2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Optimisation with Zeroth-Order Oracles in Formation

Elad Michael, Daniel Zelazo, Tony A. Wood, Chris Manzie, and Iman Shames

***Abstract*— In this paper, we consider the optimisation of time varying functions by a network of agents with no gradient information. We propose a novel method to estimate the gradient at each agent's position using only neighbour information. The gradient estimation is coupled with a formation controller, to minimise gradient estimation error and prevent agent collisions. Convergence results for the algorithm are provided for functions which satisfy the Polyak-Łojasiewicz inequality. Simulations and numerical results are provided to support the theoretical results.**

I. INTRODUCTION

In time varying optimisation tasks, the goal is to optimise a sequence of problems where each new objective is a variation of the previous. The time varying function can represent the position of a moving source, with measurements capturing signal strength. We assume that only measurements, with no higher order information, are available at each iteration. To compensate for the lack of gradient information, we consider a cooperating formation of agents, sharing information to minimise the time-varying function. In general, this may be posed as a sequence of independent optimisation problems [1]. One could simply treat every new cost function as an entirely new optimisation problem, although this may be computationally infeasible. Additionally, solving for the optimum at each iteration is unnecessary if it is sufficient to remain within some neighbourhood of the optimum at every iteration. If there is a limit on the variation of objective parameters between iterations, the solution of the previous iteration can be updated to approach of the solution of the current iteration.

At each iteration, some amount of information about the changing function f_k must be measured. Here we adopt the term *p-th order oracle* [2] to describe the type of available information. If $p = 0$ then the zeroth-order oracle only makes available the current function value $f_k(x_k)$, and not any gradient or higher order information. Gradient descent makes use of a first-order oracle, Newton's method a second-order oracle, etc. We derive an iterative approach to track the optima of a changing cost function using zeroth-order oracles, and minimal assumptions on the behavior of f_k. This is similar to finite difference stochastic approximation (FDSA), except in this case the function f_k can only be observed at the locations of the agents, rather than user chosen sample points. Therefore, the agents receive information from wherever their neighbours are to compute an approximate descent direction at each iteration. As the accuracy of the agent's gradient estimate is dependent on the geometry of its neighbours, we incorporate a formation control strategy to ensure the gradient estimation is accurate.

In the area of online optimisation, we will give a short review of the time varying optimisation problems, but largely focus on the gradient free solutions which are more relevant to our formulation. Time varying optimisation problems are well studied, frequently under the name Online Convex Optimisation or OCO. A predictive/corrective method for OCO is presented in [3], using gradient information and line search methods. Online convex optimisation with constraints is addressed by [4], with regret bounds and convergence results. These approaches use gradient information which we assume is unavailable in this formulation. The term bandit feedback is also used to describe this problem coupled with a zeroth-order oracle, as it conforms to a multi-armed bandit problem with convex costs [2]. Regret bounds assuming compactness and convexity are derived in [5], and a similar technique is used in [6] with a multi-point estimate at each iteration for bandit feedback problems. A similar technique but using only a stochastic two point sampling each iteration is derived in [7]. These results utilise random or user chosen function sampling at each iteration, are entirely centralised, and assume convexity of the unknown cost functions. A network of zeroth-order oracles localising the source of a static scalar field is examined in [8], with existence and convergence results, by assuming the existence of controllers with given properties. We extend this analysis to time-varying scalar fields, as well as focusing on the controller design.

In this paper, we present a novel algorithm which combines the information from a network of zeroth oracles to optimise a time-varying cost function. We assume the agents follow single integrator dynamics, and construct a gradient estimate which uses only local information. As well, we provide a novel method of bounding the gradient estimation error, which has an interesting geometric interpretation. As such, we incorporate formation control, along with the gradient descent, to minimise the gradient estimation error. Both gradient estimation and formation control laws require only local information, leading to an entirely distributed approach. Additionally, we allow for a time-varying objective function, and the assumptions on the time-varying objective functions are only the Lipschitz continuity of the gradient and the Polyak-Łojasiewicz inequality. These assumptions are weaker than many which are used to provide the linear

E. Michael, T. A. Wood, C. Manzie, and I. Shames are with the Department of Electrical and Electronic Engineering, University of Melbourne. {eladm@student, wood.t@,manziec@,ishames@}unimelb.edu.au

D. Zelazo is with the Faculty of Aerospace Engineering, Israel Institute of Technology, Israel. dzelazo@technion.ac.il

978-1-7281-7447-1/20/$31.00 ©2020 IEEE

convergence of gradient descent algorithms [9].

The paper is organised as follows. Section II is devoted to basic assumptions on the time-varying function and agent dynamics. Section III covers the approximation of the gradient given only zeroth-order information from an agent and its neighbours and derives an error bound on the gradient approximation. Section IV introduces and unifies formation control with the minimisation. Finally, simulation and conclusions are covered in Section V.

II. Problem Formulation

Consider a network of n agents where $x_k^i \in \mathbb{R}^d$ denotes the position of the i-th agent for $i \in \{1, ..., n\}$ at iteration k in dimension d. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the underlying graph of the network with the vertex set $\mathcal{V} = \{1, ..., n\}$ and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The edge set $\mathcal{E}$ captures the communication topology of the network, i.e. agent i receives information from agent j if $(i, j) \in \mathcal{E}$. Denote the neighbour set of each agent i by $\mathcal{N}^i$ where $\mathcal{N}^i = \{j \mid (i, j) \in \mathcal{E}\}$.

In this paper, we consider the edges to be bidirectional, i.e., if $(i, j) \in \mathcal{E}$ then $(j, i) \in \mathcal{E}$. We begin with an assumption on the network structure.

Assumption 1. *Assume that the network is* undirected, connected*, and that* $|\mathcal{N}^i| \geq d \; \forall \, i \in \mathcal{V}$.

If the network is disconnected, then each connected subnetwork would display the same behavior as is presented in this paper. The assumption that the neighbour set has cardinality greater than or equal to the dimension is necessary for the algorithm presented, and as the authors primarily envision physical applications (2 or 3 dimensional), is not seen as a restrictive assumption.

The agents are modeled as single integrators, with dynamics

$$x_{k+1}^i = x_k^i + u_k^i. \tag{1}$$

At each time instance k, each agent i can measure $y_k^i = f_k(x_k^i)$. Let $\mathcal{X}_k^*$ denote the set of minimisers of the time-varying function f_k. The following assumptions hold for the functions being minimised f_k.

Assumption 2. *(Differentiability and Lipschitz Gradient): The function $f_k : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable. The gradient is Lipschitz with constant L, that is there exists a positive scalar L such that, $x \in \mathbb{R}^d$, $y \in \mathbb{R}^d$,*

$$||\nabla f_k(x) - \nabla f_k(y)|| \leq L||x - y||,$$

or equivalently

$$f_k(y) \leq f_k(x) + \nabla f_k(x)^T (y - x) + \frac{L}{2}||y - x||^2.$$

We allow for the cost function f_k to change at each iteration, however we make two assumptions on the changing cost functions.

Assumption 3. *(Polyak Condition): There exists a positive scalar s such that*

$$||\nabla f_k(x)||^2 \geq 2s(f_k(x) - f_k(x_k^*))$$

where $x_k^* \in \mathcal{X}_k^*$ are the minimisers of f_k.

Assumption 4. *(Bounded Drift in Time): There exist positive scalars η_0 and η^* such that $|f_{k+1}(x) - f_k(x)| \leq \eta_0$ for all $x \in \mathbb{R}^d$ and $|f_k(x_k^*) - f_{k+1}(x_{k+1}^*)| \leq \eta^*$.*

The problem of interest is given below.

Problem 1. *Let $\mathcal{X}_k^*$ denote the set of minimisers of the time-varying function f_k. For a network of n agents modeled in* (1)*, under Assumptions 1-4, find inputs u_k^i for all agents $i \in \mathcal{V}$ given $\mathcal{Y}_k^i = \{y_k^j \mid j \in \mathcal{N}^i \cup \{i\}\}$, i.e., the set of measurements available to agent i at iteration k, and a positive constant M such that $||x_k^i - x_k^*|| \leq M$ as $k \to \infty$ where $x_k^* = \arg\min_{x \in \mathcal{X}_k^*} \|x_k^i - x\|$.*

III. Zeroth Order Network

If, at each time step k, each agent i was able to query an oracle and receive $\nabla f_k(x_k^i)$, then a standard gradient descent method could be used to reach the set of minimisers. Motivated by this, we construct an approximate gradient oracle which combines the set of measurements from the agent and its neighbours to produce a descent direction at each iteration k.

Consider the directional derivative along the path from agent i to agent $j \in \mathcal{N}^i$

$$\nabla_{ji} f_k(x_k^i) = \frac{(x_k^j - x_k^i)^T}{||x_k^j - x_k^i||} \nabla f_k(x_k^i).$$

We construct an estimate of the gradient $\Lambda^i(x_k)$ with an error term ϵ_k^{ji}

$$\langle v_k^{ji}, \Lambda^i(x_k) \rangle = \frac{y_k^j - y_k^i}{||x_k^{ji}||}, \tag{2}$$

$$\langle v_k^{ji}, \nabla f_k(x_k^i) \rangle = \langle v_k^{ji}, \Lambda^i(x_k) \rangle - \epsilon_k^{ji}. \tag{3}$$

where we are using the shortening $x_k^{ji} = x_k^j - x_k^i$ to represent the difference vector and $v_k^{ji} = x_k^{ji}/||x_k^{ji}||$ as the unit vector in the difference's direction. We use $\langle u, v \rangle$ to denote the standard inner product when superscripts make $u^T v$ cumbersome. Note that if the function f was linear, (2) would be the exact directional derivative with $\epsilon_k^{ji} = 0$. Using the estimate $\langle v_k^{ji}, \Lambda^i(x_k) \rangle$ and Assumption 2, the error term ϵ_k^{ji} is bounded by

$$y_k^j - y_k^i - \langle x_k^{ji}, \nabla f_k(x_k^i) \rangle \leq \frac{L}{2}||x_k^{ji}||^2,$$

$$\epsilon_k^{ji} \leq \frac{L}{2}||x_k^{ji}||. \tag{4}$$

However, computing an approximation of $-\nabla f_k(x_k^i)$ to use as a descent direction with bounded error requires more than just information in the x_k^{ji} direction. By summing (2) across all neighbours, with some algebraic manipulation, the gradient approximation is

$$\Lambda^i(x_k) = \left[\sum_{j \in \mathcal{N}^i} v_k^{ji} (v_k^{ji})^T \right]^{-1} \sum_{j \in \mathcal{N}^i} \frac{y_k^j - y_k^i}{||x_k^{ji}||} v_k^{ji}. \tag{5}$$

Remark 1. *Each agent i can compute a local estimate of $\nabla f_k(x_k^i)$ via (5) using only x_k^i and x_k^j, $j \in \mathcal{N}^i$.*

Note that if the sum of outer products on the left of (5) is not of appropriate rank, it cannot be inverted to estimate the gradient. In $2/3$ dimensions, this requires that there exists $2/3$ neighbours which are not co-linear/co-planar. Additionally, if any adjacent agents coincide, then the gradient estimate $\Lambda^i(x_k)$ cannot be computed. Both the rank requirement and the requirement that no agents coincide will be addressed using formation control strategies in Section IV.

Theorem 1. *For a function f_k and a set of agents satisfying Assumptions 1-4, the vector $\Lambda^i(x_k)$ as defined in (5) satisfies*

$$||\Lambda^i(x_k) - \nabla f_k(x_k^i)|| \leq \delta_k^i, \tag{6}$$

where, in $\mathbb{R}^2$, the error δ_k^i is defined to be

$$\delta_k^i := \min_{j,l \in \mathcal{N}^i} \frac{L}{|\langle v_k^{li}, \bar{v}_k^{ji} \rangle|} \max(||x_k^{ji} + x_k^{li}||, ||x_k^{jl}||), \tag{7}$$

and we have used $\bar{v}^{ji}$ to indicate a vector which is orthogonal to v^{ji}.

Proof. Recall the error bound on a single directional derivative state in (4). Let $a_k^{ji} := \frac{L}{2}||x_k^{ji}||$ and $d_k^{ji} := \frac{y_k^j - y_k^i}{||x_k^{ji}||}$. Rearranging the error bound into a set of inequalities yields

$$d_k^{ji} - a_k^{ji} \leq \langle v_k^{ji}, \nabla f_k(x_k^i) \rangle \leq d_k^{ji} + a_k^{ji}. \tag{8}$$

In $\mathbb{R}^2$ representing the space of all possible gradients, these two inequalities enclose a band of $\mathbb{R}^2$ of width $2a_k^{ji}$ bordered by two parallel lines perpendicular to v_k^{ji}. Consider the set of inequalities from an additional neighbour $l \in \mathcal{N}^i$

$$d^{il} - a_k^{li} \leq \langle v_k^{li}, \nabla f_k(x_k^i) \rangle \leq d^{il} + a_k^{li}, \tag{9}$$

Then as long as v_k^{li} and v_k^{ji} are not parallel they enclose a finite area parallelogram $\mathcal{P}_{ijl} \in \mathbb{R}^2$, and $\nabla f_k(x_k^i) \in \mathcal{P}_{ijl}$. The gradient $\nabla f_k(x_k^i)$ and estimate $\Lambda^i(x_k)$ are both inside the parallelogram $\mathcal{P}_{ijl}$ because they each satisfy (8) and (9). Therefore, the error $||\Lambda^i(x_k) - \nabla f_k(x_k^i)||$ is bounded by diameter of the smallest ball containing $\mathcal{P}_{ijl}$. The diagonals of $\mathcal{P}_{ijl}$ have lengths

$$l = \frac{L}{|\langle v_k^{li}, \bar{v}_k^{ji} \rangle|} ||(x_k^j - x_k^i) \pm (x_k^l - x_k^i)||. \tag{10}$$

For an agent i with neighbours j and l, the longest diagonal thus has length

$$l = \frac{L}{|\langle v_k^{li}, \bar{v}_k^{ji} \rangle|} \max(||x_k^{ji} + x_k^{li}||, ||x_k^{jl}||), \tag{11}$$

which is the bound used in the Theorem. □

An example of the parallelogram $\mathcal{P}_{ijl}$ is shown in Figure 1, generated with $v_k^{ji} = \frac{1}{\sqrt{2}}[1, 1]^T$, $a_k^{ji} = 2$, $d_k^{ji} = 0$, and $v_k^{li} = \frac{1}{\sqrt{5}}[1, -2]^T$, $a_k^{li} = 2$, $d_k^{li} = 3$.

This bound does not take into account where within the parallelogram the gradient estimate $\Lambda^i(x_k)$ falls, which may decrease the distance to the farthest point by up to a factor of 2.

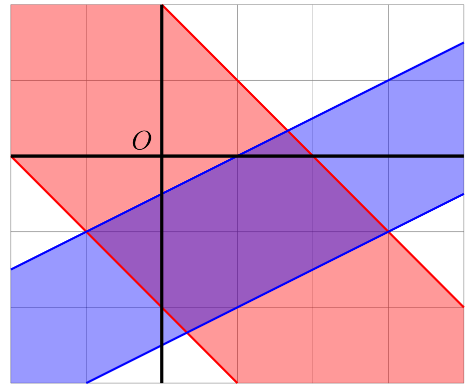

Fig. 1. The set of feasible gradients which satisfy (8) and (9).

An almost identical bounding procedure for the error is possible in $\mathbb{R}^n$, with each neighbour specifying a pair of parallel hyper plane constraints, which given n neighbours result in an n-parallelotope. The approximate gradient formulation (5) is the same for any dimension.

Repeating this technique of partitioning the space of possible gradients with information from additional neighbours, the bound can be tightened. However, using additional agents significantly increases the computational burden, as the resulting polytope of possible gradients will have uncertain structure, and maximising the norm under linear constraints is itself an NP-hard problem. Additionally, empirically the error bound from the complete set of neighbours largely seems to be determined by the pair of neighbours from Theorem 1. Finally, computing the bound involving only 2 neighbours is independent of the function measurements y_k^i, y_k^j, y_k^l. With additional neighbours forming a polytope with more facets, this computational convenience is lost.

Using the gradient approximation from (5), the agents are able to find an approximate descent direction which has an error bounded by (7). To minimise the bound (7), thereby creating a better gradient approximation, we combine formation control with the decentralised minimisation.

IV. Optimisation in Formation

Examining the parallelogram $\mathcal{P}_{ijl}$ in Fig. 1, there are two intuitive methods to minimise the diameter of the smallest bounding ball. We can ensure that the two bands are orthogonal, "squaring" the parallelogram, and bring the parallel edges closer together, "thinning" the parallelogram. The former can be achieved by keeping the vectors v_k^{li} and v_k^{ji} orthogonal, i.e. ensuring that $\langle v_k^{li}, \bar{v}_k^{ji} \rangle = 1$ and the latter by keeping the agents as close as possible while maintaining a non-collision guarantee. Finally, it is critical to prevent collinearity of the neighbours, when $\langle v_k^{li}, \bar{v}_k^{ji} \rangle = 0$, which geometrically corresponds to both bands in Fig. 1 being parallel to each other. We use decentralised navigation functions [10], [11] to maintain a desirable formation while minimising f_k.

Definition 1. *Let $\phi^i : \mathbb{R}^{n_i d} \to \mathbb{R}^+$ be the navigation potential function for agent i where $n_i = |\mathcal{N}_i|+1$, with the* *following properties:*

1) *The function ϕ^i is continuously differentiable on $\mathbb{R}^d$.*
2) *The function ϕ^i has a unique minimum, only attained when the agents are in the desired formation configuration.*
3) *The function ϕ^i is Morse (critical points are non-degenerate).*
4) *The function can be computed decentrally, i.e., each agent i can compute $\phi^i(x_k)$ using only x_k^i and x_k^j, $j \in \mathcal{N}^i$.*

Note that these navigation potential functions exclude distance based approaches such as in [12], as they are not Morse and we cannot, as of yet, prove the global convergence properties derived here. Using the decentralised navigation functions $\phi^i(x_k)$, and information available locally to each agent i, the agents are able to decrease a common global potential function

$$\phi(x_k) = \sum_{i \in \mathcal{V}} \phi^i(x_k). \tag{12}$$

Remark 2. *To evaluate $\phi^i(x_k)$, agent i needs access only to x_k^i and x_k^j, $j \in \mathcal{N}^i$. No information about the position of all other agents is required. Consequently, i can compute $\nabla_i \phi(x_k)$ using x_k^i and x_k^j, $j \in \mathcal{N}^i$.*

We make the following assumption throughout the remainder of the paper.

Assumption 5. *The global potential function $\phi(x_k)$ is continuously differentiable, and the gradient is Lipschitz with constant L_ϕ.*

Defining the control input $u_k^i := -\alpha_k^i p_k^i$, the dynamics are

$$x_{k+1}^i = x_k^i - \alpha_k^i p_k^i, \tag{13}$$

where $\alpha_k^i > 0$ is a design constant. Define p_k^i for agent i at k as

$$p_k^i = \lambda_k^i \Lambda^i(x_k) + (1 - \lambda_k^i) \nabla_i \phi(x_k), \tag{14}$$

where $\Lambda^i(x_k)$ is the estimate of the gradient from (5) and $\lambda_k^i \in [0, 1]$ allows the agents to "focus" on the primary goal of minimising f_k while maintaining formation. The rules for deciding the weight λ_k^i and constant α_k^i are laid out in Theorem 2.

Theorem 2. *Let $\phi(x) = \sum_{i \in \mathcal{V}} \phi^i(x_k)$ be the sum of functions $\phi_i(x_k)$, with a Lipschitz continuous gradient with constant L_ϕ. Let Φ^i, $i \in \{1, \dots, n\}$ be positive constants. For a set of agents with dynamics as in* (13)*, with step direction* (14)*, define the weighting parameter λ_k^i*

$$\lambda_k^i := \min(1, \frac{||\nabla_i \phi(x_k)||}{||\Lambda^i(x_k) - \nabla_i \phi(x_k)||} \frac{\sigma(\Phi^i)}{\sigma(\phi^i(x_k))}), \tag{15}$$

where σ is a class $\mathcal{K}$ function. Define $\bar{\alpha}_k^i$ to be

$$\bar{\alpha}_k^i = \frac{2c}{||\Lambda^i(x_k) - \nabla_i \phi(x_k)||^2}, \tag{16}$$

where c is a constant. Let the design constant α_k^i be in the interval

$$\alpha_k^i \in (0, \min(\frac{1}{L_\phi}, \frac{1}{L}, \bar{\alpha}_k^i)] \tag{17}$$

where L is the Lipschitz constant for the gradient of f_k. Then the system is stable, and $\exists k_0$ such that $\forall k \geq k_0$ the global potential function is bounded $\phi(x_k) \leq \sum_i \Phi^i + c$.

Proof. Proof in [13]. □

Remark 3. *In light of* (15) *and* (17)*, and Remarks 1 and 2, it can be seen that each agent i can compute u_k^i using only x_k^i and x_k^j, $j \in \mathcal{N}^i$.*

Let D_0^i be the projection of set $\{x \mid \phi^i(x) \leq \Phi^i + c\}$ onto the subspace defined by x^i and x^j, $j \in \mathcal{N}^i$. The boundedness of $\phi(x_k)$ corresponds to trajectories converging to D_0^i. We should choose the formation potential functions ϕ^i and design constants Φ^i, c such that having two collinear neighbours (in 2-D) or three coplanar neighbours (in 3-D) is impossible. Thus, we ensure that the matrix in (5) is full-rank.

We may also leverage the convergence of the potential function to bound the gradient estimate error, as the formation fixes the geometry of the estimation. Using the new step direction, the the modified gradient error term is

$$||\nabla f_k(x_k^i) - p_k^i|| \leq \delta_k^i + (1 - \lambda_k^i)||\Lambda^i(x_k) - \nabla_i \phi(x_k)||, \tag{18}$$

$$\leq \delta_k^i + ||\Lambda^i(x_k)|| + ||\nabla_i \phi(x_k)||, \tag{19}$$

where δ_k^i is the upper bound on the error of the original gradient estimate from (7). Define ρ^i be the radius of the smallest ball which contains D_0^i and is centred at x^i. By the Lipshitz gradient property of $\phi(x)$, we then have

$$||\nabla f_k(x_k^i) - p_k^i|| \leq \delta_k^i + ||\Lambda^i(x_k)|| + L_\phi \rho^i \ \forall\ k \geq k_0 \tag{20}$$

Finally if we assume that $||\Lambda^i(x_k)|| \leq \gamma^i \ \forall\ k \geq k_0$,

$$||\nabla f_k(x_k^i) - p_k^i|| \leq \delta_k^i + \gamma^i + L_\phi \rho^i \ \forall\ k \geq k_0 \tag{21}$$

To achieve the desired cooperative tasks agent i executes the following steps at each k, (i) the gradient of f_k at x_k^i is estimated using (5); (ii) $\nabla_i \phi(x_k)$ is computed; (iii) the value of λ_k^i is chosen via (15); (iv) the state is updated through (14) and an appropriate choice of α_k^i.

We conclude this section by commenting on the overall performance of the agents in tracking the minimiser(s) of f_k. To this end, note that the directions p_k at each iteration are still only approximations of the true gradients. The formation of zeroth-order agents cooperating is thus equivalent to individual agents querying a $\delta-$*first order oracle* at each iteration k. The definition of a $\delta-$*first order oracle* is given in Definition 2.

Definition 2. (δ-first order Oracle): *Given the function f and a point x the oracle returns $p(x) = \nabla f(x) + \delta(x)$ such that $||\delta(x)|| \leq \bar{\delta}$ for some positive scalar δ.*

Here we show that a δ-first order oracle is sufficient to converge to a neighbourhood of the minimisers $\mathcal{X}_k^*$, using (21) to construct an error bound on the δ-first order Oracle

$$\bar{\delta}^i := \delta_k^i + \gamma^i + L_\phi \rho^i. \tag{22}$$

With the bounds introduced in (21), we may also define a constant α for each agent,

$$\alpha^i \in (0, \min(\frac{1}{L_\phi}, \frac{1}{L_\phi}, \frac{2c}{\gamma^i(\gamma^i + 2L_\phi \rho^i)})], \tag{23}$$

which satisfies all of the required properties for Theorem 2. Note that the $\bar{\delta}$ used in Proposition 1 includes the formation control term $L_\phi \rho^i$, because it is an additional error in the gradient estimate, although it benefits the network as a whole.

Proposition 1. *If α^i is chosen such that $|(1-\alpha^i s)|< 1$, then an agent using the δ-first order oracle will reach an $M = \frac{\eta_0+\eta^*}{2\alpha^i s^2} + \frac{(\bar{\delta}^i)^2}{4s^2}$ neighbourhood of the optimiser $\mathcal{X}_k^*$ as the time steps $k \to \infty$.*

Proof. Proof in [13]. □

V. Simulations

In this section we will implement, illustrate, and analyze the method described in the previous sections. We use a formation potential adapted from [10], where each agent uses the following potential function

$$\phi^i(x_k) = \frac{\sum_{j \in \mathcal{N}^i} ||x_k^i - x_k^j - c^{ij}||_2^2}{e^{\beta(x_k)}}. \tag{24}$$

In the formation potential function given in (24), fully explored in [10], the numerator is a quadratic attraction potential to the desired difference between agents i and j. The function in the denominator $\beta(x_k)$ is described as a "collision function", which is nominally equal to 1 but quickly vanishes as the agents reach a prescribed safety distance of each other or an obstacle. The decentralised formation control from [10], [11] is shown to almost always converge, except from a set of initial conditions with measure zero. We have chosen the desired displacements c^{ij} to form a hexagon with side lengths $s = 4$. The gradient error bound (7) is

$$||\Lambda^i(x_k) - \nabla f_k(x_k^i)|| \le \frac{L||x_k^{ji} + x_k^{li}||}{|\langle v_k^{li}, \bar{v}_k^{ji}\rangle|} = 2sL + \epsilon(\lambda_0, \Phi^i).$$

If the agents were in a perfect hexagon formation, $2sL$ would be their error bound, for s the side length and L the Lipschitz constant. However, the formation maintenance is balanced with the minimisation goal, so $\epsilon(\lambda_0, \Phi^i)$ represents the additional error introduced from the choices of nominal weight λ_0 and the acceptable deviation bound Φ^i. The specific choices of these parameters, and their impacts, are examined further in this section. Each agent's eventual 2 nearest neighbours in the hexagon are its neighbour set $\mathcal{N}$.

For the function to be minimised f_k, we use convex quadratic function in two dimensions,

$$f_k(x_k^{(i)}) = x_k^{(i)T} Q x_k^{(i)} + \zeta(t)^T x_k^{(i)},$$

with $Q \succeq 0$. To simulate a moving source, the linear term $\zeta(t)$ is used to translate the quadratic along a path in the plane at a constant speed. The nominal weighting between the formation gradient and minimisation gradient was $\lambda_0 = 1$, i.e. fully weighted on the minimisation. This ensures that as long as the formation is "good enough", the agents will be attaining the best gradient estimate. The class $\mathcal{K}$ function $\sigma(\phi(x))$ used in (15) is $\sigma(z) = z^2$, and the upper bounds for all agents potential functions' are $\Phi^i = 1$. The trajectories of the agents and source function are shown in Figure 2, with the dots and star symbolising the final position of the agents and optimum of f_k.

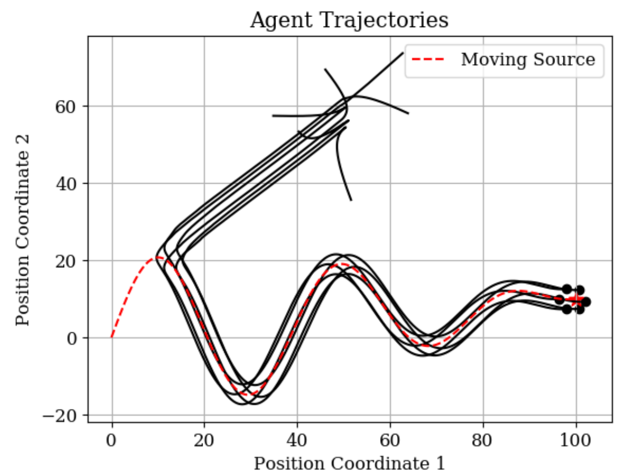

Fig. 2. Agent trajectories with upper bound $\Phi = 1$.

Immediately after the random initialisation, the agents are not in formation. Therefore, their individual formation potential values $\phi^i(x_k)$ far exceed the prescribed upper bound Φ^i, and they coalesce into formation. Once in formation, or "close enough" as determined by $\phi^i(x_k) \le \Phi^i$, the formation begins converging to the neighbourhood of the minimisers of f_k. The minimisation error $f_k(x_k) - f_k(x_k^*)$ and derived bound (see online Arxiv version) are shown for individual agents in Fig. 3. The agents quickly converge to formation around the minimiser, and remains within the neighbourhood, oscillating beneath the bound as the source function f_k changes. If the upper bound Φ^i was decreased, representing a more stringent requirement on the formation control, the formation would converge to the minimisers $\mathcal{X}_k^*$ more slowly. The choice of upper bounded is also clearly tied with the choice of the potential function $\phi(x)$. If there is a critical safety distance, between UAVs for example, then the upper bound Φ^i must be chosen such that the individual safety distance corresponds to a potential function value *greater than* the upper bound $\Phi^i + c$.

To demonstrate the benefits of the formation control, Fig. 4 shows the trajectories of the same agents without any formation control ($\lambda_k^i = 1$). The minimisation error in Fig. 5 converges more quickly to a value which is approximately an order of magnitude larger than in Fig. 3. The error of the gradient estimate is high in this case largely due to

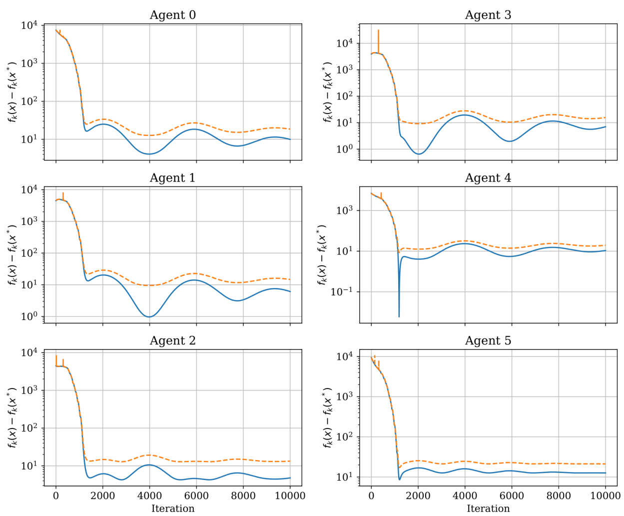

Fig. 3. Minimisation error with $\Phi = 1$.

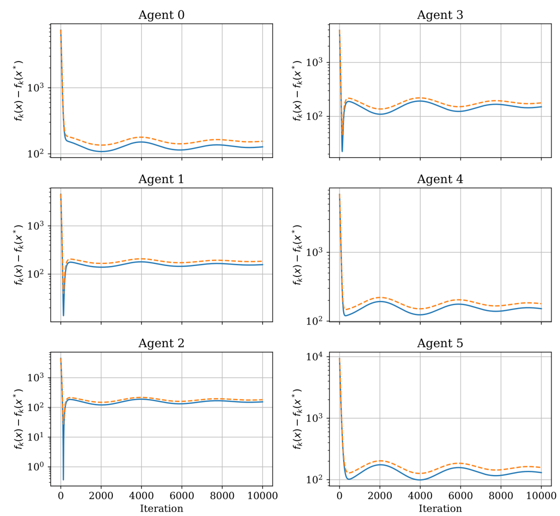

Fig. 5. Minimisation error with no formation control.

the distance between the agents being larger than in the formation controlled case Fig. 2.

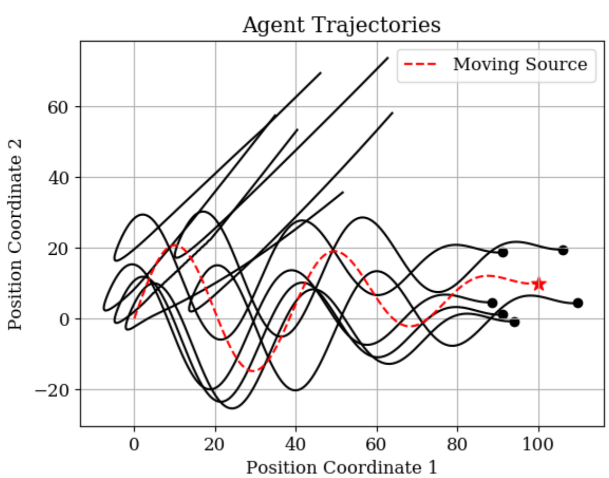

Fig. 4. Agent trajectories with no formation.

VI. Conclusion

In this paper we consider a formation of agents tracking the optimum of a time varying function f_k with no gradient information. At each iteration, the agents take measurements, compute an approximate descent direction, and converge to a neighbourhood of the optimum. We derive bounds on the neighbourhood of convergence, as a function of the error in the gradient estimate, using minimal assumptions on the time-varying function. As the gradient approximation is constructed in a decentralised way, formation control is used to encourage the agents to formations which improve the gradient estimates, while not overwhelming the task of minimising the source function f_k. We show that the formation control remains within a bounded distance of the optimal formation, and the implications to the convergence of the network to the minima of f_k. In the future, a more flexible formation control approach with convergence guarantees as well as hardware experiments will be investigated.

References

[1] B. Gutjahr, L. Gröll, and M. Werling, “Lateral vehicle trajectory optimization using constrained linear time-varying mpc,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1586–1595, 2016.

[2] I. Shames, D. Selvaratnam, and J. H. Manton, “Online optimisation using zeroth order oracles,” *IEEE Control Systems Letters*, 2019.

[3] A. Lesage-Landry, I. Shames, and J. A. Taylor, “Predictive online convex optimization,” *Automatica*, vol. 113, p. 108771, 2020.

[4] A. Simonetto and E. Dall’Anese, “Prediction-correction algorithms for time-varying constrained optimization,” *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5481–5494, 2017.

[5] S. Bubeck, N. Cesa-Bianchi, *et al.*, “Regret analysis of stochastic and nonstochastic multi-armed bandit problems,” *Foundations and Trends® in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[6] A. Agarwal, O. Dekel, and L. Xiao, “Optimal algorithms for online convex optimization with multi-point bandit feedback.” in *COLT*. Citeseer, 2010, pp. 28–40.

[7] O. Shamir, “An optimal algorithm for bandit and zero-order convex optimization with two-point feedback.” *Journal of Machine Learning Research*, vol. 18, no. 52, pp. 1–11, 2017.

[8] S. Z. Khong, Y. Tan, C. Manzie, and D. Nešić, “Multi-agent source seeking via discrete-time extremum seeking control,” *Automatica*, vol. 50, no. 9, pp. 2312–2320, 2014.

[9] H. Karimi, J. Nutini, and M. Schmidt, “Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 795–811.

[10] H. G. Tanner and A. Kumar, “Formation stabilization of multiple agents using decentralized navigation functions.” in *Robotics: Science and systems*, vol. 1. Boston, 2005, pp. 49–56.

[11] D. V. Dimarogonas and E. Frazzoli, “Analysis of decentralized potential field based multi-agent navigation via primal-dual lyapunov theory,” in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 1215–1220.

[12] V. Gazi, B. Fİdan, Y. S. Hanay, and İ. Köksal, “Aggregation, foraging, and formation control of swarms with non-holonomic agents using potential functions and sliding mode techniques,” *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 15, no. 2, pp. 149–168, 2007.

[13] E. Michael, D. Zelazo, T. A. Wood, C. Manzie, and I. Shames, “Optimization with zeroth-order oracles in formation,” *arXiv preprint*, 2020.