5

Virginia Tech's Twin Contenders: A Comparative Study of Reactive and Deliberative Navigation

Brett M. Leedy, Joseph S. Putney, Cheryl Bauman, Stephen Cacciola, J. Michael Webster, and Charles F. Reinholtz

Unmanned Systems Group, Virginia Tech, Blacksburg, Virginia 24060

Summary. There are two commonly accepted paradigms for organizing intelligence in robotic vehicles, namely, reactive and deliberative. Although these paradigms are well known to researchers, there are few published examples directly comparing their development and application on similar vehicles operating in similar environments. Virginia Tech's participation, with two nearly identical vehicles in the DARPA Grand Challenge, afforded a practical opportunity for such a case study. The two Virginia Tech vehicles, Cliff and Rocky, proved capable of off-road navigation, including road following and obstacle avoidance in complex desert terrain. Under the conditions of our testing, the reactive paradigm developed for Cliff produced smoother paths and proved to be more reliable than the deliberative paradigm developed for Rocky. The deliberative method shows great promise for planning feasible paths through complex environments, but it proved unnecessarily complex for the desert road navigation problem posed by the Grand Challenge. This case study, while limited to two specific software implementations, may help to shed additional light on the tradeoffs and performance of competing approaches to machine intelligence.

5.1 Introduction

The 2005 DARPA Grand Challenge was a 132 mile race of autonomous ground vehicles through the Mojave Desert. Virginia Tech produced two off-road autonomous vehicles (Figure 5.1) to compete for the $2 million prize. From an initial field of 195 teams, both Virginia Tech vehicles passed a series of qualifying events and ultimately qualified for the main Grand Challenge Event, along with 21 other teams. Although they were built on two similar base vehicle platforms, one vehicle was developed using a reactive paradigm, while the other vehicle was developed using a deliberative navigation paradigm (Murphy, 2000). These competing strategies were developed and evaluated independently for the Challenge. This paper discusses the strategy, capability, and performance of both of the Virginia Tech Grand Challenge entries.

M. Buehler, K. Iagnemma, and S. Singh (Eds.): DARPA'05, STAR 36, pp. 155–182, 2007.
springerlink.com

Fig. 5.1. Virginia Tech's entries to the 2005 DARPA Grand Challenge, Cliff (left) and Rocky (right)

5.2 Base Platform

Both Virginia Tech Grand Challenge vehicles were initially designed as interchangeable platforms on which to develop two very different navigation strategies. However, the terrain mapping for Rocky's deliberative path planning required additional terrain mapping LADAR units, resulting in some hardware differences between the two vehicles. This section includes the details of the base vehicles, power system, drive-by-wire conversion, and network architecture (Leedy, 2006).

5.2.1 Base Vehicle

The Virginia Tech Grand Challenge base vehicles are Club Car XRT 1500s, utility vehicles produced by Ingersoll-Rand. This base platform may seem like an unlikely choice for a desert race due to its diminutive size, but it has proven to be a capable off-road vehicle. The XRT 1500 is extremely agile with a turning radius of 3.5 m. The vehicle also provides a top speed of 40 km per hour and a minimum ground clearance of 16.5 cm under the rear differential skid plate. Stock vehicle weight is 567 kg with the capability of carrying a 454 kg payload.

Cliff, a redesign of Virginia Tech's entry to the 2004 DARPA Grand Challenge, is built on a prototype XRT 1500 which had not yet gone into production at the time of the vehicle's donation. An aircooled 20 horse power (hp) Honda GX620 gasoline engine supplies power to the drive train. The vehicle's roll cage was customized to provide protection for electronic equipment located in the payload area as well as mounting locations for vision and laser sensors.

Rocky's platform is also a Club Car XRT 1500. However, Rocky is a production vehicle powered by a Kubota D722 20 hp liquid-cooled diesel engine. The roll cage on Rocky was replaced with a custom built cage constructed of thicker wall tubing. Rocky also makes use of Club Car's optional heavy-duty suspension upgrade.

5.2.2 Drive-by-Wire Conversion

To enable full computer control of the vehicle actuation systems, the throttle, brake, and steering were converted to drive by wire. The drive-by-wire systems on Cliff and Rocky are nearly identical. Both vehicles actuate the throttle using

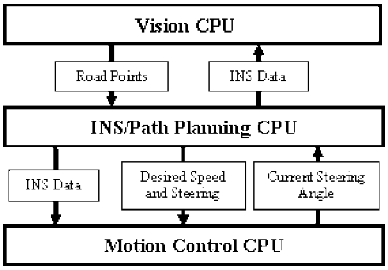

Fig. 5.2. System-level data flow diagram for Cliff

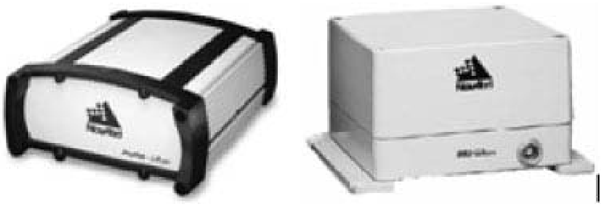

Fig. 5.3. Novatel Propak LBplus positioning system. This system consists of a Novatel Propak LBplus GPS receiver (left) and a Honeywell HG1700 IMU in a Novatel IMUG2 enclosure (right).

a dc gear motor with integrated encoder feedback. The throttle cable is wrapped around a pulley mounted to the output shaft of the motor.

The steering wheel and column were removed from the vehicles to make space for the drive-by-wire system. Both vehicles use right-angle gear motors fitted with quadrature encoders to actuate the steering. The output shaft of the right-angle gear head is directly coupled to the input shaft of the stock steering rack with a chain coupling.

The drive-by-wire braking is accomplished by replacing the master cylinder and brake pedal assembly with an electronically controlled hydraulic pump. The braking system uses a Hydrastar HBA-16 actuator from Carlisle Industrial Brake. Operator control of this electrohydraulic brake is accomplished using a Teleflex-Morse ECFP electric foot pedal connected to the vehicle's Motion Control computer.

In addition to the drive-by-wire brake, a manual/emergency brake can be applied by the onboard operator or emergency stop system. The stock parking brake consists of a pedal assembly that pulls a steel cable directly connected to the rear brake calipers. After removing the ratcheting mechanism, the modified parking brake assembly is used as a manual auxiliary brake. This brake is also automatically activated in the event of a power loss to the vehicle, such as a DARPA "disable" emergency stop condition. To use this braking system as an autoengaging safety brake, an air tank and air actuated piston were installed. Air hold-off pressure must be applied to the unit to release the brake. In the event of vehicle power loss, the brakes are engaged when a solenoid valve opens the air passage and allows the hold-off air to escape from the air-actuated piston. Once power has been restored to the system, the reserve tank recharges the piston and disengages the brake.

5.2.3 Computing Architecture

Both vehicles are equipped with National Instruments PXI-8176 controllers. These controllers are high-performance compact personal computers containing Pentium processors with up to 1 GB of random access memory. The controllers can run at speeds ranging from 1.2 to 2.6 GHz. Due to their high shock resistance, the PXI computers are rigidly mounted to the electronics enclosure without additional shock isolation. The Windows XP operating system provides a familiar visual user interface.

The three computers on Cliff each perform a specific task: Vision, INS/Path Planning, and Motion Control. The INS/Path Planning computer determines Cliff's current position and target location while monitoring for obstacles in front of the vehicle. The Vision computer uses monocular vision to look for roads in the vehicle's field of view and stereovision to localize points along the road. The information from the Vision computer is passed to the INS/Path Planning computer, which determines the appropriate behavior for perceived orientation and surroundings. The Motion Control computer executes speed and steering commands from Path Planning by handling the closed-loop control of all vehicle actuators. Figure 5.2 illustrates the computing architecture of Cliff's three computers.

Rocky uses the same basic architecture as Cliff except with four computers: Vision, Path Planning, INS/Local Mapping, and Motion Control. The INS/Local Mapping computer creates a map of the perceived obstacles and terrain. The Vision computer passes a map of perceived roads to the Path Planning computer. The Path Planning computer then determines the optimal path to take through the surrounding area using data from local map and road map data. These decisions are passed to the Motion Control computer and handled exactly as on Cliff.

5.3 Sensors

Cliff and Rocky both share the same basic sensor suite: GPS/INS for positioning, horizontal LADAR for obstacle detection, and stereovision for road following. Rocky is also equipped with a set of two downward-looking scanning LADAR units for local terrain mapping. These LADAR units are used only with the map-based deliberative scheme to provide terrain information and to detect negative obstacles, such as holes and ditches. This section describes in detail the physical arrangement, purpose, and data format of the main exteroceptive sensor components.

5.3.1 Positioning

Both vehicles used Novatel Propak LBplus positioning systems (Figure 5.3) in the Grand Challenge and in the comparative studies described later in this paper. The system consists of a Novatel Propak LBplus GPS receiver and a Novatel IMUG2 enclosure housing a Honeywell HG1700 inertial measurement unit (IMU). The Propak LBplus unit provides singlepoint position accuracy of 1.5 m CEP. As previously mentioned, this accuracy is increased to 10 cm CEP by L-band differential corrections through the subscription service, OmniSTAR. The position, velocity, and heading from the Propak LBplus are collected at the maximum output rate of 20 Hz. In the event that the global positioning system (GPS) signal becomes occluded, the inertial measurements from the IMU take over seamlessly to provide position and heading.

5.3.2 Obstacle Detection

A single SICK LADAR is used to detect obstacles by scanning a horizontal plane in front of Cliff and Rocky. Anything detected by this scanner is marked as an obstacle. Unfortunately, this includes false obstacles, such as hills and other nonobstacle objects that may pass in front of the scanner. The unit is mounted to the front of each vehicle directly below the brush guard approximately 0.38 m above the ground. By angling the sensor up approximately 1.5° from horizontal, problems related to false hill detection are minimized. The serial data output of the LADAR returns a near-instantaneous twodimensional (2D) polar coordinate array of the range and angle to any solid objects in the sensor's viewing plane. Only the most recent scan from the LADAR is used for both navigation strategies.

5.3.3 Vision

The monocular/stereovision system allows Cliff and Rocky to perceive roads ahead of the vehicle and mark them as preferred areas of travel. The vision system examines the monocular image of the scene, extracts areas that look like roads, then finds the relative position of the road areas using the camera's stereo capabilities. Finally, it passes the road information to the path planning computer.

For the Grand Challenge Event, it was assumed that most of the competition course would follow desert service paths and that these roads would be less likely to contain obstacles than the surrounding terrain. For this reason, a vision system was designed to identify roads and adjust the path of the vehicle to be down the center of the road. A Point Grey Bumblebee stereovision camera, mounted to the top center of the vehicle's roll cage, is used to observe the area in front of the vehicle. Each of the Bumblebee's cameras is capable of outputting progressively scanned 640×480 stereoimages at 30 Hz. The stereoimage processing algorithm operates at approximately 5 Hz.

Before each image is processed, it undergoes a number of modifications to reduce processing time. The image resolution is reduced to 160×120 pixels to lessen the number of recognition operations required on each frame. The image is also converted from the red-green-blue (RGB) image representation to hue-saturation-luminosity representation. The HSL representation allows simpler color definitions in a variety of lighting conditions.

To further reduce the required processing per image, each frame is run through a k-clustering (Green, Yao & Zhang, 1998) algorithm that separates every pixel in each color plane into eight categories defined by the center of the cluster. This clustering method reduces the colors in an image from 16 million to only 512. Instead of using rigidly defined color windows, this method reduces the colors using dynamic logical segments. This operation ensures that each pixel will be converted into colors that are as close to the original color as possible. The reduced-color image is still stored as a 24-bit image to preserve the original color differences.

After simplifying the image, the software searches for the basic geometric characteristics that define a road. This search is done using only one of the 2D images provided by the stereocameras. The software determines if a uniform texture and color pattern form a shape close to that of a desert road (Rasmussen, 2004). Once a road is found, its color patterns are logged to a color look-up table. This table correlates a specific color to a road certainty value.

For each subsequent 2D image frame, every pixel is compared to the color look-up table to create a confidence value for that pixel. This confidence value represents the level of certainty that the specific pixel is part of a road. To simplify further processing, the pixels corresponding to the lowest 30% of the confidence values are removed. Pixels that are not adjacent to or near high confidence value pixels are also removed. Using morphological techniques to connect and separate the areas of an image, the software attempts to create a single area of the image that may be a road (Rasmussen, 2004). The suspected road area must be confirmed as a road by checking its vertical narrowing, edge continuity, and size relative to the image.

If an area of the 2D image is determined to be a road, the image is sent to the stereo-processing software where it is combined with its synchronous image to convert the pixel locations to coordinates in the vehicle reference frame. These coordinates are rechecked to ensure that there are no discontinuities, and that they lie on the same plane as the vehicle. If confirmed, the road points are transformed into UTM global coordinates and sent to the Path Planning computer to be used in the navigation algorithm. The color look-up table for the confirmed road is averaged with the previous ten tables to form the new table for future iterations. To ensure that the look-up table contains enough values to operate accurately, five successful iterations must occur before road points are confirmed and sent to the Path Planning computer.

5.4 Motion Control and Vehicle Safety

Although the two vehicles were designed to use different navigation algorithms and sensor configurations, the drive-by-wire systems were designed to be the same on both vehicles. This standardization allows any navigational algorithm to be implemented on either vehicle, as long as it is compatible with the standard interface. The Motion Control system provides the necessary software to turn desired steering and speed commands into vehicle movement. The motion of both Cliff and Rocky is controlled by three actuators: Steering motor, throttle motor, and brake actuator. The desired commands are generated by the Path Planning software in autonomous operation or a human driver in manual mode.

5.4.1 Speed Control

The vehicle acceleration is controlled by a closed-loop manipulation of an electric gear motor attached to the vehicle's throttle. Since the throttle motor cannot slow the vehicle, a parallel brake control is needed for controlling the speed of the vehicle. The braking system uses an open-loop control to translate a desired reduction in speed to the appropriate brake percent command for the hydraulic brake driver. If the commanded speed is greater than the current speed, a proportional integral differential (PID) control loop handles throttle inputs. If the commanded speed is less than the current speed, brakes are applied based on the commanded urgency of deceleration. Even though the brake and throttle control the speed in parallel, the vehicle will never attempt to increase throttle when braking.

5.4.2 Rollover Prevention and Vehicle Safety

After experiencing two vehicle rollovers, one during Rocky's DARPA site visit, attention was focused on preventing another rollover. A simple dynamic model of the vehicle, that considers gravity and centripetal force, was developed. The basis for this model is shown in Figure 5.4. To account for the rollover effects of unpredictable terrain, a factor of safety is implemented in each calculation.

A rollover condition exists when the resultant of the centripetal force and the weight vector point outside the footprint of the vehicle. Stability can be achieved by slowing the vehicle's forward velocity and reducing the magnitude of the steering angle.

Since the stability calculation depends on velocity and steering angle feedback, the stability calculation described above is not foolproof. Rocky's rollover in qualifying was due to LABVIEW DataSocket communication failures between the INS computer and Motion Control computer. When the failure occurred, the Motion Control computer falsely perceived a zero-speed value from the GPS/INS DataSocket. When the vehicle attempted to accelerate to the commanded speed, the GPS/INS feedback speed remained zero. As a result, the PID controller continued to apply full throttle, and the vehicle rolled in its first turn. The

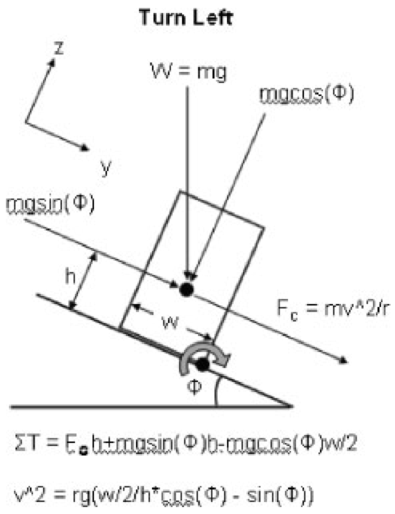

Fig. 5.4. Model of the vehicle on a side slope and in a turn

team replaced the less reliable communication with simple UDP messages. In addition, Motion Control monitors GPS/INS data for communication failures, ensuring that the data are being updated on every iteration. The software will pause the vehicle if a failure occurs, which prevents it from driving without speed feedback. Similar safety systems monitor other potential failures, such as problems with steering.

5.5 Navigation Strategies

An important objective in developing the two Virginia Tech Grand Challenge vehicles was to compare the reactive navigation strategy used on Cliff with the deliberative path planning strategy used on Rocky. These two approaches are usually considered to be opposite ends of the spectrum of navigation strategies (Murphy, 2000). Both approaches were given equal attention during the design and development phase in preparation for the competition. This section provides an explanation of each of the two navigation strategies.

5.5.1 Reactive Navigation with Dynamic Expanding Zones (DEZ)

Waypoint navigation, road following, and obstacle avoidance on Cliff all use a reactive scheme (Murphy, 2000). Reactive algorithms only use the most recent sensor information to make navigational decisions. A technique called the Dynamic Expanding Zones algorithm was developed by Virginia Tech as the main obstacle avoidance strategy for the reactive approach. A set of zones around the vehicle dictate the behavior that the vehicle will exhibit. If obstacles are not detected within these zones, the vehicle will proceed with waypoint/road following.

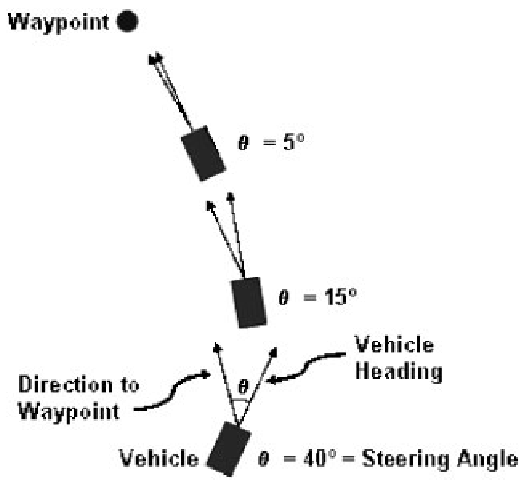

Fig. 5.5. An illustration of the vehicle's commanded steering angle converging toward zero during waypoint navigation

Otherwise, the vehicle will take appropriate action to avoid the obstacles. These zones vary in size and shape, depending on vehicle speed, steering, and sensor status.

5.5.1.1 Waypoint Navigation

A critical component of the 2005 Grand Challenge Event was successful navigation through globally defined waypoints. As with all decision-making software on Cliff, waypoint navigation does not generate a planned path to reach a desired waypoint. Instead, an instantaneous steering angle, equal to the difference between the current heading and direction to the waypoint, is commanded (Figure 5.5). This waypoint navigation strategy acts as a closed-loop feedback control that requires very little computation to calculate the commanded steering angle.

The vehicle reaches a desired waypoint when the vehicle enters a radius defined by the distance between the corridor intersection point and waypoint (Figure 5.6). Whether the vehicle is saving time or avoiding an obstacle, the waypoint radius eliminates the need to travel directly over the waypoint. Traveling over a waypoint was not required for the Grand Challenge Event, as long as the vehicle stayed within the lateral boundary offset (LBO).

5.5.1.2 Road Following

Since roads are generally easier to traverse and have fewer obstacles than unstructured desert terrain, road following is a desirable behavior. If a road exists that leads the vehicle in the general direction of the waypoint, the vehicle ignores waypoint navigation to follow the road. Road data are received from the vision

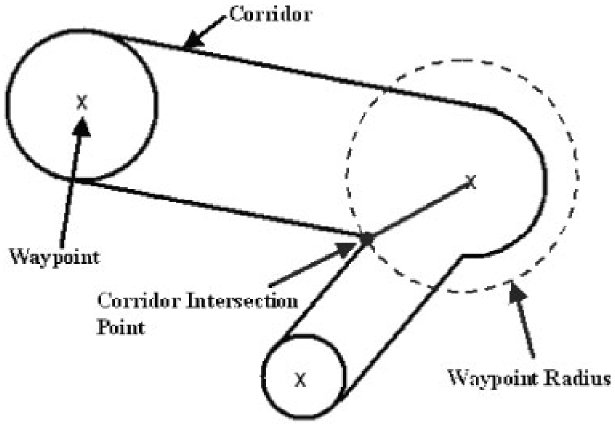

Fig. 5.6. A waypoint radius is created using the LBO of the intersecting corridors

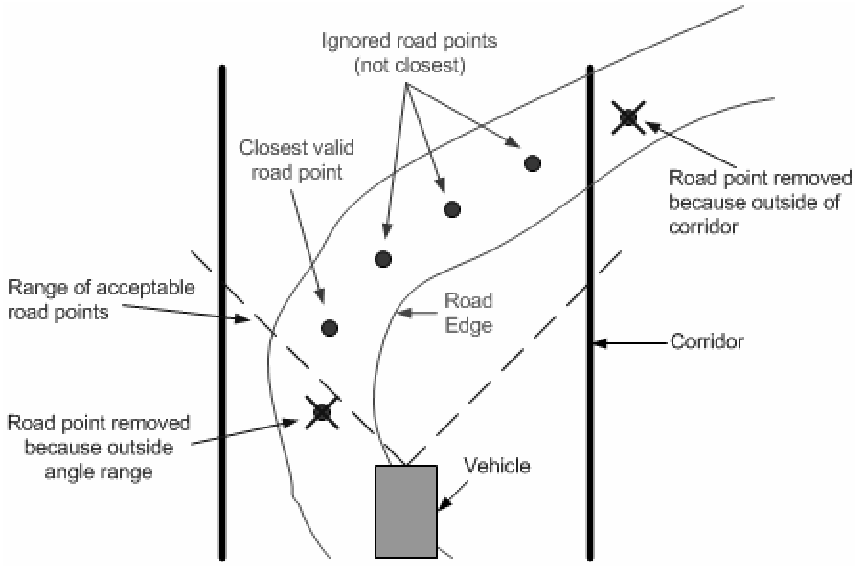

Fig. 5.7. An illustration of how a road point is selected from the road point array

computer as an array of perceived road center points. Points outside of the LBO and points not within 30° of the current heading are ignored, and the closest valid road point is chosen to be the desired road point (Figure 5.7).

Using this desired road point, fuzzy logic control is used to determine if the road point is in the general direction of the desired waypoints. Fuzzy logic is able to substitute numerical variables with linguistic variables to solve ill-defined problems (Zadeh, 1965, 1973). For example, if the vehicle is heading *somewhat*

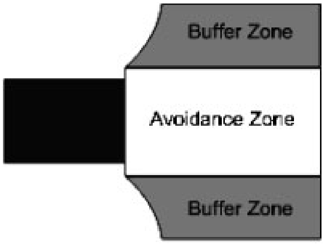

Fig. 5.8. Dynamic Expanding Zone layout

toward the waypoint and away from the corridor boundary, fuzzy control will determine that road following is appropriate. If road following is desired, Cliff steers toward the road point in the same manner as waypoint navigation.

5.5.1.3 Obstacle Avoidance

If a perceived obstacle prevents the vehicle from driving directly to a waypoint or following a road, Cliff ignores the waypoint navigation and road following behaviors to avoid the obstacle. A reactive obstacle avoidance approach, called Dynamic Expanding Zones, has been developed for robust obstacle avoidance. This algorithm is not limited to a specific sensor configuration, and can use any type of instantaneous obstacle map with Boolean elements (obstacle or no obstacle).

5.5.1.3.1 Obstacle Zones. The Dynamic Expanding Zones algorithm uses two zones to determine the avoidance behavior when an obstacle is present (Figure 5.8). The avoidance zone is located directly in front of the vehicle (Reynolds, 1999). If an obstacle is in the avoidance zone, the vehicle must avoid it to continue safely toward the desired waypoint. This zone has a constant width, slightly larger than the width of the vehicle, which prevents the vehicle from clipping the sides of obstacles. The length of the avoidance zone expands dynamically, hence the name Dynamic Expanding Zones. Dynamic Expanding Zones commands a steering angle and speed to avoid any obstacles in this zone.

The second zone, the buffer zone, is adjacent to and of the same length as the avoidance zone. The purpose of the buffer zone is to prevent the vehicle from turning into an obstacle. It also eliminates oscillatory behavior between waypoint/road following and obstacle avoidance. For example, if the vehicle attempts to make a turn to the left when there is an obstacle in the left buffer, Dynamic Expanding Zones will override the turn command. The vehicle will drive straight forward, until the obstacle exits the buffer zone. Once both the avoidance and buffer zones are clear, waypoint/road following will resume.

5.5.1.3.2 Dynamic Expanding Capability. The length of the avoidance and width of the buffer zone are the key factors in the success of this obstacle avoidance algorithm. The length and width of the zones are adjusted based on the current driving conditions. For example, the avoidance zone is shortening when the vehicle is turning. This keeps it from unnecessarily trying to avoid obstacles that are straight ahead. It is possible for the vehicle to make an unnecessary maneuver to avoid an obstacle if the buffers are too wide. For these reasons, the size of the avoidance and buffer zones are dynamically modified to optimize navigation for different situations.

The length of the avoidance zone is controlled by the projected clothoid path and the speed of the vehicle (Shin & Singh, 1990). Using the current steering angle and steering velocity (assumed to be constant), the corresponding clothoid path is calculated. The avoidance zone shrinks to the most distant intersection of this path with the avoidance zone. This means that the length of the avoidance zone shrinks as the steering angle increases. This length control prevents the vehicle from reacting to obstacles too far ahead, but ensures that the vehicle will have ample time to respond as it approaches an obstacle. In addition, as the vehicle increases its speed, the avoidance zone length must also increase to react to obstacles in the distance (Putney, 2006).

Similar to the avoidance zone, the buffer is also dynamically controlled based on the steering angle. Both buffer zones widen symmetrically as the steering angle increases. A larger steering angle requires the vehicle to look for obstacles farther away in the lateral direction. Again, this zone expansion ensures that the vehicle will only avoid the necessary obstacles.

5.5.1.3.3 Commanded Steering Control. Similar to road following, the steering direction is determined by fuzzy logic control. The controller intelligently decides a steering direction, which is optimal for both avoiding an obstacle and staying on course. The fuzzy input variables include distance to obstacles and obstacle summing, discussed below. When a collision with an obstacle is imminent, the Dynamic Expanding Zones method uses only obstacle summing to choose a safe steering direction. For example, if an obstacle is located on the left side of the avoidance zone near the vehicle, the safest steering direction is to the right. On the other hand, when the obstacle is farther ahead, the vehicle has more decision flexibility. As a result, the vehicle can choose a direction that will avoid the obstacle while keeping the vehicle within the boundaries.

Obstacle summing allows Dynamic Expanding Zones to decide which direction is optimal given current obstacle data. An obstacle window is a defined area of interest that encompasses known obstacles in front of the vehicle. The height and width of the obstacle window is defined by the fixed lengths for lateral and length expansion. Only obstacles detected in this window are considered in the obstacle summing calculation. This window allows the vehicle to respond to multiple objects in close proximity instead of just the closest detected obstacle. Using the obstacle window, a value is determined by summing the distances from each of the obstacles to the centerline of the vehicle. Figure 5.9 illustrates how obstacle summing would work if the obstacle window contained two obstacles. The negative values, left of the centerline, represent the wall obstacle (sensed as three points by the laser scanner); while the positive value represents the round

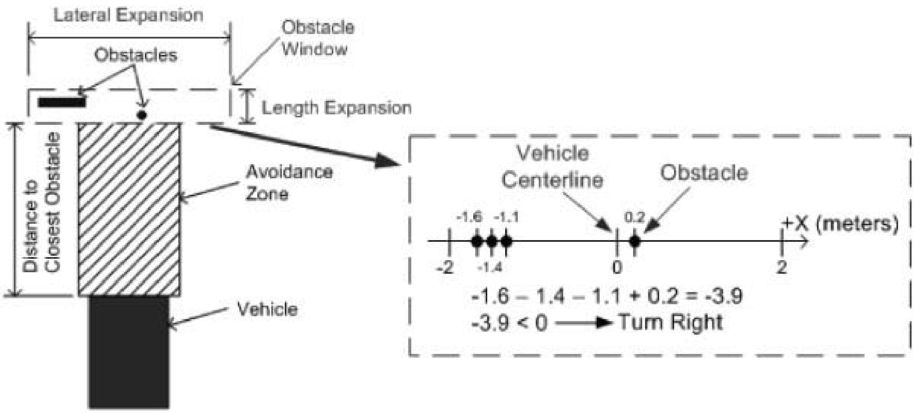

Fig. 5.9. Example calculation for determining a steering direction based on obstacle location

obstacle to the right. This example results in a negative obstacle sum; therefore, a right turn requires a smaller steering maneuver (Putney, 2006).

The magnitude of the commanded steering angle is calculated using the distance to the closest obstacle within the avoidance zone. The steering angle calculation is not an attempt to model the vehicle's actual projected path. Dynamic Expanding Zones varies this approximate path and steering angle with each iteration until the obstacle is avoided. This steering angle calculation eliminates the need for accurate path calculations on each update, which can be computationally expensive. As a result, Dynamic Expanding Zones requires minimal processing power when compared to many deliberative approaches (Putney, 2006).

Once an obstacle has left the peripheral view of the LADAR, it is no longer considered by the reactive strategy. This has proven to be a safe assumption, since only the most extreme maneuvers of the vehicle would cause it to turn back into an obstacle it has already seen without seeing it again.

5.5.1.3.4 Commanded Speed Control. Speed control is critical for properly avoiding obstacles, staying within boundaries, and preventing rollover. The vehicle will always attempt to run at its top speed. However, to prevent rollovers, the speed is limited when the vehicle executes a turn. Though speed control follows the reactive approach, the vehicle can anticipate future maneuvers and take precautionary action. For example, the vehicle slows down when it detects an obstacle in its avoidance zone. The vehicle also anticipates the turn at a waypoint by slowing to a safe speed before it reaches the waypoint radius.

5.5.2 Deliberative Strategy

The Deliberative NonUniform Terrain Search (NUTS) algorithm, developed for Rocky, uses simultaneous sensor fusion and storage to create a homogeneous local terrain traversibility map. This map contains information on discontinuities in terrain height, course boundaries, and roads recognized by the vision system. The map is scanned by an A* graph search (Hart, Nilsson & Raphael, 1968) to determine the desired future path of the vehicle. This operation iterates in real time at a rate of 16 Hz. The goal of the deliberative Terrain Search path planning strategy is to build a continuously updated best path on which to drive. The benefits include planned reaction to perceived future obstacles and holistic driving decisions based on all sensor data (Leedy, 2006).

5.5.2.1 Terrain Mapping

The Terrain search algorithm uses two types of LADAR data to describe the local terrain and obstacles: A two-and-one-half dimensional geometric terrain map, and a binary obstacle map. The geometric terrain map is built using two ground-scanning LADAR units. These can be seen on Rocky in Figure 5.1. The obstacle map is built using a horizontal scanning, front mounted LADAR.

To detect variations in terrain that might affect the planned path of the vehicle, two groundscanning LADAR systems are employed. Figure 5.10 shows the fields of view of Rocky's sensors.

The ground-scanning LADAR units measure ranges to solid objects in a 100° 2D swath about the z axes of the sensors. On a perfectly flat surface, this would allow the scanner beams to reach the ground at a maximum distance of 15 m in front of the vehicle. As the vehicle approaches an obstacle, the scanners record a higher altitude at the location the scan plane intersects the object. Tall obstacles occlude the LADAR; leaving a "shadow" behind the obstacle, which cannot be

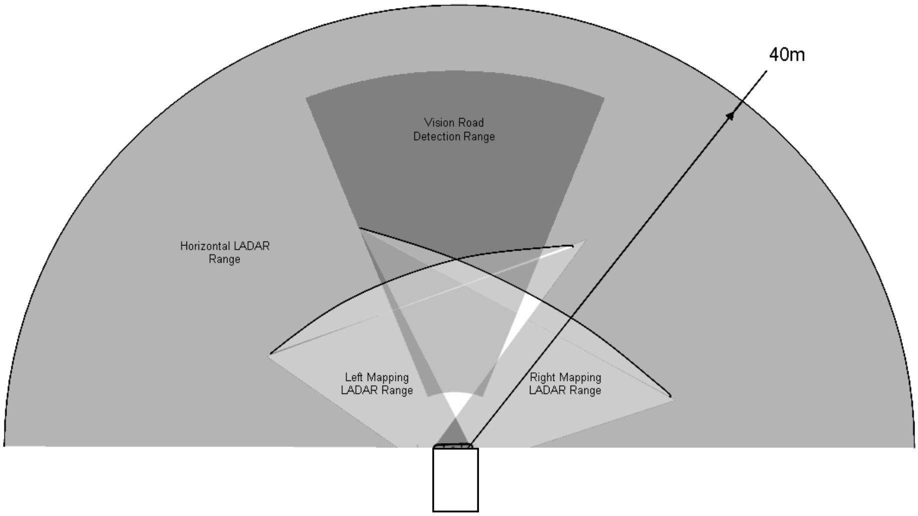

Fig. 5.10. Fields of view of Rocky's ground mapping LADAR, horizontal LADAR, and camera

Fig. 5.11. The image on the left shows a parked car from Rocky's point of view. The image on the right shows scanned terrain colored black. The position and orientation of Rocky is denoted by the grey arrow. A parked car blocks LADAR scans, leaving an unscanned shadow, circled on the right image.

scanned. Figure 5.11 shows the location of scanned points collected as the vehicle approaches car on the left.

The scan planes of the two scanners overlap in front of the vehicle, giving more data directly ahead. This extra information can be used to identify potential obstacles more readily.

Each point collected by the LADAR scanners is transformed from the sensor coordinate frame to the vehicle reference and then rotated into global UTM coordinates. This transformation uses the most recent position, attitude, and heading. The new data are stored as an array of height and position values, then added to the corresponding location in the local map. The new data overwrite any older values stored in the same location. The local map is stored as a 2D array of height values from an arbitrary baseline set at the start of the vehicle's run. The map array is aligned with true north-south and east-west, and represents a 40 m by 40 m field of 20 cm square grid elements. This local terrain map is continuously updated with new LADAR scans at a rate of 16 Hz. As the vehicle moves, the LADAR scanners measure the height of solid objects in their scan plane, and adds these data to the local terrain map, creating a three-dimensional geometric description of the terrain that has been scanned. The data are always represented with the vehicle at a fixed location and varying attitudes and the map grid aligned with the global UTM coordinate frame. This map could readily accept a priori terrain data, if any were available. Figure 5.12 shows a diagram of the vehicle on a local terrain map.

In each program iteration, a 12.5 m by 12.5 m rectangular section of the terrain map is extracted for path planning analysis. This section is transformed back into the vehicle coordinate frame to another grid of 20 cm×20 cm squares. This extracted section is processed using the sigma filter method (Murphy, 2000; Lee, 1984) to find the slope (first derivative) of the perceived terrain. Areas of

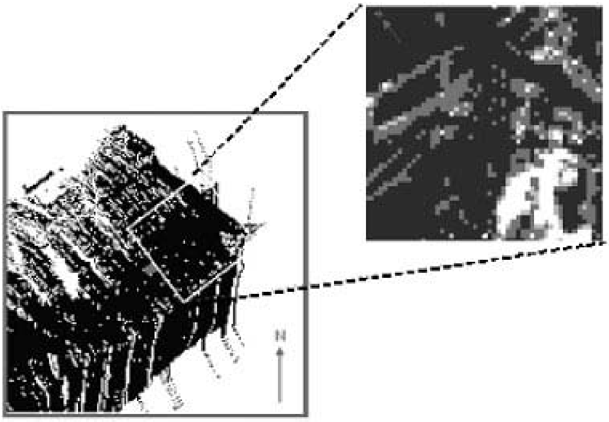

Fig. 5.12. Black areas in the local terrain map (left) indicate scanned points. The local cost map is extracted from this area (right), and sent to the A* decision algorithm. On the terrain map, the vehicle is held at a fixed position on the scrolling globally referenced terrain map, but it may change attitude. The cost map area is held fixed relative to the vehicle.

Fig. 5.13. The local cost map (top-down view, left) generated by the horizontal LADAR for a typical scene with obstacles (vehicle view, right)

high slope are considered to be less passable than areas of little or no slope, so the cost map is scaled by a tuned gain value to return high cost in areas of high slope and low cost in areas of low slope. This map format allows the grid to be searched using standard graph-search algorithms.

Horizontal LADAR scan points, which return a range of less than 40 m, are imported into the local cost map as high-cost obstacles. Any point returned to the scanner is considered impassible, and is marked with an extremely high cost. The data from this sensor are refreshed on every program cycle using only the most recent scan for the cost map. This treatment of the sensor data provides easily interpreted high-cost obstacles wherever the horizontal scanning LADAR detects a solid object. The main drawback of this treatment is the occurrence of "false positives" when the vehicle pitches momentarily or when it approaches a hill. This has the effect of slowing the vehicle speed and causing the vehicle to approach steep hills at an angle. In testing, we found that this rarely created a situation where the vehicle would veer off course. Figure 5.13 shows a local cost map generated by the horizontal scanning LADAR beside a photo of the scene.

Vision Road Mapping – A computer vision road finding algorithm also contributes to the local cost map. The vision processing approach is identical to the one used by the Dynamic Expanding Zones algorithm with data passed as a Boolean map of suspected road points. The scene in front of the vehicle is processed to find points along the road. Figure 5.14 shows one frame of a stereotest scene with the map array of the corresponding road map.

The vision road map is passed to the mapbuilding software as a binary array of the same dimensions as the LADAR local cost maps. If confidence in the vision-recognized road falls below a specific threshold, a blank map is sent to the path planning. When received by the map-building software, all areas marked as road centerline are marked with a lower cost than the surrounding areas. The difference between road and nonroad cost values was tuned through extensive field testing.

5.5.2.2 Deliberative Driving Decision

The NUTS deliberative driving paradigm attempts to drive the optimum path over continuously changing nonuniform terrain perceived by the vehicle. To optimize the path based for both the current and intended future position of the vehicle, the NUTS algorithm computes a new optimum path at each program iteration using an A* graph search. Figure 5.15 is a flow schematic of a single iteration of the NUTS program.

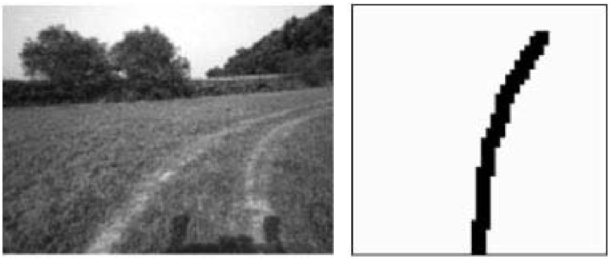

Fig. 5.14. The recognized road from the RGB image (left) is translated into a local road map (right)

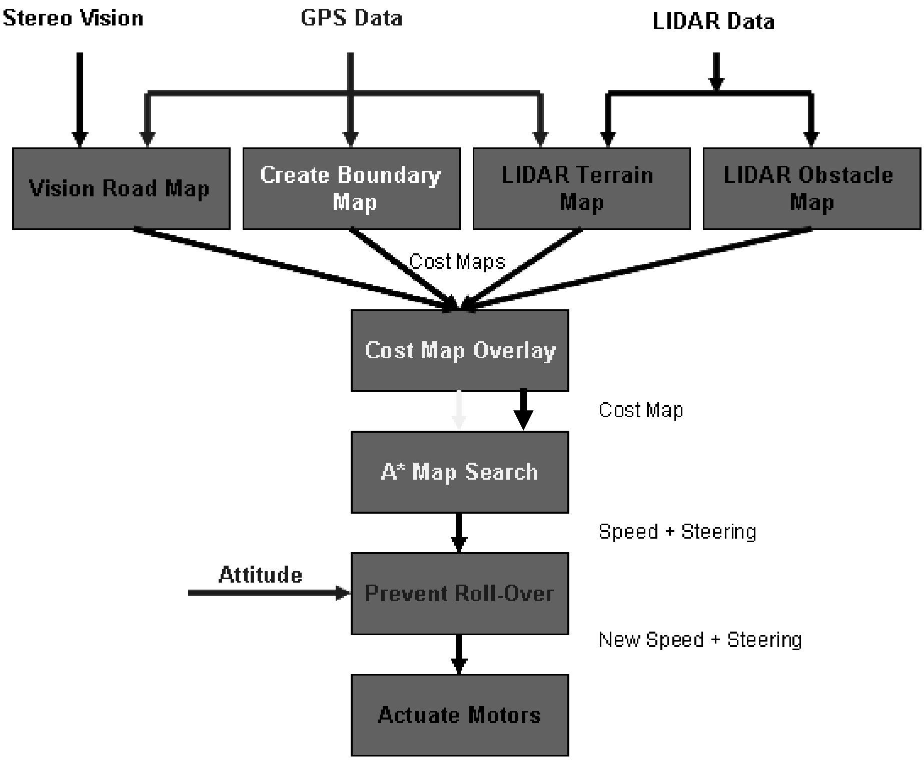

Fig. 5.15. Sensor data for the NUTS algorithm is processed in parallel then combined in the form of a local cost map for the A* search

To generate the final search map, NUTS overlays the four local cost maps generated by the sensor cognition components. Figure 5.16 shows an example of a typical cost map used by NUTS.

Obstacles detected by the horizontal LADAR and areas outside the course boundary are marked with the highest possible value using the obstacle and boundary cost maps. Areas with a geometric change in altitude are assigned a cost based on the "steepness" of the terrain. The road layer adds cost to areas not believed to be a road. The cost overlay values are weighted such that course boundaries and obstacles have the greatest influence over driving decisions. Terrain LADAR and road data are used to guide the vehicle through the optimum path for navigation.

Using the overlaid map, NUTS next attempts to find the best path using an A* least-cost path search (Hart et al., 1968). If the destination waypoint is on the cost map, it is taken as the search goal point. If the destination point is out of the map, NUTS generates a goal point on the border of the map. The path generated by the A* search is passed to the driving component. The driving component generates steering and speed commands based on the vehicle's current pose. This

Fig. 5.16. Typical LADAR overlay map of sensor data for the Virginia Tech Grand Challenge A* graph search (right). Dark shades indicate areas of low cost, while light areas indicate areas of high cost. The area enclosed in the white oval indicated a significant drop off; the area circled in dashes indicates the trees pictured at left.

is accomplished by selecting a point on the path a certain range from the vehicle. The vehicle steers using a pure pursuit algorithm (Coulter, 1992) to head toward the path. Before the steering and speed commands are passed on to the vehicle motion control system, a final check is performed to ensure that the commands are safe.

5.6 Comparative Study

In preparation for the DARPA Grand Challenge, Virginia Tech compared the reactive and deliberative navigation strategies side-by-side. The comparison described in this section attempts to capture the data and lessons learned from applying each navigation strategy. In the future, the team intends to use the results to improve future designs of hybrid paradigms using the best elements of both reactive and deliberative path planning. By implementing both strategies simultaneously with similar developmental teams, this study also sheds light on the nuances of developing and implementing both types of algorithms.

5.6.1 Performance

At the DARPA site visit to Virginia Tech on May 5, 2005, both the reactive and the deliberative algorithms demonstrated their ability to navigate global waypoints, avoid obstacles, and stay within the course boundaries on an off-road obstacle course. This section discusses the differences in performance of the reactive Dynamic Expanding Zones and deliberative NUTS driving algorithms in the areas of waypoint following, driving smoothness/efficiency, obstacle avoidance, and repeatability/reliably.

Fig. 5.17. The lower Plantation Road test field with the RDDF course centerline superimposed

To quantify the ability of each strategy to navigate waypoints, GPS/inertial data were collected during test runs of both vehicles on an open rolling-hill terrain test course set up at the Plantation Road test facility on the Virginia Tech campus (Figure 5.17). These initial tests were run with identical route definition data files (RDDF) paths and with no obstacles to avoid.

Special care was taken to examine the driving algorithms under weather and terrain conditions that were as similar as possible. All test runs were collected on the same day in clear weather with alternating reactive and deliberative runs for measurements at the same speed/weather/lighting combination. All exteroceptive sensors except the high-precision Novatel GPS/INS were shut down on the vehicles. In essence, this test focused on the ability of the algorithm to follow a given path in the absence of obstacles. Both data sets were collected on the same vehicle, using the same sensors and peripheral software. Our goal was to isolate the behavioral differences in the decision-making software.

The RDDF length of this course was 3.173 miles (approximately 5 km) over open field terrain. Each driving algorithm was tested over five laps at maximum commanded speeds of 5, 10, and 15 mph. Position, velocity, actuator state, commanded vehicle state, and the values of many other parameters were collected at a rate of 5 Hz during the tests.

The data, summarized in Table 5.1, clearly show an overall performance edge for the reactive Dynamic Expanding Zones algorithm. Rocky running Dynamic Expanding Zones averaged significantly higher speeds than the same vehicle running the deliberative NUTS path planning. Although the vehicle was fully capable of driving at higher speeds, the rollover safety processes in both algorithms prevented the vehicle from selecting turn/speed combinations that might put it in jeopardy. At 5 and 10 mph, the Dynamic Expanding Zones algorithm averaged speeds near the top speed limit imposed on the software. At 15 mph, however, the serpentine nature of the course triggered safety slowdown procedures for sharp turns, and limited the overall speed.

By attempting to steer a course defined by a square grid, the deliberative method must make more frequent steering adjustments to follow the desired

Table 5.1. Overall performance statistics for non-obstacle avoidance test runs

		Test Run Overall Performance		
Top Set Speed		5 mph	10 mph	15 mph
Reactive (Dynamic Expanding Zones)	Top Speed (mph)	6.6	11.4	15.6
	Average Speed (mph)	4.8	7.2	7.9
	Total Time (s)	2365.2	1568.8	1443.6
Deliberative (Non-Uniform Terrain Search)	Top Speed (mph)	6.6	10.8	14.4
	Average Speed (mph)	3.5	4.0	4.30
	Total Time (s)	3203.6	2829.4	2594.4

path. The overall course performance highlights one main difference between the reactive and deliberative paradigms: decisiveness. The reactive Dynamic Expanding Zones algorithm is less sensitive to subtle changes in the perceived sensor state. It is important for the vehicle to be flexible and react quickly to a dynamic environment when selecting a desired path, but subtle errors in the vehicle position or orientation can cause the deliberative algorithm to reroute the path, which diminishes the overall performance. While this rerouting may be desirable for long-term planning, it does not seem to be desirable for simple waypoint following. As a result, the purely reactive paradigm had better overall performance on the test course.

Unnecessarily using the steering, brake, or throttle actuators consumes energy and may degrade the dynamic performance of the vehicle, or even result in a rollover. This unnecessary actuation may also cause wear on the involved components, such as the brake pads and steering rack. Unnecessary actuation also burdens the vehicle's power system and reduces the vehicles overall efficiency. The steering actuator consumes 373 W at peak power, and the electrohydraulic brakes consume 240 W at peak power. Brake actuation also takes significant kinetic energy from the vehicle, and dissipates it as waste heat at the brake pads. Hence, unnecessary steering and braking can be significant factors in reducing the efficiency of the vehicle.

While collecting the test run data, it appeared that the reactive algorithm was able to steer more smoothly and efficiently on the course. To measure the efficiency and smoothness of steering on the test runs, the percent of the time the vehicle commands a change in steering angle was examined. The percent of time spent changing steering angle is an indicator of the driving algorithm's smoothness and energy efficiency. If the vehicle constantly seeks a new steering position, it uses a large amount of energy to drive the steering motors. Continuous steering actuation also indicates more weaving of the vehicle.

Table 5.2 shows the percentage of time the steering actuator was running for each algorithm on the same course. The reactive algorithm is nearly an order of magnitude more efficient than the deliberative algorithm under the

Table 5.2. Steering and braking actuation percentage (no obstacle avoidance)

		Steering and Braking Actuation		
		5 mph	10 mph	15 mph
Dynamic Expanding Zones	Travel Distance (mi)	3.22	3.17	3.17
	% Time Turning	6.3%	9.4%	10.0%
	% Time Braking	3.6%	10.6%	13.6%
Non-Uniform Terrain Search	Travel Distance (mi)	3.24	3.21	3.21
	% Time Turning	54.4%	56.3%	57.2%
	% Time Braking	5.1%	10.6%	12.9%

same conditions. The likely cause of this wide disparity in steering efficiency can be traced to the means by which each algorithm generates its path. The reactive driving scheme determines the difference between its current heading and the heading to the next waypoint. As long as the vehicle is heading toward the waypoint, no steering adjustments need to be made. The deliberative approach uses a pure pursuit algorithm to drive the planned path as closely as possible.

In tests with lower maximum speeds, the reactive NUTS algorithm exhibited slightly more frequent braking than Dynamic Expanding Zones (Table 5.2). As the maximum speeds increase, the relationship reverses, but these differences are probably not significant. The Dynamic Expanding Zones algorithm is able to achieve higher average speeds on the course. If the vehicle is allowed to accelerate up to full speed in some sections, it will have to brake for turns. On the other hand, the NUTS algorithm has a high frequency of turning, which means the vehicle must first to slow to a safe speed to execute the maneuver. These effects seem to require roughly the same amount of braking effort.

The mission of the Grand Challenge requires the vehicles to navigate reliably throughout the 134 mile off-road course. To accomplish this reliably, extensive testing and tuning was performed prior to the event. It was found during testing that the reactive algorithm was more robust and less sensitive to small variations in sensor data than the deliberative algorithm. The graphs in Figure 5.18 are overlays of repeated runs on a practice course for the two algorithms. The reactive algorithm clearly produces more repeatable paths than the deliberative algorithm at all speeds.

5.6.2 Application

During development of both navigation strategies, the Virginia Tech Grand Challenge team made several notable observations that provide insight into the practical application of the reactive and deliberative driving schemes. The vehicle's

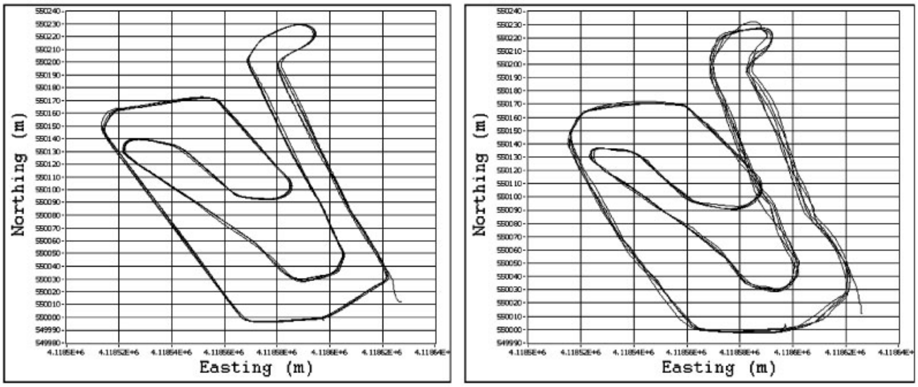

Fig. 5.18. Overlays of repeated runs for the reactive (left) and deliberative (right) algorithms

reactions to sensor stimuli and errors, such as high grass and error in GPS position, show key differences between the strategies. The measures taken to address these and other issues shed light on the characteristics of development and practical application of these strategies. Overall, the Virginia Tech Grand Challenge team found the reactive approach to be the most conducive to upgrades and gradual improvements based on field testing. The "general solution" approach of the deliberative scheme promises higher intelligence in navigation, but requires fundamental changes in navigation strategy to influence small changes in behavior.

5.6.2.1 GPS Error

A common experience for the Virginia Tech Grand Challenge GPS and inertial-based positioning systems is the "GPS pop" (Figure 5.19). This occurs when, after running on inertial-only positioning, the GPS/ INS regains the GPS signal. The perceived position of the vehicle instantaneously jumps from the INS-computed location to the GPS-based position. This "jump" has been measured as up to 2.5 m, in an arbitrary direction, based on the error in INS and GPS.

The reactive approach handles this type of sensor aberration without issue. There is no change in obstacle avoidance or waypoint following performance, except for a small heading adjustment based on the new perceived position with respect to the waypoint. This adjustment is inversely proportional to the vehicle's distance from the waypoint. Because obstacle avoidance is based entirely on the instantaneous measurement of obstacle position relative to the vehicle, no change is affected by noise or error in the positioning system.

The deliberative algorithm is significantly more susceptible to problems due to varying position error. Since the deliberative path planner attempts to drive a path based on data collected in a previous time period, it will use the most recent (corrected) position data to drive a path generated using an older, offset position frame. The now-offset path can potentially carry the vehicle through

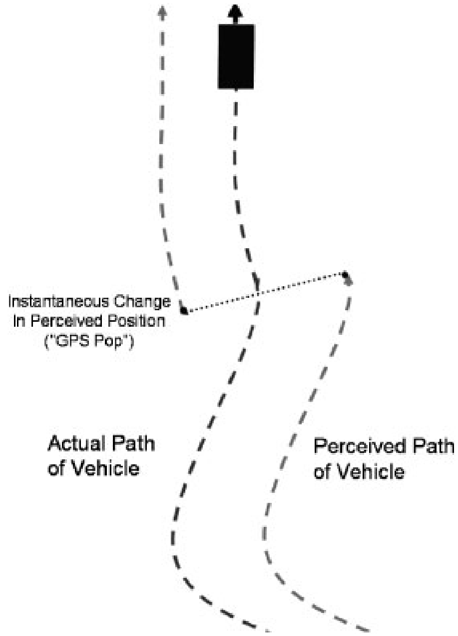

Fig. 5.19. When the GPS/INS reacquires satellites, the system corrects the INS-computed position

obstacles that were detected, but no longer perceived, in the correct location relative to the vehicle.

The general direction and shape of the "correct" path is still valid from the old computed position, but the data are no longer usable for obstacle avoidance. The long-term general planning capability of the NUTS approach could benefit from a reactive driver for more reliable close-in obstacle avoidance.

5.6.2.2 Navigation Range

The Dynamic Expanding Zones navigation approach has proven superior to the NUTS in robustness and reliability for obstacle avoidance in simple situations. The DARPA Grand Challenge is one such case where intelligent planning of complex maneuvers is not required. The reactive navigation strategy is superb at navigation through simple obstacles, such as passing cars and tunnel walls, but lacks the ability to plan through complex situations (Figure 5.20).

Because of the avoidance zone in front of the vehicle, a vehicle running Dynamic Expanding Zones might not be capable of maneuvering through close-quarters situations. In practice, Dynamic Expanding Zones showed a particular weakness in the offset-gate configuration of obstacles (Figure 5.21).

The offset-gate obstacle is traversable by the NUTS deliberative strategy, which is capable of planning a path through any area, regardless of the complexity of the

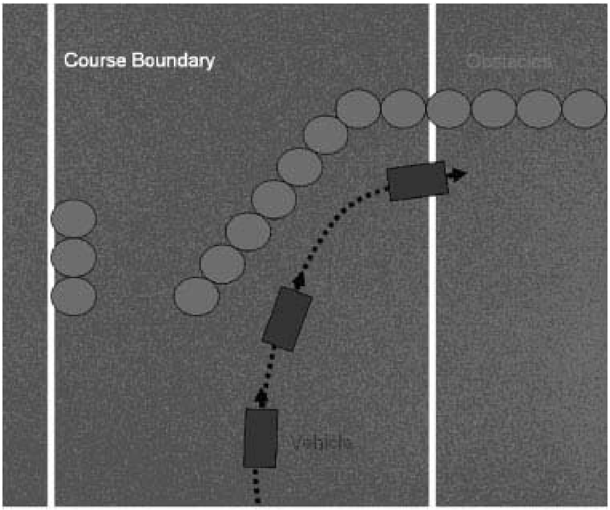

Fig. 5.20. Dynamic Expanding Zones does not take the optimum path in some situations

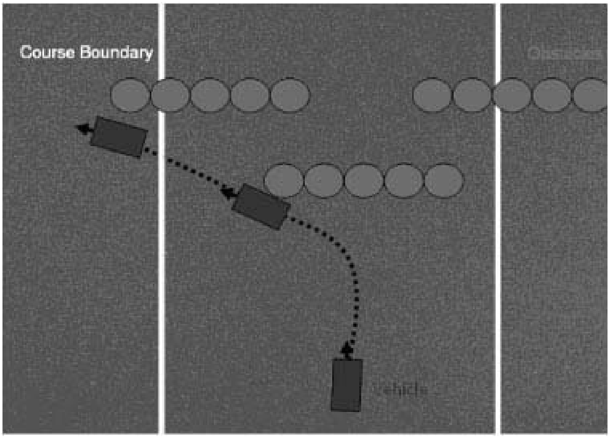

Fig. 5.21. Dynamic Expanding Zones reaction to the offset-gate obstacle

obstacle field (Figure 5.22). This long-range intelligence demonstrates the main attraction of the deliberative approach: The larger the area and complexity of data available for path planning, the greater the advantage to the deliberative approach. The drawback to using this strategy is that, as discussed in this paper, it lacks the adaptability and smooth obstacle avoiding performance of the reactive approach.

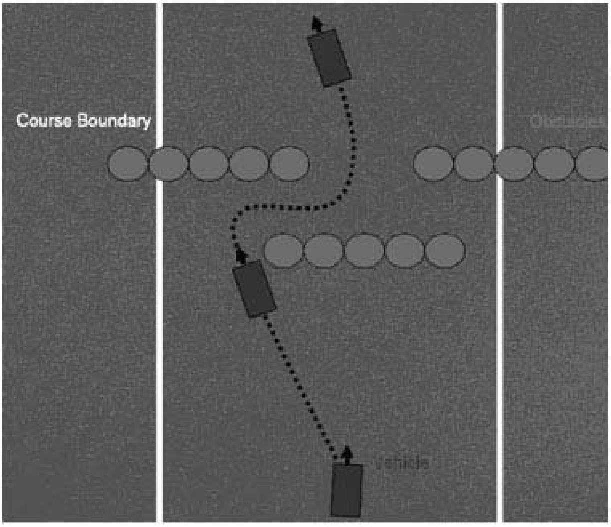

Fig. 5.22. NUTS reaction to the offset-gate obstacle

5.6.2.3 Road Following

Another area in which NUTS excels is road following. The data output from the Virginia Tech Grand Challenge road recognition algorithm to the NUTS is optimal for map-based path following. Rather than simply steer toward a point ahead of the vehicle suspected to be a road, NUTS attempts to find the shortest path onto the low-cost road terrain. Unlike Dynamic Expanding Zones, which ignores road following data in the presence of an obstacle, NUTS is capable of

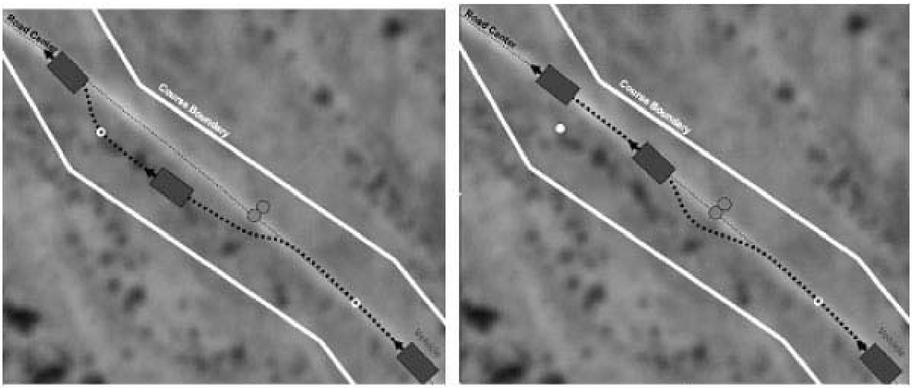

Fig. 5.23. In some situations, Dynamic Expanding Zones (left) can lose a road due to obstacle avoidance while NUTS (right) will maintain the optimal path (aerial photo from Google Local ®)

intelligently planning a path down a road, around an obstacle, and back onto the road again given sufficient sensor data (Figure 5.23).

5.7 Grand Challenge Performance

Neither Cliff nor Rocky were able to finish the 132 mile course. Rocky traveled just over 39 miles along the course, and Cliff traveled just over 44 miles. Both vehicles failed due to mechanical problems, rather than poor navigation decisions. Cliff's drive engine stalled when it briefly slowed to an idle, and Rocky's on-board generator shut down due to a suspected false low-oil reading. Had the base platforms not failed, the Virginia Tech Grand Challenge team is confident that the sensors and navigation systems would have allowed both vehicles to finish the race in just under the 10 h time limit. From the start, the Virginia Tech Grand Challenge strategy was to finish the race in the allotted amount of time and focus on solid navigation rather than higher-speed performance. In overall distance traveled, Cliff and Rocky finished eighth and ninth, respectively.

5.8 Conclusions

A team of dedicated undergraduate and graduate engineering students built the Virginia Tech GrandChallenge vehicles as an exercise in engineering design. Although the conversion to drive-by-wire, power system, motion control, and computing architecture are nearly identical on Cliff and Rocky, these platforms are designed to test and run two very different navigation strategies.

Cliff was designed to use the Dynamic Expanding Zones algorithm, a reactive navigation approach specifically created for the Grand Challenge. This algorithm consists of a number of variable size zones around the vehicle. In these zones, the presence or lack of obstacles dictates the behavior of the vehicle. The zones vary in size depending on the speed and surroundings of the robot in order to only take in the essential information to avoid obstacles. Rocky uses a deliberative navigation approach, making use of a terrain map and an A* algorithm to search through the map for the easiest route to travel. This allows Rocky to navigate more efficiently than Cliff through complicated terrain.

For the actual Grand Challenge Event and the National Qualifying Event, the reactive navigation software was used on both Cliff and Rocky. The deliberative software was not used, simply because the implementation was not mature enough to perform reliably in competition. One of the biggest lessons learned for implementing a deliberative navigation strategy was that it is essential to have a way to translate the optimal grid-based path to a path that is smoothly drivable. Another consideration is the size of terrain data. The deliberative algorithm planned a path from a terrain map of 12.5×12.5 m. This was too small an area to generate a useful plan.

In summary, the Virginia Tech case study describes and emphasizes some of the key design considerations for development of deliberative and reactive navigation.

The reactive strategy is simple and efficient, while the deliberative approach shows the potential to deliver higher navigational intelligence and planning.

References

Coulter, R.C. (1992). Implementation of the pure pursuit path tracking algorithm (Tech. Rep. CMU-RI-TR-92-01). Pittsburgh, PA: Carnegie Mellon University, Robotics Institute.

Gillespie, T.D. (1992). Fundamentals of vehicle dynamics. Warrendale, PA: SAE International.

Green, D., Yao, F., & Zhang, T. (1998). A linear algorithm for optimal context clustering with application to bilevel image coding. Paper presented the International Conference on Image Processing. Chicago, IL.

Hart, P.E., Nilsson, N.J., & Raphael, B. (1968). A formal basis for heuristic determination of minimum path cost. IEEE Transactions on Systems, Science, and Cybernetics 4, 100–107.

Lee, J.S. (1984). Edge detection by partitioning. In E.G. Wegman (Ed.), Statistical image processing (pp. 59–69). New York: Dekker.

Leedy, B.M. (2006). Two minds for one vehicle: A case study in deliberative and reactive navigation. Unpublished Masters thesis, Virginia Tech, Department of Mechanical Engineering, Blacksburg, VA.

Murphy, R. (2000). Introduction to AI robotics. Cambridge, MA: MIT Press.

Putney, J. (2006). Reactive navigation of an autonomous ground vehicle using dynamic expanding zones. Unpublished Masters thesis, Virginia Tech, Department of Mechanical Engineering, Blacksburg, VA.

Rasmussen, C. (2004). Grouping dominant orientations for Ill-structured road following. Paper presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04), San Diego, CA.

Reynolds, C.W. (1999). Steering behaviors for autonomous characters. Foster City, CA: Sony Computer Entertainment America.

Shin, D.H., & Singh, S. (1990). Path generation for robot vehicle using composite clothoid segments (Tech Rep. CMU-RI-TR-90-31). Pittsburgh, PA: Carnegie Mellon University, The Robotics Institute.