Reactive Navigation of an Autonomous Ground Vehicle Using Dynamic Expanding Zones

Joseph Satoru Putney

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Masters of Science
In
Mechanical Engineering

Dr. Charles F. Reinholtz, Chairman
Dept. of Mechanical Engineering

Dr. Alfred L. Wicks
Dept. of Mechanical Engineering

Dr. Dennis W. Hong
Dept. of Mechanical Engineering

May 11, 2006
Blacksburg, Virginia

Keywords: Reactive Navigation, Autonomous Vehicles, Mobile Robots, Unmanned Ground Vehicle, Fuzzy Logic, Dynamic Expanding Zones

Reactive Navigation of an Autonomous Ground Vehicle Using Dynamic Expanding Zones

Joseph Satoru Putney

ABSTRACT

Autonomous navigation of mobile robots through unstructured terrain presents many challenges. The task becomes even more difficult with increasing obstacle density, at higher speeds, and when a priori knowledge of the terrain is not available. Reactive navigation schemas are often dismissed as overly simplistic or considered to be inferior to deliberative approaches for off-road navigation. The Potential Field algorithm has been a popular reactive approach for low speed, highly maneuverable mobile robots. However, as vehicle speeds increase, Potential Fields becomes less effective at avoiding obstacles.

The traditional shortcomings of the Potential Field approach can be largely overcome by using dynamically expanding perception zones to help track objects of immediate interest. This newly developed technique is hereafter referred to as the Dynamic Expanding Zones (DEZ) algorithm. In this approach, the Potential Field algorithm is used for waypoint navigation and the DEZ algorithm is used for obstacle avoidance. This combination of methods facilitates high-speed navigation in obstacle-rich environments at a fraction of the computational cost and complexity of deliberative methods.

The DEZ reactive navigation algorithm is believed to represent a fundamental contribution to the body of knowledge in the area of high-speed reactive navigation. This method was implemented on the Virginia Tech DARPA Grand Challenge vehicles. The results of this implementation are presented as a case study to demonstrate the efficacy of the newly developed DEZ approach.

Acknowledgements

My work in this thesis and success throughout my engineering career would not have been possible without the support of many people. First, I would like to thank my family for always offering support and love.

I would like to express my deepest appreciation to my committee chair, Dr. Reinholtz, who has been instrumental in my successful college career. Without his advice and encouragement, I would not be working on autonomous vehicles today. Also, thanks to my committee members, Alfred Wicks and Dennis Hong, who offered guidance and support.

Finally, I would like to give a special thanks to everyone on the 2005 Virginia Tech Grand Challenge Team! Without the whole team, Cliff and Rocky would not have been as successful in the Grand Challenge Event. Thanks to my graduate student mentors, Andrew Bacha and Brett Leedy, for the endless encouragement, trust, and answers to my many questions. I would like to especially thank Ernest Velasquez for originally coming up with the "box" idea.

Table of Contents

Acronyms and Abbreviations

AIO	Analog Input/Output
ASCII	American Standard Code for Information Interchange
CEP	Circular Error Probable
DARPA	Defense Advance Research Projects Agency
DEZ	Dynamic Expanding Zones
DGC	DARPA Grand Challenge
DIO	Digital Input/Output
GCE	Grand Challenge Event
GPS	Global Positioning System
ICR	Instantaneous Center of Rotation
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
I/O	Input/Output
LADAR	Laser Detecting and Ranging
NI	National Instruments
NQE	National Qualifying Event
PID	Proportional-Integral-Derivative Controller
PLGR	Rockwell Collins Precision Lightweight GPS Receiver
RDDF	Route Definition Data File
TALIN	Honeywell Tactical Advanced Land Inertial Navigator
UDP	User Datagram Protocol
UTM	Universal Transverse Mercator
VT	Virginia Tech
VTGC	Virginia Tech Grand Challenge
ZOH	Zero Order Hold

List of Figures

List of Tables

Chapter 1
Introduction

The Dynamic Expanding Zones algorithm presented in this thesis was developed to as part of the Virginia Tech effort to compete in the 2005 DARPA Grand Challenge. A student-faculty team from Virginia Tech entered two vehicles to compete in the 130+ mile off-road race through the Mojave Desert. This chapter first gives an overview of the thesis organization. The last two sections cover the goals and rules for the Grand Challenge Competition. The rules for the competition were used to define the problem for the reactive navigation algorithm discussed in this thesis. Although the Dynamic Expanding Zones algorithm was developed for the Grand Challenge, the method can be broadly applied to other autonomous ground vehicles for a wide variety of tasks.

1.1 Thesis Overview

Reactive navigation schemas are often dismissed as overly simplistic or considered to be inferior to deliberative approaches for off-road navigation. The Potential Field algorithm has been a popular reactive approach for low speed, highly maneuverable mobile robots. However, as vehicle speeds increase, Potential Fields becomes less effective at avoiding obstacles. This thesis explains how the traditional shortcomings of the Potential Field approach can be largely overcome by using dynamically expanding perception zones to help track objects of immediate interest.

The first chapter explains the motivation for the development of Dynamic Expanding Zones. The Grand Challenge rules define the problem and mission which the navigation software must solve and execute. Chapter 2 gives an overview of the base platform, drive by wire system, sensor package, and architecture this algorithm was designed for. Chapter 3 is a literature review of the three main navigation paradigms: reactive, deliberative, and hybrid reactive-deliberative.

The Dynamic Expanding Zone algorithm is a reactive navigation algorithm that provides a computationally simple means for avoiding obstacles at high speed. As a reactive strategy, Dynamic Expanding Zones uses concurrent behaviors to command steering and propulsion directly from sensor data. Instead of commanding a trajectory or a path, the resulting path of a reactive strategy is emergent. The commands are recalculated each iteration of the algorithm. Chapter 4 discusses how Dynamic Expanding Zones fits into this reactive architecture.

The following four chapters give a detailed description of each concurrent behavior. Chapter 5 describes the implementation of the Potential Fields algorithm for waypoint navigation. Road following, explained in Chapter 6, uses a hybrid Pure Pursuit-Potential Fields algorithm. Chapter 7 focuses on obstacle avoidance using the newly developed Dynamic Expanding Zone algorithm. Chapter 8 concentrates on the rollover prevention behavior.

The combination of these concurrent behaviors facilitates high-speed navigation in obstacle-rich environments at a fraction of the computational cost and complexity of deliberative methods. Chapter 9 evaluates the advantages and disadvantages of each behavior and suggests future research and improvements.

1.2 Grand Challenge Competition Background

The Defense Advanced Research Projects Agency (DARPA) developed the Grand Challenge in response to the congressional mandate to make one-third of the operational ground combat vehicles unmanned by 2015 [1]. The goal of DARPA was to "bring together individuals and organizations from the research and development community, industry, Government, the Armed Services, academia, professional societies, and from the ranks of students, backyard inventors, and automotive enthusiasts" to accelerate autonomous ground vehicle technology [2]. The second Grand Challenge was held on October 8, 2005 with a cash prize of $2 million. This paper focuses on the software development and implementation for the second Grand Challenge competition.

Virginia Tech produced two off-road autonomous vehicles, Cliff and Rocky, to compete in the Grand Challenge. Both vehicles were selected for the National Qualifying Event (NQE) and went on to the main Grand Challenge Event (GCE). In the final even, Cliff finished 8th and Rocky finished 9th. Cliff stopped at the 44 mile marker, because the drive train engine stalled. Rocky failed at the 39 mile marker due to a generator failure. Before the mechanical failures occurred, both vehicles were successfully navigating at speeds up to 25 mph.

1.3 Competition Rules

The description of rules presented in this section is adapted from the rules posed on the DARPA website [3]. To win the cash prize, a team had to be the fastest team to complete the 132 mile course within 10 hours. The course included paved/unpaved roads, trails, and off-road desert terrain. Examples of obstacles given by DARPA were "ditches, berms, washboard, sandy ground, standing water, rocks and boulders, narrow underpasses, construction equipment, concrete safety rails, power line towers, barbed wire fences and cattle guards" [3]. Although the course was not provided to the competitors until two hours before the race, the teams were assured that the course would be traversable by a commercial 4x4 pickup truck.

Each vehicle was required to complete the course autonomously without any human interaction. The vehicles were required to be "propelled and steered by traction with the ground" [3]. The vehicle dimensions were limited to 10 feet in width and 9 feet in height with a maximum weight of 20 tons.

The route definition data file (RDDF) was the official definition of the route and defined the corridor through which all vehicles were required to travel. The RDDF contained waypoints, lateral boundary offsets (LBO), and maximum speed limits. Vehicles could navigate any area within the boundaries, but they were required to avoid any obstacles. The track line, line connecting waypoints, was not guaranteed to be free of obstacles; therefore, vehicles were expected to determine the best way to navigate from waypoint to waypoint.

The lateral boundary offsets define the width of the course corridors. They are measured perpendicular to the line connecting the two waypoints (Figure 1-1). Vehicles that leave the boundaries could be disqualified. At the endpoints each path segment, the boundary was defined by semicircle centered on the waypoint with a radius equal to the LBO. The boundary of the course is defined by the outer boundary (Figure 1-2) [4].

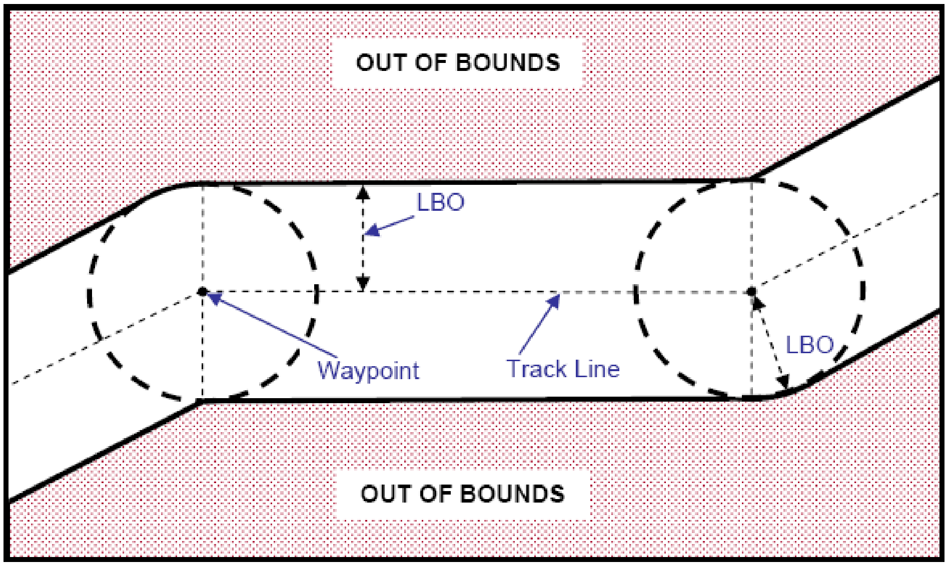


Figure 1-1. DARPA course defined by Lateral Boundary Offset (LBO) and waypoints [4].

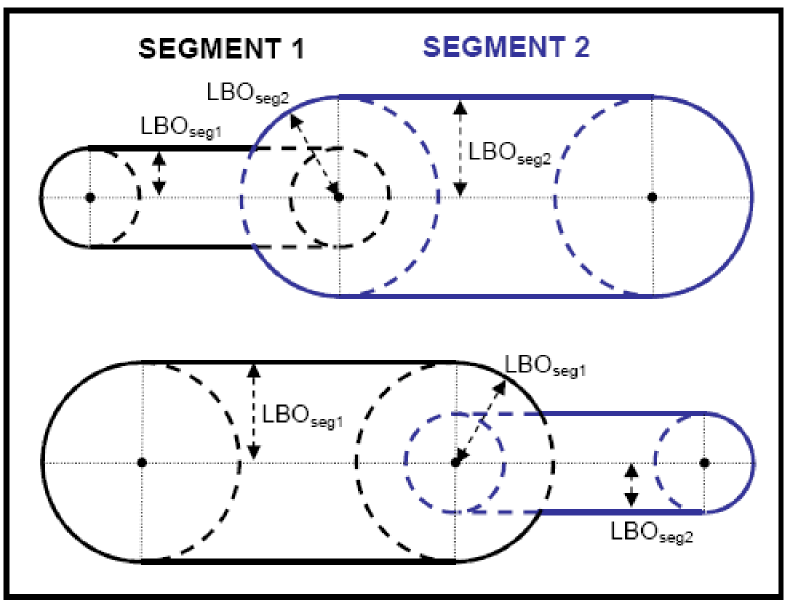


Figure 1-2. Boundary of the route is defined by the least restrictive segment boundary. The solid lines indicate the actual boundaries [4].

Chapter 2
Virginia Tech Vehicles

Virginia Tech produced two off-road autonomous vehicles to compete in the Grand Challenge Event. Both vehicles, shown in Figure 2-1, were designed off the same base platform with the same sensors and drive by wire hardware. Although both vehicles were physically similar, they were designed to run with completely different navigation approaches. A reactive strategy was developed to run on Cliff, while a deliberative path planning strategy was developed for Rocky [5]. These competing strategies were developed and evaluated in parallel for the Challenge. Since the reactive approach was the furthest developed, both vehicles employed the reactive algorithm for the Grand Challenge. For didactic purposes, any reference to the vehicle hereafter will be to the vehicle Cliff. The slight differences between Cliff and Rocky are insignificant in the explanation of Dynamic Expanding Zones. This chapter will discuss the design of the various subsystems of the Virginia Tech vehicles: base platform, drive by wire system, sensors, and system architecture.

Figure 2-1. Virginia Tech's entries to the 2005 DARPA Grand Challenge, Cliff (left) and Rocky (right)

2.1. Base Platform

Ground navigation through an unstructured, desert terrain presents many challenges in vehicle design. Not only must the vehicle be sufficiently stable enough to traverse washboard, sand, mud, shallow water, and gravel, the vehicle must be adequately maneuverable to safely navigate obstacle rich terrain at high speeds.

Mobile robots have utilized various wheel types and geometries ranging from a two wheeled bicycle to a Swedish omni-directional wheel configuration. The type of geometry is typically driven by the application for the mobile robot. Based on the environment the vehicle has to navigate, the designer must choose a platform that provides the ideal amount of maneuverability, controllability, and stability. For the Grand Challenge, Virginia Tech used an Ackerman steered vehicle, which provides the required stability for off-road terrain, while providing enough maneuverability to navigate the course.

From a practical standpoint, static stability can be ensured by three wheels as long as the center of mass is within the triangle formed by the points of contact [6]. Increased stability can be obtained by adding wheels. Additional wheels cause the vehicle to be over-constrained, requiring the vehicle to have a suspension to ensure each wheel is in contact with uneven terrain. An Ackerman steered vehicle is an example of a wheel configuration with two fixed wheels in the rear and two steerable wheels in the front (four total points of contact). As shown in Figure 2-2, the inside tire (of a turn) must be steered at a larger angle than the outside tire to theoretically eliminate slip/skid of the front tires. This difference in angle allows the longitudinal force of each tire to be pointed perpendicular to the turning radius of the corresponding tire. An off-road suspension allows an Ackerman steered vehicle to maintain stability in the unstructured terrain of the Mojave desert.

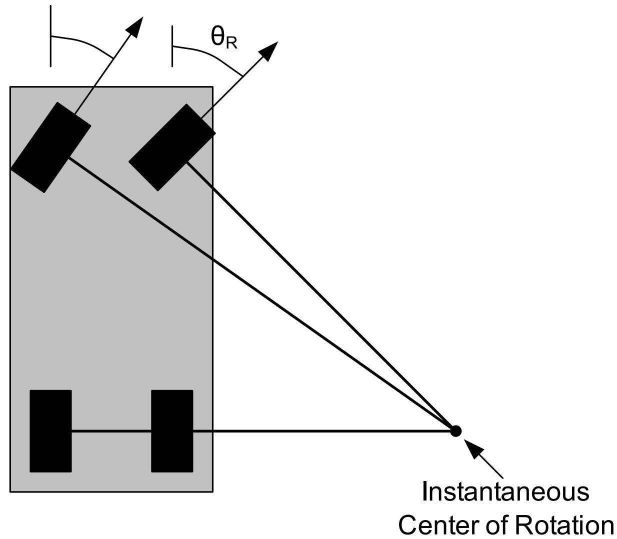


Figure 2-2. Ackerman steered vehicle geometry [8]

The slight disadvantage to having a stable Ackerman steered wheel configuration is the limited maneuverability [6]. The nonholonomic wheel geometry prevents the vehicle from making zero radius turns and traveling in the lateral direction. The vehicle must perform a series of forward and reverse maneuvers to move laterally. The maneuverability limitation actually improves lateral stability in high speed turns. Ackerman steering is also much easier to control than such geometries as differential steering and omni-directional steering. Ackerman steering has 2 degrees of freedom: steering angle and propulsion.

The Virginia Tech Grand Challenge (VTGC) base vehicles are Ingersoll-Rand Club Car XRT 1500 utility vehicles. The XRT 1500 is extremely agile Ackerman steered vehicle with a small turning radius of 11.5 feet. The vehicle also drives off-road at a top speed of 25 mph with auto-engaging four wheel drive and off-road run flat tires. The stock vehicle weighs 1250 pounds and has the capability of carrying a 1050 pound payload. This payload capacity was easily sufficient for the autonomous conversion package. An air-cooled 20hp Honda GX620 gasoline engine supplies power to the drive train.

2.2. Drive by Wire Conversion

To enable full computer control of the vehicle actuation systems, the throttle, steering, and brake were converted to drive-by-wire (DBW). Steering and throttle are both controlled by DC servo motors with integrated quadrature encoders. The braking system utilizes an electronically controlled hydraulic pump.

The steering wheel and column were removed from the vehicle to allow for a right angle gear motor to couple directly to the shaft of the stock steering rack and pinion system. The throttle DC servo motor attached to the throttle cable using a pulley. Both of these motors are controlled using PID position control (Figure 2-3). The PID controllers were tuned to provide the fastest, stable response at all times. The reactive navigation does not require control over steering and throttle rates. The maximum steering rate was approximately equal to 34 degrees/second. For steering and throttle response data, please refer to Appendix A.

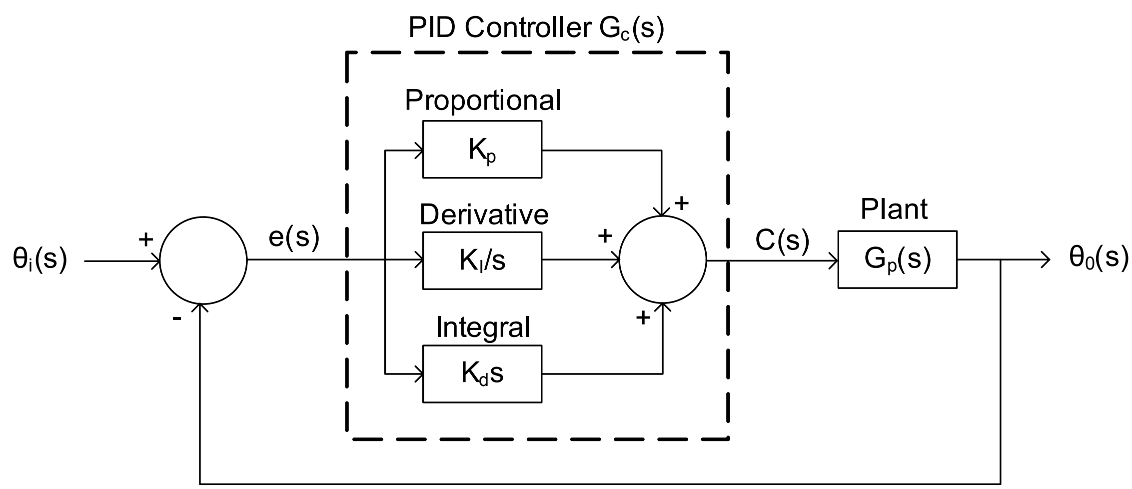


Figure 2-3. PID control for motors, where θ_i, θ_o, e(s), C(s), and plant are commanded angular motor position, actual angular motor position, error, motor input voltage, and the motor behavior, respectively.

The vehicle speed is controlled by a closed-loop PID with the throttle PID in series (Figure 2-4). This PID requires speed as an input and outputs a throttle position. In turn, this throttle position goes through the throttle PID to output a desired throttle motor voltage.

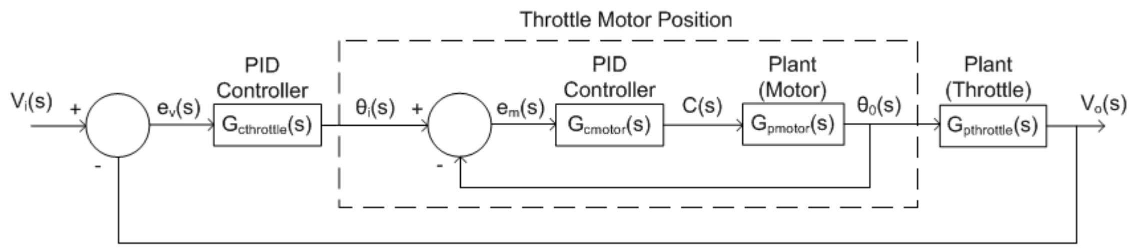


Figure 2-4. PID control for speed, where V_i, V_o, and $e_v(s)$ are input speed, output speed, and speed error, respectively

The throttle and brake control run in parallel; however, the vehicle will never attempt to throttle while braking and vice versa. The drive by wire brake was implemented by replacing the master cylinder and brake pedal assembly with an electronically controlled hydraulic pump. For the competition, the braking system used an open-loop control shown in Figure 2-6 to translate a desired reduction in speed, e(s), to the appropriate brake percent command, C(s), for the hydraulic brake driver. After the competition, a closed loop PID brake control shown in the bottom of Figure 2-5 has been implemented, where acceleration is the input and brake pressure is the output.

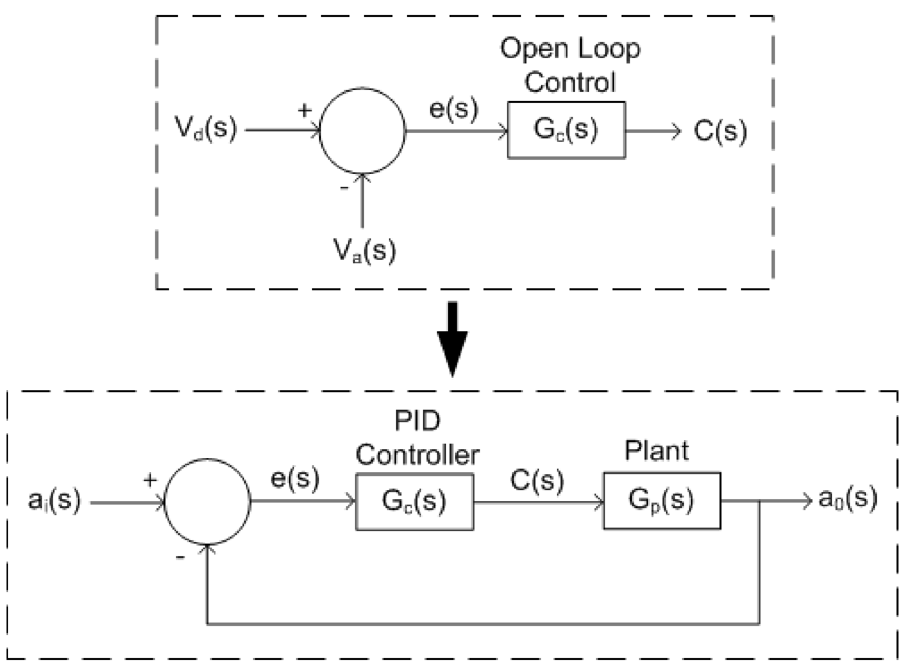


Figure 2-5. The top system illustrates the open loop control of the brakes. The bottom system illustrates the new PID controlled system.

2.3. Sensors

The VTGC vehicles use a sensor suite which includes a GPS/INS for global positioning, LADAR for obstacle detection, and stereo vision camera for road detection/positioning (Figure 2-6). Each sensor obtained data for a unique task, so no sensor fusion was necessary. See Chapter 3 for details on why each sensor is coupled to a behavior in a reactive architecture.

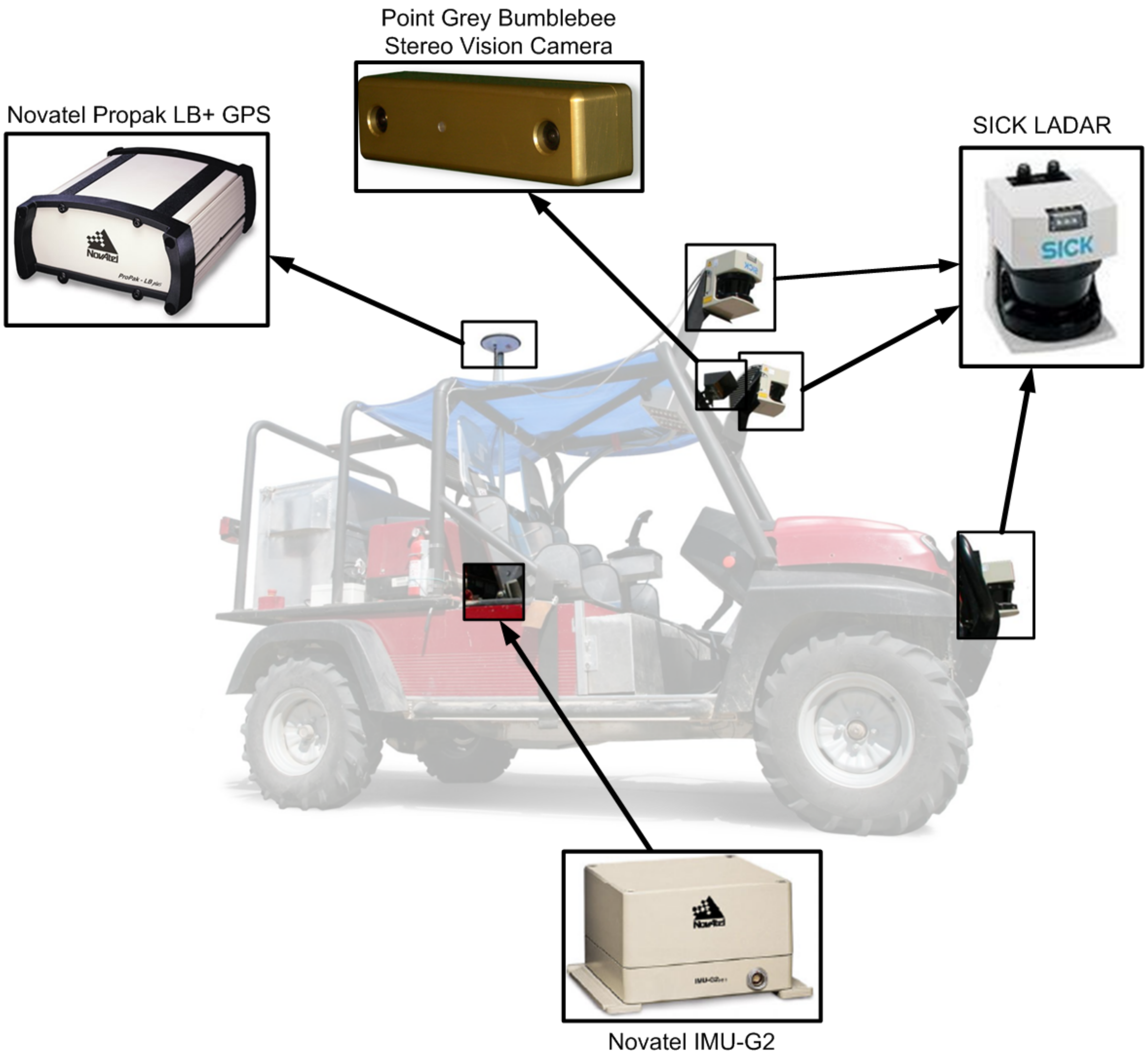


Figure 2-6. Photo of sensors and their corresponding locations.

2.3.1. Positioning Sensor

The Grand Challenge competition required the vehicles to navigate a course defined in the Latitude-Longitude global coordinate reference system. A Global Positioning System (GPS) sensor is a true position sensor which will provide accurate

Latitude-Longitude position. A GPS with different corrections can provide position with sub-meter Circular Error Probable (CEP) accuracy. Sub-meter accuracy is sufficient for a mid-size vehicle to navigate the DARPA defined course.   However, DARPA specified that the vehicles must be able to navigate through tunnels, where access to satellites is lost.   An Inertial Measurement System (IMU) is an excellent relative position sensor to use over short distances. As the traversed distance increases, the position error increases. By fusing the GPS and IMU together, the vehicle can obtain accurate position, while in open land and in short tunnels.

Novatel provides a system that includes a Novatel Propak-LBplus GPS receiver and a Novatel IMU-G2 enclosure housing a Honeywell HG1700 IMU (Inertial Measurement Unit) (Figure 2-7). The Propak-LBplus unit provides single-point position accuracy of 1.5m CEP. This accuracy is increased to 10cm CEP by L-band differential corrections through the subscription service, OmniSTAR. The position, velocity, and pose data from the Propak-LBplus is collected at 20 Hz.

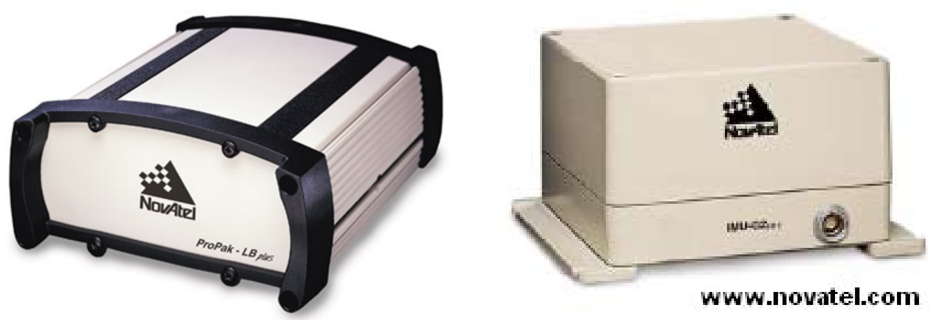


Figure 2-7. Novatel Propak-LBplus GPS receiver (left) and a Honeywell HG1700 IMU in a Novatel IMU-G2 enclosure (right).

2.3.2. Obstacle Detection

A single SICK LADAR is used to detect obstacles by sweeping 180 degrees with 0.5 degree resolution along the horizontal plane in front of the vehicle. The LADAR returns an instantaneous 2-D polar coordinate array of the range and angle to any solid objects in the sensor's viewing plane (Figure 2-8). The SICK LMS-290 can return distances up to 70m with 10mm resolution at a rate of 75 Hz. Dynamic Expanding Zones used a horizontal scanning LADAR for obstacle detection; however, the algorithm is not limited to this configuration or sensor. The algorithm only requires obstacle data positioned in the vehicle frame of reference.

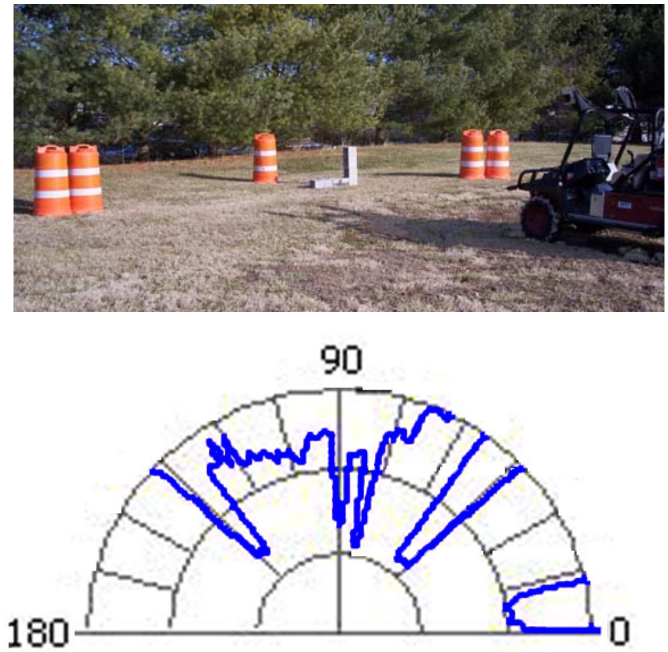


Figure 2-8. Single plane LADAR polar plot [9].

2.3.3. Vision

The course of the Grand Challenge competition was assumed to follow desert service path. The VTGC vision system is designed to identify roads and adjust the path of the vehicle to navigate along the center of the road. A Point Grey Bumble Bee stereo vision camera, mounted to the top center of the vehicle's roll cage, is used to observe the area in front of the vehicle (Figure 2-9). The Bumble Bee camera is capable of outputting progressively scanned 640x480 stereo images at 30Hz. The stereo processing algorithm operates at approximately 5Hz.

Figure 2-9. Point Grey Research Bumblebee stereovision camera

If an area of the image is determined to be a road, center points are calculated and sent to the stereo processing software in order to convert the pixel locations to coordinates in the vehicle reference frame (Figure 2-10). These coordinates are rechecked in order to ensure no discontinuities exist and the coordinates are on the same plane as the vehicle. If every check is confirmed, the road center points are sent to the navigation algorithm.

Figure 2-10. Virginia Tech road recognition data.

2.4. System Architecture

The Virginia Tech autonomous package included three 1.2 GHz Pentium processors and National Instruments motor controller, analog I/O, and digital I/O. The three computers each perform a specific task: vision, sensors/navigation, and motion control. The sensors/navigation computer determines the current position of the vehicle and waypoint location, while identifying obstacles in front of the vehicle. The vision computer uses stereo and monocular vision techniques to recognize roads. The information from the vision computer is passed to the sensors/navigation computer, which determines the appropriate navigation behavior based on the LADAR, GPS, and vision data. The motion control computer executes propulsion and steering commands from the sensors/navigation computer by PID control of all vehicle actuators. Figure 2-11 illustrates the network architecture of the three computers. Communication between the computers utilizes the User Datagram Protocol (UDP). The majority of the software for the 2005 Grand Challenge competition was developed using National Instruments

LabVIEW. Appendix B discusses the advantages to using this graphical software development tool.

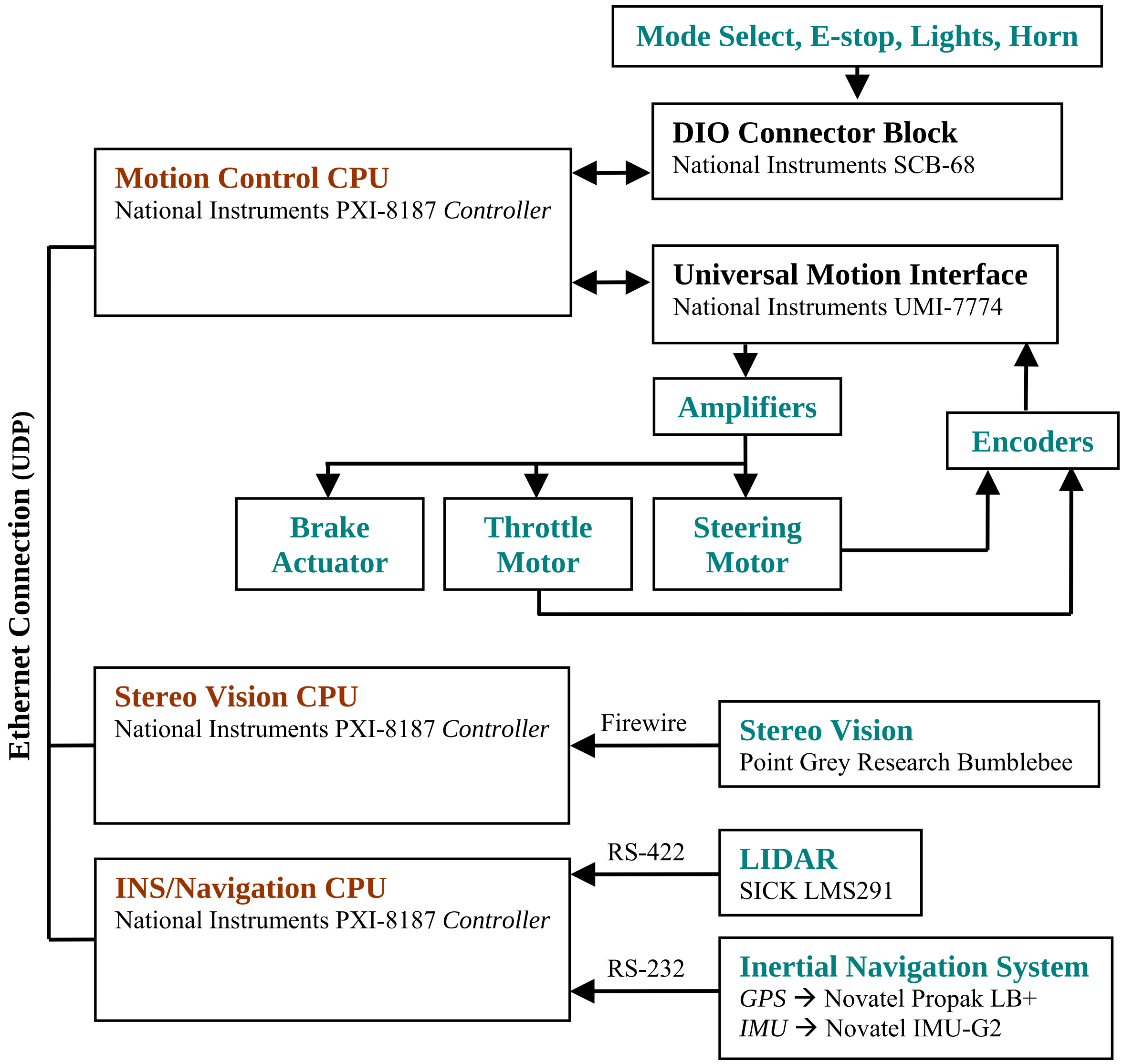


Figure 2-11. System level data flow diagram of VTGC Vehicle.

Chapter 3
Navigation Paradigms

Currently, the three main paradigms for autonomous ground mobile robot navigation are reactive, deliberative, and hybrid reactive-deliberative. These paradigms can be described by the three primitives of robotics: Sense, Plan, and Act [5]. The Sense primitive includes any function that parses sensor data and/or converts the data into another form useful for other functions. The Plan primitive is a function that uses the output of the Sense primitive or stored knowledge to create a group of tasks or trajectory for the Act primitive to execute. The Act primitive controls the actuators on the mobile robot. The following table, Table 3-1, gives an overview of the three navigation paradigms.

Table 3-1. Overview of the three navigation paradigms [5].

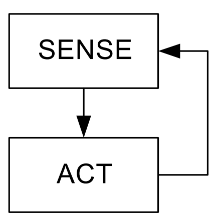
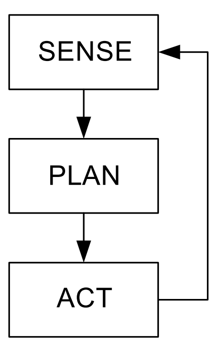
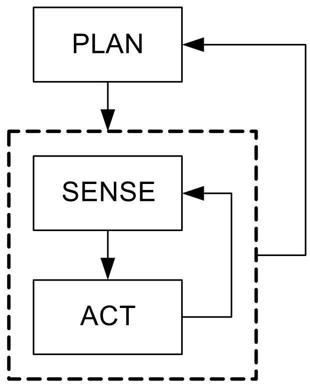
Navigation Paradigm	Architecture	Advantages	Disadvantages
Reactive		-Computationally Inexpensive -Concurrent Behaviors -Incremental Behavior Expansion	-Overall Behavior is Emergent -Limited Robustness -Can Become Complex
Deliberative		-Creates Optimal Trajectory -Reduces Actuator Effort	-Require Accurate Map -Computationally Expensive -Requires Drivable Trajectory

Table 3-1 (continued). Overview of the three navigation paradigms

Hybrid Reactive-Deliberative		-Creates Optimal Trajectory -Reduces Actuator Effort -Requires Less Computation than Deliberative	-Require Accurate Map -Vehicle Must Track Trajectory Accurately

3.1. Reactive Paradigm

A reactive paradigm utilizes two of the three primitives, Sense and Act (S-A), to navigate [5]. Instead of creating a path or trajectory, a reactive navigation scheme creates steering and/or propulsion commands to react to current snapshot of sensor data. Similar to a closed loop feedback system, the robot acts, changes the world, and modifies the action to react to the new world. The robot has no knowledge of where it will be in the future and no memory of where it has been. The overall behavior is dictated by the series of commands (emergent behavior) rather than one single trajectory.

Robin Murphy refers to this Sense-Act pair as a behavior [5]. To prevent the robot from only performing one type of action, multiple behaviors can be processed concurrently (Figure 3-1). Each concurrent behavior is independent of other behaviors. A behavior can be added or removed without affecting the other behavior. In addition, a change to one behavior will not affect other behaviors. As a result, each rule can be developed and tested independently without affecting one another. Not only does the parallel architecture support incremental development of robot capabilities, but it follows good modular software development practice.

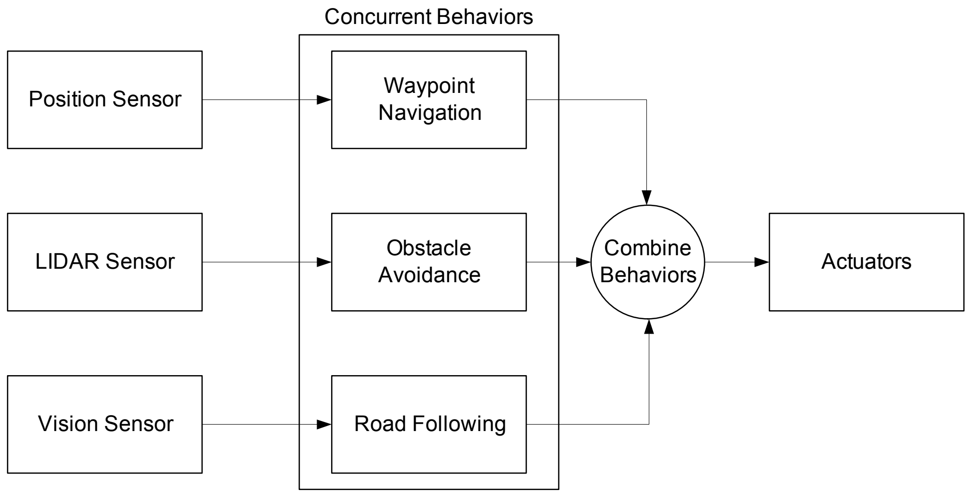


Figure 3-1. Example of the reactive paradigm architecture.

Since the reactive paradigm permits concurrent behaviors, it requires a means for combining these behaviors or selecting a single behavior to actuate steering and propulsion. In Robin Murphy's book, she lists Equilibrium, Dominance, and Fuzzy Logic as methods for combining/selecting the concurrent behaviors [5]. The Equilibrium Method sums the output of each behavior. Potential Fields is an example of a reactive algorithm which uses the Equilibrium Method to sum the steering vector of each behavior to create a resultant steering vector. The Dominance Method selects the behavior which has the highest priority or level. Subsumption is an example of an algorithm which higher level behaviors subsume lower level behaviors. Fuzzy Logic has also been utilized to combine parallel behaviors. For more details on Fuzzy Logic, see Chapter 4.

Since the sensing and acting is tightly coupled, the algorithm can operate extremely quickly. No computation is needed to map data to the earth frame or store data in a map. Instead, all sensor data, except for GPS data, is egocentric. Behaviors which use exteroceptive sensory data, such as LADAR range data, typically do not require complex manipulation of the data to make decisions. The data can retain its reference to the vehicle frame. Reactive behaviors typically have low computational complexity, often on the order of O(n) [5].

This low computational and development complexity comes at the expense of poor navigation decisions under certain circumstances. One of the most common problems with the reactive approaches is its propensity to become trapped in dead end conditions, such as cul-de-sacs. Shown in Figure 3-2, Potential Fields can lead a vehicle into a local minima (dead end), where the sum of the vectors equal zero. Reactive strategies also typically exhibit "jerky" motion on the perimeter of an obstacle, especially at high speeds. Like any discrete controller, the reactive navigation scheme is susceptible to overshoot (jerky motion) due to the lag of the control effort and the discrete measurement of the feedback. Since reactive approaches are susceptible to overshoot, these approaches are better suited for low speeds and vehicles with high controllability, such as zero radius turn vehicles.

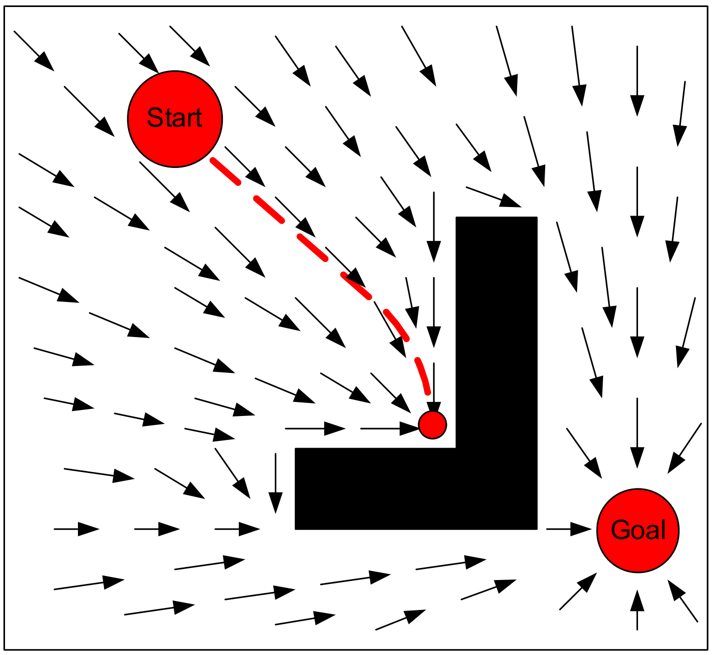


Figure 3-2. Local minima problem for Potential Fields

The behavior of a vehicle using Potential Fields is highly dependent on the magnitude profiles around obstacles (Figure 3-3). These profiles are not easily portable to other vehicles or even different applications with the same vehicle. A magnitude profile, which works for a vehicle traveling at 5 mph, will not work for the same vehicle traveling at 25 mph as shown in A) of Figure 3-4. The faster the vehicle travels, the larger the magnitude profile must expand to allow for the vehicle to react in time (Figure 3-4B). However, a slow speed robot will circumnavigate the obstacle with an excessively large maneuver if the magnitude profile is designed for a high speed vehicle (Figure 3-4C). The emergent overall behavior of the vehicle is very sensitive to the magnitude profile design. Many techniques have been developed to enable smooth navigation around obstacles at different speeds and prevent the local minima problem; however, they all greatly increase the complexity of the algorithm. One of the largest benefits to the reactive paradigm is its simplicity. If the reactive algorithm becomes too complex, other paradigms become more attractive.

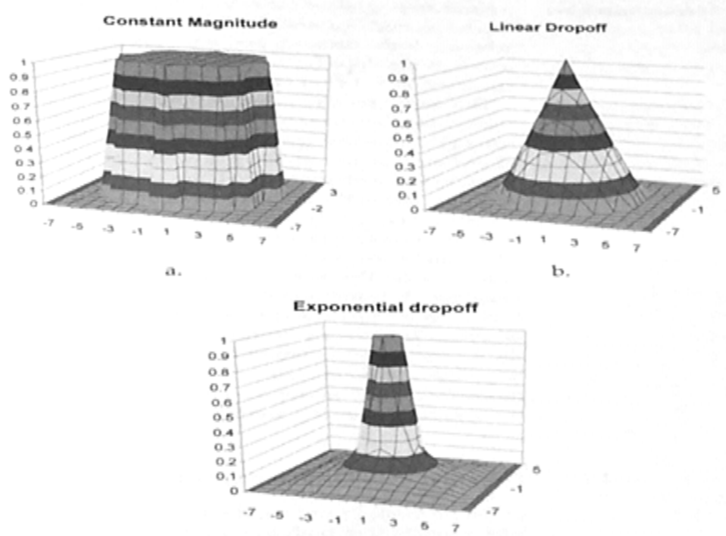


Figure 3-3. Magnitude profiles for Potential Fields [5].

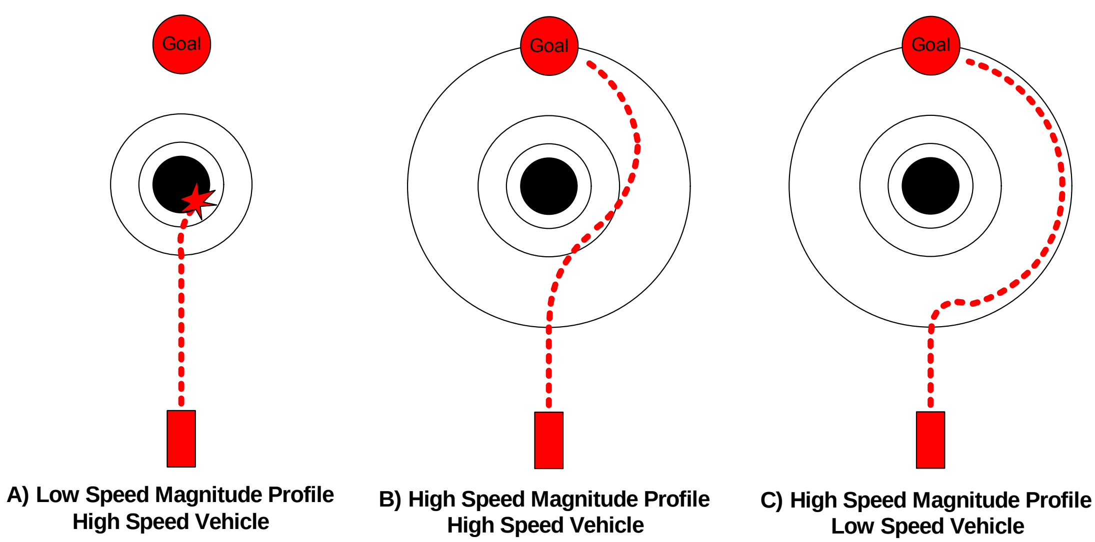


Figure 3-4. Magnitude profile effects on navigation. Similar to elevation lines on a topographic map, the concentric circles indicate positions of similar magnitudes of the repulsive field.

3.2. Deliberative Paradigm

Since designing behaviors to create the desired emergent behavior is more of an art than a science, the deliberative paradigm specifically attempts to generate the path [5]. A deliberative paradigm utilizes all three primitives, Plan, Sense, and Act (P, S-A), to navigate. This approach requires a map to intelligently plan an 'optimal' path. The map can be constructed from a fusion of sensor data to include terrain heights, traversability, obstacle locations, and/or a number of other necessary navigation parameters. A graph search algorithm, such as Dijkstra's algorithm and A*, uses a map where each cell is assigned a cost. A low cost corresponds to an easily traversable area, while a high cost corresponds to an impassable object. Using this map, a graph search algorithm will find the path of least cost to the goal point. Figure 3-5 illustrates a graph search optimal path (green) from the start point (red) to the goal point (blue). The black blocks correspond to obstacles, and the non-black blocks are traversable terrain.

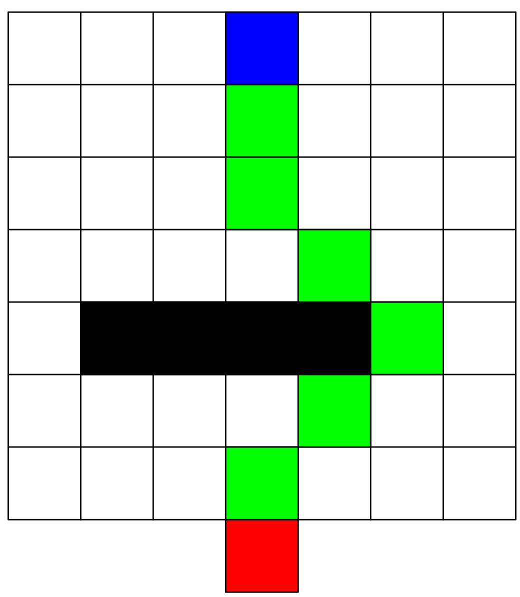


Figure 3-5. Example of a path least cost generated by a graph search approach.

The deliberative approaches have the most potential to be a robust navigation solution, because of its ability to deal with complex obstacle configurations, such as cul-de-sacs. However, this algorithm makes the assumption that the map is accurate, projected path can be accurately followed, and the software computational frequency is adequate for the required speeds.

The accuracy of the map is limited to the accuracy of the knowledge about the environment. An accurate map is extremely hard to create in an unstructured terrain without a priori knowledge. A vehicle driving with an inaccurate map is essentially a vehicle driving blind. The deliberative map can be constructed in two ways: instantaneously or populated over time. A map populated over time is susceptible to spurious sensor data. If the mapping software misidentifies an obstacle, the deliberative navigator does not have the intelligence to ignore the ‘false’ obstacle. Several algorithms exist to clear the graph of these false detections; however, this adds to the complexity of the software. In addition, maps which use stored data are also extremely susceptible to position errors. If the navigation unit outputs a position with an error of 5m in the Easting direction, all the obstacles in the map are now offset 5m East (Figure 3-6). Since the obstacles in the map may be located in the wrong position, the generated path could lead the vehicle directly into an obstacle it assumes it is avoiding.

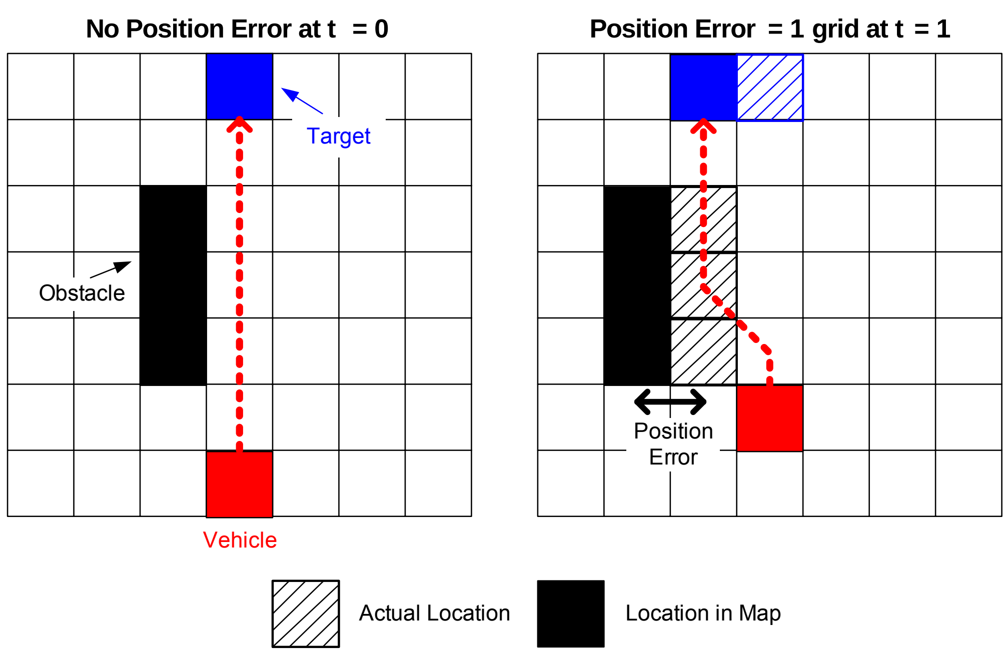


Figure 3-6. Obstacle locations in the map are inaccurate if a position error occurs while populating the map.

No matter how low cost the generated path is, the path is useless if the vehicle is unable to follow it. Map searching algorithms, such as A*, create paths that require discrete heading change. An A* path is ideal for a vehicle that can make discrete heading changes, such as a holonomic, low speed vehicle. An Ackerman steered vehicle, on the other hand, is incapable of turning in place. The change in heading is a function of steering angle and traversed distance. As a result, an Ackerman vehicle incapable of following the generated path may collide with obstacles (Figure 3-7).

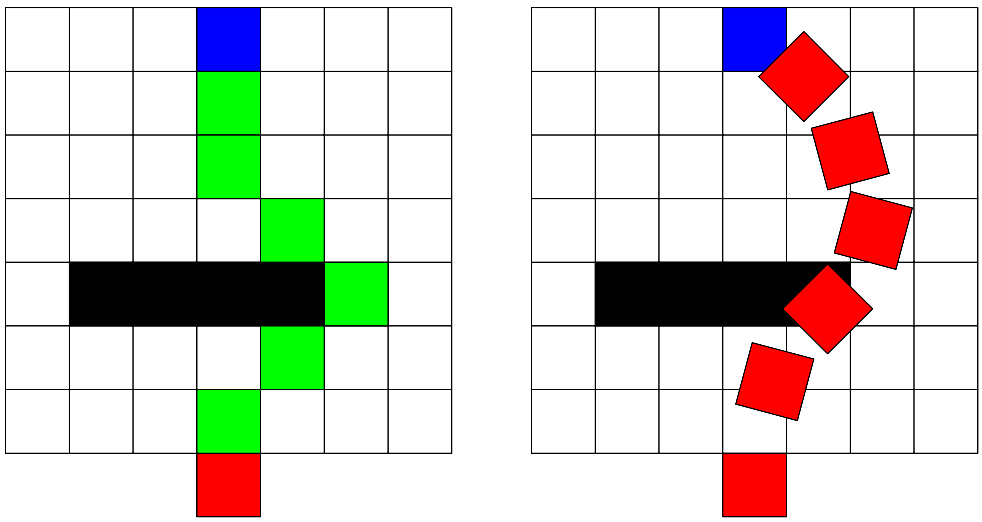


Figure 3-7. If the path is not drivable, the vehicle could hit obstacles.

Drivable trajectories can be created by using such mathematical expressions as Clothoids, Cuboids, and Splines. For example, clothoids paths are continuous curves where the curvature varies linearly with arc length defined by

$$\gamma(s) = ks + \gamma_i \tag{3-1}$$

where γ, γ_i, k, and s are curvature, initial curvature, rate of change of curvature, and distance along the curve [10]. When clothoids are applied to mobile robot trajectories, the curvature is the inverse of turning radius and the rate of change of curvature is controlled by the steering angular rate of change. Various trajectories with different initial conditions and steering rates can be calculated off-line and placed in an ego-graph (Figure 3-8) [11]. This list can include anywhere from a few trajectories to thousands of trajectories. A graph search algorithm can be used to search the ego-graph for an optimal, drivable path.

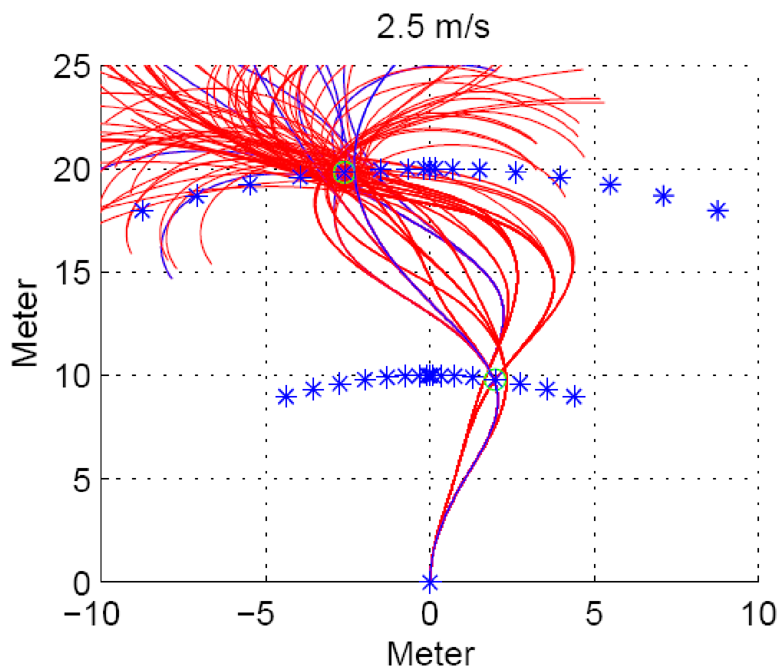


Figure 3-8. An ego-graph of clothoids(© [1998] IEEE) [11].

Even though a drivable path can be generated, the deliberative approach still assumes that the vehicle will track the pre-computed path. On off-road terrain, the vehicle's actual path is affected heavily by the terrain. Not only does the geometry of terrain affect the behavior of the vehicle, but the type of surface and how aggressive the vehicle is driving has a large affect on the overall behavior[8]. Error between the projected path and the actual path can propagate due to terrain effects, slip of the wheels/tracks, and errors in the vehicle model. As a result, deliberative approaches are vulnerable to unpredictable terrain effects.

As you can see, deliberative approaches can become extremely complex. The development time for a deliberative planning can outweigh the advantages. Not only is the development time costly, the software is computationally expensive. High speeds may not be attainable, when the computationally expensive algorithm limits the update frequency of the navigation software.

3.3. Hybrid Reactive-Deliberative Paradigm

Since the reactive and deliberative paradigms each have desirable properties, a hybrid approach can be utilized to maintain the high execution frequency of the reactive paradigm and the path planning of the deliberative paradigm [5]. The hybrid paradigm uses a deliberative path planning algorithm with a reactive path tracking algorithm to perform low level control. The deliberative path planner is no longer required to operate at a high frequency, since a reactive tracking algorithm can follow the commanded path at a higher frequency. Figure 3-9 shows an example of a tracking algorithm called the Pure Pursuit method [12, 13]. The Pure Pursuit commands a steering actuation that will move the vehicle toward an intermediate point on the planned path, which is a *lookahead* distance from the vehicle.

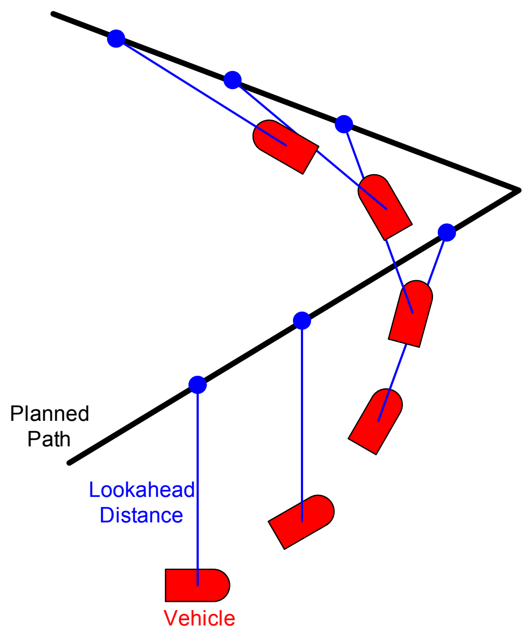


Figure 3-9. Pure Pursuit tracking algorithm.

Even though the limitations with slow deliberative update rates can be lessened, the hybrid reactive-deliberative paradigm still has many similar problems as the deliberative paradigm. The hybrid approach still assumes that the deliberative map accurately represents the environment and the planned path is drivable. Figure 3-10 illustrates that a tracking algorithm can not accurately track a path which is not drivable.

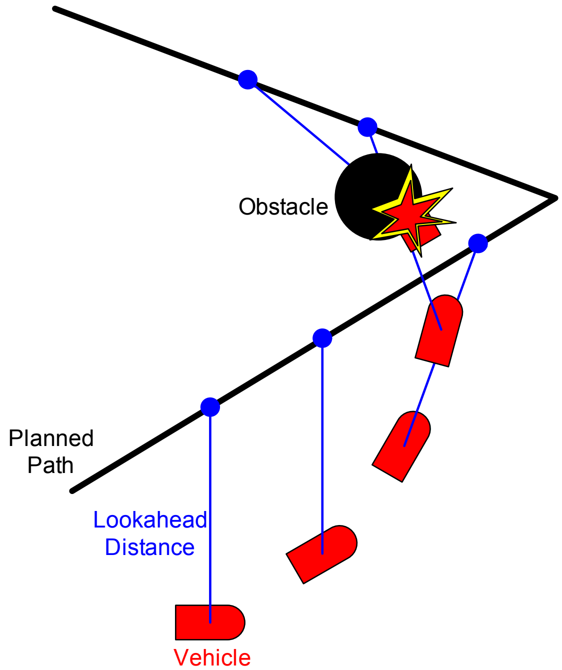


Figure 3-10. A tracking algorithm can not accurately track a path which is not drivable.

Chapter 4
Dynamic Expanding Zones

The reactive paradigm facilitates efficient and simple computation and algorithm development for mobile robot navigation. However, many shortcomings of the reactive approach has led developers to believe it is overly simplistic and insufficient for high speed navigation. Dynamic Expanding Zones has been implemented to overcome the traditional shortcomings of Potential Field obstacle avoidance. The use of dynamically expanding perception zones allows the vehicle to only focus on objects obstructing the path of the vehicle. In this approach, the Potential Field algorithm is used for waypoint navigation, Pure Pursuit is used for road following, and Dynamic Expanding Zones is used for obstacle avoidance. The ability to separate these three tasks from one another comes from the reactive concurrent behavior architecture. This chapter will first give an introduction on Fuzzy Logic, a method to blend the concurrent behaviors. The second section outlines the reactive architecture used for Dynamic Expanding Zones.

4.1. Fuzzy Logic

In Chapter 3, the reactive paradigm section discussed the various methods for concurrent behavior combination. The reactive approach using Dynamic Expanding Zones, discussed in detail in the next four chapters, uses a hybrid Dominance-Fuzzy Control behavior combination. This section will give an overview and the advantages of Fuzzy Logic.

Conventional quantitative approaches are often inappropriate for such humanistic decision making as deciding if shower water is warm enough. The real world is extremely complex and difficult to model with certainty; however, humans are able to make intelligent decisions everyday, often subconsciously. A human does not require accurate data to make intelligent decisions. To adjust the temperature in a shower, a human does not need a thermometer reading; instead, he only needs to know if the water is too cold. Fuzzy logic is able to substitute numerical variables with linguistic variables. For example, a fuzzy variable can describe temperature as *cold, not cold, somewhat cold, very cold, not very cold, very very cold,* or *cold but not very cold* [14, 15].

Binary logic's principle of bivalence requires that a proposition is either true or false (1 or 0). Constraining a proposition to true or false is often unreasonable [16]. In binary logic, a *hot* set can be defined by any temperature above 80 degrees Fahrenheit. As a result, an air temperature of 79.99999999999 degrees Fahrenheit would be considered *not hot*. A human does not require a clear cutoff to determine if the air is hot outside. Fuzzy logic allows this humanistic flexibility by allowing the degree of membership to be anything within the range of 0 and 1. As shown in Figure 4-1, the temperatures from 75 to 85 degrees have a degree of membership between 0 and 1. The difference between 80 and 79.999999999 degree Fahrenheit is now Hot(0.5) and Hot(0.499999999).

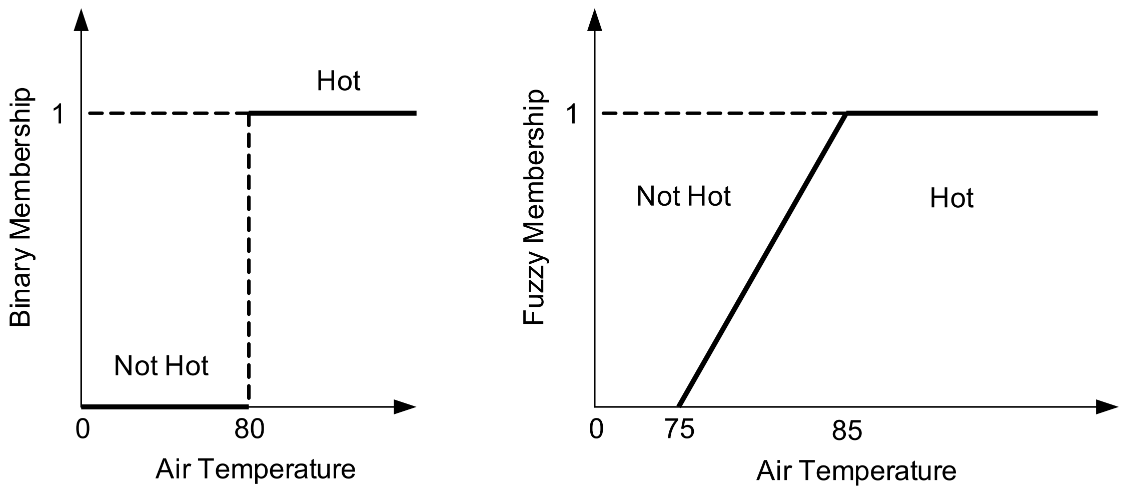


Figure 4-1. Difference between binary membership and fuzzy membership [16].

Fuzzy Logic should not be mistaken as another form of probability theory [16]. The degree of membership is not equivalent to probability. For example, an iron skillet can have 0.95 cool fuzzy membership or a 95% chance of being cool. Would you rather grab an iron skillet that has 5% chance of being scalding hot or the alternative? Fuzzy logic represents uncertainty; however, uncertainty does not mean random uncertainty. Unlike binary logic, fuzzy sets can overlap. Binary logic states that if the air is hot, it cannot be cold at the same time. Figure 4-2 shows the linguistic variables *cold*, *somewhat hot*, and *hot* overlapping. If the temperature is 80 degrees Fahrenheit, the degrees of membership are Hot(0.5) and Somewhat Hot(0.33). All the fuzzy membership functions developed for Dynamic Expanding Zones were created using intuition. Several other methods exist to assign membership functions: inference, rank ordering, angular fuzzy sets, neural nets, genetic algorithm, inductive reasoning, soft partitioning, meta rules, and fuzzy statistics [16].

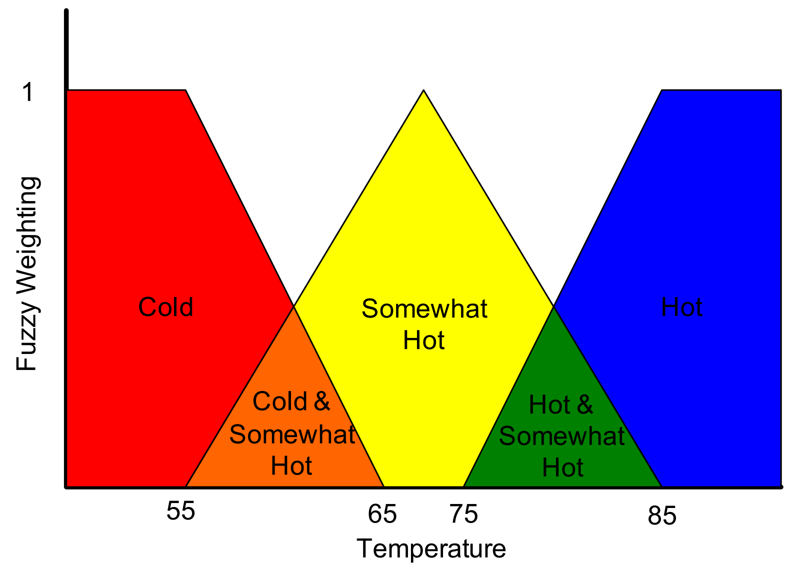


Figure 4-2. Example of overlapping linguistic variables.

Using linguistic variables, fuzzy conditional statements (rule base) can be made to solve ill defined problems. If a road leads directly to the waypoint, the decision to follow the road is easy. However, the further the road strays from the waypoint, the more "fuzzy" the decision becomes. A quantitative road following limit does not exist for these inputs. Fuzzy logic provides a simple robust method for combining these inputs and making a humanistic decision. For example, if the vehicle is heading s*omewhat toward the waypoint,* but *near the course boundary*, the Fuzzy rule base will determine that road following is inappropriate (Figure 4-3A). On the other hand, if the vehicle is heading s*omewhat toward the waypoint,* but *far from the course boundary*, the Fuzzy rule base will determine that road following is appropriate (Figure 4-3B). Figure 4-4 shows a simple example of a Fuzzy rule base for determining if a road or a waypoint should be followed. The actual Fuzzy rule base for road following is shown in Appendix D.

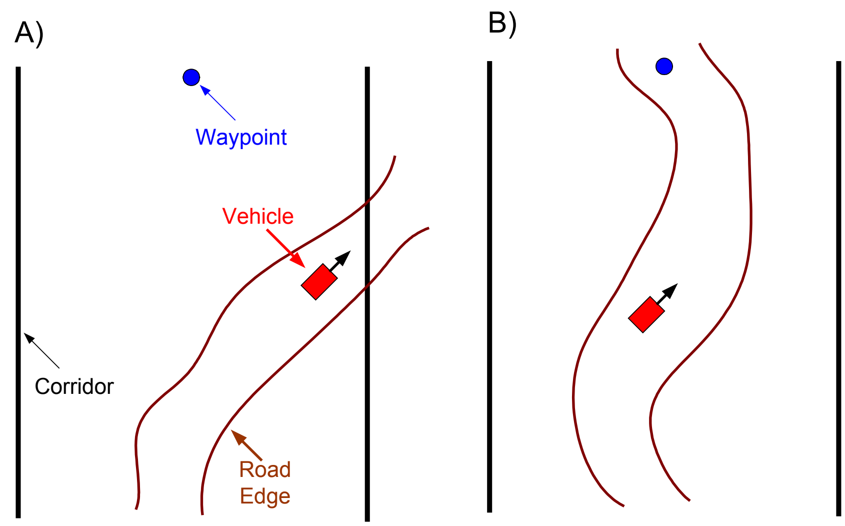


Figure 4-3. An example application of Fuzzy Control. Using the fuzzy rule base in Figure 4-6, the vehicle will choose to follow a waypoint for A) and follow the road for B).

Road Following Fuzzy		Direction to Waypoint: Toward	Direction to Waypoint: Somewhat Toward	Direction to Waypoint: Away
Distance to Course Boundary	Near	Follow Road	Follow Waypoint	Follow Waypoint
Distance to Course Boundary	Moderate	Follow Road	Follow Road	Follow Waypoint
Distance to Course Boundary	Far	Follow Road	Follow Road	Follow Waypoint

Figure 4-4. Example of how a Fuzzy rule base creates a fuzzy output. For didactic purposes, the Road Follow Fuzzy Logic is being shown in a simplified form.

Fuzzy Logic is not a universal solution to all control problems [16]. For systems that have little complexity, mathematical expressions describe the system with the most accuracy. For systems with slightly more complex behavior, learning behaviors, such as neural networks, can be optimal. Fuzzy Logic control is excellent solution for systems which are too complex to model. Fuzzy control makes no attempt to model the system precisely, thus the name *Fuzzy*. It instead makes imprecise but intelligent decisions on complex and/or ambiguous information. Fuzzy logic excels when there is a lack of quantitative information and only qualitative knowledge exists. Fuzzy logic is not a good solution for high precision, high accuracy control such as a hard drive read-write arm control.

Many mobile robot applications do not require high precision and lend themselves well to Fuzzy Logic control. Autonomous navigation is very humanistic in nature. Unnecessary precision can lead to excess cost for development and production. Fuzzy logic has been successfully implemented in many commercial products, especially in Japan. Everything from a rice cooker to a subway train system have been controlled by Fuzzy Logic [16].

4.2. Reactive Navigation Using Dynamic Expanding Zones

Reactive navigation paradigms have been used successfully in a wide variety of autonomous ground vehicles. Nevertheless, these reactive approaches are viewed by many researchers as overly simplistic and unable to deal with complex real-world environments. The DEZ algorithm was developed to evaluate the effectiveness of a reactive approach for high speed navigation through unstructured terrain.

Similar to other reactive algorithms, DEZ uses a set of concurrent behaviors which make steering and propulsion decisions based on the sensor data. The reactive strategy in this paper includes four main behaviors: waypoint navigation, road following, obstacle avoidance, and rollover prevention. Each behavior was developed independently of one another. This flexibility and ease in development is not otherwise achievable using a deliberative approach.

Each behavior has a priority, where the higher priority behaviors subsume the lower priority behaviors. The levels of priority for the behaviors are listed from lowest to highest: waypoint navigation, road following, obstacle avoidance, and rollover prevention. As a result, if a road exists, the vehicle will follow a road instead of the waypoint. However, following roads just because it exists can lead the vehicle to navigate away from the waypoint. Fuzzy Logic is implemented to intelligently blend road following behavior with the waypoint navigation behavior (Figure 4-5). Similarly fuzzy logic is also implemented to blend obstacle avoidance with waypoint navigation. The fuzzy logic control determines if following a road is "advantageous" and which direction to avoid an obstacle is "better." As a result, the DEZ algorithm uses a Hybrid Dominance-Fuzzy Control behavior blending scheme. The following chapters will go into more detail about each behavior.

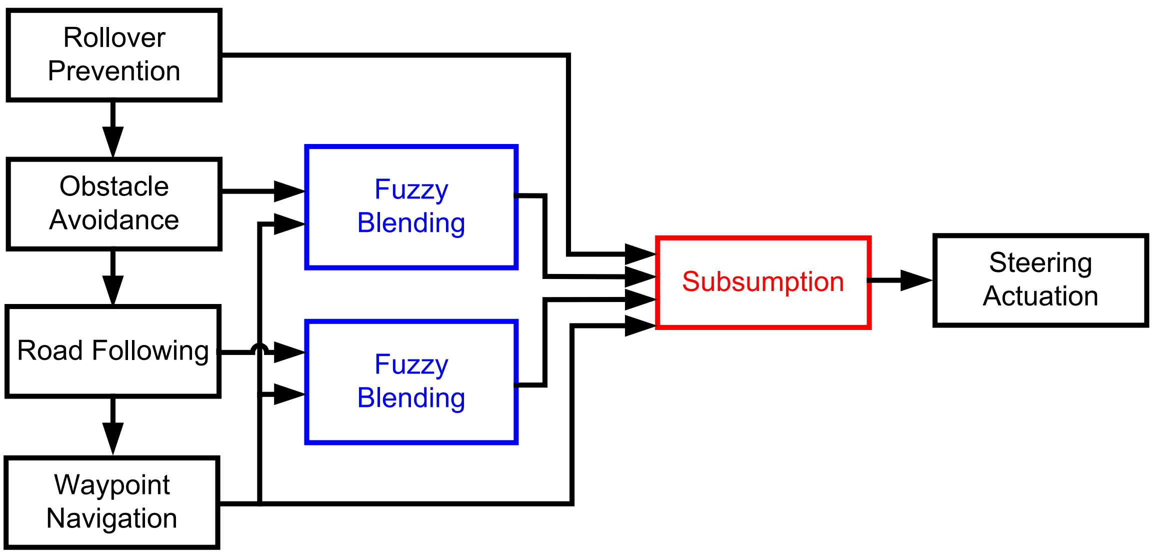


Figure 4-5. Dynamic Expanding Zones reactive architecture

Chapter 5
Waypoint Navigation

A critical component of the Grand Challenge objective is the successful navigation of globally defined waypoints. Waypoint navigation is the lowest level behavior in the reactive framework (Figure 5-1). When no "advantageous" roads and obstacles exist in front of the vehicle, the waypoint navigation behavior will be executed. The reactive navigation software does not generate a planned path in order to reach a desired waypoint. Before going into detail about the waypoint navigation algorithm, this chapter will first define the vehicle model. The following section will discuss how the Potential Fields algorithm was implemented for waypoint navigation.

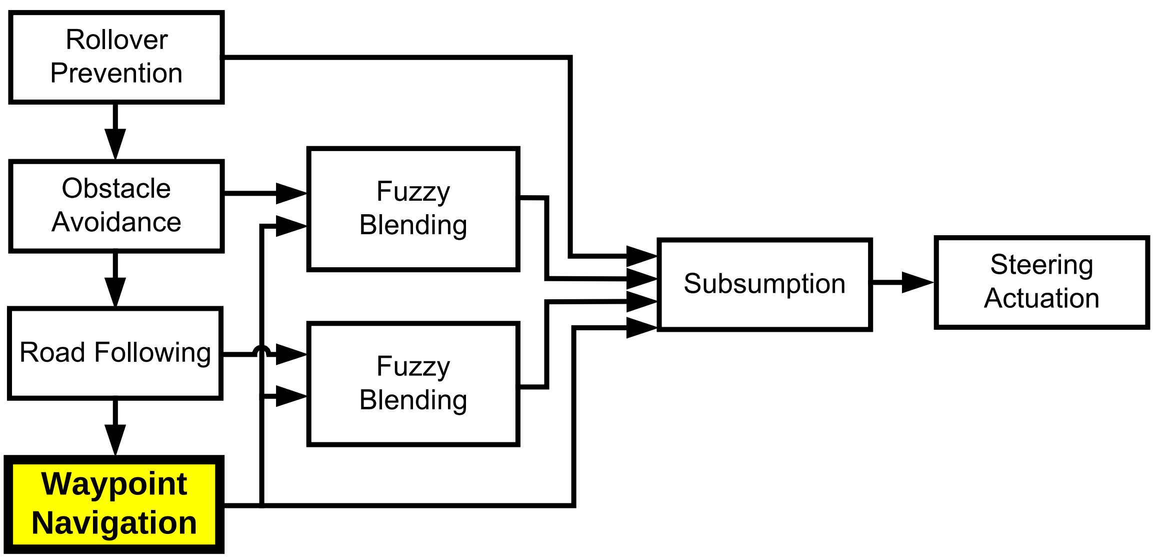


Figure 5-1. Waypoint navigation behavior within the Dynamic Expanding Zones reactive architecture

5.1 Vehicle Model

For reasons stated in Chapter 2, an Ackerman steered vehicle was selected as the mobile robot platform. Like automobiles, an Ackerman steered vehicle has four points of contact through the wheels. The two front wheels are steered, while the two rear wheels are fixed. A true Ackerman steered vehicle has all the wheels perpendicular to the instantaneous center of rotation (ICR). As a result, the inside wheel of a turn has a larger steering angle than the outside wheel. As shown in Figure 5-2, an Ackerman steered vehicle can be modeled as a bicycle [10]. To use the bicycle model, the vehicle is assumed to be a true Ackerman vehicle. In reality, the steering rack and pinion system only approximates a true Ackerman geometry. However, the minimal error between the two is negligible for a reactive paradigm.

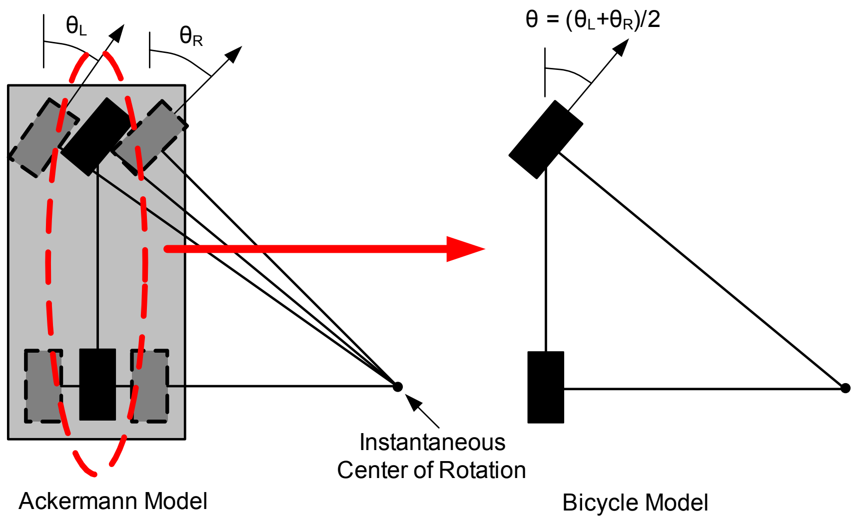


Figure 5-2. Ackerman steered vehicle can be approximated by the bicycle model. θ_L and θ_R are the steering angle of the left and right wheels.

A bicycle is a two degree of freedom system, where the user controls the steering and propulsion. The relationship between steering column position (front wheel angle) and the turning radius for the bicycle model is

$$\tan(\theta) = L/r \tag{5-1}$$

where θ, L, and r are steering angle, wheel base, and turning radius, respectively (Figure 5-3). For this thesis, a steering angle to the right of the longitudinal axis is positive and to the left is negative. The longitudinal axis of the vehicle is equivalent to the heading of the vehicle.

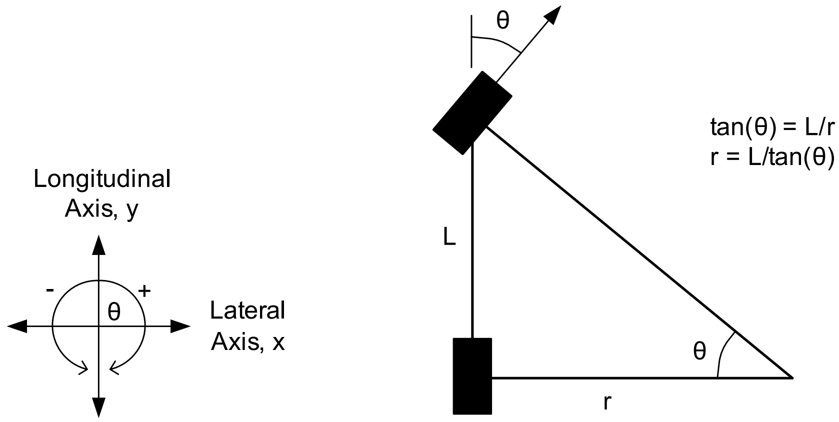


Figure 5-3. Geometry of the bicycle model. The variables θ, r, and L are steering angle, turning radius, and wheel base, respectively.

An Ackerman steered vehicle is nonholonomic, incapable of a zero radius turn. Since the vehicle is nonholonomic, a change in steering angle does not correspond to a change in vehicle heading (Figure 5-4). The relationship between heading and steering angle is

$$\Delta\varphi_v = \frac{L}{\Delta s \tan(\theta)} \tag{5-2}$$

where Φ_v, L, s, and θ are vehicle heading, wheel base, vehicle displacement, and steering angle [17]. The effects of this relationship on waypoint navigation will be discussed further in the last section of this chapter.

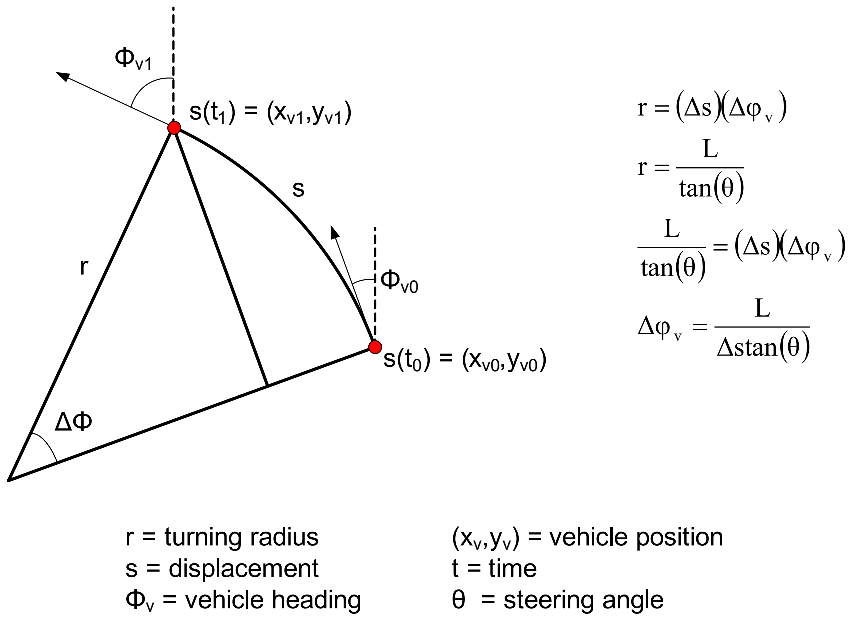


Figure 5-4. The heading of a non-holonomic Ackerman vehicle is a function of steering angle and vehicle displacement [17].

5.2 Potential Field Algorithm

The mission requirements of the robot greatly affect which reactive strategy to use. If the robot needs to follow the track line connecting waypoints, a path tracking algorithm, such as Pure Pursuit, is ideal (Figure 5-5). If the robot is not required to follow the track line, a point-to-point algorithm is simpler and more ideal than a tracking algorithm to implement (Figure 5-6). Since the DARPA rules make no requirement to follow the track line, the Potential Field algorithm was implemented for waypoint navigation. Navigating independently of the track line is extremely important when the lateral boundaries are miles wide. As Figure 5-7 shows, the overall distance traveled by a Pure Pursuit controlled vehicle can be much larger than a Potential Fields algorithm, if the LBO is large.

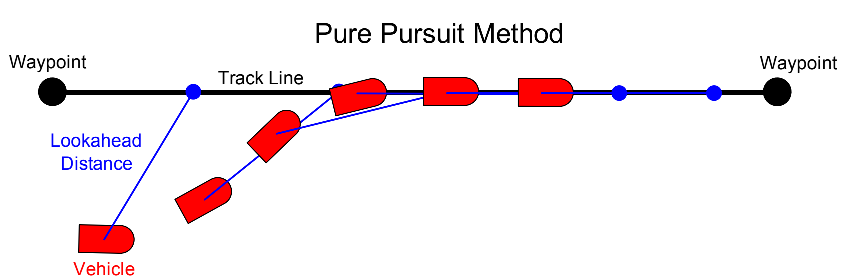


Figure 5-5. Track line waypoint navigation.

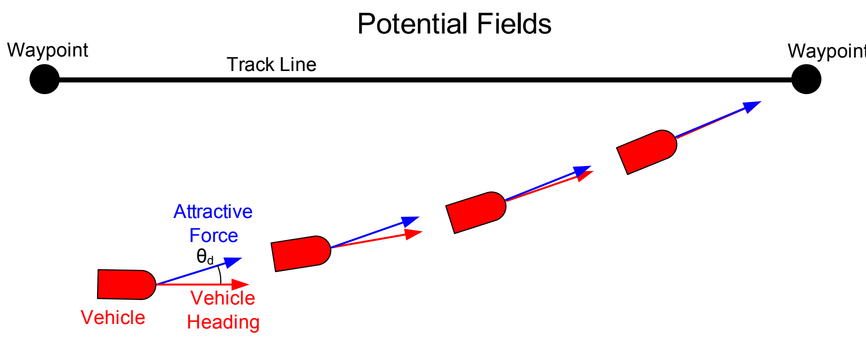


Figure 5-6. Point-to-point waypoint navigation.

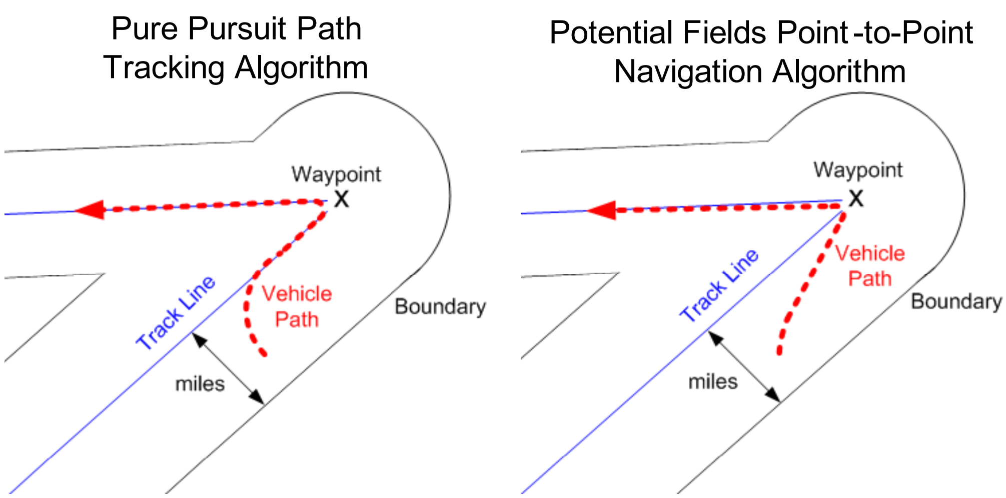


Figure 5-7. When the LBO is extremely wide, the Potential Field algorithm overall path is more efficient.

The Potential Fields algorithm utilizes attractive and repelling fields to control the mobile robot. Attractive potential fields are typically used for waypoints (Figure 5-8), while repelling forces are typically used for obstacles. Attractive potential fields for waypoints are similar to a gravitational force field around a mass, such as a planet. A secondary smaller mass, such as a meteor, will be attracted to the larger mass. Similarly, the vehicle is attracted toward the waypoint. Instead of following a track line, the vehicle tends to aim straight for the waypoint. Unlike gravitation acceleration, the magnitude of the attractive potential field is not limited to an inverse square relationship. The magnitude of the potential field is a constant of one regardless of the distance from the waypoint.

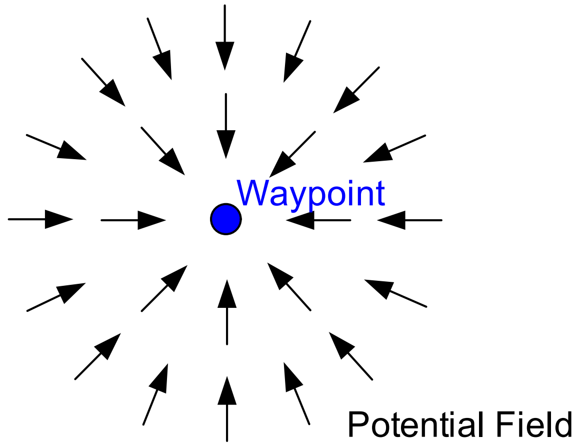


Figure 5-8. Potential field around a waypoint.

Since a physical force field is not pushing the robot, the attractive force must be converted to steering and propulsion commands. The steering is calculated by first determining the heading error between the heading of the vehicle and the attractive force (Figure 5-9). The vehicle is then commanded to steer this calculated heading error. If the heading is -30 degrees, the commanded steering is 30 degrees to the left. If the heading error is larger than the max steering angle, the commanded steering value is capped at the max steering angle.

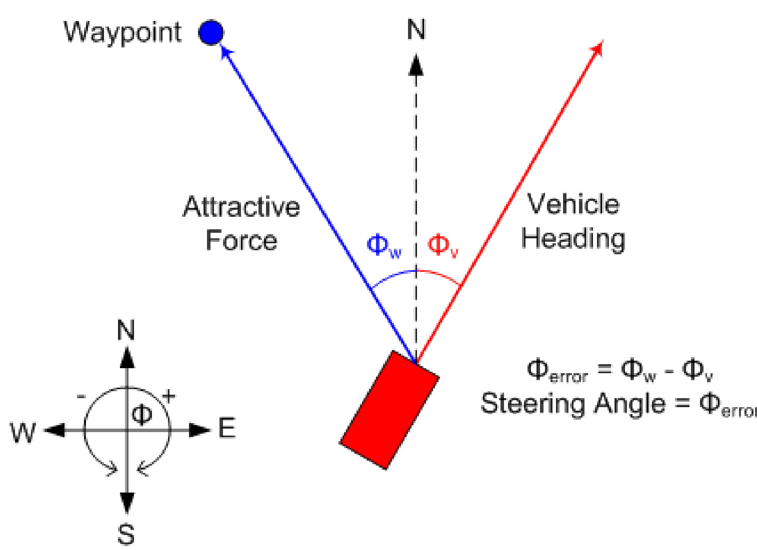


Figure 5-9. Method for calculating the steering angle

5.3 Proportional Closed Loop Feedback Controller

The Potential Fields algorithm acts as a proportional closed loop feedback control with a gain, K_p, of 1 (Figure 5-10). Since the algorithm is running on a digital computer, the controller is discretely executed with a period of T. As a result, the steering angle is commanded every T seconds, similar to a zero order hold (ZOH). Discrete control can be assumed continuous (analog) control, if the sampling frequency is large and the feedback resolution is small. According to the *Digital Control of Dynamic Systems* text, the continuous assumption can be applied, if "the sampling frequency is 30 or more times the systems bandwidth with a 16-bit word size" [18]. The Potential Fields algorithm was run at frequencies as low as 10 Hz at top vehicle speeds of 25mph without any discrete control effects. Chapter 9 discusses the problems which can occur if this algorithm is run at a slow frequency.

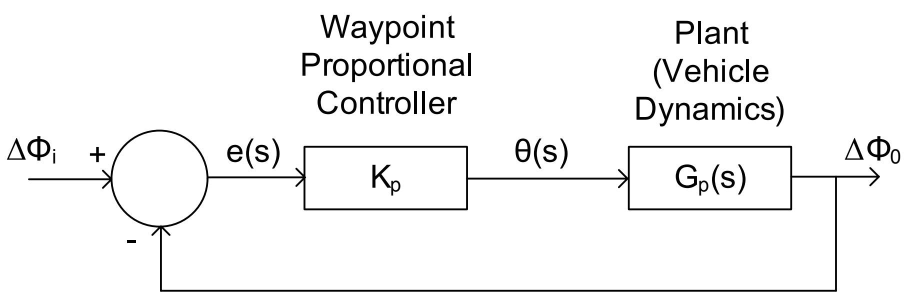


Figure 5-10. Control diagram for the proportional waypoint controller.

As the vehicle progresses toward the waypoint, the heading error approaches zero (Figure 5-11). As long as the waypoint navigator acts as a continuous controller the resulting overall behavior is smooth and requires very little computation.

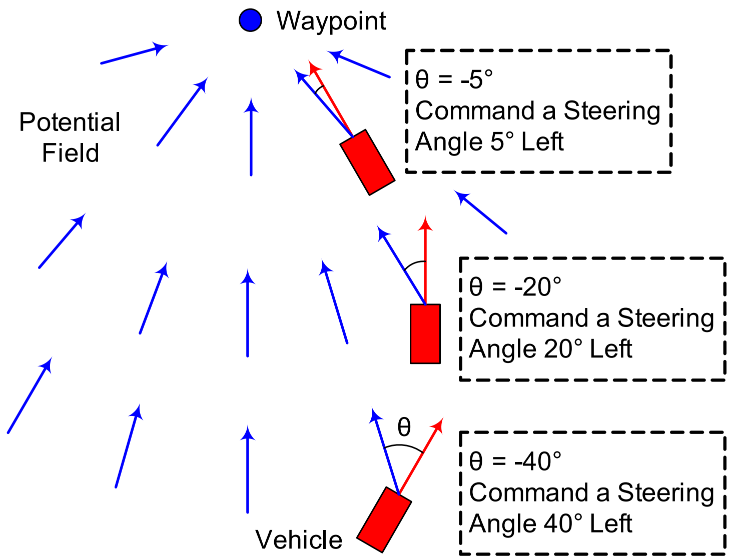


Figure 5-11. Heading error converges to zero.

The Potential Field algorithm also tends to go unstable as the vehicle's distance to the waypoint approaches zero. As the vehicle gets closer to the waypoint, the angular rate of change of the attractive force becomes larger. Since DARPA does not require the vehicle to drive over the waypoints, this problem can be prevented by expanding the size of the waypoint from a point to a circular area. The radius of the waypoint is defined by the distance between waypoint and the corridor intersection point (Figure 5-12). Once the vehicle enters this waypoint radius, the vehicle will move onto the next waypoint. The implementation of the waypoint radius also eliminates problems with navigating to a waypoint where an obstacle exists.

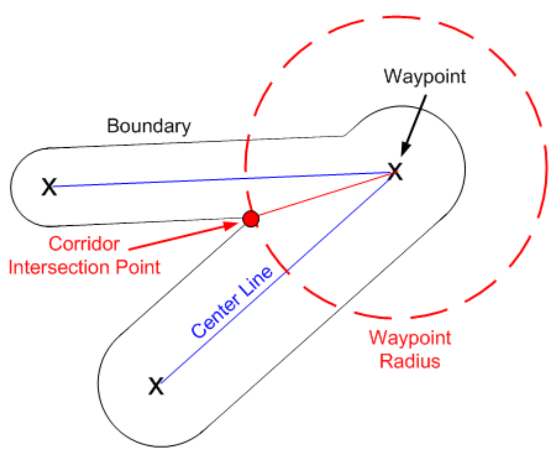


Figure 5-12. A waypoint radius is created using the Lateral Boundary Offset (LBO) of the intersecting corridors.

Chapter 6
Road Following

Since roads are generally easier to traverse and have fewer obstacles than unstructured desert terrain, road following was critical for the DARPA Grand Challenge. As long as no obstacles or roads are perceived on the path to the waypoint, the vehicle will progress using simple waypoint following behavior described in the previous section. However, if a road exists that leads the vehicle in the general direction of the waypoint, the road following behavior will subsume the waypoint navigation behavior (Figure 6-1). The following sections discuss how the Pure Pursuit algorithm was implemented for road following and how Fuzzy Control determined if road following will lead the vehicle in the general direction of the waypoint.

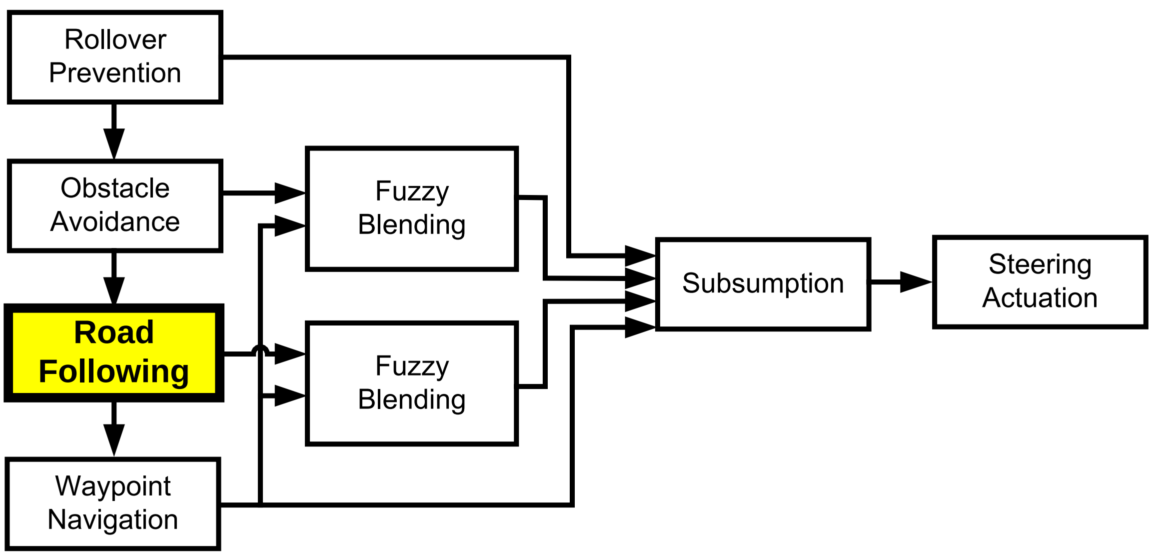


Figure 6-1. Road following behavior within the Dynamic Expanding Zones reactive architecture

6.1. Hybrid Pure Pursuit-Potential Fields Algorithm

Unlike waypoint navigation, the objective of road following is not to aim toward one static point, but reactively track a path. Road following is not to be confused with deliberative navigation. No path is being generated; instead, road following is tracking a path (the road), which already physically exists. The road recognition software mentioned in Chapter 2 outputs road data as an array of perceived road center points, instead of a continuous centerline. The center points are reported in 0.25 m increments.

The area examined for a road is a 12.5m x 12.5m area in front of the vehicle. The array of road center points is updated at 4Hz. Road following uses a hybrid Pure Pursuit-Potential Fields method for tracking these road center points (Figure 6-2).

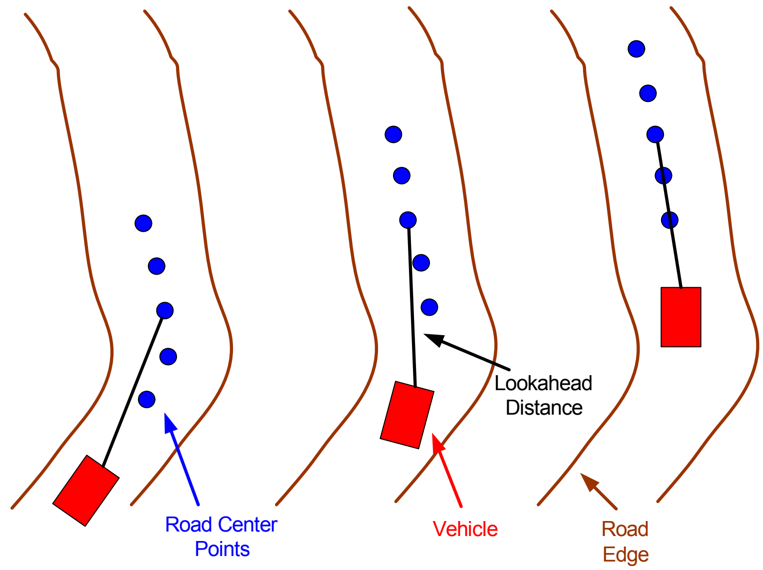


Figure 6-2. Road following using hybrid Pure Pursuit-Potential Fields method

Typically, the Pure Pursuit method assigns a lookahead distance, which defines a goal point to track on the path. However, the hybrid approach used for DEZ is the opposite; the goal road point defines the lookahead distance. The hybrid lookahead distance is the distance from the vehicle to the closest acceptable road center point (Figure 6-3). All road points outside the LBO are considered unacceptable and removed to prevent the vehicle from following a road off the course. Points within 3m of the vehicle and 40 degrees from the heading of the vehicle are also eliminated to prevent aggressive maneuvers. After the unacceptable road points are removed, the closest valid road point is picked to track.

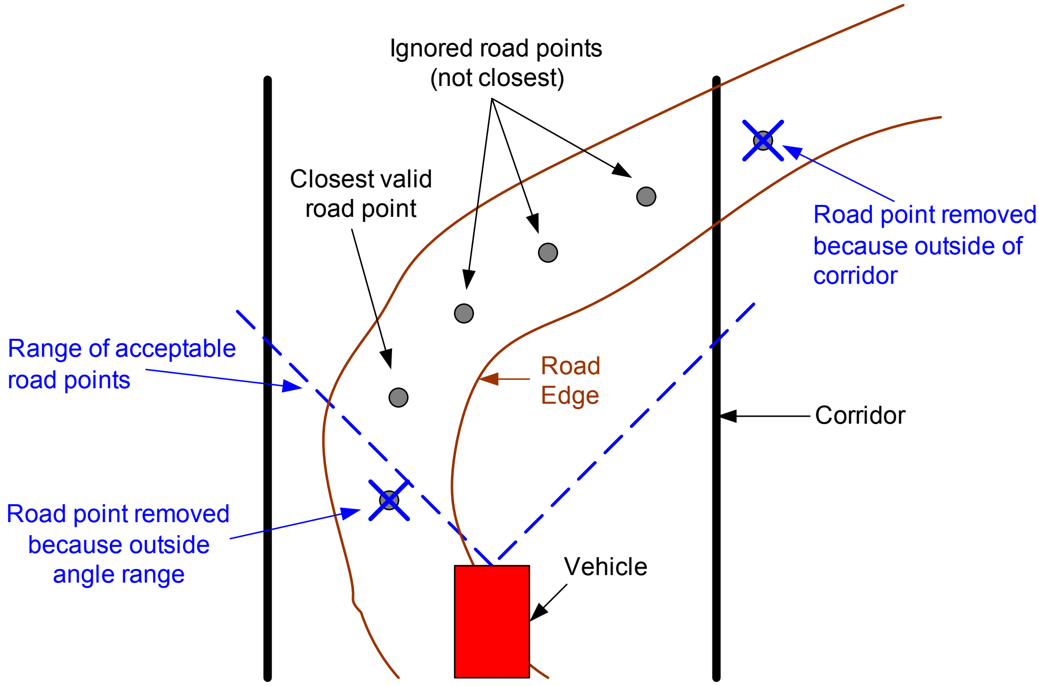


Figure 6-3. An illustration of how a road point is selected from the road point array.

Once the closest valid road center point is identified, the algorithm is identical to waypoint navigation. The road center point acts as a waypoint with an attractive force. The steering is calculated by commanding a steering angle equal to the heading error. The main difference between waypoint navigation and road following is that a waypoint is static and a road point is dynamic, similar to a carrot hanging in front of a donkey.

6.2. Fuzzy Control for Road Following

As discussed in Chapter 4, concurrent behaviors are both combined and subsumed. Waypoint navigation behavior is subsumed by the higher priority road following behavior. However, following a road away from the waypoint is more detrimental than ignoring road following completely. For this reason, Fuzzy Logic was implemented to add more intelligence in determining if following road is advantageous (Figure 6-4). Conventional quantitative approaches are often inappropriate for such humanistic decision making as deciding if following a road is advantageous.

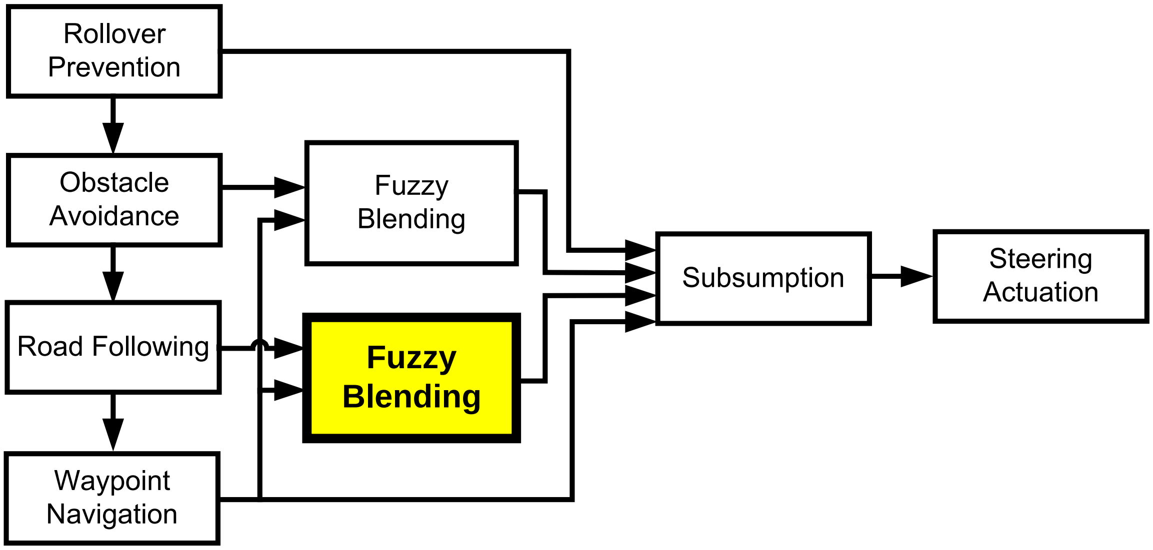


Figure 6-4. Road Following Fuzzy Control within the Dynamic Expanding Zones reactive architecture

The three fuzzy variables used in the road following fuzzy control are heading error, distance to the boundary, and direction to the boundary (Figure 6-5). Road point heading error is the angle between the vehicle's heading and the attractive force to the road point. Heading error is linguistically described as small, medium, and large. Distance to the boundary is measured by percent, where 0% is the LBO and 100% is the center line. The distance to the boundary is described as near, moderate, and far. The third variable is direction to the boundary. This variable is described by a Boolean away or toward the boundary. If the vehicle is pointed at the center line, the vehicle is traveling away from the boundary; otherwise, the vehicle is traveling toward the boundary. The fuzzy membership and fuzzy rule base is shown in Appendix C.

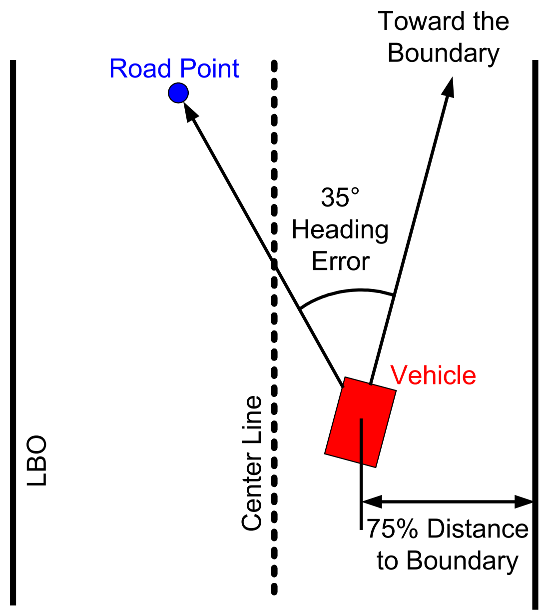


Figure 6-5. An illustration of the three fuzzy variables used for road following

Chapter 7
Obstacle Avoidance

For low speeds, the Potential Field algorithm has been effectively implemented for obstacle avoidance. However, as the vehicle speeds increase the magnitude profiles of the obstacles must also expand to allow for the vehicle to have enough reaction time. Implementing dynamic magnitude profiles for high speed navigation is computationally and developmentally complex. Creating a smooth overall emergent path is more of an art than a science. Not only are these dynamic profiles difficult to develop, but they are vehicle specific. Dynamic Expanding Zones (DEZ) has been developed as an alternative high speed obstacle avoidance algorithm. A set of perception zones focus on objects of immediate interest while ignoring objects that are unlikely to enter the path of the vehicle. If an obstacle is located in these zones, obstacle avoidance behavior will subsume waypoint navigation and road following as shown in Figure 7-1.

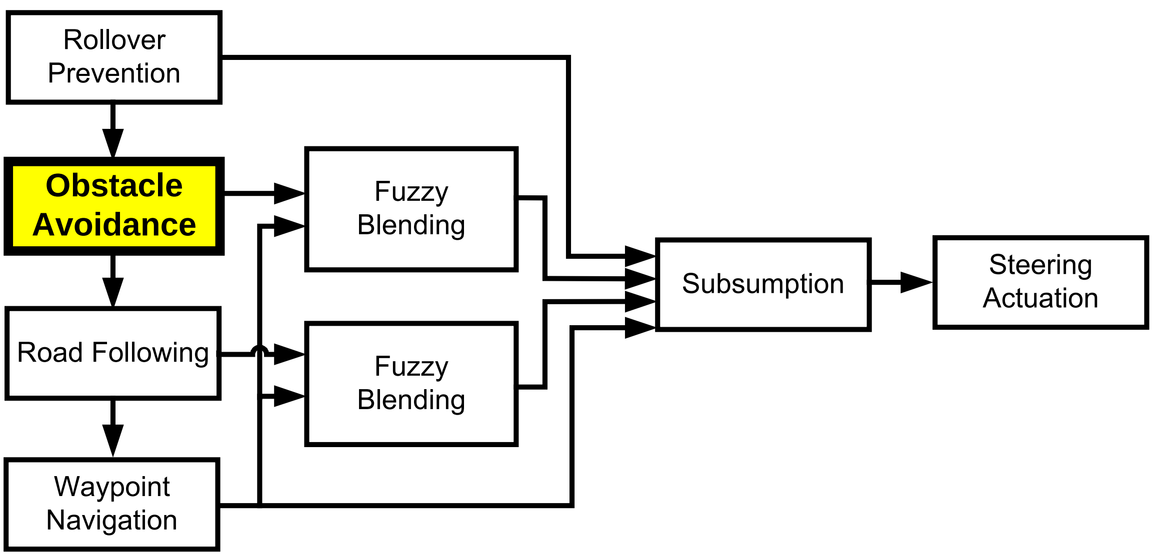


Figure 7-1. Obstacle avoidance behavior within the Dynamic Expanding Zones reactive architecture

Dynamic Expanding Zones takes advantage of the tight Sense-Act couple of the reactive paradigm. The algorithm only requires an egocentric sensor to create an instantaneous obstacle map with boolean elements (obstacle or no obstacle). For the Grand Challenge, Virginia Tech used a SICK LADAR to obtain a polar map of the obstacles. The LADAR data is not transformed to the world the frame nor is it stored over time. If any spurious data is detected, the data is forgotten in the next iteration. Dynamic Expanding Zones can use a deliberative map, but this would only negate the benefits of the reactive paradigm and make the software more computationally expensive. This chapter will define the perception zones, steering angle calculations, and dynamic behavior of the zones for the Dynamic Expanding Zones algorithm.

7.1 Perception Zones

The Dynamic Expanding Zones algorithm uses two zones to determine the avoidance behavior when an obstacle is present (Figure 7-2). The first zone, the Avoidance Zone, is located directly in front of the vehicle was originally adapted from Reynold's "Steering Behaviors for Autonomous Characters" [19]. If an obstacle is detected in the rectangular Avoidance Zone, the vehicle must avoid it to continue toward the waypoint safely. The width of the Avoidance Zone is slightly larger than the width of the vehicle to maintain a safe lateral distance away from obstacles. The Avoidance Zone width remains constant; however, the length expands dynamically, hence the name Dynamic Expansion Zones. DEZ commands a steering angle to avoid any obstacles in this zone.

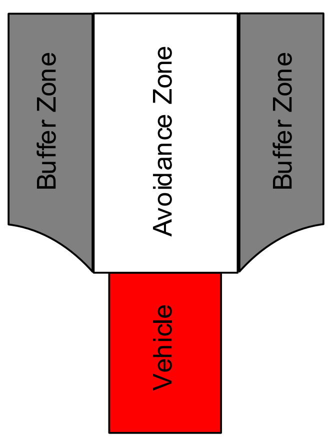


Figure 7-2. Dynamic Expanding Zones layout.

The Avoidance Zone prevents the vehicle from needlessly avoiding objects safely outside the path of the vehicle. The classical Potential Field algorithm, on the other hand, does not attempt to determine if an object is actually obstructing the path of the vehicle. The object repels the vehicle if it is within close proximity, even if the vehicle is in no danger of colliding with it (Figure 7-3). The ability to determine if an obstacle is in the path is critical for high speed, efficient obstacle avoidance.

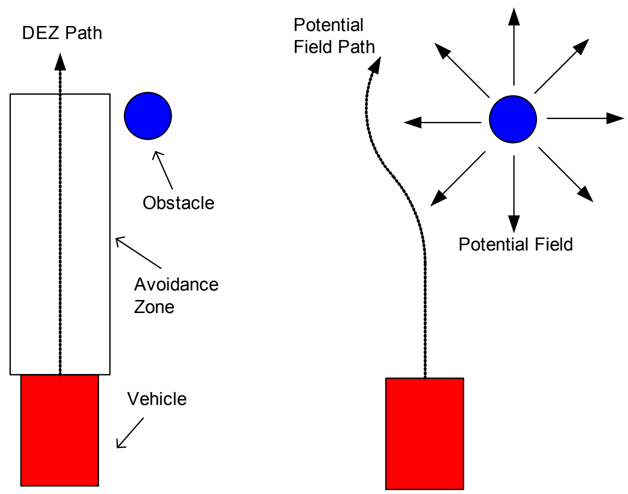


Figure 7-3. The Avoidance Zone prevents the vehicle from needlessly avoiding obstacles.

The second zone, the Buffer Zone, is adjacent to the Avoidance Zone. If the vehicle attempts to make a turn when there is an obstacle in the buffer, Dynamic Expanding Zones will override the turn command. The vehicle will drive straight forward, until the obstacle exits the Buffer Zone. Once all the zones are clear, waypoint/road following will resume.

7.2 Avoidance Zone Steering Direction

If only one obstacle is located on the right side of the Avoidance Zone, the obvious avoidance maneuver would be to steer left around it. However, when several obstacles are within the Avoidance Zone, the decision becomes more complex. No matter how many obstacles are in the Avoidance Zone, the steering direction decision is made by only focusing on the obstacles within the Obstacle Window (Figure 7-4). The Obstacle Window examines an area with a length of 1m starting from the closest obstacle and a width twice the Avoidance Zone width.

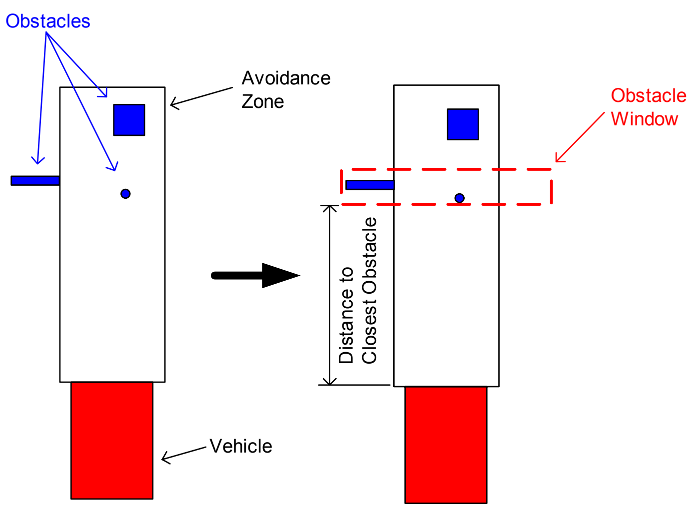


Figure 7-4. Illustration of obstacle window

The steering direction is determined by a method referred to as Obstacle Summing. The longitudinal distances of all the obstacles in the Obstacle Window are summed (Figure 7-5). The distances are measured from the centerline, where all values left of the centerline are negative and all values on the right are positive. If the resulting value is negative, the vehicle will steer right to avoid the obstacle(s). By summing the values, larger obstacles are given a higher weighting. Obstacle Summing does not reveal a path through a cluster of obstacles. If obstacles exist in the obstacle window, the vehicle can not physically navigate between the obstacles without a collision.

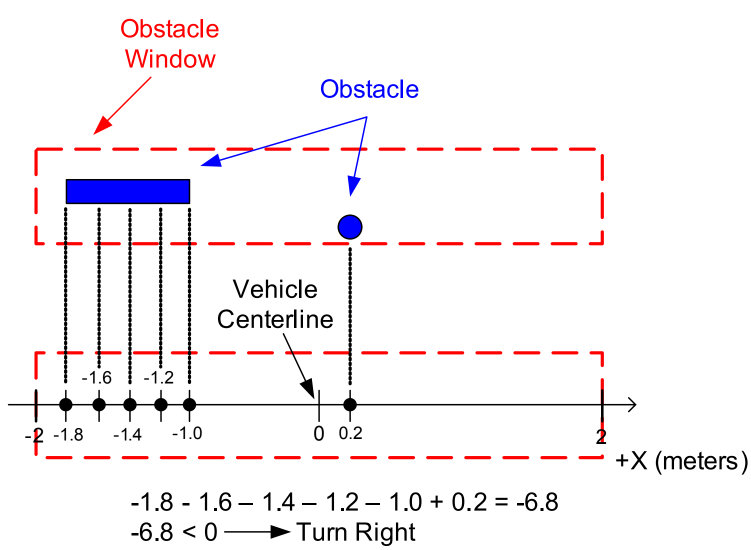


Figure 7-5. Illustration of the Obstacle Summing method

The Obstacle Summing method can lead to jerky steering decisions, if the summed value oscillates between positive and negative. This indecisive behavior is extremely undesirable, largely because the reaction time to avoid the obstacle is reduced and may result in a collision. To prevent indecision, a software version of a Schmitt trigger was implemented. A Schmitt Trigger is a special comparator circuit, which converts a continuous input signal to a digital output signal with hysteresis (Figure 7-6). When the Obstacle Summing value is below the "low threshold", the output is "turn right." When the input is above the "high threshold", the output is "turn left." When the input is between the "low threshold" and "high threshold," the output remains constant. The Schmitt Trigger uses hysteresis to reduce the effects of noise on the stability of the trigger.

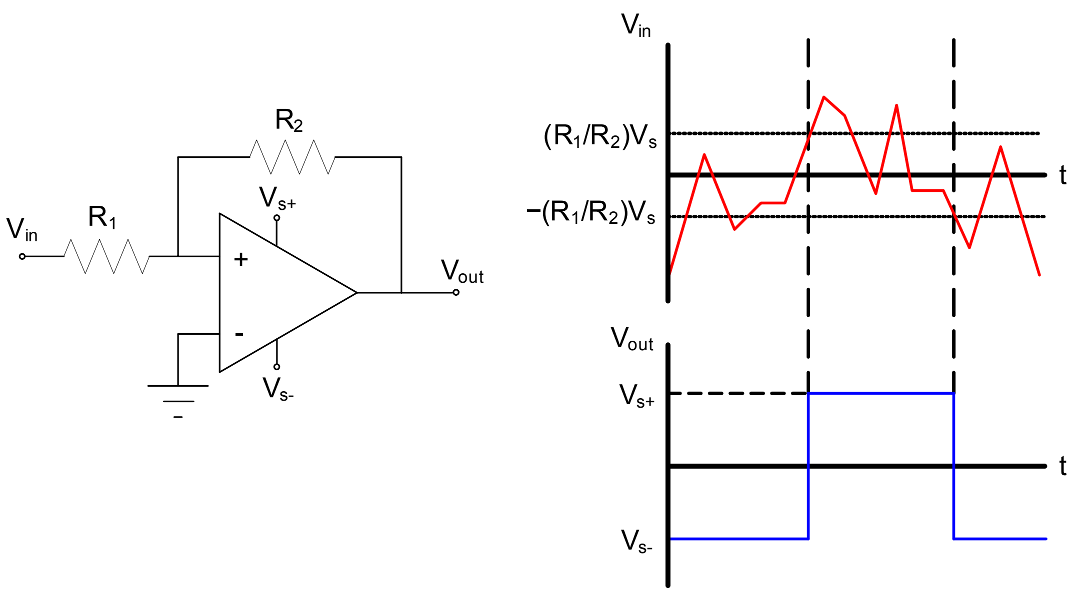


Figure 7-6. Schmitt Trigger.

7.3 Avoidance Zone Fuzzy Control

If an obstacle is present in the Avoidance Zone, obstacle avoidance subsumes both waypoint navigation and road following. However, pure Subsumption can lead to poor navigation decisions. Figure 7-7 shows how an efficient avoidance maneuver results in a longer overall path toward the waypoint. Fuzzy Logic is implemented to blend waypoint navigation and obstacle avoidance (Figure 7-8).

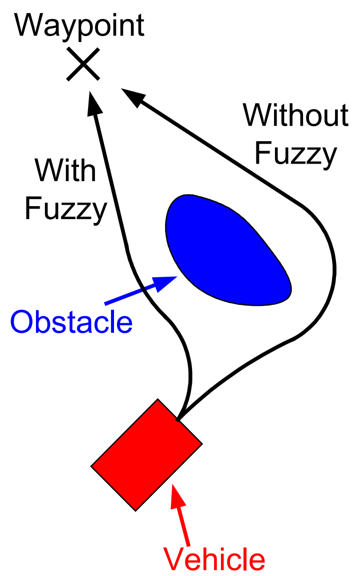


Figure 7-7. Fuzzy control is needed to make intelligent avoidance decisions. Path A uses fuzzy blending, while Path B uses pure Subsumption.

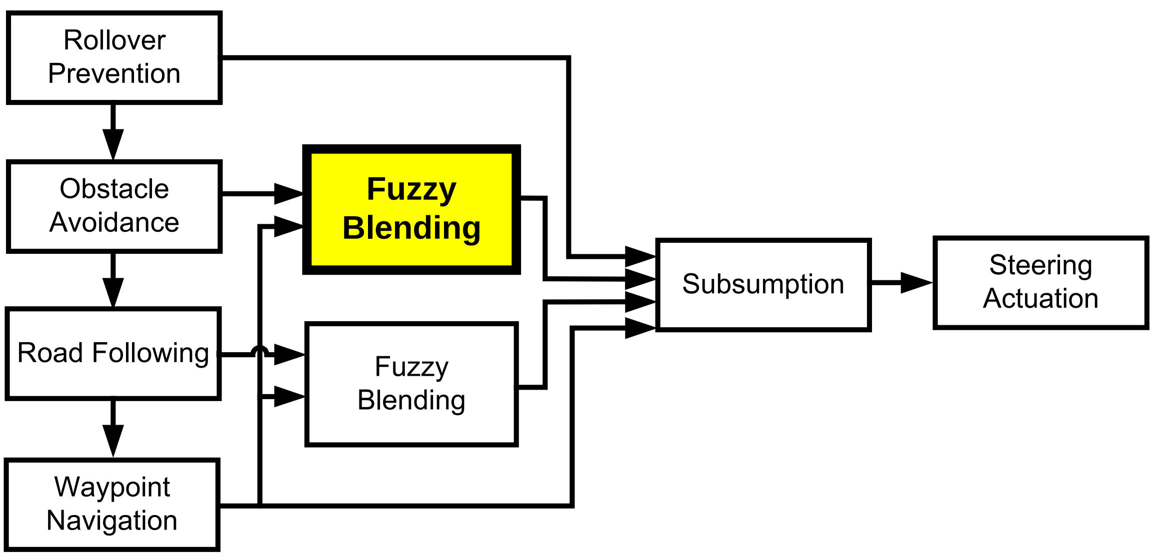


Figure 7-8 .Road Following Fuzzy Control within the Dynamic Expanding Zones reactive architecture

The steering direction (left/right) is determined by fuzzy logic control. The controller intelligently decides a steering direction which is optimal for not only avoiding an obstacle but steering toward the waypoint. When an obstacle is *close* to the vehicle, the vehicle may only be able to avoid the obstacle in one direction. Steering in the wrong direction could cause a collision. As a result, DEZ will weight Obstacle Summing heavier for *close* obstacles. On the other hand, when the obstacle is *far* ahead of the vehicle, the vehicle is capable of avoiding the obstacle on the left or right side. In this case, steering in the direction of the waypoint will reduce the overall path length.

The fuzzy variables used to control obstacle avoidance steering direction are Obstacle Summing, Distance to Closest Obstacle, Distance from Centerline, and Angle to Boundary. The Distance from Centerline is measured in percent, where the centerline, right boundary, and left boundary are 0%, 100%, and -100%, respectively. The Angle to Boundary is the angle between the vehicle heading and the vector perpendicular to the centerline. The angle is negative when the vehicle is pointing toward the left boundary and positive when pointing to the right boundary. An illustration of Angle to Boundary and Distance from Centerline is shown in Figure 7-9. The fuzzy membership and fuzzy rule base is shown in Appendix D.

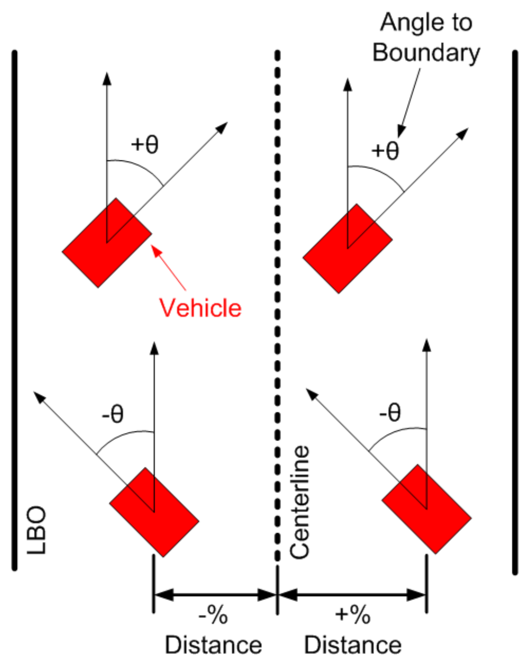


Figure 7-9. Illustration of Distance from Centerline and Angle to Boundary.

7.4 Avoidance Zone Steering Magnitude

The magnitude of the commanded steering angle is calculated using the distance to the closest obstacle within the Avoidance Zone. First, the radius of an arc that connects the wheel of greater turning radius to the obstacle is determined (Figure 7-10). This arc trajectory is actually the trajectory of the back left tire shifted forward by the length of the wheel base. Note that the magnitude of the resulting turning radius is the same for left or right turns. Using this steering radius, the magnitude of the steering angle is calculated by using the bicycle model.

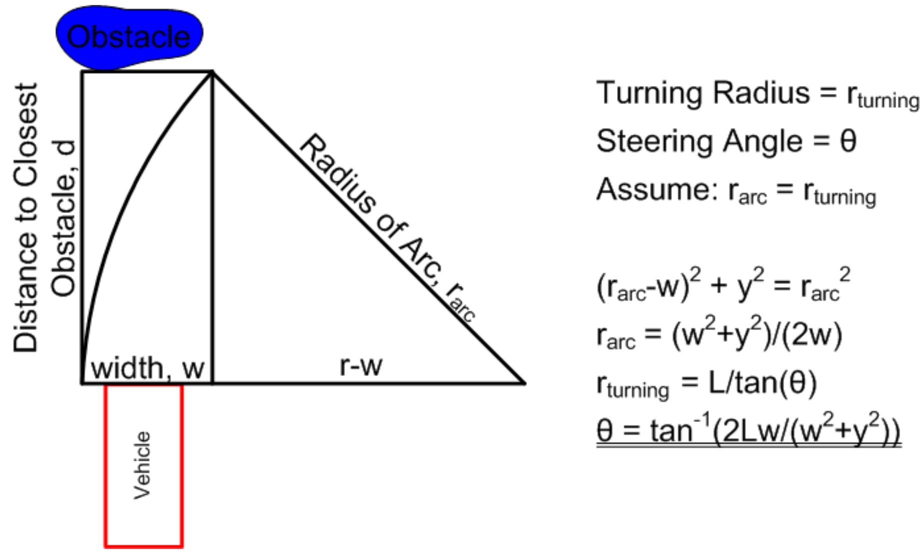


Figure 7-10. Steering angle calculation using arc assumption.

The steering angle calculation is not an attempt to model the actual projected path of the vehicle. An approximate path is all that is required for this reactive approach. If the commanded steering angle is not sufficiently large enough to avoid an obstacle, the next iteration the commanded steering angle will increase. If the obstacle is not avoided, the steering angle will increase each iteration as the distance to the obstacle decreases. Because of closed loop feedback nature, Dynamic Expanding Zones does not need to compute computationally complex paths often required for deliberative navigation algorithms. As a result, DEZ requires minimal processing power when compared to many deliberative approaches.

7.5 Buffer Zone

The Buffer Zone, adjacent to the Avoidance Zone, prevents the vehicle from turning into an obstacle outside of the Avoidance Zone. The Buffer Zone is only used if the vehicle is attempting to make a turn; otherwise, only the Avoidance Zone is utilized. In addition, the left Buffer Zone is used for left turns and the right Buffer Zone for right turns. If an obstacle is within the Buffer Zone, the vehicle will drive straight forward until the obstacle exits the Buffer Zone. Once all the zones are clear, waypoint/road following will resume.

By preventing the vehicle from turning into an obstacle, the Buffer Zone helps smooth out the obstacle avoidance maneuver. Without a Buffer Zone, the vehicle will avoid an obstacle, then turn back into it when waypoint navigation resumes (Figure 7-11). As a result, the vehicle exhibits jerky navigation around an obstacle. By implementing a Buffer Zone, the vehicle will avoid the obstacle, drive straight until the obstacle is clear of the Buffer Zone, and then steer for the waypoint (Figure 7-12).

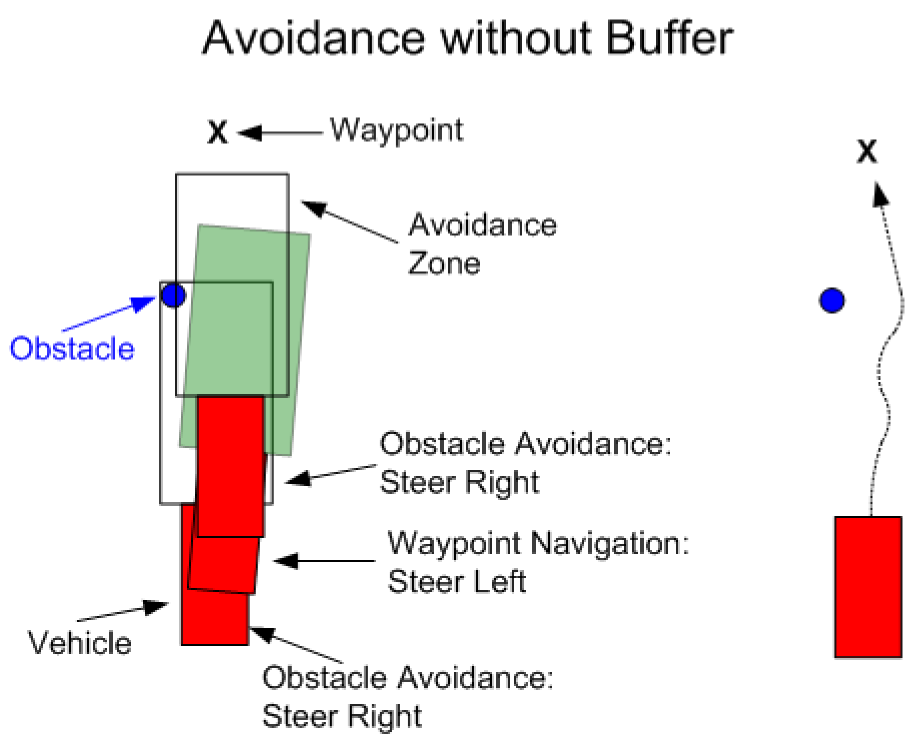


Figure 7-11. Obstacle avoidance without a Buffer Zone exhibits jerky navigation

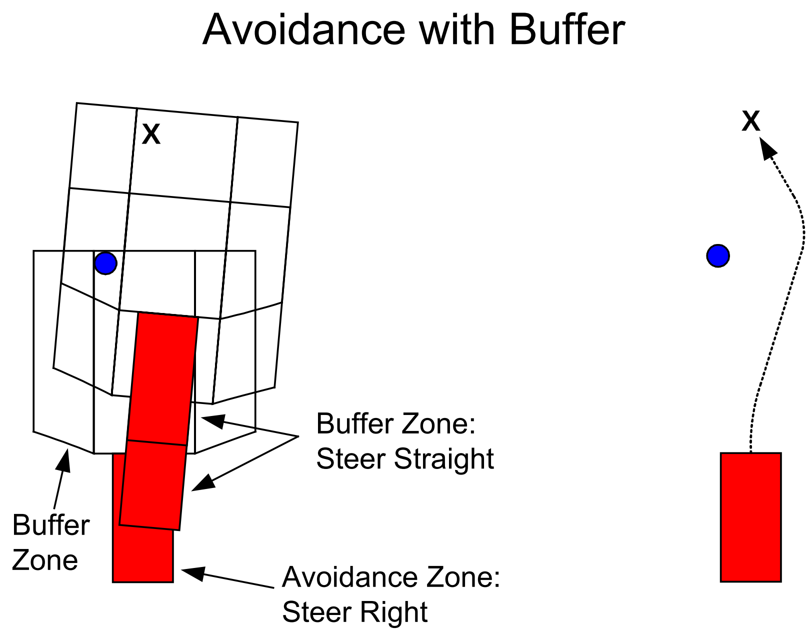


Figure 7-12. Obstacle avoidance with a Buffer Zone exhibits smooth navigation

7.6 Dynamic Expanding Capability

The length and width of the Avoidance and Buffer Zone play a large role in the obstacle avoidance capability of the Dynamic Expanding Zone algorithm. If the Avoidance Zone is too long, the vehicle may unnecessarily attempt to avoid obstacles. If the path of the vehicle naturally avoids an obstacle, no obstacle avoidance is required, even if the object is directly in front of the vehicle. If the Buffer Zones are too large, the vehicle will make an excessively large maneuver to avoid even a small obstacle. To optimize navigation for different situations, the length of the Avoidance Zone and Buffer Zone change dynamically.

The length of the Avoidance Zone is controlled by the projected path of the vehicle, which is assumed to be a clothoid curve. Clothoid curves have a linear change in curvature as a function of the arc length. Such a curve gives a good approximation of the trajectory an Ackerman steered vehicle would take if the angular steer velocity of the wheels are constant [10]. The equations of motion are

$$\frac{d\theta(s)}{ds} = \gamma(s) = ks + \gamma_i$$
$$\theta(s) = \frac{ks^2}{2} + \gamma_i s + \theta_i$$
$$\frac{dx}{ds} = \cos(\theta)$$
$$\frac{dy}{ds} = \sin(\theta),$$
$$x(s) = \int_0^s \cos\left(\frac{kx^2}{2} + \gamma_i x + \theta_i\right)dx$$
$$y(s) = \int_0^s \sin\left(\frac{kx^2}{2} + \gamma_i x + \theta_i\right)dx$$

where θ, γ, k, and s are heading, curvature, curvature rate, and distance along the clothoid. The equations of motion can be parameterized in Cartesian space as shown in the x(s) and y(s) equations. By inputting both the current steering angle and steering rate, the DEZ algorithm approximates the distance the wheels will travel within the fixed widths of the Avoidance Zone (Figure 7-13). The length of the Avoidance Zone is determined by the intersection of the clothoid path and zone width. Similar to all the other reactive behaviors, the length of the avoidance zone is recalculated every iteration, allowing the zone to dynamically change.

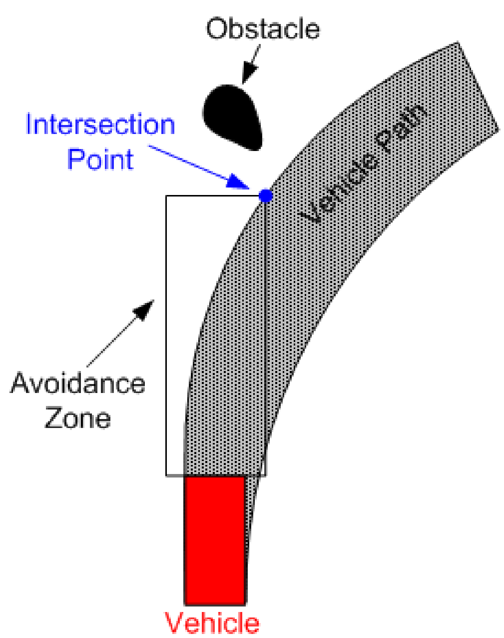


Figure 7-13. Avoidance Zone length is limited by Clothoid trajectory

Similar to the Avoidance Zone, the buffer is also dynamically controlled based on the steering angle. The buffer length is always equal to the length of the Avoidance Zone. The width of the buffer, however, increases in size as the steering angle increases. The vehicle will hit an obstacle laterally further away if the steering angle is larger. This expansion keeps the vehicle from turning into an obstacle that is only hazardous in aggressive turns (Figure 7-14).

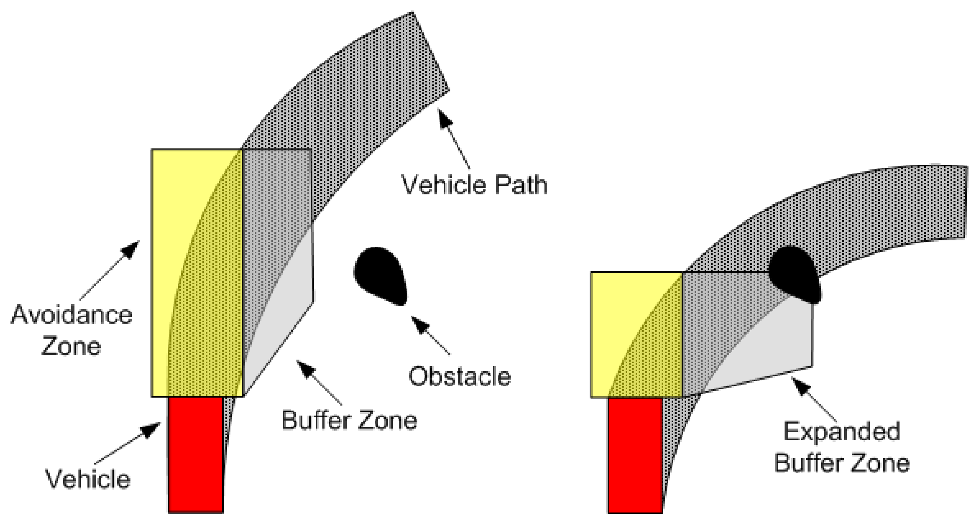


Figure 7-14. Buffer Zone width expands as the steering angle increases

Chapter 8
Speed Control and Rollover Prevention

As the demand for higher speed mobile robots increases, the need for intelligent speed control and rollover prevention becomes increasingly critical for vehicle safety. When traveling at low speeds, vehicles can easily navigate through obstacles without considering any vehicle dynamics. The vehicle can make extremely tight turns without worry of skidding or rollover. However, as Figure 8-1 shows, vehicles traveling at higher speeds are extremely susceptible to rollover, especially if driven offroad. The DARPA Grand Challenge pushed autonomous vehicles to extremely high speeds (25+ mph). Autonomous vehicles now traveling at these speeds require a reflexive behavior based on a vehicle model to prevent dangerous maneuvers.

Figure 8-1. Rocky rolled during the 2005 Grand Challenge site visit

The speed control and rollover prevention are the highest priority behavior. If the vehicle is driving at a speed and steering angle deemed as unsafe, the rollover prevention will reduce the speed and limit the steering angle. Since obstacle avoidance is subsumed by rollover prevention, the vehicle would collide with an obstacle over rolling over (Figure 8-2). This chapter will first discuss speed control and later rollover prevention.

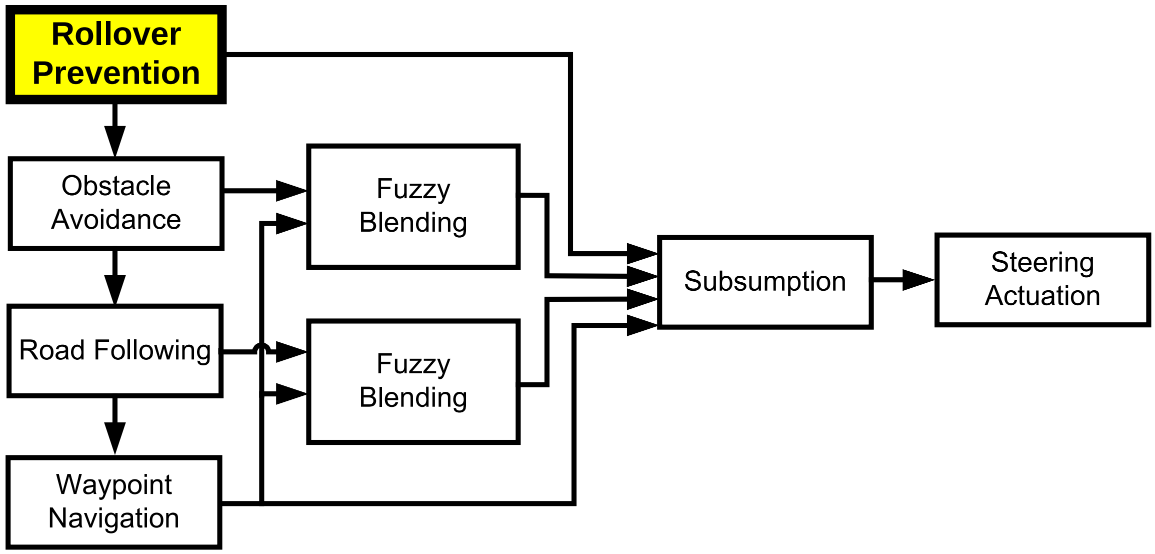


Figure 8-2. Rollover prevention behavior within the Dynamic Expanding Zones reactive architecture

8.1. Speed Control

Similar to all reactive behaviors speed control is a Sense-Act couple, where a new speed is commanded every cycle. Speed control is critical for properly avoiding obstacles, staying within boundaries, and preventing rollover. Ideally, the vehicle would run at top speed throughout the course; however, this is generally not achievable since the vehicle must make turns to follow waypoints and maneuver around obstacles. As a result, when the vehicle is executing a turn, the speed will be reduced to prevent rollovers. Also, since the wheels can not be turned instantaneously, the vehicle is slowed to allow the actual steering to track the desired path.

Even though speed control follows the reactive approach, the vehicle can anticipate future maneuvers and take precautionary action similar to a deliberative approach. For example, the vehicle will slow down if an obstacle is in its path before it starts to avoid the obstacle. The speed is governed by how fast the vehicle can safely maneuver around the obstacle without rolling over. The vehicle also anticipates a turn at a waypoint by slowing to a safe speed before it enters the turn.

8.2. Rollover Prevention

After experiencing two vehicle rollovers, a simple rigid body model of the vehicle was created. Rollover typically occurs when the moment around the outside tire from the centripetal force is larger than from the vehicle weight [8]. Centripetal forces also cause suspension deflections (body roll), resulting in a shift of the center of gravity toward the outside wheel. This shift reduces the counteracting moment from the vehicle weight. A rollover can also be caused by an obstacle or negative obstacle (i.e. ditch) "tripping" the vehicle.

The vehicle can be more accurately modeled by accounting for such factors as body roll and suspension effects. However, for simplicity these effects were neglected. A straightforward rigid-body approach was favored over a more complex and computationally expensive model (Figure 8-3). Since the off-road environment is highly unpredictable, an extremely accurate model is unnecessary. To account for the effects of unpredictable terrain, a factor of safety is implemented in each calculation.

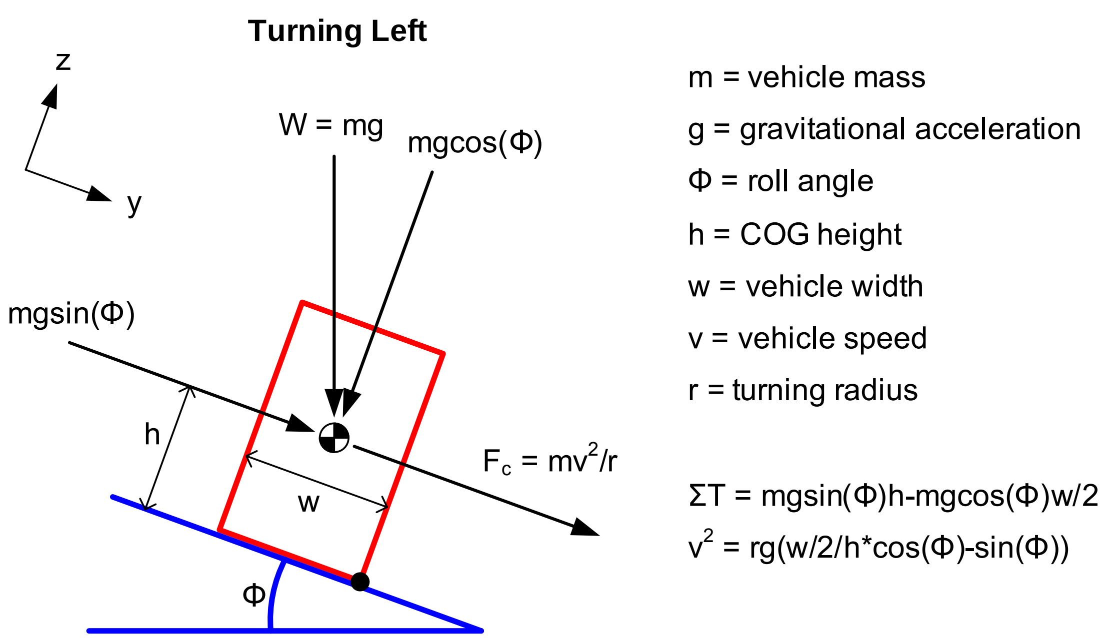


Figure 8-3. Rigid body model of the vehicle in a turn.

Using this model, four values are calculated: safe speed, urgent speed, safe steering angle range, and steering angle that results in the most stability. The safe speed and safe steering angle range specify an envelope in which the vehicle can operate safely. The vehicle will always attempt to operate within this envelope. The urgent speed signifies that a rollover is imminent. In the event that the current speed is greater than the urgent speed, the vehicle will ignore the steering and speed commands and will attempt to prevent rollover. Stability can be achieved by steering to an angle which results in the an even distribution of the vehicle weight on all the wheels. Meanwhile, the speed is reduced to the safe speed. Once the speed is no longer above the urgent speed, the vehicle will resume steering commands to the algorithm flowcharted in Figure 8-4. On flat terrain, this stability occurs when the wheels are straight forward. Moving perpendicular to a hill may cause the vehicle to lean right. If instability occurs in this case, the vehicle will steer right in order to regain stability.

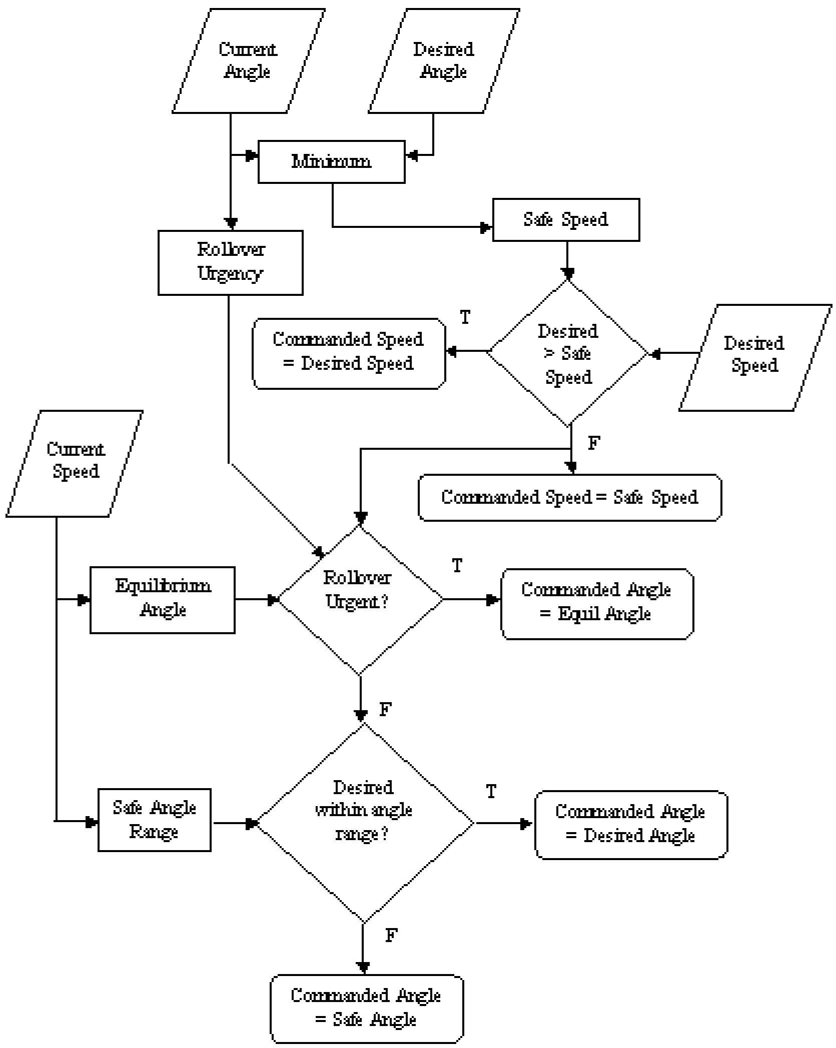


Figure 8-4. Flowchart of rollover prevention

Chapter 9

Results, Future Work, and Conclusions

This chapter will investigate the strengths and weaknesses of each concurrent behavior of the Dynamic Expanding Zones algorithm. No claim has been made that this reactive algorithm is the optimal solution for high speed navigation; however, Dynamic Expanding Zones has proven effective for high speed (25+ mph) reactive based obstacle avoidance. DEZ produces smooth, decisive navigation control without use of any generated optimal trajectories or paths. The complexities of generating a map and paths are replaced with simple modular Sense-Act behaviors. The high computational speed allows this approach to be used effectively for high speed navigation. These concurrent behaviors allow for efficient incremental development and testing. With less than a year to develop a fully autonomous vehicle for the Grand Challenge, this incremental development proved to be extremely beneficial. Many developers believe that a deliberative approach is the only way for autonomous navigation. Virginia Tech has shown that a reactive approach can still be effectively utilized for such a complex problem as the Grand Challenge. This approach was used on both Virginia Tech vehicles, and the vehicles successfully placed 8th and 9th in the 2005 Grand Challenge.

9.1. Waypoint Navigation

As Chapter 5 discussed, Dynamic Expanding Zones utilizes the Potential Field algorithm instead of a path tracking algorithm for waypoint navigation. Many other waypoint reactive navigation algorithms have been effectively implemented in the past; however, the point-to-point waypoint navigation algorithm provides a computationally simple means for creating a smooth, efficient navigation to the waypoints. The Potential Field algorithm uses a simple proportional controller to aim the vehicle toward the waypoint.

Chapter 3 discussed the problems associated with designing Potential Field magnitude profiles for obstacles. Designing a smooth emergent overall path around obstacles, especially a cluster of obstacles, becomes extremely complex at high speeds. For this reason, the Potential Field algorithm is only used for waypoint navigation. The magnitude of the field is constant and independent of the proximity to the waypoint. No special techniques are needed to make the Potential Field algorithm work for waypoint navigation.

As long as the frequency of the controller is high, it can be modeled in continuous time. Figure 9-1 shows the vehicle's overall path, when the Potential Field algorithm is run at 10 Hz. The initial position, initial heading, and waypoint position are (5m, 5m), 10 degrees West of North, and (1000m, 1000m), respectively. The vehicle runs at a constant speed of 5mph. Figure 9-2 shows how the heading error tracks to zero with an initial condition of 55 degrees heading error. Heading error is the angle between the vehicle's heading and the direction of the attractive force. This particular case has a 31.2% overshoot. The lower the overshoot, the less steering is required. For the future, different types of controllers, such as PID and lead-lag, should be evaluated for Potential Fields. Alternative controllers can be used to decrease overshoot and settling time.

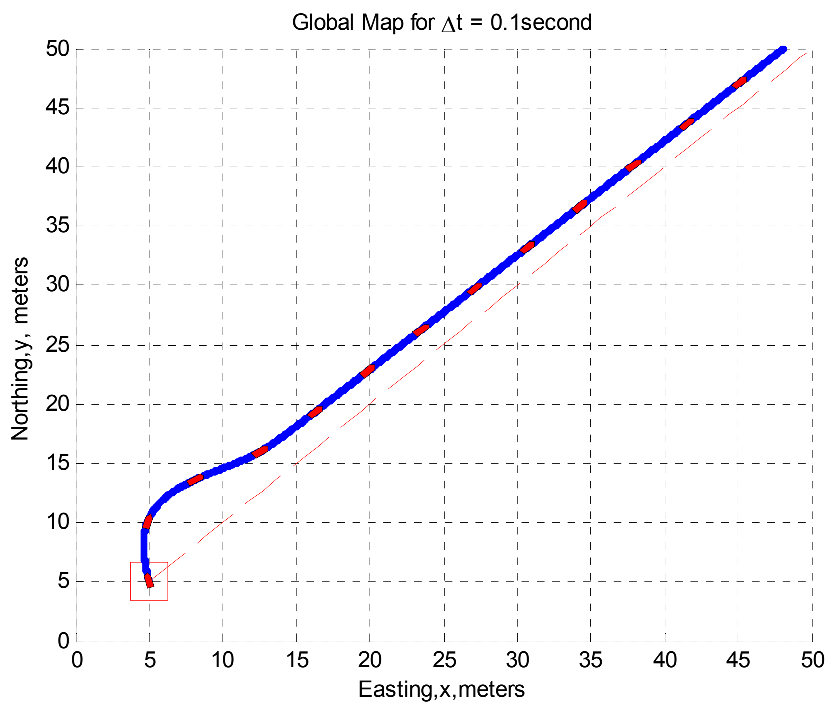


Figure 9-1. Top view of a simulated Potential Field generated overall path. The blue line indicates the path of the vehicle. The red square displays the initial position of the vehicle. The red dashed line connects the initial position and the waypoint. The solid red pentagon shows the position and orientation of the vehicle at a specified time interval.

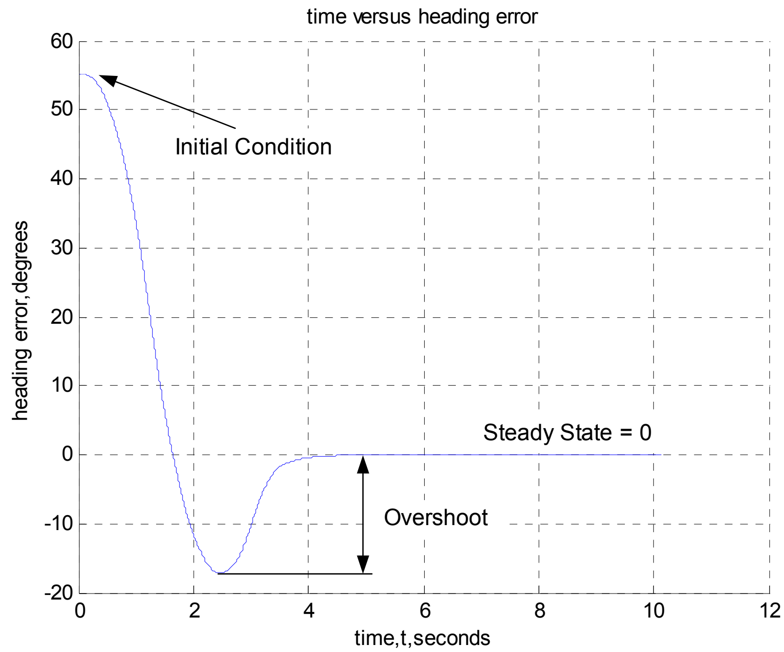


Figure 9-2. Simulated Potential Field time response.

The proportional control of the Potential Field algorithm is robust to system disturbances. A common disturbance is the phenomena called GPS pop. For many commercially available GPS/INS integrated systems, the GPS data is fused with the inertial navigation data to produce the most accurate position solution. The consequence of outputting the most accurate available position is the loss of a smooth solution. The GPS/INS system can "pop" from an inaccurate solution one iteration to a more accurate solution the next (Figure 9-3). These "pops" can be position changes of meters in some cases, and is completely independent of the vehicle motion. A GPS pop typically occurs when a GPS signal is occluded temporarily. The position solution is temporarily computed solely from the INS data. The INS, a relative position sensor, accrues absolute position error over time. When the GPS signal is reacquired, the commercial GPS/INS systems typically "pop" the inaccurate position back to the more accurate GPS position.

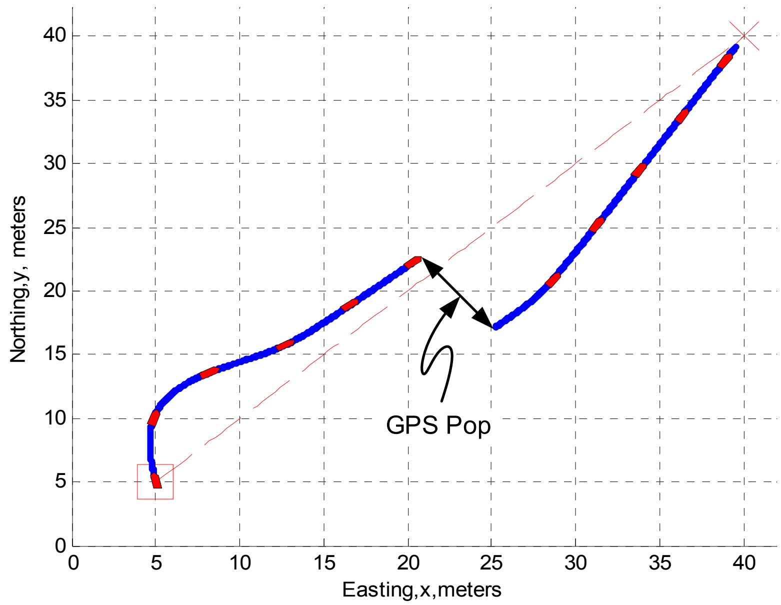


Figure 9-3. Simulated waypoint navigation with a GPS "pop."

For deliberative maps, a GPS Pop can cause all the stored map data to offset by the amount of the GPS Pop. With an inaccurate map, a deliberative approach could plan a path directly into an obstacle it assumes it is avoiding. Since Potential Field does not use a deliberative map to plan a path, this algorithm can easily react to disturbances. Essentially, a GPS pop is an impulse disturbance into the plant (Figure 9-4). Figure 9-5 shows how the system tracks the heading error of zero after the disturbance occurs. The system maintains stability through a GPS pop.

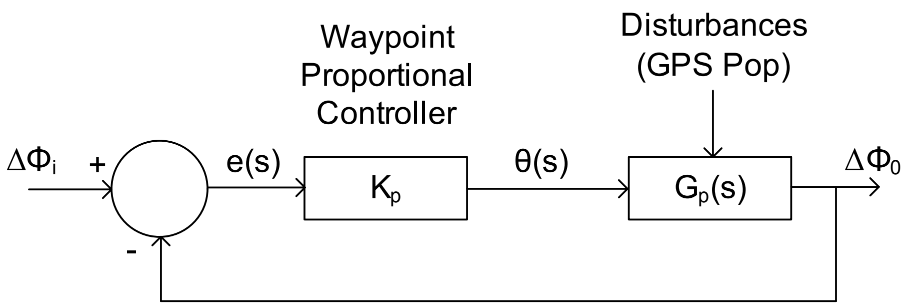


Figure 9-4. Proportional control including a GPS pop disturbance.

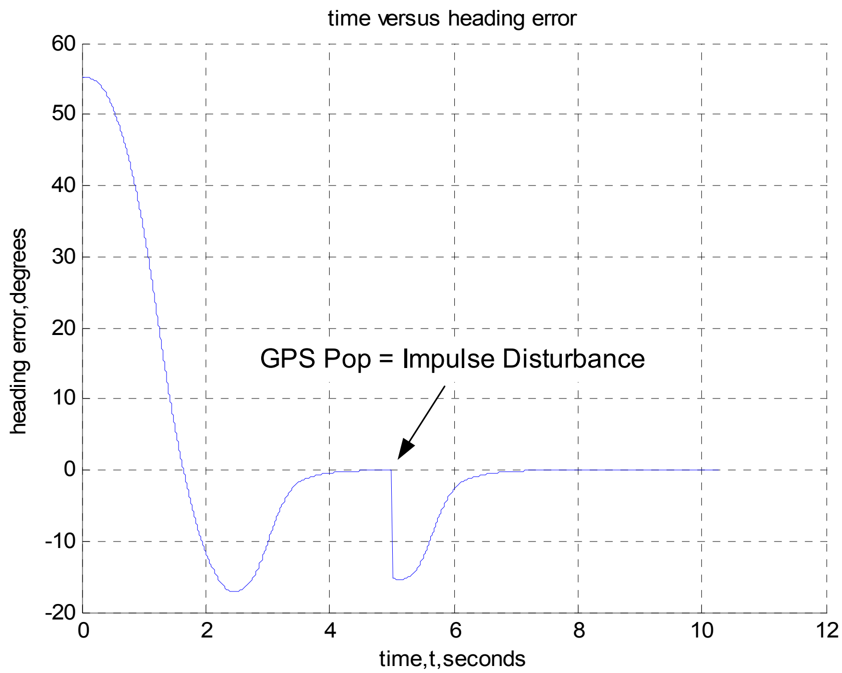


Figure 9-5. Potential Field is stable through a GPS pop.

Since the vehicle is non-holonomic (non-zero radius turn), the heading of the vehicle will not always align with the attractive field. As shown earlier in Figure 9-2, the system takes 4 seconds for the vehicle's heading to align with the attractive field. Since the simulation was run at a vehicle speed of 5 mph, the vehicle must drive about 9m for the heading to align to the field. As a result a non-holonomic vehicle using Potential Field will exhibit instability or steady state error in close proximity to waypoints. This effect is similar to the effects of gravitational force field on a satellite. A satellite that has enough tangential velocity will orbit a planet. Similarly, a vehicle that has "tangential velocity" will attempt to "orbit" the waypoint as shown in Figure 9-6. A more complex potential field can be created to deal with this problem; however, adding complexity to the attractive field nullifies the advantage of the Potential Field algorithm being computationally simple. As discussed in Chapter 5, the Dynamic Expanding Zones algorithm completely removes the attractive field around a waypoint, once the vehicle is within the waypoint radius. A new field is created around the next waypoint.

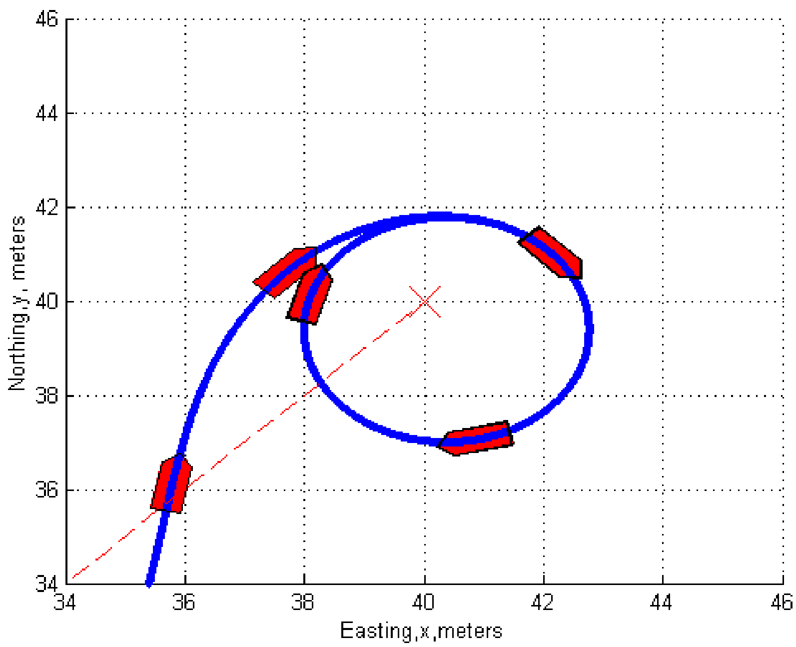


Figure 9-6. Simulated "orbit" around a waypoint. The red 'x' indicates the waypoint location.

With the high update rates and the implementation of the waypoint radius, the Potential Field is a computationally simple algorithm for smooth waypoint navigation. Figure 9-7 displays position data taken from the VTGC vehicle, Rocky, during waypoint navigation. The white line is the path taken by the vehicle. The red line is the track line, the line connecting waypoints. The yellow dot is the waypoint. The blue arrow illustrates the direction of travel. The first path segment clearly shows that vehicle is not tracking the red center line. Instead, the vehicle is aiming directly at the waypoint. The vehicle is also not attempting to drive over the waypoints. Once it enters the waypoint radius, the vehicle proceeds to the next waypoint.

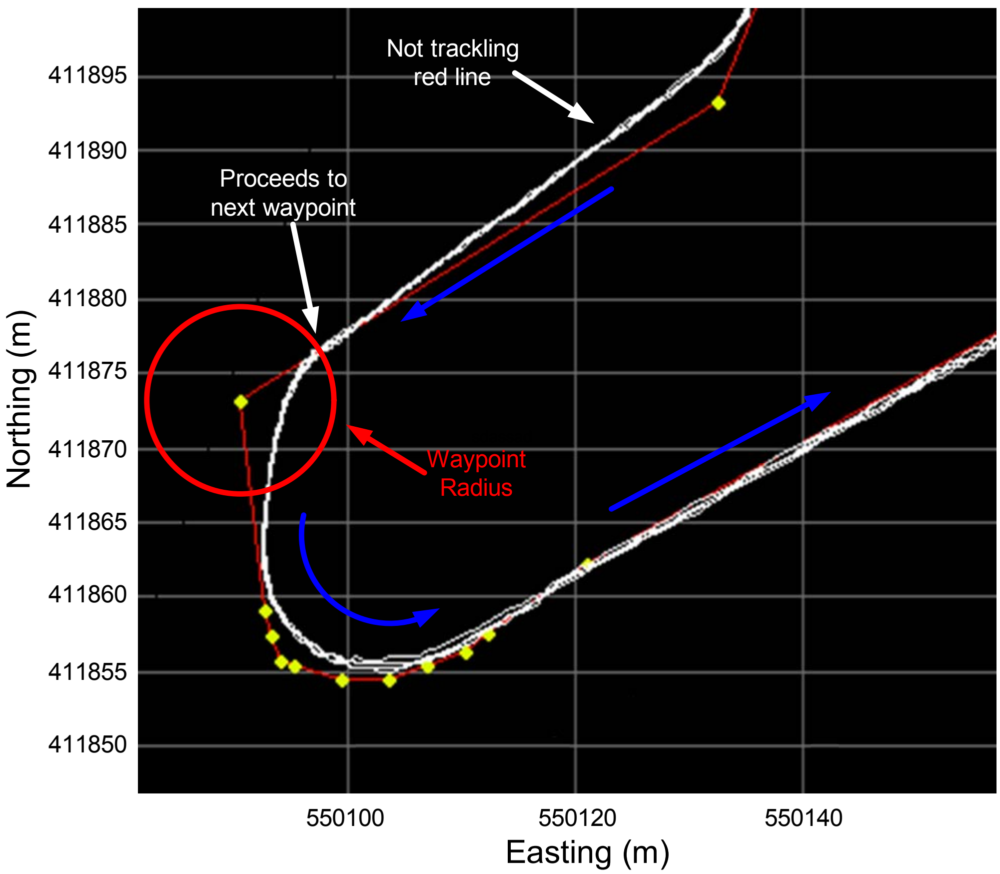


Figure 9-7. Waypoint navigation data taken from the vehicle. The red line, white line, yellow dots, and blue arrows are the course centerline, vehicle path, waypoints, and direction, respectively [9].

As mentioned in the previous chapter, this algorithm is sensitive to update rates. When the algorithm is run at a high frequency, the control can be assumed a continuous controller. However, as frequency decreases, the control exhibits discrete control behavior. The damping in the system actually decreases as the update frequency decreases. If the update frequency decreases even more, the system will go unstable and eventually chaotic (Figure 9-8). In addition, increasing the vehicle speed and decreasing the steering rate reduces the damping in the system. The simulation shown in Figure 9-8 has the same system parameters at the simulation shown in Figure 9-1; however, the update rate is varied.

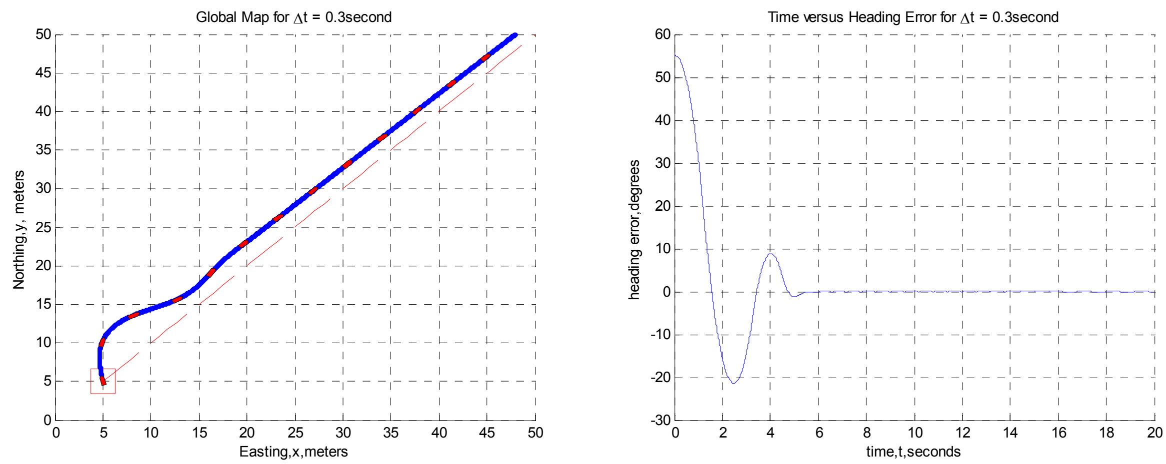


Figure 9-8A. Simulated waypoint navigation at 3.33Hz. The left figure is the top view of vehicle path. The right figure illustrates a system response plot

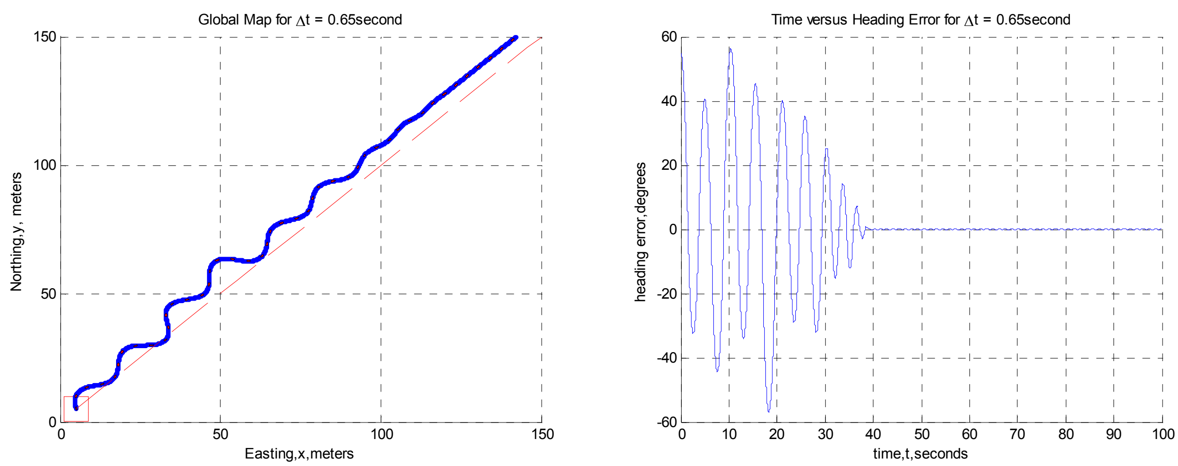


Figure 9-8B. Simulated waypoint navigation at 1.54Hz.

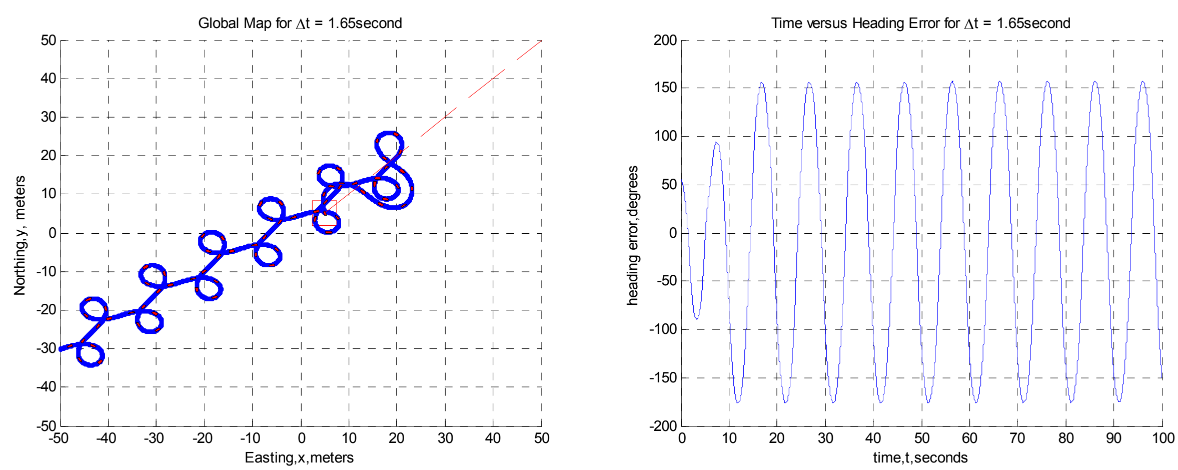


Figure 9-8C. Simulated waypoint navigation at 0.61Hz.

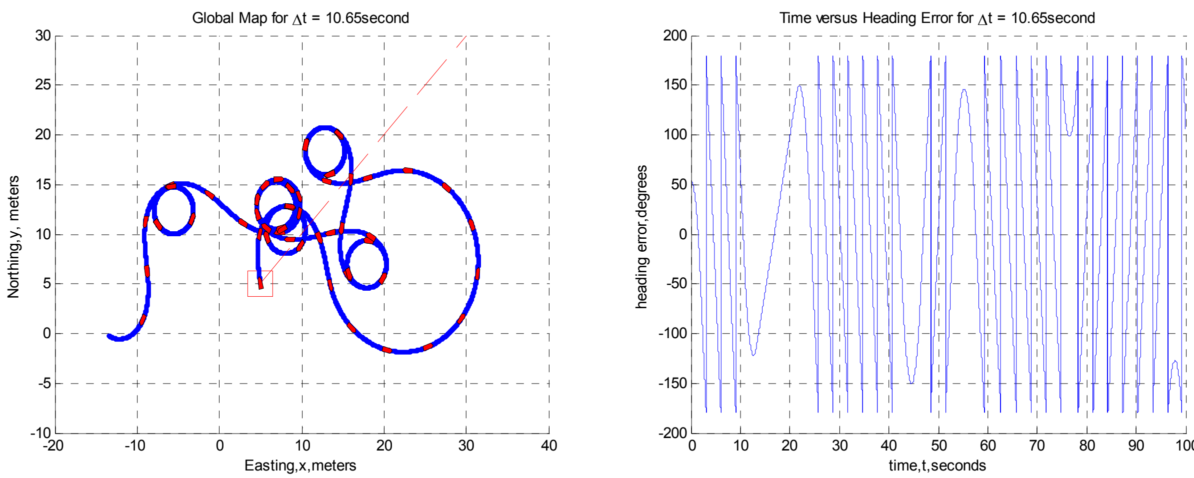


Figure 9-8D. Simulated waypoint navigation 0.09Hz.

The discrete effects on the waypoint controller can be further explained using the block diagram shown in Figure 9-9. Even though the controller is discrete, the transfer function of the plant is continuous. The discrete controller samples a continuous signal and commands a control effort at a period T. The k refers to the iteration of the controller. The discrete input to the plant is a zero order hold (ZOH) as shown in Figure 9-10. As a result, the average discrete input lags a continuous input by half the period [18]. This lag is the reason for the reduction in damping and instability.

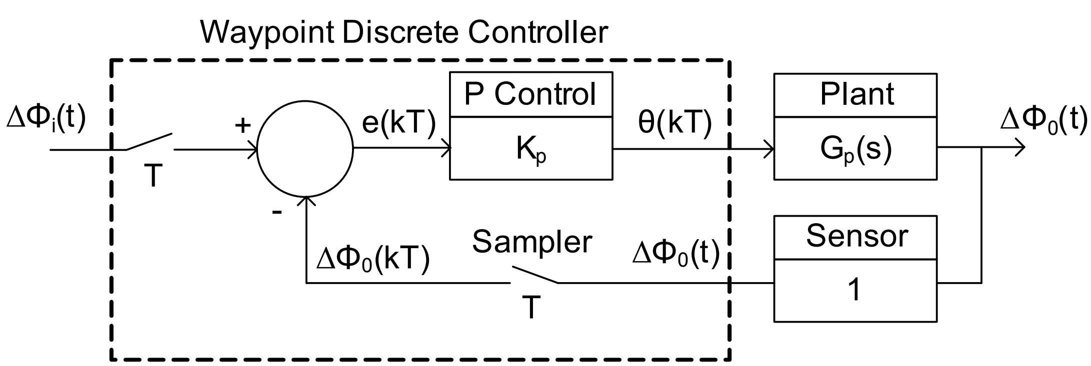


Figure 9-9. Discrete waypoint control block diagram.

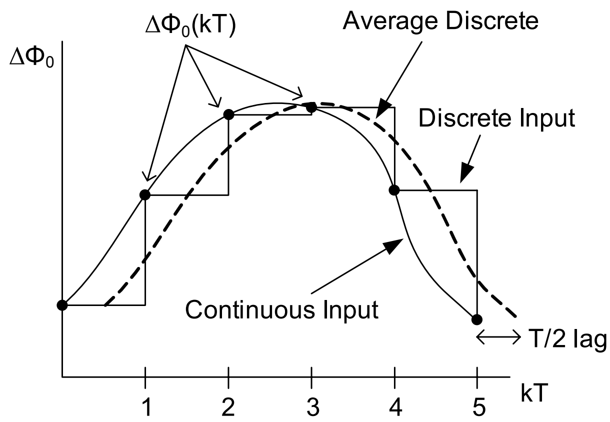


Figure 9-10. Discrete input to the plant is a zero order hold (ZOH) [18].

9.2. Road Following

As Chapter 6 discussed, road following uses a hybrid Pure Pursuit-Potential Field method to track the road. Instead of driving toward a set of static road point similar to the waypoint navigation, the ideal course of action is to track the path (road). The Pure Pursuit algorithm has been successfully implemented by many as a path tracking algorithm. The lookahead distance for the Pure Pursuit algorithm is critical to the behavior of the system. The lookahead distance essentially acts as the damper in the system. The longer the lookahead distance, the larger the damping is. To prevent overshoot in the navigation of a vehicle traveling and varying speeds, the lookahead distance needs to be dynamic.

Currently, the distance to the closest valid road center point is the Pure Pursuit lookahead distance. In the future, the lookahead distance should be dependent on vehicle speed instead of the closest road point. By controlling the lookahead distance, the overshoot of the vehicle navigation can be reduced. For example, if the vehicle is traveling at a high speed, the lookahead distance can be increased to reduce overshoot problems. Once a lookahead distance is determined, a valid road point at a lookahead distance away should be tracked.

Once the closest valid road point is determined, the current road following algorithm uses the Potential Field algorithm to track the point. The decision to use Potential Fields was mainly based on the successful implementation of the waypoint navigation. In the future, road following should use the true Pure Pursuit algorithm to command a steering angle. Instead of using the heading error as the controller input, the Pure Pursuit algorithm uses the lateral distance from the path as the controller input. The derivation of the steering command and illustration of the Pure Pursuit geometry is shown in Figure 9-11. The Pure Pursuit algorithm still acts as a discrete proportional controller, thus it is sensitive to the update frequency of the controller. All the instability and overshoot problems associated with Potential Fields also apply to the Pure Pursuit algorithm.

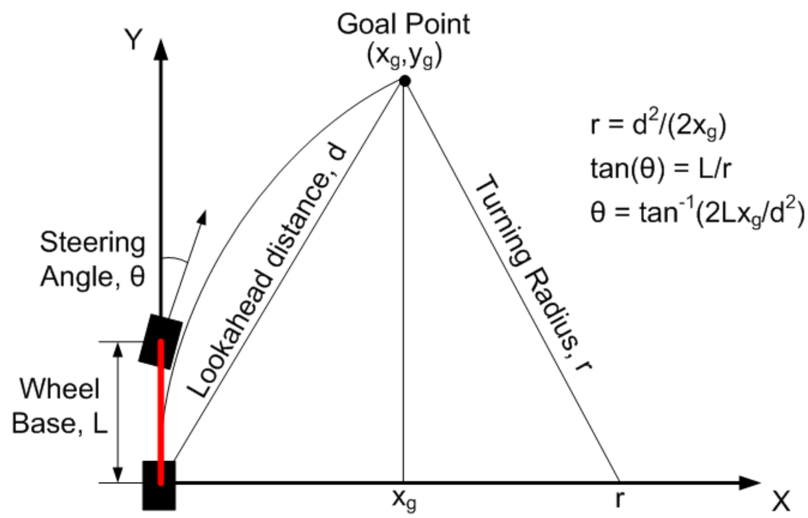


Figure 9-11. Illustration of Pure Pursuit geometry and derivation of steering command [12].

9.3. Obstacle Avoidance

The Potential Fields has been a popular reactive algorithm for implementing obstacle avoidance in many low speed vehicles. The attractiveness of this approach is largely due to the simplicity in summing attractive/repulsive forces to generate a desired steering direction and speed. However, in its simple form, Potential Fields is sensitive to obstacles not even in the path of the vehicle [6]. Potential Fields algorithm is not easily portable from one vehicle/application to another. The overall behavior is largely dependent on the speed of the vehicle and maneuverability of the vehicle. For example, if the vehicle in Figure 9-12 was traveling at a higher speed, the magnitude profile around the obstacle would have to prevent the vehicle from hitting the second obstacle. In addition, if multiple obstacles are in close proximity, Potential Fields can exhibit oscillatory navigation. Even worse, the vehicle can stop at a local minima, where all the attractive/repulsive forces sum to zero.

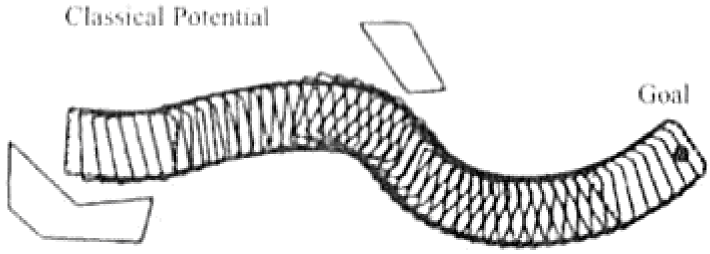


Figure 9-12. Vehicle is repelled from the first obstacle [6].

Techniques have been developed to create a smooth overall behavior for Potential Fields. Figure 9-13 shows how navigation templates (NaTs), developed by Marc Slack at JPL, are used to intelligently sum the force vectors with a heuristic toward the goal point [5]. This heuristic is a rotation (tangential) potential field based on the orientation of the robot in relation to the obstacle. Figure 9-14 illustrates how rotation potential fields developed by Raja Chatila keep the vehicle from avoiding obstacles outside the path of the vehicle. Many other techniques exist; however, by adding more and more complexity to Potential Fields, the main benefit of simplicity is lost. Furthermore, added complexity reduces the portability of the software.

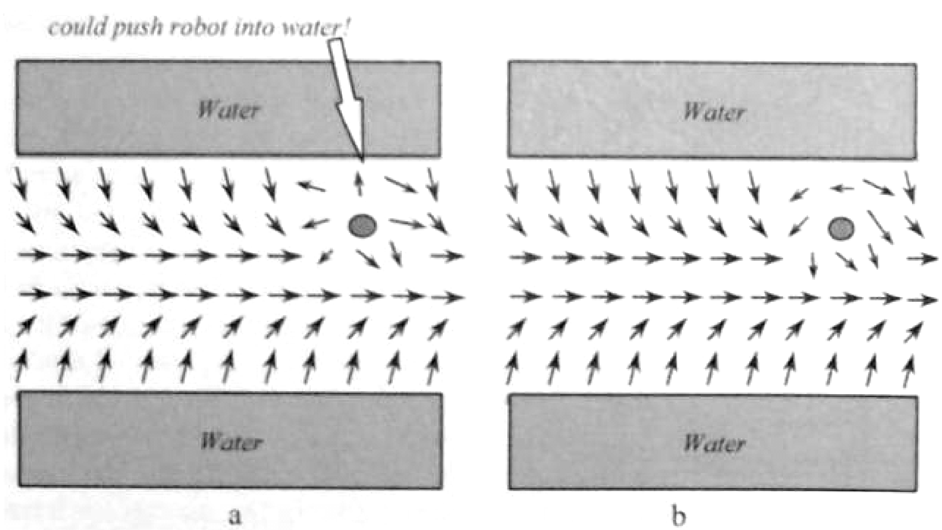


Figure 9-13. a) Classical Potential Fields may force the vehicle into the water.
b) NaTs keeps the vehicle in the center of the bridge [5]

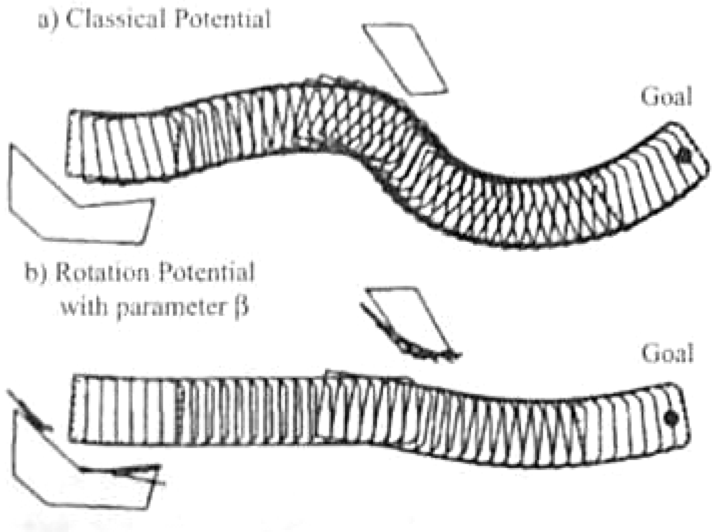


Figure 9-14. a) Vehicle using Classical Potential is repelled from the first obstacle.
b) Vehicle using Rotational Potential is not repelled. [6]

Dynamic Expanding Zones takes advantage of the simplicity of Potential Fields for waypoint navigation, while removing the complications associated with obstacle repulsive fields. Instead of utilizing a creative approach for summing the force vectors, Dynamic Expanding Zones evaluates only the area the vehicle will be traversing across. The perception zone expands and shrinks to fit the path of the vehicle. Figure 9-15 shows how Dynamic Expanding Zones navigates the same course as Figure 9-12. Dynamic Expanding Zones does not avoid obstacles outside the vehicle's path and produces a smooth overall behavior toward the goal point. Dynamic Expanding Zones is a simple method for smooth high speed obstacle avoidance. This algorithm was successfully employed for the Grand Challenge Qualification Course and Event Course at speeds up to 25 mph.

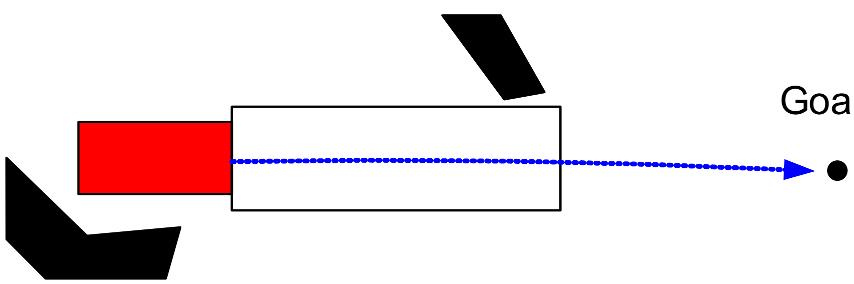


Figure 9-15. Dynamic Expanding Zones does not avoid obstacles outside the path as Potential Fields does in Figure 9-12.

As discussed in Chapter 7, the steering angle is calculated assuming the vehicle will drive a constant arc. Figure 9-16 shows a circular arc and a clothoid arc projected from the left front tire when the vehicle is making a right turn. Assuming the clothoid path accurately models the actual vehicle trajectory, the circular arc does not accurately project the vehicle's trajectory. The inaccuracy of the circular arc is handled by the control feedback nature of the reactive behavior. If the steering angle is too small during a particular iteration, the next iteration will command a larger steering angle. Clothoids could be used to calculate an accurate steering angle, but this would add unnecessary complexity.

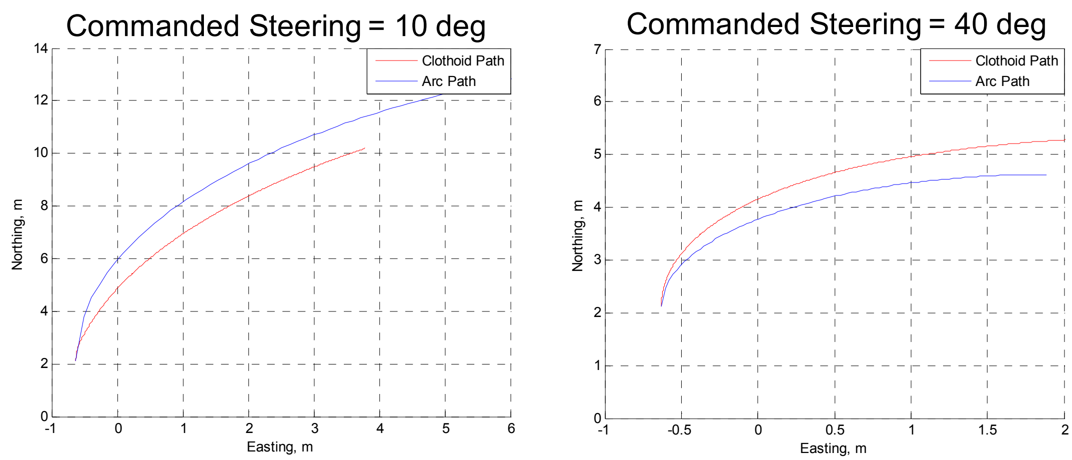


Figure 9-16. Predicted path using circular arcs and clothoids.

The Dynamic Expanding Zones approach still has many of the same problems as other reactive approaches. DEZ does not attempt to plan an optimal path; instead, an emergent path is generated by commands in discrete time. Since the future state of the vehicle is not considered, the algorithm can navigate the vehicle into a dead end, like a cul-de-sac (Figure 9-17). A simple method for dealing with the dead-end condition does not currently exist. Future research can be done to find a solution to this problem.

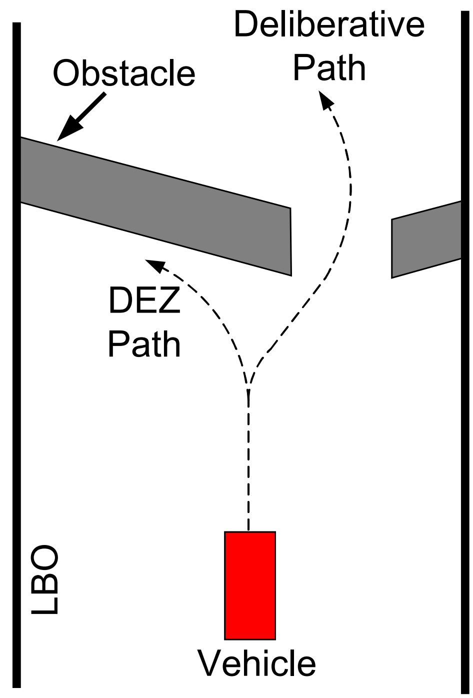


Figure 9-17. Illustrates two situation when Dynamic Expanding Zones could lead the vehicle into a dead end.

The geometry and size of the perception zones are extremely important in the overall emergent behavior. As shown in Figure 9-18, the vehicle will avoid unnecessary obstacles if the Avoidance Zone is too long or the Buffer Zone is too wide. For simplicity, the avoidance and buffer zone are modeled as a rectangle and trapezoid. However, to increase the effectiveness of the obstacle avoidance, the zones need to mirror the actual vehicle path as shown in Figure 9-19. Clothoids can be used to create the boundaries of both the avoidance zone and buffer zone. The overall complexity of the algorithm is not greatly increased, because clothoids are not used to determine the path in any way. Instead, using the current steering angle and steering rate, the boundary can be found with little computation.

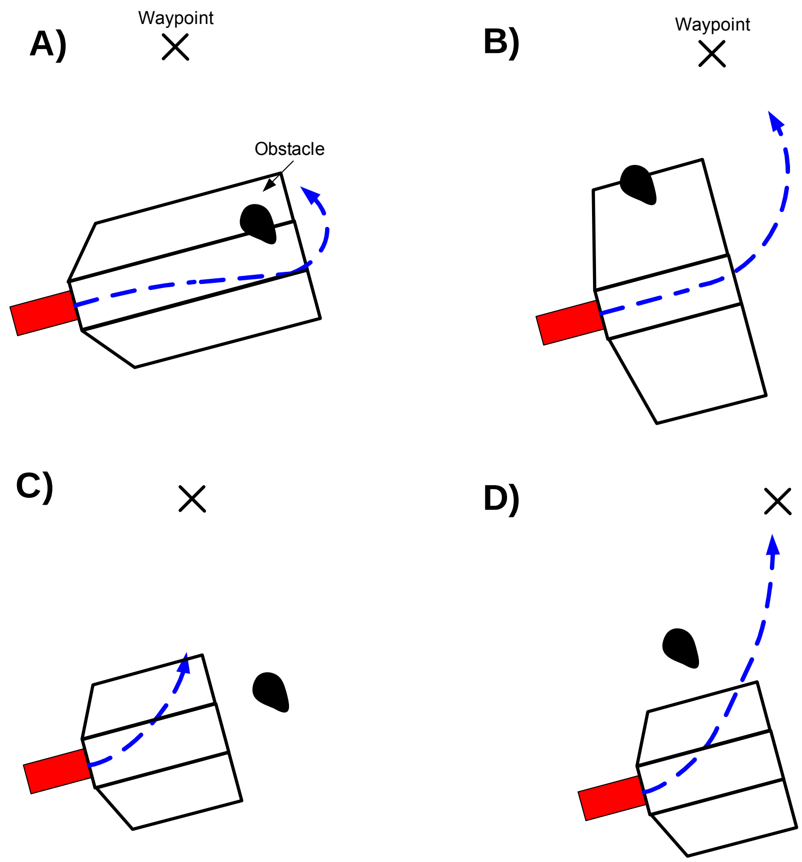


Figure 9-18. Size and geometry of perception zones effect overall emergent behavior. A) and B) display the resulting behavior of overly large zones. C) and D) show the resulting behavior if the zones are the correct size.

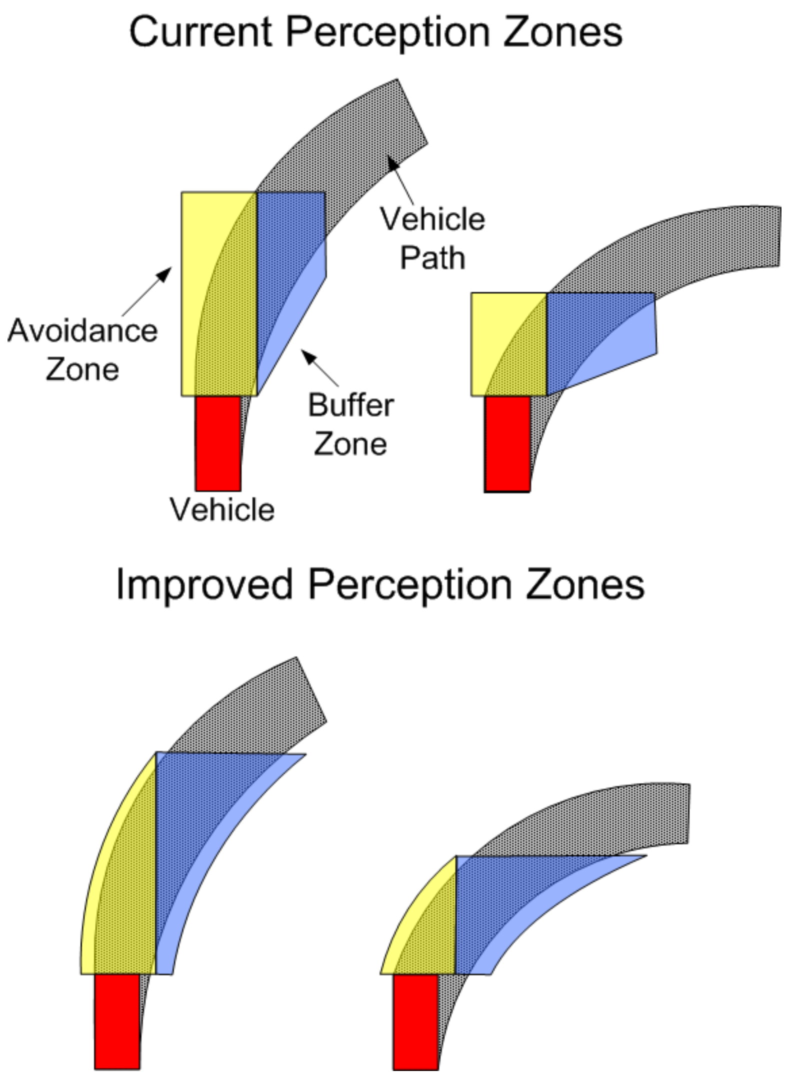


Figure 9-19. Clothoid curves can improve obstacle avoidance by shaping the zones to mirror the vehicle's path.

9.4. Speed Control and Rollover Prevention

The dynamics of an Ackerman vehicle on predictable, structured terrain is well understood and well documented. Several commercial modeling software packages are available to help engineers accurately model the dynamics of vehicle. However, as soon as the vehicle traverses unstructured terrain, the predictability of the vehicle dynamics worsens. The vehicle dynamics is extremely sensitive to the interaction between tires and the terrain. Unstructured terrain is no longer flat, homogeneous, or easily modeled. For example, a vehicle is more likely to slide if the vehicle is on loose gravel versus asphalt. Also, objects and ditches can cause the vehicle to rollover.

Since navigation over unstructured terrain is highly unpredictable, a simple rigid body model is used to determine the safety of the vehicle. The model provided a safe envelope in which the vehicle can operate without the risk of rollover. This envelope is reduced in size by a factor of safety to account for unpredictable off-road effects. This rigid model successfully prevents the vehicle from making unsafe maneuvers. However, since a simple model is used, the vehicle occasionally drives too conservatively.

Increasing the aggressiveness of the vehicle will require a more accurate model. Even though off road terrain is highly unpredictable, several effects independent of the terrain can be added to the model. For example, suspension effects and load transfer are applicable to both off-road and on-road driving. Ackerman steered vehicles will typically exhibit body roll during turns, as the centripetal force needed for turning also applies a lateral torque on the body (Figure 9-21). Similarly, braking will cause a longitudinal torque on the body. The lateral and longitudinal torques deflect the suspension, resulting in body roll and pitch. When the vehicle body rolls or pitches, the vehicle's center of gravity shifts and the weight distribution on the tires change [8]. As a result, incorporating body roll and pitch is important for creating operating envelope in which a vehicle can maneuver safely but aggressively.

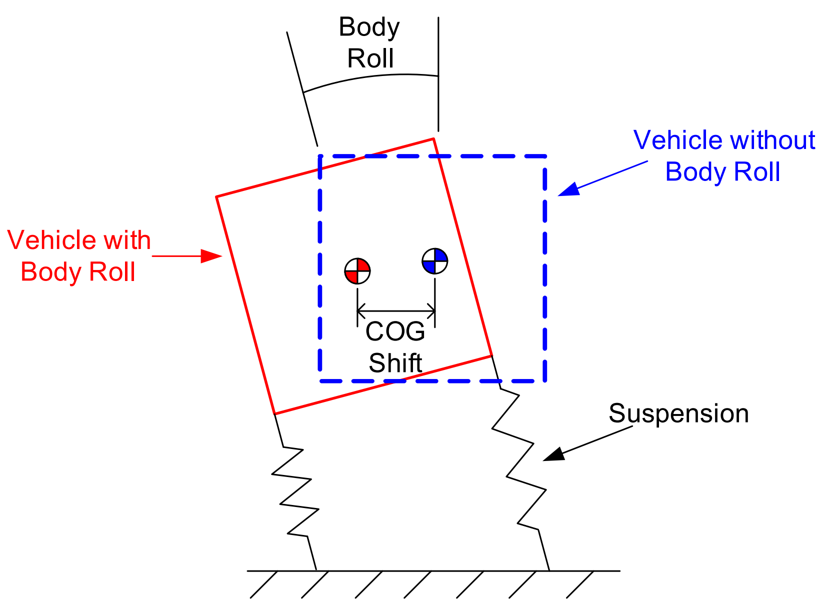


Figure 9-21. Body roll suspension effects [8].

9.5. Conclusion

By focusing on a set of perception zones, the Dynamic Expanding Zones facilitates smooth high speed obstacle avoidance with little computational complexity. The DEZ obstacle avoidance algorithm has been designed to work in parallel with existing reactive waypoint and path tracking algorithms. Similar to any reactive strategy, an optimal path or trajectory is not generated; instead, the overall path is emergent. As a result, DEZ does not have the potential of being as robust as a deliberative approach. On the other hand, a reactive architecture allows for incremental development/testing. For many high speed applications the algorithm and computation complexity of a deliberative approach outweighs the benefit of its capabilities. The Virginia Tech entry into DARPA Grand Challenge is a case where a reactive approach has been successfully implemented for high speed off road autonomous navigation.

References

1. *DARPA Grand Challenge Congressional Mandate.* Available: http://www.darpa.mil/grandchallenge04/sponsor_toolkit/congress_lang.pdf
2. *Grand Challenge Initial Press Release.* Available: http://www.darpa.mil/grandchallenge05/InitialPressRelease.pdf
3. *DARPA Grand Challenge Rules*. Available: http://www.darpa.mil/grandchallenge05/Rules_8oct04.pdf
4. *DARPA RDDF Document.* Available: http://www.darpa.mil/grandchallenge05/RDDF_Document.pdf
5. Murphy, R., *Introduction to AI Robotics*. 2000: MIT Press.
6. R Siegwart, I.N., *Introduction to Autonomous Mobile Robots*. 2004: MIT Press.
7. *Blue Team.* Available: http://www.ghostriderrobot.com/
8. Gillespie, T., *Fundamentals of Vehicle Dynamics*. 1992, Warrendale: SAE, Inc.
9. Leedy, B., *Two Minds for One Vehicle: A Case Study in Deliberative and Reactive Navigation*, in *Mechanical Engineering*. 2006, Virginia Tech: Blacksburg.
10. D. Shin, S.S., *Path Generation for Robot Vehicle Using Composite Clothoid Segments*, in *Robotics Institute*. 1990, Carnagie Mellon: Pittsuburgh.
11. A. Lacaze, Y.M., N. DeClaris, K. Murphy, *Path Planning for Autonomous Vehicles Driving Over Rough Terrain*, in *IEEE International Symposium Intelligent Control*. 1998.
12. Coulter, C., *Implementation of the Pure Pursuit Path Tracking Algorithm*, in *The Robotics Institute*. January 1992, Carnagie Mellon University: Pittsburgh.
13. Murphy, K., *Analysis of Robotic Vehicle Steering and Controller Delay*, in *IEEE International Symposium on Robotics and Manufacturing*. 1994: Wailea, HI.
14. Zadeh, L., *Fuzzy Sets.* Information and Control, 1965. 8(1): p. 338-353.
15. Zadeh, L., *Outline of a New Approach to the Analysis of Complex Systems and Decision Processes.* IEEE Trans. on Systems, Man, and Cybernetics, 1973. SMC-3: p. 28-44.
16. Ross, T., *Fuzzy Logic with Engineering Application*. 1995: McGraw-Hill, Inc.
17. Amidi, O., *Integrated Mobile Robot Control*, in *The Robotics Institute*. 1990, Carnagie Mellon University: Pittsburgh.
18. G. Franklin, J.P., M. Workman, *Digital Control of Dynamic Systems*. Third Edition ed. 1998: Addison Wesley.
19. Reynolds, C., *Steering Behavior for Autonomous Characters*, in *Game Developers Conference*. 1999, Miller Freeman Game Group: San Fransisco.

Appendix A

Grand Challenge Congressional Mandate

Congressional Mandate
http://www.darpa.mil/grandchallenge04/sponsor_toolkit/congress_lang.pdf

The DARPA Grand Challenge for autonomous robotic ground vehicles is the first in a series of grand challenges designed to reward scientific and technological achievement with cash prizes. The program was established in Congressional legislation, and is summarized the following excerpt:

> "Prizes for achievements in promoting science, mathematics, engineering, or technology education
>
> "The Secretaries of the military departments and the heads of defense agencies may each carry out a program to award cash prizes in recognition of outstanding achievements that are designed to promote science, mathematics, engineering, or technology education in support of the missions of the U.S. Department of Defense."
>
> -*National Defense Authorization Act for Fiscal Year 2003* (H.R. 4546, Sec. 2374b)

The decision to make the first DARPA Grand Challenge focused on autonomous robotic ground vehicles also reflects a Congressional mandate summarized below:

> "Unmanned Advanced Capability Aircraft and Ground Combat Vehicles
>
> It shall be a goal of the Armed Forces to achieve the fielding of unmanned, remotely controlled technology such that by 2015, one-third of the operational ground combat vehicles of the Armed Forces are unmanned."
>
> -*National Defense Authorization Act for Fiscal Year 2001* (S. 2549, Sec. 217)

Appendix B

Actuator PID Response

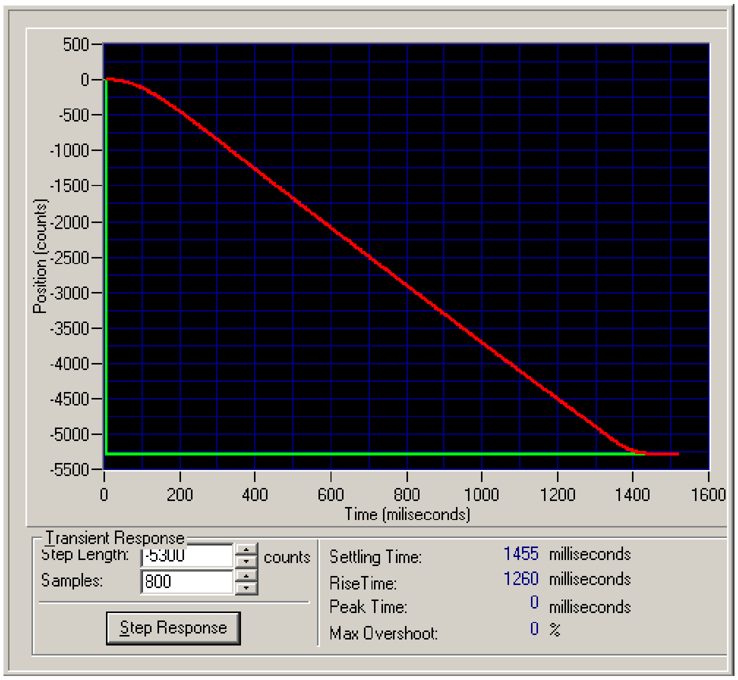


Figure B-1. Steering actuation from 0 to 40 degrees left.

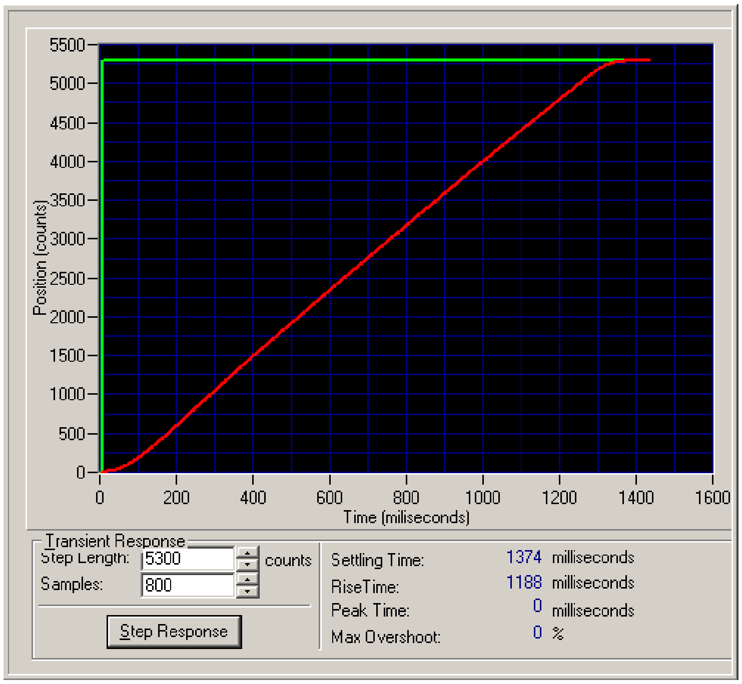


Figure B-2. Steering actuation from 0 to 40 degrees right

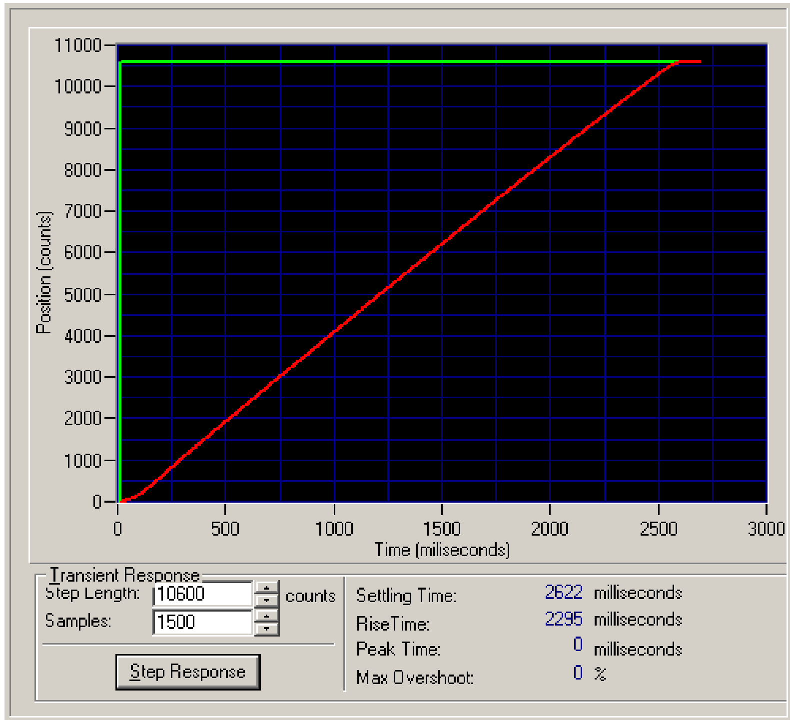


Figure B-3. Steering actuation from 40 degrees left to 40 right

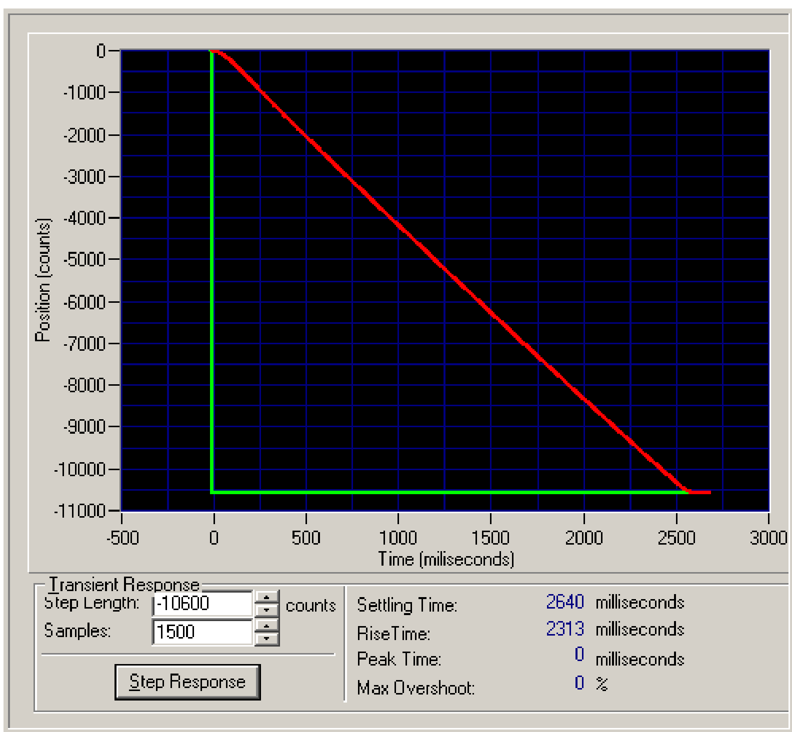


Figure B-4. Steering actuation from 40 degrees right to 40 left.

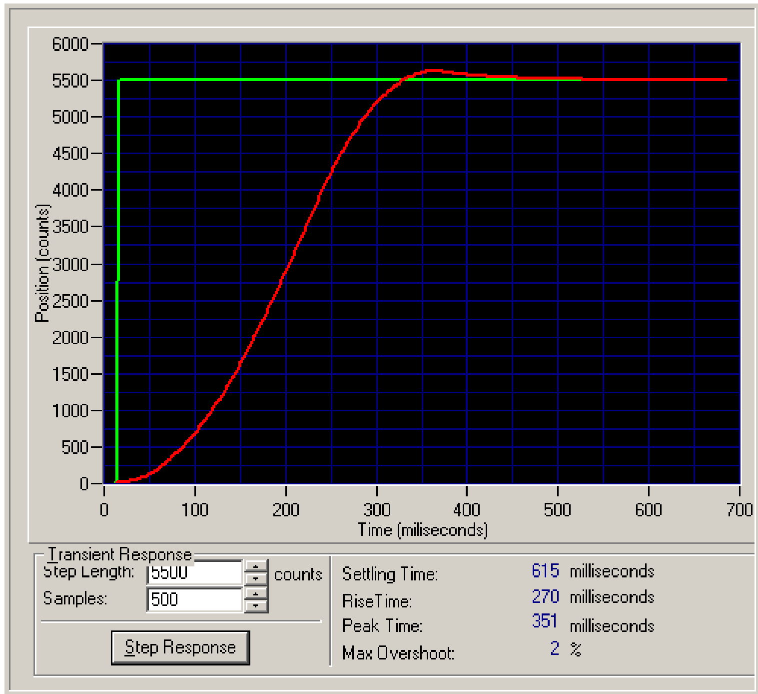


Figure B-5. Throttle actuation from 0 to full throttle.

Appendix C
LabVIEW Evaluation

The Virginia Tech Grand Challenge team was created as a senior design project for the Mechanical Engineering department. As a result, the team was comprised of 30 undergraduate student, 2 grad students, and 2 faculty advisors. The majority of the team members were Mechanical Engineering majors with little robotics experience. Several decisions were driven by the lack of expertise in such areas as embedded systems and software design. Instead of using commonly used programming languages C++ and Java, the team decided to use a graphical software development tool called LabVIEW (Laboratory Virtual Instrument Engineering Workbench). All the software mentioned in this paper was developed using National Instruments LabVIEW 7.1.

LabVIEW was initially designed to provide engineers and scientists with little programming experience with intuitive tools to acquire and analyze data. After 20 years of development, LabVIEW provides the flexibility of a programming language with specialized tools for data acquisition and instrument control. LabVIEW graphically represents the flow of data in a block diagram. It allows the user to manipulate and view the data through an intuitive graphical user interface with familiar switches, dials, and displays. For novice programmers, LabVIEW provides higher-level control for common tasks. For more experienced programmers, LabVIEW delivers the performance, flexibility, and compatibility of a traditional programming language such as C.

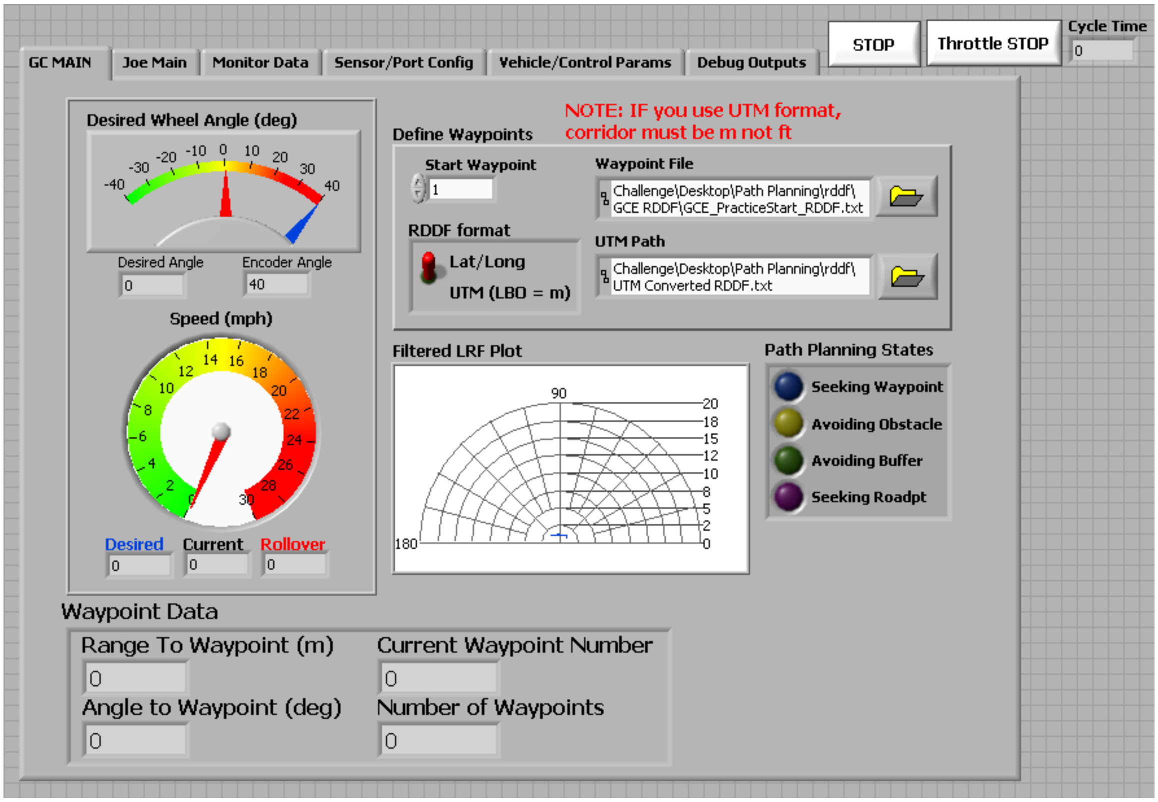


Figure C-1. Example of a LabVIEW graphical interface of the DEZ software

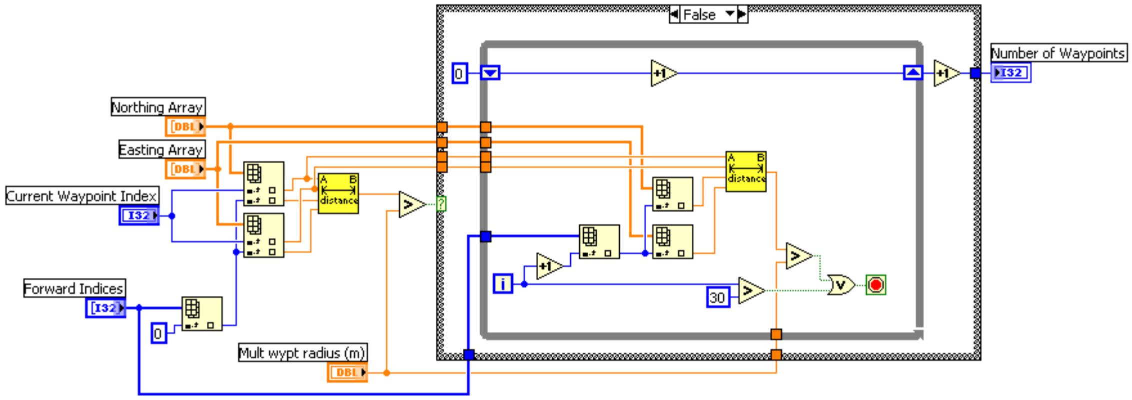


Figure C-2. Example of a LabVIEW block diagram

Appendix D

Fuzzy Logic Example

Fuzzy Fluid Level Control Example:

This appendix is an example of how Fuzzy Logic can be used for controlling the fluid level in a tank. The Fuzzy Logic controller outputs a control effort than adjusts the inlet flow rate to make the fluid level reach the desired level. The Fuzzy variables are Fluid Level Error and Fluid Level Error Rate. A Max-Min composition operation and a Weighted Average Defuzzification method is used in this example [16].

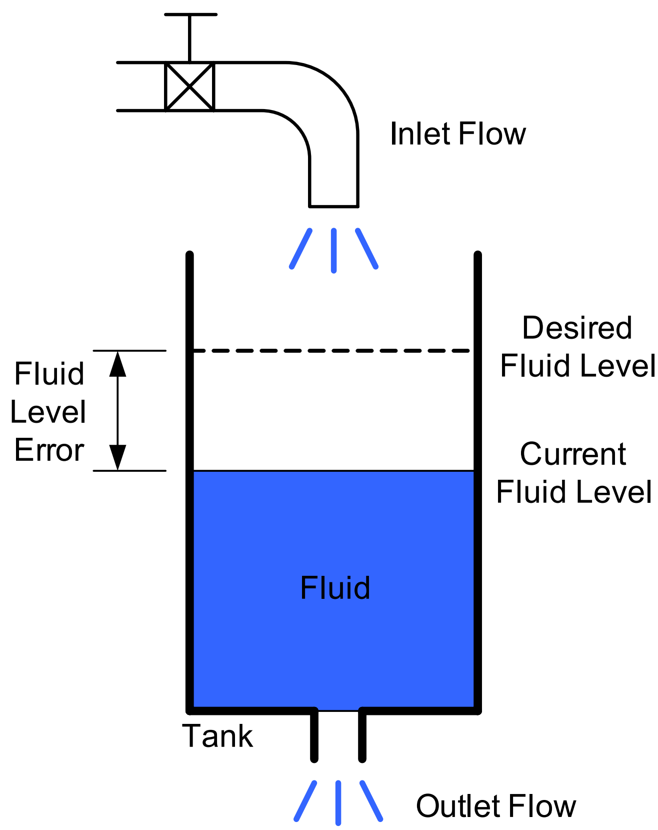


Figure D-1. Fluid level Fuzzy Logic control problem.

Initial Conditions:
Fluid Level Error = 1.5 m
Fluid Level Error Rate = -1 m/min

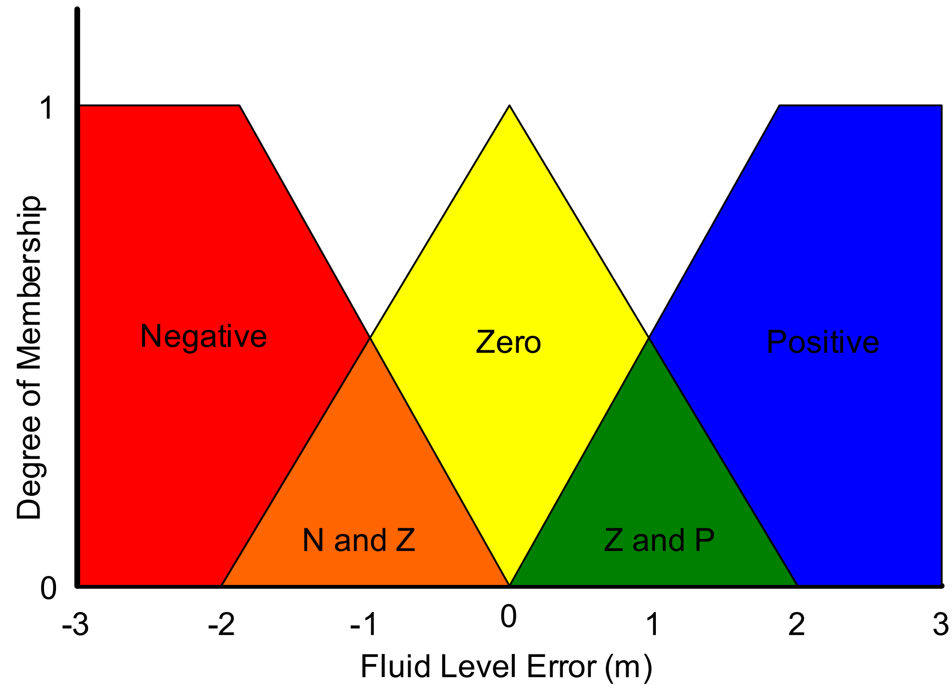


Figure D-2. Fluid Level Error Fuzzy variable.

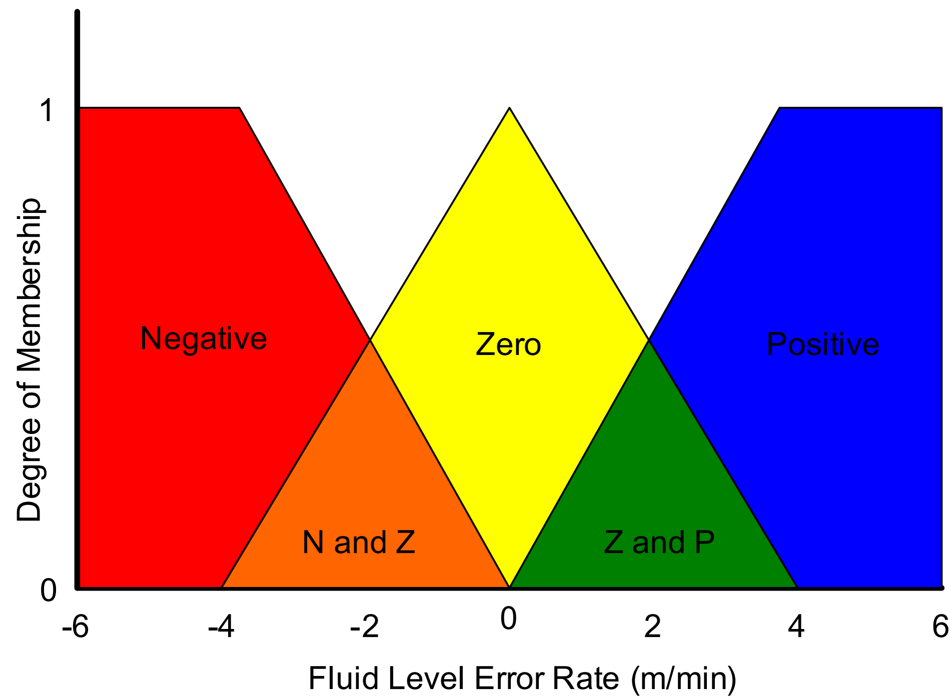


Figure D-3. Fluid Level Error Rate Fuzzy variable.

Fuzzy Rule Base		Fluid Level Error: Negative	Fluid Level Error: Zero	Fluid Level Error: Positive
Fluid Level Error Rate	Negative	Rule 1: High	Rule 2: Medium	Rule 3: Medium
Fluid Level Error Rate	Zero	Rule 4: High	Rule 5: Medium	Rule 6: Low
Fluid Level Error Rate	Positive	Rule 7: High	Rule 8: Low	Rule 9: Low

Figure D-4. Fluid Level Fuzzy Rule Base.

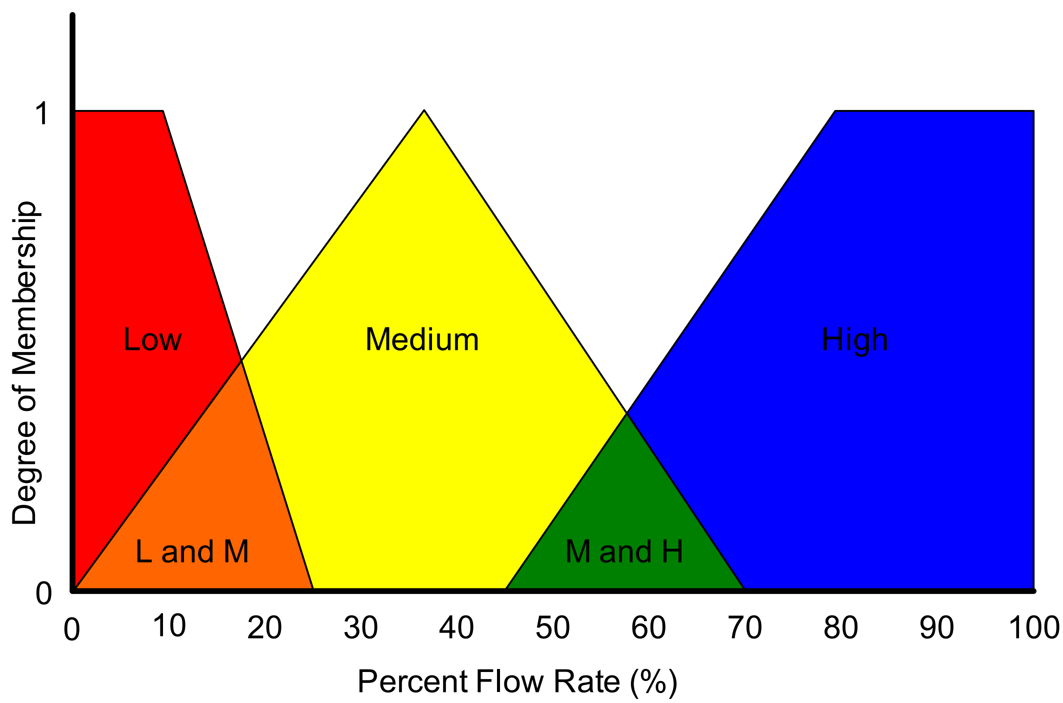


Figure D-5. Fuzzy Logic control output.

Input Degree of Membership

Fluid Level Error = 1.5 m

Negative = 0
Zero = 0.25
Positive = 0.75

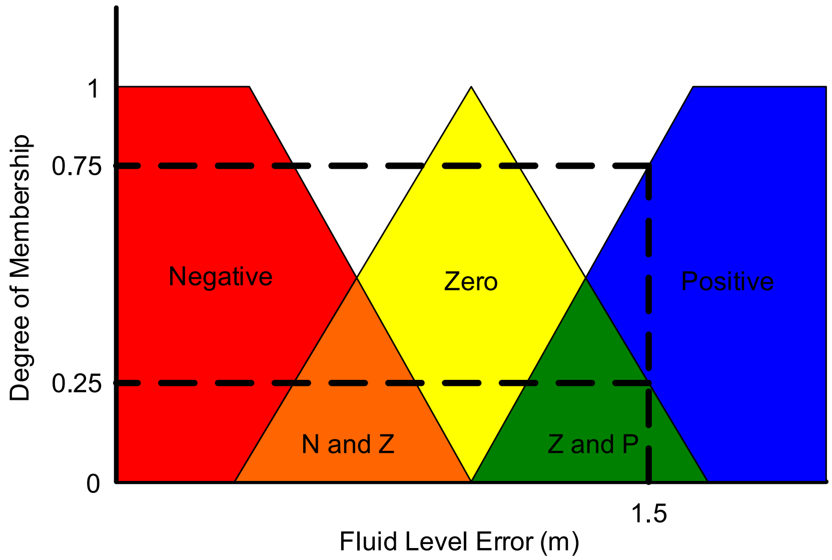


Figure D-6. Fluid level error degree of membership.

Fluid Level Error Rate = -1 m/min

Negative = 0.75
Zero = 0.25
Positive = 0

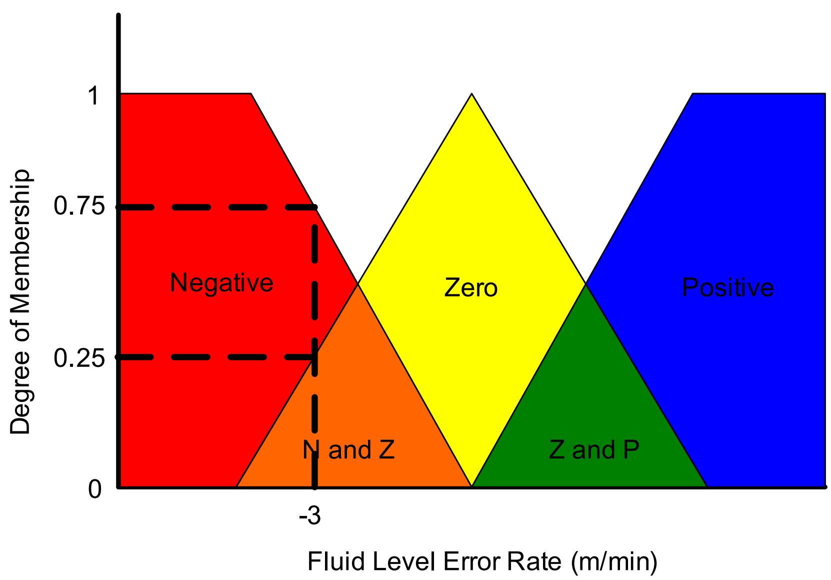


Figure D-7. Fluid level error rate degree of membership.

Fuzzy Rule Base (Min-Max Inference Method)

Rule 1) If Error = Negative(0) AND Rate = Negative(0.75)
THAN Flow = High[min(0 & 0.75)] = **High(0)**

Rule 2) If Error = Zero(0.25) AND Rate = Negative(0.75)
THAN Flow = Medium[min(0.25 & 0.75)] = **Medium(0.25)**

Rule 3) If Error = Positive(0.75) AND Rate = Negative(0.75)
THAN Flow = Medium[min(0.75 & 0.75)] = **Medium(0.75)**

Rule 4) If Error = Negative(0) AND Rate = Zero(0.25)
THAN Flow = High[min(0 & 0.25)] = **High(0)**

Rule 5) If Error = Zero(0.25) AND Rate = Zero(0.25)
THAN Flow = Medium[min(0.25 & 0.25)] = **Medium(0.25)**

Rule 6) If Error = Positive(0.75) AND Rate = Zero(0.25)
THAN Flow = Low[min(0.75 & 0.25)] = **Low(0.25)**

Rule 7) If Error = Negative(0) AND Rate = Positive(0)
THAN Flow = High[min(0 & 0)] = **High(0)**

Rule 8) If Error = Zero(0.25) AND Rate = Positive(0)
THAN Flow = Low[min(0.25 & 0)] = **Low(0)**

Rule 9) If Error = Positive(0.75) AND Rate = Positive(0)
THAN Flow = Low[min(0.75 & 0)] = **Low(0)**

Maximum of each output membership:
Max(Low(0.25) & Low(0) & Low(0)) = **Low(0.25)**
Max(Medium(0.25) & Medium(0.75) & Medium(0.25)) = **Medium(0.75)**
Max(High(0) & High(0) & High(0)) = **High(0)**

<u>Defuzzification (Weighted Average Method)</u>

$$z^* = \frac{\sum \mu(z) \bullet z}{\sum \mu(z)}$$

where z* = crisp output after defuzzification

$\mu(z)$ = strength of each output member

z = centroid of the "symmetrical" member

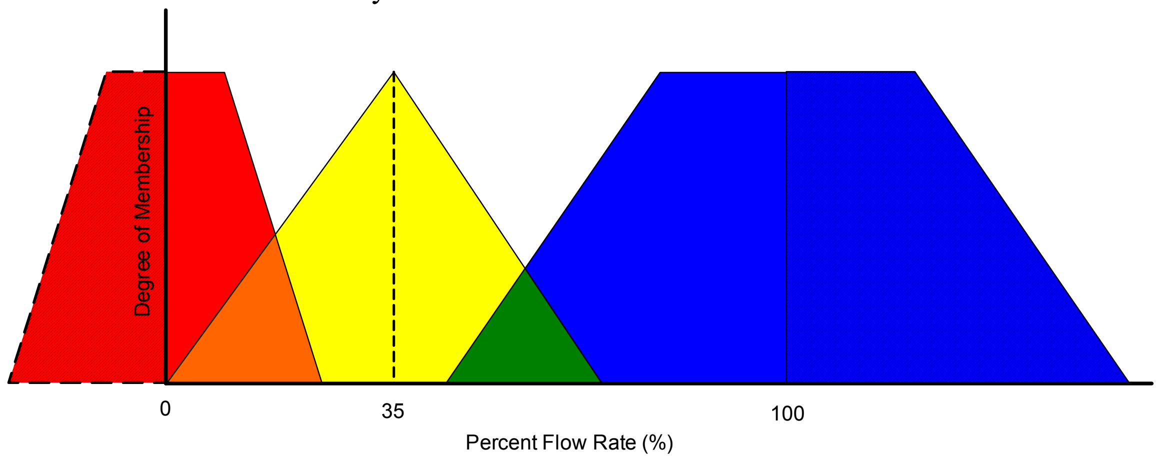


Figure D-8. Weighted Average Method.

Centroids:
Low = 0
Medium = 35
High = 100

Strengths:
Low = 0.25
Medium = 0.75
High = 0

Crisp Output:

$$z^* = \text{Percent Output} = \frac{0.25(0) + 0.75(35) + 0(100)}{0.25 + 0.75 + 0} = 26.25\%$$

<u>Percent Flow Rate = 26.25%</u>

Appendix E

Road Following Fuzzy Parameters

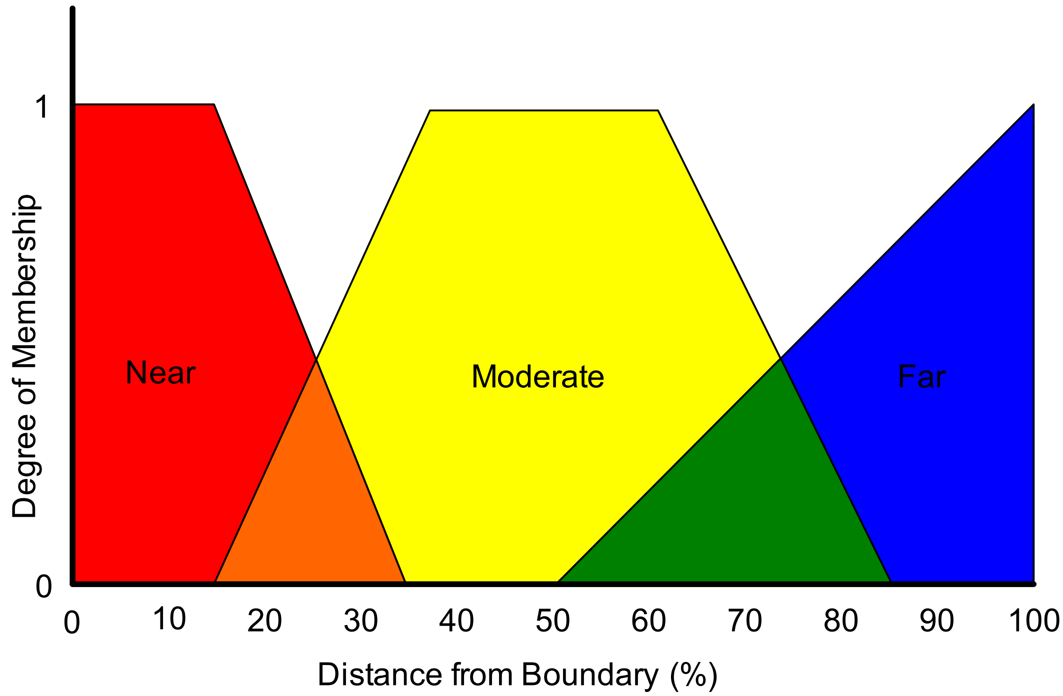


Figure E-1. Distance from Boundary Fuzzy input variable.

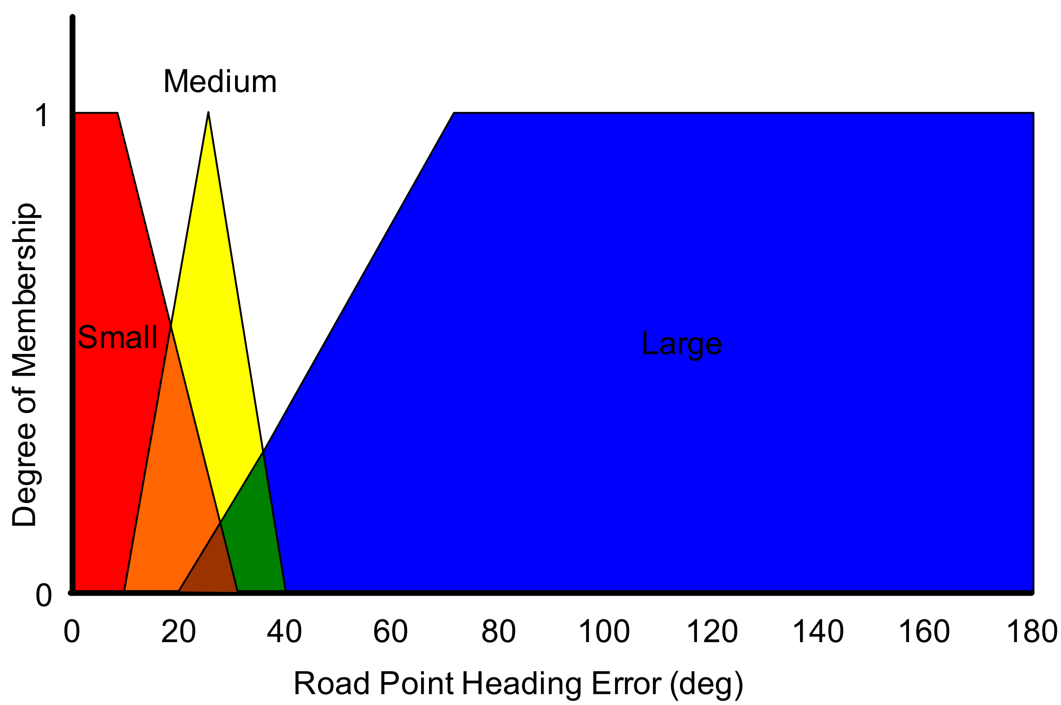


Figure E-2. Road Point Heading Error Fuzzy input variable.

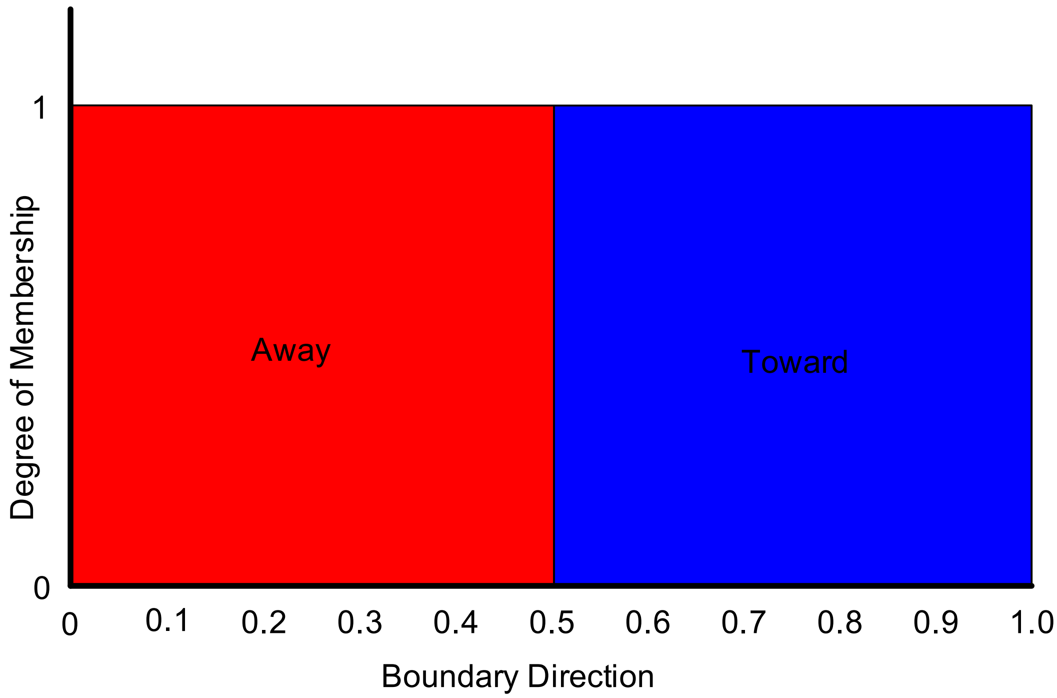


Figure E-3. Boundary Direction Fuzzy input variable.

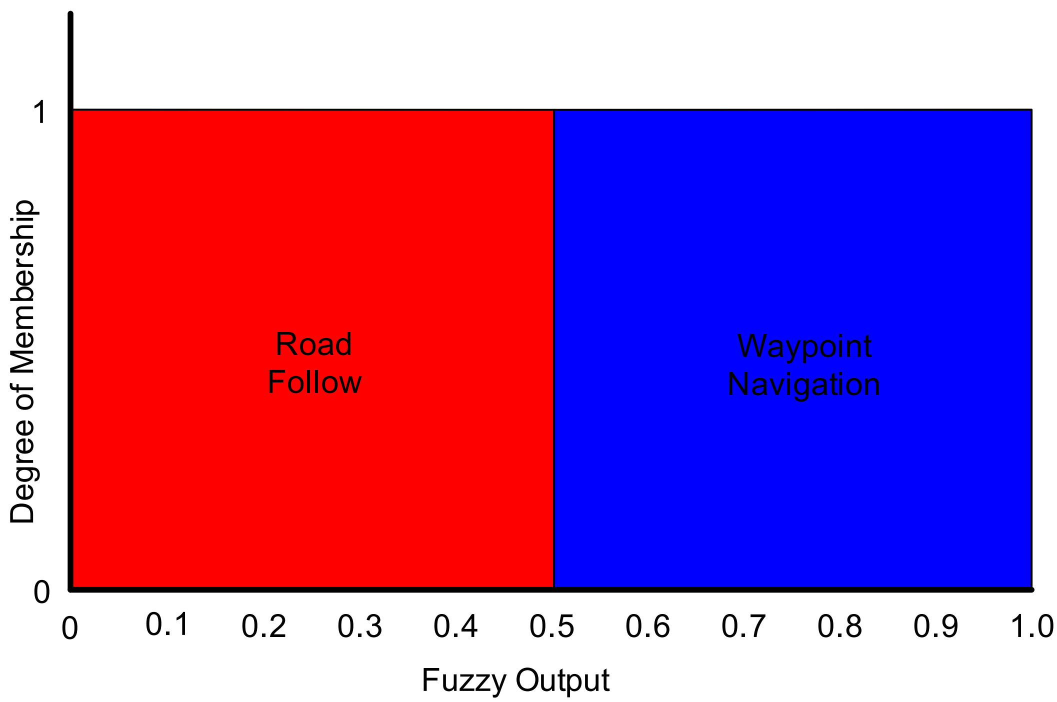


Figure E-4. Road Following Fuzzy output.

Table E-1. Road Following Rule Base

IF			THEN
Bound Distance	Angle to Waypoint	Boundary Direction	Output
near	small	Away	Waypoint
near	small	Center	Waypoint
near	small	Toward	Road
near	medium	Away	Waypoint
near	medium	Center	Waypoint
near	medium	Toward	Road
near	large	Away	Waypoint
near	large	Center	Waypoint
near	large	Toward	Waypoint
moderate	small	Away	Road
moderate	small	Center	Road
moderate	small	Toward	Road
moderate	medium	Away	Waypoint
moderate	medium	Center	Waypoint
moderate	medium	Toward	Road
moderate	large	Away	Waypoint
moderate	large	Center	Waypoint
moderate	large	Toward	Waypoint
far	small	Away	Road
far	small	Center	Road
far	small	Toward	Road
far	medium	Away	Road
far	medium	Center	Road
far	medium	Toward	Road
far	large	Away	Waypoint
far	large	Center	Waypoint
far	large	Toward	Waypoint

Appendix F

Obstacle Avoidance Fuzzy Parameters

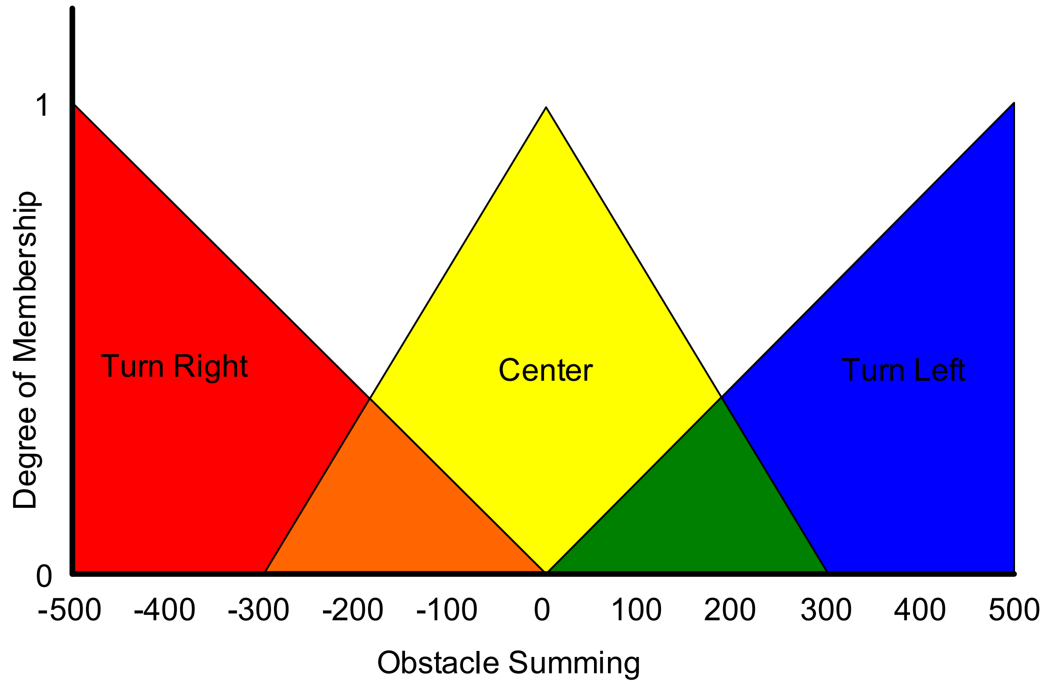


Figure F-1. Obstacle Summing Fuzzy input variable.

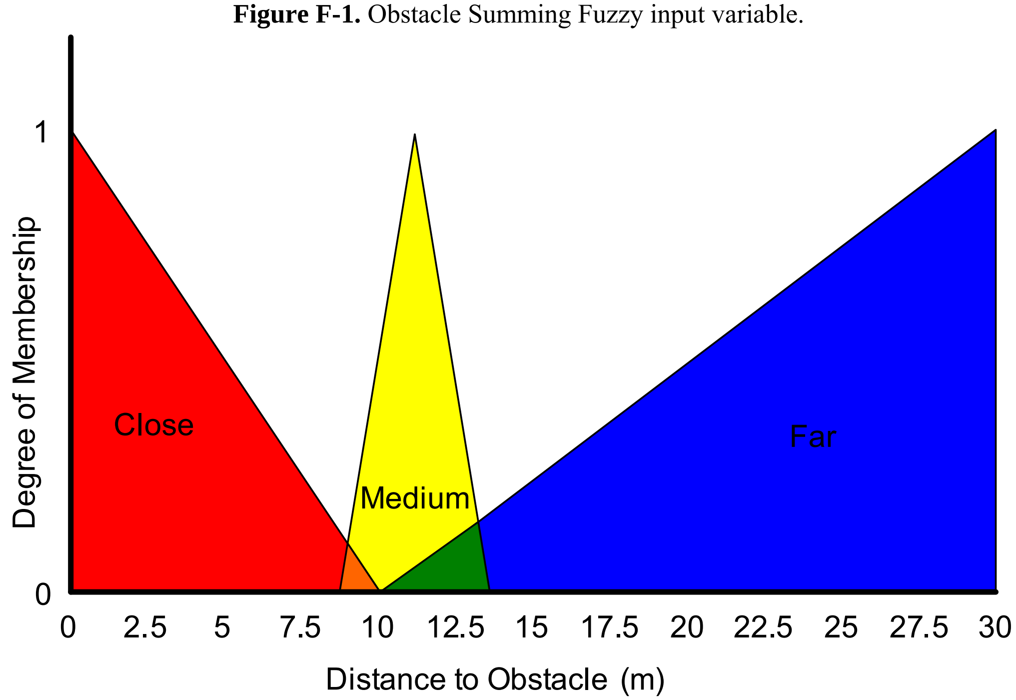


Figure F-2. Distance to Obstacle Fuzzy input variable.

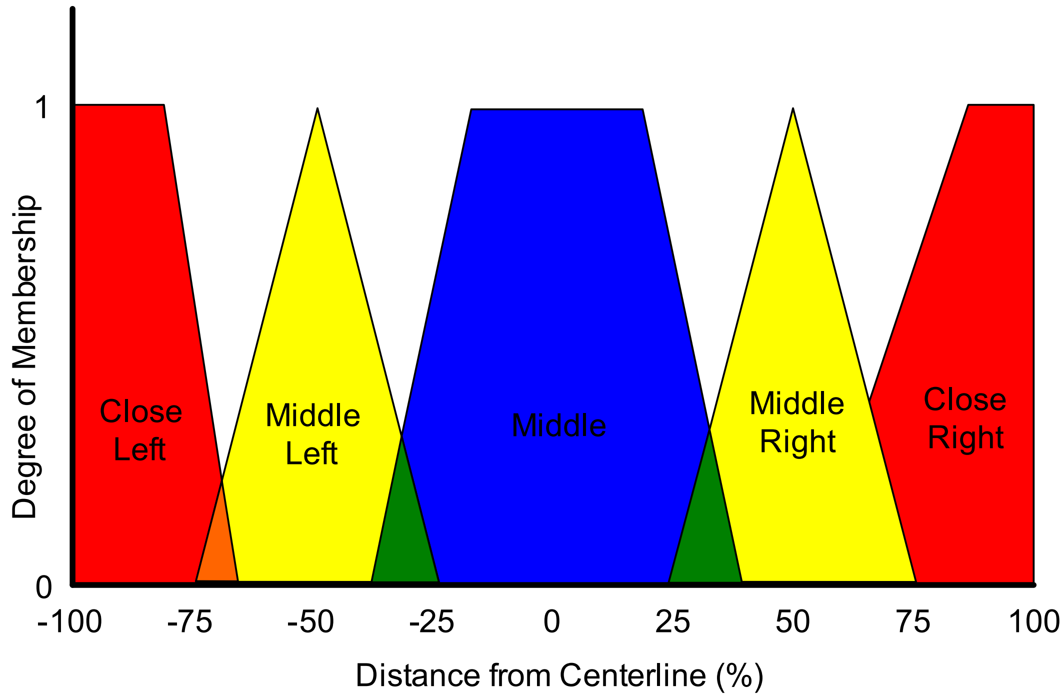


Figure F-3. Distance from Centerline Fuzzy input variable.

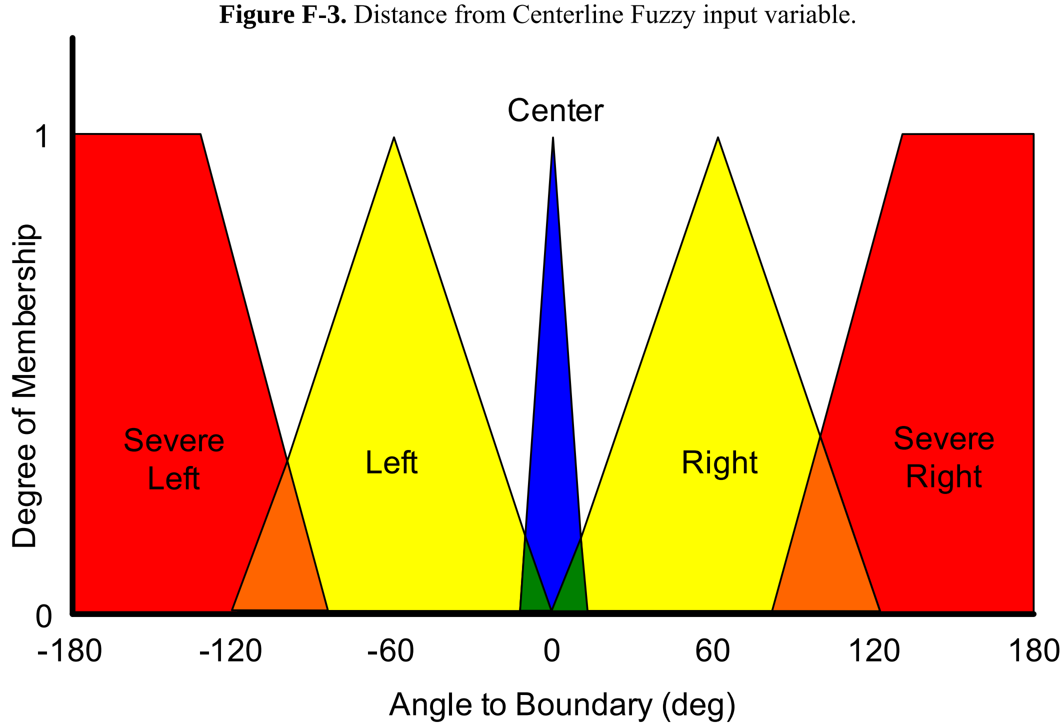


Figure F-4. Angle to Boundary Fuzzy input variable.

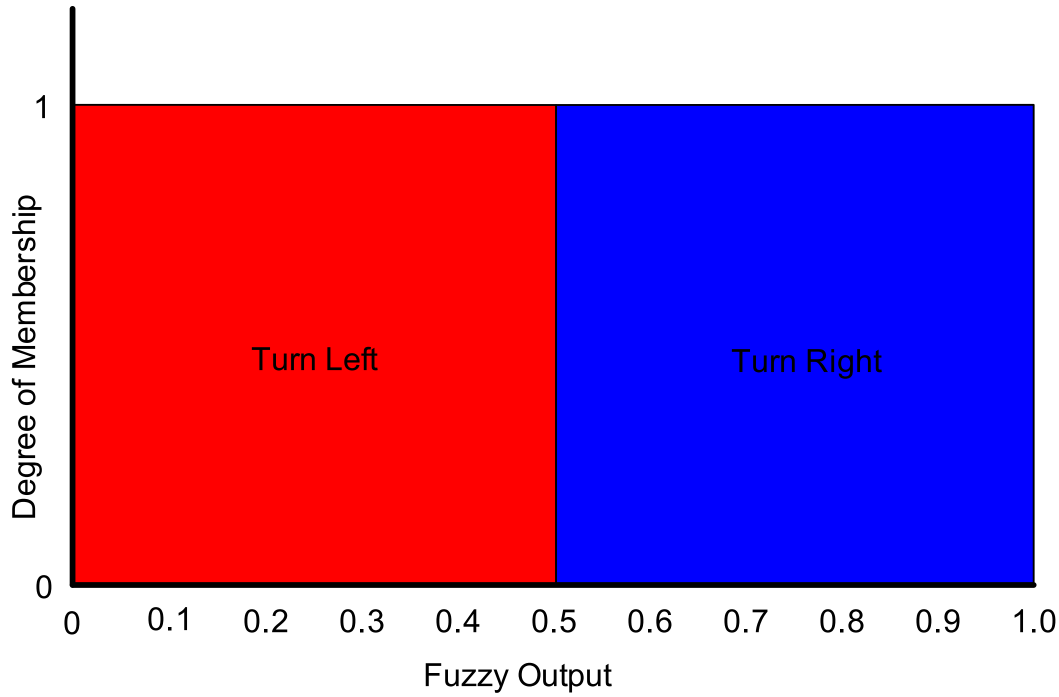


Figure F-5. Obstacle Avoidance Fuzzy output.

Table F-1. Obstacle Avoidance Rule Base

IF				THEN
Sum	% Distance	Boundary Angle	Obstacle Distance	OUTPUT
right	Close L	Severe R	Close	Left
right	Close L	Severe R	Medium	Left
right	Close L	Severe R	Far	Left
right	Close L	R	Close	Right
right	Close L	R	Medium	Right
right	Close L	R	Far	Left
right	Close L	Center	Close	Right
right	Close L	Center	Medium	Right
right	Close L	Center	Far	Right
right	Close L	L	Close	Right
right	Close L	L	Medium	Right
right	Close L	L	Far	Right
right	Close L	Severe L	Close	Left
right	Close L	Severe L	Medium	Left
right	Close L	Severe L	Far	Left
right	Med L	Severe R	Close	Left
right	Med L	Severe R	Medium	Left
right	Med L	Severe R	Far	Left
right	Med L	R	Close	Right
right	Med L	R	Medium	Left

right	Med L	R	Far	Left
right	Med L	Center	Close	Right
right	Med L	Center	Medium	Right
right	Med L	Center	Far	Right
right	Med L	L	Close	Right
right	Med L	L	Medium	Right
right	Med L	L	Far	Right
right	Med L	Severe L	Close	Left
right	Med L	Severe L	Medium	Left
right	Med L	Severe L	Far	Left
right	Middle	Severe R	Close	Left
right	Middle	Severe R	Medium	Left
right	Middle	Severe R	Far	Left
right	Middle	R	Close	Right
right	Middle	R	Medium	Left
right	Middle	R	Far	Left
right	Middle	Center	Close	Right
right	Middle	Center	Medium	Right
right	Middle	Center	Far	Right
right	Middle	L	Close	Right
right	Middle	L	Medium	Right
right	Middle	L	Far	Right
right	Middle	Severe L	Close	Right
right	Middle	Severe L	Medium	Right
right	Middle	Severe L	Far	Right
right	Med R	Severe R	Close	Right
right	Med R	Severe R	Medium	Right
right	Med R	Severe R	Far	Right
right	Med R	R	Close	Right
right	Med R	R	Medium	Left
right	Med R	R	Far	Left
right	Med R	Center	Close	Right
right	Med R	Center	Medium	Right
right	Med R	Center	Far	Left
right	Med R	L	Close	Right
right	Med R	L	Medium	Right
right	Med R	L	Far	Right
right	Med R	Severe L	Close	Right
right	Med R	Severe L	Medium	Right
right	Med R	Severe L	Far	Right
right	Close R	Severe R	Close	Right
right	Close R	Severe R	Medium	Right
right	Close R	Severe R	Far	Right
right	Close R	R	Close	Left
right	Close R	R	Medium	Left
right	Close R	R	Far	Left
right	Close R	Center	Close	Left
right	Close R	Center	Medium	Left

right	Close R	Center	Far	Left
right	Close R	L	Close	Right
right	Close R	L	Medium	Right
right	Close R	L	Far	Right
right	Close R	Severe L	Close	Right
right	Close R	Severe L	Medium	Right
right	Close R	Severe L	Far	Right
center	Close L	Severe R	Close	Left
center	Close L	Severe R	Medium	Left
center	Close L	Severe R	Far	Left
center	Close L	R	Close	Right
center	Close L	R	Medium	Right
center	Close L	R	Far	Left
center	Close L	Center	Close	Right
center	Close L	Center	Medium	Right
center	Close L	Center	Far	Right
center	Close L	L	Close	Right
center	Close L	L	Medium	Right
center	Close L	L	Far	Right
center	Close L	Severe L	Close	Left
center	Close L	Severe L	Medium	Left
center	Close L	Severe L	Far	Left
center	Med L	Severe R	Close	Left
center	Med L	Severe R	Medium	Left
center	Med L	Severe R	Far	Left
center	Med L	R	Close	Right
center	Med L	R	Medium	Left
center	Med L	R	Far	Left
center	Med L	Center	Close	Right
center	Med L	Center	Medium	Right
center	Med L	Center	Far	Right
center	Med L	L	Close	Right
center	Med L	L	Medium	Right
center	Med L	L	Far	Right
center	Med L	Severe L	Close	Left
center	Med L	Severe L	Medium	Left
center	Med L	Severe L	Far	Left
center	Middle	Severe R	Close	Left
center	Middle	Severe R	Medium	Left
center	Middle	Severe R	Far	Left
center	Middle	R	Close	Left
center	Middle	R	Medium	Left
center	Middle	R	Far	Left
center	Middle	Center	Close	Left
center	Middle	Center	Medium	Left
center	Middle	Center	Far	Left
center	Middle	L	Close	Right
center	Middle	L	Medium	Right

center	Middle	L	Far	Right
center	Middle	Severe L	Close	Right
center	Middle	Severe L	Medium	Right
center	Middle	Severe L	Far	Right
center	Med R	Severe R	Close	Right
center	Med R	Severe R	Medium	Right
center	Med R	Severe R	Far	Right
center	Med R	R	Close	Left
center	Med R	R	Medium	Left
center	Med R	R	Far	Left
center	Med R	Center	Close	Left
center	Med R	Center	Medium	Left
center	Med R	Center	Far	Left
center	Med R	L	Close	Left
center	Med R	L	Medium	Right
center	Med R	L	Far	Right
center	Med R	Severe L	Close	Right
center	Med R	Severe L	Medium	Right
center	Med R	Severe L	Far	Right
center	Close R	Severe R	Close	Right
center	Close R	Severe R	Medium	Right
center	Close R	Severe R	Far	Right
center	Close R	R	Close	Left
center	Close R	R	Medium	Left
center	Close R	R	Far	Left
center	Close R	Center	Close	Left
center	Close R	Center	Medium	Left
center	Close R	Center	Far	Left
center	Close R	L	Close	Left
center	Close R	L	Medium	Left
center	Close R	L	Far	Right
center	Close R	Severe L	Close	Right
center	Close R	Severe L	Medium	Right
center	Close R	Severe L	Far	Right
left	Close L	Severe R	Close	Left
left	Close L	Severe R	Medium	Left
left	Close L	Severe R	Far	Left
left	Close L	R	Close	Left
left	Close L	R	Medium	Left
left	Close L	R	Far	Left
left	Close L	Center	Close	Right
left	Close L	Center	Medium	Right
left	Close L	Center	Far	Right
left	Close L	L	Close	Right
left	Close L	L	Medium	Right
left	Close L	L	Far	Right
left	Close L	Severe L	Close	Left
left	Close L	Severe L	Medium	Left

left	Close L	Severe L	Far	Left
left	Med L	Severe R	Close	Left
left	Med L	Severe R	Medium	Left
left	Med L	Severe R	Far	Left
left	Med L	R	Close	Left
left	Med L	R	Medium	Left
left	Med L	R	Far	Left
left	Med L	Center	Close	Left
left	Med L	Center	Medium	Left
left	Med L	Center	Far	Right
left	Med L	L	Close	Left
left	Med L	L	Medium	Right
left	Med L	L	Far	Right
left	Med L	Severe L	Close	Left
left	Med L	Severe L	Medium	Left
left	Med L	Severe L	Far	Left
left	Middle	Severe R	Close	Left
left	Middle	Severe R	Medium	Left
left	Middle	Severe R	Far	Left
left	Middle	R	Close	Left
left	Middle	R	Medium	Left
left	Middle	R	Far	Left
left	Middle	Center	Close	Left
left	Middle	Center	Medium	Left
left	Middle	Center	Far	Left
left	Middle	L	Close	Left
left	Middle	L	Medium	Right
left	Middle	L	Far	Right
left	Middle	Severe L	Close	Right
left	Middle	Severe L	Medium	Right
left	Middle	Severe L	Far	Right
left	Med R	Severe R	Close	Right
left	Med R	Severe R	Medium	Right
left	Med R	Severe R	Far	Right
left	Med R	R	Close	Left
left	Med R	R	Medium	Left
left	Med R	R	Far	Left
left	Med R	Center	Close	Left
left	Med R	Center	Medium	Left
left	Med R	Center	Far	Left
left	Med R	L	Close	Left
left	Med R	L	Medium	Right
left	Med R	L	Far	Right
left	Med R	Severe L	Close	Right
left	Med R	Severe L	Medium	Right
left	Med R	Severe L	Far	Right
left	Close R	Severe R	Close	Right
left	Close R	Severe R	Medium	Right

left	Close R	Severe R	Far	Right
left	Close R	R	Close	Left
left	Close R	R	Medium	Left
left	Close R	R	Far	Left
left	Close R	Center	Close	Left
left	Close R	Center	Medium	Left
left	Close R	Center	Far	Left
left	Close R	L	Close	Left
left	Close R	L	Medium	Left
left	Close R	L	Far	Right
left	Close R	Severe L	Close	Right
left	Close R	Severe L	Medium	Right
left	Close R	Severe L	Far	Right