An Adaptive Distributed Asynchronous Algorithm with Application to Target Localization

Rahul Mourya*, Adil Salim *, Pascal Bianchi* and Cédric Richard‡
*LTCI, Télécom ParisTech, Univesité Paris-Saclay, Paris 75013, France
‡Laboratoire Lagrange, Observatoire de la Côte d'Azur, Université de Nice Sophia Antipolis, Nice 06108, France

***Abstract*—This paper considers a generic stochastic optimization problem arising in many applications, and presents a distributed adaptive optimization algorithm based on Douglas-Rachford splitting algorithm. The proposed algorithm is generic enough to be easily adapted in both synchronous and asynchronous settings. To illustrate the usability of the algorithm, we consider an application of localizing slowly moving target using the measurements from a network of sensors (nodes) of multistatic continuous active sonar (MCAS) system, which provides range and direction-of-arrival measurements. The nodes are endowed with private cost functions, and seek to find a consensus on the minimizer of the aggregate cost. In particular, the target localization is based on least squares estimate using mixed range-and-DOA measurements, which is indeed a nonconvex optimization problem. Despite the difficulty of the problem, the proposed algorithm operates *asynchronously*, and is able to estimate efficiently the exact (global) solution and track the target with good accuracy.**

***Index Terms*—Stochastic approximation, distributed optimization, proximal projection, target localization.**

I. INTRODUCTION

A broadly investigated subject in optimization and signal processing consists in solving the optimization problem

$$\min_{x\in\mathcal{X}} \mathbb{E}(f(x,\theta))$$

where $\mathcal{X}$ is an Euclidean space, and $\mathbb{E}$ is the expectation over the random variable (r.v.) θ and $f(\,.\,,\theta)$ is some random function. A case of interest is the scenario where the expectation cannot be directly computed, either because the probability distribution of θ is unavailable or because the corresponding integral cannot be numerically evaluated. Instead, it is commonly assumed that the expectation is progressively revealed by means of i.i.d. copies of θ. Iterative algorithms such as the standard stochastic gradient algorithm or the stochastic proximal point algorithm [1] can be used to track a minimizer. At every step, the iterate is updated using one realization of θ. A key-parameter is the *step size* which can either be kept constant along the iteration or vanishing. In the vanishing step size case, the iterates can be shown to converge to a minimizer as the iteration index tends to infinity, under some hypotheses. In the constant step size case, the iterate do not converge in the almost sure sense, but eventually fluctuate in a certain vicinity of a minimizer. The constant step size case is especially interesting in the context of adaptive signal processing: the step size is kept constant in order to maintain the tracking abilities of the algorithm [2].

The work is supported by the Agence Nationale pour la Recherche (ANR), France, (ODISSEE project, ANR-13-ASTR-0030).

In this paper, we propose a *distributed* stochastic algorithm over-graphs with constant step size. Similar distributed algorithms are proposed in [3,4] that focus on diffusion-based adaptive solutions to least-mean-square problems. Consider an undirected and connected graph $\mathsf{G} = (\mathsf{V},\mathsf{E})$ with N vertices/nodes, where $\mathsf{V} := \{1,\dots,N\}$ is the set of vertices and E is the set of edges.The problem of interest has the form:

$$\min_{\boldsymbol{x}\in\mathsf{X}^{\mathsf{V}}} \sum_{v\in\mathsf{V}} \mathbb{E}(f_v(\boldsymbol{x}_v,\theta_v)) + \sum_{\substack{v<w\\ \{v,w\}\in\mathsf{E}}} g_{\{v,w\}}(\boldsymbol{x}_v,\boldsymbol{x}_w)\,, \qquad (1)$$

where for every $v \in \mathsf{V}$, θ_v is random variable on some probability space into some measurable space Ξ_v, $f_v(\,.\,,\theta_v)$ is a random function in the set $\Gamma_0(\mathcal{X})$ of convex, proper and lower semicontinuous functions on $\mathcal{X} \to (+\infty,+\infty]$ and g_e ($e \in \mathsf{E}$) are functions in $\Gamma_0(\mathsf{X}\times\mathsf{X})$, referred to as *regularizers*. The functions f_v represent some private cost, known only locally at the node $v \in \mathsf{V}$. The regularizers g_e ($e \in \mathsf{E}$) ensure the coupling between the variables ($x_v : v \in \mathsf{V}$). A special case of Problem (2) is given when every function g_e is defined by

$$g_e(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} 0 & \text{if } \boldsymbol{x}=\boldsymbol{y} \\ +\infty & \text{otherwise.} \end{cases}$$

In this case, using the fact that the graph is connected, the sum in the second term of (1) is equal to zero if $\boldsymbol{x}_1 = \cdots = \boldsymbol{x}_N$ and to $+\infty$ otherwise. Hence, Problem (2) is equivalent to the *consensus problem*:

$$\min_{\boldsymbol{x}\in\mathsf{X}^{\mathsf{V}}} \sum_{v\in\mathsf{V}} \mathbb{E}(f_v(\boldsymbol{x}_v,\theta)) \text{ s.t. } \boldsymbol{x}_1 = \cdots = \boldsymbol{x}_N\,. \qquad (2)$$

In other words, all nodes are seeking to find a common minimizer of the aggregate cost $\sum_{v\in\mathsf{V}} \mathbb{E}(f_v(\,.\,,\theta))$. Compared to (2), the generic formulation (1) is useful to cover the case of total variation regularization ($g_{\{v,w\}}(x,y) = \|x-y\|$) of Laplacian regularization ($g_{\{v,w\}}(x,y) = \|x-y\|^2$). It is also useful in practical applications such as *distributed target localization*, which we will consider and discuss at length in this paper as the main motivation for our algorithm.

In this paper, we propose an asynchronous distributed algorithm to solve the problem (1) and its special case (2). The algorithm is asynchronous in the sense that, at every iteration, only a certain number of (randomly chosen) nodes update and exchange their variables, other nodes of the network being idle. We derive our algorithm as a special instance of an algorithm recently derived in [**?**]. The algorithm of [**?**] is a stochastic version of the celebrated Douglas-Rachford algorithm [5]. The Douglas-Rachford algorithm is used to minimize the sum of two functions. In its stochastic counterpart [?], the latter are replaced by random functions observed at every iteration of the algorithm. An asymptotic analysis is provided under the hypothesis of a constant step size. The distributed problem (1) can be seen as an instance of the stochastic programming problem solved by the stochastic/adaptive Douglas-Rachford algorithm of [?]. The nature of the randomness is twofolds. First, the innovation: every node locally observes some random realization of every function $f_v(\,.\,,\theta_v)$ at each step. Second, the asynchronous communications: only certain nodes chosen at random communicate at a given time. Here, the main idea in order to incorporate asynchronous communications in the algorithm, consists in reformulating the second sum in (1) as an expectation over the (random) active edges and then to apply the adaptive Douglas-Rachford algorithm.

As a final contribution, we apply our adaptive algorithm to the problem of target (slowly moving) localization using a network of MCAS receivers.

The paper is organized as follows. In Section II, we present a generic formulation of problems (1), and an adaptive algorithm to solve it. In Section III, we consider an application of localizing slowly moving target using measurements from sensors network, and compare the performance of the synchronous and asynchronous version of the proposed adaptive algorithm.

II. Main Algorithm

The notation prox represents the *proximity operator*, defined for every $h \in \Gamma_0(\mathcal{X})$ and every $x \in \mathcal{X}$ by

$$\mathrm{prox}_h(x) = \arg\min_{y\in\mathcal{X}} h(y) + \frac{\|y-x\|^2}{2}\,.$$

If $\mathcal{A}$ is a set, the notation $\iota_{\mathcal{A}}$ stands for the indicator function of the set $\mathcal{A}$, equal to zero on that set and to $+\infty$ elsewhere.

A. Adaptive Douglas Rachford Algorithm

Let ξ be a random variable (r.v.) defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ into an arbitrary measurable space $(\Xi, \mathcal{G})$. We say that a mapping $f : \mathcal{X} \times \Xi \to (-\infty, +\infty]$ is a normal convex integrand if $f(\,.\,, s) \in \Gamma_0(\mathcal{X})$ for every $s \in \Xi$ and if $f(x,\,.\,)$ is measurable for every $x \in \mathcal{X}$.

Consider a generic problem of the form

$$\min_{x\in\mathcal{X}} \mathbb{E}(f(x,\xi)) + \mathbb{E}(g(x,\xi))$$

where f, g are normal convex integrands. Let $(\xi^{(k)} : k \in \mathbb{N})$ a sequence of iid copies of the r.v. ξ. In [?], the following algorithm is proposed:

$$\begin{aligned}
\boldsymbol{x}^{k+1} &= \mathrm{prox}_{\gamma f(\,.\,,\xi^{(k+1)})}(\boldsymbol{u}^{(k)}) \\
\boldsymbol{z}^{k+1} &= \mathrm{prox}_{\gamma g(\,.\,,\xi^{(k+1)})}(2\boldsymbol{x}^{(k+1)} - \boldsymbol{u}^{(k)}) \\
\boldsymbol{u}^{k+1} &= \boldsymbol{u}^{(k)} + \boldsymbol{z}^{(k+1)} - \boldsymbol{x}^{(k+1)}\,,
\end{aligned}$$

where $\gamma > 0$ is the step size of the algorithm. The algorithm is an immediate extension of the Douglas-Rachford algorithm [5], where deterministic functions are replaced by random realizations.

Convergence result

B. Application to the Asynchronous Consensus Problem (2)

To simplify the presentation, we first start by describing our algorithm in the special case of the consensus problem (2). Define $\boldsymbol{\theta} = (\theta_v : v \in V)$ and let $(\boldsymbol{\theta}^{(k)} : k \in \mathbb{N})$ be a sequence of iid copies of the r.v. $\boldsymbol{\theta}$. We assume the following **asynchronous communication model**. At every iteration k, a random node $v^{(k)}$ is chosen according to the uniform distribution on V. This node observes the r.v. $\theta_v^{(k)}$ and updates some local variable. Next, during some communication step, a node $w^{(k)}$ is chosen uniformly amongst the neighbors of node $v^{(k)}$, and the two nodes $v^{(k)}$, $w^{(k)}$ exchange some local variables. Other nodes are idle during this step. The sequence $((v^{(k)}, w^{(k)}) : k \in \mathbb{N})$ is supposed i.i.d. and independent from the sequence $(\boldsymbol{\theta}^{(k)} : k \in \mathbb{N})$. The following lemma follows easily from the fact that the graph is connected. The proof is omitted.

Lemma II.1. *Problem (2) is equivalent to*

$$\min_{\boldsymbol{x}\in\mathsf{X}^{\mathsf{V}}} \mathbb{E}\left(f_{v^{(1)}}(\boldsymbol{x}_{v^{(1)}}, \theta^{(1)}_{v^{(1)}})\right) + \mathbb{E}\left(\iota_{\mathcal{C}}(\boldsymbol{x}_{v^{(1)}}, \boldsymbol{x}_{w^{(1)}})\right) \quad (3)$$

where $\iota_{\mathcal{C}}$ is the indicator function of the set $\mathcal{C} := \{(x,x) : x \in \mathcal{X}\}$.

We now apply the adaptive Douglas Rachford algorithm to (3), substituting $\mathcal{X}$ with $\mathcal{X}^V$, letting $\xi = (\boldsymbol{\theta}^{(1)}, v^{(1)}, w^{(1)})$ and choosing $f(\boldsymbol{x},\xi) := f_{v^{(1)}}(\boldsymbol{x}_{v^{(1)}}, \theta^{(1)}_{v^{(1)}})$ and $g(\boldsymbol{x},\xi) := \iota_{\mathcal{C}}(\boldsymbol{x}_{v^{(1)}}, \boldsymbol{x}_{w^{(1)}})$. After some algebra, the iterates read as follows.

At iteration $k+1$, denote for simplicity $v = v^{k+1}$, $w = w^{k+1}$ and set $f_v^{k+1} := f_v(\,.\,,\theta_v^{(k+1)})$:

$$\begin{aligned}
\boldsymbol{x}_v^{(k+1)} &= \mathrm{prox}_{\gamma f_v^{(k+1)}}(\boldsymbol{u}_v^{(k)}) \\
\boldsymbol{u}_v^{(k+1)} &= \frac{1}{2}(u_v^{(k+1)} + u_w^{(k+1)}) \\
\boldsymbol{u}_w^{(k+1)} &= \boldsymbol{x}_v^{(k+1)} + \frac{1}{2}(u_w^{(k+1)} - u_v^{(k+1)})\,,
\end{aligned}$$

and for every $\ell \notin \{v,w\}$, $\boldsymbol{u}_\ell^{(k+1)} = \boldsymbol{u}_v^{(k)}$.

convergence result

C. Generalization

In this paragraph, we generalize our algorithm to the following case:

- We address the general problem (1);
- We use a more general asynchronous communication model. Specifically, we distinguish between *computing nodes* and *communicating nodes*. Several random nodes v (the computing nodes) observe their local r.v. $\theta_v^{(k)}$ at iteration k and compute the output of a proximity operator. These nodes are refered to as the computing nodes. In addition, a certain set of nodes (not necessarily restricted to a single edge) participate to the exchange of variables at iteration k. These nodes are refered to as the communicating nodes. As in the previous paragraph, there might be an overlap between computing and communicating nodes, but it is not mandatory: we make no such assumption here. This way, our general model

encompasses a large number of scenarios. We refer to Section III for an example.

Let us be formal. We introduce a random variable $\nu : \Omega \to 2^V$ taking its values in the set of subsets of V. The elements of ν are the computing nodes. We also introduce a random variable $\varepsilon : \Omega \to 2^E$ taking its values in the set of subsets of E. Elements of ε are the active edges, and we identify the communicating nodes with all nodes belonging to at least one active edge. We define for every $v \in V$, $e \in E$, the probabilities: $p_v := \mathbb{P}(v \in \nu), q_e := \mathbb{P}(e \in \varepsilon)$.

Assumption II.1. *For every $v \in \mathsf{V}$, $p_v > 0$, and for every $e \in \mathsf{E}$, $q_e > 0$.*

Assumption II.2. *The r.v. θ and ν are independent.*

Define the r.v. $\xi := (\theta, \nu, \varepsilon)$ on the space $\Xi := \Theta \times 2^V \times 2^E$ and introduce some maps $f, g : \mathcal{X}^V \times \Xi \to (-\infty, +\infty]$ s.t. w.p.1,

$$f(\boldsymbol{x}, \xi) := \sum_{v \in \nu} p_v^{-1} f_v(\boldsymbol{x}_v, \theta)$$

$$g(\boldsymbol{x}, \xi) := \sum_{\substack{v<w \\ \{v,w\} \in \varepsilon}} q_{\{v,w\}}^{-1} g_{\{v,w\}}(\boldsymbol{x}_v, \boldsymbol{x}_w),$$

for every $\boldsymbol{x} \in \mathsf{X}^V$. Under the Assumptions II.1 and II.2, we can write:

$$\mathbb{E}(f(\boldsymbol{x}, \xi)) = \sum_{v \in V} \mathbb{E}(f_v(\boldsymbol{x}, \theta)),$$

$$\mathbb{E}(g(\boldsymbol{x}, \xi)) = \sum_{\substack{v<w \\ \{v,w\} \in E}} g_{\{v,w\}}(\boldsymbol{x}_v, \boldsymbol{x}_w).$$

Now, we can thus apply the adaptive Douglas-Rachford algorithm with the sequence $\xi^{(k)} := (\theta^{(k)}, \nu^{(k)}, \varepsilon^{(k)})$. In the sequel we define $\mathsf{V}(\varepsilon) := \{v \in V : \exists w \in V, \{v, w\} \in \varepsilon\}$. The algorithm yields the following iterates at time $k+1$.

In the first step, every computing node $v \in \nu^{(k+1)}$ generates

$$\boldsymbol{x}_v^{(k+1)} = \operatorname{prox}_{\gamma p_v^{-1}, f_v(\,.\,, \theta^{(k+1)})}(\boldsymbol{u}_v^{(k)}) \tag{4}$$

whereas other nodes $v \notin \nu^{(k+1)}$ simply set $\boldsymbol{x}_v^{(k+1)} = \boldsymbol{u}_v^{(k)}$. The set of communicating nodes $\mathsf{V}(\varepsilon^{(k+1)})$ jointly compute

$$\begin{aligned}(\boldsymbol{z}_v^{(k+1)} : v \in \mathsf{V}(\varepsilon^{(k+1)})) = \\ \arg \min_{\boldsymbol{z} \in \mathsf{X}^{\mathsf{V}(\varepsilon^{(k+1)})}} \sum_{\substack{v<w \\ \{v,w\} \in \varepsilon}} q_{\{v,w\}}^{-1} g_{\{v,w\}}(\boldsymbol{z}_v, \boldsymbol{z}_w) \\ + \frac{1}{2\gamma} \sum_{v \in \mathsf{V}(\varepsilon^{(k+1)})} \|\boldsymbol{z}_v - 2\boldsymbol{x}_v^{(k+1)} + \boldsymbol{u}_v^{(k)}\|^2,\end{aligned} \tag{5}$$

whereas other nodes $v \notin \mathsf{V}(\varepsilon^{(k+1)})$ simply set $\boldsymbol{z}_v^{(k+1)} = 2\boldsymbol{x}_v^{(k+1)} - \boldsymbol{u}_v^{(k)} = \boldsymbol{u}_v^{(k)}$. Finally, all nodes $v \in V$ set

$$\boldsymbol{u}_v^{(k+1)} = \boldsymbol{u}_v^{(k)} + \boldsymbol{z}_v^{(k+1)} - \boldsymbol{x}_v^{(k+1)} \tag{6}$$

which boils down to $\boldsymbol{u}_v^{(k+1)} = \boldsymbol{x}_v^{(k)} = \boldsymbol{u}_v^{(k)}$ if v neither belongs to $\nu^{(k+1)}$ nor to $\mathsf{V}(\varepsilon^{(k+1)})$.

Theorem II.1. *Assume that $f_v(\cdot, \theta)$ and $g_{\{v,w\}}$ are w.p.1 convex, and that there exists $L > 0$ such that $f_v(\cdot, \theta)$ is w.p.1 $L-$smooth (differentiable with L-Lipschitz gradient) and that $\nabla f_v(\boldsymbol{x}, \theta) \in \mathcal{L}^2$ (square-norm integrable) for at least one $\boldsymbol{x}$. Besides, assume that the objective function is coercive and that the function $\boldsymbol{x} \mapsto \partial_0 g_{\{v,w\}}(\boldsymbol{x}_v, \boldsymbol{x}_w)$ is bounded over compact sets. We also assume that $\sum_{\substack{v<w \\ \{v,w\} \in E}} d(x, \operatorname{dom}(g_{\{v,w\}})) \geq \boldsymbol{d}(x)$ where $\operatorname{dom}(h)$ denotes the domain of function h and d the distance function and $\boldsymbol{d}$ the distance function to $\cap_{\substack{v<w \\ \{v,w\} \in E}} \operatorname{dom}(g_{\{v,w\}})$. Finally, we assume that*

$$\mathbb{E}\left(\|\nabla f_{v,\gamma}(x, \xi)\| + \frac{1}{\gamma}\|\operatorname{prox}_{\gamma g(\cdot,\xi)}(\boldsymbol{x}) - \Pi_{cl(D(s))}(\boldsymbol{x})\|\right)\mu(ds) \leq C(1+|$$

Then, if $\arg\min F + G \neq \emptyset$, for all $\varepsilon > 0$,

$$\limsup_{n \to \infty} \frac{1}{n+1} \sum_{k=0}^{n} \mathbb{P}\left(d(\boldsymbol{u}^{(k),\gamma}, \arg\min(F+G)) > \varepsilon\right) \xrightarrow[\gamma \to 0]{} 0.$$

where we wrote explicitly the dependance of $\boldsymbol{u}^{(k)}$ on γ.

III. Target Localization

We consider an application of underwater target localization using the range and DOA measurements obtained from network of sensors in MCAS system. The target localization problem is based upon least squares estimate, which is indeed a *nonconvex* optimization, thus, rather difficult to find global solution. Despite such a difficulty, efficient methods for exact (global) solutions are proposed in [6,7]. Based up on their idea, we apply the proposed adaptive distributed algorithm described in Section II for tracking a slowly moving target.

A. System Description and Problem Formulation

MACS systems are consist of multiple transmitter and receiver units spatially distributed over a ROI [7,8]. MCAS systems involve transmission and reception of multiple continuous probing sequences [9], thus, each receiver can operate asynchronously. Here, we considered two-dimensional (2D) space, but it can be easily extended to 3D space. Let the MCAS system be equipped with stationary M transmitters and N receivers, and a target moving in the ROI. Let $\boldsymbol{t}_m = [x_m^t, y_m^t]^{\mathrm{T}}$, $\boldsymbol{r}_n = [x_n^r, y_n^r]^{\mathrm{T}}$, and $\boldsymbol{\theta} = [x^\theta, y^\theta]^{\mathrm{T}}$ denote the Cartesian coordinate of the mth transmitter, nth receiver, and the target, respectively, for $m = 1, 2, \cdots, M$, and $n = 1, 2, \cdots, N$; see [7] for detailed description about the placement scheme of the transmitters and receivers in MCAS system. A signal transmitted by the mth transmitter echo back to the nth receiver after propagating the distance: $\rho_{m,n} = \|\boldsymbol{\theta} - \boldsymbol{t}_m\| + \|\boldsymbol{\theta} - \boldsymbol{r}_n\|$. Assume that we have a stream of range $\{\rho_{m,n}^{(k)}\}$ and DOA measurements $\{\varphi_n^{(k)}\}$ sampled at the instances $k \in \mathbb{N}$, which are corrupted by white Gaussian noise. Let receivers be equipped with processing units that form nodes of undirected and connected graph G. At certain instant k, target position estimation [7] is written as:

$$\arg\min_{\boldsymbol{\theta}} \sum_{v=1}^{N} \|\mathbf{B}_v^{(k)} \begin{bmatrix} \boldsymbol{\theta} \\ \|\boldsymbol{\theta} - \boldsymbol{r}_v\| \end{bmatrix} - \boldsymbol{g}_v^{(k)}\|^2 \tag{7}$$

See [7] for the definitions of matrix $\mathbf{B}_v$ and vector $\boldsymbol{g}_v$. As in [6], we assume that $\mathbf{B}_n^{(k)}$ have full column rank so that ${\mathbf{B}_n^{(k)}}^{\mathrm{T}} \mathbf{B}_n^{(k)}$ is nonsingular. One can notice that the problem (7) is non-convex problem in $\boldsymbol{\theta}$ but convex in the vectors $\tilde{\boldsymbol{\theta}}_v = \left[\boldsymbol{\theta}^{\mathrm{T}}, \|\boldsymbol{\theta} - \boldsymbol{r}_v\|\right]^{\mathrm{T}} \in \mathbb{R}^3$ for $v = 1, 2, \cdots, N$.

Thus, as suggested in [7], we use $\tilde{\boldsymbol{\theta}}_v$ to relax the problem (7), and formulate it as distributed optimization problem. Let $\mathcal{X} = \{\tilde{\boldsymbol{\theta}}_1, \tilde{\boldsymbol{\theta}}_2, \cdots, \tilde{\boldsymbol{\theta}}_N\}$ and $\mathcal{C}$ be set of vectors $(\tilde{\boldsymbol{\theta}}_1, \tilde{\boldsymbol{\theta}}_2, \cdots, \tilde{\boldsymbol{\theta}}_N) \in \mathcal{X}$ satisfying the consensus condition: $\forall (n,m) \in \{1,2,\cdots,N\}^2$ $\tilde{\boldsymbol{\theta}}_n(\ell) = \tilde{\boldsymbol{\theta}}_m(\ell)$, for $\ell = 1,2$, where $\boldsymbol{\theta}(\ell)$ represents ℓth element of the vector. Target position estimation from streams of noisy measurements at N different receiver nodes is an instance of the problem (2), which is written as:

$$\arg\min_{\tilde{\boldsymbol{\theta}} \in \mathcal{X}^{\mathsf{V}}} \sum_{n \in \mathsf{V}} \mathbb{E}(f_v(\tilde{\boldsymbol{\theta}}_v, \theta)) \text{ s.t. } (\tilde{\boldsymbol{\theta}}_1, \tilde{\boldsymbol{\theta}}_2, \cdots, \tilde{\boldsymbol{\theta}}_N) \in \mathcal{C} \tag{8}$$

where $f_v(\tilde{\boldsymbol{\theta}}_v, \theta^{(k+1)}) = \frac{1}{2}\|\mathbf{B}_v^{(k+1)}\tilde{\boldsymbol{\theta}}_v - \boldsymbol{g}_v^{(k+1)}\|^2$.

B. Solution

We solve the problem (8) by the distributed adaptive asynchronous D-R algorithm proposed in Section II-B. Here, we consider that the both nodes $v^{(k+1)}$ and $w^{(k+1)}$ do the local estimations and exchange their estimates for consensus step. The iterations of the algorithm writes:

At nodes $v \in \{v^{(k+1)}, w^{(k+1)}\}$ perform:

$$\tilde{\boldsymbol{\theta}}_v^{(k+1)} = \operatorname{prox}_{\gamma, f_v(\tilde{\boldsymbol{\theta}}_v, \theta^{(k+1)})}(\boldsymbol{u}_v^{(k)}) \tag{9}$$

$$\boldsymbol{z}_v^{(k+1)}(\ell) = \begin{cases} \frac{1}{2}\sum_{i \in v}\left(2\tilde{\boldsymbol{\theta}}_i^{(k+1)} - \boldsymbol{u}_i^{(k)}\right)(\ell) & \text{for } \ell = 1,2 \\ \left(2\tilde{\boldsymbol{\theta}}_v^{(k+1)} - \boldsymbol{u}_v^{(k)}\right)(\ell) & \text{for } \ell = 3 \end{cases}$$

$$\boldsymbol{u}_v^{(k+1)} = \boldsymbol{u}_v^{(k)} + \boldsymbol{z}_v^{(k+1)} - \tilde{\boldsymbol{\theta}}_v^{(k+1)}$$

The position of the target is given by the first two elements of $\tilde{\boldsymbol{\theta}}_n$. As pointed out in [6,7], although the problem (8) is convex in $\tilde{\boldsymbol{\theta}}_v$, but solving it can produce only a suboptimal solution to the problem (7) due to the fact that (8) discards the quadratic relationship

$$[\tilde{\boldsymbol{\theta}}_v - \tilde{\boldsymbol{r}}_v]^{\mathrm{T}}\mathbf{S}[\tilde{\boldsymbol{\theta}}_v - \tilde{\boldsymbol{r}}_v] = 0 \tag{10}$$

among the elements of $\tilde{\boldsymbol{\theta}}_v$, where

$$\tilde{\boldsymbol{r}}_v = [\boldsymbol{r}_v^{\mathrm{T}}, 0]^{\mathrm{T}} \text{ and } \mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

Thus, we introduce the above quadratic constraint into (9), and rewrite it as following constrained optimization problems:

$$\tilde{\boldsymbol{\theta}}_v = \arg\min_{\tilde{\boldsymbol{\theta}}_v} \frac{1}{2}\|\mathbf{B}_v\tilde{\boldsymbol{\theta}}_v - \boldsymbol{g}_v\|_2^2 + \frac{1}{2\gamma}\|\tilde{\boldsymbol{\theta}}_v - \boldsymbol{u}_v\|_2^2$$
$$\text{s.t. } [\tilde{\boldsymbol{\theta}}_v - \tilde{\boldsymbol{r}}_v]^{\mathrm{T}}\mathbf{S}[\tilde{\boldsymbol{\theta}}_v - \tilde{\boldsymbol{r}}_v] = 0 \tag{11}$$

where we dropped the index k for sake of notational simplicity. Note that the problem (11) is no more convex in $\tilde{\boldsymbol{\theta}}_v$ since the quadratic constraint is not convex. The problems of these type are called generalized trust region subproblems (GTRS) [10]. To find the global solution of the problem (11), we follow the idea in [6].

C. Numerical Simulation

In our numerical simulation, we considered two transmitters and six receivers, whose positions in 2D Cartesian coordinates are: $\boldsymbol{t}_1 = [0,0]$, $\boldsymbol{t}_2 = [2000, 2000]$, $\boldsymbol{r}_1 = [-1000, -1000]$, $\boldsymbol{r}_2 = [1500, -1000]$, $\boldsymbol{r}_3 = [-1000, 1000]$, $\boldsymbol{r}_4 = [1500, 1000]$, $\boldsymbol{r}_5 = [1500, 2500]$, and $\boldsymbol{r}_t = [2500, 1500]$, respectively (unit of distance is meter). The receivers form nodes of the connected graph G with edges $\mathsf{E} = \{\{1,2\},\{1,3\},\{2,4\},\{3,4\},\{4,5\},\{4,6\}\}$. Let the initial position of the target be $\boldsymbol{\theta}^{(0)} = [500, 500]$ and it is moving in spiral with the target position is given by parametric equation: $x^{(k)} = R^{(k)}\cos(t^{(k)}) + \boldsymbol{\theta}_x^{(0)}$, $y^{(k)} = R^{(k)}\sin(t^{(k)}) + \boldsymbol{\theta}_y^{(0)}$, $R^{(k+1)} = R^{(k)} + \nabla R$, and $t^{(k+1)} = t^{(k)} + \nabla t$ sampled at intervals $\nabla R = 2.5$ and $\nabla t = 0.25$. The range measurements $\rho_{m,n}$ and DOA measurements φ_v are corrupted by Gaussian noise with standard deviations 5 and 0.5, respectively. We choose $\omega_v = 1, v = 1, \cdots, N$.

We compare the tracking ability of the proposed adaptive distributed (both synchronous and asynchronous) algorithms. For both settings, we chose parameter $\gamma = 2E-8$. Figure 1(a) clearly shows the effect of imposing the quadratic constraint (10), thus it is necessary for the accurate solution. Figure 1(b) shows the true track of the target, and the tracks estimated by the two algorithms. Between two sample points of the true track (i.e. between two blue star markers on blue curve), we allowed 50 iterations for both the algorithms, and it is sufficient to track continuously the target with good accuracies. In spite of using only two nodes in estimation at each iteration, the asynchronous algorithm, after certain initial lag, performs almost similar to the synchronous one that involved all six nodes at each iteration. We also observe that when target moves faster (at outer periphery of the spiral), then the two algorithm make larger errors, which suggests that the receivers should sample the measurements at shorter intervals, and should have faster computation capability to do more iterations in between the two samples.

In this paper, we presented an adaptive/stochastic distributed optimization algorithm, which can be easily adapted to both synchronous and asynchronous settings. We performed numerical experiment on localizing slowly moving target, and showed that proposed asynchronous algorithm has similar performance as synchronous one, which is very useful in the applications where one cannot have fusion center.

References

[1] P. Bianchi, "Ergodic convergence of a stochastic proximal point algorithm," *SIAM J. Optim.*, vol. 26, no. 4, pp. 2235–2260, 2016. [Online]. Available: http://dx.doi.org/10.1137/15M1017909

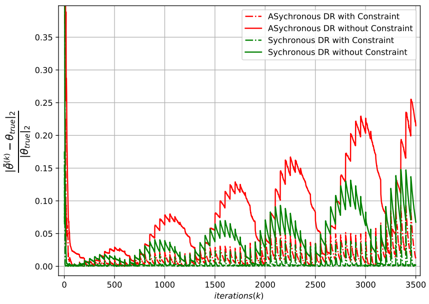
[2] H. J. Kushner and G. G. Yin, *Stochastic approximation and recursive algorithms and applications*, 2nd ed., ser. Applications of Mathematics (New York). Springer-Verlag, New York, 2003, vol. 35, stochastic Modelling and Applied Probability.

[3] F. S. Cattivelli and A. H. Sayed, "Diffusion lms strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

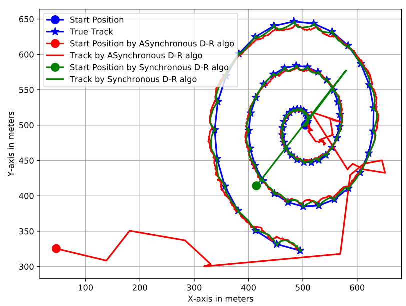
[4] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[5] P. L. Lions and B. Mercier, "Splitting Algorithms for the Sum of Two Nonlinear Operators," *SIAM Journal of Numerical Analysis*, vol. 16, no. 6, pp. 964–979, 1979.

[6] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1770–1778, 2008.



(a) Convergence of solultions against iterations (k): $\bar{\boldsymbol{\theta}}$ represent mean of $\boldsymbol{\theta}_v, v = 1, \cdots, N$, and $\boldsymbol{\theta}_{\text{true}}$ represents true positions of traget.



(b) True and the estimated tracks of the target with the quadratic constraint

Fig. 1. Numerical simulation results on tracking slowly moving target.

[7] J. Liang, L. Xu, J. Li, and P. Stoica, "On designing the transmission and reception of multistatic continuous active sonar systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 285–299, 2014.

[8] F. Ehlers, M. Daun, and M. Ulmke, "System design and fusion techniques for multistatic active sonar," in *OCEANS 2009-EUROPE*. IEEE, 2009, pp. 1–10.

[9] S. Coraluppi, "Multistatic sonar localization," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 964–974, 2006.

[10] J. J. Moré, "Generalizations of the trust region problem," *Optimization methods and Software*, vol. 2, no. 3-4, pp. 189–209, 1993.