Localization and Follow-the-Leader Control of a Heterogeneous Group of Mobile Robots

Jiangyang Huang, Shane M. Farritor, Ala' Qadi, *Member, IEEE*, and Steve Goddard, *Member, IEEE*

***Abstract*—This paper investigates the control and localization of a heterogeneous (e.g., different sensing, mechanical, computational capabilities) group of mobile robots. The group considered here has several inexpensive sensor-limited and computationally limited robots, which follow a leader robot in a desired formation over long distances. This situation is similar to a search, demining, or planetary exploration situation where there are several deployable/disposable robots led by a more sophisticated leader. Specifically, the robots in this paper are designed for highway safety applications where they automatically deploy and maneuver safety barrels commonly used to control traffic in highway work zones. Complex sensing and computation are performed by the leader, while the followers perform simple operations under the leader's guidance. This architecture allows followers to be simple, inexpensive, and have minimal sensors. Theoretical and statistical analysis of a tracking-based localization method is provided. A simple follow-the-leader control method is also presented, including a method for changing follower's configuration. Experimental results of localization and follow-the-leader formation-motion are included.**

***Index Terms*—Bezier trajectory, follow-the-leader control, heterogeneous robot groups.**

I. Introduction

MOST investigations of multirobot cooperative control involve groups of robots with similar capabilities. It is often desirable to have a heterogeneous group of mobile robots with dramatically different onboard sensing, control, and computation. This allows some robots to have minimal sensors and therefore be less expensive and much simpler. Such simple robots can be considered disposable to the overall mission and can be used for high-risk activities and therefore increase the capabilities of the group. Such high-risk and large-scale team tasks include highway traffic control, battlefield environments, or space exploration, where the system's "attitude" toward risk must be explicitly considered.

The application considered in this paper is the use of a heterogeneous group of robotic barrels to improve the safety of highway construction and maintenance workers [1]. Safety barrels (Fig. 1) guide traffic and serve as a visible barrier between traffic and work crews. These barrels consist of a brightly colored plastic drum (approximately 130-cm high and 50-cm diameter) that is attached to a heavy base. The robotic safety barrel replaces the heavy base with a mobile robot that transports the safety barrel. The robots work in teams to provide traffic control. Independent, autonomous barrel motion has several advantages. First, the barrels can self-deploy and self-retrieve, eliminating the dangerous task of manually placing barrels. Second, their positions can be quickly and remotely reconfigured as the work zone changes thus reducing the work zone size. Finally, barrels could continuously follow work crews.

Fig. 1. The follower robot(s).

Proper traffic control is critical in highway work zone safety. Traffic control devices such as signs, barricades, cones, and safety barrels are used. Accidents can occur because of improper work zone design, improper work zone housekeeping, and driver negligence. Implementing reliable robotic safety devices could improve highway work zone safety and could also increase the safety to the general public by creating better and more efficient work zones [1], [2].

The system is designed with distributed planning and control that reduces the per-robot cost (several robots are needed and they are often struck by traffic) by centralizing the intelligence and sensing while keeping communication bandwidth low by distributing local control [2].

The application specifically considered is where a group of barrel robots would autonomously follow a slow-moving maintenance operation, eliminating the need to create enormous worker zones for tasks performed over long distances. Such

Manuscript received September 15, 2005; revised November 30, 2005. Recommended by Guest Editors S. Agrawal and V. Krovi. This work was supported by the National Research Council Transportation Research Board's NCHRP IDEA program.

J. Huang and S. M. Farritor are with the Department of Mechanical Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588 USA (e-mail: jyhuang@bigred.unl.edu; sfarritor@unl.edu).

A. Qadi and S. Goddard are with the Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588 USA (e-mail: aqadi@cse.unl.edu; goddard@cse.unl.edu).

Digital Object Identifier 10.1109/TMECH.2006.871100

tasks include lane striping, mowing, or asphalt overlay [3]. The goal of this paper is to create a method to achieve the follow-the-leader formation for such a heterogeneous robot team. The main challenge is that simple sensor-limited followers do not have the capability of localization and navigation because of their lack of on-board sensing, computation, and communication bandwidth.

To address this challenge, this paper presents a centralized localization and control method for follow-the-leader movement. The complexity of sensing, control, and computation is relegated to the leader while followers only perform simple local operations under the leader's guidance. The leader has sufficient sensors to allow it to self-navigate. The leader may even take on the form of a maintenance pickup or utility vehicle driven by a highway worker. The follower robots have minimal sensors (only wheel encoders for dead-reckoning over short distances) and no hazard avoidance sensing or global localization. To guide these extremely sensor-limited followers during the long distance travel, the leader robot estimates the pose of followers and commands their motion. This method reduces system cost and system power requirements as compared to a completely distributed approach [4]. This heterogeneous group also protects the sophisticated leader robot (or maintenance worker) while using inexpensive robots for the high-risk traffic control.

Section II gives an overview of related work. Section III presents the localization method for the follower robots based on a cubic polynomial least-squares method. The accuracy of this method is then evaluated with a statistical analysis. A simple follow-the-leader control approach that uses Bezier trajectory generation is described in Section IV. Section V presents a centralized reconfiguration method for the follower robots during the follow-the-leader motion while they avoid collision each other. Experimental results in both localization and follow-the-leader motion are presented in Section VI.

II. Related Work

A variety of approaches have been proposed for localization and follow-the-leader formation control of a group of mobile robots. Balch and Arkin [5] propose a behavior-based approach for formation keeping of a team of military unmanned ground vehicles as a scout unit. Each powerful vehicle is equipped with GPS, vision, and hazard sensors. Lewis and Tan [6] apply the concept of virtual rigid structure for formation maintenance. Each robot is controlled to maintain a rigid geometric relationship to each other and to a frame of reference. But their application range is constrained because localization of each robot is computed by a fixed and independent vision-based camera system.

Das *et al*. [7] present a paradigm for switching between decentralized controllers that allows for changes in formation. A single omni-directional camera is used in all robots and a host computer is used as a centralized processing unit. The host computer receives video from all robots and calculates relative velocity and pose between each follower and its leaders. Vidal *et al*. [8] translate the formation control problem into a separate visual servoing task for each follower using similar omni-directional vision. The follower robot uses motion segmentation to estimate the position and velocities of its leader. Using distributed camera sensors in these approaches requires significant sensing and intense computation at each robot (therefore higher per robot cost).

Chiem *et al*. [9] propose a simple and efficient method to control robot formations using Bezier trajectories but only present quantitative results in simulation. Again, each follower robot uses a color-tracking vision camera to estimate the relative pose of the leader. Then a local Bezier trajectory is created and followed. Bezier trajectories have been used in many planning applications [10]–[12]. The work presented in this paper uses Bezier curves for trajectory generation because it is computationally simple and well suited for the simple follower robots used in this paper.

Fredslund *et al*. [13] propose another local sensing method for robot formations. Each follower uses a laser rangefinder to determine the distance to its leader and then pans the camera to center the leader in the camera's field of view. This method can achieve global formation movement. Michaud *et al*. [14] also use similar directional panning cameras and propose a hybrid control approach to dynamically initialize the formation. However, neither method controls the relative orientation between the robots and again each robot requires significant sensing (i.e., expense) and has identical capabilities (homogeneous).

Parker *et al*. [15] present a control approach for heterogeneous robots where a more capable leader assists simpler follower robots to navigate using a "chaining" formation method. Here, the followers use a camera and a color-tracking algorithm to follow the next robot in the "chain."

All of the above approaches allow a team of robots to move in formation. However, each of the above approaches requires the followers to have significant (e.g., vision processing) sensing and computation capabilities. The follower robots in this work are very sensor-limited and much simpler, prohibiting any of the above approaches to be directly applied here. This constraint provides new challenges for localization and follow-the-leader control of such sensor, computation, and communication-limited groups of heterogeneous robots.

III. Follower Localization

The follower robots estimate their global position and orientation via dead reckoning; however, errors (due to a variety of issues, such as wheel slippage) are accumulated over time and thus dead reckoning is only reliable for short distance navigation. This localization method is unacceptable for long travel. Here it is assumed that the leader robot has a precise positioning system. This global knowledge could take the form of accurate GPS or a maintenance worker (who obviously knows his/her location) driving a pickup. The leader (or pickup) then locates the group of sensor-limited followers using a single-laser rangefinder. This method is used to determine the position of the follower robots and successive positions are carefully used to estimate their orientation.

A. Position Estimation

A SICK laser scanner LMS200 mounted on the leader robot is used to estimate the relative position of the follower robots

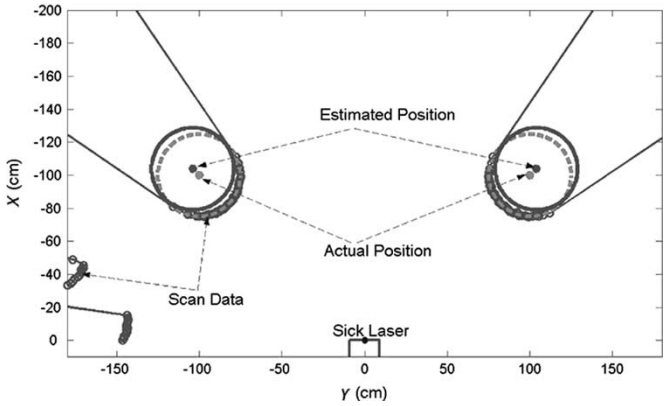


Fig. 2. Positions of two followers with respect to the leader robot.

with respect to the leader. The laser sensor provides distance measurement every 0.5° up to a 32-m range throughout a 180° scanning field. The cylindrical safety barrel (as in Fig. 1) with a 50-cm diameter creates a circle pattern in the two-dimensional plane of the laser scanner. Fig. 2 shows a typical laser scan indicating the positions of two follower robots with respect to the leader robot. The follower robot (i.e., center of the circle) is localized with a modified Hough transform algorithm [16]. This algorithm uses one common search window for each cluster of follower robots to decrease the required execution time. The location of the search windows is updated based on the last location of the follower robots. The current position of the follower robot is determined by comparing the possible positions in the search window with the previous position using the nearest-distance method. The laser scanner functions at a data transfer rate of 500 KBaud and the position of each follower is estimated every 60 ms (total time including laser scanning and algorithm processing).

The position estimation error for each follower is largely a result of the error in the laser scanner's distance measurements ($\sim$1.5 cm rms). The optical properties of the target surfaces can also affect sensing performance [17]. To evaluate the accuracy of the Hough transform algorithm a statistical experiment shows the estimation error in both X (Fig. 3, top) and Y (Fig. 3, bottom). The position estimate is approximately normally distributed with standard deviation less than 0.2 cm. In this experiment the actual robot position is 100 cm behind the leader robot ($X = -100$ cm) and 100 cm to its port ($Y = 100$ cm). Coordinates are defined relative to the leader as in Fig. 2.

With this information, the global position of each follower can be calculated from (1), where $P_r = (X_r, Y_r)^T$ is the measured relative position of the follower with respect to the leader and $P_L = (X_L, Y_L)^T$ and θ_L are the pose of the leader with respect to the global coordinate system

$$P_F = \begin{pmatrix} \cos\theta_L & -\sin\theta_L \\ \sin\theta_L & \cos\theta_L \end{pmatrix} P_r + P_L. \tag{1}$$

B. Orientation Estimation

Now, the position estimates will be used over time to track the motion of the follower robots. This tracking is then used to provide an estimate of the follower's current orientation. Follower orientation can be estimated from the past positions since they are nonholonomic in their mechanical design.

First a cubic polynomial function is adopted to approximate the follower's path based on some number of recent positions of the follower. A cubic polynomial is used because the follower robot's local trajectory generation uses a cubic Bezier curve (see Section IV-A) and this curve uses Bernstein polynomials of third degree as the basis function [18], [19]. The follower robot's path is approximated by (2)

$$Y = a_0 + a_1 X + a_2 X^2 + a_3 X^3 \tag{2}$$

where (X, Y) is the global position of the follower and a_0, a_1, a_2, and a_3 are the coefficients of the cubic polynomial function.

Now, $P_F(t), P_F(t-1), P_F(t-2), \ldots, P_F(t-n+1)$ are n recent-tracked global positions of the follower robot(s) with equal distance d between each location and t the current time step. At least four $(n = 4)$ positions are needed to perfectly fit a cubic polynomial function. However, using exactly four recent positions can lead to error because each position estimate has a normally distributed error (in X and Y, Fig. 3).

Here, $n = 8$ is used to reduce the effects of the error at any given location and a least-squares fit is used to approximate these eight positions with the cubic polynomial function (as in Fig. 4). The coefficients a_0, a_1, a_2, and a_3 can hence be obtained by solving a series of linear equations using Gaussian elimination [20]. Then the approximate orientation θ_F is evaluated from (3) by differentiating (2)

$$\left.\frac{dY}{dX}\right|_{X_F(t)} = a_1 + 2a_2 X + 3a_3 X^2 \theta_F$$
$$= a\tan\left(\left.\frac{dY}{dX}\right|_{X_F(t)}\right). \tag{3}$$

It is important to note that the leader robot may obtain any orientation (i.e., it could drive in circles). In that case, the cubic polynomial function would not accurately estimate the orientation of the follower because a one-to-one function cannot represent the recent path of the follower. Therefore, this requires that the follower robot's tracked positions must be monotonically increasing or decreasing along the global X- or Y-axis (or another suitable set of axes). This constrains the radius of curvature relative to the distance between tracked positions (further discussed in Section III-B2).

The above method is sensitive to the distance d between the tracked positions and the error expected for each location (or the ratio of these values). In following sections, statistical analysis is used to find the appropriate value for distance d between each of the two tracked global positions and the accuracy of the proposed approach is evaluated for common trajectories of the follower robots.

1) Distance Between Tracked Positions d: If the position measurements of the follower robot are precise, the distance d can be small and an accurate estimate of the follower's orientation can be obtained. However, error in this position (as quantified in Fig. 3) can create large errors in the estimation

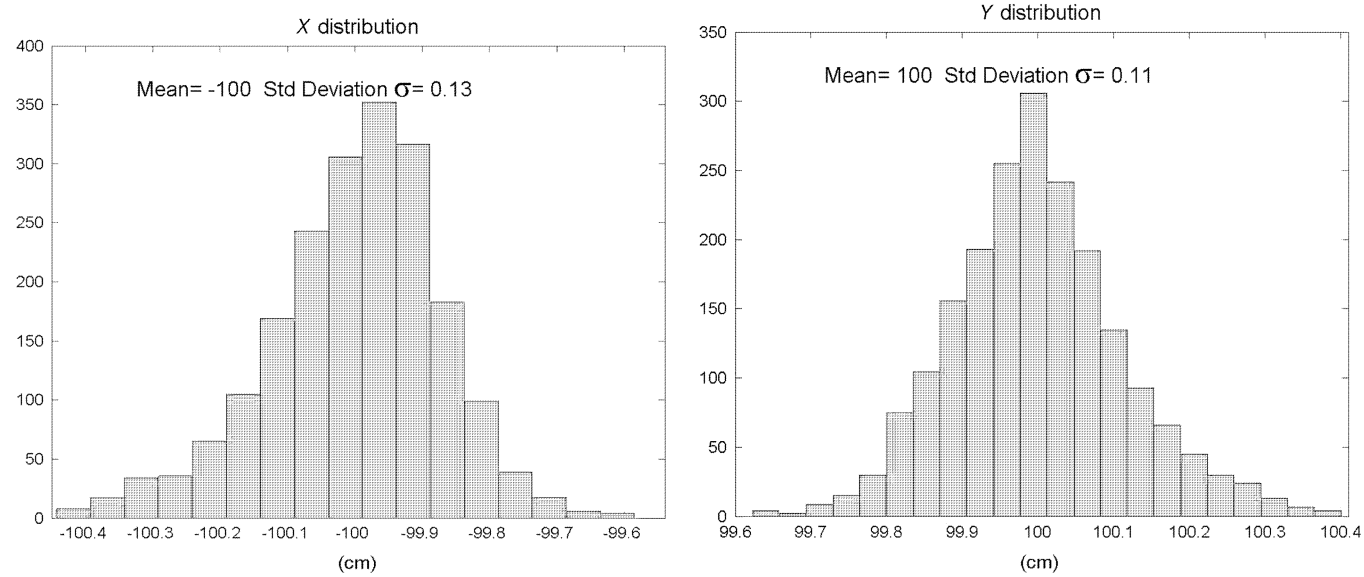


Fig. 3. Position error.

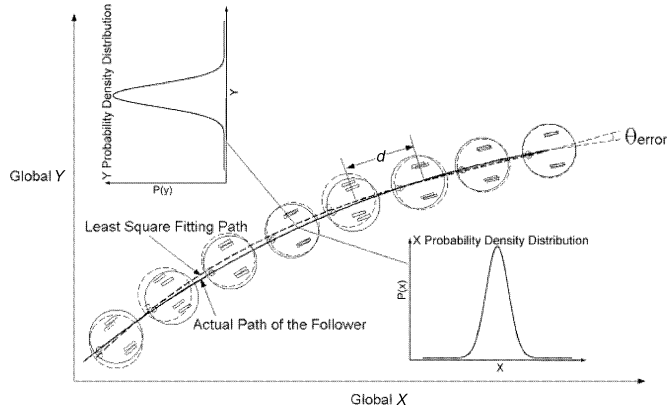


Fig. 4. Eight recent global positions fitted with a cubic polynomial.

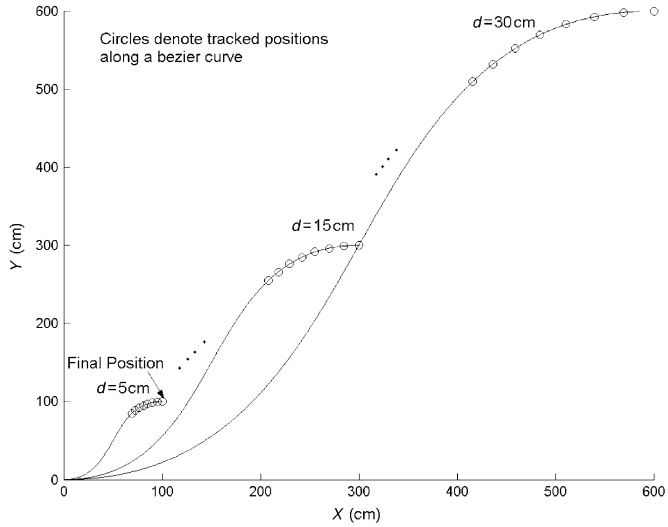


Fig. 5. Tracked positions along a series of Bezier curves.

of the follower's orientation (for a given distance d between positions). The choice of an appropriate distance d should be considered with respect to the standard deviation of error in position estimates. The statistical nature of the error in the follower's position makes an analytical approach to the problem difficult. Therefore, statistical methods will be used along with simple Monte Carlo simulations.

Consider the common (Bezier) motion of the follower shown in Fig. 5. Here, a basis Bezier curve is defined from the pose $(0, 0, 0^\circ)$ to $(100, 100, 0^\circ)$ (shortest curve in Fig. 5). Eight tracked positions are now sampled along this Bezier curve and an estimate of the robot's final position is made. This process is then repeated 1000 times with random noise (as described in Fig. 3, normal distribution and standard deviation 0.2 cm) included in each of the eight estimates of the follower's position. This produces a mean and standard deviation in the estimate of the follower's final orientation. It is known that the desired orientation at the final position of the robot (as defined by the Bezier curve) is 0°. Therefore, an estimate is created of the error in orientation using a given value of d. This process is then repeated with increasing distances between the samples (increasing d). To do this, the Bezier curve is expanded proportionally (the two longer curves in Fig. 5) and each curve produces a mean and standard deviation of the error in the orientation estimation.

The results for several values of d are shown in Figs. 6 and 7. The mean of orientation error is less than 1° for all distances d. This confirms that the cubic polynomial function is a good method to approximate the orientation of the robot for Bezier trajectories. Furthermore, the standard deviation of orientation error decreases when the distance d increases (to a limit). However, to create frequent estimates of the orientation of the follower robot, the distance d should be small. Considering this tradeoff, a distance of $d = 15$ cm with a standard deviation of error of 1.2° is a good choice for the paths analyzed by this simulation (at the knee of the curve). With this d value, it is expected that 99.9999998% of the orientation estimations will be within six standard deviations (7.2°) of the mean error (0.86°).

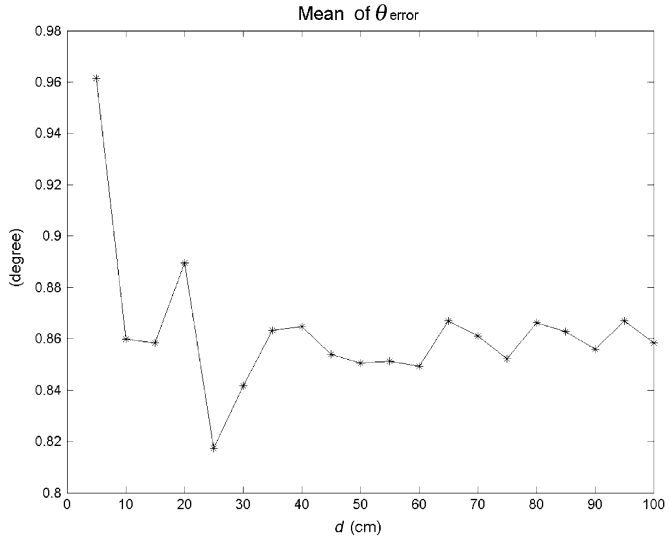


Fig. 6. Mean of orientation error versus distance d.

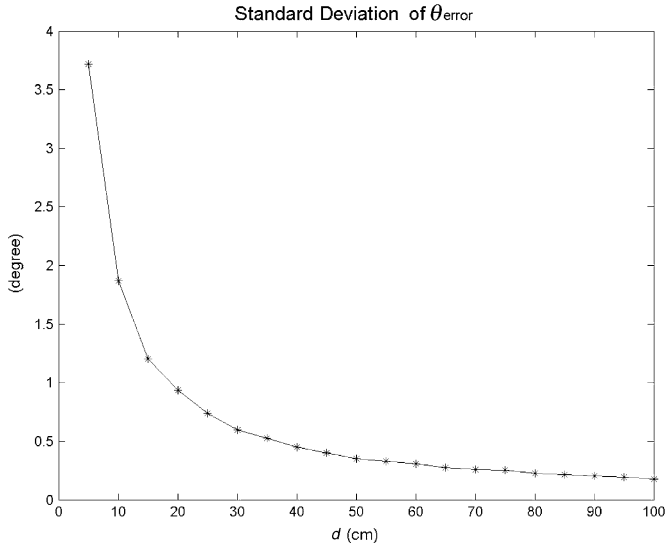


Fig. 7. Standard deviation of orientation error versus distance d.

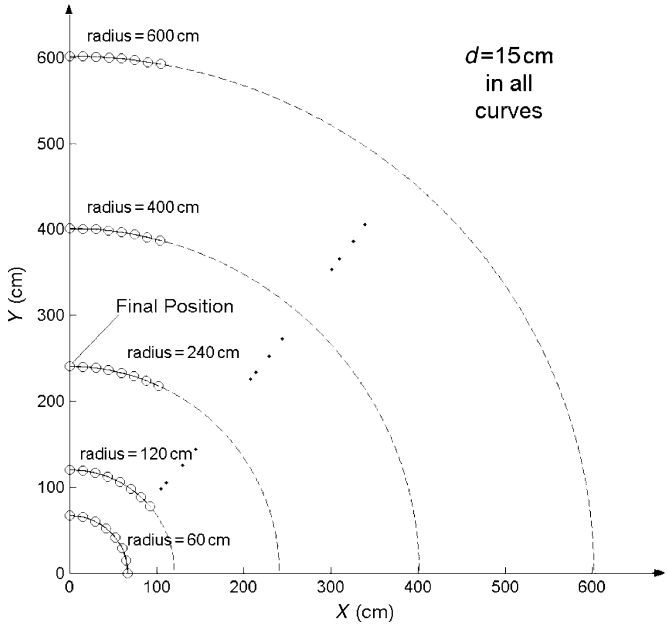


Fig. 8. Eight positions with constant d and increasing radius.

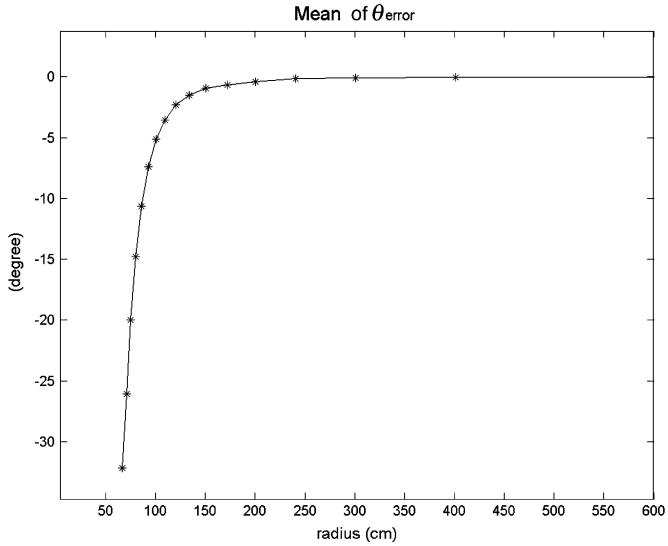


Fig. 9. Mean of orientation error versus radius.

2) Analysis of Arc Following Motion: In Section III-B1 we saw that the appropriate distance d between the tracked follower positions is required to produce an accurate tracking estimate of the follower's orientation (for a given position measurement accuracy). However, the radius of curvature of the follower's path can also produce errors in the estimate of the orientation. This is because the polynomial least-squares fitting method will fail in arcs with small radii. This is because the follower's path (consisting of eight tracked positions) is distinctly different from a cubic polynomial.

To avoid this problem, it is assumed that the motion of the follower has large radius when the follower robot turns. A minimum radius that guarantees accurate estimate of the follower's orientation is determined through statistical simulation.

Consider the arc motions of the follower in Fig. 8. Here the distance d between tracked positions is held constant at 15 cm and the radius of curvature of the follower's arc motion is changed. Again, the X and Y data of each position are normally distributed with standard deviation 0.2 cm (as in Fig. 3). All arcs lie in the first quadrant (rotated here). From a basis arc with the length 105 cm (7×15 cm), the radius is increased while keeping the final position on the Y-axis. Therefore, the desired orientation at the final position is 180° for all arcs. This analysis is then repeated 1000 times for each arc and a mean and standard deviation of the error in orientation estimation are developed corresponding to different radii. The results are shown in Figs. 9 and 10. It can be seen that the mean of orientation error goes to zero and the standard deviation decreases as the radius increases. This indicates that the orientation estimation becomes better and much more certain with increasing radius. This simulation suggests that a minimum turning radius

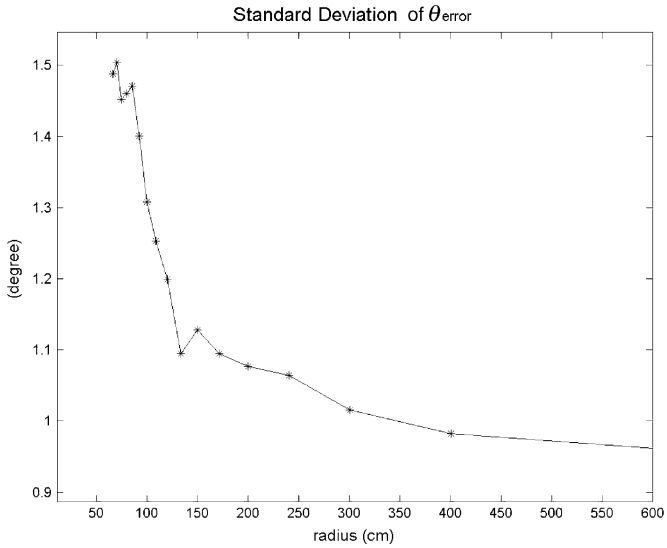


Fig. 10. Standard deviation of orientation error versus radius.

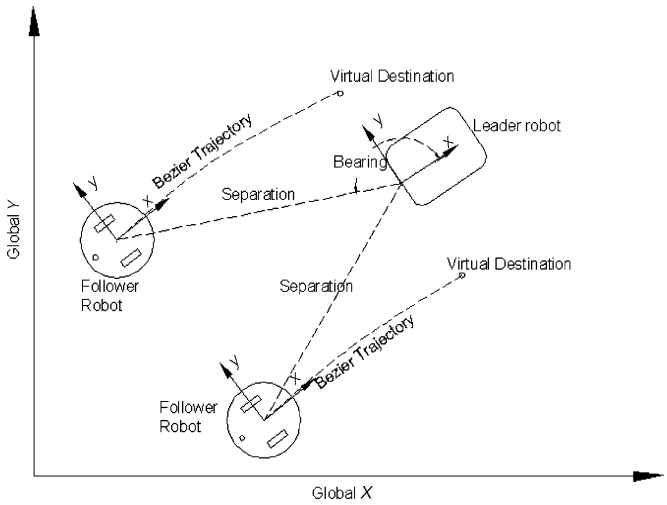


Fig. 11. Follower-the-leader control diagram.

of 150 cm with the standard deviation of 1.13° is an appropriate (knee in curve) choice for the given path and sensor error. This radius creates an expectation that 99.9999998% of the orientation estimates will be within six standard deviations (6.78°) of the mean error (−0.93°). Therefore, the cubic polynomial least-squares method can estimate the orientation of the follower robot with satisfactory accuracy if the radius of the arc is >150 cm.

IV. Follow-the-Leader Control

This section presents a simple follow-the-leader formation-keeping control method that uses the localization scheme outlined in the previous sections. As shown in Fig. 11, each follower (two in this figure) attempts to maintain the desired separation and bearing with respect to the leader. A virtual destination is assigned to each follower to maintain this geometric formation. Then a cubic Bezier trajectory is defined between the follower and this virtual destination to allow the robots to follow the

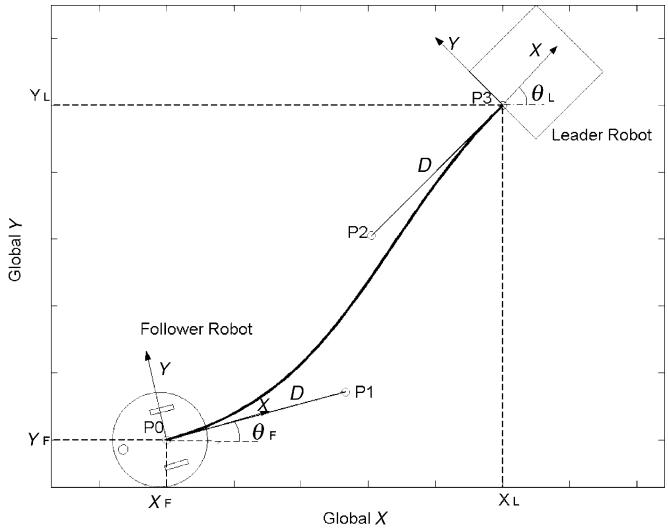


Fig. 12. Bezier trajectory between the leader and follower.

leader. The trajectory is updated in real time since the virtual destination varies as the leader robot moves. This approach keeps communication bandwidth very low since only the updated follower location, the Bezier control points (described in the following section), and the leader robot's velocity need to be communicated to the follower. A simple PID velocity controller is executed by the followers to track the Bezier trajectory with the desired linear velocity. The result is that the followers perform only computationally simple operations and can have minimal onboard computing.

A. Bezier Trajectory

A cubic Bezier curve is used for local trajectory planning on the follower robots. This curve is the weighted sum of four control points (P_0, P_1, P_2, and P_3) as shown in Fig. 12. The shape of the curve can be easily deformed by modifying the relative locations of these points. The virtual destination commanded by the leader robot can be calculated in the local coordinate frame (local to the follower) given the global pose of the leader robot (X_L, Y_L, θ_L) and follower robot (X_F, Y_F, θ_F) (coordinate systems defined in Fig. 12)

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{pmatrix} \cos(-\theta_F) & -\sin(-\theta_F) \\ \sin(-\theta_F) & \cos(-\theta_F) \end{pmatrix} \begin{bmatrix} X_L - X_F \\ Y_L - Y_F \end{bmatrix}$$
$$\theta = \theta_L - \theta_F. \tag{4}$$

Then the local cubic Bezier trajectory is defined by

$$P(q) = (1-q)^3 P_0 + (1-q)^2 q P_1 + (1-q) q^2 P_2 + q^3 P_3 \tag{5}$$

where $P_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, P_1 = \begin{bmatrix} D \\ 0 \end{bmatrix}, P_2 = \begin{bmatrix} x - D\cos\theta \\ y - D\sin\theta \end{bmatrix}, P_3 = \begin{bmatrix} x \\ y \end{bmatrix}, q \in [0,1]$ is the curve parameter, and $D = \|P_1 - P_0\| = \|P_3 - P_2\|$. To determine P_1 and P_2, the distance $\|P_1 - P_0\|$ and $\|P_3 - P_2\|$ are assumed to be equal and is simply given $D = \alpha\|P_3 - P_0\|$. The initial curvature of the Bezier trajectory is related to the choice of D and this curvature affects the performance of the local velocity controller of the follower.

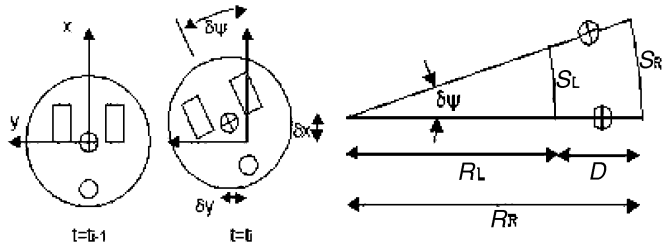


Fig. 13. Differential robot motion.

$\alpha = 0.4$ is chosen from experimental experience. Also, note that the Bezier curve is tangent to both vectors $P_1 - P_0$ and $P_3 - P_2$ as in Fig. 12. Therefore, if the leader robot moves in straight line, the followers can follow their desired positions in the formation.

B. Local Follower Control

The design of a velocity controller is critical to execute the Bezier trajectory, which will allow the follower to maintain the desired formation with respect to the leader robot. Again, the follower receives its updated pose information, the linear velocity of the leader, and the Bezier control points. The follower then defines the Bezier trajectory and tries to track the trajectory with a linear velocity V. The velocity V is adjusted with a factor if the length of Bezier curve is different from the desired separation between the follower and the virtual destination (i.e., P-control on velocity). The length L of the Bezier trajectory is calculated numerically using Simpson's rule [20]. The total time to follow (T_{total}) the full Bezier trajectory is then calculated from (6)

$$T_{\text{total}} = \frac{L}{V}. \tag{6}$$

The parameter q to the control time t is defined by a simple linear function (7) between q and t

$$q(t) = \frac{1}{T_{\text{total}}} t. \tag{7}$$

Therefore, when t varies in the range $[0, T_{\text{total}}]$, the parameter q varies between $[0, 1]$. In each cycle of the control loop, the curvature of the Bezier curve and the angular velocity can be calculated by (8) and (9)

$$\kappa(t) = \frac{\frac{dx}{dq}\frac{d^2y}{dq^2} - \frac{dy}{dq}\frac{d^2x}{dq^2}}{\left(\sqrt{\left(\frac{dx}{dq}\right)^2 + \left(\frac{dy}{dq}\right)^2}\right)^3} \tag{8}$$

$$\omega(t) = V\kappa(t). \tag{9}$$

In this study, the followers are differential drive robots with two actuated wheels and one passive caster (Fig. 1). During a period of differential time between t_{i-1} and t_i as shown in Fig. 13, the robot turns about a given radius by $\delta\Psi$. Here, S_L and S_R are the paths of the left and right wheels (and measured by the wheel encoders). No slip is assumed so the path lengths are related to the change in wheel angles ($S = \delta\theta r$, where r is the wheel radius and $\delta\theta$ is given by the motor encoders). D_w is the distance between the wheels, and R_L and R_R are the turning radii for each wheel.

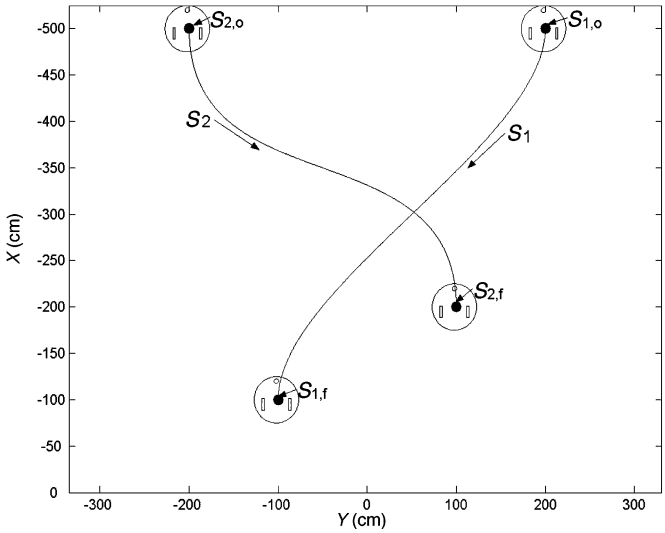


Fig. 14. Centralized reconfiguration.

Now, given the desired linear velocity and angular velocity of the follower, the angular velocities of left and right wheels are obtained from (10) using the follower's kinematics

$$\omega_L = \frac{2V - \omega D_w}{2}$$
$$\omega_R = \frac{2V + \omega D_w}{2} \tag{10}$$

where $V = (V_L + V_R/2), \omega = (V_R - V_L/D_w)$, and $V_L = \omega_L r$ and $V_R = \omega_R r$ are the velocities of the left and right wheel centers, respectively.

A PID velocity controller has been designed to control the wheel angular velocity of the follower robot. Therefore, the follower robot can follow the defined Bezier trajectory with the desired linear velocity by controlling the angular velocity of each wheel.

V. Centralized Reconfiguration

Changing task requirements (e.g., desire to close a different highway lane, etc.) may require the followers to change their position with respect to one another or with respect to the leader. A centralized (performed by the leader) planning method is adopted to coordinate the movement of the sensor-limited followers. For movement of highway safety markers, it is assumed that the primary obstacles for the follower robot are other robots; however, the proposed approach can be expanded to include fixed obstacles or obstacles with known motion. The objective of the reconfiguration algorithm is to avoid collision between followers while maintaining line of sight between the leader and followers.

Consider the example reconfiguration shown in Fig. 14. Here two robots are required to change configuration with respect to the leader (shown in the leader frame as Fig. 2). A Bezier curve

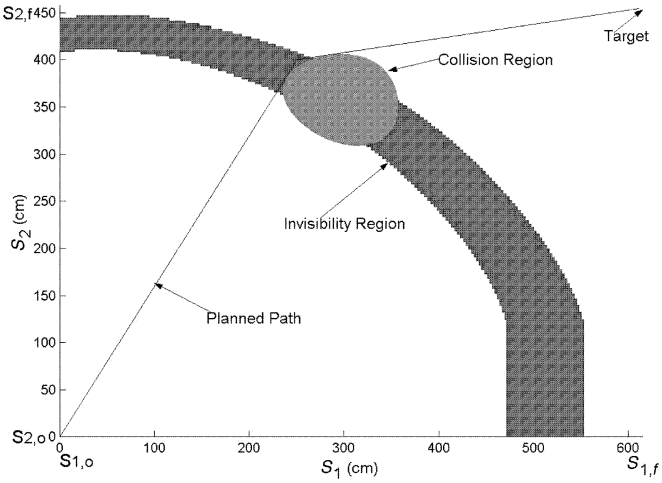


Fig. 15. Motion path shown in variables (S_1 and S_2).

is again used to create the desired robot path. The path can then be rewritten as a distance along the Bezier curve given by S_1 and S_2, where 1 and 2 denote each robot.

The planning problem for the ith robot can then be expressed in terms of the path variable S_i, where S_i changes from $S_{i,o}$ to $S_{i,f}$ as shown in Fig. 15. This approach is similar to a planning method for fixed-base manipulators [23]. Showing the reconfiguration in these coordinates allows the regions of collision (around $S_1 = 300$ cm and $S_2 = 350$ cm) between the robots to be explicitly shown. Collision is defined as the values of S_i such that the distance between the follower robots is below a specified safety threshold. The "invisibility" region is also shown and is defined as configurations where the leader does not have line-of-sight contact with a follower. For example, when the first follower robot is at a location of $S_1 = 200$ cm and the second robot has $S_2 = 400$ cm, the second robot prevents the leader robot from "seeing" the first robot. To cross these regions, the follower must temporarily localize itself with dead reckoning. When visibility is regained, the follower robot will communicate its current positions (estimated from dead reckoning) back to the leader robot. The robot's motion along the path with respect to time can now be chosen to make a fast reconfiguration while avoiding the collision region and minimizing invisibility. Several methods can be implemented and a modified tangent graph method is used here.

The path shown in Fig. 15 gives the ratio of the linear velocities of the two robots. These velocities can then be normalized by a nominal desired velocity and transferred back into the leader robot's coordinate frame. If the leader robot is moving along a straight line with a constant velocity ν_L and orientation θ_L, the relative linear velocity of the follower ${}^r\nu_F$ with respect to the frame of the leader robot can be found from

$$\begin{bmatrix} {}^r\boldsymbol{\nu}_{FX} \\ {}^r\boldsymbol{\nu}_{FY} \end{bmatrix} = \frac{dP}{dq}\frac{dq}{dt} \tag{11}$$

where ${}^r\boldsymbol{\nu}_{FX}$ and ${}^r\boldsymbol{\nu}_{FY}$ are the X and Y components of the relative linear velocity in the leader robot coordinate frame, P is the relative position of the follower, and q is the Bezier curve parameter as indicated in (5) and (7).

The absolute linear velocity of the followers can then be computed from

$$\begin{bmatrix} \boldsymbol{\nu}_{FX} \\ \boldsymbol{\nu}_{FY} \end{bmatrix} = \begin{pmatrix} \cos\theta_L & -\sin\theta_L \\ \sin\theta_L & \cos\theta_L \end{pmatrix} \begin{bmatrix} {}^r\boldsymbol{\nu}_{FX} \\ {}^r\boldsymbol{\nu}_{FY} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\nu}_{\mathrm{LX}} \\ \boldsymbol{\nu}_{\mathrm{LY}} \end{bmatrix}. \tag{12}$$

where $\boldsymbol{\nu}_{\mathrm{LX}}$ and $\boldsymbol{\nu}_{\mathrm{LY}}$ are the components of the linear velocity of the leader robot in the global coordinate system.

The magnitude of the absolute linear velocity of the follower is then given by

$$\|v_F\| = \sqrt{v_{FX}^2 + v_{FY}^2}. \tag{13}$$

The absolute angular velocity of the followers can be calculated from the curvature $\kappa(t)$ of its path given by

$$\boldsymbol{\kappa}(t) = \frac{\boldsymbol{\nu}_{\mathrm{Fx}}\frac{d\boldsymbol{\nu}_{\mathrm{Fy}}}{dt} - \nu_{\mathrm{Fy}}\frac{d\nu_{\mathrm{Fx}}}{dt}}{\|v_F\|^3}. \tag{14}$$

With this, the instant angular velocity of the followers is

$$\omega(t) = \|v_F\|\kappa(t). \tag{15}$$

Therefore, given the desired relative linear velocity and the absolute linear and angular velocity of the leader, the desired absolute linear and angular velocities of the follower robot can be found from (14) and (15). These two velocities are then input into the follower's local PID controller and reconfiguration is accomplished.

VI. Experimental Results

A. Experiment Setup

The leader is a car-like robot evolved from [21] and is equipped with a standard Pentium laptop. The leader robot localizes the position and orientation of the followers and sends this pose information, in addition to its velocity and pose, to each follower through RF wireless communication. Each follower communicates with the leader through a unique identification number. When the follower receives the communication data, it defines a local Bezier trajectory, determines the desired linear velocity and executes the velocity controller to follow the defined trajectory. The Bezier trajectory is dynamically updated as new information is obtained from the leader.

The experiments conducted here require all follower robots to follow the leader as the leader travels along a straight line for at least 105 cm so that the first eight positions (with $d = 15$ cm) for the first estimate of the follower's orientation can be acquired. After this initial distance the cubic polynomial least-squares localization method creates a new estimate of the follower's orientation for every 15 cm of travel. During the 15-cm movement, the follower robot navigates and calculates its pose via dead reckoning. The follower also updates its pose once it receives the estimated position and orientation from the leader. This method bounds the errors in dead reckoning and allows the leader and followers to travel long distances.

B. Localization Experiment

In this experiment, the localization method (developed in Section III) is used to estimate the orientation of the follower

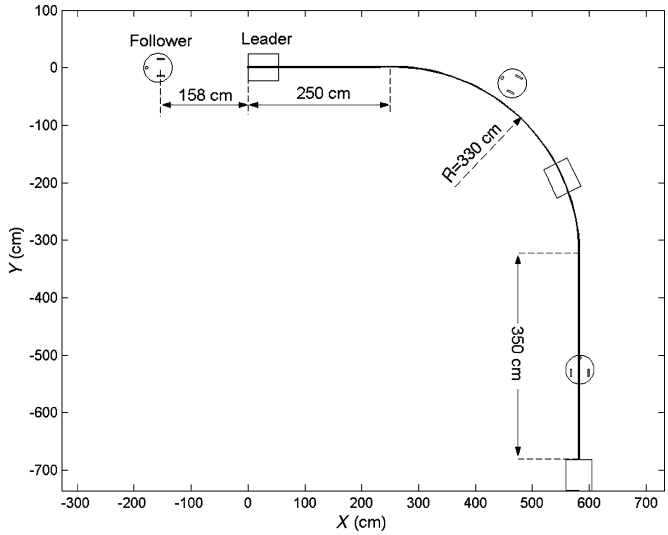


Fig. 16. Leader/follower motion during localization experiment.

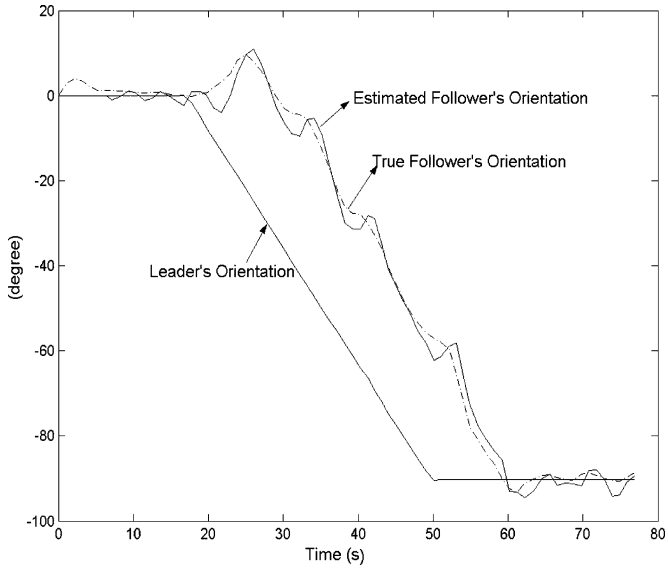


Fig. 17. Orientation of the leader and follower robot.

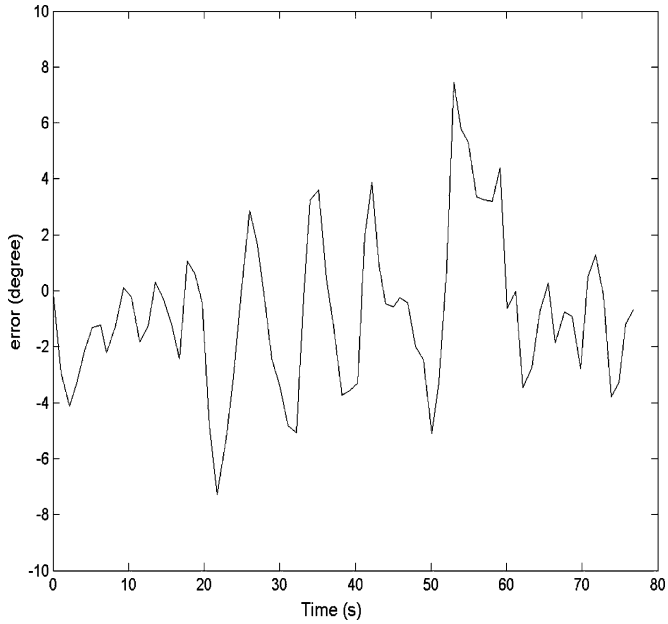


Fig. 18. Orientation error of the follower robot.

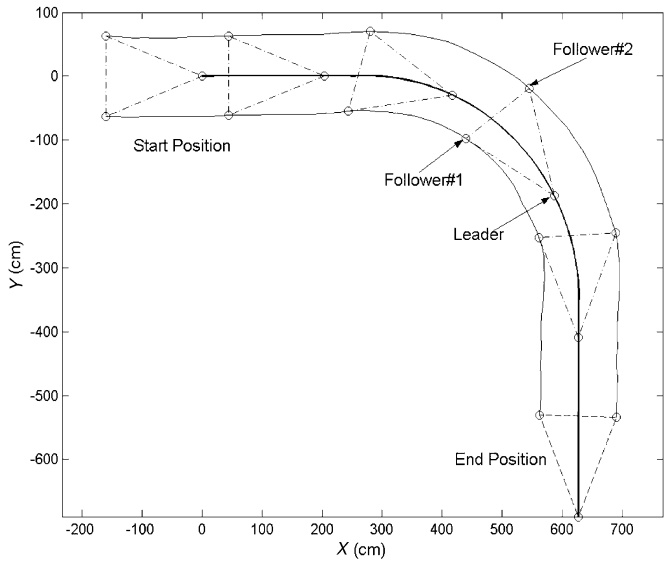


Fig. 19. Three robots maintain the triangular formation.

robot. The leader robot moves at a constant linear velocity of 15 cm/s. Fig. 16 shows the experimental setup where the leader first moves 250 cm along a line. Then it turns with a radius 330 cm. After that, it continues to move 350 cm along another line. The follower robot follows the leader at a distance of 158 cm with a bearing of 0°.

During this experiment, a special sensor is used to estimate the true orientation of the follower robot with respect to the leader. This additional sensor is independent of the sensors on either robot. The sensor is an NDI Polaris camera system [22], which is mounted on the leader robot and detects retro-reflective optical markers mounted on the followers. This commercial system (used in surgical registration) provides real-time (60 Hz maximum updating rate) 6-D tracking with a position accuracy less than 0.35 mm.

Fig. 17 shows the orientation of the leader and follower robots. From Fig. 18, it is seen that the error between the estimated and true orientation of the follower is bounded within $\pm 8^\circ$ and this error does not grow with time. The mean error during this experiment is -0.7°, with a standard deviation of 2.8°. This experimental result confirms the validity of the localization approach. It also compares favorably with the accuracy predicted by the simulation in Figs. 6 and 7 and Figs. 9 and 10. Since the error is bounded and does not grow with distance, this localization approach can be used for formation control over long distances.

C. Follow-the-Leader Experiment

Now, two followers follow the leader robot while they attempt to maintain the triangular geometric formation shown in Fig. 19

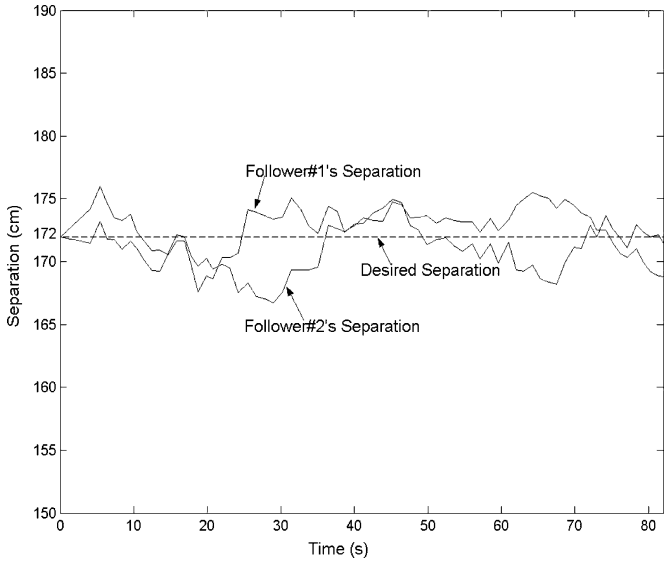


Fig. 20. Separation vs. time.

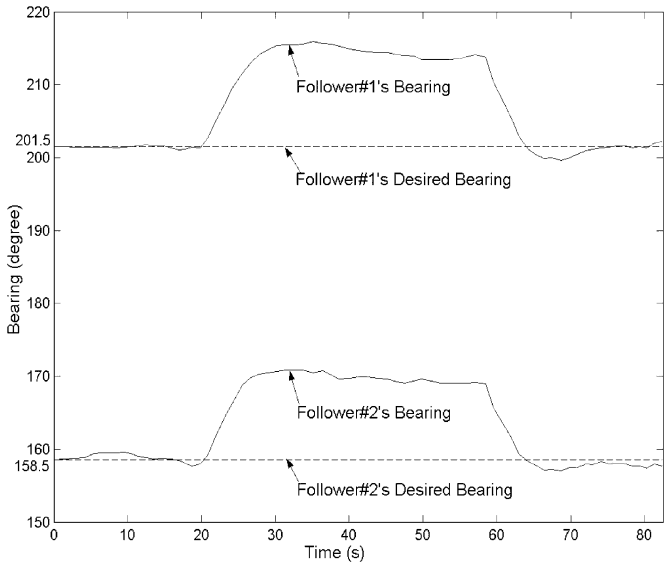


Fig. 21. Bearing versus time.

(the camera system is not used). The initial global position of the leader is (0, 0), and initial global position of two followers are $(-160, 63\,\text{cm})$ and $(-160, -63\,\text{cm})$, respectively (as given in the coordinate system in Fig. 19). Therefore, the desired separation for each follower with respect to the leader is 172 cm, and the desired bearing for the follower robot #1 and follower robot #2 is 201.5° and 158.5°, respectively. The leader moves with a constant linear velocity of 15 cm/s.

As depicted in the experimental data in Fig. 19, the two follower robots maintain the formation with respect to the leader. When the leader completes the turn, the two followers converge to their respective desired positions in the triangular formation. The separation and bearing can be easily computed from the relative localization through the laser scanner. Figs. 20 and 21 show the variance of separation and bearing of two followers over time. Both separation errors are within ± 5 cm and the bearings of two followers have small errors $(\leq 2^\circ)$, when the three robots move along a straight line. Bearing error is larger $(\sim 15^\circ)$ when the formation turns, largely due to the latency of the information transmitted to the follower with the limited communication bandwidth. This is a very acceptable error for applications such as maintaining a wedge formation for lane closure in highway safety tasks.

VII. Summary and Conclusion

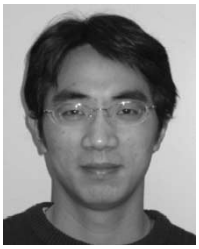
This paper presents a centralized localization and control method for follow-the-leader movement of a heterogeneous group of mobile robots. This procedure allows the follower robots to be computationally simple, have minimal sensing, be inexpensive, and therefore be suited to high-risk tasks. A special centralized localization method for such a group of extremely sensor-limited follower robots is presented. Errors in localization are bounded and therefore the method is useful for long-distance navigation. A statistical simulation of the localization method is included, which analyzes the choice of appropriate distance between two tracked positions and for the minimum allowable radius of curvature for follower motion. This analysis shows the cubic polynomial least-squares method can estimate the orientation of the follower robot with the desired accuracy. An algorithm to alter the formation is also presented. Finally, a simple formation-keeping control approach that uses Bezier trajectories is presented. Experimental results confirm the validity of both localization and follow-the-leader control methods.

The proposed method is limited in that the leader needs to generally maintain line-of-sight contact with the followers. This means a perfectly "single file" formation could not be maintained over long distances. Also, the computation requirements of the leader associated with follower localization and planning grows linearly with the number of followers.

References

[1] S. Farritor and M. Rentschler, "Robotic highway safety markers," presented at the ASME Int. Mech. Eng. Congr., New Orleans, LA, 2002.
[2] X. Shen, J. Dumpert, and S. Farritor, "Design and control of robotic highway safety markers," *IEEE/ASME Trans. Mechatron.*, vol. 10, no. 5, pp. 513–520, Oct. 2005.
[3] S. Farritor and S. Goddard, "Intelligent highway safety markers," *IEEE Intell. Syst.*, vol. 19, no. 6, pp. 8–11, Nov.–Dec. 2004.
[4] B. Khoshnevis and G. Bekey, "Centralized sensing and control of multiple mobile robots," *Comput. Ind. Eng.*, vol. 35, no. 3/4, pp. 503–506, 1998.
[5] T. Balch and R. C. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, Dec. 1998.
[6] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Auton. Robots*, vol. 4, no. 4, pp. 387–403, 1997.
[7] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 813–825, Oct. 2002.
[8] R. Vidal, O. Shakernia, and S. Sastry, "Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation," in *IEEE Int. Conf. Robot. Autom.*, vol. 1, Taipei, Taiwan, R.O.C., 2003, pp. 584–589.
[9] S. Y. Chiem and E. Cervera, "Vision-based robot formations with Bezier trajectories," in *Proc. 8th Intelligent Auton. Robots*, 2004, pp. 191–198.

[10] A. Arakawa, M. Hiyama, T. Emura, and Y. Kagami, "Trajectory generation for wheeled mobile robot based on landmarks," in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics, Intelligent Syst. 21st Century*, vol. 2, Vancouver, Canada, 1995, pp. 1273–1278.
[11] M. Khatib, H. Jaouni, R. Chatila, and J. P. Laumond, "Dynamic path modification for car-Like nonholonomic mobile robots," in *IEEE Int. Conf. Robot. Autom.*, Albuquerque, New Mexico, 1997, pp. 2920–2925.
[12] J. Hwang, R. C. Arkin, and D. Kwon, "Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, vol. 2, Las Vegas, NV, 2003, pp. 1444–1449.
[13] J. Fredslund and M. J. Mataric, "A general algorithm for robot formations using local sensing and minimal communication," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 837–846, Oct. 2002.
[14] F. Michaud, D. Letourneau, M. Guilbert, and J. Valin, "Dynamic robot formations using directional visual perception," in *IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, vol. 3, 2002, pp. 2740–2745.
[15] L. E. Parker, B. Kannan, F. Tang, and M. Bailey, "Tightly-coupled navigation assistance in heterogeneous multi-robot teams," in *Proc. IEEE Int. Conf. Intelligent Robots Syst.*, vol. 1, Sendai, Japan, 2004, pp. 1016–1022.
[16] J. Shi, S. Goddard, A. Lal, and S. Farritor, "A real-time model for the robotic highway safety marker system," *J. Real-time Syst.*, vol. 29, no. 2/3, pp. 183–204, Mar. 2005.
[17] Y. Cang and J. Borenstein, "Characterization of a 2D laser scanner for mobile robot obstacle negotiation," in *Proc. IEEE Int. Conf. Robotics Autom.*, vol. 3, Washington, DC, 2002, pp. 2512–2518.
[18] F. Andersson "Bezier and B-Spline technology," Master's thesis, Dept. Comput. Sci., UMEA Univ., UMEA, Sweden, 2003.
[19] G. G. Lorentz, *Bernstein Polynomials*. Toronto: Univ. Toronto Press, 1953.
[20] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis*. 7th ed. Reading, MA: Addison-Wesley, 2003.
[21] W. Kaliardos, M. Cosgrove, T. Chow, S. Steiner, and S. Farritor, "Companion—Overview of a telerobotic ground vehicle," presented at the Assoc. for Unmanned Vehicle Systems International (AUVSI), 1996.
[22] Polaris Technical Specifications, Northern Digital Inc., Waterloo, ON, Canada.
[23] A. Mohri, M. Yamamoto, and S. Marushima, "Collision-free trajectory planning for two manipulators using virtual coordination space," in *IEEE Int. Conf. Robotics Autom.*, 1993, pp. 674–679.



Jiangyang Huang received the B.S. degree in mechanical engineering and automation from the Xi'an Jiaotong University, Xi'an, China, in 1999, and the M.S. degree in mechatronics engineering from the Zhejiang University, Hangzhou, China, in 2002. He is currently pursuing the Ph.D. degree at the Department of Mechanical Engineering, University of Nebraska-Lincoln, Lincoln.

His current research interests include heterogeneous mobile robotic systems, cooperative localization and control, and real-time scheduling in robotic systems.

Mr. Huang is a student member of ASME.

Shane M. Farritor received the B.S. degree from the University of Nebraska-Lincoln, Lincoln, in 1992, and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1994 and 1998, respectively, all in mechanical engineering.

He is currently an Associate Professor in the Department of Mechanical Engineering, University of Nebraska-Lincoln, Lincoln, and holds courtesy appointments in both the Department of Surgery and the Department of Orthopaedic Surgery at the University of Nebraska Medical Center, Omaha, NE. His research interests include space robotics, surgical robotics, biomedical sensors, and robotics for highway safety.

Dr. Farritor is the Chairman of the AIAA Space Robotics and Automation technical committee and serves on the ASME Dynamic Systems and Control Robotics Panel.

Ala' Qadi (M'05) received the B.S. degree in electrical engineering from the An-Najah National University, Nablus, Palestine, and the M.S. degree in computer engineering from the the University of Nebraska-Lincoln, Lincoln, in 2000 and 2003, respectively. He is currently pursuing the Ph.D. degree in computer engineering with a minor in mechanical engineering, University of Nebraska-Lincoln, Lincoln.

His research interests include real-time systems, robotics and control systems.

Steve Goddard (M'98) received the B.A. degree in computer science and mathematics from the University of Minnesota, St. Paul, in 1985, and the M.S. and Ph.D. degrees from the University of North Carolina, Chapel Hill, in 1995 and 1998, respectively, both in computer science.

He is a J.D. Edwards Associate Professor and College of Engineering Distinguished Scholar in the Department of Computer Science and Engineering, University of Nebraska–Lincoln, Lincoln. His research interests are embedded, real-time and distributed systems with emphasis in high assurance systems engineering and real-time, rate-based scheduling theory.