Proceedings of the 41st IEEE
Conference on Decision and Control
Las Vegas, Nevada USA, December 2002

Ad Hoc Networks for Localization and Control

Aveek Das, John Spletzer, Vijay Kumar and Camillo Taylor
{aveek,spletzer,kumar,cjtaylor}@grasp.cis.upenn.edu

Abstract

We consider a team of mobile robots equipped with sensors and wireless network cards and the task of navigating to a desired location in a formation. We develop a set of algorithms for (a) discovery; (b) cooperative localization; and (c) cooperative control. Discovery involves the use of sensory information to organize the robots into a mobile network allowing each robot to establish its neighbors and, when necessary, one or more leaders. Cooperative control is the task of achieving a desired goal position and orientation and desired formation *shape* and maintaining it. Cooperative localization allows each robot to estimate its relative position and orientation with respect to its neighbors and hence the formation shape. We show numerical results and simulations for a team of nonholonomic, wheeled robots with omnidirectional cameras sharing a wireless communication network.

1 Introduction

In real-world situations multi-agent robotic systems are subject to sensor, actuator and communication constraints, and have to operate within uncertain and unstructured environments. We are interested in tasks that include surveillance [4], search and rescue operations [6], exploration and mapping of unknown or partially known environments [12], and distributed manipulation [8] and transportation of large objects [10]. In all these applications, there is a need to have the robots estimate their relative positions and orientations with respect to their neighbors and maintain a desired formation.

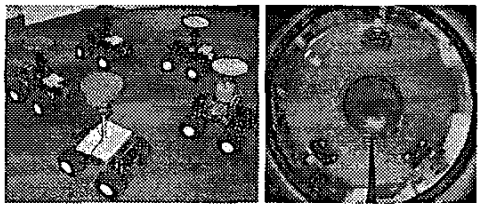

Figure 1: Our Clodbuster wheeled mobile robot platform (left) with omnidirectional cameras (right) and wireless networking.

We are interested in the leader-follower assignment paradigm where each robot follows one or two leaders [3]. The choice of leader-follower controllers leads to the description of a *control graph*, that describes the assignment of these controllers and the interconnections in the system. The stability and performance of the system depends on this graph [5]. We use distributed cooperative sensing as our tool for localization of robots relative to each other. We are motivated by our experimental platform of wheeled mobile robots with omnidirectional cameras and IEEE 802.11b wireless networking (see Figure 1) [2, 5]. In this paper, we present a framework for building dynamic, *ad hoc* computational networks of mobile robots for multi-robot coordination tasks based on distributed cooperative sensing and cooperative control. Specifically, we will address the problem of navigating a group of n nonholonomic wheeled robots to a desired goal position and orientation while achieve a desired *shape*, where shape refers to the distribution of the robots relative to their neighbors.

2 Modeling

Kinematic Model: We consider a simple kinematic model for our wheeled robot platforms shown in Figure 1. The i^{th} robot has the model:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{f}(\boldsymbol{x}_i)\boldsymbol{u}_i \tag{1}$$

where $\boldsymbol{x}_i = (x_i,\ y_i,\ \theta_i) \in SE(2)$, inputs $\boldsymbol{u}_i = (v_i,\ \omega_i) \in \mathbb{R}^2$, and v_i and ω_i are the linear and angular control velocities, respectively.

We advocate a leader-follower paradigm [3] in which each robot, except for the leader, can follow one or two other robots. To see this, consider the example in Figure 2. Robot R_i leads the group. R_j maintains a specified separation (l_{ij}) and bearing (ψ_{ij}) with respect to R_i, while R_k regulates its separation with respect to R_i (l_{ik}) and R_j (l_{jk}). A directed edge from R_i to R_j denotes a Separation-Bearing Controller ($SB_{ij}C$) in the figure, and edges from R_i to R_k and R_j to R_k are used to represent a Separation-Separation Controller ($SS_{ijk}C$). Details of the controllers are provided in [3, 5]. Very briefly, the Separation-Bearing controller guarantees exponential convergence of the two relevant shape variables — the separation and relative bearing with respect to a leader — to desired values. Similarly, the Separation-Separation controller ensures exponential convergence of the separations with respect to two leaders to desired values. In both cases, the orientation of the robot is not directly controlled. The shape $\rho = (l_{ij}, \psi_{ij}, l_{ik}, l_{jk})$ determines the relative positions of the robots. The position and orientation of robot R_i can be used to describe the gross position and orientation of the group g. This basic idea will be generalized to a n robot team next.

0-7803-7516-5/02/$17.00 ©2002 IEEE

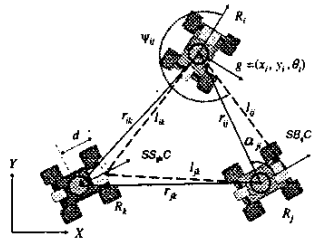

Figure 2: A 3 robot triangular formation. R_j follows R_i using $SB_{ij}C$, while R_k follows both R_i and R_j using $SS_{ijk}C$.

Network Model: We can associate with a team of n robots three different networks: a *physical network* that captures the physical constraints on the dynamics, control and sensing of each robot; a *communication network* that describes information sharing between robots; and a *computational network* that describes the computations performed at each robot and the flow of information across the group.

We model each network by a graph with n nodes, one node for each agent. R is a finite set of nodes, $R_1, R_2, \ldots, R_n$. The physical network is a directed graph, $G_p = (R, E_p)$, where E_p consists of edges each of which represent the flow of sensory information (relative state). Thus the edge $(R_i, R_j) \in E_p$ whenever robot R_i can see robot R_j. $G_c = (R, E_c)$ is an undirected graph representing the communication network. The edge set E_c consists of pairs of robots that can communicate with each other (assuming omnidirectional transmitters and receivers on each robot) G_p and G_c are determined by constraints of the hardware, the physical distribution of the robots, and the characteristics of the environment.

The key goal is to design the computational network. This network is modeled by a directed acyclic graph $H = (R, E)$. In our work, E_c is fully connected with $n^2 - n$ edges, and E consists of edges that belong to $E_p \cup E_c$. There are two special cases: (a) robots are guided simply via communication ($E = E_c$); and (b) only via line-of-sight sensing ($E = E_p$). The ability of R_j to listen to R_i allows R_i to broadcast its state and feedforward information for R_j to use feedforward control and to improve its estimates of the relative state. The design of the graph H is based on the task. In this paper we will primarily interested in controlling the formation of a group of mobile robots and hence we call H a *control graph*.

Control Graphs: The control graph is a directed graph $H = (R, E)$. An edge $(R_i, R_j) \in E$ if the input u_j associated with the robot R_j depends on the state of the agent R_i. Because each robot has two inputs, we will also assume the in-degree of each vertex is also at most two [3]. If a robot R_j has an in-degree of one, we will associate with the robot a Separation-Bearing controller which maintains the robots separation, l_{ij}, and bearing, ψ_{ij}, with respect to R_i (see Figure 2). If a robot R_k has an in-degree of two, and incoming edges from R_i and R_j, this will mean the robot is controlled by a Separation-Separation controller which maintains the robot separation with respect R_i and R_j. A column in the adjacency matrix with all zeros corresponds to the group leader. A row of zeros signifies a terminal follower.

In addition, we are interested in: (a) the position and orientation of the formation reference frame in space denoted by g; and (b) the shape of the formation denoted by $\boldsymbol{r}$. In our case, g is an element of the motion group $SE(2)$, while $\boldsymbol{r} \in \mathbb{R}^{2(n-1)}$ (for our planar formations), the *shape vector*, describes the distribution of the robots around g in Cartesian coordinates. If we assume that g denotes the position and orientation of the leader of the formation, *i.e.* $g = (x_1, y_1, \theta_1)$, the shape vector is: $\boldsymbol{r} = [\bar{x}_2, \bar{y}_2, \bar{x}_3, \bar{y}_3, \ldots, \bar{x}_n, \bar{y}_n]^T$ where $(\bar{x}_j, \bar{y}_j) = (x_i - x_j, y_i - y_j)$ are the coordinates of robot R_j in a formation fixed frame (relative to R_i).

Given the control graph H, we can reparametrize the shape using the variables that are being regulated by the controllers corresponding to the edges. For example, if R_i is the leader, the coordinates $(\bar{x}_j, \bar{y}_j)$ can be replaced by $\rho_j = [l_{ij}\ \psi_{ij}]$ (see Eqn. 1 and Figure 2) where the transformation $\rho_j = T_j(\boldsymbol{r})$ yields:

$$
\begin{aligned}
l_{ij} &= \sqrt{(\bar{x}_j - d\cos\theta_j)^2 + (\bar{y}_j - d\sin\theta_j)^2}, \\
\psi_{ij} &= \pi - \arctan 2(d\sin\theta_j - \bar{y}_j, \bar{x}_j - d\cos\theta_j) - \theta_i.
\end{aligned}
$$

where d is the offset on the follower shown in Figure 2. We will use ρ to define this reparametrized shape vector since it is more closely related to the control graph. Obviously, for a given $\boldsymbol{r}$, ρ is not unique. There is an equivalence relation $\sim$ on pairs (ρ, H), so that $(\rho, H) \sim (\tilde{\rho}, \tilde{H})$ if they correspond to the same shape vector $\boldsymbol{r}$.

Thus our group of robots can be modeled by a tuple $\mathcal{F} = (g, \rho, H)$ where $g(t) \in SE(2)$ is the reference trajectory of the lead robot, ρ represents the shape with respect to the formation reference frame (attached to the lead robot in our case), and H is the *control graph*. It is important to note when orientations are considered (as is the case for nonholonomic systems), the shape ρ (or $\boldsymbol{r}$) does not capture the complete shape of the system.

3 Organization into an Ad Hoc Network

The main goal of this paper is to be able to organize and move a group of robots to a desired location while achieving and maintaining a desired formation. We assume that each robot has a definite identity that can be determined by visual observations as well as by communication. Our robots use a broadcast protocol for communication. We assume all robots can hear each other. Robots that cannot "talk" or cannot "listen" are left out of the group. However, we do allow visibility constraints. Our robots have a limited field of view and each robot can only see neighbors within a spec-

ified disk. Finally, we assume a planned trajectory $g^{des}(t)$ and a desired formation r^{des} are specified either by a human operator or by an agent that is at a superior level. In order to accomplish our goal, it is necessary to develop the computational network based on edges in the physical and communication network. Because the leader-follower controllers for this network must be assigned on the fly, we call the network *ad hoc*.

We envision four steps to establishing the *ad hoc* network (see Figure 3) - (a) *leader election* to determine the leader of the group; (b) a *discovery* process in which the sensory information and the physical network associated with visibility are used to establish a spanning tree rooted at the leader; (c) a *control graph* or assignment of controllers to each robot to maximize control performance, and (d) *cooperative localization* to combine sensory information and information broadcast from other robots in order to obtain relative position and orientation information required for control.

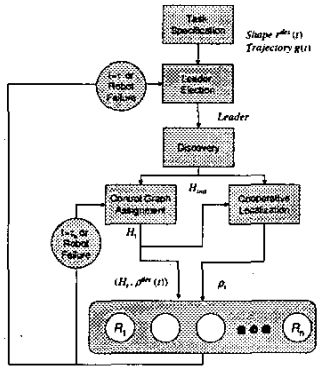

Figure 3: The four step procedure for creating *ad hoc* computational networks for mobile robots.

The overall approach is described in Figure 3. The leader election process allows the group to establish a leader for the specified task. The discovery process allows robots to use sensory information to establish their neighbors, and construct a spanning tree rooted at the leader. This spanning tree is used to initialize the robot controllers by establishing a default control graph, H_{init}. Local heuristics are used to adapt and refine the control graph depending on the shape of the formation and on environmental conditions. The cooperative localization process explained later in Section 5 provides the shape vector $\rho(t)$ corresponding to the control graph $H(t)$ at time t.

We utilize a simple *election* scheme where every robot connected by the wireless network *votes* for leadership. We use a broadcast protocol (UDP based) for communication. Recall our assumption that every robot can communicate to every other robot in the network through broadcast messages. Our election algorithm ensures that the system stabilizes to elect a single leader from any initial state. It is also robust to failure of leader nodes (leads to re-election) and we avoid deadlock by assigning *time outs* for dormancy at each robot. The choice of a number for a *vote* by each robot can be either randomly generated or, as in our case, can depend on a local metric at that robot (node) that quantifies its "leadership" capability. For given distribution of n nonholonomic robots and a desired goal position we can define the vote for the k^{th} robot to be

$$\Lambda_k = \frac{1}{d_{goal,k} - \frac{1}{n}\sum_{j \in J_k} d_{j,k}} \tag{2}$$

where, $d_{j,k} = d(x_j, x_k)$ is a distance metric for points in $SE(2)$, and $J_k \subset (\{1, 2, \ldots, n\} - k)$ is the set of all visible neighbors of R_k. Thus, the robot that is nearest to the goal with the furthest mean distance to its neighbors has the highest vote. Obviously a different task would merit a different choice of Λ_k.

The discovery process is the process of identifying all robots in the group based on sensor measurements and on broadcast information, and establishing a spanning tree that connects all robots in the group. The broadcast protocol is used to implement a fast distributed breadth first search [1] on the graph G_p induced by the sensor visibility constraints.

In Figure 3, the cooperative localization runs at sensor frame rate every τ_s seconds, while the refinement of the control graph is done every τ_c seconds, and the leader-election followed by the discovery process is repeated every τ_l seconds, where $\tau_l > \tau_c > \tau_s$. In the next two sections, we describe distributed approaches to cooperative localization and the assignment of edges in the control graph.

4 Cooperative Control

We now specifically address the following problem — given a distribution of n robots and a desired planar shape parametrized by ρ, find a control graph H that assigns a controller for each robot subject to the following two constraints: (a) *kinematic constraints* that must be satisfied by the relative position and orientation between neighboring robots; and (b) *sensor and communication constraints* based on the limits on range and field of view of sensor and communication device(s) that prevent a robot from obtaining complete information about its neighbors. Among the feasible control graphs that satisfy the constraints, we select those control graphs that locally maximize safety (minimize the likelihood of collisions) and formation stability.

The nonlinear kinematic controllers SBC and SSC have singularities (see [5]) which constrain the choice of controllers based on the configuration of the group. The Separation-Bearing Controller is not defined for a pair of robots (R_i, R_j) having initial relative orientation satisfying $|\theta_i - \theta_j| = \pi$, which corresponds to them facing towards or away from each other. Further, the Separation-Separation Controller is not applicable for robot R_k to follow R_i, R_j when the inter-robot separations are such that

$\epsilon_{ijk} = l_{ik} + l_{jk} - l_{ij}$ is zero (refer to Figure 2). Thus, the assignment of edges in the control graph is constrained by these fundamental limitations of the Separation-Bearing and Separation-Separation Controllers.

In order to prevent collisions, we want to ensure that the separation between robots R_i and R_j is above a threshold. In addition, we will consider the rate of change of this separation and ensure that relative motion between the robots do not cause this separation to decrease below the threshold rapidly. Consider each robot R_j with control inputs $\boldsymbol{u}_j$ has dynamics given by Eqn. (1). Suppose R_j has to maintain a separation constraint $c_{ij} = c(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq 0$ with a neighboring robot R_i.

$$\dot{c}_{ij} = \mathcal{L}_{f_i} c_{ij} \boldsymbol{u}_i + \mathcal{L}_{f_j} c_{ij} \boldsymbol{u}_j \tag{3}$$

where, $\mathcal{L}_{f_i} c_{ij}$ denotes the Lie derivative of c_{ij} along $\boldsymbol{f}(\boldsymbol{x}_i)$. R_j can estimate the *time to collision* with R_i as:

$$\delta t_{ij} = c_{ij} / \dot{c}_{ij} \tag{4}$$

If robot R_j can estimate $\boldsymbol{u}_i$ either by using an estimator [2] or by explicit communication [9] (this is what we call feedforward information), it can compute c_{ij} and thus estimate δt_{ij}. Both the magnitude and sign of δt_{ij} can be used to identify pairs of robots (R_i, R_j), that are on a collision course.

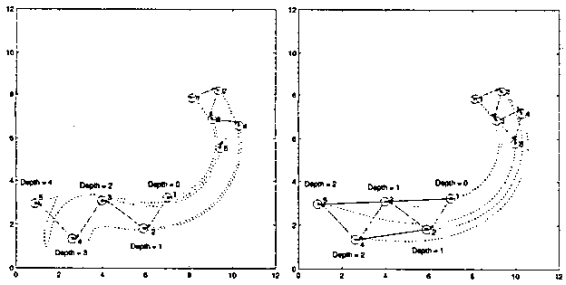

Figure 4: Two 5 robot formations with all SBC links (left), and 1 SBC and 4 SSC links (right).

Although acyclicity of the control graph guarantees stability, the performance associated with a control graph depends on the the *maximum depth*, which we define to be the maximum length of the shortest directed path (assuming all control links have same weights) from the leader to any follower. As this depth becomes greater, the formation shape errors have a tendency to grow. A more precise result that quantifies this relationship can be found in [11]. The comparison between the two formations in Figure 4 illustrates this. The nodes in the formation on the left have a greater depth and this results in larger transient errors. We use a simple heuristic that locally minimizes the maximum depth. When deciding between two control graphs that are otherwise similar we prefer the one with smaller maximum depth.

The control graph assignment algorithm (see Algorithm 1) has to select the appropriate controller for every robot so as

Algorithm 1 Control graph assignment algorithm (CGA)

initialize adjacency matrix $H(i,j) := 0$;
for all robot $k \in \{1, 2, \ldots, n\}$, $k \neq leader$ **do**
  $H(i,k)$:=1 for $SB_{ik}C$, edges(i,k) $\in$spanning tree of G_p;
  d_k:= depth of node k in G_c;
  find set P_k of robots visible to k with depths $d_k, d_k - 1$;
  if $P_k = \emptyset$(disconnected) **then**
    report failure at k, break;
  S_k:=P_k sorted by ascending $timeToCollision$ with k;
  if $numOfElements(S_k) \geq 2$ **then**
    pick last two elements $i, j \in P_k$;
    if $\epsilon_{ijk} = (l_{ik} + l_{jk} - l_{ij}) \neq 0$ **then**
      $H(i,k)$:= 1, $H(j,k)$:= 1 for $SS_{ijk}C$;
    else
      repeat above check for remaining $j \in S_k$ in order;

to maintain connectivity of the graph with the maximum allowable in-degree of two. We assume that each robot can recover the required relative state information for implementing the chosen controller from the cooperative localization process. We follow a two step procedure – (a) assign an initial acyclic leader-follower graph H_{init} with single leader based control links (this is a tree); and (b) refine (add/delete edges) control graph based on local optimality measures. Once the leader is identified, H_{init} is derived via communication by having each robot identifying its neighbors in the physical network. If each robot broadcasts the identities of its neighbors in a prescribed order, a breadth-first search can be used to establish a spanning tree H_{init}. If there are robots with no neighbors in the physical network (*i.e.* with no visible neighbors), we have a disconnected graph.

We use Figure 5 to illustrate a typical scenario showing assignment of a feasible control graphs that optimizes the local heuristics at each node. The visibility graph (left) with an assigned leader allows the robots to use a broadcast protocol to to establish the spanning tree (H_{init}) (middle). The local addition of links results in a refined control graph (right).

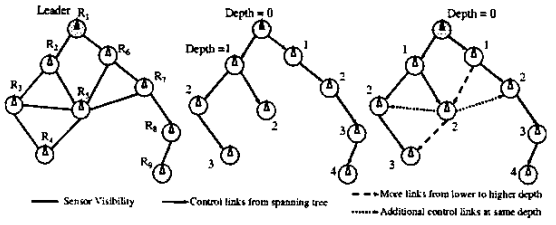

Figure 5: Control graph assignment procedure.

An obvious concern regarding stability of the formation arises when we switch between control graphs and shape vectors in continuous time to achieve and maintain a desired physical shape. In [5] we presented a switching strategy for a triangular formation of three nonholonomic robots. The

system was modelled as a closed-loop hybrid system with three basic modes corresponding to three control graphs with the same desired physical shape. It was shown under some assumptions on the sensor and motion constraints that the system had a common quadratic Lyapunov function [7] and a stable equilibrium point. We are currently researching similar techniques for proving convergence for n robot formations.

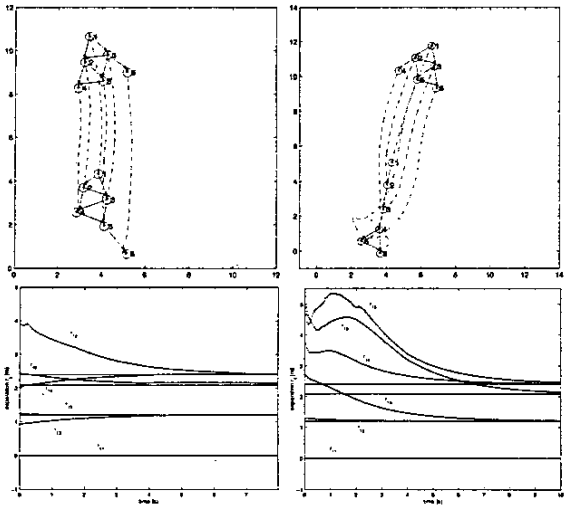

Figure 6: A 6 robot formation with different initial configurations (top). The separations r_{ij} (relative to the leader R_i) converge to the desired values (bottom).

In Figure 6 we simulate the performance of the CGA algorithm from two initial configurations with the same desired shape - an equilateral triangle with six equally spaced robots ($1.2m$ half length for each side). Notice the final control graph H is different in the two cases even thought the desired (and the final) formation shape is the same in both cases.

5 Cooperative Localization

Our use of a broadcast model for intra-formation communication provides a surplus of measurements and constraints. These can be used by individual robots to improve their state estimates as necessary. A natural question is then how should each robot assimilate these additional data to best estimate the formation pose, and consequently improve controller performance?

We assume a sensor model capable of estimating both range and bearing to other teammates, uncertainty expressed in a known variance σ and covariance matrix C, and an objective of localizing a formation of n robots in $SE(2)$. With the above sensor model, each robot is able to estimate the position of its *visible* neighbors without communication. However, estimating relative orientations in such a manner is not practicable. To accomplish this, team members must communicate. By exchanging respective angular measurements, two robots can infer their relative orientations from the measurement differences (this approach extends to $SE(3)$ as well if three or more robots share measurements [9]).

Estimating the pose of the formation can then be accomplished simply by chaining together mutually visible, communicating pairs of robots. This can be formulated as a breadth-first search (BFS) on the graph G_p. The pose of each of the robots can then be estimated as its corresponding node is visited in the graph. The approach is computationally efficient. However, it fails to fully exploit sensor measurements (redundant edges) and geometric constraints (cycles), and as a consequence provides the least accurate estimates of the approaches discussed.

In contrast, we could attempt to estimate the pose of the entire formation at once using standard, iterative optimization techniques with all sensor inputs. Our objective function would be the disparity between the robot position estimates and those obtained from sensor measurements. While such an approach should yield an optimal estimate for the formation pose, there are several drawbacks. Neither solution convergence nor a global minimum is guaranteed, though both might be expected with a good initial estimate (from BFS for example). More importantly, the dimension of the problem state space is $3n - 3$, which for large n makes the method unsuitable for real-time applications.

To address these shortcomings we propose a sequential least-squares (SLS) approach, where the problem of estimating the formation pose is decoupled into two sequential optimization sub-problems: recovering the orientation of robots within the formation, and then recovering the positions. Each can be posed individually as a linear least-squares problem. Consider first the task of recovering robot orientations. Noting that the orientations of the robots θ_i can be related by their relative azimuth angle measurements α_{ij}, α_{ji}, this can then be posed as the following linear least squares problem

$$\min_{\theta_i,\theta_j} \sum \frac{((\theta_i - \theta_j) - (\alpha_{ji} - \alpha_{ij}) - \pi)^2}{\sigma_i^2 + \sigma_j^2} \qquad (5)$$

With the orientations estimated, the positions of the robots can be approximated similarly. Let $[tx_{ij}, ty_{ij}]^T$ represent the translation from R_i to R_j as estimated by R_i, C_i is the sensor covariance matrix for R_i, and J_{ij} corresponds to the Jacobian relating polar and Cartesian coordinates evaluated at r_{ij}, θ_i, and α_{ij}. Solving for the robot positions (x_i, y_i) can then be expressed as the following linear least-squares problem

$$\min_{x_i,y_i} \sum \begin{bmatrix} \bar{x}_j + tx_{ij} \\ \bar{y}_j + ty_{ij} \end{bmatrix}^T J_{ij}^{-T} C_i^{-1} J_{ij}^{-1} \begin{bmatrix} \bar{x}_j + tx_{ij} \\ \bar{y}_j + ty_{ij} \end{bmatrix}$$

where $\bar{x}_j = x_i - x_j$ and $\bar{y}_j = y_i - y_j$. This problem can be solved in closed form. A shortcoming of the SLS approach is the assumption that orientations and positions can be recovered independently, when in fact the two are tightly

coupled. As a result, there is a tradeoff between computational efficiency and optimality of the solution.

In an attempt to characterize the relative performance of these localization approaches, we conducted simulations of a six agent robot formation as reflected in both the initial (A) and final (B) configurations of Figure 6 (left). Range data were subjected to normally distributed noise with a variance proportional to the fourth power of the range, while azimuth angle readings suffered from normally distributed noise of constant variance. Using these imperfect measurements, we proceeded to apply the three schemes to recover an estimate for the formation pose.

Table 1 shows a sample localization trial of the initial formation (A), emphasizing the parameters of interest for the respective controllers as inferred from the localization data. Results from over 150 simulation trials indicate that the performance correlated well with the complexity of the approach. BFS's failure to utilize redundant measurements resulted in the least accurate estimates, with average errors of $10.6cm$ in position, and $0.8°$ in orientation. In contrast, using the SLS approach resulted in significantly lower errors in both position and orientation ($5.9cm$, $0.6°$), though not as good as the quasi-Newton based global approach provided ($4.4cm$, $0.6°$). The latter's inability to improve orientation estimates over the SLS approach was not unexpected, as the latter generates an independent optimal estimate for θ_i. Though by no means exhaustive, these results show the potential benefits from cooperatively fusing distributed sensor measurements, and for the SLS approach as an effective, computationally efficient method for assimilating data.

			BFS	SLS	Global
R_i	Ctrl Param	Actual	Est	Est	Est
2	l_{12}	1.034	0.903	0.972	1.002
	ψ_{12}	2.434	2.413	2.418	2.419
3	l_{13}	1.613	1.604	1.601	1.599
	l_{23}	1.189	1.188	1.208	1.204
4	l_{24}	1.458	1.385	1.523	1.490
	l_{34}	1.854	1.712	1.868	1.877
5	l_{35}	1.312	1.404	1.385	1.365
	l_{45}	1.128	1.088	1.137	1.123
6	l_{56}	1.373	1.422	1.423	1.423
	ψ_{56}	4.059	4.055	4.055	4.055

Table 1: Localizer results in pose estimation.

6 Conclusions

We present a general procedure and a set of algorithms that allows a group of nonholonomic mobile robots to organize themselves into an *ad hoc* computational network in order to move to a specified position in a specified formation along a specified group trajectory. Our procedure guarantees the robots are able to organize themselves into a team utilizing cooperative localization and control strategies. The properties of the leader-follower feedback controllers guarantee convergence to the desired shape provided changes in the control graph do not occur. Since our algorithms scale linearly with the number of robots, it is reasonable to expect good performance for a large number of robots. The two important issues that are yet to be addressed are: (a) stability under changes in control graphs; and (b) fault-tolerance with respect to failures in sensors, actuators, and communication links. Our preliminary work in stability analysis [5] points to stability results for a team of three robots and provides some answers to the first question. The second area is an area of current interest for us and we hope to report on this in future publications.

Acknowledgements: This work was supported by AFOSR grant no. F49620-01-1-0382 and NSF grant no. CDS-97-03220.

References

[1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 1997.

[2] A. Das, R. Fierro, V. Kumar, J. Southall, J. Spletzer, and C. J. Taylor. Real-time vision based control of a nonholonomic mobile robot. In *IEEE Int. Conf. Robot. Automat.*, pages 157–162, Seoul, Korea, May 2001.

[3] J. Desai, J. P. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2864–2869, Leuven, Belgium, May 1998.

[4] J. Feddema and D. Schoenwald. Decentralized control of cooperative robotic vehicles. In *Proc. SPIE Vol. 4364, Aerosense*, Orlando, Florida, April 2001.

[5] R. Fierro, A. Das, V. Kumar, and J. P. Ostrowski. Hybrid control of formations of robots. *Proc. IEEE Int. Conf. Robot. Automat.*, pages 157–162, May 2001.

[6] J. S. Jennings, G. Whelan, and W. F. Evans. Cooperative search and rescue with a team of mobile robots. *Proc. IEEE Int. Conf. on Advanced Robotics*, 1997.

[7] D. Liberzon and A. S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems*, 19(5):59–70, Oct. 1999.

[8] D. Rus, B. Donald, and J. Jennings. Moving furniture with teams of autonomous robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 235–242, Pittsburgh, PA, Aug. 1995.

[9] J. Spletzer, A. Das, R. Fierro, C. J. Taylor, V. Kumar, and J. P. Ostrowski. Cooperative localization and control for multi-robot manipulation. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, October 2001.

[10] T. Sugar and V. Kumar. Control and coordination of multiple mobile robots in manipulation and material handling tasks. In P. Corke and J. Trevelyan, editors, *Experimental Robotics VI: Lecture Notes in Control and Information Sciences*, volume 250, pages 15–24. Springer-Verlag, 2000.

[11] H. Tanner, V. Kumar, and G. Pappas. The effect of feedback and feedforward on formation iss. In *Proc. IEEE Intl. Conf. Robot. Automat.*, pages 3448–3453, Washington D.C., May 2002.

[12] S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *Proc. IEEE Int. Conf. Robot. Automat.*, San Francisco, CA, May 2000.