Chapter 7

Radar waveform design via the majorization–minimization framework

Linlong Wu[1] *and Daniel P. Palomar*[1]

7.1 Introduction

A multiple-input–multiple-output (MIMO) radar transmits several probing waveforms simultaneously from its transmit antennas. It can improve the interference rejection capability and parameter identifiability as well as provide flexibility for transmit beam pattern design [1]. In addition, a cognitive approach for radar systems, which capitalizes on the information obtained from the surrounding environment or the prior knowledge stored in the platform, was proposed [2]. The significance of MIMO radar and the cognitive approach has recently motivated active research into the waveform design. A well-designed waveform can allow for more accurate detection and estimation. However, we are facing several challenges when designing radar waveform:

1. Radar system is an application with extreme emphasis on real time, which requires a fast implementation of waveform design. Moreover, for a MIMO radar, the simultaneously transmitted waveforms are independent from each other, which requires more computation cost compared with the traditional radars. In addition, if the radar system runs in a cognitive manner, the adaptability is achieved by redesigning the waveform before the next transmission, which means a short time window for the design phase.
2. With the consideration of hardware configuration and application scenarios, some waveform constraints are necessarily incorporated. However, these constraints probably raise the design difficulty. For example, in [3], closed-form solutions can be found when only the finite-energy constraint is considered. If other constraints are considered, then it is not as easy as the former case.

Thus, for radar waveform design, an algorithmic framework is desired that it is not only computationally efficient but also flexible so that various waveform constraints can be handled. One type of the promising solutions is the derived algorithms based on the MM method. The MM method refers to the majorization–minimization or

[1] Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology (HKUST), Hong Kong, China

the minorization–maximization method, which is a general algorithmic framework to solve the original problem via iteratively solving a sequence of simpler problems. The idea of MM dates back to the 1970s [4] and started taking shape as a general algorithm framework around 2000 [5–7]. Many existing algorithms can be interpreted from the MM's perspective, and it has already been successfully applied to a large number of problems and applications. Recently, some works have applied the MM method successfully on radar waveform design problems [8–12]. The results presented in these papers have shown that the algorithms based on the MM method are competitive alternatives compared with the benchmark algorithms for those design problems.

This chapter presents, in a self-contained manner, the MM framework and its applications to two radar waveform design problems. Through these design problems, several key issues of applying MM and its properties are illustrated. We hope this self-contained chapter can introduce you to this useful algorithmic framework on solving waveform design problems. The rest of the chapter is organized as follows. In Section 7.2, we give a relatively complete description of the MM method. In Section 7.3, we apply the MM method to the joint design of transmit waveform and receive filter. In Section 7.4, we apply the extension of the vanilla MM method to the robust joint design problem. Finally, conclusions are given in Section 7.5.

7.2 Preliminaries: the MM method

The MM framework is a powerful optimization scheme, especially when the problem is hard to tackle directly. The idea behind the MM algorithm is to convert the original problem into a sequence of simpler problems to be solved until convergence. The key to using the MM method is to construct a simple majorized or minorized problem that can be solved efficiently. This section serves as a concise description of the MM method, which includes the general procedure of MM, convergence analysis, acceleration schemes, and the extension to the maximin case. Interested readers may refer to [13], and references therein for more details.

7.2.1 The vanilla MM method

Consider a general optimization problem:

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\mathsf{min}} & f\left(\boldsymbol{x}\right) \\ \mathsf{subject\ to} & \boldsymbol{x} \in \mathscr{X}, \end{array} \tag{7.1}$$

where $\mathscr{X}$ is a nonempty closed set and $f : \mathscr{X} \rightarrow \mathbb{R}^1$ is a continuous function. Notice that the convexity of the feasible set $\mathscr{X}$ and the objective function f is not assumed.

Initialized at $\boldsymbol{x}^{(0)} \in \mathscr{X}$, the MM method generates a sequence of feasible points $\{\boldsymbol{x}^{(\ell)}\}_{\ell\in\mathbb{N}}$ according to the following update rule:

$$\boldsymbol{x}^{(\ell+1)} \in \underset{\boldsymbol{x}\in\mathscr{X}}{\operatorname{argmin}}\, u\left(\boldsymbol{x},\boldsymbol{x}^{(\ell)}\right), \tag{7.2}$$

Algorithm 1: Majorization–minimization (MM)

```
Require: Initial point $\boldsymbol{x}^{(0)}$
Ensure: Converged point $\boldsymbol{x}$
$\ell = 0$
repeat
    Construct a majorizer $u\left(\boldsymbol{x}, \boldsymbol{x}^{(\ell)}\right)$
    $\boldsymbol{x}^{(\ell+1)} \in \underset{\boldsymbol{x} \in \mathscr{X}}{\arg\min}\, u\left(\boldsymbol{x}, \boldsymbol{x}^{(\ell)}\right)$
    $\ell \leftarrow \ell + 1$
until convergence
```

where $\boldsymbol{x}^{(\ell)}$ is the point generated by the MM method at the ℓth iteration, and $u\left(\boldsymbol{x}, \boldsymbol{x}^{(\ell)}\right)$ is a majorizing function (majorizer) of $f(\boldsymbol{x})$ at $\boldsymbol{x}^{(\ell)}$, which should satisfy at least the following two properties:

$$u(\boldsymbol{x}, \boldsymbol{y}) \geq f(\boldsymbol{x}), \text{ for } \forall \boldsymbol{x}, \boldsymbol{y} \in \mathscr{X} \tag{7.3}$$

$$u(\boldsymbol{y}, \boldsymbol{y}) = f(\boldsymbol{y}), \text{ for } \forall \boldsymbol{y} \in \mathscr{X}. \tag{7.4}$$

In other words, the majorizer $u\left(\boldsymbol{x}, \boldsymbol{x}^{(\ell)}\right)$ should be a global upper bound of $f(\boldsymbol{x})$ and coincide with $f(\boldsymbol{x})$ at $\boldsymbol{x}^{(\ell)}$.

Therefore, the algorithm based on the MM method for problem (7.1) is summarized as Algorithm 1.

One interesting and useful property of MM-based methods is the monotonicity of the generated sequence of objective values $\left\{f\left(\boldsymbol{x}^{(\ell)}\right)\right\}$:

$$f\left(\boldsymbol{x}^{(\ell+1)}\right) \leq u\left(\boldsymbol{x}^{(\ell+1)}, \boldsymbol{x}^{(\ell)}\right) \leq u\left(\boldsymbol{x}^{(\ell)}, \boldsymbol{x}^{(\ell)}\right) = f\left(\boldsymbol{x}^{(\ell)}\right), \tag{7.5}$$

where the first inequality follows from (7.3), the second one follows from (7.2), and the last equality follows from (7.4). Furthermore, even if $\boldsymbol{x}^{(\ell+1)}$ is not chosen as the minimizer of $u\left(\boldsymbol{x}, \boldsymbol{x}^{(\ell)}\right)$, the monotonicity can still be guaranteed as long as it improves the function $u\left(\boldsymbol{x}^{(\ell+1)}, \boldsymbol{x}^{(\ell)}\right) \leq u\left(\boldsymbol{x}^{(\ell)}, \boldsymbol{x}^{(\ell)}\right)$.

The key to applying the MM method lies in the careful construction of the majorizer. On the one hand, the majorizer should be tight to the original function for a fast convergence speed in the sense of the number of iterations. On the other hand, it should be simple enough so that the majorized problem per iteration can be solved at low computational cost. Thus, there is clearly a trade-off between the complexity of each iteration and the number of iterations, which should be considered when constructing the majorizer.

7.2.2 Convergence analysis

Given the monotonicity of $\left\{f\left(\boldsymbol{x}^{(\ell)}\right)\right\}$, it is obvious that $\left\{f\left(\boldsymbol{x}^{(\ell)}\right)\right\}$ is guaranteed to converge as long as $f(\boldsymbol{x})$ is bounded below over the feasible set. In the following, we will focus on the convergence of $\left\{\boldsymbol{x}^{(\ell)}\right\}$. The convexity of the constraint set $\mathscr{X}$ decides which type of stationary points the MM method converges to. Suppose the constraint set $\mathscr{X}$ is convex. The majorizer must satisfy the following conditions to guarantee the convergence to a d(irectional) stationary point [14]:

$$u\left(\boldsymbol{x},\boldsymbol{y}\right) \geq f\left(\boldsymbol{x}\right), \text{ for } \forall \boldsymbol{x},\boldsymbol{y} \in \mathscr{X} \tag{7.6}$$

$$u\left(\boldsymbol{y},\boldsymbol{y}\right) = f\left(\boldsymbol{y}\right), \text{ for } \forall \boldsymbol{y} \in \mathscr{X} \tag{7.7}$$

$$u'\left(\boldsymbol{y},\boldsymbol{y};\boldsymbol{d}\right) = f'\left(\boldsymbol{y};\boldsymbol{d}\right), \text{ for } \forall \boldsymbol{d} \text{ with } \boldsymbol{y}+\boldsymbol{d} \in \mathscr{X} \tag{7.8}$$

$$u\left(\boldsymbol{x},\boldsymbol{y}\right) \text{ is continuous on } \left(\boldsymbol{x},\boldsymbol{y}\right), \tag{7.9}$$

where $f'\left(\boldsymbol{y};\boldsymbol{d}\right) = \lim_{\lambda\to 0}\inf\left(\left(f\left(\boldsymbol{y}+\lambda\boldsymbol{d}\right) - f\left(\boldsymbol{y}\right)\right)/\lambda\right)$. In [14], it is proved that the limit point $\boldsymbol{x}^{(\infty)}$ satisfies

$$f'\left(\boldsymbol{x}^{(\infty)};\boldsymbol{d}\right) \geq 0, \text{ for } \forall \boldsymbol{d} \text{ with } \boldsymbol{x}^{(\infty)}+\boldsymbol{d} \in \mathscr{X}. \tag{7.10}$$

Thus, $\boldsymbol{x}^{(\infty)}$ is called d(irectional) stationary point.

Suppose the constraint set $\mathscr{X}$ is nonconvex. The condition (7.8) needs to be modified to guarantee the convergence:

$$u'\left(\boldsymbol{y},\boldsymbol{y};\boldsymbol{d}\right) = f'\left(\boldsymbol{y};\boldsymbol{d}\right), \text{ for } \forall \boldsymbol{d} \in \mathscr{T}_{\mathscr{X}}\left(\boldsymbol{y}\right), \tag{7.11}$$

where u and f are defined on the whole $\mathbb{R}^1$ space and $\mathscr{T}_{\mathscr{X}}\left(\boldsymbol{y}\right)$ is the Boulingand tangent cone of $\mathscr{X}$ at $\boldsymbol{y}$. Following this modification, we can prove that the limit point $\boldsymbol{x}^{(\infty)}$ satisfies

$$f'\left(\boldsymbol{x}^{(\infty)};\boldsymbol{d}\right) \geq 0, \text{ for } \forall \boldsymbol{d} \in \mathscr{T}_{\mathscr{X}}\left(\boldsymbol{x}^{(\infty)}\right). \tag{7.12}$$

Thus, $\boldsymbol{x}^{(\infty)}$ is called the B(oulingand) stationary point [15,16].*

7.2.3 Acceleration schemes

In some cases, if the majorizer is ill-constructed in the sense that it yields a closed-form solution but at the cost of slow convergence speed, some acceleration technique has to be adopted as a remedy. Luckily, the MM algorithm can be implemented with some acceleration techniques without loss of convergence.† Different types of accelerators have been proposed in the literature based on line search [18], inexact descent [19], and fixed-point equations [20,21]. Among them, the accelerator named SQUAREM (squared iterative method) was proposed by Varadhan and Roland [20] and can be easily implemented as an "off-the-shelf" accelerator for the MM algorithm.

*Following the notations and problem settings of [14], the proof mostly follows that of [14, Theorem 1] with only "$z + d \in \mathscr{X}$" replaced by $d \in \mathscr{T}_{\mathscr{X}}\left(z\right)$, which is based on [17, Proposition 4.7.1].

†The proof of the convergence of the accelerated MM mostly follows that of [14, Theorem 1] with slight modifications on one equation: (following the notations and problem settings of [14]) $u\left(x,x^{r_j}\right) \geq u\left(\mathrm{MM}\left(x^{r_j}\right),x^{r_j}\right) \geq f\left(\mathrm{MM}\left(x^{r_j}\right)\right) \geq f\left(x^{r_j+1}\right) \geq f\left(x^{r_{j+1}}\right) \geq u\left(x^{r_{j+1}},x^{r_{j+1}}\right)$, where $MM\left(\cdot\right)$ is the MM algorithm mapping and x^{r_j+1} is the next iteration point found by the acceleration technique. Thus, every limit point of the generated sequence by the accelerated MM algorithm is a stationary point of the solving problem.

Algorithm 2: MM accelerated via SQUAREM

```
Require: Initial point $\boldsymbol{x}^{(0)}$
Ensure: Converged point $\boldsymbol{x}$
$\ell = 0$
repeat
    $\boldsymbol{x}_0 = \boldsymbol{x}^{(\ell)}$
    $\boldsymbol{x}_1 = \mathscr{F}_{MM}(\boldsymbol{x}_0)$
    $\boldsymbol{x}_2 = \mathscr{F}_{MM}(\boldsymbol{x}_1)$
    $\boldsymbol{r} = \boldsymbol{x}_1 - \boldsymbol{x}_0$
    $\boldsymbol{v} = (\boldsymbol{x}_2 - \boldsymbol{x}_1) - \boldsymbol{r}$
    $\alpha = -\|\boldsymbol{r}\|_2 / \|\boldsymbol{v}\|_2$
    $\boldsymbol{x} = \boldsymbol{x}_1 - 2\alpha\boldsymbol{r} + \alpha^2\boldsymbol{v}$
    $\boldsymbol{x}^{(\ell+1)} = \mathscr{F}_{MM}(\boldsymbol{x})$
    $\ell \leftarrow \ell + 1$
until convergence
```

Let $\mathscr{F}_{MM}(\cdot)$ denote the fixed-point iteration map of the MM algorithm. Then, the MM updating scheme can be expressed as $\boldsymbol{x}^{(\ell+1)} \in \mathscr{F}_{MM}\left(\boldsymbol{x}^{(\ell)}\right)$. Define $\mathscr{F}(\boldsymbol{x}) = \mathscr{F}_{MM}(\boldsymbol{x}) - \boldsymbol{x}$, a Newton step update of finding a zero of $\mathscr{G}$ is given by

$$\boldsymbol{x}^{(\ell+1)} = \boldsymbol{x}^{(\ell)} - \left(\nabla\mathscr{F}\left(\boldsymbol{x}^{(\ell)}\right)\right)^{-1}\mathscr{F}\left(\boldsymbol{x}^{(\ell)}\right), \tag{7.13}$$

where $\nabla\mathscr{F}$ is the Jacobian of $\mathscr{F}$, which will be further approximated via the secant method. Combining this approximate update rule with the Cauchy–Barzilai–Borwein method [52], we obtain the SQUAREM given in Algorithm 2.

7.2.4 Extension to the maximin case

We have illustrated the MM framework by considering a minimization problem. For a maximization problem, the above results can be directly applied with simple modifications. Specifically, the update rule (7.2) should be replaced by

$$\boldsymbol{x}^{(\ell+1)} \in \underset{\boldsymbol{x}\in\mathscr{X}}{\operatorname{argmax}}\; u\left(\boldsymbol{x}, \boldsymbol{x}^{(\ell)}\right), \tag{7.14}$$

where $u\left(\boldsymbol{x}, \boldsymbol{x}^{(\ell)}\right)$ is the minorizing function (minorizer) of $f(\boldsymbol{x})$ at $\boldsymbol{x}^{(\ell)}$ with

$$u(\boldsymbol{x}, \boldsymbol{y}) \leq f(\boldsymbol{x}), \text{ for } \forall \boldsymbol{x}, \boldsymbol{y} \in \mathscr{X} \tag{7.15}$$

$$u(\boldsymbol{y}, \boldsymbol{y}) = f(\boldsymbol{y}), \text{ for } \forall \boldsymbol{y} \in \mathscr{X}. \tag{7.16}$$

Now, we consider a maximin optimization problem:

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{max}} & \underset{i=1,\ldots,I}{\min} f_i(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in \mathscr{X}, \end{array} \tag{7.17}$$

where the objective takes the form of $F(\boldsymbol{x}) = \min_{i=1,\ldots,I} f_i(\boldsymbol{x})$ (the f_i's are assumed differentiable). One minorizer of $F(\boldsymbol{x})$ is given by

$$U(\boldsymbol{x},\boldsymbol{y}) = \min_{i=1,\ldots,I} u_i(\boldsymbol{x},\boldsymbol{y}), \tag{7.18}$$

with each u_i being a tight lower bound of f_i, satisfying: $\forall i$,

$$u_i(\boldsymbol{x},\boldsymbol{y}) \le f_i(\boldsymbol{x}), \text{ for } \forall \boldsymbol{x},\boldsymbol{y} \in \mathscr{X} \tag{7.19}$$

$$u_i(\boldsymbol{y},\boldsymbol{y}) = f_i(\boldsymbol{y}), \text{ for } \forall \boldsymbol{y} \in \mathscr{X} \tag{7.20}$$

$$\nabla u_i(\boldsymbol{y},\boldsymbol{y}) = \nabla f_i(\boldsymbol{y}), \text{ for } \forall \boldsymbol{y} \in \mathscr{X} \tag{7.21}$$

$$u_i(\boldsymbol{x},\boldsymbol{y}) \text{ is continuous on } (\boldsymbol{x},\boldsymbol{y}). \tag{7.22}$$

Now, we check whether $U(\boldsymbol{x},\boldsymbol{y})$ satisfies the conditions for stationarity convergence:

Checking (7.16)**:** $\forall \boldsymbol{y} \in \mathscr{X}$,

$$U(\boldsymbol{y},\boldsymbol{y}) = \min_{i=1,\ldots,I} u_i(\boldsymbol{y},\boldsymbol{y}) = \min_{i=1,\ldots,I} f_i(\boldsymbol{y}) = F(\boldsymbol{y}). \tag{7.23}$$

Checking (7.15)**:** $\forall \boldsymbol{x},\boldsymbol{y} \in \mathscr{X}$,

$$u_i(\boldsymbol{x},\boldsymbol{y}) \le f_i(\boldsymbol{x}) \Rightarrow \min_{i=1,\ldots,I} u_i(\boldsymbol{x},\boldsymbol{y}) \le \min_{i=1,\ldots,I} f_i(\boldsymbol{x}) \Rightarrow U(\boldsymbol{x},\boldsymbol{y}) \le F(\boldsymbol{x}). \tag{7.24}$$

Checking (7.8)**:** According to [22, Theorem 9.16], given $\boldsymbol{d}$, the directional derivative of U in (7.14) can be expressed as

$$U'(\boldsymbol{y},\boldsymbol{y};\boldsymbol{d}) = \max\{\langle \boldsymbol{\xi},\boldsymbol{d}\rangle : \boldsymbol{\xi} \in \partial U(\boldsymbol{y},\boldsymbol{y})\}, \tag{7.25}$$

where $\partial U(\boldsymbol{y},\boldsymbol{y}) = conv(\{\nabla u_i(\boldsymbol{y},\boldsymbol{y}) : U(\boldsymbol{y},\boldsymbol{y}) = u_i(\boldsymbol{y},\boldsymbol{y})\})$ and conv($\mathscr{A}$) is the convex hull of the set $\mathscr{A}$. We also derive the directional derivative of F:

$$F'(\boldsymbol{y};\boldsymbol{d}) = \max\{\langle \boldsymbol{\xi},\boldsymbol{d}\rangle : \boldsymbol{\xi} \in \partial F(\boldsymbol{y})\}, \tag{7.26}$$

where $\partial F(\boldsymbol{y}) = \text{conv}(\{\nabla f_i(\boldsymbol{y}) : F(\boldsymbol{y}) = f_i(\boldsymbol{y})\})$. From $U(\boldsymbol{y},\boldsymbol{y}) = F(\boldsymbol{y})$ and $u_i(\boldsymbol{y},\boldsymbol{y}) = f_i(\boldsymbol{y})$, we obtain $\{i|U(\boldsymbol{y},\boldsymbol{y}) = u_i(\boldsymbol{y},\boldsymbol{y})\} = \{i|F(\boldsymbol{y}) = f_i(\boldsymbol{y})\}$. Besides, f_i satisfies $\nabla u_i(\boldsymbol{y},\boldsymbol{y}) = \nabla f_i(\boldsymbol{y})$, $\forall i$, so $\partial U(\boldsymbol{y},\boldsymbol{y}) = \partial F(\boldsymbol{y})$. If $\mathscr{X}$ is convex, $U'(\boldsymbol{y},\boldsymbol{y};\boldsymbol{d}) = F'(\boldsymbol{y};\boldsymbol{d})$, $\forall \boldsymbol{d}$ with $\boldsymbol{y} + \boldsymbol{d} \in \mathscr{X}$. If $\mathscr{X}$ is nonconvex, $U'(\boldsymbol{y},\boldsymbol{y};\boldsymbol{d}) = F'(\boldsymbol{y};\boldsymbol{d})$, $\forall \boldsymbol{d} \in \mathscr{T}_{\mathscr{X}}(\boldsymbol{y})$.

Checking: (7.9)**:** Obvious.

Thus, the piecewise function $U(\boldsymbol{x},\boldsymbol{y})$ in (7.18) satisfies the four conditions of convergence and the limit point of (7.14), i.e., $\boldsymbol{x}^{(\infty)}$, shall be the stationary point:

- When $\mathscr{X}$ is a convex set, d-stationarity is achieved.
- When $\mathscr{X}$ is a nonconvex set, B-stationarity is achieved.

7.3 Joint design of transmit waveform and receive filter

In this section, we consider the joint design of transmit waveform and receive filter in the presence of signal-dependent interference for a colocated MIMO radar. For the joint design problem, a popular approach is to adopt the alternating optimization scheme and then resort to semidefinite relaxation (SDR) and rank-one reconstruction [23–30]. Recently, [11] has applied the MM method successfully on this problem, on which the content of this section is based. Interested readers may refer to [13], and references therein for more details. The rest of this section is organized as follows. We first introduce the system model and formulate the problem. Then, we derive the general algorithm within the MM framework and consider several constraints on the problem. Finally, we analyze the numerical performance of the proposed algorithm for each case and compare our methods with the corresponding benchmarks.

7.3.1 System model and problem formulation

Consider a colocated MIMO radar system with N_t transmit antennas and N_r receive antennas. Each transmit antenna emits a waveform $s_m(n)$, $m = 1, \ldots, N_t$, $n = 1, \ldots, N$, with N being the number of samples of each transmitted pulse. Let $\boldsymbol{s}(n) \in \mathbb{C}^{N_t \times 1}$ represent the nth sample of the N_t waveforms. Suppose the target of interest is located at the range-angle position (r_0, θ_0) with $r_0 = 0$. Thus, the signals at the receive antennas can be represented by

$$\boldsymbol{x}(n) = \alpha \boldsymbol{a}_r(\theta_0)\, \boldsymbol{a}_t(\theta_0)^T \boldsymbol{s}(n)\, e^{j2\pi(n-1)\nu_0} + \boldsymbol{d}(n) + \boldsymbol{v}(n), \tag{7.27}$$

where

- α accounts for the target radar cross section with $\mathbb{E}\left[|\alpha|^2\right] = \sigma_0^2$.
- ν_0 is the Doppler frequency of the target.
- $\boldsymbol{a}_t(\theta_0) \in \mathbb{C}^{N_t \times 1}$ denotes the steering vector and $\boldsymbol{a}_r(\theta) \in \mathbb{C}^{N_r \times 1}$ is the propagation vector. Both the transmit and receive antennas are assumed to be uniform linear arrays (ULAs) with half-wavelength separation so that

$$\boldsymbol{a}_t(\theta) = \frac{1}{\sqrt{N_t}} \left[1, e^{-j\pi 1 \sin\theta}, \ldots, e^{-j\pi(N_t-1)\sin\theta}\right]^T, \tag{7.28}$$

  and similarly for $\boldsymbol{a}_r(\theta)$.
- $\boldsymbol{d}(n)$ accounts for the superposition of K signal-dependent uncorrelated point-like interferers. Specifically, the kth interferer is located at the range-angle position (r_k, θ_k), where $r_k \in \{0, 1, \ldots, N\}$, $\theta_k \in \{0, 1, \ldots, L\} \times (2\pi/(L+1))$ with L being the number of discrete azimuth sectors. The received interfering vector $\boldsymbol{d}(n)$ can be expressed as follows:

$$\boldsymbol{d}(n) = \sum_{k=1}^{K} \alpha_k \boldsymbol{a}_r(\theta_k)\, \boldsymbol{a}_t(\theta_k)^T \boldsymbol{s}(n - r_k)\, e^{j2\pi(n-1)\nu_k}, \tag{7.29}$$

  where α_k is the complex amplitude of the kth interferer with $\mathbb{E}\left[|\alpha_k|^2\right] = \sigma_k^2$ and ν_k is its Doppler frequency.
- $\boldsymbol{v}(n)$ denotes the additive Gaussian noise with $\boldsymbol{v}(n) \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_v^2 \boldsymbol{I}\right)$.

Let $\boldsymbol{x} = \left[\boldsymbol{x}(1)^T \ldots \boldsymbol{x}(N)^T\right]^T$, $\boldsymbol{s} = \left[\boldsymbol{s}(1)^T \ldots \boldsymbol{s}(N)^T\right]^T$, and $\boldsymbol{v} = \left[\boldsymbol{v}(1)^T \ldots \boldsymbol{v}(N)^T\right]^T$. Expression (7.27) can be represented as

$$\boldsymbol{x} = \alpha \boldsymbol{A}(r_0, \theta_0)\boldsymbol{s} + \sum_{k=1}^{K} \alpha_k \boldsymbol{A}(r_k, \theta_k)\boldsymbol{s} + \boldsymbol{v}, \tag{7.30}$$

where $\boldsymbol{A}(r_k, \theta_k)$ is given by

$$\boldsymbol{A}(r_k, \theta_k) = \left[\mathrm{Diag}(\boldsymbol{p}(\nu_k)) \otimes \left(\boldsymbol{a}_r(\theta_k)\boldsymbol{a}_t(\theta_k)^T\right)\right]\boldsymbol{J}_{r_k}, \tag{7.31}$$

where $\boldsymbol{J}_{r_k}$ is the shift matrix given by

$$\boldsymbol{J}_{r_k}(m,\ n) = \begin{cases} 1, & m - n = N_t \times r_k \\ 0, & m - n \neq N_t \times r_k \end{cases} \quad (m,\ n) \in \{1, \ldots, N_t N\}^2, \tag{7.32}$$

and

$$\boldsymbol{p}(\nu_k) = \left[1, e^{j2\pi\nu_k}, \ldots, e^{j2\pi(N-1)\nu_k}\right]^T \tag{7.33}$$

with ν_k being the Doppler frequency for $k = 0, 1, \ldots, K$. Hereafter, the Doppler frequencies $\{\nu_k\}_{k=0}^{K}$ are assumed to be zero without loss of generality (i.e., both the target and the interferers are assumed to be slowly moving or stay still), and $\boldsymbol{A}(r_k, \theta_k)$ will be denoted by $\boldsymbol{A}_k$ for simplicity of notation.

Suppose linear finite impulse response receive filters $\boldsymbol{w}$ are deployed. Then, the output of the filter is given by

$$r = \boldsymbol{w}^\dagger \boldsymbol{x} = \alpha \boldsymbol{w}^\dagger \boldsymbol{A}_0 \boldsymbol{s} + \boldsymbol{w}^\dagger \sum_{k=1}^{K} \alpha_k \boldsymbol{A}_k \boldsymbol{s} + \boldsymbol{w}^\dagger \boldsymbol{v}. \tag{7.34}$$

Based on the model, the output SINR is given by

$$\mathrm{SINR} = \frac{\sigma_0^2 \left|\boldsymbol{w}^\dagger \boldsymbol{A}_0 \boldsymbol{s}\right|^2}{\boldsymbol{w}^\dagger \left(\sum_{k=1}^{K} \sigma_k^2 \boldsymbol{A}_k \boldsymbol{s}\boldsymbol{s}^\dagger \boldsymbol{A}_k^\dagger\right)\boldsymbol{w} + \sigma_v^2 \boldsymbol{w}^\dagger \boldsymbol{w}}. \tag{7.35}$$

Thus, the design problem can be formulated from the perspective of maximizing the SINR as follows:

$$\begin{aligned} \max_{\boldsymbol{s}, \boldsymbol{w}} \quad & \frac{\left|\boldsymbol{w}^\dagger \boldsymbol{A}_0 \boldsymbol{s}\right|^2}{\boldsymbol{w}^\dagger \Psi(\boldsymbol{s})\boldsymbol{w} + \boldsymbol{w}^\dagger \boldsymbol{w}} \\ \text{subject to} \quad & \boldsymbol{s} \in \mathscr{S}, \end{aligned} \tag{7.36}$$

where $\Psi(\boldsymbol{s}) = \sum_{k=1}^{K} q_k \boldsymbol{A}_k \boldsymbol{s}\boldsymbol{s}^\dagger \boldsymbol{A}_k^\dagger$ with $q_k = \sigma_k^2/\sigma_v^2 > 0$, $\mathscr{S} = \left\{\boldsymbol{s} \mid \|\boldsymbol{s}\|^2 = 1,\ \boldsymbol{s} \in \mathscr{S}_c\right\}$, with $\mathscr{S}_c \in \mathbb{C}^{NN_t}$ denoting a nonempty but not necessarily convex set.

7.3.2 MM-based method for joint design with multiple constraints

We first derive an algorithmic framework for the joint design problem based on the MM method. Then, we will consider four waveform constraints for the derived algorithmic framework. Finally, we will summarize the proposed algorithms with analysis of their computational complexity.

7.3.2.1 Majorized iteration method for joint design

For a given $\boldsymbol{s}$, problem (7.36) with respect to $\boldsymbol{w}$ can be equivalently reformulated into

$$\begin{aligned} &\min_{\boldsymbol{w}} && \boldsymbol{w}^{\dagger}\left[\Psi\left(\boldsymbol{s}\right)+\boldsymbol{I}\right]\boldsymbol{w} \\ &\text{subject to} && \boldsymbol{w}^{\dagger}\boldsymbol{A}_{0}\boldsymbol{s}=1, \end{aligned} \tag{7.37}$$

with a closed-form solution given by

$$\boldsymbol{w}^{\star}=\frac{\left[\Psi\left(\boldsymbol{s}\right)+\boldsymbol{I}\right]^{-1}\boldsymbol{A}_{0}\boldsymbol{s}}{\boldsymbol{s}^{\dagger}\boldsymbol{A}_{0}^{\dagger}\left[\Psi\left(\boldsymbol{s}\right)+\boldsymbol{I}\right]^{-1}\boldsymbol{A}_{0}\boldsymbol{s}}. \tag{7.38}$$

Substituting (7.38) into (7.36), and after some algebraic manipulations, problem (7.36) reduces to

$$\begin{aligned} &\max_{\boldsymbol{s}} && \boldsymbol{s}^{\dagger}\left(\boldsymbol{A}_{0}^{\dagger}\left[\Psi\left(\boldsymbol{s}\right)+\boldsymbol{I}\right]^{-1}\boldsymbol{A}_{0}\right)\boldsymbol{s} \\ &\text{subject to} && \boldsymbol{s}\in\mathscr{S}. \end{aligned} \tag{7.39}$$

Defining $\boldsymbol{s}=\boldsymbol{s}\boldsymbol{s}^{\dagger}$, problem (7.39) is equivalent to

$$\begin{aligned} &\min_{\boldsymbol{s},\boldsymbol{S}} && -\boldsymbol{s}^{\dagger}\left(\boldsymbol{A}_{0}^{\dagger}\left[\Psi\left(\boldsymbol{s}\right)+\boldsymbol{I}\right]^{-1}\boldsymbol{A}_{0}\right)\boldsymbol{s} \\ &\text{subject to} && \boldsymbol{s}=\boldsymbol{s}\boldsymbol{s}^{\dagger}, \quad \boldsymbol{s}\in\mathscr{S}, \end{aligned} \tag{7.40}$$

where $\Psi\left(\boldsymbol{s}\right)=\sum_{k=1}^{K}q_{k}\boldsymbol{A}_{k}\boldsymbol{s}\boldsymbol{A}_{k}^{\dagger}$.

Lemma 7.1. *Denote the objective function of problem* (7.40) *by* $f\left(\boldsymbol{s},\boldsymbol{S}\right)$. *Then,* $f\left(\boldsymbol{s},\boldsymbol{S}\right)$ *is a concave function of* **s** *and* **s** *jointly, and a majorizer of* $f\left(\boldsymbol{s},\boldsymbol{S}\right)$ *is*

$$u_{1}\left(\boldsymbol{s},\boldsymbol{S};\boldsymbol{s}_{\ell},\boldsymbol{s}_{\ell}\right)=-2\mathrm{Re}\left(\boldsymbol{z}_{\ell}^{\dagger}\boldsymbol{s}\right)+2\mathrm{Tr}\left(\boldsymbol{p}_{\ell}\boldsymbol{s}\right)-2\mathrm{Tr}\left(\boldsymbol{p}_{\ell}\boldsymbol{s}_{\ell}\right)-f\left(\boldsymbol{s}_{\ell},\boldsymbol{s}_{\ell}\right), \tag{7.41}$$

where $\boldsymbol{z}_{\ell}=\boldsymbol{A}_{0}^{\dagger}\left[\Psi\left(\boldsymbol{s}_{\ell}\right)+\boldsymbol{I}\right]^{-1}\boldsymbol{A}_{0}\boldsymbol{s}_{\ell}$, $\boldsymbol{p}_{\ell}=\sum_{k=1}^{K}q_{k}\left(\boldsymbol{Q}_{\ell}^{k}\right)^{\dagger}\boldsymbol{s}_{\ell}\boldsymbol{Q}_{\ell}^{k}$, *and* $\boldsymbol{Q}_{\ell}^{k}=\boldsymbol{A}_{0}^{\dagger}\left[\Psi\left(\boldsymbol{s}_{\ell}\right)+\boldsymbol{I}\right]^{-1}\boldsymbol{A}_{k}$.

Proof. See Appendix A. □

Ignoring the constant terms and undoing the change of variables $\boldsymbol{s}=\boldsymbol{s}\boldsymbol{s}^{\dagger}$ in the function (7.40), the first majorized problem is then given by

$$\begin{aligned} &\min_{\boldsymbol{s}} && \boldsymbol{s}^{\dagger}\boldsymbol{p}_{\ell}\boldsymbol{s}-\mathrm{Re}\left(\boldsymbol{z}_{\ell}^{\dagger}\boldsymbol{s}\right) \\ &\text{subject to} && \boldsymbol{s}\in\mathscr{S}. \end{aligned} \tag{7.42}$$

Lemma 7.2. *Let* $\boldsymbol{L}$ *be an* $n\times n$ *Hermitian matrix and* $\boldsymbol{M}$ *be another* $n\times n$ *Hermitian matrix such that* $\boldsymbol{M}\succeq\boldsymbol{L}$. *Then for any point* $\boldsymbol{x}_{0}\in\boldsymbol{C}^{n}$, *the quadratic function* $\boldsymbol{x}^{\dagger}\boldsymbol{L}\boldsymbol{x}$ *is majorized by* $\boldsymbol{x}^{\dagger}\boldsymbol{M}\boldsymbol{x}+2\mathrm{Re}\left(\boldsymbol{x}^{\dagger}\left(\boldsymbol{L}-\boldsymbol{M}\right)\boldsymbol{x}_{0}\right)+\boldsymbol{x}_{0}^{\dagger}\left(\boldsymbol{M}-\boldsymbol{L}\right)\boldsymbol{x}_{0}$ *at* $\boldsymbol{x}_{0}$ *[31].*

By using the above lemma, a majorizer of the objective function of problem (7.42) can be constructed as follows:

$$\begin{aligned} u_2(\boldsymbol{s}, \boldsymbol{S}_\ell) = &\lambda_u(\boldsymbol{P}_\ell)\boldsymbol{s}^\dagger\boldsymbol{s} + 2\mathrm{Re}\left(\boldsymbol{s}^\dagger(\boldsymbol{P}_\ell - \lambda_u(\boldsymbol{P}_\ell)\boldsymbol{I})\boldsymbol{s}_\ell\right) \\ &+ \boldsymbol{s}_\ell^\dagger(\lambda_u(\boldsymbol{P}_\ell)\boldsymbol{I} - \boldsymbol{P}_\ell)\boldsymbol{s}_\ell - \mathrm{Re}\left(\boldsymbol{z}_\ell^\dagger\boldsymbol{s}\right), \end{aligned} \tag{7.43}$$

where $\lambda_u(\boldsymbol{P}_\ell)$ is an upper bound of the eigenvalues of the positive semidefinite matrix $\boldsymbol{P}_\ell$, which could be simply chosen as $\mathrm{Tr}(\boldsymbol{P}_\ell)$ considering the computation cost. Please note that the tightness of the upper bound $\lambda_u(\boldsymbol{P}_\ell)$ affects the performance of the convergence speed. Due to $\boldsymbol{s}^\dagger\boldsymbol{s} = 1$, we have the second majorized problem:

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \mathrm{Re}\left(\boldsymbol{v}_\ell^\dagger\boldsymbol{s}\right) \\ \text{subject to} \quad & \boldsymbol{s} \in \mathscr{S}, \end{aligned} \tag{7.44}$$

where

$$\boldsymbol{v}_\ell = 2(\boldsymbol{P}_\ell - \lambda_u(\boldsymbol{P}_\ell)\boldsymbol{I})\boldsymbol{s}_\ell - \boldsymbol{z}_\ell. \tag{7.45}$$

7.3.2.2 Four waveform constraint cases

We consider four constraints on the problem: the constant modulus, the similarity, the peak-to-average power ratio (PAR), and the spectrum compatibility. For each constraint case, we will derive the closed-form solution or algorithms for the corresponding majorized problems.

Constant modulus constraint

Note that in practice, due to the limitations of hardware components (such as the maximum signal amplitude clip of A/D converters and power amplifiers), it is usually desirable to transmit constant modulus waveforms. Problem (7.44) becomes

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \mathrm{Re}\left(\boldsymbol{v}_\ell^\dagger\boldsymbol{s}\right) \\ \text{subject to} \quad & |s_n| = \frac{1}{\sqrt{NN_t}}, \text{ for } i = 1, \ldots, NN_t, \end{aligned} \tag{7.46}$$

which has the closed-form solution

$$\boldsymbol{s} = -\frac{e^{j\arg(\boldsymbol{v}_\ell)}}{\sqrt{NN_t}}, \tag{7.47}$$

where $e^{j\arg(\cdot)}$ is an element-wise operation that extracts the phase.

Similarity constraint

We will consider the similarity constraint for the first and second majorized problems. If the finite energy constraint is reduced to the constant modulus constraint, the similarity constraint can be recast as in [32], and the first majorized problem (7.42) becomes

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \boldsymbol{s}^{\dagger}\boldsymbol{P}_{\ell}\boldsymbol{s} - \operatorname{Re}\left(\boldsymbol{z}_{\ell}^{\dagger}\boldsymbol{s}\right) \\ \text{subject to} \quad & |s_n| = \frac{1}{\sqrt{NN_t}}, \quad \arg\left(s_n\right) \in \left[\gamma_n, \gamma_n + \delta\right], \end{aligned} \tag{7.48}$$

where $\gamma_n = \arg\left(\boldsymbol{s}_{\mathsf{ref}}\left(n\right)\right) - \arccos\left(1 - NN_t\epsilon^2/2\right)$ with $\boldsymbol{s}_{\mathsf{ref}}$ being the reference sequence and $\delta = 2\arccos\left(1 - NN_t\epsilon^2/2\right)$ with ϵ being the similarity parameter defined by $\|\mathbf{s} - \mathbf{s}_{\mathrm{ref}}\|_{\infty} \leq \epsilon$.

The block coordinate descent method (BCD) [33, Chapter 2] can be employed for problem (7.48). Assuming all the elements of $\boldsymbol{s}$, except s_n, are fixed, the problem of s_n is given by

$$\begin{aligned} \min_{s_n} \quad & \operatorname{Re}(a_n^* s_n) \\ \text{subject to} \quad & |s_n| = \frac{1}{\sqrt{NN_t}}, \quad \arg\left(s_n\right) \in \left[\gamma_n, \gamma_n + \delta\right], \end{aligned} \tag{7.49}$$

where $a_n = 2\sum_{i=1,i\neq n}^{NN_t} s_i P_{n,i} - z_n$ with $P_{n,i}$ being the (n,i)th entry of $\boldsymbol{P}_{\ell}$, and z_n is the nth element of $\boldsymbol{z}$. The closed-form solution to problem (7.49) is already shown in [30] and rewritten as follows:

$$s_n = \begin{cases} \frac{e^{ja_n}}{\sqrt{NN_t}} & \arg\left(a_n\right) \in \left[\gamma_n + \frac{\delta}{2} + \left(2k-2\right)\pi, \gamma_n + \left(2k-1\right)\pi\right] \\ \frac{-e^{j\arg(a_n)}}{\sqrt{NN_t}} & \arg\left(a_n\right) \in \left[\gamma_n + \left(2k-1\right)\pi, \gamma_n + \delta + \left(2k-1\right)\pi\right] \\ \frac{e^{j(a_n+\delta)}}{\sqrt{NN_t}} & \text{otherwise,} \end{cases} \tag{7.50}$$

where $\exists k \in \mathbb{Z}$, for $n = 1, 2, \ldots, NN_t$.

For the second majorized problem (7.44), we have

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \operatorname{Re}\left(\boldsymbol{v}_{\ell}^{\dagger}\boldsymbol{s}\right) \\ \text{subject to} \quad & |s_n| = \frac{1}{\sqrt{NN_t}}, \quad \arg\left(s_n\right) \in \left[\gamma_n, \gamma_n + \delta\right], \end{aligned} \tag{7.51}$$

which has a closed-form solution similar to (7.50).

PAR constraint

The PAR constraint is expressed as

$$\mathrm{PAR} = \frac{\max\limits_{1\leq n\leq NN_t}\left\{|s_n|^2\right\}}{\left(1/NN_t\right)\|\boldsymbol{s}\|^2} \leq \epsilon, \tag{7.52}$$

where ϵ is the parameter controlling the acceptable level of PAR with $1 \leq \epsilon \leq NN_t$.

Considering the PAR constraint and letting $\gamma = (\epsilon/NN_t)$, the problem becomes

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \operatorname{Re}\left(\boldsymbol{v}_{\ell}^{\dagger}\boldsymbol{s}\right) \\ \text{subject to} \quad & \|\boldsymbol{s}\|^2 = 1, \quad |s_n| \leq \sqrt{\gamma}, \end{aligned} \tag{7.53}$$

Lemma 7.3. *Without loss of generality, we assume that* $|v_1| \geq |v_2| \geq \cdots \geq |v_N|$ *and the number of nonzero elements of* $\boldsymbol{v}$ *is m. Then, the solution to problem* (7.53) *is*

$$\boldsymbol{s} = \mathscr{P}_{\mathscr{S}}(\boldsymbol{v}), \tag{7.54}$$

where

$$\begin{aligned} \mathscr{P}_{\mathscr{S}}(\cdot) = &-\left(\boldsymbol{1}_{\mathbb{R}_{\geq 0}}(1-m\gamma)\right)\sqrt{\gamma}\boldsymbol{u}_m \odot e^{jarg(\cdot)} \\ &-\left(\boldsymbol{1}_{\mathbb{R}_{<0}}(1-m\gamma)\right) min\{\beta|\boldsymbol{v}|, \sqrt{\gamma}\boldsymbol{1}\} \odot e^{jarg(\cdot)}, \end{aligned} \tag{7.55}$$

$min\{\cdot,\cdot\}$, $|\cdot|$ *and* $e^{j\arg(\cdot)}$ *are element-wise operations,*

$$\boldsymbol{1}_A(x) = \begin{cases} 1, & if\, x \in A, \\ 0, & otherwise, \end{cases} \tag{7.56}$$

$$\boldsymbol{u}_m = \left[\underbrace{1,\ldots,1}_{m}, \underbrace{\sqrt{\frac{1-m\gamma}{NN_t\gamma - m\gamma}},\ldots,\sqrt{\frac{1-m\gamma}{NN_t\gamma - m\gamma}}}_{NN_t - m}\right]^T,$$

and

$$\beta \in \left\{\beta | \sum_{n=1}^{N} min\left\{\beta^2 |v_n|^2, \gamma\right\} = 1, \beta \in \left[0, \frac{\sqrt{\gamma}}{min\{|v_n| \mid |v_n| \neq 0\}}\right]\right\}. \tag{7.57}$$

Proof. A derivation has been given in [34, Algorithm 2]. □

Spectral compatibility constraint

The interference control for the coexistence has been extensively researched in cognitive radio [35,36] and also applies to the radar field. In order to control the interference brought to the coexisting telecommunication systems, the spectral compatibility constraint is imposed to introduce a trade-off between the SINR and the power spectral density (PSD). The spectral compatibility constraint is given by [37]

$$\boldsymbol{c}^{\dagger}\boldsymbol{r}\boldsymbol{c} \leq E_I, \tag{7.58}$$

where $\boldsymbol{C}$ is a transmitted coherent burst of N sub-pulses, and E_I is the maximum allowed interference; the spectral compatibility matrix is defined as

$$\boldsymbol{R} = \sum_{i=1}^{M} \omega_i \boldsymbol{R}_i, \tag{7.59}$$

where ω_i is the weight corresponding to the ith coexisting wireless network, and

$$\boldsymbol{R}_i(m,l) = \begin{cases} f^i_{upper} - f^i_{lower} & \text{if } m = l \\ \frac{e^{j2\pi f^i_2 (m-l)} - e^{j2\pi f^i_1 (m-l)}}{j2\pi(m-l)} & \text{if } m \neq l, \end{cases} \tag{7.60}$$

where f^i_{lower} and f^i_{upper} denote the lower and upper normalized frequencies for the ith wireless network, respectively. Thus, $\boldsymbol{c}^{\dagger}\boldsymbol{R}_i\boldsymbol{c}$ represents the energy of the radar system transmitted on the ith band $\left[f^{i}_{lower}, f^{i}_{upper}\right]$, and consequently, $\boldsymbol{c}^{\dagger}\boldsymbol{R}\boldsymbol{c}$ represents the total weighted energy of the sequence $\boldsymbol{C}$ transmitted on all M bands.

Recall that in the model, $\boldsymbol{s}$ consists of the N_t waveforms with $\boldsymbol{s} = \left[\boldsymbol{s}(1)^T \ldots \boldsymbol{s}(N)^T\right]^T$, where $\boldsymbol{s}(n) \in \mathbb{C}^{N_t \times 1}$ for $n = 1, \ldots, N$. Thus, the waveform transmitted by the kth antenna is given by

$$\boldsymbol{s}_k = \left[\boldsymbol{s}(1,k)^T \ldots \boldsymbol{s}(N,k)^T\right]^T \text{ for } k = 1, 2, \ldots, N_t, \tag{7.61}$$

where $\boldsymbol{s}(i,k)$ represents the kth element of $\boldsymbol{s}(i)$. The kth waveform is expressed as

$$\boldsymbol{s}_k = \left(\boldsymbol{I}_N \otimes \boldsymbol{u}_k\right)\boldsymbol{s} = \boldsymbol{u}_k\boldsymbol{s}, \tag{7.62}$$

where $\boldsymbol{u}_k = [\overbrace{0, \ldots, 0}^{k-1}, 1, \overbrace{0, \ldots, 0}^{N_t-k}]$ and $k = 1, \ldots, N_t$. Thus, the spectrum compatibility constraint for global design is expressed as

$$\boldsymbol{s}^{\dagger}\left(\boldsymbol{U}_1^{\dagger}\boldsymbol{R}\boldsymbol{U}_1 + \cdots + \boldsymbol{U}_{N_t}^{\dagger}\boldsymbol{R}\boldsymbol{U}_{N_t}\right)\boldsymbol{s} = \boldsymbol{s}^{\dagger}\widetilde{\boldsymbol{R}}\boldsymbol{s} \le E_I, \tag{7.63}$$

where $\boldsymbol{r}$ is defined in (7.59). The inequality (7.63) means that the total energy of all the N_t transmit waveforms on those M bands is no more than a threshold. Therefore, the optimization problem based on the first majorization is formulated as

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \boldsymbol{s}^{\dagger}\boldsymbol{P}_{\ell}\boldsymbol{s} - \operatorname{Re}\left(\boldsymbol{z}_{\ell}^{\dagger}\boldsymbol{s}\right) \\ \text{subject to} \quad & \|\boldsymbol{s}\|^2 = 1, \quad \boldsymbol{s}^{\dagger}\widetilde{\boldsymbol{R}}\boldsymbol{s} \le E_I, \end{aligned} \tag{7.64}$$

where the maximum allowed interference E_I is with $\lambda_{min}\left(\widetilde{\boldsymbol{R}}\right) \le E_I \le \lambda_{max}\left(\widetilde{\boldsymbol{R}}\right)$.

It is obvious that the constraint $\|\boldsymbol{s}\|^2 = 1$ can be equivalently rewritten as $1 \le \|\boldsymbol{s}\|^2 \le 1$. After this reformulation, the FPP-SCA algorithm [38] can be applied. At the kth iteration of the FPP-SCA algorithm, we need to solve the following problem:

$$\begin{aligned} \min_{\boldsymbol{s},\epsilon} \quad & \boldsymbol{s}^{\dagger}\boldsymbol{P}_{\ell}\boldsymbol{s} - \operatorname{Re}\left(\boldsymbol{z}_{\ell}^{\dagger}\boldsymbol{s}\right) + \mu\varepsilon \\ \text{subject to} \quad & \|\boldsymbol{s}\|^2 \le 1 + \varepsilon \\ & \boldsymbol{s}_k^{\dagger}\boldsymbol{s}_k - 2\operatorname{Re}\left(\boldsymbol{s}_k^{\dagger}\boldsymbol{s}\right) \le \varepsilon - 1 \\ & \boldsymbol{s}^{\dagger}\widetilde{\boldsymbol{R}}\boldsymbol{s} \le E_I \\ & \varepsilon \ge 0, \end{aligned} \tag{7.65}$$

where μ is the penalty parameter to scale the impact of the penalty term.

Note that due to the slack variable ε, problem (7.65) is always feasible and being a convex QCQP, which can be solved efficiently by off-the-shelf solvers, e.g., MOSEK [39]. In order to guarantee the feasibility of the solution to the original problem, a large parameter μ is suggested in [38] to force the slack variable toward zero. In terms of convergence, if FPP-SCA converges, it converges to the KKT point of problem (7.65). Further, if the converged slack variable ε is zero, then the remaining variable $\boldsymbol{s}$ is the KKT point of problem (7.64).

Algorithm 3: Majorized iterative algorithm (MIA)

```
Require: Initial sequence $\boldsymbol{s}_0$, convergence threshold(s) $\varepsilon_{obj}$ (and $\varepsilon_{slc}$) (and penalty
    parameters $\mu_0$, $\{\mu_0^i\}_{i=1}^{N_t}$)
Ensure: Designed sequence $\boldsymbol{s}$ and receive filter $\boldsymbol{w}$
repeat
    $\boldsymbol{z}_\ell = \boldsymbol{A}(\theta)^\dagger [\Psi(\boldsymbol{s}_\ell) + \boldsymbol{I}]^{-1} \boldsymbol{a}(\theta) \boldsymbol{s}_\ell$
    $\boldsymbol{P}_\ell = \sum_{k=1}^{K} q_k \left(\boldsymbol{Q}_\ell^k\right)^\dagger \boldsymbol{s}_\ell \boldsymbol{Q}_\ell^k$
    (I) Constant modulus:
        update $\lambda_u(\boldsymbol{P}_\ell)$
        $\boldsymbol{v}_\ell = 2(\boldsymbol{P}_\ell - \lambda_u(\boldsymbol{P}_\ell)\boldsymbol{I})\boldsymbol{s}_\ell - \boldsymbol{z}_\ell$
        obtain $\boldsymbol{s}_{\ell+1}$ according to (7.47)
    (II-i) Similarity based on the 1st majorization:
        for $n = 1$ to $NN_t$
            $a_n = 2\sum_{i=1, i \neq n}^{NN_T} s_i P_{n,i} - z_n$
            obtain $s_n$ according to (7.50)
        end
        $\boldsymbol{s}_{\ell+1} = \left[s_1, \ldots, s_{NN_t}\right]^T$
    (II-ii) Similarity based on the 2nd majorization:
        update $\lambda_u(\boldsymbol{P}_\ell)$
        $\boldsymbol{v}_\ell = 2(\boldsymbol{P}_\ell - \lambda_u(\boldsymbol{P}_\ell)\boldsymbol{I})\boldsymbol{s}_\ell - \boldsymbol{z}_\ell$
        obtain $\boldsymbol{s}_{\ell+1}$ similar to (7.50)
    (III) PAR:
        update $\lambda_u(\boldsymbol{P}_\ell)$
        $\boldsymbol{v}_\ell = 2(\boldsymbol{P}_\ell - \lambda_u(\boldsymbol{P}_\ell)\boldsymbol{I})\boldsymbol{s}_\ell - \boldsymbol{z}_\ell$
        obtain $\boldsymbol{s}_{\ell+1}$ according to (7.55)
    (IV) Global design for spectrum compatibility:
        obtain $\boldsymbol{s}_{\ell+1}$ by solving the QCQP (7.65)
        if $t > \varepsilon_{slc}$ : $\mu_{\ell+1} = 2\mu_\ell$; else: $\mu_{\ell+1} = \mu_\ell$
    $\ell \leftarrow \ell + 1$
until (I,II,III):  $|f(\boldsymbol{s}_\ell) - f(\boldsymbol{s}_{\ell-1})| \leq \varepsilon_{obj}$
      (IV) :       $|f(\boldsymbol{s}_\ell) - f(\boldsymbol{s}_{\ell-1})| \leq \varepsilon_{obj}$ & $t \leq \varepsilon_{slc}$
$\boldsymbol{w} = \frac{[\Psi(\boldsymbol{s}_\ell)+\boldsymbol{I}]^{-1}\boldsymbol{a}(\theta)\boldsymbol{s}_\ell}{\boldsymbol{s}_\ell^\dagger \boldsymbol{a}(\theta)^\dagger [\Psi(\boldsymbol{s}_\ell)+\boldsymbol{I}]^{-1}\boldsymbol{a}(\theta)\boldsymbol{s}_\ell}$
```

7.3.2.3 Summary of algorithm and complexity analysis

To summarize, the description of the algorithm with respect to the above constraints is given in Algorithm 3. Hereafter, we use **MIA-CMC**, **MIA-CMSC**, **MIA-PC**, and

MIA-SCCG to denote the proposed Majorized Iterative Algorithm with the Constant Modulus Constraint, Constant Modulus and Similarity Constraint, PAR Constraint and Spectrum Compatibility Constraint for Global design, respectively. **MIA-CMSC** can be subdivided into **MIA-CMSC1** (MIA-CMSC based on the first majorization) and **MIA-CMSC2** (MIA-CMSC based on the second majorization).

We now analyze the computational complexity of the MIA-based algorithms. The overall complexity of each MIA-XXX method is linear with respect to the number of iterations. For the convenience of analysis, we focus on the deterministic cost on a per-iteration basis, which comes from the following three sources: $\boldsymbol{z}_\ell$, $\boldsymbol{P}_\ell$, and $\boldsymbol{s}_{\ell+1}$. The computational cost of $\boldsymbol{z}_\ell$ is $\mathcal{O}\left((N_rN)^3\right)$ because of the inversion operation. With $[\Psi(\boldsymbol{s})+\boldsymbol{I}]^{-1}$ already computed, the computational cost of $\boldsymbol{Q}_\ell^k$ is $\mathcal{O}\left((N_rN)\cdot(N_tN)^2\right)$, and, thus, the computational cost of $\boldsymbol{P}_\ell$ should be $\mathcal{O}\left((N_rN)\cdot(N_tN)^2\right)+\mathcal{O}\left((N_tN)^3\right)$. To sum up, $\boldsymbol{z}_\ell$ and $\boldsymbol{P}_\ell$ contributes the total amount of complexity $\mathcal{O}\left(N^3\cdot(\max\{N_r,N_t\})^3\right)$, neglecting the lower order terms.

The update of $\boldsymbol{s}_{\ell+1}$ varies case by case: for MIA-CMC, MIA-CMSC2, and MIA-PAR, the computation cost mainly comes from the computation of $\lambda_u(\boldsymbol{P}_\ell)$ and $\boldsymbol{v}_\ell$. The update of $\lambda_u(\boldsymbol{P}_\ell)$ is simply chosen as $\mathrm{Tr}(\boldsymbol{P}_\ell)$, so the computation cost is $\mathcal{O}(N_tN)$. The computational cost of $\boldsymbol{v}_\ell$ is $\mathcal{O}\left((N_tN)^2\right)$. For MIA-CMSC1, there are NN_t subproblems due to the BCD scheme, and the computational cost of each subproblem is $\mathcal{O}(N_tN)$ because of the update of a_n. Thus, the total computational cost is $\mathcal{O}\left((N_tN)^2\right)$. For MIA-SCCG, MOSEK will reformulate problem (7.65) into the epigraph form by introducing one more slack variable. Then, there is one linear constraint and three second-order cone (SOC) constraints. The computational complexity of solving the problem should be upper bounded by $\mathcal{O}\left((N_tN)^{3.5}\right)$, the same order as SOC programing (SOCP).

7.3.3 Numerical experiments

Assume that both the transmitter and receiver are ULAs with half-wavelength separation, $N_t=8$, $N_r=8$, and $N=20$. A target is located at the range-angle position $(0,15°)$ with power $|\alpha_0|^2=20$ dB, and three fixed interferers are located at the range-angle positions $(0,-50°)$, $(1,-10°)$, and $(2,40°)$, respectively. The power for each interferer is $|\alpha_i|^2=20$ dB, for $i=1,2,3$. The noise variance is $\sigma_v^2=0$ dB. The orthogonal linear frequency modulation (LFM) waveforms are set as the initial and also the reference waveforms for MIA-CMC, MIA-CMSC, and MIA-PC. Denote the space-time sequence matrix of the LFM waveform by $\boldsymbol{S}_0$. The (k,n)th entry of $\boldsymbol{S}_0$ is given by

$$\boldsymbol{S}_0(k,n)=\frac{1}{\sqrt{NN_t}}\exp\{j2\pi(n-1)(k+n-1)/N\}, \tag{7.66}$$

where $k=1,\dots,N_t$ and $n=1,\dots,N$. The initial sequence $\boldsymbol{S}_0\in\mathbb{C}^{NN_t\times 1}$ is obtained by stacking the columns of $\boldsymbol{s}_0$. The initial filter $\boldsymbol{w}_0$ is obtained according to (7.38) by using $\boldsymbol{s}_0$. Unless otherwise specified, all the parameters are the same in the numerical experiments. In the following experiment, we also implement the accelerated version of the corresponding MIA-type method, denoted by MIA-XXX-Accelerated.

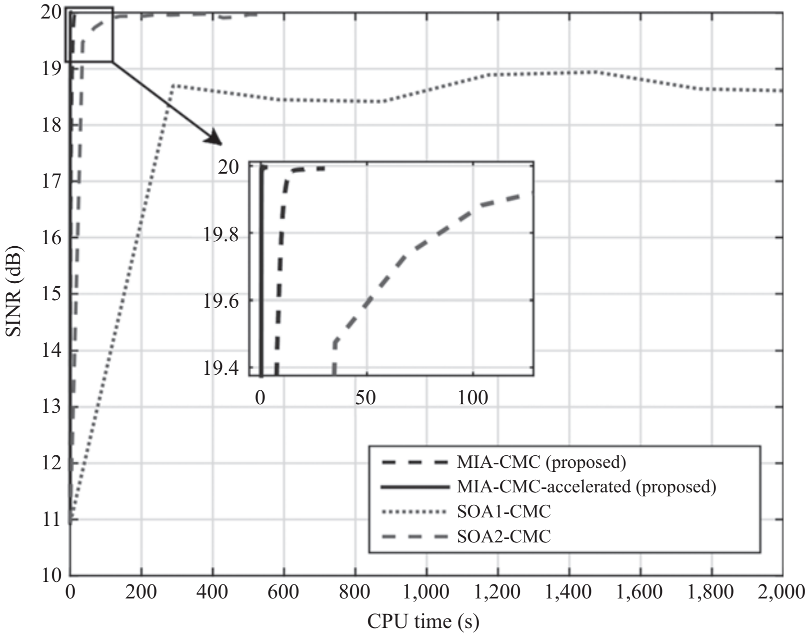

Figure 7.1 Convergence plot: SINR versus CPU time for the constant modulus case

SQUAREM [20], as an off-the-shelf acceleration scheme, is deployed as the acceleration scheme. All experiments were carried out on a Windows laptop with a 2.60 GHz i7-5600U CPU and 8 GB RAM.

7.3.3.1 Joint design with the constant modulus constraint

The benchmark methods for the constant modulus constraint are SOA1-CMC (sequential optimization algorithm 1 with constant modulus constraint) and SOA2-CMC (sequential optimization algorithm 2 with constant modulus constraint) [23]. Both of them employ the SDR and randomization to solve the rank-one constrained SDP problems iteratively. In Figure 7.1, we can see clearly that an obvious advantage of the MIA-type methods is the guarantee of monotonicity, compared with the fluctuation of the SINR achieved by the SOA methods. Besides, both MIA-CMC and MIA-CMC-Accelerated are much faster than the SOA-type methods in terms of the CPU time.

7.3.3.2 Joint design with the similarity constraint

The benchmarks are SOA1-CMSC (sequential optimization algorithm 1 with constant modulus and similarity constraints), SOA2-CMSC (sequential optimization algorithm 2 with constant modulus and similarity constraints) in [23], and Algorithm 2 in [40] named ALT-DB. SOA1-CMSC and SOA2-CMSC are similar to SOA1-CMC and

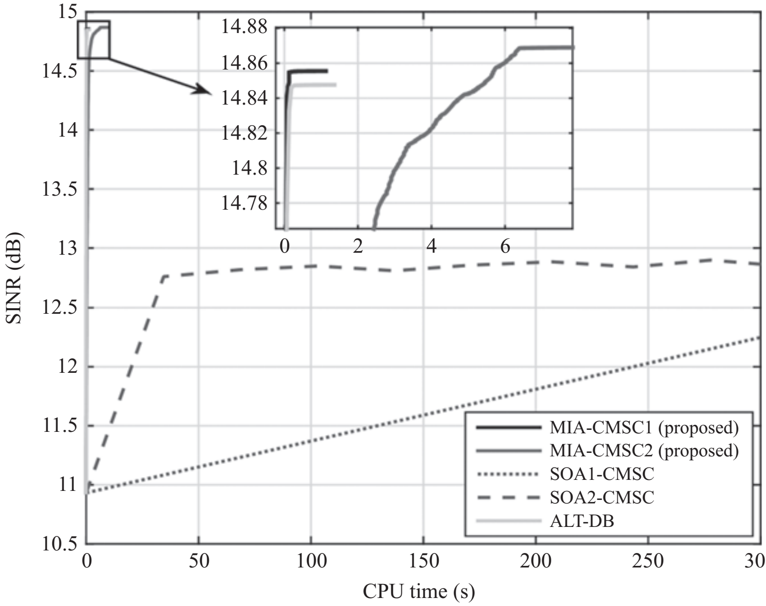

Figure 7.2 Convergence plot: SINR versus CPU time for the similarity case. Similarity parameter: $\epsilon = 1/\sqrt{NN_t}$

SOA2-CMC, respectively. ALT-DB deploys the alternating scheme, and for the problem with respect to $\boldsymbol{s}$, the BCD method is used within the Dinkelbach framework. From Figure 7.2, we can see that the MIA-type methods and ALT-DB are better than the SOA-type methods in terms of both CPU time and converged SINR. Further, MIA-CMSC2 is slightly better than ALT-DB in terms of the achieved SINR with almost the same performance on CPU time.

7.3.3.3 Joint design with the PAR constraint

The SOA1-type method is modified and named SOA1-PC to be a benchmark for the PAR case. The other benchmark is the method proposed in [41] named ALT-SDP here. ALT-SDP can be decomposed into two steps: SDPs of $\boldsymbol{w}$ and $\boldsymbol{s}$ are solved alternately in the first step, and $\boldsymbol{w}$ and $\boldsymbol{s}$ are synthesized by randomization in the second step. In the experiment about computation efficiency, we only consider the first step of ALT-SDP, where the SDP of $\boldsymbol{w}$ is replaced by the closed-form solution of $\boldsymbol{w}$. Note that the SINR achieved after the second step may be smaller than that obtained in the first step, and the running time should include the time for the synthesis stage. Figure 7.3 shows that the converged SINR achieved by MIA-PC is about 3 dB higher than both SOA1-PC and ALT-SDP. In addition, MIA-PC converges much faster in terms of CPU time.

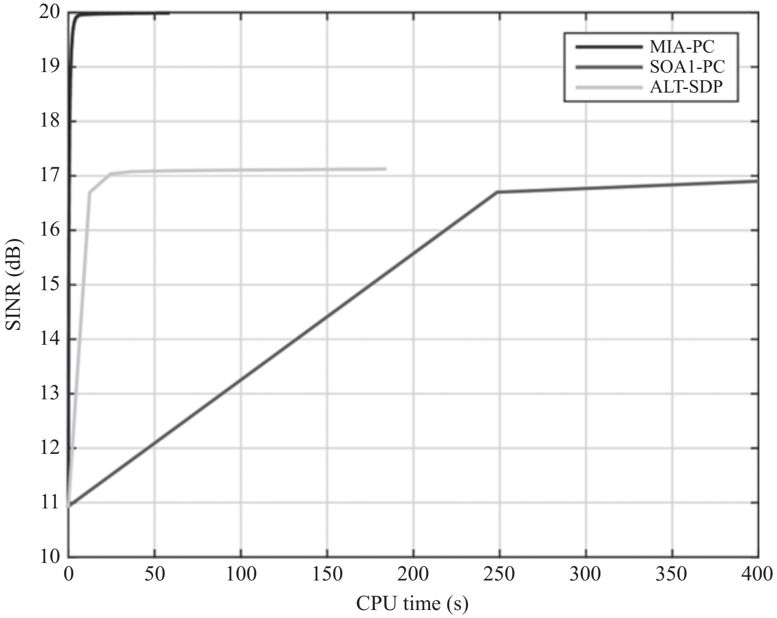

Figure 7.3 Convergence plot: SINR versus CPU time for the PAR case. PAR parameter: $\gamma = 0.5$

7.3.3.4 Joint design with the spectrum compatibility constraint

The experiment settings are as follows: $N = 40$, $N_t = 2$, and $N_r = 4$. The first frequency interval is $\left[f^1_{lower}, f^1_{upper}\right] = [0.2, 0.3]$, together with the spectral compatibility matrix $\boldsymbol{R}_1$ and the maximum allowed interference E_1. The second frequency interval is $\left[f^2_{lower}, f^2_{upper}\right] = [0.75, 0.85]$, together with the spectral compatibility matrix $\boldsymbol{R}_2$ and the maximum allowed interference E_2. For global design, the spectral compatibility matrix is $\widetilde{\boldsymbol{R}} = \sum_{k=1}^{N_t} \boldsymbol{U}_k^\dagger \boldsymbol{R}_k \boldsymbol{U}_k$, and the total allowed interference is E_I. The benchmark is the method proposed in [42] named ARCO here. Figure 7.4 shows the convergence curves and the PSD comparison of the designed waveforms. Our method is faster than the benchmark although both methods converge to almost the same SINR. Two deep nulls of the PSD designed by MIA-SCCG are much deeper than the counterparts designed by the benchmark.

7.4 Robust joint design for the worst-case SINR maximization

In the previous section, we formulate problem as a maximization of SINR. Further, the maximization of the minimum of SINR is also of interest in radar signal processing, where the maximin metric aims at ensuring the worst-case performance guarantee. In radar target detection, many works of waveform design are based on

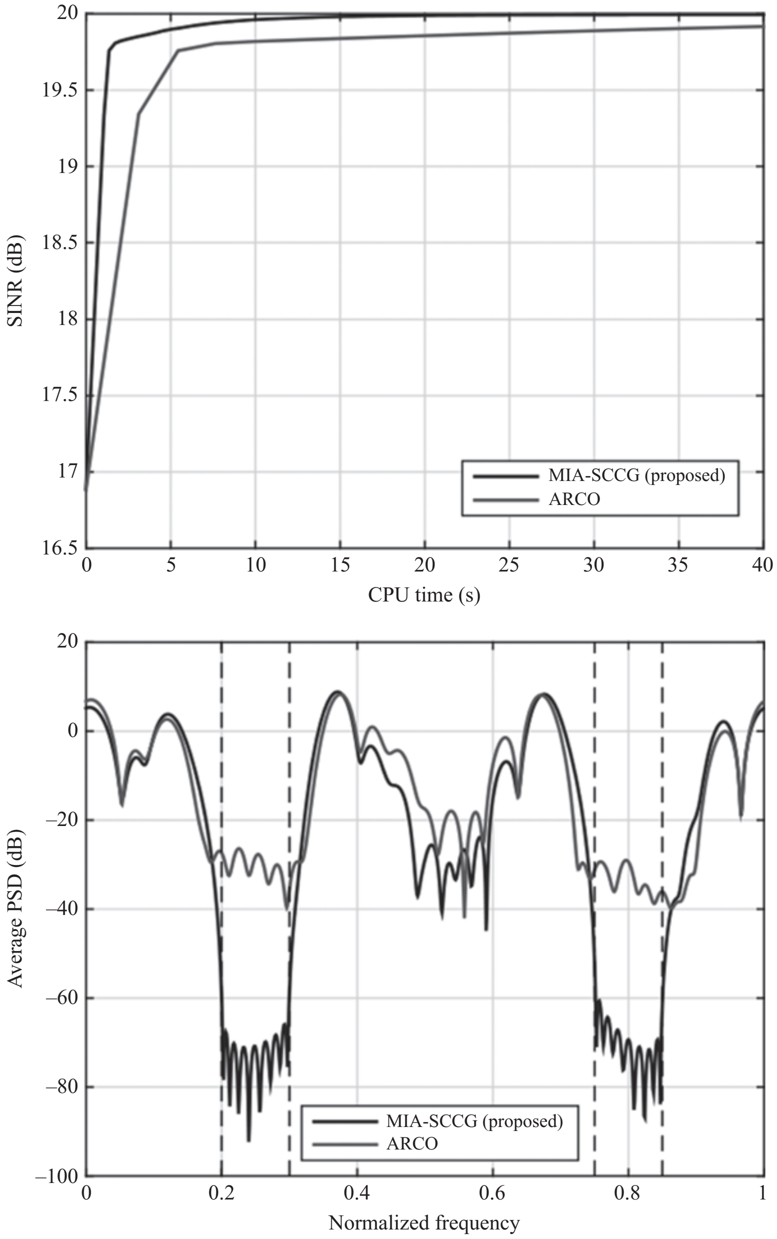

Figure 7.4 Convergence plot (left) and designed average PSD (right) for spectrum compatibility case. Parameter settings: $E_I = |\lambda_{max}(\boldsymbol{r}_1) + \lambda_{min}(\boldsymbol{r}_1)| \times 10^{-4} + |\lambda_{max}(\boldsymbol{r}_2) + \lambda_{min}(\boldsymbol{r}_2)| \times 10^{-4}$

either known Doppler shifts [3,43,44] or signal-independent interference [37,45,46]. In practice, Doppler shifts are often unknown, especially when the detection process has just been launched and the target has not yet been tracked. The assumption of signal-independent interference fails to take into account possible reflections of transmitting signals from other objects (hence signal-dependent interference). One pioneering work combining these two considerations is [26], which proposed a novel algorithm, DESIDE, to conduct the maximin optimization. The DESIDE algorithm is cyclic and SDR-based. The more recent work [27] improved the design of [26] by incorporating a filter bank. Apart from the problems with unknown Doppler, there exists a similar problem in a colocated MIMO radar system [23,47–49]. Both signal-dependent interference and some uncertainty in the target angle are considered. For the introduced works above, the optimization problems take basically the same form as follows:

$$\begin{aligned} &\max_{\boldsymbol{s}\,(\text{or }\boldsymbol{s}_i),\,\boldsymbol{w}_i} \min_{i=1,2,\ldots,I} \text{SINR}_i \\ &\text{subject to } \boldsymbol{s}\ (\text{or } \boldsymbol{s}_i) \in \mathscr{S}, \end{aligned} \tag{7.67}$$

where $\boldsymbol{s}$ (or $\boldsymbol{s}_i$) $\in \mathbb{C}^N$ denotes the transmitting sequence, $\boldsymbol{w}_i \in \mathbb{C}^M$ represents the receiving filter, and $\mathscr{S}$ models the constraint set.

In this section, we focus on solving the above robust design problem with consideration of uncertain parameters such as Doppler frequency. The content of this section is based on [12]. Interested readers may refer to [12], and references therein for more details. The rest of this section is organized as follows. We first specify the problem formulation and then propose the MM-based algorithmic framework for solving the maximin problem, which is followed by the numerical simulations.

7.4.1 Problem formulation

We denote the length of the transmitting sequence(s) and receiving filters as N and M, respectively. Recall that $\boldsymbol{s}$ is the transmitting sequence and $\boldsymbol{w}_i$ is the ith receiving filter of the filter bank. The SINR is expressed as (the derivation is similar to (7.38) in Section 7.3.2.1)

$$\text{SINR}_i = \frac{\alpha_i \left|\boldsymbol{w}_i^\dagger \boldsymbol{H}_i \boldsymbol{s}\right|^2}{\boldsymbol{w}_i^\dagger \boldsymbol{\Sigma}_I\left(\boldsymbol{s}\right) \boldsymbol{w}_i + \boldsymbol{w}_i^\dagger \boldsymbol{R} \boldsymbol{W}_i}. \tag{7.68}$$

with

$$\boldsymbol{\Sigma}_I\left(\boldsymbol{s}\right) = \sum_j \beta_j \boldsymbol{M}_j \boldsymbol{s}\boldsymbol{s}^\dagger \boldsymbol{M}_j^\dagger, \tag{7.69}$$

where $\alpha_i > 0$ is the parameter representing the path gain (or loss), $\boldsymbol{H}_i \in \mathbb{C}^{M\times N}$ represents the channel matrix, $\boldsymbol{\Sigma}_I\left(\boldsymbol{s}\right)$ (cf. [23,29,50]) is the interference covariance matrix with $\beta_j > 0$, and the matrix $\boldsymbol{M}_j \in \mathbb{C}^{M\times N}$ is an application-dependent constant matrix, which will be specified later.

Therefore, the problem of interest is given by

$$\begin{aligned} &\max_{\boldsymbol{s},\boldsymbol{w}_i} \quad \min_{i=1,2,\ldots,I} \frac{\alpha_i \left|\boldsymbol{w}_i^\dagger \boldsymbol{H}_i \boldsymbol{s}\right|^2}{\boldsymbol{w}_i^\dagger \boldsymbol{\Sigma}_I(\boldsymbol{s})\boldsymbol{w}_i + \boldsymbol{w}_i^\dagger \boldsymbol{R}\boldsymbol{w}_i} \\ &\text{subject to } \boldsymbol{s} \in \mathscr{S}, \end{aligned}$$

where $\mathscr{S} = \left\{\boldsymbol{s} \in \mathbb{C}^N \middle| \, \|\boldsymbol{s}\|_2 = 1, \|\boldsymbol{s}\|_\infty \le \sqrt{(\rho/N)}\right\}$ with ρ being the parameter of the PAR constraint and I is the number of the filters of the filter bank.

7.4.2 MM-based method for robust joint design

We are now prepared to present the MM algorithmic framework for solving the formulated problem. The MM method is naturally split into two stages: the minorization and the maximization, which will be illustrated in two subsections.

7.4.2.1 Minorizer construction

It is not hard to show that given $\boldsymbol{s}$, the optimal solution for $\boldsymbol{w}_i$ is (up to a positive scaling factor)

$$\boldsymbol{w}_i^\star = \frac{\left(\boldsymbol{\Sigma}_I(\boldsymbol{s}) + \boldsymbol{R}\right)^{-1} \boldsymbol{H}_i \boldsymbol{s}}{\boldsymbol{s}^\dagger \boldsymbol{H}_i^\dagger \left(\boldsymbol{\Sigma}_I(\boldsymbol{s}) + \boldsymbol{R}\right)^{-1} \boldsymbol{H}_i \boldsymbol{s}}. \tag{7.70}$$

Then the original problem (7.67) is reduced to

$$\begin{aligned} &\max_{\boldsymbol{s}\in\mathbb{C}^N} \quad \min_{i=1,2,\ldots,I} \mathrm{SINR}_i(\boldsymbol{s}) \\ &\text{subject to } \boldsymbol{s} \in \mathscr{S}, \end{aligned} \tag{7.71}$$

where

$$\mathrm{SINR}_i(\boldsymbol{s}) = \alpha_i \boldsymbol{s}^\dagger \boldsymbol{H}_i^\dagger \left(\boldsymbol{\Sigma}_I(\boldsymbol{s}) + \boldsymbol{R}\right)^{-1} \boldsymbol{H}_i \boldsymbol{s}. \tag{7.72}$$

We already know from Section 7.2 that finding a minorizing function for $\min_{i=1,2,\ldots,I} \mathrm{SINR}_i(\boldsymbol{s})$ boils down to finding one for each $\mathrm{SINR}_i(\boldsymbol{s})$. Thus, we can focus on the expression of $\mathrm{SINR}_i(\boldsymbol{s})$ only. In the following, we are going to find a tight lower bound for $\mathrm{SINR}_i(\boldsymbol{s})$ at the current iteration value $\boldsymbol{s}^{(n)}$.

We do a change of variable: let $\boldsymbol{G}$ be $\boldsymbol{\Sigma}_I(\boldsymbol{s})$, and then $\mathrm{SINR}_i = \mathrm{SINR}_i(\boldsymbol{s}, \boldsymbol{G}) = \alpha_i \boldsymbol{s}^\dagger \boldsymbol{H}_i^\dagger (\boldsymbol{G} + \boldsymbol{R})^{-1} \boldsymbol{H}_i \boldsymbol{s}$. We can see that SINR_i is a matrix fractional function and proves to be jointly convex in $(\boldsymbol{s}, \boldsymbol{G})$, as can be seen from [51, Example 3.4]. In that sense, a simple first-order Taylor expansion with respect to $(\boldsymbol{s}, \boldsymbol{G})$ at $(\boldsymbol{s}_0, \boldsymbol{G}_0)$ gives us a tight lower bound:

$$\mathrm{SINR}_i(\boldsymbol{s}, \boldsymbol{G}) \ge \mathrm{SINR}_i(\boldsymbol{s}_0, \boldsymbol{G}_0) + 2\mathrm{Re}\left[\boldsymbol{b}_i^\dagger (\boldsymbol{s} - \boldsymbol{s}_0)\right] - \mathrm{Tr}\left(\boldsymbol{a}_i \boldsymbol{a}_i^\dagger \cdot (\boldsymbol{G} - \boldsymbol{G}_0)\right), \tag{7.73}$$

where

$$\boldsymbol{b}_i = \alpha_i \boldsymbol{H}_i^\dagger (\boldsymbol{G}_0 + \boldsymbol{R})^{-1} \boldsymbol{H}_i \boldsymbol{s}_0, \tag{7.74}$$

$$\boldsymbol{a}_i = \sqrt{\alpha_i}\, (\boldsymbol{G}_0 + \boldsymbol{R})^{-1} \boldsymbol{H}_i \boldsymbol{s}_0, \tag{7.75}$$

and $-\boldsymbol{a}_i\boldsymbol{a}_i^\dagger$ is the gradient with respect to $\boldsymbol{G}$. Now, we undo the change of variable $\boldsymbol{G} = \boldsymbol{\Sigma}_I(\boldsymbol{s})$ and let $\boldsymbol{s}_0$ be $\boldsymbol{s}^{(n)}$, the transmitting code at the nth iteration:

$$\begin{aligned}\mathrm{SINR}_i(\boldsymbol{s}, \boldsymbol{\Sigma}_I(\boldsymbol{s})) &\geq \mathrm{SINR}_i\left(\boldsymbol{s}^{(n)}, \boldsymbol{\Sigma}_I\left(\boldsymbol{s}^{(n)}\right)\right)\\ &+ 2\mathrm{Re}\left[\boldsymbol{b}_i^\dagger\left(\boldsymbol{s}-\boldsymbol{s}^{(n)}\right)\right] - \mathrm{Tr}\left(\boldsymbol{a}_i\boldsymbol{a}_i^\dagger \cdot \left(\boldsymbol{\Sigma}_I(\boldsymbol{s}) - \boldsymbol{\Sigma}_I\left(\boldsymbol{s}^{(n)}\right)\right)\right),\end{aligned} \tag{7.76}$$

where $\boldsymbol{b}_i$ and $\boldsymbol{a}_i$ should also be adjusted:

$$\boldsymbol{b}_i = \alpha_i\boldsymbol{H}_i^\dagger\left(\boldsymbol{\Sigma}_I\left(\boldsymbol{s}^{(n)}\right)+\boldsymbol{R}\right)^{-1}\boldsymbol{H}_i\boldsymbol{s}^{(n)} \tag{7.77}$$

and

$$\boldsymbol{a}_i = \sqrt{\alpha_i}\left(\boldsymbol{\Sigma}_I\left(\boldsymbol{s}^{(n)}\right)+\boldsymbol{R}\right)^{-1}\boldsymbol{H}_i\boldsymbol{s}^{(n)}. \tag{7.78}$$

Lemma 7.4 provides a minorizer of $\mathrm{SINR}_i(\boldsymbol{s}, \boldsymbol{\Sigma}_I(\boldsymbol{s}))$.

Lemma 7.4. *A minorizing function of SINR$_i$ ($\boldsymbol{s}$) at $\boldsymbol{s} = \boldsymbol{s}^{(n)}$ is given as*

$$\overline{SINR}_i\left(\boldsymbol{s},\boldsymbol{S}^{(n)}\right) = SINR_i\left(\boldsymbol{s}^{(n)}\right) + 2Re\left[\boldsymbol{c}_i^\dagger\left(\boldsymbol{s}-\boldsymbol{s}^{(n)}\right)\right] - \lambda_{u,i}\left\|\boldsymbol{s}-\boldsymbol{s}^{(n)}\right\|_2^2, \tag{7.79}$$

where

$$\boldsymbol{c}_i = \boldsymbol{b}_i - \boldsymbol{A}_i\boldsymbol{s}^{(n)}, \tag{7.80}$$

$$\boldsymbol{A}_i = \sum_j \beta_j\boldsymbol{M}_j^\dagger\boldsymbol{a}_i\boldsymbol{a}_i^\dagger\boldsymbol{M}_j \succeq \boldsymbol{0}, \tag{7.81}$$

and

$$\lambda_{u,i} = \boldsymbol{\lambda}_{max}(\boldsymbol{A}_i) > 0. \tag{7.82}$$

Proof. See Appendix B. □

Then, the minorizing function for $\min_{i=1,2,\ldots,I}\mathrm{SINR}_i(\boldsymbol{s})$ is

$$\min_{i=1,2,\ldots,I}\overline{\mathrm{SINR}}_i\left(\boldsymbol{s},\boldsymbol{S}^{(n)}\right). \tag{7.83}$$

According to the framework of the MM method, at every iteration we just need to solve the following problem:

$$\begin{aligned}&\max_{\boldsymbol{s}\in\mathbb{C}^N}\ \min_{i=1,2,\ldots,I}\overline{\mathrm{SINR}}_i\left(\boldsymbol{s},\boldsymbol{S}^{(n)}\right)\\ &\text{subject to } \boldsymbol{s}\in\mathscr{S}.\end{aligned} \tag{7.84}$$

7.4.2.2 Maximization solution pursuit

Since $\|\boldsymbol{s}\|_2 = 1$ and $\left\|\boldsymbol{s}^{(n)}\right\|_2 = 1$, we can now rewrite (7.84) as

$$\max_{\boldsymbol{s}\in\mathscr{S}}\ \min_{i=1,2,\ldots,I} d_i + 2\mathrm{Re}\left[\left(\boldsymbol{c}_i+\lambda_{u,i}\boldsymbol{s}^{(n)}\right)^\dagger\boldsymbol{s}\right], \tag{7.85}$$

where

$$d_i = \mathrm{SINR}_i\left(\boldsymbol{s}^{(n)}\right) - 2\mathrm{Re}\left[\boldsymbol{c}_i^\dagger\boldsymbol{s}^{(n)}\right] - 2\lambda_{u,i}. \tag{7.86}$$

The discrete minimum in (7.85) can be equivalently rewritten as follows:

$$\begin{aligned}&\min_{i=1,2,\ldots,I} d_i + 2\mathrm{Re}\left[\left(\boldsymbol{c}_i + \lambda_{u,i}\boldsymbol{s}^{(n)}\right)^{\dagger}\boldsymbol{s}\right]\\ &\overset{(a)}{=}\min_{\boldsymbol{p}\in\mathscr{P}}\sum_{i=1}^{I}p_i\left(d_i + 2\mathrm{Re}\left[\left(\boldsymbol{c}_i + \lambda_{u,i}\boldsymbol{s}^{(n)}\right)^{\dagger}\boldsymbol{s}\right]\right)\\ &\overset{(b)}{=}\min_{\boldsymbol{p}\in\mathscr{P}}\boldsymbol{p}^T\boldsymbol{d} + 2\mathrm{Re}\left[\left(\left(\boldsymbol{C} + \boldsymbol{s}^{(l)}\boldsymbol{\lambda}_u^T\right)\boldsymbol{p}\right)^{\dagger}\boldsymbol{s}\right],\end{aligned} \tag{7.87}$$

where (a) $\mathscr{P} = \left\{\boldsymbol{p} \in \mathbb{R}^I | \boldsymbol{1}^T\boldsymbol{p} = 1, \boldsymbol{p} \geq \boldsymbol{0}\right\}$ is a simplex and (b) $\boldsymbol{d} = [d_1, d_2, \cdots, d_I]^T$, $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_I]$, and $\boldsymbol{\lambda}_u = \left[\lambda_{u,1}, \lambda_{u,2}, \cdots, \lambda_{u,I}\right]^T$. Thus, (7.85) is equivalent to

$$\max_{\boldsymbol{s}\in\mathscr{S}}\min_{\boldsymbol{p}\in\mathscr{P}} 2\mathrm{Re}\left[\left(\left(\boldsymbol{C} + \boldsymbol{s}^{(l)}\boldsymbol{\lambda}_u^T\right)\boldsymbol{p}\right)^{\dagger}\boldsymbol{s}\right] + \boldsymbol{p}^T\boldsymbol{d}. \tag{7.88}$$

Lemma 7.5. *In problem* (7.88)*, a saddle point exists and it can be obtained from solving the relaxed problem where the nonconvex constraint set* $\mathscr{S}$ *is relaxed to*

$$\mathscr{S}_{\text{relaxed}} = \left\{\boldsymbol{s} \in \mathbb{C}^N | \, \|\boldsymbol{s}\|_2 \leq 1, \|\boldsymbol{s}\|_\infty \leq \sqrt{\frac{\rho}{N}}\right\}. \tag{7.89}$$

Proof. See Appendix C. □

Now we look into the relaxed problem:

$$\max_{\boldsymbol{s}\in\mathscr{S}_{\text{relaxed}}}\min_{i=1,2,\ldots,I} d_i + 2\mathrm{Re}\left[\left(\boldsymbol{c}_i + \lambda_{u,i}\boldsymbol{s}^{(l)}\right)^{\dagger}\boldsymbol{s}\right], \tag{7.90}$$

or, equivalently,

$$\max_{\boldsymbol{s}\in\mathscr{S}_{\text{relaxed}}}\min_{\boldsymbol{p}\in\mathscr{P}} 2\mathrm{Re}\left[\left(\left(\boldsymbol{C} + \boldsymbol{s}^{(l)}\boldsymbol{\lambda}_u^T\right)\boldsymbol{p}\right)^{\dagger}\boldsymbol{s}\right] + \boldsymbol{p}^T\boldsymbol{d}. \tag{7.91}$$

Problem (7.91) is derived from (7.88) by changing $\mathscr{S}$ to $\mathscr{S}_{\text{relaxed}}$ and problem (7.90) is a reformulation of (7.91) into the discrete minimum format. We are going to propose two approaches for solving the relaxed problem.

The first approach: If we focus on (7.90), we can solve it via an off-the-shelf solver directly. To accelerate the convergence speed of the MM algorithm, we adopt the following technique[‡]: at any iteration, say the nth iteration, we utilize the optimal

[‡] The convergence result also holds for the accelerated MM algorithm and the proof mostly follows that of [14, Theorem 1] with slight modifications on one equation: (following the notations and problem settings of [14]) $u\left(x, x^{r_j}\right) \geq u\left(\mathrm{MM}\left(x^{r_j}\right), x^{r_j}\right) \geq f\left(\mathrm{MM}\left(x^{r_j}\right)\right) \geq f\left(x^{r_j+1}\right) \geq f\left(x^{r_{j+1}}\right) = u\left(x^{r_{j+1}}, x^{r_{j+1}}\right)$, where MM () is the MM algorithm mapping and x^{r_j+1} is the next iteration point found by the acceleration technique. In this case, subsequence stationarity convergence is maintained.

Algorithm 4: Accelerated solver-based MM algorithm

```
Require: feasible $\boldsymbol{s}^{(0)}$, $n = 0$;
repeat
    Compute $\boldsymbol{d}$, $\boldsymbol{C}$, and $\boldsymbol{\lambda}_u$ (cf. (7.86), (7.80), and (7.82), respectively);
    Solve (7.90) via some off-the-shelf solver and get its optimal solution $\hat{\boldsymbol{s}}^{(n)}$;
    Apply acceleration technique (7.92) for step size $\beta$;
    $\boldsymbol{s}^{(n+1)} = \frac{\boldsymbol{s}^{(n)}+\beta\left(\hat{\boldsymbol{s}}^{(n)}-\boldsymbol{s}^{(n)}\right)}{\left\|\boldsymbol{s}^{(n)}+\beta\left(\hat{\boldsymbol{s}}^{(n)}-\boldsymbol{s}^{(n)}\right)\right\|_2}$;
    $n = n + 1$;
until convergence
```

$\boldsymbol{s}$, denoted by $\hat{\boldsymbol{s}}^{(n)}$, to provide an ascent direction, $\hat{\boldsymbol{s}}^{(n)} - \boldsymbol{s}^{(n)}$, and do the line search as [52]:

$$\begin{aligned}
&\textbf{\textit{choose}}\ \alpha\ (>1)\,;\\
&\beta = 1;\\
&\boldsymbol{s}_{\text{temp}} = \frac{\boldsymbol{s}^{(n)}+\alpha\beta\left(\hat{\boldsymbol{s}}^{(n)}-\boldsymbol{s}^{(n)}\right)}{\left\|\boldsymbol{s}^{(n)}+\alpha\beta\left(\hat{\boldsymbol{s}}^{(n)}-\boldsymbol{s}^{(n)}\right)\right\|_2};\\
&\textbf{\textit{while}}\ \min_{i=1,2,\ldots,I}\text{SINR}_i\left(\boldsymbol{s}_{\text{temp}}\right) \geq \min_{i=1,2,\ldots,I}\text{SINR}_i\left(\hat{\boldsymbol{s}}^{(n)}\right)\ \text{and}\ \left\|\boldsymbol{s}_{\text{temp}}\right\|_\infty \leq \sqrt{\tfrac{\rho}{N}}\\
&\quad \beta = \alpha\beta;\\
&\quad \boldsymbol{s}_{\text{temp}} = \frac{\boldsymbol{s}^{(n)}+\alpha\beta\left(\hat{\boldsymbol{s}}^{(n)}-\boldsymbol{s}^{(n)}\right)}{\left\|\boldsymbol{s}^{(n)}+\alpha\beta\left(\hat{\boldsymbol{s}}^{(n)}-\boldsymbol{s}^{(n)}\right)\right\|_2};\\
&\textbf{\textit{end}}
\end{aligned} \tag{7.92}$$

The first algorithm is summarized in Algorithm 4.

The second approach: Now we focus on (7.91). The objective function in (7.91) is bilinear in $\boldsymbol{s}$ and $\boldsymbol{p}$; $\mathscr{S}_{\text{relaxed}}$ and $\mathscr{P}$ are both nonempty compact convex sets. Following the results of [53, Corollary 37.6.2 and Lemma 36.2], a saddle point exists and we can swap maximin to be minimax without affecting the solutions:

$$\min_{\boldsymbol{p}\in\mathscr{P}}\ \max_{\boldsymbol{s}\in\mathscr{S}_{\text{relaxed}}}\ 2\text{Re}\left[\left(\left(\boldsymbol{C}+\boldsymbol{s}^{(n)}\boldsymbol{\lambda}_u^T\right)\boldsymbol{p}\right)^\dagger\boldsymbol{s}\right]+\boldsymbol{p}^T\boldsymbol{d}, \tag{7.93}$$

which can be compactly rewritten as

$$\min_{\boldsymbol{p}\in\mathscr{P}}\ h\left(\boldsymbol{p}\right), \tag{7.94}$$

where

$$h\left(\boldsymbol{p}\right) = \max_{\boldsymbol{s}\in\mathscr{S}_{\text{relaxed}}}\ 2\text{Re}\left[\left(\boldsymbol{B}\boldsymbol{p}\right)^\dagger\boldsymbol{s}\right]+\boldsymbol{p}^T\boldsymbol{d} \tag{7.95}$$

and $\boldsymbol{b} = \boldsymbol{C} + \boldsymbol{s}^{(n)}\boldsymbol{\lambda}_u^T$. In particular,

- when $\rho = 1$, $\mathscr{S}_{\text{relaxed}} = \left\{\boldsymbol{s}\in\mathbb{C}^N \mid \|\boldsymbol{s}\|_\infty \leq \sqrt{1/N}\right\}$ and $h\left(\boldsymbol{p}\right) = 2\sqrt{1/N}\,\|\boldsymbol{B}\boldsymbol{p}\|_1 + \boldsymbol{p}^T\boldsymbol{d}$;
- when $\rho = N$, $\mathscr{S}_{\text{relaxed}} = \left\{\boldsymbol{s}\in\mathbb{C}^N \mid \|\boldsymbol{s}\|_2 \leq 1\right\}$ and $h\left(\boldsymbol{p}\right) = 2\,\|\boldsymbol{B}\boldsymbol{p}\|_2 + \boldsymbol{p}^T\boldsymbol{d}$.

Algorithm 5: MDA algorithm

```
Require: feasible $\boldsymbol{p}^{(0)}$, $m = 0$;
1: repeat
2:     Get subgradient: $\boldsymbol{h}^{(m)} \in \partial h\left(\boldsymbol{p}^{(m)}\right)$;
3:     $\boldsymbol{p}^{(m+1)} = \frac{\boldsymbol{p}^{(m)} \odot \exp\left(-\gamma_m \boldsymbol{h}^{(m)}\right)}{\boldsymbol{1}^T\left(\boldsymbol{p}^{(m)} \odot \exp\left(-\gamma_m \boldsymbol{h}^{(m)}\right)\right)}$;
4:     $m = m + 1$;
5: until convergence
```

We solve (7.94) via the mirror descent algorithm (MDA) [54]. Since $\mathscr{P}$ is the unit simplex, we omit the derivation details and present the MDA for problem (7.94) in Algorithm 5.

Now we are only left with computing $\boldsymbol{h}^{(m)}$ in line 2 of Algorithm 5. The subgradient $\boldsymbol{h}^{(m)}$ is given as

$$\boldsymbol{h}^{(m)} = 2\mathrm{Re}\left[\boldsymbol{B}^{\dagger}\boldsymbol{x}^{(m)}\right] + \boldsymbol{d}, \tag{7.96}$$

where $\boldsymbol{x}^{(m)} = \arg\max_{\boldsymbol{x}\in\mathscr{S}_{\text{relaxed}}} \mathrm{Re}\left[\left(\boldsymbol{B}\boldsymbol{p}^{(m)}\right)^{\dagger}\boldsymbol{x}\right]$. In particular,

- when $\rho = 1$, $\boldsymbol{x}^{(m)} = \sqrt{1/N}\left(\left|\boldsymbol{B}\boldsymbol{p}^{(m)}\right|^{-1} \odot \left[\boldsymbol{B}\boldsymbol{p}^{(m)}\right]\right)$ ($|\cdot|^{-1}$ operation is imposed element wisely);
- when $\rho = N, \boldsymbol{x}^{(m)} = \left(\boldsymbol{B}\boldsymbol{p}^{(m)}\right) / \left\|\boldsymbol{B}\boldsymbol{p}^{(m)}\right\|_2$;
- when $1 < \rho < N$, $\boldsymbol{x}^{(m)}$ follows the closed-form solution in [34, Algorithm 2]. The phases of $\boldsymbol{x}^{(m)}$ are aligned with those of $\boldsymbol{B}\boldsymbol{p}^{(m)}$. Denote the number of nonzero elements of $\boldsymbol{B}\boldsymbol{p}^{(m)}$ as K ($\leq N$), and the set containing all the corresponding indexes as $\mathscr{K}$. The solution of $\left|\boldsymbol{x}^{(m)}\right|$ is as follows:
  - if $K\rho/N \leq 1$, the solution is
    $$\left|x_n^{(m)}\right| = \begin{cases} \sqrt{\frac{\rho}{N}} & \forall n \in \mathscr{K}, \\ \sqrt{\frac{1-K\rho/N}{N-K}} & \forall n \notin \mathscr{K}; \end{cases} \tag{7.97}$$
  - if $K\rho/N > 1$, the solution is
    $$\left|\boldsymbol{x}^{(m)}\right| = \left[\eta\left|\boldsymbol{B}\boldsymbol{p}^{(m)}\right|\right]_0^{\sqrt{\rho/N}}, \tag{7.98}$$
    where η satisfies $\left\|[\eta|\boldsymbol{B}\boldsymbol{p}^{(m)}|]_0^{\sqrt{\rho/N}}\right\|_2 = 1$ ($|\cdot|$ and $[\cdot]_a^b$ denote the element-wise absolute value and the element-wise projection onto $[a, b]$, respectively). Observing that $g(\eta) = \left\|[\eta|\boldsymbol{B}\boldsymbol{p}^{(m)}|]_0^{\sqrt{\rho/N}}\right\|_2$ is a strictly increasing function on $\left[0, \left((\sqrt{\rho/N})/\left(\min_{n\in\mathscr{K}}\left\{\left|\left(\boldsymbol{B}\boldsymbol{p}^{(m)}\right)_n\right|\right\}\right)\right)\right]$, there is a unique η satisfying $g(\eta) = 1$.

The second algorithm is finally summarized in Algorithm 6.

7.4.2.3 Computational complexity

Now, we discuss the computational complexity of Algorithm 4 and 6. The only difference between the two algorithms is the way they solve the subproblem (7.84). We

Algorithm 6: Accelerated MDA-based MM algorithm

```
Require: feasible $\boldsymbol{s}^{(0)}$, $n = 0$;
repeat
    Compute $\boldsymbol{d}$, $\boldsymbol{C}$, and $\boldsymbol{\lambda}_u$ (cf. (7.86), (7.80), and (7.82), respectively);
    Solve (7.94) via MDA (Algorithm 5) for $\boldsymbol{p}^\star$ and $\hat{\boldsymbol{s}}^{(n)}$;
    Apply acceleration technique (7.92) for step size $\beta$;
    $\boldsymbol{s}^{(n+1)} = \frac{\boldsymbol{s}^{(n)}+\beta(\hat{\boldsymbol{s}}^{(n)}-\boldsymbol{s}^{(n)})}{\|\boldsymbol{s}^{(n)}+\beta(\hat{\boldsymbol{s}}^{(n)}-\boldsymbol{s}^{(n)})\|_2}$;
    $n = n + 1$;
until convergence
```

analyze the computational complexity on a per-iteration basis or, more precisely, on a per-outer-iteration basis. For analytical convenience, we focus on the deterministic cost only. The deterministic computational cost mainly comes from two sources: (1) computing $\boldsymbol{d}$, $\boldsymbol{C}$, and $\boldsymbol{\lambda}_u$ and (2) solving the simple convex problem (7.90) or (7.91). We assume M and N are of the same order ($\boldsymbol{H}_i, \boldsymbol{M}_j \in \mathbb{C}^{M\times N}$).

First, we look into the computation of $\boldsymbol{d}$, $\boldsymbol{C}$, and $\boldsymbol{\lambda}_u$ (cf. (7.86), (7.80), and (7.82), respectively). The most costly operation in computing one element of $\boldsymbol{d} \in \mathbb{R}^I$ and one column of $\boldsymbol{C} \in \mathbb{C}^{N\times I}$ needs $\mathcal{O}\left(N^3\right)$ because of $\left(\boldsymbol{\Sigma}_I\left(\boldsymbol{s}^{(n)}\right) + \boldsymbol{r}\right)^{-1}$, so the overall complexity is $\mathcal{O}\left(IN^3\right)$. Recall that $\lambda_{u,i} = \boldsymbol{\lambda}_{\max}(\boldsymbol{a}_i)$, where $\boldsymbol{a}_i \succeq \boldsymbol{0}$. The computation of $\lambda_{u,i}$ can be replaced by $\operatorname{Tr}(\boldsymbol{a}_i)$ because, first, this change does not violate any of the inequalities in the algorithm design and, second, computing $\operatorname{Tr}(\boldsymbol{a}_i)$ is very efficient, only $\mathcal{O}(N)$. So, the overall cost is $\mathcal{O}(IN)$. To this moment, the first source contributes a total amount of complexity $\mathcal{O}\left(IN^3\right)$, neglecting the lower order terms.

Next, we move on to the simple convex problem. An off-the-shelf solver, e.g., MOSEK [39], will reformulate the problem into the epigraph form with one more slack variable. Thus, we have I linear constraints. The ℓ_2- and ℓ_∞-norm constraints can be rewritten as SOC constraints: (1) ℓ_2: $\|\boldsymbol{s}\|_2 \leq 1$ and (2) ℓ_∞: $\forall n$, $|s_n| \leq \sqrt{\rho/N} \Longrightarrow \|[\operatorname{Re}[s_n], \operatorname{Im}[s_n]]\|_2 \leq \sqrt{\rho/N}$, hence a total of $N+1$ SOC constraints. To sum up, there are I linear constraints and $N+1$ SOC constraints, so the computational complexity of solving the simple convex problem should be upper bounded by $\mathcal{O}\left(N^{3.5}\right)$, the same order as SOCP. If we take a closer look at those SOC constraints, we find that they are of very small size (only two variables) and much simpler than those in [27]: no Hadamard product, no matrix decomposition, and no affine transformation. The resulting SOCP is quite sparse, and modern conic solvers such as MOSEK can exploit the sparsity of the problem very efficiently. That is why the practical complexity is far below the worst-case complexity $\mathcal{O}\left(N^{3.5}\right)$. The per-iteration complexity of MDA is elaborated as follows. MDA consists of two main steps in each iteration: (1) computation of subgradient $\boldsymbol{H}^{(m)}$: this step involves matrix multiplications $\boldsymbol{B}\boldsymbol{p}^{(m)}$ and $\boldsymbol{b}^\dagger\boldsymbol{x}^{(m)}$, of complexity $\mathcal{O}(NI)$ ($\boldsymbol{b} \in \mathbb{C}^{N\times I}$, $\boldsymbol{p}^{(m)} \in \mathbb{R}^I$, and $\boldsymbol{x}^{(m)} \in \mathbb{C}^N$); (2) update of $\boldsymbol{p}^{(m)}$ to $\boldsymbol{p}^{(m+1)}$: this step involves element-wise exponent, Hadamard product, and summation, of complexity $\mathcal{O}(I)$. Therefore, the per-iteration complexity of MDA is $\mathcal{O}(NI)$.

7.4.2.4 Robust design with the uncertainty of Doppler

Following the setting in [27], we set $M = N$ (the filter has the same length as the sequence); the channel matrix is given as

$$\boldsymbol{H}_i = \operatorname{Diag}\left(\boldsymbol{p}\left(\nu_{d_T}^i\right)\right), \tag{7.99}$$

where $\boldsymbol{p}(\nu) = \left[1, e^{j2\pi\nu}, \ldots, e^{j2\pi(N-1)\nu}\right]^T$ is the Doppler steering vector and $\nu_{d_T}^i$ is the ith sampled normalized target Doppler frequency, falling within $\left[\nu_{d_T,\text{lower}}, \nu_{d_T,\text{upper}}\right]$. The interference covariance matrix $\boldsymbol{\Sigma}_I(\boldsymbol{s})$ is specifically expressed as

$$\boldsymbol{\Sigma}_I(\boldsymbol{s}) = \sum_{n_c=0}^{N_c-1}\sum_{l=0}^{L-1} \sigma_{(n_c,l)}^2 \boldsymbol{J}_{n_c}\left(\boldsymbol{\Phi}_{\epsilon_{(n_c,l)}}^{\bar{\nu}_{d(n_c,l)}} \odot \boldsymbol{s}\boldsymbol{s}^\dagger\right)\boldsymbol{J}_{n_c}^\dagger, \tag{7.100}$$

where $N_c\,(< N)$ is the number of range rings, L is the number of azimuth sectors, the range-azimuth bin is denoted as (n_c, l), the bin of interest is $(0, 0)$ (where we receive signals), $\sigma_{(n_c,l)}^2$ is the mean interfering power associated with the clutter patch located at the range-azimuth bin (n_c, l) whose (normalized) Doppler shift $\nu_{d_{(n_c,l)}}$ is uniformly distributed over the interval $\left(\bar{\nu}_{d_{(n_c,l)}} - \epsilon_{(n_c,l)}/2,\ \bar{\nu}_{d_{(n_c,l)}} + \epsilon_{(n_c,l)}/2\right)$, $\boldsymbol{J}_{n_c}$ is a Toeplitz matrix with the n_cth subdiagonal entries being 1 and 0 elsewhere, and $\boldsymbol{\Phi}_{\epsilon_{(n_c,l)}}^{\bar{\nu}_{d(n_c,l)}}$ is the covariance matrix of $\boldsymbol{p}\left(\nu_{d_{(n_c,l)}}\right)$, given as

$$\boldsymbol{\Phi}_{\epsilon_{(n_c,l)}}^{\bar{\nu}_{d(n_c,l)}}(m,n) = \exp\left(j2\pi\bar{\nu}_{d_{(n_c,l)}}(m-n)\right) \times \operatorname{sinc}\left(\epsilon_{(n_c,l)}(m-n)\right) \tag{7.101}$$

with $\operatorname{sinc}(x) = \sin(\pi x)/(\pi x)$. In this case, the expression of $\boldsymbol{a}_i$ is specified as

$$\boldsymbol{a}_i = \sum_{n_c=0}^{N_c-1}\sum_{l=0}^{L-1} \sigma_{(n_c,l)}^2 \boldsymbol{J}_{n_c}^\dagger \boldsymbol{a}_i \boldsymbol{a}_i^\dagger \boldsymbol{J}_{n_c} \odot \left(\boldsymbol{\Phi}_{\epsilon_{(n_c,l)}}^{\bar{\nu}_{d(n_c,l)}}\right)^T. \tag{7.102}$$

7.4.3 Numerical experiments

We present numerical results with respect to Doppler robust design. All experiments were performed on a PC with a 3.20-GHz i5-4570 CPU and 8-GB RAM. The off-the-shelf solver is specified as (1) MOSEK [39] built in the CVX toolbox [55], shortly denoted as CVX, and/or (2) the Fusion MATLAB® API in MOSEK, shortly denoted as MOSEK. The proposed algorithms are terminated when the improvement between iterations is smaller than a threshold (by default 10^{-6}) or the number of iterations reaches a predetermined maximum (by default 500).

7.4.3.1 Experiment settings

The transmitting sequence length is $N = 20$. We assume $N_c = 2$ interfering range rings and $L = 100$ azimuth sectors. A homogeneous ground clutter is adopted: $\forall (n_c, l)$, a uniformly distributed clutter is assumed with $\sigma_{(n_c,l)}^2 = \sigma^2 = 1000$ and the Doppler shift of the clutter scatterer $\nu_{d_{(n_c,l)}}$ is uniformly distributed over $\Omega_c = \left(\bar{\nu}_{d_{(n_c,l)}} - \epsilon_{(n_c,l)}/2,\ \bar{\nu}_{d_{(n_c,l)}} + \epsilon_{(n_c,l)}/2\right) = (-0.065,\ 0.065)$. As for the target, $\alpha_i = \alpha = 10$ dB, $\forall i$. The background noise covariance matrix $\boldsymbol{r}$ is $\boldsymbol{I}$ (white noise). The filter bank is designed by assuming $\nu_{d_T}^i \notin \Omega_c$, $\forall i$, i.e., the uncertainty interval of the

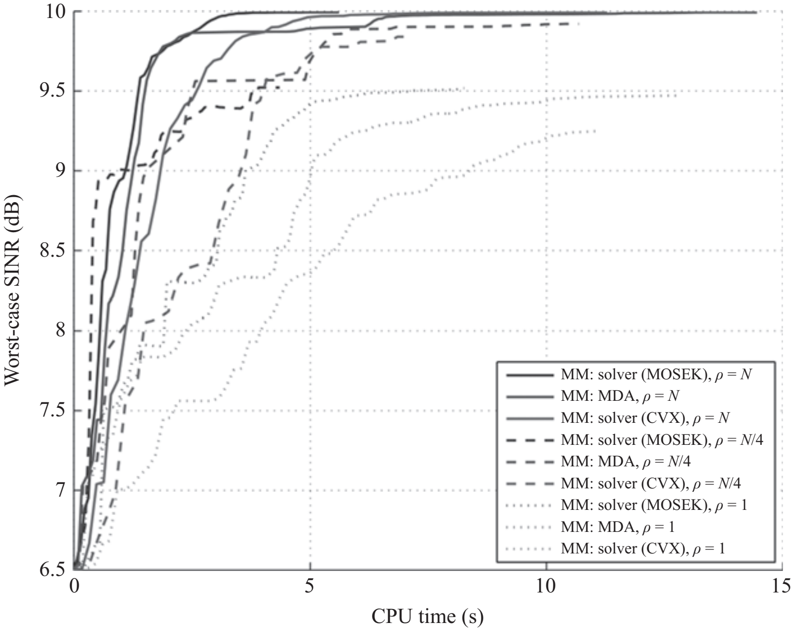

Figure 7.5 Convergence plot: worst-case SINR versus CPU time, $N = 20$

target Doppler frequency $\Omega_T = \left[\nu_{d_T,\text{lower}}, \nu_{d_T,\text{upper}}\right]$ does not overlap with Ω_c. We set $\Omega_T = [0.34, 0.5]$. The number of filters is $I = 10$. For the PAR constraint threshold, we set $\rho = 1$, $N/4$, and N for performance comparison. Unless otherwise specified, all the values of the parameters are unchanged in the numerical experiments.

7.4.3.2 Monotonic property of the proposed algorithms

We implement both solver-based MM and MDA-based MM; both algorithms initiate from a known sequence: the generalized Barker code (with unit energy) [56]. In Figure 7.5, we show the monotonic property of the proposed algorithms. The worst-case SINR (i.e., the objective function value) monotonically increases with the time, until it becomes saturated at a certain level. When we increase the parameter ρ, the optimized worst-case SINR also increases because the constraint set becomes more and more relaxed. We may notice that the two algorithms need different numbers of iterations and time to converge, and they may not converge to exactly the same solution. In the current settings, when $\rho = N/4$ and N, MDA-based MM reaches a slightly higher optimized value, while solver-based MM converges slightly faster, especially in the case of MOSEK; when $\rho = 1$, solver-based MM using MOSEK directly reaches the highest optimized value and converges the fastest.

7.4.3.3 Robust versus non-robust design

We adopt the MDA-based MM as the proposed method, which initiates from the generalized Barker code (with unit energy). In the non-robust design, only the nominal target Doppler frequency is considered. Here we set the nominal value to be the center of the uncertainty interval, i.e., $\hat{\nu}_{d_T} = (0.34 + 0.5)/2 = 0.42$. The radar detection performance is measured by $\text{SINR}(\nu)$, which is defined as

$$\text{SINR}(\nu) = \max_{i=1,2,\ldots,I} \frac{\alpha_i \left| \boldsymbol{w}_i^\dagger \boldsymbol{H}(\nu)\, \boldsymbol{s} \right|^2}{\boldsymbol{w}_i^\dagger \boldsymbol{\Sigma}_I(\boldsymbol{s})\, \boldsymbol{w}_i + \boldsymbol{w}_i^\dagger \boldsymbol{r} \boldsymbol{w}_i}, \tag{7.103}$$

with $\boldsymbol{H}(\nu) = \text{Diag}\left(\left[1, e^{j2\pi\nu}, \ldots, e^{j2\pi(N-1)\nu}\right]^T\right)$. The variable $\boldsymbol{s}$ is derived from optimization; once $\boldsymbol{s}$ is known, the optimal $\{\boldsymbol{w}_i\}$ is also known (cf. (7.70)). The reason for using $\text{SINR}(\nu)$ is related to the detection mechanism of the filter bank: once the received signal is passed through the filter bank, we pick the largest SINR to compare with a predetermined threshold for detection; when the target Doppler is actually ν (still falling within the uncertainty interval), the largest SINR for threshold comparison is thus expressed as $\text{SINR}(\nu)$ and in the performance evaluation, we want $\text{SINR}(\nu)$ to be as large as possible. In Figure 7.6, we compare between robust and non-robust designs under different PAR levels and noise. Under both white noise and colored noise, the robust design has a much smaller scale of fluctuation than its non-robust counterpart. Although the non-robust design achieves slightly higher SINR in a small neighborhood around the nominal value, its worst-case performance across the interval can be arbitrarily bad. Moreover, when imposing different levels of the PAR constraint, we see no significant change of $\text{SINR}(\nu)$ in the robust design different from the non-robust case.

7.4.3.4 Comparison with existing methods

We compare the proposed two algorithms with the existing DESIDE [26] and the SOCP-based algorithm in [27]. To enable fair comparison, only the energy constraint is enforced, i.e., $\rho = N$, and all four methods initiate from the same initial sequence $\boldsymbol{s}^{(0)}$. In Figure 7.7, we plot $\text{SINR}(\nu)$ in the uncertainty interval for the four methods. Our proposed methods achieve a worst-case SINR (the smallest value across the uncertainty interval, i.e., $\min_{\nu \in \Omega_T} \text{SINR}(\nu)$) of around 9.8 dB, while both benchmark algorithms achieve around 8.2 dB.

7.5 Conclusion

In this chapter, we have first given a complete description of the MM optimization method, followed by its application to two radar waveform design problems. The first problem is the joint design of transmit waveform and receive filter for a colocated MIMO radar, which is formulated as maximizing the SINR subject to multiple waveform constraints. We have derived the MIA based on the MM method, which can handle multiple waveform constraints, namely, the constant modulus, the similarity,

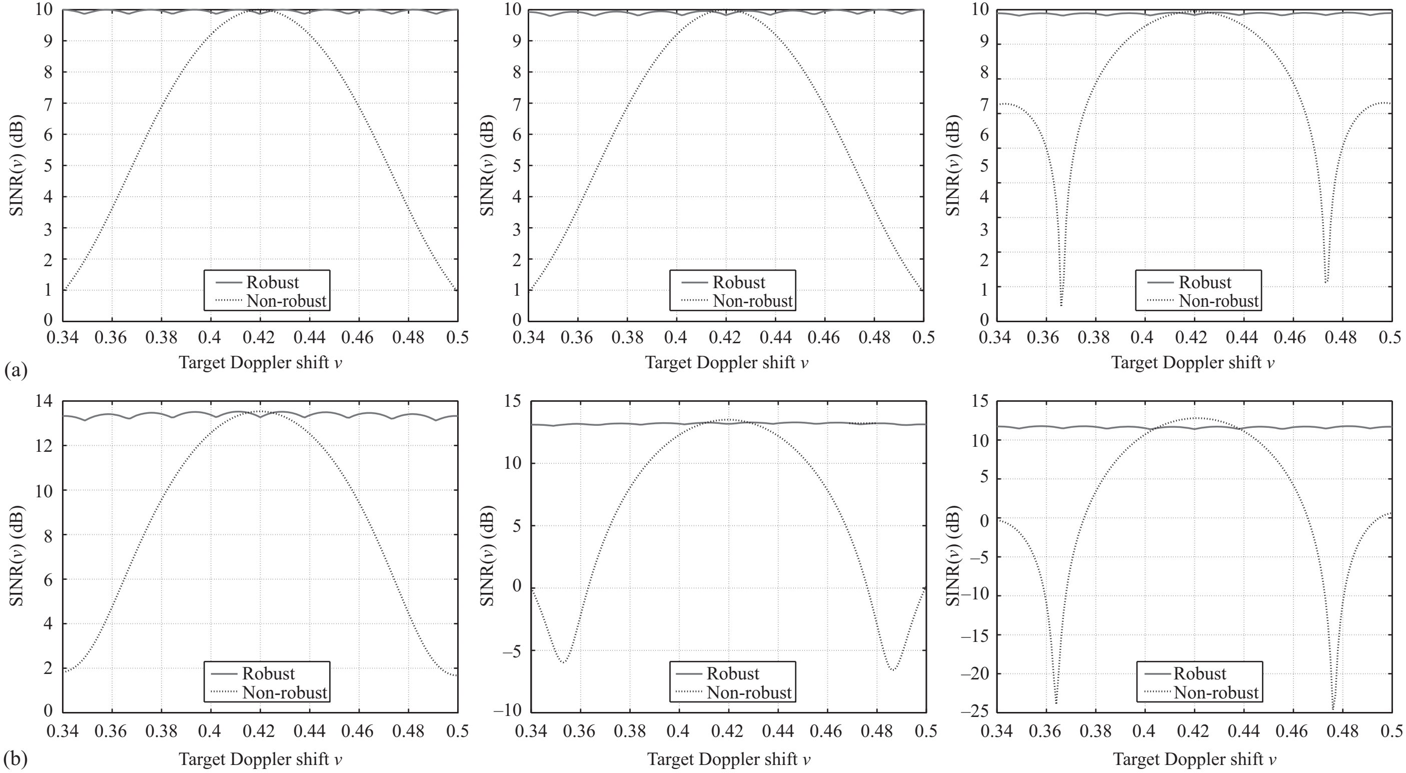

Figure 7.6 Robust designs versus non-robust designs under different PAR levels and noise: (a) White noise ($\boldsymbol{R} = \boldsymbol{I}$): $\rho = N$ (left), $\rho = N/4$ (middle), and $\rho = 1$ (right); $N = 20$ and (b) colored noise ($\boldsymbol{R}(m, n) = 0.4^{|m-n|}$): $\rho = N$ (left), $\rho = N/4$ (middle), and $\rho = 1$ (right); $N = 20$

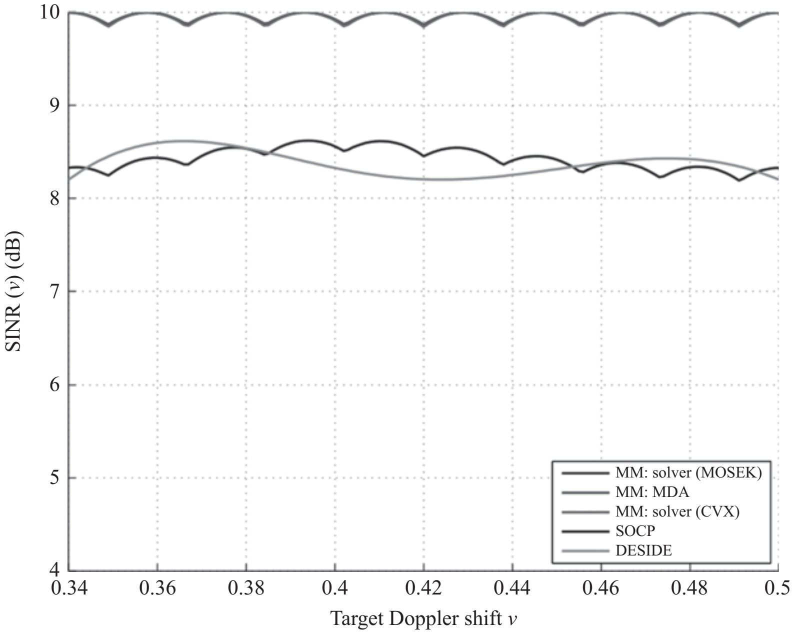

Figure 7.7 SINR (ν) *versus Doppler shift* ν *for four methods: MM: MDA, MM: solver (proposed methods) and SOCP, DESIDE (benchmark methods)*

the PAR, and the spectrum compatibility. Numerical experiments show the good performance of MIA under the above four constraints and emphasize its efficiency in terms of the achieved SINR and the CPU time. The second problem is the robust joint design with the uncertainty of Doppler frequency. It is an extension of the first problem in the sense of mathematical expression, which is formulated as a of the minimum of several SINR's. Based on the maximin extension of MM, an MM-based algorithm for solving the robust design problem has been proposed. Within the proposed algorithm, we have provided two solving approaches for the minorized problem at each iteration. Numerical simulations have shown that the proposed MM algorithms, both solver-based and MDA-based, achieve higher objective values as well as a faster convergence speed compared with the benchmarks, and the achieved robust SINR is higher than the benchmark. The results for the two problems have shown that the algorithms based on the MM method are competitive alternatives compared with the benchmark algorithms for radar waveform design.

Appendix A Proof of Lemma 7.1

Proof. Denote the objective function of (7.40) by $f(\boldsymbol{s}, \boldsymbol{S})$ and define $g(\boldsymbol{x}, \boldsymbol{Y}) = \boldsymbol{x}^{\dagger}\boldsymbol{Y}^{-1}\boldsymbol{x}$, where $\boldsymbol{Y} \succ \boldsymbol{0}$. The function $g(\boldsymbol{x}, \boldsymbol{Y}) = \boldsymbol{x}^{\dagger}\boldsymbol{Y}^{-1}\boldsymbol{x}$ is jointly convex in $\boldsymbol{x}$ and $\boldsymbol{Y}$

[51]. Let $\boldsymbol{x} = \boldsymbol{A}_0\boldsymbol{s}$ and $\boldsymbol{Y} = \boldsymbol{I} + \sum_{k=1}^{K} q_k\boldsymbol{A}_k\boldsymbol{S}\boldsymbol{A}_k^\dagger$. Both are affine transformations. Thus, $f\left(\boldsymbol{s},\boldsymbol{S}\right)$ is jointly concave of $\boldsymbol{s}$ and $\boldsymbol{S}$. Since $f\left(\boldsymbol{s},\boldsymbol{S}\right)$ is jointly concave in $\boldsymbol{s}$ and $\boldsymbol{S}$, the first-order approximation of $f\left(\boldsymbol{s},\boldsymbol{S}\right)$, denoted by $u_1\left(\boldsymbol{s},\boldsymbol{S};\boldsymbol{s}_\ell,\boldsymbol{S}_\ell\right)$, is a majorizer of $f\left(\boldsymbol{s},\boldsymbol{S}\right)$ at the point $\left(\boldsymbol{s}_\ell,\boldsymbol{S}_\ell\right)$, which is given by

$$\begin{aligned}
& u_1\left(\boldsymbol{s},\boldsymbol{S};\boldsymbol{s}_\ell,\boldsymbol{S}_\ell\right)\\
={}& f\left(\boldsymbol{s}_\ell,\boldsymbol{S}_\ell\right) + \mathscr{D}_{\boldsymbol{s}}f|_{\boldsymbol{s}_\ell}\left(\boldsymbol{s}-\boldsymbol{s}_\ell\right) + \mathscr{D}_{\boldsymbol{s}^*}f|_{\boldsymbol{s}_\ell^*}\left(\boldsymbol{s}^*-\boldsymbol{s}_\ell^*\right)\\
& +\mathrm{Tr}\left(\left(\tfrac{\partial f}{\partial \boldsymbol{S}}|_{\boldsymbol{S}_\ell}\right)^T\left(\boldsymbol{S}-\boldsymbol{S}_\ell\right)\right) + \mathrm{Tr}\left(\left(\tfrac{\partial f}{\partial \boldsymbol{S}^*}|_{\boldsymbol{S}_\ell^*}\right)^T\left(\boldsymbol{S}^*-\boldsymbol{S}_\ell^*\right)\right)\\
={}& f\left(\boldsymbol{s}_\ell,\boldsymbol{S}_\ell\right) - \boldsymbol{s}_\ell^\dagger\left(\boldsymbol{A}_0^\dagger\left[\Psi\left(\boldsymbol{S}_\ell\right)+\boldsymbol{I}\right]^{-1}\boldsymbol{A}_0\right)\left(\boldsymbol{s}-\boldsymbol{s}_\ell\right)\\
& -\boldsymbol{s}_\ell^T\left(\boldsymbol{A}_0^\dagger\left[\Psi\left(\boldsymbol{S}_\ell\right)+\boldsymbol{I}\right]^{-1}\boldsymbol{A}_0\right)^T\left(\boldsymbol{s}^*-\boldsymbol{s}_\ell^*\right)\\
& +\mathrm{Tr}\left(\left(\textstyle\sum_{k=1}^{K} q_k\left(\boldsymbol{Q}_\ell^k\right)^\dagger\boldsymbol{S}_\ell\boldsymbol{Q}_\ell^k\right)\left(\boldsymbol{S}-\boldsymbol{S}_\ell\right)\right)\\
& +\mathrm{Tr}\left(\left(\textstyle\sum_{k=1}^{K} q_k\left(\boldsymbol{Q}_\ell^k\right)^\dagger\boldsymbol{S}_\ell\boldsymbol{Q}_\ell^k\right)^T\left(\boldsymbol{S}^*-\boldsymbol{S}_\ell^*\right)\right)\\
={}& -f\left(\boldsymbol{s}_\ell,\boldsymbol{S}_\ell\right) - 2\mathrm{Re}\left(\left(\boldsymbol{s}_\ell^\dagger\boldsymbol{A}_0^\dagger\left[\Psi\left(\boldsymbol{S}_\ell\right)+\boldsymbol{I}\right]^{-1}\boldsymbol{A}_0\right)\boldsymbol{s}\right)\\
& +2\mathrm{Tr}\left(\left(\textstyle\sum_{k=1}^{K} q_k\left(\boldsymbol{Q}_\ell^k\right)^\dagger\boldsymbol{S}_\ell\boldsymbol{Q}_\ell^k\right)\boldsymbol{S}\right)\\
& -2\mathrm{Tr}\left(\left(\textstyle\sum_{k=1}^{K} q_k\left(\boldsymbol{Q}_\ell^k\right)^\dagger\boldsymbol{S}_\ell\boldsymbol{Q}_\ell^k\right)\boldsymbol{S}_\ell\right).
\end{aligned} \tag{A.1}$$

□

Appendix B Proof of Lemma 7.4

Proof. According to $\boldsymbol{\Sigma}_I\left(\boldsymbol{s}\right)$ from (7.69), one term in (7.76) can be rewritten as

$$-\mathrm{Tr}\left(\boldsymbol{a}_i\boldsymbol{a}_i^\dagger\cdot\left(\boldsymbol{\Sigma}_I\left(\boldsymbol{s}\right)-\boldsymbol{\Sigma}_I\left(\boldsymbol{s}^{(n)}\right)\right)\right) = -\boldsymbol{s}^\dagger\boldsymbol{A}_i\boldsymbol{s} + \left(\boldsymbol{s}^{(n)}\right)^\dagger\boldsymbol{A}_i\boldsymbol{s}^{(n)}, \tag{B.1}$$

where $\boldsymbol{a}_i$ follows (7.81). Thus,

$$\mathrm{SINR}_i\left(\boldsymbol{s}\right) \geq \mathrm{SINR}_i\left(\boldsymbol{s}^{(n)}\right) + 2\mathrm{Re}\left[\boldsymbol{b}_i^\dagger\left(\boldsymbol{s}-\boldsymbol{s}^{(n)}\right)\right] - \boldsymbol{s}^\dagger\boldsymbol{A}_i\boldsymbol{s} + \left(\boldsymbol{s}^{(n)}\right)^\dagger\boldsymbol{A}_i\boldsymbol{s}^{(n)}. \tag{B.2}$$

We minorize $\mathrm{SINR}_i\left(\boldsymbol{s}\right)$ by applying $\boldsymbol{s}^\dagger\boldsymbol{A}_i\boldsymbol{s} \leq \left(\boldsymbol{s}^{(n)}\right)^\dagger\boldsymbol{A}_i\boldsymbol{s}^{(n)} + 2\mathrm{Re}\left[\boldsymbol{s}^{(n)H}\boldsymbol{A}_i\left(\boldsymbol{s}-\boldsymbol{s}^{(n)}\right)\right] + \boldsymbol{\lambda}_{max}\left(\boldsymbol{A}_i\right)\left\|\boldsymbol{s}-\boldsymbol{s}^{(n)}\right\|_2^2$:

$$\mathrm{SINR}_i\left(\boldsymbol{s}\right) \geq \mathrm{SINR}_i\left(\boldsymbol{s}^{(n)}\right) + 2\mathrm{Re}\left[\boldsymbol{c}_i^\dagger\left(\boldsymbol{s}-\boldsymbol{s}^{(n)}\right)\right] - \lambda_{u,i}\left\|\boldsymbol{s}-\boldsymbol{s}^{(n)}\right\|_2^2, \tag{B.3}$$

where $\boldsymbol{c}_i$ and $\lambda_{u,i}$ are defined in (7.80) and (7.82), respectively. □

Appendix C Proof of Lemma 7.5

Proof. If we relax $\mathscr{S}$ to be $\mathscr{S}_{\text{relaxed}}$, then the problem becomes (same as (7.91))

$$\max_{\boldsymbol{s}\in\mathscr{S}_{\text{relaxed}}}\ \min_{\boldsymbol{p}\in\mathscr{P}}\ 2\mathrm{Re}\left[\left(\left(\boldsymbol{C}+\boldsymbol{s}^{(l)}\boldsymbol{\lambda}_u^T\right)\boldsymbol{p}\right)^\dagger\boldsymbol{s}\right]+\boldsymbol{p}^T\boldsymbol{d}. \tag{C.1}$$

The objective function is concave–convex in $\boldsymbol{s}$ and $\boldsymbol{p}$, and $\mathscr{S}_{\text{relaxed}}$ and $\mathscr{P}$ are both nonempty compact convex sets. Following the results of [53, Corollary 37.6.2 and Lemma 36.2], a saddle point exists for the relaxed problem. The saddle point of the relaxed problem, denoted by $(\boldsymbol{s}^\star, \boldsymbol{p}^\star)$, must satisfy $\|\boldsymbol{s}^\star\|_2 = 1$. Suppose $\|\boldsymbol{s}^\star\|_2 < 1$. There always exist some element of $\boldsymbol{s}^\star$, denoted by $\boldsymbol{s}_j^\star$, such that $\left|\boldsymbol{s}_j^\star\right| < \sqrt{\rho/N}$. If not, then $\|\boldsymbol{s}^\star\|_2 \geq \sqrt{\left(\sqrt{\rho/N}\right)^2 \times N} = \sqrt{\rho} \geq 1$, causing contradiction. Then we reset the phase of $\boldsymbol{s}_j^\star$ to be aligned with the jth element of $\left(\boldsymbol{C} + \boldsymbol{s}^{(l)}\boldsymbol{\lambda}_u^T\right)\boldsymbol{p}^\star$ and increase its modulus by a small amount without violating feasibility. The objective can be pushed up from the side of $\boldsymbol{s}$, causing contradiction with the saddle point nature of $\boldsymbol{s}^\star$. The jth element of $\left(\boldsymbol{C} + \boldsymbol{s}^{(l)}\boldsymbol{\lambda}_u^T\right)\boldsymbol{p}^\star$ has been assumed to be nonzero for simplicity. In case it becomes zero, the optimal solution of $\boldsymbol{s}_j$ may be nonunique (and thus the saddle point is nonunique), but we can always find one on the boundary by properly increasing the modulus of the currently obtained $\boldsymbol{s}_j^\star$ if necessary. Since the saddle point (or at least one saddle point) of the relaxed problem naturally satisfies $\boldsymbol{s}^\star \in \mathscr{S}$ and $\boldsymbol{p}^\star \in \mathscr{P}$, there must exist a saddle point for problem (7.88), and the saddle point can be obtained from solving the relaxed problem. □

Acknowledgment

This work was supported by the Hong Kong RGC Theme-based Research Scheme (TRS) Grant T21-602/15R.

References

[1] Li J, Stoica P. MIMO Radar Signal Processing. Hoboken, NJ: Wiley Online Library; 2009.

[2] Haykin S. Cognitive radar: a way of the future. IEEE Signal Processing Magazine. 2006;23(1):30–40.

[3] Chen CY, Vaidyanathan P. MIMO radar waveform optimization with prior information of the extended target and clutter. IEEE Transactions on Signal Processing. 2009;57(9):3533–3544.

[4] Ortega JM, Rheinboldt WC. Iterative Solution of Nonlinear Equations in Several Variables. Philadelphia, PA: SIAM; 2000.

[5] Becker MP, Yang I, Lange K. EM algorithms without missing data. Statistical Methods in Medical Research. 1997;6(1):38–54.

[6] Heiser WJ. Convergent computation by iterative majorization: theory and applications in multidimensional data analysis. In: Recent Advances in Descriptive Multivariate Analysis. New York, NY: Oxford University Press; 1995. p. 157–189.

[7] Lange K, Hunter DR, Yang I. Optimization transfer using surrogate objective functions. Journal of Computational and Graphical Statistics. 2000;9(1):1–20.

[8] Song J, Babu P, Palomar DP. Sequence design to minimize the weighted integrated and peak sidelobe levels. IEEE Transactions on Signal Processing. 2016;64(8):2051–2064.

[9] Song J, Babu P, Palomar DP. Sequence set design with good correlation properties via majorization-minimization. IEEE Transactions on Signal Processing. 2016;64(11):2866–2879.

[10] Wu L, Babu P, Palomar DP. Cognitive radar-based sequence design via SINR maximization. IEEE Transactions on Signal Processing. 2017;65(3):779–793.

[11] Wu L, Babu P, Palomar DP. Transmit waveform/receive filter design for MIMO radar with multiple waveform constraints. IEEE Transactions on Signal Processing. 2018;66(6):1526–1540.

[12] Zhao L, Palomar DP. Maximin joint optimization of transmitting code and receiving filter in radar and communications. IEEE Transactions on Signal Processing. 2017;65(4):850–863.

[13] Sun Y, Babu P, Palomar DP. Majorization-minimization algorithms in signal processing, communications, and machine learning. IEEE Transactions on Signal Processing. 2017;65(3):794–816.

[14] Razaviyayn M, Hong M, Luo ZQ. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM Journal on Optimization. 2013;23(2):1126–1153.

[15] Pang JS. Partially B-regular optimization and equilibrium problems. Mathematics of Operations Research. 2007;32(3):687–699.

[16] Pang JS, Razaviyayn M, Alvarado A. Computing B-stationary points of nonsmooth DC programs. arXiv preprint arXiv:151101796. 2015.

[17] Bertsekas DP, Nedić A, Ozdaglar AE. Convex Analysis and Optimization. Belmont, MA: Athena Scientific; 2003.

[18] Jamshidian M, Jennrich RI. Acceleration of the EM algorithm by using quasi-Newton methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 1997;59(3):569–587.

[19] Salakhutdinov R, Roweis ST. Adaptive overrelaxed bound optimization methods. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03); 2003. p. 664–671.

[20] Varadhan R, Roland C. Simple and globally convergent methods for accelerating the convergence of any EM algorithm. Scandinavian Journal of Statistics. 2008;35(2):335–353.

[21] Zhou H, Alexander D, Lange K. A quasi-Newton acceleration for high-dimensional optimization algorithms. Statistics and Computing. 2011;21(2):261–273.

[22] Rockafellar RT, Wets RJB. Variational analysis. vol. 317. Berlin, Heidelberg: Springer Science & Business Media; 2009.

[23] Cui G, Li H, Rangaswamy M. MIMO radar waveform design with constant modulus and similarity constraints. IEEE Transactions on Signal Processing. 2014;62(2):343–353.

[24] Setlur P, Rangaswamy M. Joint filter and waveform design for radar STAP in signal dependent interference. arXiv preprint arXiv:151000055. 2015; [Online] Available: http://arxiv.org/abs/1510.00055.

[25] Naghsh MM, Soltanalian M, Stoica P, *et al.* Radar code design for detection of moving targets. IEEE Transactions on Aerospace and Electronic Systems. 2014;50(4):2762–2778.

[26] Naghsh MM, Soltanalian M, Stoica P, *et al.* A Doppler robust design of transmit sequence and receive filter in the presence of signal-dependent interference. IEEE Transactions on Signal Processing. 2014;62(4):772–785.

[27] Aubry A, De Maio A, Naghsh MM. Optimizing radar waveform and Doppler filter bank via generalized fractional programming. IEEE Journal of Selected Topics in Signal Processing. 2015;9(8):1387–1399.

[28] Stoica P, He H, Li J. Optimization of the receive filter and transmit sequence for active sensing. IEEE Transactions on Signal Processing. 2012;60(4):1730–1740.

[29] Aubry A, De Maio A, Farina A, *et al.* Knowledge-aided (potentially cognitive) transmit signal and receive filter design in signal-dependent clutter. IEEE Transactions on Aerospace and Electronic Systems. 2013;49(1):93–117.

[30] Yu X, Cui G, Kong L, *et al.* Space-time transmit code and receive filter design for colocated MIMO radar. In: Proceedings of the IEEE Radar Conference (RadarConf); 2016. p. 1–6.

[31] Song J, Babu P, Palomar DP. Optimization methods for designing sequences with low autocorrelation sidelobes. IEEE Transactions on Signal Processing. 2015;63(15):3998–4009.

[32] De Maio A, De Nicola S, Huang Y, *et al.* Design of phase codes for radar performance optimization with a similarity constraint. IEEE Transactions on Signal Processing. 2009;57(2):610–621.

[33] Bertsekas DP. Nonlinear programming. Journal of the Operational Research Society. 1997;48(3):334–334.

[34] Tropp J, Dhillon IS, Heath RW, *et al.* Designing structured tight frames via an alternating projection method. IEEE Transactions on Information Theory. 2005;51(1):188–209.

[35] Scutari G, Palomar DP, Barbarossa S. Cognitive MIMO radio. IEEE Signal Processing Magazine. 2008;25(6):46–59.

[36] Scutari G, Palomar DP. MIMO cognitive radio: a game theoretical approach. IEEE Transactions on Signal Processing. 2010;58(2):761–780.

[37] Aubry A, De Maio A, Piezzo M, *et al.* Radar waveform design in a spectrally crowded environment via nonconvex quadratic optimization. IEEE Transactions on Aerospace and Electronic Systems. 2014;50(2):1138–1152.

[38] Mehanna O, Huang K, Gopalakrishnan B, *et al.* Feasible point pursuit and successive approximation of non-convex QCQPs. IEEE Signal Processing Letters. 2015;22(7):804–808.

[39] The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 63). Denmark: MOSEK ApS. 2015; [Online] Available: http://docs.mosek.com/7.1/toolbox/index.html.

[40] Cui G, Yu X, Carotenuto V, *et al.* Space-time transmit code and receive filter design for colocated MIMO radar. IEEE Transactions on Signal Processing. 2017;65(5):1116–1129.

[41] Karbasi SM, Aubry A, De Maio A, *et al.* Robust transmit code and receive filter design for extended targets in clutter. IEEE Transactions on Signal Processing. 2015;63(8):1965–1976.

[42] Aubry A, De Maio A, Piezzo M, *et al.* Cognitive radar waveform design for spectral coexistence in signal-dependent interference. In: Proceedings of the IEEE Radar Conference (RadarConf). IEEE; 2014. p. 474–478.

[43] Stoica P, Li J, Xie Y. On probing signal design for MIMO radar. IEEE Transactions on Signal Processing. 2007;55(8):4151–4161.

[44] Friedlander B. Waveform design for MIMO radars. IEEE Transactions on Aerospace and Electronic Systems. 2007;43(3):1227–1238.

[45] De Maio A, Huang Y, Piezzo M, *et al.* Design of optimized radar codes with a peak to average power ratio constraint. IEEE Transactions on Signal Processing. 2011;59(6):2683–2697.

[46] De Maio A, Huang Y, Piezzo M. A Doppler robust max-min approach to radar code design. IEEE Transactions on Signal Processing. 2010;58(9):4943–4947.

[47] Li J, Stoica P. MIMO radar with colocated antennas. IEEE Signal Processing Magazine. 2007;24(5):106–114.

[48] Xu H, Blum RS, Wang J, *et al.* Colocated MIMO radar waveform design for transmit beampattern formation. IEEE Transactions on Aerospace and Electronic Systems. 2015;51(2):1558–1568.

[49] Karbasi SM, Aubry A, Carotenuto V, *et al.* Knowledge-based design of space-time transmit code and receive filter for a multiple-input-multiple-output radar in signal-dependent interference. IET Radar, Sonar & Navigation. 2015;9(8):1124–1135.

[50] Soltanalian M, Tang B, Li J, *et al.* Joint design of the receive filter and transmit sequence for active sensing. IEEE Signal Processing Letters. 2013;20(5): 423–426.

[51] Boyd S, Vandenberghe L. Convex Optimization. Cambridge: Cambridge University Press; 2004.

[52] Lipp T, Boyd S. Variations and extension of the convex–concave procedure. Optimization and Engineering. 2016;17(2):263–287.

[53] Rockafellar RT. Convex Analysis. Princeton, NJ: Princeton University Press; 1970.

[54] Beck A, Teboulle M. Mirror descent and nonlinear projected subgradient methods for convex optimization. Operations Research Letters. 2003;31(3): 167–175.

[55] Grant M, Boyd S. CVX: Matlab Software for Disciplined Convex Programming, version 2.1. 2014. Availablehttp://cvxr.com/cvx.

[56] Golomb S, Scholtz R. Generalized barker sequences. IEEE Transactions on Information theory. 1965;11(4):533–537.