An Adaptive Distributed Asynchronous Algorithm with Application to Target Localization

Rahul Mourya*, Pascal Bianchi*, Adil Salim*, 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 introduces a constant step size adaptive algorithm for distributed optimization on a graph. The algorithm is of diffusion-adaptation type and is asynchronous: at every iteration, some randomly selected nodes compute some local variable by means of a proximity operator involving a locally observed random variable, and share these variable with neighbors. The algorithm is built upon a stochastic version of the Douglas-Rachford algorithm. A practical application to target localization using measurements from multistatic continuous active sonar systems is investigated at length.**

***Index Terms*—Adaptive algorithms, stochastic approximation, distributed optimization, proximal operator, target localization.**

I. Introduction

A broadly investigated subject in optimization and signal processing consists in solving the problem $\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}(f(\boldsymbol{x},\theta))$ where $\mathcal{X}$ is an Euclidean space, θ is a random variable (r.v.) and $f(\,.\,,\theta)$ is some random function. We consider the case where the expectation $\mathbb{E}$ cannot be computed, but is 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. In adaptive signal processing, the step size of the algorithm is constant in order to maintain the tracking abilities of the algorithm. In this paper, we propose a *distributed* stochastic algorithm over graphs with constant step size. Consider an undirected and connected graph $\mathsf{G} = (\mathsf{V},\mathsf{E})$ 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:

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

where for every $v\in\mathsf{V}$, θ_v is random variable on some probability space into some measurable space Θ_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 $\Gamma_0(\mathcal{X}\times\mathcal{X})$. 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 ($\boldsymbol{x}_v : v\in\mathsf{V}$). A special case of Problem (1) is given when every function g_e is defined by $g_e(\boldsymbol{x},\boldsymbol{y}) = \iota_{\mathcal{C}}(\boldsymbol{x},\boldsymbol{y})$ where $\iota_{\mathcal{C}}$ is the indicator function of the set $\mathcal{C} := \{(\boldsymbol{x},\boldsymbol{x}) : \boldsymbol{x}\in\mathcal{X}\}$, equal to zero on that set and to $+\infty$ elsewhere. 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 (1) is equivalent to the *consensus problem*:

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

In other words, all nodes are seeking to find a common minimizer of the aggregate cost $\sum_{v\in\mathsf{V}}\mathbb{E}(f_v(\,.\,,\theta_v))$. Compared to (2), the generic formulation (1) is useful to cover the case of total variation regularization ($g_{\{v,w\}}(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|$) of Laplacian regularization ($g_{\{v,w\}}(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|^2$). 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). Similar distributed algorithms are proposed in [2,3]. Our 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 [4], which is a stochastic version of the celebrated Douglas-Rachford algorithm [5] used to minimize the sum of two functions. In its stochastic counterpart [4], the latter are replaced by random functions observed at every iteration of the algorithm. The distributed problem (1) can be seen as a special case of the problem solved the algorithm of [4]. The nature of the randomness is twofold. 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. In order to incorporate asynchronous communications in the algorithm, the idea is to reformulate the second sum in (1) as an expectation over the (random) active edges and then to apply the adaptive Douglas-Rachford algorithm. Finally, we apply our algorithm to the problem of adaptive and distributed target localization.

II. Main Algorithm

A. Adaptive Douglas Rachford Algorithm

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

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

The positive parameter $\gamma > 0$ is regularization parameter that controls the approximation. Let ξ be a random variable defined on some probability space $(\Omega,\mathcal{F},\mathbb{P})$ into an arbitrary

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

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 $f(\boldsymbol{x}, \,.\,)$ is measurable for every x. Consider the problem

$$\min_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}(f(\boldsymbol{x}, \xi)) + \mathbb{E}(g(\boldsymbol{x}, \xi)) \tag{3}$$

where f, g are normal convex integrands. Let $(\xi^{(k)} : k \in \mathbb{N})$ a sequence of i.i.d. copies of ξ. Consider the algorithm of [4]

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

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. The convergence analysis of the algorithm is provided in [4] under the hypothesis of a constant step size. Under some hypotheses, it is proved that the algorithm is stable (in a sense made clear in [4]) and that it converges to the set of solutions to (3) in the doubly asymptotic regime where $k \to \infty$ and $\gamma \to 0$ (see again [4] for details).

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 $\boldsymbol{\theta} = (\theta_v : v \in \mathsf{V})$ and let $(\boldsymbol{\theta}^{(k)} : k \in \mathbb{N})$ be a sequence of i.i.d. 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. The sequence $((v^{(k)}, w^{(k)}) : k \in \mathbb{N})$ is supposed i.i.d. and independent from the sequence $(\boldsymbol{\theta}^{(k)} : k \in \mathbb{N})$. As the graph is connected, Problem (2) is equivalent to

$$\min_{\boldsymbol{x} \in \mathcal{X}^{\mathsf{V}}} \mathbb{E}\left(f_{v^{(1)}}(\boldsymbol{x}_{v^{(1)}}, \theta^{(1)}_{v^{(1)}})\right) + \mathbb{E}\left(\iota_{\mathcal{C}}(\boldsymbol{x}_{v^{(1)}}, \boldsymbol{x}_{w^{(1)}})\right). \tag{4}$$

The application of adaptive Douglas Rachford algorithm to (4) yields following iterations:

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

$$\boldsymbol{x}_v^{(k+1)} = \operatorname{prox}_{\gamma, f_v^{(k+1)}}(\boldsymbol{u}_v^{(k)})$$
$$\boldsymbol{u}_v^{(k+1)} = \frac{1}{2}(\boldsymbol{u}_v^{(k+1)} + \boldsymbol{u}_w^{(k+1)})$$
$$\boldsymbol{u}_w^{(k+1)} = \boldsymbol{x}_v^{(k+1)} + \frac{1}{2}(\boldsymbol{u}_w^{(k+1)} - \boldsymbol{u}_v^{(k+1)}),$$

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

C. Generalization

We generalize the above 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 referred 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 referred 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 $\nu : \Omega \to 2^{\mathsf{V}}$ taking its values in the set of subsets of V. The elements of ν are the computing nodes. We also introduce a random variable $\varepsilon : \Omega \to 2^{\mathsf{E}}$ 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 $v \in \mathsf{V}$, $e \in \mathsf{E}$, the probabilities $p_v := \mathbb{P}(v \in \nu)$ and $q_e := \mathbb{P}(e \in \varepsilon)$ and we assume that the latter are positive. Define the r.v. $\xi := (\theta, \nu, \varepsilon)$ on the space $\Xi := \Theta \times 2^{\mathsf{V}} \times 2^{E}$ where $\Theta := \Theta_1 \times \cdots \times \Theta_N$. Introduce the maps $f, g : \mathcal{X}^{\mathsf{V}} \times \Xi \to (-\infty, +\infty]$ s.t. $f(\boldsymbol{x}, \xi) := \sum_{v \in \nu} p_v^{-1} f_v(\boldsymbol{x}_v, \theta)$ and $g(\boldsymbol{x}, \xi) := \sum_{\substack{v<w \\ \{v,w\} \in \varepsilon}} q^{-1}_{\{v,w\}} g_{\{v,w\}}(\boldsymbol{x}_v, \boldsymbol{x}_w)$, for every $\boldsymbol{x} \in \mathcal{X}^V$. Define $F(\boldsymbol{x}) := \mathbb{E}(f(\boldsymbol{x}, \xi))$ and $G(\boldsymbol{x}) := \mathbb{E}(g(\boldsymbol{x}, \xi))$. Whenever θ and ν are independent, it holds that

$$F(\boldsymbol{x}) = \sum_{v \in \mathsf{V}} \mathbb{E}(f_v(\boldsymbol{x}_v, \theta)), \; G(\boldsymbol{x}) = \sum_{\substack{v<w \\ \{v,w\} \in \mathsf{E}}} g_{\{v,w\}}(\boldsymbol{x}_v, \boldsymbol{x}_w).$$

We apply the adaptive Douglas-Rachford algorithm with the sequence $\xi^{(k)} := (\theta^{(k)}, \nu^{(k)}, \varepsilon^{(k)})$. We define $\mathsf{V}(\varepsilon) := \{v \in \mathsf{V} : \exists w \in \mathsf{V}, \{v, w\} \in \varepsilon\}$. The algorithm writes:

Every computing node $v \in \nu^{(k+1)}$ generates

$$\boldsymbol{x}_v^{(k+1)} = \operatorname{prox}_{\gamma, p_v^{-1}, f_v(\,.\,, \theta^{(k+1)})}(\boldsymbol{u}_v^{(k)}) \tag{5}$$

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

$$(\boldsymbol{z}_v^{(k+1)} : v \in \mathsf{V}(\varepsilon^{(k+1)})) = \arg \min_{\boldsymbol{z} \in \mathcal{X}^{\mathsf{V}(\varepsilon^{(k+1)})}} \sum_{\substack{v<w \\ \{v,w\} \in \varepsilon}} q^{-1}_{\{v,w\}} g_{\{v,w\}}(\boldsymbol{z}_v, \boldsymbol{z}_w) + \frac{1}{2\gamma} \sum_{v \in \mathsf{V}(\varepsilon^{(k+1)})} \|\boldsymbol{z}_v - 2\boldsymbol{x}_v^{(k+1)} + \boldsymbol{u}_v^{(k)}\|^2, \tag{6}$$

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 $v \in \mathsf{V}$ set

$$\boldsymbol{u}_v^{(k+1)} = \boldsymbol{u}_v^{(k)} + \boldsymbol{z}_v^{(k+1)} - \boldsymbol{x}_v^{(k+1)} \tag{7}$$

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 $\mathsf{V}(\varepsilon^{(k+1)})$.

The above algorithm boils down to the adaptive algorithm of Section II-B in the case where $\nu^{(k)} = \{v^{(k)}\}$, $\varepsilon^{(k)} = \{\{v^{(k)}, w^{(k)}\}\}$ and every $g_e = \iota_{\mathcal{C}}$ for all $e \in \mathsf{E}$. The proof of the following theorem is omitted. It consists in checking the assumptions of the main theorem in [4]. We denote by $\mathrm{dom}(h)$ the domain of a function h. If $\mathcal{A}$ is a set, let $d(\,.\,,\mathcal{A})$ be the distance function to $\mathcal{A}$. Set $\boldsymbol{d}(\boldsymbol{x})$ denote the distance of point $\boldsymbol{x} \in \mathcal{X}^{\mathsf{V}}$ to the intersection of the sets $\mathrm{dom}(g_e)$ w.r.t. $e \in \mathsf{E}$.

Assumption II.1. *Assume the following for every $v \in \mathsf{V}$, $e \in \mathsf{E}$. The maps $f_v(\cdot,\theta)$ and g_e are w.p.1 convex. There exists $L > 0$ such that $f_v(\cdot,\theta)$ is w.p.1 differentiable with L-Lipschitz gradient. Moreover, $p_v > 0$, $q_e > 0$.*

Assumption II.2. *There exists $\bar{\boldsymbol{x}} \in \mathcal{X}^{\mathsf{V}}$ s.t. $\nabla f_v(\bar{\boldsymbol{x}}_v,\theta)$ is square-norm integrable for every $v \in \mathsf{V}$. There exists $\alpha > 0$ s.t. $\sum_{e\in E} d(\boldsymbol{x}, \mathrm{dom}(g_e)) \geq \alpha \boldsymbol{d}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}^{\mathsf{V}}$.*

We denote by $h^\gamma : \boldsymbol{x} \mapsto \min_{\boldsymbol{x}\in\mathcal{X}} h(\boldsymbol{y}) + \|\boldsymbol{y} - \boldsymbol{x}\|^2/(2\gamma)$ the Moreau's envelope of a mapping $h \in \Gamma_0(\mathcal{X})$. We denote by $\Pi_e(\boldsymbol{x})$ the projection of a point $\boldsymbol{x} \in \mathcal{X}^{\mathsf{V}}$ onto the closure $\mathrm{dom}(g_e)$. We denote by $D_{\{v,w\}}(\boldsymbol{x})$ the least norm element in $\partial g_{\{v,w\}}(\boldsymbol{x}_v, \boldsymbol{x}_w)$. Define $\mathcal{S} := \arg\min F + G$ the set of solutions. As the algorithm is dependent on the step size γ, we write in the sequel $\boldsymbol{u}^{(k),\gamma}$ instead of $\boldsymbol{u}^{(k)}$.

Theorem II.1. *Let Assumptions II.1-II.2 hold true. Assume that θ and ν be independent. Assume that $F + G$ is coercive and that for every $e \in \mathsf{E}$, D_e is bounded over compact sets. Assume that there exists $C > 0$ s.t. for all $\boldsymbol{x}$,*

$$\sum_{v\in\mathsf{V}} \mathbb{E}\left(\|\nabla f_v^\gamma(\boldsymbol{x}_v, \theta_v)\| + \frac{1}{\gamma} \sum_{e\in\mathsf{E}} \|\operatorname{prox}_{\gamma g_e}(\boldsymbol{x}) - \Pi_e(\boldsymbol{x})\| \right)$$
$$\leq C \left(1 + |\sum_{v\in\mathsf{V}} \mathbb{E}(f_v^\gamma(\boldsymbol{x}_v, \theta_v)) + \sum_{\{v,w\}\in\mathsf{E}} \mathbb{E}(g_{\{v,w\}}^\gamma(\boldsymbol{x}_v, \boldsymbol{x}_w))| \right). \tag{8}$$

If $\mathcal{S} \neq \emptyset$, then for all $\delta > 0$,

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

Note that in the special case of Section II-B, the mappings g_e ($e \in \mathsf{E}$) coincide with indicator functions, and the condition (8) simplifies to a condition on the functions f_v only, which is mild and easily verifiable.

III. Target Localization

We consider an application of underwater target localization using the range and direction-of-arrival (DOA) measurements obtained from network of sensors in multistatic continuous active sonar system (MCAS). 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-B for tracking a slowly moving target.

A. System Description and Problem Formulation

MCAS system consists multiple transmitter and receiver units spatially distributed over a region-of-interest (ROI) [7,8]. MCAS system involves transmission and reception of multiple continuous probing sequences [9], thus, each receiver can operate asynchronously. Here, we considered two-dimensional

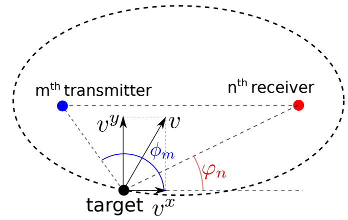


Fig. 1. A generic setting of active sonar [7].

(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 $n = 1, 2, \cdots, N$. A generic scenario of MCAS system is shown in the Fig. 1 with a mth transmitter, a nth receiver, and a moving target; see [7] for detailed description about the placement scheme of the transmitters and receivers in MCAS system. A signal transmitted by the mth transmitter echoes 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 $k \in \mathbb{N}$ that are corrupted by white Gaussian noise. Let receivers be equipped with processing units that form nodes of undirected and connected graph G. At a certain instant k, target position estimation [7] is written as:

$$\arg\min_{\boldsymbol{\theta}} \sum_{v=1}^{N} \|\mathbf{B}_v^{(k)} \begin{bmatrix} \boldsymbol{\theta} \\ \|\boldsymbol{\theta} - \boldsymbol{r}_v\| \end{bmatrix} - \boldsymbol{g}_v^{(k)}\|^2 \tag{9}$$

where

$$\mathbf{B}_v^{(k)} = \begin{bmatrix} 2(\boldsymbol{t}_1 - \boldsymbol{r}_v)^{\mathrm{T}} & -2\rho_{1,v}^{(k)} \\ \vdots & \vdots \\ 2(\boldsymbol{t}_M - \boldsymbol{r}_v)^{\mathrm{T}} & -2\rho_{M,v}^{(k)} \\ \omega_v \;\; 0 & -\omega_v \cos(\varphi_v^{(k)}) \\ 0 \;\; \omega_v & -\omega_v \sin(\varphi_v^{(k)}) \end{bmatrix} \in \mathbb{R}^{(M+2)\times 3},$$

$$\boldsymbol{g}_v^{(k)} = \Big[\|\boldsymbol{t}_1\|^2 - (\rho_{1,v}^{(k)})^2 - \|\boldsymbol{r}_v\|^2, \cdots$$
$$\|\boldsymbol{t}_M\|^2 - (\rho_{M,v}^{(k)})^2 - \|\boldsymbol{r}_v\|^2,\ \omega_v x_v^r,\ \omega_v y_v^r \Big]^{\mathrm{T}} \in \mathbb{R}^{(M+2)}$$

and $\omega_v > 0$ are weights to balance between the coefficients of range and DOA measurements. 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 (9) 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 (9), and formulate it as distributed convex 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:

$$\underset{\tilde{\boldsymbol{\theta}} \in \mathcal{X}^{\mathsf{V}}}{\arg\min} \sum_{n \in \mathsf{V}} \mathbb{E}(f_v(\tilde{\boldsymbol{\theta}}_v, \theta)) \text{ s.t. } (\tilde{\boldsymbol{\theta}}_1, \tilde{\boldsymbol{\theta}}_2, \cdots, \tilde{\boldsymbol{\theta}}_N) \in \mathcal{C} \quad (10)$$

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 (10) 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:

$$\tilde{\boldsymbol{\theta}}_v^{(k+1)} = \operatorname{prox}_{\gamma, f_v(\tilde{\boldsymbol{\theta}}_v, \theta^{(k+1)})}(\boldsymbol{u}_v^{(k)}) \quad (11)$$

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

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

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 (10) is convex in $\tilde{\boldsymbol{\theta}}_v$, but solving it can produce only a suboptimal solution to the problem (9) due to the fact that (10) discards the quadratic relationship

$$[\tilde{\boldsymbol{\theta}}_v - \tilde{\boldsymbol{r}}_v]^{\mathrm{T}}\mathbf{S}[\tilde{\boldsymbol{\theta}}_v - \tilde{\boldsymbol{r}}_v] = 0 \quad (12)$$

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

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

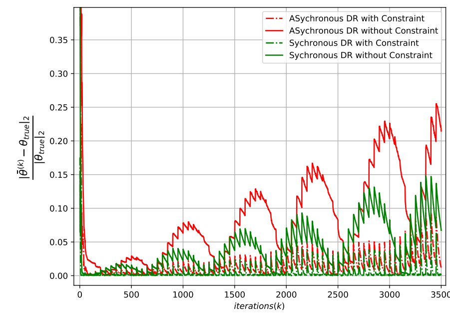
Thus, we introduce the above quadratic constraint into (11), and rewrite it as following constrained optimization problems:

$$\tilde{\boldsymbol{\theta}}_v^* = \underset{\tilde{\boldsymbol{\theta}}_v}{\arg\min} \frac{1}{2}\|\mathbf{B}_v\tilde{\boldsymbol{\theta}}_v - \boldsymbol{g}_v\|_2^2 + \frac{1}{2\gamma}\|\tilde{\boldsymbol{\theta}}_v - \boldsymbol{u}_v\|_2^2$$
$$\text{s.t. } [\tilde{\boldsymbol{\theta}}_v - \tilde{\boldsymbol{r}}_v]^{\mathrm{T}}\mathbf{S}[\tilde{\boldsymbol{\theta}}_v - \tilde{\boldsymbol{r}}_v] = 0 \quad (13)$$

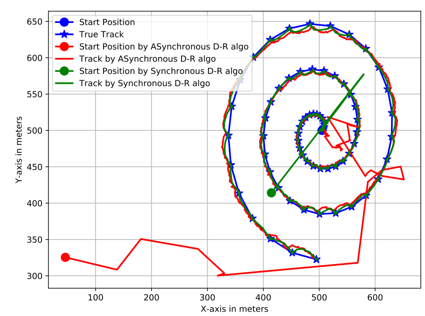
where we dropped the index k for sake of notational simplicity. Note that the problem (13) is no more convex in $\tilde{\boldsymbol{\theta}}_v$ since the quadratic constraint is not convex. The problems of these type are called generalized trust region subproblems (GTRS) [10]. To find the global solution of the problem (13), 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



(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. 2. Numerical simulation results on tracking slowly moving target.

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 2(a) clearly shows the effect of imposing the quadratic constraint (12), thus it is necessary for the accurate solution. Figure 2(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.

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] 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.

[3] 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.

[4] P. Bianchi, W. Hachem, and A. Salim, "A constant step stochastic Douglas Rachford algorithm," Telecom ParisTech, Paris, Tech. Rep., 2017. [Online]. Available: https://github.com/mouryarahul/TechReport/blob/master/TechReport.pdf

[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.

[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.