POWER-EFFICIENT BEAM PATTERN SYNTHESIS VIA SEQUENTIAL OUTER APPROXIMATION PROCEDURE

Jianjun Zhang[1], *Jiaheng Wang*[1]

[1]School of Info. Sci. & Eng.
Southeast University
Nanjing, China 210000

Qingjiang Shi[2], *Yongming Huang*[1]

[2]School of Software Engineering
Tongji University
Shanghai, China 201804

ABSTRACT

The hardware implementation of large-scale multi-antenna systems requires power-efficient power amplifiers (PAs). However, the existing beamforming designs often cause a large peak-to-average power ratio and have to rely on power-inefficient PAs. In this paper, we propose a unified power-efficient beamforming design framework, which incorporates per-antenna constant envelope constraints to improve the power efficiency. We further propose an efficient algorithm named "sequential outer approximation procedure" (SOAP) to search a feasible point. Power-efficient design for beam pattern synthesis is developed based on SOAP.

1. INTRODUCTION

Large-scale multi-antenna communication systems, by employing a large number of antennas at the bases station (BS), can significantly improve the spectral efficiency via simple signal processing [1, 2]. On the other hand, the hardware implementation of such systems requires highly power-efficient power amplifiers (PAs), since the number of PAs usually scales with that of antennas. The power efficiency (PE) of a PA is mainly limited by the linear range of the transmitted signal, which, however, may have a high peak-to-average power ratio (PAPR) as a result of adapting channel conditions and modulation schemes. Consequently, traditional techniques (e.g., maximum ratio transmission) requiring a large linear range for each PA inevitably lead to lower PE [3].

To facilitate the use of power-efficient and cost-effective PAs, per-antenna constant envelope (PACE) precoding was proposed in [3, 4] to control the PAPR. In the PACE precoding, the instantaneous power of each antenna is restricted to be a constant, irrespective of channel state information and transmitted symbols. In this case, only phase variation of each antenna is used to form desired signals, and thus it is possible to use nonlinear but cheap and highly power-efficient PAs. Motivated by the advantages of PACE precoding, there has been an upsurge of research interests in constant envelope precoding for MISO and MIMO systems [4–8]. For example, in [8] a joint transceiver design under constant envelope constraints was recently investigated for MIMO point-to-point systems.

The works in [3–8] mainly considered precoding designs. However, to improve the PE is not only essential in precoding, but also crucial in other aspects of multi-antenna system designs. Particularly, due to a large path-loss of mmwave signals, large antenna arrays with power-efficient beamforming techniques are crucial in mmwave communications [9–12]. For example, since the mmwave PAs usually have lower PE, nonlinear but highly power-efficient PAs are preferred to design training beams. The type of beams that facilitates the use of power-efficient and cheap PAs is PACE beams. Moreover, PACE beams enable different PAs to output their maximal transmit powers simultaneously to increase the received SNR, which is important in mmwave communications.

To improve the PE and facilitate the use power-efficient PAs, we investigate beamforming designs for large-scale multi-antenna systems in this paper. For this purpose, we propose a unified power-efficient beamforming design framework, by formulating the beamforming design as an optimization problem with the goal of optimizing an interested performance metric under the PACE constraints. However, it is difficult to even identify a feasible point to the formulated optimization problem. To tackle this difficulty, we propose an efficient feasible point search method, i.e., sequential outer approximation procedure (SOAP). An application to beam pattern synthesis (BPS) design corroborates the excellent performance of the SOAP method.

2. PROBLEM FORMULATION

Consider a BS equipped with a large-scale antenna array consisting of $N(\gg 1)$ antennas. The BS communicates with U single-antenna users. To improve the performance (e.g., increase the received SNR, achieve a large array gain and etc.), the BS shall design a transmit/receive beamforming vector, denoted by $\mathbf{x}$, to process transmitted or received signals. To improve the PE of PAs, it is desirable that the normalized transmit power of antenna i is a constant c_i [4, 5], i.e.,

$$|\mathbf{x}(i)| = c_i, \; (\forall\, i \in \mathcal{N}), \tag{1}$$

978-1-5386-4658-8/18/$31.00 ©2019 IEEE

where $\mathcal{N} = \{1, 2, \cdots, N\}$, which guarantees that the designed beamforming vector $\mathbf{x}$ has PACE.

In practice, it is difficult for all PAs to exactly output the specified constant transmit power. To improve the robustness of the beamforming design, we allow the transmit power of the i-th antenna to vary in a small interval $[a_i, b_i]$, where $a_i > 0$, $a_i \le c_i \le b_i$, and $b_i - a_i$ is a small positive number. Accordingly, the constraints in (1) are relaxed to

$$a_i \le \left|\mathbf{e}_i^{\mathrm{H}}\mathbf{x}\right| \le b_i,\ (i \in \mathcal{N}), \tag{2}$$

where $\mathbf{e}_i = (\mathbf{0}_{i-1}^{\mathrm{T}}, 1, \mathbf{0}_{N-i}^{\mathrm{T}})^{\mathrm{T}}$. For convenience, the constraints in (2) are referred to as PE constraints.

The system design goal is to maximize a certain performance metric $f(\mathbf{x}, \mathbf{y})$ under the PE constraints and other constraints (e.g., QoS constraints), which can be formulated as

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & f(\mathbf{x},\mathbf{y}) \\ & g_i(\mathbf{x},\mathbf{y}) \le 0,\ (i \in \mathcal{I}), && \text{(3a)} \\ \text{s.t.} \quad & h_j(\mathbf{x},\mathbf{y}) \le 0,\ (j \in \mathcal{J}), && \text{(3b)} \\ & c_l(\mathbf{x},\mathbf{y}) \le 0,\ (l \in \mathcal{L}), && \text{(3c)} \end{aligned}$$

where $\mathbf{y}$ absorbs other optimization variables (e.g., transmit power) and may be absent, $g_i\,(\forall\, i \in \mathcal{I})$ and $h_j\,(\forall\, j \in \mathcal{J})$ are scalar continuously differentiable and non-convex functions, and $c_l\,(\forall\, l \in \mathcal{L})$ is a convex function. Note that the constraints in (3a) and (3b) represent different types of non-convex constraints. Typically, the non-convex PE constraints $\left\{a_i \le \left|\mathbf{e}_i^{\mathrm{H}}\mathbf{x}\right|,\ (i \in \mathcal{N})\right\}$ are absorbed in (3a), while the other types of non-convex constraints, e.g., QoS constraints, are absorbed in (3b). All convex constraints, e.g., the convex PE constraints $\left\{\left|\mathbf{e}_i^{\mathrm{H}}\mathbf{x}\right| \le b_i,\ (i \in \mathcal{N})\right\}$, are absorbed in (3c).

Let $\mathbf{z} = (\mathbf{x}, \mathbf{y})$ for convenience. For simplicity, we focus on the case where $\{g_i(\mathbf{z}) \le 0\}$ and $\{h_j(\mathbf{z}) \le 0\}$ are difference of convex (DC) functions, although the proposed SOAP is still applicable to more general optimization problems. Let $\{g_i(\mathbf{z}) \le 0\}$ and $\{h_j(\mathbf{z}) \le 0\}$ be DC functions, i.e., $g_i(\mathbf{z}) = p_i(\mathbf{z}) - q_i(\mathbf{z})\ (i \in \mathcal{I})$ and $h_j(\mathbf{z}) = m_j(\mathbf{z}) - n_j(\mathbf{z})\ (j \in \mathcal{J})$, where $p_i(\mathbf{z})$, $q_i(\mathbf{z})$, $m_j(\mathbf{z})$ and $n_j(\mathbf{z})$ are convex functions. Then, problem (3) can be written as

$$\begin{aligned} \min_{\mathbf{z}} \quad & f(\mathbf{z}) \\ & p_i(\mathbf{z}) - q_i(\mathbf{z}) \le 0,\ (i \in \mathcal{I}) \\ \text{s.t.} \quad & m_j(\mathbf{z}) - n_j(\mathbf{z}) \le 0,\ (j \in \mathcal{J}) \\ & c_l(\mathbf{z}) \le 0,\ (l \in \mathcal{L}). \end{aligned} \tag{4}$$

The feasible set $\mathcal{F}$ of problem (4) is assumed to be non-empty.

Since problem (4) is a DC programming, convex-concave procedure (CCP) can be used to find local solutions, which, however, requires an initial feasible point [13]. Since the number of constraints in (4) is very large due to the large number of antennas and different types of the constraints are coupled, it is difficult to even identify a feasible point to problem (4).To remove the need for an initial feasible point, penalty-CCP was proposed in [14]. However, as pointed out in [14], the convergence may not be a feasible point of the original problem. For these reasons, we develop a sequential outer approximation procedure (SOAP) to find a feasible point.

3. SEQUENTIAL OUTER APPROXIMATION PROCEDURE

To describe the SOAP, we first define outer approximation (OA) sequence and OA set.

Definition 1. *An OA sequence associated with the constraints* $\{p_i(\mathbf{z}) - q_i(\mathbf{z}) \le 0, (i \in \mathcal{I})\}$ *in problem* (4) *is defined as* $\mathcal{U} = \{\mathbf{u}_k \,|\, \mathbf{u}_k \in \mathbb{R}^{|\mathcal{I}|}, k = 1, \cdots, l, l+1, \cdots\}$ *such that*

$$\mathbf{u}_1(i) \ge \cdots \ge \mathbf{u}_l(i) \ge \mathbf{u}_{l+1}(i) \ge \cdots \ge 0,\ (\forall\, i \in \mathcal{I}).$$

Definition 2. *An OA set associated with an element* $\mathbf{u} \in \mathcal{U}$ *(or a real vector* $\mathbf{u} \in \mathbb{R}^{|\mathcal{I}|}$*) is defined as*

$$\mathcal{A}(\mathbf{u}) = \big\{\mathbf{z} \,|\, p_i(\mathbf{z}) - q_i(\mathbf{z}) \le \mathbf{u}(i),\ m_j(\mathbf{z}) - n_j(\mathbf{z}) \le 0,\ c_l(\mathbf{z}) \le 0,\ (i \in \mathcal{I}, j \in \mathcal{J}, l \in \mathcal{L})\big\}.$$

According to Definition 2, $\mathcal{F}$ can also be written as $\mathcal{F} = \mathcal{A}(\mathbf{0})$. To find a feasible point, the key of SOAP is to generate a series of points $\{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{\mathrm{f}}\}$ such that: 1) $\mathbf{z}_k$ does not belong to $\mathcal{F}$, i.e., $\mathbf{z}_k \notin \mathcal{F}$; 2) $\mathbf{z}_{k+1}$ is closer to $\mathcal{F}$ than $\mathbf{z}_k$; 3) $\mathbf{z}_{k+1}$ is generated based on $\mathbf{z}_k$; and 4) $\mathbf{z}_{\mathrm{f}}$ finally falls into the feasible set $\mathcal{F}$. For this purpose, as shown in Fig.1, we construct a series of OA sets $\{\mathcal{A}(\mathbf{u}_k)\}$ such that $\mathcal{F} \subseteq \cdots \subseteq \mathcal{A}(\mathbf{u}_k) \subseteq \cdots \subseteq \mathcal{A}(\mathbf{u}_2) \subseteq \mathcal{A}(\mathbf{u}_1)$, i.e., $\{\mathcal{A}(\mathbf{u}_k)\}$ approximate to $\mathcal{F}$ from the outside. With the OA set $\mathcal{A}(\mathbf{u}_k)$ available, the point $\mathbf{z}_k$ is obtained by choosing within $\mathcal{A}(\mathbf{u}_k)$ based on $\mathbf{z}_{k-1}$. In this sequential manner, we can finally obtain the desired point $\mathbf{z}_{\mathrm{f}}$.

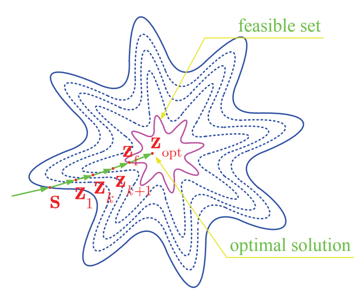


Fig. 1. An illustration of SOAP for finding a feasible point.

The main steps of SOAP are as follows. First, we choose a starting point $\mathbf{s}$. Then, we construct OA sets and choose points in an alternate fashion. Specifically, given the starting point $\mathbf{s}$, we construct the OA set $\mathcal{A}(\mathbf{u}_1)$ and choose $\mathbf{z}_1$ within $\mathcal{A}(\mathbf{u}_1)$. Then, based on $\mathbf{z}_1$, we construct the OA set $\mathbf{A}(\mathbf{u}_2)$ and choose $\mathbf{z}_2$ within $\mathcal{A}(\mathbf{u}_2)$. Generally, with $\mathbf{z}_k$ available, we construct the OA set $\mathcal{A}(\mathbf{u}_{k+1})$ and choose $\mathbf{z}_{k+1}$ within $\mathcal{A}(\mathbf{u}_{k+1})$. We repeat this procedure until reach the desired feasible point. Next, we elaborate on this procedure.

• **Choosing the Starting Point**: The first step of SOAP is to choose a starting point $\mathbf{s}$ from the following set:

$$\mathcal{A} = \left\{ \mathbf{z} \,\middle|\, m_j(\mathbf{z}) - n_j(\mathbf{z}) \le 0 \,(j \in \mathcal{J}),\, c_l(\mathbf{z}) \le 0 \,(l \in \mathcal{L}) \right\}.$$

It is seen that the first type of the non-convex constraints, i.e., $\{p_i(\mathbf{z}) - q_i(\mathbf{z}) \le 0\}$, are completely relaxed in $\mathcal{A}$. Because $\mathcal{A}$ only contains one type of the non-convex constraints, it is relatively easier to manipulate than $\mathcal{F}$. Then, we consider choosing a starting point from $\mathcal{A}$ by solving

$$\min_{\mathbf{s}} \; f(\mathbf{s}) \quad \text{s.t. } \mathbf{s} \in \mathcal{A}. \tag{5}$$

• **Choosing $\mathbf{z}_{k+1} \in \mathcal{A}(\mathbf{u}_{k+1})$ Based on $\mathbf{z}_k$**: To choose the next point $\mathbf{z}_{k+1}$ based on the current point $\mathbf{z}_k$, we first try to find a point $\mathbf{z}' \in \mathcal{F}$ based on $\mathbf{z}_k$. If we succeed in finding $\mathbf{z}'$, we set $\mathbf{z}_{\mathrm{f}} = \mathbf{z}'$ and terminate iterations. Otherwise, we construct an OA set $\mathcal{A}_1 \supseteq \mathcal{F}$ and try to find a point $\mathbf{z}' \in \mathcal{A}_1$. If we find the $\mathbf{z}'$, we set $\mathbf{z}_{k+1} = \mathbf{z}'$. Otherwise, we construct a larger OA set $\mathcal{A}_2 \supseteq \mathcal{A}_1$ and try to find a point $\mathbf{z}' \in \mathcal{A}_2$. We repeat this procedure until find a point, which is set to $\mathbf{z}_{k+1}$. To achieve this goal, we associate the OA sets $\{\mathcal{A}(\mathbf{u}_k)\}$ with constants $\{T_{k-1,i}\}$. The constants associated with $\mathcal{A}(\mathbf{u}_1)$ are $\{T_{0,i} = |p_i(\mathbf{s}) - q_i(\mathbf{s})|,\, i \in \mathcal{I}\}$. **Given** $\mathbf{z}_k \in \mathcal{A}(\mathbf{u}_k)$ **and** $\{T_{k,i}\}$, the operations (OPs) to **find** $\mathbf{z}_{k+1}$ are as follows.

Op 1: We define an integer variable e and initialize it as 0, i.e., $e = 0$. Then, we construct $\mathbf{u}_{k+1}$ as

$$\mathbf{u}_{k+1}(i) = T_{k,i}(1 - 2^{-e}) \,(\forall\, i \in \mathcal{I}). \tag{6}$$

Once $\mathbf{u}_{k+1}$ has been constructed, we obtain the corresponding OA set $\mathcal{A}(\mathbf{u}_{k+1})$ according to Definition 2.

Op 2: Using $\mathbf{z}_k$ as an initial point, we try to find a point within $\mathcal{A}(\mathbf{u}_{k+1})$, typically, by solving

$$\begin{aligned} \min_{\mathbf{z}} \quad & f(\mathbf{z}) \\ \text{s.t.} \quad & p_i(\mathbf{z}) - q_i(\mathbf{z}) \le \mathbf{u}_{k+1}(i),\, (i \in \mathcal{I}) \\ & m_j(\mathbf{z}) - n_j(\mathbf{z}) \le 0,\, (j \in \mathcal{J}) \\ & c_l(\mathbf{z}) \le 0,\, (l \in \mathcal{L}). \end{aligned} \tag{7}$$

Using $\mathbf{z}_k$ as an initial point, problem (7) can be solved via CCP.

If we successfully find a point (e.g., $\mathbf{z}'$) by solving problem (7) via CCP based on $\mathbf{z}_k$, we set $\mathbf{z}_{k+1} = \mathbf{z}'$ and $T_{k+1,i} = T_{k,i}(1 - 2^{-e})\,(\forall\, i \in \mathcal{I})$. Otherwise, we increase e (e.g., let $e \leftarrow e + 1$) and reconstruct $\mathbf{u}_{k+1}$ according to (6) and solve the new optimization problem. For clarity, given $\mathbf{z}_k$ and $T_{k,i}\,(i \in \mathcal{I})$, the algorithm to find $\mathbf{z}_{k+1}$ is summarized in Algorithm 1. The algorithm to find a feasible point of problem (4) via SOAP is summarized in Algorithm 2.

The convergence of SOAP is characterized in Theorems 1 and 2. Theorem 1 indicates that given $\mathbf{z}_k \in \mathcal{A}(\mathbf{u}_k)$, it is guaranteed to find the next point $\mathbf{z}_{k+1} \in \mathcal{A}(\mathbf{u}_{k+1})$ as long as e is sufficiently large. Theorem 2 indicates that it is sufficient to construct finite OA sets.

Algorithm 1: Given $\mathbf{z}_k$ and $T_{k,i}\,(i \in \mathcal{I})$, Find $\mathbf{z}_{k+1}$

1: **initialize:** $e = 0$
2: **repeat**
   (a) construct $\mathbf{u}_{k+1}$ according to (6), i.e.,
       $\mathbf{u}_{k+1}(i) = T_{k,i}(1 - 2^{-e}),\ (i \in \mathcal{I})$
   (b) construct OA set $\mathcal{A}(\mathbf{u}_{k+1})$
   (c) solve problem (7): if $\mathbf{z}_{k+1}$ cannot be found
       by solving problem (7), let $e \leftarrow e + 1$
   until find the point $\mathbf{z}_{k+1} \in \mathcal{A}(\mathbf{u}_{k+1})$
3: **output:** $\mathbf{z}_{k+1}$ and $T_{k+1,i} = T_{k,i}(1 - 2^{-e})\,(i \in \mathcal{I})$

Algorithm 2: Feasible Point Search - SOAP

1: **solve** problem (5) $\Longrightarrow$ starting point $\mathbf{s}$
2: **let** $T_{0,i} = |p_i(\mathbf{s}) - q_i(\mathbf{s})|$, $\mathbf{z}_0 = \mathbf{s}$ and $k = 0$
3: **repeat**
   (a) find $\mathbf{z}_{k+1}$ and $T_{k+1,i}\,(i \in \mathcal{I})$ via Algorithm 1
   (b) $k \leftarrow k + 1$
   until some termination criterion is met
4: **output:** feasible point $\mathbf{z}_{\mathrm{f}}$ of problem (4).

Theorem 1. *Suppose that $\mathbf{z}_k$ is a regular local minimizer of $\mathbf{f}(\mathbf{z})$ on $\mathcal{A}(\mathbf{u}_k)$ and $\mathbf{u}_{k+1}$ is constructed according to* (6)*. Then, if e is sufficiently large, it is guaranteed to find a point $\mathbf{z}_{k+1} \in \mathcal{A}(\mathbf{u}_{k+1})$ via CCP using $\mathbf{z}_k$ as the initial point.*

Theorem 2. *Suppose that 1) the feasible set $\mathcal{F}$ of problem* (4) *contains an interior point; 2) $\{\nabla q_i(\mathbf{z}), \nabla n_j(\mathbf{z}) \,|\, i \in \mathcal{I}, j \in \mathcal{J}\}$ are continuous functions; and 3) $\forall\, \mathbf{z} \notin \mathcal{F}$, it is able to find a point closer to $\mathcal{F}$. Then, there exists finite K such that Algorithm 2 is guaranteed to terminate for $k \ge K$.*

4. APPLICATION: BEAM PATTERN SYNTHESIS

4.1. Problem Statement

The goal of beam pattern synthesis (BPS) is to design a beamforming vector having desired beam pattern in both mainlobe and sidelobe [15–17]. Different applications require different problem formulations. In this section, we consider BPS for beam alignment (BA) in mmwave communication.

To achieve a good BA performance, the synthesized beam should have flat and large beam pattern in the mainlobe and flat but small beam pattern in the sidelobe [11], which yields

$$\begin{aligned} \min_{\mathbf{x},\varepsilon} \quad & \varepsilon \\ & a_i \le |\mathbf{e}_i^{\mathrm{H}}\mathbf{x}| \le b_i,\ (i \in \mathcal{N})\,, && \text{(8a)} \\ \text{s.t.} \quad & 1 - \varepsilon \le \left|\mathbf{a}^{\mathrm{H}}(\psi)\mathbf{x}\right| \le 1 + \varepsilon,\ (\psi \in \mathcal{D}_{\mathrm{M}})\,, && \text{(8b)} \\ & \left|\mathbf{a}^{\mathrm{H}}(\psi)\mathbf{x}\right| \le \varepsilon,\ (\psi \in \mathcal{D}_{\mathrm{S}})\,, && \text{(8c)} \end{aligned}$$

where $\mathcal{D}_{\mathrm{M}} = \{\psi_1, \psi_2, \cdots\}$ and $\mathcal{D}_{\mathrm{S}} = \{\psi'_1, \psi'_2, \cdots\}$ contain the sampling points within the mainlobe and the sidelobe, respectively. The constraints in (8b) and (8c) are used to control the ripples in the mainlobe and the sidelobe, respectively. The design goal is to minimize the maximal ripples.

Letting $\mathbf{y} = \varepsilon$ and casting problem (8) into problem (3), we have the following identification

$$\begin{aligned} f(\mathbf{x}, \mathbf{y}) &= (\mathbf{0}_N^{\mathrm{T}}, 1)(\mathbf{x}^{\mathrm{T}}, \varepsilon)^{\mathrm{T}} \\ \{g_i(\mathbf{x}, \mathbf{y}) \le 0\} &= \left\{a_i \le \left|\mathbf{e}_i^{\mathrm{H}}\mathbf{x}\right|, \, (i \in \mathcal{N})\right\} \\ \{h_j(\mathbf{x}, \mathbf{y}) \le 0\} &= \left\{1 - \varepsilon \le \left|\mathbf{a}^{\mathrm{H}}(\psi)\mathbf{x}\right|, \, (\psi \in \mathcal{D}_{\mathrm{M}})\right\} \\ \{c_l(\mathbf{x}, \mathbf{y}) \le 0\} &= \left\{\left|\mathbf{e}_i^{\mathrm{H}}\mathbf{x}\right| \le b_i, \, (i \in \mathcal{N})\right\} \cup \\ &\qquad \left\{\left|\mathbf{a}^{\mathrm{H}}(\psi)\mathbf{x}\right| \le 1 - \varepsilon, \, (\psi \in \mathcal{D}_{\mathrm{M}})\right\}. \end{aligned} \tag{9}$$

With the identification in (9), Algorithm 2 can be used to find a feasible point to problem (8).

4.2. Numerical Results

The transmit power of each antenna is restricted to $[0.9, 1.0]$. Fig.2 shows the normalized beam pattern (NBP) of the initial feasible solution (via Algorithm 2) and the stationary solution. It is seen that the feasible solution has almost the same performance as the stationary solution, e.g., small ripples in both mainlobe and sidelobe. Generally, the feasible solution obtained by Algorithm 2 is not only feasible, but also possesses desired properties and satisfies most applications.

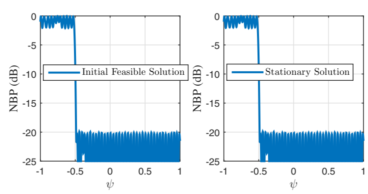
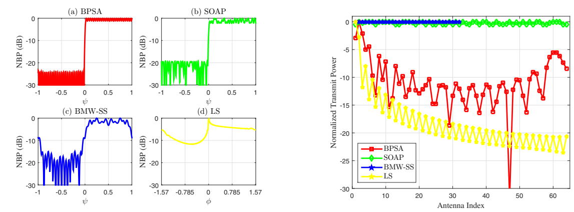


Fig. 2. The NBP of the initial feasible solution and the stationary solution.

The normalized per-antenna transmit power (NPTP) defined as $10 \log \left(|\mathbf{x}(i)| / \max\{|\mathbf{x}(i)|\}\right)$ and NBP of the beams designed by different methods is shown in Fig.3. For comparison, the beams designed with the LS method [10], the BMW-SS method [18], and the BPSA [11] are shown as well. It is seen from Fig.3-(a) that BPSA obtains the best performance in terms of suppressing the ripples in both mainlobe and sidelobe. Compared to BPSA, SOAP is inferior in ripple suppression. The reason is that the PACE constraints are imposed in SOAP. On the other hand, it is seen from Fig.3-(b) that the SOAP beam has the smallest difference in per-antenna transmit power, and thus its PAPR is also minimal. Though for the BMW-SS beam, the transmit power of the 32 active antennas is the same, the remaining antennas are inactive. Note that deactivating antennas is not a good approach for the design of wider beams, since it will lead to some loss in array gain.



(a) NBP (b) NPTP

Fig. 3. The NBP and NPTP of the beams designed with different methods - $N = 64$.

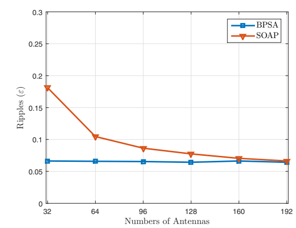


Fig. 4. The magnitude of the ripples (i.e., ε) of BPSA and SOAP beams for varying numbers of antennas.

The magnitude of the ripples of the beams designed with BPSA and SOAP for varying numbers of antennas are shown in Fig.4. It is seen that the gap of the magnitude of the ripples between the BPSA and SOAP beams becomes smaller, as the number of antennas increases. In particular, when $N \ge 160$, the magnitude of the ripples of the SOAP beam is close to that of the BPSA beam. The reason is that more antennas provide more degrees of freedom, which can be fully utilized by SOAP. Consequently, the performance loss due to the PACE constraints becomes negligible for large-scale antenna arrays.

5. CONCLUSION

We proposed a unified power-efficient beamforming design framework and devised the SOAP to search feasible points. Power-efficient design for beam pattern synthesis was developed based on the SOAP. Simulations verified the excellent performance of SOAP and revealed some important insights.

6. ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 61571107, 61711540305, 61671411, U1709219, 61720106003, 61801435, the National Science and Technology Major Project of China under Grant 2018ZX03001002-003the National Key Research and Development Project of China under Grant 2017YFE0119300, and the Research Project of Jiangsu Province under Grant BK20160069, BE2018121.

7. REFERENCES

[1] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, November 2010.

[2] S. He, J. Wang, Y. Huang, B. Ottersten, and W. Hong, "Codebook-based hybrid precoding for millimeter wave multiuser systems," *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5289–5304, Oct 2017.

[3] S. K. Mohammed and E. G. Larsson, "Constant envelope precoding for power-efficient downlink wireless communication in multi-user mimo systems using large antenna arrays," in *2012 IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*, March 2012, pp. 2949–2952.

[4] S. K. Mohammed and E. G. Larsson, "Single-user beamforming in large-scale miso systems with per-antenna constant-envelope constraints: The doughnut channel," *IEEE Trans. Wireless Commun.*, vol. 11, no. 11, pp. 3992–4005, November 2012.

[5] S. K. Mohammed and E. G. Larsson, "Per-antenna constant envelope precoding for large multi-user mimo systems," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 1059–1071, March 2013.

[6] P. V. Amadori and C. Masouros, "Constant envelope precoding by interference exploitation in phase shift keying-modulated multiuser transmission," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 538–550, Jan 2017.

[7] J. Zhang, Y. Huang, J. Wang, B. Ottersten, and L. Yang, "Per-antenna constant envelope precoding and antenna subset selection: A geometric approach," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6089–6104, Dec 2016.

[8] S. Zhang, R. Zhang, and T. J. Lim, "Constant envelope precoding for mimo systems," *IEEE Trans. Commun.*, vol. 66, no. 1, pp. 149–162, Jan 2018.

[9] V. Raghavan, J. Cezanne, S. Subramanian, A. Sampath, and O. Koymen, "Beamforming tradeoffs for initial ue discovery in millimeter-wave mimo systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 543–559, April 2016.

[10] A. Alkhateeb, O. El Ayach, G. Leus, and R.W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 831–846, Oct 2014.

[11] J. Zhang, Y. Huang, Q. Shi, J. Wang, and L. Yang, "Codebook design for beam alignment in millimeter wave communication systems," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4980–4995, Nov 2017.

[12] J. Song, J. Choi, and D. J. Love, "Common codebook millimeter wave beam design: Designing beams for both sounding and communication with uniform planar arrays," *IEEE Trans. Commun.*, vol. 65, no. 4, pp. 1859–1872, April 2017.

[13] A. L. Yuille and A Rangarajan, "The concave-convex procedure," *Neural Computation*, vol. 15, no. 4, pp. 915–936, 2003.

[14] Thomas Lipp and Stephen Boyd, "Variations and extension of the convexcconcave procedure," *Optimization & Engineering*, vol. 17, no. 2, pp. 263–287, 2016.

[15] H. Lebret and S. Boyd, "Antenna array pattern synthesis via convex optimization," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 526–532, March 1997.

[16] Fan Wang, V. Balakrishnan, P.Y. Zhou, J.J. Chen, R. Yang, and C. Frank, "Optimal array pattern synthesis using semidefinite programming," *IEEE Trans. Signal Process.*, vol. 51, no. 5, pp. 1172–1183, May 2003.

[17] B. Fuchs, "Application of convex relaxation to array synthesis problems," *IEEE Trans. Antennas Propag.*, vol. 62, no. 2, pp. 634–640, Feb 2014.

[18] Z. Xiao, T. He, P. Xia, and X. G. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3380–3392, May 2016.