Computationally Efficient Algorithms for Radar Signal Design in Spectrally Busy Environment

Author: Markus Yli-Niemi
Instructor: Sergiy Vorobyov

November 2, 2018

Master's Thesis,
Aalto University School of Electrical Engineering,
Department of Signal Processing and Acoustics

Author Markus Yli-Niemi

Title of thesis Computationally Efficient Algorithms for Radar Signal Design in Spectrally Busy Environment

Degree Master of Science

Degree programme Signal Processing and Data Science

Thesis advisor(s) Sergiy Vorobyov

Year of approval 2018 **Number of pages** 87+8 **Language** English

Abstract

In this thesis the problem of designing radar transmitter waveforms in a spectrally busy environment is considered. Spectrally busy environment is an environment where radar operates in a congested frequency spectrum with other radiators. Both single waveform design and multiple waveforms design problems are considered.

The solution algorithms for the design problems are based on Alternating Direction Method of Multipliers (ADMM) algorithm alongside computationally efficient projection techniques. Solution algorithms are verified to run on quadratic (i.e., $O(N^2)$, where N is the problem dimension) or cubic (i.e., $O(N^3)$) time complexities by time complexity graphs. Solution algorithms are also tested with example environment simulations.

The quality of designed transmitter waveforms are assessed with signal-to-interference-plus-noise ratio (SINR) and ambiguity function figures. According to these figures designed waveforms have adequate SINR while ambiguity functions have small Doppler leakages and sharp autocorrelation functions.

Keywords Radar transmitter waveform design, ADMM, Majorization-Minimization

Publications based on this thesis

title: Computationally Efficient Waveform Design in Spectrally Dense Environment
authors: Markus Yli-Niemi & Sergiy A. Vorobyov
conference: IEEE SAM 2018 in Sheffield, UK
description: This paper considers unimodular radar transmitter waveform design in spectrally dense environment. New computationally efficient method for the design problem is proposed. New method is based on ADMM-algorithm alongside Majorization-Minimization (MaMi) and computationally efficient projection techniques.

title: Computationally Efficient Algorithms for Designing Multiple Waveforms in Spectrally Dense Environment
authors: Markus Yli-Niemi, Sergiy A. Vorobyov & Patrik Dammert
journal: Not decided yet, article is under preparation.
description: This article considers problems of designing multiple unimodular radar transmitter waveforms in spectrally dense environment. Two new problem formulations are derived and they are solved in computationally efficient manner. This article is extension on our earlier work on design of single unimodular radar transmitter waveform in spectrally dense environment.

Contents

1 Introduction **2**

2 Prerequisites **4**
2.1 Dual Ascent algorithm 4
2.2 Method of Multipliers 7
2.3 ADMM 8
2.4 Majorization-Minimization (MM) techniques in Large-Scale Optimization 10

3 Single waveform design in spectrally busy environment **13**
3.1 SINR maximization problem in spectrally busy environment . . . 13
3.2 ADMM-type algorithm derivation 18
3.2.1 c-variable update 21
3.2.2 z-variable update 28
3.2.3 Proposed algorithm 31
3.3 Performance analysis and simulation example 35
3.3.1 Simulation set up 35
3.4 Single waveform algorithm extension 41
3.4.1 z-variable update 41

4 Multiple waveforms design in spectrally busy environment **44**
4.1 Simplified version of $\mathcal{P}_2$ 46
4.2 1st algorithm for multiple waveform design 48
4.2.1 z-variable update 48
4.2.2 Performance analysis 54
4.2.3 Simulation set up 54
4.3 2nd algorithm for multiple waveform design 62
4.3.1 Single waveform update 64
4.3.2 Performance analysis and example simulation 70
4.4 3rd algorithm for multiple waveform design 73
4.4.1 c-variable update 74
4.4.2 Performance analysis 77
4.4.3 Simulation set up 79

5 Conclusion **86**

6 References **87**

7 Appendices **88**
7.1 Appendix A 88
7.2 Appendix B 90
7.3 Appendix C 91

1 Introduction

Waveform design is one of the classical problems in radar and communication systems [1, 2, 3, 4, 5, 6, 7]. In recent years there have been a lot of interest in designing radar signals in spectrally busy environments, when frequency bands are highly congested. In this type of environment, it is challenging to design radar signals that have adequate signal-to-interference-plus-noise ratio (SINR), while undesired radiation to overlaid systems is kept under control. This problem leads to an NP-hard SINR-value maximization problem, as derived in [8] and [9], with constraints applied both to disturbance level on overlaid bandwiths and fast-time radar code energy. The basic setting of the SINR-value maximization problem in spectrally busy environment is derived in Section 3.1 of this Master's Thesis.

When the waveform design problem needs to be solved in radar systems that operate in GHz region, the resulting maximization problem is essentially large-scale optimization problem with fast-time radar code vector reaching to dimensions of tens of thousands. This large-scale optimization problem requires very efficient algorithm to be solved accurately enough in a very short time.

In [8], certain type of relaxation and randomization procedure is introduced to solve SINR maximization problem. Unfortunately, this type of solution method is very inefficient, with the time complexity of $O(N^{3.5})$ to $O(N^{4.5})$, where N is the problem dimension, which equals the transmitter waveform code length squared in here because of the lifting of the code optimization variable to a positive semidefinite (PSD) matrix variable as a relaxation step. Thus, such problem cannot be solved in feasible time if fast-time radar code vector has dimension of tens of thousands. Therefore, more efficient solution methods must be developed.

To tackle the problem, we will use Alternating Direction Method of Multipliers (ADMM) algorithm as our base approach. ADMM is one of the best algorithms to date to solve convex optimization problems. ADMM provides fast convergence rate with adequate solution resolution, which is desirable in this application. The details of ADMM and its preceding methods can be found in Section 2 of this Master's Thesis.

In this thesis, we will consider both single waveform and multiple waveform design problems in a spectrally busy environment. Multiwaveform design problem is a straight extension of the single waveform design problem with objective to maximize the sum of SINR-values of designed signals. Also, additional constraints for integrated sidelobe level (ISL) and peak-to-average-power ratio (PAPR) are considered. Single waveform design problem is considered in Section 3, while multiple waveform design problem is considered in Section 4. For all derived algorithms, we will execute the performance analysis to evaluate the time-complexities of the methods, and run simulations in which we implement

our design algorithms in an example environment.

2 Prerequisites

In this section, we introduce optimization techniques and mathematical frameworks that are needed in the next sections. The aim is to introduce ADMM by first discussing details of its precursors, called Dual Ascent and Method of Multipliers. ADMM combines the advantageous properties of these precursors. It motivates the use of ADMM in the later algorithm derivations (see subsections 3.2, 4.2, 4.3 and 4.4).

In addition to ADMM, Majorization-Minimization (MM) techniques using first-order surrogate functions are discussed. In many cases, the objective functions in optimization problems are so complex that they cannot be efficiently solved in their original forms. One way to tackle this problem is to use approximate functions that bound the original objectives. In the case of minimization problem, objective function is upper-bounded by surrogate, and instead of minimizing objective, surrogate is minimized, which gives an approximation of the minimum point of the objective. This technique is called then MM. In Subsection 3.2, MM methods turn out to play a central role in the algorithm derivation.

If reader is interested to deepen the knowledge of prerequisites introduced here, more details about the techniques and results can be found in [10], [11], [12].

2.1 Dual Ascent algorithm

Consider a minimization problem of the form:

$$\begin{cases} \text{minimize } f(\mathbf{x}) & \text{(2.1.1a)} \\ \text{subject to: } \mathbf{A}\mathbf{x} \leq \mathbf{b}, & \text{(2.1.1b)} \end{cases}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $f : \mathbb{R}^n \to \mathbb{R}$ is convex. The Lagrangian function for the problem (2.1.1a)-(2.1.1b) is given as:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{b}), \tag{2.1.2}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of Lagrange multipliers. By rewriting constraint (2.1.1b), $\mathbf{A}\mathbf{x} \leq \mathbf{b} \Rightarrow \mathbf{A}\mathbf{x} - \mathbf{b} \leq 0$, and requiring $\boldsymbol{\lambda}^T \geq 0$, i.e. $\boldsymbol{\lambda}_i \geq 0$, $\forall i$, the Lagrangian can be upper-bounded by the objective function f:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \underbrace{\boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{b})}_{\leq 0} \leq f(\mathbf{x}). \tag{2.1.3}$$

Infimum of the Lagrangian function can be written as

$$\begin{aligned}\inf_{\mathbf{x}} L(\mathbf{x},\boldsymbol{\lambda}) &= \inf_{\mathbf{x}} \left\{ f(\mathbf{x}) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x}-\mathbf{b}) \right\} = \inf_{\mathbf{x}} \left\{ f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{A}\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{b} \right\} \\ &= \inf_{\mathbf{x}} \left\{ f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{A}\mathbf{x} \right\} - \boldsymbol{\lambda}^T\mathbf{b} \\ &\overset{\boldsymbol{\lambda}^T\mathbf{b}=\mathbf{b}^T\boldsymbol{\lambda}}{=} \sup_{\mathbf{x}} \left\{ -\left(f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{A}\mathbf{x} \right) \right\} - \mathbf{b}^T\boldsymbol{\lambda} \\ &= \sup_{\mathbf{x}} \left\{ -\boldsymbol{\lambda}^T\mathbf{A}\mathbf{x} - f(\mathbf{x}) \right\} - \mathbf{b}^T\boldsymbol{\lambda} \\ &= f^*\left(-\mathbf{A}^T\boldsymbol{\lambda} \right) - \mathbf{b}^T\boldsymbol{\lambda}, \end{aligned} \tag{2.1.4}$$

where convex conjugate $f^*(\boldsymbol{\lambda})$ is defined as [10]:

$$f^*(\boldsymbol{\lambda}) = \sup_{\mathbf{x}\in \mathrm{dom}(f)} \left\{ -\boldsymbol{\lambda}^T\mathbf{x} - f(\mathbf{x}) \right\}. \tag{2.1.5}$$

Function $\mathrm{dom}(f)$ refers to the domain where $f(\mathbf{x})$ is defined. Denote $g(\boldsymbol{\lambda}) = \inf_{\mathbf{x}} L(\mathbf{x},\boldsymbol{\lambda})$ and $p^* = \inf_{\mathbf{x}} f(\mathbf{x})$. From (2.1.3) it is clear that $g(\boldsymbol{\lambda}) \leq p^*$. To attain p^* using $g(\boldsymbol{\lambda})$, we write:

$$p^* = \max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}) = \sup_{\boldsymbol{\lambda}} \inf_{\mathbf{x}} L(\mathbf{x},\boldsymbol{\lambda}). \tag{2.1.6}$$

Expression (2.1.6) is called Lagrangian dual problem. The idea of dual problem is that minimum point of objective can be found by maximizing dual function and vice versa. Maximum point obtained from (2.1.6) is called dual optimal point $\boldsymbol{\lambda}^*$, while primal optimal point $\mathbf{x}^*$ is the point where objective f is minimized.

Primal optimal point $\mathbf{x}^*$ can be restored from dual optimal point $\boldsymbol{\lambda}^*$:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} L\left(\mathbf{x},\boldsymbol{\lambda}^*\right). \tag{2.1.7}$$

To iteratively find primal optimal point $\mathbf{x}^*$, approximations of $\mathbf{x}^*$ and $\boldsymbol{\lambda}^*$ are calculated as long as feasible tolerance is reached. Steps can be written as:

$$\begin{cases} \mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} L\left(\mathbf{x},\boldsymbol{\lambda}_k\right) & \text{(2.1.8a)} \\ \boldsymbol{\lambda}_{k+1} = \arg\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}) = \sup_{\boldsymbol{\lambda}} \inf_{\mathbf{x}} L(\mathbf{x},\boldsymbol{\lambda}). & \text{(2.1.8b)} \end{cases}$$

Step (2.1.8a) restores primal optimal point from dual optimal point, and step (2.1.8b) finds update for dual optimal point. In Dual Ascent algorithm, gradient ascent method, i.e., $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k \nabla g(\boldsymbol{\lambda})$, where α_k is step-length, is used to find dual optimal point. Gradient of the dual function $g(\boldsymbol{\lambda})$ is given as:

$$\begin{aligned}\nabla g(\boldsymbol{\lambda}) &= \nabla_{\boldsymbol{\lambda}} \left(\inf_{\mathbf{x}} \left\{ f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{A}\mathbf{x} \right\} - \boldsymbol{\lambda}^T \mathbf{b} \right) \\ &= \nabla_{\boldsymbol{\lambda}} \left(f\left(\mathbf{x}^+\right) + \boldsymbol{\lambda}^T \mathbf{A}\mathbf{x}^+ - \boldsymbol{\lambda}^T \mathbf{b} \right) \\ &= \mathbf{A}\mathbf{x}^+ - \mathbf{b}, \end{aligned} \tag{2.1.9}$$

where $\mathbf{x}^+$ is optimal point of $\inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda})$. Hence Dual Ascent algorithm can be written as:

$$\begin{cases} \mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} L\left(\mathbf{x}, \boldsymbol{\lambda}_k\right) & \text{(2.1.10a)} \\ \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k \left(\mathbf{A}\mathbf{x}_{k+1} - \mathbf{b}\right). & \text{(2.1.10b)} \end{cases}$$

2.2 Method of Multipliers

For optimization problem (2.1.1a)-(2.1.1b) augmented Lagrangian is defined as:

$$L_\rho(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2, \tag{2.2.1}$$

where $\rho > 0$ is penalty parameter and $\|\mathbf{v}\|_2 = \sqrt{\sum_{k=1}^{N} v[k]^2}, \mathbf{v} = \mathbf{A}\mathbf{x} - \mathbf{b}$, denotes the L^2-norm of a vector. Augmented Lagrangian (2.2.1) can also be considered as Lagrangian of optimization problem of the form (2.1.1a-2.1.1b), where objective function is augmented as $F(\mathbf{x}) = f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$. Dual Ascent for this problem can be written as:

$$\begin{cases} \mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} L_\rho\left(\mathbf{x}, \boldsymbol{\lambda}_k\right) & \text{(2.2.2a)} \\ \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \rho\left(\mathbf{A}\mathbf{x}_{k+1} - \mathbf{b}\right), & \text{(2.2.2b)} \end{cases}$$

where gradient ascent step length α is chosen to be the penalty parameter ρ. Steps (2.2.2a)-(2.2.2b) are called Method of Multipliers.

The advantage of the Method of Multipliers compared to the Dual Ascent method is that it converges faster and for more general conditions [10]. The disadvantage however is that it does not support parallel computations (as the Dual Ascent does, if the objective is separable) because the augmented objective F is not separable due to L^2-norm term. This means that the Method of Multipliers can be significantly slower than the Dual Ascent method in some cases.

2.3 ADMM

Alternating Direction Method of Multipliers (ADMM) can be considered as a combination of the Dual Ascent method and the Method of Multipliers introduced in Subsections 2.1 and 2.2, respectively. In ADMM, objective function is assumed to be separable. Thus, ADMM can be applied to minimization problem of the form:

$$\begin{cases} \text{minimize } f(\mathbf{x}) + g(\mathbf{z}) & \text{(2.3.1a)} \\ \text{subject to: } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, & \text{(2.3.1b)} \end{cases}$$

where $\mathbf{x} \in \mathbb{R}^n, \mathbf{z} \in \mathbb{R}^m, \mathbf{c} \in \mathbb{R}^p, \mathbf{A} \in \mathbb{R}^{p\times n}, \mathbf{B} \in \mathbb{R}^{p\times m}$, and functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^m \to \mathbb{R}$ are convex. Augmented Lagrangian for (2.3.1a)-(2.3.1b) is given as:

$$L_\rho(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}) = f(\mathbf{x}) + g(\mathbf{z}) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2. \quad \text{(2.3.2)}$$

The Method of Multipliers for problem (2.3.1a)-(2.3.1b) can be expressed as:

$$\begin{cases} (\mathbf{x}_{k+1}, \mathbf{z}_{k+1}) = \underset{\mathbf{x},\mathbf{z}}{\arg\min}\, L_\rho\left(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}_k\right) & \text{(2.3.3a)} \\ \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \rho\left(\mathbf{A}\mathbf{x}_{k+1} - \mathbf{b}\right). & \text{(2.3.3b)} \end{cases}$$

By splitting step (2.3.3a) for separate $\mathbf{x}$- and $\mathbf{z}$-variable updates, the following three-step iteration is obtained:

$$\begin{cases} \mathbf{x}_{k+1} = \underset{\mathbf{x}}{\arg\min}\, L_\rho\left(\mathbf{x}, \mathbf{z}_k, \boldsymbol{\lambda}_k\right) & \text{(2.3.4a)} \\ \mathbf{z}_{k+1} = \underset{\mathbf{z}}{\arg\min}\, L_\rho\left(\mathbf{x}_{k+1}, \mathbf{z}, \boldsymbol{\lambda}_k\right) & \text{(2.3.4b)} \\ \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \rho\left(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1} - \mathbf{c}\right). & \text{(2.3.4c)} \end{cases}$$

Steps (2.3.4a)-(2.3.4c) are called ADMM. The popularity of ADMM in convex objective optimization is due to its general convergence properties and the support for parallel computations [10]. ADMM can be thought of having the advantages of both the Dual Ascent method and the Method of Multipliers.

In some applications, ADMM steps are written in so-called scaled form. In this form constraint (2.3.1b) is written as residual $\mathbf{r}$:

$$\mathbf{r}(\mathbf{x}, \mathbf{z}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}. \quad \text{(2.3.5)}$$

Augmented Lagrangian (2.3.3) in this case becomes:

$$
\begin{aligned}
L_\rho(\mathbf{x},\mathbf{z},\boldsymbol{\lambda}) &= f(\mathbf{x}) + g(\mathbf{z}) + \boldsymbol{\lambda}^T\mathbf{r} + \frac{\rho}{2}\|\mathbf{r}\|_2^2 \\
&= f(\mathbf{x}) + g(\mathbf{z}) + \boldsymbol{\lambda}^T\mathbf{r} + \frac{\rho}{2}\mathbf{r}^T\mathbf{r} \\
&= f(\mathbf{x}) + g(\mathbf{z}) + \boldsymbol{\lambda}^T\mathbf{r} + \frac{\rho}{2}\mathbf{r}^T\mathbf{r} + \frac{1}{2\rho}\boldsymbol{\lambda}^T\boldsymbol{\lambda} - \frac{1}{2\rho}\boldsymbol{\lambda}^T\boldsymbol{\lambda} \\
&= f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2}\left(\frac{2}{\rho}\boldsymbol{\lambda}^T\mathbf{r} + \mathbf{r}^T\mathbf{r} + \frac{1}{\rho^2}\boldsymbol{\lambda}^T\boldsymbol{\lambda}\right) - \frac{1}{2\rho}\boldsymbol{\lambda}^T\boldsymbol{\lambda} \\
&\overset{\boldsymbol{\lambda}^T\mathbf{r}=\mathbf{r}^T\boldsymbol{\lambda}}{=} f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2}\underbrace{\left(\frac{1}{\rho}\boldsymbol{\lambda}^T\mathbf{r} + \frac{1}{\rho}\mathbf{r}^T\boldsymbol{\lambda} + \mathbf{r}^T\mathbf{r} + \frac{1}{\rho^2}\boldsymbol{\lambda}^T\boldsymbol{\lambda}\right)}_{=\left(\mathbf{r}+\frac{1}{\rho}\boldsymbol{\lambda}\right)^T\left(\mathbf{r}+\frac{1}{\rho}\boldsymbol{\lambda}\right)} - \frac{1}{2\rho}\boldsymbol{\lambda}^T\boldsymbol{\lambda} \\
&= f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2}\left(\mathbf{r} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T\left(\mathbf{r} + \frac{1}{\rho}\boldsymbol{\lambda}\right) - \frac{1}{2\rho}\boldsymbol{\lambda}^T\boldsymbol{\lambda} \\
&= f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2}\left\|\frac{\boldsymbol{\lambda}}{\rho}\right\|_2^2 - \frac{\rho}{2}\left\|\frac{\boldsymbol{\lambda}}{\rho}\right\|_2^2 \\
&\overset{\mathbf{u}=\boldsymbol{\lambda}/\rho}{=} f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{r}+\mathbf{u}\|_2^2 - \frac{\rho}{2}\|\mathbf{u}\|_2^2.
\end{aligned}
$$

Then ADMM in scaled form becomes:

$$
\begin{cases}
\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \left(f(\mathbf{x}) + \frac{\rho}{2}\left\|\mathbf{r}\left(\mathbf{x},\mathbf{z}_k\right) + \mathbf{u}_k\right\|_2^2\right) & \text{(2.3.6a)} \\
\mathbf{z}_{k+1} = \arg\min_{\mathbf{z}} \left(g(\mathbf{z}) + \frac{\rho}{2}\left\|\mathbf{r}\left(\mathbf{x}_{k+1},\mathbf{z}\right) + \mathbf{u}_k\right\|_2^2\right) & \text{(2.3.6b)} \\
\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{r}\left(\mathbf{x}_{k+1},\mathbf{z}_{k+1}\right). & \text{(2.3.6c)}
\end{cases}
$$

2.4 Majorization-Minimization (MM) techniques in Large-Scale Optimization

In Majorization-Minimization (MM) optimization methods the idea is to find a simpler surrogate function $g(\mathbf{x})$ that upper-bounds a more complex objective function $f(\mathbf{x})$. Instead of minimizing the objective function, the surrogate function is then minimized. Since the objective is upper-bounded by the surrogate, minimum value of the surrogate also approaches minimum value of the objective. This approach is very useful in situations where the objective function cannot be easily minimized.

In this work, Lipschitz gradient surrogate and Proximal gradient surrogate functions are used. These two surrogate function families belong to the first-order surrogate functions. The definition of the first-order surrogate functions is given in [11] and [12]:

Definition 2.4.1 (Family of first-order surrogate functions) *Let $\boldsymbol{\kappa}$ be in convex region Θ and $f \in C^0$ be the objective function. Denote by $\mathcal{S}_{L,\rho}(f, \boldsymbol{\kappa})$ the set of ρ-strongly convex functions g such that $g \geq f, g(\boldsymbol{\kappa}) = f(\boldsymbol{\kappa})$, the approximation error $e = g - f$ is differentiable, and the gradient ∇e is L-Lipschitz continuous. Then the functions g in $\mathcal{S}_{L,\rho}(f, \boldsymbol{\kappa})$ are called "first-order surrogate functions".*

In Definition 2.4.1, the function family C^n denotes family of n-times continuously differentiable functions, ρ-strong convexity is defined as:

$$g(\boldsymbol{\kappa}) \geq \underbrace{g(\boldsymbol{\theta}) + \nabla g(\boldsymbol{\theta})(\boldsymbol{\kappa} - \boldsymbol{\theta})}_{\text{tangent hyperplane}} + \frac{\rho}{2}\|\boldsymbol{\kappa} - \boldsymbol{\theta}\|_2^2 \tag{2.4.2}$$

and L-Lipschitz continuity as:

$$|g(\boldsymbol{\kappa}) - g(\boldsymbol{\theta})| \leq L\,|\boldsymbol{\kappa} - \boldsymbol{\theta}|\,. \tag{2.4.3}$$

It is important to notice that inequality (2.4.3) also applies to L^2-norm:

$$\begin{aligned}
\|g(\boldsymbol{\kappa}) - g(\boldsymbol{\theta})\|_2 &= \langle g(\boldsymbol{\kappa}) - g(\boldsymbol{\theta}), g(\boldsymbol{\kappa}) - g(\boldsymbol{\theta})\rangle^{1/2} \\
&= \left(\int_{\boldsymbol{\theta}=\Theta} |g(\boldsymbol{\kappa}) - g(\boldsymbol{\theta})|^2\, d\boldsymbol{\theta}\right)^{1/2} \\
&\leq \left(\int_{\boldsymbol{\theta}=\Theta} (L\,|\boldsymbol{\kappa} - \boldsymbol{\theta}|)^2 d\boldsymbol{\theta}\right)^{1/2} \\
&= L\left(\int_{\boldsymbol{\theta}=\Theta} |\boldsymbol{\kappa} - \boldsymbol{\theta}|^2\, d\boldsymbol{\theta}\right)^{1/2} \\
&= L\|\boldsymbol{\kappa} - \boldsymbol{\theta}\|_2.
\end{aligned}$$

The definitions of Lipschitz gradient surrogates and Proximal gradient surrogates can be found in [4] and [5], and are given below.

Definition 2.4.4 (Lipschitz gradient surrogates) *Let $f \in C^1$ and its gradient ∇f is L-Lipschitz. Lipschitz gradient surrogate g in $\mathcal{S}_{2L,L}(f, \boldsymbol{\kappa})$ is:*

$$g(\boldsymbol{\theta}) = f(\boldsymbol{\kappa}) + \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) + \tfrac{L}{2}\|\boldsymbol{\theta} - \boldsymbol{\kappa}\|_2^2.$$

If f is convex, g is in $\mathcal{S}_{L,L}(f, \boldsymbol{\kappa})$, and when f is μ-strongly convex, g is in $\mathcal{S}_{L-\mu,L}(f, \boldsymbol{\kappa})$. Surrogate $g(\boldsymbol{\theta})$ can be minimized with gradient descent step $\boldsymbol{\theta} \leftarrow \boldsymbol{\kappa} - \frac{1}{L}\nabla f(\boldsymbol{\kappa})$.

Definition 2.4.5 (Proximal gradient surrogates) *Let $f = f_1 + f_2$, where $f_1 \in C^1, \nabla f_1$ is L-Lipschitz continuous and f_2 is convex. The proximal gradient surrogate g in $\mathcal{S}_{2L,L}(f, \boldsymbol{\kappa})$ is:*

$$g(\boldsymbol{\theta}) = f_1(\boldsymbol{\kappa}) + \nabla f_1(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) + \tfrac{L}{2}\|\boldsymbol{\theta} - \boldsymbol{\kappa}\|_2^2 + f_2(\boldsymbol{\theta}).$$

If f_1 is convex, g is in $\mathcal{S}_{L,L}(f, \boldsymbol{\kappa})$, and when f_1 is μ-strongly convex, g is in $\mathcal{S}_{L-\mu,L}(f, \boldsymbol{\kappa})$. Surrogate $g(\boldsymbol{\theta})$ can be minimized with proximal gradient step $\boldsymbol{\theta} \leftarrow \arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{2}\left\|\boldsymbol{\kappa} - \frac{1}{L}\nabla f_1(\boldsymbol{\kappa}) - \boldsymbol{\theta}\right\|_2^2 + \frac{1}{L} f_2(\boldsymbol{\theta}) \right\}$.

In Definition 2.4.4, the objective function is differentiable and hence its first-order Taylor expansion can be written as:

$$f(\boldsymbol{\theta}) \overset{\text{Taylor}}{=} f(\boldsymbol{\kappa}) + \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) + g_i(\boldsymbol{\theta}), \tag{2.4.6}$$

where $g_i(\boldsymbol{\theta})$ denotes the difference $f(\boldsymbol{\theta}) - (f(\boldsymbol{\kappa}) + \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}))$. It is easy to check that

$$\begin{aligned}
f(\boldsymbol{\theta}) &= f(\boldsymbol{\kappa}) + \int_0^1 \nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa}))(\boldsymbol{\theta} - \boldsymbol{\kappa}) d\tau \qquad |\mathbf{u} = \boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa}), d\mathbf{u} = (\boldsymbol{\theta} - \boldsymbol{\kappa})d\tau \\
&= f(\boldsymbol{\kappa}) + \int_{\mathbf{u}=\boldsymbol{\kappa}}^{\boldsymbol{\theta}} \nabla f(\mathbf{u}) d\mathbf{u} \qquad |\tau = 0 \Rightarrow \mathbf{u} = \boldsymbol{\kappa} \text{ and } \tau = 1 \Rightarrow \mathbf{u} = \boldsymbol{\theta} \\
&= f(\boldsymbol{\kappa}) + f(\boldsymbol{\theta}) - f(\boldsymbol{\kappa}) = f(\boldsymbol{\theta}).
\end{aligned}$$

Now (2.4.6) can be rewritten as:

$$
\begin{aligned}
f(\boldsymbol{\theta}) &= f(\boldsymbol{\kappa}) + \int_0^1 \nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa}))(\boldsymbol{\theta} - \boldsymbol{\kappa}) d\tau \\
&= f(\boldsymbol{\kappa}) + \int_0^1 \left[\nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa}))(\boldsymbol{\theta} - \boldsymbol{\kappa}) \underbrace{- \boldsymbol{\nabla} f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) + \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa})}_{=0} \right] d\tau \\
&= f(\boldsymbol{\kappa}) + \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) + \int_0^1 \left[\nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa}))(\boldsymbol{\theta} - \boldsymbol{\kappa}) - \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) \right] d\tau
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow g_i(\boldsymbol{\theta}) &= f(\boldsymbol{\theta}) - f(\boldsymbol{\kappa}) - \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) \\
&= \int_0^1 \left[\nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa}))(\boldsymbol{\theta} - \boldsymbol{\kappa}) - \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) \right] d\tau.
\end{aligned}
$$

Absolute value of $g_i(\boldsymbol{\theta})$ is:

$$
\begin{aligned}
|g_i(\boldsymbol{\theta})| &= \left| \int_0^1 \left[\nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa}))(\boldsymbol{\theta} - \boldsymbol{\kappa}) - \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) \right] d\tau \right| \\
&\leq \int_0^1 \left\| \nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa}))(\boldsymbol{\theta} - \boldsymbol{\kappa}) - \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) \right\|_2 d\tau \\
&= \int_0^1 \left\| (\nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa})) - \nabla f(\boldsymbol{\kappa}))(\boldsymbol{\theta} - \boldsymbol{\kappa}) \right\|_2 d\tau \\
&\leq \int_0^1 \left\| \nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa})) - \nabla f(\boldsymbol{\kappa}) \right\|_2 \left\| \boldsymbol{\theta} - \boldsymbol{\kappa} \right\|_2 d\tau \\
&= \left(\int_0^1 \left\| \nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa})) - \nabla f(\boldsymbol{\kappa}) \right\|_2 d\tau \right) \left\| \boldsymbol{\theta} - \boldsymbol{\kappa} \right\|_2 \\
&\overset{*}{\leq} \left(\int_0^1 \tau L \left\| \boldsymbol{\theta} - \boldsymbol{\kappa} \right\|_2 d\tau \right) \left\| \boldsymbol{\theta} - \boldsymbol{\kappa} \right\|_2 \\
&= \frac{L}{2} \left\| \boldsymbol{\theta} - \boldsymbol{\kappa} \right\|_2^2 .
\end{aligned}
$$

(*) Lipschitz continuity has been used here: $\left\| \nabla f(\boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa})) - \nabla f(\boldsymbol{\kappa}) \right\|_2 \leq L \left\| \boldsymbol{\kappa} + \tau(\boldsymbol{\theta} - \boldsymbol{\kappa}) - \boldsymbol{\kappa} \right\|_2 = \tau L \left\| \boldsymbol{\theta} - \boldsymbol{\kappa} \right\|_2$.

With the above calculation and using the fact $\frac{L}{2}\|\boldsymbol{\theta} - \boldsymbol{\kappa}\|_2^2 \geq 0$, the surrogate function can be written as $g(\boldsymbol{\theta}) = f(\boldsymbol{\kappa}) + \nabla f(\boldsymbol{\kappa})(\boldsymbol{\theta} - \boldsymbol{\kappa}) + \frac{L}{2}\|\boldsymbol{\theta} - \boldsymbol{\kappa}\|_2^2 \geq f(\boldsymbol{\theta})$ and the Lipschitz surrogate function $g(\boldsymbol{\theta})$ indeed upper-bounds $f(\boldsymbol{\theta})$. The proof above follows closely the proof of Lemma 1.2.3 in [13].

In Definition 2.4.5, the objective function is splitted into two parts, where f_1 has identical properties with those for the objective in Definition 2.4.4. Hence, similar surrogate can be used. For f_2 surrogate is not needed since f_2 is already convex.

3 Single waveform design in spectrally busy environment

In this chapter, SINR maximization problem of radar signal design is introduced. In spectrally busy environment, radar signal radiation energy is constrained in specified frequency bands. Also fast-time radar-code energy is constrained to make sure that the radar signal can be sent with feasible energy. In addition to these constraints, we require that designed radar-code signal is close to reference signal which is usually a linearly modulated signal. The details about the problem derivation can be found from [8] and [9].

After introduction of SINR-maximization problem, ADMM-type algorithm for solving it is developed [14]. The tools provided in Section 2 are used in the derivation together with some fundamental results from linear algebra. All steps of derivations will be shown in details with explaining figures so that a reader can follow the algorithm construction without a need for additional material.

3.1 SINR maximization problem in spectrally busy environment

Let us consider linearly modulated sinusoidal signals $x(t), x, t \in \mathbb{R}$, which are often used in radar applications due to their beneficial properties (e.g., pulse compression):

$$\begin{aligned} x(t) &= A\cos\left(2\pi\int_0^t f(\tau)d\tau\right) \\ &= A\cos\left(2\pi\int_0^t f_0 + f_\Delta r(\tau)d\tau\right), \end{aligned} \tag{3.1.1}$$

where f_Δ denotes frequency range, $A \in \mathbb{R}$ and $r(\tau) = \alpha\ \tau, \alpha \in \mathbb{R}$. Change in frequency with respect to time is linear:

$$\frac{df}{d\tau} = \alpha\ f_\Delta.$$

Signal $x(t)$ can be decomposed to orthogonal components using trigonometric identity $\cos(a+b) = \cos a \cos b - \sin a \sin b$:

$$
\begin{aligned}
x(t) &= A\cos\left(2\pi\int_0^t f_0 + f_\Delta r(\tau)d\tau\right) \\
&= A\cos\left(2\pi(f_0 + \frac{1}{2}\alpha\ f_\Delta\ t)t\right) \\
&= A\cos\left(2\pi f_0 t + \frac{1}{2}\alpha\ f_\Delta\ t^2\right) \\
&= A\left(\cos(2\pi f_0 t)\cos\left(\frac{1}{2}\alpha\ f_\Delta\ t^2\right) - \sin(2\pi f_0 t)\sin\left(\frac{1}{2}\alpha\ f_\Delta\ t^2\right)\right) \\
&= A(I(t) + Q(t)), \qquad (3.1.2)
\end{aligned}
$$

where $I(t)$ and $Q(t)$ are called in-phase and quadrature components respectively. In complex notations, (3.1.2) can be written as:

$$
x_I(t) = A(I(t) + jQ(t)) = A * Z(t), \qquad (3.1.3)
$$

where $Z(t) = I(t) + jQ(t)$ is called baseband signal of $x(t)$.

To extract only positive frequencies of baseband signal $Z(t)$ (and hence represent the physical signal) single-sided frequency spectrum of $Z(t)$ needs to be considered, and it is given as:

$$
S_+(f) = \mathcal{F}_+\{Z(t)\}(f) = 2u(f)\mathcal{F}\{Z(t)\}(f), \qquad (3.1.4)
$$

where $\mathcal{F}\{Z(t)\}(f) = \int_{\mathbb{R}} Z(t)e^{-j2\pi ft}dt$ is Fourier transform of $Z(t)$ and $u(f)$ is the Heaviside step function. Hence, physical signal is

$$
Z_+(t) = \mathcal{F}^{-1}\{S_+(f)\}(t) = \int_{\mathbb{R}} S_+(f)e^{+j2\pi ft}df. \qquad (3.1.5)
$$

This yields (see for example [1]):

$$
Z_+(t) = s(t) + j\hat{s}(t), \qquad (3.1.6)
$$

where $s(t) = I(t)\cos(2\pi f_0 t) - Q(t)\sin(2\pi f_0 t)$ and $\hat{s}(t) = I(t)\sin(2\pi f_0 t) + Q(t)\cos(2\pi f_0 t)$. To use physical signal representation introduced in (3.1.1)-(3.1.2), signal $s(t)$ is chosen to represent physical signal.

With the use of (3.1.6), the relationship between (3.1.2) and (3.1.3) is expressed as:

$$
\begin{aligned}
x(t) &= \mathrm{Re}\{x_I(t)\}\cos(2\pi f_0 t) - \mathrm{Im}\{x_I(t)\}\sin(2\pi f_0 t) \\
&= I(t)\cos(2\pi f_0 t) - Q(t)\sin(2\pi f_0 t) \\
&= \mathrm{Re}\{x_I(t)e^{j2\pi f_0 t}\}. \qquad (3.1.7)
\end{aligned}
$$

Now baseband representation of $x(t)$ in (3.1.2) can be written as:

$$\begin{aligned} x_I(t) &= A(\cos\left(\frac{1}{2}\alpha\ f_\Delta\ t^2\right) + j\sin\left(\frac{1}{2}\alpha\ f_\Delta\ t^2\right)) \\ &= Ae^{j\frac{1}{2}\alpha f_\Delta t^2}. \end{aligned} \tag{3.1.8}$$

Representation (3.1.8) is called complex representation of chirp signal. In [8] and [9], chirp is defined with $A = 1/\sqrt{N}$, $\alpha = 4\pi$, and $f_\Delta = K_s$.

Let us define the fast-time radar code signal $\mathbf{c}$ (transmitted signal) as a baseband signal of transmitted radar pulse. The fast-time radar code is a digital signal and it can be represented as vector of length N in which each element is linearly modulated subpulse:

$$\mathbf{c} = [c_1, c_2, ..., c_N]^T \in \mathbb{C}^N, \tag{3.1.9}$$

where $c_n, n \in \{1, ..., N\}$ is code element of n'th subpulse. An example of code signal is discrete version of (3.1.8) with parameters $A = 1/\sqrt{N}$, $\alpha = 4\pi$ and $f_\Delta = K_s$:

$$\mathbf{c}_{\text{chirp}}(n) = \frac{1}{\sqrt{N}}e^{j2\pi K_s(nT)^2} = \frac{1}{\sqrt{N}}e^{j2\pi K_s(n/f_s)^2}. \tag{3.1.10}$$

The transmitted signal (the fast-time radar code) is called fast-time observation signal once it reaches radar receiver. The received fast-time observation signal can be written as:

$$\mathbf{v} = [v_1, v_2, ..., v_N]^T \in \mathbb{C}^N. \tag{3.1.11}$$

The observation signal $\mathbf{v}$ can be modeled with the use of the radar code signal $\mathbf{c}$ as:

$$\mathbf{v} = \alpha_T\mathbf{c} + \mathbf{n}, \tag{3.1.12}$$

where $\mathbf{n}$ is called a vector of filtered disturbance echo samples and α_T is a parameter contributing to backscatter effects. Vector $\mathbf{n}$ includes both noise and disturbance signals (e.g. signals from jammers and other radiators) that deteriorate the observation signal quality. Vector $\mathbf{n}$ is usually modeled as a complex-valued, zero-mean Gaussian random vector with covariance matrix:

$$\mathbb{E}\left[\mathbf{n}\mathbf{n}^H\right] = \mathbf{M}. \tag{3.1.13}$$

It is important to notice that the covariance matrix $\mathbf{M}$ in (3.1.13) is Hermitian, i.e., $\mathbf{M}^H = \mathbf{M}$. The energy transmitted by radar to specific bandwith is expressed as:

$$\int_{f_1^k}^{f_2^k} S_c(f)df = \mathbf{c}^H \mathbf{R}_I^k \mathbf{c}, \tag{3.1.14}$$

where f_1^k is the lower-bound and f_2^k is the upper-bound of k'th bandwith and $S_c(f)$ is the energy of radar-code signal frequency components, that is:

$$\begin{aligned} S_c(f) &= |\mathcal{F}_{\mathbb{N}}\{\mathbf{c}(n)\}|^2 \\ &= \left| \sum_{n=-\infty}^{\infty} \mathbf{c}(n) e^{-j2\pi fn} \right|^2 \\ &= \left| \sum_{n=0}^{N-1} \mathbf{c}(n) e^{-j2\pi fn} \right|^2, \end{aligned} \tag{3.1.15}$$

where $\mathcal{F}_{\mathbb{N}}\{\mathbf{c}(n)\}$ denotes the discrete-time Fourier transform (DTFT) of signal $\mathbf{c}$. The matrix $\mathbf{R}_I^k$ can be written as:

$$\mathbf{R}_I^k(m,l) = \begin{cases} f_2^k - f_1^k, & m = l \\ \frac{e^{j2\pi f_2^k (m-l)} - e^{j2\pi f_1^k (m-l)}}{j2\pi(m-l)} & m \neq l, \end{cases} \tag{3.1.16}$$

where $(m,l) \in \{1,...,N\}^2$.

If radar transmits energy over K different frequency bands, the total transmitted energy over the frequency bands is:

$$\sum_{k=1}^{K} \mathbf{c}^H \mathbf{R}_I^k \mathbf{c}. \tag{3.1.17}$$

In spectrally busy environments, it is important to constraint the amount of energy that radar transmits over different frequency bands. Otherwise, radar can deteriorate the performance of other radiators (e.g., telecommunication devices etc.) significantly. Let us denote the total maximum allowed disturbance to frequency bands by E_I. The disturbance constraint can now be expressed as:

$$\mathbf{c}^H \mathbf{R}_I \mathbf{c} \leq E_I, \tag{3.1.18}$$

where the left-hand-side is the weighted sum (with weights $w_k \geq 0$) of energies radiated to frequency bands:

$$\mathbf{R}_I = \sum_{k=1}^{K} w_k \mathbf{R}_I^k. \tag{3.1.19}$$

For designing the fast-time radar code signal, one way to determine quality of the signal is to determine its SINR:

$$\text{SINR} = |\alpha_T|^2 \mathbf{c}^H \mathbf{R} \mathbf{c}, \tag{3.1.20}$$

where $\mathbf{R} = \mathbf{M}^{-1}$ ($\mathbf{M}$ defined in (3.1.13)). Because $\mathbf{M}$ is Hermitian, the inverse $\mathbf{M}^{-1}$ is Hermitian. SINR tells how distinguishable is the given radar code signal $\mathbf{c}$ from disturbance signals. The greater the SINR value the better. Thus, the objective for designing $\mathbf{c}$ is to maximize SINR (3.1.20).

In addition to the energy radiated to frequency bands, the fast-time radar code energy is also constrained to make sure that the transmitted signal can be sent with feasible energy:

$$\|\mathbf{c}\|_2^2 = \mathbf{c}^H \mathbf{c} = 1. \tag{3.1.21}$$

Moreover, to make sure that radar signal is well-behaving (in the sense that for example the radar signal has desired ambiguity function properties) it needs to be close to a given unit energy (i.e., $\|\mathbf{c}_0\|_2 = 1$) reference signal $\mathbf{c}_0$:

$$\|\mathbf{c} - \mathbf{c}_0\|_2^2 \leq \epsilon, \tag{3.1.22}$$

where ϵ defines the similarity region.

To maximize SINR (3.1.20) under constraints (3.1.18), (3.1.21) and (3.1.22), the following maximization problem $\mathcal{P}_1$ is introduced:

$$\mathcal{P}_1 : \begin{cases} \max\limits_{\mathbf{c}} & |\alpha_T|^2 \mathbf{c}^H \mathbf{R} \mathbf{c} & \text{(3.1.23a)} \\ \text{s.t. :} & \|\mathbf{c}\|^2 = 1 & \text{(3.1.23b)} \\ & \mathbf{c}^H \mathbf{R}_I \mathbf{c} \leq E_I & \text{(3.1.23c)} \\ & \|\mathbf{c} - \mathbf{c}_0\|^2 \leq \epsilon. & \text{(3.1.23d)} \end{cases}$$

Because α_T is constant, it can be dropped from the objective function and $\mathcal{P}_1$ simplifies to:

$$\mathcal{P}_1^{(1)} : \begin{cases} \max\limits_{\mathbf{c}} & \mathbf{c}^H \mathbf{R} \mathbf{c} & \text{(3.1.24a)} \\ \text{s.t. :} & \|\mathbf{c}\|^2 = 1 & \text{(3.1.24b)} \\ & \mathbf{c}^H \mathbf{R}_I \mathbf{c} \leq E_I & \text{(3.1.24c)} \\ & \|\mathbf{c} - \mathbf{c}_0\|^2 \leq \epsilon. & \text{(3.1.24d)} \end{cases}$$

Problem $\mathcal{P}_1^{(1)}$ can be equivalently written as minimization problem:

$$\mathcal{P}_1^{(2)}: \begin{cases} \min\limits_{\mathbf{c}} & -\mathbf{c}^H\mathbf{R}\mathbf{c} & (3.1.25a) \\ \text{s.t. :} & \|\mathbf{c}\|^2 = 1 & (3.1.25b) \\ & \mathbf{c}^H\mathbf{R}_I\mathbf{c} \leq E_I & (3.1.25c) \\ & \|\mathbf{c}-\mathbf{c}_0\|^2 \leq \epsilon. & (3.1.25d) \end{cases}$$

3.2 ADMM-type algorithm derivation

Before tackling minimization problem $\mathcal{P}_1^{(2)}$, it is important to notice that objective (3.1.25a) is non-convex because matrix $\mathbf{R}$ is positive semidefinite (PSD). Indeed, it is inverse of PSD noise covariance matrix. If matrix $\mathbf{R}$ was negative semidefinite ($R \preceq 0$) it would guarantee that (3.1.25a) is convex. This can be seen using convexity property (2.4.2):

$$\begin{aligned}
&f(\mathbf{x}) + \nabla f(\mathbf{x})(\mathbf{y}-\mathbf{x}) && |f(\mathbf{x}) = \mathbf{x}^T\mathbf{R}\mathbf{x} \\
&= \mathbf{x}^T\mathbf{R}\mathbf{x} + ((\mathbf{R}+\mathbf{R}^T)\mathbf{x})^T(\mathbf{y}-\mathbf{x}) && |\nabla f(\mathbf{x}) = (\mathbf{R}+\mathbf{R}^T)\mathbf{x} \\
&= \mathbf{x}^T\mathbf{R}\mathbf{x} + \mathbf{x}^T(\mathbf{R}+\mathbf{R}^T)(\mathbf{y}-\mathbf{x}) && \\
&= \mathbf{x}^T\mathbf{R}\mathbf{x} + \mathbf{x}^T\mathbf{R}\mathbf{y} + \mathbf{x}^T\mathbf{R}^T\mathbf{y} - \mathbf{x}^T\mathbf{R}\mathbf{x} - \mathbf{x}^T\mathbf{R}^T\mathbf{x} && \\
&= \mathbf{x}^T\mathbf{R}^T\mathbf{y} + \mathbf{x}^T\mathbf{R}\mathbf{y} - \mathbf{x}^T\mathbf{R}^T\mathbf{x} && |\mathbf{x} = \mathbf{y} + \boldsymbol{\alpha} \\
&= (\mathbf{y}+\boldsymbol{\alpha})^T\mathbf{R}^T\mathbf{y} + (\mathbf{y}+\boldsymbol{\alpha})^T\mathbf{R}\mathbf{y} - (\mathbf{y}+\boldsymbol{\alpha})^T\mathbf{R}^T(\mathbf{y}+\boldsymbol{\alpha}) && \\
&= \mathbf{y}^T\mathbf{R}^T\mathbf{y} + \boldsymbol{\alpha}^T\mathbf{R}\mathbf{y} - \mathbf{y}^T\mathbf{R}^T\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{R}^T\boldsymbol{\alpha} && \\
&= \mathbf{y}^T\mathbf{R}^T\mathbf{y} + \boldsymbol{\alpha}^T\mathbf{R}\mathbf{y} - \left(\boldsymbol{\alpha}^T\mathbf{R}\mathbf{y}\right)^T - \boldsymbol{\alpha}^T\mathbf{R}^T\boldsymbol{\alpha} && |\boldsymbol{\alpha}^T\mathbf{R}\mathbf{y} = \left(\boldsymbol{\alpha}^T\mathbf{R}\mathbf{y}\right)^T \text{ since } \mathbf{R}^T = \mathbf{R} \\
&= \mathbf{y}^T\mathbf{R}^T\mathbf{y} - \boldsymbol{\alpha}^T\mathbf{R}^T\boldsymbol{\alpha} && |\text{If } \mathbf{R} \succeq 0,\ \boldsymbol{\alpha}^T\mathbf{R}^T\boldsymbol{\alpha} \geq 0 \\
&\leq \mathbf{y}^T\mathbf{R}^T\mathbf{y} = f(\mathbf{y}). && |\text{convexity: } f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})(\mathbf{y}-\mathbf{x})
\end{aligned}$$

It is worth noticing that for any $\mathbf{R} \preceq 0$, convexity of $-\mathbf{c}^H\mathbf{R}\mathbf{c}$ holds, not only for Hermitian $\mathbf{R}$, although in this work only Hermitian $\mathbf{R}$ needs to be considered. To ensure that objective (3.1.25a) is convex, let us rewrite $\mathcal{P}_1^{(2)}$ as:

$$\mathcal{P}_1^{(3)}: \begin{cases} \min\limits_{\mathbf{c}} & \mathbf{c}^H\mathbf{Q}\mathbf{c} & (3.2.1a) \\ \text{s.t. :} & \|\mathbf{c}\|^2 = 1 & (3.2.1b) \\ & \mathbf{c}^H\mathbf{R}_I\mathbf{c} \leq E_I & (3.2.1c) \\ & \|\mathbf{c}-\mathbf{c}_0\|^2 \leq \epsilon, & (3.2.1d) \end{cases}$$

where $\mathbf{Q} = \mu\mathbf{I} - \mathbf{R}$, $\mathbf{I}$ is identity matrix and μ is positive constant such that $\mathbf{Q} \succeq 0$. Because $\mathbf{Q} \succeq 0$, the objective function is convex. Also $\mathbf{Q}$ is Hermitian because $\mathbf{R}$ is Hermitian.

Objective (3.2.1a) upper-bounds objective (3.1.25a):

$$\begin{aligned}\mathbf{c}^H\mathbf{Q}\mathbf{c} &= \mathbf{c}^H(\mu\mathbf{I} - \mathbf{R})\mathbf{c} \\ &= \mu\mathbf{c}^H\mathbf{c} - \mathbf{c}^H\mathbf{R}\mathbf{c} \\ &= \underbrace{\mu\|\mathbf{c}\|_2^2}_{\geq 0} - \mathbf{c}^H\mathbf{R}\mathbf{c} \\ &\geq -\mathbf{c}^H\mathbf{R}\mathbf{c},\end{aligned}$$

which implies that the minimum value of objective (3.2.1a) is an approximation of the minimum value of (3.1.25a). Hence transformation from $\mathcal{P}_1^{(2)}$ to $\mathcal{P}_1^{(3)}$ can be seen as Majorization-Minimization step.

Next let us write the complex valued matrix $\mathbf{Q} \in \mathbb{C}^{N\times N}$ and vectors $\mathbf{c}, \mathbf{c}_0 \in \mathbb{C}^N$ in real-valued notations as:

$$\mathbf{Q} = \begin{bmatrix}\mathrm{Re}\{\mathbf{Q}\} & -\mathrm{Im}\{\mathbf{Q}\} \\ \mathrm{Im}\{\mathbf{Q}\} & \mathrm{Re}\{\mathbf{Q}\}\end{bmatrix}, \mathbf{c} = \begin{bmatrix}\mathrm{Re}\{\mathbf{c}\} \\ \mathrm{Im}\{\mathbf{c}\}\end{bmatrix} \text{ and } \mathbf{c}_0 = \begin{bmatrix}\mathrm{Re}\{\mathbf{c}_0\} \\ \mathrm{Im}\{\mathbf{c}_0\}\end{bmatrix}.$$

To justify the transformation above when real and imaginary parts are independent, we check that the product of complex numbers can be written as:

$$\begin{aligned}z_1 z_2 &= (a + jb)(c + jd) \\ &= \mathrm{Re}\{z_1\}\mathrm{Re}\{z_2\} - \mathrm{Im}\{z_1\}\mathrm{Im}\{z_2\} + j\left(\mathrm{Re}\{z_1\}\mathrm{Im}\{z_2\} + \mathrm{Im}\{z_1\}\mathrm{Re}\{z_2\}\right) \\ &\overset{*}{=} \begin{bmatrix}\mathrm{Re}\{z_1\}\mathrm{Re}\{z_2\} - \mathrm{Im}\{z_1\}\mathrm{Im}\{z_2\} \\ \mathrm{Re}\{z_1\}\mathrm{Im}\{z_2\} + \mathrm{Im}\{z_1\}\mathrm{Re}\{z_2\}\end{bmatrix} \\ &= \begin{bmatrix}\mathrm{Re}\{z_2\} & -\mathrm{Im}\{z_2\} \\ \mathrm{Im}\{z_2\} & \mathrm{Re}\{z_2\}\end{bmatrix}\begin{bmatrix}\mathrm{Re}\{z_1\} \\ \mathrm{Im}\{z_1\}\end{bmatrix}.\end{aligned}$$

(*) Move to vector notation $z = a + jb = \begin{bmatrix}a \\ b\end{bmatrix}$.

Also it is worth noting that L^2-norm remains unchanged in this transformation:

$$\begin{aligned}\|z_1 - z_2\|_2^2 &= |z_1 - z_2|^2 \\ &= \left(\mathrm{Re}\{z_1\} - \mathrm{Re}\{z_2\}\right)^2 + \left(\mathrm{Im}\{z_1\} - \mathrm{Im}\{z_2\}\right)^2 \\ &= \begin{bmatrix}\mathrm{Re}\{z_1\} - \mathrm{Re}\{z_2\} & \mathrm{Im}\{z_1\} - \mathrm{Im}\{z_2\}\end{bmatrix}\begin{bmatrix}\mathrm{Re}\{z_1\} - \mathrm{Re}\{z_2\} \\ \mathrm{Im}\{z_1\} - \mathrm{Im}\{z_2\}\end{bmatrix}.\end{aligned}$$

and

$$\begin{aligned}\|z\|_2^2 = |z|^2 &= (\mathrm{Re}\{z\})^2 + (\mathrm{Im}\{z\})^2 \\ &= \begin{bmatrix}\mathrm{Re}\{z\} & \mathrm{Im}\{z\}\end{bmatrix}\begin{bmatrix}\mathrm{Re}\{z\} \\ \mathrm{Im}\{z\}\end{bmatrix}.\end{aligned}$$

Now $\mathcal{P}_1^{(3)}$ can be written as the following real-valued optimization problem:

$$\mathcal{P}_1^{(4)}: \begin{cases} \min\limits_{\mathbf{c}} & \mathbf{c}^T\mathbf{Q}\mathbf{c} & \text{(3.2.2a)} \\ \text{s.t. :} & \|\mathbf{c}\|^2 = 1 & \text{(3.2.2b)} \\ & \mathbf{c}^T\mathbf{R}_I\mathbf{c} \le E_I & \text{(3.2.2c)} \\ & \|\mathbf{c}-\mathbf{c}_0\|^2 \le \epsilon, & \text{(3.2.2d)} \end{cases}$$

where $\mathbf{Q}, \mathbf{R}_I \in \mathbb{R}^{2N\times 2N}$, $\mathbf{c} \in \mathbb{R}^{2N}$, and $\mathbf{c}_0 \in \mathbb{R}^{2N}$.

In $\mathcal{P}_1^{(4)}$, objective (3.2.2a) is convex but not separable which is needed in ADMM. To allow separability let us introduce the slack variable $\mathbf{z}$ and the constraint $\mathbf{c} = \mathbf{z}$. Now, it is easy to write Augmented Lagrangian $L_\rho(\mathbf{c}, \mathbf{z}, \boldsymbol{\lambda})$ for the minimization problem $\min_{\mathbf{c}} \mathbf{c}^T\mathbf{Q}\mathbf{c}$ s.t.: $\mathbf{c} = \mathbf{z}$:

$$L_\rho(\mathbf{c}, \mathbf{z}, \boldsymbol{\lambda}) = \mathbf{c}^T\mathbf{Q}\mathbf{c} + \boldsymbol{\lambda}^T(\mathbf{c}-\mathbf{z}) + \frac{\rho}{2}\|\mathbf{c}-\mathbf{z}\|^2. \tag{3.2.3}$$

ADMM-steps for $\mathcal{P}_1^{(4)}$ are:

$$\begin{cases} \mathbf{c}_{k+1} = \arg\min\limits_{\mathbf{c}} L_\rho(\mathbf{c}, \mathbf{z}_k, \boldsymbol{\lambda}_k) & \text{(3.2.4a)} \\ \mathbf{z}_{k+1} = \arg\min\limits_{\mathbf{z}} L_\rho(\mathbf{c}_{k+1}, \mathbf{z}, \boldsymbol{\lambda}_k) & \text{(3.2.4b)} \\ \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \rho(\mathbf{c}_{k+1} - \mathbf{z}_{k+1}), & \text{(3.2.4c)} \end{cases}$$

where Augmented Lagrangian $L_\rho(\mathbf{c}, \mathbf{z}, \boldsymbol{\lambda})$ is defined in (3.2.3). It is important to notice that constraints (3.2.2b)-(3.2.2d) are not included in Augmented Lagrangian. These constraints need to be addressed either in ADMM step (3.2.4a) or (3.2.4b). The order in which constraints (3.2.2b)-(3.2.2d) are addressed in steps (3.2.4a) and (3.2.4b) does not matter since $\mathbf{z}$ and $\mathbf{c}$ are essentially the same variable, which is ensured by the constraint $\mathbf{c} = \mathbf{z}$.

Step (3.2.4c) ($\boldsymbol{\lambda}$-variable update) is trivial, while steps (3.2.4a) and (3.2.4b) need further simplifications. In Subsections 3.2.1 and 3.2.2, these steps respectively, are simplified to their final forms. Also in Appendix A (Subsection 7.1) step (3.2.4a) is further analyzed. In Subsection 3.2.3, final ADMM-type algorithm for solving problem $\mathcal{P}_1^{(4)}$ is proposed.

3.2.1 c-variable update

c-variable update (3.2.4a) can be written as:

$$\begin{aligned}
\mathbf{c}_{k+1} &= \arg\min_{\mathbf{c}} L_\rho\left(\mathbf{c}, \mathbf{z}_k, \boldsymbol{\lambda}_k\right) \\
&= \arg\min_{\mathbf{c}} \left\{ \mathbf{c}^T\mathbf{Q}\mathbf{c} + \boldsymbol{\lambda}^T(\mathbf{c}-\mathbf{z}) + \frac{\rho}{2}\|\mathbf{c}-\mathbf{z}\|^2 \right\} \\
&= \arg\min_{\mathbf{c}} \left\{ \mathbf{c}^T\mathbf{Q}\mathbf{c} + (\boldsymbol{\lambda}-\rho\mathbf{z})^T\mathbf{c} + \left(\frac{\rho}{2}\left(1+\|\mathbf{z}\|^2 - \boldsymbol{\lambda}^T\mathbf{z}\right)\right) \right\} \\
&= \arg\min_{\mathbf{c}} \left\{ \mathbf{c}^T\mathbf{Q}\mathbf{c} + (\boldsymbol{\lambda}-\rho\mathbf{z})^T\mathbf{c} \right\} \\
&= \arg\min_{\mathbf{c}} h(\mathbf{c}) \qquad\qquad |\text{s.t. } \|\mathbf{c}\|^2 = 1,\ \|\mathbf{c}-\mathbf{c}_0\|^2 \le \epsilon. \qquad (3.2.5)
\end{aligned}$$

To efficiently solve minimization problem (3.2.5), Majorization-Minimization techniques (as introduced in Subsection 2.4) need to be used. Let us split $h(\mathbf{c})$ into two parts $h(\mathbf{c}) = h_1(\mathbf{c}) + h_2(\mathbf{c})$. Now $h_1(\mathbf{c}) = \mathbf{c}^T\mathbf{Q}\mathbf{c}$ is convex and $h_2(\mathbf{c}) = (\boldsymbol{\lambda}-\rho\mathbf{z})^T\mathbf{c} \in C^1$ is continuously differentiable. Gradients of $h_1(\mathbf{c})$, $h_2(\mathbf{c})$, and $h(\mathbf{c})$ are given as:

$$\begin{aligned}
\nabla_{\mathbf{c}} h_1(\mathbf{c}) &= \nabla_{\mathbf{c}}\left(\mathbf{c}^T\mathbf{Q}\mathbf{c}\right) \\
&= \left(\nabla_{\mathbf{c}}\mathbf{c}^T\right)\mathbf{Q}\mathbf{c} + \mathbf{c}^T \\
&= \left(\nabla_{\mathbf{c}}\mathbf{Q}\mathbf{c}\right) = \mathbf{Q}\mathbf{c} + \mathbf{Q}^T\mathbf{c} \\
&= \left(\mathbf{Q}+\mathbf{Q}^T\right)\mathbf{c}, \qquad (3.2.6)
\end{aligned}$$

$$\nabla_{\mathbf{c}} h_2(\mathbf{c}) = (\boldsymbol{\lambda}-\rho\mathbf{z}), \qquad (3.2.7)$$

and

$$\begin{aligned}
\nabla_{\mathbf{c}} h(\mathbf{c}) &= \nabla_{\mathbf{c}} h_1(\mathbf{c}) + \nabla_{\mathbf{c}} h_2(\mathbf{c}) \\
&= \left(\mathbf{Q}+\mathbf{Q}^T\right)\mathbf{c} + (\boldsymbol{\lambda}-\rho\mathbf{z}). \qquad (3.2.8)
\end{aligned}$$

Gradient $\nabla_{\mathbf{c}} h(\mathbf{c})$ is L-Lipschitz continuous because we can find L such that $|\nabla_{\mathbf{c}} h(\boldsymbol{\kappa}) - \nabla_{\mathbf{c}} h(\mathbf{c})| \le L\,|\boldsymbol{\kappa}-\mathbf{c}|$:

$$
\begin{aligned}
|\nabla_{\mathbf{c}} h(\boldsymbol{\kappa}) - \nabla_{\mathbf{c}} h(\mathbf{c})| &= \left| \left(\mathbf{Q} + \mathbf{Q}^T \right) \boldsymbol{\kappa} + (\boldsymbol{\lambda} - \rho \mathbf{z}) - \left(\left(\mathbf{Q} + \mathbf{Q}^T \right) \mathbf{c} + (\boldsymbol{\lambda} - \rho \mathbf{z}) \right) \right| \\
&= \left| \left(\mathbf{Q} + \mathbf{Q}^T \right) \boldsymbol{\kappa} - \left(\mathbf{Q} + \mathbf{Q}^T \right) \mathbf{c} \right| \\
&= \left| \left(\mathbf{Q} + \mathbf{Q}^T \right) (\boldsymbol{\kappa} - \mathbf{c}) \right| \\
&= \left| \left(\mathbf{Q} + \mathbf{Q}^T \right) \right| |(\boldsymbol{\kappa} - \mathbf{c})| \leq L \, |\boldsymbol{\kappa} - \mathbf{c}| \\
&\Rightarrow \left| \left(\mathbf{Q} + \mathbf{Q}^T \right) \right| \leq L \\
&\Leftrightarrow \begin{bmatrix} \left| \left[\mathbf{Q} + \mathbf{Q}^T \right]_1 \right| \leq L \\ \left| \left[\mathbf{Q} + \mathbf{Q}^T \right]_2 \right| \leq L \\ \vdots \\ \left| \left[\mathbf{Q} + \mathbf{Q}^T \right]_{2N} \right| \leq L \end{bmatrix},
\end{aligned}
$$

where $\left[\mathbf{Q} + \mathbf{Q}^T \right]_i$ denotes i'th row of the matrix $\mathbf{Q} + \mathbf{Q}^T$ and L is Lipschitz constant.

Because $h(\mathbf{c}) \in C^1$ and $\nabla_{\mathbf{c}} h(\mathbf{c})$ is L-Lipschitz continuous, according to Definition 2.4.4, the following Lipschitz gradient surrogate $g \in \mathcal{S}_{2L,L}(h, \boldsymbol{\kappa})$ for $h(\mathbf{c})$ can be used:

$$
\begin{aligned}
g(\mathbf{c}) &= h(\boldsymbol{\kappa}) + \nabla h(\boldsymbol{\kappa})(\mathbf{c} - \boldsymbol{\kappa}) + \frac{L}{2} \|\mathbf{c} - \boldsymbol{\kappa}\|_2^2 \\
&= \boldsymbol{\kappa}^T \mathbf{Q} \boldsymbol{\kappa} + (\boldsymbol{\lambda} - \rho \mathbf{z})^T \boldsymbol{\kappa} + \left(\left(\mathbf{Q} + \mathbf{Q}^T \right) \boldsymbol{\kappa} + (\boldsymbol{\lambda} - \rho \mathbf{z}) \right)^T (\mathbf{c} - \boldsymbol{\kappa}) + \frac{L}{2} \|\mathbf{c} - \boldsymbol{\kappa}\|_2^2.
\end{aligned}
$$

Surrogate $g(\mathbf{c})$ can be minimized using gradient descent step:

$$
\begin{aligned}
\mathbf{c} &\leftarrow \boldsymbol{\kappa} - \frac{1}{L} \nabla h(\boldsymbol{\kappa}) \\
&= \boldsymbol{\kappa} - \frac{1}{L} \left(\left(\mathbf{Q} + \mathbf{Q}^T \right) \boldsymbol{\kappa} + (\boldsymbol{\lambda} - \rho \mathbf{z}) \right),
\end{aligned}
$$

where $\boldsymbol{\kappa} \in \Theta = \{ \mathbf{c} \in \mathbb{R}^{2N} \mid \|\mathbf{c}\|^2 = 1 \text{ and } \|\mathbf{c} - \mathbf{c}_0\|^2 \leq \epsilon, \text{ for some } \mathbf{c}_0 \in \mathbb{R}^{2N} \}$. Hence the iteration update becomes:

$$
\mathbf{c}_{k+1} = \mathbf{c}_k - \frac{1}{L} \left(\left(\mathbf{Q} + \mathbf{Q}^T \right) \mathbf{c}_k + (\boldsymbol{\lambda} - \rho \mathbf{z}) \right). \tag{3.2.9}
$$

Gradient descent yields updated $\mathbf{c}$ that has $\|\mathbf{c}\|_2^2 \neq 1$. Cheap way to, possibly suboptimally, project it back to the feasible region is to divide updated $\mathbf{c}$ by its L^2-norm:

$$\hat{\mathbf{c}}_{k+1} = \mathbf{c}_{k+1}/\|\mathbf{c}_{k+1}\|_2 \tag{3.2.10}$$

Now obviously $\|\hat{\mathbf{c}}_{k+1}\|_2^2 = 1$. The idea of the projection is illustrated in Figure 1.

Next, the similarity constraint $\|\mathbf{c} - \mathbf{c}_0\|_2^2 \leq \epsilon$ needs to be considered. The idea is to find the component of $\mathbf{c}_0$ perpendicular to $\mathbf{c}$ and move to that direction length α and then project back to the region $\|\mathbf{c}\|_2^2 = 1$. This operation is performed as long as region $\|\mathbf{c} - \mathbf{c}_0\| \leq \epsilon$ is reached. Component perpendicular to $\mathbf{c}$ can be found by Gram-Schmidt process.

Vector $\mathbf{v}$ can be projected to vector $\mathbf{u}$ by operator:

$$\text{proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\langle \mathbf{v}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u}. \tag{3.2.11}$$

Hence component of $\mathbf{v}$ perpendicular to $\mathbf{u}$ is simply

$$\widetilde{\mathbf{v}} = \mathbf{v} - \text{proj}_{\mathbf{u}}(\mathbf{v}) = \mathbf{v} - \frac{\langle \mathbf{v}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u} \tag{3.2.12}$$

and unit vector to perpendicular direction:

$$\mathbf{e} = \frac{\widetilde{\mathbf{v}}}{\|\widetilde{\mathbf{v}}\|_2}. \tag{3.2.13}$$

With the help of (3.2.11)-(3.2.13) rotation steps for $\mathbf{c}$-variable can be written as in Table 1. The steps mentioned in Table 1 are illustrated in Figure 2.

Table 1: $\mathbf{c}$-variable rotation.

Step 1:	Find component of $\mathbf{c}_0$ perpendicular to $\mathbf{c}$: $\widetilde{\mathbf{c}} = \mathbf{c}_0 - \text{proj}_{\mathbf{c}}(\mathbf{c}_0) = \mathbf{c}_0 - \frac{\langle \mathbf{c}_0, \mathbf{c} \rangle}{\langle \mathbf{c}, \mathbf{c} \rangle} \mathbf{c}$.
Step 2:	Find unit vector to this direction: $\mathbf{e} = \frac{\widetilde{\mathbf{c}}}{\|\widetilde{\mathbf{c}}\|_2}$.
Step 3:	Move the length α in direction of unit vector $\mathbf{e}$: $\mathbf{c}^* = \mathbf{c} + \alpha \mathbf{e}$.
Step 4:	Project $\mathbf{c}^*$ back to region $\|\mathbf{c}\| = 1$.
Step 5:	Check if $\|\mathbf{c}^* - \mathbf{c}_0\|_2^2 \leq \epsilon$. If not repeat steps, otherwise exit.

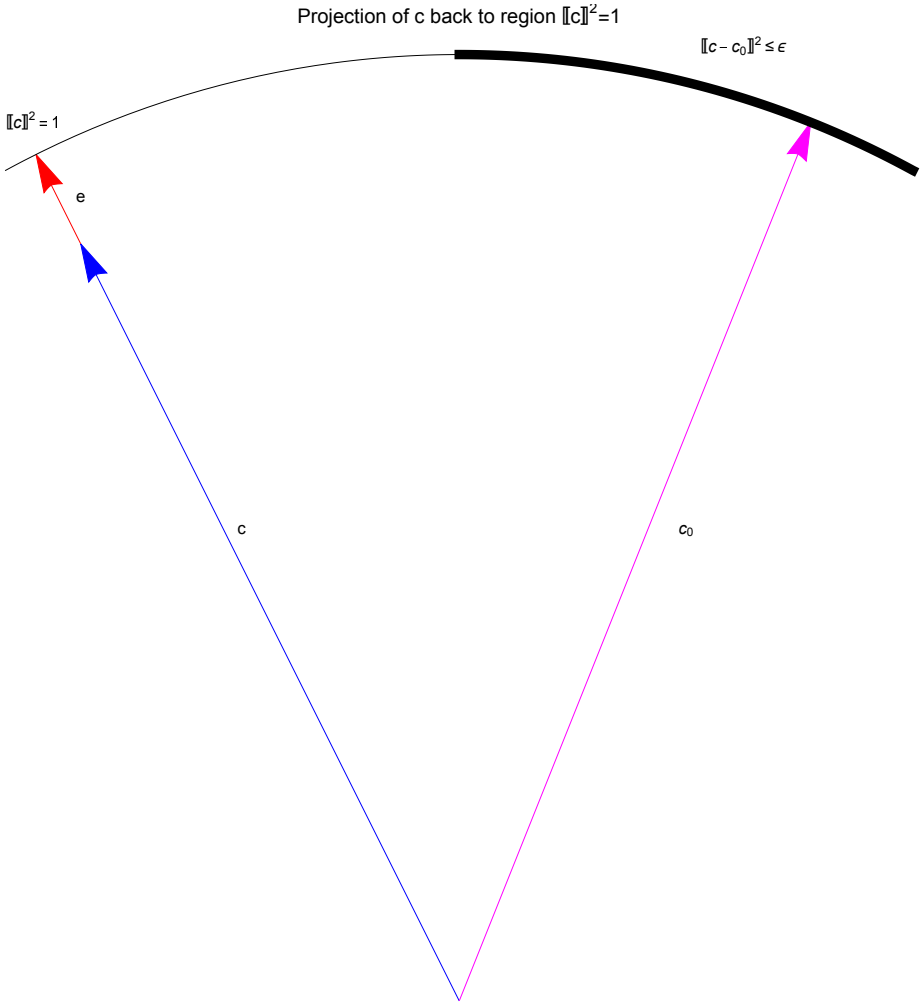

Figure 1: Projection of $\mathbf{c}$ back to region $\|\mathbf{c}\|_2^2 = 1$

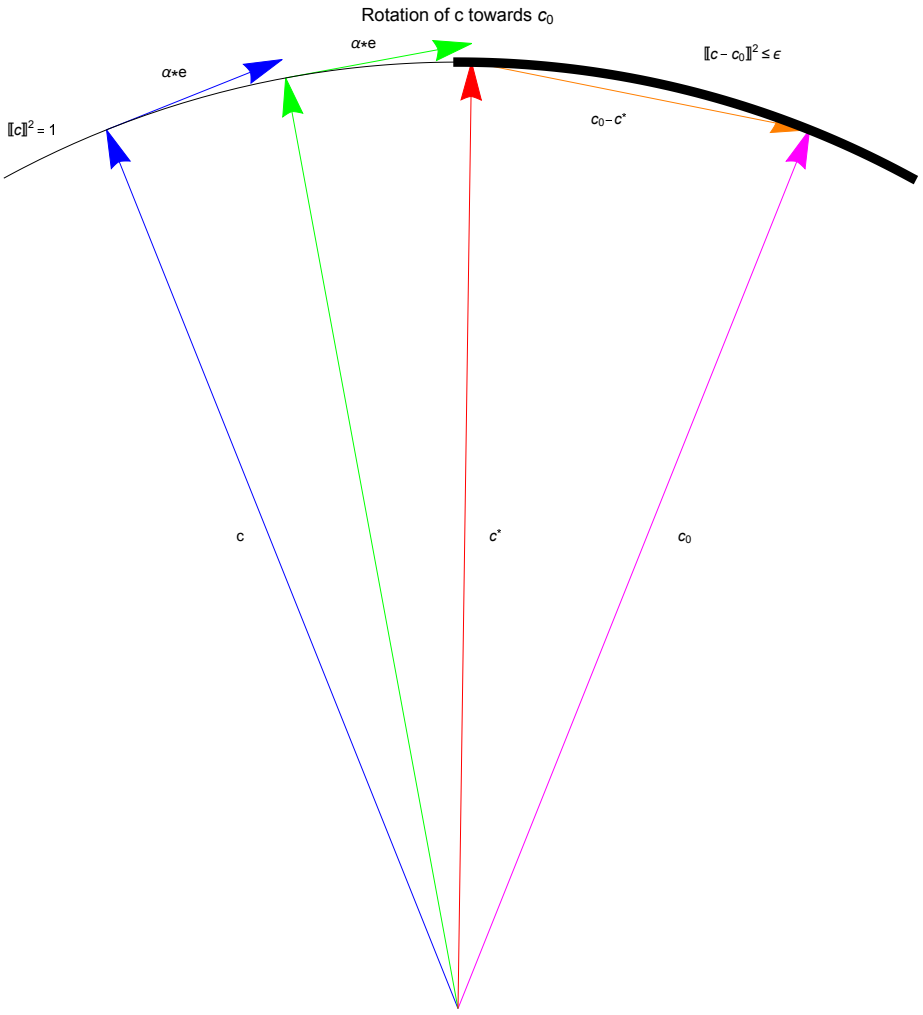

Figure 2: Rotation of $\mathbf{c}$ towards $\mathbf{c}_0$

The steps in Table 1 can be written as a function as shown in Algorithm 1.

Algorithm 1: Rotate $\mathbf{c}$ toward $\mathbf{c}_0$ as long as region $\|\mathbf{c}-\mathbf{c}_0\| \leq \epsilon$ is reached

```
function RotateVector(c, c_0, α, ε);
  Input  : c, c_0, α, and ε
  Output : c
while ‖c − c_0‖ > ε do
    c̃ = c_0 − proj_c(c_0) = c_0 − (⟨c_0, c⟩/⟨c, c⟩) c;
    e = c̃/‖c̃‖_2;
    c* = c + αe;
    c = c*/‖c*‖_2;
end
```

The combination of steps (3.2.9), (3.2.10) and Table 1 are the steps to the projected gradient step for the problem:

$$\min_{\mathbf{c}} g(\mathbf{c}) \text{ subject to } \mathbf{c} \in \Theta. \tag{3.2.14}$$

Projected gradient step for (3.2.14) is:

$$\begin{cases} \mathbf{y}_{k+1} = \mathbf{c}_k - \dfrac{1}{L}\nabla h(\mathbf{c}_k) & \text{(3.2.15a)} \\ \mathbf{c}_{k+1} = \underset{\mathbf{c}\in\Theta}{\arg\min} \left\|\mathbf{y}_{k+1} - \mathbf{c}\right\|_2 . & \text{(3.2.15b)} \end{cases}$$

The region Θ is a part of $2N$-dimensional sphere surface. By using angular coordinates $\boldsymbol{\phi} = [\phi_1,\ \phi_2,\ ...,\ \phi_{2N-1}]$, where $c_1 = \cos(\phi_1)\,, c_n = \left(\prod_{j=1}^{n-1}\sin(\phi_j)\right)\cos(\phi_n)\,,$ $n \in [2, 2N-1]$ and $c_{2N} = \prod_{j=1}^{2N-1}\sin(\phi_j)$, $\mathbf{c} \in \Theta$ can be expressed as a function of $\boldsymbol{\phi}$. Hence step (3.2.15b) can be written as:

$$\mathbf{c}_{k+1} = \underset{\boldsymbol{\phi}\in\Omega}{\arg\min} \left\|\mathbf{y}_{k+1} - \mathbf{c}(\boldsymbol{\phi})\right\|_2 , \tag{3.2.16}$$

where $\Omega = \{\boldsymbol{\phi} \in \mathbb{R}^{2N-1} \mid \underbrace{\|\mathbf{c}(\boldsymbol{\phi})\|_2^2 = 1}_{\text{satisfied } \forall\boldsymbol{\phi}} \text{ and } \|\mathbf{c}(\boldsymbol{\phi}) - \mathbf{c}_0(\boldsymbol{\phi})\|_2^2 \leq \epsilon\}$
$= \left\{\boldsymbol{\phi} \in \mathbb{R}^{2N-1} \mid \|\mathbf{c}(\boldsymbol{\phi}) - \mathbf{c}_0(\boldsymbol{\phi})\|_2^2 \leq \epsilon\right\}.$

Let us define optimal $\boldsymbol{\phi}^* \in \mathbb{R}^{2N-1}$ for which $g(\mathbf{c}(\boldsymbol{\phi}^*)) \leq g(\mathbf{c}(\boldsymbol{\phi}))\ , \forall\boldsymbol{\phi} \in \mathbb{R}^{2N-1}$. Let us also assume that $\boldsymbol{\phi}^*$ is reached by the projection $\hat{\mathbf{y}}_{k+1} = \mathbf{y}_{k+1}/\|\mathbf{y}_{k+1}\|_2 = \hat{\mathbf{y}}_{k+1}(\boldsymbol{\phi}^*)$. We can show that $g(\mathbf{c}(\boldsymbol{\phi}))$ is convex in Ω by writing the Taylor ex-

pansion of g:

$$
\begin{aligned}
&g(\mathbf{c}) = \boldsymbol{\kappa}^T\mathbf{Q}\boldsymbol{\kappa} + (\boldsymbol{\lambda} - \rho\mathbf{z})^T\boldsymbol{\kappa} + ((\mathbf{Q}+\mathbf{Q}^T)\mathbf{c} + (\boldsymbol{\lambda} - \rho\mathbf{z}))^T(\mathbf{c} - \boldsymbol{\kappa}) + \frac{1}{2}\|\mathbf{c} - \boldsymbol{\kappa}\|_2^2,\\
&\nabla_{\mathbf{c}} g(\mathbf{c}) = (\mathbf{Q}+\mathbf{Q}^T)(2\mathbf{c} - \boldsymbol{\kappa}) + (\boldsymbol{\lambda} - \rho\mathbf{z}) + L(\mathbf{c} - \boldsymbol{\kappa}),\\
&\nabla_{\mathbf{cc}} g(\mathbf{c}) = 2(\mathbf{Q}+\mathbf{Q}^T) + L\mathbf{I} \succeq \mathbf{0},\\
&\nabla_{\mathbf{ccc}} g(\mathbf{c}) = 0\\
&\Rightarrow g(\mathbf{c}+\mathbf{r}) = g(\mathbf{c}) + \mathbf{r}^T\nabla_{\mathbf{c}} g(\mathbf{c}) + \frac{1}{2}\mathbf{r}^T\nabla_{\mathbf{cc}} g(\mathbf{c})\mathbf{r},\ \forall \mathbf{r} \in \mathbb{R}^N\\
&\Rightarrow g(\mathbf{c}+\mathbf{r}) - g(\mathbf{c}) = \mathbf{r}^T\nabla_{\mathbf{c}} g(\mathbf{c}) + \underbrace{\frac{1}{2}\mathbf{r}^T\nabla_{\mathbf{cc}} g(\mathbf{c})\mathbf{r}}_{\geq 0, \forall \mathbf{r} \in \mathbb{R}^N},\ \forall \mathbf{r} \in \mathbb{R}^N,
\end{aligned}
\tag{3.2.17}
$$

where $\mathbf{r}$ is a small step to a certain direction. From (3.2.17) it can be seen that g fulfills convexity property $g(\mathbf{c}+\mathbf{r}) - g(\mathbf{c}) \geq \mathbf{r}^T\nabla_{\mathbf{c}} g(\mathbf{c})$ because $\nabla_{\mathbf{cc}} g(\mathbf{c}) \succeq \mathbf{0}$. By choosing $\mathbf{r}$ such that $\|\mathbf{c}+\mathbf{r}\|_2 = 1$ and $\|\mathbf{c}+\mathbf{r}-\mathbf{c}_0\|_2 \leq \epsilon$ (i.e. $\mathbf{c}+\mathbf{r} \in \Theta$), it is clear that g is convex in the region Θ. The operation $\mathbf{c}+\mathbf{r} \in \Theta$ is equal to $\mathbf{c}(\boldsymbol{\phi}+\boldsymbol{\phi}')$ for some $\boldsymbol{\phi}' \in \mathbb{R}^{2N-1}$ such that $\boldsymbol{\phi}+\boldsymbol{\phi}' \in \Omega$.

Since g is convex and $\nabla_{\mathbf{cc}} g(\mathbf{c}) = 2(\mathbf{Q}+\mathbf{Q}^T) + L\mathbf{I} \succeq \mathbf{0}$, we have $g(\boldsymbol{\phi}_1) \leq g(\boldsymbol{\phi}_2)$ if $\|\boldsymbol{\phi}_1 - \boldsymbol{\phi}^*\|_2 \ll \|\boldsymbol{\phi}_2 - \boldsymbol{\phi}^*\|_2$. This can be seen by choosing $\mathbf{r}_1$ and $\mathbf{r}_2$ such that $\|\mathbf{r}_1\|_2 \ll \|\mathbf{r}_2\|_2$ and $\mathbf{c}(\boldsymbol{\phi}^*) + \mathbf{r}_1 \in \Theta$ and $\mathbf{c}(\boldsymbol{\phi}^*) + \mathbf{r}_2 \in \Theta$. Now we have:

$$
\begin{aligned}
&g(\mathbf{c}(\boldsymbol{\phi}^*) + \mathbf{r}_2) - g(\mathbf{c}(\boldsymbol{\phi}^*) + \mathbf{r}_1)\\
&= \mathbf{r}_2^T\underbrace{\nabla_{\mathbf{c}} g(\mathbf{c}(\boldsymbol{\phi}^*))}_{=0} + \frac{1}{2}\mathbf{r}_2^T\nabla_{\mathbf{cc}} g(\mathbf{c}(\boldsymbol{\phi}^*))\mathbf{r}_2 - (\mathbf{r}_1^T\underbrace{\nabla_{\mathbf{c}} g(\mathbf{c}(\boldsymbol{\phi}^*))}_{=0} + \frac{1}{2}\mathbf{r}_1^T\nabla_{\mathbf{cc}} g(\mathbf{c}(\boldsymbol{\phi}^*))\mathbf{r}_1)\\
&= \frac{1}{2}\mathbf{r}_2^T\nabla_{\mathbf{cc}} g(\mathbf{c}(\boldsymbol{\phi}^*))\mathbf{r}_2 - \frac{1}{2}\mathbf{r}_1^T\nabla_{\mathbf{cc}} g(\mathbf{c}(\boldsymbol{\phi}^*))\mathbf{r}_1 \geq 0.
\end{aligned}
$$

This allows to rewrite (3.2.16) as

$$
\begin{cases}
\boldsymbol{\phi}_{k+1} = \underset{\boldsymbol{\phi} \in \Omega}{\arg\min}\, \|\boldsymbol{\phi}^* - \boldsymbol{\phi}\|_2 & \text{(3.2.18a)}\\
\mathbf{c}_{k+1} = \mathbf{c}(\boldsymbol{\phi}_{k+1}). & \text{(3.2.18b)}
\end{cases}
$$

It is important to notice that (3.2.18) is a convex problem, while (3.2.15) is not because Θ is not convex region. Problem (3.2.18a) can be solved using rotation step introduced in Table 1. One complete $\mathbf{c}$-variable update is illustrated in Appendix A.

3.2.2 z-variable update

z-variable update (3.2.4b) can be written as:

$$\begin{aligned}
\mathbf{z}_{k+1} &= \arg\min_{\mathbf{z}} L_\rho\left(\mathbf{c}_{k+1}, \mathbf{z}, \boldsymbol{\lambda}_k\right) \\
&= \arg\min_{\mathbf{z}} \left\{ \mathbf{c}^T\mathbf{Q}\mathbf{c} + \boldsymbol{\lambda}^T(\mathbf{c}-\mathbf{z}) + \frac{\rho}{2}\|\mathbf{c}-\mathbf{z}\|^2 \right\} \\
&= \arg\min_{\mathbf{z}} \left\{ \boldsymbol{\lambda}^T(\mathbf{c}-\mathbf{z}) + \frac{\rho}{2}\|\mathbf{c}-\mathbf{z}\|^2 \right\} \\
&= \arg\min_{\mathbf{z}} u(\mathbf{z}). \qquad \text{|s.t. } \mathbf{z}^T\mathbf{R}_I\mathbf{z} \le E_I.
\end{aligned} \tag{3.2.19}$$

Let us rewrite the objective function $u(\mathbf{z})$ as:

$$\begin{aligned}
u(\mathbf{z}) &= \boldsymbol{\lambda}^T(\mathbf{c}-\mathbf{z}) + \frac{\rho}{2}\|\mathbf{c}-\mathbf{z}\|^2 \\
&= \boldsymbol{\lambda}^T(\mathbf{c}-\mathbf{z}) + \frac{\rho}{2}(\mathbf{c}-\mathbf{z})^T(\mathbf{c}-\mathbf{z}) \\
&= \boldsymbol{\lambda}^T(\mathbf{c}-\mathbf{z}) + \frac{\rho}{2}(\mathbf{c}-\mathbf{z})^T(\mathbf{c}-\mathbf{z}) + \frac{1}{2\rho}\boldsymbol{\lambda}^T\boldsymbol{\lambda} - \frac{1}{2\rho}\boldsymbol{\lambda}^T\boldsymbol{\lambda} \\
&= \frac{\rho}{2}(\frac{2}{\rho}\boldsymbol{\lambda}^T(\mathbf{c}-\mathbf{z}) + (\mathbf{c}-\mathbf{z})^T(\mathbf{c}-\mathbf{z}) + \frac{1}{\rho^2}\boldsymbol{\lambda}^T\boldsymbol{\lambda}) - \frac{1}{2\rho}\boldsymbol{\lambda}^T\boldsymbol{\lambda} \\
&= \frac{\rho}{2}(\frac{1}{\rho}\boldsymbol{\lambda}^T(\mathbf{c}-\mathbf{z}) + \frac{1}{\rho}(\mathbf{c}-\mathbf{z})^T\boldsymbol{\lambda} + (\mathbf{c}-\mathbf{z})^T(\mathbf{c}-\mathbf{z}) + \frac{1}{\rho^2}\boldsymbol{\lambda}^T\boldsymbol{\lambda}) - \frac{1}{2\rho}\boldsymbol{\lambda}^T\boldsymbol{\lambda} \\
&= \frac{\rho}{2}((\mathbf{c}-\mathbf{z}) + \frac{1}{\rho}\boldsymbol{\lambda})^T((\mathbf{c}-\mathbf{z}) + \frac{1}{\rho}\boldsymbol{\lambda})) - \frac{1}{2\rho}\boldsymbol{\lambda}^T\boldsymbol{\lambda} \\
&= \frac{\rho}{2}\left\|\mathbf{c}-\mathbf{z}+\frac{1}{\rho}\boldsymbol{\lambda}\right\|_2^2 - \frac{1}{2\rho}\|\boldsymbol{\lambda}\|_2^2.
\end{aligned}$$

The above manipulation is essentially the same as converting ADMM steps to scaled form (see Subsection 2.3). Now minimization problem (3.2.19) can be written as:

$$\begin{aligned}
\mathbf{z}_{k+1} &= \arg\min_{\mathbf{z}} \left\{ \boldsymbol{\lambda}^T(\mathbf{c}-\mathbf{z}) + \frac{\rho}{2}\|\mathbf{c}-\mathbf{z}\|^2 \right\} \\
&= \arg\min_{\mathbf{z}} \left\{ \frac{\rho}{2}\left\|\mathbf{c}-\mathbf{z}+\frac{1}{\rho}\boldsymbol{\lambda}\right\|_2^2 - \frac{1}{2\rho}\|\boldsymbol{\lambda}\|_2^2 \right\} \\
&= \arg\min_{\mathbf{z}} \left\{ \left\|\mathbf{c}-\mathbf{z}+\frac{1}{\rho}\boldsymbol{\lambda}\right\|_2^2 \right\} \\
&= \arg\min_{\mathbf{z}} \left\{ \left\|\mathbf{z}-(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda})\right\|_2^2 \right\} \qquad \text{|s.t. } \mathbf{z}^T\mathbf{R}_I\mathbf{z} \le E_I.
\end{aligned} \tag{3.2.20}$$

It is important to notice that minimization problem (3.2.20) is non-convex because matrix $\mathbf{R}_I$ is indefinite. Lagrangian for (3.2.20) is given as:

$$L(\mathbf{z}, \gamma) = \left\| \mathbf{z} - (\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}) \right\|_2^2 + \gamma(\mathbf{z}^T \mathbf{R}_I \mathbf{z} - E_I), \tag{3.2.21}$$

where $\gamma \in \mathbb{R}$ is the Lagrange multiplier.

Karush-Kuhn-Tucker (KKT) conditions for minimization problem (3.2.20) are given as:

$$\begin{cases} \nabla_{\mathbf{z}} L(\mathbf{z}^*, \gamma^*) = 0 & \text{(3.2.22a)} \\ \gamma^* \geq 0 & \text{(3.2.22b)} \\ \gamma^*((\mathbf{z}^*)^T \mathbf{R}_I \mathbf{z}^* - E_I) = 0 & \text{(3.2.22c)} \\ (\mathbf{z}^T \mathbf{R}_I \mathbf{z} - E_I) \leq 0 & \text{(3.2.22d)} \\ \nabla_{\mathbf{zz}} L(\mathbf{z}^*, \gamma^*) \succeq 0, & \text{(3.2.22e)} \end{cases}$$

where $\mathbf{z}^*$ and γ^* denote critical points of the Lagrangian $L(\mathbf{z}, \gamma)$. From (3.2.22a), we find:

$$\begin{aligned} \nabla_{\mathbf{z}} L(\mathbf{z}^*, \gamma^*) &\overset{*}{=} \gamma^*(\mathbf{R}_I + \mathbf{R}_I^T)\mathbf{z}^* + 2(\mathbf{z}^* - (\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda})) = 0 \\ &\Rightarrow \left(\mathbf{I} + \gamma^* \left(\frac{\mathbf{R}_I + \mathbf{R}_I^T}{2} \right) \right) \mathbf{z}^* = \mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \\ &\overset{**}{\Leftrightarrow} (\mathbf{I} + \gamma^* \mathbf{R}_I)\, \mathbf{z}^* = \mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}. \end{aligned} \tag{3.2.23}$$

(*) $\nabla_{\mathbf{z}} \gamma(\mathbf{z}^T \mathbf{R}_I \mathbf{z} - E_I) = \nabla_{\mathbf{z}} \gamma \mathbf{z}^T \mathbf{R}_I \mathbf{z} = \gamma(\mathbf{R}_I + \mathbf{R}_I^T)\mathbf{z}$ and
$\nabla_{\mathbf{z}} \left\| \mathbf{z} - (\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right\|_2^2 = \nabla_{\mathbf{z}} \left(\mathbf{z} - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right) \right)^T \left(\mathbf{z} - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right) \right)$
$= \left(\mathbf{z} - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right) \right) + \left(\mathbf{z} - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right) \right) = 2 \left(\mathbf{z} - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right) \right)$
(**) For symmetric matrices $\frac{\mathbf{A} + \mathbf{A}^T}{2} = \mathbf{A}$.

Now with (3.2.22c) and (3.2.23), the iteration steps for γ_{k+2} and $\mathbf{z}_{k+1}$ can be written as:

$$\begin{cases} \gamma_{k+2} \left(\mathbf{z}_{k+1}^T \mathbf{R}_I \mathbf{z}_{k+1} - E_I \right) = 0 & \text{(3.2.24a)} \\ (\mathbf{I} + \gamma_{k+2} \mathbf{R}_I)\, \mathbf{z}_{k+1} = \mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}. & \text{(3.2.24b)} \end{cases}$$

Solve (3.2.24b) for $\gamma_{k+1} > 0$:

$$\mathbf{z}_{k+1} = \left(\mathbf{I} + \gamma_{k+1}\mathbf{R}_I\right)^{-1}\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right), \tag{3.2.25}$$

where γ_{k+1} can be found as the solution to (3.2.24a):

$$\begin{aligned}
&\mathbf{z}_{k+1}^T\mathbf{R}_I\mathbf{z}_{k+1} - E_I = 0 \\
&\overset{*}{\Rightarrow} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T \left(\mathbf{I} + \gamma_{k+1}\mathbf{R}_I\right)^{-1}\mathbf{R}_I\left(\mathbf{I} + \gamma_{k+1}\mathbf{R}_I\right)^{-1}\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) = E_I. \quad (3.2.26)
\end{aligned}$$

(*) $\left(\left(\mathbf{I} + \gamma_{k+1}\mathbf{R}_I\right)^{-1}\right)^T = \left(\mathbf{I} + \gamma_{k+1}\mathbf{R}_I\right)^{-1}$ (see appendix B).

We can further simplify (3.2.25):

$$\begin{aligned}
\mathbf{z}_{k+1} &= \left(\mathbf{I} + \gamma_{k+1}\mathbf{R}_I\right)^{-1}\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) \\
&= \left(\mathbf{I} - \sum_{n=1}^{2N}\frac{\gamma_{k+1}\sigma_n}{1 + \gamma_{k+1}\sigma_n}\mathbf{p}_n\mathbf{p}_n^T\right)\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) \qquad (3.2.27)
\end{aligned}$$

where σ_n is n'th eigenvalue and $\mathbf{p}_n$ is the corresponding eigenvector of $\mathbf{R}_I$, which can be precomputed for given $\mathbf{R}_I$.

3.2.3 Proposed algorithm

By collecting the results from Subsections 3.2.1 and 3.2.2, The ADMM steps (3.2.4a)-(3.2.4c) can be written as Algorithm 2.

Algorithm 2: Single waveform design algorithm

1 function Singlewaveform($\mathbf{Q}, \mathbf{c}_0, \mathbf{R}_I, E_I, \epsilon, K$);
  Input : $\mathbf{Q} = \mu\mathbf{I} - \mathbf{R} \succeq \mathbf{0}$, $\mathbf{c}_0$, $\mathbf{R}_I$, E_I, ϵ and K
  Output : $\mathbf{c}$
2 Initialize $\mathbf{c}$, $\mathbf{z}$ and $\boldsymbol{\lambda}$;
3 **for** $k = 1, k \leq K, k$ ++ **do**
4 $\quad \hat{\mathbf{c}}_{k+1} = \mathbf{c}_k - \frac{1}{L}\left(\left(\mathbf{Q} + \mathbf{Q}^T\right)\mathbf{c}_k + (\boldsymbol{\lambda} - \rho\mathbf{z})\right)$;
5 $\quad \widetilde{\mathbf{c}}_{k+1} = \frac{\hat{\mathbf{c}}_{k+1}}{\|\hat{\mathbf{c}}_{k+1}\|}$;
6 $\quad \mathbf{c}_{k+1} = \text{RotateVector}(\widetilde{\mathbf{c}}_{k+1}, \mathbf{c}_0, \alpha, \epsilon)$;
7 $\quad$ Solve $\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T (\mathbf{I} + \gamma_{k+1}\mathbf{R}_I)^{-1} \mathbf{R}_I (\mathbf{I} + \gamma_{k+1}\mathbf{R}_I)^{-1} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) = E_I$ for $\gamma_{k+1} > 0$;
8 $\quad \mathbf{z}_{k+1} = (\mathbf{I} + \gamma_{k+1}\mathbf{R}_I)^{-1}\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) = \left(\mathbf{I} - \sum_{n=1}^{2N} \frac{\gamma_{k+1}\sigma_n}{1+\gamma_{k+1}\sigma_n}\mathbf{p}_n\mathbf{p}_n^T\right)\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)$;
9 $\quad \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \rho(\mathbf{c}_{k+1} - \mathbf{z}_{k+1})$;
10 **end**

In Algorithm 2, function RotateVector is defined as in Algorithm 1 (see Subsection 3.2.1).

One way to efficiently solve the equation in line 7 of Algorithm 2 is to define function

$$\begin{aligned} f(\gamma) &= \left(\mathbf{c} + \frac{\boldsymbol{\lambda}}{\rho}\right)^T (\mathbf{I} + \gamma_{k+1}\mathbf{R}_I)^{-1} \mathbf{R}_I (\mathbf{I} + \gamma_{k+1}\mathbf{R}_I)^{-1} \left(\mathbf{c} + \frac{\boldsymbol{\lambda}}{\rho}\right) - E_I \\ &= \left(\mathbf{c} + \frac{\boldsymbol{\lambda}}{\rho}\right)^T \mathbf{X}(\gamma) \left(\mathbf{c} + \frac{\boldsymbol{\lambda}}{\rho}\right) - E_I, \end{aligned} \tag{3.2.28}$$

and find roots of f (i.e., find $f(\gamma) = 0$) using Newton's method.

By Woodbury's matrix identity:

$$(\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}, \tag{3.2.29}$$

where the kernel matrix $\mathbf{X}(\gamma)$ can be written as:

$$
\begin{aligned}
\mathbf{X}(\gamma) &= \left(\mathbf{I}+\gamma_{k+1}\mathbf{R}_I\right)^{-1}\mathbf{R}_I\left(\mathbf{I}+\gamma_{k+1}\mathbf{R}_I\right)^{-1} \\
&= \left(\mathbf{I}+\gamma_{k+1}\mathbf{P}\mathbf{D}\mathbf{P}^{-1}\right)^{-1}\mathbf{P}\mathbf{D}\mathbf{P}^{-1}\left(\mathbf{I}+\gamma_{k+1}\mathbf{P}\mathbf{D}\mathbf{P}^{-1}\right)^{-1} \\
&= \left(\mathbf{I}-\mathbf{P}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{P}^{-1}\right)\mathbf{P}\mathbf{D}\mathbf{P}^{-1}\left(\mathbf{I}-\mathbf{P}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{P}^{-1}\right) \\
&= \left(\mathbf{P}\mathbf{D}\mathbf{P}^{-1}-\mathbf{P}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{D}\mathbf{P}^{-1}\right)\left(\mathbf{I}-\mathbf{P}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{P}^{-1}\right) \\
&= \mathbf{P}\mathbf{D}\mathbf{P}^{-1}-\mathbf{P}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{D}\mathbf{P}^{-1}-\mathbf{P}\mathbf{D}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{P}^{-1}+ \\
&\quad \mathbf{P}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{D}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{P}^{-1} \\
&\overset{*}{=} \mathbf{P}\mathbf{D}\mathbf{P}^{-1}-\mathbf{P}\mathbf{C}\mathbf{D}\mathbf{P}^{-1}-\mathbf{P}\mathbf{D}\mathbf{C}\mathbf{P}^{-1}+\mathbf{P}\mathbf{C}\mathbf{D}\mathbf{C}\mathbf{P}^{-1} \\
&= \mathbf{P}\mathbf{D}\mathbf{P}^{-1}+\mathbf{P}\mathbf{C}\mathbf{D}\mathbf{C}\mathbf{P}^{-1}-2\mathbf{P}\mathbf{C}\mathbf{D}\mathbf{P}^{-1} \\
&= \mathbf{P}\left(\mathbf{D}+\mathbf{C}\mathbf{D}\mathbf{C}-2\mathbf{C}\mathbf{D}\right)\mathbf{P}^{-1} \\
&= \mathbf{P}\left(\mathbf{C}^2-2\mathbf{C}+\mathbf{I}\right)\mathbf{D}\mathbf{P}^{-1},
\end{aligned}
$$

$$
(*)\ \mathbf{C} = \left(\tfrac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1} = \begin{bmatrix} \frac{\sigma_1\gamma}{1+\sigma_1\gamma} & & \\ & \ddots & \\ & & \frac{\sigma_{2N}\gamma}{1+\sigma_{2N}\gamma} \end{bmatrix}.
$$

where $\mathbf{R}_I = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ is eigenvalue decomposition of $\mathbf{R}_I$. Let the vector of eigenvalues be denoted as $\boldsymbol{\sigma} = [\sigma_1 \ ... \ \sigma_{2N}]$.

The derivative of the kernel matrix $\mathbf{X}(\gamma)$ is:

$$
\begin{aligned}
\frac{d\mathbf{X}}{d\gamma} &= -\mathbf{P}\left[\frac{d}{d\gamma}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\right]\mathbf{D}\mathbf{P}^{-1}-\mathbf{P}\mathbf{D}\left[\frac{d}{d\gamma}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\right]\mathbf{P}^{-1}+ \\
&\quad \mathbf{P}\left[\frac{d}{d\gamma}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\right]\mathbf{D}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{P}^{-1}+ \\
&\quad \mathbf{P}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{D}\left[\frac{d}{d\gamma}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\right]\mathbf{P}^{-1},
\end{aligned}
$$

where $\frac{d}{d\gamma}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}$ is simply:

$$\frac{d}{d\gamma}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1} = \mathbf{G}(\gamma) = \begin{bmatrix} \frac{d}{d\gamma}\frac{\sigma_1\gamma}{1+\sigma_1\gamma} & & \\ & \ddots & \\ & & \frac{d}{d\gamma}\frac{\sigma_{2N}\gamma}{1+\sigma_{2N}\gamma} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{\sigma_1(1+\sigma_1\gamma)-\sigma_1^2\gamma}{(1+\sigma_1\gamma)^2} & & \\ & \ddots & \\ & & \frac{\sigma_{2N}(1+\sigma_{2N}\gamma)-\sigma_{2N}^2\gamma}{(1+\sigma_{2N}\gamma)^2} \end{bmatrix}.$$

Now it is easy to write the derivative of f with respect to γ as:

$$\begin{aligned}
\frac{df}{d\gamma} &= \left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right)^T \frac{d\mathbf{X}}{d\gamma}\left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right) \\
&= \left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right)^T \left(-\mathbf{PGDP}^{-1}-\mathbf{PDGP}^{-1}+\mathbf{PGD}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{P}^{-1}+\right. \\
&\left.\mathbf{P}\left(\frac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1}\mathbf{DGP}^{-1}\right)\left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right) \\
&\overset{*}{=} \left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right)^T \left(-\mathbf{PGDP}^{-1}-\mathbf{PDGP}^{-1}+\mathbf{PGDCP}^{-1}+\mathbf{PCDGP}^{-1}\right)\left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right) \\
&= \left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right)^T \left(-2\mathbf{PGDP}^{-1}+2\mathbf{PGDCP}^{-1}\right)\left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right) \\
&= 2\left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right)^T \left(\mathbf{PGDCP}^{-1}-\mathbf{PGDP}^{-1}\right)\left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right) \\
&= 2\left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right)^T \left(\mathbf{P}\left(\mathbf{GDC}+\mathbf{GD}\right)\mathbf{P}^{-1}\right)\left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right) \\
&= 2\left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right)^T \left(\mathbf{PGD}\left(\mathbf{C}+\mathbf{I}\right)\mathbf{P}^{-1}\right)\left(\mathbf{c}+\frac{\boldsymbol{\lambda}}{\rho}\right). \qquad (3.2.30)
\end{aligned}$$

$$(*)\ \mathbf{C} = \left(\tfrac{1}{\gamma}\mathbf{D}^{-1}+\mathbf{I}\right)^{-1} = \begin{bmatrix} \frac{\sigma_1\gamma}{1+\sigma_1\gamma} & & \\ & \ddots & \\ & & \frac{\sigma_{2N}\gamma}{1+\sigma_{2N}\gamma} \end{bmatrix}.$$

Newton's method for solving the equation in line 7 of Algorithm 2 is summarized in Algorithm 3.

Algorithm 3: Newton's method to find positive root of f

```
1  function NewtonMethod(x_0, N);
   Input  : x_0 and N
   Output : x
2  while true do
3  |  for k = 1, k ≤ N, k ++ do
4  |  |  x_k = x_{k-1} − f(x_{k-1}) / f'(x_{k-1});
5  |  end
6  |  if x_N < 0 then
7  |  |  x_0 = x_0 + a, where a > 0;
8  |  else
9  |  |  break;
10 |  end
11 end
```

3.3 Performance analysis and simulation example

In this section, we evaluate the performance of Algorithm 2 in terms of simulation example. The time complexity graph of Algorithm 2 is shown in Figure 3, alongside reference curves ranging from $O(Nlog(N)$ to $O(N^{4.5})$. The problem dimension (x-axis) is the number of elements in the fast-time radar code vector $\mathbf{c}$, and the runtime is the time of one iteration of Algorithm 2 ran on the desktop computer (HP Z240 Tower Workstation with Xeon E3-1230v5 3.40GHz 8MB processor). By comparing the slope of Algorithm 2 runtime to slope of reference curves, we see that the time-complexity of Algorithm 2 is approximately quadratic (i.e., $O(N^2)$). This is expected since the most expensive operation in Algorithm 2 is matrix to vector product which essentially requires $O(N^2)$ operation.

Next, we use Algorithm 2 in simulation example. We consider frequency band occupied by several unlicensed and licensed radiators. For reference signal, we use linearly modulated signal.

3.3.1 Simulation set up

Let us consider radar system with transmit bandwidth of 6GHz. Radar uses sampling frequency $f_s = 12$GHz. Radar pulse has length $T = 2\mu s$ with duty cycle $d = 0.5$. This implies that the fast-time radar code has the length $1\mu s$, which corresponds to 12000 dimensional fast-time radar code vector by using sampling frequency f_s.

Covariance matrix $\mathbf{M}$ is modelled as:

$$\mathbf{M} = \sigma_0 \mathbf{I} + \sum_{k=1}^{K} \frac{\sigma_{I,k}}{\Delta f_k} \mathbf{R}_I^k + \sum_{k=1}^{K_J} \sigma_{J,k} \mathbf{R}_{J,k}, \tag{3.3.1}$$

where

- $\sigma_0 = 0$dB (thermal noise level);
- $K = 7$ (number of licensed radiators);
- $\sigma_{I,k} = 10\text{dB}, \forall k \in \{1, ..., K\}$ (energy of coexisting telecom network operating on normalized frequency band $\Omega_k = [f_1^k, f_2^k]$);
- $\Delta f_k = f_2^k - f_1^k, \forall k \in \{1, ..., K\}$ (bandwidth associated with the k'th licensed radiator);
- $K_J = 2$ (number of active and unlicensed narrowband jammers);
- $\sigma_{J,k} = \begin{cases} 50\text{dB}, & k = 1 \\ 40\text{dB}, & k = 2, \end{cases}$ (energy of active jammers);

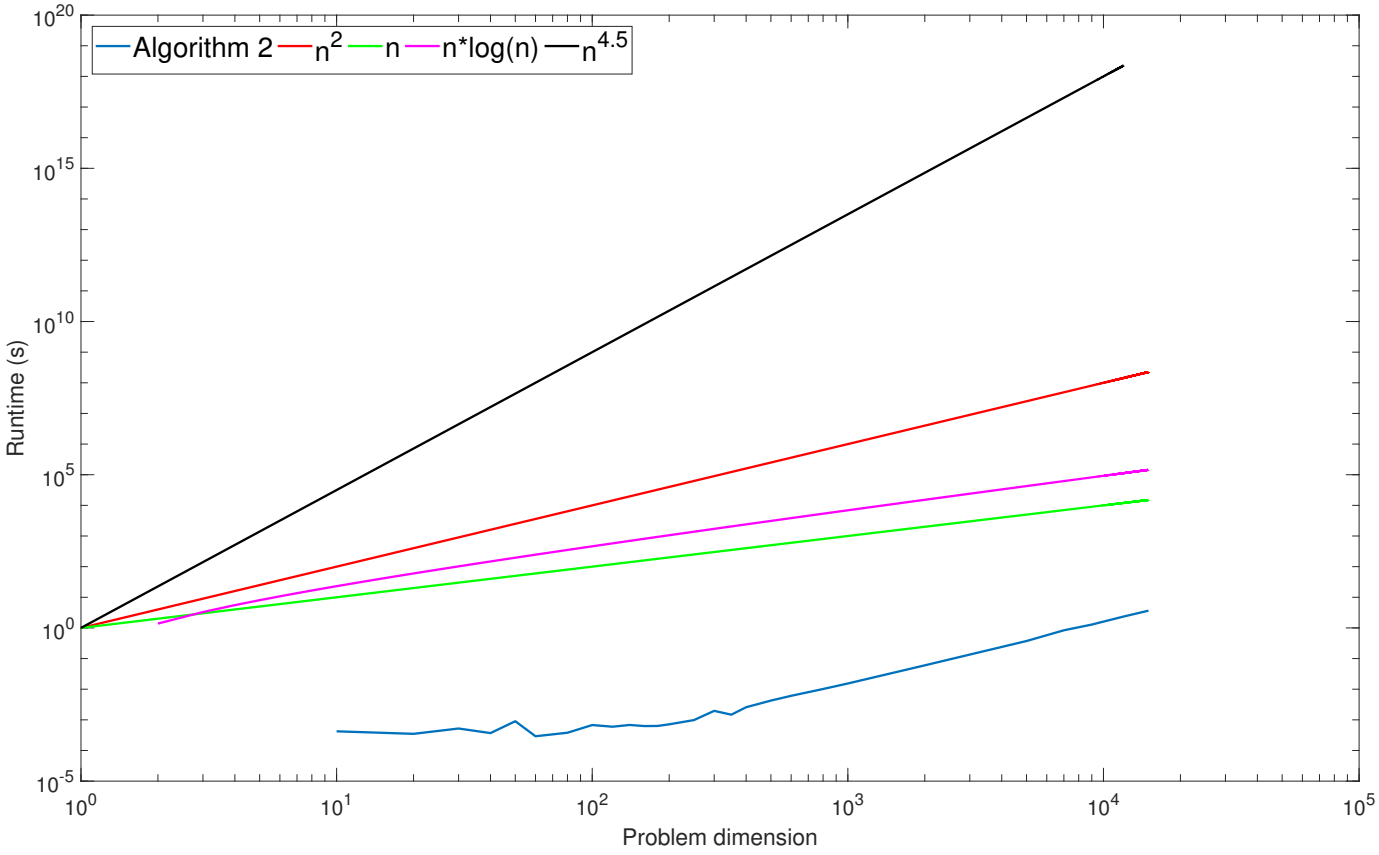

Figure 3: Time complexity graph of Algorithm 2

- $\mathbf{R}_{J,k} = \mathbf{r}_{J,k}\mathbf{r}_{J,k}^H, k = 1, ..., K_J$ (normalized disturbance covariance matrix of the k'th active unlicensed jammer);
- $\mathbf{r}_{J,k} = e^{j2\pi f_{j,k}n/f_s}$, $f_{J,1}/f_s = 0.7$ and $f_{J,2}/f_s = 0.75$;
- $w_k = 1, \forall k \in \{1, ..., 7\}$ (weights in $\mathbf{R}_I$).

The parameter values for noise covariance matrix $\mathbf{M}$ are exactly the same as the parameter values in [9]. For reference signal, we use linearly modulated signal $c_0 = e^{i2\pi(f_\Delta t + f_0)t}$ with carrier frequency $f_0 = 1.8$GHz and frequency range $f_\Delta = 3.6\text{GHz}/\mu s$. The licensed radiators operate at normalized frequency bands:

- $\Omega_1 = [f_1^1, f_2^1] = [0.0000, 0.0617]$
- $\Omega_2 = [f_1^2, f_2^2] = [0.0700, 0.1247]$
- $\Omega_3 = [f_1^3, f_2^3] = [0.1526, 0.2540]$
- $\Omega_4 = [f_1^4, f_2^4] = [0.3086, 0.3827]$
- $\Omega_5 = [f_1^5, f_2^5] = [0.4074, 0.4938]$
- $\Omega_6 = [f_1^6, f_2^6] = [0.6185, 0.7600]$
- $\Omega_7 = [f_1^7, f_2^7] = [0.8200, 0.9500]$

We set the similarity and radiation constraint levels to $\epsilon = 0.9$ and $E_I = 0.87$. In Figure 4, convergence graph of Algorithm 2 is shown alongside constraint levels per each iteration. In Figure 5, frequency spectrum of designed signal (i.e. signal maximizing SINR) is shown. In Figure 6, SINR of the designed signal per iteration is shown, and in Figure 7, the ambiguity function of designed signal is shown.

From Figure 5, we see that designed signal uses all available bands except the bands in the range 0.0617-0700 and 0.3827-0.4074. The reference signal c_0 is in the range 0.3-0.9, and the level of ϵ determines how freely designed signal can use allowed bands outside this region.

From Figure 6, we see that while minimizing surrogate objective, we are simultaneously maximizing original objective which is the SINR of the designed signal. Finally, from Figure 7, we see that our designed waveform has similar ambiguity function as the reference linearly modulated signals. Autocorrelation is very narrow and Doppler leakage is small.

Here the ambiguity transform $\mathcal{X}(\mathbf{u}, \mathbf{u}) : \mathbb{Z}_N \times \mathbb{Z}_N \to \mathbb{C}$ ($\mathbb{Z}_N$ is N-periodic discrete space) is defined as:

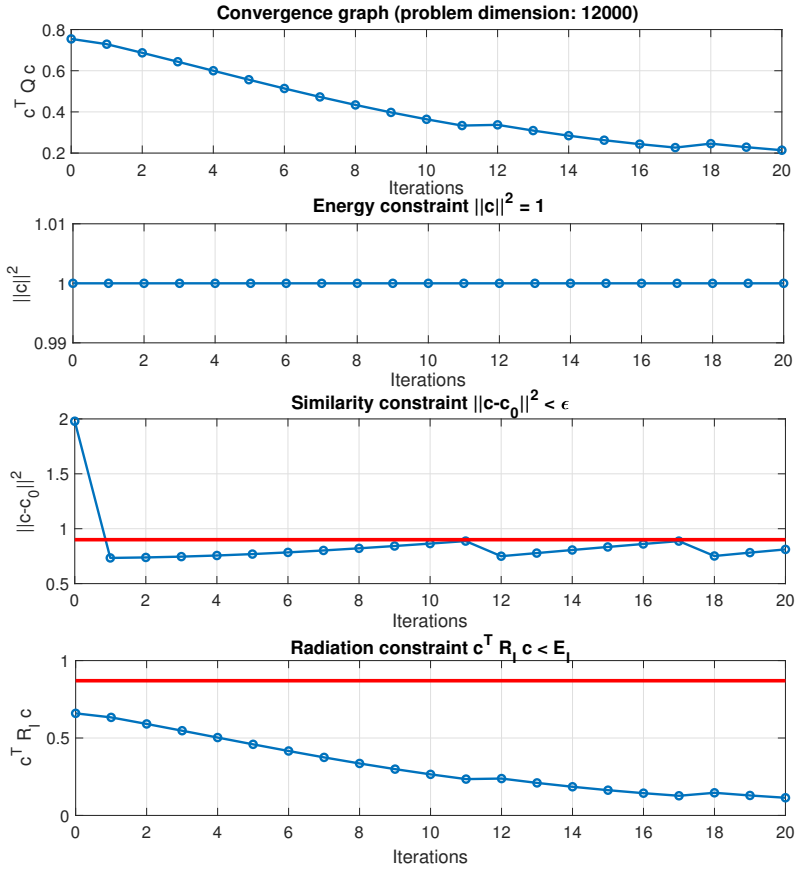

Figure 4: Convergence graph of Algorithm 2

$$\mathcal{X}(\mathbf{u},\mathbf{u})(\xi,y)=\frac{1}{N}\sum_{k=0}^{N-1}e^{-j2\pi k\xi/N}\mathbf{u}(k+y/2)\mathbf{u}(k-y/2)^*, \tag{3.3.2}$$

where ξ denotes frequency shift, y is time shift, and $(\bullet)^*$ is complex conjugate operation. Autocorrelation has the form:

$$\mathcal{X}(\mathbf{u},\mathbf{u})(0,y)=\frac{1}{N}\sum_{k=0}^{N-1}\mathbf{u}(k+y/2)\mathbf{u}(k-y/2)^*, \tag{3.3.3}$$

i.e., it is a 0-Doppler cut of (3.3.2).

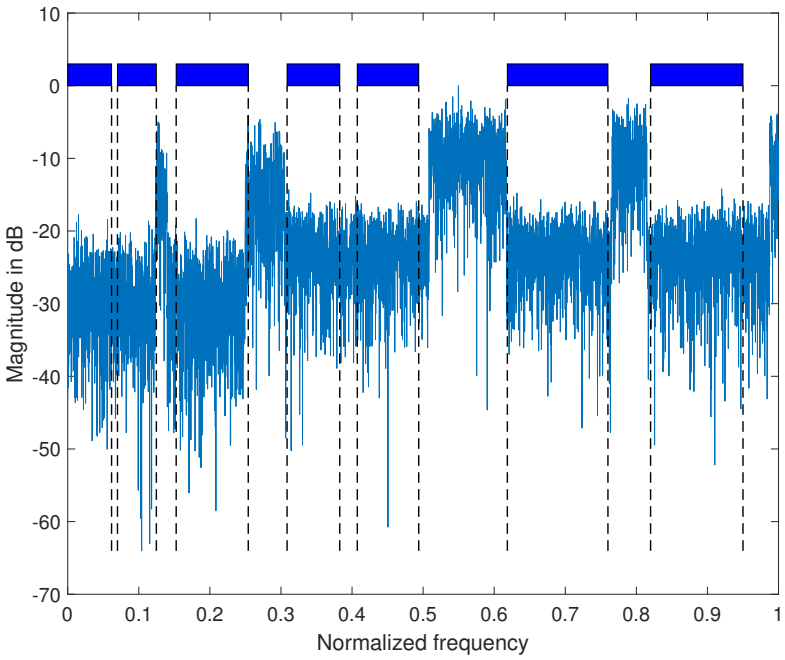

Figure 5: Frequency spectrum of designed signal

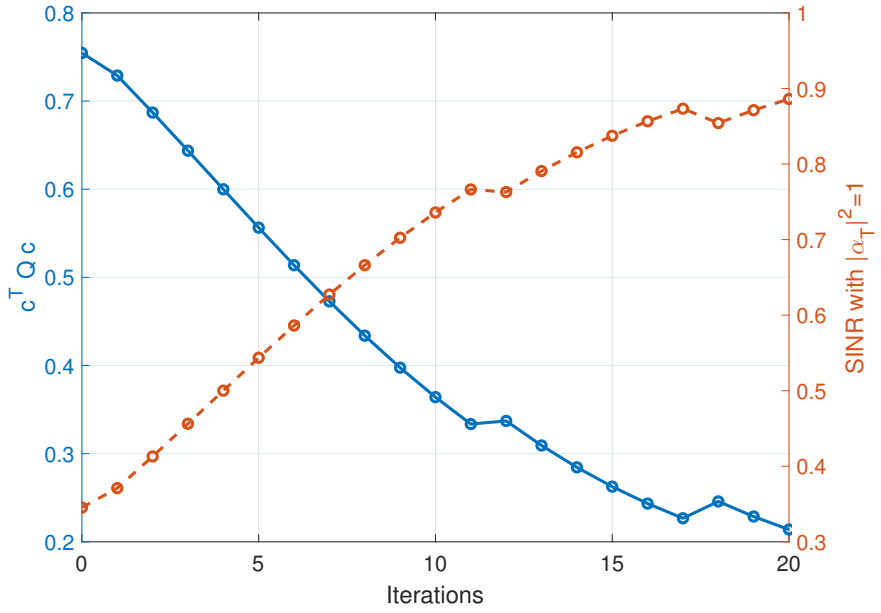

Figure 6: SINR of designed signal per iteration

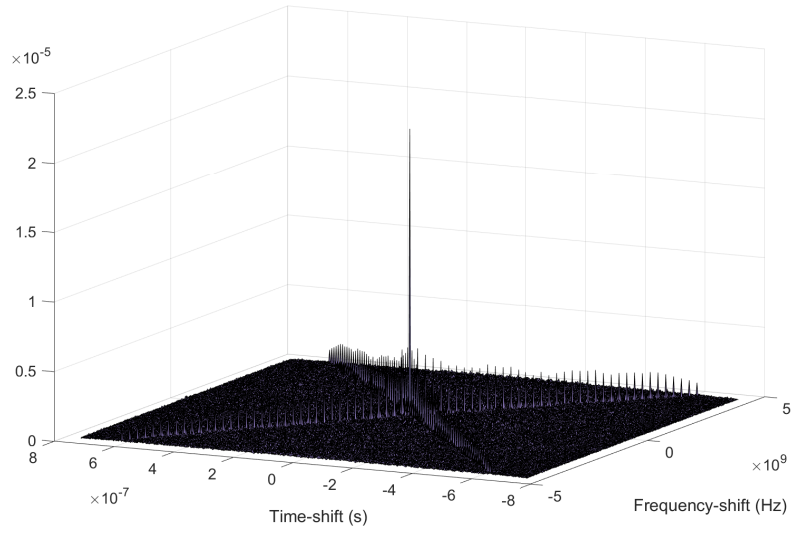

Figure 7: Ambiguity function of designed signal

3.4 Single waveform algorithm extension

In this subsection we consider radiation energy constraint (3.2.2c) separately for each constrained frequency band instead of summing them together as we do in Subsection 3.2. This is essentially the extension from paper [9] to [8]. Motivation for doing this is based on the discussions we had in IEEE SAM 2018 conference in Sheffield.

Therefore, the minimization problem $\mathcal{P}_1^{(4)}$ is written as:

$$\mathcal{P}_1^{(5)}: \begin{cases} \min\limits_{\mathbf{c}} & \mathbf{c}^T\mathbf{Q}\mathbf{c} & \text{(3.4.1a)} \\ \text{s.t. :} & \|\mathbf{c}\|^2 = 1 & \text{(3.4.1b)} \\ & \mathbf{c}^T\mathbf{R}_I^k\mathbf{c} \le E_I^k, \quad \forall k \in \{1,...,K\} & \text{(3.4.1c)} \\ & \|\mathbf{c}-\mathbf{c}_0\|^2 \le \epsilon, & \text{(3.4.1d)} \end{cases}$$

where $\mathbf{Q}, \mathbf{R}_I \in \mathbb{R}^{2N\times 2N}$, $\mathbf{c} \in \mathbb{R}^{2N}$ and $\mathbf{c}_0 \in \mathbb{R}^{2N}$.

To solve $\mathcal{P}_1^{(5)}$ we address constraints (3.4.1c) in $\mathbf{z}$-variable update. The $\mathbf{c}$-variable update remains the same as in Algorithm 2.

3.4.1 z-variable update

We have minimization problem of form:

$$\mathbf{z}_{k+1} = \arg\min_{\mathbf{z}} \left\{ \left\| \mathbf{z} - (\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}) \right\|_2^2 \right\} \quad |\text{s.t. } \mathbf{z}^T\mathbf{R}_I^k\mathbf{z} \le E_I^k, \quad \forall k \in \{1,...,K\}. \tag{3.4.2}$$

Lagrangian for (3.4.2) is given as:

$$L(\mathbf{z},\boldsymbol{\gamma}) = \left\| \mathbf{z} - (\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}) \right\|_2^2 + \sum_{k=1}^{K} \gamma^{(k)}(\mathbf{z}^T\mathbf{R}_I^k\mathbf{z} - E_I^k), \tag{3.4.3}$$

where $\boldsymbol{\gamma} = (\gamma^{(1)}, \gamma^{(2)}, ..., \gamma^{(K)})^T$, with $\gamma^{(k)} \in \mathbb{R}$, $k \in \{1,...,K\}$, are the Lagrange multipliers.

Karush-Kuhn-Tucker (KKT) conditions for the minimization problem (3.4.2) are given as:

$$\begin{cases} \nabla_{\mathbf{z}} L(\mathbf{z}^*, \gamma^*) = 0 & \text{(3.4.4a)} \\ \gamma^* \ge 0 & \text{(3.4.4b)} \\ (\gamma^{(k)})^*((\mathbf{z}^*)^T\mathbf{R}_I^k\mathbf{z}^* - E_I^k) = 0, \quad \forall k \in \{1,...,K\} & \text{(3.4.4c)} \\ (\mathbf{z}^T\mathbf{R}_I\mathbf{z} - E_I) \le 0 & \text{(3.4.4d)} \\ \nabla_{\mathbf{zz}} L(\mathbf{z}^*, \gamma^*) \succeq 0, & \text{(3.4.4e)} \end{cases}$$

where $\mathbf{z}^*$ and $\boldsymbol{\gamma}^*$ denote critical points of the Lagrangian $L(\mathbf{z}, \boldsymbol{\gamma})$. To solve γ's we use complementary slackness conditions (3.4.4c) and $\mathbf{z}$-variable update is obtained by gradient condition (3.4.4a). Gradient $\nabla_{\mathbf{z}} L(\mathbf{z}^*, \gamma^*)$:

$$\begin{aligned}\nabla_{\mathbf{z}} L(\mathbf{z}^*, \gamma^*) &= 2(\mathbf{z} - (\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda})) + \sum_{k=1}^{K} \gamma^{(k)} \left((\mathbf{R}_I^k)^T + \mathbf{R}_I^k \right) \mathbf{z} \\ &= 2(\mathbf{z} - (\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda})) + \left(\sum_{k=1}^{K} 2\gamma^{(k)} \mathbf{R}_I^k \right) \mathbf{z} \\ &= \left(2\mathbf{I} + \sum_{k=1}^{K} 2\gamma^{(k)} \mathbf{R}_I^k \right) \mathbf{z} - 2 \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right) \end{aligned} \tag{3.4.5}$$

Therefore condition (3.4.4a) yields:

$$\mathbf{z} = \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k \right)^{-1} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right) \tag{3.4.6}$$

Substitution of (3.4.6) to complementary slackness conditions (3.4.4c) yields:

$$\begin{aligned} &\mathbf{z}_{k+1}^T \mathbf{R}_I^k \mathbf{z}_{k+1} - E_I^k = 0 \\ &\overset{*}{\Rightarrow} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right)^T \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k \right)^{-1} \mathbf{R}_I^k \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k \right)^{-1} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right) = E_I^k. \end{aligned} \tag{3.4.7}$$

(*) $\left(\left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k \right)^{-1} \right)^T = \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k \right)^{-1}$ (see appendix B).

By denoting:

$$f_i(\boldsymbol{\gamma}) = \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right)^T \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k \right)^{-1} \mathbf{R}_I^i \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k \right)^{-1} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda} \right) - E_I^i, \tag{3.4.8}$$

we have system of K equations with K unknowns:

$$\begin{cases} f_1(\boldsymbol{\gamma}) = 0, & \text{(3.4.9a)} \\ f_2(\boldsymbol{\gamma}) = 0, & \text{(3.4.9b)} \\ \vdots & \\ f_K(\boldsymbol{\gamma}) = 0. & \text{(3.4.9c)} \end{cases}$$

To solve system of equations (3.4.9a)-(3.4.9c) we use Newton's method:

$$\boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k - \left[\mathbf{J}\left(\boldsymbol{\gamma}_k\right)\right]^{-1} \mathbf{F}\left(\boldsymbol{\gamma}_k\right), \tag{3.4.10}$$

where $\mathbf{J}\left(\boldsymbol{\gamma}_k\right) = \left(\begin{bmatrix} \frac{\partial f_1}{\partial \gamma^{(1)}}\left(\boldsymbol{\gamma}_k\right) & \frac{\partial f_1}{\partial \gamma^{(2)}}\left(\boldsymbol{\gamma}_k\right) & \cdots & \frac{\partial f_1}{\partial \gamma^{(K)}}\left(\boldsymbol{\gamma}_k\right) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_K}{\partial \gamma^{(1)}}\left(\boldsymbol{\gamma}_k\right) & \frac{\partial f_K}{\partial \gamma^{(2)}}\left(\boldsymbol{\gamma}_k\right) & \cdots & \frac{\partial f_K}{\partial \gamma^{(K)}}\left(\boldsymbol{\gamma}_k\right) \end{bmatrix}\right)$ (Jacobian) and $\mathbf{F}\left(\boldsymbol{\gamma}_k\right) = \left(f_1(\boldsymbol{\gamma}_k), f_2(\boldsymbol{\gamma}_k), ..., f_K(\boldsymbol{\gamma}_k)\right)^T$. Jacobian entries are given as:

$$\begin{aligned}
\frac{\partial f_i(\boldsymbol{\gamma})}{\partial \gamma^{(j)}} &= \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T \frac{\partial}{\partial \gamma^{(j)}} \left[\left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1}\right] \mathbf{R}_I^i \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) + \\
&\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \mathbf{R}_I^i \frac{\partial}{\partial \gamma^{(j)}} \left[\left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1}\right] \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) \\
&\overset{*}{=} -\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \left[\frac{\partial}{\partial \gamma^{(j)}} \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)\right] \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} * \\
&\mathbf{R}_I^i \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \mathbf{R}_I^i * \\
&\left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \left[\frac{\partial}{\partial \gamma^{(j)}} \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)\right] \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) \\
&= -\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \mathbf{R}_I^j \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \mathbf{R}_I^i \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) - \\
&\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \mathbf{R}_I^i \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \mathbf{R}_I^j \left(\mathbf{I} + \sum_{k=1}^{K} \gamma^{(k)} \mathbf{R}_I^k\right)^{-1} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)
\end{aligned} \tag{3.4.11}$$

(*) Derivative of inverse matrix $(K^{-1})' = -K^{-1}K'K^{-1}$.

4 Multiple waveforms design in spectrally busy environment

Next, we extend our single waveform design algorithm to multiple waveforms design, i.e., we simultaneously design P signals. To do this, we need to add constraint for integrated side-lobe level (ISL). ISL controls auto- and cross-correlation between designed signals. This is required to make sure that each signals are not deteriorating the detection performance of distinctive transmitter signals.

For P radar codes (or sequences in general) of length N, $\mathbf{c}^{(p)} = \{c_j^p\}_{j=0}^{N-1}$, we define ISL as (see e.g. [15]):

$$ISL(\mathbf{c}^{(p)}, \mathbf{c}^{(q)}) = \sum_{p=1}^{P} \sum_{k=-N+1, k\neq 0}^{N-1} |X_{\mathbf{c}^{(p)}\mathbf{c}^{(p)}}(k)|^2 + \sum_{p=1}^{P} \sum_{q=1, p\neq q}^{P} \sum_{k=-N+1}^{N-1} |X_{\mathbf{c}^{(\mathbf{p})}\mathbf{c}^{(\mathbf{q})}}(k)|^2, \tag{4.0.1}$$

where the cross-correlation $X_{\mathbf{c}^{(p)}\mathbf{c}^{(q)}}(k)$ is defined as:

$$X_{\mathbf{c}^{(\mathbf{p})}\mathbf{c}^{(\mathbf{q})}}(k) = \sum_{j=\max\{0,-k\}}^{\min\{N-k,N\}-1} c_j^{(p)} (c_{j+k}^{(q)})^*, \qquad k = \{-N+1, ..., N-1\}. \tag{4.0.2}$$

In (4.0.2), c^* denotes complex conjugate. We aim to obtain sequences $\mathbf{c}^{(p)}$ such that the autocorrelation for any non-zero time shifts, i.e., $X_{\mathbf{c}^{(p)}\mathbf{c}^{(p)}}(k), \forall k \neq 0$, is low. This means sequences that are uncorrelated with time-shifted versions of themselves. Also, the cross-correlations between each pair of sequences, i.e., $X_{\mathbf{c}^{(p)}\mathbf{c}^{(q)}}(k), \forall k\{-N+1, ..., N-1\}$, need to be low. This means that each sequence must be uncorrelated with other sequences. These requirements are achieved by requiring ISL to be low, more precisely by ISL to be upper-bounded by some small constant.

Next, we will write the similarity constraint (3.1.22) in more general form (as discussed in [8]):

$$\|\mathbf{c} - \alpha_{\mathbf{c}_0}\mathbf{c}_0\|^2 \leq \epsilon, \tag{4.0.3}$$

where $|\alpha_{\mathbf{c}_0}|^2 \leq 1$, $\|\mathbf{c}\|^2 \leq 1$ and $\|\mathbf{c}_0\|^2 = 1$. It is worth pointing out that if $\|\mathbf{c}\|^2 = 1$ (i.e., radar code is unit energy signal), we have:

$$
\begin{aligned}
f(\alpha_{\mathbf{c}_0}) &= \|\mathbf{c} - \alpha_{\mathbf{c}_0}\mathbf{c}_0\|^2 \\
&= \|\mathbf{c}\|^2 + \alpha_{\mathbf{c}_0}^2\|\mathbf{c}_0\|^2 - 2\alpha_{\mathbf{c}_0}\mathbf{c}^T\mathbf{c}_0 \\
&= 1 + \alpha_{\mathbf{c}_0}^2 - 2\alpha_{\mathbf{c}_0}\mathbf{c}^H\mathbf{c}_0.
\end{aligned}
$$

To minimize $f(\alpha_{\mathbf{c}_0})$, we find:

$$
\begin{aligned}
f'(\alpha_{\mathbf{c}_0}) = 2\alpha_{\mathbf{c}_0} - 2\mathbf{c}^H\mathbf{c}_0 = 0 \\
\Rightarrow \alpha_{\mathbf{c}_0} = \mathbf{c}^H\mathbf{c}_0,
\end{aligned}
$$

which leads to similarity $\|\mathbf{c} - \alpha_{\mathbf{c}_0}\mathbf{c}_0\|^2 = 1 - |\mathbf{c}^H\mathbf{c}_0|^2$. By comparing it to similarity constraint (3.1.22) $\|\mathbf{c} - \mathbf{c}_0\|^2 = 2(1 - \mathbf{c}^H\mathbf{c}_0)$, we see that both expressions are minimized when $\mathbf{c} = \mathbf{c}_0$ and they are monotonically increasing around this point. Hence, these constraints can be said to be equivalent.

For multiple waveform design, let us use the sum of SINR values $\sum_{p=1}^{P}(\mathbf{c}^{(p)})^H\mathbf{M}^{-1}\mathbf{c}^{(p)}$ as the objective. Now, we can write the optimization problem for multiple waveform design as $\mathcal{P}_2$:

$$
\mathcal{P}_2 : \begin{cases}
\max\limits_{\mathbf{c}} & \sum\limits_{p=1}^{P}(\mathbf{c}^{(p)})^H\mathbf{M}^{-1}\mathbf{c}^{(p)} & \text{(4.0.4a)} \\
\text{s.t. :} & \|\mathbf{c}^{(p)}\|^2 \le 1, \quad \forall p \in \{1, ..., P\} & \text{(4.0.4b)} \\
& (\mathbf{c}^{(p)})^H\mathbf{R}_I\mathbf{c}^{(p)} \le E_I, \quad \forall p \in \{1, ..., P\} & \text{(4.0.4c)} \\
& \left\|\mathbf{c}^{(p)} - \alpha_{\mathbf{c}_0}\mathbf{c}_0^{(p)}\right\|^2 \le \epsilon, \quad \forall p \in \{1, ..., P\} & \text{(4.0.4d)} \\
& |\alpha_{\mathbf{c}_0}|^2 \le 1 & \text{(4.0.4e)} \\
& ISL(\mathbf{c}^{(p)}, \mathbf{c}^{(q)}) \le \delta, \quad \forall p, q \in \{1, ..., P\}, p \ne q, & \text{(4.0.4f)}
\end{cases}
$$

where $\mathbf{c}^{(p)}, \mathbf{c}_0^{(p)} \in \mathbb{C}^N$, $p \in 1, ..., P$, $\mathbf{M}, \mathbf{R}_I \in \mathbb{C}^{N\times N}$, $\alpha_{\mathbf{c}_0} \in \mathbb{C}$, and $E_I, \epsilon, \delta \in \mathbb{R}_+$. It is worth pointing out that objective (4.0.4a) is separable which makes it natural to be approached by splitting methods such as ADMM.

In addition to constraints (4.0.4b-4.0.4f), we can consider PAPR (peak-to-average power ratio) constraint. PAPR constraint can be written as:

$$
\frac{|\mathbf{c}_{peak}^{(p)}|}{\sqrt{\frac{1}{N}\|\mathbf{c}^{(p)}\|^2}} \le \beta, \tag{4.0.5}
$$

where $\beta > 0$. However in this section, we will ignore the PAPR-constraint and focus on solving problem $\mathcal{P}_2$.

4.1 Simplified version of $\mathcal{P}_2$

We first write problem $\mathcal{P}_2$ as the following minimization problem:

$$\mathcal{P}_2^{(1)}: \begin{cases} \min\limits_{\mathbf{c}} & -\sum\limits_{p=1}^{P}(\mathbf{c}^{(p)})^H\mathbf{M}^{-1}\mathbf{c}^{(p)} & (4.1.1a)\\ \text{s.t. :} & \|\mathbf{c}^{(p)}\|^2 \le 1, \quad \forall p \in \{1,...,P\} & (4.1.1b)\\ & (\mathbf{c}^{(p)})^H\mathbf{R}_I\mathbf{c}^{(p)} \le E_I, \quad \forall p \in \{1,...,P\} & (4.1.1c)\\ & \left\|\mathbf{c}^{(p)} - \alpha_{\mathbf{c}_0}\mathbf{c}_0^{(p)}\right\|^2 \le \epsilon, \quad \forall p \in \{1,...,P\} & (4.1.1d)\\ & |\alpha_{\mathbf{c}_0}|^2 \le 1 & (4.1.1e)\\ & ISL(\mathbf{c}^{(p)},\mathbf{c}^{(q)}) \le \delta, \quad \forall p,q \in \{1,...,P\}, p \ne q. & (4.1.1f) \end{cases}$$

Next step is to define side-lobe level as:

$$SL\left(\mathbf{c}^{(p)},\mathbf{c}^{(q)}\right) = \sum_{k=-N+1,k\ne 0}^{N-1}\left|X_{\mathbf{c}^{(p)}\mathbf{c}^{(p)}}(k)\right|^2 + \sum_{q=1,p\ne q}^{P}\sum_{k=-N+1}^{N-1}\left|X_{\mathbf{c}^{(p)}\mathbf{c}^{(q)}}(k)\right|^2. \tag{4.1.2}$$

Equation (4.1.2) can be equivalently (from optimization point of view) written as:

$$SL2\left(\mathbf{c}^{(p)}\right) = \sum_{k=1}^{N-1}|\underbrace{\sum_{j=0}^{N-1}\mathbf{c}^{(p)}(j)\left(\mathbf{c}^{(p)}(j+k)\right)^*}_{\text{Autocorrelation}}|^2 + \sum_{q=1,p\ne q}^{P}\sum_{k=0}^{N-1}|\underbrace{\sum_{j=0}^{N-1}\mathbf{c}^{(p)}(j)\left(\mathbf{c}^{(q)}(j+k)\right)^*}_{\text{Cross-correlation}}|^2. \tag{4.1.3}$$

In matrix notation, we can write the autocorrelation as:

$$\sum_{k=1}^{N-1}\left|\sum_{j=0}^{N-1}\mathbf{c}^{(p)}(j)\left(\mathbf{c}^{(p)}(j+k)\right)^*\right|^2 = \mathbf{1}^T\left|\begin{pmatrix}\left(\mathbf{c}^{(p)}(j+1)\right)^H\\ \left(\mathbf{c}^{(p)}(j+2)\right)^H\\ \vdots\\ \left(\mathbf{c}^{(p)}(j+N-2)\right)^H\\ \left(\mathbf{c}^{(p)}(j+N-1)\right)^H\end{pmatrix}\mathbf{c}^{(p)}(j)\right|^2 = \mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2,$$

and the cross-correlation as:

$$\sum_{q=1,p\neq q}^{P}\sum_{k=0}^{N-1}\left|\sum_{j=0}^{N-1}\mathbf{c}^{(p)}(j)\left(\mathbf{c}^{(q)}(j+k)\right)^{*}\right|^{2}=\sum_{q=1,p\neq q}^{P}\mathbf{1}^{T}\left|\begin{pmatrix}\left(\mathbf{c}^{(q)}(j)\right)^{H}\\ \left(\mathbf{c}^{(q)}(j+1)\right)^{H}\\ \vdots\\ \left(\mathbf{c}^{(q)}(j+N-2)\right)^{H}\\ \left(\mathbf{c}^{(q)}(j+N-1)\right)^{H}\end{pmatrix}\mathbf{c}^{(p)}(j)\right|^{2}$$

$$=\sum_{q=1,p\neq q}^{P}\mathbf{1}^{T}\left|\mathbf{B}_{q}\mathbf{c}^{(p)}\right|^{2}.$$

In above expressions, $|\mathbf{v}|^{2}=\begin{pmatrix}|(\mathbf{v})_{1}|^{2}\\ |(\mathbf{v})_{2}|^{2}\\ \vdots\\ |(\mathbf{v})_{N}|^{2}\end{pmatrix}$, where $(\mathbf{v})_{i}$ is the i'th element of the vector $\mathbf{v}$. Hence SL2 is written as:

$$\mathrm{SL2}\left(\mathbf{c}^{(p)}\right)=\mathbf{1}^{T}\left|\mathbf{A}\mathbf{c}^{(p)}\right|^{2}+\sum_{q=1,p\neq q}^{P}\mathbf{1}^{T}\left|\mathbf{B}_{q}\mathbf{c}^{(p)}\right|^{2}. \tag{4.1.4}$$

By using SL2, we can rewrite $\mathcal{P}_{2}^{(1)}$ as:

$$\mathcal{P}_{2}^{(2)}:\begin{cases}\min\limits_{\mathbf{c}} & -\sum\limits_{p=1}^{P}(\mathbf{c}^{(p)})^{H}\mathbf{M}^{-1}\mathbf{c}^{(p)} & \text{(4.1.5a)}\\ \text{s.t. :} & \|\mathbf{c}^{(p)}\|^{2}=1,\quad\forall p\in\{1,...,P\} & \text{(4.1.5b)}\\ & (\mathbf{c}^{(p)})^{H}\mathbf{R}_{I}\mathbf{c}^{(p)}\leq E_{I},\quad\forall p\in\{1,...,P\} & \text{(4.1.5c)}\\ & \left\|\mathbf{c}^{(p)}-\mathbf{c}_{0}^{(p)}\right\|^{2}\leq\epsilon,\quad\forall p\in\{1,...,P\} & \text{(4.1.5d)}\\ & SL2(\mathbf{c}^{(p)})\leq\delta,\quad\forall p\in\{1,...,P\}. & \text{(4.1.5e)}\end{cases}$$

Let us rewrite the objective $-\sum_{p=1}^{P}\left(\mathbf{c}^{(p)}\right)^{H}\mathbf{M}^{-1}\mathbf{c}^{(p)}$ as $\sum_{p=1}^{P}\left(\mathbf{c}^{(p)}\right)^{H}\mathbf{Q}\mathbf{c}^{(p)}$, where $\mathbf{Q}=\mu\mathbf{I}-\mathbf{R}$, $\mathbf{R}=\mathbf{M}^{-1}$ (Hermitian), and μ is positive constant such that $\mathbf{Q}\succeq\mathbf{0}$. Because $\mathbf{Q}\succeq\mathbf{0}$, the objective function $f(\mathbf{c})$ is convex. Also $\mathbf{Q}$ is symmetric. Let us write the complex valued matrices $\mathbf{Q},\mathbf{R}_{I},\mathbf{B}_{q}\in\mathbb{C}^{N\times N}$, $\mathbf{A}\in\mathbb{C}^{N-1\times N}$ and vectors $\mathbf{c}^{(p)},\mathbf{c}_{0}^{(p)}\in\mathbb{C}^{N}$ in real-valued notation as we did in the single waveform design case. We get the real-valued optimization problem:

$$\mathcal{P}_2^{(3)}: \begin{cases} \min\limits_{\mathbf{c}} & \sum\limits_{p=1}^{P} (\mathbf{c}^{(p)})^T \mathbf{Q} \mathbf{c}^{(p)} & (4.1.6a) \\ \text{s.t. :} & \|\mathbf{c}^{(p)}\|^2 = 1, \quad \forall p \in \{1, ..., P\} & (4.1.6b) \\ & (\mathbf{c}^{(p)})^T \mathbf{R}_I \mathbf{c}^{(p)} \leq E_I, \quad \forall p \in \{1, ..., P\} & (4.1.6c) \\ & \left\| \mathbf{c}^{(p)} - \mathbf{c}_0^{(p)} \right\|^2 \leq \epsilon, \quad \forall p \in \{1, ..., P\} & (4.1.6d) \\ & SL2(\mathbf{c}^{(p)}) \leq \delta, \quad \forall p \in \{1, ..., P\}. & (4.1.6e) \end{cases}$$

To address $\mathcal{P}_2^{(3)}$, we derive two different algorithms. The first one is straightforward extension of our single-waveform design algorithm developed in Section 3, and the second one is a new approach which aims to achieve lower computational complexity.

4.2 1st algorithm for multiple waveform design

For given $\mathbf{c}^{(k)}, \forall k \neq i$, $\mathcal{P}_2^{(3)}$ can be decomposed into the single waveform update:

$$\mathcal{P}_3^{(i)}: \begin{cases} \min\limits_{\mathbf{c}^{(i)}} & (\mathbf{c}^{(i)})^T \mathbf{Q} \mathbf{c}^{(i)} & (4.2.1a) \\ \text{s.t. :} & \|\mathbf{c}^{(i)}\|^2 = 1, & (4.2.1b) \\ & (\mathbf{c}^{(i)})^T \mathbf{R}_I \mathbf{c}^{(i)} \leq E_I, & (4.2.1c) \\ & \left\| \mathbf{c}^{(i)} - \mathbf{c}_0^{(i)} \right\|^2 \leq \epsilon, & (4.2.1d) \\ & SL2(\mathbf{c}^{(i)}) \leq \delta.. & (4.2.1e) \end{cases}$$

Notice that $\mathcal{P}_3^{(i)}$ is exactly the same problem as we had in the single waveform design case, except that we also have side-lobe level constraint (4.2.1e). Therefore, we can use our single-waveform design algorithm embedded with side-lobe constraint. We can do this by addressing the side-lobe constraint in $\mathbf{z}$-variable update, while $\mathbf{c}$-variable update remains the same.

4.2.1 z-variable update

We have the following minimization problem (derived in Subsection 3.2.2):

$$\mathbf{z}_{k+1} = \arg\min_{\mathbf{z}} \left\{ \left\| \mathbf{z} - (\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}) \right\|_2^2 \right\} \qquad |\text{s.t. } \mathbf{z}^T \mathbf{R}_I \mathbf{z} \leq E_I, SL2(\mathbf{c}^{(i)}) \leq \delta \tag{4.2.2}$$

The Lagrangian $L(\mathbf{z}, \gamma, \nu)$ for the problem is:

$$L(\mathbf{z},\gamma,\nu) = \left\|\mathbf{z} - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)\right\|_2^2 + \gamma\left(\mathbf{z}^T\mathbf{R}_I\mathbf{z} - E_I\right) + \nu(\mathrm{SL2}(\mathbf{z}) - \delta)$$
$$= \left\|\mathbf{z} - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)\right\|_2^2 + \gamma\left(\mathbf{z}^T\mathbf{R}_I\mathbf{z} - E_I\right) + \nu\left(\mathbf{1}^T|\mathbf{A}\mathbf{z}|^2 + \sum_{q=1,p\neq q}^{P}\mathbf{1}^T|\mathbf{B}_q\mathbf{z}|^2 - \delta\right).$$

The KKT conditions for the problem are then:

$$\begin{cases} \nabla_{\mathbf{z}} L(\mathbf{z}^*,\gamma^*,\nu^*) = 0 & \text{(4.2.3a)} \\ \gamma^* \geq 0 & \text{(4.2.3b)} \\ \nu^* \geq 0 & \text{(4.2.3c)} \\ \gamma^*((\mathbf{z}^*)^T\mathbf{R}_I\mathbf{z}^* - E_I) = 0 & \text{(4.2.3d)} \\ \nu^*(SL2(\mathbf{z}^*) - \delta) = 0. & \text{(4.2.3e)} \end{cases}$$

From (4.2.3a), we obtain:

$$\nabla_{\mathbf{z}} L\left(\mathbf{z}^*,\gamma^*,\nu^*\right) = 2\left(\mathbf{z}^* - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)\right) + \gamma^*\left(\mathbf{R}_I + \mathbf{R}_I^T\right)\mathbf{z}^* +$$
$$\nu^*\left(4\mathbf{A}^T\left(\mathbf{A}\mathbf{z}^*\right) + 2\sum_{q=1,p\neq q}^{P}\mathbf{B}_q^T\left(\mathbf{B}_q\mathbf{z}^*\right)\right) = 0$$
$$\Leftrightarrow \left(\mathbf{z}^* - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)\right) + \gamma^*\underbrace{\left(\frac{\mathbf{R}_I + \mathbf{R}_I^T}{2}\right)}_{=\mathbf{R}_I}\mathbf{z}^* +$$
$$\nu^*\left(2\mathbf{A}^T\left(\mathbf{A}\mathbf{z}^*\right) + \sum_{q=1,p\neq q}^{P}\mathbf{B}_q^T\left(\mathbf{B}_q z^*\right)\right) = 0$$
$$\Rightarrow \mathbf{z}^* - \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) + \gamma^*\mathbf{R}_I\mathbf{z}^* + 2\nu^*\mathbf{A}^T\mathbf{A}\mathbf{z}^* + \sum_{q=1,p\neq q}^{P}\nu^*\mathbf{B}_q^T\mathbf{B}_q\mathbf{z}^* = 0$$
$$\Rightarrow \mathbf{z}^* + \gamma^*\mathbf{R}_I\mathbf{z}^* + 2\nu^*\mathbf{A}^T\mathbf{A}\mathbf{z}^* + \sum_{q=1,p\neq q}^{P}\nu^*\mathbf{B}_q^T\mathbf{B}_q\mathbf{z}^* = \mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}$$
$$\Rightarrow \left(\mathbf{I} + \gamma^*\mathbf{R}_I + 2\nu^*\mathbf{A}^T\mathbf{A} + \sum_{q=1,p\neq q}^{P}\nu^*\mathbf{B}_q^T\mathbf{B}_q\right)\mathbf{z}^* = \mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}.$$

By using (4.2.3a), (4.2.3d) and (4.2.3e), we get system of three equations with three unknowns:

$$\begin{cases} \gamma\left(\mathbf{z}^T\mathbf{R}_I\mathbf{z} - E_I\right) = 0 & (4.2.4a) \\ \nu(\mathrm{SL2}(\mathbf{z}) - \delta) = 0 & (4.2.4b) \\ \left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum\limits_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)\mathbf{z} = \mathbf{c} + \dfrac{1}{\rho}\boldsymbol{\lambda}. & (4.2.4c) \end{cases}$$

Then we solve (4.2.4c) for $\gamma > 0$ and $\nu > 0$:

$$\mathbf{z} = \left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right). \quad (4.2.5)$$

Now the system reduced to two equations with two unknowns:

$$\begin{cases} \mathbf{z}^T\mathbf{R}_I\mathbf{z} = E_I & (4.2.6a) \\ SL2(\mathbf{z}) = \delta & (4.2.6b) \end{cases}$$

$\Leftrightarrow$

$$\begin{cases} \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T \left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum\limits_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1} \mathbf{R}_I * \\ \left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum\limits_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) = E_I, \\ \mathbf{1}^T\left|\mathbf{A}\left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum\limits_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2 + \\ \sum\limits_{q=1,p\neq q}^{P} \mathbf{1}^T\left|\mathbf{B}_q\left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum\limits_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2 = \delta. \end{cases}$$

This can be rewritten as:

$$\begin{cases} f_1(\gamma,\nu) = 0 & (4.2.8a) \\ f_2(\gamma,\nu) = 0, & (4.2.8b) \end{cases}$$

where $f_1(\gamma,\nu) = \left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)^T \left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1} \mathbf{R}_I *$
$\left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right) - E_I$, and
$f_2(\gamma,\nu) = \mathbf{1}^T\left|\mathbf{A}\left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c} + \frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2 +$

$$\sum_{q=1,p\neq q}^{P} \mathbf{1}^T \left| \mathbf{B}_q \left(\mathbf{I} + \gamma \mathbf{R}_I + 2\nu \mathbf{A}^T \mathbf{A} + \sum_{q=1,p\neq q}^{P} \nu \mathbf{B}_q^T \mathbf{B}_q \right)^{-1} \left(\mathbf{c} + \frac{1}{\rho} \boldsymbol{\lambda} \right) \right|^2 - \delta.$$

System (4.2.8a-4.2.8b) can be solved by Newton's method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[\mathbf{J} \left(\mathbf{x}_k \right) \right]^{-1} \mathbf{F} \left(\mathbf{x}_k \right), \tag{4.2.9}$$

where $\mathbf{x}_k = \left(\gamma_k, \nu_k \right)^T$, $\mathbf{J} \left(\mathbf{x}_k \right) = \begin{pmatrix} \frac{\partial f_1}{\partial \gamma} \left(\mathbf{x}_k \right) & \frac{\partial f_1}{\partial \nu} \left(\mathbf{x}_k \right) \\ \frac{\partial f_2}{\partial \gamma} \left(\mathbf{x}_k \right) & \frac{\partial f_2}{\partial \nu} \left(\mathbf{x}_k \right) \end{pmatrix}$ (Jacobian) and $\mathbf{F} \left(\mathbf{x}_k \right) = \left(f_1(\mathbf{x}_k), f_2(\mathbf{x}_k) \right)^T$.

Let us calculate Jacobian (notice that Jacobian does not depend on $\mathbf{x}$):

$$
\begin{aligned}
\frac{\partial f_1}{\partial \gamma} &= \frac{\partial}{\partial \gamma}[\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)^T \left(\mathbf{I}+\gamma\mathbf{R}_I+2\nu\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\mathbf{R}_{I*} \\
&\left(\mathbf{I}+\gamma\mathbf{R}_I+2\nu\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)-E_I] \\
&= \left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)^T\mathbf{R}_I^{-1}\mathbf{R}_I\mathbf{R}_I^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right) \\
&= \left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)^T\mathbf{R}_I^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right), \\
\frac{\partial f_1}{\partial \nu} &= \frac{\partial}{\partial \nu}[\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)^T \left(\mathbf{I}+\gamma\mathbf{R}_I+2\nu\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\mathbf{R}_{I*} \\
&\left(\mathbf{I}+\gamma\mathbf{R}_I+2\nu\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)-E_I] \\
&= \left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)^T\left(2\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\mathbf{R}_I\left(2\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right), \\
\frac{\partial f_2}{\partial \gamma} &= \frac{\partial}{\partial \gamma}[\mathbf{1}^T\left|\mathbf{A}\left(\mathbf{I}+\gamma\mathbf{R}_I+2\nu\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2+ \\
&\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\left(\mathbf{I}+\gamma\mathbf{R}_I+2\nu\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2-\delta] \\
&= \mathbf{1}^T\left|\mathbf{A}\mathbf{R}_I^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{R}_I^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2, \\
\frac{\partial f_2}{\partial \nu} &= \frac{\partial}{\partial \nu}[\mathbf{1}^T\left|\mathbf{A}\left(\mathbf{I}+\gamma\mathbf{R}_I+2\nu\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2+ \\
&\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\left(\mathbf{I}+\gamma\mathbf{R}_I+2\nu\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2-\delta] \\
&= \mathbf{1}^T\left|\mathbf{A}\left(2\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2+ \\
&\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\left(2\mathbf{A}^T\mathbf{A}+\sum_{q=1,p\neq q}^{P}\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}+\frac{1}{\rho}\boldsymbol{\lambda}\right)\right|^2.
\end{aligned}
$$

Now we can write the 1^{st} multiple waveform design algorithm as Algorithm 4:

Algorithm 4: 1^{st} multiple waveform design algorithm

1 function Multiplewaveform1($\mathbf{Q}, \{\mathbf{c}_0^{(p)}\}_{p\in\{1,...,P\}}, \mathbf{R}_I, E_I, \epsilon, \delta, K$);
Input : $\mathbf{Q} = \mu\mathbf{I} - \mathbf{R} \succeq \mathbf{0}$, $\{\mathbf{c}_0^{(p)}\}_{p\in\{1,...,P\}}$, $\mathbf{R}_I$, E_I, ϵ and K
Output : $\{\mathbf{c}^{(p)}\}_{p\in\{1,...,P\}}$
2 Initialize $\{\mathbf{c}^{(p)}\}_{p\in\{1,...,P\}}$, $\{\mathbf{z}^{(p)}\}_{p\in\{1,...,P\}}$ and $\{\boldsymbol{\lambda}^{(p)}\}_{p\in\{1,...,P\}}$;
3 **for** $k = 1, k \leq K, k$ ++ **do**
4 Solve $\mathcal{P}_3^{(i)}, \forall i \in \{1, ..., P\}$;
5 Update cross-correlation matrices $\mathbf{B}_q, \forall q \in \{1, ..., P\}$;
6 **end**

where $\mathcal{P}_3^{(i)}$ is solved as shown in Algorithm 5:

Algorithm 5: Solve $\mathcal{P}_3^{(i)}$

1 function Solve_$\mathcal{P}_3^{(i)}$($\mathbf{Q}, \mathbf{c}_0^{(i)}, \mathbf{R}_I, E_I, \epsilon, \delta, K$);
Input : $\mathbf{Q} = \mu\mathbf{I} - \mathbf{R} \succeq \mathbf{0}$, $\mathbf{c}_0^{(i)}$, $\mathbf{R}_I$, E_I, ϵ and K
Output : $\mathbf{c}^{(i)}$
2 Initialize $\mathbf{c}^{(i)}$, $\mathbf{z}^{(i)}$ and $\boldsymbol{\lambda}^{(i)}$;
3 **for** $k = 1, k \leq K, k$ ++ **do**
4 $\hat{\mathbf{c}}_{k+1}^{(i)} = \mathbf{c}_k^{(i)} - \frac{1}{L}\left(\left(\mathbf{Q} + \mathbf{Q}^T\right)\mathbf{c}_k^{(i)} + \left(\boldsymbol{\lambda}^{(i)} - \rho\mathbf{z}^{(i)}\right)\right)$;
5 $\widetilde{\mathbf{c}}_{k+1}^{(i)} = \frac{\hat{\mathbf{c}}_{k+1}^{(i)}}{\left\|\hat{\mathbf{c}}_{k+1}^{(i)}\right\|}$;
6 $\mathbf{c}_{k+1}^{(i)} = \text{RotateVector}(\widetilde{\mathbf{c}}_{k+1}^{(i)}, \mathbf{c}_0^{(i)}, \alpha, \epsilon)$;
7 Solve system (4.2.8a-4.2.8b);
8 $\mathbf{z}_{k+1}^{(i)} = \left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}^{(i)} + \frac{1}{\rho}\boldsymbol{\lambda}^{(i)}\right)$;
9 $\boldsymbol{\lambda}_{k+1}^{(i)} = \boldsymbol{\lambda}_k^{(i)} + \rho(\mathbf{c}_{k+1}^{(i)} - \mathbf{z}_{k+1}^{(i)})$;
10 **end**

The problem in line 7 in Algorithm 5 can be solved by Newton's method (4.2.9).

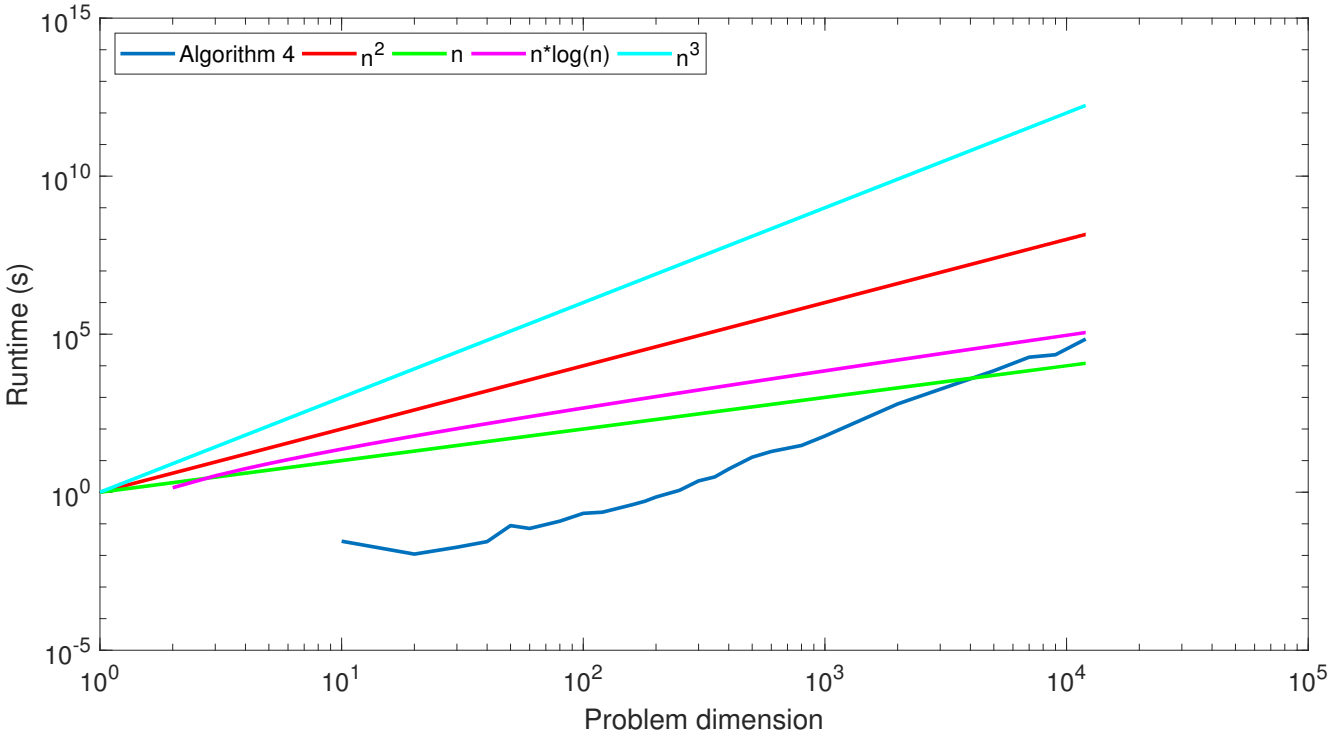

Figure 8: Time-complexity graph of Algorithm 4

4.2.2 Performance analysis

In this section, we evaluate the performance of Algorithm 4 by plotting its time complexity graph and other relevant figures of merit. In Figure 8, time complexity graph of Algorithm 4 is drawn alongside reference curves ranging from time-complexities $Nlog(N)$ to N^3. The blue line is the runtime of Algorithm 4 for one iteration when we design 3 different waveforms (i.e., $P = 3$) on desktop computer (HP Z240 Tower Workstation with Xeon E3-1230v5 3.40GHz 8MB processor). By comparing the slope of Algorithm 4 runtime curve to reference curves, we can see that time complexity is approximately cubic (i.e. $O(N^3)$). This is due to the fact that in line 7 of Algorithm 5 we use Newton's method which requires many matrix inversions. Also line 8 of Algorithm 5 requires matrix inversion. Matrix inversions are roughly obtained with $O(N^3)$ operations.

4.2.3 Simulation set up

Let us consider radar system with transmit bandwidth of 1.5 GHz and sampling frequency $f_s = 3$GHz. Radar pulse has length $T = 2\mu s$ with duty cycle $d = 0.5$. This implies that the fast-time radar code has length of $1\mu s$, which corresponds

to 3000 dimensional fast-time radar code vector by using sampling frequency f_s. We use covariance matrix $\mathbf{M}$ as defined in (3.3.1) with same parameter values as in section 3.3.1.

We aim to design three (i.e., $P = 3$) waveforms to frequency band, with reference signals being $\{\mathbf{c}_0^{(p)} = e^{i2\pi(f_\Delta^{(p)}t + f_0^{(p)})t}\}_{p\in\{1,2,3\}}$, where $f_\Delta^{(1)} = f_\Delta^{(2)} = f_\Delta^{(3)} = 0.2\text{GHz}/\mu s$ and $f_0^{(1)} = 250\text{MHz}$, $f_0^{(2)} = 700\text{MHz}$ and $f_0^{(3)} = 1.1\text{GHz}$.

The licensed radiators operate at normalized frequency bands:

- $\Omega_1 = [f_1^1, f_2^1] = [0.0000, 0.1500]$
- $\Omega_2 = [f_1^2, f_2^2] = [0.2300, 0.3000]$
- $\Omega_3 = [f_1^3, f_2^3] = [0.4200, 0.4800]$
- $\Omega_4 = [f_1^4, f_2^4] = [0.5000, 0.6000]$
- $\Omega_5 = [f_1^5, f_2^5] = [0.8300, 0.8800]$
- $\Omega_6 = [f_1^6, f_2^6] = [0.9200, 0.9600]$

For constraint levels we use values $E_I = 0.45, \epsilon = 0.6$, and $\delta = 7$.

In Figure 9, the convergence graph of the objective is shown alongside constraint levels per iteration. In Figure 10, the frequency spectrums of the designed waveforms $\{\mathbf{c}^{(p)}\}_{p\in\{1,2,3\}}$ are plotted. In Figure 11, the SINR values of the designed waveforms are shown per each iteration, and finally in Figures 12, 13 and 14, the ambiguity functions of the designed waveforms are plotted.

We see from Figure 10 that the first and second designed waveforms occupy mainly one allowed band, while the third designed waveform occupies two different allowed bands. The difference between allowed and constrained band radiation levels is about 20-30dB. From Figures 12, 13 and 14, we can see that the ambiguity functions of the designed waveforms are close to the ambiguity functions of the linearly modulated reference signals.

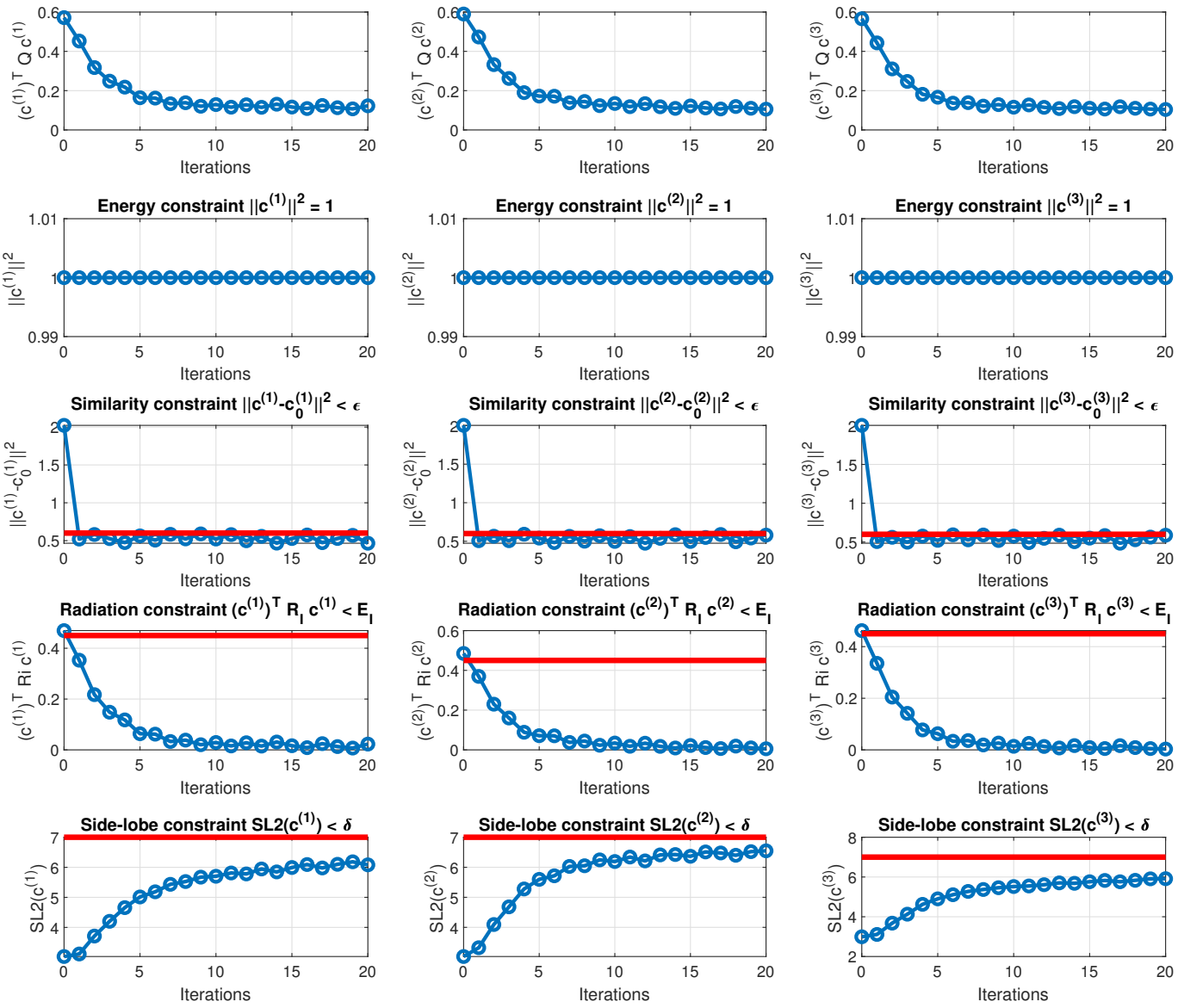

Figure 9: Convergence graph of Algorithm 4

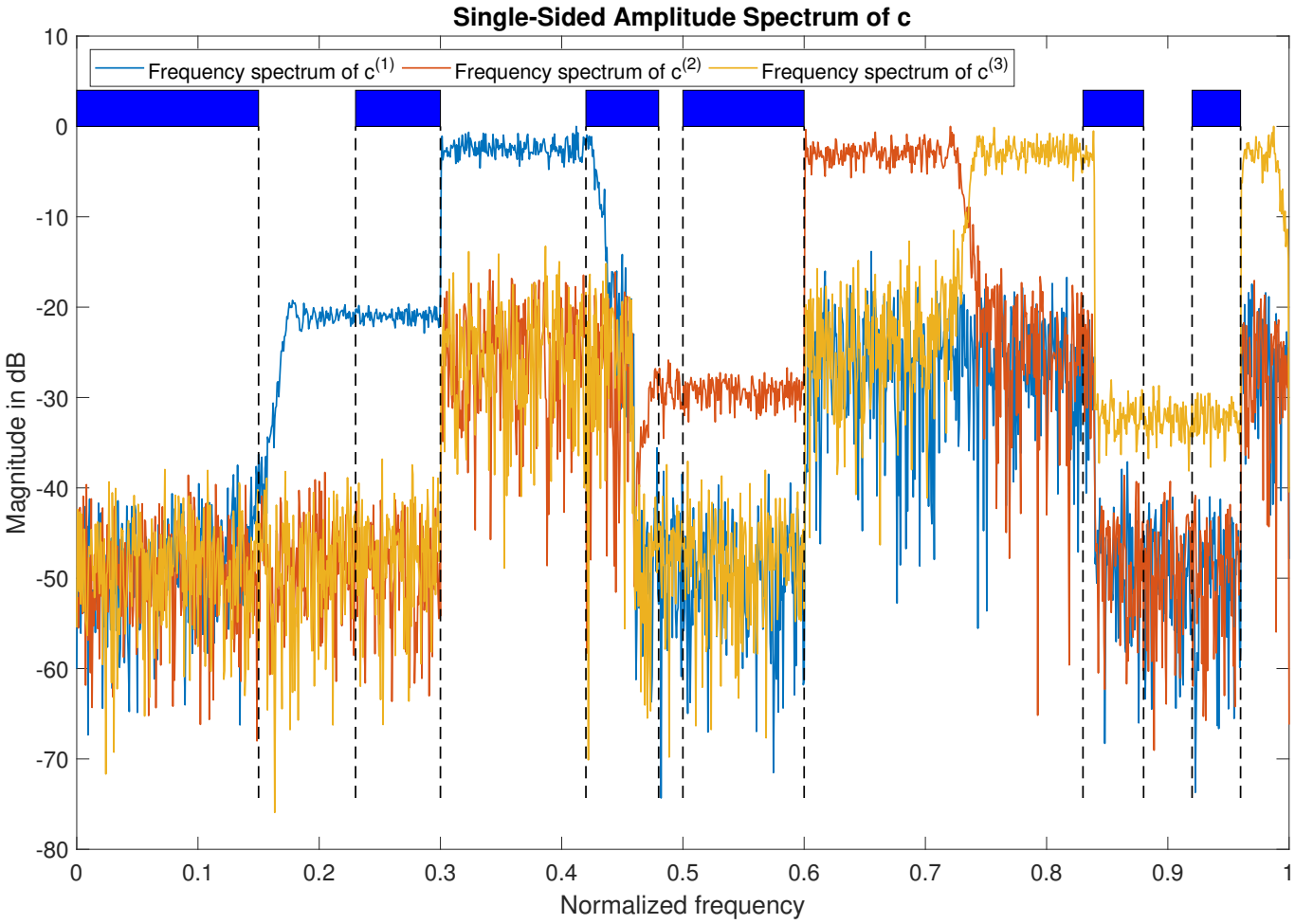

Figure 10: Frequency spectrum of designed signals $\{\mathbf{c}^{(p)}\}_{p\in\{1,2,3\}}$

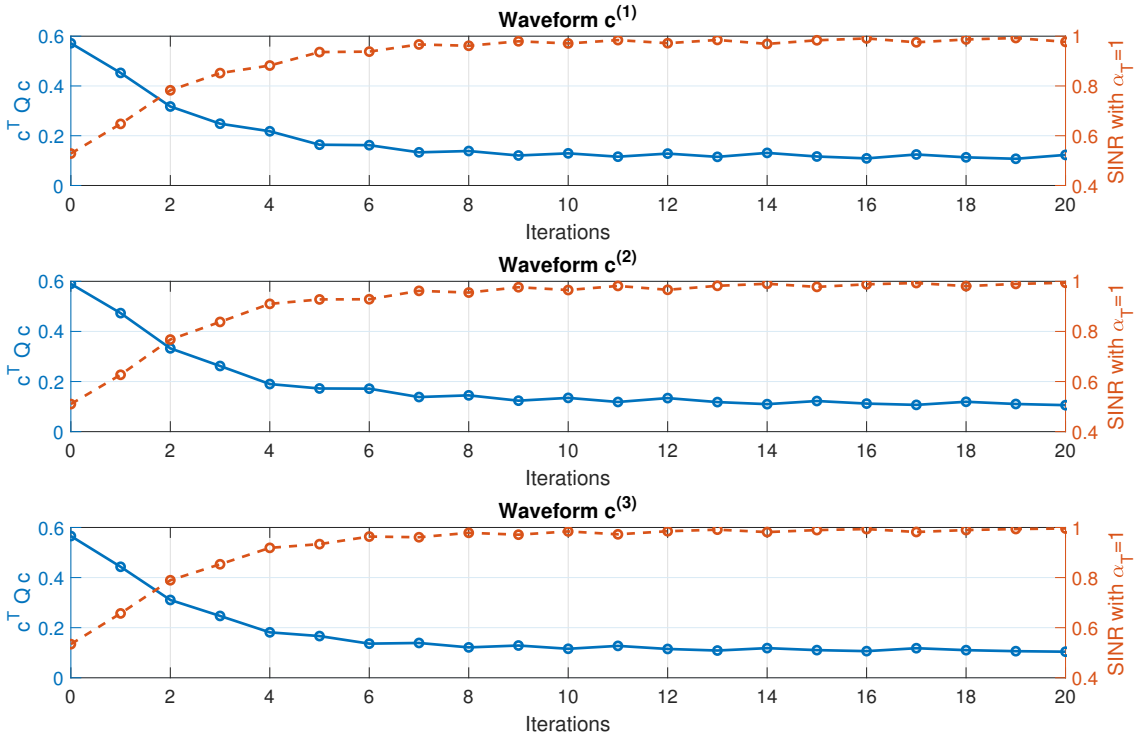

Figure 11: SINR values per iteration of designed signals $\{\mathbf{c}^{(p)}\}_{p\in\{1,2,3\}}$

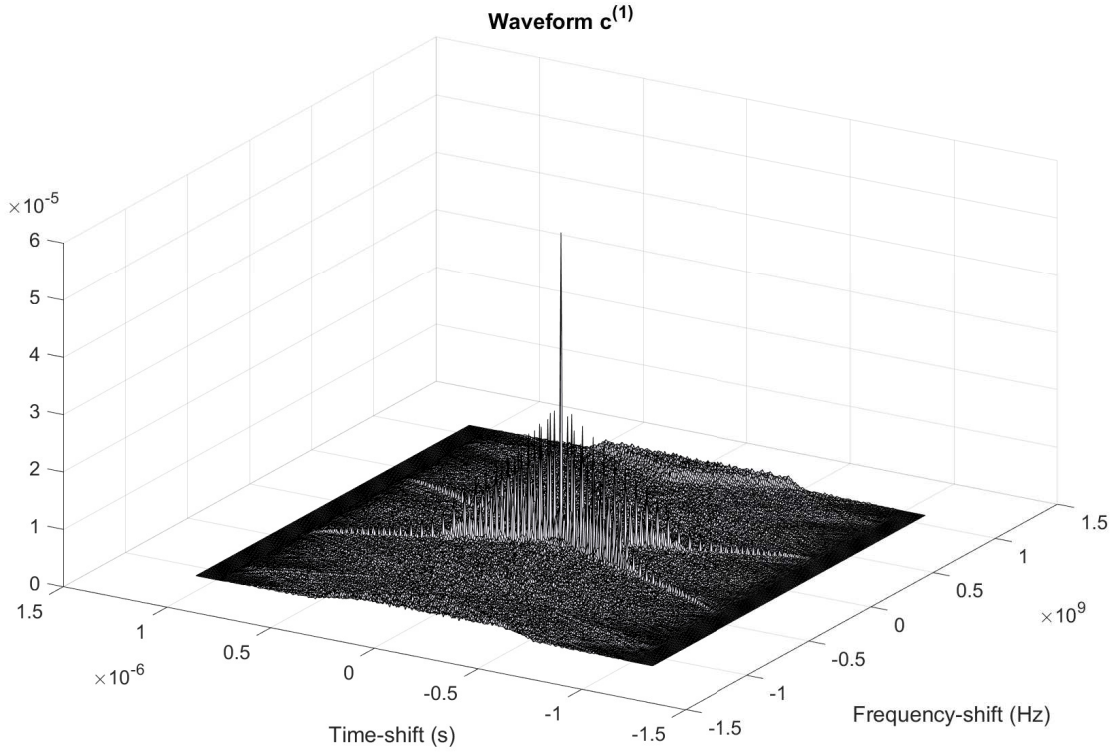

Figure 12: Ambiguity function of designed waveform $\mathbf{c}^{(1)}$

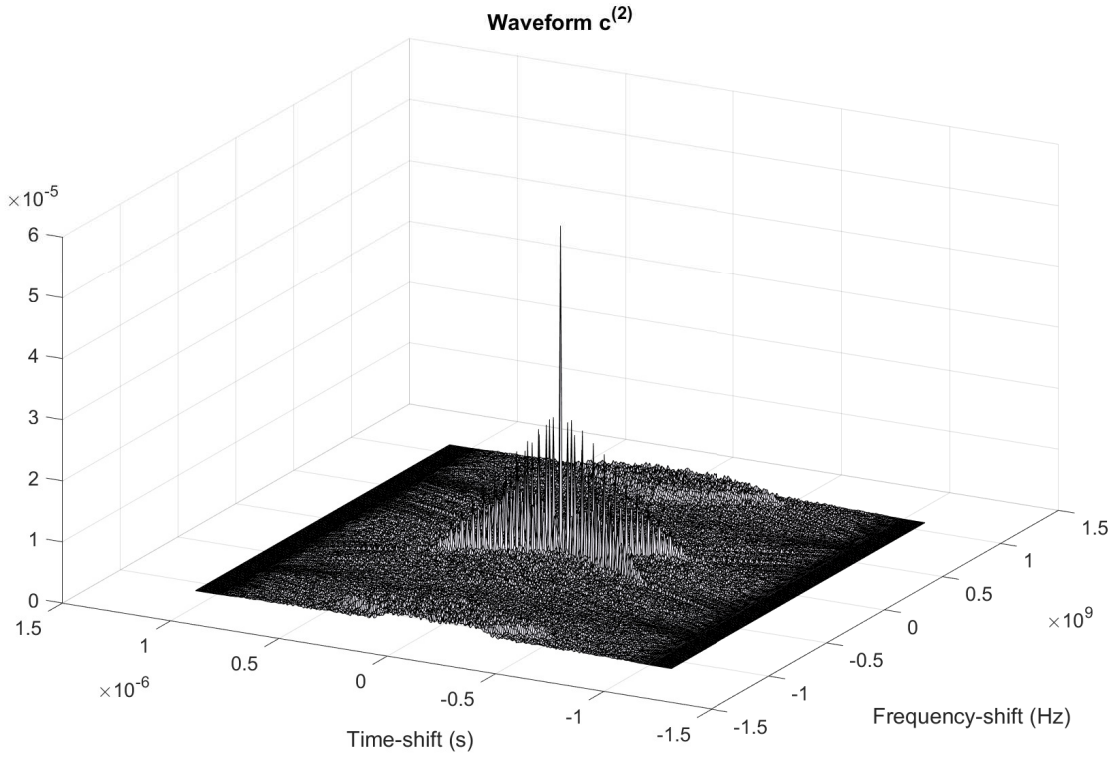

Figure 13: Ambiguity function of designed waveform $\mathbf{c}^{(2)}$

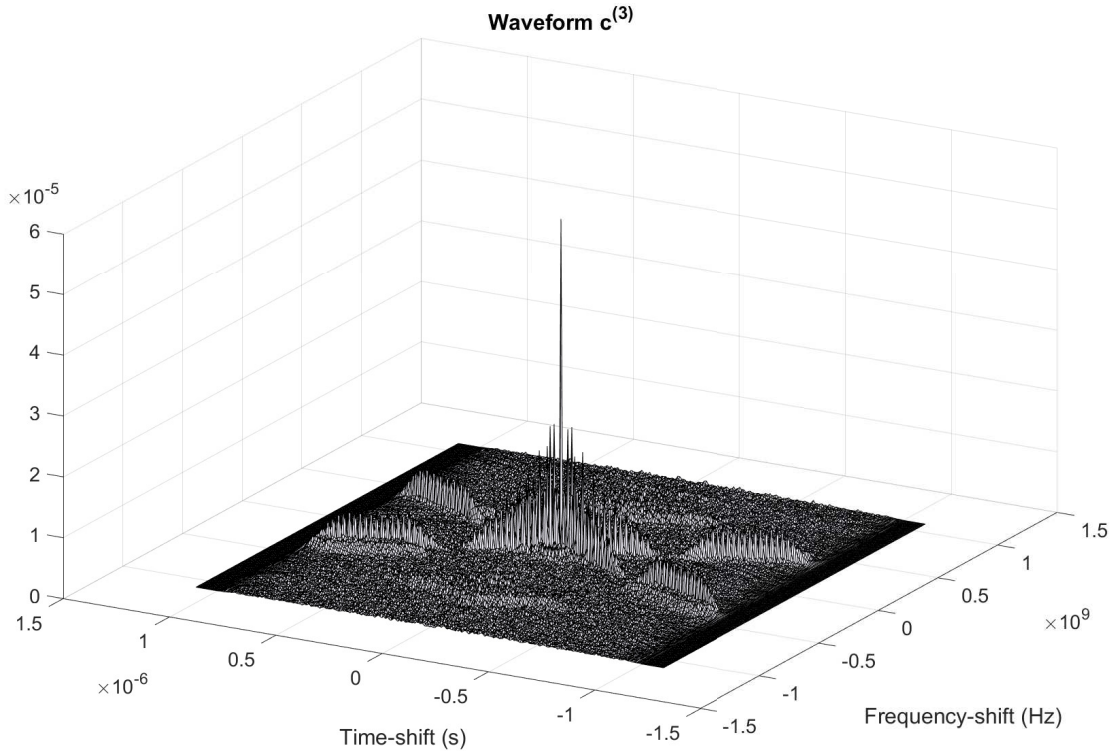

Figure 14: Ambiguity function of designed waveform $\mathbf{c}^{(3)}$

4.3 2nd algorithm for multiple waveform design

In this section we develop the second algorithm to solve problem $\mathcal{P}_2^{(3)}$ introduced in Subsection 4.1. The motivation for this development is that Algorithm 4 involves many unavoidable matrix inversions which cost $O(N^3)$ operations. Ideally, we would like to reduce the time-complexity of operations to $O(N^2)$ in order to ensure that the operations are suitable for large-scale problems.

We begin by writing the augmented Lagrangian to $\mathcal{P}_2^{(3)}$ using side-lobe level constraint $SL2(\mathbf{c}^{(p)}) \leq \delta$:

$$
\begin{aligned}
&L_\rho\left(\mathbf{c}^{(1)},\mathbf{c}^{(2)},...,\mathbf{c}^{(P)},\boldsymbol{\lambda}\right)\\
&=-\sum_{p=1}^{P}\left(\mathbf{c}^{(p)}\right)^T\mathbf{Q}\mathbf{c}^{(p)}+\sum_{p=1}^{P}\lambda^{(p)}\left(\mathrm{SL2}\left(\mathbf{c}^{(p)}\right)-\delta\right)+\underbrace{\frac{\rho}{2}\sum_{p=1}^{P}\left(\mathrm{SL2}\left(\mathbf{c}^{(p)}\right)-\delta\right)^2}_{\text{Augmentationpart}}\\
&=-\sum_{p=1}^{P}\left(\mathbf{c}^{(p)}\right)^T\mathbf{Q}\mathbf{c}^{(p)}+\lambda^{(p)}\mathrm{SL2}\left(\mathbf{c}^{(p)}\right)-\delta\lambda^{(p)}+\frac{\rho}{2}\mathrm{SL2}\left(\mathbf{c}^{(p)}\right)^2-\rho\delta\mathrm{SL2}\left(\mathbf{c}^{(p)}\right)+\frac{\rho}{2}\delta^2\\
&=-\sum_{p=1}^{P}\left(\mathbf{c}^{(p)}\right)^T\mathbf{Q}\mathbf{c}^{(p)}+\left(\lambda^{(p)}-\rho\delta\right)\mathrm{SL2}\left(\mathbf{c}^{(p)}\right)+\frac{\rho}{2}\mathrm{SL2}\left(\mathbf{c}^{(p)}\right)^2-\delta\lambda^{(p)}+\frac{\rho}{2}\delta^2\\
&=-\sum_{p=1}^{P}\left(\mathbf{c}^{(p)}\right)^T\mathbf{Q}\mathbf{c}^{(p)}+\left(\lambda^{(p)}-\rho\delta\right)\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2\right)+\\
&\frac{\rho}{2}\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2\right)^2-\delta\lambda^{(p)}+\frac{\rho}{2}\delta^2\\
&=-\sum_{p=1}^{P}\left(\mathbf{c}^{(p)}\right)^T\mathbf{Q}\mathbf{c}^{(p)}+\lambda^{(p)}\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2+\lambda^{(p)}\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2-\rho\delta\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2-\\
&\rho\delta\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2+\frac{\rho}{2}\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2\right)^2-\delta\lambda^{(p)}+\frac{\rho}{2}\delta^2\\
&=-\sum_{p=1}^{P}\left(\mathbf{c}^{(p)}\right)^T\mathbf{Q}\mathbf{c}^{(p)}+\lambda^{(p)}\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2+\lambda^{(p)}\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2-\rho\delta\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2-\\
&\rho\delta\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2+\frac{\rho}{2}\left(\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2\right)^2+2*\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2*\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2+\right.\\
&\left.\left(\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2\right)^2\right)-\delta\lambda^{(p)}+\frac{\rho}{2}\delta^2,
\end{aligned}
$$

where $\boldsymbol{\lambda}=\left[\lambda^{(1)},\lambda^{(2)},...,\lambda^{(P)}\right]^T$ is the vector of Lagrange multipliers.

Using the augmented Lagrangian $L_\rho\left(\mathbf{c}^{(1)},\mathbf{c}^{(2)},...,\mathbf{c}^{(P)},\boldsymbol{\lambda}\right)$, we can write ADMM steps:

$$
\begin{cases}
\mathbf{c}_{k+1}^{(1)} = \underset{\mathbf{c}^{(1)}}{\arg\min}\, L_\rho\left(\mathbf{c}^{(1)}, \mathbf{c}_k^{(2)}, ..., \mathbf{c}_k^{(P)}, \boldsymbol{\lambda}_k\right) & \text{(4.3.1a)} \\
\mathbf{c}_{k+1}^{(2)} = \underset{\mathbf{c}^{(2)}}{\arg\min}\, L_\rho\left(\mathbf{c}_{k+1}^{(1)}, \mathbf{c}^{(2)}, ..., \mathbf{c}_k^{(P)}, \boldsymbol{\lambda}_k\right) & \text{(4.3.1b)} \\
\vdots & \\
\mathbf{c}_{k+1}^{(P)} = \underset{\mathbf{c}^{(P)}}{\arg\min}\, L_\rho\left(\mathbf{c}_{k+1}^{(1)}, \mathbf{c}_{k+1}^{(2)}, ..., \mathbf{c}^{(P)}, \boldsymbol{\lambda}_k\right) & \text{(4.3.1c)} \\
\lambda_{k+1}^{(p)} = \lambda_k^{(p)} + \rho\left(\mathrm{SL2}\left(\mathbf{c}_{k+1}^{(p)}\right) - \delta\right) & \text{(4.3.1d)}
\end{cases}
$$

It is important to notice that the side-lobe level constraint (4.1.6e) is non-convex. This means that the convergence is not guaranteed when using ADMM. Also the fact that we have more than two primal variable updates in our ADMM steps (4.3.1) can cause problems with convergence. More details can be found from [10] Section 9.

In ADMM-steps (4.3.1a - 4.3.1c), we need to consider constraints $\|\mathbf{c}^{(p)}\|^2 = 1$, $\left(\mathbf{c}^{(p)}\right)^H \mathbf{R}_I \mathbf{c}^{(p)} \leq E_I$, and $\|\mathbf{c}^{(p)} - \mathbf{c}_0^{(p)}\|^2 \leq \epsilon, \forall p \in \{1, ..., P\}$. Hence minimization problems (4.3.1a - 4.3.1c) can be written as:

$$
\begin{cases}
\mathbf{c}_{k+1}^{(i)} = \underset{\mathbf{c}^{(i)}}{\arg\min}\, L_\rho\left(\mathbf{c}_{k+1}^{(1)}, ..., \mathbf{c}_{k+1}^{(i-1)}, \mathbf{c}^{(i)}, \mathbf{c}_k^{(i+1)}, ..., \mathbf{c}_k^{(P)}, \boldsymbol{\lambda}_k\right) & \text{(4.3.2a)} \\
\text{s.t. } \|\mathbf{c}^{(i)}\|^2 = 1 & \text{(4.3.2b)} \\
\left(\mathbf{c}^{(i)}\right)^T \mathbf{R}_I \mathbf{c}^{(i)} \leq E_I & \text{(4.3.2c)} \\
\|\mathbf{c}^{(i)} - \mathbf{c}_0^{(i)}\|^2 \leq \epsilon. & \text{(4.3.2d)}
\end{cases}
$$

Let us thus address optimization problem (4.3.2a-4.3.2d).

4.3.1 Single waveform update

Objective function (4.3.2a) can be simplified to:

$$
\begin{aligned}
&L_\rho\left(\mathbf{c}_{k+1}^{(1)},...,\mathbf{c}_{k+1}^{(i-1)},\mathbf{c}^{(i)},\mathbf{c}_{k}^{(i+1)},...,\mathbf{c}_{k}^{(P)},\boldsymbol{\lambda}_k\right)\\
&=-\sum_{p=1}^{P}\left(\mathbf{c}^{(p)}\right)^T\mathbf{Q}\mathbf{c}^{(p)}+\left(\lambda^{(p)}-\rho\delta\right)\mathrm{SL2}\left(\mathbf{c}^{(p)}\right)+\frac{\rho}{2}\mathrm{SL2}\left(\mathbf{c}^{(p)}\right)^2-\delta\lambda^{(p)}+\frac{\rho}{2}\delta^2\\
&=-\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}+\left(\lambda^{(i)}-\rho\delta\right)\mathrm{SL2}\left(\mathbf{c}^{(i)}\right)+\frac{\rho}{2}\mathrm{SL2}\left(\mathbf{c}^{(i)}\right)^2-\delta\lambda^{(i)}+\frac{\rho}{2}\delta^2\\
&\propto-\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}+\left(\lambda^{(i)}-\rho\delta\right)\mathrm{SL2}\left(\mathbf{c}^{(i)}\right)+\frac{\rho}{2}\mathrm{SL2}\left(\mathbf{c}^{(i)}\right)^2\\
&=-\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}+\left(\lambda^{(i)}-\rho\delta\right)\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(i)}\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(i)}\right|^2\right)+\\
&\frac{\rho}{2}\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(i)}\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(i)}\right|^2\right)^2.
\end{aligned}
$$

This yields the simplified minimization problem:

$$
\mathcal{P}_4^{(i)}:\begin{cases}
\mathbf{c}_{k+1}^{(i)}=\underset{\mathbf{c}^{(i)}}{\arg\min}\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}+\left(\lambda^{(i)}-\rho\delta\right)\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(i)}\right|^2+\right.\\
\left.\sum_{q=1,q\neq i}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(i)}\right|^2\right)+\frac{\rho}{2}\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(i)}\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(i)}\right|^2\right)^2 & \text{(4.3.3a)}\\
\text{s.t. } \|\mathbf{c}^{(i)}\|^2=1 & \text{(4.3.3b)}\\
\left(\mathbf{c}^{(i)}\right)^T\mathbf{R}_I\mathbf{c}^{(i)}\leq E_I & \text{(4.3.3c)}\\
\|\mathbf{c}^{(i)}-\mathbf{c}_0^{(i)}\|^2\leq\epsilon. & \text{(4.3.3d)}
\end{cases}
$$

Let us rewrite the objective function in above minimization problem as

$$
\begin{aligned}
f_0\left(\mathbf{c}^{(i)}\right)=\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}+\left(\lambda^{(i)}-\rho\delta\right)\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(i)}\right|^2+\sum_{q=1,q\neq i}^{P}\mathbf{1}^T\left|\mathbf{B}_q c^{(i)}\right|^2\right)+\\
\frac{\rho}{2}\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(i)}\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(i)}\right|^2\right)^2.
\end{aligned}
$$

Our aim is to use Dual Ascent for the following minimization problem:

$$
\begin{cases}
\mathbf{c}_{k+1}^{(i)}=\underset{\mathbf{c}^{(i)}}{\arg\min}\, f_0\left(\mathbf{c}^{(i)}\right) & \text{(4.3.4a)}\\
\left(\mathbf{c}^{(i)}\right)^T\mathbf{R}_I\mathbf{c}^{(i)}\leq E_I, & \text{(4.3.4b)}
\end{cases}
$$

and project the result of applying Dual Ascent in to region $\mathcal{D} = \left\{ \mathbf{c} \mid \|\mathbf{c}^{(i)}\|^2 = 1, \|\mathbf{c}^{(i)} - \mathbf{c}_0^{(i)}\|^2 \leq \epsilon \right\}$. Notice that the radiation energy constraint (4.3.4b) is non-convex too. As in the case of ADMM-steps (4.3.1a-4.3.1c), this means that the convergence is not guaranteed.

The Lagrangian $\mathcal{L}\left(\mathbf{c}^{(i)}, \zeta\right)$ is given as:

$$\mathcal{L}\left(\mathbf{c}^{(i)}, \zeta\right) = f_0\left(\mathbf{c}^{(i)}\right) + \zeta\left(\left(\mathbf{c}^{(i)}\right)^T \mathbf{R}_I \mathbf{c}^{(i)} - E_I\right). \tag{4.3.5}$$

The dual function $g(\zeta)$ is expressed as:

$$\begin{aligned} g(\zeta) &= \inf_{\mathbf{c}^{(i)}} \left\{ f_0\left(\mathbf{c}^{(i)}\right) + \zeta\left(\left(\mathbf{c}^{(i)}\right)^T \mathbf{R}_I \mathbf{c}^{(i)} - E_I\right) \right\} \\ &= \inf_{\mathbf{c}^{(i)}} \left\{ f_0\left(\mathbf{c}^{(i)}\right) + \zeta\left(\mathbf{c}^{(i)}\right)^T \mathbf{R}_I \mathbf{c}^{(i)} - \zeta E_I \right\} \\ &= \inf_{\mathbf{c}^{(i)}} \left\{ f_0\left(\mathbf{c}^{(i)}\right) + \zeta\left(\mathbf{c}^{(i)}\right)^T \mathbf{R}_I \mathbf{c}^{(i)} \right\} - \zeta E_I. \end{aligned}$$

The Lagrangian (4.3.5) can be minimized by Gradient Descent, and the dual function can be maximized by Gradient Ascent. This yields routine:

$$\begin{cases} \hat{\mathbf{c}}_{k+1}^{(i)} = \mathbf{c}_k^{(i)} - \dfrac{1}{L} \nabla_{\mathbf{c}^{(i)}} \mathcal{L}\left(\mathbf{c}_k^{(i)}, \zeta_k\right) & \text{(4.3.6a)} \\ \text{Project } \hat{\mathbf{c}}_{k+1}^{(i)} \text{ to region } \mathcal{D} \text{ to obtain } \mathbf{c}_{k+1}^{(i)} & \text{(4.3.6b)} \\ \zeta_{k+1} = \zeta_k + \dfrac{1}{\alpha} \dfrac{d}{d\zeta} g(\zeta). & \text{(4.3.6c)} \end{cases}$$

In (4.3.6b), we project $\hat{\mathbf{c}}_{k+1}^{(i)}$ to region $\|\mathbf{c}^{(i)}\|^2 = 1$ by dividing it by its L^2-norm as we did in Subsection 3.2.1, equation (3.2.10). The rotation to similarity region $\|\mathbf{c}^{(i)} - \mathbf{c}_0^{(i)}\|^2 \leq \epsilon$ is done by Algorithm 1. Only thing remained to be done is to find gradient $\nabla_{\mathbf{c}^{(i)}} \mathcal{L}\left(\mathbf{c}_k^{(i)}, \zeta\right)$ and derivative $\frac{d}{d\zeta} g(\zeta)$.

The derivative $\frac{d}{d\zeta} g(\zeta)$ is given as:

$$\begin{aligned} \frac{d}{d\zeta} g(\zeta) &= \frac{d}{d\zeta} \left[\inf_{\mathbf{c}^{(i)}} \left\{ f_0\left(\mathbf{c}^{(i)}\right) + \zeta\left(\mathbf{c}^{(i)}\right)^T \mathbf{R}_I \mathbf{c}^{(i)} \right\} - \zeta E_I \right] \\ &= \frac{d}{d\zeta} \left[f_0\left(\mathbf{c}_{\text{opt}}^{(i)}\right) + \zeta\left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T \mathbf{R}_I \mathbf{c}_{\text{opt}}^{(i)} - \zeta E_I \right] \\ &= \left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T \mathbf{R}_I \mathbf{c}_{\text{opt}}^{(i)} - E_I, \end{aligned}$$

where $\mathbf{c}_{\text{opt}}^{(i)}$ is obtained from steps (4.3.6a-4.3.6b). The gradient of the Lagrangian $\nabla_{\mathbf{c}^{(i)}}\mathcal{L}\left(\mathbf{c}_k^{(i)},\zeta\right)$ can be found by considering all parts of the sum separately as follows:

The 1^{st} part of the sum:

$$\nabla_{\mathbf{c}^{(i)}}\left(\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}\right)=\left(\mathbf{Q}+\mathbf{Q}^T\right)\mathbf{c}^{(i)}=2\mathbf{Q}\mathbf{c}^{(i)},$$

The 2^{nd} part of the sum:

$$\begin{aligned}
&\nabla_{\mathbf{c}^{(i)}}\left(\left(\lambda^{(i)}-\rho\delta\right)\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2\right)\right)\\
&=\left(\lambda^{(i)}-\rho\delta\right)\left(\nabla_{\mathbf{c}^{(p)}}\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2+\sum_{q=1,p\neq q}^{P}\nabla_{\mathbf{c}^{(p)}}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2\right)\\
&\overset{*}{=}\left(\lambda^{(i)}-\rho\delta\right)\left(\nabla_{\mathbf{c}^{(p)}}\mathbf{1}^T\left(\mathbf{A}\mathbf{c}^{(p)}\right)^2+\sum_{q=1,p\neq q}^{P}\nabla_{\mathbf{c}^{(p)}}\mathbf{1}^T\left(\mathbf{B}_q\mathbf{c}^{(p)}\right)^2\right)\\
&\overset{**}{=}2\left(\lambda^{(i)}-\rho\delta\right)\left(2\mathbf{A}^T\left(\mathbf{A}\mathbf{c}^{(p)}\right)+\sum_{q=1,p\neq q}^{P}\mathbf{B}_q^T\left(\mathbf{B}_q\mathbf{c}^{(p)}\right)\right)
\end{aligned}$$

The 3^{rd} part of the sum:

$$\begin{aligned}
&\nabla_{\mathbf{c}^{(i)}}\left(\frac{\rho}{2}\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(i)}\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(i)}\right|^2\right)^2\right)\\
&=\rho\left(\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(i)}\right|^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(i)}\right|^2\right)\left(\nabla_{\mathbf{c}^{(p)}}\mathbf{1}^T\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2+\sum_{q=1,p\neq q}^{P}\nabla_{\mathbf{c}^{(p)}}\mathbf{1}^T\left|\mathbf{B}_q\mathbf{c}^{(p)}\right|^2\right)\\
&\overset{*}{=}\rho\left(\mathbf{1}^T\left(\mathbf{A}\mathbf{c}^{(i)}\right)^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left(\mathbf{B}_q\mathbf{c}^{(i)}\right)^2\right)\left(\nabla_{\mathbf{c}^{(p)}}\mathbf{1}^T\left(\mathbf{A}\mathbf{c}^{(p)}\right)^2+\sum_{q=1,p\neq q}^{P}\nabla_{\mathbf{c}^{(p)}}\mathbf{1}^T\left(\mathbf{B}_q\mathbf{c}^{(p)}\right)^2\right)\\
&=2\rho\left(\mathbf{1}^T\left(\mathbf{A}\mathbf{c}^{(i)}\right)^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left(\mathbf{B}_q\mathbf{c}^{(i)}\right)^2\right)\left(2\mathbf{A}^T\left(\mathbf{A}\mathbf{c}^{(p)}\right)+\sum_{q=1,p\neq q}^{P}\mathbf{B}_q^T\left(\mathbf{B}_q\mathbf{c}^{(p)}\right)\right)
\end{aligned}$$

The 4^{th} part of the sum:

$$\nabla_{\mathbf{c}^{(i)}}\zeta\left(\left(\mathbf{c}^{(i)}\right)^T\mathbf{R}_I\mathbf{c}^{(i)}-E_I\right)=\zeta\left(\mathbf{R}_I+\mathbf{R}_I^T\right)\mathbf{c}^{(i)}\overset{\mathbf{R}_I^T=\mathbf{R}_I}{=}2\zeta\mathbf{R}_I\mathbf{c}^{(i)}$$

Then the overall gradient can be found as:

$$\nabla_{\mathbf{c}^{(i)}}\mathcal{L}\left(\mathbf{c}_k^{(i)},\zeta\right)=2\mathbf{Q}\mathbf{c}^{(i)}+2\left(\lambda^{(i)}-\rho\delta\right)\left(2\mathbf{A}^T\left(\mathbf{A}\mathbf{c}^{(i)}\right)+\sum_{q=1,p\neq q}^{P}\mathbf{B}_q^T\left(\mathbf{B}_q\mathbf{c}^{(i)}\right)\right)+$$

$$2\rho\left(\underbrace{\mathbf{1}^T\left(\mathbf{A}\mathbf{c}^{(i)}\right)^2+\sum_{q=1,p\neq q}^{P}\mathbf{1}^T\left(\mathbf{B}_q\mathbf{c}^{(i)}\right)^2}_{\in\mathbb{R}(\text{Scalar})}\right)\left(\underbrace{2\mathbf{A}^T\left(\mathbf{A}\mathbf{c}^{(i)}\right)}_{\in\mathbb{R}^{2N}}+\sum_{q=1,p\neq q}^{P}\underbrace{\mathbf{B}_q^T\left(\mathbf{B}_q\mathbf{c}^{(i)}\right)}_{\in\mathbb{R}^{2N}}\right)+2\zeta\mathbf{R}_I\mathbf{c}^{(i)}.$$

(*) Because the vector $\mathbf{A}\mathbf{c}^{(p)}$ is real valued $\left|\mathbf{A}\mathbf{c}^{(p)}\right|^2=\left(\mathbf{A}\mathbf{c}^{(p)}\right)^2$, where $(\bullet)^2$ is elementwise second power.
(**) $\nabla_{\mathbf{c}^{(p)}}\mathbf{1}^T\left(\mathbf{A}\mathbf{c}^{(p)}\right)^2=4\mathbf{A}^T\mathbf{A}\mathbf{c}^{(p)}$.

We obtained the final algorithm for multiple waveforms design as in Algorithm 6.

Algorithm 6: 2^{nd} multiple waveforms design algorithm

1 function Multiplewaveform2($\mathbf{Q},\{\mathbf{c}_0^{(p)}\}_{p\in\{1,...,P\}},\mathbf{R}_I,E_I,\epsilon,\delta,K$);
  Input : $\mathbf{Q}=\mu\mathbf{I}-\mathbf{R}\succeq\mathbf{0}$, $\{\mathbf{c}_0^{(p)}\}_{p\in\{1,...,P\}}$, $\mathbf{R}_I$, E_I, ϵ and K
  Output : $\{\mathbf{c}^{(p)}\}_{p\in\{1,...,P\}}$
2 Initialize $\{\mathbf{c}^{(p)}\}_{p\in\{1,...,P\}}$ and $\{\lambda^{(p)}\}_{p\in\{1,...,P\}}$;
3 **for** $k=1,k\leq K,k$ ++ **do**
4 Solve $\mathcal{P}_4^{(i)},\forall i\in\{1,...,P\}$;
5 $\lambda_{k+1}^{(i)}=\lambda_k^{(i)}+\rho\left(\text{SL2}\left(\mathbf{c}_{k+1}^{(i)}\right)-\delta\right),\forall i\in\{1,...,P\}$;
6 **end**

where $\mathcal{P}_4^{(i)}$ are solved as shown in Algorithm 7.

Algorithm 7: Solve $\mathcal{P}_4^{(i)}$

function Solve_$\mathcal{P}_4^{(i)}(\mathbf{Q}, \mathbf{c}^{(i)}, \mathbf{c}_0^{(i)}, \mathbf{R}_I, E_I, \epsilon, \delta, K)$;

Input : $\mathbf{Q} = \mu\mathbf{I} - \mathbf{R} \succeq \mathbf{0}$, $\mathbf{c}_0^{(i)}$, $\mathbf{R}_I$, E_I, ϵ and K

Output : $\mathbf{c}^{(i)}$

Initialize $\zeta^{(i)}$;

for $k = 1, k \leq K, k$ ++ **do**

$\hat{\mathbf{c}}_{k+1}^{(i)} = \mathbf{c}_k^{(i)} - \frac{1}{L}\nabla_{\mathbf{c}^{(i)}}\mathcal{L}\left(\mathbf{c}_k^{(i)}, \zeta_k\right)$;

$\widetilde{\mathbf{c}}_{k+1}^{(i)} = \frac{\hat{\mathbf{c}}_{k+1}^{(i)}}{\left\|\hat{\mathbf{c}}_{k+1}^{(i)}\right\|}$;

$\mathbf{c}_{k+1}^{(i)} = \text{RotateVector}(\widetilde{\mathbf{c}}_{k+1}^{(i)}, \mathbf{c}_0^{(i)}, \alpha, \epsilon)$;

$\zeta_{k+1}^{(i)} = \zeta_k^{(i)} + \frac{1}{\alpha}\left(\left(\mathbf{c}_{\mathrm{k+1}}^{(i)}\right)^T \mathbf{R}_I \mathbf{c}_{\mathrm{k+1}}^{(i)} - E_I\right)$;

end

4.3.2 Performance analysis and example simulation

In Figure 15, the time complexity graph of Algorithm 6 is plotted alongside reference curves ranging from time-complexities $O(Nlog(N))$ to $O(N^2)$. The runtime is for one iteration of Algorithm 6 when we design three distinctive waveforms (i.e., $P = 3$) on desktop computer (HP Z240 Tower Workstation with Xeon E3-1230v5 3.40GHz 8MB processor). By comparing the slope of runtime of Algorithm 6 to slopes of reference curves, we see that time complexity of Algorithm 6 is approximately quadratic, i.e., $O(N^2)$, which is expected since the heaviest operation we do is matrix to vector product. This was also the case in Algorithm 2, and therefore the time complexity graphs of Algorithm 2 and Algorithm 6 are similar.

Although Algorithm 6 is efficient, it has some technical issues to point out. We discussed in Subsection 4.3 that the side-lobe level constraint (4.1.6e) and radiation energy constraint (4.1.6c) are non-convex. In addition to these, we had more than two primal variable updates in ADMM steps (4.3.1). These features mean that by applying ADMM and Dual Ascent the convergence of the method is not guaranteed.

We also face this issue when applying the method in example environment. For randomly generated data (i.e. matrices $\mathbf{Q} \succeq \mathbf{0}, \mathbf{R}_I$, s.t. $\mathbf{R}_I^T = \mathbf{R}_I$ and vectors $\mathbf{c}, \mathbf{c}_0$ are randomly generated) the method is able to minimize the objective and satisfy constraints, as can be seen in Figure 16, but in an example environment similar to the environment of Subsection 4.2.3, the method is able to find a solution inside feasible region (i.e., region where constraints are satisfied), but cannot do any minimization inside the feasible region. For future work, it would make sense to analyze this method further and try to make it more reliable, since although it is quick it does not guarantee convergence in all situtations.

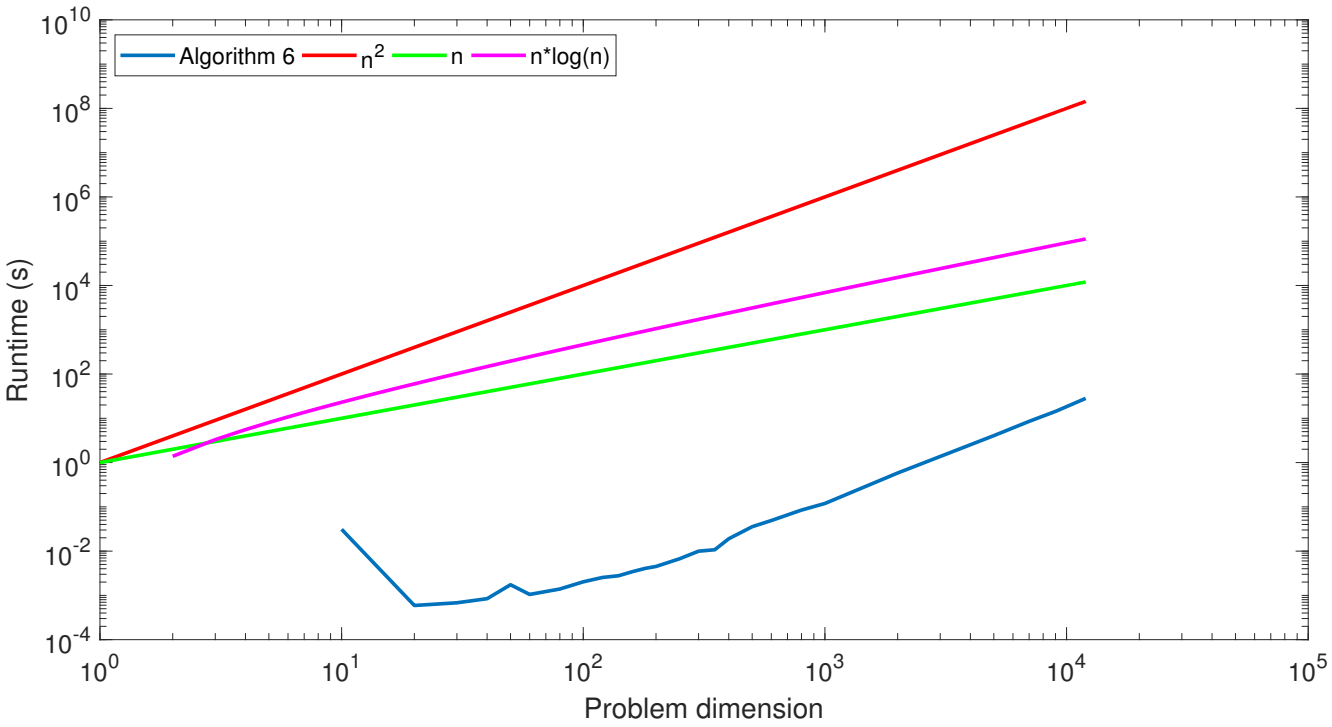

Figure 15: Time-complexity graph of Algorithm 6

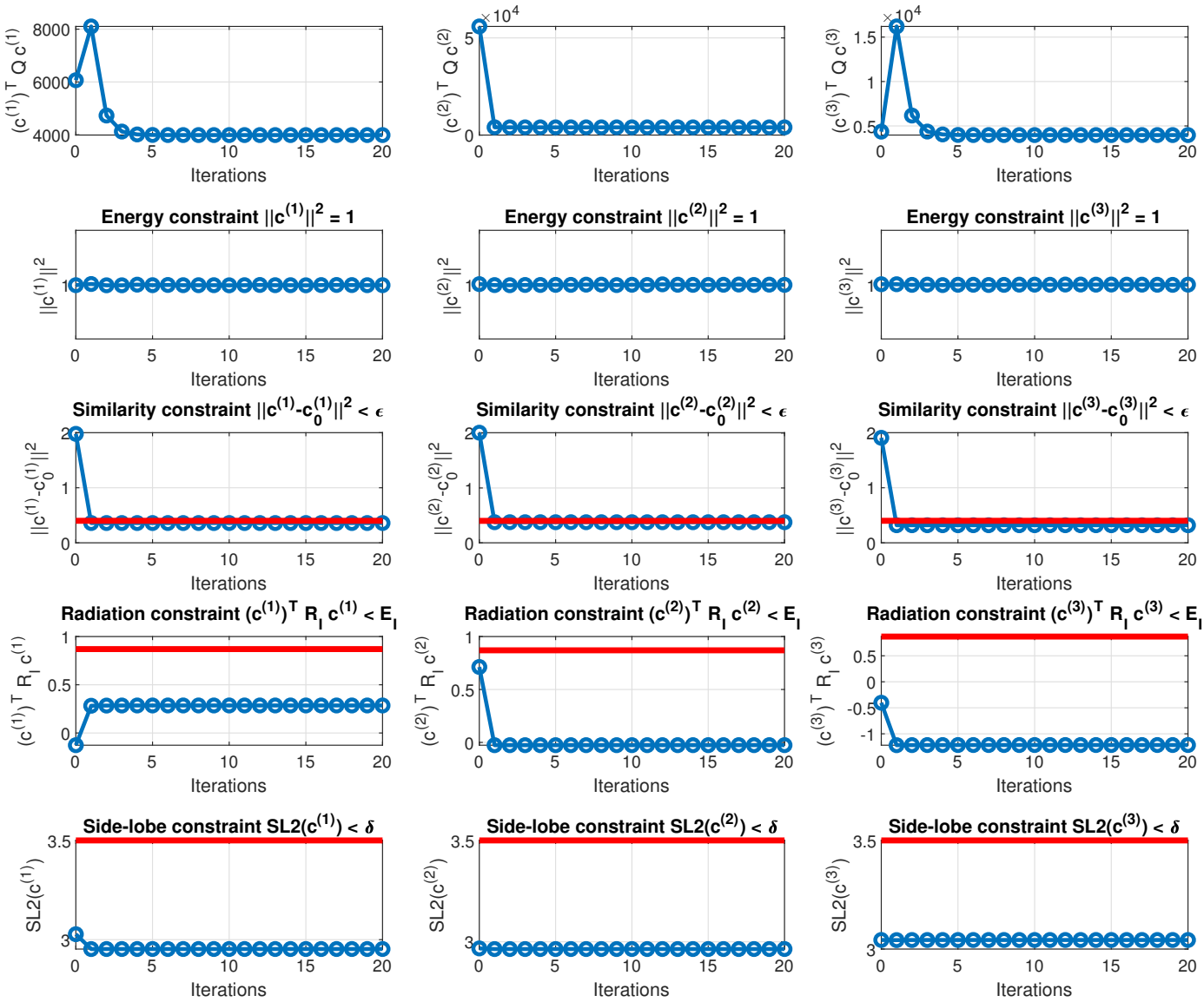

Figure 16: Convergence graph of Algorithm 6 on randomly generated data

4.4 3rd algorithm for multiple waveform design

In this section, we develop yet another algorithm for addressing multiple waveforms design problem $\mathcal{P}_2^{(3)}$. In addition to constraints (4.1.6b-4.1.6e), we consider now PAPR constraint. PAPR constraint can be written as:

$$\text{PAPR}(\mathbf{c}^{(p)}) = \frac{\max_n \left|\mathbf{c}^{(p)}(n)\right|^2}{\frac{1}{N}\|\mathbf{c}^{(p)}\|^2} \leq \eta, \eta \geq 1, \tag{4.4.1}$$

where N denotes waveform length. The maximum operator in numerator of (4.4.1) can be eliminated by rewriting (4.4.1) as:

$$\frac{\left|\mathbf{c}^{(p)}(n)\right|^2}{\frac{1}{N}\|\mathbf{c}^{(p)}\|^2} \leq \eta, \forall n \in \{1, 2, ..., N\}. \tag{4.4.2}$$

By constraint (4.1.6b), we have that $\|\mathbf{c}^{(p)}\|^2 = 1$. Thus:

$$\left|\mathbf{c}^{(p)}(n)\right|^2 \leq \frac{\eta}{N}, \forall n \in \{1, 2, ..., N\}. \tag{4.4.3}$$

If we introduce matrices $\mathbf{E}_n$, in which $\mathbf{E}_n(i,j) = \begin{cases} 1, & i = n \text{ and } j = n \\ 0, & \text{otherwise} \end{cases}$, we can rewrite (4.4.3) as:

$$\left(\mathbf{c}^{(p)}\right)^H \mathbf{E}_n \mathbf{c}^{(p)} \leq \frac{\eta}{N}, \forall n \in \{1, 2, ..., N\}. \tag{4.4.4}$$

By adding constraint (4.4.1) to $\mathcal{P}_2^{(3)}$, we get the following optimization problem:

$$\mathcal{P}_5 : \begin{cases} \min\limits_{c} & \sum\limits_{p=1}^{P} (\mathbf{c}^{(p)})^T \mathbf{Q} \mathbf{c}^{(p)} & \text{(4.4.5a)} \\ \text{s.t. :} & \|\mathbf{c}^{(p)}\|^2 = 1, \quad \forall p \in \{1, ..., P\} & \text{(4.4.5b)} \\ & \left(\mathbf{c}^{(p)}\right)^T \mathbf{E}_n \mathbf{c}^{(p)} \leq \frac{\eta}{N}, n \in \{1, ..., N\}, \forall p & \text{(4.4.5c)} \\ & (\mathbf{c}^{(p)})^T \mathbf{R}_I \mathbf{c}^{(p)} \leq E_I, \quad \forall p \in \{1, ..., P\} & \text{(4.4.5d)} \\ & \left\|\mathbf{c}^{(p)} - \mathbf{c}_0^{(p)}\right\|^2 \leq \epsilon, \quad \forall p \in \{1, ..., P\} & \text{(4.4.5e)} \\ & SL2(\mathbf{c}^{(p)}) \leq \delta, \quad \forall p \in \{1, ..., P\}. & \text{(4.4.5f)} \end{cases}$$

For given $\mathbf{c}^{(k)}, \forall k \neq i$, $\mathcal{P}_5$ can be decomposed into single waveform updates as:

$$\mathcal{P}_5^{(i)}: \begin{cases} \min\limits_{\mathbf{c}^{(i)}} & \sum\limits_{p=1}^{P} (\mathbf{c}^{(i)})^T \mathbf{Q} \mathbf{c}^{(i)} & (4.4.6a) \\ \text{s.t. :} & \|\mathbf{c}^{(i)}\|^2 = 1, & (4.4.6b) \\ & \left(\mathbf{c}^{(i)}\right)^T \mathbf{E}_n \mathbf{c}^{(i)} \leq \frac{\eta}{N}, n \in \{1, ..., N\}, & (4.4.6c) \\ & (\mathbf{c}^{(i)})^T \mathbf{R}_I \mathbf{c}^{(i)} \leq E_I, & (4.4.6d) \\ & \left\| \mathbf{c}^{(i)} - \mathbf{c}_0^{(i)} \right\|^2 \leq \epsilon, & (4.4.6e) \\ & SL2(\mathbf{c}^{(i)}) \leq \delta. & (4.4.6f) \end{cases}$$

To solve $\mathcal{P}_5^{(i)}$ we use 1^{st} algorithm we developed for multiple waveforms design in Section 4.2. We implement additional constraint (4.4.6c) in **c**-variable update, while the **z**-variable update remains the same.

4.4.1 c-variable update

We have the following minimization problem:

$$\mathbf{c}_{k+1}^{(i)} = \arg\min_{\mathbf{c}^{(i)}} (\mathbf{c}^{(i)})^T \mathbf{Q} \mathbf{c}^{(i)} + (\boldsymbol{\lambda} - \rho \mathbf{z})^T \mathbf{c}^{(i)},$$
$$\text{s.t. } \|\mathbf{c}^{(i)}\|^2 = 1, \ \left\| \mathbf{c}^{(i)} - \mathbf{c}_0^{(i)} \right\|^2 \leq \epsilon, \left(\mathbf{c}^{(i)}\right)^T \mathbf{E}_n \mathbf{c}^{(i)} \leq \frac{\eta}{N}, n \in \{1, ..., N\}. \quad (4.4.7)$$

The Lagrangian $L\left(\mathbf{c}^{(i)}, \boldsymbol{\xi}\right)$ for (4.4.7) by using constraint (4.4.6c) can be written as:

$$L\left(\mathbf{c}^{(i)}, \boldsymbol{\xi}\right) = \left(\mathbf{c}^{(i)}\right)^T \mathbf{Q} \mathbf{c}^{(i)} + (\boldsymbol{\lambda} - \rho z)^T \mathbf{c}^{(i)} + \sum_{n=1}^{N} \xi_n \left(\left(\mathbf{c}^{(i)}\right)^T \mathbf{E}_n \mathbf{c}^{(i)} - \frac{\eta}{N} \right). \quad (4.4.8)$$

The dual function $g(\boldsymbol{\xi})$ can be found as:

$$
\begin{aligned}
g(\boldsymbol{\xi}) &= \inf_{\mathbf{c}^{(i)}} L\left(\mathbf{c}^{(i)}, \boldsymbol{\xi}\right) = \inf_{\mathbf{c}^{(i)}} \left\{ \left(\mathbf{c}^{(i)}\right)^T \mathbf{Q}\mathbf{c}^{(i)} + (\boldsymbol{\lambda} - \rho\mathbf{z})^T \mathbf{c}^{(i)} + \right. \\
&\left. \sum_{n=1}^{N} \xi_n \left(\left(\mathbf{c}^{(i)}\right)^T \mathbf{E}_n \mathbf{c}^{(i)} - \frac{\eta}{N} \right) \right\} \qquad (4.4.9) \\
&= \inf_{\mathbf{c}^{(i)}} \left\{ \left(\mathbf{c}^{(i)}\right)^T \mathbf{Q}\mathbf{c}^{(i)} + (\boldsymbol{\lambda} - \rho\mathbf{z})^T \mathbf{c}^{(i)} + \sum_{n=1}^{N} \xi_n \left(\mathbf{c}^{(i)}\right)^T \mathbf{E}_n \mathbf{c}^{(i)} - \frac{\eta}{N} \sum_{n=1}^{N} \xi_n \right\} \\
&= \inf_{\mathbf{c}^{(i)}} \left\{ \left(\mathbf{c}^{(i)}\right)^T \mathbf{Q}\mathbf{c}^{(i)} + (\boldsymbol{\lambda} - \rho\mathbf{z})^T \mathbf{c}^{(i)} + \sum_{n=1}^{N} \xi_n \left(\mathbf{c}^{(i)}\right)^T \mathbf{E}_n \mathbf{c}^{(i)} \right\} - \frac{\eta}{N} \sum_{n=1}^{N} \xi_n \\
&\overset{*}{=} \left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T \mathbf{Q}\mathbf{c}_{\text{opt}}^{(i)} + (\boldsymbol{\lambda} - \rho\mathbf{z})^T \mathbf{c}_{\text{opt}}^{(i)} + \sum_{n=1}^{N} \xi_n \left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T \mathbf{E}_n \mathbf{c}_{\text{opt}}^{(i)} - \frac{\eta}{N} \sum_{n=1}^{N} \xi_n . \qquad (4.4.10)
\end{aligned}
$$

(*) $\mathbf{c}_{\text{opt}}^{(i)}$ obtained by first step in dual ascent (primal problem minimization).

Denote $\Omega^{(i)} = \left\{ \mathbf{c}^{(i)} \in \mathbb{R}^{2N} \mid \|\mathbf{c}^{(i)}\|^2 = 1 \text{ and } \|\mathbf{c}^{(i)} - \mathbf{c}_0^{(i)}\|^2 \leq \epsilon \right\}$. By using Dual ascent to the problem

$$
\begin{cases}
\min\limits_{\mathbf{c}} & (\mathbf{c}^{(i)})^T \mathbf{Q}\mathbf{c}^{(i)} + (\boldsymbol{\lambda} - \rho\mathbf{z})^T \mathbf{c}^{(i)} \qquad (4.4.11a) \\
\text{s.t.:} & \left(\mathbf{c}^{(i)}\right)^T \mathbf{E}_n \mathbf{c}^{(i)} \leq \frac{\eta}{N}, n \in \{1, ..., N\}, \qquad (4.4.11b)
\end{cases}
$$

we get the routine:

$$
\begin{cases}
\hat{\mathbf{c}}_{k+1}^{(i)} = \mathbf{c}_k^{(i)} - \dfrac{1}{L} \nabla_{\mathbf{c}^{(i)}} \mathcal{L}\left(\mathbf{c}_k^{(i)}, \boldsymbol{\xi}_k\right) \qquad (4.4.12a) \\
\text{Project } \hat{\mathbf{c}}_{k+1}^{(i)} \text{ to region } \Omega^{(i)} \text{ to obtain } \mathbf{c}_{k+1}^{(i)} \qquad (4.4.12b) \\
\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \dfrac{1}{\alpha} \nabla_{\boldsymbol{\xi}} g(\boldsymbol{\xi}_k). \qquad (4.4.12c)
\end{cases}
$$

The gradients can be found as:

$$\nabla_{\mathbf{c}^{(i)}}\mathcal{L}\left(\mathbf{c}^{(i)},\boldsymbol{\xi}\right)=\left(\mathbf{Q}^T+\mathbf{Q}\right)\mathbf{c}^{(i)}+(\boldsymbol{\lambda}-\rho\mathbf{z})+\sum_{n=1}^{N}\xi_n\left(\mathbf{E}_n^T+\mathbf{E}_n\right)\mathbf{c}^{(i)}$$

$$\overset{\mathbf{Q}^T=\mathbf{Q},\mathbf{E}_n^T=\mathbf{E}_n}{=}2\mathbf{Q}\mathbf{c}^{(i)}+(\boldsymbol{\lambda}-\rho\mathbf{z})+2\sum_{n=1}^{N}\xi_n\mathbf{E}_n\mathbf{c}^{(i)},$$

$$\begin{aligned}\frac{\partial}{\partial\xi_k}g(\boldsymbol{\xi})&=\frac{\partial}{\partial\xi_k}\left[\left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T\mathbf{Q}\mathbf{c}_{\text{opt}}^{(i)}+(\boldsymbol{\lambda}-\rho\mathbf{z})^T\mathbf{c}_{\text{opt}}^{(i)}+\right.\\&\left.\sum_{n=1}^{N}\xi_n\left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T\mathbf{E}_n\mathbf{c}_{\text{opt}}^{(i)}-\frac{\eta}{N}\sum_{n=1}^{N}\xi_n\right]\\&=\left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T\mathbf{E}_k\mathbf{c}_{\text{opt}}^{(i)}-\frac{\eta}{N},\end{aligned}$$

$$\nabla_{\boldsymbol{\xi}}g(\boldsymbol{\xi})=\begin{pmatrix}\left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T\mathbf{E}_1\mathbf{c}_{\text{opt}}^{(i)}-\frac{\eta}{N}\\\left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T\mathbf{E}_2\mathbf{c}_{\text{opt}}^{(i)}-\frac{\eta}{N}\\\vdots\\\left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T\mathbf{E}_N\mathbf{c}_{\text{opt}}^{(i)}-\frac{\eta}{N}\end{pmatrix}.$$

Steps (4.4.12a) and (4.4.12c) are straightforward. The projections in step (4.4.12b) are performed as in the single waveform update case.

Now we can write the 3^{rd} multiple waveforms design algorithm as Algorithm 8, where $\mathcal{P}_5^{(i)}$ is solved as shown in Algorithm 9. The system of equations in line 8 in Algorithm 9 can be solved by Newton's method (4.2.9).

Algorithm 8: 3^{rd} multiple waveforms design algorithm

1 function Multiplewaveform3($\mathbf{Q}, \{\mathbf{c}_0^{(p)}\}_{p\in\{1,...,P\}}, \mathbf{R}_I, E_I, \epsilon, \delta, K$);
  Input : $\mathbf{Q}=\mu\mathbf{I}-\mathbf{R}\succeq\mathbf{0}$, $\{\mathbf{c}_0^{(p)}\}_{p\in\{1,...,P\}}$, $\mathbf{R}_I$, E_I, ϵ, δ and K
  Output : $\{\mathbf{c}^{(p)}\}_{p\in\{1,...,P\}}$
2 Initialize $\{\mathbf{c}^{(p)}\}_{p\in\{1,...,P\}}$, $\{\mathbf{z}^{(p)}\}_{p\in\{1,...,P\}}$ and $\{\boldsymbol{\lambda}^{(p)}\}_{p\in\{1,...,P\}}$;
3 **for** $k=1, k\leq K, k$ ++ **do**
4 Solve $\mathcal{P}_5^{(i)}, \forall i\in\{1,...,P\}$;
5 Update cross-correlation matrices $\mathbf{B}_q, \forall q\in\{1,...,P\}$;
6 **end**

Algorithm 9: Solve $\mathcal{P}_5^{(i)}$

function Solve_$\mathcal{P}_5^{(i)}(\mathbf{Q}, \mathbf{c}_0^{(i)}, \mathbf{R}_I, E_I, \epsilon, \delta, K)$;

Input : $\mathbf{Q} = \mu\mathbf{I} - \mathbf{R} \succeq \mathbf{0}$, $\mathbf{c}_0^{(i)}$, $\mathbf{R}_I$, E_I, ϵ, δ and K

Output : $\mathbf{c}^{(i)}$

Initialize $\mathbf{c}^{(i)}$, $\mathbf{z}^{(i)}$, $\boldsymbol{\zeta}^{(i)}$ and $\boldsymbol{\lambda}^{(i)}$;

for $k = 1, k \leq K, k$ ++ **do**

$\hat{\mathbf{c}}_{k+1}^{(i)} = \mathbf{c}_k^{(i)} - \frac{1}{L}\nabla_{\mathbf{c}^{(i)}}\mathcal{L}\left(\mathbf{c}_k^{(i)}, \boldsymbol{\zeta}_k\right)$;

$\widetilde{\mathbf{c}}_{k+1}^{(i)} = \frac{\hat{\mathbf{c}}_{k+1}^{(i)}}{\left\|\hat{\mathbf{c}}_{k+1}^{(i)}\right\|}$;

$\mathbf{c}_{k+1}^{(i)} = \text{RotateVector}(\widetilde{\mathbf{c}}_{k+1}^{(i)}, \mathbf{c}_0^{(i)}, \alpha, \epsilon)$;

$\boldsymbol{\zeta}_{k+1}^{(i)} = \boldsymbol{\zeta}_k^{(i)} + \frac{1}{\alpha}\nabla_{\boldsymbol{\zeta}} g(\boldsymbol{\zeta}^{(i)})$;

Solve system (4.2.8a-4.2.8b);

$\mathbf{z}_{k+1}^{(i)} = \left(\mathbf{I} + \gamma\mathbf{R}_I + 2\nu\mathbf{A}^T\mathbf{A} + \sum_{q=1,p\neq q}^{P} \nu\mathbf{B}_q^T\mathbf{B}_q\right)^{-1}\left(\mathbf{c}^{(i)} + \frac{1}{\rho}\boldsymbol{\lambda}^{(i)}\right)$;

$\boldsymbol{\lambda}_{k+1}^{(i)} = \boldsymbol{\lambda}_k^{(i)} + \rho(\mathbf{c}_{k+1}^{(i)} - \mathbf{z}_{k+1}^{(i)})$;

end

4.4.2 Performance analysis

In Figure 17, the time complexity graph of Algorithm 8 is drawn alongside reference curves ranging from time-complexities $Nlog(N)$ to N^3. The blue line is the runtime of Algorithm 8 for one iteration when designing 3 different waveforms (i.e., $P = 3$) on desktop computer (HP Z240 Tower Workstation with Xeon E3-1230v5 3.40GHz 8MB processor). By comparing the slope of Algorithm 8 runtime curve to reference curves, we can see that the time complexity is approximately cubic (i.e. $O(N^3)$). This is similar to runtime of Algorithm 4, which is expected because Algorithms 4 and 8 are based on the same implementation.

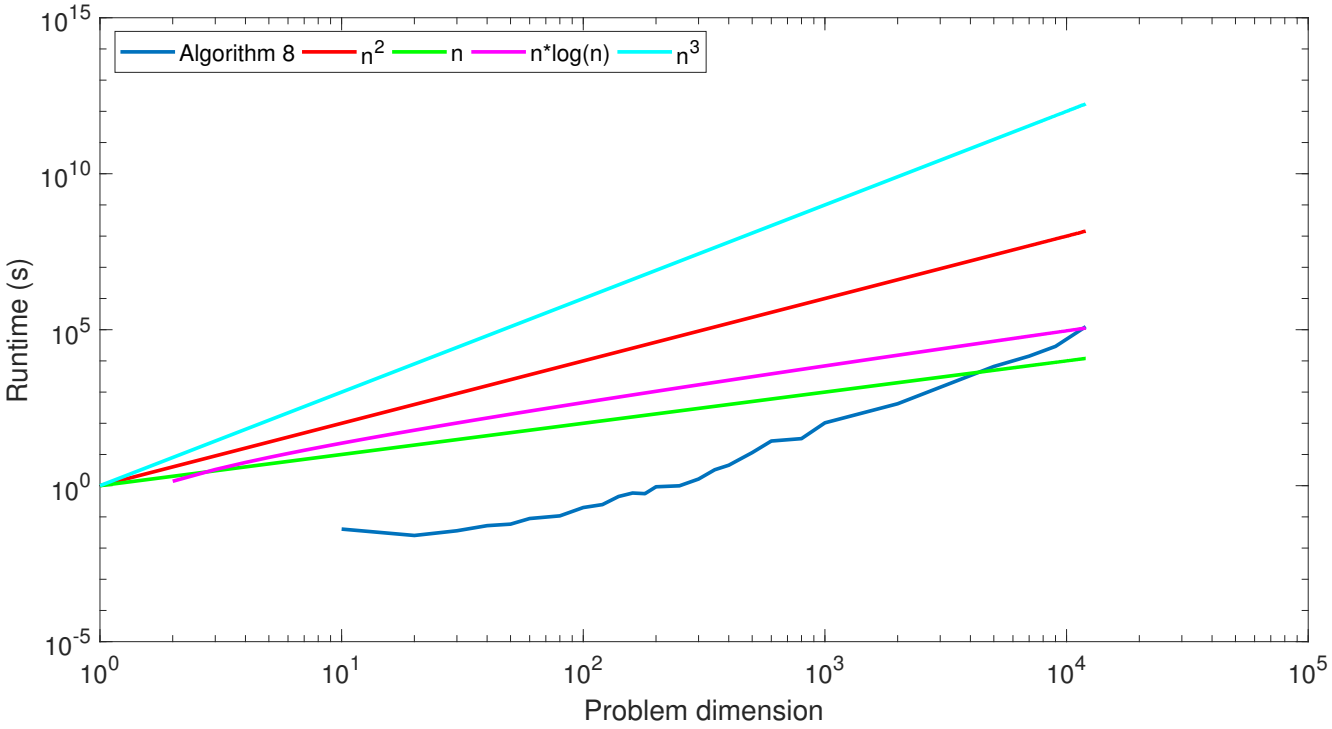

Figure 17: Time-complexity graph of Algorithm 8

4.4.3 Simulation set up

In this Subsection we use Algorithm 8 in an example environment similar to the environment of Subsection 4.2.3. All parameter values are the same as in simulation in Subsection 4.2.3, but we also add PAPR-constraint with the constraint level of $\delta = 5.5$.

In Figure 18, the convergence graph of the objective is shown alongside constraint levels per iteration. In Figure 19, the frequency spectrums of the designed waveforms $\{\mathbf{c}^{(p)}\}_{p\in\{1,2,3\}}$ are plotted. In Figure 20, SINR values of the designed waveforms are shown per each iteration, and finally in Figures 21, 22 and 23, the ambiguity functions of the designed waveforms are plotted.

By comparing designed waveforms to the waveforms obtained by Algorithm 4, we can see that Algorithms 4 and 8 yield very similar waveforms. The only difference between these algorithms is that Algorithm 8 includes PAPR-constraint, and in the cases where PAPR-constraint is active these two methods can yield different results.

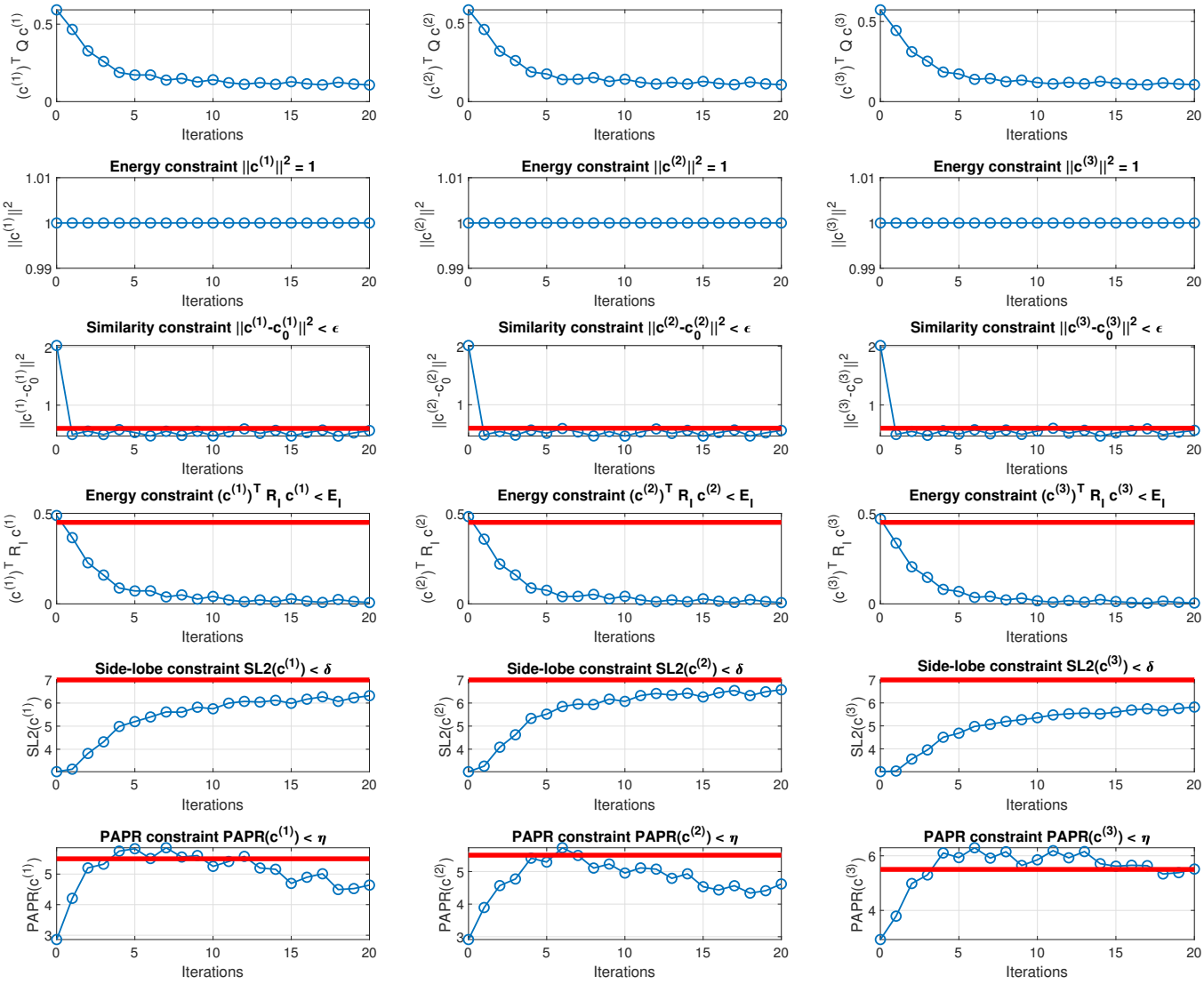

Figure 18: Convergence graph of Algorithm 8

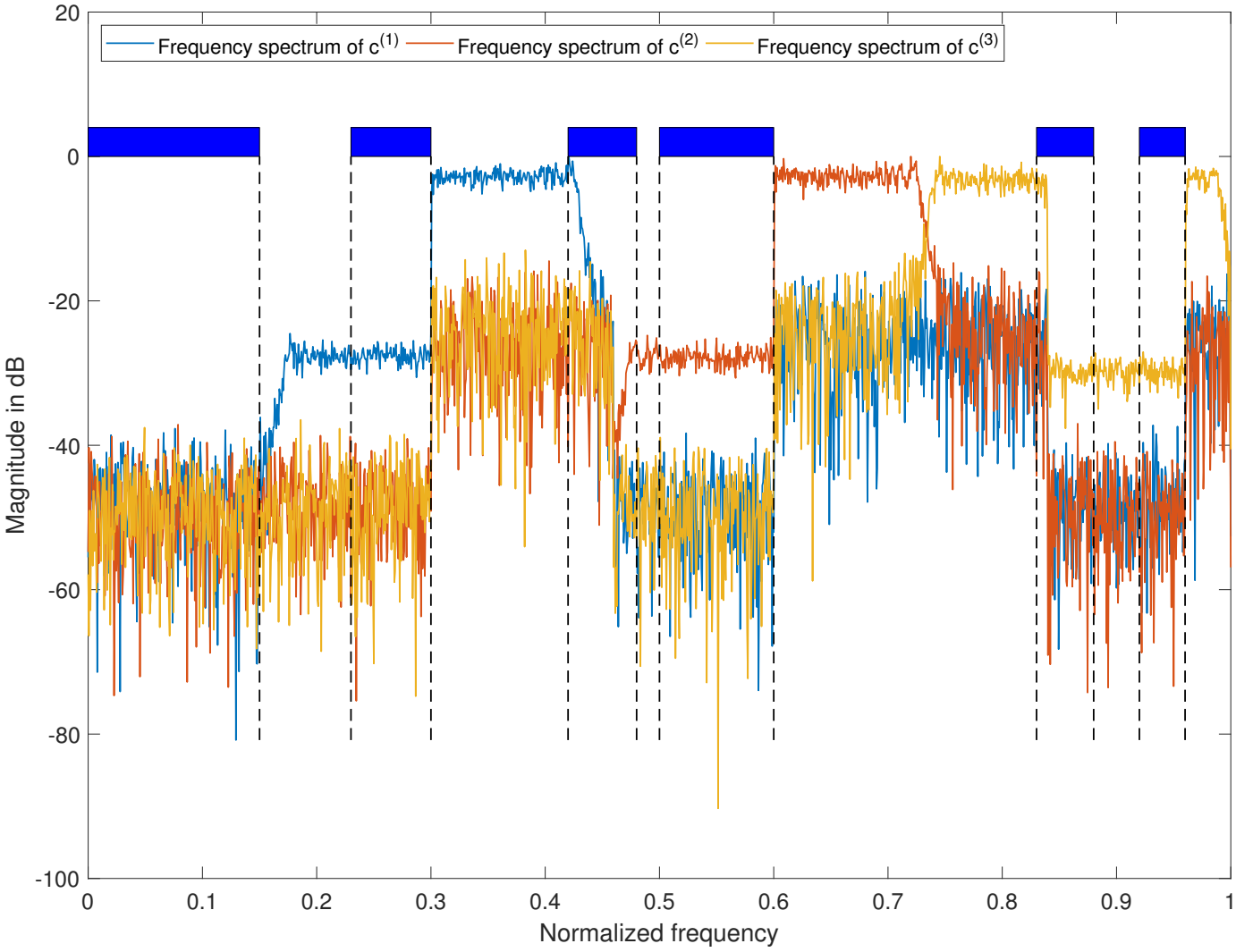

Figure 19: Frequency spectrum of designed signals $\{\mathbf{c}^{(p)}\}_{p\in\{1,2,3\}}$

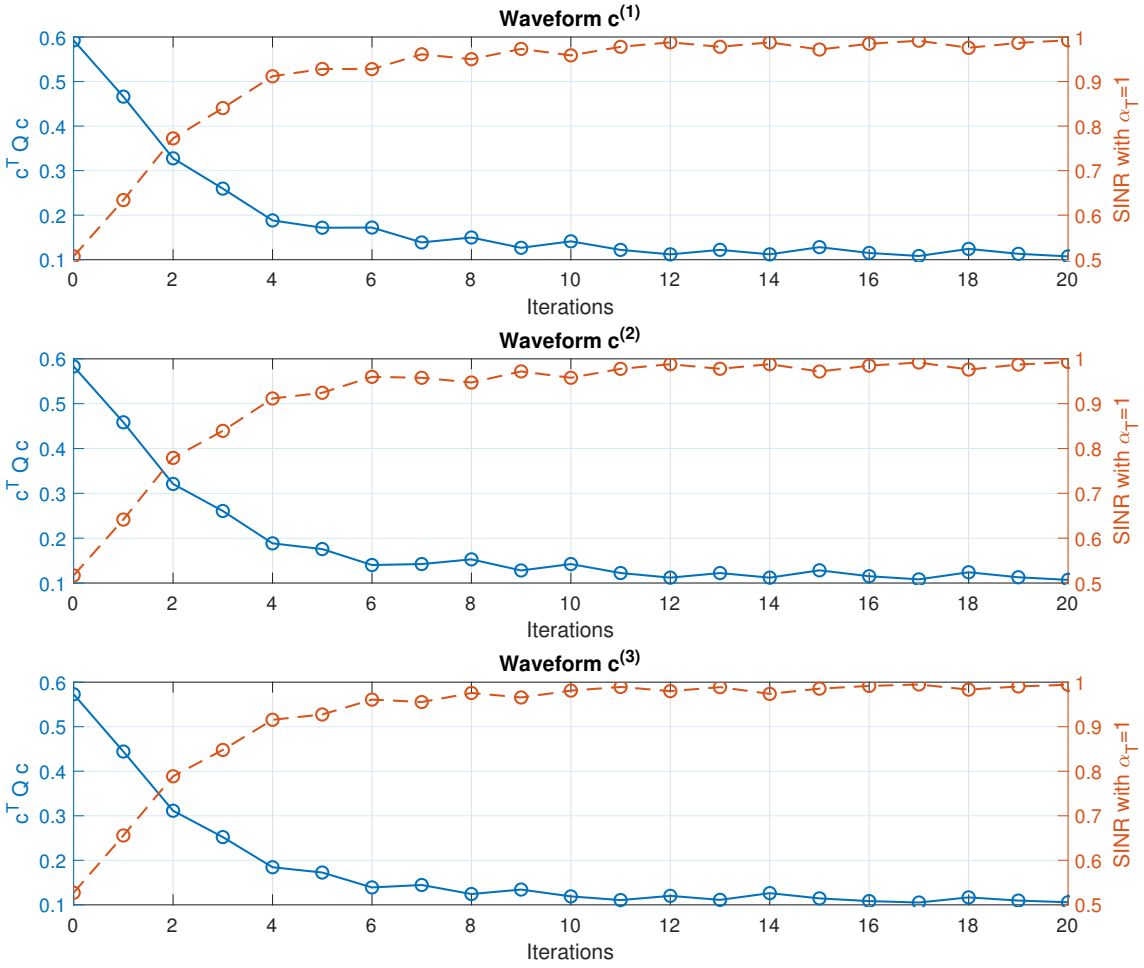

Figure 20: SINR values per iteration of designed signals $\{\mathbf{c}^{(p)}\}_{p\in\{1,2,3\}}$

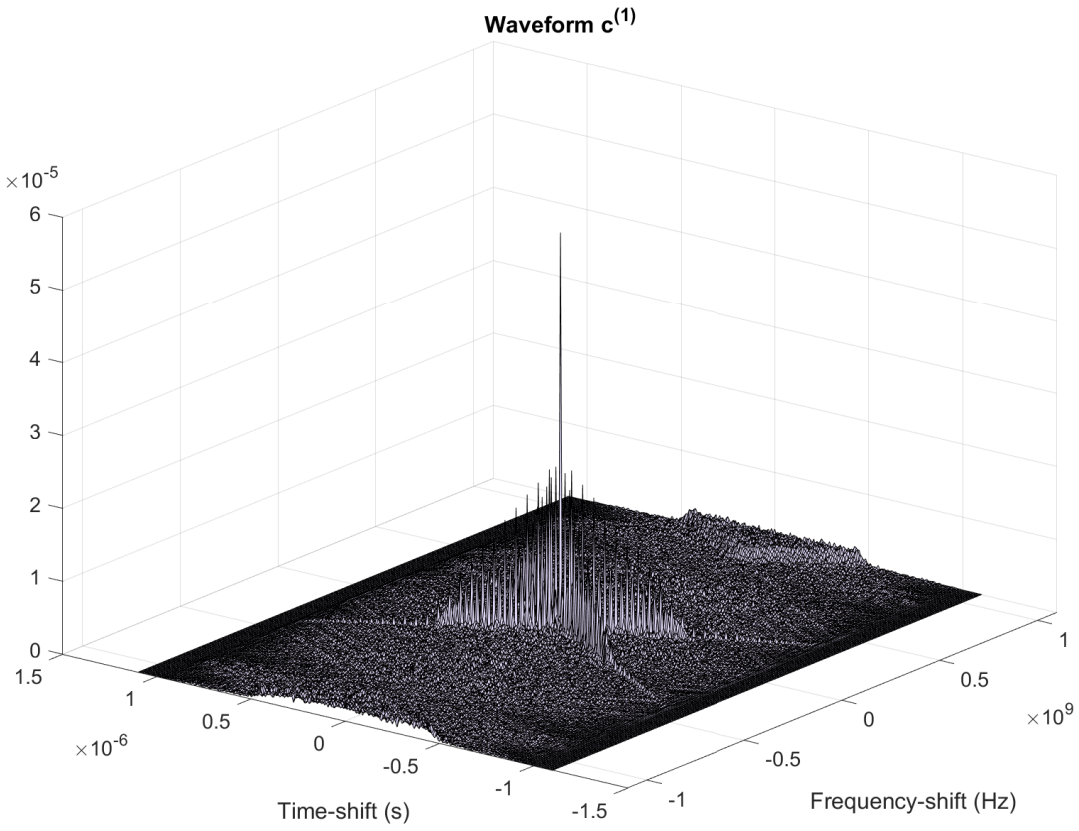

Figure 21: Ambiguity function of designed waveform $\mathbf{c}^{(1)}$

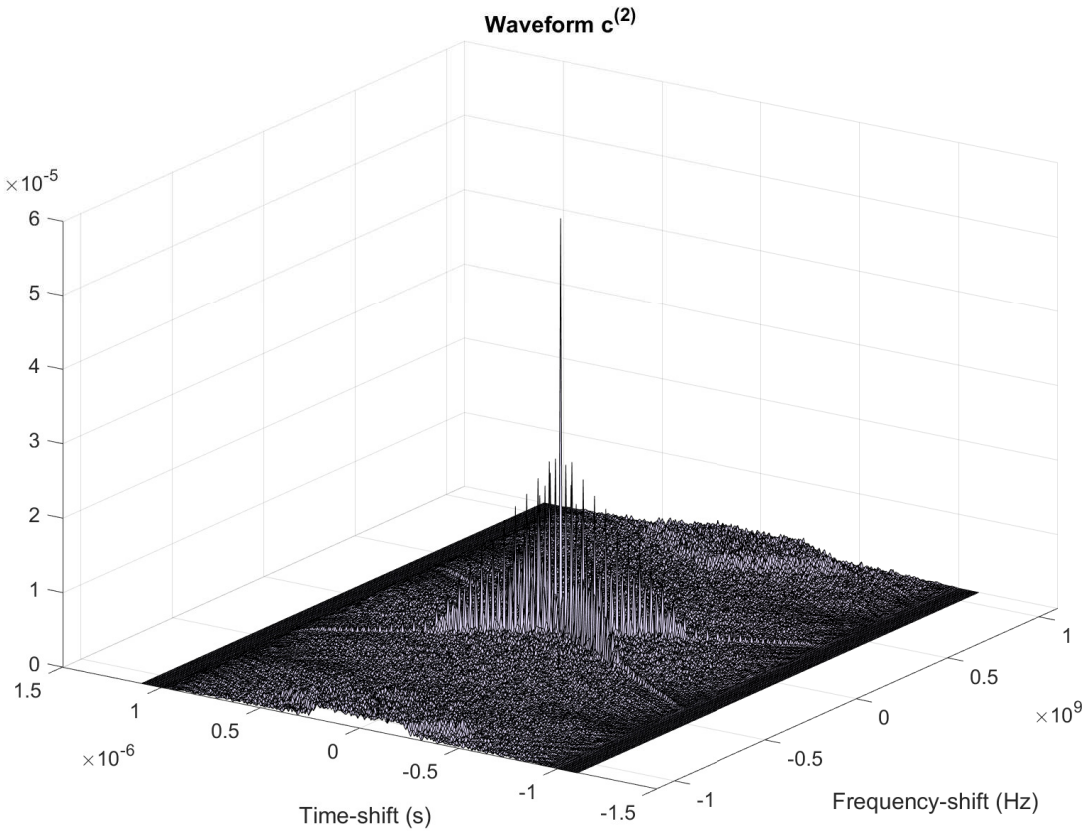

Figure 22: Ambiguity function of designed waveform $\mathbf{c}^{(2)}$

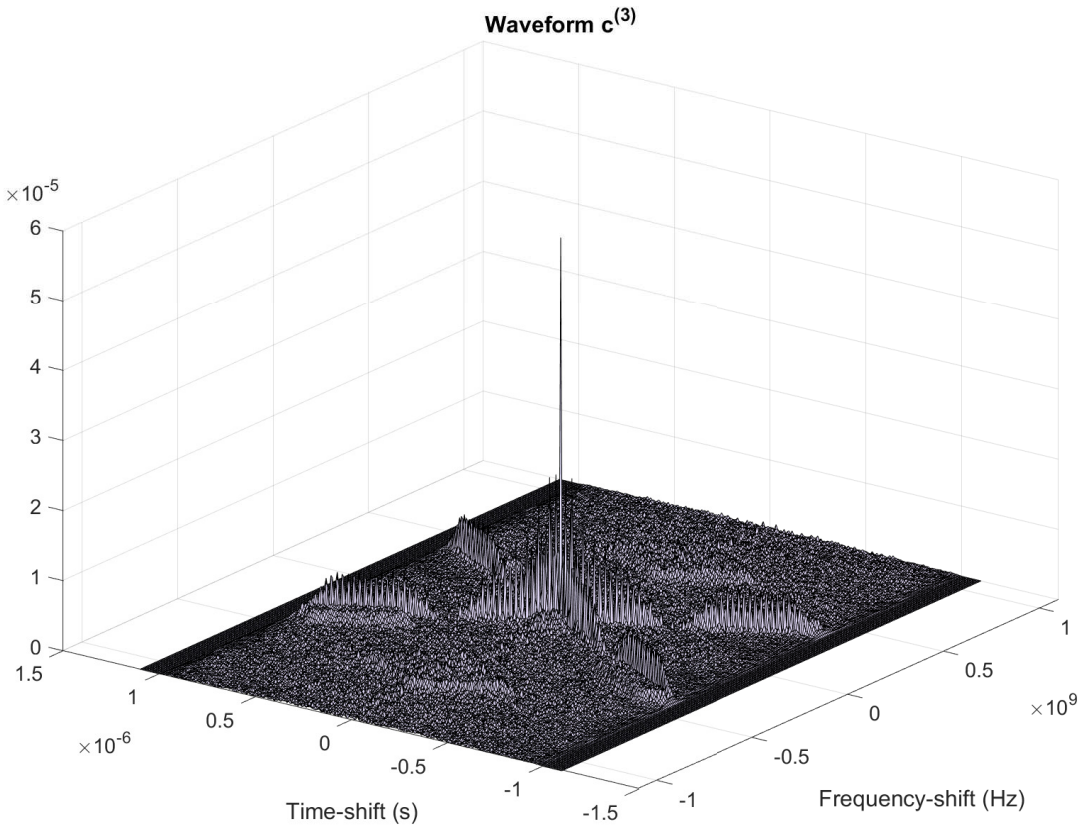

Figure 23: Ambiguity function of designed waveform $\mathbf{c}^{(3)}$

5 Conclusion

In this thesis, we developed computationally efficient algorithms for radar transmit waveform design in spectrally busy environment. We derived one algorithm (i.e., Algorithm 2) for single waveform design, in which only one transmit waveform is designed, and three algorithms (i.e., Algorithm 4, 6 and 8) for multiple waveform design, in which multiple transmit waveforms are designed simultaneously. The developed algorithms have been tested by means of simulations and their computational complexities have been shown in terms of time complexity graphs. The multiple waveforms algorithm simulations have been performed by designing three different transmit signals simultaneously. For Algorithm 6 we have only shown simulation with random data (i.e., vectors and matrices have been random) because of the convergence issues with the algorithm. This is due to the shortcomings of Algorithm 6 which have been discussed in Subsection 4.3.

All four algorithms are based on ADMM, which is one of the most powerful convex optimization algorithms up-to-date. ADMM guarantees fast convergence with general convergence properties which are desirable in this type of applications. Also Dual Ascent algorithm have been used in sub-problems of Algorithms 6 and 8.

By examining the time complexity graphs of the algorithms (see Figures 3, 8, 15, and 17) we have shown that Algorithms 2 and 6 reached quadratic time-complexity (i.e., time-complexity $O(N^2)$, where N is problem dimension), while Algorithms 4 and 8 reached cubic time complexity (i.e., $O(N^3)$). This is due to the fact that Algorithms 4 and 8 performed several matrix inversions, while Algorithms 2 and 6 avoided them completely. Matrix inversion has roughly cubic time-complexity (i.e., $O(N^3)$). Although passing quadratic time complexity, Algorithms 4 and 8 are still suitable for large-scale optimization problems as long as matrix inversion is computationally feasible operation.

By examining the ambiguity graphs (i.e., Figures 7, 12, 13, 14, 21, 22, and 23), we have shown that the designed signals by Algorithms 2, 4 and 8 have had the ambiguity properties similar to that of the linearly modulated signals. This means that the designed signals have had small Doppler-leakages and autocorrelation functions have been sharp and narrow. These properties are desired because they help radar system to identify possible targets.

Finally, by examining frequency spectrum graphs (i.e., Figures 5, 10, and 19) we have shown that the designed signals by Algorithms 2, 4, and 8 have efficiently used available frequency bands while constrained bands have been left unused. The designed waveforms have had radiation magnitude difference of 10 to 50 dB between allowed and constrained bands which can be considered adequate for radar systems.

6 References

[1] Proakis, G. John, *Digital Communications (Chapter 4-1), 4th edition.* McGraw-Hill, 2001.

[2] S. W. Golomb, *Shift Register Sequences.* San Francisco: Holden-Day, 1967.

[3] L. R. Welch, "Lower bounds on the maximum cross correlation of signals," *IEEE Trans. Inf. Theory*, vol. 20, no. 3, pp. 397–399, 1974.

[4] M. R. Bell, "Information theory and radar waveform design," *IEEE Trans. Inf. Theory*, vol. 39, no. 5, pp. 1578–1597, 1993.

[5] N. Levanon and E. Mozeson, *Radar Signals.* Hoboken, NJ: Wiley, 2004.

[6] H. Hao, P. Stoica, and J. Li, "Designing unimodular sequences sets with good correlations—Including an application to MIMO radar," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4391–4405, 2009.

[7] P. Stoica, H. He, and J. Li, "New algorithms for designing unimodular sequences with good correlation properties," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1415–1425, 2009.

[8] A. Aubry, V. Carotenuto and A. De Maio, "Forcing multiple spectral compatibility constraints in radar waveforms," *IEEE Signal Processing Letters*, vol. 23, no. 4, pp. 483–487, 2016.

[9] A. Aubry, A. De Maio, M. Piezzo and A. Farina, "Radar waveform design in a spectrally crowded environment via nonconvex quadratic optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 2, pp. 1138–1152, 2014.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[11] J. Mairal, "Stochastic majorization-minimization algorithms for large-scale optimization," 2013.

[12] J. Mairal, "Optimization with first-order surrogate functions," 2013.

[13] Y. Nesterov, *Introductory Lectures on Convex Optimization, A Basic Course.* Springer US, 2004.

[14] M. Yli-Niemi and S. A. Vorobyov, "Computationally efficient waveform design in spectrally dense environment," *IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 277–281, 2018.

[15] J. Haboba, R. Rovatti, G. Setti, "Determination of the integrated sidelobe level of sets of rotated legendre sequences," 2011.

7 Appendices

7.1 Appendix A

One **c**-variable update step is illustrated in Figure 24.

Steps in Figure 24:

(0) *Beginning of the iteration update:*

Initially we have $\mathbf{c}_k$ with length $\|\mathbf{c}_k\|_2^2 = 1$ and angle $\boldsymbol{\phi}_k$. This can be outside of the feasible region $\Theta = \{\mathbf{c} \in \mathbb{R}^{2N} \mid \|\mathbf{c}\|^2 = 1 \text{ and } \|\mathbf{c} - \mathbf{c}_0\|^2 \leq \epsilon, \text{ for some } \mathbf{c}_0 \in \mathbb{R}^{2N}\}$ (initial guess) or inside the feasible region (at least one iteration done or initial guess is inside feasible region).

(1) *Restoring primal optimal $\boldsymbol{c}^*$ from dual optimal $\boldsymbol{\lambda}^*$:*

Primal optimal is restored by identity $\mathbf{c}^* = \min_{\mathbf{c}} L_\rho(\mathbf{c}, \boldsymbol{\lambda}^*)$. We have $f(\mathbf{c}^*) \leq f(\mathbf{c})$, where f is the objective. Minimization problem (3.2.5) is solved with MM-method which yields iteration update $\hat{\mathbf{c}}_{k+1} = \mathbf{c}_k - \frac{1}{L}\left(\left(\mathbf{Q} + \mathbf{Q}^T\right)\mathbf{c}_k + (\boldsymbol{\lambda} - \rho\mathbf{z})\right)$. This update both scales and rotates vector $\mathbf{c}_k$, which yields better length $\|\mathbf{c}\|_2$ and angle $\boldsymbol{\phi}_{k+1}$. Thus, $\hat{\mathbf{c}}_{k+1}^T\mathbf{Q}\hat{\mathbf{c}}_{k+1} \leq \mathbf{c}_k^T\mathbf{Q}\mathbf{c}_k$.

(2) *Projecting $\hat{\boldsymbol{c}}_{k+1}$ back to region $\|\boldsymbol{c}\|_2^2 = 1$:*

$\tilde{\mathbf{c}}_{k+1} = \hat{\mathbf{c}}_{k+1}/\|\hat{\mathbf{c}}_{k+1}\|_2^2$. Still $\tilde{\mathbf{c}}_{k+1}^T\mathbf{Q}\tilde{\mathbf{c}}_{k+1} \leq \mathbf{c}_k^T\mathbf{Q}\mathbf{c}_k$, since angle $\boldsymbol{\phi}_{k+1}$ is better than $\boldsymbol{\phi}_k$.

(3) *Rotation of $\tilde{\boldsymbol{c}}_{k+1}$ to region $\|\boldsymbol{c} - \boldsymbol{c}_0\|_2^2 \leq \epsilon$:*

Vector $\tilde{\mathbf{c}}_{k+1}$ is rotated to region $\|\mathbf{c} - \mathbf{c}_0\|_2^2 \leq \epsilon$ with Gram-Schmidt process if its outside feasible region. If $\boldsymbol{\phi}^*$ is outside the region $\|\mathbf{c}(\boldsymbol{\phi}) - \mathbf{c}_0(\boldsymbol{\phi})\|_2^2 \leq \epsilon$, rotation yields worse $\boldsymbol{\phi}$ than $\boldsymbol{\phi}'_{k+1}$, but since $\nabla_{\boldsymbol{\phi}} f(\boldsymbol{\phi}^*) = 0$ and $\nabla_{\boldsymbol{\phi}\boldsymbol{\phi}} f(\boldsymbol{\phi}^*) \succeq \mathbf{0}$ (Hessian), where $\boldsymbol{\phi}^*$ is the angle for which $f(\boldsymbol{\phi})$ is minimized, $\nabla_{\boldsymbol{\phi}} g(\boldsymbol{\phi}) = \mu\left(\mathbf{c}(\boldsymbol{\phi}^*) - \frac{\langle \mathbf{c}(\boldsymbol{\phi}^*), \mathbf{c}(\boldsymbol{\phi})\rangle}{\langle \mathbf{c}(\boldsymbol{\phi}), \mathbf{c}(\boldsymbol{\phi})\rangle}\mathbf{c}(\boldsymbol{\phi})\right)$, $\mu \leq 0$, the best feasible angle $\boldsymbol{\phi}$ is at the boundary of the feasible region and hence cannot be worse than $\boldsymbol{\phi}_k$ if $\boldsymbol{\phi}_k$ is in the feasible region.

(4) *$\boldsymbol{c}_{k+1}$ has been found and it complies with constraints $\|\boldsymbol{c}\|_2^2 = 1$ and $\|\boldsymbol{c} - \boldsymbol{c}_0\|_2^2 \leq \epsilon$.*

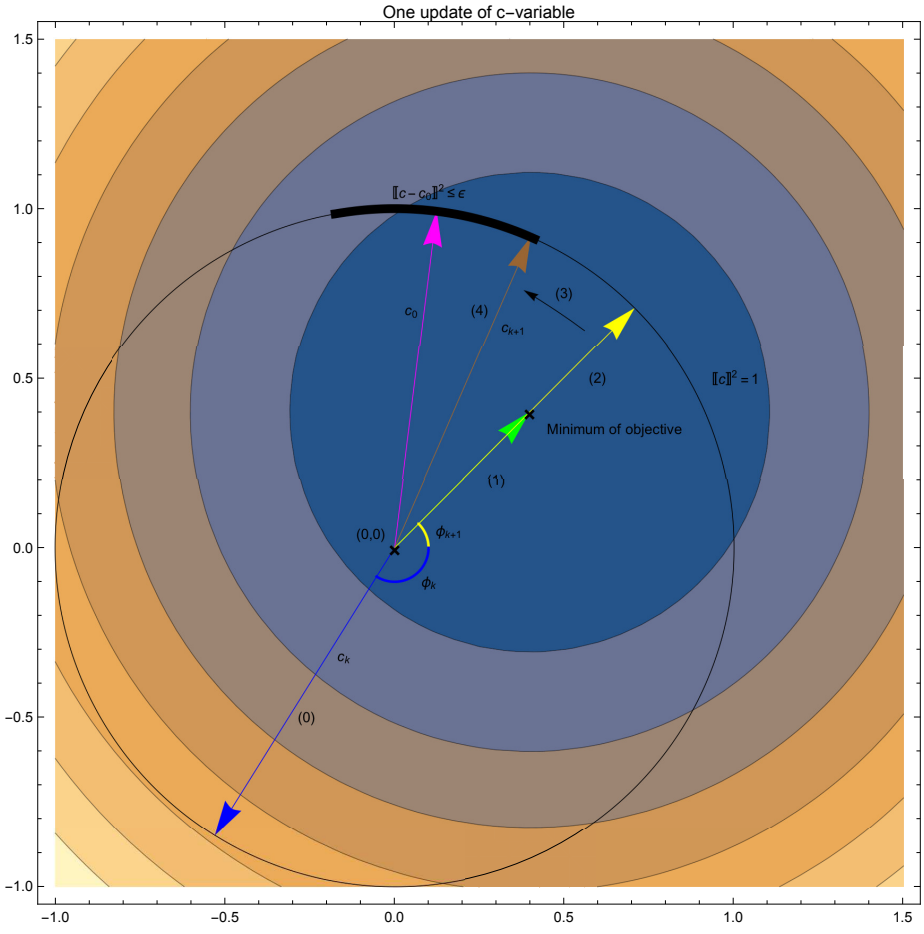

Figure 24: **c**-variable update

7.2 Appendix B

Matrix $\mathbf{R}_I = \begin{bmatrix} \mathrm{Re}\{\hat{\mathbf{R}}_I\} & -\mathrm{Im}\{\hat{\mathbf{R}}_I\} \\ \mathrm{Im}\{\hat{\mathbf{R}}_I\} & \mathrm{Re}\{\hat{\mathbf{R}}_I\} \end{bmatrix}$, where $\hat{\mathbf{R}}_I \in \mathbb{C}^{N\times N}$ is defined as in (3.1.8) and (3.1.11). For this construction, we have $\left(\mathrm{Re}\{\hat{\mathbf{R}}_I\}\right)^T = \mathrm{Re}\{\hat{\mathbf{R}}_I\}$ and $\left(-\mathrm{Im}\{\hat{\mathbf{R}}_I\}\right)^T = \mathrm{Im}\{\hat{\mathbf{R}}_I\}$.

Block matrix transpose identity is given as:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^T = \begin{bmatrix} \mathbf{A}^T & \mathbf{C}^T \\ \mathbf{B}^T & \mathbf{D}^T \end{bmatrix}$$

which yields

$$\mathbf{R}_I^T = \begin{bmatrix} (\mathrm{Re}\{\hat{\mathbf{R}}_I\})^T & (\mathrm{Im}\{\hat{\mathbf{R}}_I\})^T \\ (-\mathrm{Im}\{\hat{\mathbf{R}}_I\})^T & (\mathrm{Re}\{\hat{\mathbf{R}}_I\})^T \end{bmatrix} = \begin{bmatrix} \mathrm{Re}\{\hat{\mathbf{R}}_I\} & -\mathrm{Im}\{\hat{\mathbf{R}}_I\} \\ \mathrm{Im}\{\hat{\mathbf{R}}_I\} & \mathrm{Re}\{\hat{\mathbf{R}}_I\} \end{bmatrix} = \mathbf{R}_I.$$

Hence $\mathbf{R}_I^T = \mathbf{R}_I$, that is $\mathbf{R}_I$ is symmetric.

For symmetric $\mathbf{R}_I$, identity $((\mathbf{I} + \gamma_{k+1}\mathbf{R}_I)^{-1})^T = (\mathbf{I} + \gamma_{k+1}\mathbf{R}_I)^{-1}$ holds $\forall \gamma \in \mathbb{R}$, because

$$((\mathbf{I} + \gamma_{k+1}\mathbf{R}_I)^{-1})^T = \left(\mathbf{I} + \gamma_{k+1}\mathbf{R}_I^T\right)^{-1} = (\mathbf{I} + \gamma_{k+1}\mathbf{R}_I)^{-1}.$$

7.3 Appendix C

In Subsection 4.4 we introduced the PAPR-constraint which essentially yielded an additional constraint of a form:

$$\left|\mathbf{c}^{(p)}(n)\right|^2 \leq \frac{\eta}{N}, \forall n \in \{1,2,...,N\}. \tag{7.3.1}$$

In the simulation setup this constraint did not seem to contribute to the waveform design. We were not able to set up simulation in such a way that modifying the η parameter (i.e., suppressing the PAPR-constraint) would affect the waveform design. If η was chosen to be small, we ran out of feasible region.

To avoid this problem we can reformulate the constraint (7.3.1) to following forms:

$$\begin{cases} \left|\mathbf{c}^{(p)}(n)\right| \leq 1 + \frac{\eta}{N}, \forall n \in \{1,2,...,N\}, & \text{(7.3.2a)} \\ \left|\mathbf{c}^{(p)}(n)\right| \geq 1 - \frac{\eta}{N}, \forall n \in \{1,2,...,N\}, & \text{(7.3.2b)} \end{cases}$$

or

$$\begin{cases} \left|\mathbf{c}^{(p)}(n)\right| \leq \frac{1+\eta}{\sqrt{N}}, \forall n \in \{1,2,...,N\}, & \text{(7.3.3a)} \\ \left|\mathbf{c}^{(p)}(n)\right| \geq \frac{1-\eta}{\sqrt{N}}, \forall n \in \{1,2,...,N\}. & \text{(7.3.3b)} \end{cases}$$

In (7.3.2a)-(7.3.2b) we require all elements of the designed waveforms to be close to 1. It is worth noting that with this type of constraints we cannot use sinusoidal reference codes, because otherwise the similarity constraint (4.4.5e) would not be attainable. This means we would run out of feasible region in optimization.

By using constraints (7.3.2a)-(7.3.2b) instead of (7.3.1) we get a minimization problem:

$$\mathcal{P}_6 : \begin{cases} \min\limits_{c} & \sum\limits_{p=1}^{P} (\mathbf{c}^{(p)})^T \mathbf{Q} \mathbf{c}^{(p)} & \text{(7.3.4a)} \\ \text{s.t. :} & \|\mathbf{c}^{(p)}\|^2 = 1, \quad \forall p \in \{1,...,P\} & \text{(7.3.4b)} \\ & \left|\mathbf{c}^{(p)}(n)\right| \leq 1 + \frac{\eta}{N}, n \in \{1,...,N\}, \forall p & \text{(7.3.4c)} \\ & -\left|\mathbf{c}^{(p)}(n)\right| \leq \frac{\eta}{N} - 1, n \in \{1,...,N\}, \forall p & \text{(7.3.4d)} \\ & (\mathbf{c}^{(p)})^T \mathbf{R}_I \mathbf{c}^{(p)} \leq E_I, \quad \forall p \in \{1,...,P\} & \text{(7.3.4e)} \\ & \left\|\mathbf{c}^{(p)} - \mathbf{c}_0^{(p)}\right\|^2 \leq \epsilon, \quad \forall p \in \{1,...,P\} & \text{(7.3.4f)} \\ & SL2(\mathbf{c}^{(p)}) \leq \delta, \quad \forall p \in \{1,...,P\}. & \text{(7.3.4g)} \end{cases}$$

Then again, by using constraints (7.3.2a)-(7.3.2b) instead of (7.3.1) we get a minimization problem:

$$\mathcal{P}_7 : \begin{cases} \min\limits_{c} & \sum\limits_{p=1}^{P} (\mathbf{c}^{(p)})^T \mathbf{Q} \mathbf{c}^{(p)} & (7.3.5a) \\ \text{s.t. :} & \|\mathbf{c}^{(p)}\|^2 = 1, \quad \forall p \in \{1, ..., P\} & (7.3.5b) \\ & \left|\mathbf{c}^{(p)}(n)\right| \leq \frac{1+\eta}{\sqrt{N}}, \forall n \in \{1, 2, ..., N\}, \forall p & (7.3.5c) \\ & -\left|\mathbf{c}^{(p)}(n)\right| \leq \frac{\eta-1}{\sqrt{N}}, n \in \{1, ..., N\}, \forall p & (7.3.5d) \\ & (\mathbf{c}^{(p)})^T \mathbf{R}_I \mathbf{c}^{(p)} \leq E_I, \quad \forall p \in \{1, ..., P\} & (7.3.5e) \\ & \left\|\mathbf{c}^{(p)} - \mathbf{c}_0^{(p)}\right\|^2 \leq \epsilon, \quad \forall p \in \{1, ..., P\} & (7.3.5f) \\ & SL2(\mathbf{c}^{(p)}) \leq \delta, \quad \forall p \in \{1, ..., P\}. & (7.3.5g) \end{cases}$$

The constraints (7.3.4c)-(7.3.4d) and (7.3.5c)-(7.3.5d) are handled in the $\mathbf{c}$-variable update similarly as the original PAPR-constraint in 3^{rd} multiple waveform algorithm.

In case of the minimization problem $\mathcal{P}_6$, we have a following minimization problem:

$$\begin{aligned} &\mathbf{c}_{k+1}^{(i)} = \arg\min_{\mathbf{c}^{(i)}} (\mathbf{c}^{(i)})^T \mathbf{Q} \mathbf{c}^{(i)} + (\boldsymbol{\lambda} - \rho \mathbf{z})^T \mathbf{c}^{(i)} \\ &\text{s.t. } \|\mathbf{c}^{(i)}\|^2 = 1, \ \left\|\mathbf{c}^{(i)} - \mathbf{c}_0^{(i)}\right\|^2 \leq \epsilon, \left|\mathbf{c}^{(i)}(n)\right| \leq 1 + \frac{\eta}{N}, \\ &-\left|\mathbf{c}^{(i)}(n)\right| \leq \frac{\eta}{N} - 1, n \in \{1, ..., N\}, \end{aligned} \tag{7.3.6}$$

and in case of the minimization problem $\mathcal{P}_7$, we have a following minimization problem:

$$\begin{aligned} &\mathbf{c}_{k+1}^{(i)} = \arg\min_{\mathbf{c}^{(i)}} (\mathbf{c}^{(i)})^T \mathbf{Q} \mathbf{c}^{(i)} + (\boldsymbol{\lambda} - \rho \mathbf{z})^T \mathbf{c}^{(i)} \\ &\text{s.t. } \|\mathbf{c}^{(i)}\|^2 = 1, \ \left\|\mathbf{c}^{(i)} - \mathbf{c}_0^{(i)}\right\|^2 \leq \epsilon, \left|\mathbf{c}^{(i)}(n)\right| \leq \frac{1+\eta}{\sqrt{N}}, \\ &-\left|\mathbf{c}^{(i)}(n)\right| \leq \frac{\eta-1}{\sqrt{N}}, n \in \{1, ..., N\}, \end{aligned} \tag{7.3.7}$$

The Lagrangians $\mathcal{L}_{\mathcal{P}_6}\left(\mathbf{c}^{(i)}, \boldsymbol{\xi}, \boldsymbol{\zeta}\right)$ and $\mathcal{L}_{\mathcal{P}_7}\left(\mathbf{c}^{(i)}, \boldsymbol{\xi}, \boldsymbol{\zeta}\right)$ for (7.3.6) and (7.3.7)

by using the constraints (7.3.4c)-(7.3.4d) and (7.3.5c)-(7.3.5d), respectively:

$$\mathcal{L}_{\mathcal{P}_6}\left(\mathbf{c}^{(i)},\boldsymbol{\xi},\boldsymbol{\zeta}\right)=\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}+(\boldsymbol{\lambda}-\rho z)^T\mathbf{c}^{(i)}+\sum_{n=1}^{N}\xi_n\left(\left|\mathbf{c}^{(i)}(n)\right|-\left(1+\frac{\eta}{N}\right)\right)$$
$$-\sum_{n=1}^{N}\zeta_n\left(\left|\mathbf{c}^{(i)}(n)\right|-\left(1-\frac{\eta}{N}\right)\right), \tag{7.3.8}$$

$$\mathcal{L}_{\mathcal{P}_7}\left(\mathbf{c}^{(i)},\boldsymbol{\xi},\boldsymbol{\zeta}\right)=\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}+(\boldsymbol{\lambda}-\rho z)^T\mathbf{c}^{(i)}+\sum_{n=1}^{N}\xi_n\left(\left|\mathbf{c}^{(i)}(n)\right|-\left(\frac{1+\eta}{\sqrt{N}}\right)\right)$$
$$-\sum_{n=1}^{N}\zeta_n\left(\left|\mathbf{c}^{(i)}(n)\right|-\left(\frac{1-\eta}{\sqrt{N}}\right)\right), \tag{7.3.9}$$

where $\boldsymbol{\xi}=(\xi_1,\xi_2,...,\xi_N)$ and $\boldsymbol{\zeta}=(\zeta_1,\zeta_2,...,\zeta_N)$ are Lagrange multipliers. The dual functions $g_{\mathcal{P}_6}(\boldsymbol{\xi},\boldsymbol{\zeta})$ and $g_{\mathcal{P}_7}(\boldsymbol{\xi},\boldsymbol{\zeta})$ are given as:

$$\begin{aligned}
g_{\mathcal{P}_6}(\boldsymbol{\xi},\boldsymbol{\zeta}) &= \inf_{\mathbf{c}^{(i)}} L\left(\mathbf{c}^{(i)},\boldsymbol{\xi},\boldsymbol{\zeta}\right)\\
&= \inf_{\mathbf{c}^{(i)}}\left\{\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}+(\boldsymbol{\lambda}-\rho z)^T\mathbf{c}^{(i)}+\sum_{n=1}^{N}\xi_n\left(\left|\mathbf{c}^{(i)}(n)\right|-\left(1+\frac{\eta}{N}\right)\right)\right.\\
&\left.-\sum_{n=1}^{N}\zeta_n\left(\left|\mathbf{c}^{(i)}(n)\right|-\left(1-\frac{\eta}{N}\right)\right)\right\}\\
&= \inf_{\mathbf{c}^{(i)}}\left\{\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}+(\boldsymbol{\lambda}-\rho\mathbf{z})^T\mathbf{c}^{(i)}+\sum_{n=1}^{N}(\xi_n-\zeta_n)\left|\mathbf{c}^{(i)}(n)\right|\right.\\
&\left.+\sum_{n=1}^{N}\left[\zeta_n\left(1-\frac{\eta}{N}\right)-\xi_n\left(1+\frac{\eta}{N}\right)\right]\right\}\\
&= \inf_{\mathbf{c}^{(i)}}\left\{\left(\mathbf{c}^{(i)}\right)^T\mathbf{Q}\mathbf{c}^{(i)}+(\boldsymbol{\lambda}-\rho\mathbf{z})^T\mathbf{c}^{(i)}+\sum_{n=1}^{N}(\xi_n-\zeta_n)\left|\mathbf{c}^{(i)}(n)\right|\right\}\\
&+\sum_{n=1}^{N}\left[\zeta_n\left(1-\frac{\eta}{N}\right)-\xi_n\left(1+\frac{\eta}{N}\right)\right]\\
&\overset{*}{=}\left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T\mathbf{Q}\mathbf{c}_{\text{opt}}^{(i)}+(\boldsymbol{\lambda}-\rho\mathbf{z})^T\mathbf{c}_{\text{opt}}^{(i)}+\sum_{n=1}^{N}(\xi_n-\zeta_n)\left|\mathbf{c}_{\text{opt}}^{(i)}(n)\right|\\
&+\sum_{n=1}^{N}\left[\zeta_n\left(1-\frac{\eta}{N}\right)-\xi_n\left(1+\frac{\eta}{N}\right)\right],
\end{aligned} \tag{7.3.10}$$

$$
\begin{aligned}
g_{\mathcal{P}_7}(\boldsymbol{\xi},\boldsymbol{\zeta}) &= \inf_{\mathbf{c}^{(i)}} L\left(\mathbf{c}^{(i)},\boldsymbol{\xi},\boldsymbol{\zeta}\right) \\
&\overset{*}{=} \left(\mathbf{c}_{\text{opt}}^{(i)}\right)^T \mathbf{Q}\mathbf{c}_{\text{opt}}^{(i)} + (\boldsymbol{\lambda}-\rho\mathbf{z})^T\mathbf{c}_{\text{opt}}^{(i)} + \sum_{n=1}^{N}(\xi_n-\zeta_n)\left|\mathbf{c}_{\text{opt}}^{(i)}(n)\right| \\
&+ \sum_{n=1}^{N}\left[\zeta_n\left(\frac{1-\eta}{\sqrt{N}}\right) - \xi_n\left(\frac{1+\eta}{\sqrt{N}}\right)\right]. \qquad (7.3.11)
\end{aligned}
$$

(*) $\mathbf{c}_{\text{opt}}^{(i)}$ obtained by the first step in dual ascent (primal problem minimization).

Denote $\Omega^{(i)} = \left\{\mathbf{c}^{(i)} \in \mathbb{R}^{2N} \mid \|\mathbf{c}^{(i)}\|^2 = 1 \text{ and } \|\mathbf{c}^{(i)} - \mathbf{c}_0^{(i)}\|^2 \le \epsilon\right\}$. By using Dual ascent to problem

$$
\begin{cases}
\min\limits_{\mathbf{c}} & (\mathbf{c}^{(i)})^T\mathbf{Q}\mathbf{c}^{(i)} + (\boldsymbol{\lambda}-\rho\mathbf{z})^T\mathbf{c}^{(i)} & (7.3.12a)\\
\text{s.t.:} & \left|\mathbf{c}^{(i)}(n)\right| \le 1 + \frac{\eta}{N}, n \in \{1,...,N\}, & (7.3.12b)\\
& -\left|\mathbf{c}^{(i)}(n)\right| \le \frac{\eta}{N} - 1, n \in \{1,...,N\} & (7.3.12c)
\end{cases}
$$

or

$$
\begin{cases}
\min\limits_{\mathbf{c}} & (\mathbf{c}^{(i)})^T\mathbf{Q}\mathbf{c}^{(i)} + (\boldsymbol{\lambda}-\rho\mathbf{z})^T\mathbf{c}^{(i)} & (7.3.13a)\\
\text{s.t.:} & \left|\mathbf{c}^{(i)}(n)\right| \le \frac{1+\eta}{\sqrt{N}}, n \in \{1,...,N\}, & (7.3.13b)\\
& -\left|\mathbf{c}^{(i)}(n)\right| \le \frac{\eta-1}{\sqrt{N}}, n \in \{1,...,N\} & (7.3.13c)
\end{cases}
$$

we get routines:

$$
\begin{cases}
\hat{\mathbf{c}}_{k+1}^{(i)} = \mathbf{c}_k^{(i)} - \dfrac{1}{L}\nabla_{\mathbf{c}^{(i)}}\mathcal{L}_{\mathcal{P}_{\{6,7\}}}\left(\mathbf{c}_k^{(i)},\boldsymbol{\xi}_k,\boldsymbol{\zeta}_k\right) & (7.3.14a)\\
\text{Project } \hat{\mathbf{c}}_{k+1}^{(i)} \text{ to region } \Omega^{(i)} \text{ to obtain } \mathbf{c}_{k+1}^{(i)} & (7.3.14b)\\
\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \dfrac{1}{\alpha}\nabla_{\boldsymbol{\xi}} g_{\mathcal{P}_{\{6,7\}}}(\boldsymbol{\xi}_k,\boldsymbol{\zeta}_k) & (7.3.14c)\\
\boldsymbol{\zeta}_{k+1} = \boldsymbol{\zeta}_k + \dfrac{1}{\beta}\nabla_{\boldsymbol{\zeta}} g_{\mathcal{P}_{\{6,7\}}}(\boldsymbol{\xi}_k,\boldsymbol{\zeta}_k). & (7.3.14d)
\end{cases}
$$

Gradients in (7.3.14) are given as:

$$\nabla_{\mathbf{c}^{(i)}} \mathcal{L}_{\mathcal{P}_6} \left(\mathbf{c}_k^{(i)}, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k \right) = 2\mathbf{Q}\mathbf{c}^{(i)} + (\boldsymbol{\lambda} - \rho z)^T + \boldsymbol{\xi} - \boldsymbol{\zeta}, \tag{7.3.15}$$

$$\nabla_{\mathbf{c}^{(i)}} \mathcal{L}_{\mathcal{P}_7} \left(\mathbf{c}_k^{(i)}, \boldsymbol{\xi}_k, \boldsymbol{\zeta}_k \right) = 2\mathbf{Q}\mathbf{c}^{(i)} + (\boldsymbol{\lambda} - \rho z)^T + \boldsymbol{\xi} - \boldsymbol{\zeta}, \tag{7.3.16}$$

$$\nabla_{\boldsymbol{\xi}} g_{\mathcal{P}_6}(\boldsymbol{\xi}, \boldsymbol{\zeta}) = \mathbf{c}^{(i)} - \left(1 + \frac{\eta}{N} \right), \tag{7.3.17}$$

$$\nabla_{\boldsymbol{\zeta}} g_{\mathcal{P}_6}(\boldsymbol{\xi}, \boldsymbol{\zeta}) = -\mathbf{c}^{(i)} + \left(1 - \frac{\eta}{N} \right), \tag{7.3.18}$$

$$\nabla_{\boldsymbol{\xi}} g_{\mathcal{P}_7}(\boldsymbol{\xi}, \boldsymbol{\zeta}) = \mathbf{c}^{(i)} - \frac{1+\eta}{N}, \tag{7.3.19}$$

$$\nabla_{\boldsymbol{\zeta}} g_{\mathcal{P}_7}(\boldsymbol{\xi}, \boldsymbol{\zeta}) = -\mathbf{c}^{(i)} + \frac{1-\eta}{N}. \tag{7.3.20}$$