SAND99-0783C

Proceedings of SPIE Vol. 3693, Unmanned Ground Vehicle Technology, Orlando, April 7-8, 1999.

Control of Multiple Robotic Sentry Vehicles

John Feddema, Chris Lewis, and Paul Klarer
Intelligent Systems and Robotics Center
Sandia National Laboratories*
Albuquerque, NM 87185-1003

ABSTRACT

As part of a project for the Defense Advanced Research Projects Agency, Sandia National Laboratories is developing and testing the feasibility of using of a cooperative team of robotic sentry vehicles to guard a perimeter and to perform surround and diversion tasks. This paper describes on-going activities in the development of these robotic sentry vehicles. To date, we have developed a robotic perimeter detection system which consists of eight "Roving All Terrain Lunar Explorer Rover" (RATLER™) vehicles, a laptop-based base-station, and several Miniature Intrusion Detection Sensors (MIDS). A radio frequency receiver on each of the RATLER vehicles alerts the sentry vehicles of alarms from the hidden MIDS. When an alarm is received, each vehicle decides whether it should investigate the alarm based on the proximity of itself and the other vehicles to the alarm. As one vehicle attends an alarm, the other vehicles adjust their position around the perimeter to better prepare for another alarm. We have also demonstrated the ability to drive multiple vehicles in formation via tele-operation or by waypoint GPS navigation. This is currently being extended to include mission planning capabilities. At the base-station, the operator can draw on an aerial map the goal regions to be surrounded and the repulsive regions to be avoided. A potential field path planner automatically generates a path from the vehicles' current position to the goal regions while avoiding the repulsive regions and the other vehicles. This path is previewed to the operator before the regions are downloaded to the vehicles. The same potential field path planner resides on the vehicle, except additional repulsive forces from on-board proximity sensors guide the vehicle away from unplanned obstacles.

Keywords: Mobile Robots, Cooperative Control, Distributed Autonomous Systems

1. INTRODUCTION

The field of mobile robotics is quite advanced. The ability to build robotic vehicles that can navigate over long distances either using tele-operation or autonomous control has been demonstrated by a number of researchers, see for instance [1]. In recent years, this field has expanded to consider large numbers or squads of vehicles [2]. The underlying goal of multi-vehicle systems is expanded capability through cooperation. Methods for controlling groups of vehicles range from distributed autonomy [3] to intelligent squad control and general purpose cooperative mission planning [4]. The types of tasks under study range from moving large objects [5] to troop hunting behaviors [6]. Conceptually, large groups of mobile vehicles outfitted with sensors should be able to automatically perform military tasks like formation following, localization of chemical sources, de-mining, perimeter control, surveillance, and search and rescue missions [7-10]. In simulation, it has been shown that by sharing concurrent sensory information the group can better estimate the shape of a chemical plume and therefore localize its source [11]. Similarly, for a search and rescue operation a moving target is more easily found using an organized team [12-13]. Simulation has also shown that enhanced perimeter control can be achieved by dispersing the group uniformly and by communicating when possible intrusions occur.

As a proof-of-concept, Sandia National Labs is developing a squad of semi-autonomous all terrain vehicles for remote cooperative sensing applications. This system serves as a test platform to verify communications, control, and sensing strategies. The system is being used to demonstrate the viability of using a cooperative team of robotic sentry vehicles to investi-

* Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000. This research is partially funded by the Tactical Technology Office of the Defense Advanced Research Projects Agency under contract e169/09.

gate alarms from intrusion detection sensors, and surround and monitor an enemy facility. This paper will first describe the system and the basic control modes that have been implemented. Then it will describe how the system performs perimeter patrol and surround tasks.

2. SYSTEM DESCRIPTION

Eight RATLER™ vehicles have been built at Sandia as a test platform for cooperative control and sensing applications. The RATLER design originated from a lunar exploration mission. These electric, all wheel drive vehicles sport two composite bodies joined by a passive central pivot. This flexible structure when combined with an aggressive asymmetric tread on custom carbon composite wheels provides agile off road capabilities. With a PC104 Intel 80486, the RATLER vehicles are fully equipped with a wide range of sensors and peripherals. Software on the vehicles is currently a single-threaded DOS-based application for simplicity. The vehicles have been programmed to operate either through tele-operation or autonomously. Bristling with antennae, the RATLER vehicles rely heavily on Radio Frequency (RF) signals for communications. Currently, the vehicles are outfitted with differential GPS sensors, and two spread spectrum RF modems. One modem is for inter-vehicle and base-to-vehicle communication and the other is for the differential GPS signal. Video cameras communicate to the base-station via a separate RF video link.

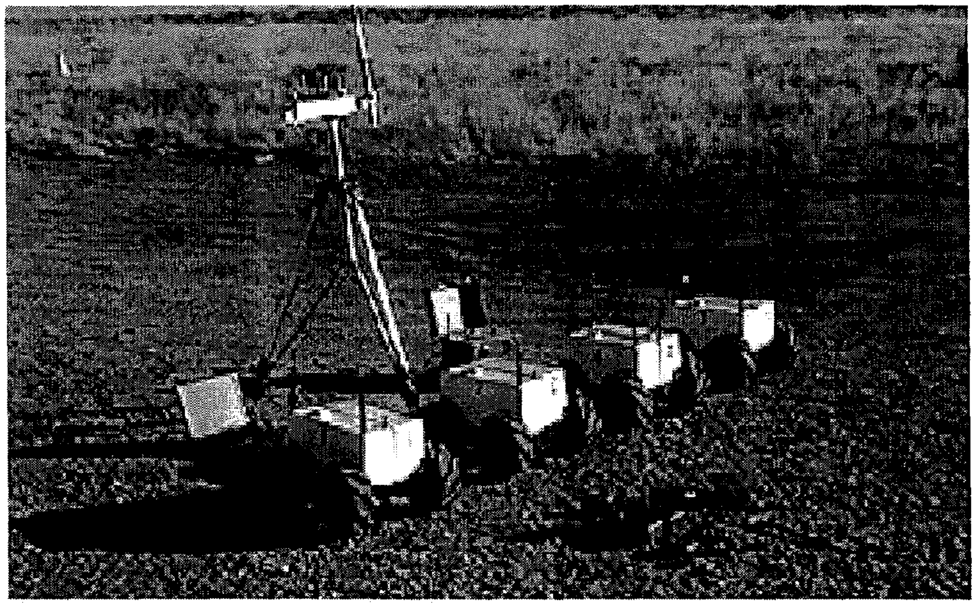
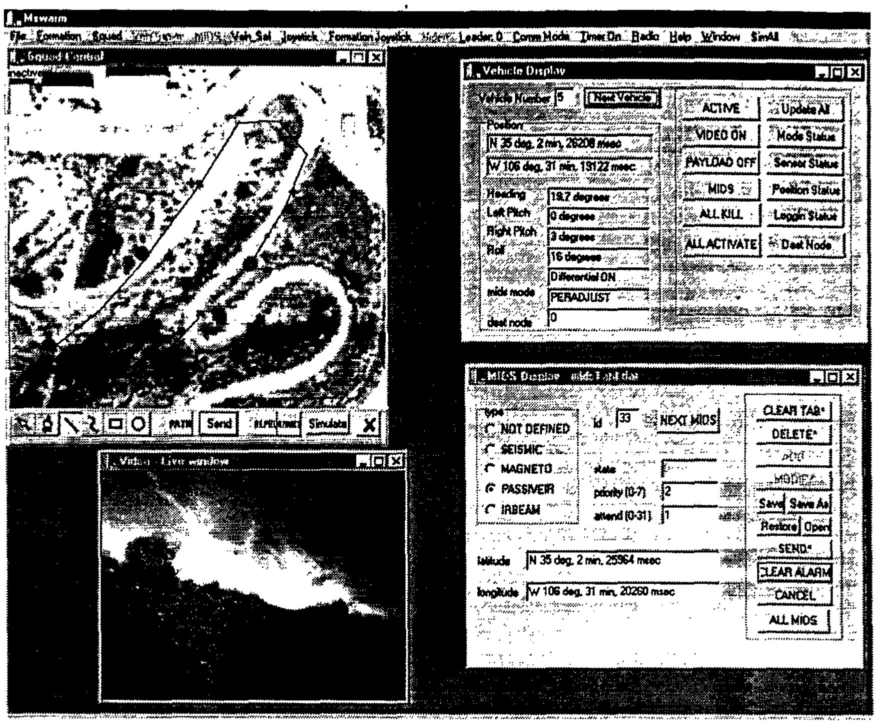


Figure 1. Perimeter detection system: 4 of the 8 RATLER vehicles, MIDS, and base-station.

A laptop computer is used as the base-station. A Windows NT application was written to control the vehicles from the base-station. A Graphical User Interface (GUI) displays vehicle status information and allows the operator to monitor the vehicles positions on a Geographic Information System (GIS) map – either aerial photo or topological data, as well as view the live video from a selected vehicle (see Figure 2). Mission specific control modes such as tele-operation, formation following, autonomous navigation, and perimeter detection can be initiated and monitored using this GUI interface.



Vehicle Number 5

Position
N 35 deg. 2 min, 26208 msec
W 106 deg. 31 min. 19122 msec

Heading 19.7 degrees
Left Pitch 0 degrees
Right Pitch 3 degrees
Roll 16 degrees
Differential ON
mids mode PERADJUST
dest node 0

type
NOT DEFINED
SEISMIC
MAGNETO
[✓ PASSIVEIR]
IRBEAM

id 33
state
priority (0-7) 2
attend (0-31) 1
latitude N 35 deg. 2 min. 25964 msec
longitude W 106 deg. 31 min. 20260 msec

Figure 2. The base-station's graphical user interface displays vehicle status, MIDS status, video, and GPS position on a GIS map.

3. COMMUNICATION

In the initial implementation of the RATLER squad, all radio communication was coordinated by the base-station. Recently, token ring communications has been added as an alternative to the star network topology centered on the base-station (see Figure 3). The token ring network is more fault tolerant than the star network since there is no single point of failure as there is with the star network. Also, the token ring network allows the vehicles to continue to operate in perimeter detection mode even if the base station is shutdown. In a token ring network, each node (either vehicle or base station) speaks only when it receives the token. In our case, an actual token packet was not needed since each vehicle has an id number, and communication order is determined from this id number. All messages are broadcast in half-duplex mode so that each vehicle knows when the other vehicles or the base-station has transmitted a message. If a node does not communicate when expected, a timer on the next node expires, signaling that the next node should transmit.

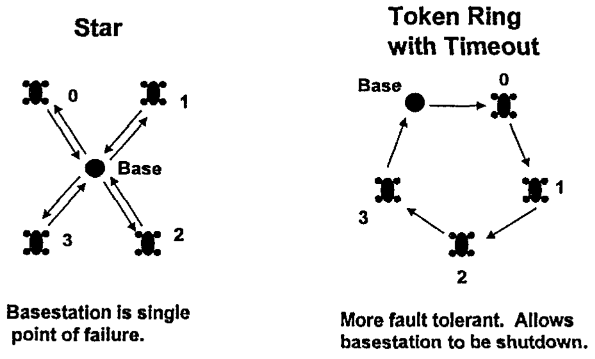

Star
0
1
Base
3
2
Basestation is single point of failure.
Token Ring with Timeout
Base
0
1
2
3
More fault tolerant. Allows basestation to be shutdown.

Figure 3. Two communication options are available. The star network requires the base station to coordinate all communication. The token ring network provides a decentralized solution.

3. TELE-OPERATION

Tele-operation is useful for navigating complex terrain, and for investigating the environment manually. In tele-operation the vehicle is controlled with a standard joystick attached to the laptop's MIDI port. The joystick's trigger button acts as a deadman's switch, while the other buttons are mapped to functions like focus and throttle. The operator has direct control over the right and left wheel speed and direction of rotation. The video signal from any one of the vehicles can be displayed within the multi-document interface to aid in the control. Tele-operation of one vehicle does not necessarily effect the behavior of the other vehicles. However, a squad can be maneuvered by specifying a formation relative to the vehicle being tele-operated.

4. AUTONOMOUS CONTROL

The ability to automatically navigate from one location to another is a fundamental capability of the squad system. In autonomous mode, the vehicles rely on their GPS, compass heading, and dual tilt sensors. A simple control strategy monitors these sensors as it steers toward desired world coordinate locations and avoids obstacles along the way. A skid-driven vehicle model is used for autonomous control. In this model, the vehicle's linear and angular velocity are related to the right and left wheel velocities as follows:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \frac{1}{2}\begin{bmatrix} r\cos(\theta) & r\cos(\theta) \\ r\sin(\theta) & r\sin(\theta) \\ \frac{r}{l} & -\frac{r}{l} \end{bmatrix}\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = B\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \tag{1}$$

where ***l*** is the distance from the center of the vehicle to the wheel, r is the wheel radius, and θ is the compass direction. The differences between desired and actual location and orientation are used as feedback as follows:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} x_d - x \\ y_d - y \\ k_1(\theta - a\tan 2(y_d - y, x_d - x)) + k_2(\theta - \theta_d) \end{bmatrix} \tag{2}$$

where k_1 and k_2 are gains. The wheel velocity commands are determined by

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = B^+\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}. \tag{3}$$

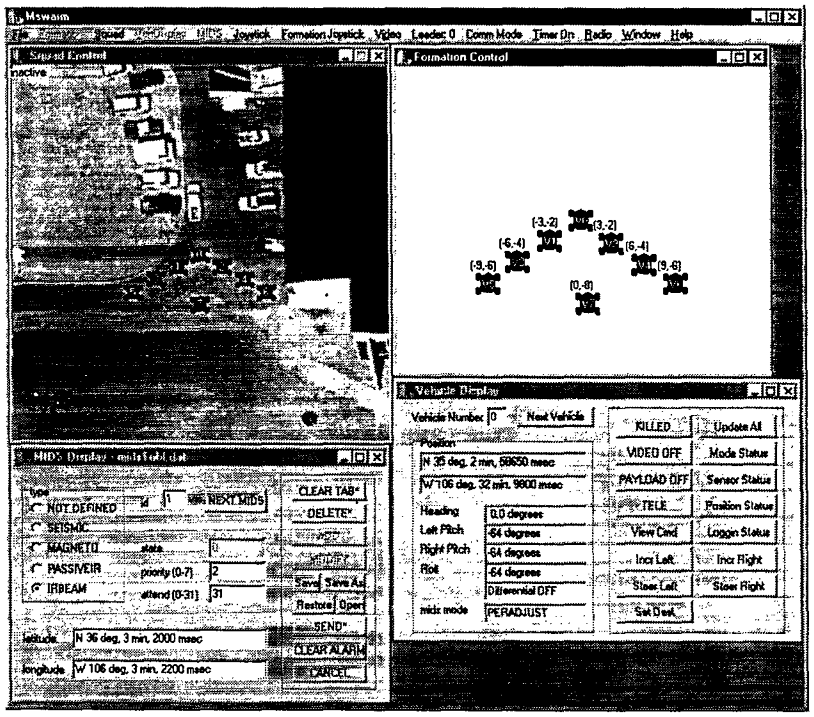
where B^+ is the pseudo-inverse of the system matrix B. For relatively large values of k_1, the vehicle will turn first and proceed toward the goal. Smaller values produce an arching trajectory depending on the initial conditions. The value of k_2 is kept small so that its only effect is to orient the vehicle once it achieves the desired position.

Because the vehicles move relatively slowly and have very high torque capabilities, obstacle avoidance is achieved using the tilt sensors. When the vehicle runs into an obstacle such as a bush the front wheels will climb until the tilt sensor reaches a cutoff value. At this point the vehicle will reverse direction, turn, move forward and turn again. At this point automatic control is re-started. This simple strategy has proven effective in avoiding most of the obstacles in a desert environment.

The basic ability to navigate to GPS locations is vital to the capability of the squad system. Currently, the operator may graphically select goal locations and paths on a GIS map for the vehicles. In the future, search strategies will be automatically generated and executed.

5. FORMATION

One of the goals of this project is to develop a simple user interface that allows a single soldier to guide multiple robot vehicles. Formation following is one way to accomplish this task. The ability to maintain a formation is useful for conducting searches and for moving the squad from place to place. This capability has been implemented using the base-station's GUI and utilizing the autonomous navigation capability described earlier. To initiate formation control the operator selects a lead vehicle, and then graphically places the other vehicles relative to the leader as shown in Figure 4. When the formation is initiated, the vehicles are automatically commanded to autonomously navigate to their respective locations surrounding the leader. As the leader moves either by autonomous navigation or using tele-operation control the other vehicles maintain the formation. In the current implementation, orientation is not considered so that the vehicles always traverse nominally the same distance as the formation moves along. A formation always remains aligned to the compass frame rather than to the lead vehicle's frame. In the future, we will be implementing a formation control mode in the lead vehicle's frame. In this case, the lead vehicles turning rate will have to be limited to account for the greater distances traveled by vehicles that are farther away. A number of control strategies are being investigated for actively maintaining tight formations regardless of the differences in the terrain and path lengths traversed by each vehicle participating in the formation.



Vehicle Number: 0

Position:
N 35 deg. 2 min. 59650 msec
W 106 deg. 32 min. 9800 msec

Heading: 0.0 degrees
Left Pitch: -64 degrees
Right Pitch: -64 degrees
Roll: -64 degrees
Differential OFF
mids mode: PERADJUST

type: NOT DEFINED / SEISMIC / MAGNETO / PASSIVEIR / [✓ IRBEAM]
id: 1
state: 0
priority (0-7): 2
attend (0-31): 31
latitude: N 36 deg. 3 min. 2000 msec
longitude: W 106 deg. 3 min. 2200 msec

Figure 4. Formation control windows. Also shown are vehicle and MIDS status windows.

6. POTENTIAL FIELD PATH PLANNER

In addition to formation following, we are developing an interactive playbook capability where the operator first draws the goal regions and obstacles on a map, and then vehicle paths are automatically generated and previewed to the operator. In Figure 5, the operator has used a drawing tool bar to outline the obstacles and indicate goal regions to attract the vehicles. A simple attractive and repelling potential field algorithm is used to generate the desired paths for the vehicles to take. The algorithm uses the distance and direction to the nearest goal, obstacle, and neighboring vehicle to determine the gradient used to update the vehicle's position as each vehicle moves from its initial position to the closest goal.

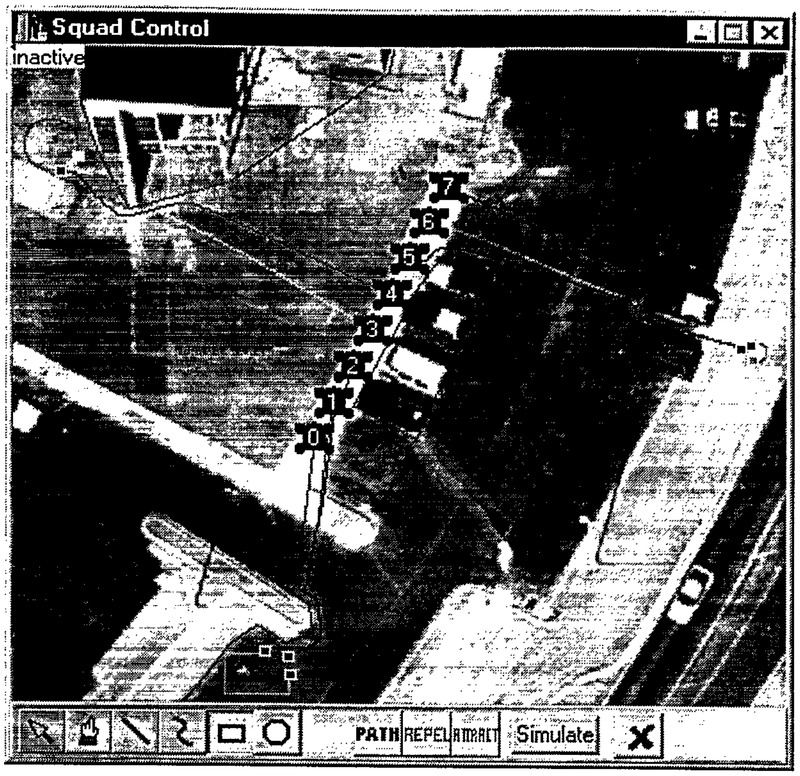


Figure 5. Base-station control window. The initial positions of the vehicles are the numbered circles in the middle of the screen. The final positions of the vehicles are spread out along the square at the bottom and the circles on the left and right sides. Although not visible here, the obstacles are drawn in red, the goals are drawn in green, and the vehicle paths are drawn in other colors.

The distance to the closest goal and obstacle is computed as described in the algorithm below. The polygons and line segments drawn by the operator are stored in a linked list. The algorithm assumes that the robot is at the origin, and (x_1, y_1) and (x_2, y_2) are the end points of the line segments with respect to the robot.

```
$x_{\min} = x_1;\ y_{\min} = y_1;$          /* Initialize to first point relative to the robot's position*/
$d_{\min}^2 = x_1^2 + y_1^2;$               /* Initialize minimum squared distance */

For each line segment in the linked list

    If $|x_2 - x_1| \neq 0$ then          /* Not a vertical line */
        $a = \frac{y_1 - y_2}{x_2 - x_1};\ b = 1;\ c = -ax_2 - y_2;$    /* Equation of a line ax+by+c=0 */
        $k = 1 + a^2;\ d_\perp^2 = \frac{c^2}{k};$                       /* Perpendicular distance to origin */
        $c_{t1} = x_1 - ay_1;\ c_{t2} = x_2 - ay_2;$                     /* y-axis intercept of tangent lines at end points*/
    Else
        $a = 1;\ b = 0;\ c = -x_2;$                                      /* Equation of a line ax+by+c=0 */
        $k = 1;\ d_\perp^2 = c^2;$                                       /* Perpendicular distance to origin */
        $c_{t1} = y_1;\ c_{t2} = y_2;$                                   /* y-axis intercept of tangent lines at end points */
        End

        If ( $c_{t1} < c_{t2}$ and $c_{t1} < 0$ and $c_{t2} > 0$ ) or ( $c_{t1} > c_{t2}$ and $c_{t1} > 0$ and $c_{t2} < 0$ ) then
            /* If robot location (at origin) is between the y intercepts of the tangent lines */
            If $d_{\perp}^2 < d_{\min}^2$ then     /* If perpendicular distance is smaller than previous line segments */
                $d_{\min}^2 = d_{\perp}^2$;
                $x_{\min} = \frac{-ac}{k}$;  $y_{\min} = \frac{-bc}{k}$
            End
        Else  /* Check distance to second point */
            $d_2^2 = x_2^2 + y_2^2$;
            If $d_2^2 < d_{\min}^2$ then     /* If squared distance to second point is smaller than previous */
                $d_{\min}^2 = d_2^2$;
                $x_{\min} = x_2$;  $y_{\min} = y_2$;
            End
        End
    End
```

After the closest obstacle, goal, and vehicle positions are computed, the direction of the vehicle is given by

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = K_a \begin{bmatrix} x_a - x_v \\ y_a - y_v \end{bmatrix} - K_r \begin{bmatrix} x_r - x_v \\ y_r - y_v \end{bmatrix} - K_{v2} \begin{bmatrix} x_{cv} - x_v \\ y_{cv} - y_v \end{bmatrix} \tag{4}$$

where (x_v, y_v) is the vehicle's position, (x_a, y_a) is the closest attractive point (goal), (x_r, y_r) is the closest repulsive point (obstacle), (x_{cv}, y_{cv}) is the closest vehicle, and K_a, K_r, and K_{cv} are positive gains.

Notice in Figure 5 that the vehicles split up to go to three different goals. Also notice that the vehicles spread out along the goal polygon because of the third term in Equation (4). By simulating the vehicle's response, the operator can predict the paths of the vehicles before initiating the motion. Once the operator has validated the path, the polygons and line segments, which represent the obstacles and goals, are downloaded to the vehicles, instead of the actual paths. On board the vehicles, the same potential field algorithm generates the path except that it uses the true location of itself and the nearest neighboring vehicle.

7. PERIMETER DETECTION SYSTEM

The squad of vehicles is also being tested for a perimeter detection mission. The objective is to demonstrate the viability of using a cooperative team of robotic sentry vehicles to investigate alarms from intrusion detection sensors. To demonstrate this capability, additional hardware and software were added to the squad system. A variety of remote intrusion sensors, hidden on a defensive perimeter, broadcast unique identification codes to all the vehicles when tripped. These miniature intrusion detection sensors (MIDS) were developed by Sandia, and are now commercially available [14]. There are four different types available including: magnetometer, seismic, passive infrared and beam break (or active) infrared.

The vehicles were outfitted with receivers to detect when the sensors are tripped. The vehicles were also programmed to maintain an internal representation of the location of the MIDS sensors and the other vehicles. Additional code was also added to the base-station to enter and display the MIDS information.

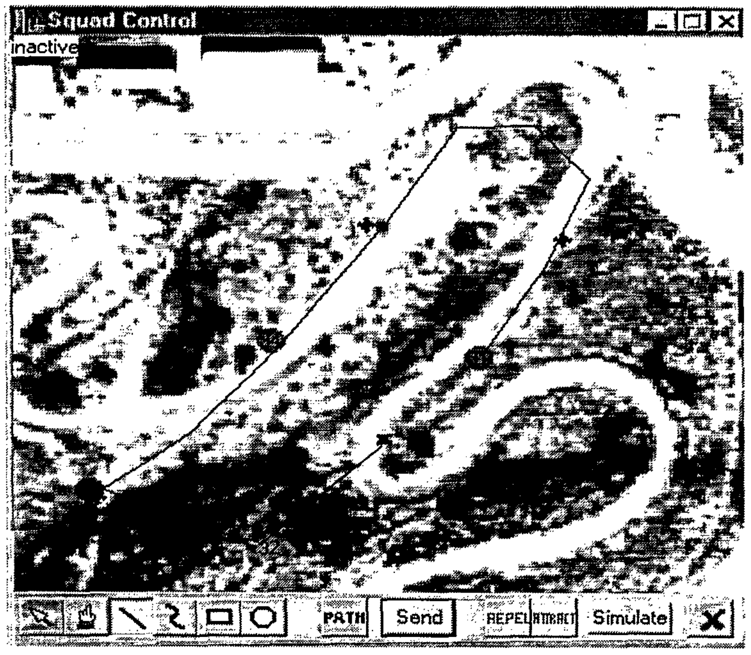


Figure 6. Perimeter being guarded by robot sentries.

As the sensors are hidden, the operator enters the MIDS attributes including:

1. the type of sensor
2. the GPS location of the sensor
3. the number of RATLERs to attend the alarm
4. the priority of the alarm.

The operator next draws a perimeter on the GIS map as shown in Figure 6. The MIDS information and the perimeter region are broadcast from the base-station to all the vehicles. Once the operator places the vehicles in the MIDS sentry mode, the vehicles spread out uniformly along the perimeter maintaining equal distance between their two nearest neighbors. When a sensor is alarmed, the vehicles decide, without base-station intervention, which of the vehicles can best investigate the intrusion, and how the remaining vehicles should adapt to maintain the perimeter.

To maintain the perimeter, the vehicles periodically broadcast their location and the status of the sensors. In this way, each vehicle can maintain a local representation of where the other vehicles are and which sensors have been tripped. When a vehicle receives an alarm signal, it broadcasts the new information. If one vehicle receives an alarm and the others don't, the other vehicles will receive the alarm through the broadcast. The base-station displays the location of the vehicles and the MIDS sensors on a GIS map. When a MIDS sensor is alarmed, the icon of the MIDS sensor changes color. The status display indicates which of the vehicles are moving to attend the alarm.

The determination of which vehicles attend an alarm is made independent of the base-station. When an alarm is received each vehicle computes it's distance to the alarmed sensor as well as the distance of the other vehicles to the same sensor. If the vehicle is closest to the alarmed sensor within the number of vehicles that are to attend the sensor, then it will head toward the alarm. That is unless a MIDS of higher priority is alarmed, in which case it heads towards the MIDS of higher priority. All of these decisions occur several times per second; therefore, a vehicle may be heading towards one alarmed MIDS, when a higher priority MIDS is alarmed, causing it to change directions. When a vehicle is not attending to an alarm, it tries to maintain an equi-distant position around the perimeter from the other unalarmed vehicles.

8. ANALYSIS OF COOPERATION

In this section, we formulate the perimeter detection problem mathematically. Consider a perimeter detection problem in which four vehicles must guard a military asset (see Figure 7). The position of the vehicles around the perimeter is represented by a single variable, θ. Around the perimeter are intrusion detection sensors. When a sensor is alarmed, the nearest vehicle to the sensor attends the alarm, while the others adjust their position in preparation for the next alarm. One option is to send all sensor data to a base station and have the base station tell each vehicle its position around the circle. Unfortunately, if the base station is destroyed, the vehicles become inoperable. A better solution is to embed a distributed sensing and control methodology in the vehicles which allows the vehicles to collectively decide which vehicles should attend alarms and which vehicles should adjust around the perimeter.

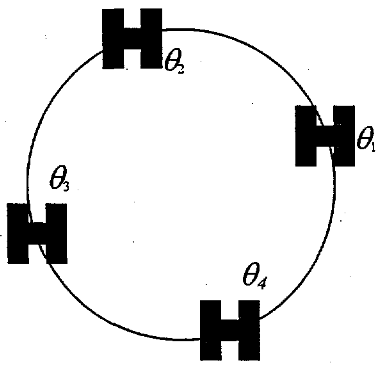


Figure 7. Perimeter detection problem.

Our embedded algorithm makes the vehicles that are adjusting their position move half way between their adjacent neighbors. Consider Figure 7, where vehicles 1 and 4 are attending to an alarm, while vehicles 2 and 3 adjust their position. The vehicles communicate with each other via a RF token ring network. Each vehicle transmits in turn: first 1, then 2, 3, 4, and then 1 again. Since each transmitted message is broadcast, each vehicle receives the other vehicles' messages. In the case of vehicle 2, it is going to use the positions of vehicles 1 and 3 to update its position. Similarly, vehicle 3 is going to use the positions of vehicles 2 and 4 to update its position. The state space equations which represent the calculated position of the vehicles is

$$\begin{bmatrix}\theta_1(k)\\ \theta_2(k+1)\\ \theta_3(k+2)\\ \theta_4(k+3)\end{bmatrix}=\begin{bmatrix}0&0&0&0\\0.5&0&0&0\\0&0.5&0&0\\0&0&0&0\end{bmatrix}\begin{bmatrix}\theta_1(k)\\ \theta_2(k+1)\\ \theta_3(k+2)\\ \theta_4(k+3)\end{bmatrix}+\begin{bmatrix}0&0&0&0\\0&0&0.5&0\\0&0&0&0.5\\0&0&0&0\end{bmatrix}\begin{bmatrix}\theta_1(k-4)\\ \theta_2(k-3)\\ \theta_3(k-2)\\ \theta_4(k-1)\end{bmatrix}+\begin{bmatrix}\theta_1(0)\\0\\0\\ \theta_4(0)\end{bmatrix} \tag{5}$$

where k represents each discrete communication time, and θ_i is the position of vehicle i. Notice that before vehicles 2 and 3 transmit their desired position, they update it by calculating the average value of the position of the adjacent vehicles. Also, notice that when k increments, it increments by the total number of vehicles, in this case, 4. Since vehicles 1 and 4 are stationary, the state equations for vehicles 2 and 3 reduce to

$$\begin{aligned}\theta_2(k') &= 0.5\theta_3(k'-1)+0.5\theta_1(0)\\ \theta_3(k') &= 0.25\theta_3(k'-1)+0.25\theta_1(0)+0.5\theta_4(0)\end{aligned} \tag{6}$$

where k' represents the discrete update after a complete token ring communication cycle. Here, k' increments by one each time through the token ring cycle. If $\theta_1(0)=0$ and $\theta_4(0)=1.5\pi$, then the position of vehicles 2 and 3 are governed by

$$\begin{aligned}\theta_2(k') &= 0.5\theta_3(k'-1)\\ \theta_3(k') &= 0.25\theta_3(k'-1)+0.75\pi\end{aligned} \tag{7}$$

In the steady state, $\theta_3(k') = \theta_3(k'-1) = 0.5\pi$ and $\theta_2(k') = \pi$. Therefore, vehicles 2 and 3 are evenly spaced between vehicles 1 and 4.

There are other approaches to this problem. One might wonder why vehicle 2 does not determine its position relative to vehicle 1, and vehicle 3 does not determine its position relative to vehicle 2. There are several reasons. First, this algorithm provides equal spacing between the vehicles regardless of how many vehicles are present. Second, because of the algorithm's parallel nature, the positioning is less susceptible to accumulated positioning error than a serial algorithm. Third, a parallel algorithm is more fault tolerant because it relies on data from two or more agents rather than a single agent.

Even more interesting is the inverse problem: determine the interaction coefficients (currently 0.5) which place the vehicles at the desired locations $\theta_3(k') = \theta_3(k'-1) = \theta_3^d$ and $\theta_2(k') = \theta_2^d$. The state equations become

$$\begin{aligned}\theta_2(k') &= a_{23}\theta_3(k'-1) + a_{21}\theta_1(0)\\ \theta_3(k') &= a_{32}a_{23}\theta_3(k'-1) + a_{32}a_{21}\theta_1(0) + a_{34}\theta_4(0)\end{aligned} \tag{8}$$

where the terms a_{ij} are the interaction coefficients used to determine the desired position of vehicle *i* based on the position of vehicle *j*. There are six parameters ($\theta_1(0)$, $\theta_4(0)$, a_{23}, a_{21}, a_{32}, a_{34}) which can be adjusted to arrive at the desired position of vehicles 2 and 3. However, there are only 2 equations. If we assume that $\theta_1(0)$ and $\theta_4(0)$ are known, $a_{23} = a_{21}$, and $a_{32} = a_{34}$, then we can uniquely determine the coefficients as

$$a_{23} = \frac{\theta_2^d}{\theta_1(0) + \theta_3^d} \qquad \text{and} \qquad a_{32} = \frac{\theta_3^d}{\theta_4(0) + \theta_2^d} \tag{9}$$

The importance of this result is that we can determine the interaction necessary between agents to achieve the desired position. This problem could be extended to a two- or three-dimensional problem where you would like to determine the interactions between agents such that they form an equally spaced mesh or grid.

In general, the state equations for *N* cooperative robot agents will be given by

$$\begin{bmatrix} x_1(k') \\ x_2(k') \\ \vdots \\ x_N(k') \end{bmatrix} = A_1 \begin{bmatrix} x_1(k'-1) \\ x_2(k'-1) \\ \vdots \\ x_N(k'-1) \end{bmatrix} + \ldots + A_N \begin{bmatrix} x_1(k'-N) \\ x_2(k'-N) \\ \vdots \\ x_N(k'-N) \end{bmatrix} + \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_N(0) \end{bmatrix} \tag{10}$$

where $x_i \in \Re^M$ is the state information of vehicle *i*, and the A_j's depend on the communication connectivity and the protocols (e.g. token ring). Assuming that the initial conditions $(x_1(0), x_2(0), \ldots, x_N(0))$ and the final conditions $(x_1(k_f'), x_2(k_f'), \ldots, x_N(k_f'))$ are known, and we want to determine the connectivity matrices that drive the system to the desired outputs; then we need to solve

$$(I - A_1 - \ldots - A_N) \begin{bmatrix} x_1(k_f') \\ x_2(k_f') \\ \vdots \\ x_N(k_f') \end{bmatrix} = \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_N(0) \end{bmatrix} \tag{11}$$

for the parameters in A_j. There are $N \times M$ equations and $N \times (N \times M)^2$ unknowns if the interaction matrices A_j's are fully populated matrices. Fortunately, as seen in the previous example, the A_j's are usually very sparse matrices because of the communication protocols. In the future, this type of analysis will be used to perform more complex cooperative missions such as surveillance and searching.

9. CONCLUSION

This paper described a squad of mobile robotic vehicles being developed at Sandia National Laboratories to demonstrate cooperative robotic sensing capabilities. The squad currently consists of eight RATLER™ style vehicles, a laptop base-station, and a battery powered antenna array. The system has been used to demonstrate autonomous navigation of a cooperative team of robots and their use for surround and perimeter detection missions. For the perimeter detection mission, MIDS receivers were added to the robots. The vehicles and the base-station were programmed to maintain a perimeter. Because the vehicles share information about the location and status of the sensors and the other vehicles, they are able to effectively allocate vehicle resources to investigate intrusions. Using the base-station, the operator may monitor the system's operation, and if necessary assume tele-operation control of a vehicle already at the scene should an intrusion need further investigation. This perimeter monitoring task demonstrates some of the cooperative robotic sensing capabilities of Sandia's squad of mobile robots.

REFERENCES

1. D Bapna, E et .al., "The Atracama Desert Trek: Outcomes," *Proceedings of the 1998 Conference on Robotics & Automation,* Leuven, Belgium, May 1998, 597-604.

2. Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng, "Cooperative Mobile Robotics: Antecedents and Directions," *Proceedings of the 1995 IEEE/RSJ IROS Conference,* 226-234.

3. T. Fukuda, et al , "Evaluation on Flexibility of Swarm Intelligent System, "*Proceedings of the 1998 Conference on Robotics & Automation*, Leuven, Belgium, May 1998, 3210-3215.

4. B. Brummitt and Anthony Stentz, "GRAMMPS: A Generalized Mission Planner for Multiple Mobile Robots In Unstructured Environments," *Proceedings of the 1998 Conference on Robotics & Automatio*n, Leuven, Belgium, May 1998, 1564-1571.

5. K. Kosuge, T. Oosumi, M. Satou, K. Chiba, K. Takeo, "Transporation of a Single Object by Two Decentralized-Controlled Nonholonomic Mobile Robots," *Proceedings of the 1998 Conference on Robotics & Automatio*n, Leuven, Belgium, May 1998, 2989-2994.

6. Hiroaki Yamaguchi, "A Cooperative Hunting Behavior by Mobile Robot Troops," *Proceedings of the 1998 Conference on Robotics & Automation,* Leuven, Belgium, May 1998, 3204-3209.

7. F.R. Noreils, "Multi-Robot Coordination for Battlefield Strategies," *Proceedings of the 1992 IEEE.RSJ International Conference on Intelligent Robots and Systems,* Raleigh, NC, July 1992, 1777-1784.

8. Q. Chen, J.Y.S. Luh, "Coordination and Control of a Group of Small Mobile Robots," *Proceedings of the 1994 IEEE.RSJ International Conference on Intelligent Robots and Systems,* 2315-2320.

9. J.P. Desai, J. Ostrowski, V. Kumar, "Controlling Formations of Multiple Mobile Robots," *Proceedings of the 1998 Conference on Robotics & Automatio*n, Leuven, Belgium, May 1998, 2864-2869.

10. H. Yamaguchi, J.W. Burdick, "Asymptotic Stabilization of Multiple Nonholonomic Mobile Robots Forming Group Formations," *Proceedings of the 1998 Conference on Robotics & Automation*, Leuven, Belgium, May 1998, 3573-3580.

11. J.E. Hurtado, R.D. Robinett, C.R. Dohrmann, S.Y. Goldsmith, "Distributed Sensing and Cooperating Control for Swarms of Robotic Vehicles," *Proc. IASTED Conference Control & Applications*, Honolulu, Hawaii, Aug. 12-14, 1998.

12. J.S. Jennings, G. Whelan, W.F. Evans, "Cooperative Search and Rescue with a Team of Mobile Robotis," *Proc. IEEE International Conference of Advanced Robotics*, Monterey, CA, 1997.

13. S. Goldsmith, J. Feddema, R. Robinett, "Analysis of Decentralized Variable Structure Control for Collective Search by Mobile Robots," SPIE98, *Proc. Sensor Fusion and Decentralized Control in Robotic Systems*, Boston, November 1-6, 1998.

14. Qual-Tron, Inc., 4339 South 93rd East Ave, Tulsa, OK 74145, 918-622-7052.

15. C. Lewis, J.T. Feddema, P. Klarer, "Robotic Perimeter Detection System," Proceedings of SPIE Vol. 3577, Boston, MA, November 3-5, 1998, pp. 14-21.