A Perspective on Emerging Automotive Safety Applications, Derived from Lessons Learned through Participation in the DARPA Grand Challenges

J. R. McBride, J. C. Ivan, D. S. Rhode, J. D. Rupp, and M. Y. Rupp
Ford Motor Company—Research and Advanced Engineering
Research and Innovation Center
2101 Village Road
Dearborn, Michigan 48121
e-mail: jmcbride@ford.com

J. D. Higgins and D. D. Turner
Delphi Corporation
5725 Delphi Drive
Troy, Michigan 48098

R. M. Eustice
University of Michigan
2600 Draper Drive
Ann Arbor, Michigan 48109

Received 21 February 2008; accepted 27 August 2008

This paper reports on various aspects of the Intelligent Vehicle Systems (IVS) team's involvement in the recent 2007 DARPA Urban Challenge, wherein our platform, the autonomous "XAV-250," competed as one of the 11 finalists qualifying for the event. We provide a candid discussion of the hardware and software design process that led to our team's entry, along with lessons learned at this event and derived from participation in the two previous Grand Challenges. In addition, we give an overview of our vision-, radar-, and LIDAR-based perceptual sensing suite, its fusion with a military-grade inertial navigation package, and the map-based control and planning architectures used leading up to and during the event. The underlying theme of this article is to elucidate how the development of future automotive safety systems can potentially be accelerated by tackling the technological challenges of autonomous ground vehicle robotics. Of interest, we will discuss how a production manufacturing mindset imposes a unique set of constraints upon approaching the problem and how this worked for and against us, given the very compressed timeline of the contests. © 2008 Wiley Periodicals, Inc.

1. INTRODUCTION AND OVERVIEW

The narrative presented in this paper is derived from experiences gained by the various authors through their participation in the series of DARPA Grand Challenges (DGC1, DGC2, DGC3, also variously referred to by year, etc.). During the inaugural edition of the Grand Challenge, two of the authors worked as volunteers for DARPA and performed tasks such as reviewing the participating teams' technical proposals, conducting safety evaluations of the entrants' robots, and serving as chase vehicle crew members. In the process, they were able to gain insight into the state of the art of ground vehicle robotics and established an invaluable array of personal contacts, ranging from government officials to university researchers to corporate manufacturers. By the end of the event, the consensus was that many of the problems that were vexing ground vehicle robotics were of the same technological nature as the hurdles impeding the rapid development of advanced safety systems in the automotive industry.

With this conclusion in mind, in 2005 a collaborative effort was formed between Delphi, Ford, Honeywell, and PercepTek, with the goal of conducting joint research directed toward the creation of safe and robust intelligent ground vehicle systems for production-intent commercial and military applications. Participating under the collaborative name "Intelligent Vehicle Safety Technologies," or IVST, they entered the 2005 DGC, fielding the autonomous Ford F-250 dubbed the "Desert Tortoise." In their first attempt at ground vehicle robotics, the team impressively made it all the way through the selection process, earning the fifth starting pole position at the finals (IVST, 2005a, 2005b).

Upon announcement of the 2007 DARPA Urban Challenge (DUC/DGC3), three core members from the previous effort formed a new collaboration, this time known as "Intelligent Vehicle Systems" (IVS), which was initially composed primarily of employees from Delphi, Ford, and Honeywell. However, as the project evolved, the collaboration expanded to include contributions from a variety of external organizations, including Cybernet, the University of Michigan (UMich), and the Massachusetts Institute of Technology (MIT). The team once again used the F-250 as its base platform, but significant modifications were made to the sensing suite and computing architecture. The "XAV-250" (eXperimental Autonomous Vehicle-250) as it was now called, might seem to be a surprising choice for an urban driving environment, given its size and mass, and particularly given that its long wheel base resulted in a turning radius in excess of 8 m (Figure 1). However, the truck had already proved its merits in the DGC2 and moreover precisely represents the type of vehicle that would likely be employed in a realistic autonomous mission capable of carrying any sort of significant cargo. If humans could keep it centered in their lane on the road, we saw no reason to believe that computers could not do the same. The team was one of only 11 to achieve the finals and one of only six to have appeared consecutively in the past two DGC final events.

Figure 1. The XAV-250 poses for the camera at the 2007 NQE.

It should be noted that the striking difference between the first Grand Challenges and the latest is the dynamic nature of the urban event, which introduced moving targets, intermittently blocked pathways, and extended navigation in regions of denied-global positioning system (GPS) reception. The team's intent was to build upon the lessons learned from the Desert Tortoise and specifically to evolve from a reactive, arbiter-based methodology to an architecture that dynamically senses the three-dimensional (3D) world about the vehicle and performs complex, real-time path planning through a dense global obstacle map populated from multiple types of fused sensor inputs.

The remainder of this article is organized as follows. In Section 2 we parallel how advances in ground robotics research can lead to advances in automotive active safety applications, thereby

motivating our corporate research participation in the DARPA challenges. In Section 3 we provide an overview of the hardware and software architecture in place prior to our post-DARPA-site-visit redesign, which entailed a migration to the MIT software architecture (described in Section 4). Section 5 reviews our performance during the National Qualifying Event (NQE) and Finals, and Section 6 offers a post-DGC reflection on engineering lessons learned through our participation. Finally, Section 7 provides some concluding remarks.

2. THE CONNECTION BETWEEN ROBOTICS RESEARCH AND AUTOMOTIVE SAFETY

In common with participants in other long-term, visionary research projects, we are often asked to explain the relevance of our work. The question of "just how does playing with robots deliver benefits to the company, its stakeholders, and customers?" is not uncommon. Our opinion is that by solving the complex challenges required to make a vehicle capable of autonomous driving, we will simultaneously enable and accelerate the development of technologies that will eventually be found on future accident mitigation and avoidance systems throughout the industry.

As we will discuss later in the text, we do not anticipate that mass-production, fully autonomous automobiles (*autonomobiles*) will appear on the market anytime in the foreseeable near future. However, we do envision a steady and systematic progression of improvements in automotive convenience, assistance, and safety features, wherein the vehicle becomes capable of assuming an ever-increasing role in the shared human–machine driving experience. The general trend in the automotive industry is evolving from merely providing value-added driver information to assisting with mundane driving tasks, warning and helping the driver to choose a safe solution when presented with potential hazards, and ultimately assuming increasing control when it becomes indisputably determined that the driver is incapable of avoiding or mitigating an imminent accident on his own accord.

The bulleted list below provides a generic overview of several of the upcoming features being touted by automotive OEMs (original equipment manufacturers) and which will undoubtedly benefit from many of the algorithms derived from autonomous vehicle research. Roughly speaking, the first four examples fall under the categories of braking, throttle, steering, and vehicle dynamics. The end point for each of these evolutionary systems converges with the others to provide a comprehensive collision mitigation strategy. The remaining items on the list involve infrastructure and human–machine interactions.

- antilock brake systems (ABS), imminent collision warning, panic brake assist, collision mitigation by braking (CMbB)
- cruise control, adaptive cruise control (ACC), ACC plus stop-and-go capability, urban cruise control (UCC; recognizes stop signs and traffic signals)
- lane departure warning (LDW), lane keeping assistance (LKA), electronic power assist steering (EPAS), adaptive front steering (AFS), active steer (EPAS + AFS), emergency lane assist (ELA)
- traction control, electronic stability control (ESC), roll stability control (RSC), active suspension
- integration of precrash sensing with occupant protection equipment (airbags, seat belts, pretensioners)
- collision mitigation by integrated braking, throttle, steering, and vehicle dynamics control
- vehicle-to-vehicle and infrastructure integration (VII/V-V), intelligent vehicle highway systems (IVHS)
- blind spot detection, pedestrian detection, parking assistance, night vision
- driver drowsiness and distraction monitoring
- total accident avoidance/autonomous vehicle control

2.1. The magnitude of the problem

Although we are frequently reminded of the annual impact to society caused by vehicular accidents (NHTSA, 2005), it is nevertheless worth summarizing the statistics for the United States (worldwide statistics, although less well documented, are more than an order of magnitude worse):

- 43,000 deaths
- 2.7 million injuries
- $230 billion in economic costs

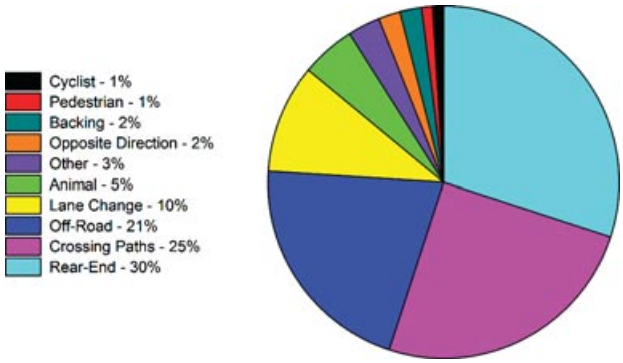

Figure 2. Distribution of light vehicle crashes by type (Volpe, 2007).

To put this in perspective, this is roughly the equivalent of one major airline crash per day. Although we somehow seem to accept the inevitability of traffic accidents, it is quite unlikely that the American public would board airplanes if they knew one of them would crash every day, killing all occupants aboard.

Figure 2 shows a distribution of light vehicle crashes by type, as compiled in "Pre-Crash Scenario Typology for Crash Avoidance Research" (Volpe, 2007). Whereas many of these scenarios are well understood and safety systems are either in place or under development to address them, a noteworthy percentage of the crash scenarios are presently not well covered by emerging near- to midterm automotive safety technologies. To expand a bit upon this point, the rear-end collision prevention scenario is the closest to production deployment, because the bulk of the hardware and algorithms required to achieve this application are logical extensions of ACC systems, which are presently available on selected OEM models. Whereas the off-road and crossing path scenarios, which comprise the largest share of unresolved safety challenges, could benefit greatly from the results of the DGC research efforts, solutions to these issues could be developed even faster if technologies that were explicitly omitted from the DGCs were incorporated, notably vehicle-to-vehicle and vehicle-to-infrastructure communications and enhanced roadway maps replete with ample metadata. We conclude that there are ample research opportunities for delivering improvements in automotive safety to members of our society, and it can be argued that these gaps in coverage obviously represent scenarios that require a greater degree of situational awareness of the world around the vehicle, advanced sensors, and more sophisticated algorithms.

2.2. How Does the Reliability of a Present-Day Robot Compare with That of a Human?

Before we can make any sort of intelligent comments about the impending appearance of the autonomobiles that popular science writers have been promising for the past half century (e.g., Figure 3) or, for that matter, any number of the advanced active safety features enumerated in the bulleted list above, we need to have a rough idea of how the performance of present-day autonomous vehicles compares with that of human drivers. Given that the primary aim of this paper is not to scientifically analyze robotic

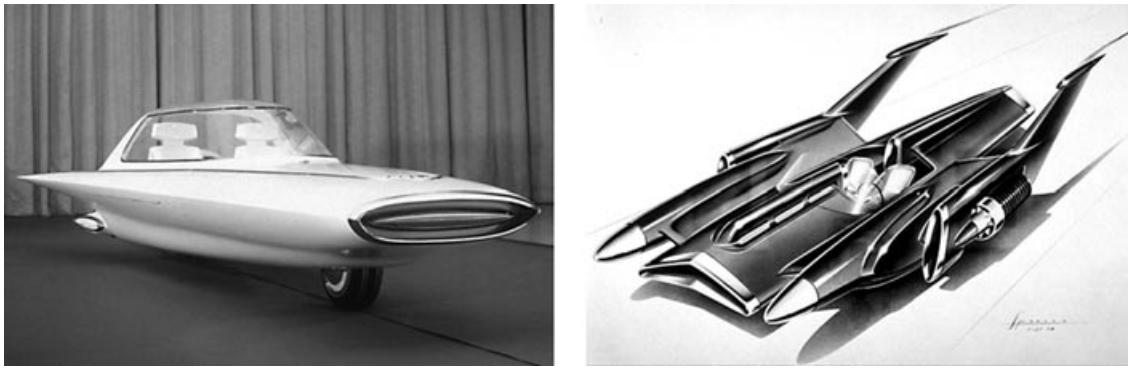

Figure 3. Yesterday's "cars of tomorrow." A couple of fairly typical concept cars from the dawning of the Space Age: photograph of the 1961 Ford Gyron (left) and preproduction sketch of the 1954 Ford FX-Atmos (right).

safety but rather to discuss our experiences at the 2007 DARPA Urban Challenge, we present instead a "back-of-the-envelope" calculation, which serves to initiate the discussions to follow.

For small distances, we can assume that the probability of a "failure" is approximately given by

$$P_f \approx \alpha\, \delta x,$$

where α is the mean failure rate per distance and δx is the incremental distance traveled. Conversely, the probability of "success" is given by

$$P_s = 1 - P_f.$$

If we wish to repeatedly achieve success and note that the total distance traveled is simply $x = n\, \delta x$, where n is the number of path segments (i.e., trials), then the overall probability becomes

$$P_s = (1 - P_f)^n = \lim_{n\to\infty} \left(1 - \frac{\alpha x}{n}\right)^n = e^{-\alpha x}.$$

Of course, this derivation makes some simplifying assumptions, most notably that the events are randomly distributed and independent of one another so that they can be modeled as a Poisson process. If, for example, our vehicle operated flawlessly except whenever a train passed by, this equation would undoubtedly fail to hold. Nevertheless, it makes a good starting point for a provocative discussion.

If we examine the most recently published data from NHTSA's "Traffic Safety Facts 2005" (NHTSA, 2005), we observe that there are roughly

3.0×10^{12} miles driven per year, and
2.0×10^{8} registered drivers, translating to
1.5×10^{4} average annual miles driven per person.

We also find that there are

14.5 fatalities per billion miles, and
900 injuries per billion miles.

Expressed in another manner, on average this is

68.8 mean million miles between fatality, and
1.1 mean million miles between injury.

Using these NHSTA statistics and our derived equation, we can estimate the lifetime odds of suffering a vehicular injury or fatality. Given that life expectancy, at birth, for a middle-aged American was roughly 68 years (it is longer for children born today), let us for the sake of simplicity assume that the average person drives for 50 years of his lifetime. We now find that

$$P_{\text{injury}} \approx 1 - e^{-(9.0\times10^{-7} i/\text{mi})(1.5\times10^{4}\text{mi/yr})(50\text{ yr})} = 49.1\%,$$

$$P_{\text{fatality}} \approx 1 - e^{-(1.45\times10^{-8} f/\text{mi})(1.5\times10^{4}\text{mi/yr})(50\text{ yr})} = 1.1\%.$$

Whereas most government sources state that the lifetime odds of being involved in a "serious" or "major" vehicle accident are about 1 in 3, they do not uniformly define what the metrics are for this categorization, nor do they comment on minor injuries. Fatalities, on the other hand, are not ambiguous, and numerous sources such as the National Safety Council (2004) put the lifetime odds of death due to injury in a vehicular accident at 1 in 84, or 1.2%. This is in excellent agreement with our simple estimation.

We can also apply this line of reasoning to the behavior of the robots in the DGCs. Although we did not explicitly acquire statistics on the mean time or distance between failures, particularly because we frequently tested and worked out one behavioral bug at a time, we did on occasion conduct some long, uninterrupted test runs, and during these outings we generally felt that in the absence of grossly anomalous road features, we could probably travel on the order of 100 miles between significant failures. Of course this value is highly variable, being dependent on the type of road conditions being traversed. If one were lane tracking on a freeway, the results would likely be an order of magnitude better than those observed while negotiating dense urban landscapes. Nonetheless, this average value of $\alpha \sim 0.01$ mean failures per mile led one of our team members to speculate that our chances of completing the DGC2 course would be

$$P_{\text{success}} \approx e^{-(0.01\, f/\text{mi})(132\text{ mi})} = 26.7\%.$$

Now what makes this interesting is that we can turn this argument inside-out and consider the implications upon the other vehicles that participated in the finals of the last two events. If we solve for the mean failure rate α, we find

$$\alpha = -\frac{\ln(P_s)}{x}.$$

At the 2005 DGC2, 5 of the 23 finalists successfully finished the 132-mile course, whereas at the 2007 UCE, 6 of the 11 finalists finished (albeit with a few instances of helpful human intervention) a 60-mile course (DARPA, 2008). Let us see what this suggests:

$$\alpha_{\mathrm{DGC2}} = -\frac{\ln(5/23)}{132\,\mathrm{miles}} = 0.012 \text{ mean failures per mile},$$

$$\alpha_{\mathrm{UCE}} = -\frac{\ln(6/11)}{60\,\mathrm{miles}} = 0.010 \text{ mean failures per mile}.$$

Remarkably, the observational values between the two DARPA final events not only agree with one another but also agree with the crude estimate we had formulated for our own vehicle. To be clear, we do not in any manner wish to suggest that the quality of any team's accomplishments was simply a matter of statistical fortune, nor do we want to make any scientific claims with regard to the accuracy of these observations, given the very small statistics and large assumptions made.

However, we do want to put the present-day capabilities of fully autonomous vehicles in perspective with regard to human abilities. In this context, it would appear that humans are four orders of magnitude better in preventing minor accidents and perhaps six orders of magnitude better in avoiding fatal (mission-ending) accidents. Therefore, the undeniable and key message is that the robotics community has abundant challenges yet to be solved before we see the advent of an autonomobile in each of our garages. To be fair, one can make a valid case for very specific applications in which semi-autonomous or autonomous vehicles would excel, either presently or in the near future, but bear in mind that these are far outside the scope of the general automotive driving problem as framed above. Examples of near-term applications would likely include a range of missions from robotic mining operations to platooning of vehicles in highway "road trains."

2.3. Observations Regarding Customer Acceptance and Market Penetration

The availability of a technology or its proven effectivenes is by no means a guarantee that the customer will actually use it. Although present-day attitudes in society have tipped in favor of not only accepting but demanding more safety applications and regulations, nonetheless a sizable portion of the population views the driving experience as something that should be unencumbered by assistance (or perceived intrusions) from the vehicle itself. For some, it is simply a matter of enjoying the freedom and thrill of driving, whereas for others, it can amount to a serious mistrust of technology, especially that which is not under their direct control. The reader will undoubtedly recall the public commotions made over the introduction of ABS, airbags, and electronic stability control. However, once drivers became sufficiently familiar with these features, their concerns tended to subside. On the other hand, it is yet to be seen how the public will react to convenience and safety features that employ semi- or fully autonomous technologies.

To illustrate this point, consider the case of seat belts, arguably one of the simplest and most effective safety technologies invented. Although they were patented at essentially the same time as the automobile was invented (1885), 70 years passed before they were offered as optional equipment on production automobiles. Furthermore, it took a full century, and legislative actions beginning in the mid-1980s, before customers began to use them in any significant numbers (refer to Figure 4). Even at the present time, nearly 20% of Americans still refuse to wear them (NHTSA, 2005).

Another issue confronting new technologies is the speed at which we can expect penetration into the marketplace. For some features, this is not a big concern, whereas for others, the utility of the technology depends on universal acceptance. For example, it does not matter very much if one customer chooses not to purchase a premium sound system, but on the other hand, the entire traffic system would fail to work if headlamps were not a required feature. Many factors enter into how fast a new technology is implemented, including customer acceptance, availability, cost, and regulatory requirements.

For the case of semi- or fully autonomous driving, success will undoubtedly depend on having as many vehicles as possible equipped with these features. It has often been suggested that some portion of roadways, the federal interstate system for example, may in the future be regulated so as to exclude vehicles that do not conform to a uniform set of equipment requirements. Whereas having individual autonomous vehicles on the roadway may improve safety, having an entire fleet on the roadway could also increase vehicular density, improve throughput,

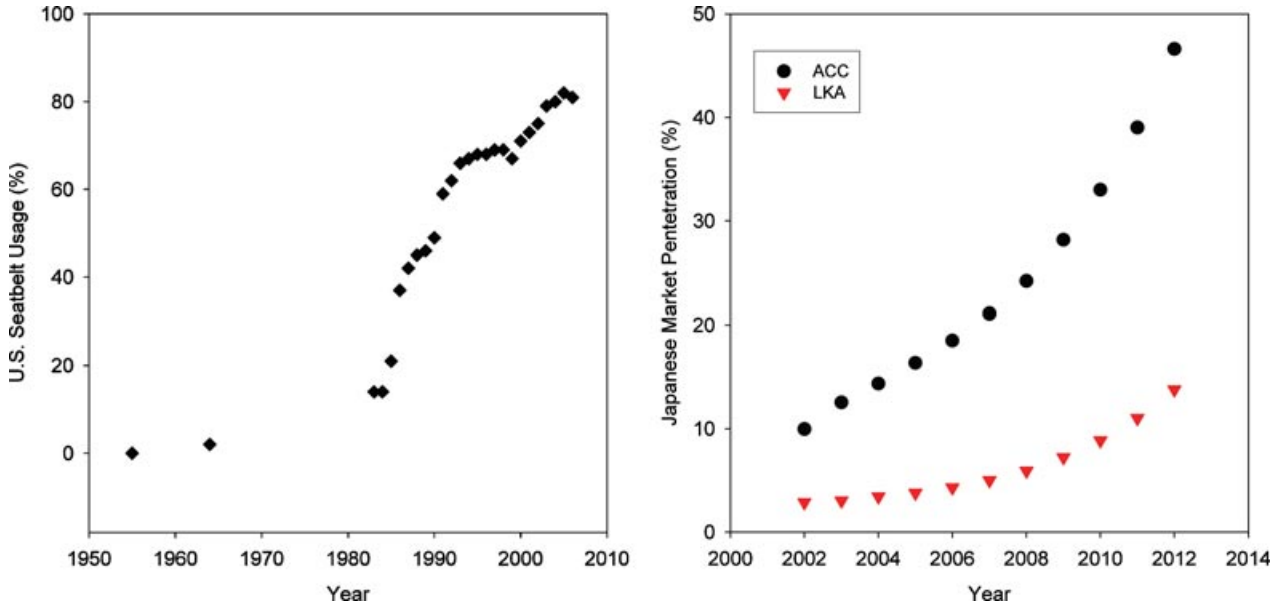

Figure 4. Seat belt usage in the United States as a function of time (left). Note the dramatic rise in acceptance following legislation passed in 1984. Right: Actual and projected Japanese market penetration for ACC and LKA automotive driving features.

and by platooning also reduce drag and improve fuel economy.

Although statistics regarding the customer "take rate" of optional features on a new car are often closely guarded by individual OEMs, Figure 4 (Frost & Sullivan, 2005) also presents a prediction of what "fast adopters," such as the Japanese market, are expected to do with regard to two of the basic robotic building blocks leading toward autonomous operation: ACC and LKA. It should be noted that whereas present-day acceptance for ACC in Japan and parts of Europe is ~15%, in the United States, it is a mere 4%. Perhaps more telling, the market penetration for mature systems that merely provide informational content (not vehicular control), such as GPS-based navigation devices, is still limited to a small subset of new vehicles.

2.4. Concluding Remarks Regarding Implementation of Autonomy in Production Vehicles

Whereas the accident statistics and relative reliability of robots vs. human drivers clearly indicate ample opportunities for future autonomous research solutions, we have also illustrated that a number of factors, including customer acceptance and delivery speed to the marketplace, will ultimately determine when fully autonomous passenger vehicles become a commonplace reality. In this regard, the data refute optimistic projections of production-level autonomobiles by model year (MY) 2015 as some have claimed (military and industrial robotic applications obviously constitute a separate conversation) but rather indicate a slower and continual progression of semi-autonomous driver support and safety features. In this regard, we feel that the field of active safety will ultimately lead the way, with robotics and autonomous vehicle research becoming the key enablers for these future systems.

3. VEHICLE ARCHITECTURE

One of the guiding principles in the design of our entry was to ensure that the hardware, sensors, and algorithms developed not only addressed the mission goals of the DUC but also offered a practical path toward future production integration into other active safety applications. In many phases of the project, this philosophy implied that the wisest choice was to employ production components, whereas there were 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, success depended on choosing the right balance between status quo and innovation. In the following, we provide an overview of our vehicle hardware platform (Section 3.1), our perceptual sensing suite (Section 3.2), and our software architecture (Section 3.3) prior to our DARPA site visit.

3.1. Platform

Based on the success of and lessons learned from the Desert Tortoise used in the DGC2, the 2005 MY Ford F250 again served as the platform for the IVS research efforts. This truck series has been extensively used as a rugged all-purpose workhorse, operating on roads of all sizes and surface conditions around the world, making it an ideal platform for a realistic autonomous mission, commercial or military. We fabricated two identical trucks for the DUC dubbed the "XAV-250"s, models A and T.

Overarching the theme of simplicity stated earlier, safety was always at the forefront of our efforts. Each of the by-wire systems (throttle, brakes, steering, and transmission) operated 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 was further enhanced by the use of redundant, fully independent, remote wireless e-stop systems. When a pause or disable command was initiated, there were multiple means by which it was obeyed.

3.1.1. Vehicle Control Unit

The vehicle control unit (VCU) was implemented using a dSPACE AutoBox rapid control prototyping system and was primarily used to coordinate the by-wire systems that control vehicle speed and steering. Commonly used by automotive OEMs and suppliers for algorithm development, it uses MATLAB/Simulink/Stateflow and Real-Time Workshop for automatic code generation. The VCU algorithms ran a position control algorithm to control steering wheel (hand wheel) position and a speed control algorithm that coordinated the throttle and brake systems, and they also processed launch timing and pause requests.

3.1.2. Throttle-by-Wire

The throttle on the production engine communicates with the engine electronic control unit (ECU) via three voltage signals. A mixing circuit inserted between the throttle pedal and ECU essentially added the by-wire control commands to the outputs from the throttle pedal itself. A dedicated microprocessor with a watchdog timer was used for this interface; nominally the throttle interface commands are issued every 10 ms by the VCU, and if a valid command is not received within 83 ms, the by-wire commands default to zero. This approach did not require modifications to the ECU or throttle pedal and has proven to be simple, safe, and effective.

3.1.3. Brake-by-Wire

Modern active safety systems such as ABS, AdvanceTrac, ESC, and RSC control brake pressures to increase vehicle stability. The XAV-250 has production-representative ESC/RSC hydraulic brake hardware and ECUs, with modified software and hardware containing an additional CAN interface. These systems are regularly used by OEMs and suppliers to develop and tune vehicle stability algorithms. Through this interface, the VCU can command individual brake pressures at each corner of the vehicle with the full capability of the braking system. By using this production-proven hardware, our vehicle robustness and reliability has been very high.

The parking brake was automated to improve safety and durability and to provide redundancy to the main braking system. Keeping with the desire to use production-proven parts, a MY 2000 Lincoln LS electronic parking brake actuator was used to activate the parking brake. To prevent overheating of the brake modulator, the vehicle was shifted into park whenever the vehicle was at zero speed for an extended period of time, and the hydraulic pressure was released on the main brakes. This allowed the brake valves and disks to cool when they were not needed. In the event of an unmanned e-stop disable, the parking brake was actuated by a relay circuit independent of all computing platforms.

3.1.4. Steer-by-Wire

The steering system was actuated by a permanent-magnet dc motor, chain-coupled to the upper steering column. The gear ratio (3.5:1) and inertia of the motor are low enough that manual operation is not

affected when power is removed. By coupling to the upper steering column, the hydraulic power steering system aids the dc motor. For production vehicles, the maximum driver torque needed to command full power steering assist is ~10 N-m. The motor can deliver this torque at approximately 20% of its maximum capacity. To drive it, an off-the-shelf OSMC H-bridge was used to create a high-current 12-V PWM signal. Using a 12-V motor and drive electronics simplified the energy management and switching noise issues within the vehicle.

3.1.5. Shift-by-Wire

Transmission control was accomplished using a linear actuator to pull or push the transmission shift cable. The position was determined using the production sensor located inside the transmission housing. A microprocessor controls the linear actuator and provides the interface to the manual shift selection buttons and VCU. It also senses vehicle speed from the ECU and the vehicle state (run, disable, pause) from the e-stop control panel and affords simple push-button manual operation with appropriate safety interlocks. This approach did not require any modification of the transmission or engine ECU and resulted in a robust actuation that provided the same retrotraverse capability that the Desert Tortoise had in the DGC2.

3.1.6. E-Stop Interface System

The e-stop interface system connects the radio-controlled e-stop system and the various by-wire subsystems to control the operating modes of the vehicle. This system was implemented using automotive relays to increase reliability and reduce complexity. The interface has two modes, development and race, and two states, run and disable. In the development mode, used when a safety driver is in the vehicle, the disable state allows for full manual operation of the vehicle and the run state provides full autonomous operation. In the race mode, the vehicle is unmanned and the by-wire systems conform to DGC rules. Pause requests are handled by the VCU to bring the vehicle to a gradual stop while still obeying steering commands. Communication faults were monitored within dSPACE, and a signal to actuate an e-stop was issued when a fault was detected.

3.1.7. Navigation

Integration and support for the XAV-250 navigation system was provided by Honeywell, as described in previously published reports (IVS, 2006, 2007a, 2007b). The system incorporated a commercially available NovAtel GPS receiver using OmniSTAR HP satellite corrections and was coupled with Honeywell's internally proprietary PING (Prototype Inertial Navigation Gyro) package. The PING has a high degree of flexibility, being capable of using a variety of inertial measurement units (IMUs), and can input various state observations besides GPS, such as wheel speed odometry derived from the vehicle. The navigation algorithms from the PING exported position, attitude, and acceleration and rotation rates at 100 Hz, with the pose information remaining stable and accurate even during GPS outages of up to 20 or 30 min, owing to the high quality of ring laser gyroscopes and accelerometers used in their IMUs.

3.2. Sensors

Changes in the mission specifications, as well as the transition from a desert environment to an urban environment, required that many alterations be made to the sensing philosophy and implementation on the XAV-250. In the earlier DGC2, obstacle detection and path planning were essentially constrained to a narrow corridor closely aligned with a predefined dense set of GPS waypoints. In the DUC, the game became wide open, with sparse waypoints, large urban expanses such as parking lots and intersections, and more important, moving traffic surrounding the vehicle. Clearly this dictated a new solution, capable of sensing the dynamic world surrounding the vehicle but also able to process the wealth of data in a real-time manner.

The introduction of the revolutionary Velodyne HDL-64E LIDAR seemed to be the answer to it all, with 64 laser beams, a huge field of view (FOV), 360 deg in azimuth and 26.5 deg in elevation, 120-m range, and ability to capture one million range points per second at a refresh rate of up to 20 Hz. However, this sensor had yet to be field tested, and with the known limitations associated with previous LIDAR systems, there was certainly some degree of hesitancy to rely too heavily on this device. As such, it was apparent that we would need to provide a redundant set of coverage for the vehicle. This would not only offer

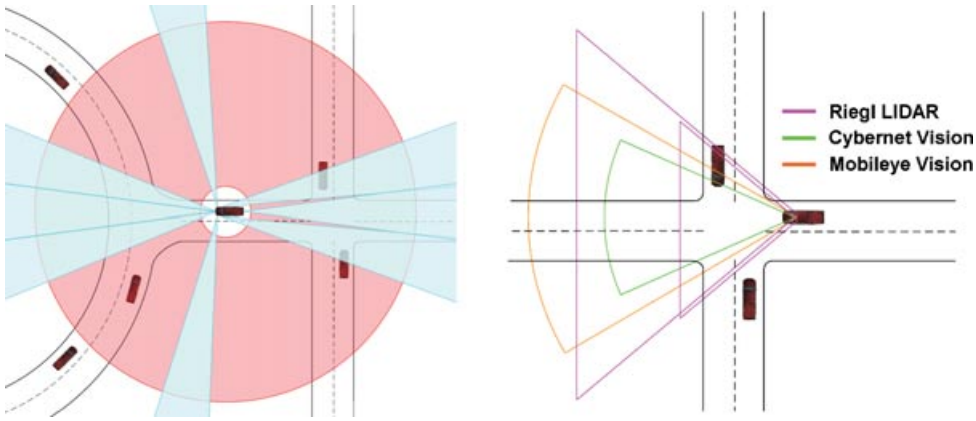

Figure 5. Depiction of the long-range (left) and mid-range (right) sensing FOVs for typical roadways. The light blue triangles depict ACC radars; the shaded pink circle the Velodyne LIDAR. Note that the various depictions are not necessarily drawn to scale.

added confirmation of obstacle detections but would also serve to enhance the robustness of the system to the failure of a single sensor.

Whereas one could take the approach of adding as many sensors as possible to the vehicle (and some teams did in fact do so), this adds an unwieldy burden to computational, electrical power, and thermal management requirements. A better solution was to determine where the XAV-250 most frequently needed to "look" in order to satisfy the required DUC mission maneuvers (Figure 5) and to place sensors of the appropriate modality accordingly to fulfill these needs. We conducted a detailed study to optimize this problem, which included considerations such as the following:

- mission requirements of the DUC (GPS waypoint density, road network geometry, route replanning, merging, passing, stopping at intersections, dealing with intermittent GPS reception)
- infrastructure (intersections, traffic circles, parking lots, dead-end roads)
- roadway design guidelines (line-of-sight requirements, minimum and maximum road grade and curvature, pavement vs. other roadway surfaces)
- highway driving rules (observance of lane markings, intersection precedence, spacing between vehicles)
- closing velocity of traffic (following time and look-ahead requirements)
- potential obstacles to be encountered (vehicles, curbs, buildings, signs, power-line poles, fences, concrete rails, construction obstacles, foliage)

On the basis of our analysis, we initially settled upon a sensing suite that included (in addition to the Velodyne LIDAR) eight Delphi Forewarn ACC radars, four Delphi dual-beam Back Up Aid (BUA) radars, two Cybernet cameras, one Mobileye camera, and two Riegl LIDARs. The overall sensor placement is shown in the truck collage in Figure 6 and is also described in more detail below.

Delphi's 76-GHz ACC radars (Figure 7) are long-range, mechanically scanning radars that have been in production for years on Jaguars and Cadillacs. This radar senses targets at 150 m with a 15-deg FOV and updates its tracking list every 100 ms. A grouping of three of these sensors were coupled together to form a combined 45-deg forward FOV, enabling multilane, curved road coverage. Three more ACC units were strategically placed to also create a wide rearward FOV, with one on the rear bumper and two placed on the front outboard corners of the truck to provide rear and adjacent lane obstacle tracking. Additionally, two radars were mounted in a transverse direction on the front corners to provide coverage of obstacles at intersection zones.

In the forward center FOV, a Mobileye camera was used, primarily to provide confirmation to the ACC radars that targets were vehicles, but also to aid with lane tracking. The other cameras were dedicated

Figure 6. Collage of XAV-250 images revealing key elements of the external hardware. Truck overview (top left); protective frontal exoskeleton with three embedded in-line ACC radars and single centered BUA radar underneath (top center); rear view showing GPS mast on rooftop and protected pair of ACC and BUA radars (top right); side exoskeleton with ACC and BUA radars (bottom left); climate-controlled and shock-isolated box containing computing cluster (bottom center); and LIDAR and vision systems (bottom right), with Velodyne at apex, Riegls left and right, and cameras hidden within the Velodyne tower and behind the windshield glass.

Figure 7. Delphi radars: 76-GHz ACC (left) and 24-GHz BUA (right).

to detect roadway lane markings and curbs and to log visual data. The two Riegl LMS Q120 LIDARs each had an 80-deg FOV with very fine (0.02 deg per step) resolution. They were nominally set at distances of 12.5 and 25 m and used to detect curbs and aid the Velodyne in placing static obstacles in the map. For close-proximity sensing scenarios, e.g., U-turns, backing up, maneuvering into a parking space, passing a close-by vehicle, multiple Delphi BUA radars were used (Figure 8). These sensors have an effective range of ~5 m; however, they do not return azimuth information on the target.

3.3. Software

3.3.1. DGC2: An Arbiter-Based Design

In the prior DARPA Grand Challenge (DGC2), the IVST team employed a situational dependent,

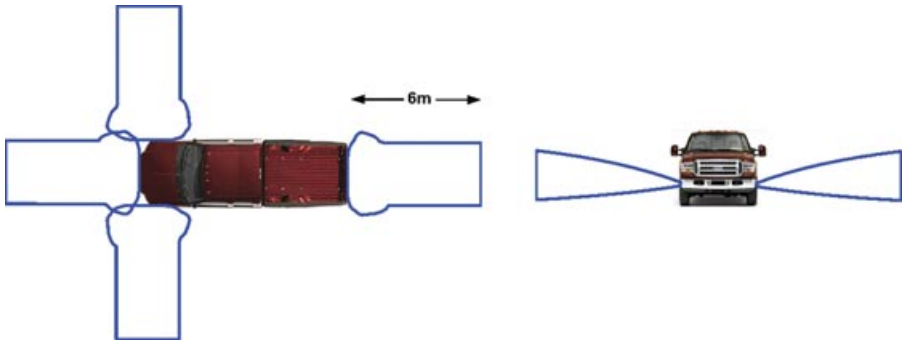

Figure 8. Short-range sensor map illustrating the coverage of the Delphi BUAs.

arbitrating a behavior-based solution (Figure 9) as reported in IVST (2005a, 2005b). The arbiter combined the outputs from the current set of contextual behaviors to produce a resultant steering and speed response. Each positive behavior would send its desired steering response to the arbiter in the form of a vector that represents the full range of steering, with the value at each vector element being the degree to which that specific behavior believes the vehicle should steer. Negative behaviors sent a response over the full steering range that represents steering directions not to go. The arbiter produced a weighted sum of all positive behaviors in which the weight of a behavior is the product of an assigned relative

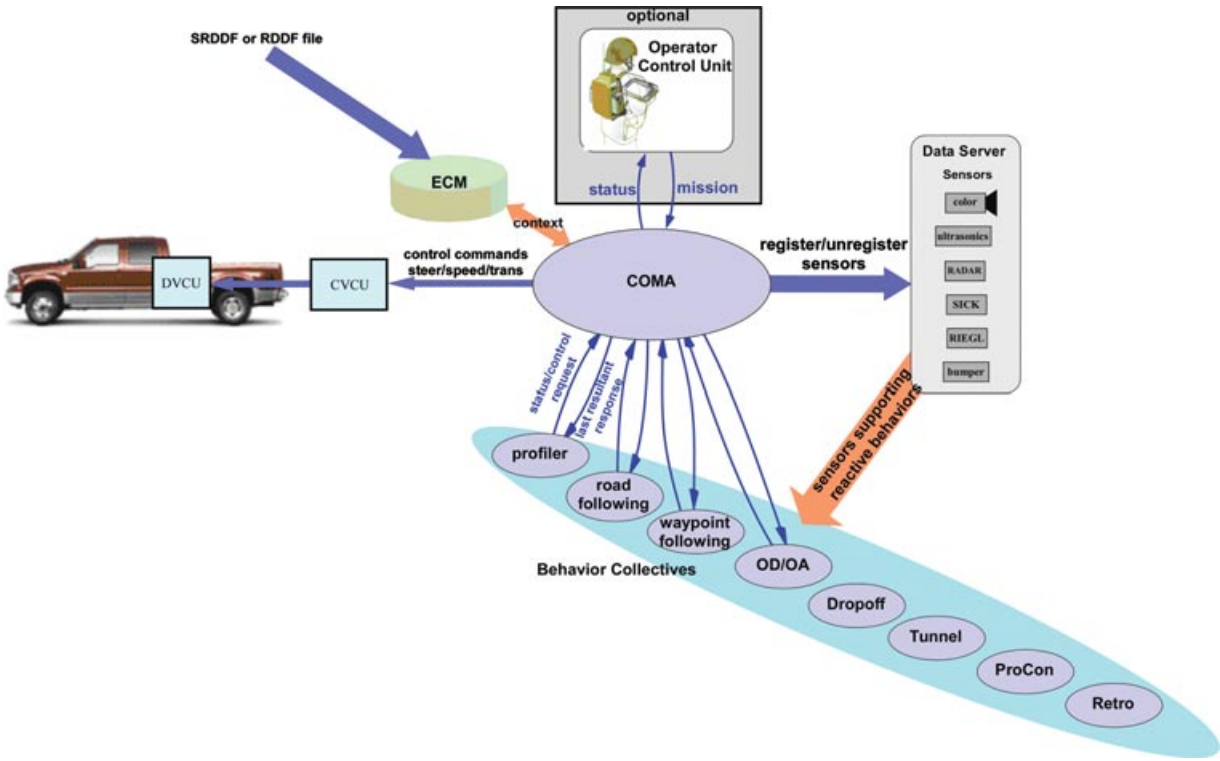

Figure 9. Schematic of the arbiter software architecture used by IVST during DGC2 (IVST, 2005a, 2005b).

Figure 10. Location on the DGC2 course where the IVST Desert Tortoise departed corridor boundaries. Multiple competing behaviors within the arbiter framework caused the vehicle to delay at a fork in the road. The correct route is the narrow hidden pathway on the left, as indicated by our team member standing where a DARPA photographer had been filming.

weight of the behavior to other behaviors in a specific environmental context and the confidence of the behavior. The superposition of the sum of negative behaviors was used to cancel out hazardous directions, and then the peak of the final response was used as the desired steering direction. Those behaviors that control speed also provided a speed vector over the full steering range, where the value of a vector element represents the speed the behavior wants to go for that steering direction. The arbiter took the minimum speed over all behaviors for the chosen steering direction.

Although this framework proved effective at the NQE, during the final event a situation occurred in which multiple competing behaviors with high confidence came into play at one time. These included, but were not necessarily limited to, vision-based road following (a very successful feature) locked onto the wide forward path while having an obstructed view toward the turn, a navigation request to turn sharply left onto a very narrow road, and apparent in-path obstacle detections of a pedestrian as well as from a cloud of dust created by the rapid deceleration to the intersection speed limit. Though the correct behavior did eventually prevail, the arbitration delay initiated a chain of events (Figure 10) that led to a minor departure from the course and ultimately the team's disqualification. Learning from this lesson, the IVS team decided to pursue a map-based sensor fusion strategy for the DUC that was neither situational dependent nor arbitrated.

3.3.2. Site-Visit Software Architecture

At the time of the DUC DARPA site visit, the bulk of the XAV-250 software was provided by Honeywell and is described in detail in previously published reports (IVS, 2006, 2007a, 2007b). We briefly review some key elements of the architecture, as depicted in Figure 11.

Mission controller: The function of the mission controller is to encapsulate the high-level logic needed to accomplish a mission and to distribute that information to the other components in the system. Each software element is responsible for producing certain events, which can trigger state changes and react to changes in state.

Mapper: The one map (TOM) accepts classification data (unseen, empty, road, line, curb, obstacle) from each sensor, fuses them spatially and temporally, and provides a single map that contains the probabilities of each classification per sensor per cell. Updates to the map are asynchronous per sensor. The map is composed of 1024 × 1024, 0.25 × 0.25 m grid cells, centered on the vehicle. The map is implemented as a doubly circular buffer (or torus), and TOM scrolls this map with the movement of the truck (determined from the PING). Each cell contains

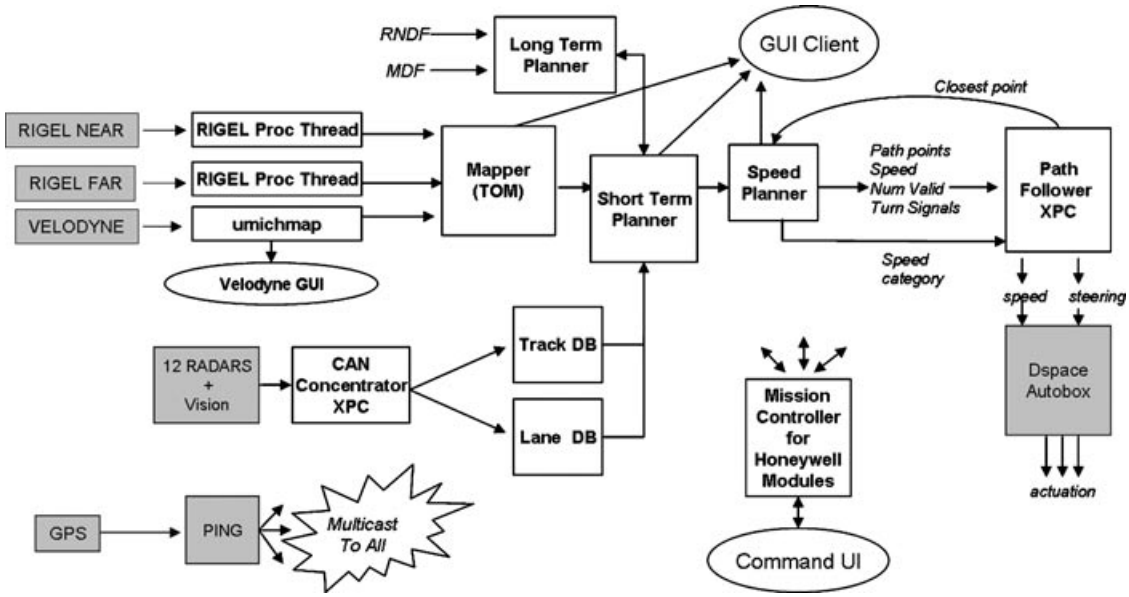

Figure 11. Schematic drawing of the software architecture employed on the IVS vehicle at the 2007 DARPA site visit. White boxes indicate major software components and gray boxes embedded hardware.

every sensor's estimated probability of the classification as being unseen, empty, road, line, curb, or obstacle. A fusion algorithm is run at 10 Hz across the center 512 × 512 cells (128 × 128 m) to derive a single probability of classification for each cell. TOM feeds these data to both the short term planner and the graphical user interface (GUI).

Long-term planner: The long-term planner is responsible for determining the route the truck will take through its mission. It reads the route network definition file (RNDF) and mission definition file (MDF) and then based on the GPS position of the truck, determines where it is on the route network and creates a high-level plan consisting of a sequence of RNDF waypoints to reach the checkpoints specified by the MDF. This plan is devised using Dijkstra's shortest-path algorithm, with edge weights in the constructed graph determined based on a variety of factors, including estimated distance, speed limit specified in the MDF, and whether the system has recently experienced that road to be blocked. The long-term planner provides the planned route to the short-term planner.

Short-term planner: The short-term planner takes as input the position of the truck, the next goal from the long-term planner, and the fused map from TOM and produces a path to that next goal. The planning problem is formulated as a gradient descent in a Laplacian potential field.

Speed planner: The speed planner takes as input the obstacle map and the path computed by the short-term planner and calculates a safe speed along that path by considering a number of issues, including intersecting tracks, proximity to obstacles, curvature, speed limits, and zones.

Path follower: The path follower takes as input the position, heading and speed of the truck, and a list of path points from the speed planner and calculates a goal point on the desired path just ahead of the front axle using a vehicle model. The position and curvature of the path at the goal point is used to calculate a steering wheel position command, which is passed to the VCU, implemented in a dSPACE AutoBox.

4. TRANSITION TO THE IVS/MIT VEHICLE ARCHITECTURE

An internal assessment of the state of the project was conducted after the IVS team took a few weeks to digest the results of the DARPA site visit. Although the demonstrated functionality was sufficient to satisfactorily complete all the required site-visit milestones, it had become obvious that there were a number of problems with our approach that would preclude

completing the final system development on schedule. These issues included unexpected delays introduced by both hardware and software development and were compounded by team staffing limitations. Although we will not elaborate on the details, we will point out a couple of examples to help the reader understand our subsequent and seemingly radical shift in plans. At the time of the site visit, our middle ware employed a field-based, Laplacian path planner, which had been demonstrated in other robotic applications (IVS, 2007a, and references therein), notably with unmanned air vehicles (UAVs). In the context of the DUC, successful implementation of the Laplacian planner required the inclusion of pseudo obstacles to prevent "undesired optimal" paths, a simple example being a four-way intersection. Without painted lane markings existing within the intersection itself, the "un-aided" Laplacian planner would calculate the best path as one passing directly through the center of the intersection, obviously causing the vehicle to depart its lane. Whereas these limitations could be overcome in principle, in practice the myriad of topological possibilities made this algorithmic approach time consuming and cumbersome.

Also of major concern at the time of site visit were infrequent but significant positional errors exported by the inertial navigation system (INS). Although the PING IMU was undeniably orders of magnitude more sensitive than the commercial units employed by most of the other teams, this also resulted in complications with tuning its prototype software (Kalman filter parameters) to accommodate a commercial GPS input as one of the state estimators. Given that the functionality of virtually everything on the vehicle relied on having accurate pose information, Honeywell focused its efforts on this system, and it was decided that additional external collaborative resources would be solicited to complete final system development on schedule.

MIT was a logical candidate in this regard, owing largely to the preexisting, well-established Ford–MIT Research Alliance. A fair number of projects falling under this umbrella included students and professors who were also part of MIT's DUC team. Furthermore, the teams had been in contact beforehand, as Ford had provided them prior support, including help with the acquisition of their Land Rover LR3 platform. As a side benefit, expanded contact between the teams would allow both sides to assess options for future joint autonomous vehicle research.

By mid-August, an agreement in principle had been made to work together. After clearing this proposal with DARPA, an implementation plan was devised. Although IVS had an existing set of code, albeit with gaps, it was instantly apparent that it would be far quicker to simply migrate to the MIT middleware code (Leonard et al., 2008) than it would be to try to merge disparate pieces of software. On the positive side, the MIT software was already successfully running on their platform, and the code structure was generic enough to incorporate most sensor types into their mapper. On the negative side, we would be dealing with a vast amount of code for which we had no inner knowledge, the transition would require a large rip up of hardware and software architecture, and we would no longer have a second identical truck for development.

4.1. Undeployed IVS Sensor Technology

Another one of the negatives of transitioning to the MIT architecture was that several novel sensing systems that were being developed by IVS (and partners) had to be put on hold so that all personnel resources could be reallocated to ensure successful completion of the ambitious software transition. The remainder of this section illustrates a few examples, not only to point out that they could have been migrated to the MIT platform given enough time, but also because they are still under consideration and/or development, and moreover there may be useful new information for the reader (particularly concerning results from the Velodyne LIDAR effort).

4.1.1. Lane detection

Redundant vision systems were under development to aid in the fault tolerance of lane detection. A customized Mobileye vision system with a primary focal point of 40 m and a Cybernet vision system with a primary focal point of 25 m were in development prior to the site visit. Each system was capable of detecting lane markings and vision gradient differences between the roadway and side of the road. Figure 12 is an example of urban driving data acquired from the Delphi/Mobileye fused radar and vision application taken on a surrogate vehicle using a production intent system composed of a single radar and camera. The XAV-250 implementation was being developed to use three radars, effectively increasing the radar FOV for forward object detection by a factor of

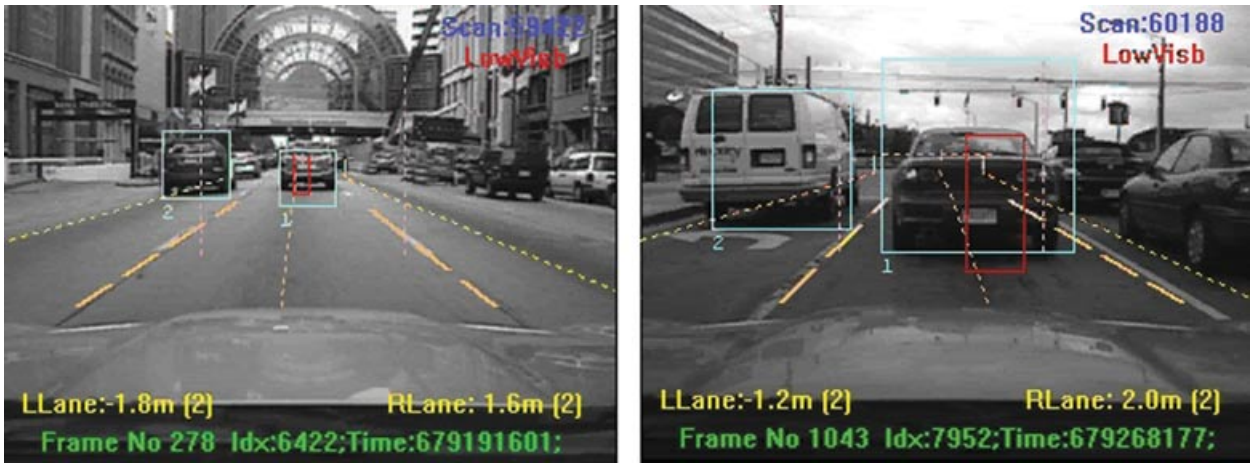

Figure 12. Delphi/Mobileye radar/vision fusion with lane detection and in-path targets.

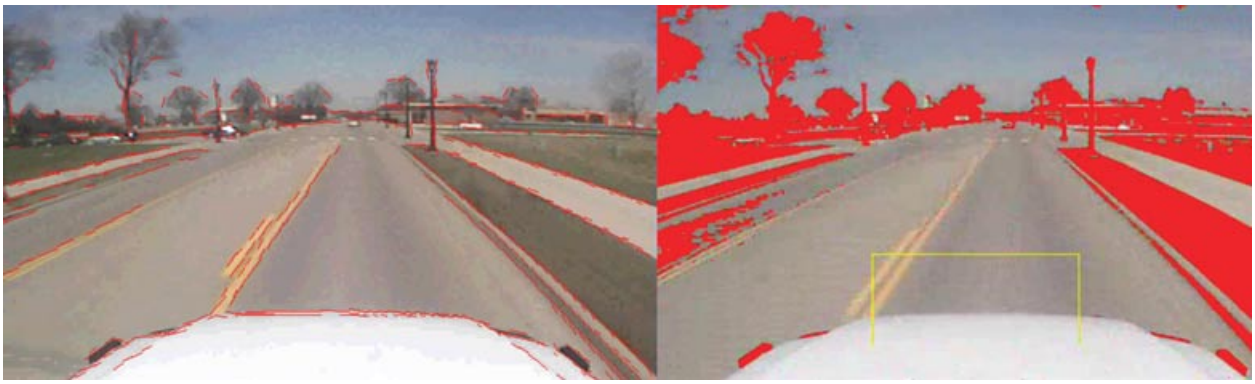

Figure 13. Cybernet vision system roadway and traversability detection example.

three. Although the vision system has a fairly wide FOV, it was initially intended to confirm radar targets only from the center channel. Figure 13 illustrates two alpha-version feature-extraction applications derived from the Cybernet vision system as actually 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 on contrasts in color and texture, and is heavily influenced by the sample roadway immediately in front of the vehicle (yellow box).

4.1.2. Velodyne Processing

The Velodyne HDL-64E was the primary perceptual sensor used by team IVS, both before and after the transition to the MIT code base. It provided 360-deg FOV situational awareness and was used for both static and dynamic obstacle detection and tracking and curb and berm detection, and preliminary results also suggested that painted-line lane detection on asphalt was possible via the intensity channel. The HDL-64E operates on the premise that instead of a single laser firing through a rotating mirror, 64 lasers are mounted on upper and lower blocks of 32 lasers each, and the entire unit spins. This design allows for 64 separate lasers to each fire thousands of times per second, providing far more data points per second and a much richer point cloud than conventional "push-broom" LIDAR designs. Each laser/detector pair is precisely aligned at predetermined vertical angles, resulting in an effective 26.8-deg vertical FOV. By spinning the entire unit at speeds up to 900 rpm

(15 Hz), a 360-deg FOV is achieved (resulting in 1 million 3D points per second).

Sampling characteristics: Each Velodyne data frame consists of 12 "shots" and is acquired over a period of 384 μs. This data frame is packetized and transmitted over the network via UDP at a fixed rate. To properly decode and transform points into the world frame requires compensating for the rotational effect of the unit, as all of the lasers within a block are not fired coincidentally. Each shot consists of data from one block of 32 lasers; however, the Velodyne fires only four lasers at a time with a 4-μs lapse between firings during collection of a full shot. Therefore, 32 lasers per block divided by 4 lasers per firing yields 8 firings per shot, with a total elapsed time of $8 \times 4\ \mu\text{s} = 32\ \mu\text{s}$ (thus $32\ \mu\text{s} \times 12$ shots $= 384\ \mu$s to acquire 1 data frame). This means that per shot, the head actually spins a finite amount in yaw while acquiring one shot's worth of data. The reported yaw angle for each shot is the sampled encoder yaw at the time the *first* group of four lasers are fired, so that in actuality, the yaw changes by δ yaw = spin_rate $\times$ 4 μs between the groups of four firings, such that on the eighth firing the unit has spun by spin_rate $\times$ 28 μs. For example, on a unit rotating at 10 Hz (i.e., 600 rpm), this would amount to 0.1 deg of motion intrashot, which at a range of 100 m would result in 17.6 cm of lateral displacement.

Coordinate transformation: Figure 14 illustrates the HDL-64E sensor geometry used for coordinate frame decomposition. The unit is mechanically actuated to spin in a clockwise direction about the sensor's vertical z axis, z_s. Encoder yaw Ψ is measured with 0.01-deg resolution with respect to the base and is positive in the direction shown. Each laser is individually calibrated and parameterized by its azimuth ϕ and elevation θ angle (measured with respect to the rotating x_s–y_s sensor head coordinate frame) and by two parallax offsets, p_v and p_h (measured orthogonal to the laser axis), which account for the noncoaxial laser/detector optics. Thus, a time-of-flight range return can be mapped to a 3D point in the sensor coordinate frame as

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = d \underbrace{\begin{bmatrix} \cos\theta \sin\alpha \\ \cos\theta \cos\alpha \\ \sin\theta \end{bmatrix}}_{\hat{e}} + \underbrace{p_v \begin{bmatrix} -\sin\theta \sin\alpha \\ -\sin\theta \cos\alpha \\ \cos\theta \end{bmatrix} + p_h \begin{bmatrix} -\cos\alpha \\ \sin\alpha \\ 0 \end{bmatrix}}_{\boldsymbol{p}}, \quad (1)$$

where d is the measured range and $\alpha = \Psi - \phi$. For real-time computation, this mapping can be precached into a look-up table indexed by laser ID, i, and encoder yaw (i.e., a 64 $\times$ 3,600 LUT) so that only three multiplies and three additions are needed per laser for decoding to the sensor frame:

$$\boldsymbol{x}_s = d\ \hat{e}(i, \psi) + \boldsymbol{p}(i, \psi). \quad (2)$$

Points can then subsequently be mapped to the world frame based on vehicle pose. In our system, navigation updates were provided at a rate of 100 Hz;

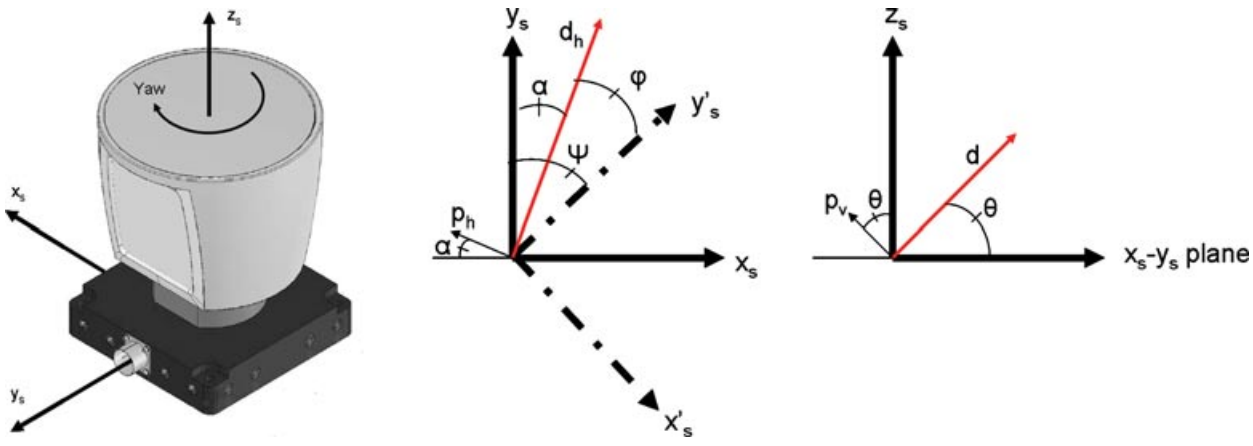

Figure 14. Velodyne HDL-64E sensor coordinate frame used for decomposition (laser beam depicted in red).

Figure 15. UMichMap Velodyne LIDAR interface.

therefore, we causally interpolated vehicle pose to the time stamp of each shot using forward Euler integration with a constant-velocity kinematic model.

Site-Visit Technology—UMichMap: An independent alliance between Ford and the University of Michigan was responsible for developing UMichMap, the software package that interfaced and processed the raw data from the HDL-64E. At the time of the site visit this software comprised two well-tested algorithms: obstacle/traversable area detection and lane marking detection. A third algorithm, dynamic obstacle tracking, was undergoing alpha testing. All algorithms classified 0.25-m grid cells within a sliding tessellated map that was output to a global map for fusion with other sensor maps.

Obstacle detection involved first eliminating overhead features (bridges, trees, etc.) that did not obstruct the safe passage of the vehicle. An object within a cell was then classified as an obstacle if it vertically exceeded an above-ground-plane threshold of 0.25 m and there was confirmational evidence from neighboring cells. During the demonstration runs at the actual site visit, this threshold was temporarily increased to 1 m due to a ranging problem in the presence of retroreflective tape (discussed below). Preliminary results of lane detection were positive. The algorithm was based on thresholding the Velodyne intensity channel returns and then fitting a contour to the remaining data points using a RANSAC (Fischler & Bolles, 1981) framework. Simply speaking, if the return amplitude ranged from a value of 200 up to the maximum of 255, it was considered to be a candidate lane marking (for comparison, typical range intensities for asphalt were well below a value of 100). Of those remaining thresholded points, those with a sufficient RANSAC model consensus were deemed actual lanes. Finally, a GUI using GLUT (OpenGL Utility Toolkit) was developed for real-time display, data log playback, analysis, and calibration.

Figure 15 is a composite image illustrating how data captured from the Velodyne are processed in the UMichMap system architecture. The photo on the left is a typical road segment at Ford's Dearborn campus. The center image depicts a single data frame captured from the Velodyne HDL-64E, with each LIDAR beam color coded by the intensity of the return. One 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 exported attribute features) is shown on the right for each 0.25 × 0.25 m cell surrounding the vehicle. It should be noted that as the vehicle moved, the driveability map would rapidly fill in as the laser beams swept the entire region in front of the truck.

Site-Visit Lesson—LIDAR Issues Associated with Highly Reflective Materials: The software developed for the Velodyne LIDAR proved to be a huge success, especially after incorporating the latest firmware upgrades that were designed to correct some vexing hardware issues. However, one item that had yet to be perfected, and for that matter is still unresolved, involves the intensity channel information from each of the 64 beams in the HDL-64E model. The last

Figure 16. Michigan Proving Grounds vehicle dynamics test area, precision steering course, and site-visit course. Lane markings for the site-visit course were laid out using reflective highway construction tape. The intersection area contained in the red rectangle is discussed in the next figure.

firmware upgrade enabled our unit to detect obstacles to beyond 100 m, and when our vehicle was parked anywhere within the site visit course, we were easily able to detect all of the painted lines.

Furthermore, the center of the lines registered in our map to within one pixel or less of where they were calculated to be from the geometry of the site-visit course (Figure 16) as determined by the GPS survey markers. (For the record, it is not known how accurately the lane marking tape was actually laid with respect to these points.)

Surprisingly, when we processed these data through the obstacle detection algorithm, we saw phantom obstacles appearing on the lines, growing in size with distance from the Velodyne (see Figures 17 and 18). A brief discussion of LIDAR is needed in order to explain this effect. Most LIDAR manufacturers calculate the range based on the time at which the back-scattered (returned) beam intensity reaches its maximum. Under normal circumstances, this would physically correspond to the brightest spot of the projected beam, typically the center of the dispersed spot. The problem is that laser returns from highly reflective materials, such as retroreflecting paint found on traffic signs (in our case, road construction tape), are orders of magnitude brighter than those from

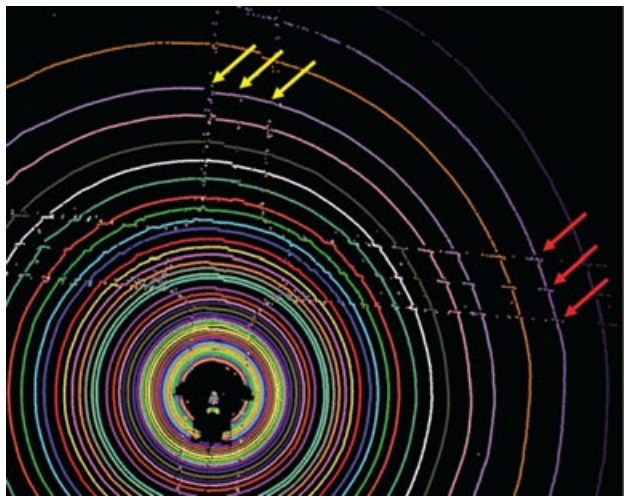

Figure 17. We observed large range variations as the beams swept over road lane marking tape at our site-visit course. These delta ranges ended up causing our obstacle detection algorithm to put spurious pixels on the map in these locations. Note the fourfold symmetry in the lane "zigzag" about the lateral and longitudinal axes. The red arrows highlight that this effect is more pronounced as the laser incidence angle to the line increases; the yellow arrows show that this effect is less so when the sweeping angle of incidence is small.

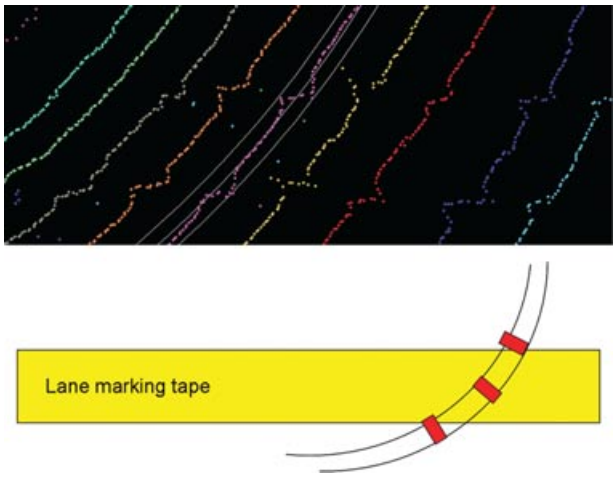

Figure 18. Screen capture (top) of some of the beams crossing a single piece of lane marking tape, ~10 cm wide. The lilac colored one is laser 0 and in our configuration falls on the ground at the 20 m horizontal range. The concentric circles are ±0.25 m from the nominal arc everywhere else. Conceptually, a simple-minded explanation (bottom) would be provided by the following observation: if the maximum intensity in a range sample return defines the range, then we could see something like that which occurs in the real data.

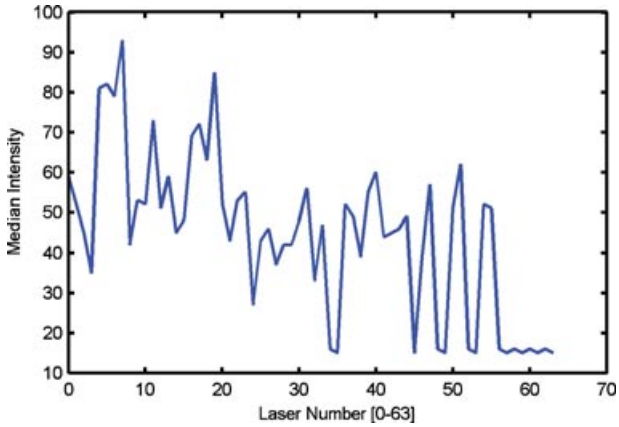

Figure 19. Interlaser intensity variation within a single scan across a uniform asphalt surface.

normal materials. As such, the return will immediately saturate the detector, and the peak signal will occur at the time when the beam first strikes the reflector and not necessarily at the center of the beam spot. As such, the circular sweeping beam from the Velodyne produces a zigzag discontinuity in the range measurements upon crossing a retroreflecting line. Just prior to and after striking the line, the range is correct. However, when the beam first strikes the line, the perceived range is too short, and at the point where the beam exits the line, the range is perceived as too long. The magnitude of the discontinuity increases with distance owing to the divergence of the projected beam and over the span of the site-visit course could exceed 0.5 m. The implications for obstacle detection algorithms are obvious.

We considered at least three methods to deal with this. One was to simply leave the lines as real obstacles in the map. The problem with this approach is that the planner needed to ignore lines when passing stalled vehicles, and additional logic would have been required to establish that these were in fact lines. Additionally, declaring them as physical objects would slightly narrow the lane widths (by at least two pixels, each 0.25 m wide), something we could ill afford with a vehicle the size of an F250. Second, we had characterized the intensity of laser returns for each of the 64 beams when scattering off asphalt (Figure 19); hence we could easily identify the lines from the pavement. Although we did in fact demonstrate this method as a means to easily discriminate lane markings, it required more logic to implement than the temporary solution we settled upon—simply raising the threshold for what was declared an obstacle.

Interestingly, a somewhat similar phenomenon was observed while the vehicle was parked in the garage at our site-visit location, which had a smooth cement floor. A gaping hole appeared in the range map in front of the vehicle. Upon further investigation, we noted a large retroreflecting sign on the wall several meters in front of the vehicle. The backscattered laser return from the floor was obviously far weaker than the forward-scattered signal off the floor and back again from the bright sign. The sign was thus observed twice, once in the correct location by the higher elevation beams that hit it directly and second "beneath the floor" by the beams that reflected off the cement and then struck the sign. Unexpected artifacts of this nature plague nearly all sensors and as such explain the need for redundant sensing systems and conformational data in production products such as automobiles. Whereas we were concerned that this might be an issue in the race, as far as we could ascertain, traffic signs and license plates were never aligned so as to produce these ghost obstacles.

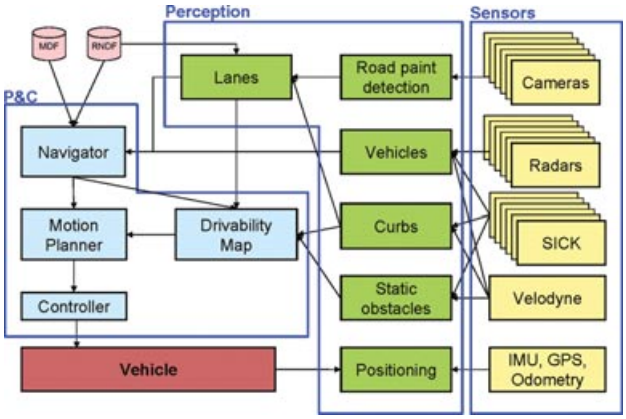

Figure 20. Schematic drawing of the MIT base code software architecture (Leonard et al., 2008).

4.2. Making the XAV-250 "Look" Like MIT's LR3

At the start of the transition to the MIT architecture (Figure 20), the MIT team members were actively engaged with tasks to add advanced navigation traffic capabilities to their platform. To minimize distractions to their efforts, it was decided to change the XAV-250 actuation, infrastructure, and sensing suite to match MIT's LR3 as closely as possible. With these alterations, it was also necessary for Ford to write several software and interface modules, as illustrated by the schematic in Figure 21. Some of the notable major changes are as follows:

- In the site-visit configuration of the IVS vehicles, the compute cluster used a mixture of Advantech and Dell servers. The Dell servers were smaller and had higher speed CPUs, whereas the physically larger Advantech servers had increased external I/O card capacity and were compatible with the MathWorks xPC rapid prototyping software. To increase computational power, the Advantech computers were removed and the Dell computers from both IVS vehicles were combined into the race vehicle, resulting in a compute cluster with 24 compute cores. Although fewer than the 40 cores used by MIT, they were sufficient to run their core software with our reduced sensor set.
- With the removal of the Advantech computers, the CAN concentrator was replaced with an array of EthCAN modules. These modules were based on a Keil software evaluation board (model MCB2370) using an ARM9 processor and were programmed to pass CAN messages to the compute cluster via Ethernet. Each module supported two CAN networks. For each CAN message received, the EthCAN module would transmit the CAN header information and data bits using one Ethernet packet. Similarly, an Ethernet message could be sent to the EthCAN module, and it would repackage the information to produce one CAN message. The EthCAN array was used

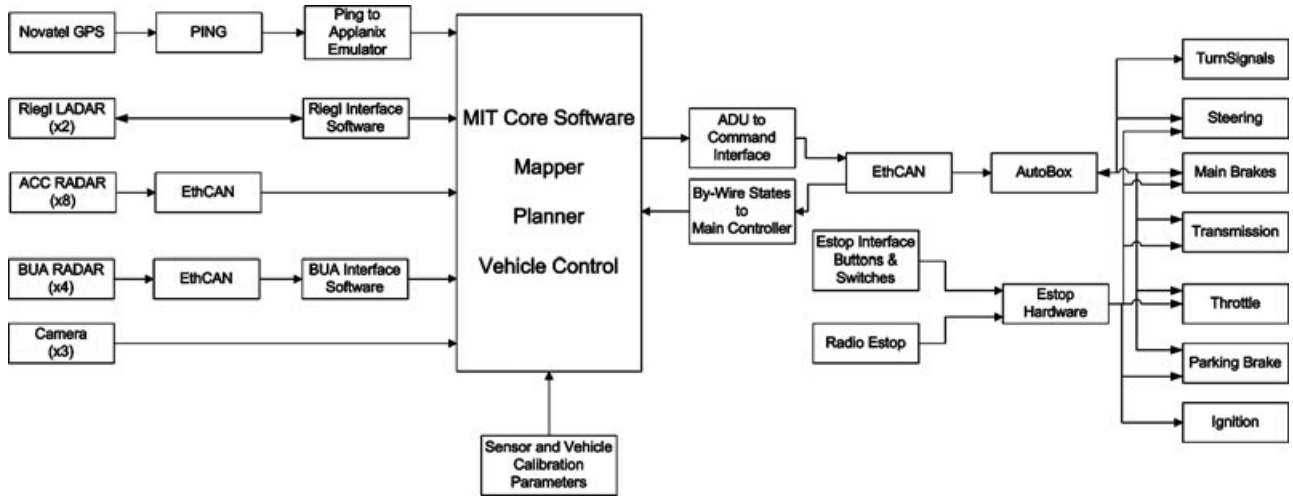

Figure 21. Schematic drawing of the software architecture employed on the IVS vehicle after the transition to the MIT code.

to interface the radars and VCU (dSPACE AutoBox) to the main compute cluster. It should be noted that the MIT software architecture does not take advantage of the preprocessing that resides within the ACC radar units (e.g., closest in path target identification). Relying solely on raw radar data, the MIT development team created their own radar processing software. The reader is directed to the MIT documentation (Leonard et al., 2008) for details related to this data processing.

- The EthCAN modules and the PING had difficulties supporting high-speed Ethernet traffic. In the final configuration, two Ethernet switches were added to form low- (10 Mb/s), medium- (100 Mb/s), and high- (1 Gb/s) speed networks.
- In the ADU command interface, control messages sent from the MIT core software were repackaged and sent via CAN messaging to a dSPACE AutoBox for by-wire execution. An EthCAN module performed the conversion between Ethernet and the AutoBox CAN network. The AutoBox contained the VCU software that controlled the low-level functions of the by-wire systems and monitored signals for fault detection. Hardware-based monitoring was also implemented if the CAN connections were broken or the network failed. If a hardware, software, or out-of-range signal was detected, an emergency stop was requested. In a similar fashion, the vehicle by-wire states were sent back to the main controller module running in the MIT core software for state information and for fault monitoring.
- To avoid camera interface issues, the team decided that the quickest way to implement the MIT lane detection algorithms on the IVS vehicle would be to add three roof-mounted Point Grey Firefly MV cameras. The number of cameras was limited by computational capability. Unfortunately, the Mobileye system was abandoned due to its incompatibility with the MIT software architecture. On a similar note, the Cybernet algorithms, which had been operational well in advance of the site visit, were also never integrated.
- In contrast to the LR3, the XAV-250 was equipped with Delphi BUA radars that gave nearby obstacle information out to a range of 5 m. This improved reliability in detecting low-lying, close-by obstacles that fell within the Velodyne vehicle shadow. To take advantage of the BUA units, Ford developed a sensor interface function that allowed BUA data to be processed by the MIT software. This software function first read the BUA messages from the CAN bus in real time and then transformed the range returns into map coordinates based on the BUA calibration and vehicle pose within the map. If sufficient returns had accumulated in a particular location, that position, with the inclusion of a dilatational radius, was classified as a high-weighted obstacle in the map.
- The LR3 incorporated a large number of SICK line-scanning LIDARs, which have a nominal resolution of 1 deg per step. The XAV-250 used two high-resolution Riegl LIDARs for the same "push-broom" functionality, acquiring range returns at 0.02 deg per step. The Riegl interface function generically repackaged the data into laser scan packets that the MIT software could use.
- The MIT software expected input from an Applanix POS LV 220 INS; thus an emulator was written by Ford to pass the Honeywell PING data in the same format.

Much of the effort to adapt the MIT software to the IVS platform was spent changing calibrations concerning vehicle parameters and sensor reference locations. In most cases, the changes could be made rather easily via calibration files; however, in some instances, these values were hard-coded constants that needed to be identified and changed within the code itself. Within 3 weeks after deciding to reconfigure the XAV-250, testing began using the MIT software and toolset.

Once testing was underway, it was determined that some additional physical, electrical, and software changes were needed to accommodate the "denser" computing cluster, including the installation of larger battery capacity, redistribution of electrical loads between the front and rear electrical systems, and redirection of the cooling air flow in the rear environmental computer enclosure.

5. PERFORMANCE ANALYSIS

5.1. Testing at El Toro with MIT

MIT team members visited Dearborn in early October, with the primary objective to help fine-tune the parameters that are used by the vehicle prediction model portion of the planner code. Prior to this time, we were having limited success in operating their code on our vehicle. However, once this exercise was complete, we quickly were able to demonstrate a variety of autonomous behaviors on the XAV-250, many of which exhibited peculiarities such as MIT was reporting from their LR3. It was at this point that the potential utility of collaborative testing was fully realized, and MIT suggested that we join them at a test facility on the El Toro Marine Corps Base. With some last-minute alterations to our schedule, we were able to divert the truck to southern California and achieved approximately 1 week of joint testing prior to the NQE.

Testing with two different robotic vehicles on the course at the same time proved to be very productive, with each team learning from the other. MIT had been at El Toro for a couple of weeks prior to our arrival and had constructed a RNDF of the road network, as well as a variety of MDFs. On our first attempt at their course, we had serious difficulties staying in our lane due to a constant bias in position. We had witnessed this before, watching a number of elite teams exhibit this behavior at the first two Grand Challenges. The problem was obvious to us: MIT used an Applanix INS, which by default exports coordinates in the NAD83 datum, whereas our system used the WGS84 datum, the same as DARPA uses. In southern California, these happen to differ by approximately 1.5 m, or roughly half a lane width. After a code fix by MIT and a translation of coordinates in the RNDF, we were soon driving robotically past one another without issues.

With some cooperative help from MIT, we were able to successfully demonstrate the operational capability of each of the sensors and processes (specific task algorithms, such as curb detection) that the MIT code would support on our platform. A highlight of the testing was the ability to validate all of the intersection precedence scenarios with both robots and traffic vehicles involved. Numerous consecutive runs were made to ensure consistent behavior, and the success of this effort was later apparent at the NQE and UCE, where (to our knowledge) neither MIT nor IVS ever made a traffic-related driving error. The only real downside of traveling to El Toro was the interruption caused by the Los Angeles wildfires, which shortened our available test time and introduced some hardware problems associated with the fallout of very fine ash.

As a final note, we would like to clarify that when IVS and MIT finished testing at El Toro, there was a code split and no further technical interaction occurred between the teams until after the race. We felt strongly that there should be no advantage afforded to either team, relative to the field of contenders, based on any prior knowledge gained while undergoing testing at the NQE.

5.2. NQE and UCE

At various points during the NQE and/or UCE, we successfully demonstrated each of the sensor modalities and software algorithms that were capable of being supported by the MIT code. As it turned out, it was not always possible to operate the full sensing and software suite simultaneously, and as such, in the end we converged upon the simplest stable configuration possible. This consisted of GPS waypoint following, the Velodyne LIDAR, and a small set of Delphi ACC radars, notably including the front forward unit. Throughout our vehicle evaluation on the NQE sites, and during additional testing, we encountered and solved numerous problems, with both the hardware and the software. There were, however, some bugs for which no near-term solution existed, and this impacted what we could reliably run. Even though some of the observed anomalies occurred on a rare basis and we could likely have operated more of our system, we chose not to, as we did not understand the root causes, and moreover because the same functionality could be obtained with a simpler solution. To reiterate, although some sensors were not used for autonomous decision making, the sensor hardware itself was operational, and in many cases data from these systems were recorded for later resimulation studies.

5.2.1. NQE: Area C, the "Belt Buckle"

Our first test session occurred in Area C (see Figure 22), referred to by many as the "belt buckle." This test was presumably designed to evaluate navigation, intersection logic and traffic precedence, and route replanning when presented with a blocked path.

Figure 22. Area C: intersection logic and dynamic replanning. Aerial photo of Area C course (top); red arrow indicates the intersection depicted in the figures below. XAV-250 successfully exhibits advanced traffic behavior (bottom left). MIT viewer rendering of the RNDF and tracked cars at the intersection (bottom right).

For both of our runs in Area C, we demonstrated flawless execution of intersection logic and traffic precedence, with the truck stopping precisely at the painted stop lines and no errors occurring in any of the intersection scenarios. In each run, we accurately navigated the course via GPS waypoint tracking and by utilizing the curb detection process fed from both types of LIDAR—the two push-broom Riegls and the Velodyne. Although video was recorded for data logging purposes, the lane detection process was not employed for navigational guidance. This decision was made primarily in light of the abundance of curbs and the faintness of painted lines in this neighborhood but to some extent by issues we were experiencing with our vision hardware and software. At the time of the first run, we had not had an opportunity to validate the camera calibration (following transport from El Toro), and on the second run, we did not want to introduce changes to what had worked successfully the first time.

For us, the route replanning proved to be among the most problematic of any of the NQE tasks, and we would spend the majority of our remaining free time at the event in an effort to solve this issue. On our first run in Area C, the truck was issued a DARPA pause command after it attempted to circumnavigate the road blockage by cutting between the construction barrels and a large tree in the adjacent yard. We were allowed to reposition the vehicle on the road, and on the second attempt it executed a U-turn; however, it did so by departing the street again and completing the maneuver on a lawn. When the truck reached

the second blockage constructed from stop signs on gated arms, it immediately recognized them as obstacles, stopped for several seconds, and again appeared as if it was going to seek a route around them. Coincidentally, our test time ran out at this moment, so the final outcome in this scenario remains uncertain.

The behavior exhibited here was initially unexpected, and explaining it requires a discussion of the MIT planner code. The blockage occurred immediately in front of a checkpoint on what was essentially a circular loop. In this case, it is topologically impossible, following highway rules of the road, for the vehicle to replan a route reaching the checkpoint. The only way this point could be achieved would be to a priori assume that the same blockage would exist after the vehicle had circled the course in the opposite direction and furthermore that the vehicle could and would execute another U-turn so as to be in the correct lane in the correct orientation. Unfortunately, this scenario was overlooked in the planning logic and had not been discovered in our limited prior testing, as there had always been an intersecting roadway, providing a valid alternative route, between blockages and the next checkpoint.

5.2.2. NQE: Area A, the "Circles of Death"

The second test site we visited was Area A, a place we personally called the "circles of death" (see Figure 23). By our account, there were about a dozen traffic cars traveling bidirectionally around an outer oval of roughly 300 m in circumference. Our task was to make left-hand loops on a subsection of the course, yielding through gaps in the oncoming traffic, and pulling out from a stop sign onto a very narrow section of road abutted on one side by a solid concrete barrier.

We felt that our performance here was very good, despite receiving a fair number of honks from the traffic vehicles. In each of our two attempts, we completed more than a dozen laps, many of them perfect. Our robot always recognized traffic and precedence and never came close to hitting a moving object. The difficulty in this task stemmed not only from the density of traffic, but also from our interpretation of the rules, in which we assumed a requirement of allotting several vehicle lengths of spacing between our truck and the traffic vehicles. Given that our vehicle is roughly 7 m in length and we had to allow spacing in both directions when exiting the stop sign, very little room or time was left for traffic gaps along the 60-m stretch we were merging onto. Adding to the challenge was the 9 s it took for the XAV-250 to accelerate from a stop to 10 mph through the tight 90-deg turn. There were quite a few cases in which our vehicle would determine it was clear to go and then balk as a traffic vehicle would make the far turn from the transverse segment of the course onto the segment we were merging onto. Although the Velodyne was identifying the traffic vehicles across the entire span of the test site, our intersection algorithm did not classify these vehicles as obstacles of concern and assign them a track file until they entered a predefined zone surrounding the intersection itself.

On our first attempt at the course, we used GPS waypoint tracking and vision-based detection of lane markings for navigation, and left the curb detection algorithm off. There were few curbs on the loop, and we had also recently discovered the potential for a software bug to appear when the vision lane tracker

Figure 23. Area A "circles of death." XAV-250 waits to merge into traffic (left). MIT viewer showing the RNDF course with obstacles derived from the radars and Velodyne point cloud (right).

and curb detection algorithm reported conflicting estimates of the lane width or position. This approach worked on all but one lap. In that instance, a group of traffic vehicles passed very close to the F250 on the corner farthest from the stop sign; the truck took the corner perhaps 1 m wide and struck or drove atop the curb. We were generally very pleased, however, with the registration of our sensors (LIDAR and radar) with respect to ground truth, as the vehicle maintained the center of its lane while tracking very close to the concrete barriers.

We did make a change to our sensing strategy on the second run, however. In this case, we chose to run the curb detection algorithm and turn the vision-based lane tracking off. This decision was prompted in part by some bugs that had cropped up in the vision software, as well as by performing a resimulation of our previous run. This simulation showed that the curb function provided excellent guidance along the concrete barriers and on the curbs on the back side of the loop, with the GPS waypoints on the other two segments being sufficient to easily maintain lane centers. With this configuration, all loops were navigationally perfect.

5.2.3. NQE: Area B, the "Puzzle"

Area B was the last of the sites in our testing order (Figure 24). This area was very representative of the final event, with the exception that it had an abundance of stalled vehicles littering the puzzle-patterned streets. It was also very similar to the roads we tested on at El Toro, and hence we expected to perform well. To the contrary, we experienced our worst outings on this portion of the course, with most of our failures the result of some bewildering hardware failures, software bugs, and a bit of misfortune. We were not able to fully complete the mission on either of the runs and as a result, ended up having scant data to analyze in order to prepare for the finals. It would have been greatly beneficial to our team if DARPA had provided a practice site resembling this area.

On our first attempt at Area B, we chose to navigate using the same sensor set successfully employed in Area C: GPS waypoint tracking and curb detection derived from LIDAR. Absent from our sensing suite was the forward-facing radar cluster, as we had been observing a fair number of false detects from ground clutter on these radars and we feared this would be a bigger concern in Area B than in Areas A and C. It should also be recalled that the MIT code utilized the raw radar signals, as opposed to filtered output that normally is exported from the Delphi production radars. Given that the Velodyne LIDAR had been reliably detecting all obstacles of interest, this was deemed an acceptable solution. The vehicle demonstrated the ability to execute parking maneuvers and navigate around stalled obstacles and again performed without flaw at intersections and in the presence of traffic. However, we did experience occasional issues with the curb detection algorithm and in some cases missed curbs that existed, resulting in behaviors such as cutting corners between waypoints. In other cases, we misclassified obstacles that

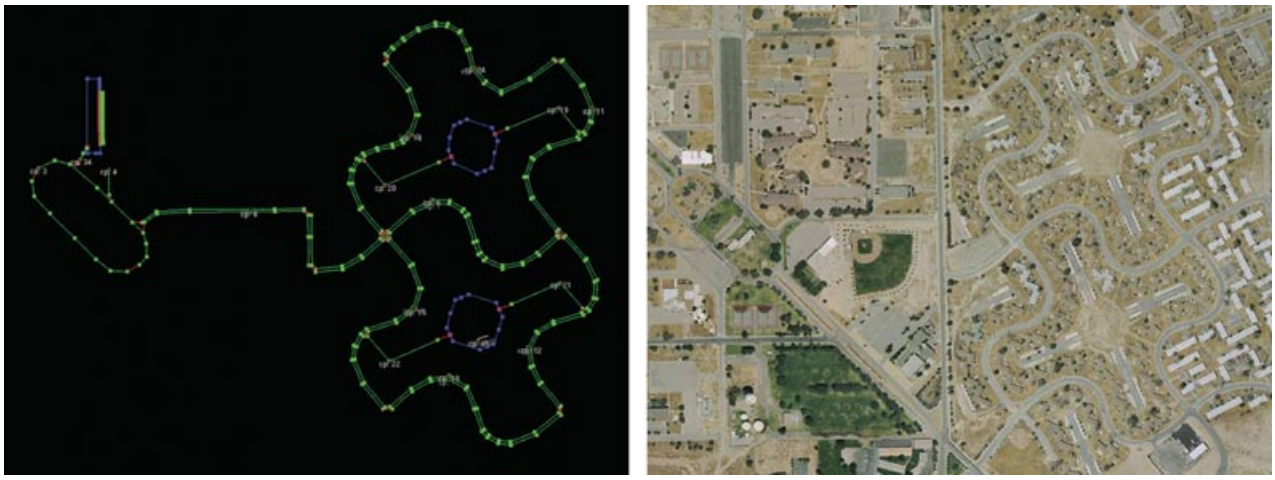

Figure 24. Site B course RNDF and aerial photo.

were not curbs as curbs, resulting in fail-safe modes being invoked, in which case curbs could again be ignored and similar driving behaviors would result. After jumping a curb about midway through the course, the Velodyne process crashed due to the physical unseating of a memory chip. Presented with no Velodyne obstacles in the map, the vehicle drifted at idle speed and was DARPA paused just as it was about to strike a plastic banner located at the interior of one of the puzzle pieces. Although it is highly improbable that we would have completed the course without the Velodyne, had the front radar been on, we would have likely detected the banner and stopped. Shortly thereafter, we realigned the front radars to a higher projection angle above the ground plane. To preclude the possibility of getting a false return from an overhead object, such as an overpass or low-hanging tree branch, we filtered the data to reject returns from ranges in excess of ~20 m.

Our second attempt at Area B came after sitting for 7 h in the sun on one of the hottest days of the event. We were scheduled to begin promptly at 0700; however, at each of the areas where we were tested, we were continually leapfrogged in the schedule by other teams in the field. Given that this was our last run before the finals, we decided to run all sensors and processes, including the vision lane tracking and LIDAR curb detection, having felt we had resolved the conflict between these two processes and wanting to acquire a complete data set. Upon launch, the vehicle proceeded nominally for a few hundred meters and then began to stutter, starting and stopping abruptly. DARPA immediately halted the test and sent us back to the start chute for another opportunity. Upon examining the data logs, it was found that we were flooding the computer network with traffic, and as a result navigation and pose messages were being periodically dropped, causing the stuttering motion observed in the vehicle. We terminated the lane detection function and relaunched the robot. It proceeded normally until reaching a stop sign at an intersection exiting the interior of one of the puzzle pieces, and at this location a significant discontinuity in the ground plane was formed by the crown of the facing road and/or the rain gutter between the two perpendicular roads. This was perceived to be a potential curb, causing the planner to keep the vehicle halted while it attempted to resolve the situation. After a brief stoppage, the truck idled across the street and into a lawn. A DARPA pause command was ineffectual, and we were subsequently disabled. The data logs revealed that the brake controller module indicated an over-temperature warning and refused to command brake pressure, which is consistent with the observed behavior with regard to the DARPA pause vs. disable (which commands the parking brake) commands.

At least three serious issues were raised from testing in this area:

Vision: Our vehicle employed three cameras, two forward facing with a combined FOV of ~100 deg and one center rearward facing. Although MIT had significantly more visual coverage, their lane detection algorithm was designed to accept a variable number of camera inputs and had been shown during testing at El Toro to work acceptably with our configuration when clear lane markings were present. However, during the NQE, we were unable to demonstrate reliable functionality from the vision system on our platform, and we are not certain whether hardware or software contributed to these shortcomings.

Curb detection: The primary functions of the LIDARs were to detect obstacles and to determine the topography of the ground around the vehicle, information that was used to determine traversable terrain, as well as to infer the existence of curbs. Generally speaking, the algorithms that perform these functions do so by making a comparison of the elevation changes or slopes of neighboring map grid cells. Declaring something an obstacle is much easier than declaring something a curb, especially in the case of the F250, where the ground clearance is more than 0.25 m. On the other hand, curbs are often less than 0.10 m in height relative to the surrounding terrain. The trick is to tune the thresholds in the software to maximize detection while minimizing false positives. An additional complication is that some features will correctly trigger a detection, yet not be an obstacle of concern. Examples of this would include speed bumps or other ground discontinuities such as grated sewer covers, rain gutters, etc. At present, when this type of detection arises, the MIT code relies on fail-safe modes to make forward progress. Given additional time to develop more sophisticated algorithms, and given redundant sensing corroboration, this problem could be addressed in other manners, but that was beyond the containable scope of the DUC effort.

Although these false positives were infrequent, we seemed to be more sensitive to them than MIT's LR3, again presumably due to the differences

between our platforms. Whereas MIT operated far more LIDARs—which could provide a greater degree of redundancy—we feel that the issue was more likely related to the fidelity of the LIDAR data. MIT's push-broom scanners, produced by SICK, sampled at 1-deg intervals, and their Velodyne, which they operated at 15 Hz, sampled at 0.15-deg intervals. On the other hand, our push-broom Riegls acquired data at 0.02-deg intervals, and our Velodyne, rotating at 10 Hz, sampled at 0.09-deg intervals. All things being equal, our sensing set would be more prone to elevation noise with an algorithm that compares neighboring cells. Once we realized this, we went back and resimulated all of our prior data sets, tuning the available parameters until we no longer saw these false positives. In the process, we recognized that the Velodyne alone was sufficient to detect curbs, and to avoid potential noise from the Riegls, we did not incorporate them in this algorithm in the UCE.

Brake controls: The IVS vehicle braking system was implemented by the use of a brake by-wire system furnished by TRW, one of Ford's primary brake suppliers. A production-level ABS module with the addition of custom hardware and software modifications allowed for independent dynamic braking of each wheel of the vehicle with superior braking accuracy and increased braking bandwidth, as compared with brake pedal displacement devices such as employed by some of the competing vehicles. This module is capable of providing smooth control of vehicle velocities down to 0.5 m/s. Another advantage of this system is its quick recovery time, which significantly enhances safety for development engineers occupying the vehicle during autonomous test runs. The main disadvantage of this prototype system was the inability to hold the vehicle at a complete stop for more than 10 min. As with all production ABS systems, the control module's heat dissipation characteristics are primarily specified for intermittent use, and therefore for our application, the continuous braking utility of the ABS module was limited. To protect the brake module from potential physical damage under these continuous braking applications, an internal software integrator timer was employed.

If the brake module actually overheats and/or the internal software integrator times out, all primary braking for the IVS vehicle is lost, and the vehicle would begin to roll at the idle speed of the engine, with an e-stop pause command being ineffectual. A similar time-out failure had occurred during the site visit, and at that time, DARPA suggested shifting to park position and releasing the brakes during an extended pause condition. This suggestion was an idea we had also contemplated but had not yet implemented due to time constraints. Once the site visit was over, we did follow this approach.

5.2.4. UCE: The Finals

When we started the truck on the morning of the race, one of the servers was inoperative. After some inspection, we discovered that the power supply to this server was not functional, and we had to remove all the servers in the rack to access it. Fortunately, we had a spare and reassembled the system just in time to make it into the queue. While we were going through our checklist in preparation for the start, we discovered that the low-level controller was exporting bad data. This was quickly traced to a faulty connector, which had likely resulted from the pandemonium in replacing the power supply. This connection was fixed, and the low-level controller subsequently seemed to be behaving correctly. Given that we were only two vehicles from the starting chute, and that it takes about 20 min for the INS to stabilize, we opted not to power down and perform a complete reboot of the system. This was a clear mistake, as corrupt data remained in the system, causing our steering controller to command a lock left turn upon launch. We were restaged, and during this time, did reboot the system from start. The second launch proceeded with nominal behavior. This incident points out a clear problem that needs to be resolved before autonomous systems ever reach production product viability—as has been shown with numerous other systems, the customer is unwilling to wait for even a few seconds after ignition before driving, much less the minutes it currently takes for computers (and INS systems) to initialize or reinitialize.

On the basis of the lessons we had learned during the NQE, we decided to run the UCE with the simplest stable configuration of sensors and algorithms possible. This consisted of GPS waypoint following, curb detection using only the Velodyne LIDAR, and obstacle detection using both LIDAR types and a small set of Delphi ACC radars, notably including the front forward unit and the 90-deg intersection scanning units. Because we were somewhat handicapped by not being able to run the vision system (aside from data logging purposes), we did insert a limited number of additional waypoints into the RNDF in regions we deemed might present issues.

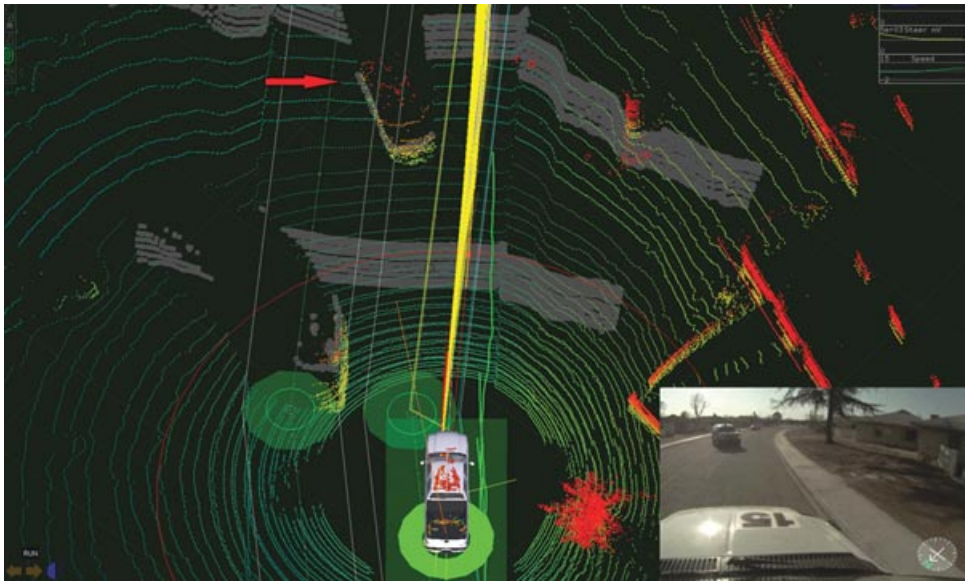

Figure 25. Incident in which the Cornell team tried to pass into XAV-250's lane, requiring evasive maneuvering on XAV-250's part to avoid collision. The red arrow denotes the Cornell vehicle as seen in the Velodyne point cloud, and the overlaid camera image to the lower right clearly shows the Cornell team in our lane.

During the time our vehicle was operational, it navigated well, obeyed the rules of the road, passed numerous robots and traffic vehicles, displayed correct intersection logic and traffic precedence, and successfully demonstrated parking maneuvers. Furthermore, it exhibited intelligent behavior when presented with the scenario of an oncoming robot approaching us in the wrong lane, slowing down and taking evasive actions to avoid a collision (Figure 25).

The failure mode for our vehicle occurred when we again detected a false positive upon exiting the interior of one of the puzzle pieces. While at the stop sign between this small road and the main road, the curb detection process incorrectly perceived either the crown in the facing road or the sharp discontinuity formed by the rain gutter to be a potential curb (Figure 26). Under normal circumstances, the vehicle would have waited for a short period (nominally 90 s) and invoked a fail-safe mode, which would have relaxed the curb constraint. However, following the difficulties we had with the topological conundrum in Area C, the timer for this process had been increased by an order of magnitude to rigidly enforce curb constraints while we evaluated a potential fix for this issue. Unfortunately, through oversight on our part, the timer had not been restored to its default value and we were subsequently and fairly disqualified for excessive delay on the course. When we rescued the truck, the planner was indicating a valid path, waiting for the fail-safe timer to expire. Although we cannot say what would have happened for the remainder of the course, we do know that this oversight prevented us from ever finding out.

6. GENERAL OBSERVATIONS AND LESSONS LEARNED

This section presents, in no particular order, a variety of the remarks contributed by team members during the writing of the DARPA final report and this article. Whereas these comments obviously pertain to our perception of the DUC experience, we suspect that many of these general observations and lessons learned will be shared by other teams as well.

- We expect that all teams will complain to some extent about having inadequate developmental and testing time between the announcement of the DUC program and the UCE. It is a very ambitious goal to create a test vehicle within a year, much less one that autonomously drives itself in simulated urban

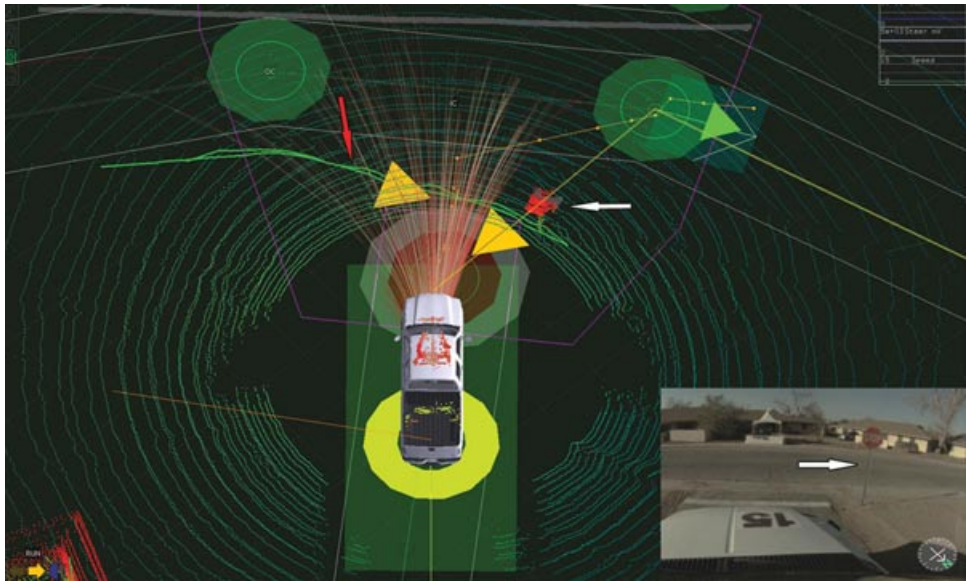

Figure 26. Failure mode of the XAV-250 during the UCE. The red arrow indicates a false detect of an in-path curb at an intersection. For reference, the white arrow indicates the stop sign in both the Velodyne intensity channel and camera imagery.

traffic. Complicating the challenge is the large number of intermediate milestones. Although we can certainly understand DARPA's need to assess interim performance, the Track A funding proposal, video submission, kick-off meeting, informal interim reports, technical paper, site visit, multiple revisions to rules and procedures, etc., are nevertheless distractions, especially for teams with few members.

- Many of us felt that the DARPA site visit and NQE did not adequately represent the final events at any of the three Grand Challenges. In some sense they are actually a bit of a detour—the site visit because it requires an integrated, fully functional system too early in the development timeline and the NQE because it demands performance objectives that are not actualized again in the finals. From our discussions with other teams at the event, we found that a significant number had designed specifically for the site visit—often with surrogate platforms, sensors, or algorithms—knowing in advance that they would operate different systems if they were invited to participate at NQE.
- From our perspective, we would encourage any potential future event to create mission goals that are both clearly defined and realistic. Conversely, we do understand the opposing perspective, in that specifying requirements too succinctly can result in less-innovative solutions. Although we felt DARPA did an excellent job of conveying goals at this Challenge, we also feel that the goals were not entirely representative of a practical mission application. Our assumption is that maps, with ample metadata, will exist for both automotive and military applications. Referring back to an example shown at the Washington briefing, it seems highly improbable to expect a robot to stop within 1-m registration of a stop line at one intersection, when the neighboring intersection is completely devoid of GPS coordinates and requires the vehicle to execute a 90-deg turn through it. Corporate entrants, such as IVS, are driven by a production mindset demanding system reliability, redundancy, and robustness and as such are already prone to overdesign for the Challenge, without the added burden of trying to correctly guess in advance what the metrics for success will be.
- Similarly, corporate teams are often disadvantaged with respect to universities or

military contractors, wherein the metrics for success are very different. Universities can draw upon a vast pool of inexpensive, talented, and highly motivated labor, and there is very little downside to not performing well, as they are, after all, "just a bunch of students." On the other side of the coin, corporate teams must justify the high costs of (very long-range) internal research and development and carefully weigh the potential rewards vs. the numerous risks, ranging from liability to negative publicity. Given that major corporations, and not universities, are ultimately going to deliver military and commercial hardware solutions, we would encourage DARPA to consider how to better engage their participation without making all but the winner appear to be losers.

- Testing in a realistic environment is absolutely critical to uncovering system weaknesses ranging from flaws in logic to bugs in algorithms. A thousand laps in a parking lot is no match for a mere few blocks of urban roadway. Unfortunately, finding safe and secure test facilities requires connections, time, and money. The IVS team was fortunate to have tested at more than half a dozen locations prior to NQE, yet one of our most critical bugs was not realized until we attempted Area C. It was difficult to test potential fixes to this flaw, however, as the practice areas at NQE did not contain representative features of the UCE, one of the very few disappointments we had with DARPA's execution of this event. It would have also been useful if DARPA had allowed teams a couple of days in which to attempt the courses after the UCE was complete, so as to close the loop on the learning process. On the basis of our mutual testing with MIT prior to the NQE, we are convinced that if DARPA could arrange for a common testing venue for all teams, autonomous ground vehicle technologies would advance at a much faster pace.
- One of the lessons we learned, and not necessarily by choice, was that the vehicle system does not need to be too complex to accomplish an amazing amount of autonomous behaviors. Whereas we did drive several portions of the course with only the INS and Velodyne LIDAR, we would not necessarily advocate implementing a system without added redundancy. It should further be noted that we did not even come close to fully exploiting the capabilities of the LIDAR, particularly in light of the incomplete developmental work on the intensity channel data from the Velodyne HDL-64E. If this hardware/firmware were reliably functioning, one could essentially derive black and white vision simultaneously from the unit and apply the wealth of existing image processing algorithms to the data to significantly expand sensing capabilities. We are looking forward to pursuing this area of research in the near future.
- When the IVS team initially started testing the Velodyne LIDAR, we frequently lost GPS reception and hence the INS pose information that was necessary for correcting sensor data in our map. Given our close working relationship with Velodyne, we were able to rapidly validate that the HDL-64E was indeed generating sufficient EMI to jam the electronics on our NovAtel GPS receiver. To solve this, it was deemed necessary to mount our GPS antennas above the Velodyne and to use a choke-ring design that minimized multipath interference from below the antenna phase center. Without data it is impossible to prove, but we believe that many of the difficulties encountered by other teams during the prefinal practice sessions were due to EMI emanating from the many Velodyne units. There was some anecdotal evidence that other electronic devices could also interfere with our system, including 802.11 wireless communications from laptop computers in the test vehicle. Although it was not employed at NQE, our secondary e-stop system was known to fail if more than one 2.4-GHz device (hand-held radios) was keyed simultaneously, something we actually encountered during the site visit. In a similar vein, the hand-held radios used by DARPA during NQE/UCE were powerful enough to cause the Velodyne units to drop Ethernet packets. (This was first observed by Stanford and later verified in the lab by Velodyne.) If we are to allow the fate of the vehicle to rely on a stack of electronics and not a human driver, it is

clear that more care must be taken in the future to properly address EMI issues.

- During our prerace testing, particularly when we were collaborating with MIT, we came to appreciate the importance and power of customized software toolsets. There were several notable tasks that one team or another could do within minutes, whereas it would take the other team hours to accomplish. Lots of time can be expended laying waypoints on maps, creating RNDFs, or visualizing data, to cite but a few examples. Perhaps DARPA could solicit contributions from the participating teams in this regard and create a public domain repository of available tools, so that each subsequent effort is not slowed by these mundane tasks. On a similar note, we would like to extend kudos to DARPA for supplying aerial imagery of the Victorville facility in advance of NQE and for allowing us to preview the UCE course prior to race day.
- An inspection of the entrants from the three Grand Challenges reveals that, with rare exceptions (most notably the first-generation Velodyne LIDAR), most of the hardware and sensors utilized were essentially off-the-shelf technologies. It is clear that the cutting edge of autonomous vehicle research really lies in the algorithms and software architecture. As such, the customized construction of a by-wire vehicle platform could be viewed as an unnecessary distraction, and an interesting future twist for a DARPA Challenge might be to outfit each of the teams with an identical platform and see what they could accomplish by virtue of innovative software alone. This places the competitors on even ground and is somewhat akin to the DARPA PerceptOr program. (Of course, this is the converse of what IVS and MIT did this year, i.e., run common code on vastly different platforms.) Given the success of the Gray Team at the last Challenge and VTU at the DUC (and with some biased self-promotion), we might suggest the Ford Hybrid Escape as a platform that is by-wire capable with minimal modifications.

7. CONCLUSION

In conclusion, we have demonstrated the successful operation of an autonomous vehicle capable of safely maneuvering in simulated urban driving conditions. Moreover, we have achieved this, to varying degrees of driving complexity, with the implementation of two very different computer and software architectures. Our switch to MIT's architecture, which included a substantial amount of hardware reconfiguration, was accomplished in a span of less than 2 months and not only demonstrated the versatility of their code but also our resolve and devotion to completing the mission. Although we have only partially explored the bounds of what is possible via autonomous vehicle operations, we have learned a great deal and have ample reason for optimism. Although we have estimated, on the basis of our performance and that of the other contenders, that the capabilities of present-day robots are still orders of magnitude inferior to those of human drivers, we have witnessed a rapid progression in autonomous technologies, despite the relatively short developmental time afforded to the teams that participated in the Urban Challenge. As such, we anticipate that this general trend will continue and foresee that many of the lessons learned from this and similar endeavors will soon find their way into commercially available automotive safety features.

ACKNOWLEDGMENTS

We thank our colleagues at the University of Michigan and at the Massachusetts Institute of Technology. Without the incredible support of these organizations, we would not have been able to complete our project goals. It is a testament to the quality of the MIT code that we were able to install it on a completely different platform, with a completely different sensor set, and demonstrate advanced autonomous driving behaviors within months of implementation. Together, we learned a great deal and hopefully planted the seeds for many future collaborative efforts.

Finally, we sincerely thank DARPA for conducting what we believe to be the best Challenge to date. It has been a privilege and honor to participate in this event. Although we dealt with an inordinate amount of adversity and ultimately may not have produced the results we had hoped for, we were nonetheless thrilled to once again make it all the way to race day. The understanding and support we received from the DARPA personnel no doubt contributed significantly to our success. We look forward to productive

future interactions with DARPA as opportunities become available.

REFERENCES

DARPA (2008). Various archived data and written accounts found on the Grand Challenge website, http://darpa.mil/grandchallenge.

Fischler, M., & Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. Communications of the ACM (Association for Computing Machinery), 24, 381–395.

Frost & Sullivan. (2005). Japanese passenger car and passive safety, systems markets, Report 4B79–18.

Intelligent Vehicle Safety Technologies. (2005a). Intelligent vehicle safety technologies 1—Final technical report. Submitted by W. Klarquist & J. McBride, August 29, 2005, and archived at http://www.darpa.mil/grandchallenge05/TechPapers/IVST.pdf.

Intelligent Vehicle Safety Technologies. (2005b). User manual, M. Rosenblum, November 11, 2005. Littleton, CO: PercepTek, Inc.

Intelligent Vehicle Systems (2006). Team proposal for the DARPA Urban Challenge, Proposal for BAA 06-36, submitted by Honeywell Laboratories, June 26, 2006.

Intelligent Vehicle Systems (2007a). DARPA Urban Challenge technical paper, submitted on behalf of the IVS Team by J. McBride, April 13, 2007, and archived at http://www.darpa.mil/grandchallenge/TechPapers/Honeyell_IVS.pdf.

Intelligent Vehicle Systems (2007b). DARPA Urban Challenge final report, submitted by J. McBride, December 22, 2007.

Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S., Koch, O., Kuwata, Y., Moore, D., Olson, E., Peters, S., Teo, J., Truax, R., Walter, M., Barrett, D., Epstein, A., Maheloni, K., Moyer, K., Jones, T., Buckley, R., Antone, M., Galejs, R., Krishnamurthy, S., & Williams, J. (2008). A perception-driven autonomous urban vehicle. Journal of Field Robotics, 25(10), 727–774.

NHTSA (National Highway Traffic Safety Administration, U.S. Department of Transportation). (2005). Traffic Safety Facts 2005, from http://www-nrd.nhtsa.dot.gov/Pubs/TSF2005.PDF.

National Safety Council. (2004). Mortality classifications. http://www.nsc.org/lrs/statinfo/odds.htm.

Volpe (2007). In Pre-crash scenario typology for crash avoidance research, Volpe National Transportation Systems Center, Project Memorandum, DOT-VNTSC-NHTSA-06-02, DOT HS 810 767, April 2007, http://www-nrd.nhtsa.dot.gov/departments/nrd-12/pubs_rev.html.