DARPA Urban Challenge

1 June 2007

Team Sting

Authors:
Tucker Balch (Georgia Tech) (tucker.balch@cc.gatech.edu, 678-523-8685)
Henrik Christensen (Georgia Tech) (hic@cc.gatech.edu, 404-385-7480)
Vince Camp (Georgia Tech)
Tom Collins (Georgia Tech)
Magnus Egerstedt (Georgia Tech)
Michael Hoffman (SAIC)
Ben Johnson (Georgia Tech)
Karl Kluge (SAIC)
Donald Logston (SAIC)
Matt Powers (Georgia Tech)
Robert Schafrik (SAIC)
Joshua Witter (SAIC)
Dave Wooden (Georgia Tech)

EXECUTIVE SUMMARY

Sting Racing, a collaboration between the Georgia Institute of Technology and SAIC, has designed and implemented an unmanned system for entry into the DARPA Urban Challenge. The vehicle is a Porsche Cayenne retrofitted for complete computer control, and using a combination of camera, radar and LADAR data to generate situational awareness. The challenge of operating in an urban environment with other traffic requires adoption of mixed strategy with deliberative mission execution and reactive handling of traffic and structures in the environment. In response to these challenges, a novel hybrid system architecture has been proposed that identifies a number of high-level modes of operation to be used throughout the mission. Within each mode of operation, several possible variations are possible, which is modeled by a number of different control behaviors combined through a voting-based fusion strategy. A key finding is that, based on extensive tests using simulations, small-scale test sites, and facilities for urban traffic training, this modular architecture seems to be able to address the challenges posed by the Urban Grand Challenge.

1 Introduction and Overview

1.1 Team Sting

Georgia Tech and Science Applications International Corporation (SAIC) have joined to form Team Sting in order to implement and test a reliable autonomous ground vehicle capable of safely operating in a dynamic urban environment and winning the Urban Challenge. Collectively, we bring significant autonomous robotics experience to this program. Team Sting's experience begins with the DARPA ALV program, and continues through DARPA's Demo II and Demo III, TMR, MARS and MARS 2020, PerceptOR, SDR, LAGR, and Grand Challenges I and II. Key subcontractors and vendors include Telcordia (software assurance), EMC (vehicle actuation), and Eaton Vorad (automotive collision warning radar), bringing together players capable of providing reliable state-of-the-art capabilities in all the key technologies required for the DARPA Urban Challenge.

1.2 Identification of key problems in the Urban Challenge

The two previous grand challenges organized by DARPA emphasized autonomy and robust operation in cross-country off-road environments [1][2]. The environment was assumed to be static, with few or no moving objects. If other vehicles were encountered, one of them would be paused while the other vehicle continued its route towards the goal. The desired route to be followed was defined by a relatively dense list of waypoints rather than by perceptual features (i.e., roads) in the environment. The objectives of the previous challenges therefore focused on endurance, robustness to local sensory dropouts, and trajectory following within a corridor defined by waypoints, with local deviations to accommodate static obstacles. As witnessed by the number of finishers in the last Grand Challenge, the lower level sensing, control, and vehicle reliability required to drive between waypoints while avoiding sparse static obstacles are now largely solved problems [1][2].

The Urban Challenge (UC) poses a number of very different higher-level cognition challenges for the design of a system. First of all, navigation must be performed with respect to locally defined structures such as lane-markings, stop lines, etc. Driving is required to perform lane keeping in situations with widely spaced waypoints. The vehicle is required to come to a stop at a stop line. Navigation must be performed relative to these markings, not with respect to global coordinate frames as defined by GPS. In addition, global position estimation methods such as GPS might have limited availability. In short, instead of being told where it is relative to a detailed path to follow, the vehicle must reason as to its location and the associated appropriate control responses.

Another major challenge in terms of urban driving is handling of traffic, where the exact behavior of other vehicles is not known. The implications are several, and vehicles must

- perform anticipatory planning based upon a shared set of behavioral guidelines,
- perform dynamic planning in the presence of moving vehicles in the vicinity,
- detect routes of safe passage in the event of objects blocking the present lane, and
- queue at intersections and points of congestion.

In contrast to earlier Grand Challenges, the vehicle is required to show situational awareness of dynamic as well as stationary vehicles and structures within changing areas around the vehicle. Situational awareness is required to allow the vehicle to plan its actions in response to the context. For example, if a slow moving vehicle is in front of the car, and the lane marking is a double yellow line, then following at an appropriate distance is the correct action. But the same situation with a slow moving vehicle alone in a lane with *dashed* lane dividers might allow an overtake maneuver, provided there are no vehicles in front and there are no oncoming vehicles with the segment needed for passage. For the overtake maneuver, there is a need for long-range detection of vehicles in other lanes to ensure safe passage. At intersections there is a need to detect vehicles that are waiting or approaching, which calls for long-range lateral coverage.

In earlier competitions, the trajectory to be traversed was specified in detail, including tolerances on a corridor within which the vehicle had to remain. In the Urban Challenge, the density of points in the RNDF is sparse. In addition, the mission definition file (MDF) specifies a number of "waypoints" to visit during a mission. The detailed planning of segments to travel along to ensure passage through the specified set of "points" is not defined *a priori*. The vehicle is required to generate a strategic plan for execution of a mission, which details a number of road segments and intersections to pass for completion of a mission. Once a strategic plan is in place, the system is required to navigate each chosen segment using particular modes of operation to accomplish the mission. In a dynamic world, the vehicle might encounter difficulties such as blocked segments and slow moving traffic. Such events might require online revision of the strategic plan to ensure efficient execution of a particular mission. In contrast to earlier competitions, the vehicle is required to perform strategic planning, dynamic assessment of plan feasibility and replanning as needed to handle unplanned events.

In summary, the key problems to be addressed as part of the Urban Challenge include:

- Navigation by local analysis of scene content relative to behavioral goals rather than by map-based navigation using absolute geospatial coordinates.
- Dynamic planning and replanning of missions – deciding what to do deliberatively and then re-planning as needed in response to dynamic events in the environment.
- Operation and planning in the context of uncertainty due to the presence of other moving agents in the world with only semi-predictable behavior.
- Handling "emergency" unexpected events that require an immediate response.

Our architecture to address these challenges is based upon the assumption that the required capabilities can be broken down into a small (enumerable) number of operating modes, each mode consists of a collection of parameterized behaviors and a behavior arbitration mechanism. This modularization makes design and development tractable, as well as provides a mechanism for structured, incremental testing. Traffic laws and conventions structure the world dynamics into this small set, though robust behavior within an operating mode requires being robust with respect to large variety of possibilities *relevant to that mode.* Selection of a particular mode of operation can be performed based on the RNDF and the situation derived for the presence of other vehicle and location with respect to the RNDF segment.

1.3 Sting architectural features and DARPA Technical Criteria

The set of capabilities required by the Urban Challenge are partly specified by the DARPA Technical Criteria for Evaluation. In this section the architectural features required to implement a successful entry into the competition are listed and they are briefly outlined.

Proven graph-based path planning: A standard algorithm (D*) for path planning in a graph is used. If the required order of checkpoints were not specified, planning would involve solving the NP-hard Traveling Salesman Problem. Because the order of checkpoint visitation is specified, low-order polynomial methods can be used to plan the shortest path between successive checkpoints in the RNDF. (See section 2.4.2.)

Vision-based lane detection: Lane detection and tracking is done with computer vision. Marked paved roads are detected and tracked using a hybrid approach that combines deformable template based lane detection with tracking using local feature detection and least squares shape parameter estimation. Dirt roads are tracked using a combination of geometric cues (berm detection) and vision (adaptive color-based classification). (See section 2.4.1.)

MDF/RNDF exploitation: MDF speed limits are combined with global position and RNDF information to constrain the allowable speed of the robot on lane segments as well as in safety zones and RNDF zones.

Parallel processing/networking architecture: On-board processing capability has been sized to support required level of real-time perception and decision making, with room for expansion if needed. Network architecture is doubly redundant, also with ability to expand as required. (See section 2.3.1 and Figure 4.)

Complementary range/velocity sensors: SICK LADARs provide 360º coverage within the robot's stopping distance; a combination of a high resolution/range Riegl LADAR and Eaton-Vorad EVT-300 radars provide detection and tracking of other moving vehicles at ranges over 100 m. (See section 2.3.2.)

Longitudinal control: Longitudinal control algorithms provide speed control to maintain the required speed- and zone-type dependent headway. (See section 2.1.)

Multimodal stop line detection: A combination of GPS cueing and vision is used to detect and track stop lines; dead reckoning is used once camera view of line occluded by vehicle. (See sections 2.4.1 and 2.4.2.)

Fused target tracking: LADAR range scans are segmented to detect groups of readings corresponding to obstacles and moving vehicles in the environment; these targets are tracked over time to identify and predict the behavior of other vehicles in the environment. This is integrated with target azimuth, range, and range-rate information from multiple automotive radars. This target time-history is used to identify order of arrival at intersections and emptiness of the intersection box to determine when the robot has right-of-way to proceed, and to detect appropriate gaps when turning across/merging into traffic that has right-of-way at intersections that do not have 4-way stops. (See sections 2.4.1 and 2.4.2.)

	Graph-based path planning	Vision-based lane detection	MDF/RNDF exploitation	Par. processing/netw. arch.	Compl. range/vel sensors	Longitudinal control	Multimodal stop line detection	Fused target tracking	Hybrid planner	Situational velocity control	Collision predictor	Localized decision-making
A.1 Preparation for run	X											
A.2 Mission start	X											
A.3 Checkpoints	X											
A.4. Stay in lane		X										
A.5. Speed limits			X									
A.6. Excess delay				X								
A.7. Collisions					X	X						
A.8. Stop line							X					
A.9. Vehicle separation					X	X						
A.10 Leaving lane to pass.		X									X	
A.11. Returning to lane after pass		X									X	
A.12. U-turn	X										X	X
B.1. Basic navigation	X	X	X	X	X	X	X				X	X
B.2. Intersection precedence								X				
B.3. Minimum following distance					X	X						
B.4. Queuing					X	X						
C.1. Basic traffic	X	X	X	X	X	X	X	X			X	X
C.2. Obstacle field									X			
C.3. Parking lot									X			
C.4. Dynamic re-planning	X											
C.5. Road following		X										
C.6. GPS outage		X										
D.1. Advanced navigation	X	X	X	X	X	X	X	X	X		X	X
D.2. Merge								X				
D.3. Vehicle separation during merge								X				
D.4. Left turn								X				
D.5. Vehicle separation during left turn								X				
D.6. Zones									X			
D.7. Emergency braking										X		
D.8. Defensive driving											X	
D.9. Traffic jam												X

Table 1: Mapping between Urban Challenge Technical Evaluation Criteria and Sting architecture features. The Sting architecture has been designed to address all of DARPA technical criteria.

Hybrid planner: Zones (including obstacle fields and parking lots) are navigated using a hybrid approach of behavior-based reactive control and a graph-based planner working on a local map. The reactive controllers keep the vehicle clear of obstacles and moving traffic, while the planner provides a route to either the exit of the zone or to the vehicle's parking space. Parking is handled using a precise navigation routine that plans a route for the vehicle to drive into and back out of a parking space without violating the constraints of the neighboring spaces. (See section 2.4.2.)

Situational velocity control: Lower-level velocity control module employs different controller configurations based on the situation. Emergency situations are handled with a much more aggressive controller than nominal situations, making effective emergency braking possible. (See section 2.4.2.)

Collision predictor: The vehicle's obstacle detection and tracking module identifies obstacles that are actively on a collision course with the vehicle. When such obstacles are detected, the vehicle transitions into a context-dependent active avoidance mode. See sections 2.4.1 (perception), and 2.4.2 (avoidance).

Localized decision-making: Local sensor information is used to navigate intersections that have been partially blocked. LADAR and camera data are used to detect obstructions, other vehicles, and the road, allowing the vehicle to make headway in a traffic jam. See sections 2.4.1 (perception) and 2.4.2 (reaction and navigation).

Table 1 specifies the mapping between the architectural features and the DARPA technical evaluation criteria to demonstrate the compliance of the designed system. The actual system has been implemented in a vehicle as outlined in section 2.3. The underlying top-level software architecture is shown in Figure 1. This architecture can be broadly categorized as comprising four major groups of processes:

1) *Primitive Perception*, where perceptual processes are acting on single images/scans/readings coming from individual sensors. The design of the sensor suite is described in detail in Section 2.2. with the perception module design described in section 2.4.1.
2) *Integrated Perception*, where multiple sensory sources are combined together in order to obtain more involved environmental representations. Section 2.4.1 describes this portion of the design in detail.
3) *Planning*, where mission-level decisions are made regarding what sensing modalities and controllers to use. The Planning processes consist of the Mission Mapping, Mission Planning and Task Sequencing, and Situational Awareness and Action Sequencing blocks in Figure 1. Section 2.4 provides design details of the hierarchical hybrid finite state automaton approach used for planning.
4) *Control*, where a collection of control laws and behavioral arbiters are implemented, consisting of the Reactive Behaviors, Behavior Arbitration, and Vehicle Control blocks in Figure 1 The vehicle platform and actuation system on which these control modules act is described in section 2.3.1.

2 Analysis and Design

In this section, the hardware and software design choices made by Team Sting are discussed. In particular, these choices are related to unique challenges posed by the Urban Challenge and it is shown how they map the problem characteristics to sensing, actuation, and planning modalities in a comprehensive manner.

2.1 Required modes of operation for urban driving in traffic

The novel, modular architecture employed by Team Sting was arrived at by observing that the sensing, planning, and control capabilities needed to drive down a road are fundamentally different than those needed to park the vehicle. As such, rather than choosing a single, sense-plan-act solution in which a unified planner produces references for a trajectory tracker, a number of distinctly different environments were identified, based on the unique challenges posed by the Urban Grand Challenge. In fact, the operation of the system is modeled as a finite set of "modes of operation" that each capture a nominal situation to be handled by the vehicle. Within each of these modes of operation, a dedicated set of controllers is used to handle both the nominal situation and unexpected variations.

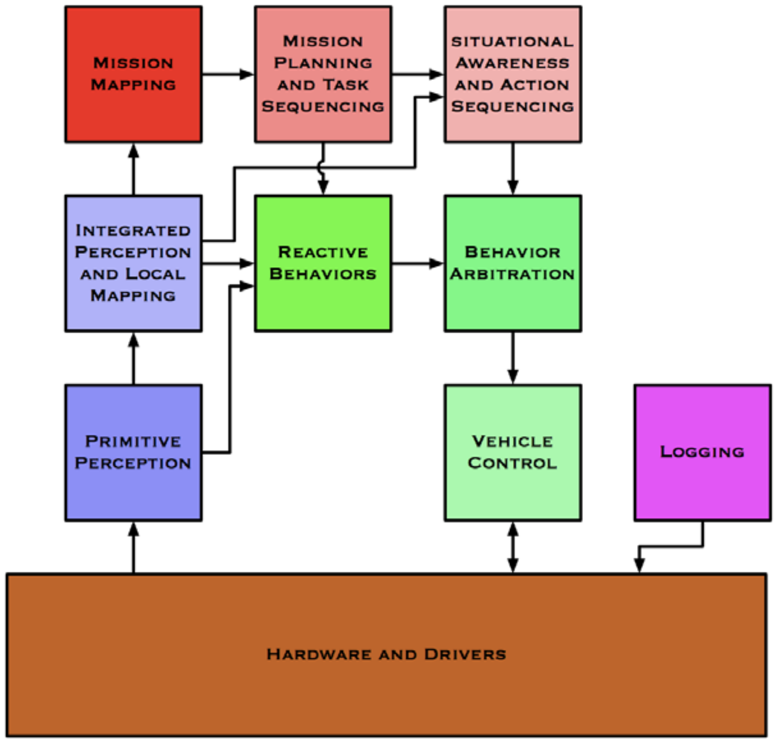

Figure 1: Sting Racing Software Architecture

Each mode of operation is represented as a hybrid automaton, as seen in Figure 2. An automaton is composed of states and transitions among the states. For example, consider a state Follow-Lanes which represents the behavior of driving along lanes on a road while obeying speed limits and recognizing the speed of nearby traffic. This state would have transitions to another state, Handle-Intersection, where the transition occurs based on a combination of the distance from the robot to the stop point (from GPS information) and other visual cues, such as the detection of a stop line.

In the left automaton in Figure 2, nodes at the highest level of abstraction are shown. These correspond to the high-level modes of operation Follow Lanes, Overtake Static Obstacle, U-Turn, Handle Intersection, Park, and Unpark. Based on the specifications of the Urban Grand Challenge mission, these are the six modes of operation that are selected by Team Sting as the minimal set of modes needed to successfully complete the mission. An important additional benefit, however, associated with the modular design is that new modes can be added, whenever the need arises further down the development cycle. Indeed, encapsulation and ease of extendibility are a key features of this software architecture and important to the short development cycles required for the Urban Grand Challenge.

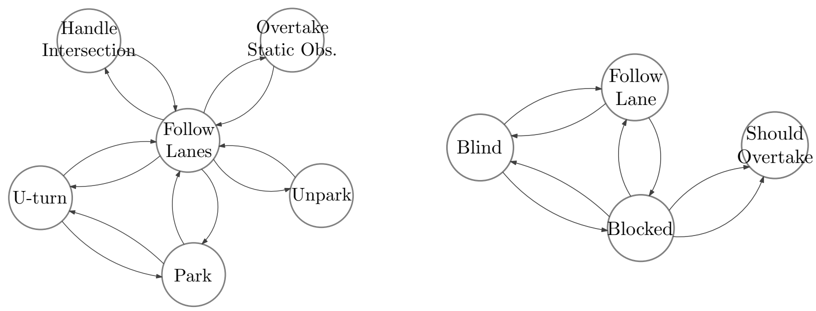

Figure 2: Modes of operation modeled as a Hybrid Automaton

The transitions between modes are guarded in the sense that environmental conditions trigger the transitions. As such, the situational awareness component of the novel Team Sting architecture can be thought of as the guard conditions (or transition conditions associated with the different edges in Figure 2), and the cognition component is encoded by the underlying state machine dynamics. And, for the sake of easy reference, each of the modes of operation are roughly described. A more detailed description is given to the Follow Lanes mode of operation. The remaining modes are discussed only cursively.

2.1.1 Follow Lanes

In the right figure of Figure 2, the Follow Lanes mode of operation is given. Here, each node corresponds to a particular set of behavioral controllers as well as to a particular arbitration mechanism. In fact, the modes that make up this high-level mode are

- *Follow Lane*: This mode corresponds to a set of behaviors that use visual perception to track lane striping and that use fused LIDAR and radar data to track nearby traffic, thereby adjusting speed and avoiding collisions.
- *Overtake*: Typically, transitions between the states are based on environmental or perceptual information. *Overtake* mode, however, is a state-based signal to switch from the larger *Follow-Lanes* model into the *Overtake-Static-Obstacle* model. This mode corresponds to a command to the behavior arbiter to stand still until the *Overtake-Static-Obstacle* model is enabled (as described in Section 2.1.2).
- *Blocked*: This mode uses the same behavior arbiter as the *Follow-Lane* mode. However, it has a transition based primarily on time. If this mode is active for a parameterized amount of time, it transitions to *Overtake*, which then signals the robot to overtake a static obstacle.
- *Blind:* This mode corresponds to a behavior arbiter that uses GPS and laser information to drive in the lane because the lane detector has failed in some way. Fused laser and radar data is used to avoid collisions and maintain speed in the lane.

2.1.2 Overtake Static Obstacle

This mode of operation governs the control of vehicle during a maneuver to overtake a static obstacle. The four modes comprising this high-level mode include the following: *Init-State*, *Change-Left*, *Change-Right*, and *Done*. *Init-State* establishes a fixed coordinate frame to govern the transitions through the subsequent modes. *Change-Left* and *Change-Right* correspond to the tracking of lane markings one lane to the left or right, respectively of the current lane being tracked. That is, the lane change maneuver is achieved primarily by shifting visual perceptual attention on the road. The lane change commands are triggered based on a combination of distance travel (relative to the coordinate frame established in *Init-State*) and the presence/absence of obstacles from fused LIDAR/radar data.

2.1.3 U-Turn

The states of this high-level mode encode a mapping from vehicle orientation (i.e., position and heading) to output primitive (e.g., drive forward, hard left; drive in reverse, hard right). This mapping stabilizes the vehicle (in the presence of imperfect vehicle control) to the desired final position and heading.

2.1.4 Handle Intersection

Intersections are handled by cycling through a string of simple modes: *Approach*, *Find Queue Position*, *Wait For Turn*, *Go*, and *Done*. *Approach* smoothly brings the vehicle to a stop at the stop line based on visual perception of the lane markings, while queueing behind other vehicles. Once stopped, *Find Queue Position* establishes the robot's precedence order based on fused LIDAR/radar data. *Wait For Turn* checks the interior of the intersection for traversals by the adjacent vehicles with higher precedence. Once its turn has come, *Go* is triggered, and the robot traverses a path through the intersection towards the entry point back onto the lane segment.

2.1.5 Park and Unpark

This pair of high-level modes guides the robot through RNDF zones and in and out of parking spots. These states govern the path of the robot (e.g., to drive it to a parking spot) given the constraints of Ackermann steering and encode the rules of driving in the unstructured RNDF zones (e.g., pass to the right of oncoming traffic).

2.2 Sensor requirements for urban driving in traffic

In this section, a brief discussion is given as to what sensory modalities were selected by Team Sting. The selection of the different sensing modalities was largely driven by a task-based decomposition of the mission. Five representative tasks were analyzed: driving in a lane with traffic, driving through intersections, driving around static obstacles, parking and unparking, and U-turning. For each task, the required tasks sensing modalities listed were analyzed in terms of their sensing requirements in an urban environment. (The quantitative analysis associated with the actual sensors and the corresponding performance requirements are given in section 2.3.2 as part of the analysis of the hardware design.)

2.2.1 Requirements for driving in lane with traffic

A significant portion of the Urban Challenge mission involves traveling along maintained roads among other moving traffic. To stay on the road, and in its respective lane, the vehicle must be able to detect the visual markings on the road or, in the case of an unmarked road, the edges of the road. The color, pattern, location and geometry of the road markings are all of interest. Additionally, to maintain appropriate separation from other traffic, the vehicle must be able to detect the location and relative velocity of other vehicles on the road. High frame rates and long ranges are especially important for all sensors used in this context, as the vehicle's highest speeds will be achieved driving along marked, paved roads.

2.2.2 Requirements for driving through intersections

Intersections pose a unique challenge to an Urban Challenge vehicle in that lane markings are not available in the middle of the intersection, yet the vehicle must navigate from one road to another, meeting the constraints of the destination road's lane markings. While the RNDF provides the location of the intersection entry and exit points in a global frame, GPS alone is unlikely to provide enough information to traverse the intersection, as the position accuracy with respect to the lane markings may not be enough to guide the vehicle into the lane, and GPS quality in an urban environment may be compromised. Though the center of the intersection lacks strong exteroceptive queues, differential sensors may be used over such short distances to measure progress through the intersection. Given that the vehicle can achieve the first point, the robot can use differential sensors, such as inertial measurement units or wheel encoders, to navigate the difference to the next point, where lane markings will again guide the vehicle.

Navigating the intersection using differential sensors is preconditioned on finding the beginning of the intersection. Four-way intersections in the Urban Challenge will be marked by visual stop lines on the road. To traverse the intersection, the vehicle must be able to detect and stop at stop lines denoting the beginning of an intersection. Then, to exit the intersection, the vehicle must attain and stay in the outgoing lane as it leaves the intersection. Both these requirements involve detecting visual road markings and describing their color, pattern, location and geometry.

2.2.3 Requirements for handling static obstacles

While much of the Urban Challenge course is well structured, (paved roads, lane markings, etc.) it is still necessary for the vehicle to navigate around obstacles that may be blocking the road or passage through a zone area. The vehicle must be able to detect the location and size of static obstacles and in order to navigate around them.

2.2.4 Requirements for parking and unparking

In contrast to driving along roads, finding the vehicle's assigned parking space is mainly a globally-defined task. Lacking obvious lane markings in parking lots and other zones, to find its parking space, the vehicle must navigate to a globally-defined position, without the aid of road markings. Accurate global localization (e.g., GPS) is useful in this respect. Additionally, there will be other moving (and static) vehicles in the parking lot. It is important to be able to detect the range, bearing and velocity of these other vehicles in order to avoid collisions in the parking lot.

2.2.5 Requirements for U-turn maneuver

Just as the vehicle may have to navigate around static obstacles along the road, it may come upon obstacles that totally impede progress along the road. In this case, the vehicle must perform a U-turn to backtrack along the road. Before executing the U-turn, it must detect traffic in the oncoming lane to confirm the road is clear for a U-turn. This requires the ability to sense the range, bearing and velocity of oncoming traffic at a significant distance. While executing the U-turn, the vehicle may not be able to sense the road edges at all times, and thus must stay within the boundaries of the road by other means.

2.3 Vehicle Design and the Sting Hardware Systems

2.3.1 The Sting-1 robotic platform

The robotic vehicle (Figure 3) prepared by Team Sting is based on a 2006 Porsche Cayenne modified for control via on-board computers. An AEVIT driving control system integrated by Electronic Mobility Controls (EMC) provides primary servo control of steering and acceleration/braking as well as controls for secondary functions, including ignition, transmission, lights, and the parking brake.

A Gigabit network of 8 Dual-Core Intel XEON 5120 processor-based computers provides the processing power for interface to the suite of sensors and software control. Power for the AEVIT servo and vehicle system controls is provided by the Cayenne 12V DC system. All sensors and associated computer equipment are powered from an auxiliary, engine-driven, 24V DC alternator configured to provide 75A at idle. This exceeds the sensor suite and computer equipment power requirements of 50-55A, 24V DC during operation and provides additional power for future requirements. Batteries, contactors, circuit breakers, and the power distribution panel for the 24V system are located in the rear vehicle compartment with the computer rack. DC-DC converters are installed to provide power for sensor and peripheral equipment requiring 12V or 5VDC. Figure 4 shows a block diagram of the sensor, processor, and actuation hardware.

Figure 3: The Sting-1 robot during testing at Georgia Tech.

Roof-mounted equipment includes an amber safety strobe and emergency stop pushbuttons accessible from either side. The safety strobe and an audible warning signal provide awareness during autonomous operation. Antennas are installed for remote pause and disable receivers, and an 802.11G WLAN interface is provided for developmental purposes. Other safety-related equipment includes internal pause and emergency stop controls and the circuitry needed to disable the vehicle, sensor suite and computer equipment when commanded.

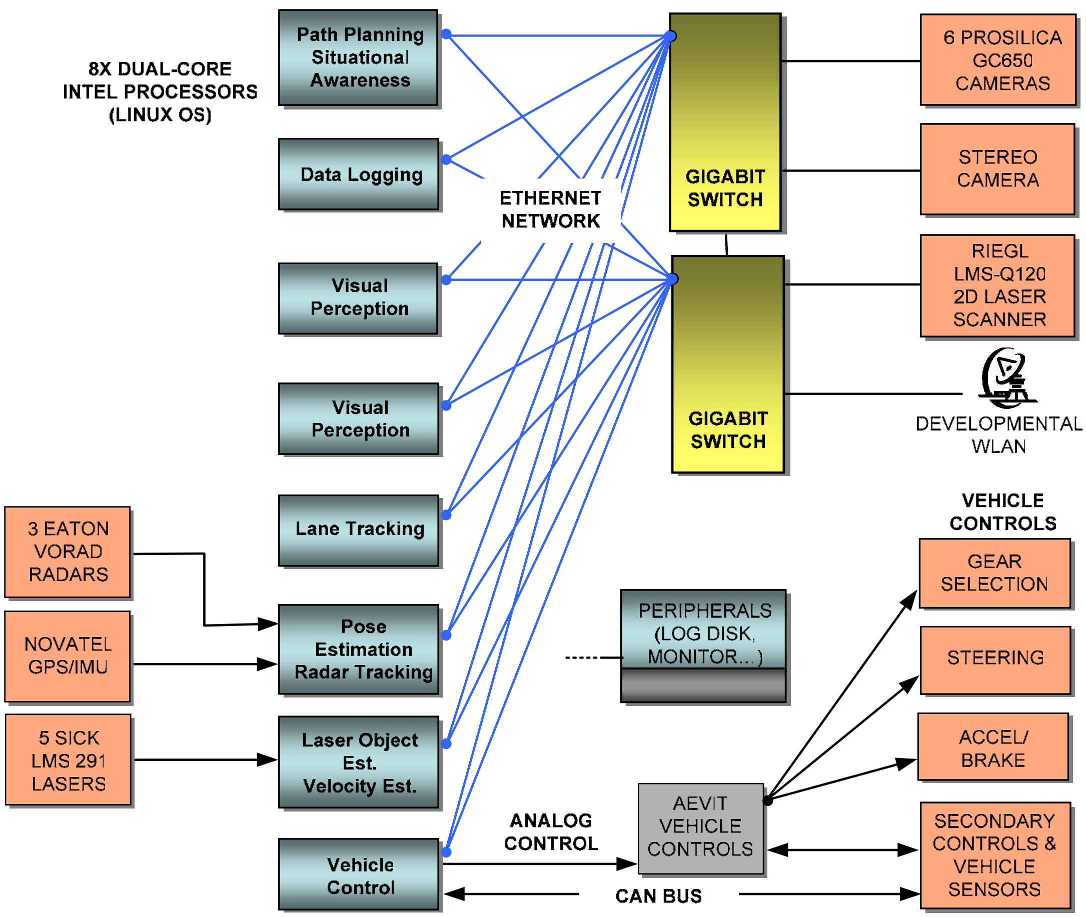

Figure 4: Sting-1 sensor and processing configuration

2.3.2 Selection of Sensors

Obstacle Detection and Tracking. Selection of sensors for obstacle detection and tracking for the Urban Challenge was driven by the requirements derived from the rules as shown below in Table 2.

Required obstacle detection range for the *Basic Navigation*, *Basic Traffic* and *Advanced Navigation* criteria are dominated by the vehicle (non-emergency) stopping distance of ~33 m. The *Advanced Traffic* criteria require detecting and tracking other vehicles in on-road settings at ranges of up to ~100 m. Note that in realistic urban environments, complete 360-degree coverage to that range is not required, as occlusion constraints due to cultural features (buildings, poles, etc.) limit the intersection geometries at which it would be reasonable for traffic control measures (i.e., stop signs or traffic lights) to be absent for safety reasons.

Direction	**Constraint(s)**	**Required max range**
Forward	Stopping distance (speeds <= 30 mph; min separation of 2 m in safety areas; assuming 0.5 sec latency, 0.5 g braking)	33.0 m
	Headway maintenance (min separation of 1 vehicle length/10 mph in travel areas, speeds <= 30 mph)	14.4 m
	Gap detection for turning across oncoming traffic (minimum of 2 vehicle lengths forward separation from oncoming vehicles, speeds <= 30 mph, 5 seconds to complete turn) [*Advanced Traffic*]	97.7 m
Side	Minimum standoff from obstacles of 1 m on all sides of vehicle in all areas	1 m
	Gap detection for turning across oncoming traffic (minimum of 2 vehicle lengths forward separation from oncoming vehicles, speeds <= 30 mph, 5 seconds to complete turn) [*Advanced Traffic*]	97.7 m
Rear	Vehicle returns to travel lane 1-4 vehicles lengths beyond obstacle when completing a passing maneuver; trailing vehicle not required to slow down when completing passing a moving vehicle (assume 5 m length for vehicle being passed)	15 – 20 m
	Gap detection for merging, leaving 1 vehicle length/10 mph to trailing vehicle after merge completed [*Advanced Traffic*]	70 m

Table 2: Obstacle detection sensor range requirements derived from the Urban Challenge rules.

Our approach is to translate these requirements to a set of ranging sensors that reliably provide either (a) multiple lateral range readings on any obstacle, or (b) reliable tracking of obstacles as targets with associated range and velocity. In both cases, this sensing must be reliable at the maximum expected distance of such objects, as defined by Table 2. Condition (a) provides the ability to discriminate obstacles and identify their shape and extent, most suitable for close ranges. Condition (b) allows for sensors that have less spatial resolution, but can discriminate based on obstacle velocity (suitable for distant objects and thus complementary).

LADAR Sensors. Three LADAR sensors emerged as candidates, and all were evaluated in tests. The SICK LMS 291 is a 2-D scanner (a rotating scan in a single plane), as is the Riegl LMS-Q120. The 3-D scanner considered was the Velodyne HDL-64E. Several team members had extensive experience with the LMS 291 and Riegl sensors, and test data for a LMS 291 was actually included in our original proposal. A rental Velodyne unit was evaluated, and satisfactory results were achieved, but it was removed from consideration for several reasons, including 1) the unit was an early production model, possibly subject to unexpected failures; 2) While range for a LADAR is difficult to quantify over all possible conditions, the range of the Velodyne seemed lower than 2-D scanners, especially the Riegl; 3) For less cost, we could arrange multiple 2-D scanners in such a way as to provide the necessary range readings at multiple elevations; and 4) Such an alternate arrangement of multiple 2-D scanners would provide greater redundancy.

The SICK LMS 291 and the Riegl LMS-Q120 sensors turned out to be complementary, with the LMS-Q120 having greater range while the LMS 291 was more cost-effective for placement in multiple vehicle locations. Table 3 shows the LADAR characteristics, including calculations of the number of lateral readings on objects at the maximum distances found in Table 2. (This is effectively a translation of angular resolution to the worst-case obstacle locations.) Based on this, LMS 291 LADARs were chosen as the primary sensors for detection and tracking of stationary or moving obstacles within the required stopping distance. A roof-mounted LMS 291 provides a 90º sweep of the road about 20 meters in front of the vehicle. A second roof-mounted LMS 291 is oriented vertically to provide data used to assist vision road-detection algorithms. Side-mounted LMS 291 sensors provide a 180º scan on either side of the vehicle. A rear bumper-mounted LMS 291 provides object detection capability behind the vehicle. While the vehicle is stationary at intersections, a LMS-Q120 can provide high-resolution target information at ranges up to 150 m at low scan rates within the 80 degrees directly ahead of the vehicle. Furthermore, while the vehicle is moving, the same LMS-Q120 can provide enhanced resolution for obstacles within the stopping distance at a 25 Hz rate.

Sensor	**Max range (m)**	**HFOV**	**Scans/sec**	**Wavelength or Frequency**	**Lateral range readings on 1.5 m wide obstacle**	
					@ 33 m	**@ 100 m**
SICK LMS-291 (4 scanning horizontally, 1 scanning vertically)	80 m	180°	19 @ 0.25° resolution; 38 @ 0.5° resolution	905 nm (Class 1 eye-safe)	10 @ 0.25° resolution, 5 @ 0.5° resolution	N/A
Riegl LMS-Q120 (1 pointing forward, scanning horizontally)	150 m @ 80% target	80°	5 @ 0.04° resolution; 25 @ 0.2° resolution	Near IR (Class 1 eye-safe)	65 @ 0.04° resolution; 13 @ 0.2° resolution	21 @ 0.04° resolution
Eaton Vorad EVT-300 (1 on front, 2 on sides)	110 m (motorcycle -sized target)	12°	15	24.725 GHz	N/A – tracks up to 20 objects, returning azimuth, range, and range rate	

Table 3: Sting obstacle detection sensor characteristics.

Radar sensors. For collision warning radar we selected an Eaton Vorad EVT-300 as a means of providing reliable detection for oncoming vehicles at large distances. The EVT-300 emits 3mW of RF power at 24.725 GHz, with 1 MHz bandwidth. Able to detect and track as many as 20 objects at up to 150 m, it can update seven tracks at 15 Hz. A motorcycle-sized target can be detected at distances as great as 110 m. Target range (+/- 3 feet), velocity, and azimuth are included for each tracked target.

With approximately a 12º field of view, multiple EVT-300 units are required to provide adequate warning for the front and sides of the vehicle. One front-mounted unit provides redundancy for the Riegl, as well as the complementary capability to maintain tracks independently of other system software. Two side-mounted units are placed near the opposite ends of the front bumper to provide indication of approaching traffic as the vehicle pulls into an intersection or out of a side road. Additional side-facing units may be mounted to increase the field of regard covered. Figure 5 below shows the current sensor configuration mounted on the vehicle. Figure 6 provides a plan view of sensor coverage.

Figure 5: Bumper-level sensors (shown at left) include side-looking SICK LMS 291 LADARs and EVT-300 radars and a forward-looking Riegl LMS-Q120 LADAR and EVT-300 radar. Roof mounted sensors include 6 Prosilica GC 650 color Gigabit Ethernet cameras and 2 SICK LADARs (1 horizontally, 1 vertically).

Sensing for lane keeping. Six Prosilica GC-650 Gigabit Ethernet color cameras provide images to the front, sides and rear of the vehicle. They are capable of providing 12 bits of intensity dynamic range per pixel, and the fast Ethernet interface supports high frame rates. All 6 cameras are roof-mounted in environmentally sealed enclosures. The two forward-looking cameras are used for road following.

Sensing for vehicle position and pose estimation. Sting-1 uses the Novatel SPAN inertial navigation system, consisting of the Novatel Propak G2plus GPS receiver with roof-mounted antenna and a Honeywell HG1700 AG17 inertial measurement unit mounted centrally in the vehicle. The GPS unit receives both L1 and L2 signals and is capable of pseudorange differential corrections. Currently, the only correction data is that supplied by WAAS. The IMU allows the vehicle's position to be estimated during periods of GPS dropout, as are expected in an environment with limited view of the sky. The HG1700 IMU specifications are given in Table 4. The SPAN technology described by Novatel [11] combine WAAS-corrected GPS readings with IMU data to provide position accuracy with 0.8 m CEP, velocity accuracy of 0.02 m/s RMS (nominal), attitude accuracy of 0.015º (pitch or roll) and 0.05º (yaw), and acceleration accuracy of 0.03 m/s^2. In referenced tests, an error of 1-3m was seen during a 60s dropout period, depending on how many GPS satellites were blocked. Additionally, the time to reacquire a GPS position after a dropout was improved dramatically, from 11s to 1s, by keeping an estimated position based on IMU data.

Gyro Input Range	± 1000 degrees/s
Gyro Rate Bias	10.0 degrees/hr
Gyro Rate Scale Factor	150 ppm
Accelerometer Range	± 50 g
Accelerometer Linearity	500 ppm
Accelerometer Scale Factor	300 ppm
Accelerometer Bias	3.0 mg

Table 4: IMU specifications for SPAN INS.

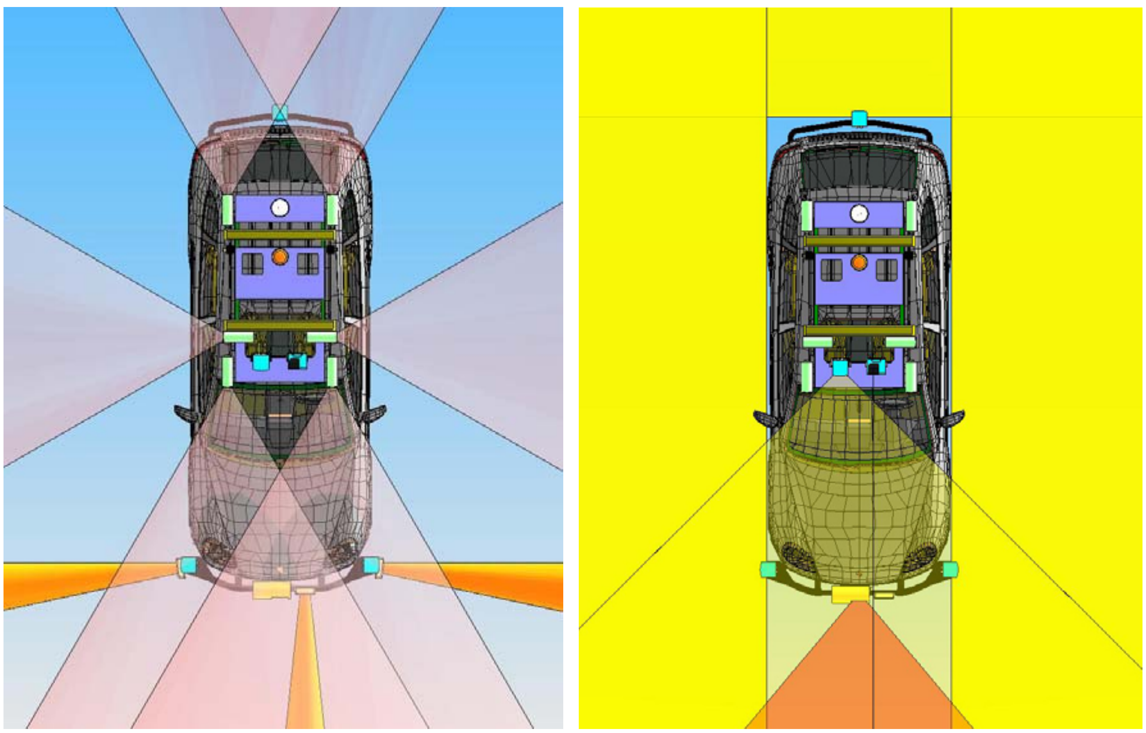

Figure 6: Sting sensor coverage.
Left – 6 Prosilica GC 650 color Gigabit Ethernet cameras (pink) and 3 EVT-300 radars (orange)
Right - 5 SICK LMS 291 LADARs (yellow) and forward-looking Riegl LMS-Q120 LADAR (orange)

2.4 Software Architecture

Figure 7 provides a more detailed view of the processes comprising the Sting software architecture shown in Figure 1. The Planning Group consists of the Mission Mapping, Mission Planning, and Situational Awareness and Action Sequencing blocks. Similarly, the Control Group consists of the Reactive Behaviors, Behavior Arbitration, and Vehicle Control blocks. This section describes the operation of these blocks in detail and outlines their functionality with respect to the key software and architectural challenges associated with the Urban Challenge.

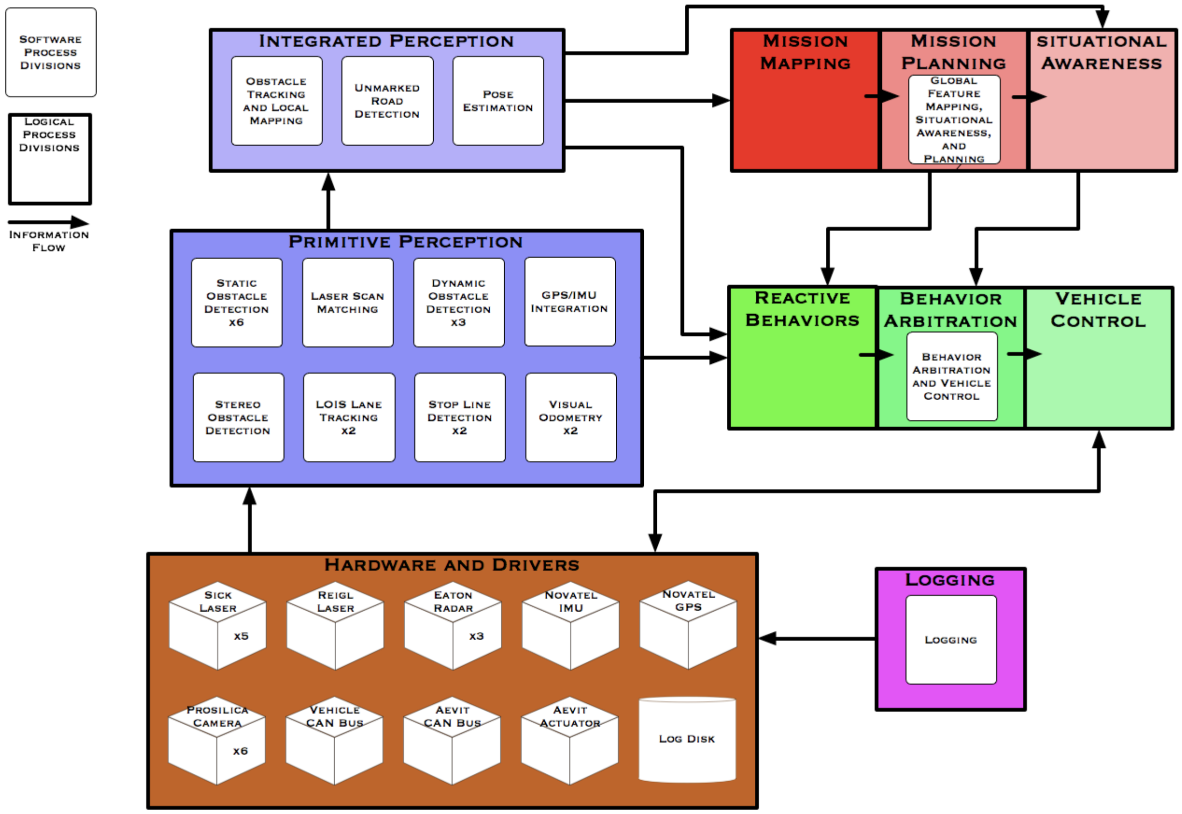

Figure 7: Software processes used within the Sting software architecture and their relationship to the conceptual architecture presented in Figure 1. Smaller boxes represent divisions of labor between software processes (e.g., Static Obstacle Detection). Larger boxes represent divisions of labor within the conceptual architecture (e.g., Primitive Perception).

2.4.1 Primitive and Integrated Perception

In order for the vehicle to estimate its own state as well as relevant environmental conditions, sensing and estimation are needed at different levels of abstraction, frequency, and fidelity. The primitive perception part of the software architecture collects and processes single scans/images/measurements from individual sensory sources. In order to arrive at a comprehensive list of perception primitives, Team Sting relied on the mission scenarios to be expected in the Urban Challenge. In particular, as safety is going to be a critically important issue, static and dynamic obstacle detection are needed as well as scan matching algorithms for obstacle classification. The dynamic obstacle detection is necessary also from a traffic management point-of-view. Moreover, as the vehicle will be operating in environments in which GPS signals may or may not be readily available, an integrated GPS/IMU primitive is needed in combination with a vision-based method for local pose estimation, i.e., visual odometry. Finally, lane and stop line tracking capabilities will also be needed in order to place the vehicle correctly in its local environment. Note that these primitives are not providing all of the perceptual skills needed, but the remaining, more complex perception tasks will be handled at the integrated perception level.

To summarize, the derived set of required primitive perception capabilities are:

- Static Obstacle Detection
- Laser Scan Matching
- Dynamic Obstacle Detection
- GPS/IMU Integration
- Stereo Obstacle Detection
- Lane and Stop Line Tracking
- Visual Odometry

The Integrated Perception functional group deals with sensor fusion, in which the data from the primitive perception group is used in an integrated fashion to achieve higher-level perceptual tasks. These tasks are Pose Estimation, Unmarked Road Detection, and Obstacle Tracking and Local Mapping. Two of the key problems associated with the Urban Challenge are driving on a road network without detailed, high accuracy information about the road location, and detecting and tracking other moving entities in the world. These critical capabilities are described in more detail below.

Vision-based road following. The Urban Challenge rules establish a number of requirements for road following:

- Lane keeping at speeds of up to 30 mph (13.4 meters/second) "…in a legal and appropriate travel lane while en route, including around sharp turns, through intersections, and while passing."
- Perception-based rather than map-based: "C.5. Road following: Vehicle navigates roads with sparse waypoints and stays in travel lane through road-following by sensing berms or road edges, or by any other sensor-based technique."
- On both paved (with variable pavement and lane stripe quality) and unpaved (dirt) roads.

Commercial lane tracking systems including Iteris' AutoVue, AssistWare's SafeTRAC, and Mobileye's ACP5 were considered, but rejected for two reasons. First, these systems have been tuned to lane departure warning tasks on highways, raising concerns about their appropriateness for the lower-speed urban setting of the Urban Challenge. Second, the proprietary nature of these systems would have made extension/modification difficult or impossible.

Team Sting's approach to detecting and tracking lanes on paved roads is illustrated in Figure 8 below. Initial detection of lane edges and acquisition of the new travel lane when crossing or turning at intersections is done using a new implementation of the LOIS Lane Detector [4]. LOIS uses a deformable template approach to find lane edges, giving it robust performance in situations with complex illumination conditions and variable quality and types of lane edges. This approach has been tested using a combination of open-loop testing on video data collected on urban streets in the Pittsburgh, PA area and closed loop tests on the sites described in the Results and Performance section below. The left side of Figure 8 shows an example result in an urban scene where the right lane edge is a curb rather than a painted stripe, and cars provide visual clutter in the adjacent lane. Initial acquisition of the lane geometry using LOIS takes ~0.1 seconds as measured on Sting's processors. The initial estimate of lane shape provided by LOIS is tracked in subsequent images by fitting a parametric model of lane shape to estimated lane edge points found using a local feature detector, similar to the methods used in [5]. The right side of Figure 8 shows the results of this method, with single stripe detections shown in green, double stripe detections shown in yellow, and the estimated lane shape outlined with red dots. Once the lane is acquired, the tracking process runs at rates of up to 30 frames/second.

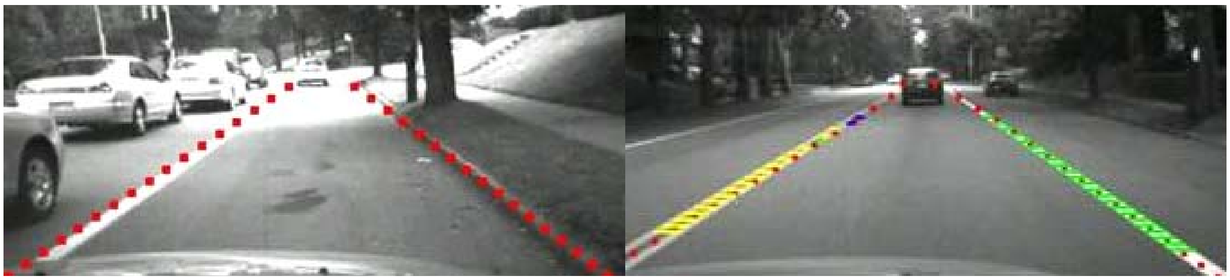

Figure 8: Paved road lane tracking used by Team Sting. Initial lane detection is done using a deformable template approach that handles a variety of lane edge types including painted lane markings and curbs (left); tracking is done at rates up to 30 frames/second by fitting a parametric model of lane shape to lane edge points found using a local detector (right)

Our planned approach to tracking dirt roads is to combine adaptive color-based methods [3][6] with extraction of geometric cues such as berms defining the road edges. A parametric model of road shape is fit to images where the value at a pixel represents the probability that it is "road" given its color value. This probability is estimated using adaptive models of the road and off-road color distributions. A reimplementation of the SCARF algorithm developed by SAIC for use by a team in Grand Challenge '05 was able to drive a robotic HMMWV down dirt roads at speeds from 2 – 6 m/s, including one run of 4.9 km at 4 m/s. Although this algorithm required rare manual interventions (~1 / km, usually at specific spots along the road), we will implement mechanisms for the Urban Challenge to detect transient failures and recover. At this stage of our system development, this remains work in progress.

Detection and tracking of moving objects. The robot has a set of LADAR sensors covering a 360 degree field of view with a range of more than 40 meters with each scanner having an overlapping field of view with multiple other scanners. Each scanner returns a set of points that is placed into a 3D coordinate frame relative to the vehicle. At a given iteration, this large set of data is then segmented based on spatial relationships into objects. Over multiple iterations, these objects are tracked such that a unique ID number is assigned to each entity in the environment. In combination with the data from the three radar sensors, the position and velocity (translational and rotational) of each object can be estimated with an extended Kalman filter.

Furthermore, our approach is to apply two levels of classification to each object, based on its spatial size and velocity over time: whether it is a static or dynamic object and what object class it belongs to. Examples of object classes are *Car* and *Building*.

This information is used in a variety of ways. The static laser data is used in the lowest-level controller to provide a last line of defense against unanticipated obstacles. The objects' velocity and size information is used by the behavior-based controllers to compute predicted intersection points, to avoid obstacles and navigate in the presence of traffic. This is also used by the nested hybrid automata in triggering the transitions between modes. The object classification is used, for example, in the *Find-Queue-Position* mode of the *Handle-Intersection* higher-level mode for determining if another car arrived at the intersection prior to the robot. That is, when assigning precedence at an intersection, the robot uses the object classification to distinguish between something like a stop sign and another vehicle.

2.4.2 Planning and Control

Planning and Control tasks span a number of processes in our software architecture, due to its multilayered hybrid continuous/discrete control strategy. Figure 9 shows the structure of these processes. At the top of this hierarchical structure is the *Mission Level Mapping* block. At the beginning of a mission, a map is produced that consists of a graph structure based on the provided RNDF. As the mission progresses, this graph structure is augmented with information about the routes it represents. Experiences of traffic congestion, dangerous obstacles, and impassible lanes are noted in the graph for future reference.

The map produced by the *Mission Level Mapping* block is passed on to the *Mission Level Planning* block. This block incorporates the MDF and plans a route through the graph-based map to achieve the specified checkpoints. Information stored in the map is used to weight edges of the graph, allowing the planner to find a route that optimizes the expected time-to-complete, rather than simply distance. The plan is passed on to the *Reactive Behaviors* block. A representation of the robot's current task (e.g., PARK, UNPARK, DRIVE TO CHECKPOINT) is passed on to the *Situational Awareness* block.

The *Situational Awareness* block implements a nested hybrid automaton (NHA), which is driven by the robot's current task and perception. The NHA implements an *a priori* representation of the structure of the robot's environment and task. The nested structure allows for asynchronous transitions at different levels of functionality. Each state in the NHA maps, in a one-to-many fashion, to actions such as FOLLOW-LANE, DRIVE-TO-POINT, and STAND-STILL. Selected actions are passed on to the *Behavior Arbitration* block.

The *Behavior Arbitration* block maps an action to a set of weights (which may be zero) which is applied to the output of the behaviors provided by the *Reactive Behavior* block. Each behavior provides a set of votes over discrete values of curvature within the vehicle's drive capabilities, and provides a maximum allowable velocity for each evaluated curvature. The *Behavior Arbitration* block chooses a commanded steering angle according to the input provided by the behaviors and their respective weights, and a commanded velocity according to the minimum allowable velocity provided by the behaviors for the selected curvature. This commanded curvature and velocity is passed on to the *Vehicle Control* block, which runs in a tight loop, controlling the actuation of the vehicle to achieve the commanded set points.

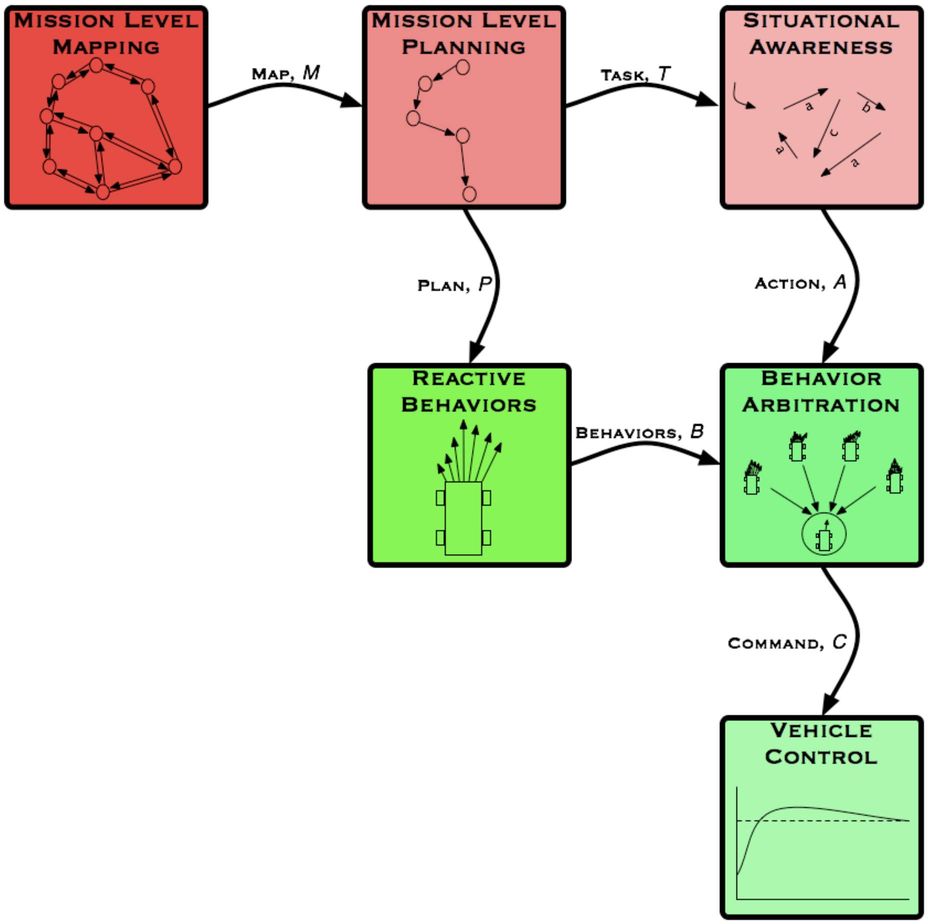

Figure 9: A detailed view of the planning and control architecture, presented as part of the full architecture in Figure 1. Arrows indicating information flow are labeled with the type of information communicated.

2.4.3 Handling Atypical and Unexpected Situations

Within the Team Sting planning and control architecture, atypical and unexpected events and situations are addressed in two different ways. First, the transitions between states at a given level of the nested hybrid automaton are asynchronous with respect to the state/transitions of lower levels. This reduces the possibility for deadlock. Moreover, by using the hybrid automaton structure, existing and well known tools for analyzing the design (e.g., assessing the reachability of bad states, finding the possibility of deadlock) are readily available. By dividing the complexity of the larger *Situational Awareness* problem into separable components – the various high-level modes described below – the standard software principles of modularity and encapsulation are employed. This planning architecture thus lends itself to quickly determining the fault in the existing design as well as allowing for a revision of that component with minimal impact on other components.

The second major way for handling unexpected situations comes from the use of a behavior-based arbitration mechanism based on the DAMN architecture [7], as shown at the Arbitration Level in Figure 9. A number of active behaviors express appropriate commands for their respective interests (such as avoiding obstacles or following the lane) by voting for or against values in a set of steering angles. Because each behavior can express multiple preferences across the set of steering angles, the behavior arbiter is less likely to arrive at a local minima or an oscillatory state. For example, a behavior dedicated to avoiding obstacles can express that turning *either* left *or* right is appropriate for avoiding an obstacle in front of the vehicle, and let the arbiter evaluate the other behaviors before deciding to turn left or right, as shown in [10].

3 Results and Performance

So far, Team Sting has operated in a number of different scenarios on four different test tracks, as well as in a simulated environment. The GPS/IMU-based waypoint tracking, lane tracking and following, static obstacle avoidance, intersection negotiation, dynamic lane changing, and headway maintenance have all been performed at various speeds and degrees of environmental complexity, as summarized in Table 5.

Date	Test Description	Comments
09/27/06	Passed integration of Planning, Control, and Simulation processes test.	
11/01/06	Check-out of digital and analog vehicle control.	
11/08/06	Calibration of gas and brake control.	Passed analog control unit tests.
12/15/06	GPS/IMU integrated through comms with Planning/Control on the vehicle.	Passed comms and GPS unit tests.
12/15/06	Passed GPS waypoint following test.	Achieved waypoints within tolerance of 1.0m and within a corridor of 4.5m.
01/24/07	Cameras and Lane Tracking integrated on the vehicle and passed unit test.	
01/24/07	Passed LIDAR/Radar integration (with comms) test on the vehicle and in simulation.	Accurately detected obstacles at >40 m (SICK) and >130 m (Riegl) from the robot.
02/09/07	Passed obstacle avoidance test in simulation.	Avoided collisions with obstacles 100% of simulation time.
02/18/07	Passed obstacle avoidance test on the vehicle.	Maintained 1 meter clearance on all sides of vehicle in obstacle field at all times.
02/24/07	Passed lane tracking speed tests at GPSTC at 5, 10, 15 mph. 5+ laps for each trial.	Speed controller maintained commanded setpoint +/- 2mph on grades of up to 6%.
02/24/07	Passed intersection logic tests at GPSTC.	Properly obeyed all intersection precedence rules.
02/25/07	Passed remote e-stop, e-pause, e-unpause tests at GPSTC.	Vehicle smoothly stopped within proper distance and time.
03/09/07	Passed speed control tests for turning and tight curves at GPSTC.	Vehicle did not exceed 0.3 rad/sec during turns. Max speed set to 20 mph.

03/09/07	Passed overtake-stopped-vehicle test at GPSTC.	Nominal speed set to 8 mph. Obstacle vehicle was the backup Porsche Cayenne.
03/09/07	Passed automata test using modes *Follow-Lanes, Handle-Intersection*, and *Overtake*.	Nominal speed set to 8 mph. Test attempted on GPSTC urban course.
03/29/07	DARPA Video demo completed.	
04/13/07	Passed lane following (visual sensing only) test on site visit course.	Minimum lane curvature on course is 7 m. Lane tracking frame rate increased to 30 fps.
04/20/07	Passed stop-at-intersection-stop-line test using combined GPS and vision sensing.	Stopped within 1 m of stop line marking for all trials.
05/28/07	Passed U-Turn test along with *Follow-Lanes*, *Overtake*, and *Handle-Intersection*.	Stayed within 9mx30m boundary during all U-Turn maneuver trials on site visit course
05/30/07	Tested headway maintenance with moving traffic. Nominal speed setpoint was 10 mph.	Robot maintained proper vehicle separation at all times.

Table 5: Summary of test activities and measured performance to date.

3.1 Testing methodology

Due to the complexity and integrated nature of the system, it is vitally important that a testing strategy is devised that allows the designers to test different aspects of the system, the validity of design modifications and additions, as well as the entire, integrated system. In order to accommodate these requirements, Team Sting's testing strategy is based on a combination of carefully engineered unit tests, integrated mission and scenario-level tests, open-loop tests in which no autonomous control of the vehicle is allowed, and simulated tests in synthetic environments.

Unit testing: Unit tests are tests designed to capture a targeted, isolated part of the system. Such tests have been conducted extensively at the early stages of development by Team Sting and they are important for capturing the basic behavior of the system from both sensing, actuation, and planning points-of-view.

Integrated system testing: One aspect of the Urban Challenge that sets it apart from previous Grand Challenges is the fact the system is forced to switch between many different modes of operation in response to environmental conditions. The high-level modes of operation (Follow Lanes, Overtake Static Obstacle, U-Turn, Handle Intersection, Park, and Unpark) identified by Team Sting as critical to a successful completion of the race are discussed in Section 2.4. These high-level modes of operation must be tested in an integrated fashion, i.e., with all low-level functionality engaged, and all transitions enabled. That is, unit tests are used to test individual perceptual and behavioral components while integrated tests are those that test the situational awareness modes that depend on these lower-level components. The hierarchical layering of the software system lends itself to translation into testing strategies at different levels of abstraction and integration.

Open loop testing in real urban environments: As safety is a key issue that must be addressed when testing the system, Team Sting is conducting Open Loop Tests, in which the vehicle is deployed in an actual, urban environment with the software system running. The only difference is that the proposed control signals are not allowed to actually control the vehicle. Instead the vehicle is controlled by a human driver. This mode of operation has proven to be very useful for evaluating the perception modules in truly complex environments. Moreover, rough qualitative estimates of the validity of the proposed control signals have been obtained in this manner. In the future, Team Sting will continue to employ this strategy in combination with a formal assessment of the proposed control signals as compared to that of the behavior of a human driver.

Modeling and simulation using OneSAF and AI.implant™ Software: Logistical and safety considerations make it impractical to test the Sting system in scenarios involving many vehicles on a large road network in real urban environments. In order to support such larger-scale tests of the system's decision logic and behaviors, the Sting software architecture is interfaced to the capabilities provided by the OneSAF constructive simulation software and the AI Implant™ commercial software. Use of physics-driven entity models and high-fidelity terrain and sensor models represented within the OneSAF simulation environment allows the actual vehicle control software to be stimulated by the simulated sensor outputs of a model running in a virtual environment to validate the system design. The system allows users to create and control actors with composable behaviors. This will allow the team to perform many more iterations on the actual vehicle control code, and to inject a high degree of complexity into the scenario without incurring the costs of providing those stimuli in a real world vehicle exercise.

3.2 Testing facilities

Our primary facilities for closed-loop testing of the Sting-1 system are the Georgia Public Safety Training Center (GPSTC) and a large parking lot located adjacent to the Georgia Tech campus. The GPSTC (http://gpstc.georgia.gov/02/gpstc/home/0,2466,31062192,00.html) has made their facilities available to Team Sting on weekends. Located in Forsyth, GA, roughly an hour south of Atlanta, the GPSTC has two training areas highly suitable for Urban Challenge testing. The first is a 1.5 mile test track for driver training, which includes a variety of turns and small hills. The second is a small grid of urban streets, complete with lane markings, intersections, and traffic signals. Figure 10 below shows these facilities.

Figure 10: Test track and urban street network at the GPSTC.

The parking lot site adjacent to campus offers a convenient location for unit tests, filming of the video demo, and the site visit. Figure 11 below shows the site painted with the Sting site visit course.

Figure 11: Parking lot test site adjacent to Georgia Tech campus, painted with the Sting site visit course.

4 References

[1] Multiple authors, "Special Issue on the DARPA Grand Challenge 2005 (Part 1)," *Journal of Field Robotics 23(8)*, 2006.

[2] Multiple authors, "Special Issue on the DARPA Grand Challenge 2005 (Part 2)," *Journal of Field Robotics 23(9)*, 2006.

[3] Jill Crisman and Charles Thorpe, "SCARF: A Color Vision System that Tracks Roads and Intersections," *IEEE Transactions on Robotics and Automation* 9(1):49-58, February 1993.

[4] Karl Kluge, Chris Kreucher, and Sridhar Lakshmanan, "Tracking Lane and Pavement Edges Using Deformable Templates," in *Enhanced and Synthetic Vision 1998 (Proc. SPIE vol. 3364)*, p. 167-176, 1998.

[5] Karl Kluge and Charles Thorpe, "The YARF System for Vision-Based Road Following," *Mathematical and Computer Modelling* 22(4-7):213-233, August - October 1995.

[6] Ola Ramstrom and Henrik Christensen, "A Method for Following Unmarked Roads," in *Proceedings of the IEEE Intelligent Vehicles Symposium, 2005*, p. 650-655, June 2005.

[7] Julio K. Rosenblatt, "DAMN: a distributed architecture for mobile navigation" in *Journal of Experimental & Theoretical Artificial Intelligence, 9:2*, p. 339-360, 1997.

[8] Ernest Dickmanns, "Vehicles capable of dynamic vision: a new breed of technical beings", in *Artificial Intelligence*, 103 (1-2), pp. 46-73, Aug. 1998

[9] Ronald C. Arkin, *Behaviour Based Robotics*, MIT Press, Cambridge, MA. 1998.

[10] J. Sun, T. Mehta, D. Wooden, M. Powers, J. Regh, T. Balch, and M. Egerstedt. Learning from Examples in Unstructured, Outdoor Environments. *Journal of Field Robotics,* Vol 23, No. 11/12, pp. 1019-1036, Nov/Dec. 2006.

[11] S. Kennedy, J. Hamilton, and H. Martell, "Architecture and System Performance of SPAN - NovAtel's GPS/INS Solution," in *IEEE/ION Position, Location, and Navigation Symposium 2006,* April 25-27, 2006.