Obstacle Avoidance Using Complex Vector Fields

Tommie Liddy[1], **Tien-Fu Lu**[1], **Peter Lozo**[2], **David Harvey**[3]
[1]The School of Mechanical Engineering
The University of Adelaide, Adelaide, SA 5005, Australia
{tommie.liddy,tien-fu.lu}@adelaide.edu.au
[2]Weapons Systems Division
Defence Science and Technology Organisation(DSTO)
PO BOX 1500 Edinburgh, SA 5111, Australia
peter.lozo@dsto.defence.gov.au
[3]Aerospace Concepts
17 Yallourn St, Fyshwick, ACT 2609, Australia
david.harvey@concepts.aero

Abstract

Obstacle avoidance is an important part of an autonomous navigation system. One method of obstacle avoidance is the use of simple vector fields. Simple vector fields used for obstacle avoidance consist of a repulsive force centred on the obstacle (obstacle vector field) that guides the vehicle away from the obstacle. This is added directly to an existing vector field (environmental vector field) that guides the vehicle towards the goal to produce a resultant force on a vehicle. This method has been shown to avoid obstacles but does have limitations, the most common of which is that the vehicle can become trapped in a 'U' shaped dead end. This study outlines a method of vector field navigation that blends the environmental and obstacle vector fields instead of adding them directly together. The navigation method will also use rotational vector fields for obstacle avoidance instead of repulsive forces. It will be shown that this method can be used for avoiding obstacles as well as evading the 'U' shaped or dead end trap.

1 Introduction

The use of vector fields is a simple and low cost method of both navigation and obstacle avoidance for an autonomous vehicle. To incorporate obstacle avoidance into the navigation method two vectors have been created and combined. The first vector usually represents the desired vehicle behaviour, created with an attractive force towards a goal or waypoint. The other vector is created by the obstacles detected or known by the vehicle and usually acts away from the obstacles. The standard combination method used is to add these vectors together [Borenstein and Koren, 1989].

Vectors are commonly created in a grid [Kim *et al.*, 1999; Loizuo *et al.*, 2003] but can also be created based on the vehicles position with respect to the desired goal or waypoint [Liddy *et al.*, 2007]. If the vectors are created in a grid there will be less calculation in real time as long as the environment does not change. If the environment changes then the entire grid has to be recalculated. If the vectors are created based on the relative positions of the vehicle, waypoint and obstacles [Liddy *et al.*, 2007] then the calculations must be done repeatedly in real time but a change in the environment will not alter the number of calculations required.

Vector field navigation systems often have one main weakness. The navigation systems are unable to escape from within a concave or 'U' shaped dead end. The vehicle often gets caught in a navigational loop where the same path is repeated continuously. To overcome this problem many navigation systems use a secondary navigation protocol [Vadakkepat *et al.*, 2000]. These secondary protocols often use a monitored value or specific environmental configuration to trigger the switch from one to the other and back again [Borenstein and Koren, 1989].

Through this paper a navigation system will be introduced that will be able to avoid obstacles and evade the dead end trap. This system will calculate an environmental and an obstacle vector field that represents the current state of the known local area instead of predetermining a vector field. The environmental vector field will be created with the method described in Liddy *et al*. [2007] while the obstacle vector field will be created using rotational instead of repulsive vector fields. These two vector fields will then be blended using a weighting function to produce desired vector for the vehicle. The results show that obstacle avoidance and trap evasion are both possible with this navigation system.

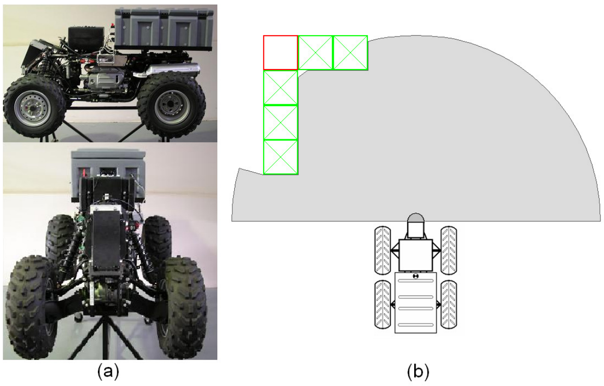

Figure 1: (a) Honda Quad Bike; (b) Vehicle sensor limitations

2 Simulation Modelling

Experiments were done in a simulated environment using a mathematical model to represent the Autonomous Ground Vehicle (AGV)[Liddy *et al*., 2007]. The vehicle model used Ackermann steering and was based on a quad bike, Figure 1(a). The environment in which the simulated AGV travelled was considered flat in the X-Y plane. Waypoints, which consisted of a position in the X-Y plane and a heading (x_{wp},y_{wp},θ_{wp}), were placed in the environment for the AGV to navigate towards. Similarly, obstacles consisting of a position in the X-Y plane as well as a length and width along the x and y axis (x_{ob},y_{ob},l_{ob},w_{ob}) were placed in the environment to represent likely structures which the AGV would be required to navigate around. For simplicity the obstacles in this study were set as squares (l_{ob}=w_{ob}=1 meter) and complex shapes were made by clustering multiple objects together in specific configurations.

The simulation was run as a real time navigational system. The AGV was given no prior knowledge of the obstacles in the environment. To locate the obstacles a modelled version of an ideal planar laser scanner was used. The modelled planar laser scanner was able to "see" up to 15 meters away between 90^o and -90^o from its centre, as shown in Figure 1(b).

3 Obstacle vector field

The main objective of an obstacle vector field is to produce a force which acts on the AGV in such a way that it avoids the obstacle. This is most commonly done by creating a repulsive force which diminishes as the distance from the obstacle increases. In an instance where there are multiple obstacles multiple forces are added together to create one obstacle vector field. Both this vector field and the environmental vector field are superimposed and the result is used to determine the movement of the vehicle, this process is shown in Figure 2.

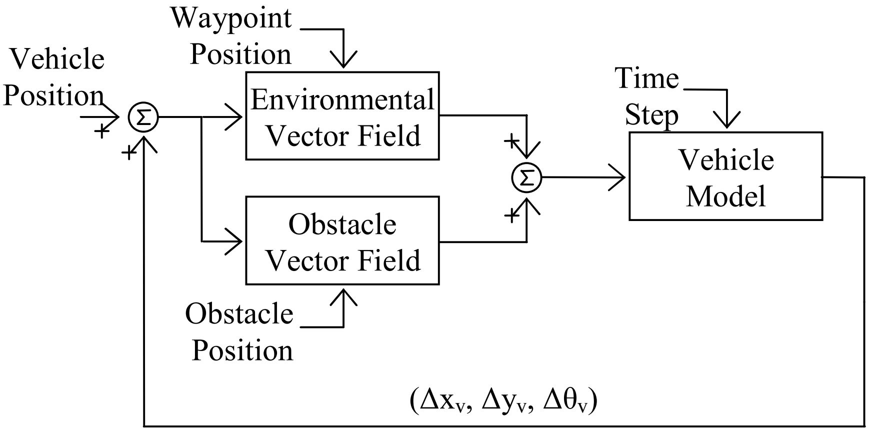

Figure 2: Model for vector field navigation

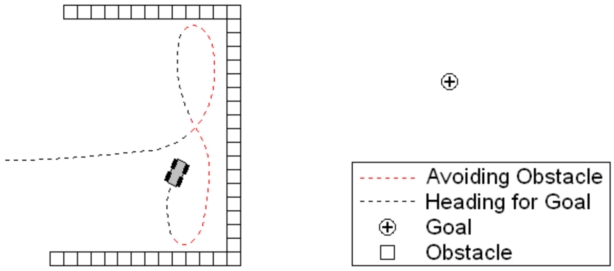

Figure 3: Common trap associated with vector field navigation

With the above method the obstacle vector field acts to move the AGV away from obstacles but it does not act to move the AGV closer to the desired point. Alternatively the environmental vector field acts to move the AGV towards a goal but does not act to avoid any obstacle. In some instances these two vector fields can act against one another diminishing the effectiveness of both. It is these instances which can allow the AGV to be stuck in a trap where it repeatedly switches focus from avoiding the obstacles to reaching the goal but achieves neither. The most common of these traps can be seen when the AGV heads into a 'U' shaped dead end where the goal is outside of the enclosed space, this can be seen in Figure 3.

By altering this method so that the obstacle vector field and the environmental vector field no longer act against one another it is possible to avoid this trap while retaining the ability to move around obstacles and reach a desired goal.

3.1 Blend method for combining vector fields

To avoid the conflict between the obstacle and the environmental vector fields the simple superposition of the two was replaced by a blending function, following the model shown in Figure 4. The blend function used a weighting value called the blend factor (Bf in Equation (1)) to merge the environmental vector field (*envVF*) and the obstacle vector field (*odsVF*) into a navigational vector field (*navVF*) as shown in Equation (2).

$$Bf(x) = \min\left(\max\left(A + B\left(\frac{x}{TP_x}\right),1\right),0\right) \qquad (1)$$

$$navVF = Bf * envVF + (1 - Bf) * obsVF \qquad (2)$$

As a weighting value the blend factor was calculated to be between zero and one as shown in Figure 5(a). The blend factor can be calculated according to different properties of the AGV and the information known about the environment. These can be chosen based on the requirements of the navigation system.

For a real time navigation system it was evident that the blend factor would become quite complex or alternatively more than one blend factor would be required. For the AGV to the obstacle and the minimum turning radius of the vehicle (x_1=D_{vob}: TP_{x1}=4*R_{min}: A=-0.5: B=2) was used and secondly a blend factor based on the absolute testing done throughout this study two blend factors were used. Firstly a blend factor based on the distance from the angle between the waypoint, AGV and the obstacle (x_2=ANG_{min}: TP_{x2}= π/2: A=-0.5: B=2) was used. The blend factors were calculated using Equations (1) and then combined using Equation (3). The variables TP_{x1} and TP_{x2} were set at twice the turning circle of the vehicle and the maximum sensor angle respectively.

$$Bf(x_1, x_2) = 1 - (1 - Bf(x_1)) * (1 - Bf(x_2)) \qquad (3)$$

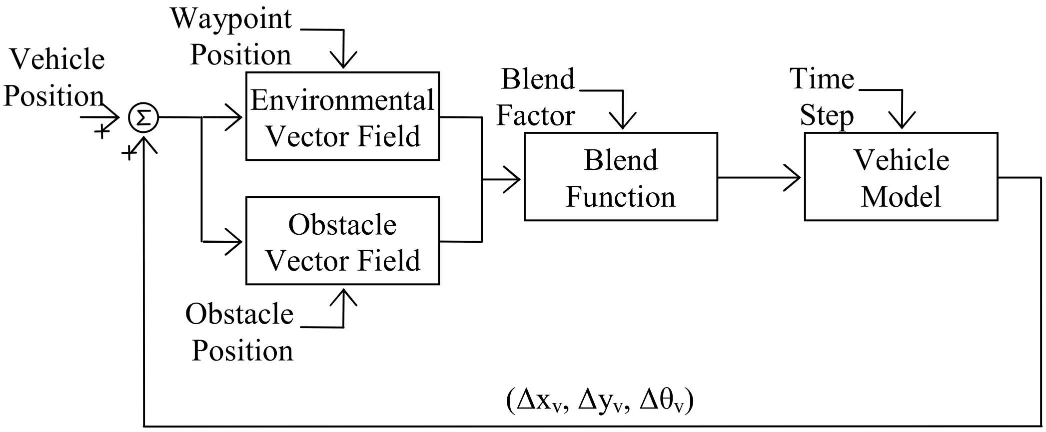

Figure 4: Model for modified vector field navigation

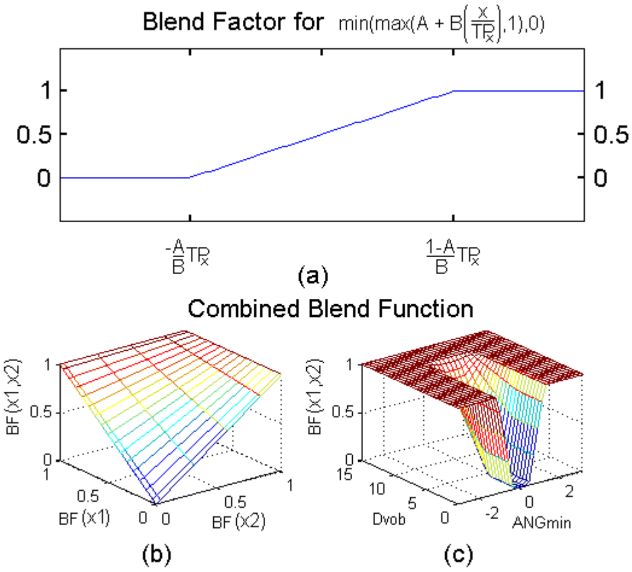

Figure 5: (a) Single factor blend function; (b) $Bf(x_1,x_2)$ created from $Bf(x_1)$ and $Bf(x_2)$; (c) $Bf(x_1,x_2)$ created from system variables

From Figure 5(b) and (c) it can be seen that the blend factor ($Bf(x_1,x_2)$) tends toward one and hence the blend function tends towards the environmental vector field when the obstacles are far away from the AGV and the AGV has clear space between it and the waypoint. Similarly it can be seen that the blend factor tends towards zero and hence the blend function tends towards the obstacle vector field when the obstacles are close to the AGV and when there are obstacles directly between the AGV and the waypoint.

3.2 Rotational obstacle vector field

With the use of a blend function to combine vector fields it was seen that the AGV was able to respond solely to the obstacle vector field during instances where an obstacle was blocking the path to a waypoint. Since the obstacle vector field no longer needed to overpower the environmental vector field it became possible to use a rotational force instead of a repulsive force. The rotational force used for the obstacle vector field allows the AGV to be guided around anything blocking its path instead of being forced away form it.

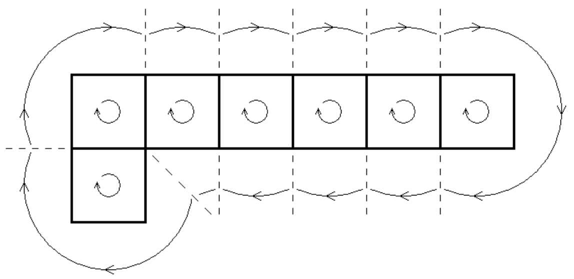

Figure 6: Desired vector field around a cluster of obstacles

The use of the blend function instead of superposition meant that the obstacle vector field needed to have a constant force instead of a force that diminished with respect to the distance from the obstacle. Using the diminishing force method allows the closest obstacle to have the greatest effect on the overall vector field and this characteristic was still desirable while using rotational obstacle vector fields. To achieve this only the force from the closest obstacle to the AGV was used to create the obstacle vector field, an example of this is shown Figure 6.

3.3 Rotational direction

Since the obstacle vector field is taken as the rotational force from the closest obstacle to the AGV it becomes important to be able to assign the direction of rotation, especially when dealing with a cluster of obstacles. It can be seen from Figure 7 that by assigning some obstacles clockwise and others counter clockwise rotational forces a cluster of obstacles can be given a separation line and a reattachment line. The separation line is of interest since it determines which direction the vehicle will travel while moving around the obstacle. Whereas the attachment line must be placed correctly otherwise it may cause the vehicle to move in an undesirable manner.

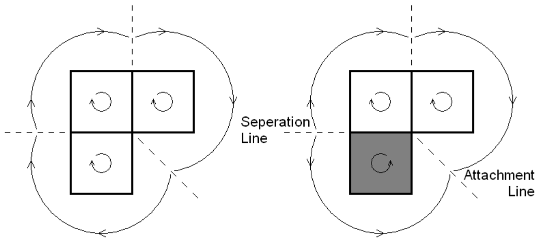

Figure 7: (a) Cluster with the same rotation direction; (b) Cluster with separation point and attachment point

The separation and attachment lines were chosen to be attached to the obstacle closest to the waypoint. This was done to move the attachment line to the waypoint side of the obstacle where the blend function tends towards the environmental vector field more than the obstacle vector field. To determine the rotational direction of the other obstacles a line was drawn between the centre of the chosen obstacle and the waypoint. The obstacles were assigned clockwise or counter clockwise rotational forces based on which side of the line they were on in order to place the separation and attachment lines where they were required. The chosen obstacle can be assigned either direction but for the duration of this study it was made clockwise, as shown in Figure 8.

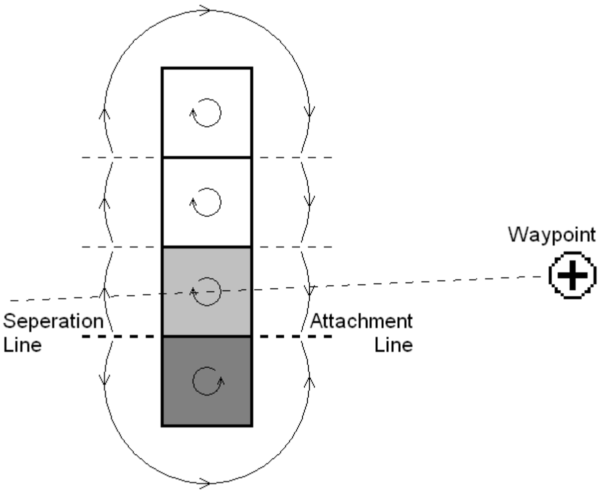

Figure 8: Method for selecting clockwise and counter-clockwise rotation

4 Navigating around static obstacles

To verify the theory in the previous section a set of test were preformed. Each of these tests consisted of the AGV being required to travel from a starting position to a waypoint while avoiding a single cluster of obstacles. The cluster varied in position, configuration and size for each of the tests. Three starting positions were chosen for each configuration of obstacles and each time the AGV was required to reach the same waypoint.

4.1 Blended vector fields

From the principles discussed in the previous section it was possible to create vector fields for areas with static obstacles. Shown in Figure 9 are four examples of such vector fields. The separation line can be seen in each of the examples starting at the closest block to the waypoint, marked in light grey and moving to the left of the plot. For all of the examples the attachment point is obscured due to the blend function heavily weighting the environmental vector field. These examples show that the blend function is capable of shifting focus from the environmental vector field to the obstacle vector field and back again at various positions on the arena.

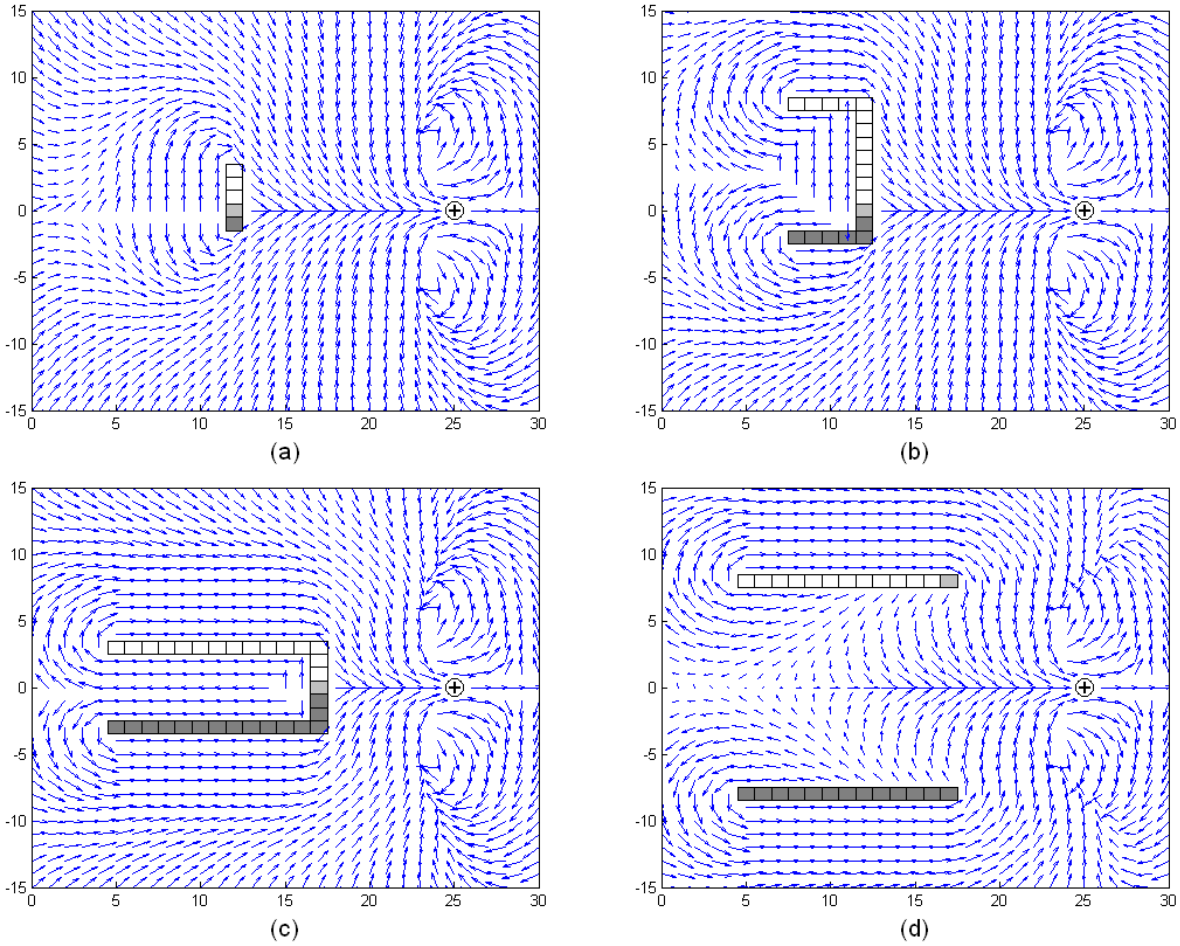

Figure 9: (a) Vector field from a small wall; (b) Vector field from a small dead end; (c) Vector field from a deep dead end; (d) Vector field from parallel walls

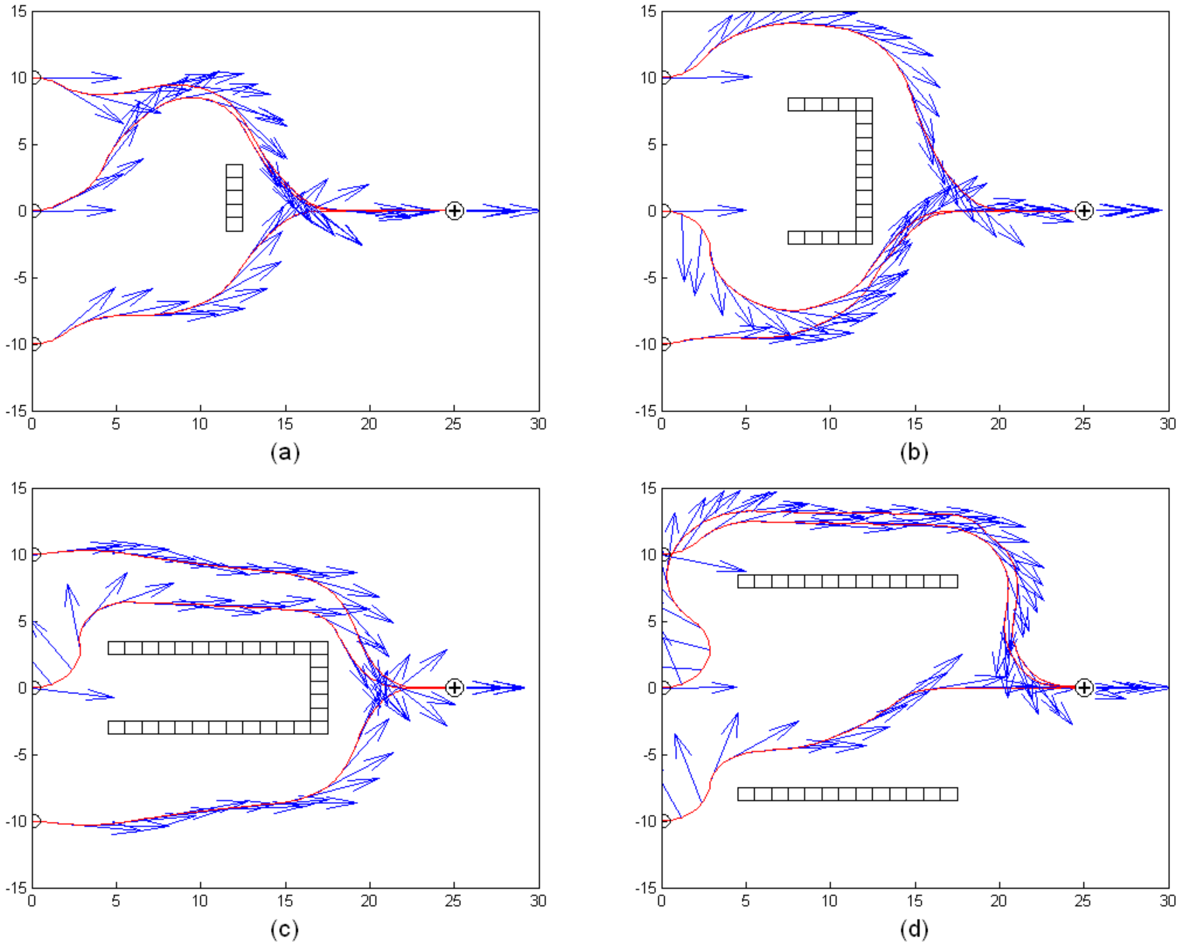

Figure 10: (a) Vehicle paths from a small wall; (b) Vehicle paths from a small dead end; (c) Vehicle paths from a deep dead end; (d) Vehicle paths from parallel walls

4.2 AGV simulation

The vector fields above were created with complete knowledge of the simulation environment which included the position of all objects. In the AGV simulations shown in Figure 10 prior knowledge of the environment was limited to the position of the waypoint. The vector field created at any given time only used the obstacles the vehicle could currently locate and those which the simulation stores in memory. Due to this fact the AGV path does not necessarily follow where the above vector field indicates it would. This, however, does not alter the result of the simulation, in all tests shown above the AGV was able to avoid the obstacles and reach the desired goal.

The way the blend function affects navigation can be seen from the results shown in Figure 10. The blend function (Equation (3)) tends towards zero as the vehicle approaches the obstacle which heavily weights the obstacle vector field allowing the vehicle to travel around the obstacle. The reverse effect can an also be seen, a good example of this is Figure 10(d) when the vehicle starts at (0,-10). Shortly after starting the vehicle is travelling parallel to the obstacles, when the vehicle is at approximately (8,-5) the blend function adds weight to the environmental vector field. The vehicle reacts to the environmental vector field and travels directly to the waypoint.

4.3 Trap Evasion

One of the shortfalls of vector field navigation is that it is susceptible to the dead end trap [Borenstein and Koren, 1989; Vadakkepat *et al.*, 2000]. The most common method used to evade this trap is to incorporate a secondary navigation method such as the wall following technique. The secondary navigation method is triggered by specific set of circumstances such as the vehicles heading with respect to the environment. By using the navigation method outlined in Section 3 the vehicle was able to avoid this trap without using a secondary navigation method. Figure 11(a) shows the vector field created with full knowledge of the environment. The separation and attachment lines can clearly be seen along the Y-axis. The vehicle paths, Figure 11(b), show how the vehicle behaves when confronted with the dead end trap. Although the vehicle does not have full knowledge of the system it is still able to evade the trap and reach the waypoint.

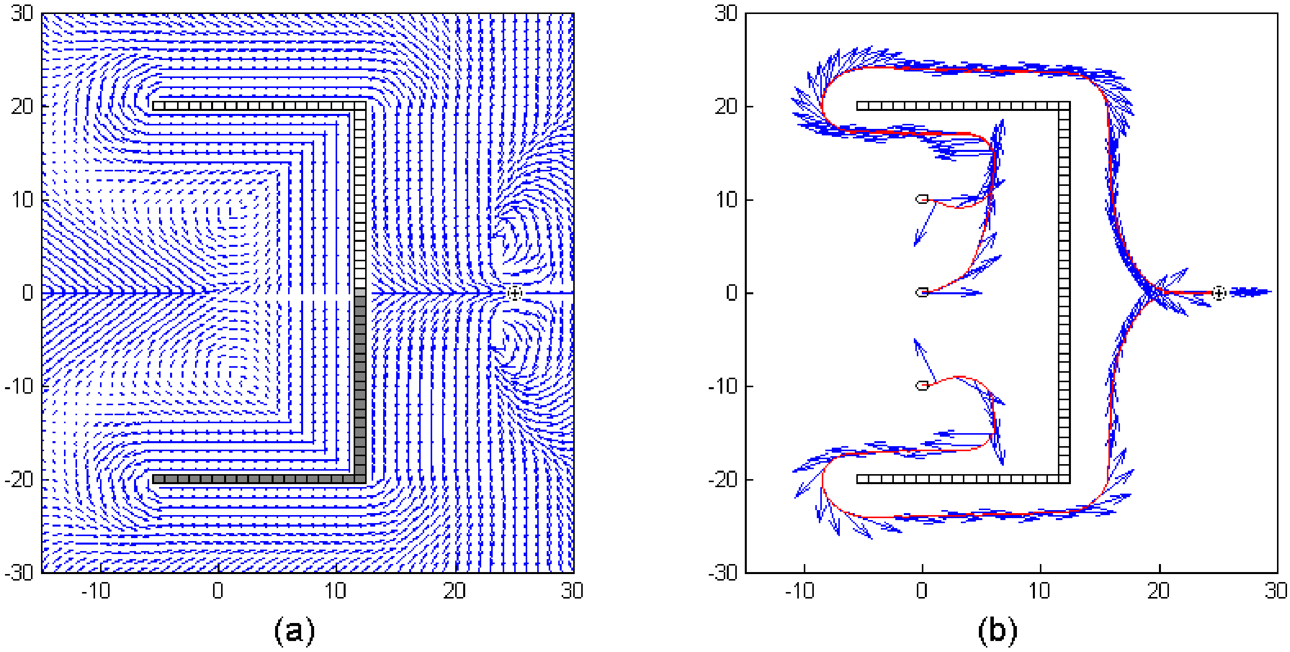

Figure 11: (a) Vector field of the trap; (b) Vehicle paths from within the trap

5 Conclusion

By modifying the method used to create and integrate vector fields into a navigation system this paper has introduced a method of navigation that is able to solve one of the weaknesses of vector field navigation. With the use of a blend function instead of using unweighted addition to combine vector fields it was shown that obstacle avoidance can be achieved with a rotational instead of a repulsive vector field. This obstacle avoidance method acted to move the vehicle around, instead of away from, obstacles making the navigation system more versatile. The results in Section 4.2 indicate that the vehicle is able to reach the desired goal while avoiding a variety of obstacles. Furthermore the results in Section 4.2 show that the navigation method is also able to guide a vehicle out of the dead end trap.

References

[Borenstein and Koren, 1989] J. Borenstein and Y. Koren. Real-Time Obstacle Avoidance for Fast Mobile Robots. IEEE Transactions on Systems, Man, and Cybernetics. Vol. 19, No. 5, pp. 1179-1187, September/October 1989.

[Kim *et al.*, 1999] Yong-Jae Kim, Dong-Han Kim and Jong-Hwan Kim. Evolutionary Programming-Based Uni-vector Field Method for Fast Mobile Robot Navigation. SEAL'98, LNCS 1585, pp. 154-161, 1999.

[Liddy *et al.*, 2007] Tommie Liddy and Tien-Fu Lu, Waypoint Navigation with Position and Heading Control using Complex Vector Fields for an Ackermann Steering Autonomous Vehicle. Australasian Conference on Robotics and Automation, December 2007.

[Loizuo *et al.*, 2003] Savvas G. Loizou, Herbert G. Tanner, Vijay Kumar and Kostas J. Kyriakopoulos. Closed Loop Motion Planning and Control for Mobile Robots in Uncertain Environments. IEEE Conference on Decision and Control, Vol. 3, pp. 2926-2931, 2003.

[Vadakkepat *et al.*, 2000] Vadakkepat, P, KC Tan and M L Wang. Evolutionary artificial potential field and their application in real-time robot path planning. Proc of IEEE Congress on Evolutionary Computation, pp. 256-263, La Jolla, CA, United States, Jan 2000.