Advocates and critics for tactical behaviors in UGV navigation

Talib S. Hussain, Gordon Vidaver, Jeffrey Berliner
BBN Technologies, 10 Moulton St., Cambridge, MA 02138*

ABSTRACT

Critical to the development of unmanned ground vehicle platforms is the incorporation of adaptive tactical behaviors for the planning of high-level navigation and tactical actions. BBN Technologies recently completed a simulation-based project for the Army Research Lab (ARL) in which we applied an evolutionary computation approach to navigating through a terrain to capture flag objectives while faced with one or more mobile enemies. Our Advocates and Critics for Tactical Behaviors (ACTB) system evolves plans for the vehicle that control its movement goals (in the form of waypoints), and its future actions (e.g., pointing cameras). We apply domain-specific, state-dependent genetic operators called advocates that promote specific tactical behaviors (e.g., adapt a plan to stay closer to walls). We define the fitness function as a weighted sum of a number of independent, domain-specific, state-dependent evaluation components called critics. Critics reward plans based upon specific tactical criteria, such as minimizing risk of exposure or time to the flags. Additionally, the ACTB system provides the capability for a human commander to specify the "rules of engagement" under which the vehicle will operate. The rules of engagement determine the planning emphasis required under different tactical situations (e.g., discovery of an enemy), and provide a mechanism for automatically adapting the relative selection probabilities of the advocates, the weights of the critics, and the depth of planning in response to tactical events. The ACTB system demonstrated highly effective performance in a head-to-head testing event, held by ARL, against two competing tactical behavior systems.

Keywords: Tactical behaviors, unmanned ground vehicles, navigation, evolutionary computation

1. INTRODUCTION

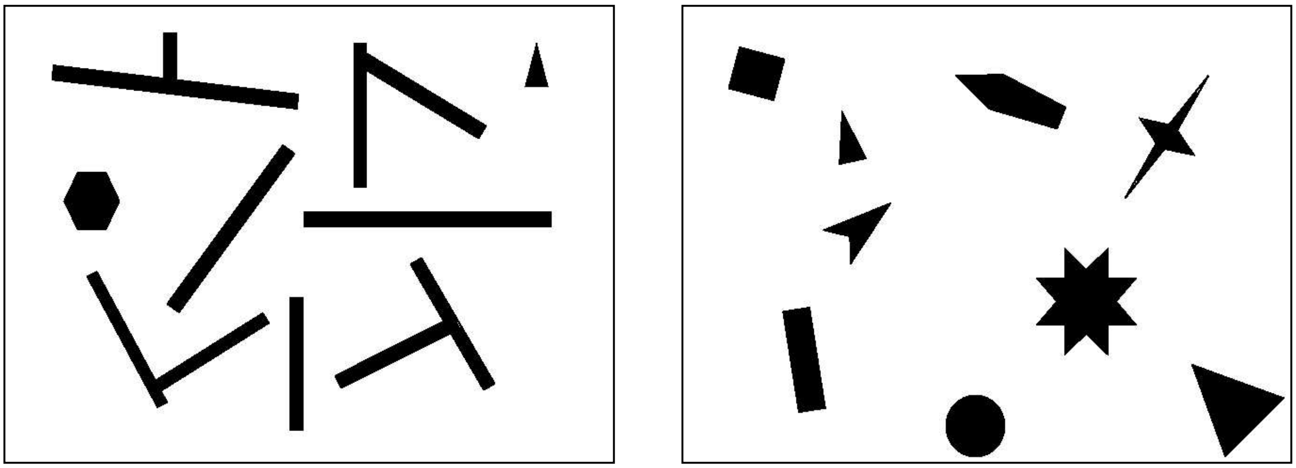
The development of unmanned ground vehicles (UGVs) for the military faces several challenges that arise from the dynamic and complex nature of the environments in which they must operate[1]. UGVs must have the capability to (a) develop and execute plans to carry out higher-level mission goals (such as reconnaissance, surveillance, and target acquisition), (b) navigate effectively through complex terrain, (c) accommodate uncertain, incomplete, dynamic and/or incorrect knowledge, (d) respond rapidly and effectively to threats posed by mobile and intelligent enemy units, and (e) perform all its actions using tactical behaviors that ensure successful accomplishment of its mission goals and its continued survival. While recent developments in UGV navigation have led to partial achievement of some of these capabilities, current technologies fall far short of enabling UGVs that behave in meaningful tactical ways given the rich variety of tasks and risks faced in even simple missions.

BBN Technologies recently participated in a project conducted for the Army Research Lab (ARL) to explore new techniques for incorporating tactical behaviors within UGVs. The challenge presented in the project was to develop UGV tactical behavior software capable of controlling a single simulated UGV operating in a simple simulated world with one or two enemy units. We developed a solution based upon our Advocates and Critics for Tactical Behaviors (ACTB) technique[2]. What made the challenge particularly interesting is that the software was to be tested in a series of trials in which the enemy units were controlled by software developed by two different project members. During a six-month development phase, each of the three project teams was provided with access to the same simulator and given the same instructions on the type of mission objectives and operating conditions they would experience during the test trials. However, there were no interactions between the participants. In a testing event held during September 2004, the tactical behavior solutions of the teams were tested against each other in hundreds of trials. Thus, in a manner similar to robot soccer competitions, each project member had to independently develop an effective strategy for accomplishing the mission objectives as well as anticipate enemy strategies and design effective means for countering them.

We describe the simulated environment in which our software operated, present the details of our approach and summarize the performance of our system.

2. SIMULATION ENVIRONMENT

Our tactical behavior software was tested in a simulator provided by General Dynamics Robotics Systems (GDRS). The simulator defined a simple world consisting of obstacles, clear areas and simulated UGVs. The terrain used was typically fairly simple. Figure 1 illustrates two of the terrains used in the test trials, where white represents clear areas and black represents obstacles. The terrains were 640 x 480 meters in size (a scale of 1 pixel to 1 meter was used).



(a) (b)

Figure 1: Simulated terrains used in test trials

A UGV had a simulated gun turret and a simulated camera, and could be controlled via five basic commands. The vehicle could *move* to a given geographical location, *stop* movement, *look* in a given direction by rotating the camera, *aim* the gun in a given direction, and *fire* the gun. The simulator performed several key tasks.

1. It defined the physics for vehicle movement. Each UGV had a maximum top speed of 5 m/s, and the simulator would automatically accelerate or decelerate the vehicle at a rate of 1 m/s^2 as new movement or stop commands were received. The simulator would also automatically turn the vehicle at a rate of 60 degrees/sec to head towards its next destination. The vehicle would stop when it hit an obstacle.
2. It defined the physics for the camera. The camera could be slewed at a fixed rate (45 degrees/sec) to any angle relative to the axis of the vehicle. The camera turned with the vehicle. The camera had a viewable area determined by the field of view angle (60 degrees) and viewing range (100 meters). Any areas behind an obstacle were not viewable.
3. It defined the physics for the gun. The gun could be slewed at a fixed rate (45 degrees/sec) to any angle relative to the axis of the vehicle. The gun turned with the vehicle. The simulator determined whether a shot fired from the gun had hit or killed an enemy. A hit occurred when the shot was accurate and the enemy was within 80 meters. A hit within 40 meters was always a kill. A hit between 40 and 80 meters had a linear decreasing probability of being a kill. Bullet speed was infinite, and there was a minimum delay of 3 seconds between successive shots. When there was only one enemy UGV, each vehicle had 5 shots available. When there were two enemies, each vehicle had 8 shots.
4. It provided communication mechanisms that enabled the tactical behavior software to send commands to the UGV it was controlling and receive information about the current state of the UGV. In particular, movement commands could be issued as a sequence of waypoints to visit. The vehicle's state included its location, heading, speed, gun angle, camera angle and remaining number of shots.
5. It provided communication mechanisms that enabled the tactical behavior software to receive information about the environment and enemies. It provided the location of all obstacles and flags that the vehicle could see within its viewable area. It also provided each enemy's last known state (i.e., position, heading, gun angle, camera angle and shots remaining).

6. It determined the mission. Each mission required that the UGVs visit three *flag* locations. All three flags were randomly located within a 100 meter circle. At the start of the mission, each vehicle was given information on the centroid of that circular area (or *flag locus*), but not the specific locations of the flags. The flag locations were discovered by a UGV only when viewed with that vehicle's camera.

The simulator performed in real time, typically with half a second delay between issuance of a command and execution of that command. All requested information was also provided in a similar time frame.

3. TACTICAL BEHAVIOR SYSTEM

The Advocates and Critics for Tactical Behaviors (ACTB) technique was initially developed to produce navigation plans for one or multiple UGVs cooperating in a simulated outdoor environment to achieve a set of mission objectives while avoiding any known unmoving enemies[2]. The technique used an evolutionary computation based approach to develop new plans in a continual cycle that sought to automatically evolve better and better plans as the UGVs moved and as the state of knowledge about the world changed. The technique was effective at generating good plans in a very short time scale (typically around a minute). However, in the current work, there was a strong need for effective real-time response given the nature of the simulator and the active nature of the enemy vehicle(s). To accommodate this need, we adapted ACTB in several ways to provide faster planning and effective situation-dependent response.

3.1. ACTB Architecture

Our new ACTB architecture is shown at the highest level in Figure 2. Planning is performed in a continual cycle that is influenced by changes in the state of knowledge about the world as well as by the current planning emphasis of the system. The simulator provides symbolic knowledge rather than raw "sensor" data, and we illustrate this as a distinct common operating picture (COP) component. The COP was maintained through a combination of simulator information queries and stored information from previous queries.

Initially, the planning process starts with a set of new plans that are generated randomly based on the initial world conditions and the general location of the flags. Each plan is a description for proposed UGV actions into the future, and specifies a sequence of move and/or "stop and look" commands. A move command is represented as a waypoint. A stop and look command is represented as a direction to point the camera in. The plan does not specify any gun aiming or firing commands. During execution, a contiguous sequence of waypoints is passed directly to the simulator as a sequence of corresponding moves. The simulator will ensure that the UGV visits those waypoints with minimal slowdown between waypoints (i.e., a smooth, fast turn). A stop and look command is associated with the immediately preceding waypoint – when the vehicle arrives at that waypoint, it will stop and slew its camera to the indicated angle. When the camera is fully turned, the next action in the plan will be executed. If that is another stop and look command, the vehicle will remain at the same waypoint until that look is completed. Otherwise the next appropriate sequence of move commands will be issued.

The new plans are evaluated by a set of *critics* to determine how effective those plans are. Critics are independent specialists. They examine each plan, where appropriate, and evaluate the plan based on that critic's criteria (e.g., "How exposed is this plan?"). They may evaluate the whole plan, a small portion of the plan or nothing in the plan. The results of the critics are combined in a weighted sum to produce a single score for the plan. Thus, the critics form the fitness function of the evolutionary search. Once all the new plans are scored, they are ranked with respect to each other and to plans from previous cycles. The highest ranked plan is always selected as the current plan for execution, and some of the lowest ranked plans are purged. If the best plan has changed since the previous planning cycle, the new plan is communicated to the UGV, otherwise, the UGV continues to execute as previously instructed.

The ranked plans are in turn modified by a set of *advocates* to create a set of new plans that exhibit tactical improvements. Advocates are the genetic operators used within the evolutionary search. The advocates examine existing plans and, when appropriate, create one or more new plans that better meet that advocate's goal (e.g., "I want to skulk along walls" or "I want to peek around the corner"). Most advocates expend a small effort and create a small, incremental change (e.g., by modifying a few move, stop or look commands). Each cycle, only a few advocates are selected to create new plans. Advocates operate independently from one another and may be invoked with a different probability. While most advocates are domain-dependent specialists, some domain-independent genetic operators are also incorporated to ensure a robust evolutionary search.

The planning cycle continues indefinitely as the new plans generated by the advocates are in turn evaluated by the critics, and so on. During each cycle, the advocates and critics operate based on the most recent information available in the COP about the terrain, the enemies and the missions.

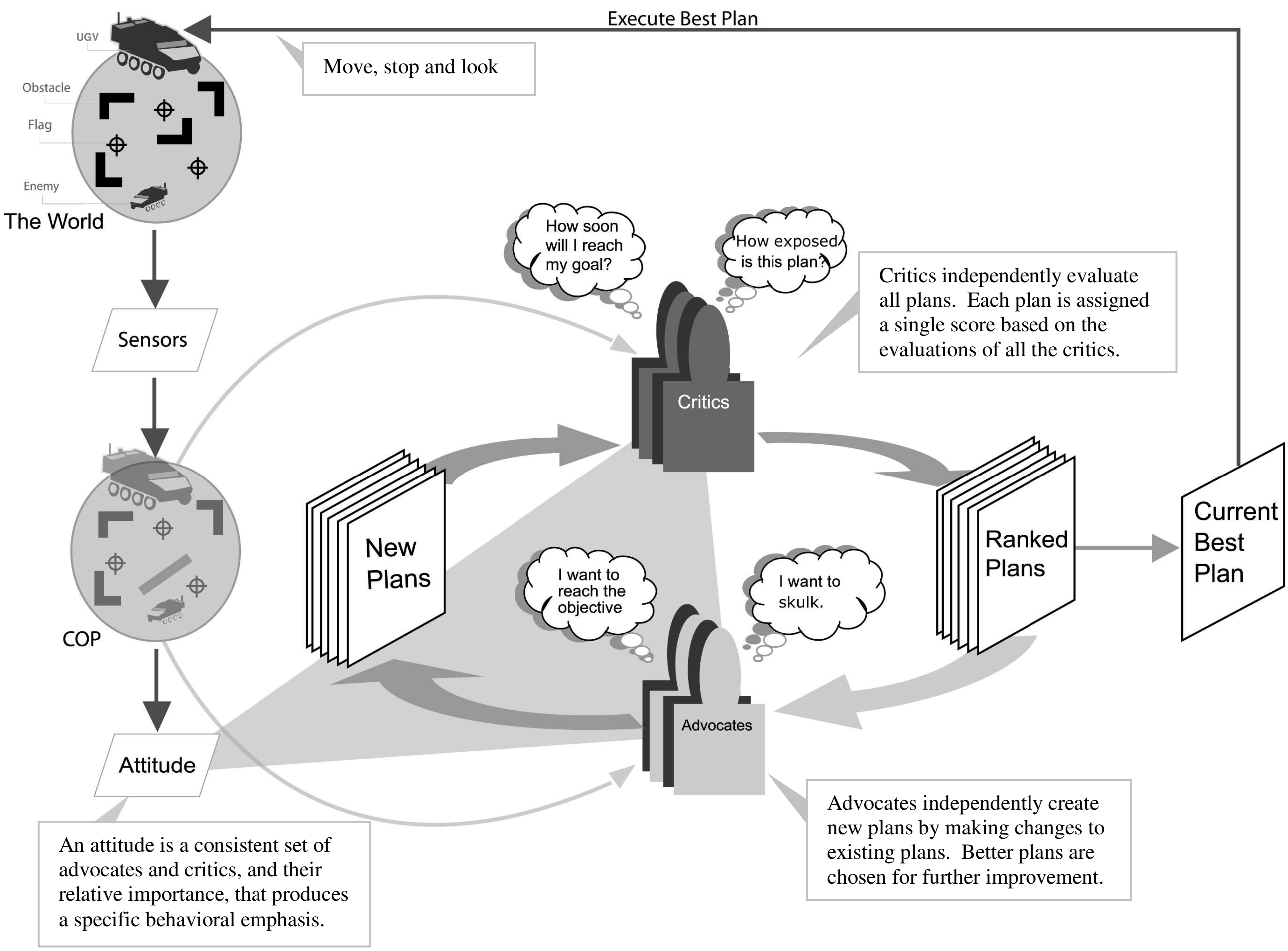


Figure 2: ACTB Architecture illustrating continual planning cycle moderated by current attitude

The new ACTB architecture also incorporates high-level strategic ”rules of engagement” that influence the manner in which planning is performed. In particular, the strategy rules determine the current *attitude* of the system. An attitude reflects a bias in the tactical behaviors used in the planning process. Specifically, an attitude is a consistent set of advocates and critics, each with certain probabilities of selection and weights, respectively. When a new attitude is chosen, the advocates and critics used by the planning process are immediately changed to reflect those specified in the new attitude. This new capability enables the ACTB planning cycle to be highly responsive to important changes in the tactical situation. Instead of trying to solve one difficult optimization problem, ACTB uses attitudes to solve several simpler ones to improve the quality and timeliness of its response. For example, during a mission it may be appropriate under different conditions for a UGV to become more cautious or more aggressive. If the UGV knows the location of a mission flag that it needs to visit and does not know where the enemy is, then perhaps it should proceed quickly to the flag, but make some effort to stay out of sight and keep an eye out to make sure the enemy doesn’t ambush it. On the other hand, if the UGV has discovered an enemy and it is in the way of accomplishing the mission, then perhaps the UGV should be more aggressive at closing and attacking – and not worry about staying out of sight.

Through the use of advocates, critics and attitudes, ACTB naturally admits the diversity of goals and constraints that are important for the tactical behavior navigation problem, but in a manner that makes those aspects intuitive to understand. For instance, it is generally easy for a human observing a UGV's performance to be a critic of some specific bad behavior and to be an advocate for some specific desired behaviors. ACTB enables this natural way of thinking about the problem to be reflected directly. As well, a human considering a tactical navigation task will generally re-orient his or her priorities depending on the tactical situation. ACTB enables this natural way of focusing on what is important to be reflected directly. A key benefit of the ACTB design is that new advocates, critics and attitudes may be readily added to reflect new tactical behaviors. As the system is developed, consistent planning errors can be identified and new components added to compensate with minimal disruption to current performance.

3.2. Critics

Nine critics were used in our tactical behavior system. All return evaluations that are greater than or equal to 0, where lower numbers indicate better plans.

The *Traversability* critic exploits terrain knowledge to identify all portions of the path that attempt to travel through known obstacles. It returns a penalty proportional to the distance traveled through obstacles (i.e., we allow a path to cross untraversable terrain but penalize accordingly).

The *Safety* critic exploits knowledge of the last known enemy location and heading and applies a simple model of enemy movement to evaluate whether a given plan puts the UGV in danger by placing it within firing range of the enemy. The enemy is assumed to move in a straight line at maximum speed based on its last known heading. The critic returns a penalty proportional to the distance traveled within danger range of any enemies (i.e., we allow a path to cross dangerously close to enemies, but penalize accordingly).

The *Proximity* critic is a variation on the safety critic that reward plans which quickly move the UGV out of the proximity of a currently observed enemy while preserving the UGV's momentum. While the enemy is visible, commands that move out of the visibility range are rewarded and commands that preserve momentum are rewarded over those that involve significant changes in direction. This critic was needed to accommodate the acceleration and deceleration properties of the vehicle.

The *Time-to-Flag* critic measures the total number of minutes that the UGV takes to travel to all the flags, including time spent stopping and looking. It rewards paths that have a shorter elapsed time.

The *Mission-success* critic measures how many of the remaining flags the vehicle will be visited if the plan is followed. Plans are penalized based on the number of missing flags.

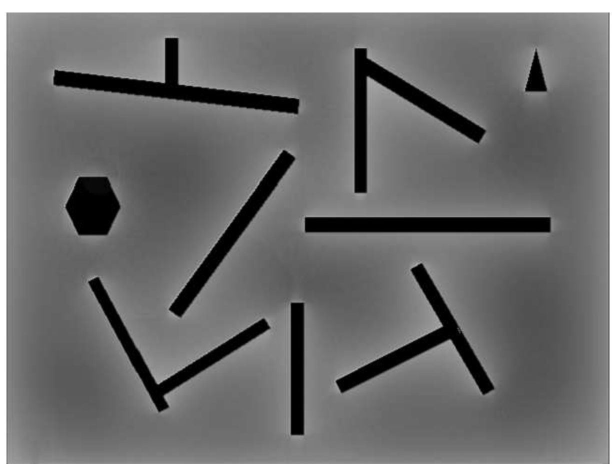
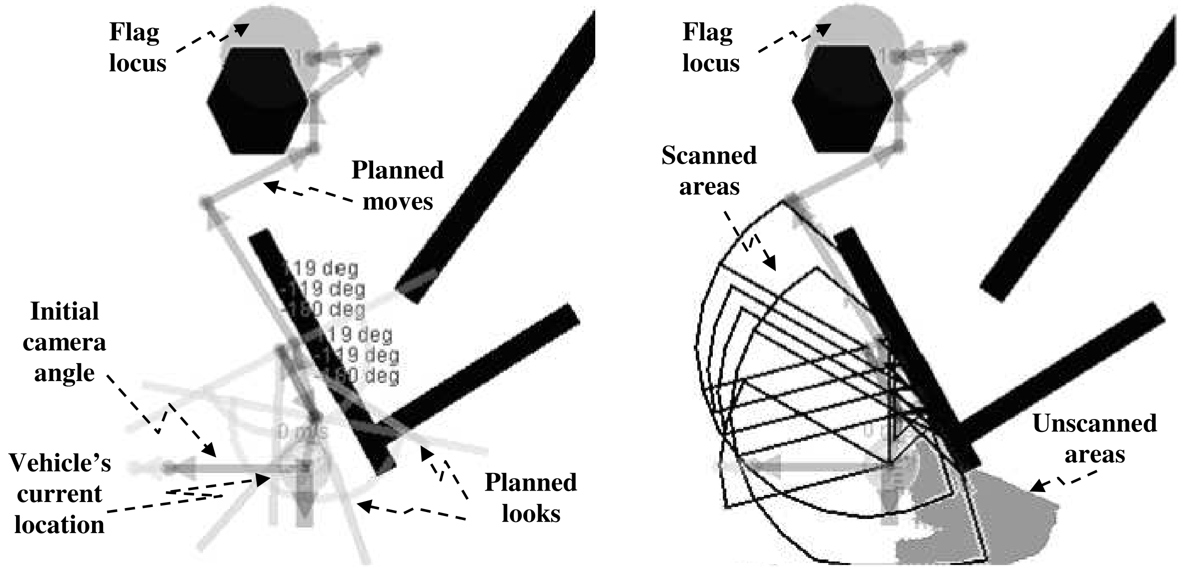


Figure 3: Exposure map

The *Exposure* critic exploits knowledge of the terrain and line-of-sight to estimate the a priori risk that the UGV may be detected by an enemy while executing a given plan. The exposure risk of a location in the terrain is proportional to the size of the area that is co-visible. The critic maintains an *exposure map* of all these risk values, and updates that map as new terrain information is received. Figure 3 illustrates the exposure map for the terrain of Figure 1a; open areas are shaded in gray, where the darker the gray, the more exposed the location. The critic returns a penalty based on the sum of the exposure of all points on a given path.

The *Awareness* critic exploits knowledge of the terrain and where the UGV plans to look to determine the relative risk that the UGV will be visible to an enemy situated in an unscanned area while executing a plan. For each point along the near-term path (up to 120m), the critic computes the percentage of the co-visible area that will be scanned shortly before the vehicle visits that point. The path is penalized based on the total risk over the first 120m of the path. The awareness critic provides a balance to the exposure critic since it takes into account the UGV's scanning actions. It will reward plans that make the UGV look towards areas not recently scanned prior to entering them (e.g., look around a corner first). Figure 4 illustrates the awareness computation made for a given plan (only part of the terrain is shown). The plan, shown in Figure 4a, is comprised of both planned moves (dark gray arrows) and looks (light gray angles). The plan defines a path towards the known flag locus (gray circle). Figure 4b illustrates the computation of the amount of viewable area that is scanned before entering (empty cones) and the amount of viewable area that is not scanned (dark gray areas). The illustrated plan is one with very good awareness.



(a) (b)

Figure 4: Illustration of the computation of the awareness critic

The *Lingering* critic computes the proportion of the near-term plan (the next 200m) that moves the UGV to areas not recently visited. An area is considered visited if it has been scanned recently (within 150 seconds). Plans that move the UGV to new areas are rewarded. Without such a critic, the best option might be to remain in one location since it is safe, scanned and enemy-free. Once all the mission flags have been visited, this critic still encourages the vehicle to move and seek the enemy.

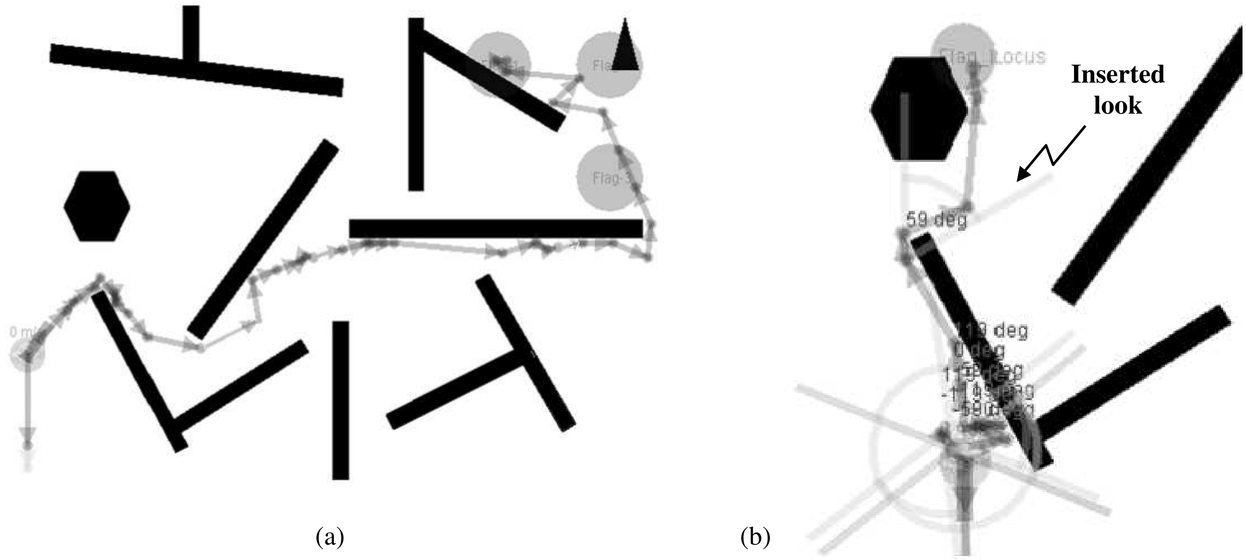
Finally, the *Complexity* critic computes the length of every segment on the path (i.e., between successive waypoints). Every segment is penalized exponentially based on the inverse of its length. The critic returns a value based on the sum over all segments. The result is that plans with many short segments (e.g., 1 or 2 meters) are very heavily penalized.

3.3. Advocates

A number of advocates were used in our tactical behavior system. The key ones are described.

The *Insert-Flag* advocate exploits knowledge of the mission to improve plans that do not visit all remaining flags. The advocate selects the waypoint of the path that is closest to a missing flag and inserts after it a new waypoint that represents visiting the flag. As new flags are discovered, the advocate begins to insert appropriate waypoints to visit them into new plans.

The *Skulk Gradient* advocate exploits knowledge about the exposure risk to detect segments along a path that are more exposed or have been flagged as a hot spot by the exposure critic. It randomly selects one of these sections and moves the endpoints in the direction of lower exposure (based on the exposure gradient at those points). This will generally lead to an improved score on the exposure critic and a changed score on the time-to-flag critic (may be better or worse). Figure 5a illustrates a plan in which the skulk gradient advocate has played a significant role - the path tends to stay near cover (i.e., walls) and avoid open areas.



(a) (b)

Figure 5: Illustration of the effects of skulk gradient and peek advocates

The *Peek* advocate exploits knowledge about the terrain and exposure risk to detect segments along a path where there is a large transition from low risk to high risk of exposure. The advocate randomly selects one of those segments and inserts a “stop and look” command at the point of maximum transition on that segment. This will generally lead to an improved score on the awareness critic. Figure 5b illustrates the result of applying the peek advocate – a look is inserted on the path just prior to end of an obstacle, enabling the UGV to see around the “corner” before reaching it.

The *Go-Around* advocate selects a section of the path that crosses an obstacle or passes close to an enemy and removes that section. A new sequence of waypoints is inserted in its place to form a circular path (either clockwise or counter-clockwise) that connects with the remainder of the path. This advocate will generally lead to improved scores in the traversability, safety and/or proximity critics and a changed score in the time-to-flag critic.

The *Jump-Back* advocate is a variation of the Go-Around advocate, but focuses on adapting a plan to quickly retreat out of the visible range of a (recently discovered) enemy. A section of the path that is in firing range of an enemy is removed. A new sequence of waypoints is inserted that first moves the UGV directly away from the enemy (i.e., in the direction of the straight line between the UGV’s current location and the enemy’s location). The angle of retreat is selected with a small amount of random variability. The sequence then moves the UGV out of visible range and (as in the Go-Around advocate) circles around to connect with the remainder of the path.

The *Jump-Left-or-Right* advocate is similar to the Jump-Back advocate, but focuses on adapting a plan to make a different evasive move. Instead of first moving the UGV directly away from the enemy, the plan first moves the UGV in a direction perpendicular to the enemy (i.e., perpendicular to the straight line between the UGV’s current location and the enemy’s location). The angle of evasion is selected with a small amount of random variability.

The *Wall Trace* advocate randomly selects two waypoints along the path and removes the section of the plan between them. Using knowledge of the terrain, it identifies the nearest obstacle to the first waypoint. It then uses a deterministic algorithm to identify the nearby boundary of the obstacle and creates a new sequence of waypoints that follow the contour of the obstacle. That sequence is inserted back into the path between the original pair of selected waypoints.

Several unbiased mutation operators and one crossover operator were also used to ensure that enough variability was maintained in the population during evolution.

- The *insert-waypoint* operator randomly selects an action (i.e., move or stop-and-look) in the plan and inserts a single waypoint before or after that action. The geographic location of the new waypoint is a small random distance in a random direction from the line connecting its neighboring waypoints.
- The *insert-look* operator randomly selects an action in the plan and inserts a single stop and look action after it. The direction of the look is randomly chosen, but will not overlap with any looks actions already associated with the same waypoint.
- The *remove-section* operator randomly selects two actions in the plan and removes them and all actions between them.
- The *nudge-waypoint* operator randomly selects a waypoint from the plan and modifies its geographical location slightly in a random direction.
- The *remove-look* operator randomly selects a look action on the plan and removes it.
- The *plan-crossover* operator is applied to two parent plans, and performs variable-length one-point crossover between the two plans.

3.4. Attitudes

Four different attitudes were used in our tactical behavior system. Each attitude represented a different set of critics and advocates. Each critic had a default weight value, and an attitude could assign a critic weight equal to, higher or lower than that default to impose an appropriate bias in the tactical planning. With one exception, all advocates were assigned a default probability and that probability was used in all attitudes (i.e., only manipulated the set of advocates, not the probabilities). Table 1 summarizes the relative differences in the critics between attitudes. A critic would either be used with the default weight (medium gray), a low weight (light gray), a high weight (dark gray) or not at all (n/a).

Critic	Attitude			
	Cautious	***Flee***	***Fight***	***Mission-Oriented***
Traversability				
Safety			Low	Low
Proximity			Low	Low
Exposure		n/a	n/a	
Awareness		n/a	n/a	Low
Time-To-Flag		Low	Low	High
Mission-Success		Low	Low	
Lingering	n/a	n/a	n/a	High
Complexity		Low		High

Table 1: Critics used in each attitude and their weights relative to default.

The *Mission-Oriented* attitude was generally the initial attitude, and placed emphasis on achieving the mission objectives quickly by encouraging travel to new regions and allowing some risk. All advocates except Skulk-Gradient were used.

The *Cautious* attitude placed emphasis on stealthy exploration. All advocates except Jump-Back and Jump-Left-or-Right were used. The Go-Around advocate was used, but with a lower probability of selection than default.

The *Flee* attitude placed emphasis on getting away from the enemy rather than accomplishing the mission or planning looks. Very few advocates were used – only the Go-Around, Jump-Back, Jump-Left-or-Right, Insert-Flag, Insert-Waypoint, Nudge-Waypoint, Remove-Section and Plan-Crossover advocates.

The *Fight* attitude placed emphasis on achieving the mission while maintaining maneuverability, with safety as a low factor. Exposure risk was ignored and no looks were planned. The same advocates as in the Flee attitude were used.

The attitude of the system was determined by tactical and environmental events, such as the discovery of an enemy, and recent UGV actions, such as accomplishment of a mission flag. A set of strategic "Rules of Engagement" was defined which determined when the system would change its attitude. The rules are illustrated in Figure 6 using a transition diagram, where each transition is triggered by a specific event. The strategy rules were developed to reflect our expectations that the enemies would be highly aggressive and likely to guard the objectives.

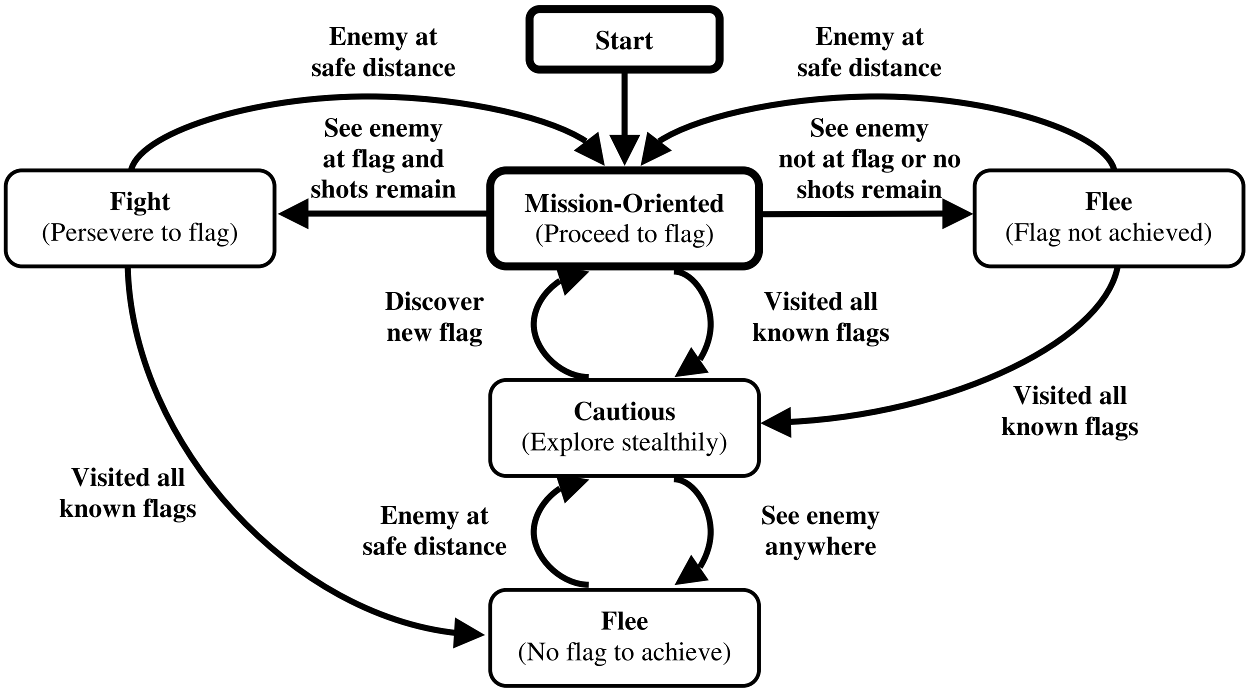


Figure 6: Rules of engagement select the attitude based on the current tactical situation

3.5. Rapid Planning and Response

In addition to attitudes, our tactical behavior system incorporated three main features designed to facilitate effective real-time response: *hot spot* advocates, thrashing control and *autonomic* behaviors.

A key property of most advocates is that they are applied with relatively little bias to any eligible point of mutation or crossover in the plan. This is generally true of both tactical advocates and the traditional genetic operators used in ACTB. For example, the peek advocate may determine a set of eligible points based on the exposure transitions encountered on the path. The choice of which point to adapt is made at random, and thus (in general) a look may be inserted near the end of the path with just as much likelihood as at the beginning of the path. Through evolution, the near-term plan may be improved significantly, but it may take a large number of generations due to a balanced focus on plan elements regardless of whether they are distant or close in time.

Within the current system, we exploited the computations performed by the critics to identify those portions of the paths that were most problematic. Each of these "hot spots" was directly marked on the plan during the evaluation phase. For instance, the traversability critic may determine that a path section crosses a large obstacle area, and then mark it as a "traversability hot spot". Every advocate was provided with the capability to access the hot spot information to quickly identify problem areas in the plan. An advocate could pay attention to any or all types of hot spots as appropriate. For instance, the Go-Around advocate may pay attention to hot spots from the Traversability, Safety and Proximity critics.

In particular, an advocate may pay attention to the first hot spot on the path. Many of the advocates described above were operated in two modes: one in which they performed an unbiased locally-computed selection of where to modify the plan, and one in which they used the hot spots to determine where to modify the plan. The modes were applied with equal likelihood. The net effect was a very rapid application of the advocate (avoiding some redundant computations) and a rapid evolution process for resolving critical problem areas. The unbiased mode was useful in maintaining population variability.

The ACTB planning cycle is very rapid, and new "best plans" potentially may be generated multiple times a second. While improving a plan is desirable in general, it is possible, especially when the situation and/or attitude changes, to generate highly frequent changes which effectively stop the vehicle from moving (e.g., too many changes of direction). To compensate for this thrashing behavior, we implemented the following planning rule. If more than 10 plan changes were made within 10 seconds (within 2 seconds under the Flee and Fight attitudes), then no more plan changes were permitted for another 10 seconds (2 seconds for Flee and Fight). This ban on changes persisted under any attitude transition except to Flee or Fight.

While the ACTB planning cycle is rapid, we determined that it was not an appropriate solution for all vehicle control. Within our tactical behavior system, we integrated several behaviors that were triggered automatically based on certain tactical events and performed autonomically using specialized code. These autonomic behaviors operated in tandem with the planned behaviors, and included:

- *Automated gun slewing*: Under normal circumstances, the gun was always slewed with the camera to ensure that it was aimed at the general area where an enemy would be revealed (and thereby facilitate rapid targeting).
- *Automated targeting and firing*: When an enemy was discovered, the UGV immediately started to aim its gun at the enemy. As soon as the enemy was within firing range (i.e., 80m) and the gun was targeted accurately, a shot would be fired. If only one shot was remaining, then the shot would not be fired until the enemy was within the guaranteed hit range (i.e., 40m). The behavior occurred independently of any planned move or stop-and-look commands. This design followed an "opportunistic" firing strategy.
- *Automated wandering*: If the UGV had accomplished all known flags, it would generate a destination to wander towards. If there were still outstanding flags, the destination would tend to be near the original flag locus. If no flags remained, it would tend to be placed at a distance. Once the wandering destination was reached, a new one would be generated if no new flags had been discovered. This autonomic behavior encouraged the UGV to explore a very wide area when appropriate.

4. RESULTS

During several days in September 2004, our tactical behavior system was tested directly against the systems of two other competing project members. In the test trials, three factors were manipulated: the amount and accuracy of a priori knowledge about the terrain, the pattern of the terrain to be traversed, and the number of vehicles. The vehicles were either presented with a complete initial map of the world or with no information at all. In the former case, the a priori information may have been incorrect - while moving, a vehicle could discover that the actual terrain features differed. In the latter case, each vehicle would have to discover terrain features by scanning with its camera as it moved. In both cases, the actual terrain was fixed (i.e., once an area was viewed, the terrain in that area remained unchanged). The terrains used in the test trials were mostly variations based upon the two terrains illustrated in Figure 1. The variations included the addition of extra obstacles, the removal of entire obstacles, and the partial removal of some obstacles (e.g., a "path" through a "wall"). The number of competing UGVs in a test trial was either two or three (i.e., one friendly vehicle and one or two enemy vehicles controlled by independent tactical behavior software). The simulator also maintained a single *wandering vehicle* that moved randomly through the terrain but never fired. The sole purpose of this vehicle was to provide a moving obstacle to avoid – if another UGV hit the wandering vehicle it would stop. Since the simulator provided the exact identity of all vehicles, the wandering vehicle could never be mistaken for an active enemy.

In each test trial, the flags were randomly located (within a 100m locus) and the initial positions of the competing UGVs were determined randomly such that they were distant from each other and the flag locus. The trial lasted 20 minutes or until only one UGV remained alive. During a trial, all competing UGVs could visit all flags in any order and at any time. However, since the last kill ended the trial, it was possible to be the last standing UGV and still have remaining flags to visit though the 20 minutes had not expired.

Two sets of trials were conducted. In the first set, 60 one-on-one trials were conducted, with each team's vehicle competing 20 times against each opponent (for a total of 40 runs for each team). In the second set, 132 trials were conducted, with each system participating one-on-one against each competitor 36 times and against both competitors 24 times (for a total of 96 runs for each team). In the three-way battles, each vehicle was given 8 shots instead of 5.

The results of the first trial are shown in Table 2. Four types of aggregate data are summarized for each team. The number of flags visited over all trials, the number of enemy kills made, the number of shots that were not fired over all trials, and the number of times the vehicle survived until end of the trial (either due to time out or being the last vehicle alive). In this set of trials, our system performed better than both competitors. We made a large number of kills and were more accurate in our shots (better ratio of shots per kill). We captured a large percentage of the flags (65%) and had a higher survival rate than our opponents (75%).

	# Flags Visited (out of 120)	# Kills (out of 40)	# Shots Left (out of 200)	# Survivals (out of 40)
BBN	78	13	173	30
Competitor 1	27	7	154	26
Competitor 2	62	13	146	23

Table 2: Aggregate results of first set of testing trials

Qualitatively, during the first set of trials, our system performed robustly under all terrain manipulations and got stuck in an unrecoverable position only once. By contrast, both competitors experienced a range of problems and frequently got stuck on obstacles or in degenerate situations (e.g., turning in a circle indefinitely). After the first set of runs, each team was allowed to make some changes to their code. In our case, we adapted our automated wandering behavior to encourage the UGV to search for missing flags more aggressively (i.e., more likely to look close to the flag locus than further away), and slightly increased the minimum distance our vehicle would maintain from obstacles.

The results of the second trial are shown in Tables 3 and 4. In these trials, our system performed very poorly relative to Competitor 2 in terms of our ability to target and shoot the enemy. However, we outperformed both opponents in terms of our ability to capture flags – Table 3 indicates a comparable number of flags captured between our system and Competitor 2, but Table 4 illustrates that in head-to-head trials against Competitor 2, we were able to capture more flags despite being killed 72% of the time.

	# Flags Visited (out of 480)	# Kills (out of 120)	# Shots Left (out of 552)	# Survivals (out of 96)
BBN	182	13	505	48
Competitor 1	32	5	542	71
Competitor 2	193	65	450	86

Table 3: Aggregate results of second set of testing trials

	# Flags Visited (out of 180)	# Kills (out of 36)	# Shots Left (out of 180)	# Survivals (out of 36)
BBN	61	5	158	10
Competitor 2	56	26	138	31

Table 4: Head-to-head testing results between BBN and Competitor 2 in second set of trials

Qualitatively, during the second set of trials, our system again performed excellently and this time never got stuck in an unrecoverable position. It again performed very robustly under all terrain manipulations and all information manipulations (a priori, a priori with changes, no a priori). For example, our UGV frequently exploited "gaps" in "walls" (see Figure 7) and was able to map out unexpected cul-de-sacs that it encountered. The opponent systems were more stable than in the first set of trials, but still frequently got stuck on obstacles or in degenerate situations. Competitor 2 followed a targeting strategy that always kept the gun in a fixed position relative to the vehicle body and turned the actual vehicle instead. Our vehicle, by contrast, would slew the gun directly, even while turning the vehicle. The difference between the gun slew rate (45 degrees/sec) and the vehicle turn rate (60 degrees/sec) made our targeting algorithms much poorer at keeping the gun trained on the enemy. As well, our UGV actively explored the entire terrain space, even after capturing all the flags. This actually led to poor performance on kills since we would frequently discover the enemy, who may have been stuck, and then lose in the ensuing gunfight.

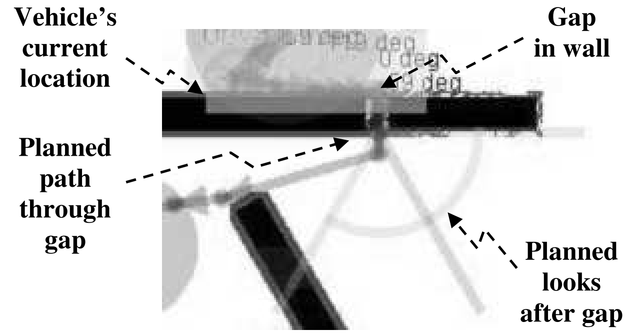


Figure 7: Intelligently exploiting a gap in a wall

In all trials, our system showed a rich set of timely responses and performed in noticeably tactical ways. It did not always make the same decisions when faced with a similar situation, but still demonstrated tactical behaviors appropriate to those situations. The performance was consistent with our high-level rules of engagement, and our autonomic behaviors and planned behaviors supported each other. For example, our UGV would often flee from an enemy, but continue to target and fire at it; a number of kills were obtained in this way.

5. CONCLUSIONS

The development of effective tactical behaviors for unmanned ground vehicles presents a number of challenges. We have developed a rich planning and control architecture that enables us to incorporate a variety of competing tactical behaviors into a single planning system and achieve robust performance under different tactical situations. Our resulting UGV control system was demonstrated to be effective against two other state-of-the-art approaches.

Our Advocates and Critics for Tactical Behaviors (ACTB) approach has been designed to be extensible and meaningful. A key strength of the ACTB approach is that advocates and critics allow insertion of common-sense tactics into the robot with minimal disruption to the current capabilities. Further, the ability to explicitly define rules of engagement and different operating attitudes may prove critical for the deployment of effective UGVs. Just as with human troops, a commander will need to adapt the rules of engagement to ensure that the UGV plays the right role at the right time.

ACKNOWLEDGEMENTS

The work presented in this paper was conducted under subcontract to General Dynamics Robotics Systems (GDRS) under contract #DAAD19-01-C-0066. We wish to thank Nancy Asendorf at GDRS for her technical support on the simulator during development. We would like to acknowledge the contributions of Richard Estrada, David Montana, Richard Lazarus and Stephen Milligan to the ideas presented in this paper. Finally, we wish to thank Amelia Fedyk for her assistance in testing our system during development and Debi Matrona for her assistance in preparing Figure 2.

REFERENCES

1. National Research Council Staff, *Technology Development for Army Unmanned Ground Vehicles*. National Academies Press, Washington, D.C., 2002
2. T. Hussain, D. Montana and G. Vidaver, "Evolution-Based Deliberative Planning for Cooperating Unmanned Ground Vehicles in a Dynamic Environment," *Genetic and Evolutionary Computation - GECCO 2004, Part II. Lecture Notes in Computer Science 3103*. K. Deb et al (Eds.), p. 1017-1029, Springer-Verlag, Berlin, 2004.