Second-Order Cone Programming (SOCP) Techniques for Coordinating Large-Scale Robot Teams in Polygonal Environments

Jason C. Derenick and John R. Spletzer

Lehigh University, Bethlehem PA 18015, USA
{jcd6,spletzer}@cse.lehigh.edu

Abstract. In this paper, we present an online optimization approach for coordinating large-scale robot teams in both convex and non-convex polygonal environments. In the former, we investigate the problem of moving a team of m robots from an initial shape to an objective shape while minimizing the total distance the team must travel within the specified workspace. Employing SOCP techniques, we establish a theoretical complexity of $O(k^{1.5}m^{1.5})$ for this problem with $O(km)$ performance in practice – where k denotes the number of linear inequalities used to model the workspace. Regarding the latter, we present a multi-phase hybrid optimization approach. In Phase I, an optimal path is generated over an appropriate tessellation of the workspace. In Phase II, model predictive control techniques are used to identify optimal formation trajectories over said path while guaranteeing against collisions with obstacles and workspace boundaries. Once again employing SOCP, we establish complementary complexity measures of $O(l^{3.5}m^{1.5})$ and $O(l^{1.5}m^{3.5})$ for this problem with $O(l^3m)$ and $O(lm^3)$ performance in practice – where l denotes the length of the optimization horizon.

1 Introduction

The robotics community has seen a tremendous increase in multi-agent systems research in recent years. This has been driven in part by the maturation of the underlying technology: advances in embedded computing, sensor and actuator technology, and perhaps most significantly pervasive wireless communication. However, the primary motivation is the diverse range of applications envisaged for large-scale robot teams, defined herein as formations ranging from tens to thousands of robots. These include support of first responders in search and rescue operations, autonomous surveillance and monitoring in support of military and homeland security operations, and environmental monitoring. Unfortunately, the effective coordination of a large-scale robot team in an arbitrary environment is a non-trivial problem – one that will need to be solved in order for such systems to find widespread use.

In this paper, we investigate an optimization approach to the coordination task. This is motivated by the realization that the effective operation of such a team is inherently a constrained resource allocation problem. A finite number of nodes are required to perform some task (*e.g.* area surveillance), perhaps with performance objectives (*e.g.* maximize coverage), while subjected to resources that are dictated by communication

M.J. Hirsch et al. (Eds.): Adv. in Cooper. Ctrl. & Optimization, LNCIS 369, pp. 95–108, 2007.
springerlink.com

and sensor ranges, motion constraints, environmental constraints, *etc.*. More precisely, we characterize the coordination task as an optimization problem geared towards minimizing the total distance traversed while transitioning the team to a new *shape* configuration subject to polygonal environmental constraints.

While the optimization construct has many advantages, its potential for use in multi-agent systems has never been fully realized due to scalability concerns. Complete solutions to problems of interest typically scale in super-linear time with the number of robots and/or the size of the environment. In this work, we leverage recent advances in convex optimization theory to develop motion planning strategies for effectively coordinating robot teams in both convex and non-convex polygonal work environments. In both cases, the proposed strategy in practice scales linearly with the number of team members. The result is a rich, optimization-based framework for coordinating a large-scale team of fully actuated robots in real-time.

2 Related Work

Control and coordination of mobile robots in polygonal environments has been extensively studied in the literature. Belta *et al*. proposed a computational framework for generating provably correct control laws for fully-actuated robots as well as unicycles in an arbitrary polygonal workspace [1]. In [15], Kloetzer and Belta define a computational framework for the deployment of robots in both 2D and 3D rectangular environments. In this work, obstacles were modeled as polytopes and robot motion was constrained to lie within polyhedral sets. Lindemann and LaValle also considered robot control in polygonal spaces [16]. In particular, they focused upon "car-like" vehicles with bounded path curvature constraints. In their work they partition the polygonal environment into a collection of convex cells before developing safe control laws that obey specified smoothness constraints. Conner *et al*. considered global control laws based upon the utility of local potential functions [6]. They partition the environment into discrete cells and then associate each with control laws which they model as vector fields.

Formations of robot teams have also been extensively studied. As a complete survey is beyond the scope of this paper, we instead focus on those where the notion of *shape* – defined differently under different contexts – was of significant relevance to the research. Das *et al*. described a vision-based formation control framework [8]. This focused on achieving and maintaining a given formation shape using a leader-follower framework. Control of formations using Jacobi shape coordinates was addressed by Zhang *et al* [22]. The approach was applied to a formation of a small number of robots which are modeled as point masses. Abstraction based control was used by Belta and Kumar as a mechanism to coordinate a large number of fully actuated mobile robots moving in formation [2]. The main idea was to map the configuration space of the robots $\mathcal{Q}$ to a lower dimensional manifold $\mathcal{A}$. The concept of *shape* refers to the area spanned by the robots. A local controller was designed based on the state of the robot and the state on the manifold $\mathcal{A}$.

There has also been significant interest in applying optimization based techniques to coordinate robot teams and deploy sensor networks. Contributions in this area include

the work by Cortes *et al* [7]. Here the focus is on autonomous vehicles performing distributed sensing tasks. Recently Feddema *et al.* applied decentralized optimization based control to solve a variety of multi-robot problems [12]. Optimal motion planning was considered by Belta and Kumar [3]. In this work, the authors generate a family of optimal smooth trajectories for a set of fully actuated mobile robots.

3 Defining the Coordination Problem

In developing our strategies, we first consider the problem of transitioning a robot team, constrained to lie within a convex polygonal space, to a new shape formation while minimizing the total distance that the team must travel. As the operating environment is assumed both convex and polygonal, we define it as the affine set:

$$E_c = \{x \in \mathbb{R}^2 \colon A_c x \leq b_c\} \tag{1}$$

where $A_c \in \mathbb{R}^{k \times 2}$ with k denoting the the finite number of linear inequalities used to model the team workspace.

Since the coordination problem is defined as a function of *shape*, it is imperative to first solidify what is precisely meant by this term, as it is often defined differently in different contexts. For our purposes, we adopt the traditional definition of shape that is often employed in statistical shape analysis [11]:

Definition 1. *The shape of a formation is the geometrical information that remains when location, scale, and rotational effects are removed.*

Thus, formation shape is invariant under the Euclidean similarity transformations of translation, rotation, and scale [11].

Given this definition, we can now provide a formal statement of the coordination problem. We begin by letting $Q = [q_1, \ldots, q_m]^T \in \mathbb{R}^{m \times 2}$ denote the concatenated coordinates of the objective shape formation with respect to some world frame $\mathcal{W}$ and by letting $S = [s_1, \ldots, s_m]^T \in \mathbb{R}^{m \times 2}$ denote an instance of our objective shape with respect to some local frame $\mathcal{F}$. Given our convex polygonal workspace E_c, we see that solving the coordination problem reduces to identifying the optimal similarity transformation that when applied to $S \subset \mathcal{F}$ yields an equivalent shape $Q \subset E_C \subset \mathcal{W}$ such that our total distance objective is minimized with respect to Q and the initial robot positions $P = [p_1, p_2, \ldots, p_m]^T$. In other words, we must identify the optimal transformation parameters $[\alpha, \theta, t^x, t^y]^T$ such that $q_i = \alpha R(\theta) s_i + [t^x, t^y]^T \in E_c$ for $i = 1, \ldots, m$ where $\alpha \in R_+$, $R(\theta) \in SO(2)$ and $t^x, t^y \in \mathbb{R}$. Given these observations, the coordination problem can be formulated as the following constrained non-linear optimization problem:

$$\begin{aligned} \min\ & f(q) = \sum_{i=1}^{m} \parallel q_i - p_i \parallel_2 \\ \text{s.t.}\ & q_i = \alpha R(\theta) s_i + [t^x, t^y]^T, i = 1, \ldots, m \\ & q_i \in E_c, i = 1, \ldots, m \\ & \alpha > 0, \theta \in [0, 2\pi) \end{aligned} \tag{2}$$

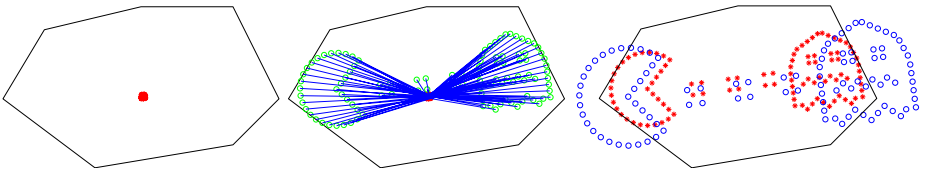

Fig. 1. (Left) The initial formation pose for 101 nodes living in a convex heptagonal (7-sided) environment. (Center) The final formation trajectories to achieve the desired shape configuration while ensuring maximal sensor network coverage in E_c. (Right) The final formation pose (red) after achieving the optimal configuration overlaid with the optimal solution (blue) obtained when the environmental model is ignored. The respecitve optimal parameters were $\alpha = 59.85$ with $[t^x, t^y]^T = [5.585, 5.169]^T$ and $\alpha = 80$ with $[t^x, t^y]^T = [9.893, 4.530]^T$. In this example, θ was fixed at 7.5^o, and scale was constrained to $\alpha \in [10, 80]$.

Unfortunately, this formulation is non-convex due to the $2m$ non-linear constraints used to capture the full set of similarity transformations (as a function of $[\alpha, \theta, t^x, t^y]^T$) that characterize the desired shape geometry S. To remedy this, we employ our results from [9]. In this work, we showed that the optimization variables $[\alpha, \theta, t^x, t^y]^T$ can be implicitly rewritten as a function of the optimal shape configuration Q. More precisely, we can supplant the non-linear equalities in (2) with the following linear (homogenous) constraints (while retaining all original problem information):

$$\left.\begin{array}{r} \| s_2 \|_2 (q_i^x - q_1^x) - (s_i^x, -s_i^y)^T (q_2 - q_1) = 0 \\ \| s_2 \|_2 (q_i^y - q_1^y) - (s_i^y, s_i^x)^T (q_2 - q_1) = 0 \end{array}\right\} i = 3, \ldots, m \tag{3}$$

by defining without loss of generality $[\, \alpha,\, \theta,\, t^x,\, t^y \,]^T \triangleq \left[\frac{\| q_2 - q_1 \|_2}{\| s_2 \|_2},\, \arctan \frac{q_2^y - q_1^y}{q_2^x - q_1^x},\, q_1^x,\, q_1^y \right]^T$.

Given this constraint set, we can now write (2) in convex form; however, doing so would be premature as the objective is non-smooth due to the Euclidean norms inherent in its definition. To handle this, we simply introduce m auxiliary variables $[t_1, t_2, \ldots, t_m]^T$. Doing so allows us to rewrite our non-smooth objective function as a sum of upper bounds on the given Euclidean measures. In other words, the introduction of these variables induces m second-order cone constraints.

Making these adjustments, we can now formally state the coordination problem as the following SOCP in standard form:

$$\begin{array}{ll} \min\limits_{q} & f(t) = 1_m^T t \\ \text{s.t.} & \| q_i - p_i \|_2 \leq t_i,\ i = 1, \ldots, m \\ & \begin{bmatrix} A_w \ I \\ A_s \ 0 \end{bmatrix} \begin{bmatrix} q \\ r \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \\ & r \geq 0 \end{array} \tag{4}$$

where $A_s \in \mathbb{R}^{2(m-1) \times 2m}$ corresponds to the coefficient structure for the linear equalities given in (3) and $A_w \in \mathbb{R}^{km \times 2m}$ denotes the structure for the linear inequalities used to model E_c. We also introduce km non-negative slack variables $r = [r_1, \ldots, r_{km}]^T$.

As SOCPs are convex programs, a local minimum corresponds to a global minimum. This allows optimal solutions to be obtained through a variety of ways such as descent techniques [13] or (more efficiently) by interior point methods (IPMs) [4,13].

Figure 1 illustrates a simple application of our framework for a team of 101 robots charged with maximizing sensor network coverage within a convex heptagonal (7-sided) space. In this case, θ was fixed at 7.5^{o}, and scale was constrained to $\alpha \in [10, 80]$.

3.1 On Complexity

In this section, we solve (4) by adapting the logarithmic penalty-barrier approach outlined in [4]. In so doing, we establish a theoretical complexity of $O(k^{1.5}m^{1.5})$, where k once again denotes the number of linear inequalities used to model E_c.

Like other IPMs, the total complexity of the penalty-barrier approach is largely defined by solving a linear system of equations. In this case, Equality-constrained Newton's method (ENM) is used for internal minimization and the linear system is in KKT form. As solving this system provides a solution to the Newton step sub-problem, we accordingly refer to it as the "Newton KKT system." We show that by reformulating (4), we can band the coefficient matrix to solve the system in $O(km)$ time via algorithms that exploit knowledge of matrix bandwidth.

Reformulating the Coordination Problem. Problem (4) can be restated in a relaxed form suitable for solving via the barrier approach by simply augmenting the objective function with log-barrier terms corresponding to both the problem's conic inequalities as well as the inequalities used to ensure the non-negativity of the associated slack variables. Doing so yields the following equivalent problem statement:

$$\begin{aligned} &\min f(q,t,r) = \tau_l 1_m^T t - \sum_{i=1}^{m} \log\left(t_i^2 - (q_i - p_i)^T(q_i - p_i)\right) - \sum_{i=1}^{km} \log r_i \\ &\text{s. t.} \begin{bmatrix} A_w & I \\ A_s & 0 \end{bmatrix} \begin{bmatrix} q \\ r \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \end{aligned} \tag{5}$$

where τ_l is the inverse log-barrier scaler for the l^{th} iteration. Essentially, solving our SOCPs reduces to solving a sequence of convex optimization problems of this form, where after each iteration τ_{l+1} is chosen such that $\tau_{l+1} > \tau_l$ [4].

Banding the Newton KKT System. During each iteration of the log-barrier approach, we aim to minimize the second-order Taylor approximation of our objective function as a function of the Newton step, $\delta x = [\delta q, \delta r]^T$, subject to $A\delta x = 0$. As a result, obtaining δx is equivalent to analytically solving the KKT conditions associated with this equality-constrained sub-problem. In other words, we must solve the following linear system of equations [4]:

$$\begin{bmatrix} H & \hat{A}^T \\ \hat{A} & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ v \end{bmatrix} = \begin{bmatrix} -g \\ 0 \end{bmatrix} \tag{6}$$

where H and g respectively denote the evaluated Hessian and gradient of the objective function given in (5) at x, v is the corresponding dual variable for δx, and $\hat{A} = \left[\begin{smallmatrix} A_w & I \\ A_s & 0 \end{smallmatrix}\right]$. Undoubtedly, solving (6) is the bottleneck of the algorithm, requiring $O(k^3m^3)$ basic operations in a naive implementation; however, we will show that it can be solved very efficiently by simply reposing the problem given in (5).

Noting that the coefficient matrix of (6) is symmetric indefinite, we employ Gaussian elimination with non-symmetric partial pivoting. The performance of this technique suffers significantly when the linear system in question features dense rows and/or columns due to fill-in [21]. In particular, the algorithm could yield a worst-case performance of $O(k^3m^3)$ when solving an instance of (6) associated with the nominal problem formulation given in (5). To illustrate this point, we include Figure 2 (Left) which shows the corresponding non-zero sparsity structure of the Newton KKT system. As the rows of system are permuted during reduction, the dense rows and columns respectively located in the upper-right and lower-left quadrants of (6) could introduce a solid sub-block of order $km \times km$, which itself would require $O(k^3m^3)$ basic operations to reduce. Such a workload is highly impractical, especially when considering large-scale configurations that inherently feature 1000's of decision variables.

To address this issue, we present the following auxiliary formulation of (5) that facilitates transforming the Newton KKT system into a mono-banded form:

$$
\begin{aligned}
&\min\ f(q,t,r) = \tau_l 1_m^T t - \sum_{i=1}^{m} \log\left(t_i^2 - (q_i - p_i)^T(q_i - p_i)\right) - \sum_{i=1}^{km} \log r_i \\
&\text{s. t.} \\
&\quad \| s_2 \|_2 \left(q_i^x - c_{ik+1}^x\right) - \left(s_i^x, -s_i^y\right)^T \left(d_{(i-1)k+1} - c_{ik+1}\right) = 0,\ i = 3, \ldots, m \\
&\quad \| s_2 \|_2 \left(q_i^y - c_{ik+1}^y\right) - \left(s_i^y, s_i^x\right)^T \left(d_{(i-1)k+1} - c_{ik+1}\right) = 0,\ i = 3, \ldots, m \\
&\quad a_j^T w_{(i-1)k+j} + r_{(i-1)k+j} = b_{(i-1)k+j},\ i = 1, \ldots, m, j = 1, \ldots, k \\
&\quad w_{(i-1)k+j} = w_{(i-1)k+j+1},\ i = 1, \ldots, m, j = 1, \ldots, k-1 \\
&\quad w_{(i-1)k+1} = q_i,\ i = 1, \ldots, m \\
&\quad c_i = c_{i+1},\ i = 1, \ldots, km-1 \\
&\quad d_i = d_{i+1},\ i = 1, \ldots, km-k-1 \\
&\quad c_1 = q_1 \\
&\quad d_1 = q_2
\end{aligned}
\tag{7}
$$

In this formulation, the shape constraints are given by the first two sets of equalities while the environmental bounds are given by the third. Essentially, the additional c and d variables allow us to "chain" the values of q_1 and q_2 respectively through the corresponding Newton KKT system, which eliminates the dense row and column features that would otherwise be present. Similarly, as the k linear inequalities defining E_c bound the final objective position of each node (*i.e.* q_i), we introduce k auxiliary variables (*i.e.* w) for each node in order to locally chain q_i. Doing so ensures a bandwidth that will ultimately remain independent of both k and m.

Given this augmented formulation, our claim is that the system can be made mono-banded. To show this, we begin by defining the nominal solution vector for the coefficient structure of (6) as follows:

$$
\left[\delta\eta^T, \delta\kappa_1^T, \ldots, \delta\kappa_{m-2}^T, \delta\zeta^T, \mu^T\right]^T \tag{8}
$$

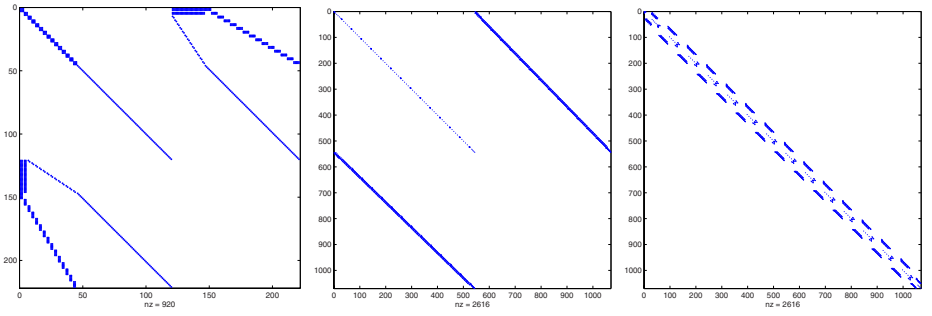

Fig. 2. (Left) The nominal Newton KKT system structure for a team of 15 robots constrained to a pentagonal workspace. (Center) The augmented Newton KKT system for the same configuration. This system is derived from (8) and (9). (Right) The banded system with lower and upper bandwidths of 37. The bandwidth is independent of both team size and the number of constraints used to model E_c. In this form, the system is now solvable in $O(km)$.

$$\delta\eta = \begin{bmatrix} \delta q_1 \\ \delta t_1 \\ \delta w_1 \\ \delta r_1 \\ \delta c_1 \\ \vdots \\ \delta w_k \\ \delta r_k \\ \delta c_k \end{bmatrix} \quad \delta\kappa_i = \begin{bmatrix} \delta q_{i+1} \\ \delta t_{i+1} \\ \delta c_{ik+1} \\ \delta d_{(i-1)k+1} \\ \delta w_{ik+1} \\ \delta r_{ik+1} \\ \vdots \\ \delta c_{(i+1)k} \\ \delta d_{ik} \\ \delta w_{(i+1)k} \\ \delta r_{(i+1)k} \end{bmatrix} \quad \delta\zeta = \begin{bmatrix} \delta q_m \\ \delta t_m \\ \delta c_{(m-1)k+1} \\ \delta d_{(m-2)k+1} \\ \delta w_{(m-1)k+1} \\ \delta r_{(m-1)k+1} \\ \vdots \\ \delta w_{mk} \\ \delta r_{mk} \end{bmatrix} \quad \mu = \begin{bmatrix} v_1 \\ \vdots \\ v_{(7m-6)k+2m} \end{bmatrix}$$

where the δ variables correspond to the primal Newton step components associated with each of the respective system variables.

In order to yield the mono-banded form, we begin by stating the constraint/row permutation for A that yields the tri-banded system appearing Figure 2 (Center). We assume that A is already arbitrarily constructed with random row and column permutations. For the sake of clarity, we group constraints by associating them with the respective nodes that introduce them into the system. In doing so, we employ a slight abuse of notation by allowing the variable q_i to also denote the i^{th} robot in the configuration. That stated, we can now define the constraints associated with q_1:

$$\left.\begin{aligned} q_1^x &= c_1^x \\ q_1^y &= c_1^y \\ q_1^x &= w_1^x \\ q_1^y &= w_1^y \\ a_1^T w_1 + r_1 &= b_1 \end{aligned}\right\} \triangleq \varrho_1 \qquad \left.\begin{aligned} c_{j-1}^x &= c_j^x \\ c_{j-1}^y &= c_j^y \\ w_{j-1}^y &= w_j^y \\ w_{j-1}^y &= w_j^y \\ a_j^T w_j + r_j &= b_j \end{aligned}\right\} \triangleq \varrho_j,\ j = 2, \ldots, k$$

Similarly for q_2, we associate

$$\left.\begin{aligned} c_k^x &= c_{k+1}^x \\ c_k^y &= c_{k+1}^y \\ q_2^x &= d_1^x \\ q_2^y &= d_1^y \\ q_2^x &= w_{k+1}^x \\ q_2^y &= w_{k+1}^y \\ a_{k+1}^T w_{k+1} + r_{k+1} &= b_{k+1} \end{aligned}\right\} \triangleq \varrho_{k+1} \qquad \left.\begin{aligned} c_{k+j-1}^x &= c_{k+j}^x \\ c_{k+j-1}^y &= c_{k+j}^y \\ d_{j-1}^x &= d_j^x \\ d_{j-1}^y &= d_j^y \\ w_{k+j-1}^y &= w_{k+j}^y \\ w_{k+j-1}^y &= w_{k+j}^y \\ a_{k+j}^T w_{k+j} + r_{k+j} &= b_{k+j} \end{aligned}\right\} \triangleq \varrho_{k+j},\ j = 2, \ldots, k$$

For $3 \leq i \leq (m-1)$, we define the constraints associated with q_i as:

$$\left.\begin{aligned} c_{(i-1)k}^x &= c_{(i-1)k+1}^x \\ c_{(i-1)k}^y &= c_{(i-1)k+1}^y \\ d_{(i-2)k}^x &= d_{(i-2)k+1}^x \\ d_{(i-2)k}^y &= d_{(i-2)k+1}^y \\ \| s_2 \|_2 \, (q_i^x - c_{(i-1)k+1}^x) &= (s_i^x, -s_i^y)^T (d_{(i-2)k+1} - c_{(i-1)k+1}) \\ \| s_2 \|_2 \, (q_i^y - c_{(i-1)k+1}^y) &= (s_i^y, s_i^x)^T (d_{(i-2)k+1} - c_{(i-1)k+1}) \\ q_i^x &= w_{(i-1)k+1}^x \\ q_i^y &= w_{(i-1)k+1}^y \\ a_{(i-1)k+1}^T w_{(i-1)k+1} + r_{(i-1)k+1} &= b_{(i-1)k+1} \end{aligned}\right\} \triangleq \varrho_{(i-1)k+1}$$

$$\left.\begin{aligned} c_{(i-1)k+j-1}^x &= c_{(i-1)k+j}^x \\ c_{(i-1)k+j-1}^y &= c_{(i-1)k+j}^y \\ d_{(i-2)k+j-1}^x &= d_{(i-2)k+j}^x \\ d_{(i-2)k+j-1}^y &= d_{(i-2)k+j}^y \\ w_{(i-1)k+j-1}^y &= w_{(i-1)k+j}^y \\ w_{(i-1)k+j-1}^y &= w_{(i-1)k+j}^y \\ a_{(i-1)k+j}^T w_{(i-1)k+j} + r_{(i-1)k+j} &= b_{(i-1)k+j} \end{aligned}\right\} \triangleq \varrho_{(i-1)k+j}, j = 2, \ldots, k$$

Finally, we associate the remaining constraints with q_m:

$$\left.\begin{aligned} c_{(m-1)k}^x &= c_{(m-1)k+1}^x \\ c_{(m-1)k}^y &= c_{(m-1)k+1}^y \\ d_{(m-2)k}^x &= d_{(m-2)k+1}^x \\ d_{(m-2)k}^y &= d_{(m-2)k+1}^y \\ \| s_2 \|_2 \, (q_m^x - c_{(m-1)k+1}^x) &= (s_m^x, -s_m^y)^T (d_{(m-2)k+1} - c_{(m-1)k+1}) \\ \| s_2 \|_2 \, (q_m^y - c_{(m-1)k+1}^y) &= (s_m^y, s_m^x)^T (d_{(m-2)k+1} - c_{(m-1)k+1}) \\ q_m^x &= w_{(m-1)k+1}^x \\ q_m^y &= w_{(m-1)k+1}^y \\ a_{(m-1)k+1}^T w_{(m-1)k+1} + r_{(m-1)k+1} &= b_{(m-1)k+1} \end{aligned}\right\} \triangleq \varrho_{(m-1)k+1}$$

$$\left.\begin{aligned} w_{(m-1)k+j-1}^y &= w_{(m-1)k+j}^y \\ w_{(m-1)k+j-1}^y &= w_{(m-1)k+j}^y \\ a_{(m-1)k+j}^T w_{(m-1)k+j} + r_{(m-1)k+j} &= b_{(m-1)k+j} \end{aligned}\right\} \triangleq \varrho_{(m-1)k+j},\ j = 2, \ldots, k$$

Again we employ a slight abuse of notation by letting each ϱ_j also denote the initial row indices of the constraints with which it is associated. Preserving the relative

ordering of the constraints as they appear in the respective definition of each ϱ_j, we provide the following row permutation for A. This ordering yields the tri-banded form as it appears in 2 (Center):

$$\left[\varrho_1^T, \varrho_2^T, \ldots, \varrho_{mk}^T\right]^T \tag{9}$$

Given this definition of A as well as (8), the mono-banded form of (6) can be constructed. Symmetrically applying the permutation that yields the following Newton KKT system solution vector ordering:

$$\left[\lambda^T, \gamma^T, \xi_1^T, \ldots, \xi_{(m-3)}^T, \chi^T\right]^T \tag{10}$$

$$\lambda = \begin{bmatrix} \delta q_1 \\ \delta t_1 \\ v_1 \\ \vdots \\ v_{5k} \\ \delta w_1 \\ \delta r_1 \\ \delta c_1 \\ \vdots \\ \delta w_k \\ \delta r_k \\ \delta c_k \end{bmatrix} \quad \gamma = \begin{bmatrix} \delta q_2 \\ \delta t_2 \\ v_{5k+1} \\ \vdots \\ v_{12k} \\ \delta c_{k+1} \\ \delta d_1 \\ \delta w_{k+1} \\ \delta r_{k+1} \\ \vdots \\ \delta c_{2k} \\ \delta d_k \\ \delta w_{2k} \\ \delta r_{2k} \end{bmatrix} \quad \xi_i = \begin{bmatrix} \delta q_{i+2} \\ \delta t_{i+2} \\ v_{(7i+5)k+2i-1} \\ \vdots \\ v_{(7i+12)k+2i} \\ \delta c_{(i+1)k+1} \\ \delta d_{ik+1} \\ \delta w_{(i+1)k+1} \\ \delta r_{(i+1)k+1} \\ \vdots \\ \delta c_{(i+2)k} \\ \delta d_{(i+1)k} \\ \delta w_{(i+2)k} \\ \delta r_{(i+2)k} \end{bmatrix} \quad \chi = \begin{bmatrix} \delta q_m \\ \delta t_m \\ v_{(7m-9)k+2m-5)} \\ \vdots \\ v_{(7m-6)k+2m} \\ \delta c_{(m-1)k+1} \\ \delta d_{(m-2)k+1} \\ \delta w_{(m-1)k+1} \\ \delta r_{(m-1)k+1} \\ \vdots \\ \delta w_{mk} \\ \delta r_{mk} \end{bmatrix}$$

produces a mono-banded coefficient structure having a respective upper and lower bandwidths of 37.

Figure 2 illustrates the process of transforming the KKT system via our approach. The "augmented" Newton KKT system derived from the permutations given in (8) and (9) is shown in Figure 2 (Center). Taking the coefficient structure of (6) in this form and symmetrically permuting its rows and columns according to (10) yields the mono-banded system appearing in Figure 2 (Right). It can now be solved in $O(km)$ using a band-diagonal LU-based solver [20].

Applying these alterations effectively reduces the per-iteration complexity of the penalty-barrier method to $O(km)$ for the coordination problem. As the iteration complexity of the barrier approach scales as $O(\sqrt{km})$, we see that the total complexity is $O(k^{1.5}m^{1.5})$ in theory. However, it should be emphasized that this bound is highly conservative as it is well-known that iteration complexity scales as $O(1)$ in practice [4]. As such, solving the coordination problem will require a number of basic operations that grows more like $O(km)$. In other words, the computational workload scales linearly with the number environmental constraints and the configuration size.

3.2 Simulation Results

The results presented thus far correspond to an application of a simple penalty-barrier approach. Although effective, such an IPM is not typically used in practice as more

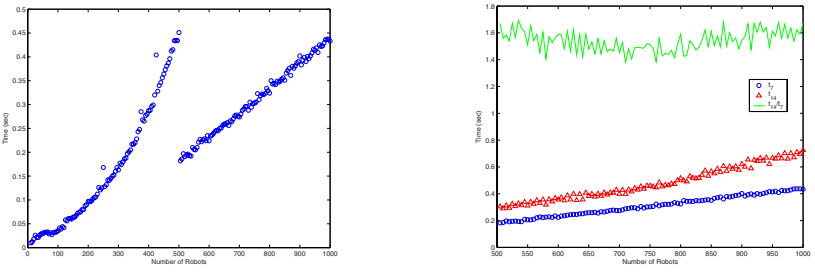

Fig. 3. (Left) MOSEK CPU utilization time for teams operating in a heptagonal environment. When problem structure is fully exploited ($m \gtrapprox 500$), the trend becomes highly linear with $r^2 = 0.9822$. (Right) The highly linear trends for teams operating in heptagonal and tetradecagonal (14-sided) environments. Regarding the latter, we have $r^2 = 0.9645$.

sophisticated and robust solvers exist [17,19]. As such, we carried-out a sequence of trials whereby the coordination problem was solved using the MOSEK industrial solver package, which utilizes a homogenous self-dual IPM [19]. For our trials, we varied the given team size m from 10 to 1000 at intervals of 5 with the mean CPU time being recorded over a sample size of 10 trials for each value. All problems were solved using a standard desktop PC having a 3.0 GHz Pentium 4 processor and 2.0 GB of RAM.

In Figure 3 (Left) the CPU utilization trend is provided for a team confined to operations in a heptagonal (7-sided) environment. Notice that below ≈ 500 nodes, the complexity scales cubicly ($r^2 = 0.9933$). This appears to be the result of the solver not fully exploiting problem structure in obtaining its solution. Beyond 500 this is not the case as performance is highly linear with linear regression analysis revealing $r^2 = 0.9822$. Perhaps more importantly, we see that solutions for configurations having up to 1000 nodes are obtainable in less than 0.45 seconds.

Figure 3 (Right) shows the highly-linear performance trends for robot teams operating respectively in heptagonal and tetradecagonal (14-sided) environments. Moreover, the linear growth of the complexity as a function of k is evident by considering the comparative performance ratio $\frac{t_{14}}{t_7}$ which remains essentially constant as $m \rightarrow 1000$. Together, these results highlight the efficacy of our approach.

4 Coordination in Non-convex Polygonal Environments

We now compose our previous results into a more general instance of the coordination task. Specifically, we consider motion planning in an arbitrary polygonal environment with obstacles. As the space of feasible robot positions is no-longer convex by assumption, solving this problem directly would require more general and less-efficient nonlinear programming techniques that guarantee only convergence to local minima. Thus, in an effort to obtain a similar complexity results as those seen in Section 3.1, we propose a hybrid multi-phase optimization approach over a discrete convex tessellation of the work environment.

4.1 Generalizing the Coordination Problem

As noted, we assume that the configuration space $\mathcal{C}$ for the robot team is a polygonal environment with obstacle subspace $\mathcal{O}$ and free space $\mathcal{C}_{free}$ such that $\mathcal{C}_{free} = \mathcal{C} - \mathcal{O}$. Using exact cell decomposition methods (*e.g.* triangulation, trapezoidal decomposition, *etc.*), $\mathcal{C}_{free}$ can be tessellated into convex polygonal cells $C_1, \ldots, C_z$, where $\mathcal{C}_{free} = \bigcup_{i=1}^{z} C_i$ [5]. The resulting partition induces an undirected graph $G = (V, E)$, where vertex $v_i \in V$ corresponds to cell C_i, and edge $e_{ij} \in E$ implies that there exists a common edge between C_i and C_j. Paths between cells can then be efficiently computed using traditional graph optimization algorithms (*e.g.* [10]). The coordination problem can then be reposed as transitioning the formation from cell to cell along the specified path. In the sequel, we assume a triangulation partition of $\mathcal{C}_{free}$. We also assume that the union of adjacent cells $C_{ij} = C_i \bigcup C_j \ \forall \ (C_i, C_j) \in E$ is convex. This is hardly restrictive as it is straightforward to refine any pair of adjacent triangles to three such triangles where both of the resulting adjacent pairs meet this constraint.

Remark 1. Given two adjacent cells $(C_i, C_j) \in E$, where $C_{ij} = C_i \bigcup C_j$ is convex, if node $x_i \in C_i$ and $x_j \in C_j$, then by convexity $\lambda x_i + (1 - \lambda) x_j \in C_{ij}, \ \lambda \in [0, 1]$. This implies that for a formation of m nodes with initial pose $X_i = (x_{i1}, \ldots, x_{im})^T \in \mathbb{R}^{2m}$ in triangle C_i, and final pose X_j in triangle C_j, the paths of each node will remain entirely in $C_{ij} \subseteq \mathcal{C}_{free}$. This guarantees against collisions with obstacles.

Let us assume that such a path $C_p = \{C_1, \ldots, C_l\} \subseteq \mathcal{C}_{free}$ has been specified by a higher level planner. The coordination problem can then be written as follows:

Problem 1. Given a path specification $C_p = \{C_1, \ldots, C_l\}$, a corresponding shape specification $S = \{S_1, \ldots, S_l\}$, and an initial formation pose X_0, find a motion sequence $X = \{X_1, \ldots, X_l\}$ for the formation such that

1. $X_i \sim S_i, \ i = 1, \ldots, l$
2. $X_i \in C_i, \ i = 1, \ldots, l$
3. The distance traveled by the formation is minimized in accordance with the criteria from Problem 4.

In solving Problem 1, we employ optimization techniques from model predictive control [14,18]. In this context however, the length of the horizon is not defined by time, but rather the length of the path over which the optimization problem is solved.

To constrain the pose of the formation during each step of the horizon, each triangle can be modeled as a set of three linear inequality constraints on the position of each robot

$$c_{ik}^T x_{ij} \leq 0, \ i = 1, \ldots, l, \ j = 1, \ldots, m, \ k = 1, 2, 3 \tag{11}$$

In a slight abuse of notation, we also let $C_i = (c_{11}, \ldots, c_{m3})^T \in \mathbb{R}^{3m \times 2m}$ denote the set of linear constraints on the formation pose such that $X_i \in C_i$ We can now write the solution to Problem 1 for our total distance metric as

$$\begin{aligned} \min_X \ & \sum_{i=1}^{l} \sum_{j=1}^{m} t_{ij}, \ i = 1, \ldots, l, \ j = 1, \ldots, m \\ \text{s.t.} \ & \| x_{ij} - x_{i-1,j} \|_2 \leq t_{ij} \\ & A_i X_i = 0 \\ & C_i X_i \leq 0 \end{aligned} \tag{12}$$

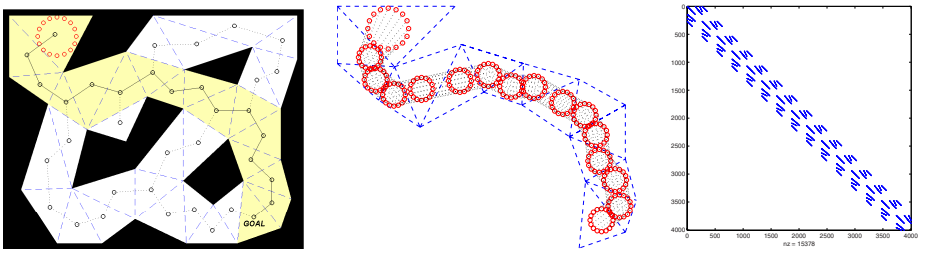

Fig. 4. (Left) $\mathcal{C}_{path}$ as specified by the higher level planner. (Center) The corresponding motion sequence obtained from solving the associated SOCP. The formation is guaranteed to follow $\mathcal{C}_{path}$ while minimizing the total distance traveled, avoiding obstacles, and maintaining the desired formation shape. In this example, both the orientation and minimum scale for the formation were constrained. (Right) The associated linear system remains mono-banded, and in this case, the bandwidth is defined as a function of the configuration size m.

where A_i are the constraints associated with shape S_i as defined in Section 3. By now, we can readily recognize the form of this problem as a SOCP. More significantly perhaps, the corresponding KKT matrix corresponds to the chaining of l instances of our single step problem. As a result, the associated linear system will remain mono-banded; however, in this case the bandwidth will grow as either a function of m or l depending upon the selected permutation of the augmented KKT system. As such, we conclude the theoretical complexity is $O(l^{1.5}m^{3.5})$ or $O(l^{3.5}m^{1.5})$ – once again depending upon the chosen ordering. In cases where the problem demands $l \gg m$ – *i.e.* the horizon length far exceeds the team size – a permutation yielding a bandwidth as a function of m is best. Similarly, when the problem requires $m \gg l$, the bandwidth is best defined as a function of horizon length as that yields the best performance bound.

Once again, these theoretical results are highly conservative as iteration complexity scales as $O(1)$ in practice [4]. Thus, solving the generalized coordination problem will require a number of basic operations that scales more like $O(l^3m)$ (or $O(lm^3)$). In the former case, complexity scales linearly with configuration size making it well-suited for coordinating a large-scale robot team.

4.2 Simulation Results

A sample simulation trial for a formation of 16 robots is shown in Figure 4. The path of the formation is specified by a higher level planner after a discrete optimization phase on the corresponding graph G (Left). The formation then solves the continuous optimization problem specified in (12). The resulting path of the formation is shown in Figure 4 (Right). In this example, the optimization was over the entire path length $(l = 16)$, the shape was held constant, and the minimum scale of the formation was specified as a premise for inter-robot collision avoidance.

5 Discussion and Future Work

In this paper, we developed strategies for coordinating large-scale robot teams in both convex and non-convex polygonal environments. We began by formulating the coor-

dination problem as a constrained optimization problem in which the objective was to minimize the distance a team, living in a convex polygonal workspace, must travel while transitioning to a new objective shape configuration. We showed that this problem can be formulated as a convex mathematical program. Solving with a log-barrier IPM, we also showed that its solvable in $O(k^{1.5}m^{1.5})$ time in theory with $O(km)$ performance in practice – where k denotes the number of affine constraints used to model the convex workspace and m denotes the configuration size.

After establishing these results, we then extended them to solve the coordination problem in a non-convex polygonal workspace. By using an appropriate tessellation of the environment along with model predictive control techniques, we showed that a large-scale team of robots can obtain an objective position while successfully avoiding collisions with both workspace boundaries and static obstacles. This problem is also presented in convex form, and we showed that complexity scales as $O(l^{3.5}m^{1.5})$ with $O(l^3m)$ performance in practice – when the bandwidth of the IPM's core linear system is defined as a function of the optimization horizon length l. In the case where bandwidth is defined in terms of configuration size m, the theoretical complexity is then $O(l^{1.5}m^{3.5})$ with $O(lm^3)$ performance in practice.

We are currently extending these results to a more general multi-objective framework for large-scale coordination in $SE(2)$. Such an extension is invaluable as many applications require teams of robots to perform well with respect to multiple goals. Additionally, we are exploring the possibility of extending the framework to $SE(3)$. However, such an extension is not obvious as a direct formulation of the coordination problem in this higher dimensional space introduces imaginary terms. As a result, alternate approaches and possible relaxations are being evaluated to achieve this end.

References

1. C. Belta, V. Isler, and G. Pappas. Discrete abstractions for robot motion planning and control in polygonal environments. *IEEE Transactions on Robotics*, 17(6):864–875, 2005.
2. C. Belta and V. Kumar. Abstraction and control for groups of robots. IEEE Trans. on Robotics and Automation, 2004.
3. C. Belta and V. Kumar. Optimal motion generation for groups of robots: a geometric approach. *ASME Journal of Mechanical Design*, 126, 2004.
4. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Unviersity Press, 2004.
5. H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. *Principles of Robot Motion Planning*. MIT Press, 2005.
6. D. C. Conner, A. A. Rizzi, and H. Choset. Composition of local potential functions for global robot control and navigation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, Nevada, USA, October 2003.
7. J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Trans. on Robotics and Automation*, 20(2):243–255, April 2004.
8. A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor. A vision-based formation control framework. *IEEE Trans. on Robotics and Automation*, 18(5):813–825, October 2002.
9. J. Derenick, C. Mansley, and J.Spletzer. Efficient motion planning strategies for large-scale sensor networks. In *Proceedings of the Seventh International Workshop on the Algorithmic Foundations of Robotics (WAFR 2006)*, New York, NY, USA, July 2006.

10. E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–272, 1959.
11. I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, 1998.
12. J. T. Feddema, R. D. Robinett, and R. H. Byrne. An optimization approach to distributed controls of multiple robot vehicles. In *Workshop on Control and Cooperation of Intelligent Miniature Robots, IEEE/RSJ IROS*, Las Vegas, Nevada, October 31 2003.
13. R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization*. Springer, 2nd edition, 2000.
14. A. Jadbabaie, J. Yu, and J. Hauser. Unconstrained receding-horizon control of nonlinear systems. *IEEE Trans. on Automatic Control*, 46(5):776–783, May 2001.
15. M. Kloetzer and C. Belta. A framework for automatic deployment of robots in 2d and 3d environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 953–958, Beijing, China, October 2006.
16. S. Lindemann and S. LaValle. Smooth feedback for car-like vehicles in polygonal environments. In *Proc. IEEE Int. Conf. Robot. Automat.*, Roma, Italy, April 2007.
17. M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and Applications, Special Issue on Linear Algebra in Control, Signals and Image Processing*, 1998.
18. D. Mayne, J. Rawings, C. Rao, and P. Scokaert. Constrained model predictive control: Stability and optimality. *Automatics*, 36(6):789–814, June 2000.
19. MOSEK ApS. *The MOSEK Optimization Tools Version 3.2 (Revision 8) User's Manual and Reference*. http://www.mosek.com.
20. W. Press et al. *Numerical Recipes in C*. Cambridge University Press, 1993.
21. Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.
22. F. Zhang, M. Goldgeier, and P. S. Krishnaprasad. Control of small formations using shape coordinates. In *Proc. IEEE Int. Conf. Robot. Automat.*, volume 2, Taipei, Sep 2003.