Apply Alternating Direction Method of Multipliers on Beampattern Synthesis using Uniform Linear Array(ULA)

YAO-MING CHEN

Department of Electrical Engineering
Advanced Digital Communication Systems Lab
Graduate Institute of Communication Engineering
National Taiwan University, Taipei, Taiwan
Advisor: Professor Borching Su

[1] Stephen Boyd; Neal Parikh; Eric Chu; Borja Peleato; Jonathan Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers , now, 2011.
[2] J. Liang, X. Fan, H. C. So and D. Zhou, "Array Beampattern Synthesis Without Specifying Lobe Level Masks," in IEEE Transactions on Antennas and Propagation, vol. 68, no. 6, pp. 4526-4539, June 2020
[3] Wei-Ting Lin, Introduction to Alternating Direction Method of Multipliers (ADMM) and its Application on Beamformer Design in [Zhang et al. 2021], March 2023

Outline

Introduction

- In order emphasize the directivity of received signal in Satellite Communication System (SAT), our goal is to attenuate the peak sidelobe beam pattern level and maintain the main lobe beam pattern level at certain level at the same time.
- Rather than using Dattorro Iterative Algorithm, I will try to use Alternating Direction Method of Multipliers (ADMM) to solve the problem proposed in this report and see if we can have better processing time in this instance.
- This report applies ULA only.

Notation

- $\{\}^T$:transpose
- $\{\}^H$:Hermitian
- bold lowercase:vector
- bold capital:matrix
- $\Re\{\}$:real part
- $\Im\{\}$:imaginary part
- *:conjugate
- $\bigtriangledown$:gradient
- $\mathrm{s.t.}$:subject to
- $\mathbf{I}_N$:$N \times N$ identity matrix
- $\angle$:angle

Beampattern

Consider a linear array with N antenna elements places on uniformly separated locations at receiver, The far-field radiated beampattern in the direction θ can be defined as follow:

Definition of beampattern

$$\mathbf{w}^H \mathbf{a}(\theta),\ \mathbf{w},\ \mathbf{a}(\theta) \in \mathbb{C}^N. \tag{2.1}$$

$$\mathbf{a}(\theta) = [1, e^{\pi \sin\theta}, e^{2\pi \sin\theta}, ..., e^{(N-1)\pi \sin\theta}]^T. \tag{2.2}$$

Problem formulation

The several condition, we expect the absolute value of beampattern at main-lobed region to be as large as possible and the one at side-lobed region to be be as small as possible. we can formulate the following beampattern synthesis model.

Problem $P1$

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}} \ \frac{\max |\mathbf{w}^H \mathbf{a}(\bar{\theta}_s)|^2}{\min |\mathbf{w}^H \mathbf{a}(\theta_m)|^2}, \quad \bar{\theta}_s \in \Theta_s, \ \theta_m \in \Theta_m \\ &\text{s.t. } \mathbf{w}^H \mathbf{w} = 1, \end{aligned} \tag{2.3}$$

where $\Theta_s = \{-\theta_e, ..., -\theta_s, \theta_s, ..., \theta_e\} = \{\bar{\theta}_1, ..., \bar{\theta}_S\}$ is the sidelobe region uniformly discretized to S angular grid points; $\Theta_m = \{-\theta_{\text{svc}}, \theta_{\text{svc}}\} = \{\theta_1, ..., \theta_M\}$ is the mainlobe region uniformly discretized to M angular grid points.

Note that $\mathbf{w}^H \mathbf{w} = 1$ is to avoid the scaling ambiguity issue.

By define upper bound of $|\mathbf{w}^H\mathbf{a}(\bar{\theta}_s)|^2$ as e^p and lower bound of $|\mathbf{w}^H\mathbf{a}(\theta_m)|^2$ as e^q for convenience, we can rewrite our prblem as below:

Problem $P1-1$

$$\underset{\mathbf{w},p,q}{\text{minimize}}\ e^{p-q} \tag{2.4}$$

$$\text{s.t. } \mathbf{w}^H\mathbf{w} = 1 \tag{2.5}$$

$$|\mathbf{w}^H\mathbf{a}(\bar{\theta}_s)|^2 \leq e^p \quad \bar{\theta}_s \in \Theta_s, \tag{2.6}$$

$$|\mathbf{w}^H\mathbf{a}(\theta_m)|^2 \geq e^q \quad \theta_m \in \Theta_m, \tag{2.7}$$

where $p, q \in \mathbb{R}$.

By using two auxiliary variables u and v, we can rewrite our prblem as below:

Problem $P1-2$

$$\underset{\mathbf{w},p,q}{\text{minimize}}\ p-q \tag{2.8}$$

$$\text{s.t. } \mathbf{w}^H\mathbf{w}=1 \tag{2.9}$$

$$|v(s)|^2 \le e^p, \quad s=1,...,S. \tag{2.10}$$

$$|u(m)|^2 \ge e^q, \quad m=1,...,M. \tag{2.11}$$

$$v(s)=\mathbf{w}^H\mathbf{a}(\bar{\theta}_s), \quad s=1,...,S. \quad \bar{\theta}_s \in \Theta_s \tag{2.12}$$

$$u(m)=\mathbf{w}^H\mathbf{a}(\theta_m), \quad m=1,...,M. \quad \theta_m \in \Theta_m \tag{2.13}$$

Introduction of Alternating Direction Method of Multipliers

Alternating Direction Method of Multipliers(ADMM) algorithm intend to blend the decomposability of dual ascent method and the superior convergence property of method of Multipliers. ADMM solve the problem in the form:

$$\underset{\mathbf{x},\mathbf{y}}{\text{minimize}}\ f(\mathbf{x}) + g(\mathbf{y}) \tag{3.1a}$$

$$\text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c} \tag{3.1b}$$

Where $\mathbf{x} \in \mathbb{R}^n,\ \mathbf{y} \in \mathbb{R}^m,\ \mathbf{A} \in \mathbb{R}^{p\times n},\ \mathbf{B} \in \mathbb{R}^{q\times m},\ f : \mathbb{R}^n \to \mathbb{R},\ g : \mathbb{R}^m \to \mathbb{R},\ \mathbf{c} \in \mathbb{R}^p$
Using the Method of Multipliers, we can find the Aumented Lagrangian:

$$L_\rho(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}) = f(\mathbf{x}) + g(\mathbf{y}) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c}) + \rho/2||\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c}||_2^2 \tag{3.2}$$

Where $\rho > 0,\ \boldsymbol{\lambda} \in \mathbb{R}^p$
ADMM algorithm has the following iteration step:

$$\mathbf{x}^{k+1} = \arg\ \underset{\mathbf{x}}{\text{minimize}}\ L_\rho(\mathbf{x},\mathbf{y}^k,\boldsymbol{\lambda}^k) \tag{3.3a}$$

$$\mathbf{y}^{k+1} = \arg\ \underset{\mathbf{y}}{\text{minimize}}\ L_\rho(\mathbf{x}^{k+1},\mathbf{y},\boldsymbol{\lambda}^k) \tag{3.3b}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^{k+1} - \mathbf{c}) \tag{3.3c}$$

k stands for the number of iteration.

ADMM in scaled form

If we define residual $\mathbf{r}$ as $\mathbf{Ax} + \mathbf{By} - \mathbf{c}$, then:

$$
\begin{aligned}
& f(\mathbf{x}) + g(\mathbf{y}) + \boldsymbol{\lambda}^T(\mathbf{Ax} + \mathbf{By} - \mathbf{c}) + \rho/2||\mathbf{Ax} + \mathbf{By} - \mathbf{c}||_2^2 && \text{(3.4a)} \\
= & f(\mathbf{x}) + g(\mathbf{y}) + \boldsymbol{\lambda}^T\mathbf{r} + \rho/2||\mathbf{r}||_2^2 && \text{(3.4b)} \\
= & f(\mathbf{x}) + g(\mathbf{y}) + \rho/2||\mathbf{r} + (1/\rho)\boldsymbol{\lambda}||_2^2 - \rho/2||\boldsymbol{\lambda}||_2^2 && \text{(3.4c)}
\end{aligned}
$$

Then the iteration process become:

$$
\begin{aligned}
\mathbf{x}^{k+1} &= \arg \underset{\mathbf{x}}{\text{minimize}} \ f(\mathbf{x}) + g(\mathbf{y}^k) + \rho/2||\mathbf{Ax} + \mathbf{By}^k - \mathbf{c} + \boldsymbol{\mu}^k||_2^2 && \text{(3.5a)} \\
\mathbf{y}^{k+1} &= \arg \underset{\mathbf{y}}{\text{minimize}} \ f(\mathbf{x}^{k+1}) + g(\mathbf{y}) + \rho/2||\mathbf{Ax}^{k+1} + \mathbf{By} - \mathbf{c} + \boldsymbol{\mu}^k||_2^2 && \text{(3.5b)} \\
\boldsymbol{\mu}^{k+1} &= \boldsymbol{\mu}^k + \mathbf{Ax}^{k+1} + \mathbf{By}^{k+1} - \mathbf{c}, && \text{(3.5c)}
\end{aligned}
$$

where $\boldsymbol{\mu} = (1/\rho)\lambda$

Algorithm 1 ADMM

Input: ρ, θ_{svc}, θ_s, θ_e. initialize $\mathbf{w}^0$, $\boldsymbol{\lambda}^0$, $\boldsymbol{\kappa}^0$.

$k \leftarrow 0$, $T \leftarrow 20000$

repeat

$k \leftarrow k+1$

Step 1 : update $\{p^k, q^k, \mathbf{u}^k, \mathbf{v}^k\} =$

$$\arg\underset{p,q,\mathbf{u},\mathbf{v}}{\operatorname{minimize}}\ L(\mathbf{w}^{k-1}, p, q, \mathbf{u}, \mathbf{v}, \boldsymbol{\lambda}^{k-1}, \boldsymbol{\kappa}^{k-1})$$

Step 2 : update $\mathbf{w}^k =$

$$\arg\underset{\mathbf{w}}{\operatorname{minimize}}\ L(\mathbf{w}, p^k, q^k, \mathbf{u}^k, \mathbf{v}^k, \boldsymbol{\lambda}^{k-1}, \boldsymbol{\kappa}^{k-1})$$
$$\text{s.t. } \mathbf{w}^H\mathbf{w} = 1$$

Step 3 : update

$$\boldsymbol{\lambda}^k(m) = \boldsymbol{\lambda}^{k-1}(m) + \rho(u(m) - \mathbf{w}^H\mathbf{a}(\theta_m)),\ \ m = 1, ..., M$$
$$\boldsymbol{\kappa}^k(s) = \boldsymbol{\kappa}^{k-1}(s) + \rho(v(s) - \mathbf{w}^H\mathbf{a}(\bar{\theta}_s)),\ \ s = 1, ..., S$$

until $k = T$ or $(\max\{u(m) - \mathbf{w}^H\mathbf{a}(\theta_m)\} < 10^{-4}$ and $max\{v(s) - \mathbf{w}^H\bar{\theta}_s\} < 10^{-4})$

Deriving Algorithm

Revisiting our problem:

Problem $P1-2$

$$\underset{\mathbf{w},p,q}{\text{minimize}}\ p-q \tag{3.6}$$
$$\text{s.t. } \mathbf{w}^H\mathbf{w}=1 \tag{3.7}$$
$$|v(s)|^2 \le e^p, \quad s=1,...,S. \tag{3.8}$$
$$|u(m)|^2 \ge e^q, \quad m=1,...,M. \tag{3.9}$$
$$v(s)=\mathbf{w}^H\mathbf{a}(\bar{\theta}_s), \quad s=1,...,S. \quad \bar{\theta}_s \in \Theta_s \tag{3.10}$$
$$u(m)=\mathbf{w}^H\mathbf{a}(\theta_m), \quad m=1,...,M. \quad \theta_m \in \Theta_m \tag{3.11}$$

Deriving Algorithm

Augmented Lagrangian:

$$
\begin{aligned}
& L(\mathbf{w}, p, q, \mathbf{u}, \mathbf{v}, \boldsymbol{\lambda}, \boldsymbol{\kappa}) \\
&= p - q \\
&+ (\rho/2) \sum_{m=1}^{M} ||u(m) - \mathbf{w}^H \mathbf{a}(\theta_m) + (1/\rho)\lambda_m||_2^2 \\
&+ (\rho/2) \sum_{s=1}^{S} ||v(s) - \mathbf{w}^H \mathbf{a}(\bar{\theta}_s) + (1/\rho)\kappa_s||_2^2 \\
\text{s.t. } & |u(m)|^2 \geq e^q, \ m = 1, ..., M \\
& |v(s)|^2 \leq e^p, \ s = 1, ..., S \\
& \mathbf{w}^H \mathbf{w} = 1,
\end{aligned}
\tag{3.12}
$$

where $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_M]^T \in \mathbb{R}^M$, $\boldsymbol{\kappa} = [\kappa_1, ..., \kappa_S]^T \in \mathbb{R}^S$ and the penalty parameter $\rho > 0$.

Deriving Algorithm

Step 1. update $\{p^{k+1}, q^{k+1}, \mathbf{u}^{k+1}, \mathbf{v}^{k+1}\} =$

$$
\begin{aligned}
\arg \underset{p,q,\mathbf{u},\mathbf{v}}{\text{minimize}}\ & L(\mathbf{w}^k, p, q, \mathbf{u}, \mathbf{v}, \boldsymbol{\lambda}^k, \boldsymbol{\kappa}^k) \\
& = p - q \\
& + (\rho/2) \sum_{m=1}^{M} ||u(m) - x(m)||_2^2 \\
& + (\rho/2) \sum_{s=1}^{S} ||v(s) - y(s)||_2^2 \\
\text{s.t. } & |u(m)|^2 \geq e^q,\ m = 1, ..., M \\
& |v(s)|^2 \leq e^p,\ s = 1, ..., S,
\end{aligned}
\tag{3.13}
$$

where
$x(m) = \mathbf{w}^H \mathbf{a}(\theta_m) - (1/\rho)\lambda_m^k,\ m = 1, ..., M;$
$y(s) = \mathbf{w}^H \mathbf{a}(\bar{\theta}_s) - (1/\rho)\kappa_s^k,\ s = 1, ..., S.$

Deriving Algorithm

We can decomposite (3.13) as follow:

$$\underset{q,\mathbf{u}}{\text{minimize}} \; -q + (\rho/2)\sum_{m=1}^{M} ||u(m) - x(m)||_2^2 \tag{3.14}$$
$$\text{s.t. } |u(m)|^2 \geq e^q, \; m = 1, ..., M$$

$$\underset{p,\mathbf{v}}{\text{minimize}} \; p + (\rho/2)\sum_{s=1}^{S} ||v(s) - y(s)||_2^2 \tag{3.15}$$
$$\text{s.t. } |v(s)|^2 \leq e^p, \; s = 1, ..., S$$

Deriving Algorithm

Assume p, q are known, (3.14) and (3.15) can rewrite as:

$$\underset{\mathbf{u}}{\text{minimize}} \sum_{m=1}^{M} ||u(m) - x(m)||_2^2 \tag{3.16}$$
$$\text{s.t. } |u(m)|^2 \geq e^q, \ m = 1, ..., M$$

$$\underset{\mathbf{v}}{\text{minimize}} \sum_{s=1}^{S} ||v(s) - y(s)||_2^2 \tag{3.17}$$
$$\text{s.t. } |v(s)|^2 \leq e^p, \ s = 1, ..., S$$

Solve (3.16), (3.17) we can have:

$$u^{k+1}(m) = \begin{cases} e^{q/2}\exp\{j\angle x(m)\} & \text{if } |x(m)| \leq e^{\frac{q}{2}} \\ x(m) & \text{otherwise} \end{cases}, \ m = 1, ..., M \tag{3.18}$$

$$v^{k+1}(s) = \begin{cases} e^{p/2}\exp\{j\angle y(s)\} & \text{if } |y(s)| \geq e^{\frac{p}{2}} \\ y(s) & \text{otherwise} \end{cases}, \ s = 1, ..., S \tag{3.19}$$

Deriving Algorithm

For convenience, we can define a step function $S(y(s), p)$:

$$S(y(s), p) = \begin{cases} 1 & \text{if } |y(s)| \geq e^{p/2} \\ 0 & \text{otherwise} \end{cases} \tag{3.20}$$

Then we can substitute (3.19) into (3.15) and rewrite it as:

$$\underset{p}{\text{minimize}}\ p + (\rho/2) \sum_{s=1}^{S} S_1(y(s), p) \times (e^{p/2} - |y(s)|)^2 \tag{3.21}$$

In order to analyze the objective function in (3.21), we can devide p within a prior-defined region $[p_0, p_{S+1}]$ with respect to turning points $p_s = 2\ln(|y(s)|),\ (s = 1, ..., S)$ into $S + 1$ subregions, i.e.$[p_0, p_1], ..., [p_S, p_{S+1}]$. By doing so, we can analyze the convexity and concavity of our objective function in (3.21) separately in each subregion.

Deriving Algorithm

For example, choose the sth segment, i.e. $p \in [p_{s-1}, p_s]$, and define the objective function in (3.16) as $f(p) = p + (\rho/2)\sum_{s=1}^{S} S_1(y_s(s), p) \times (e^{p/2} - |y_s(s)|)^2$. Then we have:

$$f'(p) = \frac{\partial f(p)}{\partial p} = \begin{cases} 1 + (\rho/2)\sum_{s=1}^{S} e^{p/2}(e^{p/2} - |y(s)|) & \text{if } p \geq 2\ln(|y(s)|) \\ 0 & \text{otherwise} \end{cases}$$

$$= 1 + (\rho/2)\sum_{s=1}^{S} S_1(y(s), p)e^{p/2}(e^{p/2} - |y(s)|) \tag{3.22}$$

$$f''(p) = \frac{\partial^2 f(p)}{\partial p^2} = (\rho/2)\sum_{s=1}^{S} S_1(y(s), p)(e^{p} - \frac{1}{2}e^{p/2}|y(s)|) \tag{3.23}$$

Now we can anaylze the convexity and concavity of $f(p)$ and find the local-minimumed point in each subregion by using the following three cases distinguishied by different $f''(p)$, $p \in [p_{s-1}, p_s]$.

Deriving Algorithm

- Case $\mathbf{A}$: $f''(p_{s-1}) > 0$ and $f''(p_s) > 0$:
  - $f'(p_{s-1}) \geq 0$ and $f'(p_s) \geq 0$, select local-minimumed point as p_{s-1}.
  - $f'(p_{s-1}) \leq 0$ and $f'(p_s) \leq 0$, select local-minimumed point as p_s.
  - otherwise, use bisection method to find local-minimumed point, where its $f'(p) = 0$.
- Case $\mathbf{B}$: $f''(p_{s-1}) < 0$ and $f''(p_s) < 0$:
  - Use (3.10) to compute $f(p)|_{p=p_{s-1}}$ and $f(p)|_{p=p_s}$, choose local-minimumed point corresponding to lower $f(p)$.
- Case $\mathbf{C}$:otherwise
  - Use bisection method to find the inflection point p_{if} then separate the discussing subregion into two segment, i.e. $[p_{s-1}, p_{\text{if}}], [p_{\text{if}}, p_s]$. Same as previous two cases, use $f''(p)$ to categorize the convexity and concavity of these two separated subregions and use Case $\mathbf{A}$ method and Case $\mathbf{B}$ method to find local-minimumed points in each separated subregion then compare the value of $f(p)$ from these two local-minimumed points. Choose the one with smaller compared value as our local-minimumed point from this subregion.

Deriving Algorithm

- After we find all local-minimumed points from every subregion, we choose the global-minimumed point p^{k+1} with the lowest value of $f(p)$.
- Use p^{k+1} and (3.19) to update $\mathbf{v}^{k+1}$
- we can use the same procedure to find q^{k+1} and $\mathbf{u}^{k+1}$

Deriving Algorithm

Step 2. update $\mathbf{w}^{k+1}$

$$= \arg \underset{\mathbf{w}}{\text{minimize}} \; L(\mathbf{w}, p^{k+1}, q^{k+1}, \mathbf{u}^{k+1}, \mathbf{v}^{k+1}, \boldsymbol{\lambda}^k, \boldsymbol{\kappa}^k) \tag{3.24}$$

$$= \sum_{m=1}^{M} |u^{k+1}(m) + \frac{\lambda^k}{\rho} - \mathbf{w}^H \mathbf{a}(\theta_m)|^2$$

$$+ \sum_{s=1}^{S} |v^{k+1}(s) + \frac{\kappa^k}{\rho} - \mathbf{w}^H \mathbf{a}(\bar{\theta}_s)|^2$$

$$\text{s.t. } \mathbf{w}^H \mathbf{w} = 1$$

We can rewrite (3.24) into the following form:

$$\mathbf{w}^{k+1} = \arg \underset{\mathbf{w}}{\text{minimize}} \; \mathbf{w}^H R \mathbf{w} + \mathbf{w}^H \mathbf{b} + \mathbf{b}^H \mathbf{w} \tag{3.25}$$

$$\text{s.t. } \mathbf{w}^H \mathbf{w} = 1,$$

where $R = \sum_{m=1}^{M} \mathbf{a}(\theta_m)\mathbf{a}^H(\theta_m) + \sum_{s=1}^{S} \mathbf{a}(\bar{\theta}_s)\mathbf{a}^H(\bar{\theta}_s)$,
$\mathbf{b} = -\sum_{m=1}^{M} \left(u^{k+1}(m) + \frac{\lambda^k(m)}{\rho}\right)^* \mathbf{a}(\theta_m) - \sum_{s=1}^{S} \left(v^{k+1}(s) + \frac{\kappa^k(s)}{\rho}\right)^* \mathbf{a}(\bar{\theta}_s)$

Deriving Algorithm

Lagrangian of (3.25):

$$L(\mathbf{w},\gamma) = \mathbf{w}^H\mathbf{R}\mathbf{w} + \mathbf{w}^H\mathbf{b} + \mathbf{b}^H\mathbf{w} + \gamma(\mathbf{w}^H\mathbf{w} - 1),\ \gamma \in \mathbb{R} \tag{3.26}$$

Set $\frac{\partial L(\mathbf{w},\gamma)}{\partial \mathbf{w}} = 0$ and $\frac{\partial L(\mathbf{w},\gamma)}{\partial \gamma} = 0$, we have:

$$\mathbf{R}\mathbf{w} + \mathbf{b}^* + \gamma\mathbf{w} = 0 \tag{3.27}$$

$$\mathbf{w}^H\mathbf{w} = 1 \tag{3.28}$$

From (3.27) and (3.28) we have:

$$\Rightarrow \mathbf{w}^{k+1} = -(\mathbf{R} + \gamma\mathbf{I}_N)^{-1}\mathbf{b}^* \tag{3.29}$$

$$g(\gamma) = \sum_{n=1}^{N} \frac{|\mathbf{b}^H\mathbf{u}_n|^2}{(\sigma_n + \gamma)^2} = 1 \tag{3.30}$$

where $\sigma_n, \mathbf{u}_n,\ n = 1, ..., N$ are the eigenvalues and corresponding eigenvectors respectfully of R in ascending order.

Deriving Algorithm

Since $g(\gamma) \to \infty$ as $\gamma = -\sigma_N$ and $g(\gamma) = -1$ as $\gamma \to \infty$ and

$$\frac{\partial g}{\partial \gamma} = \sum_{n=1}^{N} \frac{-2|\mathbf{b}^H \mathbf{u}_n|^2}{(\sigma_n + \gamma)^3} < 0, \ \gamma \in (-\sigma_N, \infty)/\{-\sigma_{N-1}, ..., -\sigma_1\} \tag{3.31}$$

$g(\gamma)$ is a monotonically decreasing function. We can use bisection method to find γ_0 such that $g(\gamma_0) \to 0$ and use (3.29) to update $\mathbf{w}^{k+1}$:

$$\mathbf{w}^{k+1} = -(\mathbf{R} + \gamma_0 \mathbf{I}_N)^{-1} b^* \tag{3.32}$$

Deriving Algorithm

Step 3. update $\lambda^{k+1},\ \kappa^{k+1}$:

$$\lambda^{k+1}(m) = \lambda^k(m) + \rho(u(m) - \mathbf{w}^H \mathbf{a}(\theta_m)),\ m = 1, ..., M \tag{3.33}$$

$$\kappa^{k+1}(s) = \kappa^k(s) + \rho(v(s) - \mathbf{w}^H \mathbf{a}(\bar{\theta}_s)),\ s = 1, .., S \tag{3.34}$$

Stopping criteria:

$$\max\{|u(m) - \mathbf{w}^H \mathbf{a}(\theta_m)|\} \leq 10^{-4},\ m = 1, ..., M \tag{3.35}$$

$$\max\{|v(s) - \mathbf{w}^H \mathbf{a}(\bar{\theta}_s)|\} \leq 10^{-4},\ s = 1, ..., S \tag{3.36}$$

Algorithm 2 ADMM

Input: ρ, θ_{svc}, θ_s, θ_e. initialize $\mathbf{w}^0$, $\boldsymbol{\lambda}^0$, $\boldsymbol{\kappa}^0$.

$k \leftarrow 0$, $T \leftarrow 20000$

repeat

$k \leftarrow k+1$

Step 1 : update $\{p^k, q^k, \mathbf{u}^k, \mathbf{v}^k\} =$

$$\underset{p,q,\mathbf{u},\mathbf{v}}{\arg\text{minimize}}\ L(\mathbf{w}^{k-1}, p, q, \mathbf{u}, \mathbf{v}, \boldsymbol{\lambda}^{k-1}, \boldsymbol{\kappa}^{k-1})$$

Step 2 : update $\mathbf{w}^k =$

$$\begin{aligned} &\underset{\mathbf{w}}{\arg\text{minimize}}\ L(\mathbf{w}, p^k, q^k, \mathbf{u}^k, \mathbf{v}^k, \boldsymbol{\lambda}^{k-1}, \boldsymbol{\kappa}^{k-1}) \\ &\text{s.t. } \mathbf{w}^H\mathbf{w} = 1 \end{aligned}$$

Step 3 : update

$$\begin{aligned} \boldsymbol{\lambda}^k(m) &= \boldsymbol{\lambda}^{k-1}(m) + \rho(u(m) - \mathbf{w}^H\mathbf{a}(\theta_m)),\ m = 1, ..., M \\ \boldsymbol{\kappa}^k(s) &= \boldsymbol{\kappa}^{k-1}(s) + \rho(v(s) - \mathbf{w}^H\mathbf{a}(\bar{\theta}_s)),\ s = 1, ..., S \end{aligned}$$

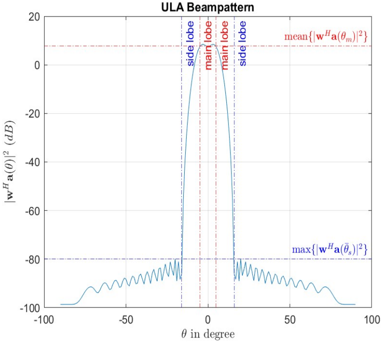
until $k = T$ or $(\max\{u(m) - \mathbf{w}^H\mathbf{a}(\theta_m)\} < 10^{-4}$ and $max\{v(s) - \mathbf{w}^H\bar{\theta}_s\} < 10^{-4})$

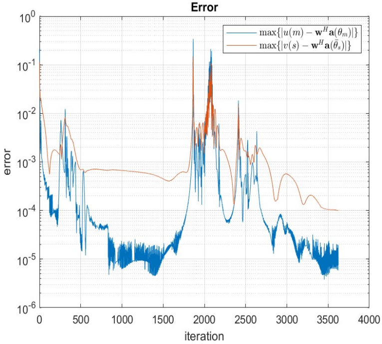
Initialization of parameters for simulation

Parameter	Symbol	Value	Units
Number of sensors	N	32	
Sampling interval of angle	$\Delta\theta$	1	degree
Initial point	$\mathbf{w}^0$	$\mathbf{1}$	
Stopping ratio	ϵ	10^{-4}	
Iteration limit	T	20000	

Simulation

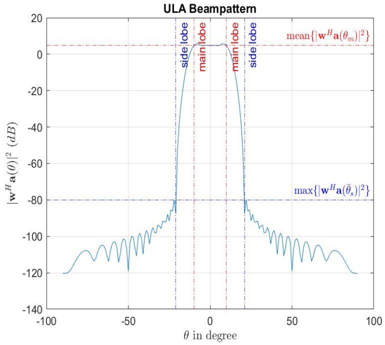
Parameter	θ_{svc}	θ_s	θ_e	ρ	iteration	elapsed time
Value	5	16	90	10	3624	84(seconds)

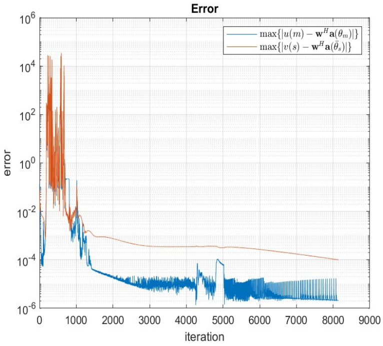




Simulation

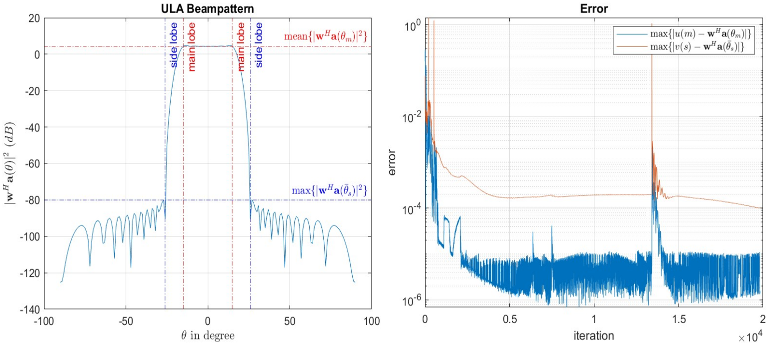
Parameter	θ_{svc}	θ_s	θ_e	ρ	iteration	elapsed time
Value	10	21	90	10	8146	183(seconds)





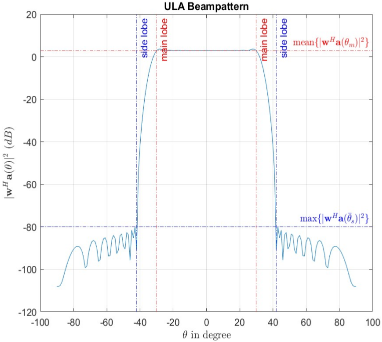
Simulation

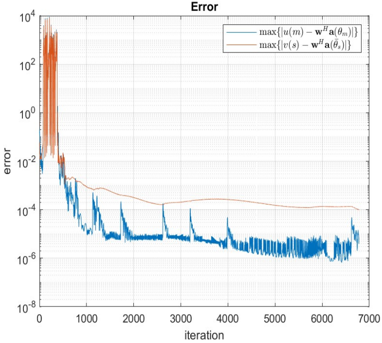
Parameter	θ_{svc}	θ_s	θ_e	ρ	iteration	elapsed time
Value	15	26	90	10	19825	324(seconds)



Simulation

Parameter	θ_{svc}	θ_s	θ_e	ρ	iteration	elapsed time
Value	30	42	90	10	6781	132(seconds)





Conclusion

Conclusion

- I tried ADMM algorithm on today's problem.
- This algorithm does not guarantee faster compiling duration.

Future work

- Modification on algorithm
- Convergence of ADMM
- Application on URA

Appendix A - Proof of existance of first order derivative

$$
\begin{aligned}
O(p) =& S_1(y_s(s),p) \times (e^{p/2} - |y_s(s)|)^2 \\
=& \begin{cases} (e^{p/2} - |y_s(s)|)^2 & if\ p \geq 2\ln(|y_s(s)|) \\ 0 & otherwise \end{cases}
\end{aligned} \tag{6.1}
$$

$$
\begin{aligned}
& \lim_{h \to 0^+} \frac{O(p+h) - O(p)}{h}\Big|_{p=2\ln|y_s(s)|} \\
=& \lim_{h \to 0^+} \frac{e^p(e^h - 1) - 2|y_s(s)|e^{p/2}(e^{h/2} - 1)}{h}\Big|_{p=2\ln|y_s(s)|} \\
=& e^{p/2}(e^{p/2} - |y_s(s)|)\Big|_{p=2\ln|y_s(s)|} \\
=& 0
\end{aligned} \tag{6.2}
$$

$$
\begin{aligned}
& \lim_{h \to 0^-} \frac{O(p+h) - O(p)}{h}\Big|_{p=2\ln|y_s(s)|} \\
=& 0
\end{aligned} \tag{6.3}
$$