ARMY RESEARCH LABORATORY

# Developing a Chemical Reconnaissance Behavior for Unmanned Ground Vehicles Using the OneSAF Battlefield Simulation Tool

by MaryAnne Fields and Bailey T. Haug

**ARL-TR-2972**                                                    **May 2003**

**20030711 050**

## NOTICES

### Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

# Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5066

---

**ARL-TR-2972** **May 2003**

---

# Developing a Chemical Reconnaissance Behavior for Unmanned Ground Vehicles Using the OneSAF Battlefield Simulation Tool

**MaryAnne Fields and Bailey T. Haug**
**Weapons and Materials Research Directorate, ARL**

# Contents

# List of Figures

# 1. Introduction

One of the goals of the U.S. Army Ground Robotics Research Program is to develop individual and group behaviors that allow the robot to contribute to battlefield missions such as reconnaissance. Since experimental time on the current robotic vehicle, referred to as the experimental unmanned ground vehicle (XUV), is divided between many organizations, it is essential that we develop a simulation tool that will allow us to develop and test behaviors in simulation before porting them to the actual vehicle. Other benefits of developing behaviors in simulation are the ability to expeditiously exercise the behavior in varied environments and the opportunity to make mistakes without catastrophic effects on the robot. In this report, we describe our efforts to develop a chemical reconnaissance behavior for a team of three XUVs.

Some general background on the chemical reconnaissance mission, as outlined in the Scout Platoon field manual [1], is presented as the basis for the robotic behavior algorithm. The reader will be introduced to the One Semi-Automated Forces (OneSAF) simulation tool and the modifications that have been made to OneSAF to support the behavior development efforts. The report includes a discussion of the basic behavior algorithm as well as enhancements required to add robustness to the behavior in the simulation environment. The next step was to port the behavior algorithm to the robotic platform. Although the ultimate target vehicle for the behavior is the XUV, surrogate robots were used to demonstrate the behavior and to facilitate experimentation and the evaluation of the behavior. The issues discovered in porting the behavior will be discussed. The report concludes with a discussion of potential extensions to the basic behavior development tool.

# 2. Background on the Mission

As stated in section 1, we intend to use the OneSAF simulation tool to develop autonomous behaviors for ground robotic systems. The primary research goal of the U.S. Army Ground Robotics Research Program is to provide autonomous mobility for robotic vehicles in future Army programs. The scout mission was selected as the application to demonstrate the utility of ground robotics and to showcase the advances in autonomous mobility. There are many aspects of the overall scout mission including reconnaissance, surveillance, and target acquisition (RSTA); classification of terrain features; and locating obstacles such as minefields or regions contaminated by nuclear, biological, or chemical (NBC) weapons. As a proof of concept, we wanted to concentrate our behavior development efforts on an important element of the scout mission that could be accomplished with minimal operator intervention and integrated with the existing Demo III XUV using currently available technologies.

We chose to concentrate on a mission to locate and mark a region contaminated by an NBC weapon. We adapted our behavior from the description given in the Scout Platoon manual [1] of the manned chemical reconnaissance mission. This mission, as performed by manned scouts, uses existing chemical sensors that alert the user of the existence of contamination. We assume that these sensors can be mounted on the XUV and that signals from these sensors can be interpreted by the XUV software.

In this work, we focus on locating and mapping a persistent contaminating agent that has already been released and settled on the surface of the terrain. The tactical use of these agents is similar to the use of minefields. Such agents are used to canalize friendly forces or to deny them access to intersections—likely avenues of approach—or other key terrain features. In the next several paragraphs, we describe the current manned mission, as outlined by the Scout Platoon manual. We discuss our implementation of this mission for the robotic vehicles in section 4.

Appendix B of the Scout Platoon manual [1] details NBC operations for a scout platoon. The preferred scenario is for an NBC reconnaissance unit to perform the marking of a contaminated region; however, few of these units exist, so the scout platoon must be prepared to perform this task. It will become apparent as we walk through the steps extracted from the Scout Platoon manual that the procedure for mapping persistent ground contamination lends itself to an algorithmic approach. The mission is to define the contaminated region only to the degree needed by the scout's commander to maneuver the main body. The minimum information is the dimensions of the rectangle enclosing the area.

Once reaching the suspected objective area, the platoon assumes a three-section organization. Using a bounding overwatch movement technique, the sections move forward sampling every 200 m. When contamination is detected, the platoon stops and a three-vehicle section is organized to mark the area.

The mapping takes on a logical step-by-step approach. The vehicle that first detected the contamination is designated the base vehicle, and its direction of movement becomes the reconnaissance direction of travel. The initial near side limit of the contaminated area is a line orthogonal to the direction of travel through the last point where the base vehicle had a negative detection. Two vehicles are selected as left and right wing vehicles, and the three vehicles are positioned on the near side limit in a line formation with 400 m of lateral separation. The remainder of the platoon is reassigned, possibly to provide security while mapping is in progress.

The base vehicle moves forward in bounds taking samples every 200 m until two consecutive negative samples are confirmed. This point defines the initial far side limit. The far side limit is a line through the second negative sample position, orthogonal to the base line. The initial near side and far side limits provide the first estimate of two sides of the rectangle.

2

The wing vehicles now proceed to mark the left and right sides. The wing vehicles bound forward in the direction of travel, sampling every 200 m. When a wing vehicle gets a positive detection, the driver turns 90° away from the base line, proceeds 200 m, and resamples. If the sample is negative, the driver turns 90°, back to the direction of travel, and resumes checking at 200-m intervals along the direction of travel. If the sample was positive, the driver turns 90°, now opposite to the direction of travel, and continues to bound and sample at 200-m intervals until a negative sample is taken. This process could result in a new near side limit. Once a negative sample is taken, the driver turns 90° away from the direction of travel, bounding 200 m and sampling. If another positive sample is taken, the vehicle would resume bounding and sampling 180° from the direction of travel; however, if a negative sample was taken, the vehicle resumes bounding and sampling along the direction of travel (Figure 1).
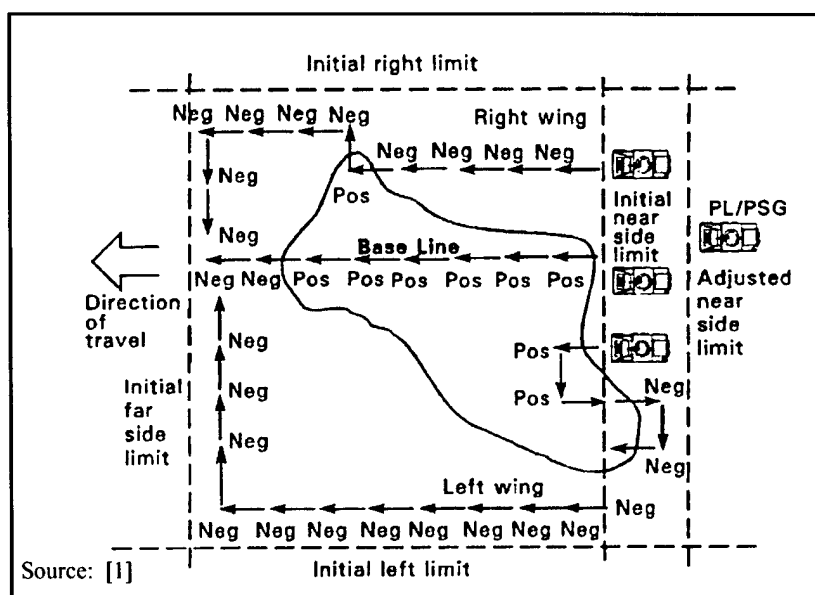


Figure 1. Example of mapping of chemical contamination.

When the wing vehicles have reached the far side limit, an orthogonal projection from their position to the near side limit defines the initial left and right side limits. The drivers turn toward the baseline and continue to bound and sample every 200 m. If they reach the base line with no positive detections, they have enclosed the contaminated region. If a wingman gets a positive detection, the driver turns 180° and proceeds back to the last negative detection, turns along the direction of travel, and begins the process that defines the relevant side limit. If after traveling 200 m in the direction of travel a negative sample is taken, this location defines the new far side limit. The wing vehicle again attempts to move toward the base line, sampling every 200 m. The other two vehicles must adjust to this new far side limit. The base vehicle starts bounding and sampling every 200 m until reaching the far side limit defined by wing vehicle. The second

wing vehicle moves back to that side limit and proceeds bounding and sampling until the new far side limit is reached. The far, left, and right side limits are adjusted by the bounding vehicles and the process continues until the area is enclosed.

The movement of the three vehicles is conditional, based on several factors including positive or negative detections, the vehicle's direction of travel, the reconnaissance direction of travel, the base line, and the far side limit. The left, right, and near side limits are defined by the bounding of the vehicles but are not considered in the decision process. The end result is a rectangle, formed by the four limit lines, that bounds the contamination. This mission is only intended to provide sufficient information to maneuver the main body, not to provide a detailed map of the contamination. The step-by-step approach and the hazardous nature of the task both make this a desirable mission for a robotic vehicle.

## 3. The OneSAF Simulation Tool

### 3.1 Baseline Features

OneSAF [2] is an interactive battlefield simulation tool developed by the U.S. Army Simulation Training and Instrumentation Command that simulates the behavior of units, their vehicles, and their weapon systems to a level of realism sufficient for training and combat development. It provides users with the capability to create and control units ranging in size from individual combatants and platforms through battalions. The simulation package also includes a representation of the physical environment, including terrain, diurnal cycle and weather, and its effect on simulated activities and behaviors.

OneSAF has many desirable features for developing and testing robotic behaviors. It is an easy-to-use, interactive tool that allows users to design test scenarios. Currently, there are several hundred different types of units that can be used in these scenarios. These units range in size from individual soldiers to battalions. The units include both air and ground systems and represent both U.S. and foreign systems. The actions of these units can be controlled by the user or, to a limited extent, controlled by OneSAF behavior algorithms. Users can add new units and behavior algorithms to the base systems to support specific projects. There are many terrain databases available for OneSAF. These terrain databases include U.S. Army installations such as Ft. Knox, KY; Ft. Hood, TX; and the National Training Center at Ft. Irwin, CA; as well as parts of Europe and Asia. In addition, commercial packages such as MultiGen Creator* can be used to provide three-dimensional visualization of the terrain databases.

---

* MultiGen Creator is a trademark of the MultiGen-Paradigm Corporation.

Finally, there is a connection between behavior development for computer-generated forces and behavior development for ground robots that should to be exploited. Within the OneSAF community, there is significant research in the area of behavior representation for simulated forces. This research includes developing representations of U.S. and foreign military doctrine for simulated units at all levels of force structure; developing militarily sound reactions to battlefield events such as enemy contact, indirect fire, and air attack; coordinating behavior among simulated entities [3, 4]; and route planning [5]. While there are fundamental differences between ground robots and simulated entities, ground robotics should be able to use some of the behavior algorithms developed for computer-generated forces.

Unfortunately, OneSAF does have limitations as a tool for developing and testing robotic behaviors. These limitations can be grouped into two categories: terrain database limitations and entity behavior limitations.

Most of the terrain databases that are available for OneSAF have elevation posts spaced 30–125 m apart. This results in a very "smooth" terrain surface that does not accurately model the terrain encountered by a small vehicle. Many terrain features, such as trees, wooded areas, roads, rivers, and buildings are "layered" on top of the elevation grid as linear or polygonal abstract features. Although these abstract features do affect the activities of the simulated entities, they are not directly sensed by the sensory equipment attached to entities. It is difficult to examine the robustness of behaviors that involve autonomous mobility without including a model of how the driving sensors acquire information about the environment. Also, most OneSAF terrain databases do not contain ditches, holes, rocks, boulders, and other small obstructions that present significant obstacles for ground robots.

The current OneSAF mobility behavior algorithms assume a competent human driver is controlling the system. This driver model "perceives" and responds appropriately to obstacles in the terrain, updating the vehicle position and velocity several times a second. In fact, since the driver is assumed to be competent, most OneSAF terrain databases do not contain small mobility obstacles to stimulate the driving algorithms. We cannot assume a competent driver for any autonomous robot since a major issue is the robustness of its driving algorithms. We have not fully investigated other behavior algorithms in OneSAF; however, many of the algorithms are trying to simulate *human* actions so they may use information and intelligence not yet available to ground robots. In general, we would like to replace the OneSAF behavior algorithms with a better representation of robotic behavior.

## 3.2 U.S. Army Research Laboratory Extensions

We have extended the basic features of the OneSAF simulation code to better represent ground robotic features. Our work can be divided into two categories: terrain modifications and robot-specific modifications. The terrain modifications overcome some of the limitations of the terrain databases described in the previous section, providing the simulated robot with a rich

environment containing both large and small obstructions that need to be sensed and incorporated into its mobility plan. There are many different approaches to modifying the OneSAF terrain databases to support mobility analysis for robotic vehicles. Fields [6] provides a detailed discussion of these modifications. In this section, we briefly describe the mobility obstacle editor that we used in analyzing and developing this behavior algorithm.

The mobility obstacle editor allows researchers to introduce obstacles to an existing terrain database to stimulate the perception and planning processes on the robotic vehicle. There are two types of obstacles: positive obstacles (representing rocks, bushes, and other obstacles above the ground plane) and negative obstacles (representing ditches, culverts, and other holes in the ground plane). Using the editor shown in Figure 2, researchers can control the size, shape, number, and distribution of these obstacles. The figure shows a divided window from a running simulation. The bottom half of the window is the obstacle editor. The top half of the window shows a portion of the battlefield map. Positive obstacles are shown in dark red; negative obstacles are shown in gray. By setting the *detectability* and *average detection distance* parameters in the editor window, the researchers control the detectability of the obstacles. Using the obstacle editor several times results in the heterogeneous group of obstacles such as the distribution shown in Figure 2. All the information for the obstacles is saved so that the distribution can be duplicated in subsequent simulations.

In this research, we needed a method to contaminate a region on the simulated battlefield. By designing an editor similar to the obstacle editor, we can place contaminated regions on the battlefield. An example of the contaminate editor is shown in Figure 3. The parameters shown in the editor window determine the size, shape, and location of the region. On the map, the contaminated area is indicated by the green polygon. Again, the parameter settings can be saved for use in other simulations and the editor can be used multiple times. In this research, we use both the obstacle and contaminate editors to test and debug the chemical reconnaissance behavior.

In addition to the obstacle and contaminate editors, we developed algorithms of robotic driving perception and robotic mobility which have been documented previously [7]. These algorithms model the perception and planning processes of the robot. The perception algorithms are "aware" of the mobility obstacles previously discussed. At each time step, the robot constructs a world model showing detected obstacles and features within a 50-m radius of the robot. The detection of a specific obstacle is a random variable whose probability distribution function is specified by the detectability parameters. Figure 4 shows a world model superimposed on the terrain map. Polygons outlined in yellow have been detected; the remaining polygons are out of range of the driving sensor.
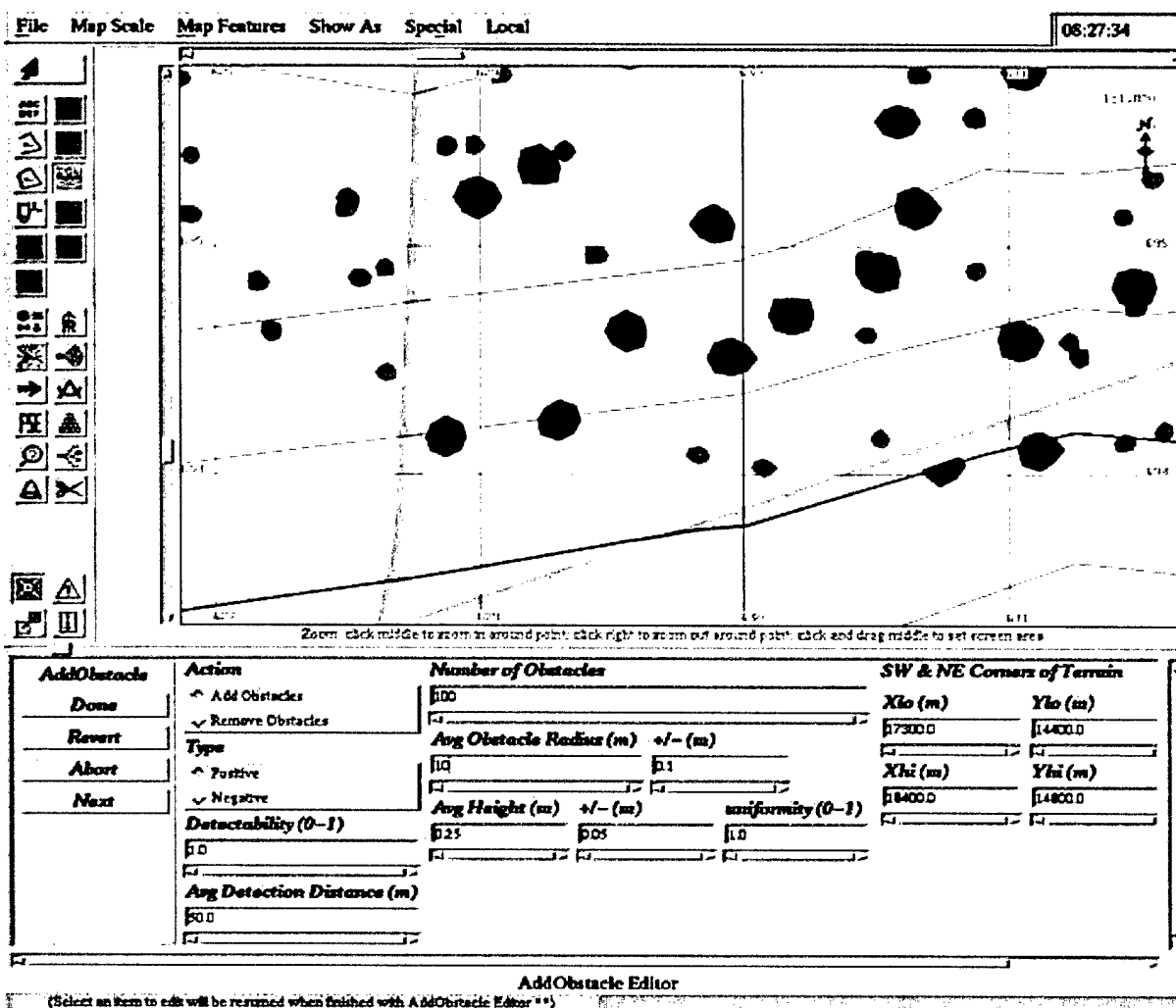
6
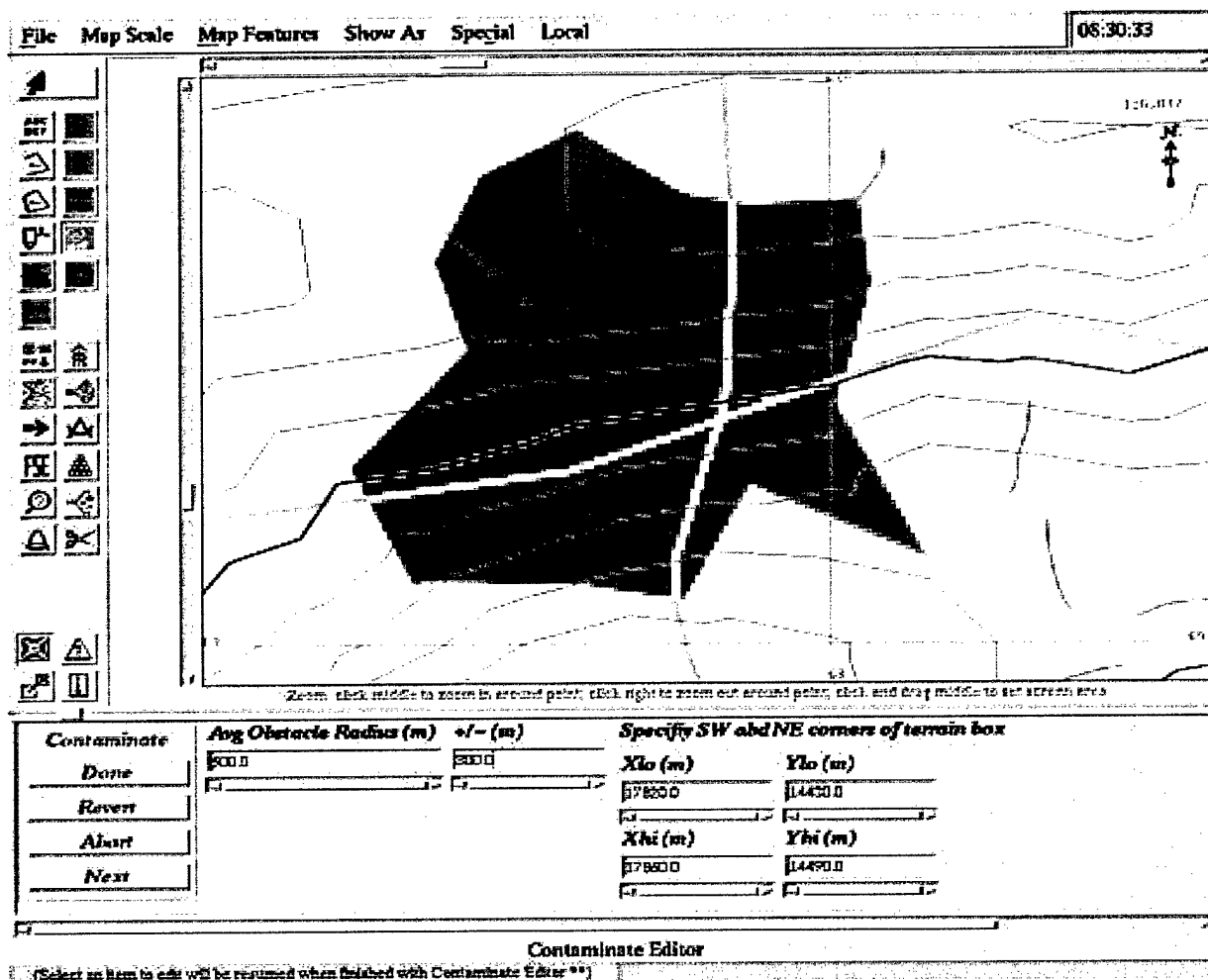
Figure 2. The obstacle editor.
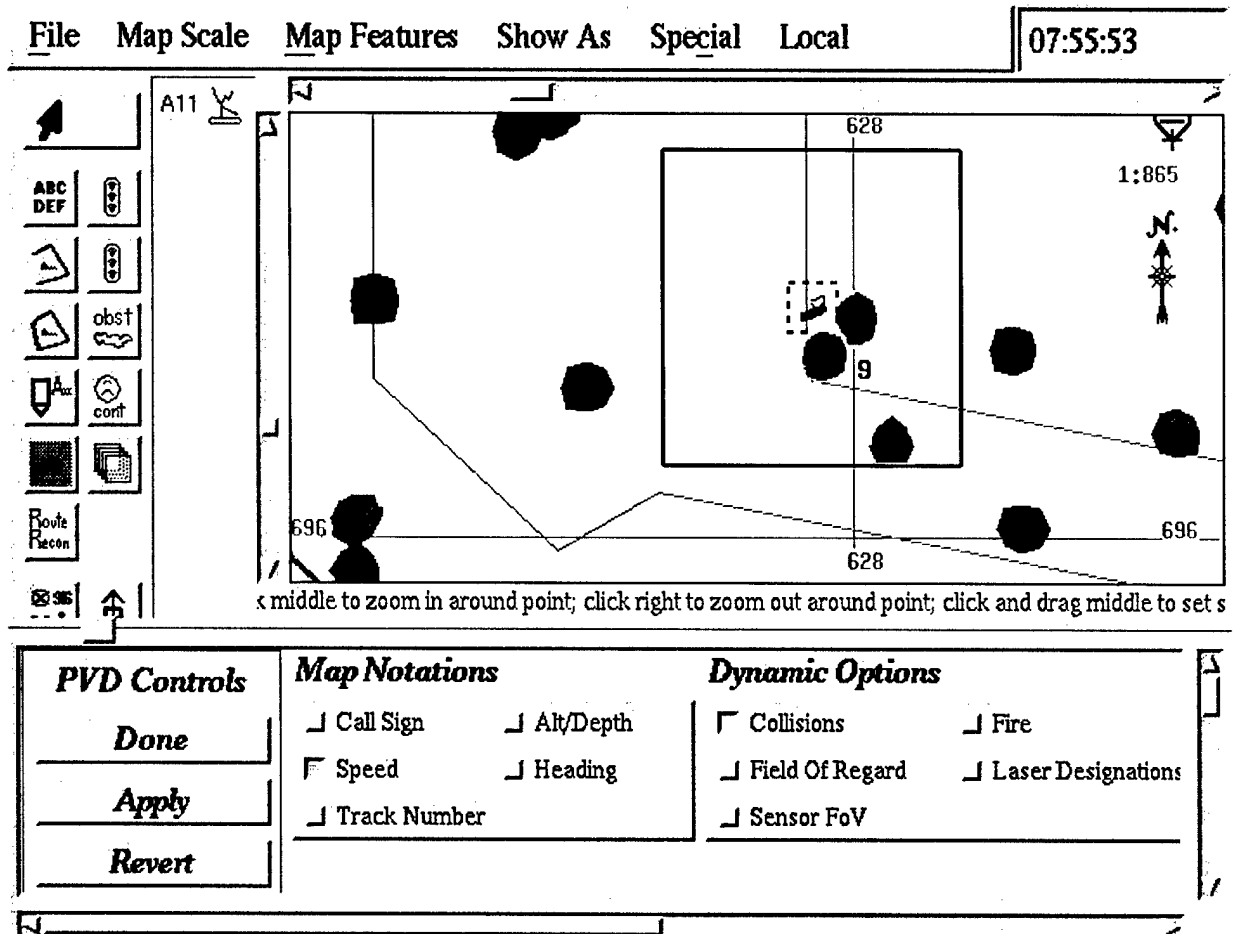
Figure 3.  The contaminate editor.

Figure 4. A simulation display from the OneSAF simulation showing the world model information for an unmanned ground vehicle (UGV).

# 4. Basic Algorithm

In this section, we describe the basic algorithm for a robotic team to locate and map a contaminated region on the ground. There are some differences between the robotic mission and the manned scout mission. Our simulation begins with the three-vehicle section that is normally organized after initial detection. This eliminates the need to model the entire platoon and simplifies the process of reorganizing the platoon once chemical contamination is detected. Later, we can extend our basic behavior to include a platoon of vehicles participating in the mission. We also did not require the robots to use a bounding overwatch movement technique. Bounding overwatch can be added later without changing the underlying mapping behavior.

We have broken the mapping algorithm into five distinct phases: (1) locating the area, (2) regrouping, (3) establishing the baseline, (4) mapping the region, and (5) completing the mission. Each phase represents a distinct behavior involving one or more of the robots. For now, we assume that transition between the phases is instantaneous—in reality, the transition times depend on the speed and reliability of the robots' communication systems.

## 4.1 Phase I: Locating the Area

In this phase the robots must find the contaminated region. We assume that the soldier/operator has intelligence information giving an approximate location of the contaminated area. From this information, the operator specifies an initial rally point that forces the robots to cross the suspected area. Figure 5 shows an example map. The suspected area is a large circular region shaded gray; the actual contaminated region is an irregularly shaped polygon shaded green. Keep in mind that the robot and the operator don't know the location of the contamination a priori. Just as on the battlefield, it is possible for the operator to select a search path that misses the contamination. In this case, the operator picked a rally point that forced the robots to travel through the actual contamination. During this phase, the robots move toward the rally point along parallel paths. The spacing between the robots can be specified by the operator. This phase ends when one of the robots makes contact with the contaminated region or all the robots reach the rally point. The NBC sensor model assumes perfect instantaneous detection so that any contact with the contaminated region will result in a detection. Unlike the NBC sensor described in section 2, this sensor samples the environment continuously. If all the robots reach the rally point, it is up to the operator to reevaluate the mission. He may choose to send the robots through the region again, continue the search to a new rally point, or he may choose to abort the mission.

## 4.2 Phase II: Regrouping

Once the robot team makes contact with the contaminated area, the team reorganizes itself to efficiently map the region. The robot that made the initial contact is designated as the base robot, the other two vehicles are designated as the left wingman and the right wingman. In the current algorithm, the left and right wingman positions are assigned arbitrarily. With future improvements, it will be possible to use information about the relative positions of the robots to make these assignments. In this phase, the base vehicle is stationary. During the second phase, the left and right wingmen prepare for the remainder of the mapping mission by repositioning themselves at a rendezvous point 50 m behind the current position of the base vehicle. (Note that this differs slightly from the task outlined in the Scout Platoon field manual [1].) Figure 6 shows a robot team positioning itself in the regrouping phase.
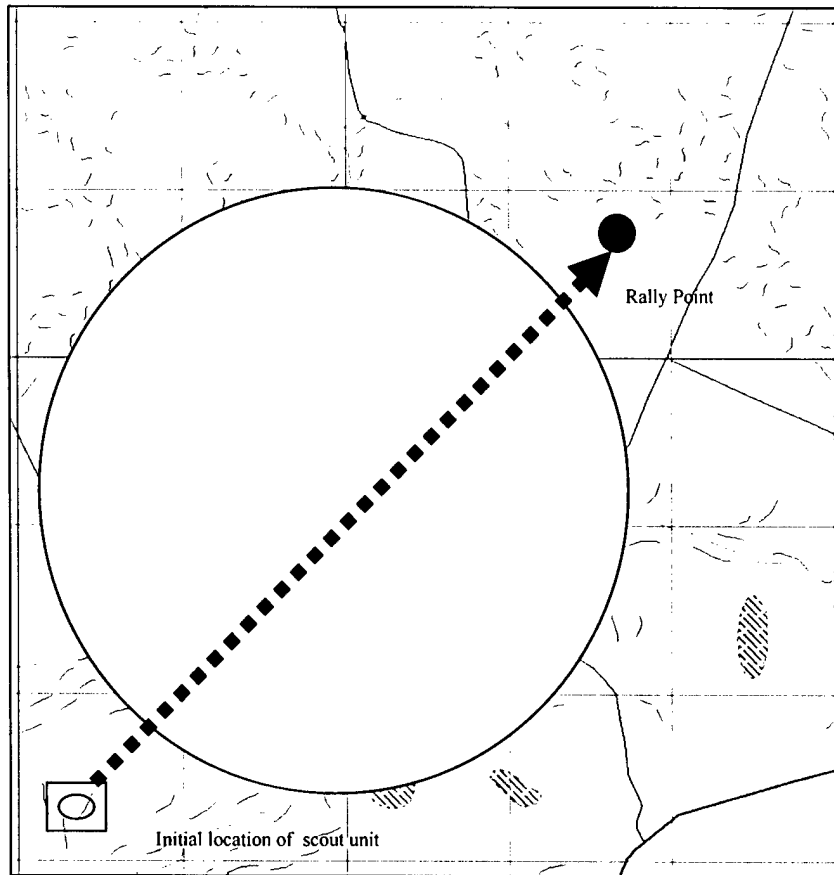
Figure 5.  A battlefield map showing region of suspected contamination (gray)
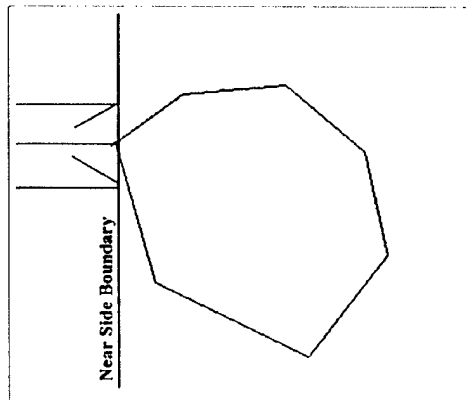and actual contamination (green).



Figure 6.  Phase II, the regrouping phase.

## 4.3    Phase III:  Establishing the Baseline

In the third phase of the mapping mission, the base robot needs to determine the extent of the
contaminated region.  It travels through the contaminated region toward the rally point, designated in
the first phase, using its NBC sensor to look for the end of the contaminated region.  Its line of travel
is referred to as the baseline.  The contamination may not be completely uniform throughout the

11

suspected area, so the base robot cannot rely on a single negative sample from its NBC sensor to determine the extent of the contaminated region. Instead, it needs to have a continuous set of negative samples, gathered over a 200-m segment of travel along the baseline to determine the extent of the contaminated region. As a guide to the two wingmen, the base robot determines a far side limit, which is perpendicular to the baseline at its current position. Figure 7 shows a robot team at the completion of the third phase and the far side boundary line.
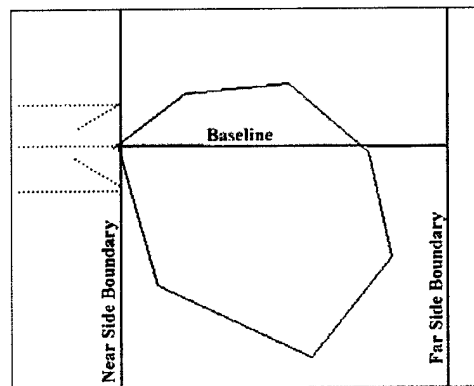


Figure 7. Phase III, establishing the baseline
and far side boundary.

## 4.4 Phase IV: Mapping the Region

The left and right wingmen map the contaminated region in the fourth phase of the mission. Before they start, the wingmen know, from communications with the base robot, the location, and direction of the baseline, and the location of the far side limit. The left wingman will map the region in a clockwise direction, and the right wingman will map the region in a counter-clockwise direction.

To map the contaminated region, the robots execute a series of FIND and REPOSITION steps. In the FIND step, the goal is to find the contamination. Normally, in this step, the robot drives in a direction parallel to the baseline until it finds contamination. In the REPOSITION step, the goal is to move a fixed distance from the contaminated region, then set up for the next FIND step.

The left wingman begins the mapping process with a FIND step by driving along the baseline toward the suspected contamination. When the NBC sensor registers contamination, the robot stops, records its current position, and turns 90° to the left. Next, the robot executes a REPOSITION step by traveling 200 m, then turning 90° to the right. At this point, the robot is pointed in a direction parallel to the baseline so it can execute a FIND step.

Likewise, the right wingman begins the mapping process with a FIND step by driving along the baseline towards the suspected contamination. When the NBC sensor registers contamination, the robot stops, records its current position, and turns 90° to the right. Next, the robot executes a

REPOSITION step by traveling 200 m then turning 90° to the left. At this point, the right wingman can execute a FIND step.

The mapping process continues until both robots reach the far side limit. Figure 8 shows the two wingmen executing the fourth phase of the mission. In this illustration, the left wingman has reached the far side limit in three steps. The right wingman reaches the far side boundary in nine steps.
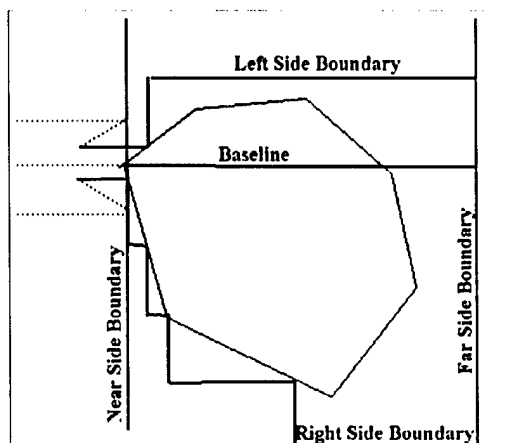


Figure 8. Phase IV, establishing the left and right side boundary.

## 4.5 Phase V: Completing the Mission

In the final phase of the mission, the wingmen rejoin the base robot. Presumably, once the robot team has reassembled, they would perform decontamination procedures and prepare for any further missions from the operator.

The algorithm described in this section works well enough to map simple convex areas of contamination in benign environments. However, it is not robust enough to adjust to more realistic missions. In the next section, we will describe improvement to the algorithm to make it more robust.

# 5. Adding Robustness

In this section, we describe our efforts to make the mapping algorithm more robust. We concentrate on three problems: (1) complex contaminated regions, (2) the loss of one or more robots, and (3) performing the mission in complex terrain.

## 5.1 Concave Regions

In the previous section, Figures 6–8 show a contaminated region consisting of a single convex polygon. In reality, the shape of the contaminated region depends on the prevailing winds, the

13

shape of the terrain, the density and type of features on the terrain surface, as well as the delivery system for the contaminant. A realistic contaminated area might contain concavities, holes, or even disconnected subregions. No algorithm can be designed to cover every contingency, but we will show improvements to our basic algorithms that will allow us to complete the mapping mission for a more realistic region.

To map concave sections, the algorithm must allow for a more complex decision process similar to what is used by the manned scout vehicles. Mapping a contaminated region requires the wingman robots to execute a series of FIND and REPOSITION steps. Consider the path of the left wingman robot as illustrated by the black dashed path shown in Figure 9 (F1, R1, F2, R2, F3, R3, and F4). FIND steps are labeled with an F, and REPOSITION steps are labeled with an R. The path segments F1 and R1 map a convex portion of the contaminated area. The F2 segment enters a concavity formed by the edges of the contaminated region. In the original algorithm, the REPOSITION step required the robot to turn 90° and then drive a fixed distance (200 m) away from the contamination. In concave regions, it may not be possible to drive 200 m without entering the contaminated area. The original algorithm misses part of the contaminated area resulting in a bounding rectangle that does not enclose the contaminated area. Note that the F3 segment has length 0—since the contamination sensors operate continuously during the FIND step, the robot immediately detects the contamination and starts the REPOSITION step.
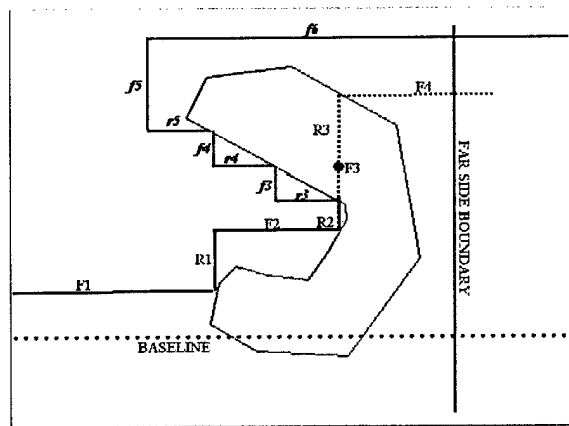


Figure 9. Mapping a concave polygon.

The basic algorithm can be extended by reconsidering the possible transitions between the FIND and REPOSITION steps. In the basic algorithm, only two transitions are possible: FIND-to-REPOSITION and REPOSITION-to-FIND. To effectively map an arbitrary polygon, we need the other two transitions, FIND-to-FIND and REPOSITION-to-REPOSITION. Figure 9 illustrates both of these transitions. The first transition occurs on the line segment R2 when the robot contacts the contaminated region. The basic algorithm requires the robot to proceed to F3. A better approach is to recognize that the region cannot be mapped from the inside, so any contact with the contaminated area should force the robot to execute a REPOSITION step. In

14

the illustration, the extended algorithm transitions from REPOSITION step R2 to REPOSITION step *r3* at the contact point. Following the blue path (*r3*, *f3*, *r4*, *f4*, *r5*, *f5*, *f6*), this approach works well until line segment *f5*. It is not possible to travel in the direction of *f5* and reach the far side boundary, so the new algorithm needs to allow FIND-to-FIND transitions. Recall that in the basic algorithm, the robot stayed in the FIND step until it encountered the contaminated region. In the extended algorithm, the distance and direction traveled can be used to signal a FIND-to-FIND transition. In the illustration, the robot travels along line segment *f5* until the distance to the baseline is larger than a set threshold, then it turns onto the *f6* segment.

Figure 10 illustrates another situation that can arise. In this case, the concavity is on the other side and the base robot incorrectly identifies the far side boundary. As the wingmen attempt to rendezvous with the base robot, they contact the contaminated region. In the extended algorithm, this information is passed to the base robot. The base robot travels along the baseline until it can establish a new far side boundary. The wingmen resume the mapping phase, using the new far side boundary as an exit criteria.
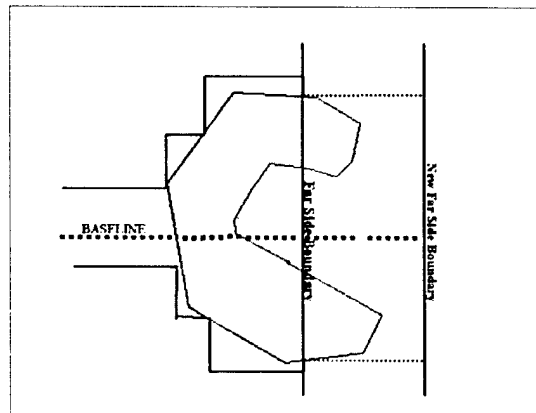


Figure 10. Adjusting far side boundary.

## 5.2 Loss of One or More Robots

The algorithm as presented in the previous sections requires three robots to map a contaminated region. What if some of the robots sustain ballistic damage or break down? Can the mission continue? The mission can continue provided there is a way to monitor the progress and health of the robots.

Extending the current algorithm so that it automatically adapts to the loss of one or more robots requires the mapping behavior to "monitor" the status of the robots and some adaptation to the mapping phase. Monitoring the status of the robots requires some simple communication between the robots and a central control unit—the robots periodically report their position and status. The robots also report the location of contaminated points, as they encounter these points. If a robot fails to report (or reports that it is damaged), it is assumed to be damaged and unavailable for the remainder of the mission. Adjusting the mapping phase involves fixing the length of the FIND step and using a new exit criteria to end the mapping phase. As an example,

consider two cases: the loss of a single robot in Phase I and the loss of two robots in Phase I. These two cases can be generalized to cover the loss of robots at any time in the mission.

Consider the loss of a single robot in Phase I. At this point in the mission, all robots are looking for the contaminated area—losing a single robot will not significantly change this phase of the mission. However, once the initial contaminated point has been discovered, the robots must reorganize themselves to map the region (Phase II). In the reduced robot team algorithm, we eliminate the role of the base robot and proceed directly to Phase IV, the mapping phase. This leaves the two remaining robots to map the outer edge of the contaminated region without prior knowledge of the location of the far side boundary normally determined by the base robot in Phase III. In the previous algorithm, the robots conducted a series of FIND and REPOSITION steps to map the region. The FIND step did not have a fixed step size—the robots drove forward until they detected contamination or until they crossed the far side boundary. After both robots crossed the far side boundary, the robots began the final phase of the mission. In the reduced robot team algorithm, the maximum length of the FIND step is a fixed length. The mapping phase continues until the two robots are within a fixed distance of each other. Figure 11 shows a comparison between a 3-robot team mapping a contaminated area using the original algorithm and a 2-robot team mapping the region with the adjusted algorithm. In this illustration, the algorithms perform similarly on the left side of the area. On the right side, the reduced robot team algorithm must "map" the area to determine the far side boundary.

The reduced robot team algorithm can be used when two robots are lost in Phase I. In this case, the remaining robot continues with Phase I until it detects contamination. The robot uses the adjusted algorithm to map the region; the mapping mission continues until the robot returns to its initial detection point. Figure 11c shows the path of a single robot as it maps the contaminated area. The 2-robot and single robot teams map the contaminated area less efficiently than the 3-robot team. To increase the efficiency, it is possible to increase the step size for the FIND and REPOSITION step.

## 5.3 Performing the Mission in Complex Terrain

In this section, we limit our discussion of complex terrain to terrain surfaces containing a significant number of mobility obstacles that force the robots to deviate from their planned course.

In some ways, performing the mission in complex terrain is similar to performing the mission with a reduced robotic team. Complex terrain can degrade the mobility of one or more of the robots to such an extent that these robots are "lost" to the mission. In the previous discussion, the mapping behavior monitored the status of the three robots as they performed the mission. Adapting the mission to complex terrain requires the behavior to monitor the progress of the robots, as well their status. Progress measures the change in the distance between the robot and its current goal for a given time period.

16
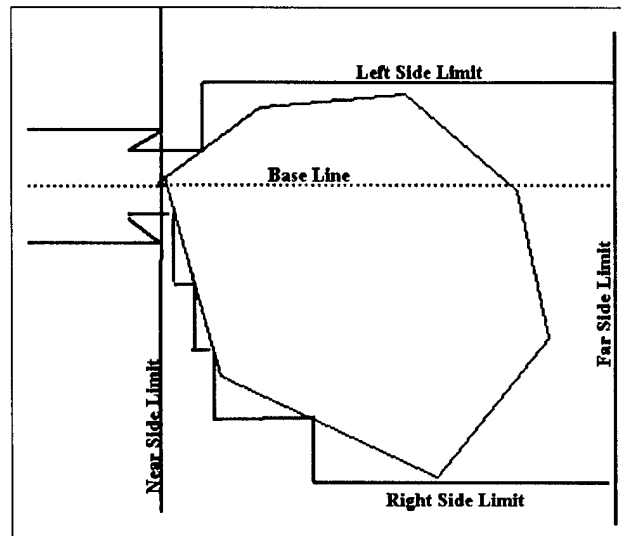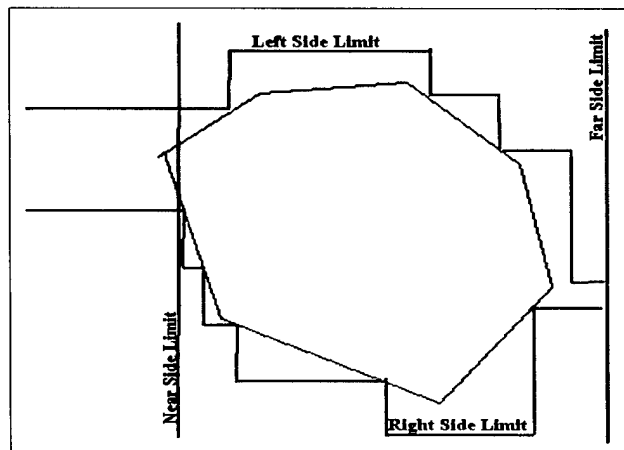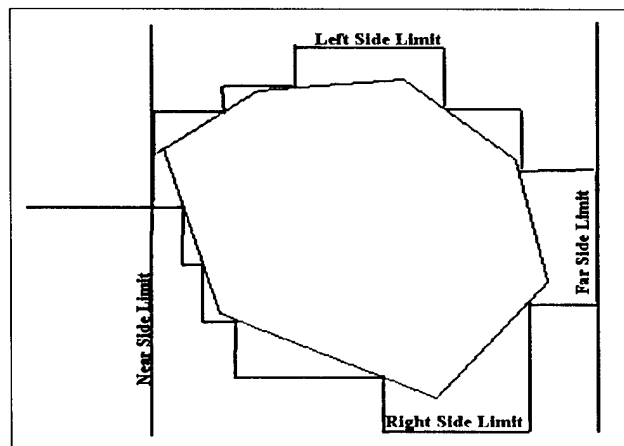
Figure 11. Mapping the contaminated area with (a)
3-robot team, (b) 2-robot team, and (c)
1-robot team.

17

The goals themselves are a function of the phase of the mission and the robot's particular assignment. In the first phase, progress measures the change in distance between the robots and the rally point (Figure 5). In Phase II, only the wingmen are moving; progress for a wingman depends on the distance to the rendezvous point. Recall that the base robot crosses the contaminated region in Phase III moving toward the Phase I rally point. We measure the progress of the base robot in Phase III by measuring the changes in distance between the base robot and the Phase I rally point. In Phase IV, progress depends on the distance between the wingmen and the far side limit. In Phase V, progress depends on the distance between the wingmen and the base robot.

In this work, we have extended the algorithm for two cases: lack of progress of the base robot in Phase III and a significant difference in the progress of the wingmen in Phase IV. In both cases, we will treat lack of progress the same as the loss of a robot—it simplifies the algorithms. The first case is a straightforward extension of the algorithm given in section 4.2. The base robot cannot find the far side boundary in a reasonable amount of time so the wingman robots begin to map the region as if they were a 2-robot team.

In the second case, one of the wingmen has difficulty mapping its side of the contaminated region. Recall that the base robot is idle during Phase IV so it can be reassigned the task to mapping a portion of the contaminated region. The base robot uses the near side boundary as its exit criteria and maps the region in the opposite direction. Once the base robot and the remaining wingman reached their respective boundaries, the bounding rectangle can be determined.

## 6. Porting the Behavior to a Surrogate Robotic Platform

The next step in this research effort was to port the mapping algorithm from the simulation package to a robotic platform. Two surrogate robots, ATRV-Jr,[*] shown in Figure 12, were selected to demonstrate the mapping behavior on actual robotic platforms. The robots are four-wheeled, skid-steered platforms that can be used indoors and outdoors. The ATRV-Jr's sensors include visible spectrum cameras, an ultrasonic range sensor array, a global positioning system, an inertial measurement unit, a compass, and a tilt sensor. To simplify the hardware and experimental requirements, the cameras were used as surrogate chemical sensors, and yellow plywood disks were used to create contaminated regions. This allowed us to focus on the algorithm and not be distracted by the integration of sensors and the use of chemical simulants.

Since there were only two robotic vehicles available, we demonstrated the reduced robot team algorithm discussed in section 4.2 (i.e., we assume that the base robot was lost and that the two remaining robots assume the roles of the left and right wingmen). Adding the base vehicle

---

[*] ATRV-Jr is a trademark of iRobot Corp.

Figure 12. ATRV-Jr robots from iRobot Corp.

would have simplified the mapping process, giving the wing vehicles a baseline and an initial far side limit.

Our experimental setup consisted of the two ATRV-Jr robots, an operator, and his/her laptop computer that acted as the operator control station. Computer code for the robots and the laptop was written for the Red Hat[*] [8] Linux[†] 6.2 operating system and used the Mobility[‡] C/CORBA libraries provided by iRobot Corp. A wireless Ethernet connected the robots and the operator's control station.

Although the basic chemical reconnaissance behavior is the same for both the real and simulated robots, there are differences in the computer programs controlling each robot. First, even though the intent of the battlefield simulation tool is to model the real world, real and simulated robots interact differently with their respective environments. Service programs such as movement, communication, and sensing need to be written specifically for each environment. The actual behavior program depends on the available programming environment. In OneSAF, behaviors are written as a finite state machine that is translated into C code by the programming environment. On the robot, the chemical reconnaissance behavior was written directly in C.

We wanted to demonstrate that the robot team could conduct the mapping behavior without significant operator involvement. In our experiments, the operator had two roles: send the "start mission" signal to the robots and act as safety officer for the experiment. Each robot conducted its portion of the mission independently. Communication was minimal. The robots reported contact points to the operator's computer. The contact points were used to draw a map on the

---

[*] Red Hat is a registered trademark of Red Hat Inc.

[†] Linux is a registered trademark of Linus Torvalds.

[‡] Mobility is a trademark of iRobot Corp.

operator control unit (OCU) screen so that the operator could compare the shape of the mapped region to the shape of the actual region on the ground.

The experiments were successful—the robots were able to consistently map the surrogate contaminated region. We demonstrated both the 2-robot team and the single-robot team mapping procedure.

The experiments identified some issues that are important future behavior development work. First is the importance of modeling system latencies. In our behavior, there were two major sources of latencies: latencies associated with detecting the contamination and latencies associated with communication. In the real world, detection is not instantaneous—the robot may actually drive into the contamination before it registers a detection. Communication latencies are important to consider for future extensions to this work—monitoring the behavior and adjusting the loss of a robot depend on reliable communication between the robots and the OCU.

## 7. Conclusions

This research represents a proof of concept—we were able to develop a behavior using computer simulation, then port it to actual robotic platforms. We chose the chemical reconnaissance behavior because it was a potential mission for robot scout unit and because it was algorithmic in nature. We developed the basic algorithm using a modified OneSAF simulation tool. Using simulation experiments to iteratively test our algorithm, we were able to improve the basic algorithm to respond automatically to loss of robots due to attrition or terrain conditions. By using laboratory robots in a controlled environment, we were able to focus on the development of the behavior without having to implement a full autonomous mobility package. The current implementation of the chemical reconnaissance mission on these robots does not take advantage of communications between robots. The robots were assigned the left and right side mapping tasks a priori, and they didn't track each other's positions. All position information was collected by the OCU. Future efforts in behavior development will take advantage of the ability of the robots to communicate with each other to more efficiently accomplish tasks.

Developing tactical behaviors in a simulation has many benefits. As discussed in this report, using the enhanced OneSAF simulation to represent current UGV capabilities facilitates the development of behaviors that can be readily transitioned to current platforms. The simulations can also point the way to new technology developments and capabilities required to accomplish more complex behaviors.

This research effort demonstrates that, with a realistic representation of a UGV and its environment, a computer simulation is a viable tool for building tactical behaviors for UGVs. The current project focused on a single team behavior that had to be designed from scratch using the simulated world to test and debug the algorithm. By structuring our future research so that

we develop libraries of common skills and behaviors first, we will be able to combine them into complex individual and group behaviors. As the library of common skills and behavior grows, development and testing time for complex behaviors may decrease since each of the common behaviors and skills will be well characterized.

## 8. References

1. U.S. Department of the Army. *Scout Platoon.* FM 17-98, Washington, DC, 10 April 1999.

2. Wittman, R., and C. Harrison. "OneSAF: A Product Line Approach to Simulation." Technical report, contract number DAAB07-01-C-C201, The Miter Corporation, Orlando, FL, 2001.

3. Ioerger, T., R. Voltz, and J. Yen. "Modeling Cooperative, Reactive Behaviors on the Battlefield With Intelligent Agents." Proceedings of the 9th Conference on Computer Generated Forces and Behavior Representation, 2000.

4. Hill, R., J. Gratch, P. Rosenbloom, and R. Whitney. "Flexible Group Behavior: Lessons Learned Building Virtual Commanders." Proceedings of the 9th Conference on Computer Generated Forces and Behavior Representation, 2000.

5. Courtemanche, A., and C. E. Campbell. "Unified Entity Maneuver." Proceedings of the 9th Conference on Computer Generated Forces and Behavior Representation, 2000.

6. Fields, M. "Modifying ModSAF Terrain Databases to Support the Evaluation of Small Weapons Platforms in Tactical Scenarios." ARL-TR-1996, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, August 1999.

7. Fields, M. "Designing a Behavior Development Environment to Support the Demo III Robotics Program." Proceedings of the SPIE Vol. 4364, AeroSense Session on Unmanned Ground Vehicle Technology, Orlando, FL, 7–8 April 2000.

8. Red Hat, Inc. Red Hat Linux 6.2. Raleigh, NC, 1999.

| 1. REPORT DATE *(DD-MM-YYYY)*<br>May 2003 | 2. REPORT TYPE<br>Final | 3. DATES COVERED *(From - To)*<br>1 October 2000–30 September 2002 |
|---|---|---|
| **4. TITLE AND SUBTITLE**<br>Developing a Chemical Reconnaissance Behavior for Unmanned Ground Vehicles Using the OneSAF Battlefield Simulation Tool | | **5a. CONTRACT NUMBER** |
| | | **5b. GRANT NUMBER** |
| | | **5c. PROGRAM ELEMENT NUMBER** |
| **6. AUTHOR(S)**<br>MaryAnne Fields and Bailey T. Haug | | **5d. PROJECT NUMBER**<br>AH03 |
| | | **5e. TASK NUMBER** |
| | | **5f. WORK UNIT NUMBER** |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**<br>U.S. Army Research Laboratory<br>ATTN: AMSRL-WM-BF<br>Aberdeen Proving Ground, MD 21005-5066 | | **8. PERFORMING ORGANIZATION REPORT NUMBER**<br>ARL-TR-2972 |
| **9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)** | | **10. SPONSOR/MONITOR'S ACRONYM(S)** |
| | | **11. SPONSOR/MONITOR'S REPORT NUMBER(S)** |

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

One of the goals of the U.S. Army Ground Robotics Research Program is to develop individual and group behaviors that allow the robots to contribute to battlefield missions such as reconnaissance. By using simulation tools, we are able to develop, debug, and test behaviors before porting them to actual robotic platforms. This affords the researchers the opportunity to expeditiously evaluate the behaviors in a diverse set of environments and to explore variations in behavior algorithms without tying up limited robotic resources or putting these robotic vehicles at risk. This report describes our efforts to develop a proof-of-principle behavior initiated in simulation then ported to a team of surrogate robotic platforms. The issues discovered in porting the algorithm from simulation to the robotic platform are discussed. The report concludes with a discussion of possible extensions to the basic tool.

**15. SUBJECT TERMS**

robotics, behavior algorithm, computer generated forces

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON<br>MaryAnne Fields |
|---|---|---|---|---|---|
| a. REPORT<br>UNCLASSIFIED | b. ABSTRACT<br>UNCLASSIFIED | c. THIS PAGE<br>UNCLASSIFIED | UL | 28 | 19b. TELEPHONE NUMBER *(Include area code)*<br>(410) 278-6675 |

INTENTIONALLY LEFT BLANK.