Kinematic Path-planning for Formations of Mobile Robots with a Nonholonomic Constraint

Timothy D Barfoot[1], Christopher M Clark[2], Stephen M Rock[2], and Gabriele M T D'Eleuterio[1]

[1] *University of Toronto, Canada, tim.barfoot@utoronto.ca, gabriele.deleuterio@utoronto.ca*
[2] *Stanford University, USA, chrisc@sun-valley.stanford.edu, rock@sun-valley.stanford.edu*

Abstract

A method of planning paths for formations of mobile robots with nonholonomic constraints is presented. The kinematics equations presented in this paper allow a general geometrical formation of mobile robots to be maintained while the group as a whole travels an arbitrary path. It is possible to represent a formation of mobile robots by a single entity with the same type of nonholonomic constraint as the individual members. Thus, any path-planner or control method may be used with the formation as would be applied to an individual robot. Equations are developed for changing the geometrical formation and hardware results are presented from the Stanford MARS Testbed.

1 Introduction

Networks of mobile robots for planetary exploration [2] and terrestrial applications are currently under much investigation. A key element in the control of such networks is path-planning [4, 5]. The ability to maintain a specific formation of robots is of particular importance in space exploration where data may need to be collected simultaneously from a distributed array of instruments (e.g., seismology, meteorology, three dimensional vision). The very nature of exploration dictates that such arrays of instruments may often need to be relocated in the work environment. This paper investigates the kinematics of formations of mobile robots, that is, how certain geometrical configurations of mobile robots can be maintained during motion. Once the geometrical configuration of the formation is defined, the equations of this paper allow the network of robots to be treated as a single large robot. Any existing path planner, controller or tele-operator can be used to specify the reference path of the group as a whole. Given this reference path each robot in the formation is able to compute its individual path. This may have particular relevance for tele-operation of such a network from Earth where communication bandwidth is low and latencies are high.

There have been several approaches to maintaining formations of mobile robots including potential field methods [1], rigid formations [10], leader-follower [11], and neural networks [7]. This paper is most similar to the leader-follower work (e.g., [6]) in that we use a reference path and define the positions of each robot relative to this path. Most methods then use nonlinear feedback controllers to allow the followers to maintain formation behind the leader. Our approach is somewhat unique in that we explicitly plan the paths of every robot in the formation and we specifically alter the formation's shape while turning [3] to accommodate the nonholonomic constraint of the robots. We do not advocate any particular feedback control method to enable each robot to actually track its planned path (there are many possibilities including [9]).

Another important aspect of travelling in formation is the ability to dynamically reconfigure the shape of the formation [6, 8]. This could be useful if, for example, the formation needs to nominally be very large (to collect data) but must become smaller to fit between two obstacles. In our approach, we hope that by planning how this reconfiguration occurs it is possible to avoid collisions (at least at the planning level). This may again be useful for tele-operation where the operator simply wants the formation to, for example, become larger, but does not want to manually program each robot nor worry about collisions.

This paper is organized as follows. We first discuss static formations, those which we simply would like to maintain during motion. We then move on to dynamic formations, those which we would like to alter during motion. These are followed by experimental results from the MARS Testbed, discussion, and conclusions.

2 Static Formations

A common model for a mobile robot with a nonholonomic constraint in Cartesian inertial space is the

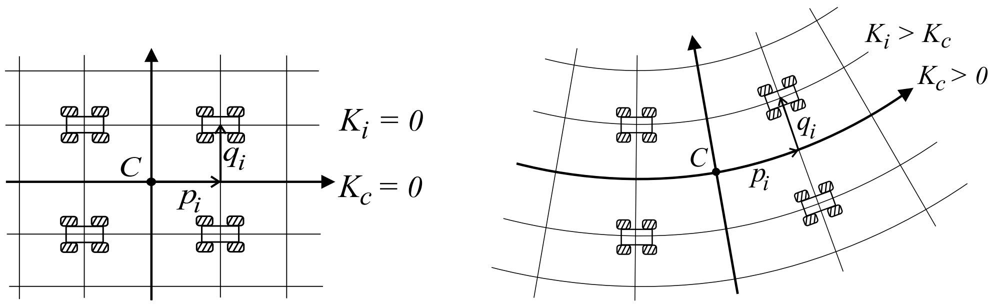

Figure 1: *(left) Square formation. (right) Square formation while turning.*

following

$$\dot{x}_c = v_c \cos\theta_c, \quad \dot{y}_c = v_c \sin\theta_c, \quad \dot{\theta}_c = v_c K_c \tag{1}$$

where $\begin{bmatrix} x_c & y_c & \theta_c \end{bmatrix}^T$ are the coordinates of the robot, v_c is the current speed, and K_c is the curvature of the path the robot is following (which a monotonic function of the steering angle of the robot). The nonholonomic constraint of this system is

$$-\dot{x}_c \sin\theta_c + \dot{y}_c \cos\theta_c = 0 \tag{2}$$

In general the speed and curvature of the robot are limited so we have the additional constraints

$$|K_c| \le K_{c,max}, \qquad |v_c| \le v_{c,max} \tag{3}$$

Note, for an ideal differentially driven robot the curvature, K, is not limited (it can turn on the spot). The first thing we must note about a geometrical formation of mobile robots is that the limits of equation (3) on the group as a whole are different than those on the individual robots. For example, consider figure 1 depicting four differentially driven mobile robots in a square. Simply because the individual robots can turn on the spot does not imply that the square formation can turn on the spot (while maintaining the square). In fact, to maintain a perfect square (including the orientation of the robots) the only possible motion is in a straight line.

In order to turn the formation, a concession must be made. We suggest that one approach is to maintain the formation in curvilinear coordinates rather than in the original rectilinear coordinate system. Figure 1 shows what the formation would look like while turning. By taking this approach we may steer the formation and thus trace any path (subject to the nonholonomic constraint). We introduce an arbitrary reference point, C, within the formation whose coordinates serve as a single set of reference coordinates for the group. This point could be the center of the formation, one of the robots in the formation, or any other point. All robots in the formation will be described relative to this reference point (but in curved space for reasons described above).

The kinematics equations for the reference point are now given by equations (1)-(3) where $K_{c,max}$ and $v_{c,max}$ are functions of the specific formation. For the remainder of this paper we assume that the path (and controls) of the reference point, C, are known. That is, we assume that

$$\begin{bmatrix} x_c(t) & y_c(t) & \theta_c(t) \end{bmatrix}^T, \qquad \begin{bmatrix} v_c(t) & K_c(t) \end{bmatrix}^T \tag{4}$$

are given (t is time). These may be obtained by any existing path-planning algorithm or controller. It is more convenient to use the controls as a function of distance, $d_c(t)$, than time. It is not difficult to use equation (4) to compute

$$v_c(d_c), \quad K_c(d_c), \quad d_c(t) = \int_0^t v_c(\tau) d\tau \tag{5}$$

This allows the geometry of the reference path to be described independently of the speed along that path. We would now like to compute the path for each of N robots in a formation relative to the reference point C. Figure 1 depicts the $\begin{bmatrix} p_i & q_i \end{bmatrix}^T$ coordinates of robot i relative to the reference point, C. If $\begin{bmatrix} p_i & q_i \end{bmatrix}$ are constant then we arrive at the following controls for robot i in order to remain in formation

$$\begin{aligned} v_i(s_i) &= v_c(s_i)\Big(1 - q_i K_c(s_i)\Big) \\ K_i(s_i) &= \frac{K_c(s_i)}{1 - q_i K_c(s_i)} \end{aligned} \tag{6}$$

where $s_i(t) = d_c(t) + p_i$. Together with

$$\dot{x}_i = v_i \cos\theta_i, \quad \dot{y}_i = v_i \sin\theta_i, \quad \dot{\theta}_i = v_i K_i \tag{7}$$

these equations describe the kinematics of robot i. Note, we must check that

$$|K_i| \le K_{i,max}, \qquad |v_i| \le v_{i,max} \tag{8}$$

at all points on the path to ensure $\begin{bmatrix} v_c(d_c) & K_c(d_c) \end{bmatrix}^T$ is a valid reference path. If not, the path cannot be implemented by robot i and this part of the formation cannot be maintained.

An advantage of this approach is that the shape of the formation can be described independently of the reference path. This implies that it is possible to compute the reference path, $\begin{bmatrix} v_c(d_c) & K_c(d_c) \end{bmatrix}^T$, in real-time. To do so, the reference point, C, must be the furthest forward point in the formation (in the direction of travel) or some individual robots will require knowledge about the reference path that has not yet been determined. An equivalent condition is that the p_i must all be opposite in sign to v_c.

3 Dynamic Formations

The more general case is to allow the shape of the formation to change over time. For example, this might be necessary to allow the group to fit through a small opening. This approach handles the general case very nicely by allowing $\begin{bmatrix} p_i & q_i \end{bmatrix}^T$ to be functions of time, or equivalently, distance. To simplify matters we again define the distance travelled by robot i along the reference trajectory as

$$s_i(t) = d_c(t) + p_i(t) \tag{9}$$

where we notice that $p_i(t)$ now changes with time. We also now have $q_i(t)$ as a function of time but we will work instead with $q_i(s_i)$. It in fact becomes necessary to determine all of the following quantities before computing the path of robot i

$$s_i(t), \quad q_i(s_i), \quad \frac{dq_i}{ds_i}(s_i), \quad \frac{d^2 q_i}{ds_i^2}(s_i) \tag{10}$$

Figure 2 interprets these quantities graphically. Once they have been determined, we may compute the local path of robot i as the following

$$\begin{aligned} v_i &= SQv_c & (11)\\ K_i &= \frac{S}{Q}\left(K_c + \frac{\left(1 - q_i K_c\right)\frac{d^2 q_i}{ds_i^2} + K_c\left(\frac{dq_i}{ds_i}\right)^2}{Q^2}\right) \end{aligned}$$

with

$$S = \operatorname{sgn}\left(1 - q_i K_c\right) \tag{12}$$

$$Q = \sqrt{\left(\frac{dq_i}{ds_i}\right)^2 + \left(1 - q_i K_c\right)^2} \tag{13}$$

where all quantities are evaluated at $s_i(t)$ not $d_c(t)$. Together with equations (7) these describe the kinematics of robot i. We again must ensure the limits in equation (8) are not exceeded at all points on the path. The factor, $\operatorname{sgn}\left(1 - q_i K_c\right)$, in both of the above equations accounts for the fact that a robot might have to move backwards to maintain formation. Notice that when the curvature of the reference path, K_c, is zero, the space is not curved and equation (11b) simplifies to

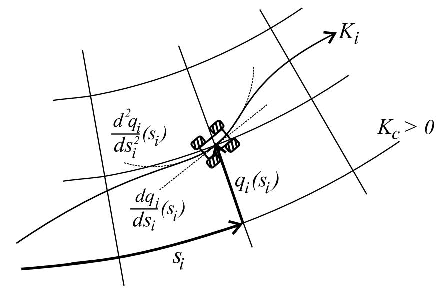

Figure 2: *Graphical depiction of dynamic robot path and associated quantities.*

$$K_i = \frac{\frac{d^2 q_i}{ds_i^2}}{\left(1 + \left(\frac{dq_i}{ds_i}\right)^2\right)^{\frac{3}{2}}} \tag{14}$$

which is a common expression for curvature. Thus we may interpret the two terms of equation (11b) as stemming from the curvature of the reference path (left) and the curvature of the maneuver (right).

A major advantage of this approach is that we may design formation maneuvers independently of the reference path. A library of transformations may be built up and used repeatedly with any reference path (e.g., switch from a square to a line, magnify the current shape, or even permute positions within the same formation).

4 Example Maneuver

It will be useful to give an example of a maneuver within a formation. One possibility is to increase the coordinate $q_{i,o}$ by an amount $\delta q_i = q_{i,f} - q_{i,o}$ over a distance, $\delta s_i = s_{i,f} - s_{i,o}$. To do this we may use the following path

$$q_i(s_i) = \begin{cases} q_{i,o} & 0 \le s_i \le s_{i,o} \\ q_{i,o} + \delta q_i b_i^2(3 - 2b_i) & s_{i,o} < s_i \le s_{i,f} \\ q_{i,f} & s_{i,f} < s_i \end{cases} \tag{15}$$

where $b_i = \frac{s_i - s_{i,o}}{\delta s_i}$. This in turn allows us to compute

$$\frac{dq_i}{ds_i}(s_i) = \begin{cases} 0 & 0 \le s_i \le s_{i,o} \\ 6\frac{\delta q_i}{\delta s_i} b_i(1 - b_i) & s_{i,o} < s_i \le s_{i,f} \\ 0 & s_{i,f} < s_i \end{cases} \tag{16}$$

$$\frac{d^2 q_i}{ds_i^2}(s_i) = \begin{cases} 0 & 0 \le s_i \le s_{i,o} \\ 6\frac{\delta q_i}{\delta s_i^2}(1 - 2b_i) & s_{i,o} < s_i \le s_{i,f} \\ 0 & s_{i,f} < s_i \end{cases} \tag{17}$$

Figure 3 shows this maneuver carried out while the formation is turning. The advantage is now that we have this maneuver worked out we may use it as many times as we like, for any reference path. We simply need to use equations (15)-(17) in (11) to compute the path of robot i either ahead of time or online.

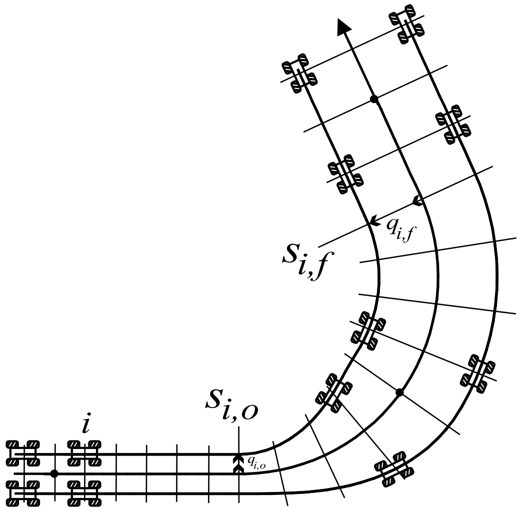

Figure 3: *An example maneuver. robot i moves further away from the reference point during a turn.*

Notice that in figure 3 not only does the q_i increase (in magnitude) for each robot but so does the p_i coordinate. This was not explained above but was accomplished by slowing down (or speeding up) each robot while still tracing out the same path.

5 Experiment

To carry out experiments, the Micro-Autonomous RoverS (MARS) Testbed (Figure 4) at Stanford University was used which models rovers in a two-dimensional workspace. The platform consists of a large 3m x 2m flat, granite table with six autonomous robots that move about the table's surface.

The robots are cylindrical in shape and use two independently driven wheels that allow them to rotate on the spot, but inhibit lateral movement so as to induce the nonholonomic constraint of equation (2). Each robot has its own motion planner located off-board. Control signal processing is also done off-board, and the control signals are sent to the individual robots via a wireless RC signal.

All communication within the MARS platform is accomplished with Real Time Innovation's Network Data Delivery Service (NDDS) software. NDDS

Figure 4: *The Micro-Autonomous RoverS (MARS) Testbed.*

is based on a publish/subscribe architecture. An overhead vision system is used to provide position sensing. Three cameras with infrared filters are used to detect LEDs mounted on the top surface of robots and obstacles. Each robot/obstacle has a distinct pattern of LEDs to distinguish it from other robots/obstacles. The vision system updates the robot's position and velocity at a rate of 30Hz.

The testbed features a Graphical User Interface (GUI) designed in Java/Swing. It provides a top-down view of the table including graphical representations of robots and obstacles (see Figure 5). Setting robot goal locations is accomplished with a drag and drop system. New goal locations are sent to the respective motion planner so trajectories can be constructed. For further details on the MARS Testbed, see [4].

In order to construct the reference trajectory that the formation will follow, a Kinodynamic Randomized Motion Planner (KRMP) similar to that described in [4] and [5] was implemented. The planner provides real-time generation of trajectories that satisfy the necessary dynamic and kinematic constraints (e.g. the nonholonomic constraint).

For real-time formation planning experiments, a single "Leader" robot is used to construct the reference trajectory. The reference point, C, coincides with this Leader. A collision-free reference trajectory for the Leader robot is constructed using the KRMP planner. When checking for collisions, the planner assumes that the Leader is a circular robot with an effective radius equal to the largest distance from the Leader to any other robot in the formation. For a sample maneuver refer to figure 5.

After the reference trajectory construction, the Leader communicates to all other robots in the formation. Assuming robots have knowledge of the

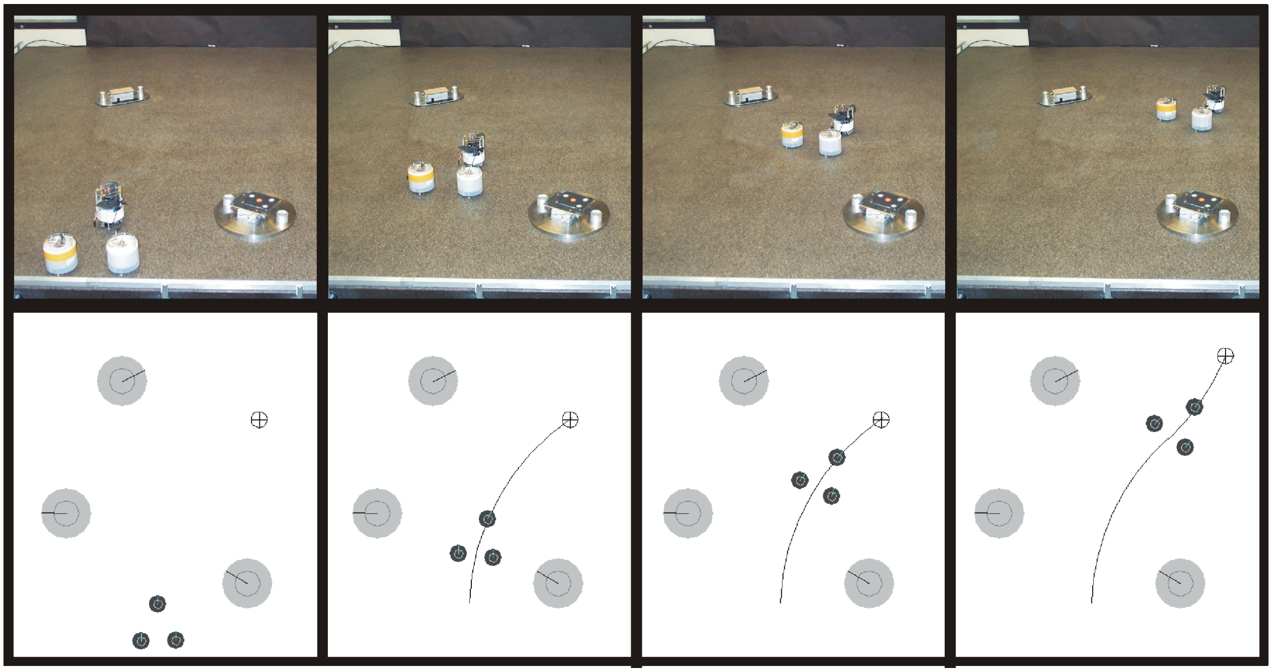

Figure 5: *A typical static formation maneuver. Top row shows time progression of hardware formation. Bottom row shows corresponding graphical display. The large circles represent obstacles and the small circles represent robots. The symbol, $\oplus$, indicates the goal which is dynamic. Note, the reference path has been altered in real-time in the right most column by moving the goal location up and to the right.*

Leader's and their own initial position, they can calculate their offsets from the Leader's initial location. This provides the static offset values p_i and q_i depicted in Figure 1. Knowing the Leader/reference trajectory and their own formation offsets, each robot can plan its path according to equations (6) through (8).

Figure 5 depicts a typical static formation maneuver involving three robots and two stationary obstacles. Photos of the maneuver are presented with their corresponding screen-shots of the GUI below. The three robots, located in the bottom left corner of the image in Figure 5 (left column), begin in a triangle formation. Once the start command is given, the Leader robot constructs a collision-free trajectory to the desired goal location.

In Figure 5 (middle two columns), the formation is progressing successfully towards the goal. Before reaching the goal, the user has moved the desired goal location of the formation to illustrate the real-time planning. In Figure 5 (right column), the formation is continuing along its new trajectory to the goal[1].

[1]The goal relocation occurred far enough in advance to allow all robots in the formation to know the new reference path in a timely fashion.

6 Discussion

The tack we have taken in this paper is somewhat unique in that we explicitly plan the paths for each robot in the formation. We believe this provides an advantage over previous methods, particularly for the case of dynamic formations, as we may plan to avoid collisions during complex formation maneuvers.

Another novelty of our method is that we allow the distances between robots to change (by design) when the formation turns. This was done to incorporate the nonholonomic constraint associated with the robots. This feature may provide a disadvantage for applications where the robots are trying to cooperatively carry large objects. For these applications, methods that attempt to maintain constant distances between robots may prevail [10]. The advantage of this feature is that we may very simply compute the paths of every robot in the formation given the reference path. In fact, we may consider the entire formation to be one robot with the same form of nonholonomic constraint as the individual robots. This not only provides a nice symmetry but allows a great many single robot methods to be recycled and applied at the formation level.

The equations provided allow a precomputed (or real-time) formation maneuver to be combined with a precomputed (or real-time) reference path at run-time which makes the method very flexible and modular. This advantage was strongly demonstrated on the MARS Testbed as an existing Kinodynamic Randomized Motion Planner [4, 5] was used to compute the reference trajectory for a static formation. This trajectory was altered in real-time and the formation was able to easily compensate. It should be stressed, however, that any path-planner (or controller) for a single robot with a nonholonomic constraint could be used to compute the reference path.

Our experimental validation of this technique has to date involved only static formations. We plan to continue this validation for dynamic formations on the MARS Testbed. There are several extensions to this work which we hope to investigate. We gave an example of a formation maneuver that was designed by hand. One extension is to use a multi-robot planner such as in [4, 5] to plan the formations maneuvers and then combine these with a reference path (which could be generated by the same planning method). This would fully automate the process of path-planning for dynamic formations and would easily accommodate knowledge about the environments (e.g., obstacles). We would further like to demonstrate the technique using limited visual capabilities (e.g., directional on-board cameras) rather than an overhead camera for robot localization.

7 Conclusions

A path-planning approach to formations of mobile robots with a nonholonomic constraint was presented. It was shown that a formation could be treated in the same way a single robot is treated thus making a great deal of single robot work relevant at the formation level. The method allows the geometry (both static and dynamic) of a formation to be considered separately from the formation's overall path by providing a way to combine these two components. The method was validated on the Stanford MARS Testbed and found to be very flexible in that an existing path planner was used to generate the reference path for the formation in real-time.

Acknowledgments

The research was funded in part by the Natural Sciences and Engineering Research Council of Canada.

References

[1] T Balch and M Hybinette. Social potentials for scalable multi-robot formations. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, USA, April 2000.

[2] T D Barfoot, E J P Earon, and G M T D'Eleuterio. A new breed: Development of a network of mobile robots for space exploration. Montréal, Canada, June 19-21 2001. The 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS).

[3] Qin Chen and J Y S Luh. Distributed motion coordination of multiple robots. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, volume 3, 1994.

[4] Christopher M Clark, Tim Bretl, and Stephen M Rock. Applying kinodynamic randomized motion planning with a dynamic priority system to multi-robot space systems. Big Sky, Montana, USA, March 9-16 2002. IEEE Aerospace Conference.

[5] Christopher M Clark and Stephen M Rock. Randomized motion planning for groups of nonholonomic robots. Montréal, Canada, June 19-21 2001. The 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS).

[6] Jaydev P Desai, Vijay Kumar, and James P Ostrowski. Control of changes in formation for a team of mobile robots. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, Michigan, May 1999.

[7] K Hirota, T Kuwabara, K Ishida, A Miyanohara, H Ohdachi, T Ohsawa, W Takeuchi, N Yubazaki, and M Ohtani. Robots moving in formation by using neural network and radial basis functions. In *Proceedings of the 1995 International Conference on Fuzzy Systems*, volume 5, pages 91–94, 1995.

[8] Jonathan R Lawton, Brett J Young, and Randal W Beard. A decentralized approach to elementary formation maneuvers. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, USA, April 2000.

[9] C Samson and K Ait-Abderrahim. Feedback control of a nonholonmic wheeled cart in cartesian space. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, USA, April 1991.

[10] Kar-Han Tan and M Anthony Lewis. Virtual structures for high-precision cooperative mobile robotics control. *Autonomous Robots*, 4:387–403, October 1997.

[11] P K C Wang. Navigation strategies for multiple autonomous mobile robots moving in formation. In *Proceedings of the 1989 IEEE International Workshop on Intelligent Robotics and Systems (IROS)*, pages 486–493, Tsukuba, Japan, September 4-6 1989.