Computationally Efficient Waveform Design in Spectrally Dense Environment

Markus Yli-Niemi & Sergiy A. Vorobyov
markus.yli-niemi@aalto.fi
sergiy.vorobyov@aalto.fi

July 5, 2018
Introduction

- Recently in radar systems waveform design in spectrally dense environment [1] has aroused noticeable interest
- Solution methods exist for the problem (see e.g. [2], [3]) but they are computationally inefficient
- When radar system operates at GHz level radar code dimension becomes large, need for computationally efficient solution methods
- Here we develop new computationally efficient method to design transmitter waveform in spectrally dense environment
Problem formulation

- Similarly to [3], denote transmitted fast-time radar code vector by $c$ and fast-time observation signal by $v$:

$$c = (c[1], c[2], \ldots, c[N])^T, \quad v = \alpha c + n, \quad c, v \in \mathbb{C}^N, \alpha \in \mathbb{C} \quad (1)$$

- Matched filtering $v$ with filter $h \in \mathbb{C}^N$ yields $y = h^H v$. Write $y = y_s + y_n$, where $y_s = \alpha h^H c$ and $y_n = h^H n$. SINR is given as:

$$
\text{SINR} = \frac{|y_s|^2}{|y_n|^2} = \frac{|\alpha|^2 |h^H c|^2}{|h^H n|^2} = \frac{|\alpha|^2 |h^H c|^2}{h^H n n^H h} = \text{M} \quad (2)
$$

- To maximize SINR w.r.t. $h$, we choose $h = M^{-1} c$, which yields $\text{SINR} = |\alpha|^2 c^H M^{-1} c$
Problem formulation

- Introduce constrained bandwidths $\{\Omega_k\}_{k \in \{1,2,...,K\}}$, where $\Omega_k = [f_1^k, f_2^k]$. The energy $c$ radiates to constrained bandwidths is (see e.g. [3]):

$$
\sum_{k=1}^{K} w_k \int_{\Omega_k} |\mathcal{F}_N\{c\}|^2 df = c^H R_l c, \quad (3)
$$

where $\{w_k\}_{k=1}^{K}$ are non-negative weights, $\mathcal{F}_N\{c\}$ stands for the discrete-time Fourier transform of $c$ given as

$$
\mathcal{F}_N\{c\} \triangleq \sum_{k=1}^{N} c[k] e^{-j2\pi kf}, \quad \text{and} \quad R_l \triangleq \sum_{k=1}^{K} w_k R_l^k \quad \text{with}
$$

$$
[R_l^k]_{m,l} = \left( e^{i2\pi f_2^k (m-l)} - e^{i2\pi f_1^k (m-l)} \right) / e^{i2\pi (m-l)}, \quad \text{if} \ m \neq l, \quad \text{and}
$$

$$
[R_l^k]_{m,l} = f_2^k - f_1^k, \quad \text{if} \ m = l.
$$
Problem formulation

- If radar code energy $\|c\|^2$ is unit constrained and required to be in similarity region with reference code $c_0$ alongside radiation energy constraint $c^H R_I c \leq E_I$, SINR maximization problem can be written:

\[
\mathcal{P}_1 : \begin{cases} 
\max_{c} \quad |\alpha|^2 c^H M^{-1} c \\
\text{s.t. :} \quad \|c\|^2 = 1 \\
\quad \quad \quad \quad \quad \quad c^H R_I c \leq E_I \\
\quad \quad \quad \quad \quad \quad \|c - c_0\|^2 \leq \epsilon
\end{cases}
\]
Problem formulation

$\mathcal{P}_1$ is equal to:

$$\mathcal{P}_1^{(1)} : \begin{cases}
\min_{c} & -c^H R c \\
\text{s.t.} : & \|c\|^2 = 1 \\
 & c^H R_I c \leq E_I \\
 & \|c - c_0\|^2 \leq \epsilon
\end{cases}$$

where $c, c_0 \in \mathbb{C}^N$ and $R_I, R = M^{-1} \in \mathbb{C}^{N \times N}$
Majorization-Minimization step

- Due to independence of real and imaginary components we can write $c, c_0, R$ and $R_I$ as:

$$R = \begin{bmatrix} \text{Re}\{R\} & -\text{Im}\{R\} \\ \text{Im}\{R\} & \text{Re}\{R\} \end{bmatrix}, \quad c = \begin{bmatrix} \text{Re}\{c\} \\ \text{Im}\{c\} \end{bmatrix} \quad \text{and} \quad c_0 = \begin{bmatrix} \text{Re}\{c_0\} \\ \text{Im}\{c_0\} \end{bmatrix}. $$

- Let us use surrogate $Q = \mu I - R \succeq 0, \mu > 0$ to upper-bound objective. We get real-valued optimization problem $P_2$:

$$P_2 : \begin{cases} \min_c c^T Q c \quad \text{(6a)} \\ \text{s.t. :} \\ \|c\|^2 = 1 \quad \text{(6b)} \\ c^T R_I c \leq E_I \quad \text{(6c)} \\ \|c - c_0\| \leq \epsilon \quad \text{(6d)} \end{cases}$$

where $c, c_0 \in \mathbb{R}^{2N}$ and $Q, R_I \in \mathbb{R}^{2N \times 2N}$. 
Apply ADMM to \( P_2 \)

To allow separability of \( c^T Qc \), let us introduce slack variable \( z \) with constraint \( c = z \). Augmented Lagrangian \( L_\rho(c, z, \lambda) \) for minimization problem \( \min_c c^T Qc \) s.t.: \( c = z \):

\[
L_\rho(c, z, \lambda) = c^T Qc + \lambda^T(c - z) + \frac{\rho}{2}||c - z||^2. \tag{7}
\]

ADMM-steps for \( P_2 \):

\[
\begin{align*}
\mathbf{c}_{k+1} &= \arg \min \limits_c L_\rho(c, z_k, \lambda_k) \quad \tag{8a} \\
\mathbf{z}_{k+1} &= \arg \min \limits_z L_\rho(c_{k+1}, z, \lambda_k) \quad \tag{8b} \\
\lambda_{k+1} &= \lambda_k + \rho \left( \mathbf{c}_{k+1} - \mathbf{z}_{k+1} \right), \tag{8c}
\end{align*}
\]

Next \( c \)-variable update and \( z \)-variable update are solved.
**c-variable update**

- **c-variable update (8a)** can be written as:

\[
c_{k+1} = \arg \min_{c} L_{\rho}(c, z_{k}, \lambda_{k}) = \arg \min_{c} \left\{ c^{T}Qc + (\lambda - \rho z)^{T}c \right\}
\]

\[
= \arg \min_{c} h(c) \quad \text{s.t.} \quad \|c\|^{2} = 1, \quad \|c - c_{0}\|^{2} \leq \epsilon.
\]

- Objective function \( h(c) \) is continuously differentiable and \( \nabla_{c}h \) is \( L \)-Lipschitz continuous. To minimize \( h(c) \) we use gradient descent:

\[
c_{k+1} = c_{k} - \frac{1}{L} \left( (Q + Q^{T}) c_{k} + (\lambda - \rho z) \right),
\]

where Lipschitz constant can be found by noticing:

\[
|\nabla_{c}h(\kappa) - \nabla_{c}h(c)| = \left| \left( Q + Q^{T} \right) (\kappa - c) \right| \leq L |\kappa - c|
\]

\[
\Rightarrow \left| \sum_{p=1}^{2N} \left( Q[i,p] + Q^{T}[i,p] \right) \right| \leq L, \forall i = 1, \ldots, 2N
\]
\( c \)-variable update

- Gradient descent yields updated \( c \) that has \( \|c\|_2^2 \neq 1 \) and possibly \( \|c - c_0\| \geq \epsilon \).
- Denote \( \Theta = \{ c \in \mathbb{R}^{2N} \mid \|c\|^2 = 1 \text{ and } \|c - c_0\|^2 \leq \epsilon, \text{ for some } c_0 \in \mathbb{R}^{2N} \} \)
- Cheap way to project \( c \) back to unitary region is to divide updated \( c \) by its \( L^2 \)-norm:

\[
\hat{c}_{k+1} = c_{k+1}/\|c_{k+1}\| \quad (11)
\]
Next $\hat{c}_{k+1}$ is rotated to region $\Theta$ with steps introduced in Algorithm 1.

**Algorithm 1:** Rotate $c$ toward $c_0$ as long as region $\|c - c_0\| \leq \epsilon$ is reached

1. function RotateVector($c, c_0, \alpha', \epsilon$);
2. Input : $c, c_0, \alpha'$ and $\epsilon$
3. Output : $c$
4. while $\|c - c_0\| > \epsilon$ do
5.     $\tilde{c} = c_0 - \text{proj}_c(c_0) = c_0 - \frac{c_0^H c}{\|c\|^2} c$;
6.     $e = \frac{\tilde{c}}{\|\tilde{c}\|}, \quad c^* = c + \alpha' e$,
7.     $c = \frac{c^*}{\|c^*\|};$
8. end

[$c$]$_2$ = 1
[$[c-c_0]$]$_2 \leq \epsilon$
$c_0$
$c$
$c^*$
$c_0$
Rotation of $c$ toward $c_0$
$\|c - c_0\|^2 \leq \epsilon$
**c-variable update**

The combination of steps (10), (11) and Algorithm 1 can be shown to be solution steps to projected gradient step for problem $\min_c h(c)$ subject to $c \in \Theta$:

$$
\begin{cases}
  y_{k+1} = c_k - \frac{1}{L} \nabla h(c_k) \\
  c_{k+1} = \min_{c \in \Theta} \| y_{k+1} - c \|.
\end{cases}
$$

By using angular coordinates $\phi \in \mathbb{R}^{2N-1}$ step (12b) can be written as:

$$
\begin{cases}
  \phi_{k+1} = \arg \min_{\phi \in \Omega} \| \phi^* - \phi \| \\
  c_{k+1} = c(\phi_{k+1}).
\end{cases}
$$

where $\Omega = \{ \phi \in \mathbb{R}^{2N-1} \mid \| c(\phi) - c_0(\phi) \|^2 \leq \epsilon \}$ and $\phi^* = \arg \min_{\phi} h(c(\phi))$. 
**z-variable update**

- **z-variable update** (8b) can be written as:

  \[
  z_{k+1} = \arg \min_z L_\rho \left( c_{k+1}, z, \lambda_k \right)
  = \arg \min_z \left\{ \lambda^T (c - z) + \frac{\rho}{2} \| c - z \|^2 \right\}
  = \arg \min_z \left\{ \| z - (c + \frac{1}{\rho} \lambda) \|^2 \right\}
  \]
  s.t. \( z^T R_I z \leq E_I \). (14)

- Lagrangian for (14) is given as:

  \[
  L(z, \gamma) = \| z - (c + \frac{1}{\rho} \lambda) \|^2 + \gamma (z^T R_I z - E_I). \] (15)
Karush-Kuhn-Tucker (KKT) conditions for the minimization problem (14):

\[
\begin{aligned}
\nabla_z L(z^*, \gamma^*) &= 0 \\
\gamma^* &\geq 0 \\
\gamma^* ((z^*)^T R_I z^* - E_I) &= 0 \\
(z^T R_I z - E_I) &\leq 0 \\
\nabla_{zz} L(z^*, \gamma^*) &\succeq 0,
\end{aligned}
\]

By (16a) and (16c):

\[
\nabla_z L(z^*, \gamma^*) = 0 \Rightarrow (I + \gamma^* R_I) z^* = c + \frac{1}{\rho} \lambda,
\]

\[
(z^*)^T R_I z^* - E_I = 0,
\]

where \( z^* \) and \( \gamma^* \) denotes critical points of Lagrangian \( L(z, \gamma) \).
**z-variable update**

- Now (17) can be written as iteration step (19):

  \[ z_{k+1} = (I + \gamma_{k+1} R_I)^{-1} \left( c + \frac{1}{\rho} \lambda \right) \]

  \[ = \left( I + \sum_{i=1}^{2N} \frac{\gamma_{k+1} \sigma_i}{1 + \gamma_{k+1} \sigma_i} p_i p_i^T \right) \left( c + \frac{1}{\rho} \lambda \right). \quad (19) \]

- \( \gamma_{k+1} > 0 \) can be found as the solution to (18):

  \[ z_{k+1}^T R_I z_{k+1} = E_I \iff \sum_{i=1}^{2N} \frac{a_i \sigma_i (1 + \gamma \sigma_i)^2}{(1 + \gamma \sigma_i)^2 - E_I} = 0 \quad (20) \]

where \( a_i = (p_i^T (c + \frac{1}{\rho} \lambda))^2 \), \( \sigma_i \) is \( i \)’th eigenvalue and \( p_i \) corresponding eigenvector of \( R_I \). Equation (20) can be efficiently solved by using Newton’s method.
Proposed algorithm

Collect $c$ and $z$-variable updates to get final algorithm:

**Algorithm 2: MM-algorithm**

```plaintext
1 function MM($Q, c_0, R, E_l, \epsilon, K'$);
  Input : $Q = \mu I - R \succeq 0, c_0, R, E_l, \epsilon$ and $K'$
  Output : $c$
2 Initialize $c, z$ and $\lambda$;
3 for $k = 1, k \leq K', k ++$ do
4   $\hat{c}_{k+1} = c_k - \frac{1}{L} \left( (Q + Q^T) c_k + (\lambda - \rho z) \right)$;
5   $\tilde{c}_{k+1} = \frac{\hat{c}_{k+1}}{||\hat{c}_{k+1}||}$;
6   $c_{k+1} = \text{RotateVector}(\tilde{c}_{k+1}, c_0, \alpha, \epsilon)$;
7   Solve $\sum_{i=1}^{2N} a_i\sigma_i - E_l = 0$ for $\gamma_{k+1} > 0$;
8   $z_{k+1} = \left( I + \sum_{i=1}^{2N} \frac{\gamma_{k+1}\sigma_i}{1+\gamma_{k+1}\sigma_i} p_ip_i^T \right) \left( c + \frac{1}{\rho} \lambda \right)$;
9   $\lambda_{k+1} = \lambda_k + \rho(c_{k+1} - z_{k+1})$;
end
```
Time-Complexity graph

![Graph showing time complexity for different algorithms.](image-url)

- **Algorithm 2**: $n^{3.5}$
- **$n^2$**: $n^2$
- **$n$**: $n$
- **$n \log(n)$**: $n \log(n)$

The graph compares the runtime (in seconds) against the problem dimension for different algorithms. The runtime is plotted on a logarithmic scale, ranging from $10^{-5}$ to $10^{15}$ seconds, while the problem dimension is shown on a logarithmic scale from $10^{-5}$ to $10^5$. The algorithms are represented by different lines: Algorithm 2 by a blue line, $n^2$ by a red line, $n$ by a green line, $n \log(n)$ by a pink line, and $n^{3.5}$ by a black line.
Simulation example

- Let us use Algorithm 2 in example environment. Consider radar with bandwidth of 6 GHz to be sampled at sampling frequency of $f_s = 12$ GHz.
- Fast-time radar code has length $T = 1$ µs (i.e. $N = 12000$).
- The radar operates in spectrally busy environment with seven constrained bandwidths
  \[ \{ \Omega_k \}^7_{k=1} = \{ [0.0000, 0.0617], [0.0700, 0.1247], [0.1526, 0.2540], [0.3086, 0.3827], [0.4074, 0.4938], [0.6185, 0.7600], [0.8200, 0.9500] \}. \]
- Covariance matrix is modelled as:
  \[
  M = \sigma_0 I + \sum_{k=1}^{K} \frac{\sigma_{I,k}}{\Delta f_k} R_{I,k} + \sum_{k=1}^{K} \sigma_{J,k} R_{J,k} \tag{21}
  \]
- For reference signal we use linearly modulated signal $c_0 = e^{j2\pi(f_\Delta t + f_0)t}$, with carrier frequency $f_0 = 1.8$ GHz and frequency range $f_\Delta = 3.6$ GHz/µs.
Simulation example

- $\sigma_0 = 0\,\text{dB}$ (thermal noise level)
- $K = 7$ (number of licensed radiators)
- $\sigma_{I,k} = 10\,\text{dB}, \forall k \in \{1, \ldots, K\}$ (energy of coexisting telecom network operating on normalized frequency band $\Omega_k = [f_{k1}^k, f_{k2}^k]$)
- $\Delta f_k = f_{k2}^k - f_{k1}^k, \forall k \in \{1, \ldots, K\}$ (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)
- $R_{J,k} = r_{J,k} r_{J,k}^H, k = 1, \ldots, K_J$ (normalized disturbance covariance matrix of the k’th active unlicensed jammer)
- $r_{J,k} = e^{j2\pi f_{J,k} n/f_s}, f_{J,1}/f_s = 0.7$ and $f_{J,2}/f_s = 0.75$
- $w_k = 1, \forall k \in \{1, \ldots, 7\}$ (weights in $R_I$).
Frequency spectrum and comparison to other method [3]

Single-Sided Amplitude Spectrum of c (Algorithm 2 in blue, comparison method in red)

Normalized frequency

Magnitude in dB

$\text{fs} = 300\text{MHz}$
SINR convergence
Ambiguity function
References


