IEEE Control Systems Letters paper presented at
2023 American Control Conference (ACC)
San Diego, CA, USA. May 31 - June 2, 2023

Distributed Target Tracking in Multi-Agent Networks via Sequential Quadratic Alternating Direction Method of Multipliers

Ola Shorinwa[1] and Mac Schwager[2]

Abstract— **We present a distributed algorithm for multi-agent target tracking, posed as a maximum a-posteriori (MAP) optimization problem. MAP estimation is, in general, a non-convex optimization that depends on each agent's local observation of the target, necessitating a distributed algorithm. In our algorithm, each agent solves a series of local optimization problems to estimate the target's trajectory, while communicating with its one-hop neighbors over a communication network. The agents do not communicate their raw observations, which may be high dimensional (e.g., images), and they do not rely on a central coordinating node or leader, minimizing the communication bandwidth requirements of our approach. We utilize the sequential quadratic programming (SQP) paradigm, with distributed computation of the ensuing sub-problems achieved via the consensus alternating direction method of multipliers (C-ADMM). We empirically demonstrate faster convergence of our algorithm to a locally optimal solution compared to other distributed methods. In addition, our algorithm achieves about the same communication overhead as the best competing distributed algorithm.**

I. Introduction

Multi-agent target tracking problems arise in many robotics applications, including, but not limited to, environmental monitoring, persistent surveillance, industrial manufacturing, and autonomous driving. In many of these problems, estimating the trajectory of the target is critical to the safe operation of autonomous agents. Although filtering methods have been applied to target tracking problems, optimization-based approaches have emerged as particularly effective methods, due to their generality and flexibility in incorporating constraints. These methods formulate the target tracking problem as an optimization problem, which depends on the local set of observations collected by all participating agents. The resulting optimization problem can be solved centrally by collating all the local information at a fusion node; however, this approach poses notable communication and computation challenges, especially in problems with limited access to high-performance computation and communication resources. Distributed approaches address this challenge by allowing each agent to solve the optimization problem locally without collating the local observations.

In this paper, we present Sequential Quadratic Alternating direction method of multipliers for Target Tracking (SQuATT), a distributed algorithm for non-convex optimization-based target tracking problems, where each agent solves a series of optimization problems locally to estimate the trajectory of a target. Each agent communicates with its one-hop neighbors over a point-to-point communication network, without relying on a central station, making our algorithm useful in problems involving agents operating in areas with limited access to existing communication infrastructure. Our method can be seen as a generalization to the non-linear, non-convex case of the prior work [1] for affine target dynamics and measurements.

Our algorithm is based on the Sequential Quadratic Programming (SQP) paradigm [2] and utilizes the Consensus Alternating Direction Method of Multipliers (C-ADMM) [3] to achieve distributed computations. At each iteration of our algorithm, each agent formulates a quadratic approximation of the underlying non-convex target tracking problem which is solved via C-ADMM. Our algorithm is amenable to problems with local constraints, allowing the incorporation of prior knowledge as constraints in target tracking problems, which can generally improve tracking performance. Our algorithm converges about three times faster to a locally optimal solution of the non-convex target tracking problem than the best competing distributed algorithm in simulation, using about the same communication budget as the best communication-efficient algorithm.

II. Related Work

Several approaches have been developed for collaborative target tracking by a group of agents, many of which were derived from well-known state estimation filters, such as Kalman, Information, and particle filters [4], [5], [6]. In filtering-based methods, each agent maintains a decentralized filter to estimate the trajectory of a target and shares information from its filters with other agents to improve its estimate. These approaches typically suffer from correlations between measurements during the data fusion process, which must be accounted for to improve the estimation accuracy [7]. Some methods have addressed this challenge by formulating the target tracking problem as a maximum likelihood estimation (MLE) or maximum a-posteriori (MAP) problem over the entire set of observations of the target collected by all agents [8], [9], [10]. Moreover, optimization-based approaches provide greater flexibility, allowing for the consideration of prior knowledge during the estimation process and the incorporation of constraints. However, the resulting MLE or MAP optimization problems are non-convex, in general. In addition, solving the resulting optimization problem requires collation of the local observations from all agents,

*This work was supported in part by NSF NRI awards 1830402 and 1925030 and ONR grant N00014-18-1-2830.

[1]O. Shorinwa is with the Department of Mechanical Engineering, Stanford University, Stanford, CA, USA. shorinwa@stanford.edu.

[2]M. Schwager is with the Department of Aeronautics and Astronautics Engineering, Stanford University, Stanford, CA, USA. schwager@stanford.edu.

if a centralized approach is utilized, which poses notable computation and communication challenges, especially in the presence of a bandwidth-constrained communication networks. Further, the central node in centralized approaches represents a single point of failure. As a result, distributed methods for solving the non-convex optimization problem are desirable for improved communication efficiency and robustness.

Gradient descent methods and other accelerated variants have been applied to the minimization of non-convex functions [11], [12], although mostly in a centralized manner. However, many distributed gradient descent methods exist for unconstrained convex optimization [13], [14], which utilize a linear consensus or push-sum technique to achieve agreement among the agents. A few extensions of distributed gradient descent to non-convex optimization exist. For example, the push-sum technique has been applied to non-convex problems by using the gradient direction obtained from a stochastic approximation of the objective function [15]. In other consensus methods, each agent performs sequential local convex approximation of the non-convex problem, followed by linear consensus procedures on the local optimization variables and gradients [16], [17].

In contrast to gradient descent methods, sequential convex programming (SCP) methods replace the non-convex objective and constraint functions with convex approximations (often quadratic in form) iteratively, and solve the resulting convex problems until some convergence criterion is satisfied [2], [18]. In separable problems where the objective and/or constraint functions consist of a sum of individual components, existing research effort has focused on deriving distributed procedures for solving the ensuing convex sub-problems arising in sequential convex programming. Some of these works distribute the resulting convex sub-problems using dual decomposition [19]. Other methods solve the convex sub-problems using proximal decomposition [20], which improves the convergence rate of the dual decomposition method, and the alternating direction method of multipliers [21] .

Augmented Lagrangian methods, another class of methods, solve non-convex optimization problems by finding minimizers of the augmented Lagrangian, which consists of a reformulation of the problem constraints with a quadratic penalty term on the satisfaction of the constraint functions [22]. Formation of the augmented Lagrangian renders the problem non-separable even when the original problem is separable, hindering distributed solutions of the non-convex problem. To overcome non-separability of the primal updates in augmented Lagrangian methods, some distributed approaches solve the primal update problem using block-coordinate decent [23], while others combine augmented Lagrangian methods with sequential quadratic programming to achieve more fully-distributed computations [24]. Although these methods provide a greater level of distributed computations, some resulting operations in the procedure remain coupled, requiring a central fusion node for their computation. The alternating direction method of multipliers (ADMM) resolves many of the challenges of augmented Lagrangian methods by solving the primal update problem in series which retains the separability of the original problem. In non-convex problems, some distributed ADMM methods employ sequential convex programming [25], [26] to compute a minimizer of the non-convex primal update sub-problem. In contrast to all these methods, our algorithm is fully-distributed, enabling each agent to compute a stationary point of the non-convex problem locally.

This paper is organized as follows: In Section III, we present the non-convex target tracking optimization problem. We derive our method in Section IV and analyze its convergence properties in Section V. In Section VI, we examine the performance of our algorithm in a non-convex target tracking problem, showing its faster convergence rates compared to existing methods. We conclude the paper in Section VII. In the subsequent discussion, we denote the identity matrix as $I_n \in \mathbb{R}^{n\times n}$ and the matrix-weighted squared-norm $x^T M x$ as $\|x\|_M^2$, given a positive definite matrix M. Further, we denote the gradient of a function f as ∇f, and its Hessian with respect to x by $\nabla_{xx} f$. We denote the set of strictly positive real numbers as $\mathbb{R}_{++}$.

III. Problem Formulation

We consider the target tracking problem with a group of N agents estimating the trajectory of a dynamic target from observations of the target collected by onboard sensors. In this work, we consider maximum a-posteriori (MAP) optimization problems, solved by the group of agents collectively to compute a trajectory that maximizes the posterior distribution of the trajectory of the target, given the set of observations of all agents. We consider the dynamics model of the target described by

$$x_{t+1} = g(x_t, u_t) + v_t, \tag{1}$$

where $x_t \in \mathbb{R}^{\bar{n}}$ denotes the state of the target at time t, $g_i : \mathbb{R}^{\bar{n}} \times \mathbb{R}^{\hat{n}} \to \mathbb{R}^{\bar{n}}$ denotes the nonlinear dynamics function of agent i, $u_t \in \mathbb{R}^{\hat{n}}$ denotes the control input of the target at time t, which may not be available to any agent, and $v_t \in \mathbb{R}^{\bar{n}}$ denotes the process noise. Since u_t is not accessible to each agent generally, we set the value of u_t in g at its nominal value in the subsequent discussion. We account for the effects of the discrepancy between the nominal and actual value of u_t in the process noise v_t. In addition, we describe the measurement model of agent i by

$$y_{i,t} = h_i(x_t) + \omega_{i,t}, \tag{2}$$

where $y_{i,t} \in \mathbb{R}^{\bar{p}_i}$ denotes the measurement obtained by agent i at time t, $h_i : \mathbb{R}^{\bar{n}} \to \mathbb{R}^{\bar{p}_i}$ denotes the nonlinear measurement function of agent i, and $\omega_{i,t} \in \mathbb{R}^{\bar{p}_i}$ denotes the measurement noise.

The form of the objective function of the MAP optimization problem depends on the modeling assumptions made on the probability distributions of the measurement noise and process noise. To simplify the exposition in this work, we model the measurement noise and process noise as zero-mean Gaussian white noise processes and assume that the initial distribution of the target's trajectory is normally distributed. However, we do admit nonlinear dynamic and measurement models, and

we do not assume the posterior over the state is Gaussian. With these assumptions, the objective function of the MAP optimization problem takes the form

$$f_i(x) = \sum_{t=0}^{T-1} \|x_{t+1} - g(x_t)\|^2_{P^{-1}} + \sum_{t=0}^{T} \|y_{i,t} - h_i(x_t)\|^2_{Q_i^{-1}} + \|x_0 - \mu\|^2_{L^{-1}} \ , \tag{3}$$

which we obtain by taking the logarithm of the posterior distribution, where μ denotes the mean of the prior probability distribution of the initial state of the target and $x = \left[x_0^\mathsf{T}, \cdots, x_T^\mathsf{T}\right]^\mathsf{T} \in \mathbb{R}^n$. In (3), we denote the covariance of the measurement noise of agent i and the prior probability distribution as $Q_i \in \mathbb{R}^{\bar{p}_i \times \bar{p}_i}$ and $L \in \mathbb{R}^{\bar{n} \times \bar{n}}$, respectively, and $P = N\bar{P}$, where $\bar{P} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ denotes the covariance of the target's process noise. Note that the objective function in (3) consists of terms relating to the dynamics of the target, the sets of measurement collected by each agent, and the probability distribution of the target's initial state. We note that only agent i has access to its set of observations $y_{i,t}$ and its measurement model $h_i(\cdot)$ and measurement noise covariance Q_i.

The resulting centralized MAP optimization problem incorporating information from all agents is given by

$$\begin{aligned} &\underset{x}{\text{minimize}} \ \sum_{i=1}^{N} f_i(x) \\ &\text{subject to } \phi_i(x) \leq 0 \quad i = 1, \cdots, N \end{aligned} \tag{4}$$

where $x \in \mathbb{R}^n$ denotes the trajectory of the target over a duration of length T, $f_i : \mathbb{R}^n \to \mathbb{R}$ denotes the objective function of agent i, and $\phi_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$ denotes the constraint function of agent i. Many MAP target tracking problems do not include constraints; however, for generality, we incorporate constraints into the MAP optimization problem in (4). To simplify the discussion, we do not explicitly include equality constraints; however, equality constraints can be handled directly, or encoded with two inequality constraints. We focus on MAP optimization problems with non-convex objective and constraint functions, f_i and ϕ_i, respectively, and note that f_i and ϕ_i are accessible to agent i only.

Aggregating the local information accessible to each agent at a central computing station (or even at each individual agent) could be challenging in general, with significant computational and communication overhead, highlighting the need for distributed approaches for solving the optimization problem (4), which we discuss in the following section.

Communication Graph

We represent the agents as nodes in an undirected graph $\mathcal{G}$ described by a set of vertices $\mathcal{V} = \{1, \cdots, N\}$ and a set of edges $\mathcal{E}$ with edge $(i, j) \in \mathcal{E}$ if agents i and j share a communication link. In addition, we define the neighbor set of agent i as $\mathcal{N}_i$, containing all agents which can communicate with agent i. We assume the communication graph $\mathcal{G}$ is connected, i.e., a path exists in $\mathcal{G}$ for every pair of agents in $\mathcal{V}$.

IV. Distributed Optimization

In this section, we derive a distributed optimization algorithm for non-convex optimization-based target tracking problems by multiple sensor-equipped agents, where each agent solves a series of local quadratic programs to obtain an estimate of the target's trajectory. Our algorithm is derived using a sequential quadratic programming (SQP) approach, which involves solving for an estimate of the solution of the non-convex MAP optimization problem (4) from its quadratic approximation and iterating over this process using the updated estimate of the solution. Considering that no single agent has access to all the problem data, the ensuing joint quadratic program (QP) cannot be solved by each agent independently. Consequently, we derive a distributed algorithm by utilizing the consensus alternating direction method of multipliers (C-ADMM) to solve the ensuing quadratic programs. Before proceeding with its derivation, we introduce the Karush-Kuhn-Tucker (KKT) conditions for the optimization problem and state some assumptions on the objective and constraint functions. The Lagrangian of the optimization problem (4) is given by

$$\mathcal{L}(x, \nu) = \sum_{i=1}^{N} f_i(x) + \sum_{i=1}^{N} \nu_i^\mathsf{T} \phi_i(x) \tag{5}$$

where $\nu_i \in \mathbb{R}^{m_i}$ denotes the Lagrange multiplier for the inequality constraint in (4). The KKT conditions for the optimization problem (4) are

C.1 Stationarity Condition

$$0 \in \partial \left(\sum_{i=1}^{N} f_i(x) + \sum_{i=1}^{N} \nu_i^\mathsf{T} \phi_i(x) \right), \tag{6}$$

C.2 Complementary Slackness

$$\nu_i^\mathsf{T} \phi_i(x) = 0 \quad \forall i \in \mathcal{V}, \tag{7}$$

C.3 Primal Feasibility

$$\phi_i(x) \leq 0 \quad \forall i \in \mathcal{V}, \tag{8}$$

C.4 Dual Feasibility

$$\nu_i \geq 0 \quad \forall i \in \mathcal{V}. \tag{9}$$

The KKT conditions are necessary conditions for optimality when strong duality holds, i.e., when the Lagrangian in (5) has a saddle-point.

We denote the index set of active constraints of agent i at x by $\mathcal{A}_i(x)$, i.e., $\mathcal{A}_i(x) = \{r \in \{1, \cdots, m_i\} \mid \phi_{i,r}(x) = 0\}$, and the Jacobian of the active constraints of agent i by $C(x)$, where $C(x) = [\nabla \phi_{i,r}(x), \forall r \in \mathcal{A}_i(x), \forall i \in \mathcal{V}]^\mathsf{T}$. Now, we make the following assumptions on the objective and constraint functions.

Assumption 1: The objective function f_i and constraint function ϕ_i are twice continuously differentiable.

Assumption 2: The Linear Independence Constraint Qualification (LICQ) condition is satisfied at a local minimum $x^\star$, i.e., the set of active constraint gradients $\{\nabla \phi_{i,r}(x), r \in \mathcal{A}_i(x)\}$ is linearly independent.

Assumption 3: The Hessian of the Lagrangian with respect to x, $\nabla_{xx}\mathcal{L}$, is non-singular at a local minimum $x^\star$ with an associated optimal multiplier $\nu^\star$, and further, $\nabla_{xx}\mathcal{L}$ is positive definite for all $d \in \mathbb{R}^n$ in the null space of $C(x^\star)$, i.e.,

$$d^T \nabla_{xx}\mathcal{L}(x^\star, \nu^\star) d > 0, \tag{10}$$

for all $d \neq 0$ such that $C(x^\star)d = 0$.

Assumption 4: Strict complementary slackness is satisfied at a local minimum $x^\star \in \mathbb{R}^n$, i.e.,

$$\begin{aligned} \phi_{i,r}(x^\star)\nu^\star_{i,r} = 0, \quad 1 \leq r \leq m_i, \\ \nu^\star_{i,r} > 0, \quad r \in \mathcal{A}_i(x^\star). \end{aligned} \tag{11}$$

We require these standard assumptions to simplify the convergence analysis of our distributed algorithm. Given that $x^\star$ satisfies the KKT conditions, Assumption 3 indicates that $x^\star$ is an isolated local minimum. Moreover, satisfaction of the LICQ at $x^\star$ implies that the KKT conditions are satisfied for some vector of Lagrange multipliers $\nu^\star$. If the strict complementarity slackness condition and the LICQ condition hold at $x^\star$, the Lagrange multiplier that satisfies the KKT conditions is unique.

In the following subsections, we describe the SQP procedure and the subsequent distributed procedure for solving the resulting quadratic program, which constitute the two phases of our algorithm.

A. Sequential Quadratic Programming

In the first phase of our distributed algorithm, we formulate a quadratic model of the MAP optimization problem (4), noting the existence of many efficient solvers for quadratic programs, which enable us to solve the resulting quadratic programs. Moreover, since our algorithm involves solving a series of closely-related QPs, we can leverage information from previously-solved QPs to speed up the solution of subsequent QPs, through a process referred to as a warm-start or hot-start, a functionality provided by many existing QP solvers. We replace the non-convex objective function in (4) with a quadratic function and linearize the non-convex constraint functions, yielding the quadratic program

$$\begin{aligned} \underset{d}{\text{minimize}} \ & \sum_{i=1}^{N} \nabla f_i(x^k)^\mathsf{T} d + \frac{1}{2} d^\mathsf{T} \nabla_{xx}\mathcal{L}_i(x^k, \nu_i^k) d \\ \text{subject to } & \nabla \phi_i(x^k)^\mathsf{T} d + \phi_i(x^k) \leq 0 \quad i = 1, \cdots, N \end{aligned} \tag{12}$$

where $d = x - x^k$ and the objective function is related to the second-order Taylor series expansion of the Lagrangian, with $\mathcal{L}_i(x, \nu_i) = f_i(x) + \nu_i^\mathsf{T}\phi_i(x)$. We assume that the QP in (12) has a non-empty feasible set, which occurs with a good initialization of x^0. In addition, we note that $\nabla_{xx}\mathcal{L}_i(x^k, \nu_i^k)$ has to be positive definite on the null space of $C(x^k)$ to guarantee the existence of a solution for the QP in (12). In many cases, $\nabla_{xx}\mathcal{L}_i(x^k, \nu_i^k)$ may not be positive definite. Consequently, we do not utilize the exact Hessian of the Lagrangian; rather, we use a positive definite approximation, denoted by $H_{i,k}$, which can be obtained through finite differences using the rank-two Powell-Symmetric-Broyden (PSB), Broyden-Fletcher-Goldfarb-Shanno (BFGS), or SALSA-SQP update schemes [27], [28]. Specifically, in this work, we use the BFGS update scheme to approximate $\nabla_{xx}\mathcal{L}_i$, which is given by

$$H_{i,k+1} = H_{i,k} - \frac{H_{i,k} s s^\mathsf{T} H_{i,k}}{s^\mathsf{T} H_{i,k} s} + \frac{y y^\mathsf{T}}{y^\mathsf{T} s} \tag{13}$$

where

$$s = x^{k+1} - x^k, \quad y = \nabla \mathcal{L}_i(x^{k+1}, u^{k+1}) - \nabla \mathcal{L}_i(x^k, u^{k+1}).$$

We note that the BFGS update scheme guarantees positive definiteness of $\{H_{i,k}\}$ provided that $H_{i,0} > 0$ and $y^\mathsf{T} s > 0$; hence, a well-defined solution exists for the QP in (12). In practice, we compute an approximation of the inverse of the Lagrangian, i.e., $H_{i,k}^{-1}$, and utilize a limited-memory BFGS (L-BFGS) in target tracking problems where T is large. L-BFGS avoids direct computation of $H_{i,k}$ by storing the last γ pairs of updates (y, s), defined in (13), which are used to compute matrix-vector products involving $H_{i,k}$, leading to a lower memory overhead.

B. Distributed Quadratic Programming

The second phase of our algorithm involves solving the quadratic program in (12) in a distributed fashion, where each agent communicates only with its one-hop neighbors. We derive a distributed procedure for solving (12) from [3]. We assign a local copy of x to each agent and enforce agreement between these local copies to arrive at a globally consistent estimate, yielding the quadratic program

$$\begin{aligned} \underset{\{x_i, \ \forall i \in \mathcal{V}\}}{\text{minimize}} \ & \sum_{i=1}^{N} \Big(\nabla f_i(x_i^k)^\mathsf{T}(x_i - x_i^k) \\ & \qquad + \frac{1}{2}(x_i - x_i^k)^\mathsf{T} H_{i,k}(x_i - x_i^k) \Big) \\ \text{subject to } & \nabla \phi_i(x_i^k)^\mathsf{T}(x_i - x_i^k) + \phi_i(x_i^k) \leq 0 \quad \forall i \in \mathcal{V} \\ & x_i = x_j \quad \forall j \in \mathcal{N}_i, \forall i \in \mathcal{V}, \end{aligned} \tag{14}$$

where x_i denotes the local copy assigned to agent i. In the subsequent discussion, we denote the concatenation of all the copies of x by $\boldsymbol{x} = \{x_i, \ \forall i \in \mathcal{V}\}$. We note the equivalence between (12) and (14) provided the communication graph is connected and $x^k = x_i^k, \ \forall i \in \mathcal{V}$. In ADMM, each agent updates its primal variables as the minimizer of the augmented Lagrangian of (14) at each iteration, using its dual variables at the previous iteration, and subsequently updates its dual variables through dual ascent on the augmented Lagrangian. We introduce the auxiliary variable c in the consensus, equality constraints in (14), resulting in the set of constraints $x_i = c_{ij}$ and $x_j = c_{ij}$, and derive the augmented Lagrangian $\mathcal{L}_a$ given by

$$\begin{aligned} \mathcal{L}_a(\boldsymbol{x}, c, u, w) = & \sum_{i=1}^{N} g_{i,k}(x_i) \\ & + \sum_{j \in \mathcal{N}_i} \Big(u_{ij}^\mathsf{T}(x_i - c_{ij}) + w_{ij}^\mathsf{T}(x_j - c_{ij}) \\ & \qquad + \frac{\rho}{2}\big(\|x_i - c_{ij}\|_2^2 + \|x_j - c_{ij}\|_2^2 \big) \Big), \end{aligned} \tag{15}$$

where $u_{ij} \in \mathbb{R}^n$ and $w_{ij} \in \mathbb{R}^n$ denote dual variables for the consensus constraints between agents i and j, with $u = [u_{ij}^\mathsf{T},\ \forall (i,j) \in \mathcal{E}]^\mathsf{T}$, $w = [w_{ij}^\mathsf{T},\ \forall (i,j) \in \mathcal{E}]^\mathsf{T}$, and $c = [c_{ij}^\mathsf{T},\ \forall (i,j) \in \mathcal{E}]^\mathsf{T}$. The augmented Lagrangian includes quadratic penalty terms on the violation of the consensus constraints, with the parameter ρ weighting the contribution of these terms to $\mathcal{L}_a$. We denote the extended-value objective function of (14) by

$$g_{i,k}(x_i) = \begin{cases} \bar{g}_{i,k}(x_i) & \text{if } \nabla\phi_i(x_i^k)^\mathsf{T}(x_i - x_i^k) + \phi_i(x_i^k) \leq 0, \\ \infty & \text{otherwise,} \end{cases} \tag{16}$$

where

$$\bar{g}_{i,k}(x_i) = \nabla f_i(x_i^k)^\mathsf{T}(x_i - x_i^k) + \frac{1}{2}(x_i - x_i^k)^\mathsf{T} H_{i,k}(x_i - x_i^k).$$

Note that $\mathcal{L}_a$ is quadratic and strongly convex in c (for $\rho > 0$). As a result, the update procedure for c can be solved in closed-form, yielding

$$\hat{c}_{ij}^{r+1} = \frac{1}{2}\sum_{j \in \mathcal{N}_i} \left(\hat{x}_i^{r+1} + \hat{x}_j^{r+1}\right) \tag{17}$$

at iteration r, by initializing u and w to zero, using the updated values of x computed by agents i and j, denoted by $\hat{x}_i$ and $\hat{x}_j$, respectively. We note that each agent updates its local copy of x, prior to updating c (refer to [3] for more details on the derivation of the update procedures). Similarly, simplifying the dual ascent steps for updating the dual variables results in

$$\hat{q}_i^{r+1} = \hat{q}_i^r + \rho \sum_{j \in \mathcal{N}_i} \left(\hat{x}_i^{r+1} - \hat{x}_j^{r+1}\right) \tag{18}$$

at iteration r, where $\hat{q}_i$ is defined in terms of u and v, with $\hat{q}_i = \sum_{j \in \mathcal{N}_i}(u_{ij} - u_{ji})$.

Each agent updates its local x variable as the minimizer of the quadratic program

$$\begin{aligned} \underset{x_i}{\text{minimize}}\ & \nabla f_i(x_i^k)^\mathsf{T}(x_i - x_i^k) + \frac{1}{2}(x_i - x_i^k)^\mathsf{T} H_{i,k}(x_i - x_i^k) \\ & + \hat{q}_i^{r\mathsf{T}} x_i + \rho \sum_{j \in \mathcal{N}_i} \left\| x_i - 0.5\left(\hat{x}_i^r + \hat{x}_j^r\right)\right\|_2^2 \\ \text{subject to } & \nabla\phi_i(x_i^k)^\mathsf{T}(x_i - x_i^k) + \phi_i(x_i^k) \leq 0, \end{aligned} \tag{19}$$

using its neighbor's previous iterates. We further simplify the objective function of the QP in (19) to obtain

$$\begin{aligned} \underset{x_i}{\text{minimize}}\ & x_i^\mathsf{T}\left(\nabla f_i(x_i^k) + \hat{q}_i^r - \rho \sum_{j \in \mathcal{N}_i}(\hat{x}_i^r + \hat{x}_j^r)\right) \\ & + \frac{1}{2} x_i^\mathsf{T}\left(H_{i,k} + 2\rho|\mathcal{N}_i| I_n\right) x_i \\ \text{subject to } & \nabla\phi_i(x_i^k)^\mathsf{T}(x_i - x_i^k) + \phi_i(x_i^k) \leq 0, \end{aligned} \tag{20}$$

which is solved by agent i.

Algorithm 1 summarizes our distributed algorithm for non-convex target tracking problems. In the *ConvexApproximation* method, each agent computes the gradient of its local objective and constraint functions, along with an approximation of the Hessian $\nabla_{xx}\mathcal{L}_i$, to create a quadratic model of (4). We assume that a good initialization of x is chosen by each agent, which is utilized in creating the quadratic model of (4). To solve the QP in (14), agent i initializes $\hat{x}_i$ as the minimizer of $g_{i,k}$, with $\hat{q}_i^0$ given by the value of $\hat{q}_i$ at the previous execution of the *Smoother* method. Our algorithm requires each agent to share only its local x-iterate with its neighbors after solving the quadratic program in (20). Each agent does not share its local set of observations of the target and the data associated with its objective and constraint functions. Our algorithm requires synchronization of the update procedures within the *Smoother* method across all agents. We note, however, that our algorithm can be extended to the asynchronous setting, by utilizing asynchronous variants of C-ADMM.

Algorithm 1: Sequential Quadratic Alternating direction method of multipliers for Target Tracking (SQuATT)

```
Initialization: ∀i ∈ V
  k ← 0
  q_i^0 ← 0, x_i^0 ∈ R^n.
do in parallel ∀i ∈ V
  g_{i,k}(x_i) ← ConvexApproximation(x_i^k)
  (x_i^{k+1}, q_i^{k+1}) ← Smoother(g_{i,k}, x_i^k, q_i^k)
  k ← k + 1
while stopping criterion not met;
```

Function Smoother($g_{i,k}, x_i^k, q_i^k$)

```
Initialization:
  r ← 0
  q̂_i^0 ← q_i^k
  x̂_i^0 ← argminimize_{x_i} g_{i,k}(x_i)
do in parallel ∀i ∈ V
  x̂_i^{r+1} ← Procedure (20)
  Agent i communicates x̂_i^{r+1} to its neighbors.
  q̂_i^{r+1} ← Procedure (18)
  r ← r + 1
while not converged or stopping criterion is not met;
return (x̂_i^r, q̂_i^r)
```

Remark 1: We note that the solution computed by each agent represents the mode of the posterior probability distribution of the target's trajectory. Moreover, an approximate covariance Σ of this estimate can be computed from the inverse of the Hessian of the Lagrangian of (4) or its approximation, given by

$$\Sigma_k^{-1} = \sum_{i=1}^{N} H_{i,k}, \tag{21}$$

at iteration k, where $\Sigma_{i,k}^{-1} = H_{i,k}$ represents the local information matrix available to each agent. As the algorithm

proceeds, the agents obtain better approximations of the mode and covariance of the posterior probability distribution of the target's trajectory. Further, computing the information matrix Σ_k^{-1} in (21) can be useful for the agents to quantify their certainty in their trajectory estimates. However, this requires a distributed consensus procedure on the approximate Hessian computed locally by each agent to aggregate information from all agents, such as the linear consensus protocol or the push sum technique [29], [30]. In addition, agent i computes $H_{i,k}$ during the execution of our algorithm. As a result, computing the summands in (21) does not incur any additional computational cost. Notably, the ℓ_2-norm of the local information matrix of each agent is not greater than the composite information matrix Σ_k^{-1}. Consequently, each agent does not become overconfident in its estimate of the target's trajectory if it utilizes its local information matrix to approximate the covariance. Hence, if necessary, the consensus procedure required to compute (21) can be omitted, to avoid additional computational overhead.

V. Convergence

In this section, we analyze the convergence properties of our algorithm, beginning with the second phase of our algorithm, which involves computing a solution for the quadratic program in (14).

Theorem 1: The iterate $\hat{x}_i^r$ converges to the optimal solution of the QP in (14), $\forall i \in \mathcal{V}$. Moreover, all the agents reach consensus, with $x_i = x$, $\forall i \in \mathcal{V}$.

Proof: Generally, convergence of ADMM requires that the sub-problems arising in the update procedures are solvable and that the Lagrangian of the associated optimization problem has a saddle-point. We note that the quadratic program in (14) satisfies Slater's condition [31], provided that the QP does not have non-empty feasible set. As a result, strong duality holds, and the Lagrangian of (14) has a saddle-point. In addition, positive definiteness of the sequence $\{H_{i,k}\}$ guarantees that the QP (14) and the resulting update procedures have well-defined solutions. For a complete proof, refer to [32]. Further, the dual residual decays to zero, indicating that the iterates of all agents converge to the same solution. ∎

In the subsequent discussion, we represent the common solution computed by all agents at iteration k by x^k, obtained at the conclusion of the ADMM method. Before proceeding, we introduce the projection operator onto the tangent space of the active inequality constraints at x:

$$\mathcal{P}(x) = I - C(x)\left(C(x)^{\mathsf{T}}C(x)\right)^{-1}C(x)^{\mathsf{T}}, \qquad (22)$$

and denote the initial positive definite approximation of the Hessian $\nabla_{xx}\mathcal{L}_i$ by $H_{i,0}$.

Lemma 1: Let Assumption 3 hold. If the BFGS update scheme (13) is utilized in generating the sequence $\{H_{i,k}\}$ of approximations of the Lagrangian of (4), $\forall i \in \mathcal{V}$, then $\{H_{i,k}\}$ satisfies the bounded deterioration condition, i.e.,

$$\|H_{i,k+1} - H_i^\star\|_2 \leq (1 + \alpha_1\alpha_k)\|H_{i,k} - H_i^\star\|_2 + \alpha_2\alpha_k \ , \qquad (23)$$

where and $H_i^\star = \nabla_{xx}\mathcal{L}_i(x^\star, \nu_i^\star)$, provided that $\|x^0 - x^\star\|_2$ and $\|H_{i,0} - H_i^\star\|_2$ are sufficiently small. In addition, $\{H_{i,k}\}$ is uniformly bounded, i.e., there exists a positive constant α such that

$$\|H_{i,k}\|_2 \leq \alpha, \qquad (24)$$

for all k, and $H_{i,k}$ has a uniformly bounded inverse, i.e., there exists a positive constant ξ such that

$$\|H_{i,k}^{-1}\|_2 \leq \xi, \qquad (25)$$

for all k.

Proof: The proof is readily available in many existing papers. We refer readers to [33] for the proof. ∎

Theorem 2: Provided that the conditions in Lemma 1 are satisfied, the sequence of approximations of the Hessian of the Lagrangian satisfy

$$\lim_{k\to\infty} \frac{\|\mathcal{P}(x^k)(H_k - H^\star)(x^{k+1} - x^k)\|_2}{\|x^{k+1} - x^k\|_2} = 0. \qquad (26)$$

Consequently, the iterates (x^k, ν^k) converge superlinearly to a local primal-dual optimal solution $(x^\star, u^\star)$. Moreover, x^k converges superlinearly to $x^\star$ and u^k converges R-superlinearly to $u^\star$.

Proof: We omit the proof here. Refer to [34], [35] for details of the proof. ∎

VI. Simulations

Now, we examine the performance of the distributed algorithm SQuATT in multi-agent non-convex target tracking problems. We consider a target with Dubins-car dynamics, given by

$$\dot{\bar{x}}_t = \begin{bmatrix} \dot{p}_{x,t} \\ \dot{p}_{y,t} \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} u_t\cos(\theta_t) \\ u_t\sin(\theta_t) \\ \frac{u_t}{L}\tan(\beta_t) \end{bmatrix}, \qquad (27)$$

where $\bar{x}_t = [p_{x,t}, p_{y,t}, \theta_t]^{\mathsf{T}} \in \mathbb{R}^3$ denotes the state of the target at time t, $p_t \in \mathbb{R}^2$ denotes the $2D$ position of the center of the rear axle of the target, θ_t denotes the heading of the vehicle, and L denotes the length of its wheelbase, i.e., the distance between the front and rear axles. The control inputs include the velocity $u_t \in \mathbb{R}$ and the steering angle of the front wheels $\beta_t \in \mathbb{R}$. Using the forward Euler discretization scheme, we obtain the discrete-time dynamics model

$$\bar{x}_{t+1} = \begin{bmatrix} p_{x,t} + \delta t \cdot u_t\cos(\theta_t) \\ p_{y,t} + \delta t \cdot u_t\sin(\theta_t) \\ \theta_t + \delta t \cdot \frac{u_t}{L}\tan(\beta_t) \end{bmatrix}, \qquad (28)$$

where δt denotes the time interval. We note that each agent does not have access to the control inputs of the target. Consequently, we approximate the dynamics model of the target using the nominal speed and steering angle of the target, $\bar{u}$ and $\bar{\beta}$, respectively, for u_t and β_t. We reiterate that $\bar{u}$ and $\bar{\beta}$ do not represent the actual speed and steering angle of the target, which are unknown. In fact, the values of $\bar{u}$ and $\bar{\beta}$ may be quite far from the actual speed and steering angle of the target at an arbitrary time instant t. As a result, we capture the inherent approximation errors in the dynamics

model of the target through the process noise term $v_t \in \mathbb{R}^3$ as well as our confidence in the resulting dynamics model through the covariance of the process noise $P \in \mathbb{R}^{3\times 3}$. The non-convex dynamics model in (28) leads to a non-convex MAP optimization problem. We assume each agent can obtain noisy measurements of the range and heading of the target using its onboard sensors, such as radars or cameras, with the associated measurement model of agent i given by

$$y_{i,t} = h_i(x_t) + \omega_{i,t} = \begin{bmatrix} \|\eta_{i,t} - p_t\|_2^2 \\ \pi_{i,t} - \theta_t \end{bmatrix} + \omega_{i,t}, \qquad (29)$$

where $\eta_{i,t} \in \mathbb{R}^2$ denotes the $2D$ position of agent i, $\pi_{i,t} \in \mathbb{R}^2$ denotes its heading angle, and $\omega_{i,t} \in \mathbb{R}^2$ denotes the measurement noise. We assume the sensors have a limited sensing range. As a result, each agent cannot observe the target over the entire duration of the problem.

The agents collectively estimate the trajectory of the target by solving (4), with the objective function given in (3). In addition, we incorporate prior knowledge that each agent must maintain a specified distance $d_i \in \mathbb{R}_{++}$ from the target vehicle at all time to avoid collisions through the constraint

$$\phi_i(x_t) = d_i^2 - \|\eta_{i,t} - p_t\|_2^2, \qquad (30)$$

in (4). We examine the convergence rates of SQuATT in comparison to the following distributed non-convex optimization methods: ADMM [25], NEXT, [16] and dual decomposition [20] to a centralized solution in this problem, with $N = 25$ robots. We set the maximum number of iterations of each method at 2000 iterations, with the convergence threshold set at $1e^{-5}$. We randomly generate the ground-truth trajectory of the target along with each agent's measurements, the process noise and measurement noise covariance matrices, and the prior distribution of the target's initial state. We show the estimated trajectory of the target in one instance of the target tracking problem in Figure 1, with a few of the robots (in orange-colored circles) displayed for easy visualization. The estimated trajectory of the target begins at the square in Figure 1 (see the attached video for additional information[1]).

Figure 2 shows the convergence rates of each algorithm on a randomly-generated connected communication network, with a connectivity ratio $\kappa = 0.74$, where κ represents a measure of the connectedness of a communication network, given by $\kappa = \frac{2|\mathcal{E}|}{N(N-1)}$. SQuATT requires the smallest number of iterations for convergence compared to the other algorithms. In Table I, we highlight the total computation time per agent required for convergence in each algorithm, across a range of randomly-generated communication graphs with different connectivity ratios. SQuATT provides about three times speedup in the computation time (with respect to the best competing algorithm), across the range of communication networks. In Table II, we provide the cumulative size of messages exchanged per agent required for convergence in each algorithm, across the same range of communication graphs. We note that SQuATT incurs a communication cost comparable to ADMM, which attains the smallest

[1]https://bit.ly/3fTGvdf

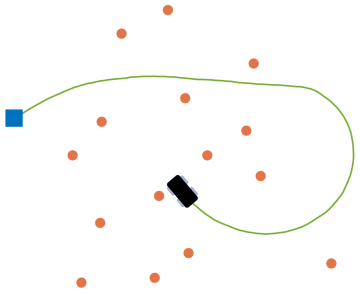

Fig. 1. The estimated trajectory of a Dubins car by a group of agents (in orange-colored circles) measuring the range and bearing of the target when the target is within sensing range.

communication cost. In general, each algorithm converges faster on networks with greater connectivity ratios and requires more communication rounds to circulate information in networks with smaller connectivity ratios.

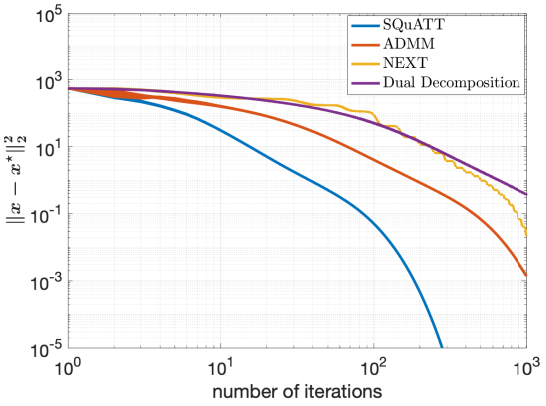

Fig. 2. On a randomly-generated communication network with $\kappa = 0.74$, SQuATT requires a smaller number of iterations to converge compared to ADMM, NEXT, and dual decomposition.

VII. Conclusion

We introduce a distributed algorithm for non-convex target tracking problems, which enables a group of agents to collaboratively estimate the trajectory of a target without communicating local observations of the target, and without requiring a central node to coordinate computation. Each agent communicates with its one-hop neighbors over a communication network to reach agreement on a common estimate of the target's trajectory. Our algorithm provides faster empirical convergence to a locally optimal solution of the non-convex target tracking problem compared to other distributed algorithms, making it suitable for target tracking problems with limited access to adequate computation and communication resources. In future work, we aim to apply trust-region optimization techniques to improve the convergence properties of our algorithm and robust asynchronous

TABLE I

THE MEAN AND STANDARD DEVIATION OF THE TOTAL COMPUTATION TIME (COMP. TIME) PER AGENT, IN SECONDS, FOR EACH DISTRIBUTED ALGORITHM, ACROSS A RANGE OF COMMUNICATION GRAPHS WITH DIFFERENT CONNECTIVITY RATIOS, κ.

Algorithm	$\kappa = 0.40$	$\kappa = 0.72$	$\kappa = 0.95$	$\kappa = 1.00$
SQuATT	3.363 ± 0.00404	3.128 ± 0.00322	2.856 ± 0.00294	1.996 ± 0.00214
ADMM	9.301 ± 0.475	8.319 ± 0.401	7.450 ± 0.361	4.042 ± 0.192
NEXT	25.425 ± 0.784	27.797 ± 0.891	25.698 ± 0.821	27.388 ± 0.839
Dual Decomposition	55.304 ± 0.0563	54.627 ± 0.0570	55.957 ± 0.0575	54.764 ± 0.0586

TABLE II

THE CUMULATIVE SIZE OF MESSAGES EXCHANGED PER AGENT, IN MB, FOR EACH DISTRIBUTED ALGORITHM, ACROSS A RANGE OF COMMUNICATION GRAPHS WITH DIFFERENT CONNECTIVITY RATIOS, κ.

Algorithm	$\kappa = 0.40$	$\kappa = 0.72$	$\kappa = 0.95$	$\kappa = 1.00$
SQuATT	3.850	3.494	3.192	2.227
ADMM	3.456	3.101	2.746	1.483
NEXT	19.210	19.210	19.210	19.210
Dual Decomposition	28.805	28.805	28.805	28.805

methods that are robust to different agents completing the update procedures at different rates.

REFERENCES

[1] O. Shorinwa, J. Yu, T. Halsted, A. Koufos, and M. Schwager, "Distributed multi-target tracking for autonomous vehicle fleets," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3495–3501.

[2] F. E. Curtis, Z. Han, and D. P. Robinson, "A globally convergent primal-dual active-set framework for large-scale convex quadratic optimization," *Computational Optimization and Applications*, vol. 60, no. 2, pp. 311–341, 2015.

[3] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, 2010.

[4] A. Dias, J. Capitan, L. Merino, J. Almeida, P. Lima, and E. Silva, "Decentralized target tracking based on multi-robot cooperative triangulation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3449–3455.

[5] B.-N. Vo, B.-T. Vo, and D. Phung, "Labeled random finite sets and the bayes multi-target tracking filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 24, pp. 6554–6567, 2014.

[6] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. S. Sukhatme, "Cooperative control for target tracking with onboard sensing," in *Experimental robotics*. Springer, 2016, pp. 879–892.

[7] L.-L. Ong, T. Bailey, H. Durrant-Whyte, and B. Upcroft, "Decentralised particle filtering for multiple target tracking in wireless sensor networks," in *2008 11th International Conference on Information Fusion*. IEEE, 2008, pp. 1–8.

[8] K. Zhou and S. I. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 678–695, 2011.

[9] A. Ahmad, G. D. Tipaldi, P. Lima, and W. Burgard, "Cooperative robot localization and target tracking based on least squares minimization," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5696–5701.

[10] J. Derenick, J. Spletzer, and A. Hsieh, "An optimal approach to collaborative target tracking with performance guarantees," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 1, pp. 47–67, 2009.

[11] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Advances in Neural Information Processing Systems*, 2015, pp. 2737–2745.

[12] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Advances in neural information processing systems*, 2014, pp. 1646–1654.

[13] A. Nedic and A. Ozdaglar, *Cooperative distributed multi-agent optimization*. Cambridge University Press, 2009, pp. 340—-386.

[14] J. Zeng and W. Yin, "Extrapush for convex smooth decentralized optimization over directed networks." *Journal of Computational Mathematics*, vol. 35, no. 4, 2017.

[15] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, 2017.

[16] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[17] G. Scutari and Y. Sun, "Distributed nonconvex constrained optimization over time-varying digraphs," *Mathematical Programming*, vol. 176, no. 1, pp. 497–544, 2019.

[18] A. Forsgren, P. E. Gill, and E. Wong, "Primal and dual active-set methods for convex quadratic programming," *Mathematical programming*, vol. 159, no. 1, pp. 469–508, 2016.

[19] Q. T. Dinh, I. Necoara, and M. Diehl, "A dual decomposition algorithm for separable nonconvex optimization using the penalty function framework," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 2372–2377.

[20] I. Necoara, C. Savorgnan, D. Q. Tran, J. Suykens, and M. Diehl, "Distributed nonlinear optimal control using sequential convex programming and smoothing techniques," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 543–548.

[21] L. Lu, "Separable nonlinear model predictive control via sequential quadratic programming for large-scale systems," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 495–500, 2015.

[22] N. Boland, J. Christiansen, B. Dandurand, A. Eberhard, and F. Oliveira, "A parallelizable augmented lagrangian method applied to large-scale non-convex-constrained optimization problems," *Mathematical Programming*, vol. 175, no. 1-2, pp. 503–536, 2019.

[23] J.-H. Hours and C. N. Jones, "An augmented lagrangian coordination-decomposition algorithm for solving distributed non-convex programs," in *2014 American Control Conference*. Ieee, 2014, pp. 4312–4317.

[24] B. Houska, J. Frasch, and M. Diehl, "An augmented lagrangian based algorithm for distributed nonconvex optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1101–1127, 2016.

[25] S. Magnússon, P. C. Weeraddana, and C. Fischione, "A distributed approach for the optimal power-flow problem based on ADMM and sequential convex approximations," *IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS*, vol. 2, no. 3, 2015.

[26] D. Hajinezhad and M. Hong, "Nonconvex alternating direction method of multipliers for distributed sparse principal component analysis," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2015, pp. 255–259.

[27] J. E. Dennis, Jr and J. J. Moré, "Quasi-newton methods, motivation and theory," *SIAM review*, vol. 19, no. 1, pp. 46–89, 1977.

[28] R. Tapia, "On secant updates for use in general constrained optimization," *Mathematics of Computation*, vol. 51, no. 183, pp. 181–202, 1988.

[29] R. Saber and R. Murray, "Consensus protocols for networks of dynamic agents," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 2. IEEE, 2003, pp. 951–956.

[30] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE, 2003, pp. 482–491.

[31] R. T. Rockafellar, *Convex analysis*. Princeton university press, 1970, vol. 18.

[32] L. Chen, D. Sun, and K.-C. Toh, "A note on the convergence of ADMM for linearly constrained convex optimization problems," *Computational Optimization and Applications*, vol. 66, no. 2, pp. 327–343, 2017.

[33] J. Dennis, H. J. Martinez, and R. A. Tapia, "Convergence theory for the structured BFGS secant method with an application to nonlinear least squares," *Journal of Optimization Theory and Applications*, vol. 61, no. 2, pp. 161–178, 1989.

[34] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta numerica*, vol. 4, pp. 1–51, 1995.

[35] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes, "Local convergence of SQP methods for mathematical programs with equilibrium constraints," *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 259–286, 2006.