A Graph Theoretic Approach to Optimal Target Tracking for Mobile Robot Teams

Jason C. Derenick, John R. Spletzer and M. Ani Hsieh

***Abstract*— In this paper, we present an optimization framework for target tracking with mobile robot teams. The target tracking problem is modeled as a generic semidefinite program (SDP). When paired with an appropriate objective function, the solution to the resulting problem instance yields an optimal robot configuration for target tracking at each time-step, while guaranteeing target coverage (each target is tracked by at least one robot) and maintaining network connectivity. Our methodology is based on the graph theoretic result where the second smallest eigenvalue of the interconnection graph Laplacian matrix is a measure for the connectivity of the graph. This formulation enables us to model agent-target coverage and inter-agent communication constraints as linear-matrix inequalities. We also show that when the communication constraints can be relaxed, the resulting problem can be reposed as a second-order cone program (SOCP) which can be solved significantly more efficiently than its SDP counterpart. Simulation results for a team of robots tracking multiple targets are presented.**

I. INTRODUCTION

We are interested in developing robot teams for use in surveillance and monitoring applications. The idea of using teams of small, inexpensive robotic agents to accomplish various tasks is one that has gained increasing currency as embedded processors and sensors become smaller, more capable, and less expensive. To this point, much of the work in multi-robot coordination has focused on control and perception. It has generally been assumed that each team member has the ability to communicate with any other member with little to no consideration for the the quality of the wireless communication network. Such an assumption, although valid in certain situations, does not generally hold - especially when considering the deployment of robot teams within unstructured and unpredictable environments.

Our previous work in target tracking made similar simplifying assumptions, as no constraints were placed on sensing and communication ranges [1]. This allowed target coverage and network connectivity requirements to be ignored in order to simplify the optimization process. In this paper however, we consider the problem of controlling the configuration of a team of mobile agents for target tracking *under both coverage and communication constraints*. Our methodology is based on the graph theoretic result where the second smallest eigenvalue of the interconnection graph Laplacian matrix is a measure for the connectivity of the graph. Recent system and control literature has shown that the maximization of the second smallest eigenvalue for a state dependent graph Laplacian matrix can be formulated as a semidefinite program [2]. We apply these results to the target tracking task and obtain a coordination strategy that maintains target coverage and network connectivity while optimizing a given objective function. Specifically, robot-target assignments and inter-agent communication constraints are embedded in *visibility* and *network* graphs, respectively. The target tracking problem is then formulated as a SDP where the coverage and communication constraints are modeled as linear-matrix inequalities (LMI).

An important advantage of this formulation is that it is agnostic to the quality metric being optimized. So long as the objective function is convex, and can be expressed in terms of linear, quadratic, or semidefinite constraints, the resulting problem will be a SDP. The convexity of semidefinite programs ensures the problem solution will be globally optimal, and solvable in polynomial time in the number of robots and targets. We also show that when communication constraints must be relaxed to ensure target coverage (e.g. to track diverging/evasive targets), the problem can be reposed as a second order cone program (SOCP) that can be solved significantly more efficiently than its SDP counterpart.

II. RELATED WORK

In the last several years, increased attention has been focused on the effects of communication networks in multi-agent teams. Earlier works generally assumed static communication ranges, [3], and/or relied on coordination strategies that require direct line-of-sight, [4]. In [5] and [6] decentralized controllers were used for concurrently moving toward goal destinations while maintaining communication constraints by maintaining line-of-sight and assuming static communication/sensor ranges respectively. Coordination strategies based on inter-agent signal strength include [7], [8], and [9]. In [10], low-level reactive controllers capable of responding to changes in signal strength or estimated available bandwidth are used to constrain robots' movements in surveillance and reconnaissance tasks. Although much of the recent works have focused on the effects of communication maintenance on navigation, few have addressed the issue of communication maintenance in tasks such as collaborative/collective localization and data fusion where team connectivity is essential to the team's ability to accomplish the given task.

Previous works in collaborative target localization include [11], [12], and [13] where strategies such as maintaining

J. C. Derenick and J. R. Spletzer are with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015, USA {derenick,josa}@lehigh.edu

M. A. Hsieh is with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA mya@seas.upenn.edu

visibility constraints and determining optimal sensor placement are considered. More recent works include [14], where distributed control strategies are used to minimize the amount of time targets are outside of mobile agents' sensor ranges. In [15], artificial potential functions are used to coordinate a team of mobile agents to track multiple moving targets. [16] addresses the same problem by formulating it as two sub-problems: target tracking for a single robot and on-line motion coordination strategy for a team of robots. In [17], the authors consider a motion coordination strategy to enable a team of mobile sensors to detect multiple targets within a given region. [18] analyses the accuracy of cooperative localization and target tracking in a team of mobile robots using an extended Kalman Filter formulation and provide upper bounds for the position uncertainty obtained by the team. In [19], a distributed control strategy is used maintain a team of mobile agents in a mesh formation to enable tracking of a discrete or diffused target. In [1], [20], [21], the authors employ particle filters to minimize the expected error in tracking mobile target(s) for a team of mobile robots without the use of explicit switching rules.

Lastly, maximizing the second smallest eigenvalue of the graph Laplacian for maintaining team connectivity has also been considered in both [2] and [22]. The former is perhaps most related to our work, where the problem of finding optimal positions for a set of nodes such that connectivity of the nodes are maintained is formulated as a SDP. The latter uses a distributed algorithm to enable a team of mobile agents to increase the connectivity of the team.

In contrast to these efforts, we propose a SDP formulation for controlling the configuration of a team of mobile agents for tracking moving targets while maintaining *both* sensing and communication constraints and optimizing an additional tracking objective. Furthermore, in situations when one is willing to forgo communication maintenance to ensure complete coverage of all targets, we show how our formulation can be simplified into a SOCP. This is relevant in situations when communication with other team members must be sacrificed to ensure all targets are appropriately tracked.

III. Problem Statement

The objective of this paper is to provide a general framework that facilitates optimal target tracking with performance guarantees. More precisely, we consider minimizing some *convex* objective function, $\Psi\colon \mathbb{R}^{3n} \to \mathbb{R}$, while ensuring:

1) Complete target coverage, where every target is tracked by at least a single agent.
2) Network connectivity across the robot formation to facilitate robot coordination.

In this case, we define Ψ as a function of our decision variable $X = (x_1^a, x_2^a, \ldots, x_n^a)^T \in \mathbb{R}^{3n}$. Here, X denotes the concatenated positions of the robot team with $x_i^a \in \mathbb{R}^3$ representing the location of agent i with respect to some world frame $\mathcal{W}$. Additionally, we assume a fully actuated robot model for each member of the team, *i.e.* where

$$\dot{x} = u \quad u \in \mathcal{U} \subseteq \mathbb{R}^2 \tag{1}$$

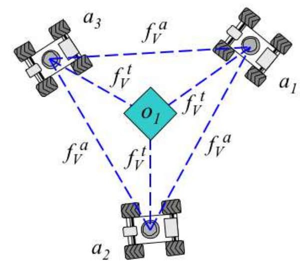

Fig. 1. The weighted visibility graph, $G_V(\mathcal{V}_V, \mathcal{E}_V)$, for a team of three robots observing a single target in $\mathbb{R}^2$. In our formulation, respective edge weights are a function of the team's positional state vector, X.

For the moment, we avoid discussion of Ψ and instead focus on formulating a sufficient constraint set to ensure that the optimal state vector X guarantees coverage of all targets.

A. A State-dependent Graph Representation

To this end, we begin by exploiting the fact that a team of robots and the targets they track collectively define a weighted graph where an agent-to-target edge corresponds to a single point-to-point sensor track. More precisely, letting $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ and $\mathcal{O} = \{o_1, o_2, \ldots, o_m\}$ respectively denote the full set of agents and the full set of observation targets, we define the graph $G_V(\mathcal{V}_V, \mathcal{E}_V)$ where $\mathcal{V}_V = \mathcal{A} \cup \mathcal{O}$ and $\mathcal{E}_V = \{e\colon e \in \mathcal{A} \times \mathcal{A} \cup \mathcal{O}\}$. We refer to this graph as the *visibility graph* and associate with its edges a mapping $f_V\colon \mathcal{E}_V \to \mathbb{R}_+$.

Letting $x_j^t \in \mathbb{R}^3$ denote the position of observation target o_j in world coordinate frame $\mathcal{W}$, we define

$$f_V(y) = \begin{cases} f_V^a(\| x_i^a - x_j^a \|_2), & y = (a_i, a_j) \in \mathcal{A} \times \mathcal{A} \\ f_V^t(\| x_i^a - x_j^t \|_2), & y = (a_i, o_j) \in \mathcal{A} \times \mathcal{O} \end{cases} \tag{2}$$

where $0 \leq f_V^a(x_i^a, x_j^a), f_V^t(x_i^a, x_j^t) \leq 1$. In other words, the weights of the corresponding edges are a direct functional of the relative Euclidean distance separating an agent from some other observable network entity. Notice that this also implicitly makes G_V a function of the positional state vector X, and as such, we accordingly denote it $G_V(X)$.

B. Ensuring Complete Target Coverage

Given the definition of $G_V(X)$, observe that all targets in the system are tracked whenever the graph itself is connected, as this implies active links between all targets and at least one member of the agent team. In other words, we would like our constraint set to capture and preserve this notion of connectivity when determining the optimal state vector X.

With this in mind, we turn our attention to recent results from spectral graph theory regarding the *connectivity* of an arbitrary graph $G(\mathcal{V}, \mathcal{E})$. In particular, we note that the constraint $\lambda_2(L(G)) > 0$ is both a necessary and sufficient condition for *guaranteeing the connectivity* of G [23], where $\lambda_2(L(G))$ denotes the second smallest eigenvalue of the weighted graph Laplacian $L(G)$ given by

$$[L(G)]_{ij} = \begin{cases} -w_{ij}, & i \neq j \\ \sum\limits_{i \neq k} w_{ik}, & i = j \end{cases} \tag{3}$$

with w_{ij} being the weight associated with the edge shared between vertices i and j.

In light of these observations, we can now pose the following initial formulation for the target tracking problem

$$\begin{aligned} &\min \Psi(X) \\ &\text{s.t. } \lambda_2(L_V(X)) > 0 \end{aligned} \tag{4}$$

where $L_V(X)$ denotes the state-dependent Laplacian of the visibility graph $G_V(X)$.

Noting the results of [2], we see that

$$\lambda_2(L_V(X)) > 0 \equiv P_V^T L_V(X) P_V \succ 0 \tag{5}$$

where $P_V \in \mathbb{R}^{(n+m)\times(n+m-1)}$ comprises an orthonormal basis for an $n+m-1$ dimensional subspace such that $\forall x \in$ **span**$(P_V), 1^T x = 0$. As such, we can further solidify the problem statement by reposing (4) as follows

$$\begin{aligned} &\min \Psi(X) \\ &\text{s.t. } P_V^T L_V(X) P_V \succ 0 \end{aligned} \tag{6}$$

In this formulation, we adopt the standard Löwner ordering.

C. Enforcing Network Connectivity Constraints

Although, solving (6) yields a positional configuration that will ensure all nodes are tracked, it makes no guarantees regarding the underlying network connectivity of the robot team. The ability to ensure a connected network graph while performing such a task is often desirable as it facilitates – among other things – distributed sensor fusion. To address this, we extend our formulation by introducing a network proximity graph $G_N(\mathcal{V}_N, \mathcal{E}_N)$ where $\mathcal{V}_N = \mathcal{A}$ and $\mathcal{E}_N = \{e\colon e \in \mathcal{A} \times \mathcal{A}\}$. Similar to the previous graph formulation, we associate a weight function $f_N\colon \mathcal{E}_N \to \mathbb{R}_+$ that is a direct functional of the Euclidean distance between network peers.

Given this definition, we can augment (6) accordingly to yield the following problem statement

$$\begin{aligned} &\min \Psi(X) \\ &\text{s.t. } P_V^T L_V(X) P_V \succ 0 \\ &\qquad P_N^T L_N(X) P_N \succ 0 \end{aligned} \tag{7}$$

Solving this problem, will yield an optimal positional configuration for the team that maintains both network connectivity as well as complete target coverage. However, it should be noted that (7) is not necessarily a convex optimization problem. In fact, the convexity of (7) hinges upon the choice of weight functions f_V^a, f_V^t and f_N. As we shall see, this fact does not prevent us from formulating the target tracking problem as a discrete-time process whereby during each iteration a convex form of (7) is solved.

IV. Defining Interactive Control Functions

Given this formulation, we now consider appropriate definitions of f_V^a, f_V^t and f_N. The choice of these functions is critical as they inherently govern the behavior of the team. As these functions dictate the relationship between one node and another as well as any observation targets, we see that at the highest level that they can be considered *interactive control* functions. With this in mind, we now consider appropriate choices for a simple target tracking scenario.

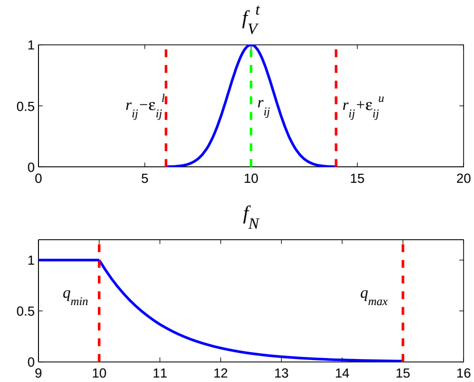

Fig. 2. (Top) An instance of f_V^a where $r_{ij} = 10$ and $\epsilon_{ij}^l = \epsilon_{ij}^u = 4, \forall i, j$. For our purposes, a simple symmetric Gaussian was utilized to govern agent-target interactions; however, a different potential function could have just as easily been used. (Bottom) An instance of a simple exponential-decay model for RF links with $q_{min} = 10$ and $q_{max} = 15$.

Momentarily neglecting discussion of f_V^a, we focus instead on characterizing the desirable properties of weight function f_V^t, which governs the interactions of systems agents and respective targets. In an ideal tracking scenario the team will have an optimal number of tracks while each agent maintains a safe standoff distance between itself and its targets. In other words, letting r_{ij} denote the desired distance between agent a_i and target o_j, we would like f_V^t to promote the team to behave such that

$$|r_{ij} - \epsilon_{ij}^l| \le d_{ij}^t \le |r_{ij} + \epsilon_{ij}^u|, \forall i, j \tag{8}$$

where $d_{ij}^t \triangleq \| x_i^a - x_j^t \|_2$ and $\epsilon_{ij}^l, \epsilon_{ij}^u \in \mathbb{R}_+$ dictate the acceptable lower/upper bound tolerances for any tracks between agent a_i and observation target o_j. When the right-hand inequality holds, we define the track as being *active*.

Seeking inspiration from the motion-planning community, we opt to consider a formulation of f_V^t modeled after common potential functions. These functions are especially attractive for our framework as they often offer nice differentiable properties and facilitate easy modeling of (8).

For our purposes, we consider the case $\epsilon_{ij}^l = \epsilon_{ij}^u$ and model f_V^t using a standard symmetric Gaussian potential – noting that the subsequent analysis can easily be adapted for an alternate (symmetric or non-symmetric) formulation using an appropriate potential function variant. That stated, we consider the following definition

$$f_V^t(x_i^a, x_j^t) = \begin{cases} e^{-\frac{\gamma^2}{2}(d_{ij}^t - r_{ij})^2}, & |d_{ij}^t - r_{ij}| \le \epsilon \\ 0, & \textit{otherwise} \end{cases} \tag{9}$$

where $\gamma = \sqrt{\frac{7}{\epsilon}}$. Figure 2 (Top) illustrates this function for an instance with $r_{ij} = 10$ and $\epsilon = 4$.

Regarding f_V^a, which governs inter-agent behaviors, we see its definition is not as obvious since a variety of formulation may lead to favorable results depending upon the chosen application and mission objectives. For instance, we can choose $f_V^a = 1$ which has the effect of removing any inter-agent observability requirements as it essentially says that no matter what the positional state of the team, the inter-agent links are connected or are observable everywhere. Whether this is feasibly possible given the agents' respective sensor suites is inconsequential for the task at hand, as in choosing the weights this way we are only concerned with ensuring complete target coverage. Another reasonable choice for f_V^a is a potential function that behaves similarly to that used to define f_V^a. This definition would be useful in a scenario where team members rely upon local observations of their peers for such things as localization. In this paper, we adopt the former definition of f_V^a.

In a similar manner, we can now address the issue of weighting the network links in $G_N(X)$. The weights of these links are very easily characterized, and such a formulation has been addressed in recent literature [2], [22]. For our purposes as well as for the sake of further discussion, we consider the exponential decay model posed by [22]. Doing so yields the following formulation for f_N

$$f_N(x_i^a, x_j^a) = \begin{cases} 1, & d_{ij}^a \leq q_{min} \\ e^{\frac{-5(d_{ij}^a - q_{min})}{q_{max} - q_{min}}}, & q_{min} < d_{ij}^a \leq q_{max} \\ 0, & d_{ij}^a > q_{max} \end{cases} \quad (10)$$

Figure 2 (Bottom) shows a single instance of f_N for $q_{min} = 10$ and $q_{max} = 15$.

V. Defining a Discrete Semi-definite Approach

In this section, we consider formulating our problem as a discrete time process whereby the agent team collectively observes the relative positions of the observation targets and then accordingly adjusts their respective trajectories so as to minimize the given objective. As the targets are assumed dynamic and control is inherently a discrete-time process, we see that at best the team can only optimize Ψ over the period Δt representing the rate at which they are able to effectively sample the environment and issue control signals. Although this approach does not guarantee optimally-convergent behavior for the team, it *does ensure* that the solution obtained will yield a trajectory that is optimal with respect to Ψ over that timestep.

A. Problem Formulation

With this in mind, we leverage the results of [2] who considered a discrete time process for maximizing network connectivity in multi-agent teams. Following suit, we perform a simple differentiation with respect to time and then apply Euler's first-order discretization method. Doing so reveals the following discrete-time representation of f_V^t

$$f_V^t(x_i^a, x_j^t)(k+1) - f_V^t(x_i^a, x_j^t)(k) = \tau_{V_k}^t \left\{ x_i^a(k) - x_j^t \right\}^T \Delta x_i^a \quad (11)$$

$$\tau_{V_k}^t = -\frac{\gamma^2(\| x_i^a(k) - x_j^t \|_2 - r_{ij})}{\| x_i^a(k) - x_j^t \|_2} f_V^t(x_i^a(k), x_j^t)$$

where $\forall l, a_l \in \mathcal{A}$, we have $\Delta x_l^a = x_l^a(k+1) - x_l^a(k)$.

Similarly for f_N, we obtain

$$f_N(x_i^a, x_j^a)(k+1) - f_N(x_i^a, x_j^a)(k) = \tau_{N_k} \left\{ x_i^a(k) - x_j^a(k) \right\}^T \left\{ \Delta x_i^a - \Delta x_j^a \right\} \quad (12)$$

$$\tau_{N_k} = -\frac{5 f_N(x_i^a(k), x_j^a(k))}{(q_{max} - q_{min}) \| x_i^a(k) - x_j^a(k) \|_2}$$

Given (11) and (12) and recalling f_V^a is chosen constant (*i.e.* $f_V^a(x_i^a, x_j^a)(k+1) - f_V^a(x_i^a, x_j^a)(k) = 0$), we can now define the discretized state-dependent Laplacians with respect to both the visibility graph $G_V(X)$ and the network proximity graph $G_N(X)$. For the former, we obtain

$$[L_V(k+1)]_{uv} = \begin{cases} -f_V(y_u, y_v)(k), & u \neq v \\ \sum_{u \neq s} f_V(y_u, y_s)(k), & u = v \end{cases} \quad (13)$$

where y_u and y_v are defined such that

$$y_l = \begin{cases} x_l^a, & l \leq n = |\mathcal{A}| \\ x_{(l-n)}^t, & n < l \end{cases}$$

Similarly for $G_N(X)$, we are able to define

$$[L_N(k+1)]_{uv} = \begin{cases} -f_N(x_u^a, x_v^a)(k), & u \neq v \\ \sum_{u \neq s} f_N(x_u^a, x_s^a)(k), & u = v \end{cases} \quad (14)$$

Putting this all together, we arrive at a discrete-time formulation for the optimal target tracking problem. At timestep k, we aim to solve the following problem

$$\begin{aligned} & \min \Psi(X(k+1)) \\ & \text{s.t.} \ \| x_i^a(k+1) - x_i^a(k) \|_2 \leq v_i \Delta t, \ i = 1, \ldots, n \\ & \quad P_V^T L_V(k+1) P_V \succ 0 \\ & \quad P_N^T L_N(k+1) P_N \succ 0 \end{aligned} \quad (15)$$

where v_i denotes the translational velocity of agent a_i.

It should be noted that in this formulation, we have augmented the problem with n second-order conic inequalities constraining the distance each agent can travel in a single step. These constraints are essential as they serve to reduce the effects of the linearization process. They can also be used to model velocity constraints on the individual robots.

As our constraint-set has been reduced to linear matrix inequalities and recalling that $\Psi(X(k+1))$ is assumed convex, we see that (15) is in fact a semi-definite program, and it can be efficiently solved using generalized interior-point methods from convex optimization theory [23].

B. Choosing an Appropriate Objective

Until this point, we have avoided any detailed discussion regarding the statement of our convex objective, Ψ. In fact, in the context of target tracking there are many useful candidate functions that fit well within this framework. One possibility is to choose Ψ as the trace of the covariance representing the uncertainty in measured target positions [24]. As such, (15) would yield a position vector that minimizes the uncertainty in the estimated target positions while ensuring full target coverage and enforcing network connectivity.

In this paper, we instead choose $\Psi(X) = -\lambda_2(L_V(X))$. Given this function, we can then maximize the second

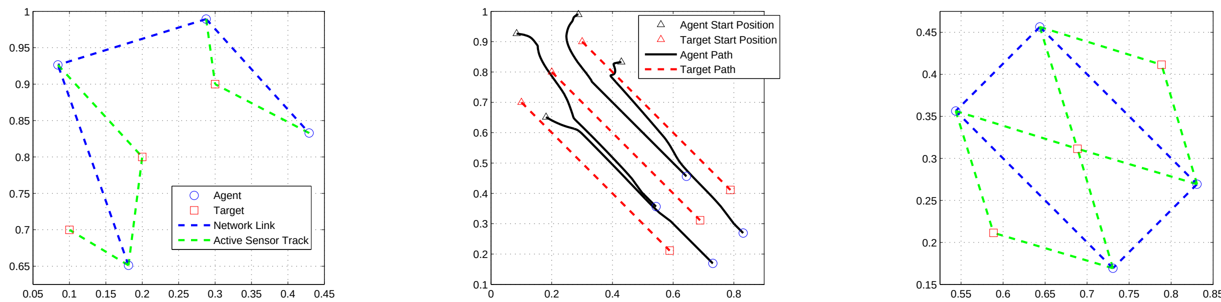

Fig. 3. (Left) The initial visibility and network proximity graphs for a team of four agents in $\mathbb{R}^2$ charged with tracking three mobile targets. In this case, each agent was modeled as using an on-board omnidirectional camera system with the team's objective being to maximize the total number of active tracks – *i.e.* $\Psi(X) = -\lambda_2(L_V(X))$. (Center) The trajectories of the respective team members as they obtain an optimal configuration. (Right) The resulting visibility and network graphs for the team after convergence to optimality ($\Psi(X) \approx 0.5857$).

smallest eigenvalue of the state-dependent graph Laplacian associated with our visibility graph. Choosing Ψ this way implicitly maximizes the number of active target tracks. Such an objective may be quite useful in surveillance applications where each member of the mobile team is outfitted with a low-grade sensor suite. In such a case, maximizing the number of target tracks while observing network connectivity will ensure maximal redundancy in the observation network.

Other appropriate objective functions (*e.g.* weighted least-squares) could be imagined. However, what is important to note is the generality of our approach. So long as the objective function is convex, and can be expressed in terms of linear, quadratic, or semidefinite constraints, the resulting problem will be a semidefinite program. The convexity of SDPs ensures the problem solution will be globally optimal while providing target coverage and network connectivity.

C. Simulation Results

In an effort to validate our discrete-time framework, we implemented our paradigm in Matlab using SeDuMi 1.1R3 [25] via YALMIP [26]. Figure 3 illustrates the results from one such trial in which the objective was to maximize the number of active target tracks in the visibility graph, *i.e.* $\Psi(X) = -\lambda_2(L_V(X))$. In this scenario, four networked agents were responsible for tracking three mobile targets while maintaining a desired standoff distance $r_{ij} = 0.15$. The minimal desired agent-target proximity bounds for active tracks was set at 0.10 with a maximum of 0.20. In this case, each agent was modeled using as its a primary sensor an omnidirectional camera system, and the network was modeled to experience exponential decay between 0.10 and 0.20. The maximal translational velocity of respective team members was 1.1 times that of the mobile targets. In this case, the team ultimately converges to an optimal configuration ($\Psi(X) \approx 0.5857$) whereby the maximal number of tracks are found while maintaining network connectivity.

Figure 4 illustrates the progression of both $\lambda_2(L_V(X))$ and $\lambda_2(L_N(X))$ as the agent team in Figure 3 converges to an optimal configuration. In this plot, we highlight the monotonically increasing behavior of the objective while noting that $\lambda_2(L_N(X))$ remains positive for the entire run. In other words, the team optimizes its objective while ensuring that network connectivity remains intact.

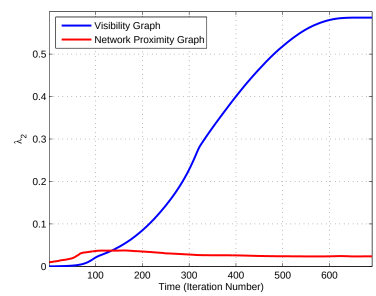

Fig. 4. The progression of $\lambda_2(L_V(X))$ and $\lambda_2(L_N(X))$ corresponding to the run illustrated in Figure 3. In this case, the objective was to maximize the number of active tracks in the agent configuration – *i.e.* $\Psi(X) = -\lambda_2(L_V(X))$. The connectivity of the visibility graph increases monotonically as the team converges to a configuration that ultimately maximizes the objective locally, yielding $\Psi(X) \approx 0.5857$.

VI. Optimal Target Tracking via SOCP

In some cases, it may be beneficial to sacrifice connectivity of the underlying network proximity graph in order to ensure full target coverage. In others, the range of communication links may far exceed the sensing range of the mobile robot team. In such scenarios, the constraint $P_N^T L_N(X) P_N \succ 0$ is redundant and/or superfluous and can be safely eliminated from the problem statement. In so doing, we obtain the original formulation of the tracking problem presented in (6), which we claim can be effectively relaxed as a SOCP.

A. Considering A Relaxed Formulation

The key to obtaining this result, is observing that the constraint $P^T L(X) P \succ 0$ reduces to a single non-linear inequality when the graph in question features only a single pair of nodes. As such, we consider a relaxed formulation of the the tracking problem whereby we associate with each target o_j a single bi-nodal graph $G_{V_j}(\mathcal{V}_j, \mathcal{E}_j)$ with one vertex serving to represent the agent team (*i.e.* $\mathcal{A}$) and the other representing the target itself. *By enforcing the connectivity of each of these graphs in our problem formulation, we ensure*

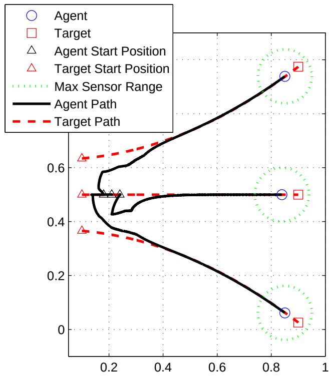

Fig. 5. A team of three agents initially deployed in a path break formation to ensure active tracks of three mobile targets. Connectivity of the state-dependent visibility graph, $G_V(X)$, is ensured by using the proposed SOCP relaxation of the target tracking problem. In this scenario, the objective was to maximize the minimal connectivity of the 3 bi-nodal visibility graphs.

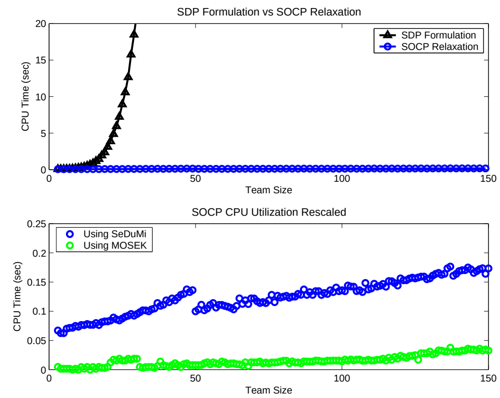

Fig. 6. (Top) CPU utilization time obtained from solving both the SDP formulation and SOCP relaxation via SeDuMi for teams having up to 150 agents tracking 3 targets in $\mathbb{R}^2$. (Bottom) CPU utilization trends for solving the relaxed problem using both a non-industrial (SeDuMi) and industrial solver (MOSEK). In both figures, a single data point corresponds to the mean CPU time obtained from solving ten instances of the problem

at least a single active track to each observation target. In other words, we ensure full target coverage.

Implicit in this statement is that an appropriate weight function can be formulated for $G_{V_j}(X)$ that fully captures the level of connectivity between the agent team and target o_j. Although a variety of functions can be considered, we extend upon our previous analysis and propose the following

$$f_{V_j}(x_1^a, x_2^a, \ldots, x_n^a, x_j^t) = \frac{1}{n}\sum_{i=1}^{n} f_V^t(x_i^a, x_j^t) \qquad (16)$$

Notice that by this definition, when target o_j is being actively tracked by all network agents with each agent observing its desired standoff distance, we have $f_{v_j} = 1$. Similarly, when no agent is actively engaging the target, we have $f_{V_j} = 0$.

In light of these results, we restate (6) in the following relaxed form

$$\begin{array}{l} \min \Psi(X) \\ \text{s.t. } P^T L_{V_j}(X) P > 0, j = 1, \ldots, m \end{array} \qquad (17)$$

where $P = [1, -1]^T$.

Once again applying Euler's first-order discretization method, we obtain the following discrete-time formulation

$$\begin{array}{l} \min \Psi(X(k+1)) \\ \text{s.t. } \| x_i^a(k+1) - x_i^a(k) \|_2 \leq v_i \Delta t, \; i = 1, \ldots, n \\ \qquad P^T L_{V_j}(k+1) P > 0, \; j = 1, \ldots, m \end{array} \qquad (18)$$

This is a standard SOCP constrained by n second-order conic inequalities along with m linear inequalities. It is readily solvable using standard SOCP techniques that are *significantly* more efficient than SDP approaches [23].

B. Simulation Results

Figure 5 illustrates a team of three robots in R^2 breaking an initial path formation in order to successfully track three evading targets. Although contrived, this example serves to highlight the governing behavior of our discrete-time SOCP formulation. By ensuring the connectivity of the $m = 3$ bi-nodal visibility graphs, we see the team is able to ensure full target coverage. In this case, the objective was to maximize the minimal respective connectivity of these graphs with $r_{ij} = 0.06$, and $\epsilon_{ij}^l = \epsilon_{ij}^u = 0.04, \forall i, j$.

In an effort to gauge the comparative difference in complexity between the two approaches, we solved instances of (15) and (18) for team sizes up to 150 nodes operating in $\mathbb{R}^2$. In our SDP implementation, we considered the objective $\Psi(X) = -\lambda_2(L_V(X))$ and maintained $f_V^a = 1$. In the SOCP implementation, we considered maximizing the minimal connectivity among the $m = 3$ bi-nodal visibility graphs. In both cases, SeDuMi was used as the underlying solver. All computations were done on a standard desktop computer having a 2.4 GHz Core 2 Duo Pentium Processor with 2GB RAM. Figure 6 shows the results of these trials where each data point corresponds to the mean utilization time obtained from solving ten random problem instances.

Not surprisingly, the computational overhead associated with solving the SDP formulation scales cubicly in time. In contrast, the computational load incurred by solving our SOCP relaxation exhibits highly linear growth ($r^2 = 0.9205$). Using SeDuMi, which is a non-industrial grade solver, we see that solving a single iteration of (18) for a team of 150 agents requires 174 milliseconds.

In practice, however, it is far more likely that an industrial grade solver will actually be used. As such, we also solved our SOCP relaxation considering the same random problem instances using the MOSEK industrial solver package [27]. The results of these trials are shown in Figure (6) (Bottom). Again, the computational overhead exhibits an approximately linear trend (in this case, $r^2 = 0.7435$). However, what is perhaps more impressive is that *solving a single iteration of (18) for a team of 150 nodes requires only 33 milliseconds!*

VII. Conclusions and Future Work

In this paper, we considered an optimization framework for dynamic target tracking. To realize this framework, we introduced the notion of a weighted visibility graph to capture the state of active target tracks as a function of the team's state positional vector, X. Noting that dynamic target tracking lends itself well to a discrete-time framework, we employed standard linearization techniques to define an iterative SDP approach for solving the target tracking problem subject to network connectivity constraints. In cases where communication constraints can be relaxed, we presented a novel SOCP relaxation to the target tracking problem that ensures connectivity of the state-dependent visibility graph while providing a tremendous reduction in computational cost when compared to a standard SDP formulation.

There are obvious areas where this work can be improved. For instance, in some applications, a simple Gaussian potential may not fully capture the desired behavior for agent-target interactions. To address this issue, we are currently considering alternate non-symmetric weight functions. Another obvious extension to this framework is a bit more challenging. In our problem formulation, we can at best only ensure that each target in the network has associated with it at least a single active track. It would be highly desirable if we could directly or indirectly constrain the minimal number of tracks for each agent. For instance, in a scenario where each team member is using only stereo vision to track the desired targets, we would like to guarantee that each agent has at least two high-level active tracks – one for each camera. Developing a formulation by which to enforce such a constraint is currently the focus of our continued work on this topic.

VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of National Research Organization as well as Yoonsoo Kim for his thoughtful remarks in response to questions on implementing the iterative SDP solver.

REFERENCES

[1] J. Spletzer and C. Taylor, "Dynamic sensor planning and control for optimally tracking targets," *International Journal of Robotics Research*, vol. 22, no. 1, pp. 7–20, 2003.

[2] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control,*, 2006.

[3] A. F. T. Winfield, "Distributed sensing and data collection via broken ad hoc wireless connected network of mobile robots," in *Distributed Autonomous Robotic Systems 4*, G. B. L. E. Parker and J. Barhen, Eds. Springer-Verlag, 2000, pp. 273–282.

[4] R. Arkin and J. Diaz, "Line-of-sight constrained eploration for reactive multiagent robotic teams," in *AMC 7th International Workshop on Advanced Motion Control*, 2002.

[5] J. Sweeney, T. J. Brunette, Y. Yang, and R. A. Grupen, "Coordinated teams of reactive mobile platforms," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Washington, DC, 2002, pp. 99–304.

[6] G. A. S. Pereira, A. K. Das, V. Kumar, and M. F. M. Campos, "Decentralized motion planning for multiple robots subject to sensing and communication constraints," in *Proceedings of the Second Multi-Robot Systems Workshop*, Washington, DC, March 2003, pp. 267–278.

[7] A. R. Wagner and R. C. Arkin, "Communication-sensitive multi-robot reconnaissance," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004, pp. 2480–2487.

[8] J. D. Sweeney, R. A. Grupen, and P. Shenoy, "Active qos flow maintenance in controlled mobile networks," in *Proceedings of the Fourth International Symposium on Robotics and Automation (ISRA)*. Queretaro, Mexico: IEEE, August 2004.

[9] M. Powers and T. Balch, "Value-based communication preservation for mobile robots," in *7th International Symposium on Distributed Autonomous Robotic Systems*, 2004.

[10] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Towards the deployment of a mobile robot network with end-to-end performance guarantees," in *International Conference on Robotics and Automation (ICRA) 2006*, Orlando, FL, April 2006.

[11] S. LaValle, H. Gonzalez-Banos, C. Becker, and J. Latombe, "Motion strategies for maintaining visibility of a moving target," in *Proceeding of the IEEE Int. Conference on Robotics and Automation*, Albuquerque, NM, April 1997, pp. 731–736.

[12] I. Stamos and P. Allen, "Interactive sensor planning," in *Computer Vision and Pattern Recognition Conference*, Santa Barbara, CA, June 1998, pp. 489–495.

[13] P. Fabiani, H. Gonzalez-Banos, J. Latombe, and D. Lin, "Tracking a partially predictable object with uncertainties and visibility constraints," *Journal of Autonomous Robots*, 2001.

[14] L. Parker, "Distributed algorithm for multi-robot observation of multiple moving targets," *Autonomous Robots*, 2002.

[15] Z. Liu, M. H. A. Jr., and W. K. G. Seah, "A potential field based approach for multi-robot tracking of multiple moving targets," in *Proc. 1st International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management*, Manila, Philippines, March 2003.

[16] B. Jung, "Cooperative target tracking using mobile robots," Ph.D. dissertation, University of Southern California, Los Angeles, CA, May 2005.

[17] K. M. Krishna, H. Hexmoor, and S. Sogani, "A t-step ahead constrained optimal target detection algorithm for a multi sensor surveillance system," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robot and Systems*, Edmonton, Canada, Oct.-Nov. 2005, pp. 1840–1845.

[18] F. M. Mirzaei, A. I. Mourikis, and S. I. Roumeliotis, "On the performance of multi-robot target tracking," in *Accepted to IEEE International Conference on Robotics and Automation (ICRA) 2007*, Rome, Italy, April 2007.

[19] B. Shucker and J. K. Bennett, "Target tracking with distributed robotic macrosensors," in *Proceedings of MILCOM 2005*, Atlantic City, NJ, October 2005.

[20] J. Spletzer and C. Taylor, "A bounded uncertainty approach to multi-robot localization," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR 01)*, 2001.

[21] J. Spletzer and C. J. Taylor, "Sensor planning and control in a dynamic environment," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC, 2002.

[22] M. D. Gennaro and A. Jadbabaie, "Decentralized control of connectivity in multi-agent systems," in *Proc. IEEE Conf. on Decision and Control*, San Diego, CA, Dec 2006.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Unviersity Press, 2004.

[24] A. I. Mourikis and S. I. Roumeliotis, "Optimal sensing strategies for mobile robot formations: Resource-constrained localization," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[25] *Addendum to the SeDuMi User Guide Version 1.1*, Advanced Optimization Laboratory, McMaster University, http://sedumi.mcmaster.ca/.

[26] J. Lofberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://control.ee.ethz.ch/ joloef/yalmip.php

[27] *The MOSEK Optimization Tools Version 3.2 (Revision 8) User's Manual and Reference*, MOSEK ApS, http://www.mosek.com.