Proportional Integral Distributed Optimization for Dynamic Network Topologies

Greg Droge, Magnus Egerstedt

***Abstract*—This paper investigates proportional-integral distributed optimization when the underlying information exchange network is dynamic. Proportional-integral distributed optimization is a technique which combines consensus-based methods and dual-decomposition methods to form a method which has the convergence guarantees of dual-decomposition and the damped response of the consensus methods. This paper extends PI distributed optimization to allow for dynamic communication networks, permitting agents to change who they can communicate with, without sacrificing convergence to the collective optimum.**

I. Introduction

Tasks solved by multi-agent systems pose challenging problems as they generally require the agents to realize collective objectives using solely information local to each agent in the communication network, e.g. [1], [2]. Moreover, many real-world examples complicate task completion as they require agents to move about the environment, changing which agents can communicate. As tasks are often defined in terms of minimizing a cost, e.g. [2], this paper will focus on the development of a distributed optimization technique which allows a system of agents to converge to a collective minimum under dynamic communication topologies.

Methods for distributed optimization have been categorized into two groups based on the underlying principles in the design of the algorithm, e.g. [3], [4]. The first category consists of consensus-based gradient methods where, at each time step, each agent combines a step its gradient direction, for optimality, with a step in a consensus direction, for agreement, e.g. [4], [5], [6], [7]. The second category consists of introducing a constraint that differing versions be equal and solve the dual problem, e.g. [7], [8], [9].

While the approach in each method is fundamentally different, their relation can be identified using a control theoretic approach. It was shown in [10] that a consensus based method can be combined with a dual method to form a new, proportional-integral (PI) distributed optimization method. The method was shown to have a damped response, similar to the consensus method, with guaranteed convergence to the optimum, similar to the dual methods. However, the method is constrained to static topologies and requires agent indexing (ie, agents must share an identification when communicating). The contribution of this paper is to develop an index free PI distributed optimization suitable for switching communication topologies.

The remainder of this paper will proceed as follows: To fully develop PI distributed optimization for switching topologies, a brief background is given in Section II. Then, Section III and IV will give modifications that allow PI distributed optimization to converge for switching topologies. To demonstrate the applicability of the algorithm, a formation control example will be given in Section V followed with some concluding remarks in Section VI.

II. Backgound

This section will introduce the approach to distributed optimization taken in this paper. It begins with a formalization of the distributed optimization problem to be solved. This will then be followed by an introduction to the graph-based model for the communication network. The section will end by using this model to introduce the PI distributed optimization approach and briefly relate it to other methods.

A. Problem Formulation

We address the distributed optimization method in terms of the problem formulation presented in [6] and continued in [4], [5], [7], [10], [11]. It is assumed that there are N agents and that agent i is aware of only its own cost, $f_i(x)$, where $x \in \mathbb{R}^n$ is a vector of parameters being optimized. It is also assumed that the costs satisfy the following assumptions:

Assumption 1. *$f_i(x) : \mathbb{R}^n \to \mathbb{R}$, $i \in \{1, ..., N\}$, is a convex, twice continuously differentiable function, with the summation of all costs being strictly-convex*[1].

Assumption 2. *The solution $f^* = \min_x \sum_{i=1}^{N} f_i(x)$ and respective optimal parameter vector, x^*, exist.*

Email: {gregdroge,magnus}@gatech.edu.
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA.

[1] A locally strictly-convex function is sufficient for convergence to a local minimum of the cost.

To clarify the notation presented in following sections, one aspect of distributed optimization must be stressed. Most gradient-based methods, e.g. [4], [5], [6], [7], [10], [11], have agent i maintain its "own version" of the parameter vector, denoted as $x_i \in \mathbb{R}^n$, $i = 1, ..., N$, with the constraint that different versions be equal. This can be viewed as solving the following problem:

$$\begin{aligned} &\min_{x_i, i=1,...,N} \sum_{i=1}^{N} f_i(x_i). \\ &\text{s.t. } x_i = x_j \ \forall\ i, j \in \{1, ..., N\} \end{aligned} \tag{1}$$

To perform the optimization in a distributed manner, the equality constraints are relaxed. Algorithms differ in the components introduced to ensure that the agents return to the constraint set at the optimal value.

B. Graph-based Model for Networked Multi-agent Systems

A graph-based model for communication between agents is now introduced to facilitate the development of the PI distributed optimization algorithm. For sake of brevity, the very basics are given in this section. The reader is encouraged to see, for example, [1] for a complete development of graph-based multi-agent control design.

The communication topology at time t can be represented as an undirected graph, $\mathcal{G}(\mathcal{V}, \mathcal{E}_{i(t)})$, where the node $v_l \in \mathcal{V}$ corresponds to agent l and the edge-set, $\mathcal{E}_{i(t)} \subseteq \mathcal{E}_a = \{\mathcal{V} \times \mathcal{V}\}$ corresponds to agents which can communicate. For sake of simplicity, $\mathcal{E}_{i(t)}$ will be denoted as $\mathcal{E}_i$. Let the index set of all possible graph topologies be denoted by I, such that $\mathcal{E}_i \subseteq \mathcal{E}_a, \forall i \in I$. In a similar fashion, let the index set of all possible connected topologies be denoted as $I_c \subset I$, such that the graph $\mathcal{G}(\mathcal{V}, \mathcal{E}_i)$ is connected $\forall i \in I_c$. The final assumption used to ensure convergence can now be stated:

Assumption 3. *Let the graph at time t be given as $G(\mathcal{V}, \mathcal{E}_{i(t)})$ where $i(t) \in I_c$ $\forall t$.*

Two matrices derived from $\mathcal{G}(\mathcal{V}, \mathcal{E}_i)$ are essential to the formulation of PI distributed optimization. Allow the elements, $e_j \in \mathcal{E}_a$ $j \in J$, $J = 1, ..., |\mathcal{E}_a|$. The incidence matrix, $\mathcal{D}_i \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}_a|}$, is then defined for graph $\mathcal{G}(\mathcal{V}, \mathcal{E}_i)$, such that $\mathcal{D}_i = [d_{kj}]$,

$$[d_{kj}] = \begin{cases} 1 & \text{if } e_j \in \mathcal{E}_i \text{ and } v_k \text{ is the head of } e_j \\ -1 & \text{if } e_j \in \mathcal{E}_i \text{ and } v_k \text{ is the tail of } e_j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In other words, column j in $\mathcal{D}_i$ is dedicated to edge $e_j \in \mathcal{E}_a$ and is only non-zero if $e_j \in \mathcal{E}_i$. The incidence matrix is typically defined with the removal of all the non-zero columns, e.g. [1], [10], but this slight change from the norm will be beneficial for dynamic topologies.

The second matrix, the graph Laplacian, can be defined in terms of the incidence matrix. As the newly defined incidence matrix differs only in the addition of zero columns, the graph Laplacian can be expressed as $L_i = \mathcal{D}_i \mathcal{D}_i^T$, where normally the typical incidence matrix replaces $\mathcal{D}_i$ in the definition, e.g. [1].

Both the incidence matrix and the graph Laplacian are used to form larger, aggregate matrices to incorporate the fact that each agent will be maintaining an entire vector of values. First, let x_{ij} denote the j^{th} element of x_i, $z_j \triangleq [x_{1j}, x_{2j}, ..., x_{Nj}]^T$, and $z \triangleq [z_1^T, ..., z_n^T]^T \in \mathbb{R}^{Nn}$. The aggregate matrices can then be written as $\mathbb{D}_i \triangleq I_n \otimes \mathcal{D}_i$ and $\mathbb{L}_i \triangleq I_n \otimes L_i$, where I_n is the $n \times n$ identity matrix.

There are several properties of these two matrices which are exploited in the development and proof of PI distributed optimization. Namely, $\mathbb{L}_i = \mathbb{L}_i^T = \mathbb{D}_i \mathbb{D}_i^T \succeq 0$ and the eigenvectors associated with the zero eigenvalues of $\mathbb{L}_i$ can be written as $\alpha \otimes \mathbf{1}$, $\alpha \in \mathbb{R}^n$ (see [1], [10] for a development of these properties). Finally, note that the constraint that $x_i = x_j$ $\forall\ i, j \in \{1, ..., N\}$ can be written as $\mathbb{D}_i^T z = 0$ for all $i \in I_c$ [10]. This allows the distributed optimization problem to be presented in a compact form:

$$\begin{aligned} &\min_z f(z) = \sum_{i=1}^{N} f_i(x_i) \\ &\text{s.t. } h(z) = \mathbb{D}_i^T z = 0 \end{aligned}. \tag{3}$$

C. PI Dynamics for Distributed Optimization

Intuitively, a gradient approach to the optimization would lead to the dynamics $\dot{z} = \frac{\partial f}{\partial z}^T(z(t)) + \frac{\partial h}{\partial z}^T \lambda(t)$, where $\frac{\partial h}{\partial z}^T \lambda(t) = \mathbb{D}_i \lambda(t)$ is a component added to maintain the constraint. However, $\lambda(t)$ may be difficult to compute in a distributed fashion. The PI distributed optimization algorithm presented in [10] relaxes the constraint in (3) and modifies the dynamics of z to ensure that z returns to the constraint set at the optimal value.

Allowing $\lambda(t)$ to be a function of error defined over each edge as $e(t) = \mathbb{D}_i^T z$, proportional and integral control can be defined, $\lambda(t) = \frac{k_P}{k_G} e(t) + \frac{k_I}{k_G} \int_{t_0}^{t} e(s) ds$, to ensure zero steady-state error with a dampened response [10]. The resultant dynamics can be expressed for a fixed graph, $\mathcal{G}(\mathcal{V}, \mathcal{E}_i)$ as

$$\begin{aligned} \dot{z} &= -k_G \frac{\partial f}{\partial z}^T - k_P \mathbb{L}_i z - k_I' \mathbb{D}_i \mu \\ \dot{\mu} &= k_I' \mathbb{D}_i^T z. \end{aligned} \tag{4}$$

which, for distributed implementation, can be written as:

$$\dot{x}_i = -k_G \frac{\partial f_i}{\partial x}^T (x_i) - k_P \sum_{j \in \mathcal{N}_i} (x_i - x_j) - k_I' \sum_{j \in \mathcal{N}_i} \mu_i^j$$
$$\dot{\mu}_i^j = k_I'(x_i - x_j), \tag{5}$$

where $\mathcal{N}_i$ denotes the set of indices of agent i's neighbors. Note that the dynamics for μ_i^j have the undesirable requirement (especially as we consider dynamic topologies) that agent i must remember the contribution agent j has made to agent i's version of the variables.

While this approach may appear ad-hoc, it was shown in [10] that the proportional component corresponds exactly to the consensus terms considered in [4], [6]. Moreover, the integral portion corresponds to the dual-decomposition term as given in [9]. Interestingly, the method presented in [5], and extended in [11] to directed networks, can also be related to PI control, but with a different representation of the constraint which adds an extra level of communication, [7], [10]. By using the incidence matrix to represent the constraint, the algorithm will be extended to dynamic networks.

III. PI Distributed Optimization for Switching Topologies

An important aspect to consider in multi-agent systems is that the underlying communication topology can vary, changing which agents are capable of communicating. Therefore, this section will exploit the newly defined incedience matrix in (2) to show convergence for the dynamics in (4) under changing topologies.

A. Convergence for Static Topologies

Under static topologies, the proofs given for convergence in [10] hold, even with the newly defined incidence matrix. However, the Lyapunov function used in the proof depends on the dynamics of the system which depend on the incidence matrix, adding complications for dynamic topologies. Therefore, we present an alternative Lyapunov function which is more readily extensible to dynamic topologies.

Theorem 1. *Given that $\mathcal{G}(\mathcal{V}, \mathcal{E}_i)$ forms a static, connected graph and Assumptions 1 and 2 hold, the dynamics given in (4) will cause (z, μ) to converge to the optimal values of the saddle-point problem, (z^*, μ^*), defined by*

$$\max_{\mu} \min_{z} (f'(z) + \mu^T \mathbb{D}_i^T z) \tag{6}$$

Proof: It was shown in [10] that at equilibrium $\dot{z} = 0, \dot{\mu} = 0$ corresponds to the optimal values, (z^*, μ^*). Define $\tilde{z} = z - z^*$, $\tilde{\mu} = \mu - \mu^*$ and the function $f'(z) = k_G f(z) + \frac{k_P}{2} z^T \mathbb{L}_i z$. Note that due to Assumption 1 and the fact that $\mathbb{L}_i \succeq 0 \Rightarrow z^T \mathbb{L}_i z$ is convex, $f'(z)$ is strictly convex, (see [8] for properties of convex functions). The proof from [7] can be readily extended to show that the candidate Lyapunov function:

$$V(\tilde{z}, \tilde{\mu}) = \frac{1}{2} \tilde{z}^T \tilde{z} + \frac{1}{2} \tilde{\mu}^T \tilde{\mu}, \tag{7}$$

has a non-positive time derivative. In fact, $\dot{V}(\tilde{z}, \tilde{\mu}) < 0$ $\forall \tilde{z} \neq 0$ and $\dot{V}(\tilde{z}, \tilde{\mu}) \leq 0$ for $\tilde{z} = 0$.

LaSalle's invariance theorem can then be invoked, e.g. [12]. Denote the smallest invariant set, $V_0 = \{(\tilde{z}, \tilde{\mu}) | \dot{V} = 0\} = \{(0, \mu), \forall \mu\}$. In V_0, $z = z* \Rightarrow \dot{z} = \ddot{z} = 0$. Note that $\ddot{z} = -k_G \frac{\partial f}{\partial z}^T \dot{z} - k_P \mathbb{L}_i \dot{z} - k_I' \mathbb{D}_i \dot{\mu} = -k_I \mathbb{D}_i \mathbb{D}_i^T z = -k_I \mathbb{L}_i z = 0$ which implies $z = \alpha \otimes \mathbf{1}$ where $\alpha \in \mathbb{R}^n$ and $\dot{\mu} = \mathbb{D}_i^T z = 0$. Therefore, the control law converges. ■

B. Convergence for Dynamic Topologies

The key factor that allows for a convergence proof for dynamic topologies is the redefinition of the incidence matrix in (2). Since (2) has a specific column dedicated to each possible edge, the elements in μ will always correspond to the integral of the error across the corresponding edge and the dimension of μ will never change, neither of which is true using the typical definition of the incidence matrix. This in turn allows in turn allows V to be continuous across switch times.

One final assumption concerning dwell time is required before giving a theroem about convergence (i.e. the system persistently encounters static topologies for intervals of length at least $\tau > 0$). It is adapted from Assumption 3 in [13] to fit the multi-agent scenario:

Assumption 4. *There exists $\tau > 0$ such that for every $T \geq 0$ a positive integer i can be found for which $t_{i+1} - \tau \geq t_i \geq T$, where t_i denotes the i^{th} switch time.*

Theorem 2. *Given that Assumptions 1, 2, 3, and 4 hold, the dynamics given in (4) will cause (z, μ) to converge to the optimal values, (z^*, μ^*), defined by (6).*

Proof: To show convergence, we invoke the LaSalle invariance principle for hybrid systems stated in Theorem 7 of [13]. To do so, the system must satisfy four conditions. Theorem 1 satisfies the first two conditions which require each set of dynamics for the switched system (corresponding to different network topologies in our case) to have a weak Lyapunov function which can be shown to converge to the equilibrium. The third condition concerns dwell time and is satisfied by Assumption 4. The final condition concerns non-increasing values

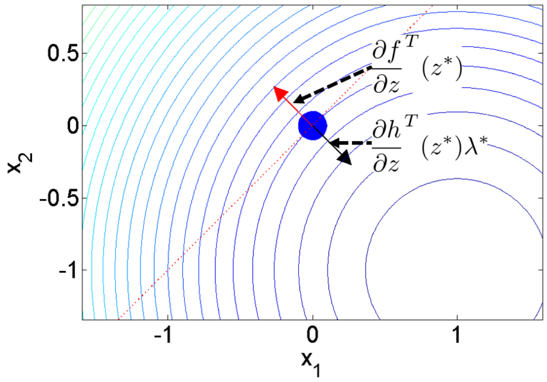

Fig. 1. This figure shows the optimal point to the cost $f(z) = (x_{11}-1)^2+(x_{21}+1)^2$ with constraint $x_{11} = x_{21}$. The unconstrained gradient, $\frac{\partial f}{\partial z}^T(z^*)$, is balanced by the vector $\frac{\partial h}{\partial z}^T(z^*)\lambda^*$

for the Lyapunov functions across switching, which is trivially satisfied as all topologies have a common Lyapunov function. ■

IV. Index-free PI Distributed Optimization

While the previous section proved that the dynamic update law given in (4) will converge to the optimal value, it is important to note that the individual dynamics given in (5) form an undesirable solution. The reason being is that the dynamic update law requires each agent to "remember" the individual contribution that every other agent has made to the integral of the error. To create an index free solution, we take a step back and evaluate the problem being solved in (3).

The structure of (3) is nothing more than a convex optimization with a linear constraint. It is well know, e.g. [14], that such a problem will have a solution satisfying:

$$0 = \frac{\partial f}{\partial z}(z^*) + \lambda^{*T}\frac{\partial h}{\partial z}, \tag{8}$$

where z^* is the optimal solution and λ^* is the optimal value for the dual problem, as shown in Figure 1. In taking a closer look at (8), the true value needing to be solved for is a vector which "offsets" the gradient, $\frac{\partial f}{\partial z}$, at the optimal value, as shown in Figure 1. In other words,

$$0 = \frac{\partial f}{\partial z}(z^*) + \nu^T, \tag{9}$$

where in actuality $\nu^T = \lambda^{*T}\frac{\partial h}{\partial z}$.

In some sense, the distributed optimization algorithm consists of the agents working together to share information in order to collectively learn the value of ν. Thus, this leads to the idea of re-formulating the distributed optimization algorithm in terms of ν as follows:

$$\dot{z} = -k_G\frac{\partial f}{\partial z}^T - k_P\mathbb{L}_i z - \nu \tag{10}$$

where the dynamics for ν can be written as

$$\dot{\nu} = k_I\mathbb{L}_i z. \tag{11}$$

In the same manner as done in (5), the aggregate dynamics can be split up between the agents in a distributed fashion as follows:

$$\begin{aligned} \dot{x}_i &= -k_G\frac{\partial f_i}{\partial x}^T(x_i) - k_P\sum_{j\in\mathcal{N}_i}(x_i - x_j) - \nu_i \\ \dot{\nu}_i &= k_I\sum_{j\in\mathcal{N}_i}(x_i - x_j), \end{aligned} \tag{12}$$

It is important to note that we have removed the agent indexing present in (5) *In (12), agent i no longer needs to keep track of agent j's contribution, rather agent i need only keep track of the aggregate contribution to the error by its neighbors.* A theorem is now given about the convergence of the newly formed dynamic update law:

Theorem 3. *Given that Assumptions 1, 2, 3, 4 and $\nu(0) = 0$, the dynamics given in (11) and (12) will cause z to converge to the optimal value, z^*, defined by*

$$\begin{aligned} &\min_z f(z) \\ \textit{s.t. } x_i =& x_j \forall i,j \in [1,...,N] \end{aligned}$$

Proof: The proof of the theorem hinges upon the fact that the state dynamics, $\dot{z}(t)$, for the aggregate state and $\dot{x}_i(t)$ for individual agents remain unchanged. Previously we had

$$\begin{aligned} \dot{\mu} &= k_I'\mathbb{D}_i^T z \\ \dot{z} &= -k_G\frac{\partial f}{\partial z}^T - k_P\mathbb{L}_i z - k_I'\mathbb{D}_i\mu \\ &= -k_G\frac{\partial f}{\partial z}^T - k_P\mathbb{L}_i z - k_I'\mathbb{D}_i\int_0^t k_I'\mathbb{D}_i^T z(s)ds \\ &= -k_G\frac{\partial f}{\partial z}^T - k_P\mathbb{L}_i z - k_I\mathbb{L}_i\int_0^t z(s)ds \end{aligned} \tag{13}$$

and corresponding agent state dynamics

$$\begin{aligned} \dot{\mu}_i^j =& k_I'(x_i - x_j) \\ \dot{x}_i =& -k_G\frac{\partial f_i}{\partial x}^T(x_i) - k_P\sum_{j\in\mathcal{N}_i}(x_i - x_j) - k_I'\sum_{j\in\mathcal{N}_i}\mu_i^j \\ =& -k_G\frac{\partial f_i}{\partial x}^T(x_i) - k_P\sum_{j\in\mathcal{N}_i}(x_i - x_j) - \\ & k_I\sum_{j\in\mathcal{N}_i}\int_0^t(x_i(s) - x_j(s))ds. \end{aligned} \tag{14}$$

The newly formed dynamics can be stated in a similar fashion:

$$\begin{aligned}\dot{\nu} &= k_I \mathbb{L}_i z \\ \dot{z} &= -k_G \frac{\partial f}{\partial z}^T - k_P \mathbb{L}_i z - \nu \\ &= -k_G \frac{\partial f}{\partial z}^T - k_P \mathbb{L}_i z - \int_0^t k_I \mathbb{L}_i z(s) ds \\ &= -k_G \frac{\partial f}{\partial z}^T - k_P \mathbb{L}_i z - k_I \mathbb{L}_i \int_0^t z(s) ds\end{aligned} \tag{15}$$

which is the same as (13). Similarly, the individual agent states can be written as

$$\begin{aligned}\dot{\nu}_i =& k_I \sum_{j \in \mathcal{N}_i} (x_i - x_j) \\ \dot{x}_i =& -k_G \frac{\partial f_i}{\partial x}^T (x_i) - k_P \sum_{j \in \mathcal{N}_i} (x_i - x_j) - \nu_i \\ =& -k_G \frac{\partial f_i}{\partial x}^T (x_i) - k_P \sum_{j \in \mathcal{N}_i} (x_i - x_j) - \\ & k_I \sum_{j \in \mathcal{N}_i} \int_0^t (x_i(s) - x_j(s)) ds\end{aligned} \tag{16}$$

which is the same as (14). Therefore, because the dynamics for z remain unchanged, z will converge to z^*. Also, note that because z converges to z^*, $\dot{\nu} \longrightarrow 0$ from the fact that $\mathbb{L}_i z^* = 0$. ■

One final note worth making is that it was shown in both [7] and [10] that when each agent does not have an opinion about a parameter in the parameter vector, the problem can be simplified to eliminate redundancies. Allowing $I_j = \{i | f_i \text{ depends on the element } j\}$ to be the set of agents which depend on element j the dynamics in (12) can be simplified in the exact way that (5) was simplified in [10]. As the arguments are the same, we simply state the modified dynamics for agent i's version of variable j as:

$$\begin{aligned}\dot{x}_{ij} &= -k_G \frac{\partial f_i}{\partial x_{ij}} - k_P \sum_{k \in \{\mathcal{N}_i \cap I_j\}} (x_{ij} - x_{kj}) - \nu_{i,j} \\ \dot{\nu}_{i,j} &= k_I \sum_{k \in \{\mathcal{N}_i \cap I_j\}} (x_{ij} - x_{kj}).\end{aligned} \tag{17}$$

Basically, this results in each variable only being maintained and updated by the agents which actually have an opinion about the variable.

V. Example: Formation Control

To demonstrate the ability for PI distributed optimization to achieve a collective objective utilizing local information, this section introduces an example of formation control based upon a relative state formulation, e.g. [1]. The basic idea being that a formation control problem can be defined by a nominal position for each agent, $y_i \in \mathbb{R}^2$. The agents must come to an agreement upon a translation, $\tau \in \mathbb{R}^2$, from the nominal position as well as a possible rotation, $\theta \in \mathbb{R}$, about the nominal origin, as well as a scaling ($\gamma \in \mathbb{R}^+$) of the formation as depicted in Figure 2. While [1] introduces methods based on feedback control to solve for τ, we show that distributed optimization can be utilized to solve for the various parameters simultaneously.

To choose the parameters, $x = [\tau^T, \theta, \gamma]^T$, the agents perform PI distributed optimization. The cost assigned to each agent takes the form

$$f_i(x_i(t)) = \frac{1}{2} ||q_i(t) - q_{d_i}(t)||^2 + k(\gamma_i(t) - 1)^2 \tag{18}$$

where $q_i(t) \in \mathbb{R}^2$ is agent i's position at time t, k is a weight on the scaling, and $q_{d_i}(t) \in \mathbb{R}^2$ is the desired position of agent i. Assuming that the nominal formation is defined with the center at the origin, $q_{d_i}(t)$ can be expressed as

$$q_{d_i}(t) = R(\theta_i(t))\gamma_i(t) y_i + \tau_i(t), \tag{19}$$

where $R \in \mathbb{R}^{2 \times 2}$ is a rotation matrix. Each variable in (18) and (19) is written as a function of time to emphasize that the variables are continually being updated.

The cost defined in (18) has two terms to guide the selection of the parameters. The first term penalizes the distance between the current position and desired position. The second term penalizes deviation from unit scaling where k is only non-zero for the GRITS example. Note that (18) is only locally convex, so agents will converge to some local minima.

While agents are optimizing they are also moving towards their respective desired position. Using integrator dynamics, $\dot{q}_i(t) = u_i(t)$, for each agent, the feedback law, $u_i(t) = q_{d_i}(t) - q_i(t)$, is used to move towards the desired position. It is also assumed that the underlying graph topology is a δ-disk graph, e.g. [1], where agents i and j are only able to communicate at time t if $\|q_i(t) - q_j(t)\| \leq \delta$.

Three examples are shown in Figure 2 and Table IV. Agents form a diamond, a line formation, and finally a GRITS formation to demonstrate the ability to specify arbitrary formations. Each example shows a significant decrease in both the average distance and standard deviation of the distance that each agent was required to travel.

VI. Conclusion

We have extended the PI distributed optimization method first presented in [10] to account for dynamic

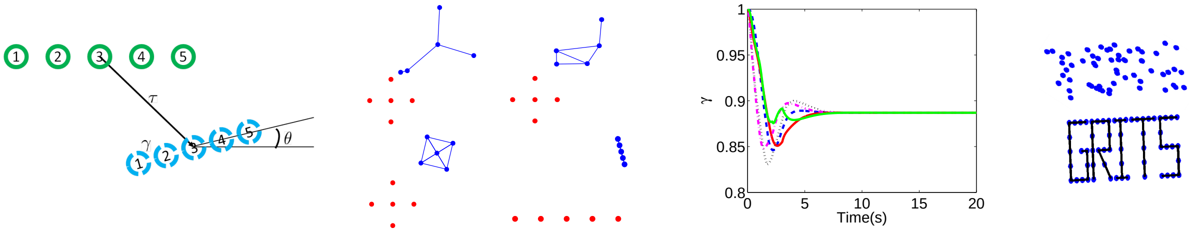

Fig. 2. The left image is a diagram showing the translation (τ), rotation (θ), and scaling (γ) from the nominal formation. The middle left image shows shapshots of agents in the line and diamond formations. The top row shows the agents while converging to the diamond formation. The bottom row shows the resulting diamond and line formations. The lines between agents show the communication topology and the bottom left of each figure is shown the nominal configuration. The middle right figure shows each agents' version of γ while converging to the diamond formation to give an example of convergence characteristics. On the right is shown 60 agents forming the word 'GRITS'. The top and bottom images show the initial and final positions, respectively.

TABLE I
THIS TABLE SHOWS THE RESULTING PARAMETERS AND DISTANCES FOR EACH SIMULATION. THE NOMINAL DISTANCE REFERS TO THE DISTANCE FROM AGENT'S STARTING POSITION TO THEIR NOMINAL POSITION. THE TRAVEL POSITION IS HOW FAR THE AGENTS ACTUALLY TRAVELLED TO REACH FORMATION.

	τ	γ	θ	Nominal Ave Distance	Nominal Std Dev	Ave Travel Distance	Travel Std Dev
Line	(2.12, 2.60)	0.25	4.97	4.03	0.98	1.28	0.67
Diamond	(2.09, 2.39)	0.89	2.12	3.35	2.11	1.49	0.57
GRITS	(1.44, 2.27)	0.98	-1.53	3.12	1.46	2.07	1.00

topologies. This has been accomplished by redefining the incidence matrix to have a column dedicated to each edge, making it possible to define a Lyapunov function that is continuous across switching topologies. However, the adjustment of the incidence matrix makes the actual implementation undesirable as it requires agents to remember the contribution every other agent has ever made to its error. By re-examining the underlying constrained optimization problem, it was shown that it was possible to reformulate the algorithm so agents solely keep track of the aggregate contribution of their neighbors.

To demonstrate the ability of PI distributed optimization to cope with changing communication topologies while maintaining convergence properties, we examined an example of formation control. Agents simultaneously moved and optimized and were able to come to agreement on several parameters in order to determine where the formation would end up. On average each agent traveled less than they would have had to in order to get to the nominal position and the distance traveled by one agent was much closer to that traveled by another.

REFERENCES

[1] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton Univerisity Press, 2010.

[2] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *Transactions-American Society of Mechanical Engineerers Journal of Dynamic Systems Measurement and Control*, vol. 129, no. 5, p. 571, 2007.

[3] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *IEEE Conference on Decision and Control*, 2012.

[4] K. Kvaternik and L. Pavel, "A continuous-time decentralized optimization scheme with positivity constraints," in *IEEE Conference on Decision and Control*, 2012.

[5] J. Wang and N. Elia, "Control approach to distributed optimization," in *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2010, pp. 557–561.

[6] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[7] J. Wang and N. Elia, "A control perspective for centralized and distributed convex optimization," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011.

[8] S. Boyd, *Convex Optimization*. Cambridge University Press, 2004.

[9] H. Terelius, U. Topcu, and R. Murray, "Decentralized multi-agent optimization via dual decomposition." IFAC, 2011.

[10] G. Droge, H. Kawashima, and M. Egerstedt, "Continuous-time proportional-integral distributed optimization for networked systems," *Journal of Control and Decision*, Under Review. Available in preprint at http://arxiv.org/abs/1309.6613.

[11] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, To appear. Preprint available at http://arxiv.org/abs/1204.0304.

[12] H. Khalil, *Nonlinear systems. 3rd ed.* Prentice hall, 2002.

[13] J. P. Hespanha, D. Liberzon, D. Angeli, and E. D. Sontag, "Nonlinear norm-observability notions and stability of switched systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 154–168, 2005.

[14] D. Luenberger and Y. Ye, *Linear and nonlinear programming (3rd ed.)*. Springer, 2008, vol. 116.