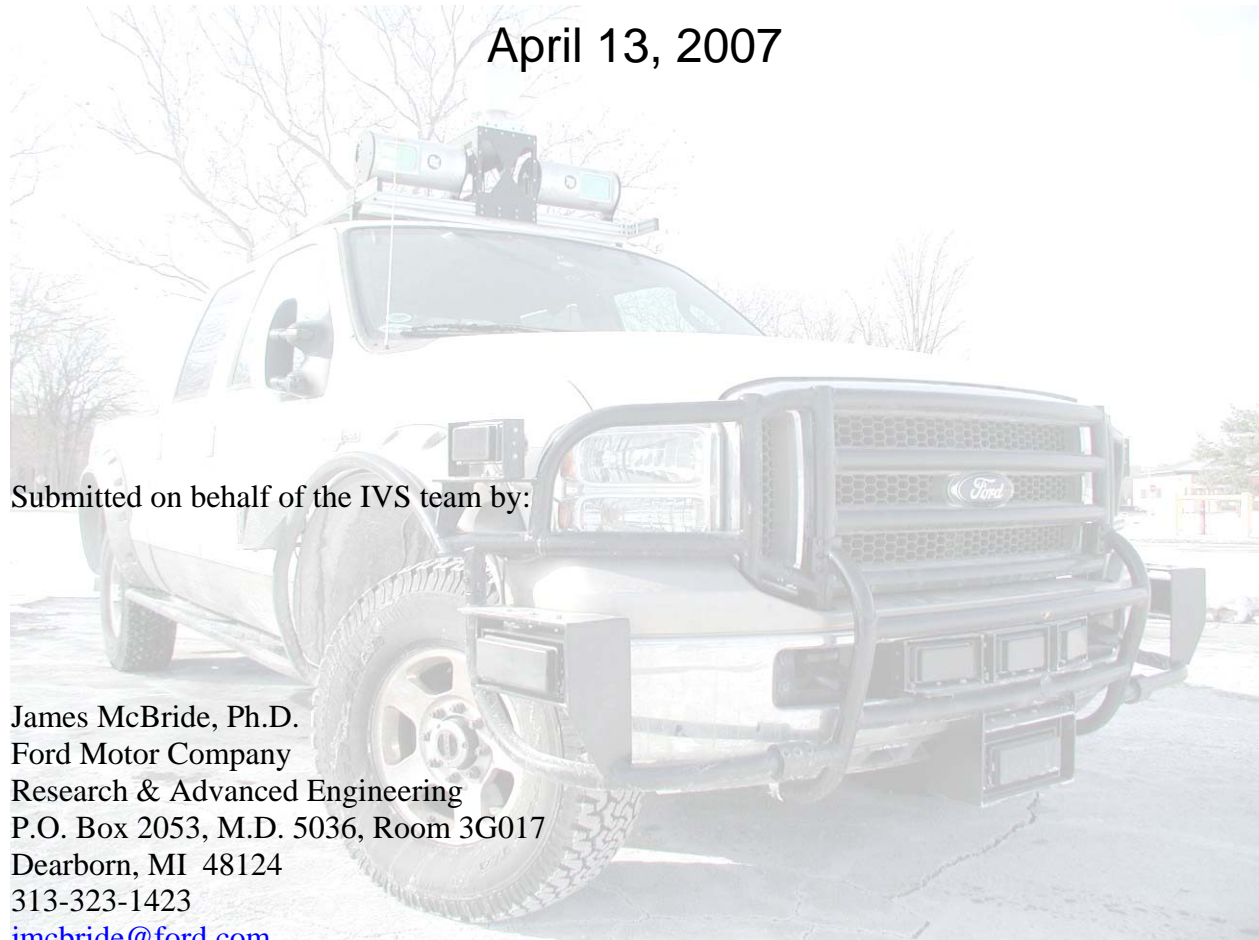


DARPA Urban Challenge

Intelligent Vehicle Systems

Technical Paper

April 13, 2007



Submitted on behalf of the IVS team by:

James McBride, Ph.D.
Ford Motor Company
Research & Advanced Engineering
P.O. Box 2053, M.D. 5036, Room 3G017
Dearborn, MI 48124
313-323-1423
jmcbride@ford.com

“DISCLAIMER: “The information contained in this paper does not represent the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA) or the Department of Defense. DARPA does not guarantee the accuracy or reliability of the information in this paper.”

Executive Summary

This paper highlights the technical status of the Intelligent Vehicle Systems (IVS) entry in the DARPA Urban Challenge (DUC). Additionally, we discuss our philosophical approach to the problem of autonomous vehicles, how this impacts the details of our design, challenges involved and testing required to achieve our goals. The motivation for participation is twofold – IVS not only wishes to contribute to the body of knowledge in the robotics community, but being comprised of major manufacturers, ultimately wants to bring these technologies to fruition for the greater benefit of society. Creating safe, intelligent vehicle systems for production-intent commercial and military applications imposes additional requirements beyond those of the DUC, both in design robustness and development time. As such, IVS is creating two identical platforms based on the Ford F-250, and dubbed eXperimental Autonomous Vehicles (XAV-250s). This dual approach will facilitate parallel testing and development. While this method carries some up-front time penalties, it will accelerate progress over the long term. At present, nearly all hardware components are installed and operational on both platforms, with testing of algorithms underway. It is fully anticipated that the team will have a vehicle capable of basic traffic operations in ample time for the site visit.

1. Introduction and Overview

1.1 Problem Statement

While the mission statement for the DUC has been clearly communicated by DARPA in a variety of forums (participant's conference, press releases, official rules, the FAQ, etc.), what DARPA doesn't state is what it will take to accomplish solving the Challenge, nor how this fundamentally differs from the first or second DARPA Grand Challenge (DGC). Nominally featuring autonomous ground vehicles maneuvering in a mock city environment executing simulated military supply missions, it could be stated that the DUC is the first step toward a realistic autonomous ground vehicle application.

From the IVS perspective, one of the most obvious differences between the first two DGCs and the upcoming DUC is the dynamic nature of this new event, which promises moving targets, intermittently blocked pathways and periodic outages in navigational guidance. Whereas the 2004 and 2005 routes could, in principal, be negotiated with a good GPS/INS path follower and a small set of stand-alone sensing techniques, this most certainly will not be the case for the 2007 mission routes. This event will require multiple real-time, fused sensing modalities, coupled with sophisticated path planning throughout regions of extended denied-GPS navigation. The concepts of redundant sensing and fault tolerance are critical to mission success and are consistent with the latest production and pre-production safety systems found in major automotive and aerospace products, such as those designed and produced by the major IVS partners – Delphi, Ford and Honeywell.

1.2 Team Description

The IVS team is a collaboration among a wide array of technical experts in vehicle platforms, navigation, path planning, sensors, perception, and intelligent autonomous control for air and ground vehicles. The team is primarily composed of employees from Delphi, Ford and Honeywell, and contains many core members from the 2005 DGC IVST team which fielded the autonomous Ford F-250 called the “*Desert Tortoise*”. As such, the team members bring a wealth of prior experience to the project. Each of the partner corporations has assumed at least one

major leadership role in the project – Delphi, as the world’s leading producer of automotive parts, is serving as the sensor and systems integration and validation leader; Ford has assumed responsibilities for the vehicle platform, vehicle dynamics and by-wire control; and Honeywell is providing the navigation solution, implementing the path planning and leading the design of the software architecture.

1.3 Design Philosophy

While the DUC presents a unique and persuasive motive for competitively participating in the solution of a difficult problem in autonomous ground vehicle navigation, it is nevertheless highly aligned with the primary objectives of the IVS corporate partners. As has been previously stated, our ultimate goal lies in the creation of safe and intelligent vehicle systems for production-intent commercial and military applications. As such, one of the guiding principles in the design of our entry is to assure that the hardware, sensors, and algorithms developed not only address the mission goals of the DUC, but also offer a practical path toward future use in other applications.

In many phases of the project, this philosophy implies that the wisest choice is to employ robust, production-proven components and rigorous systems engineering practices. On the other hand, there are certain aspects of the DUC for which no technical solution presently exists, requiring that risks be taken exploring novel sensing hardware and algorithms. With limited time and resources, the key to success lies in choosing the right balance between status quo and innovation. One of our team strengths in this regard is the ability to leverage knowledge and expertise from the diverse engineering background of our collective corporations, suppliers, and university and governmental research collaborators.

In the prior DARPA Grand Challenge, the IVST team employed a situational dependent, arbitrating behavior based solution. While this proved to be effective at the National Qualifying Event, during the Final Event a situation occurred where multiple competing behaviors with high confidence came into play at one time. While the correct behavior did eventually prevail, the arbitration delay initiated a chain of events which led to a minor departure from the course, and ultimately the team’s disqualification. Learning from this lesson, the IVS team has developed a more robust design philosophy. This philosophy is based on five basic elements:

1. Robust, production-proven vehicle platform capable of off-road and urban performance
2. Sensor fusion laws utilizing a fault tolerant strategy that are not situational dependent nor arbitrated
3. High performance navigation system aided by GPS, wheel odometry and vision to remove positional uncertainties from the problem
4. Distributed computing solution which supports parallel, fault tolerant processing.
5. Thorough multiple-step testing process which proves out levels of performance to yield a final, fully-tested system

Vehicle platform criteria/philosophy: Based upon the success of and lessons derived from the “*Desert Tortoise*” utilized in the DGC2, the 2005 model year Ford F250 has again been selected as the platform of choice for the IVS research efforts. Although a number of necessary modifications have been made to meet the operational requirements of the DUC, the guiding

principle has been to keep the design changes as simple, flexible and production-representative as possible in order to maximize system robustness, reliability and fault tolerance.

While this might be considered a rather large vehicle for an urban contest, the F250 truck series has been extensively used as a rugged all-purpose workhorse, operating on roads of all sizes and surface conditions around the world. As such, the IVS team believes it embodies the type of platform that would likely be utilized for a realistic autonomous mission – commercial or military. We have dubbed the new pair of trucks the “XAV-250”s, models A and T– (XAV being short for eXperimental Autonomous Vehicle).

Above and beyond the theme of simplicity mentioned earlier, safety is the primary guiding principle directing all of our efforts. Each of the by-wire systems – throttle, brakes, steering, and transmission – operate with redundant mechanical interfaces, enabling the XAV-250 to easily transition from human-controlled, street-legal to fully autonomous operation by the flip of a switch. Occupant and bystander safety is further enhanced by the use of redundant, fully independent, remote wireless E-Stop systems. When a disable/pause command is initiated, there are multiple means by which the command will be obeyed.

Sensing criteria/philosophy: The transition from a desert environment to an urban environment, as well as changes in the mission specifications, requires many alterations in the methods and strategies for sensing the world. In the DGC2, path planning was constrained to look for obstacles lying in a narrow corridor closely aligned with a dense set of GPS waypoints. The addition of parking lots, traffic, and intersections associated with an urban setting force the path planner to process vastly more information in a timely manner in order to determine the optimal route. To gather this new information requires a more comprehensive understanding of the dynamic world surrounding the vehicle, and hence new and additional sensors, along with associated algorithms, are being developed to satisfy this demand.

At first thought, completely surrounding the truck with as many sensors as possible might seem to be the best approach to solve the XAV-250’s increased need for situational awareness. However, this adds an unwieldy burden to the computing platform and sensor processing systems. A better solution is to determine where the XAV-250 needs to “look” to execute the required mission maneuvers and place sensors accordingly to fulfill these needs. To provide additional sensing data, assure an adequate level of redundancy and to protect against failures, multiple sensors of each type are utilized as well as different sensing modalities (e.g., vision, radar, lidar).

The first step in this design process is to focus on the driving landscape specified by the Urban Grand Challenge requirements. This includes the known data contained in the RNDF, such as types of roads, lane markings (where present), the geometries of the network and intersection nodes, etc. This information is further enhanced with additional materials derived from existing maps, knowledge related to roadway design guidelines and highway driving rules.

The second step in the sensing system design philosophy process is to understand the possible obstacles that could be encountered during the Urban Challenge. As with the infrastructure requirements, the rules specified by DARPA include a limited set of real world parameters. The

IVS team philosophy, again, is to design above and beyond these scenarios. For example, the hardware used in our fused radar/vision system is also being utilized for pedestrian protection research.

The third step is to consider the different technologies available to address the infrastructure and obstacle requirements. The IVS sensor design philosophy is to rely on proven technology whenever possible, and for the most part can be considered an evolutionary versus revolutionary approach. As an example, rather than taking a completely new, ground-up approach, the XAV-250 utilizes the proven radar sensing systems that enabled passing a paused competing robotic vehicle during the last Grand Challenge. The vehicle platform is further enhanced with a new set of technologies to address some of the more complicated set of requirements associated with the DUC. Although some of these sensors are based upon production systems with known capability and performance, they are being implemented in a unique and novel fashion. This simplifies the design of the sensing suite and allows for team resources to focus on introducing a few key new technologies (e.g. Velodyne HDL-64E lidar).

Navigation criteria/philosophy: The XAV-250 uses a NovAtel GPS, odometer, and vision aided inertial navigation system for vehicle position, attitude, acceleration, and rotation rates. Position includes latitude, longitude and elevation. Attitude includes roll, pitch and yaw in the vehicle frame relative to local level. Acceleration includes the longitudinal, lateral and vertical accelerations in the vehicle frame, and the rotation rates are the rates of rotation about the X, Y, and Z vehicle axes. All of these items are necessary for vehicle path planning or sensor data processing regardless of the availability of GPS. The philosophy is to have stable, accurate pose information during GPS outages of up to 20 or 30 minutes.

An inertial measurement unit (IMU) utilizing three GG-1320 ring laser gyroscopes (<0.005 degrees/hr error) and three QA-2000 accelerometers (less than 80 μ g errors) form the core of the inertial navigation system. The navigation algorithms use this IMU for its 100Hz outputs of position, attitude, acceleration and rotation rates. GPS is used to establish the initial location of the navigation package while the IMU quality allows self-gyro compassing to determine heading to approximately one milliradian. 1 Hz Omnistar HP GPS messages are used to establish accurate position (nominally to 20cm) and in the Kalman Filter to estimate the slowly varying inertial errors and constrain the inertial position drift. Additionally, the four individual wheel speeds are averaged and supplied to the filter as an odometer for constraining inertial drift in the absence of valid GPS measurements. The scale factors of the odometer are estimated by the Kalman Filter. The Kalman Filter rejects GPS and odometer measurements when outside the bounds possible given the quality of the IMU. A Doppler radar is also available and installed on the truck for use as another velocity aid to the navigation package should it become advantageous.

A stereoscopic vision system is also used to aid the inertial package. A SIFT algorithm is used to identify and track features in the imagery. The feature tracking through the stereo pair allows estimation of along-track and cross-track odometry as well as attitude information. This is then fed back to the Kalman filter to provide accurate navigation in the absence of GPS for 20-30 minutes.

Software criteria/philosophy: IVS software philosophy is based on a loosely connected set of distributed software components. Low-level control software executes in a predictable OS connected to the vehicle's control network; whereas the guidance and sensing software primarily executes on multiple general-purpose computing nodes connected over Ethernet. Data reduction (or abstraction) occurs as close to the data-source (e.g. the sensor processing) as practical. Tight timing synchronization is not required except at the sensing and control ends of the data flow.

Testing criteria/philosophy: As in the strategy of the system design, the basic philosophy for testing the XAV-250 is to start with smaller pieces, aka components, and then build upon these results. The testing starts at the component level, then builds to the sub-system level, and finally moves to the vehicle/system level. Key to the entire strategy is the highly integrated data acquisition system incorporated into the IVS design, enabling rapid debugging and re-simulation.

The data acquisition system is capable of archiving the relevant data at all critical junctions during the component/sub-system/system validation process. With respect to component testing, data generated by the sensors can be captured and compared against production data collection tools. In a similar manner, data that commands actuators can be recorded and compared to actual performance of the parts. As sub-systems are brought online, the data acquisition system has the capability to measure the inputs feeding a process along with the outputs from that process. In this manner, off-board simulations can then be used to further refine system performance or aid in identifying anomalies in the sub-system. The same holds true as the full vehicle system is enabled.

At the full vehicle level, the concept of building one system operational test upon the previous continues. The use of simple missions on a basic test course will be the first step. Next, the complexity of the mission and/or test course(s) will be changed. This process will also include the introduction of different testing scenarios such as the addition of basic traffic, objects, roadblocks, etc. As more confidence is built, the testing will then move to the advanced traffic and navigation requirements, including but not limited to, passing vehicles, intersection precedence, merging into moving traffic and roadway gridlock.

Another important factor in the testing philosophy of the IVS team is the use of two identical truck platforms. These dual platforms will allow for quicker testing as well as more in-depth evaluation of the various testing conditions. This will increase the robustness of the XAV-250's systems by proving the repeatability of the test. The two platform approach also allows for split testing. In this scenario one XAV-250 can be tested to investigate an anomaly while the second XAV-250 completes the test or moves onto the next test. Another distinct advantage of utilizing a dual platform approach is the capability to isolate software versus hardware issues by trying to reproduce the observed behavior in both vehicles.

2. Analysis and Design

2.1 Vehicle Platform

Based upon the success and reliability of the Ford F-250 previously used as the team platform, it has again been chosen for future research efforts by the IVS members. Following hundreds of miles of rugged testing, most taking place on harsh 4 x 4 road conditions, very few changes have been made to the vehicle, aside from removing a few cacti and some Mojave dust. A description

of the original modifications required to alter a stock F-250 into a XAV-250 can be found in the final report from the DGC2 [1] and the Track A Proposal for the DUC [2]. Among the upgrades and/or changes to the truck(s) following the last Challenge include additional protection for the sensors, modular roof racks, a dual E-stop system, replacement of the locking differentials with stock axles, street tires (as opposed to Kevlar sidewalls), an upgraded power and signal distribution system, reconfigured hazard and warning lights, turn signal indicators, wiper and washer control, and enunciator lights for system health monitoring. Note that the bulk of the by-wire systems and vehicle dynamics remain unchanged, maintaining the stable vehicle platform used previously.

Compute system: The computing cluster utilized to process sensor data, path planning, and execute autonomous driving tasks consists of five LINUX based servers, comprising eight dual-core CPUs, and four P4-class XPC based computers. The cluster is protected from the shock of the roadway through the suspension of the XAV-250 as well as being placed in a sealed, independently air conditioned box on an isolation table that rejects vibrations greater than 5Hz.

Data acquisition system: The data acquisition system incorporated into the XAV-250 design consists of an XPC computer that monitors all twelve radar sensor channels and the Ethernet traffic associated with LIDAR, vision systems, and overall system processing. A user interface has been designed to enable the selection of data storage variables on hot swappable hard disks. In this manner, off-board simulations can then be used to further refine system performance or aid in identifying anomalies in the sub-system.

2.2 Sensing Solution

Infrastructure Parameters: As mentioned in the sensor philosophy section, the sensor design solution is based upon the infrastructure environment through which the XAV-250 must traverse and the types of obstacles the XAV-250 may encounter. The infrastructure descriptions specified by DARPA include:

- The road surface
 - New pavement to potholes and broken pavement.
 - Dirt roads with low berms
- Infrastructure
 - Static lane markings
 - Sharp curbs
 - Traffic circles
 - Intersections
 - Blocked roads
 - Parking lots
 - Stop lines
 - Trees and buildings along the route may interfere with GPS.
- Obstacles
 - Moving traffic.
 - Manned vehicles
 - Tele-operated vehicles
 - Other autonomous vehicles.
 - Static vehicles may also be parked or stopped along the route
 - Downed branches
 - Traffic barrels
 - Drains and hydrants
 - Rocks and brush
 - Construction equipment
 - Concrete safety rails
 - Power line poles

The IVS team understands the critical nature of the infrastructure to the success of navigating the roadways. Calculations based upon the line of sight along roadways was utilized to determine optimal sensor placement and to assure that a complete 360° field of view is covered by the

sensing suite. See Figure 1 and Table 1 for one example of Line of Sight from the Florida Intersection Design Guideline. Sight distance values apply to both rural and urban intersections under stop sign control or flashing beacon control.

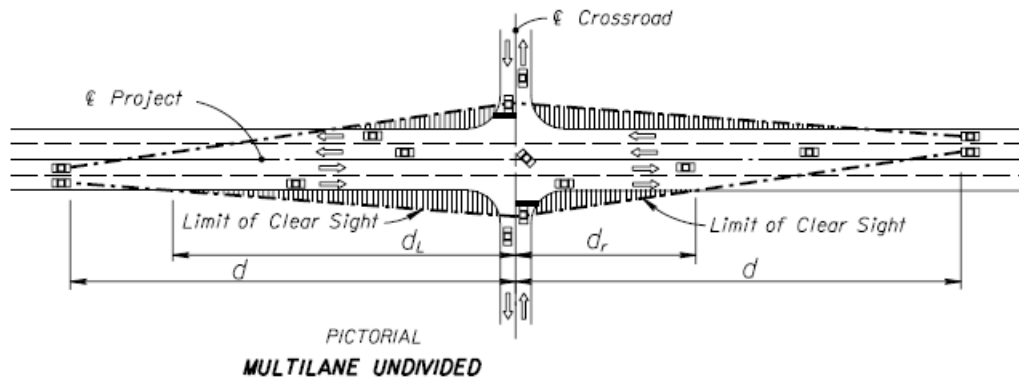


Figure 1 – Line of Sight Diagram for Multilane Undivided Road

Type Facility	Distance (feet)	Design Speed (mph)							
		30	35	40	45	50	55	60	65
Multilane Undivided	d	380	470	580	710	840	990	1150	1350
	dL	290	370	450	550	650	760	890	1040
	dr	160	190	240	290	340	400	460	540

Table 1 - Minimum Sight Distance and Related Distances

Fields of View: Once the vehicle and infrastructure design parameters were understood, the next major area of development was to select a sensor plan that would enable execution of future missions. The IVS team believes that “Completing the Mission” is an essential goal of the DUC project; as such, the failure of one sensor could not jeopardize the XAV-250 from this goal. To insure this, an overlapping sensor field of view strategy was implemented. To protect the XAV-250 in its mission, several “rings of defense” have been established. See Figures 2, 3, and 4. The outer ring is capable of detecting obstacles at long range (out to 150m) and tracks them as they approach the XAV-250. A center ring of defense exists utilizing different technologies to detect obstacles that might be difficult to detect with the long range technologies and also to perform addition tasks more relevant in a mid range setting. The inner circle, aka, the “last line of sensor defense”, consists of short range radars strategically placed around the XAV-250. These radars set up a protective field both to the front and rear of the vehicle as well as at the front corners to catch block obstacles as the XAV-250 executes a turn.

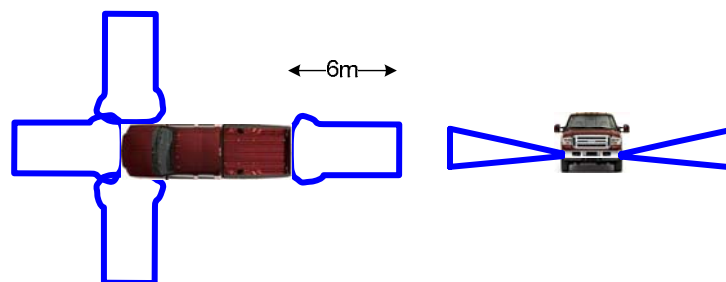


Figure 2 – Short Range Sensor Map

As mentioned, the center ring of sensing has a dual role of detecting any obstacle that might be missed by the long range sensors but is also used to perform other tasks such as detection of roadway markings and curbs. Vision systems and line scanning LIDAR were selected to support these functions. The line scanning LIDARs are focused at two different distances, 25m and 12.5m. These distances were selected to insure stopping capability if an obstacle was missed by the long range sensors but detected by the LIDARs. Derivatives of the range data is utilized to detect curbs and obstacles while the intensity of the return signal produces data used to identify roadway markings. Knowing that not all “markings” are reflective in nature, redundant vision systems have been implemented to aid in the fault tolerance of lane detection. A customized Mobileye vision system with a primary focal point of 40m and a Cybernet vision system with a primary focal point of 25m are both capable of detecting lane markings and vision gradient differences between the roadway and side of the road. See Figure 3.

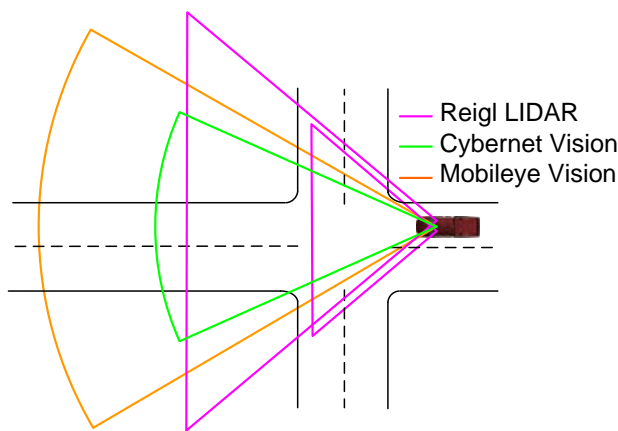


Figure 3 – Scaled Mid Range Sensor Map on Typical Roadway

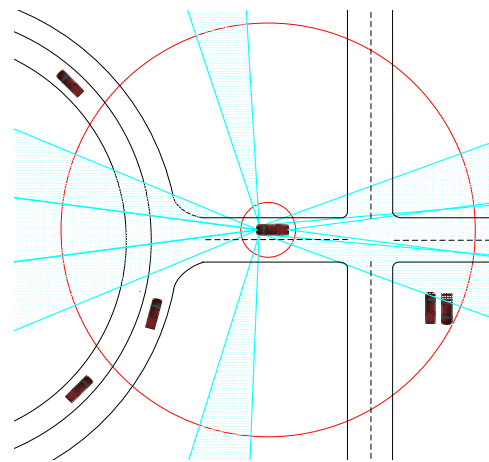


Figure 4 – Scaled Long Range Sensing Map on a Typical Roadway

Radars and scanning LIDARs were selected to identify obstacles, both moving and stationary. The radars utilized are automotive production units utilized in smart cruise control systems with a 150m range. Given the infrastructure challenges, radars were strategically placed around the XAV-250 to create a wide forward and rear field of views; See Figure 4 – blue triangles. The forward radars provide a 45° field of view enabling multi-lane, curved road area of coverage. The rear facing sensors are placed at the rear and front corners on the XAV-250 to provide rear and next lane obstacle tracking capability. Two additional radars are mounted on the front corners to provide coverage of obstacles at 90° intersection zones.

Knowing that not all intersections are 90° and to support infrastructure elements such as roundabouts, the radars are supplemented with a Velodyne scanning LIDAR. This sensor is mounted on the XAV-250 and provides a 50m range of data with a 360° field of view of both stationary and moving objects along with roadway structure data; see Figure 4 – red circle. Object track data will be generated from the Velodyne data and used for object tracking. This will be particularly useful for laterally moving objects. Figure 5 is a collage of the first completed XAV-250 build. It highlights sensor placement as well as the computer cluster.

Sensor processing: To enable a more distributed and fault tolerant system, raw data from different sensor types is first introduced to the computing cluster at different processors. If one

sensor processor should fail, the data from other sensors should still be available for use by the system. Processing is further distributed by having each associated sensor processing center perform the transformation from vehicle to earth coordinates. This process also incorporates the pitch, roll, yaw and time stamp data received from the IMU to insure proper data placement within the various map databases.



Figure 5 – Sensor placement and computer cluster on the XAV-250

Pre-Fusion: While the primary fusion engine resides with “The One Map - TOM” (see software solution), several sensors have already been designed to work together for the automotive production market. In particular, forward looking radars are used in conjunction with vision to create a “semi-fusion” data set utilized by TOM. Radar data is shared with the vision processor to identify locations in the field of view for concentrated processing. This yields target confirmation as well as addition information such as object edge points.

2.3 Software Solution

Software is a key component of the overall system architecture. It controls and coordinates various parts of the XAV-250 hardware to enable autonomous behavior. Figure 6 describes the overall software architecture for the IVS system. A key component of the architecture is "The One Map" or TOM that acts the obstacle repository. Individual sensors classify obstacles and inter-sensor fusion occurs at TOM. TOM also contains a database with track and lane information. Another key component is the "Planner" that uses the obstacle information to determine a route. The planner is a two-level planner with a "fast, long-term" planner that uses static RNDP information and a "short-term" planner that uses the actual sensed obstacles. The "Path Follower" provides steering, throttle, and brake commands to the drive-by-wire system to achieve the path guidance from the planner.

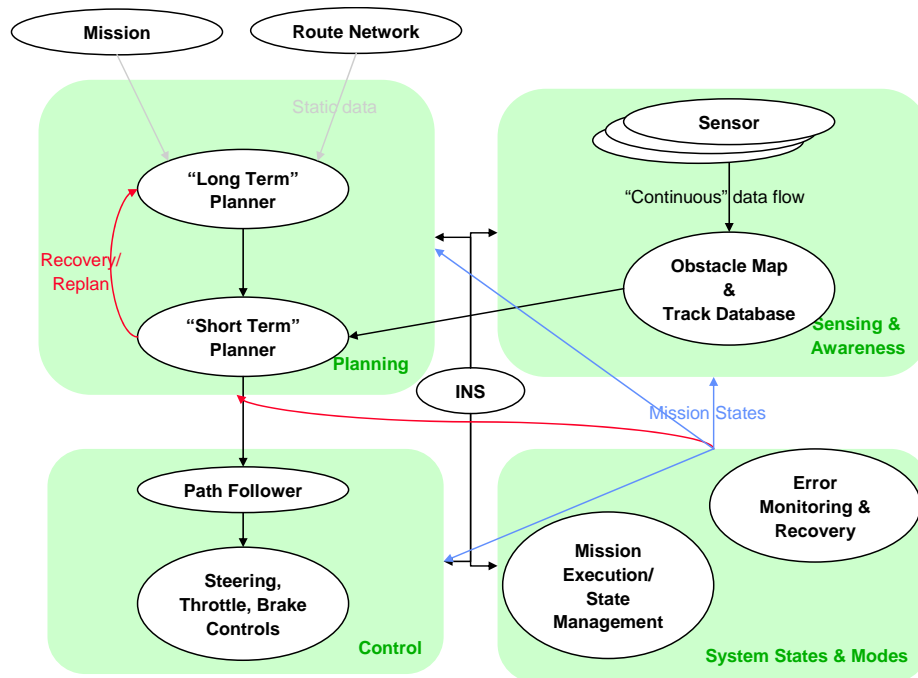


Figure 6 - Overall Software Architecture

Software infrastructure: The software infrastructure is a group of abstractions that provide an easy to use interface to the operating system. In addition to simplifying interactions with the OS it also provides higher level functionality (e.g. logging, heartbeats, etc.). By providing abstractions, the complexity of application code is reduced and its reliability is increased. This is accomplished by creating and testing utility classes that encapsulate the desired functionality. The software infrastructure components include:

- **Logging** – provides applications the ability to write (to file or network) information that will persist after the application has terminated.
- **Time Management** – manages system time between nodes in the distributed system.
- **Threading** – provides applications the ability to schedule threads (periodic and aperiodic) and provide protection for critical sections (mutexes).
- **Configuration** – provides applications the ability to configure behavior at runtime.
- **Communication** – provides applications the ability to define typed readers and writers of data that will be sent over the network.
- **Heartbeats** – provide a fault detection abstraction that allows applications to signal that they are alive. Heartbeats working in conjunction with a fault manager will allow detection and recovery from dead components.

Sensor Fusion/Aggregation Approach, Pixel Mapper & Track Database: The fusion of sensor data is a difficult task. The proper balance of the complimentary sensors is required to insure correct coverage of an area being traversed. This problem is further complicated by the fact that some sensors generate pixel based data while others generate vector oriented data. Pixel data is the term we use for data that is fused into grid cells, where those grid cells are classified by the sensor as to what is on the earth at that point. Our classifications are “unseen”, “non traversable”, “curb”, “lane”, “empty” or “moving”.

The approach for pixilated data is to have each sensor generate data at one point in time, process the data into one classification per 0.25 x 0.25 meter cell in earth coordinates, and then send this result to a single mapping process (TOM) that combines results from all sensors for each cell. The sensors classification of a cell is for one point in time, if the classification for the next reading is different, that does not concern the sensor. In doing this, the sensors are performing spatial fusion for one instance in time. Temporal fusion (changing classification over time) and sensor fusion (different classifications of a cell) are handled in the mapping process.

The purpose of TOM is to store probabilities of “classified truth” based on each sensors reports of classification for each quarter meter cell of the world, in world coordinates. Each cell stores six probabilities (Curb, Lane, Driveable, Non Traversable, Empty and Moving) for four sensors plus ancillary information including a data lifetime counter and an “unseen” flag. When a sensor reports to the map its most recent classification of a quarter meter patch of the world, the probability for that cell (for that specific sensor) is incremented and decremented appropriately. Increment and decrement rules are calibration factors adjusted during testing of the XAV-250. Table 2 shows a representation of a single cell.

	Far Riegl	Near Riegl	Velodyne	Radar Vision
“Unseen”	True / False	True / False	True / False	True / False
Life time	100 down to 0	100 down to 0	100 down to 0	100 down to 0
Moving	NONE	NONE	0-100	0-100
(L)ine on Road	0-100	0-100	0-100	0-100
(C)urb	0-100	0-100	0-100	NONE
(N)on Traversable	0-100	0-100	0-100	0-100
Driveable	100-(L+C+N)	100-(L+C+N)	100-(L+C+N)	100-(L+C+N)

Table 2 - A representation of one cell of the map.

The mapper holds a quarter kilometer of data in memory (1024x1024 cells) and swaps data in and out of files as the truck moves. In this way, when the vehicle returns to a part of the world it has already visited, detailed data is already available. Internally the map is represented as a two dimensional circular buffer (or a torus), which allows the map to “slide” as the vehicle moves by simply changing the index of the reference corner. This eliminates a large amount of data copying and only requires a few rows/columns to be either cleared or replaced with data off disk.

At a 10HZ periodic rate, the probabilities on a 128x128 meter (1/8 of a square kilometer) “window” of the map centered on the vehicle’s current location is fused for the planner. At this fusion step, the probabilities of each cell across each sensor are fused together using a trust weighting and a factor for lifetime score. This weighting is also stored off line, so that it can be easily updated in the field. Weights are per sensor / per probability, so sensors that are particularly good at sensing one type of phenomena but not another can be tuned to influence the final answer. Upon completion, spatial and temporal sensor data is available to the planner.

As mentioned some sensors produce data that is best kept in a vector form. This includes radar tracks, lanes and road edges. To handle vector data, two tables of entries centralized in one of

the processors are managed. These tables are referred as the Track Database and Lane Database. Tracks and Lane entries are maintained and updated asynchronously by the sensors as they update their estimates. The planner then accesses the databases to update its internal structures.

Long-Term Planner: In the two-level path planning paradigm, the long term planner operates at the higher level, essentially providing turn-by-turn directions to cross the mission checkpoints in sequence while utilizing the most time-efficient route possible given current knowledge. The long term planner utilizes standard graph theory principles and *Dijkstra's Shortest Path Algorithm* [3] to determine the best route. A directed graph is constructed with vertices corresponding to RNDF waypoints, and edges corresponding to lanes or road segments. Edge costs are estimated based on a combination of inputs, such as the distance between points, whether the edge constitutes a sharp turn, the estimated or observed probability of failure, and speed limits specified by the MDF.

Automated analysis is performed on the RNDF to identify the sides, or segment cross sections, of an intersection of two or more segments, and to identify parallel lanes with the same direction of travel. This analysis improves and simplifies the results of Dijkstra's algorithm. For example, it allows multiple lanes in the same direction to be merged into one graph edge with a lower probability of failure or a route that has fewer turns to be preferred over one with many.

If the short term planner detects a blocked road, it notifies the long term planner and begins to return to the last waypoint the truck crossed. The long term planner executes a *re-plan* which marks the blocked road with a high probability of failure and initiates Dijkstra's algorithm to find an alternate route to the next checkpoint starting from the most recently crossed waypoint.

Short-Term Path Planning Architecture: The role of the Short-Term Planner (STP) is to process the map fused by TOM and produce a set of path points (positions and speeds) to send to the path follower. The first step in this process is to take the fused map and produce two maps: the path planner map (PPM) and the speed planner map (SPM). SPM contains information regarding moving obstacles needed by the speed planner. The PPM map is used by the Laplacian planner to determine whether there exists an obstacle free path that converges towards the goal point. Upon convergence, a set of projected path points is computed by Laplacian and is used by the speed planner to determine the speed at each point. The speed planner communicates to the path follower a set of path points along with recommended speed at each path point.

Coordination between the speed planner, the Laplacian and the Map processor is accomplished through what is referred to as the "coordinator", which maintains a state like behavior of the short term planner and determines when the STP transitions from one state to the next. An example of a state transition is changing mode from nominal driving mode to N-point turn mode. In addition to coordinating the sub-components of the short term planner system, the coordinator serves as the interface between the short term planner and the long term planner. See Figure7 and Table 3 for an overview of the operation of the STP.

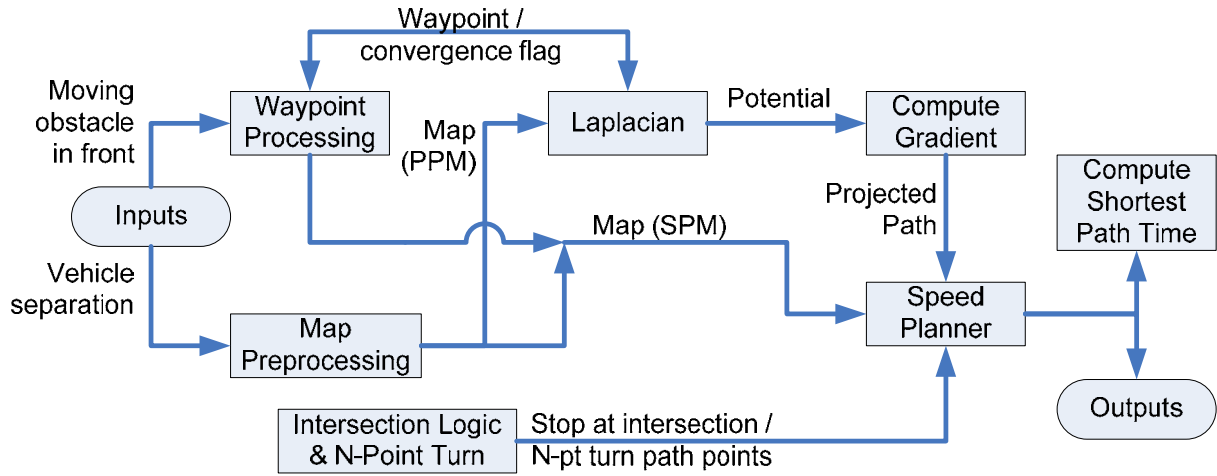


Figure7 - Short Term Planner

Inputs		Outputs	
Waypoints Sequence	Max Safe Speed	Waypoint Reached	Speed
Waypoint Reached	Max-Min Turning Radii	Unreachable Waypoint	Turn signal
Vehicle Full State	Undulation Factor	Truck Direction	Shortest Waypoint Time
Fused Map from TOM	Vehicle Separation	Path (to Path Follower)	
Is Safety Area FLAG	Map Coordinate Frame		
Speed Limit of Segment			

Table 3 – Short Term Planner Operational I/O

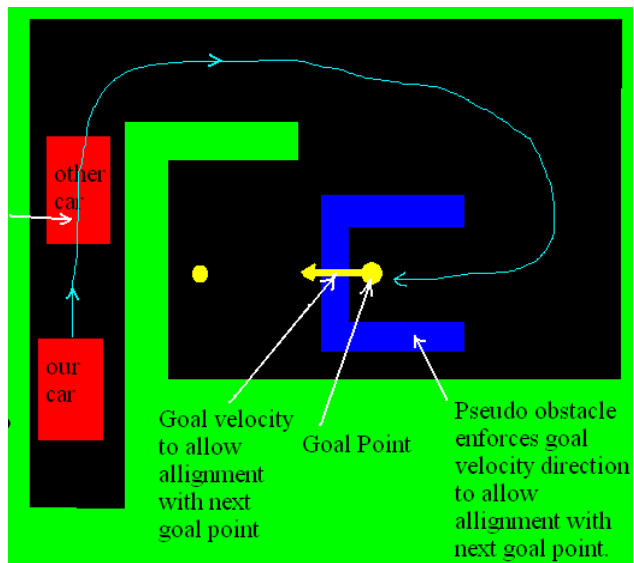


Figure 8 – Short Term Path Planner Example

Short-Term Path Planner Algorithm: The STP uses Laplace’s equation for path planning, obstacle avoidance and collision avoidance. Paths are constructed to avoid all fixed obstacles while allowing paths through moving obstacles (e.g. the vehicle ahead). The speed planner will slow the XAV-250 to avoid colliding with these moving obstacles. Obstacles considered by the path planner include physical obstacles such as curb, building, cars, as well as “pseudo obstacles” used to enforce various constraints. Example of pseudo obstacles include: painted white/yellow lane lines, as well as a turning radius and goal-velocity constraints at waypoints; see Figure 8.

Previous applications of Laplacian path-planning include [4], [5], [6], [7] and [8]. A physical analogy for paths obtained by Laplace's equation is to apply a voltage of 0 to all boundary and obstacle locations, a voltage of -1 to goal region, fill interior region with a conductor, then positive charges will follow smooth paths from anywhere in the interior to the goal region. Laplace's equation sets the divergence of a potential to zero in the domain. Solutions of Laplace's equation are harmonic functions, which have no local minima in their domain.

Numerical solutions of Laplace's equation are obtained by gridding the domain, then iteratively setting the potential at each interior point, equal to the average of its nearest neighbors. By varying the grid size (halving or doubling cell length at each step) from the crudest that still leaves paths between obstacles, to the finest that is required for smooth paths, the iteration can be made to converge in a time proportional to the number, N , of cells in the finest grid. The solution on crude grids is cheap, and is used to initialize the solution on finer grids. This multi-grid technique is described in [7] and is applied to robotic path planning in [6].

Textbook convergence proofs for empty domains, give the total number of computations to be $c \cdot N$, where N is the number of cells in the finest grid and $c \approx 5$ is some small number of iterations at each grid size. That convergence speed relies on being able to set the crudest grid cell size equal to the entire domain for the crudest solution. However, we have found that in a domain with obstacles, the number of needed iterations is given by $c \cdot (\text{path length}) / (\text{path width})$, since path-width between obstacles, limits the largest cell size of the crudest grid that still preserves the topology of the computed paths. Further computational speed up comes from iterating the Laplacian solution on a variable size grid. The original 512x512 pixel map is converted to a quadtree map, where only regions that contain obstacles are let at the fine $(0.25\text{m})^2$ resolution. Larger regions that contain no obstacles are gridded with much coarser resolution as shown in Figure 9.

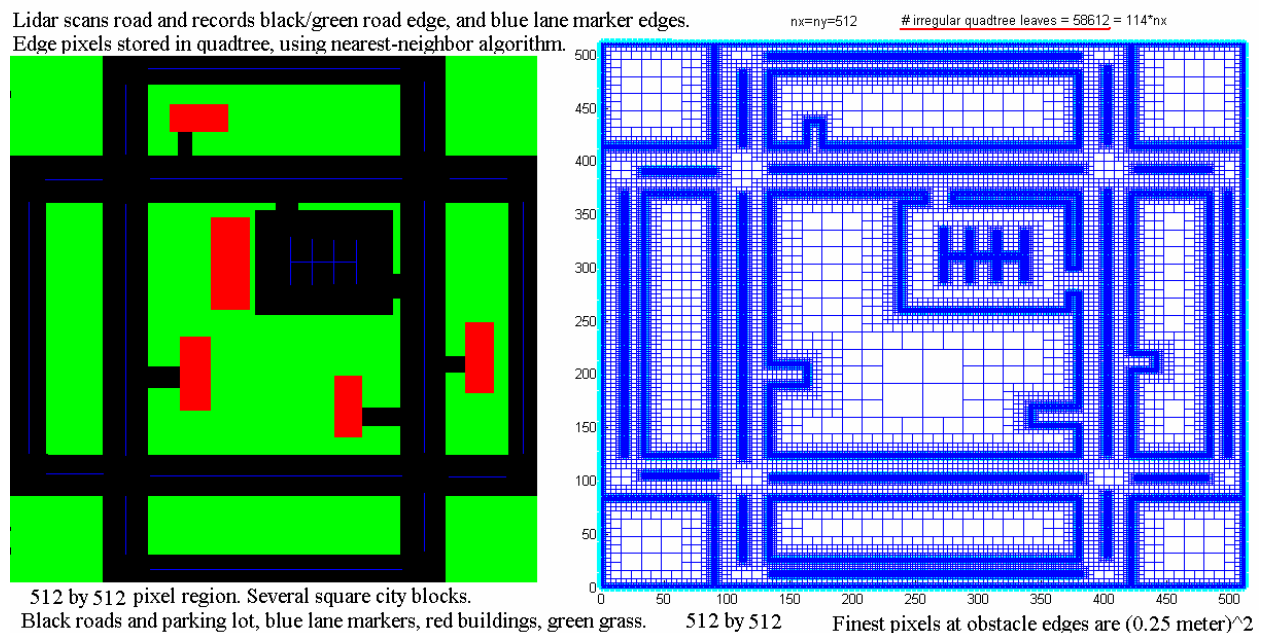


Figure 9 – Short Term Planner Grid Example

Once the potential is computed, path points are computed by evaluating the gradient of the potential at our vehicle's current position, taking a 0.25 m step in the direction of the gradient, then computing the gradient at that new location. This process is repeated for a total of 600 gradient steps, $(600 \text{ steps}) \cdot (0.25 \text{ m/step}) = 150 \text{ meter path length}$. These gradient computations at 600 points are very cheap compared to the 50 potential iterations on $512 \cdot 512$ pixels.

Dealing with structured environments and intersection logic: Various maneuvers such as lane changing, passing, n-point turns etcetera are described in the Path Replanning section, while intersection logic is described briefly here. When the XAV-250 arrives at a stop sign at an intersection, the other intersection points will be observed to determine which have queued traffic at them. The XAV-250 will wait for one vehicle to leave each of these queues and for the intersection to clear before proceeding. If a vehicle has not left its queue for 10 seconds after it could have, that vehicle will be considered stalled and the XAV-250 will proceed without waiting for it. See the following logic and figure 10 for details:

Assumptions: RNDF and LTP will define lanes and stop sign location; Unseen lanes are considered empty - normal driving rules used to deal with threats.

Let $N(i,j)$ represent intersection i , stop sign j (where $j = 0$ is the intersection itself)
 Let set K be such that there are vehicle(s) at stop signs $N(i,k)$ for k in K , j not in K

If [(truck arrives at $N(i,j)$, $j > 0$, at time t)
 AND (at time = t there is set K)
 AND (after that, for each k of K , vehicle either leaves $N(i,k)$ and enters $N(i,0)$ OR forfeits turn after 10 seconds)
 AND (after that, $N(i,0)$ is empty OR 10 second timeout occurs)]
 Then (truck can go after 1 second AND must go within 10 seconds)



Figure 10 – Intersection Logic Details

Speed Planner: The Laplacian approach to planning obstacle-free paths yields a smooth path by going down the gradient of potentials, but does not inherently indicate how fast such a path should be traversed. Therefore, the primary function of the speed planner is to process all available information about the planned path, the environment, constraints on vehicle performance, competition rules etc. in order to compute an appropriate target value for vehicle

speed. Secondary functions of the speed planner include (de-)activating the turn signal on the vehicle and re-packaging/matching specific points along the planned path with speed recommendations so that a desired spacing can be maintained between consecutive points in the path the vehicle is commanded to follow.

The primary function is accomplished by calculating the speed $v_{(.)}$ required to meet the following constraints (all are not necessarily active simultaneously):

- Curvature of the planned path (v_a)
- Need for range to closest obstacle to be greater than stopping distance at current speed and target deceleration (v_b)
- Undulation factor of the road ahead - i.e. how ‘bumpy’ the road is (v_c)
- Maximum safe speed the vehicle can travel at under current conditions (v_d)
- Road speed limit (v_e)
- Area vehicle operates in - e.g. travel vs. safety areas (v_f)
- Minimum turn radius of the vehicle (v_g)
- Need to stop at an intersection or stop sign (v_h)
- Need to trade-off enforcing the lower speed limits specified in the competition rules against moving slow enough to stop for and/or avoid obstacles (v_i)
- Need to maintain specified inter-vehicle spacing when passing (v_j)

This computation is carried out for each point along the planned path. Accordingly, if the total number of points on the path is N , the speed recommendation for the k -th point ($k = 1 \dots N$) is:

$$v_{plan-k} = \text{sgn}(dir) \min(v_a, v_b, v_c, v_d, v_e, v_f, v_g, v_h, v_i, v_j)$$

Where $\text{sgn}(dir)$ is ± 1 depending on which direction one plans for the truck (forward is ‘+’ and reverse is ‘-’).

In order to set the speed recommendation (v_b), the speed planner searches a rectangular region in front of the truck for obstacles. The length of this region depends on how fast the truck is moving while its width is currently selected to be about the same as the lane width. This same method is used to deal with both moving obstacles directly in front of the truck as well as those that are not currently in front of the truck but will be at a later time. These cases are illustrated in Figure 11. For an uncontrolled intersection, as shown on the left side of the figure, information from the track database is used to project a speed-dependent rectangular region in the direction of motion of all moving obstacles (as done for the car moving in from the left). If a collision is projected between the XAV-250 and a moving obstacle, the speed planner slows the truck down. The right side of the figure gives an example of how the speed planner slows the truck down for moving obstacles in the same lane as the XAV-250. Obstacles in the opposite lane do not impact decisions since the speed planner only searches the rectangular region projected in the direction of motion of the XAV-250 for obstacles.

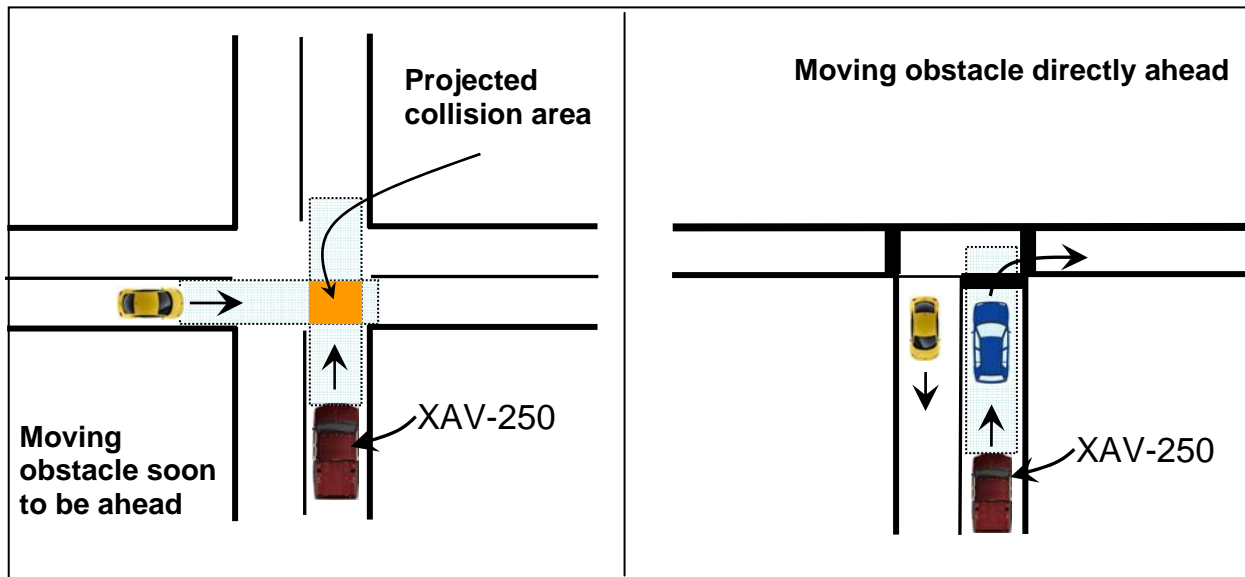


Figure 11 – Dealing with moving obstacles

Path Replanning: The key to short term planning is convergence to the goal point that the vehicle is heading towards; this is the STP nominal mode. In the case of non-convergence to the goal point, the STP will retry to converge to the goal point increasing the number of iterations used for computing the potential in the likelihood that insufficient number of iterations was used. In the case of non-convergence, the following strategies are considered:

- **Lane Change** – a lane change is considered if an un-occupied lane in same direction is available. This strategy requires reconfiguration of the lane management from the nominal STP mode.
- **Lane Passing** – a lane passing is considered if an un-occupied lane in opposite direction is available, i.e. legal lane is identified via the RNDP. Same as the lane change strategy, the lane passing strategy will also involve reconfiguration of the lane management from the nominal STP mode.
- **Direction Change** – a change direction is considered if the goal point we are heading to is relatively close and in the opposite direction of travel.
- **N-PT Turn** - an N-PT turn is considered if the goal point is relatively far and in opposite direction of travel.
- **Recover Roadway** – this action is considered if it was determined that vehicle is on a non-drivable surface and needs to return back to a drivable road.
- **Help from LTP** – this strategy is implemented if all others have failed to converge to the goal point. The short term planner notifies the long term planner which provides the short term planner with an alternative waypoint.

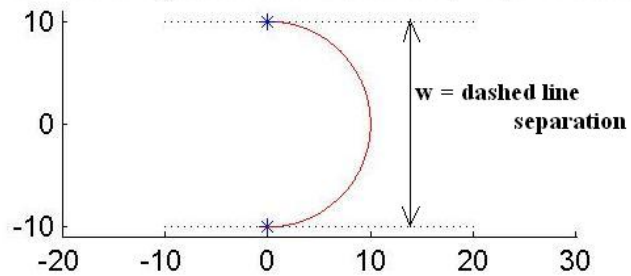
Out of these options, the lane change and passing alternatives can be considered the minimal options to reach convergence while recover roadway and help from LTP are significant actions to obtain STP convergence. Direction change and N-pt turns, however, pose an interesting challenge which fully encompasses the XAV-250's dynamic capability. To execute either, these

options, a full understanding of the roadway and vehicle turning radius is required. Furthermore, it can not be assumed that a simple U-turn will always be possible; many times an N-point turn will be required. To insure success the IVS team has created an algorithmic solution to generate the proper N-point turn whenever required.

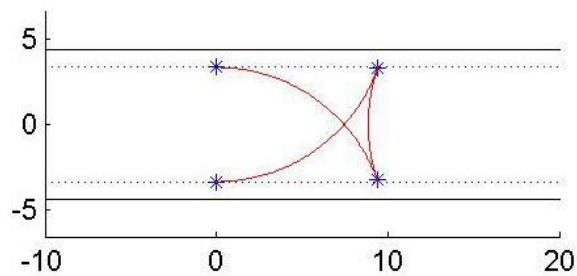
Simply stated, the algorithm functions as follows: Given a path of width w and a vehicle with turning radius R , a sequence of arcs of radius R is constructed that allows the vehicle to travel in alternating forward/reverse directions to achieve a total heading change of π radians. The solution is constructed as a sequence of arcs with each arc being tangent to its neighboring arc. The fact that the vehicle is not arbitrarily small, is accounted for by letting $w = (\text{road width}) - (\text{vehicle length})$, where vehicle length can either be the diagonal length of the entire vehicle, or the diagonal wheel base length if the bumpers may extend beyond the curbs. Figure 12 shows a simulation of $N = 1, 3,$ and 5 . Note road width requirements shrink as N increases.

Path following: The goal of path following is to follow a predefined set of path points to a goal point. The given path points and corresponding speed targets are communicated to the path follower module from the short term planner and speed planner modules. The path follower's basic function is to minimize the normal distance to the nearest path segment vector and to orient the vehicle with the correct heading angle. The follower module generates the desired steering commands and speed targets which are then sent to the low level controllers for actuation. The follower generates steering arcs based on a look-ahead goal point target relative to the current vehicle reference frame position. A speed dependant look-ahead point is used to provide accurate control at low speed and stability at higher vehicle speeds. The instantaneous vehicle position, supplied from a blended signal of the IMU and GPS data, provides smooth navigation information for feedback control. A vehicle dynamics model, driven by the state inputs from the IMU to limit the performance envelope of the vehicle, applies the necessary brake pressure and throttle intervention requests.

1-arc turn. $1 * (\text{dashed line separation}) / 2 = (\text{turn radius})$



3-arc turn. $3 * (\text{dashed line separation}) / 2 = (\text{turn radius})$



5-arc turn. $5 * (\text{dashed line separation}) / 2 = (\text{turn radius})$

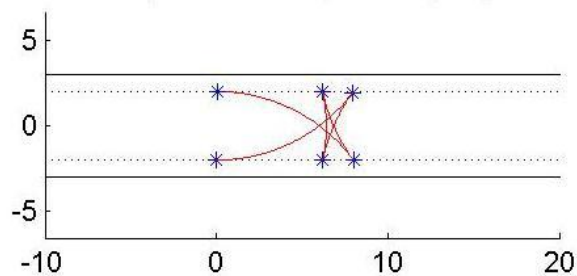


Figure 12 – N-point Turn Algorithm Examples

Mission Control: The purpose of the mission controller (MC) is to encapsulate the high-level logic needed to accomplish a mission and distribute mission information to other components. This type of logic naturally lends itself to a hierarchical state machine (HSM) model. Implementing the MC as an HSM allows reduction of complexity and allows other software components to interact with the MC states at the level of the hierarchy that makes sense to them. Each software component is responsible for producing certain events which can trigger state changes and react to changes in state.

There are two categories of logic which make up the mission controller. The first is logic used to coordinate the execution of all software components. One of the difficulties in a distributed system is how to control startup, shutdown and fault management. To solve this problem we have modeled the desired behavior as states and created events to trigger transitions between those states.

The second category is the logic necessary to control the behavior of each software component. During a mission each software component has a variety of low-level tasks they are responsible for performing. The specific low-level task that should be performed at any given time is dependant on what high-level task the vehicle is currently trying to accomplish. Modeling the high-level tasks as part of the MC allows high level decisions to be made in one place with results distributed.

3. Results and Performance

3.1 Test Approach

The strategy for test site selection follows then same method as our overall testing plan. Testing will start with basic/simple locations and test courses and builds up to more complex locations and tests. The basic site is as simple as a parking lot for stationary evaluation tests. Then next level of the test plan validates simple basic navigation tests on the Ford test track in Dearborn. Upon completion, the testing tasks will increase in difficulty and will move to the Ford's Michigan Proving Grounds (MPG). Again following the stepped approach for testing, there will be multiple courses of increasing complexity. Depending on the testing being performed, the specific test course may change. Another factor being considered in the test plan and site selection is environmental conditions. To address this issue a change of location to Ford's Arizona Proving Ground will be incorporated into the advance level testing. See Figures 13 for an example of a portion of the MPG test facility along with the site visit course layout and Table 4 for an example of the validation plan being implemented – Note multiple XAV-250 approach.



Figure 13 – MPG Vehicle Dynamic Test Area, Precision Steering Course, and Site Visit Course

Test	Location	Resource	Start/End
Subsystem Data Collection	Dearborn	Truck T	03/12/07 03/22/07
Basic Navigation	Dearborn	Truck T	03/23/07 03/30/07
Basic Traffic	Dearborn	Truck T	03/27/07 04/09/07
Site Visit Course – Basic Navigation	MPG CERT, Golden Valley	Truck T Truck A	04/09/07 04/23/07
Site Visit Course – Basic Traffic	MPG CERT, Golden Valley	Truck T Truck A	04/16/07 05/13/07
Site Visit Course - Mission Testing	MPG CERT, Golden Valley	Truck T Truck A	05/01/07 05/26/07
DARPA Site Visit Test	MPG VDA	Truck T	06/25/07
Steering Precision Course – Adv. Navigation	MPG SPC	Truck T Truck A	04/27/07 06/30/07
Steering Precision Course – Adv. Traffic	MPG SPC	Truck T Truck A	06/18/07 07/04/07
Steering Precision Course - Mission Testing	MPG SPC	Truck T Truck A	07/05/07 07/31/07
Other Test Course - Mission Testing	APG	Truck T Truck A	08/02/07 08/31/07
Other Test Course - Mission Testing	Old Base	Truck T Truck A	09/03/07 09/30/07

Table 4 – IVS Validation Plan

3.2 Sensing & Mapping Results

Figure 14 shows a screen capture of the Fused Map viewing client. It is a visualization of the same data presented to the planner. In this view, the Velodyne has populated the map when the system starts in the aisle of a parking lot. Shades of green indicate confidence of drivable terrain, while red indicates non-traversable. White pixels indicate where the map has not received data (unseen). The rings are an artifact of the Velodyne sensor. Each pixel is the classification of a 0.25 meter x 0.25 meter patch of land. World coordinates are given for the North West corner of the map. The user can query the values of individual cells of the map by clicking on them. The user can also overlay a grid or zoom in on the map.

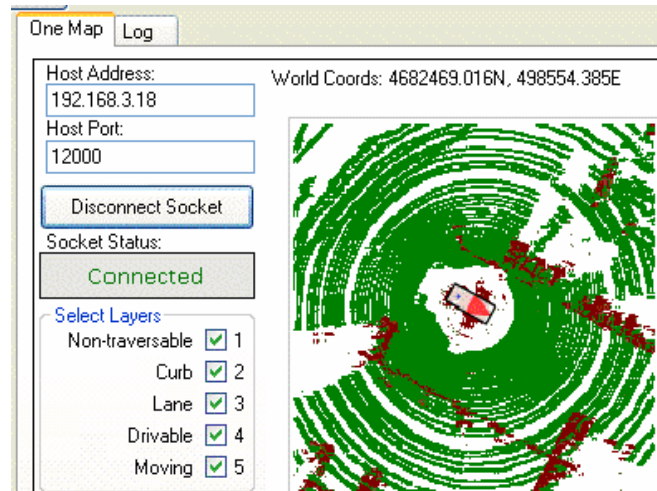


Figure 14 – Fused map presented to the planner.

Figure 15 is a composite image which illustrates how data captured from a sensor is passed to TOM in our system architecture. The photo on the left is a typical road segment at our test facilities. The center image depicts a single data frame captured from the Velodyne HD-64E, with each LIDAR beam color-coded by the intensity of the return. Ten million range measurements per second are transformed into Earth frame coordinates, used to map the ground

plane, and determine where the XAV-250 can physically drive. This “driveability” index (one of several attribute features passed to TOM) is shown on the right for each 0.25m x 0.25m cell surrounding the vehicle.

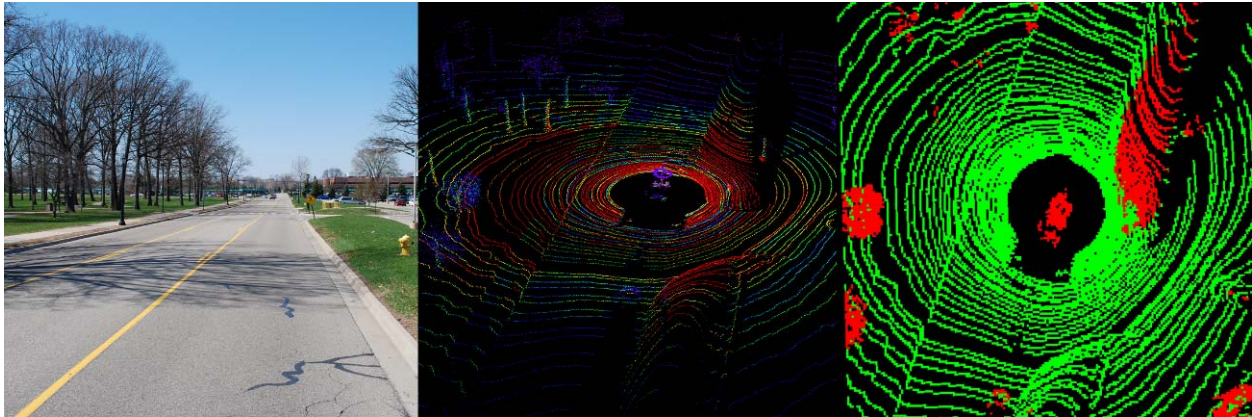


Figure 15 – Sensor data transformation to TOM

Figure 16 illustrates two alpha-version feature-extraction applications derived from the Cybernet vision system installed on the XAV-250. The algorithm running on the left identifies lane markings and other sharp-edged features, such as curbs and sidewalks. The algorithm on the right searches for traversable surfaces, based upon contrasts in color and texture, and is heavily influenced by the sample roadway immediately in front of the vehicle (yellow box).

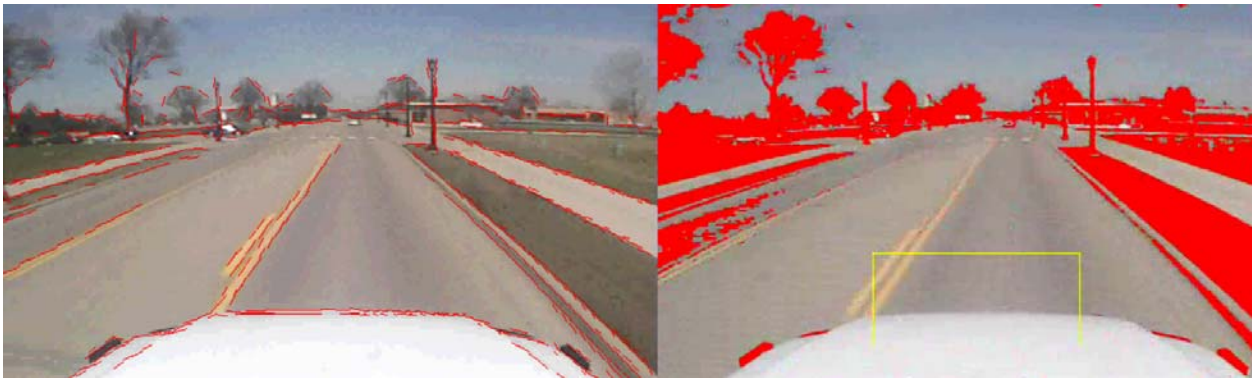


Figure 16 – Vision System Roadway and Traversability detection example.

Figure 17 is a screen shot utilized to visualize Reigl data. Points are colored on a scale from blue (low reflectivity), green (medium reflectivity), and red (high reflectivity). Red, Green and Yellow solid lines are cues to user to indicate left and right, and middle. White is a plot of the navigation position. The Reigl data is plotted in earth coordinates. Figure 18 is another example of captured Reigl data. In this case, a road line, fences, and lamp posts are visible. The green blobs are other vehicles.

Figure 19 is an example of urban driving data acquired from a fused radar and vision application. The data shown here is taken on a surrogate vehicle utilizing a production intent system comprised of a single radar and camera. The XAV-250 implementation utilizes three radars, effectively increasing the field of view for forward object detection by a factor of three,

integrated with the vision system. This “semi-fused” data is further fused into TOM. The XAV-250 employs new hardware and algorithms to process, pass, and fuse radar data from all twelve long range and four short range units to TOM. Development of this system is still underway, but all twelve sensors will be feeding TOM before the Milestone 2 Site Visit.

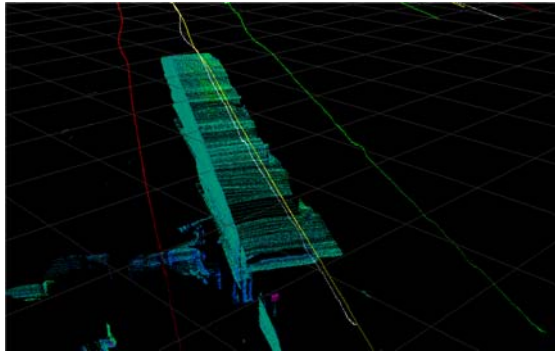


Figure 17 - Reigl intensity data. Stop sign at the bottom.

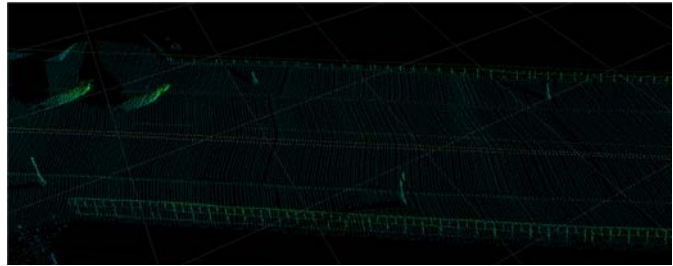


Figure 18 - Reigl data with road lines, fences and lamp posts. “Blobs” are cars.



Figure 19 – Radar/Vision fusion with lane detection and closest in-path target

3.3 Autonomous Driving Results

Path Planning: Figure 20 shows a sequence of 9 frames from a movie of the simulated vehicle driving autonomously. The vehicle starts in one parking lot in frame 1, travels through a sequence of seven waypoints and ends up in another parking lot in frame 9. At each instant, the path planner has an entire path computed to the next waypoint, but only a short part of that planned path is displayed as a red curve. The thicker black curve is where the vehicle has already traveled. Turning-radius constraints are represented by blue semicircles on each side of the vehicle. Lane management is implemented by blocking lanes that the vehicle is not supposed to be in, for example, left lanes are blocked. A small region near the vehicle has no lane restrictions, allowing the vehicle to cross yellow lines when needed, e.g. when turning left exiting parking lots (frame 1) and when making U-turns (frames 3 and 4). The final waypoint is a parking space, which is given a heading constraint to force the vehicle to enter from the desired direction. The short-term path planner planned a path through the round-about with only one waypoint in the roundabout. Lane tracking accuracies were good, except for small deviations while passing through the intersection. These small deviations in the intersection can be avoided by having the short-term planner extend the painted lane-marker lines through the intersection

along the planned path as shown in the Figure 21. The path planner devises routes around all fixed obstacles, while speed planner slows the vehicle down when planned path gets near moving obstacles.

Autonomous Navigation Simulation and Validation: The simulation environment, as shown in Figure 22, was developed to aid the control algorithm development phase. Several autonomous control techniques were investigated and simulated in a real-time environment. A MATLAB path follower simulation was utilized to test for optimal path tracking functionality. This environment is used for testing path following control algorithms, replay unique path scenarios, and provide a real-time interface during vehicle testing. Both simulation and actual vehicle tests have shown very good path tracking results. A vehicle dynamic model is used to drive positional information for laboratory Software-In-the-Loop (SIL) development. This enables the ability to replay and simulate collected data to aid software algorithm development of this and other functional modules.

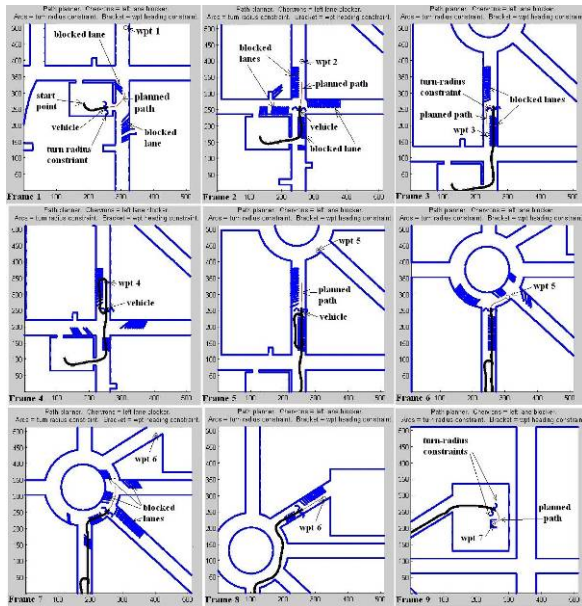


Figure 20 – Simulated path planning



Figure 21 - Extending painted lane markers though intersection, along planned path

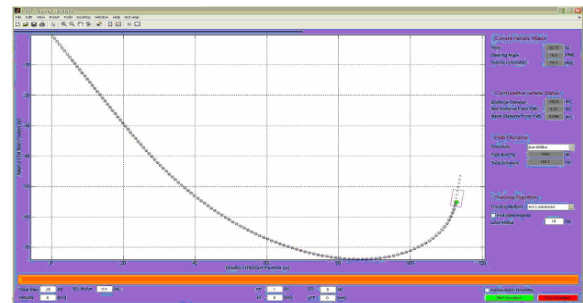


Figure 22 – Autonomous Navigation Simulation Environment

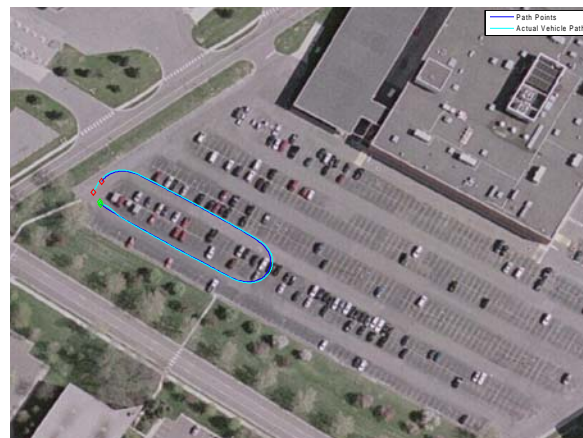


Figure 23 – Example of Autonomous Navigation Validation Testing

Autonomous Vehicle Control Validation: Many path tracking scenarios were tested to measure the performance of the autonomous control algorithms. See Figure 23 for an example of a simple course used during the validation testing of the path follower control algorithms. Vehicle testing showed very tight path tracking tolerance and smooth control.

4 Acknowledgements and References

4.1 Corporate Contacts

James McBride
DARPA Team Lead
jmcbride@ford.com

Alan Touchberry
Principle Investigator
alan.b.touchberry@honeywell.com

Douglas Turner
System Lead
doug.d.turner@delphi.com

4.2 IVS Team Contributors

M. Agrawal, W. Au, S. Dajani-Brown, D. Churchill, G. Collins, S. Cooper, M. Elgersma, R. Eustice, K. Fregene, A. Griesbach, D. Haanpaa, J. Higgins, J. Ivan, A. Levandowski, J. McBride, B. Mohr, R. Olson, S. Pratt, T. Reusser, D. Rhode, M. Rupp, J. Rye, D. Spiro, H. Szwarski, A. Touchberry, D. Turner, B. VanVoorst, R. Weaver, T. Yopp

4.3 References

- [1] Intelligent Vehicle Systems (IVS) Team Proposal for the DARPA Urban Challenge, Proposal for BAA 06-36, submitted by Honeywell Laboratories, June 26, 2006.
- [2] Intelligent Vehicle Safety Technologies 1 – Final Technical Report, submitted by William Klarquist and James McBride, August 29, 2005, archived on DARPA’s website at: <http://www.darpa.mil/grandchallenge05/TechPapers/IVST.pdf>
- [3] E. W. DIJKSTRA, “A note on two problems in connexion with graphs,” Numer. Math. 1, pp269-271, 1959
- [4] C. I. Connolly, and J. B. Burns, “Path planning using Laplace’s equation,” Proc of IEEE Int. Conference on Robotics and Automation, 1990.
- [5] C. I. Connolly, and R. A. Grupen, “The Application of Harmonic Functions to Robotics,” J. of Robotic Systems, 10(7), pp931-946, 1993.
- [6] Z. X. Li, and T. D. Bui, “Robot Path Planning Using Fluid Model,” Journal of Intelligent and Robotic Systems, v. 21, pp29-50, 1998.
- [7] Valavinis, Herbert, Kollura and Tsourveloudis, “Mobile Robot Navigation in 2-D Dynamic Environments Using an Electrostatic Potential Field,” IEEE Trans Sys and Cybernetics, v 30, 2002.
- [8] M. Jackson, V. Sharma, C. Haissig and M. Elgersma, “Airborne technology for distributed air traffic management,” European Journal of Control, v 11, pp 1-14, December 2005.