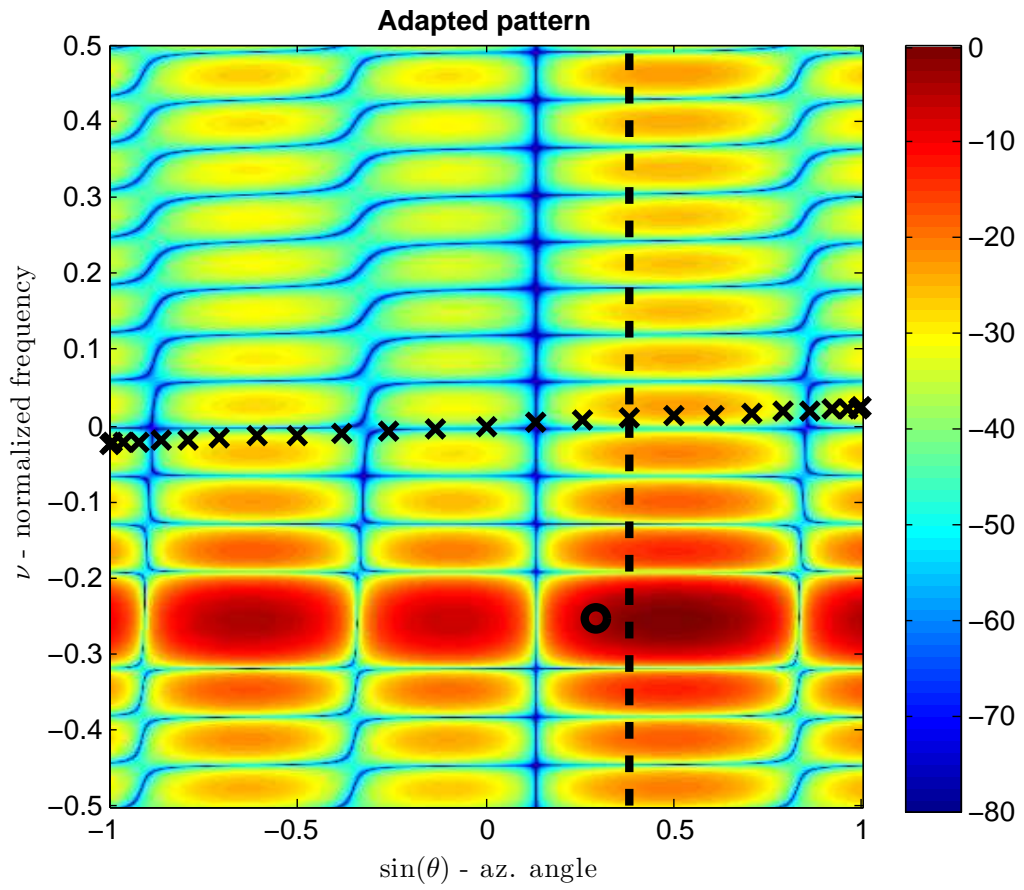
Fig. 34. Adapted Pattern, NormSolver. Target at $\bigcirc$, Clutter phase centers at $\times$, Interferer (dashed line) at $(\theta, \phi) = (0.3941, 0.3)$

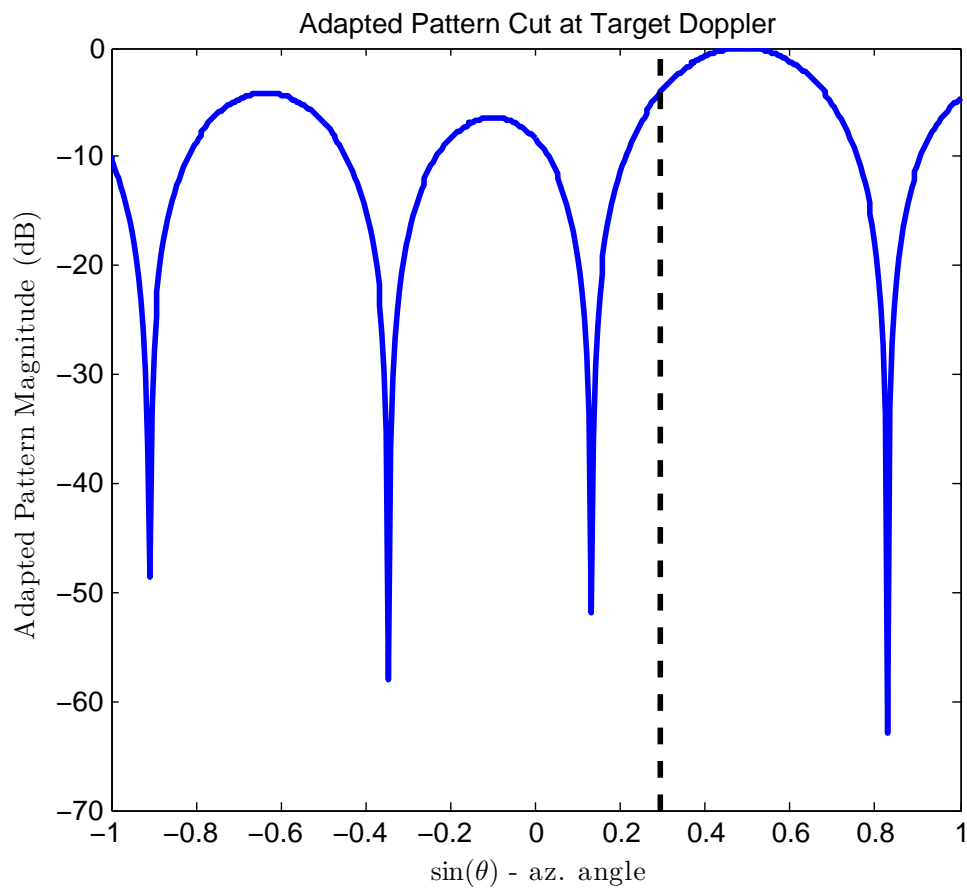Fig. 35. Adapted Pattern (dB scale), cut at target Doppler, QuadSolver. Interference.

Fig. 36. Adapted Pattern (dB scale), cut at target Doppler, NormSolver. Interference.
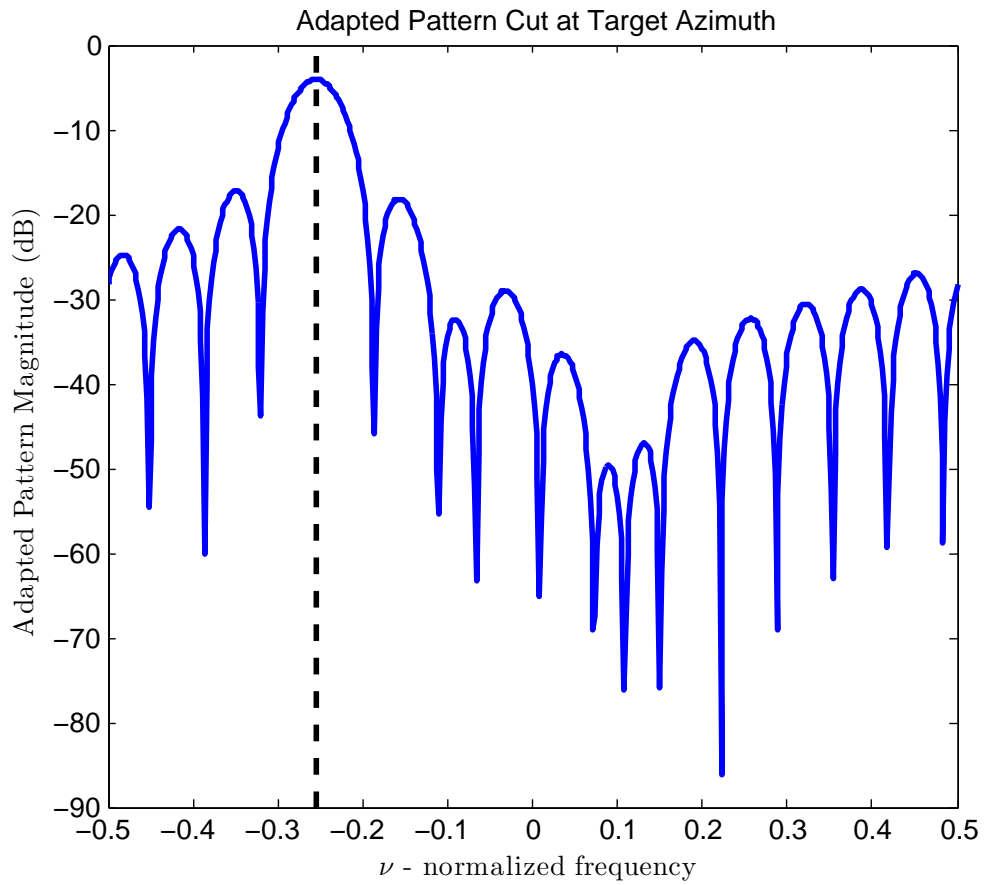
Fig. 37. Adapted Pattern (dB scale), cut at target azimuth, QuadSolver. Interference.
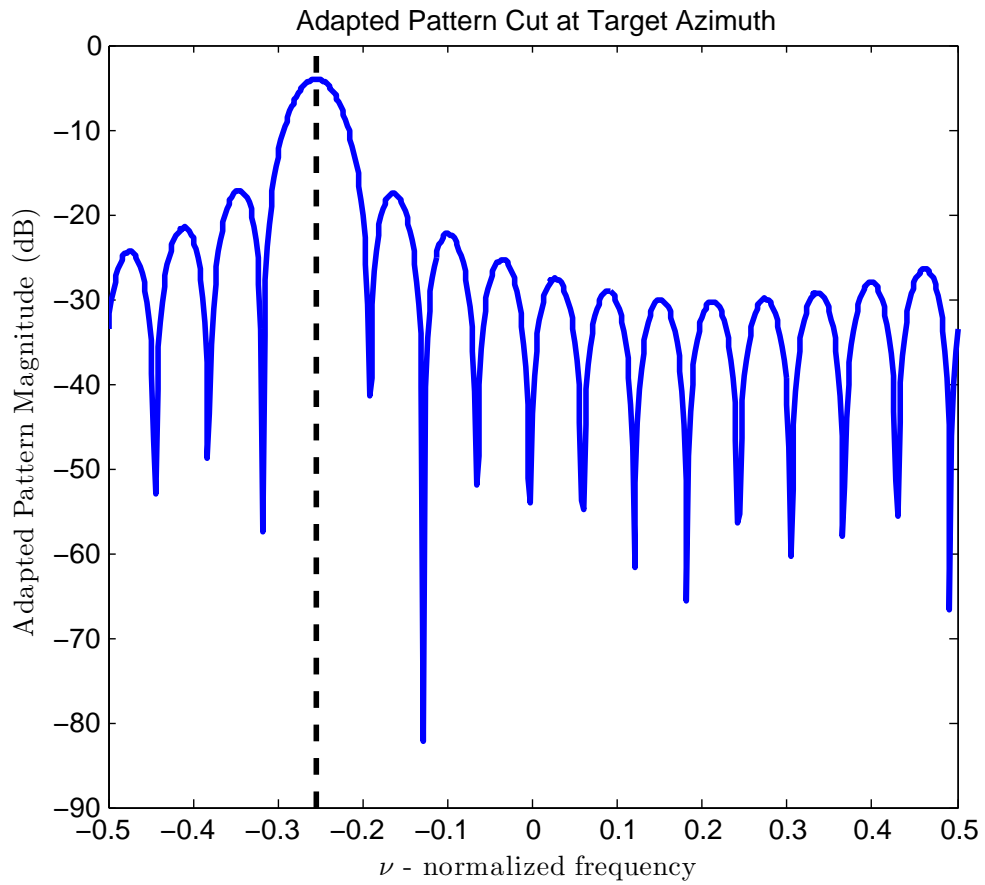
Fig. 38. Adapted Pattern (dB scale), cut at target azimuth, NormSolver. Interference.