

1-1-1994

Dynamic Obstacle Avoidance

Michael A. Craft

Jaime E. Cisneros

Rhonda L. Freeman

Clark R. Karr

Find similar works at: <https://stars.library.ucf.edu/istlibrary>
University of Central Florida Libraries <http://library.ucf.edu>

This Research Report is brought to you for free and open access by the Digital Collections at STARS. It has been accepted for inclusion in Institute for Simulation and Training by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

Recommended Citation

Craft, Michael A.; Cisneros, Jaime E.; Freeman, Rhonda L.; and Karr, Clark R., "Dynamic Obstacle Avoidance" (1994). *Institute for Simulation and Training*. 73.
<https://stars.library.ucf.edu/istlibrary/73>

INSTITUTE FOR SIMULATION AND TRAINING

Contract Number N61339-92-C-0045
December 21, 1994

Dynamic Obstacle Avoidance

Written by
Michael A. Craft, Jaime E. Cisneros, and Clark R. Karr

The logo for the Institute for Simulation and Training (IST), consisting of the lowercase letters 'i', 's', and 't' in a bold, sans-serif font.

Institute for Simulation and Training
3280 Progress Dr.
Orlando, FL 32826

University of Central Florida
Division of Sponsored Research

IST-CR-94-41

Dynamic Obstacle Avoidance

Written by
Michael A. Craft, Jaime E. Cisneros, and Clark R. Karr

December 21, 1994

IST-CR-94-41



Submitted by: Clark Karr



Reviewed by: Rhonda Freeman

Dynamic Obstacle Avoidance

IST-CR-94-41

Contract N61339-92-C-0045
December 21, 1994

Michael A. Craft, Jaime E. Cisneros, and Clark R. Karr

Table of Contents

1. EXECUTIVE SUMMARY	2
2. INTRODUCTION	3
2.1 PURPOSE	3
2.2 BACKGROUND	3
2.2.1 <i>Distributed Interactive Simulation</i>	3
2.2.2 <i>Computer Generated Forces</i>	3
3. DYNAMIC OBSTACLE AVOIDANCE	4
4. GENERAL PATH PLANNING APPROACHES	4
5. THE DOA MODEL	6
5.1 APPLICATION OF THE DOA GRIDS	6
5.2 INTERPRETATIONS OF THE DOA GRID	7
5.2.1 <i>The Neural Net Model</i>	8
5.2.2 <i>The Potential Field Model</i>	8
5.2.3 <i>The Elevation Model</i>	9
5.3 AVOIDING COLLISIONS	10
5.4 LONG RANGE VERSUS IMMEDIATE PLANNING	10
6. THE DOA TESTBED	12
6.1 EXPERIMENTAL PATHS	12
6.2 IMMEDIATE (NEXT STEP) PLANNING ALGORITHMS	12
6.2.1 <i>The Obvious Algorithm</i>	13
6.2.2 <i>The Limited Choice Algorithm</i>	13
6.2.3 <i>Selected Choices Algorithm</i>	13
6.2.4 <i>The Speed Algorithm</i>	14
7. IMPLEMENTING THE DOA MODEL IN THE IST TESTBED	15
7.1 THE MANEUVER TO POINT FSM	15
7.2 MANEUVER CONTROL WITH THE DOA MODEL	15
7.3 THE DOA CONFIGURATION FILE	16
7.4 THE DOA GRID	17
7.5 MARKING THE DOA GRID	18
8. DOA MODEL RESULTS IN THE IST CGF TESTBED	19
8.1 SCENARIOS USED TO TEST THE DOA MODEL	19
8.1.1 <i>Scenario 1: The X-scenario</i>	19
8.1.2 <i>Scenario 2: Head On Collision</i>	25
8.1.3 <i>Scenario 3: Right Angle Collision</i>	30
8.1.4 <i>Scenario 4: Competition for the Bridge</i>	38
8.1.5 <i>Head On Collision On the Bridge</i>	44
9. CONCLUSIONS AND FUTURE WORK	49
10. REFERENCES	50
11. APPENDIX A. DOA CONFIGURATION FILE	52
12. APPENDIX B: DOA CONFIGURATION FILE DATA STRUCTURES	53
13. APPENDIX C. DOA GRID CELL DATA STRUCTURES	54

1. Executive Summary

Techniques to allow simulated entities to avoid static terrain, such as trees, tree lines, canopies, etc., have been in use in Computer Generated Forces systems for many years. Avoidance of objects in motion, "dynamic obstacles", is much more complex. Both of these problems are important for Distributed Interactive Simulation (DIS) environments.

Although simple dynamic obstacle collision avoidance has been implemented in other systems, the resulting behavior is usually less than realistic. The failures in realism may be manifested as inappropriately high speed maneuvering, after-you-after-you locks (entities halting), formation deterioration, and other problems. For those trying to solve the problem well, difficulties arise related to the computation expense.

The Institute for Simulation and Training (IST) has investigated techniques to allow simulated entities to make reasonably intelligent and realistic maneuvers intended to avoid dynamic (and static) objects. The goal has been to find new methods which will yield improved collision avoidance behavior without excessive computational cost. The techniques IST investigated are likely to be of general interest although IST's work has been targeted for the DIS environment.

As a part of this research, a standalone DOA Testbed was developed for testing and evaluating solutions to the DOA problem. IST has implemented and evaluated several techniques on this foundation which without it would seem inapplicable. These range from a heuristic "best guess" method to spline fits.

As a result of this work, IST has developed a novel approach to attack the DIS dynamic obstacle avoidance (DOA) problem by combining two disparate motion planning approaches: potential fields and grid based route planning.

Measures of the realism of the resulting behavior and the execution costs have been developed and applied. Based on these metrics IST is in a position to offer recommendations for dynamic obstacle avoidance with a clear indication of both the resulting behavior and the computational overhead. Further, the foundation developed by IST is ripe for further development by applying additional techniques to the problem.

2. Introduction

2.1 Purpose

This technical report is a deliverable under STRICOM contract N61339-92-C-0045, "Intelligent Autonomous Behavior by Semi-Automated Forces in Distributed Interactive Simulation". It is one of two reports satisfying CDRL A007 "Route Planning Technical Report"; the second report is entitled "Unit Route Planning", IST-CR-94-42.

Its purpose is to report the methodology and results of experimental software development conducted at the Institute for Simulation and Training (IST) under that contract. The goal of the software development was to produce an efficient and effective mechanism whereby entities controlled by Computer Generated Forces systems can avoid collisions with other moving entities within a DIS environment.

2.2 Background

This section provides a brief description of Distributed Interactive Simulation and Computer Generated Forces. It may be skipped by readers familiar with these topics.

2.2.1 Distributed Interactive Simulation

Distributed Interactive Simulation is an architecture for building large-scale simulation models from a set of independent simulator nodes (DIS[1993]). The simulator nodes are linked by a network and communicate via a common network protocol. (The term DIS is also sometimes used to designate a particular network protocol standard; in this document "DIS" refers to the simulation architecture; the DIS protocol standard will be so identified.) In DIS, the simulator nodes each independently simulate the activities of one or more entities in the simulated environment and report their attributes and actions of interest to other simulator nodes via the network. The simulated entities coexist in a common environment (for example, a terrain database) and interact by exchanging network packets (Loper et. al.[1991]). Finally, an important characteristic of DIS simulations is that they are real-time; events in the simulation occur at the same rate as their real-world counterparts.

2.2.2 Computer Generated Forces

DIS environments are designed to provide a simulated battlefield which is often used for training military personnel. In such a battlefield, the trainees need an opposing force against which to train. One technique is to use a computer system that generates and controls multiple simulation entities using software and possibly a human operator. This type of system is known as a semi-automated force (SAF or SAFOR) or a computer generated force (CGF).

A CGF system will use built-in behavior to react autonomously to the battlefield situation or to carry out orders given by its operator. Its behavior may be encoded as algorithms, production rules, formal behavior specifications, or some other form. The intent is for the CGF system's behavior to be autonomous (i.e. not requiring human control) and realistic (i.e. true to doctrine, physics, and human responses) to the greatest possible extent.

Under the sponsorship of ARPA and STRICOM, IST has been conducting research in the area of CGF systems, seeking to increase the realism and autonomy of CGF behavior. A key product of that sponsorship is the IST CGF Testbed. The IST CGF Testbed is a CGF system that provides an environment for testing CGF behavioral control algorithms. It is documented in Danisas et.

al. [1990], Gonzalez et. al.[1990], Petty [1992], Smith et. al. [1992a], and Smith et. al. [1992b]. The research described in this report was performed using the IST CGF Testbed.

3. Dynamic Obstacle Avoidance

Vehicle route planning falls into two broad classes: planning before movement (preplanning) and planning during movement (dynamic planning). Similarly, obstacles fall into two broad classes: moving and stationary (static). The typical approach to routing in CGF systems has been the simplest: preplan considering all obstacles to be stationary (or simply ignoring moving obstacles). The troublesome issue of avoiding collisions with moving vehicles is the topic of this research.

There are several approaches to this problem. First, a route planner can be called repeatedly throughout movement treating the current position as the start and all obstacles as stationary. This approach has the disadvantage of a high computational overhead; frequent replanning of a long route to avoid nearby moving obstacles is inefficient. High frequency, local planning seems a more efficient approach. Second, a route planner could preplan but attempt to predict the locations of the moving obstacles. The prediction can be precise (i.e. a particular path will be followed) or imprecise (i.e. the obstacle is likely to be in an area or volume at a particular time). Unfortunately, precise predictions are invalid whenever obstacles change speed or heading. The imprecise prediction method creates areas in 2D or volumes in 3D where individual obstacles might be. As the time between predictions increases, the prediction becomes less certain and the areas/volumes enlarge. Eventually, the uncertainty of the imprecise prediction overwhelms its utility. High frequency prediction seems to improve both methods.

This report discusses research into Dynamic Obstacle Avoidance (DOA); that is, route planning during movement to avoid moving and stationary obstacles. An approach is presented which combines high frequency, local planning with short range, imprecise prediction of moving obstacles.

4. General Path Planning Approaches

Motion planning with particular emphasis on robot path planning and robot manipulator path planning has seen considerable work; see Hwang et. al. [1992] for a survey. There are four broad categories of path planning approaches: free space analysis, vertex graphs analysis, potential fields, and grid (regular tessellation) based algorithms (Thorpe [1984]). Each approach has strengths and weaknesses.

In the free space approach, only the space not blocked or occupied by obstacles is represented. For example, representing the center of movement corridors with Voronoi diagrams (Roos [1992]) is a free space approach. Although Voronoi diagrams are efficient representations, they and other free space approaches have some deficiencies. First, they tend to generate unrealistic paths. Paths derived from Voronoi diagrams follow the center of corridors while paths derived from visibility graphs (Mitchell [1988]) clip the edges of obstacles. Second, the width and trafficability of corridors are typically ignored. Third, distance is generally the only factor considered in choosing the optimal path.

In the vertex graph approach, only the endpoints (vertices) of possible path segments are represented (Mitchell [1988]). This approach is suitable for spaces that have sufficient obstacles to determine the endpoints; determining the vertices in "open" terrain is difficult. In addition, representing only path vertices creates two other difficulties. First, trafficability over the path segments is not represented; routes segments between arbitrary vertices are typically "open" or

"blocked". Second, factors other than distance can not be included in evaluating possible routes. In the military simulation domain, concealment and cover are important factors in route planning.

In the potential field approach, the goal (destination) is represented as an "attractor", obstacles are represented by "repellers", and the vehicle is pulled toward the goal while being repelled from the obstacles (NASA [1991]). There are two difficulties with the potential field approach. First, the vehicles can be attracted into box canyons from which they can not escape (Michell [1988]). Second, some elements of the terrain may simultaneously attract and repel. For example, an obstacle to movement, a repeller, may create an area of concealment. A vehicle should be attracted to the obstacle for concealment while being repelled from the obstacle creating the "visibility shadow".

In the regular grid approach, a grid overlays the terrain, terrain features are abstracted into the grid, and the grid rather than the terrain is analyzed. Each grid cell is typically marked as "open" or "blocked". Quadtrees are an example of the regular grid approach (Mitchell [1988]). Grid routes are converted into terrain routes typically by adding the z-coordinate to the xy-coordinates in the grid route. This approach's advantage is to simplify the analysis but has two disadvantages. First, "jagged" paths are produced because movement out of a grid cell is restricted to four (or eight) directions corresponding to the four neighboring cells (eight if diagonal moves are allowed). Second, the granularity (size of the grid cells) determines the smallest "opening" that can be identified. If the granularity is too large, small openings in obstacles (e.g. bridges over rivers) are lost. To capture the small openings, a small granularity is required which increases the computational expense of the analysis.

5. The DOA Model

The DOA Model combines the potential field and regular grid approaches into a single mechanism for avoiding moving and static obstacles during movement along a predetermined route. The Neural Net paradigm (Glasius [1994] and Hertz [1992]), was the genesis for this work, although the traditional AI hill climbing is another excellent metaphor (Charniak and McDermott [1987]).

A rectangular area is selected containing the vehicle and a (possibly interpolated) target destination. A grid is overlaid on the selected area. The cells in the grid fall into several classes:

<u>barrier</u>	a static obstacle the vehicle cannot cross and which does not change position, such as a river,
<u>target</u>	represents the position the vehicle wants to reach,
<u>vehicle</u>	represents the position of the vehicle doing the routing,
<u>obstacle</u>	corresponding to a position, or projected position, for a moving object, and
<u>open</u>	all other cells.

For example:

	A	B	C	D	E	F
1			B			
2			B			
3			B			
4						T
5	V			O		
6				O		
7						
8						

Figure 5-1 : DOA Grid.

In this example, the vehicle has no barriers (B) between it and its target, but a moving obstacle (O) is, at least temporarily, blocking a line from the vehicle (V) to the target (T).

The grids generated for the purposes of carrying DOA computations, as illustrated here and in the next sections, will be referred to as "DOA Grids."

5.1 Application of the DOA Grids

When a vehicle is to route to a destination, a long term planner selects a collection of intermediate points (route points) to be used in traveling to the target destination. This analysis is based on static terrain features. The DOA logic is applied periodically during route transversal. The vehicle's speed is used to select the frequency of DOA analysis (such that the vehicle will move approximately 1 cell before the next DOA analysis), and the grid is laid out so that the vehicle is on an edge and the target, or an interpolated target, falls on the opposite edge.

For illustration, this figure 5-2 shows the next route point (R), the vehicle (V), the interpolated target position used for the DOA calculations (T), and the DOA grid:

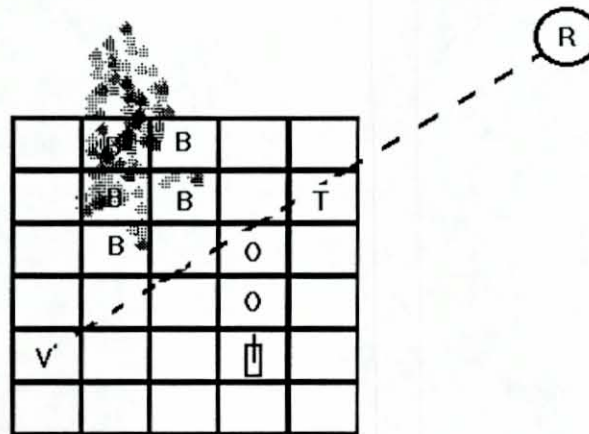


Figure 5-2 : DOA Grid with Target.

A target destination on the DOA grid may be inside the grid if the vehicle is approaching a route point.

5.2 Interpretations of the DOA Grid

The grid can be viewed as a network of cell connections; for example, consider a subset of the full graph showing a path from the vehicle in figure 5-1 to its target destination:

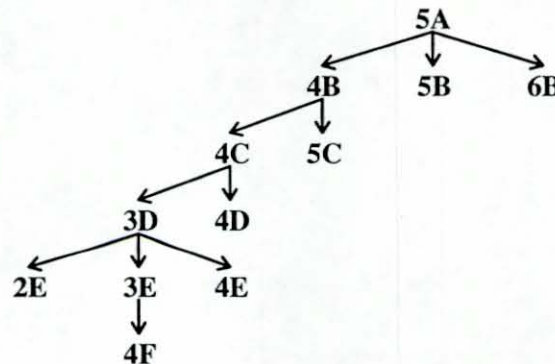


Figure 5-3 : Path as Tree Transversal.

It is the directed graph view that triggered the IST notion of looking for parallels between this problem and the neural net and potential field approaches. Traditionally, such a graph would be used to reduce the problem to that of a search avoiding "danger areas" (Cortes-Rello and Golshani [1990]). However, the search metaphor inevitably leads one to treat grid cells as open (passable) or closed (blocked) (Mitchell [1988]); something more subtle is needed. While cell 4D is not blocked, its proximity to a moving obstacle should make 4D less attractive than 3D (hence the choice of 3D rather than 4D in figure 5-1). It is not obvious how to take such things into account when viewing this as a simple search problem.

5.2.1 The Neural Net Model

If we view the cells as elements in a Hopfield neural net (Hertz [1992]), the cells (particularly 3D and 4D in the example at hand) should be influenced by their adjoining cells. In neural net theory, the "energy" of a cell is calculated in two steps. First, the contributions of neighboring cells are summed:

$$h_j = \sum w_{ij}h_i$$

Where h_j represents the energy of a cell j , w_{ij} represents the "connection weight" from cell i to cell j , and h_i represents the energy of cell i . The second step involves "clamping" h_j within a 0/1 or ± 1 range with an "activation" function, $g(h)$. Neural net theory provides two frequently used activation functions: a sigmoid function and a hyperbolic function. The sigmoid function is:

$$g(h) = \frac{1}{1 + \exp(-2Bh)}$$

The hyperbolic function is:

$$g(h) = \tanh(Bh)$$

In both equations, B is a "steepness" parameter which governs the rate of change of h .

Glasius [1994] suggests that a linear activation function, $g(x) = Kx$, is adequate for propagating the obstacle values within the net (where K is a constant in the range $[0, 1.25]$). In this research, the linear activation function was used and the range expanded to $[0, 10,000]$. Understandably, large K values tended to magnify the effects of obstacles.

The DOA Grid can be viewed as a neural net with connection weights held at 1. Thus, only cell values are manipulated.

5.2.2 The Potential Field Model

In the DOA Model, the grid and its obstacles are manipulated via the potential field metaphor. The DOA algorithm assigns the target cell a negative (attractive) potential, the barrier cells potentials of 0, and the obstacles cells positive (repulsive) potentials. Neural Net mechanics propagate the potentials throughout the grid. Thus, a cell's value represents the combined attractive and the repulsive potentials of the target, barriers, and obstacles. For simplicity, one can view these values as temperatures on a uniform sheet (the grid) and the vehicle as an object trying to negotiate the sheet to reach the coolest point. With the problem as described, we now have a set hot and cold points.

As stated, the problem still appears to be a simple search: some points are forbidden, there is a start and a finish location. However, if we view the sheet as heat conducting, and the obstacles as points where heat is being applied (perhaps with a soldering gun) and the target as a point being cooled, it becomes clear how one cell can influence those around it. The key now is to let time pass so the effects of the heating and cooling can spread. After some time we might have a new situation such as:

	A	B	C	D	E	F
1			B			
2			B			
3			B			
4						T
5	V			O		
6				O		
7						
8						

Figure 5-4 : Heated DOA Grid.

The shaded squares indicate an elevated temperature because the obstacles have heated the cells next to them. Eventually, those cells could warm those next to them, and so on. Some cells (such as 5C) are adjacent to 2 obstacle cells and so are heated from two sources, others, such as 4E are heated on one side and cooled on the other. Rather than attempt to apply any realistic temperature model, discrete time steps are taken and at each step a cell's new temperature is a weighted average of its own temperature and the temperature of the surrounding (or, if viewed as a graph, the connected) cells' temperatures. At each iteration, a new set of cells are affected by obstacle cells and cells which had already been adjusted are re-adjusted.

The resulting grid captures a great deal of information and is the basis for the algorithms developed. A cell containing an obstacle effects surrounding cells and the resulting temperature gradient is an imprecise measure of the probability the vehicle will move there. A temperature based search would now select 3D rather than 4D. It is quite easy now for people to express opinions with a glance at the grid. The path already described does not go too much out of the way but avoids 1 "hot" square. A path starting out due South East could stay on cool squares for the entire trip, albeit by taking a longer trip.

A central premise of this work is that a grid can be constructed, smoothed (which means carrying out several iterations to allow temperatures to spread), and an algorithm to select velocity applied very rapidly.

5.2.3 The Elevation Model

Before pursuing other issues, it is worth describing another metaphor for this model. The obstacles are viewed as elevated, the target is depressed, the vehicle seeks the lowest point on the grid (barriers are neither raised nor lowered, they are always at 0 elevation). Using this approach, the initial grid is flat but for some plateaus (obstacles) and a pit (the target). The smoothing process causes the ground (cells) around the poles and pits to erupt or sink. Continued iterations cause the grid to approach terrain sloping smoothly from the vehicle toward the goal and around the "hills" surrounding the obstacles and barriers. The path to be taken avoids high spots (and is blocked from obstacle and barrier squares) and runs down to the low areas. Visually, the metaphor is that of a marble rolling downhill towards a hole.

The grid approach is prone to all of the classic hill climbing problems. Our view is "upside down", we can get trapped in local minimum rather than a local maximum; in any case, local extremes

are a problem (Charniak and McDermott [1987]). To some extent, these problems are mitigated by the fact that there is only one low point at the outset; if a cell is depressed, there must be a clear path from that cell to the target because it was influenced by the target. In general, a lower point is either further from obstacles or closer to the target. This is not to say extrema problems cannot happen; a cell surrounded by obstacles at a distance will have a lower elevation than its neighbors.

While the temperature metaphor was useful for explaining the potential gradient and the smoothing process, the hill metaphor will dominate the remainder of the discussion.

5.3 Avoiding Collisions

To avoid colliding with a moving object, two options exist: change speed or direction. In real life a speed change could involve an increase in speed, but for our work we use only deceleration from the vehicle's desired speed.

The DOA algorithms (see Section 5.4 and Section 6.2) look at cells' values both for direction (smaller values generally indicate more desirable paths) and speed (smaller values indicate higher speeds can safely be used). After the direction is selected, the relative elevation of the cell being entered is used to select a speed. If the cell's value is less than zero, a clear path to the target is at hand and the vehicle moves at full speed. A relatively large elevation indicates the cell is near obstacles and lower speeds are selected.

5.4 Long Range versus Immediate Planning

Static routing (routing when no dynamic obstacles are considered) is a long range process. What direction to travel for the next few seconds is completely dependent on the long term goal. It is easy to construct examples where an error in our "next" step will make it difficult or impossible to reach our goal:

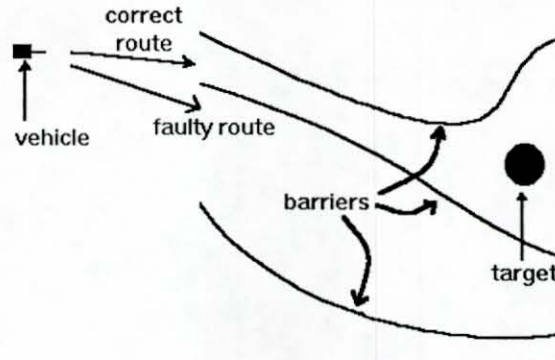


Figure 5-5 : Long Route Planning.

Given arbitrary computation time, long term planning can be used for DOA problems by treating the moving obstacles as fixed for the purpose of computation, and re-computing frequently with updated obstacle positions. Unfortunately, long term planning probably requires more effort than can be expended for rapid speed and course adjustments. Immediate planning to avoid collisions with moving obstacles is used to make a "snap decision" on what to do next without a full analysis of the situation (in particular, without guaranteeing that the target can be reached from the new position).

Both methods can be, and have been, applied to the DOA grids. The A* search algorithm (Winston [1992]), is a efficient long range search algorithm. In the DOA Model, A* treats the

grid as a terrain map where, essentially, the distance traveled is minimized (taking the elevated areas into account by charging extra distance for crossing elevated areas).

The other algorithms tested (see Section 6.2) are of the immediate planning type. Only the cells adjacent to the vehicle are considered in making course and speed adjustments. The danger to such an approach, of course, is that the vehicle could become trapped or cycle. These concerns are greatly mitigated by the way the method is applied. A clear path is known to exist (at least the first time a grid is built), obstacles will eventually move out of the way (or be passed), and the target position is re-computed (by interpolation) each time a grid is built.

6. The DOA Testbed

In order to better understand the DOA Model prior to implementing it within the IST CGF Testbed, a stand alone DOA Testbed was created. The DOA Testbed allows experimentation with DOA approaches without the complexities of terrain navigation and vehicle dynamics within a DIS environment. Although the DOA analysis was moved to the Testbed so simulated vehicle behavior could be directly examined, the fundamental work and algorithm selection was done in this stand alone DOA Testbed.

The DOA stand alone test bed uses the techniques described to produce a grid based on a configuration file. Here is a sample DOA Grid from the DOA Testbed:

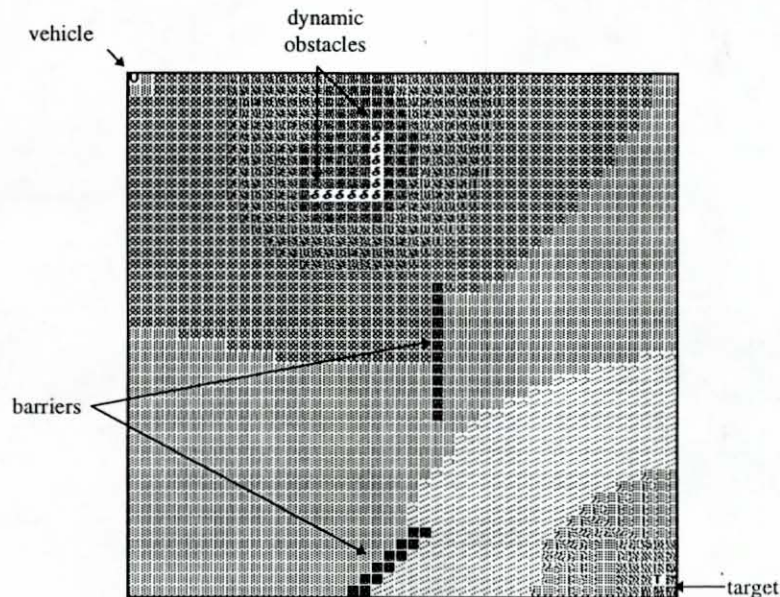


Figure 6-1 : DOA Testbed Screen.

6.1 Experimental Paths

In order to examine the quality of the path/speed selections, the DOA Testbed uses the algorithm under test repeatedly until the vehicle reaches the target. The vehicle moves across the DOA Testbed grid reapplying the algorithm under test each step. This is not completely realistic, as each time the vehicle moves the grid needs to be re-evaluated because obstacles will move and be passed and new obstacles will appear. However, this simplification was introduced because it was found that algorithms that do a good job when applied in this manner will also do a good job in the dynamic situation (when the grid is “refreshed” frequently).

Applying the algorithms to new grids after each step (a DOA Testbed option) has confirmed that algorithms that work well in finding a total path through repeated application to a fixed grid do a good job when applied a step at a time with repeated grid rebuilding.

6.2 Immediate (Next Step) Planning Algorithms

Several immediate planning algorithms were examined to perform the DOA analysis. Except for the **speed** algorithm, the path and speed were determined independently. The non-speed

algorithms select a target cell and then a speed is selected as a function of the target cell's value. If the value is negative a clear path to the target is presumed to exist and so the full specified speed is used by the vehicle. If the cell has a positive value, the vehicle's speed is reduced.

In the discussion of the algorithms, a couple of points should be borne in mind. Whenever the discussion refers to selecting an adjacent cell the intent is always that a cell is selected to which the vehicle can legally move. This eliminates the selection of cells which contain an obstacle. Also, more than one algorithm discusses selecting the cell closest to the target. It is easy to see that in some cases there may be a tie for the closest, as in:

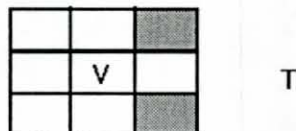


Figure 6-2 : Nearest Grid Cell.

The two shaded cells in Figure 6-2 are equidistant from T. Unless otherwise stated, a random choice is made in the case of a tie.

6.2.1 The Obvious Algorithm

The obvious way to select a path is simply to look at the cells surrounding the vehicle and move in the direction of the lowest valued cell. This approach has a serious problem. If there is an obstacle between the vehicle and target the vehicle will pass widely around the obstacle. Increasing the number of iterations in propagating cell values worsens this problem. With many iterations, the obstacles repulsive potential eventually reaches to the edge of the grid forcing the vehicle to travel along the edges of the grid. One way to avoid this is to put imaginary obstacles behind and along the sides of the grid (relative to the target) to compensate for the centralized obstacles and thus encourage the vehicle in the correct direction.

6.2.2 The Limited Choice Algorithm

To avoid the problems with the Obvious Algorithm, and to avoid introducing non-existent obstacles along the periphery of the grid, the Limited Choice Algorithm forces the vehicle to move in, approximately, the direction of the target (unless barriers force it away). Of the cells adjacent to the vehicle to which the vehicle could legally move, the closest to the target is determined and its immediate neighbors are taken as the set of cells from which a selection is made (two cells may tie for the closest, in which case these two and the cell between them are taken as the set to consider). Of the set considered, the cell with the lowest value is used (in case of tie, use the one closer to the target, if a tie still exists use either one).

6.2.3 Selected Choices Algorithm

To avoid the problems with the Obvious Algorithm, and to avoid introducing phony obstacles, the Selected Choice Algorithm does not allow the vehicle to move directly away from the target (unless barriers force it away).

If any adjacent cells have negative values, the best of these is used; the proximity of a cell with a negative value indicates a clear path to the target is available.

If all of the adjacent cells have non-negative values, the closest to the target is determined and its immediate neighbors are taken as the set of cells from which a selection is made (two cells

may tie for the closest, in which case these two and the cell between them are taken as the set to consider). Of the set considered, the cell with the lowest value is used (in case of tie, use the one closer to the target, if a tie still exists use either one).

6.2.4 The Speed Algorithm

This algorithm selects a speed first and only changes direction (from a direct route to the target) if speed is reduced by 50% or more. The idea is that the usual behavior is simply to slow down to let the obstacles pass by, but if severe braking becomes necessary (perhaps an obstacle is very near by) steering correction must also come into play. The steering algorithm used is the Limited Choice Algorithm.

Speed reduction makes sense as formations can be maintained during speed adjustments, but often break down during unanticipated maneuvers. Further, when people approach a moving obstruction the vast majority will reduce speed.

7. Implementing the DOA Model in the IST Testbed

To test the DOA Model within a DIS environment, the relevant portions of the DOA Testbed were transferred to the IST CGF Testbed. Within the CGF Testbed, the maneuver control mechanism was modified to adjust maneuvering based on the DOA Model.

Behavior control within the CGF Testbed is implemented through a code structuring technique based on Finite State Machines (FSMs) (Smith et. al. [1992a]). An FSM manages task resources and scheduling in a manner similar to that of a process in a multitasking operating system. It isolates and protects its state information much as an object does in an object oriented programming environment.

An FSM encoding the DOA Model was added to the CGF Testbed. This FSM schedules itself based on the speed of the vehicle and awakens the ManeuverToPoint FSM (see below) to make suggested changes in vehicle speed and direction.

7.1 *The ManeuverToPoint FSM*

Within the CGF Testbed, routes are piecewise linear curves, represented by a list of points. These routes are generated by a vehicle level route planner which plans route around static obstacles. Route following causes entities to maneuver towards the first route point on its route point list. As each route point is reached, it is removed from the route point list, causing the vehicle to move towards the next point on the route.

Route following is implemented in the ManeuverToPoint FSM. The states within the ManeuverToPoint FSM set requested values (such as, requested speed, requested turn) which are used by the entity's dynamics process. The ManeuverToPoint FSM maneuvers the entity so that it passes near each route point and comes to a stop near the last route point on the route point list.

7.2 *Maneuver control with the DOA Model*

The DOA FSM is started for a vehicle when the vehicle has generated a route and is beginning to move. The DOA FSM is terminated when the vehicle is at its destination.

The DOA FSM performs a "snapshot" analysis of the situation, makes recommendations for speed and heading, and schedules itself to be repeated in the near future. The scheduled time is proportional to the speed of the vehicle so that the vehicle moves less than the width of a grid cell between DOA analyses.

The DOA algorithm is:

1. Fill the DOA grid. Grid cells are marked with obstacles (moving vehicles), barriers (stationary vehicles, static objects), the interpolated target location, the vehicle's location, and information about the terrain surrounding the vehicle.
2. Propagate cell values within the grid, as described in Section 5.2.1.
3. Apply the algorithm specified in the configuration file. A suggested speed and heading are produced.
4. Change the vehicle's speed and heading:
 - a. If the nearest vehicle is less than 20 meters away, slow to a stop (i.e. halt).
 - b. If the nearest vehicle is between 20 and 30 meters away and to the right, slow to a stop.
 - c. If the vehicle has been halted and should remain halted, backup.

- d. If the vehicle has been halted and doesn't need to remain halted, resume movement.
- e. If the suggested speed is less than a threshold, maneuver along the suggested heading with the suggested speed.
- f. Otherwise, use the suggested speed heading directly at the target.

Steps 1-3 are straight forward. Step 4 makes the final maneuvering decisions. Steps 4a-d encode rules for resolving imminent collision situations. Rule 4a causes the vehicle to begin abrupt braking if a collision seems imminent. Rule 4b encodes a "yield to the right" rule of the road. 4c allows the vehicle to backup out of deadlock situations. 4d permits the vehicle to resume movement after halting and perhaps backing up. Rules 4e and 4f accept the DOA Model suggestions. 4e causes the vehicle to follow the speed and heading suggestions while 4f accepts only the speed suggestion.

In summary, the DOA algorithm causes vehicles to avoid collisions by slowing first (rule 4f), then slowing and steering away from obstacles (rule 4e), then stopping (rules 4a and 4b), and finally backing up (rule 4c). Of course only the maneuvering that is necessary to avoid collisions is performed; e.g. if slowing resolves the problem then steering, halting, and backing up are not utilized.

Runtime parameters to the DOA Model are:

- **Target location:** Specifies the grid cell of the target. This is generally interpolated from the next route point.
- **Vehicle location:** Specifies the grid cell of the vehicle.
- **Next route point:** Saves the terrain coordinates of the route point to which the vehicle is moving. Maneuvering to avoid moving objects changes the first route point on the route point list. This location is saved for later restoration.
- **Real speed:** The vehicle's speed.

The DOA Model produces:

- **DOA speed:** The speed suggested by the DOA Model. This value replaces the requested speed of the vehicle and it is calculated by:

$$\left(1 - \frac{\text{cell elevation}}{\text{speed coefficient} \times \text{minimum elevation}} \right) \times \text{real speed}$$

- **DOA heading:** A new route point suggested by the DOA Model.

7.3 The DOA configuration file.

Initializing parameters for the DOA Model are read from a configuration file. An example of the DOA configuration file can be found in Appendix A.

Upon initializing the IST Testbed, the values in the DOA configuration file are saved into a static record within the Testbed (see appendix B). This record provides the information to each vehicle as it begins to move.

Initializing parameters for the DOA Model are:

- **Algorithm:** Specifies the DOA algorithm to be used. One of A*, Limited Choice, Selected Choice, and Speed.
- **Limit of iterations to propagate the influence of cells on the DOA grid:** Following the view of cells as elements in a neural net, each cell is influenced by its neighbors. To propagate a cell's influence, the DOA algorithm iterates through the DOA grid a number of times. The more iterations, the further each cell's influence is spread but the greater the computation.
- **The cell size:** Specifies the size of each grid cell. For these experiments, the cell size was set to 10 meters.
- **Grid size:** Specifies the size of the grid. For the experiments, the grid size was set to 20 by 10 resulting in a 200x100 meter grid in front of the vehicle.
- **Temperatures for the vehicle, its target, obstacles, and barriers:** Specifies the "temperature/elevation/potential" values for target locations, obstacles, barriers, and vehicle location. Usually, a negative number is used for the target (cooler temperature/negative elevation). Positive values are used for obstacles and barriers. Values for obstacles and barriers should follow: barrier value \leq obstacle value. Zero is the most convenient value for the vehicle.
- **Speed coefficient:** Specifies how fast to increase or decrease speed.
- **Grid cell lookahead:** Specifies the number of cells in front of a moving obstacle to mark in the DOA grid. This extends a obstacle in the direction the obstacle is moving.
- **Activation constant:** Specifies a "propagation" multiplier (see Section 5.2.1). The elevation of a cell is the average of the product of surrounding cells' elevations and the propagation multiplier. This multiplier increases/decreases the affect each cell has on its neighbors.

7.4 The DOA grid

The following is a snap shot of three M1s. Only Vehicle 1 is moving on the terrain and using DOA to maneuver around obstacles (Vehicle 2 and Vehicle 3) and a barrier (a river).

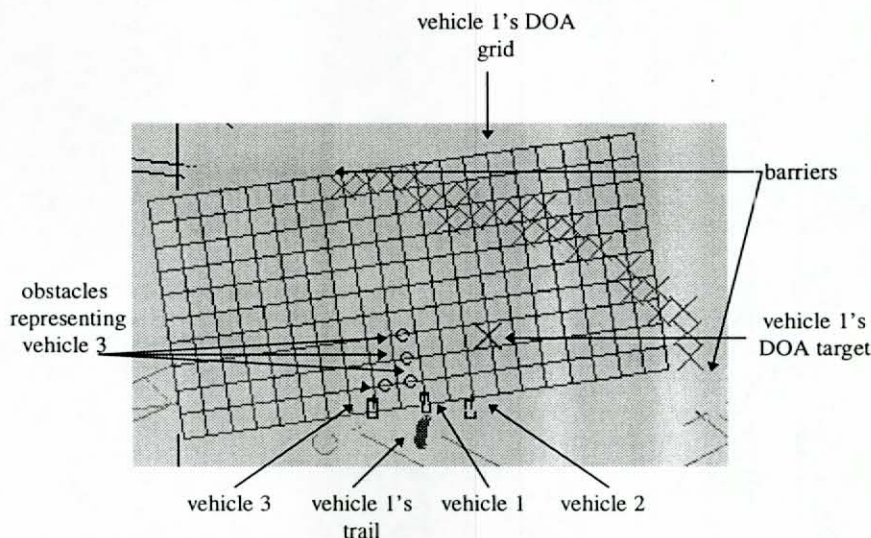


Figure 7-1 : DOA Grid in CGF Testbed.

Notice that Vehicle 3's projected position creates obstacles in front of Vehicle 1. The DOA Model has suggested a target location for Vehicle 1 to the right so as to avoid Vehicle 3. Vehicle 2 is behind and to the right of Vehicle 1 and does not influence Vehicle 1's DOA grid.

The DOA grid is a rectangular array of cells, each contains:

1. Contents of a cell:
 - EMPTY: If it contains nothing but open terrain,
 - DOA_VEHICLE: If it contains the vehicle for which the DOA analysis is being performed,
 - TARGET: If it contains the target location to which the DOA_VEHICLE is trying to reach,
 - OBSTACLE: If it contains a moving vehicle or a moving vehicle's projected position, or
 - BARRIER: If it contains a stationary object or barrier; e.g. a river.
2. Value of the cell:
 - The propagation of cell values within the grid if it is an EMPTY cell, or
 - The fixed value used for DOA_VEHICLE, TARGET, OBSTACLE or BARRIER.

The DOA grid is placed in front of the vehicle oriented along the vehicle's heading. The vehicle is located at the bottom center of the DOA grid. This arrangement allows the vehicle to detect and avoid obstacles and barriers on its forward path. Passed obstacles and barriers drop off the vehicle's DOA grid, and stop influencing its maneuvering.

7.5 Marking the DOA grid

Marking the DOA grid consists of four steps. First, using information from the Terrain Data Base, the locations of all BARRIERS are marked. Second, the locations and projected locations of all OBSTACLES are marked. Third, the TARGET location is marked by interpolating the current location of the vehicle with its next route point. Fourth, the location of the vehicle on the DOA grid is marked.

After the grid is marked, the elevation/temperature/potential values are propagated throughout the grid (see Section 5.2). The resulting DOA grid has:

1. Cells close to an obstacle or barrier have high elevations and
2. Cells close to the target by have low elevations.

The marked and value propagated DOA grid is analyzed by the DOA algorithm specified in the configuration file (see Section 7.3). The vehicle's speed and heading are adjusted based on the results of the analysis (see Section 7.2).

8. DOA Model Results in the IST CGF Testbed

Experiments with the DOA Testbed revealed that analyzing the DOA grid with the A* algorithm (step 3 in Section 7.2) gave the most stable and realistic results. The following are the results of five different scenarios using A* based DOA analysis in the IST CGF Testbed.

8.1 Scenarios used to test the DOA Model

Five scenarios were developed to test the DOA Model: X-scenario (two vehicles moving in a 315 and 225 degree collision course), Head On Collision scenario (two vehicles moving in a head on collision course), Right Angle Collision scenario (two vehicles moving in a 0 and 270 degree collision course), Competition for the Bridge scenario (three vehicles competing to cross a bridge), and Head On Collision On the Bridge scenario (two vehicles moving in a head on collision course on a bridge).

For clarity, the vehicles and their trails are shown in the figures; the DOA grids are not shown.

8.1.1 Scenario 1: The X-scenario.

The following series of pictures show two vehicles crossing fordable water from opposite banks of the river.

8.1.1.1 Start.

Figure 8.1.1.1 is a snap shot of the beginning of the scenario.

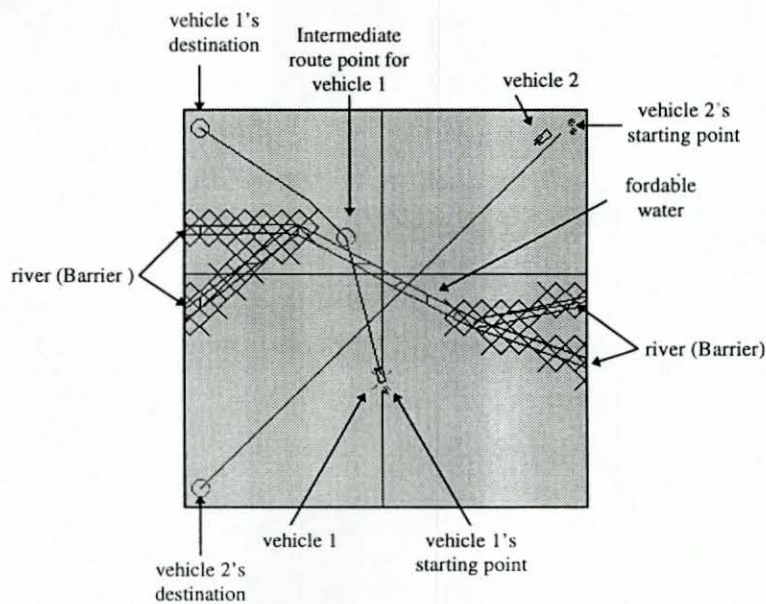


Figure 8.1.1.1 - X-Scenario: Start.

Description: Vehicle 1's route consists of three straight line segments. The first intermediate route point and destination are marked with circles. Vehicle 2's route consists of a straight line. Vehicle 2 begins moving first.

8.1.1.2 Quarter of a way through Vehicle 2's route.

Figure 8.1.1.2 is a snap shot of the X-Scenario when Vehicle 2 is one fourth of the way through its route. The small circles (Vehicle 2) or X's (Vehicle 1) behind a vehicle are its "trail". The spacing of the trail marks give an indicate of the vehicle's speed. This figure shows that Vehicle 2 recently began decelerating to avoid Vehicle 1. Because of the delay in Vehicle 1 starting its movement, it began avoiding Vehicle 2 when it began moving.

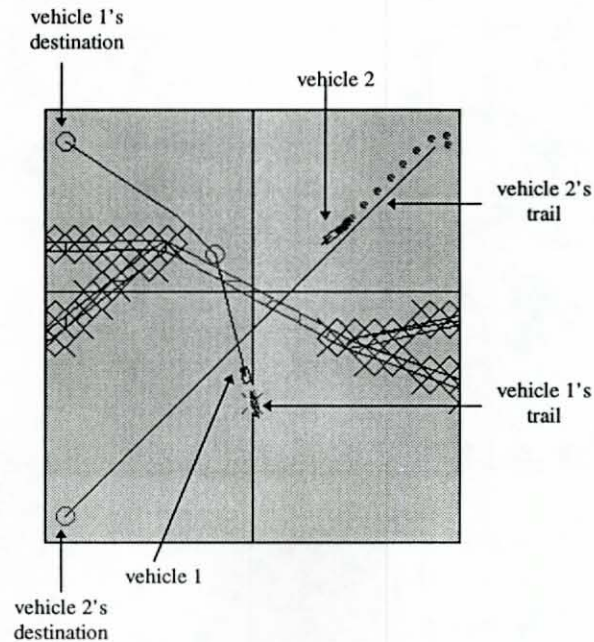


Figure 8.1.1.2 - X-Scenario: One Fourth Complete.

Description: The vehicles have started to slow down (see the dense trail that both vehicles are starting to leave). Vehicle 2 is beginning to turn away from Vehicle 1, in an attempt to avoid a collision.

8.1.1.3 *Halfway through the X-scenario.*

Figure 8.1.1.3 is a snap shot taken halfway through the scenario.

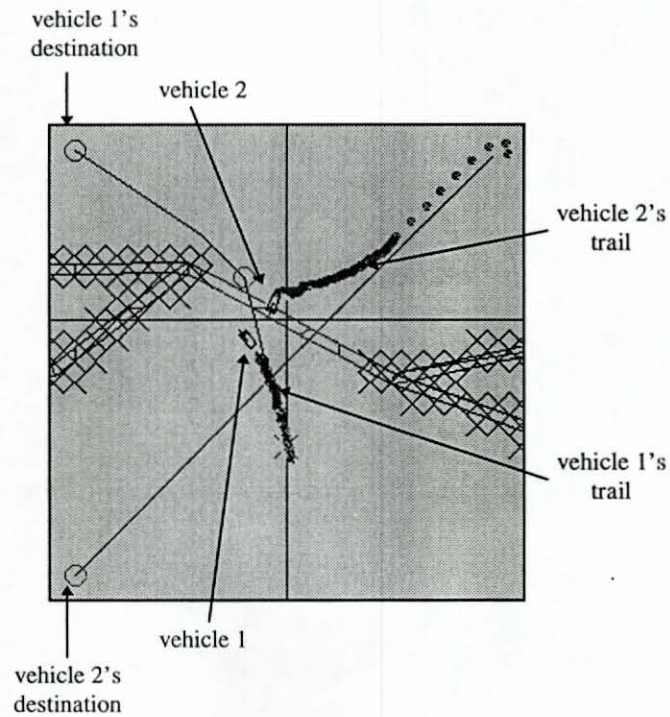


Figure 8.1.1.3 - X-Scenario: One Half Complete.

Description: Vehicle 1 has slowed and turned to its left to avoid Vehicle 2. Initially, Vehicle 2 had turned to its right to avoid Vehicle 1 (see previous figure). Now it is turning left to go behind Vehicle 1 and, thereby, let Vehicle 1 pass in front.

8.1.1.4 After avoiding each other.

Figure 8.1.1.4 is a snap shot of the vehicles after they have avoided each other.

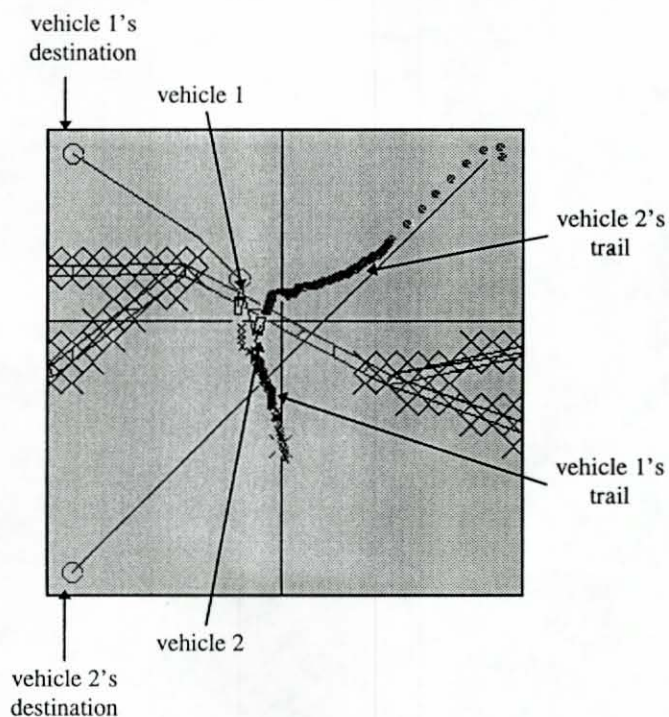


Figure 8.1.1.4 - X-Scenario: Vehicle Avoidance.

Description: Vehicle 1 has just passed Vehicle 2 and is starting to speed up (its trail is becoming sparser). Vehicle 1 is accelerating because Vehicle 2 has left Vehicle 1's DOA grid. On the other hand, Vehicle 2 is still moving slowly because Vehicle 1 is on Vehicle 2's DOA grid.

8.1.1.5 *Vehicles accelerate.*

Figure 8.1.1.5 is a snap shot of the X-scenario after the vehicles have passed one another.

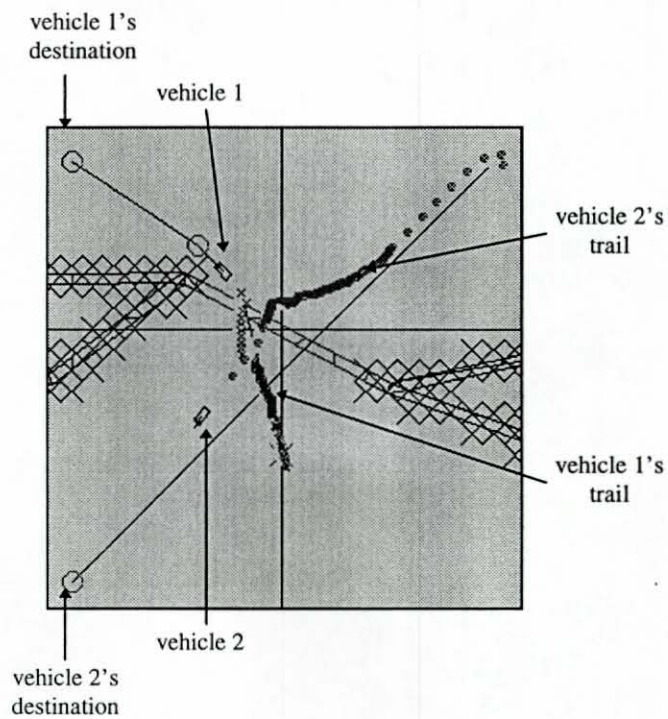


Figure 8.1.1.5 - X-Scenario: Vehicles Accelerate.

Description: Both vehicles have resumed normal speed and are moving toward their route points.

8.1.1.6 End of scenario.

Figure 8.1.1.6 is a snap shot of the end of the X-Scenario.

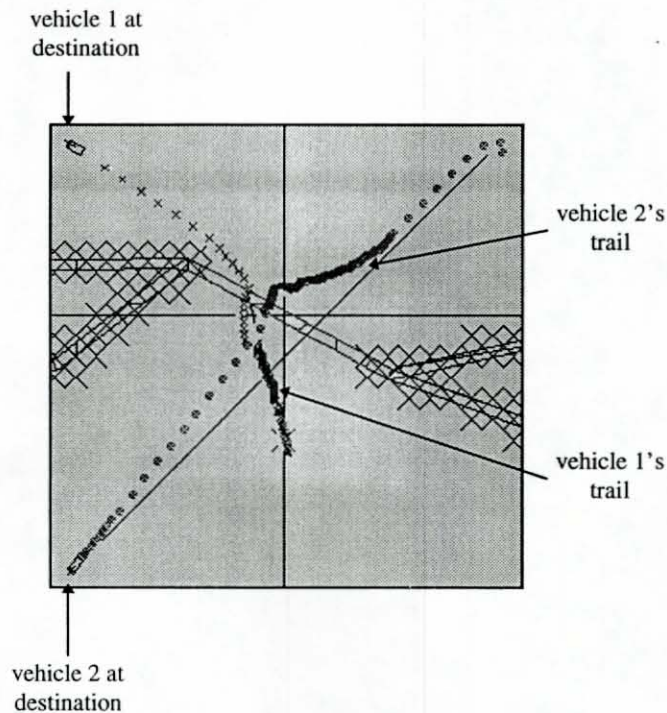


Figure 8.1.1.6 - X-Scenario: End.

Description: The vehicles have completed their routes.

8.1.1.7 Summary of X-scenario .

The maneuvers made by each vehicle in this scenario were:

Vehicle 1:

1. Started to move slowly towards its first route point (figure 8.1.1.1).
2. Is still moving slowly towards its first route point (figure 8.1.1.2).
3. Moved to its left to avoid Vehicle 2 (figure 8.1.1.3).
4. Passed Vehicle 2, and started to gain speed (figure 8.1.1.4).
5. Approached its first route point (figure 8.1.1.5).
6. Reached its destination (figure 8.1.1.6).

Vehicle 2:

1. Started to move at its normal speed towards its destination (figure 8.1.1.1).
2. Decelerated and moved to its right slightly to avoid Vehicle 1 (figure 8.1.1.2).
3. Moved to its left and passed behind Vehicle 1 (figure 8.1.1.3).
4. Passed behind Vehicle 1 but still moved slowly (figure 8.1.1.4).
5. Accelerated and began steering toward route point. (figure 8.1.1.5).
6. Reached destination (figure 8.1.1.6).

In summary, Vehicle 1 began moving slowly and steered slightly to its left to avoid Vehicle 2. Vehicle 2 began moving at full speed, decelerated, steered considerably to its right, and finally,

steered to its left to avoid Vehicle 1. This scenario demonstrates that "incorrect" decisions (e.g. Vehicle 2's turn to the right) are corrected as the vehicles continue to move.

8.1.2 Scenario 2: Head On Collision.

This scenario shows two vehicles approaching one another head on.

8.1.2.1 Start.

Figure 8.1.2.1 is a snap shot a couple of seconds after the beginning of the scenario.

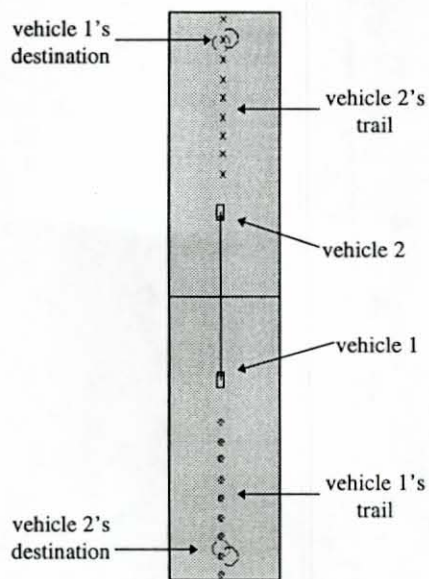


Figure 8.1.2.1 - Head On Collision: Start.

Description: The vehicles are moving at normal speed headed directly at one another

8.1.2.2 Initial Avoidance Maneuvers.

Figure 8.1.2.2 shows both vehicles shortly after they entered one another's DOA grid and have begun avoiding one another.

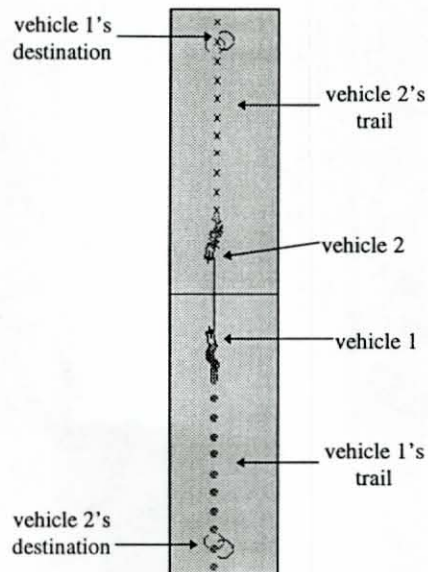


Figure 8.1.2.2 - Head On Collision: Initial Avoidance.

Description: The vehicles have decelerated (see the dense trails for both vehicles) and have begun steering to avoid one another. Vehicle 1 has turned to its left and Vehicle 2 has turned to its right. The vehicles remain in a collision course.

8.1.2.3 Second Phase Avoidance Maneuvers.

Figure 8.1.2.3 shows the vehicles avoiding each other.

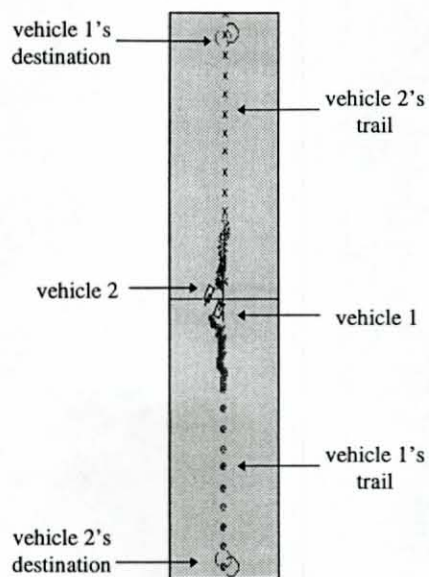


Figure 8.1.2.3 - Head On Collision: Second Phase Avoidance.

Description: Vehicle 1 changes from a left turn to a right turn. Now, the vehicles are not on a collision course but continue to move slowly because of their proximity.

8.1.2.4 Collision Avoided.

Figure 8.1.2.4 is a snap shot the moment after the vehicles pass each other.

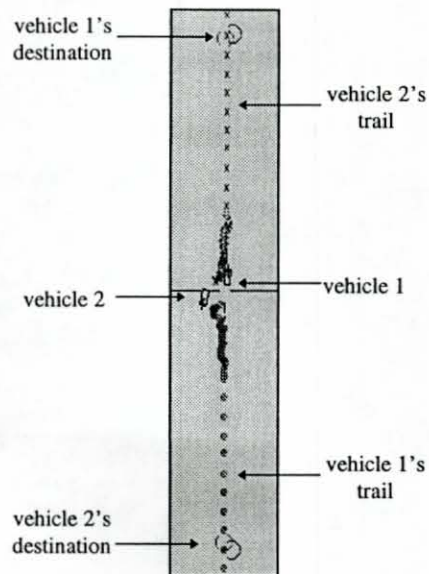


Figure 8.1.2.4 - Head On Collision: Collision Avoided.

Description: The vehicles have passed one another and disappear from one another's DOA grid. The vehicles begin accelerating and steering toward their route points.

8.1.2.5 Near Route Completion.

Figure 8.1.2.5 is a snap shot three quarters through the vehicles' routes.

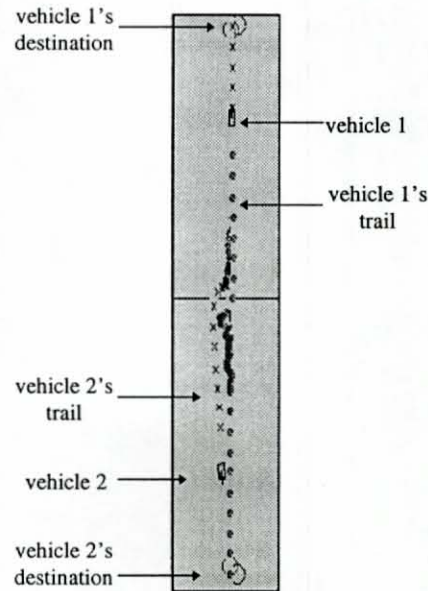


Figure 8.1.2.5 - Head On Collision: Near Route Completion.

Description: The vehicles are moving at regular speed steering directly at their destinations.

8.1.2.6 Summary of this scenario.

The maneuvers made by each vehicle in this scenario were:

Vehicle 1:

1. Started to move towards its destination at normal speed (figure 8.1.2.1).
2. Decelerated and steered left to avoid vehicle 2 but remained in a collision course (figure 8.1.2.2).
3. Turned to its right and avoided vehicle 2 (figure 8.1.2.3).
4. Passed vehicle 2 and accelerated (figure 8.1.2.4).
5. Reached normal speed and steered toward its destination (figure 8.1.2.5).

Vehicle 2:

1. Started to move towards its destination at normal speed (figure 8.1.2.1).
2. Decelerated and steered right to avoid vehicle 2 but remained in a collision course (figure 8.1.2.2).
3. Continued steering right and avoided vehicle 1 (figure 8.1.2.3).
4. Passed vehicle 1 and accelerated (figure 8.1.2.4).
5. Reached normal speed and steered toward its destination (figure 8.1.2.5).

In summary, Vehicle 1 slowed then steered first to its left and then to its right to avoid Vehicle 2. Vehicle 2 slowed and steered to its right to avoid Vehicle 1. This scenario demonstrates that "incorrect" decisions (e.g. Vehicle 1's turn to the left) are corrected as the vehicles continue to move.

8.1.3 Scenario 3: Right Angle Collision.

This scenario shows two MIs coming at each other at right angles: One moving west, and the other moving north.

8.1.3.1 Start.

Figure 8.1.3.1 shows the beginning of the scenario.

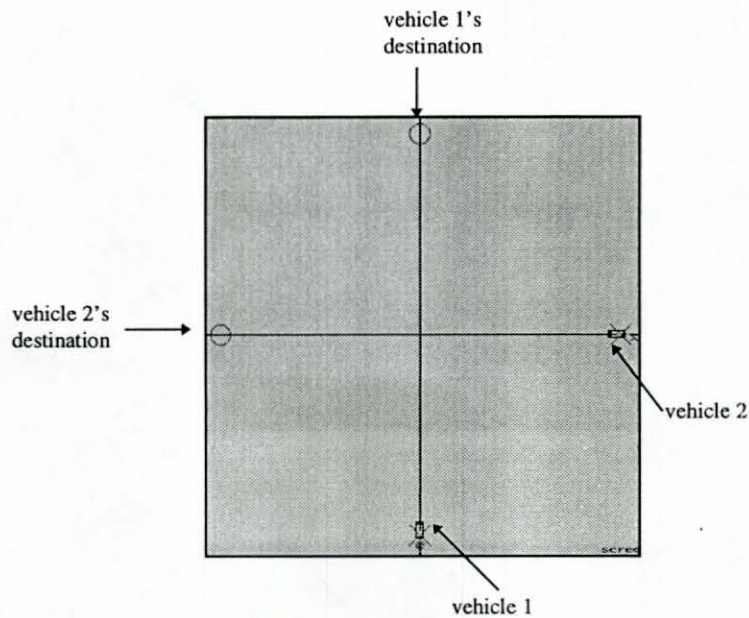


Figure 8.1.3.1 - Right Angle Collision: Start.

Description: The vehicle have routes consisting of single line segments meeting at a right angle. Without DOA, the vehicle collide.

8.1.3.2 Initial Vehicle Avoidance.

Figure 8.1.3.2 is a snap shot after each vehicle begins avoiding the other.

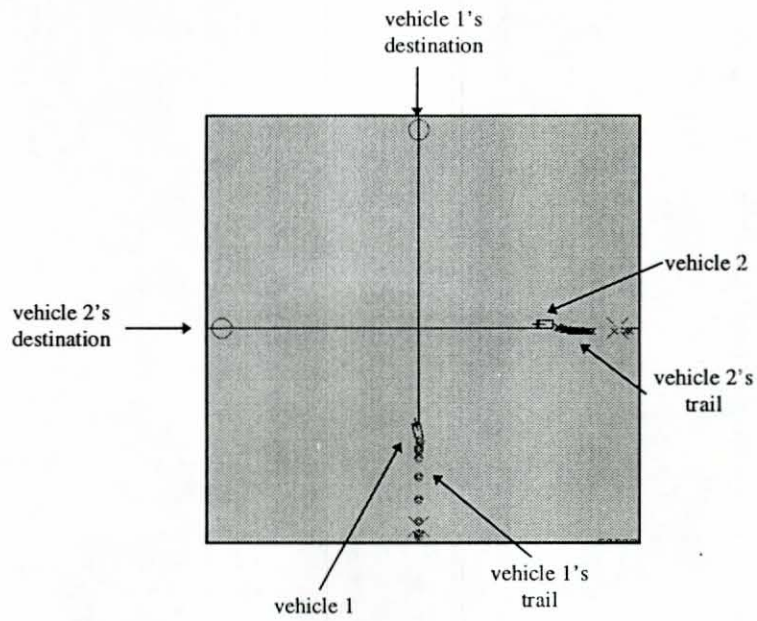


Figure 8.1.3.2 - Right Angle Collision: Initial Avoidance.

Description: The vehicles have decelerated and are beginning to steer away from one another.

8.1.3.3 Approaching a collision point.

Figure 8.1.3.3 shows the vehicles approaching a collision point.

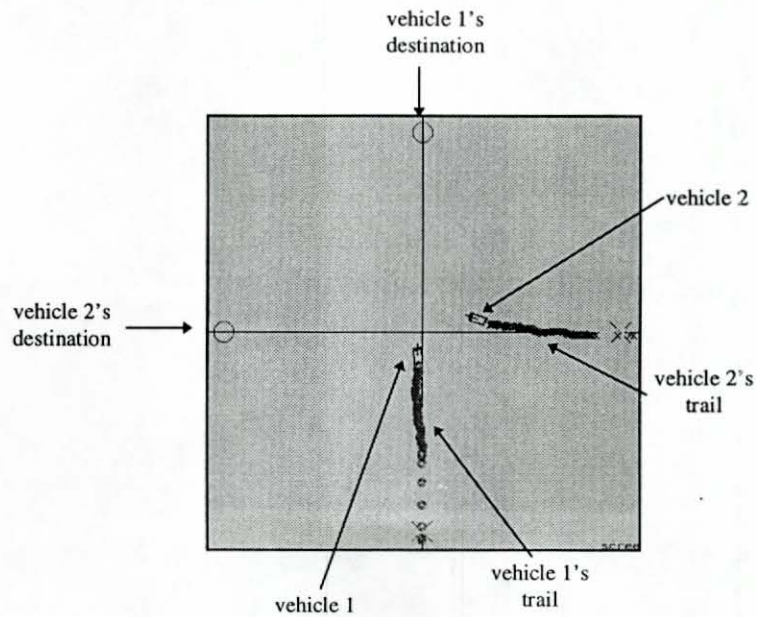


Figure 8.1.3.3 - Right Angle Collision: Approaching a Collision Point.

Description: The vehicles are closer. They have decelerated further and are continuing to steer to avoid a collision.

8.1.3.4 *Closest Approach.*

Figure 8.1.3.4 shows the vehicles at their closest.

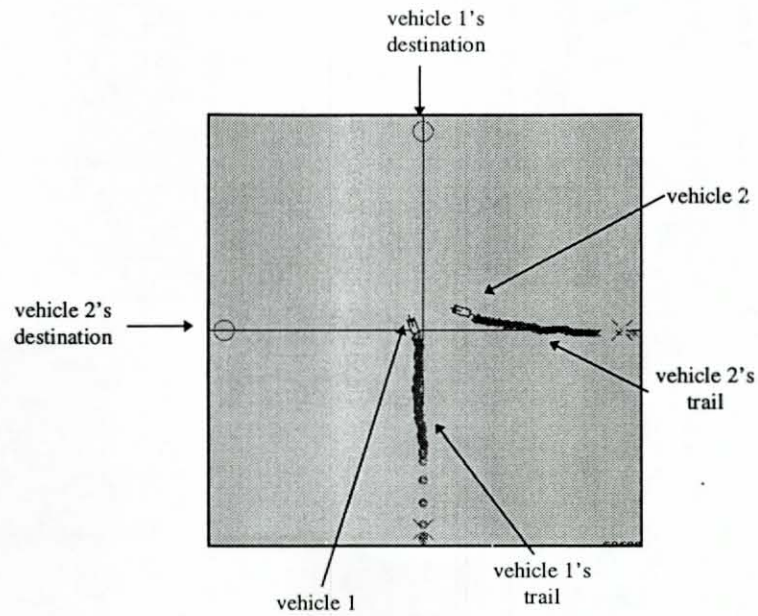


Figure 8.1.3.4 - Right Angle Collision: Closest Approach.

Description: Vehicle 1 has started left turn and, at the same time, Vehicle 2 has decelerated further and started a right turn.

8.1.3.5 Collision Avoided.

Figure 8.1.3.5 is a snap shot of the vehicles when they have avoided each other.

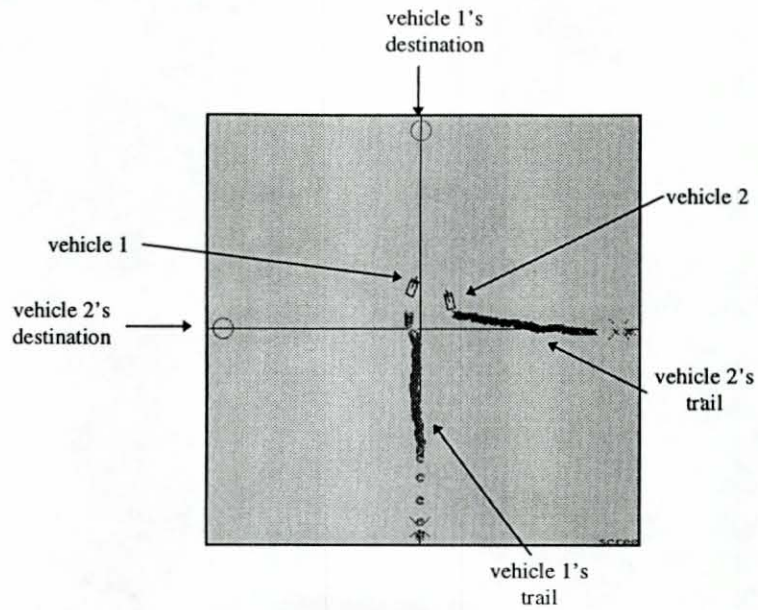


Figure 8.1.3.5 - Right Angle Collision: Collision Avoided.

Description: Vehicle 1 has passed in front of Vehicle 2 and is accelerating. Vehicle 2 has slowed to a stop.

8.1.3.6 Vehicle 2 resumes its route.

Figure 8.1.3.6 shows vehicle 2 resuming its route.

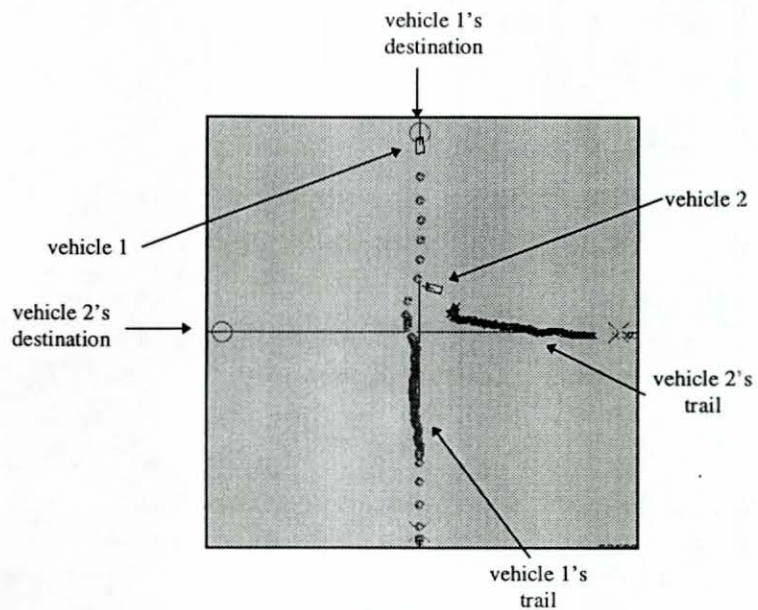


Figure 8.1.3.6 - Right Angle Collision: Vehicle 2 resumes its route.

Description: Vehicle 1 has almost reached its destination. Vehicle 2 is restarting its route.

8.1.3.7 *Vehicle 2 nears its destination.*

Figure 8.1.3.7 shows vehicle 2 nearing its destination.

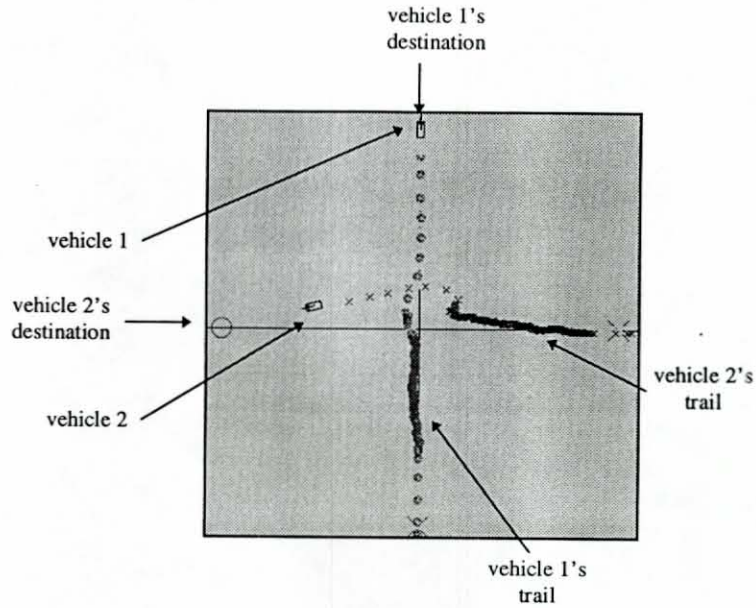


Figure 8.1.3.7 - Right Angle Collision: Vehicle 2 nears its destination.

Description: Vehicle 1 has reached its destination. Vehicle 2 is moving a normal speed and is near its destination..

8.1.3.8 End of scenario.

Figure 8.1.3.8 is a snap shot of the end of the scenario.

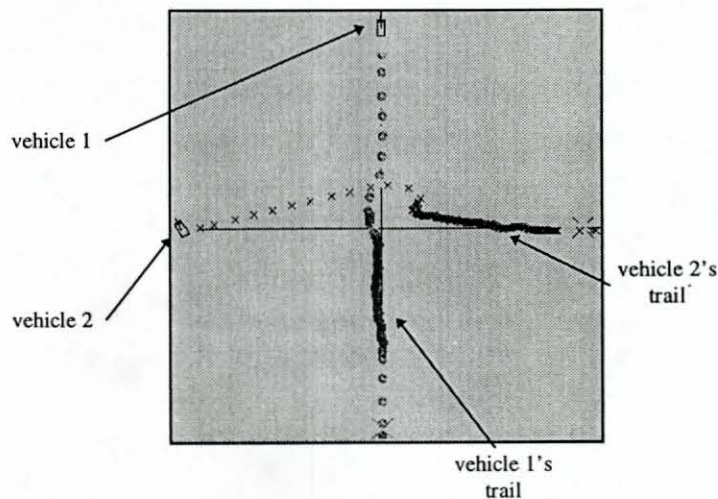


Figure 8.1.3.8 - Right Angle Collision: End.

Description: Both vehicles have reached their destinations. The vehicle trails show the maneuvering. Vehicle 1 slowed and turned slightly to its left. Vehicle 2 slowed, made a sharp turn to its right, then finished its route by turning sharply left towards its destination.

8.1.3.9 Summary of Right Angle Collision.

The maneuvers made by each vehicle in this scenario were:

Vehicle 1:

1. Started to move towards its destination at normal speed (figure 8.1.3.1).
2. Moved slightly to its left to avoid vehicle 2, but remained on a collision course (figure 8.1.3.2).
3. Turned to its right but remained on a collision course (figure 8.1.3.3).
4. Turned back to its left to avoid vehicle 2 (figure 8.1.3.4).
5. Passed vehicle 2 and accelerated (figure 8.1.3.4).
6. Moved at normal speed toward destination (figure 8.1.3.5).
7. Moved at normal speed toward destination (figure 8.1.3.6).
8. Reached destination (figure 8.1.3.8).

Vehicle 2:

1. Started to move towards its destination at normal speed (figure 8.1.3.1).
2. Decelerated but remained on a collision course (figure 8.1.3.2).
3. Decelerated further and steered to its right (figure 8.1.3.3).
4. Has almost stopped because of its proximity to vehicle 1 (figure 8.1.3.4).
5. Stopped to allow vehicle 1 to pass. (figure 8.1.3.5).
6. Restarted its route (figure 8.1.3.6).
7. Accelerated towards its destination (figure 8.1.3.7).
8. Reached its destination (figure 8.1.3.8).

In summary, Vehicle 1 decelerated and tried steering left, then right, then left to avoid Vehicle 2. However, Vehicle 1's turning did not move it far from its planned route. Vehicle 2 decelerated, steered right, and finally stopped to avoid Vehicle 1. Prior to stopping, Vehicle 2 made a sharp turn to its right away from Vehicle 1. When Vehicle 2 restarted moving, it turned sharply left towards its destination and moved at normal speed.

8.1.4 Scenario 4: Competition for the Bridge.

In contrast to the previous scenarios, this scenario involves three vehicles moving in the same direction. The complication is that they must all cross a narrow bridge. The scenario is arranged so that the vehicles would arrive at the bridge simultaneously with DOA.

8.1.4.1 The terrain features.

Figure 8.1.4.1 shows the terrain of this scenario.

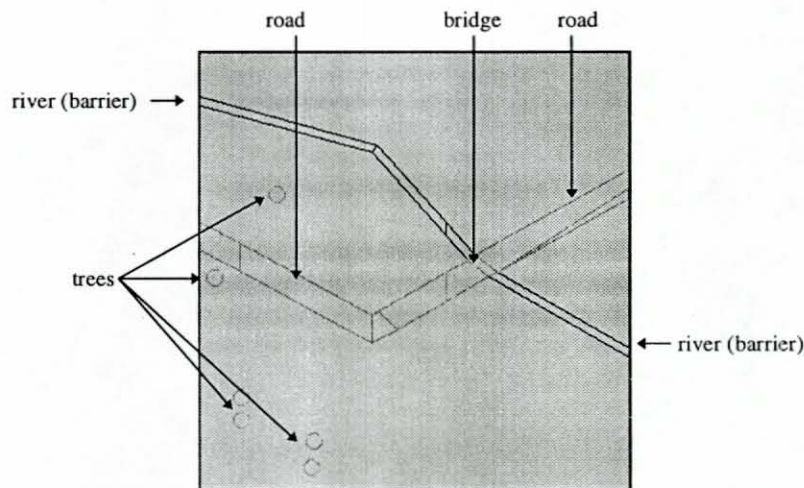


Figure 8.1.4.1 - Bridge: Terrain Features.

Description: The important features are the trees, the river, the road, and the bridge. The vehicles will cross the bridge from left to right.

8.1.4.2 Start.

Figure 8.1.4.2 is the beginning of this scenario.

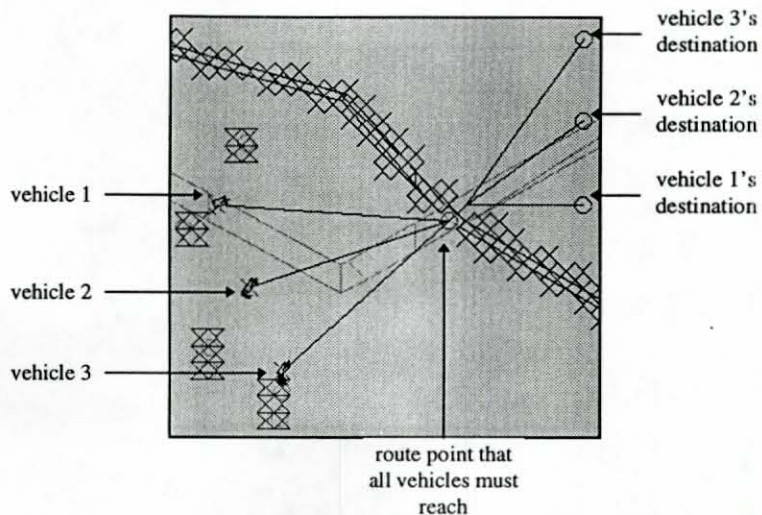


Figure 8.1.4.2 - Bridge: Start.

Description: The three vehicles move towards their destinations. Vehicle 1 is moving at normal speed. Vehicles 2 and 3, although ordered to proceed at normal speed, have limited their acceleration to avoid the other vehicles.

8.1.4.3 Vehicle 1 breaks loose from the others.

Figure 8.1.4.3 shows Vehicle 1 after it has broken loose from the influence of the other two vehicles.

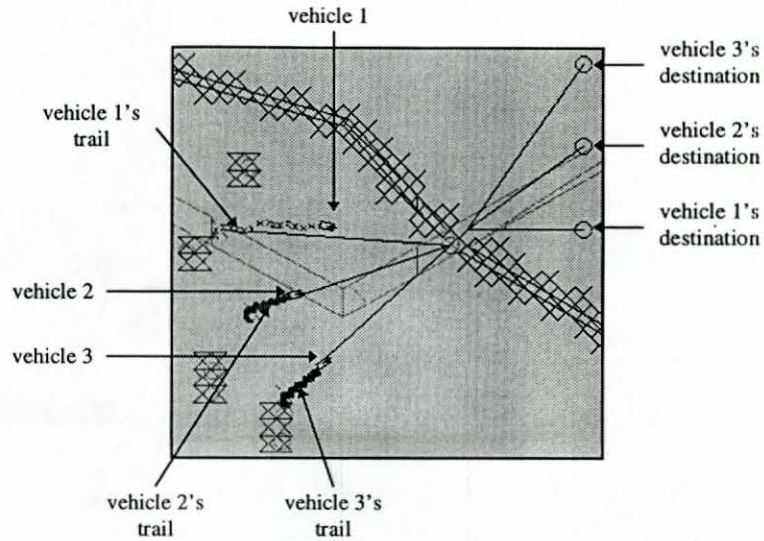


Figure 8.1.4.3 - Bridge: Vehicle 1 Accelerates.

Description: Vehicle 1 has been steering slightly to its left to avoid the other vehicles. Now the other vehicles drop off Vehicle 1's grid and Vehicle 1 accelerates toward the bridge. Vehicles 2 and 3 continue to move slowly.

8.1.4.4 *Vehicle 1 crosses the bridge.*

Figure 8.1.4.4 shows Vehicle 1 crossing the bridge.

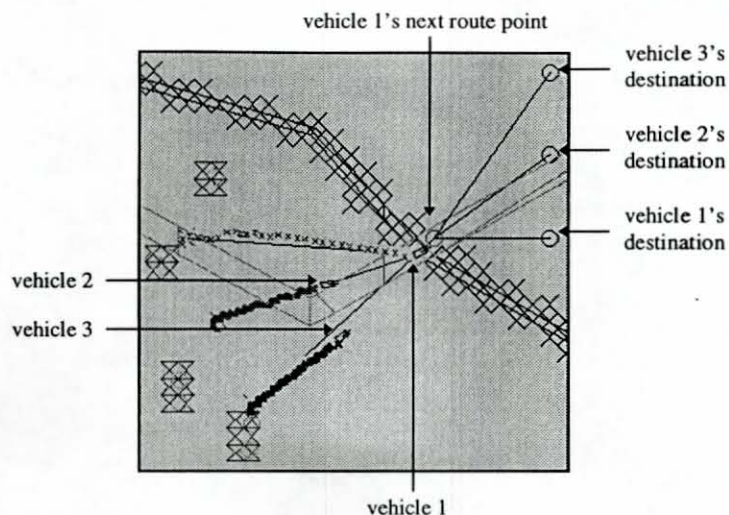


Figure 8.1.4.4 - Bridge: Vehicle 1 Crosses the Bridge.

Description: Vehicle 1 has reached the bridge and is moving at normal speed towards its route point. Vehicle 2 and 3 are still under each others influence. Both are moving slowly and Vehicle 3 is steering to its right which is allowing Vehicle 2 is get ahead of Vehicle 3.

8.1.4.5 *Vehicle 2 crosses the bridge.*

Figure 8.1.4.5 shows Vehicle 2 crossing the bridge.

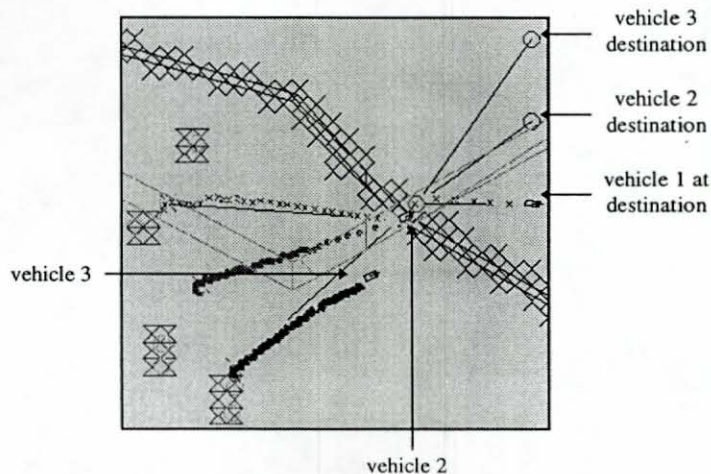


Figure 8.1.4.5 - Bridge: Vehicle 2 Crosses the Bridge.

Description: Vehicle 1 has reached its destination and stopped; it is now a static object, a BARRIER. Vehicle 2 is crossing the bridge and accelerating. Vehicle 3 is accelerating because Vehicle 2 has moved sufficiently ahead of Vehicle 3 to move off Vehicle 2's DOA grid.

8.1.4.6 *Vehicle 3 crosses the bridge.*

Figure 8.1.4.6 is a snap shot of Vehicle 3 crossing the bridge.

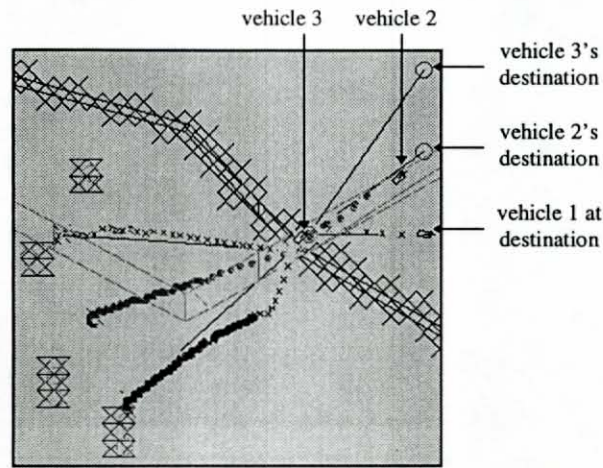


Figure 8.1.4.6 - Bridge: Vehicle 3 Crosses the Bridge.

Description: Vehicle 2 has crossed the bridge and is nearing its destination. Vehicle 3, moving at normal speed, is crossing the bridge.

8.1.4.7 *Vehicle 3 approaches its destination.*

Figure 8.1.4.7 shows vehicle 3 decelerating near its destination.

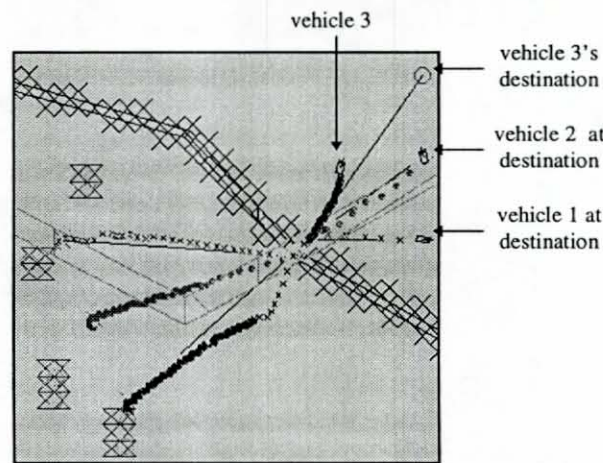


Figure 8.1.4.7 - Bridge: Vehicle 3 Approaches its Destination.

Description: Vehicle 2 has reached its destination. Vehicle 3 has decelerated because it has caught up to Vehicle 2 while Vehicle 2 was decelerating to stop at its destination.

8.1.4.8 End of scenario.

Figure 8.1.4.8 shows the end of this scenario.

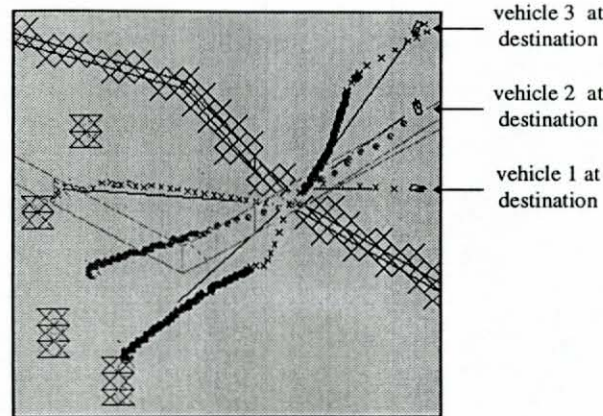


Figure 8.1.4.8 - Bridge: End.

Description: Vehicle 3 accelerated after Vehicle 2 reached its destination and stopped.

8.1.4.9 Summary of Competition for the Bridge.

The maneuvers made by each vehicle in this scenario were:

Vehicle 1:

1. Started to move slowly towards its first route point (figure 8.1.4.2).
2. Broke loose from the other vehicles influence and accelerated (figure 8.1.4.3).
3. Crossed the bridge at normal speed (figure 8.1.4.4).
4. Reached its destination (figure 8.1.4.5).

Vehicle 2:

1. Started to move slowly towards its first route point (figure 8.1.4.2).
2. Continued to move slowly to avoid Vehicles 1 and 3 (figure 8.1.4.3).
3. Moved a little bit ahead of vehicle 3 (figure 8.1.4.4).
4. Crossed the bridge at normal speed (figure 8.1.4.5).
5. Approached destination (figure 8.1.4.6).
6. Reached destination (figure 8.1.4.7).

Vehicle 3:

1. Started to move slowly towards its first route point (figure 8.1.4.2).
2. Continued to move slowly to avoid Vehicles 2 and 3 (figure 8.1.4.3).
3. Decelerated to avoid Vehicle 2 (figure 8.1.4.4).
4. Steered to its right to avoid Vehicle 2 (figure 8.1.4.5).
5. Crossed the bridge at normal speed (figure 8.1.4.6).
6. Decelerated after crossing bridge to avoid Vehicle 2 (figure 8.1.4.7).
7. Accelerated after Vehicle 2 stopped (figure 8.1.4.8).

In summary, the vehicles, through a combination of deceleration and minimal steering, passed over the bridge without collision. This scenario demonstrates that the DOA Model resolves conflicts at chokepoints without a set of "rules of right-of-way".

8.1.5 Head On Collision On the Bridge

This scenario shows two vehicles approaching one another head on, trying to cross a bridge. The beginning and final portion of this scenario will show the vehicles' trails. However, the moment one of the vehicles starts to back up, the trails are suppressed so that the backing up action can be seen in its entirety.

8.1.5.1 Start

Figure 8.1.5.1 is a snap shot of the beginning of this scenario.

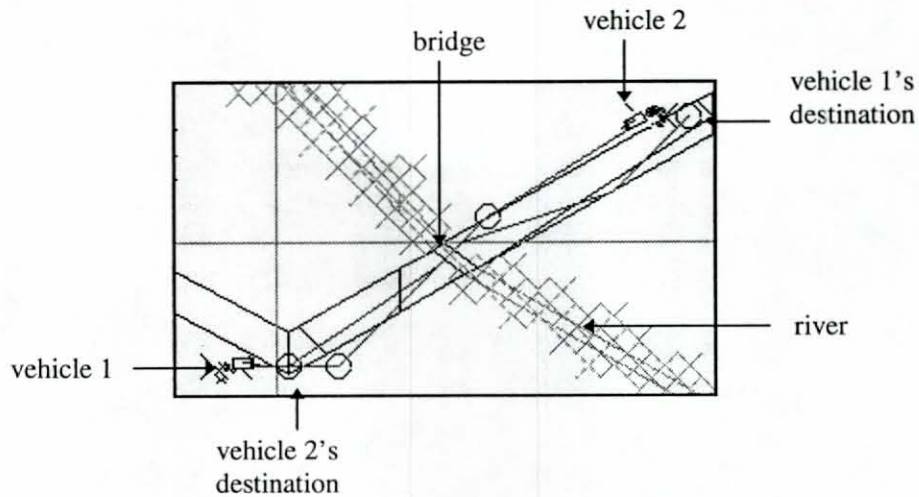


Figure 8.1.5.1 - Head On the Bridge: Start.

Description: The vehicles are starting to move towards the bridge.

8.1.5.2 The Vehicles near the Bridge.

Figure 8.1.5.2 shows both vehicles approaching the bridge from their respective sides.

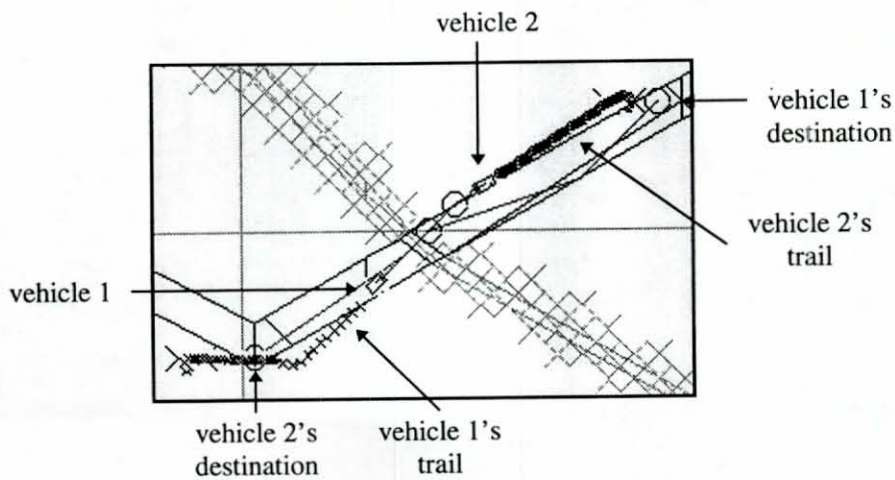


Figure 8.1.5.2 - Head On the Bridge: Vehicles near the bridge.

Description: The vehicles have been moving slowly (see the dense trails for both vehicles), and are nearing the bridge.

8.1.5.3 Stop to Avoid a Collision.

Figure 8.1.5.3 shows both vehicles after they have stopped.

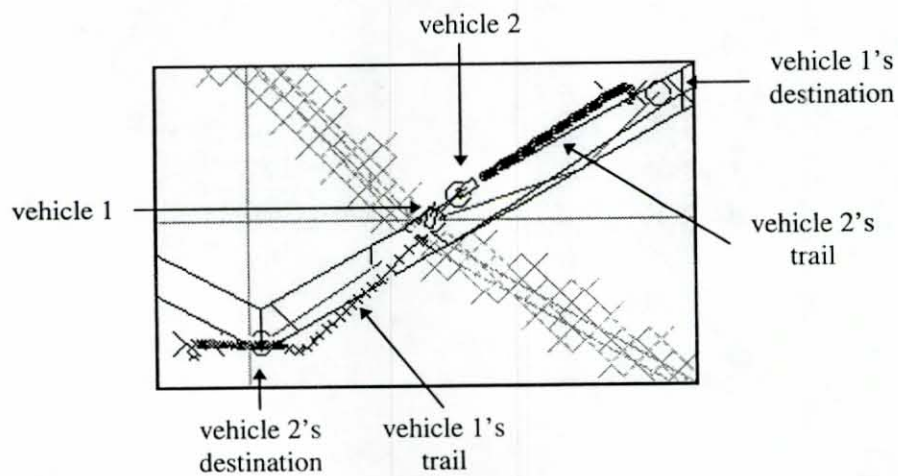


Figure 8.1.5.3 - Head On the Bridge: Stop to Avoid a Collision.

Description: The vehicles have slowed to a stop because their DOA routes have taken them too close together due to the narrow bridge.

8.1.5.4 *Start Backing up.*

Figure 8.1.5.4 shows Vehicle 2 backing up. For clarity, the back up sequence does not show the vehicle's trails.

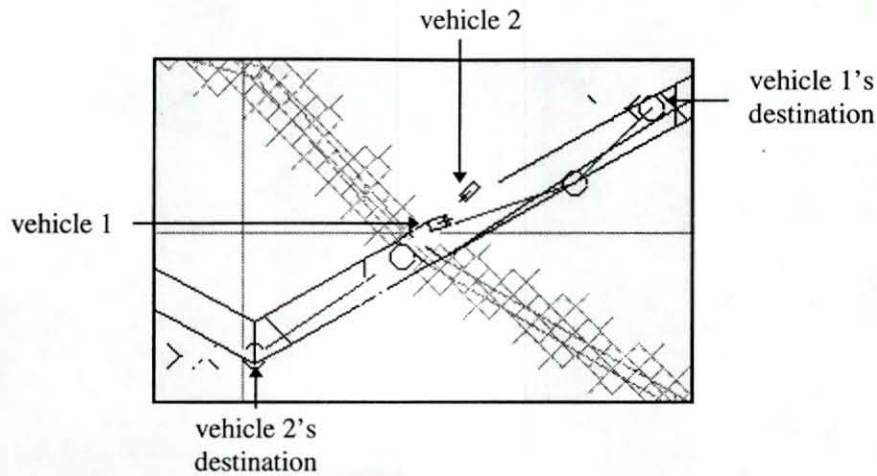


Figure 8.1.5.4 - Head On the Bridge: Start Backing up.

Description: Vehicle 1 is stationary while Vehicle 2 backs up. Only Vehicle 2 satisfies the “back up” rule in the DOA algorithm.

8.1.5.5 *Finished Backing up.*

Figure 8.1.5.5 shows Vehicle 2 finishing its back up maneuver.

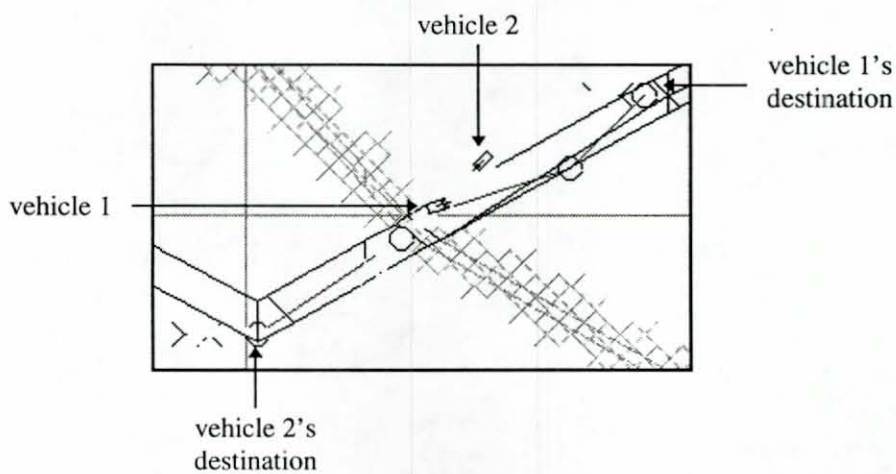


Figure 8.1.5.5 - Head On the Bridge: Finished Backing up.

Description: Vehicle 2 has finished backing up. This opens a route on Vehicle 1's DOA grid.

8.1.5.6 Collision Avoided.

Figure 8.1.5.6 is a snap shot after the vehicles pass each other. The vehicles' trails are being displayed again.

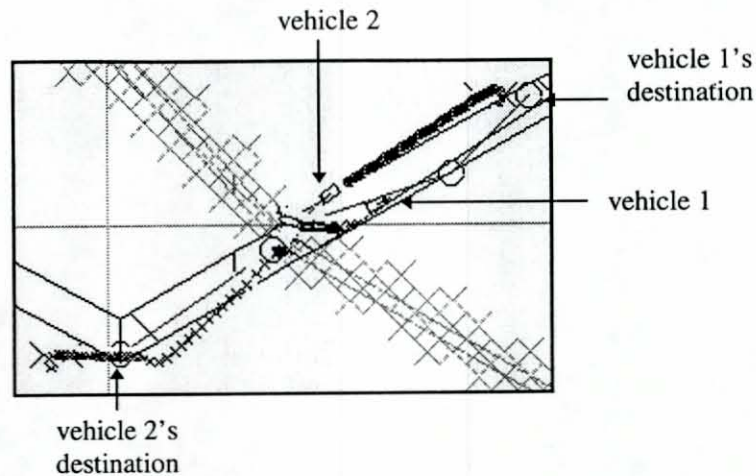


Figure 8.1.5.6 - Head On the Bridge: Collision Avoided .

Description: The vehicles have passed one another, disappeared from one another's DOA grid, and headed toward their route points.

8.1.5.7 End of Scenario.

Figure 8.1.5.7 is a snap shot of the end of the scenario.

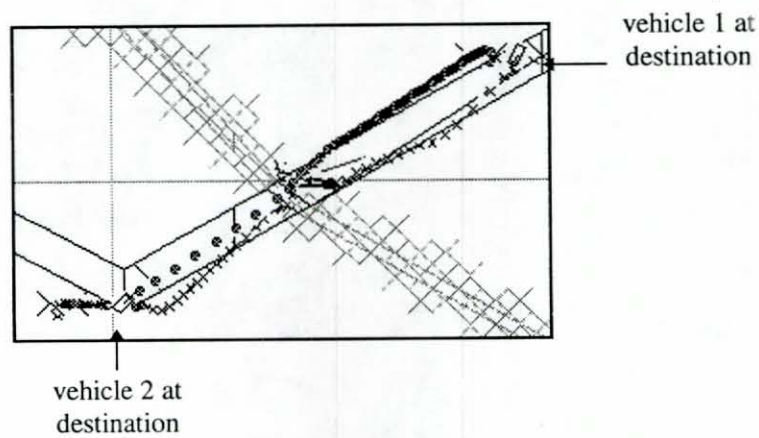


Figure 8.1.5.7 - Head On the Bridge: End .

Description: Both vehicles have reached their destinations. The vehicle trails show their maneuvering, except for Vehicle 2's back up maneuver.

8.1.5.8 *Summary of Head On Collision On the Bridge.*

The maneuvers made by each vehicle in this scenario were:

Vehicle 1:

1. Started to move towards its destination at low speed (figure 8.1.5.1).
2. Approached the bridge at slower than normal speed (figure 8.1.5.2).
3. Crossed the bridge and stopped in front of Vehicle 2 to avoid a collision (figure 8.1.5.3).
4. Remained stationary while Vehicle 2 backed up (figure 8.1.5.4).
5. Remained stationary while Vehicle 2 backup (figure 8.1.5.5).
6. Passed Vehicle 2 and headed for its destination (figure 8.1.5.6).
7. Reached destination (figure 8.1.5.7).

Vehicle 2:

1. Started to move towards its destination at low speed (figure 8.1.5.1).
2. Approached the bridge at slower than normal speed (figure 8.1.5.2).
3. Stopped to avoid a collision with Vehicle 1 (figure 8.1.5.3).
4. Backed up to make room to move (figure 8.1.5.4).
5. Finished backing up (figure 8.1.5.5).
6. Restarted its route and accelerated (figure 8.1.5.6).
7. Reached its destination (figure 8.1.5.7).

In summary, the vehicles avoided a head on collision at the bridge by both vehicles slowing down to a stop and Vehicle 2 backing up. The back up maneuver cleared a path for Vehicle 1.

9. Conclusions and Future Work

IST has developed a novel approach to attack the DIS dynamic obstacle avoidance (DOA) problem by combining two disparate motion planning approaches: potential field and regular grid analysis. This approach is rooted in neural net fundamentals and the underlying design allows various techniques to be brought to bear on the avoidance problem. To allow focused study of the DOA problem, IST has developed a stand alone DOA Testbed. On this foundation, IST has implemented and evaluated many techniques which would seem inapplicable using other approaches (from a simple "best guess" method to spline fits). To test the validity and applicability of these results, IST implemented the more successful DOA algorithms within its CGF Testbed and studied their results within a DIS environment. In particular, the A* based DOA Model shows excellent moving obstacle avoidance while maintaining reasonably close adherence to previously created piecewise linear routes.

There are several opportunities for further work in the area of Dynamic Obstacle Avoidance. Among them are the real time coordination of route following, station keeping within formation, and dynamic obstacle avoidance. This work has focused on dynamic obstacle avoidance within the context of following lengthy routes generated by route planners that ignore dynamic (moving) obstacles. The coordination of dynamic obstacle avoidance and station keeping within a formation is an interesting, untouched area for further research.

10. References

- Charniak, Eugene and McDermott, Drew (1987). *Introduction to Artificial Intelligence*, Addison-Wesley.
- Cortes-Rello, E. and Golshani, F. (1990). "Dynamic Route Planning", *Proceedings of the Fifth Rocky Mountain Conference on Artificial Intelligence: Pragmatics in Artificial Intelligence*, Las Cruces, NM, pp 87-92.
- Danisas, K., Smith, S. H., and Wood, D. D. (1990). "Sequencer/Executive for Modular Simulator Design", *Technical Report IST-TR-90-1*, Institute for Simulation and Training, University of Central Florida, 16 pages.
- DIS Steering Committee (1993). "The DIS Vision: A Map to the Future of Distributed Simulation", *IST Technical Report*, 47 pages.
- Gladius, R., Komoda, A., and Gielen, S. (1994). "Neural Network Dynamics for Path Planning and Obstacle Avoidance", *Neural Networks*, March 1994 (to appear).
- Gonzalez, G., Mullally, D. E., Smith, S. H., Vanzant-Hodge, A. F., Watkins, J. E., and Wood, D. D. (1990). "A Testbed for Automated Entity Generation in Distributed Interactive Simulation", *Technical Report IST-TR-90-15*, Institute for Simulation and Training, University of Central Florida, 37 pages.
- Hertz, John, Krogh, Anders, and Palmer, Richard G (1992). *Introduction To The Theory Of Neural Computation*, Addison-Wesley.
- Hwang, Y. K. and Ahuja, N. (1992). "Gross Motion Planning--A Survey", *ACM Computing Surveys*, Vol. 24, No. 3, pp.219-291.
- Loper, M. L., Thompson, J. R., and Williams, H. L. (1991). "Simulator Networking: What Can It Offer The Training Community?", *Military Simulation & Training*, Issue 4 1991, pp. 11-14.
- Mitchell, J. S. B. (1988). "An Algorithmic Approach to Some Problems in Terrain Navigation", *Artificial Intelligence*, Vol. 37, pp. 171-201.
- NASA (1991), "Potential-Field Scheme for Avoidance of Obstacles by a Robot", *NASA Tech Brief KSC-11491*, John F. Kennedy Space Center, FL., pp. 1-43.
- Petty, M. D. (1992). "Computer Generated Forces in Battlefield Simulation", *Proceedings of the Southeastern Simulation Conference 1992*, The Society for Computer Simulation, Pensacola FL, October 22-23 1992, pp. 56-71.
- Roos, T. and Noltemeier, H. (1991). "Dynamic Voronoi diagrams in Motion Planning", *Proceedings of the International Workshop on Computational Geometry*, Bern Switzerland, March 1991, pp 228-236.
- Smith, S. H., Karr, C. R., Petty, M. D., Franceschini R. W., and Watkins, J. E. (1992a). "The IST Computer Generated Forces Testbed", *Technical Report IST-TR-92-7*, Institute for Simulation and Training, University of Central Florida.

Smith, S. H., and Petty, M. D. (1992b). "Controlling Autonomous Behavior in Real-Time Simulation", *Proceedings of the Southeastern Simulation Conference 1992*, The Society for Computer Simulation, Pensacola FL, October 22-23 1992, pp. 27-40.

Thorpe, C. E. (1984). "Path Relaxation: Path Planning for a Mobile Robot", *CMU-RI-TR-84-5*, Carnegie-Mellon University The Robotics Institute Technical Report, April 1984.

Winston, Henry Patrick (1992). *Artificial Intelligence*, Third Edition, Addison-Wesley.

11. Appendix A. DOA Configuration File

* Dimensions of the grid.

D 20 10

* Cell size.

C 10.0

* Target temperature.

T 1.0

* Vehicle temperature and vehicle type (M1 = 0, M2 = 1, T72 = 2, and BMP = 2).

V 0.0 0

* Limit of iterations to perform cell propagation on the DOA grid.

L 10

* Obstacle temperature.

O -2.0

* Barrier temperature.

B 0.0

* Multiplier for cell updates.

M 0.9999

* Grid cell look ahead.

G 2

* Algorithm to use (A* = 0 and POSNEG = 1)

P 0

* Method to use (ZERO = 0, STEP = 1, HILL = 2, BOTH [STEP & HILL] = 3)

E 3

* Percentage of speed for vehicle to be able to start maneuvering.

% 50

* Coefficient to divide selected speed by

F 2.0

* Display GRid (FALSE = 0, TRUE = 1)

R 0

* Display PaTh (FALSE = 0, TRUE = 1)

H 0

* Display TrAil (FALSE = 0, TRUE = 1)

A 1

12. Appendix B: DOA Configuration File Data Structures

```
/* Dynamic Obstacle Avoidance information from configuration file. */
typedef struct
{
    PATH path; /* Type of path we are going to follow. */
    METHOD method; /* The method we are going to use. */
    float vehicle; /* Temperature of the vehicle. */
    float target; /* Temperature of the target. */
    float obstacle; /* Temperature of the obstacles. */
    float barrier; /* Temperature of the barriers. */
    int percentage_of_speed; /* This is the percentage of the speed
    /* to start turning.
    int display_grid, /* Boolean: TRUE displays grid.
        display_path, /* Boolean: TRUE displays path.
        display_trail; /* Boolean: TRUE displays the vehicle's
    /* trail.
    float speed_coefficient; /* Coefficient to divide selected speed.
    DOA_VEHICLES vehicle_type; /* Type of vehicle using DOA ( M1 ... )
    int grid_cell_lookahead; /* How many grid cells around a vehicle
    /* will be claimed by the vehicle?
    float cell_size; /* Cell size on the grid.
    int grid_width; /* The width of the grid.
    int grid_height; /* The height of the grid.
    int step_limit; /* Number of iterations used to fill in
    /* the grid.
    float multiplier; /* Multiplier used in cell updates.
}DOA_INFO;
```

13. Appendix C. DOA Grid Cell Data Structures

```
/* Map cells represent a square in the search. Each cell has */
/* 3 qualities: a value (a float, the "temperature"), a status */
/* (is the value computed, does it hold the target, is it an */
/* obstacle, etc.), and a step indicator (TRUE indicates a */
/* path walked through the square). */
```

```
typedef struct
{
    float value;
    STATUS kind;
    BOOLEAN used;
}CELL;
```

```
/* The status of a cell in the grid. */
```

```
typedef enum
{
    COMPUTED, /* a computed value. */
    TARGET, /* The target. */
    DOA_VEHICLE, /* The vehicle to move. */
    OBSTACLE, /* An obstacle (repels). */
    EDGE, /* A grid edge. */
    BARRIER, /* a fixed barrier. */
}STATUS;
```

0000168