DARPA Urban Challenge: Team MIT

Development Plan

October 27^{th}, 2006

John Leonard (MIT) **Jonathan How (MIT)** **Seth Teller (MIT)**

David Barrett (Olin) **Chris Sanders (Draper)**

Contents

Abbreviations

CPU Central Processing Unit

DGC DARPA Grand Challenge

EKF Extended Kalman Filter

FOV Field of View

GPS Global Positioning System

IMU Inertial Measurement Unit

LIDAR Light Detection and Ranging

MDF Mission Data File

RNDF Route Network Definition File

SLAM Simultaneous Localization and Mapping

VCU Vehicle Control Unit

Chapter 1

Introduction

This document is organized in five sections: Technical Strategy; Management Plan; Schedule and Milestones; Safety and Testing Plan; Budget Plan.

Our technical strategy has been formulated to address the three types of uncertainty inherent in the autonomous urban driving task: in the input, i.e., the relationship of the provided environment and mission descriptions to the actual driving environment; in sensing, i.e., the relationship of available sensor data to the actual static and dynamic surroundings of the vehicle; and in actuation, i.e., the relationship between commanded vehicle motions and the vehicle's actual physical progress. In the absence of any uncertainty, meeting the challenge would be a straightforward engineering exercise, albeit a very complex one. Yet such uncertainty is unavoidable in reality; recognizing this fact, and developing strategies to account for uncertainty, are (we believe) the keys to a successful Urban Challenge effort. Like other teams, we advocate the use of a nimble vehicle, using LIDAR and vision sensing in concert with GPS and IMU to achieve robust localization, dynamics, and control. In contrast to other teams, our team's central, and distinctive, focus is addressing the above sources of uncertainty in a way that is both scalable to spatially extended environments, and efficient enough for real-time on-board operation in a dynamic world.

Our management structure consists of Leonard (MIT ME) as team leader, How (MIT AA/AA) as technical lead for Planning and Control, Teller (MIT EECS) as technical lead for Perception, Barrett (Olin) as technical lead for Vehicle and Safety, and Sanders (Draper) as systems engineering lead. The core management team is rounded out by Jones (Draper) as Integration and Test coordinator and Olson and Moore (MIT) as lead graduate students on the project. Our teams consists of world leaders in the academic subdisciplines of Robotics, Artificial Intelligence, and Control, augmented by world-class systems engineering, integration and test support from Draper Laboratory.

Our schedule is aggressive, but the ambitious demands of the Urban Challenge could not be accomplished otherwise. We follow a spiral development model, with internal milestones laid out approximately once per month to create a development trajectory to lead towards successful completion of the DARPA site visit and NQE milestones.

Safety is imperative for this program. Our team has created a safety plan for testing, and has sought to embed multiple levels of safety into the system design.

Finally, our budget reflects our success in external fund-raising since the submission of our

Figure 1.1: Team MIT.

original proposal, with MIT resources of over $300,000 matched by in-kind donations of approximately $240,000 (Land Rover and Quanta Computing) and a $1,000,000 investment in internal IR&D resources from Draper Laboratory. The DARPA track A resources enable us to fund two PhD students and two postdoctoral fellows working on the program, as well as a group of undergraduates at Olin College, and to fully convert and equip several race vehicles.

Chapter 2

Technical Strategy

2.1 Architecture

Achieving an autonomous urban driving capability is a tough, multi-dimensional problem, and a key element of the difficulty is that significant *uncertainty* occurs at multiple levels: in the *input*, in *sensing*, and in *actuation*. Any successful strategy for meeting this challenge must address each of these sources of uncertainty. Moreover, it must do so in a way that is *scalable* to spatially extended environments, and efficient enough for *real-time* implementation on a rapidly moving vehicle. Our focus on managing uncertainty is reflected in our system architecture (Figure 2.1) that includes subsystems for planning, sensing and perception, navigation, control, and actuation:

- The **Mission Planner** computes the route plan that completes the mission in the minimum expected time.

- The **Situational Interpreter** provides inputs to the Mission Planner about any inferred road blockages or traffic delays and determines if changes are required in the current maneuver. This module also selects an optimal sequence of vehicle maneuvers to safely follow the route plan through the local map.

- The **Situational Planner** designs a trajectory that is consistent with the dynamics capabilities of the vehicle and avoids both static and dynamic obstacles. The input is the route produced by the mission planner and the local map given by the perceptual estimator.

- The **Perceptual State Estimator** computes local maps, by fusing all available sensor data into a local representation whose storage is managed by the **Map Fragment Database**.

- The **Vehicle Control Unit** executes the low-level control necessary to achieve the desired vehicle motions issued by the Situational Planner based on inputs from the Perceptual State Estimator.

- The **Safety Module** monitors sensor data, overriding vehicle control as necessary to avoid collisions. This module addresses safety pervasively through its interactions with vehicle hardware, firmware, and software.

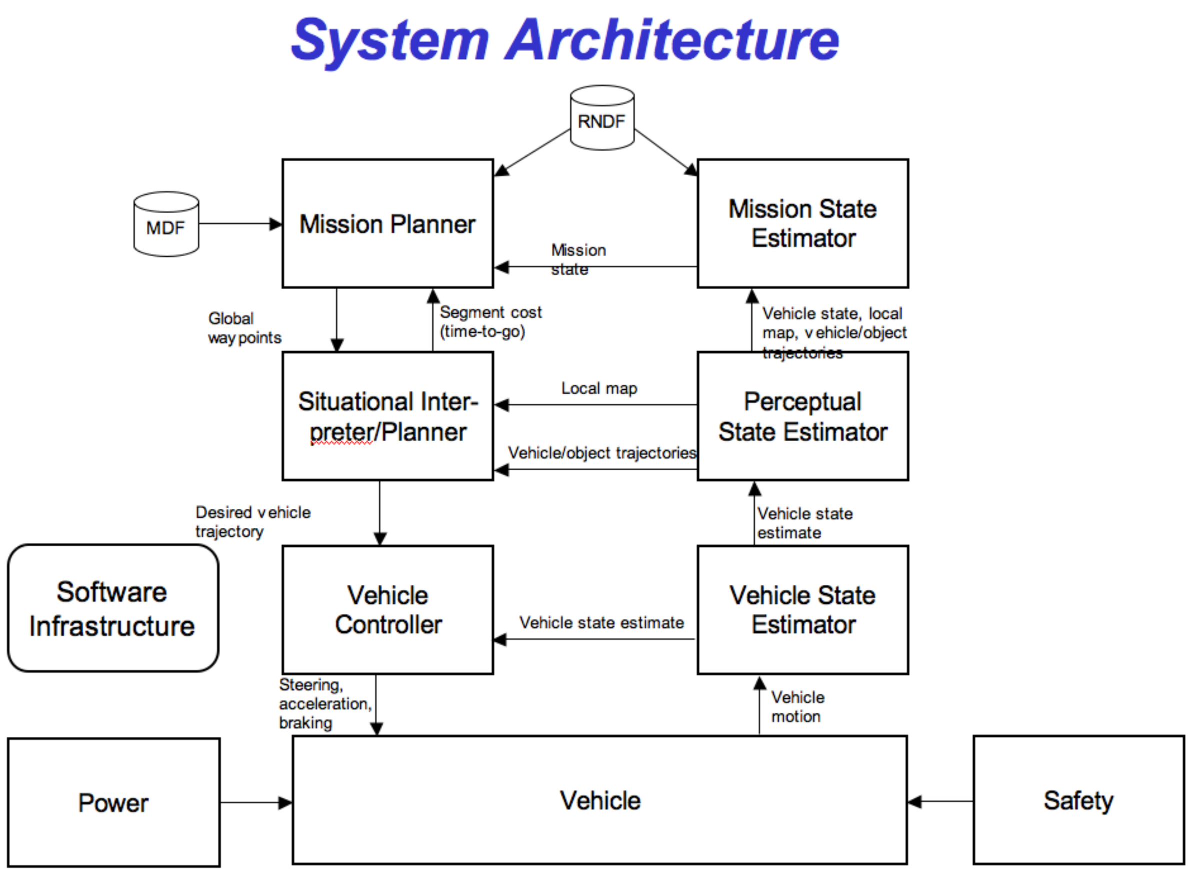

Figure 2.1: System Architecture

We have chosen to adopt a technology readiness level (TRL) framework analogous to that used by NASA and DoD. Our TRL scale can be captured approximately as follows:

- TRL 1: Sketched on a whiteboard
- TRL 3: Implemented on a benchtop or simulation
- TRL 5: Prototyped on the vehicle
- TRL 7: Fully integrated and tested on vehicle (race-ready)

XX from JH: based on this, I think we should aim for TRL 5 by 4/30/07 and TRL 7 by Aug XX

Our overall technical strategy is to (i) identify the TRL level for each major method we plan to use, (ii) identify the challenges to solving the problem and/or to using the approach at TRL 5 and higher, and (iii) formulate a rough plan to achieve TRL 7 by June 2007. For each method, we give a "strategy sentence" that outlines the plan for getting there and an indication on how and when we expect hit TRL levels 3, 5 and 7.

2.2 Sensing and Perception

LIDAR sensors and Processing The LIDAR system comprises two complementary sensor geometries: "push-broom" sensors for evaluating the navigability of nearby terrain, and a 360-degree obstacle detection and tracking system.

The "push broom" LIDAR maps the surrounding terrain as the vehicle moves. One significant innovation of the perception module is that it does not build an explicit elevation map of the terrain: these maps can generate hallucinated obstacles when the vehicle's state estimate is noisy. Instead, our implementation produces a grid-based hazard map using range discontinuities of individual scans (TRL 5). This system is scheduled to reach TRL 6 for MIT's internal milestone (12/31/06).

The obstacle detection and tracking system comprises a ring of LIDAR sensors affording 360 degree coverage. Off-groundplane objects within range, whether stationary or moving, are tracked. This produces a trajectory from which the tracking system can extrapolate future locations and the uncertainty associated with those predictions. Object detections and trajectories are passed to the situational interpreter and planner. Static obstacle detection at close range is mature (TRL 5); estimating trajectories of moving objects is substantially more difficult (TRL 3). Static detection will be at TRL 7 for our internal milestone 12/31/06, and moving obstacle tracking will be demonstrated at TRL 5 by 1/31/07. These systems are being aggressively investigated so that we can rapidly determine whether our current sensor suite is adequate, or whether we need to augment our payload with an additional sensor.

Vision Sensors and Processing The vision system detects static and moving obstacles, finds road surfaces, and locates and classifies road markings. The current configuration uses six color cameras that provide obstacle and road-marking detection. The narrow-field-of-view camera in front improves our sensing ability at long ranges, acting as a "fovea" that allows safe high-speed forward travel. When the vehicle is stopped at an intersection, the side-facing cameras detect vehicles moving along cross-streets.

Detection of road markings such as lane markers and stop lines is a major new technical requirement of the Urban Challenge. This is handled by our vision system using a matched filter that is tuned to detect fragments of lane markings, which are then clustered subject to continuity and linearity constraints. This approach handles uncertainty better than many alternatives draws upon our team's expertise in data clustering [6]. Our current implementation (TRL 3) shows the feasibility of the matched filter approach but does not yet implement any spatial clustering. We expect to be at TRL 5 by 12/31/06 and TRL 7 by 4/30/07.

Detecting far-field obstacles is another goal of the vision system, but because these objects may have unusual shapes and sizes, we cannot rely on *a priori* models of their appearance. Instead, we use a lower-level approach based on optical flow, which has recently been extended at MIT to improve robustness to the uncertainties common to natural images [7, 4]. These algorithms have been demonstrated in a laboratory environment but are not yet suited to run on the vehicle because of their slow runtime. We consider the current implementation to be at TRL 3 and expect them to be at TRL 5 by 4/30/07.

A final goal of the vision system is to detect safe portions of the road surface looking for its distinctive texture and then seeking the boundary of the road where the texture ends. We have implemented a TRL 5 version of this algorithm based on color and we intend to extend it to texture by 4/30/07 and be at TRL 6.

Perceptual State Estimation and Mapping The perceptual state estimator fuses all sensor measurements to maintain two global and local frames of reference.

The local frame is vehicle-relative, consisting of a map of obstacles (and their trajectories) and a map of safe road surfaces. Sensor uncertainty is propagated through the estimator to yield probabilistic estimates of the object paths, using methods from our recent work in robotic SLAM at MIT [2, 5]. The local frame is sufficient to do most of the fine-grained trajectory planning and obstacle avoidance because the sensor measurements do not drift relative to it. Our current development efforts (including local waypoint following) have been achieved entirely in the local frame. Our implementation of the local frame dead-reckoner is at TRL 6.

The global frame is the geo-referenced latitude/longitude frame, consistent with the RNDF. The vehicle continually estimates its global position by fusing GPS, IMU, and odometry data using a conventional Extended Kalman Filter. The noisy GPS data is only used to ensure that the local map is periodically registered with the global frame and to ensure accurate checkpoint arrivals. Since this requires only intermittent GPS availability, we ensure robustness in the presence of GPS failures. This portion of the software is at TRL 2 (not implemented). We intend to be at TRL 5 by 12/31/06 with a version of the state estimator that includes a filtered representation of GPS, without fused IMU and odometry. With further analysis and experiments on the vehicle, we intend to be at TRL 7 by 4/30/07.

2.3 Planning

Vehicle control is performed at three levels: mission, situational, and low-level. Large-scale questions are handled by the mission planner. More detailed issues are handled by the situational planner. Finally, the low-level controller allows the vehicle to precisely follow the maneuvers computed by the situational planner.

Mission Planner The mission planner generates a route plan that is expected to accomplish the mission in minimum time. This plan consists of a sequence of abstract actions using probability distributions of the completion times for each route segment. The challenges here are to produce robust optimized plans in real-time, and to ensure that we can produce workable plans to ensuring that the vehicle makes progress even in the presence of environmental uncertainty [1, 9].

The mission planner nominally runs at a rate much slower than the control loop but must be able to be triggered as soon as significant changes are detected by the situational interpreter. The algorithms that will be employed for the mission planner have been used in numerous other testbeds and simulations, and have used them in Urban Challenge simulations (TRL 3). Extensions are required to evaluate on-line the edge cost in the graph and the associated uncertainties based on the current perception of the world. The parameters will be tuned through extensive simulation testing (TRL 5 by 1/31/07). The mission planner will then be implemented on the vehicle to reach TRL 7 by 4/30/07.

Situational Interpreter and Planner The situational interpreter uses the information provided by the perception state estimation to update the local and global situational awareness. This information is then used to update the segment costs provided to the mission planner. It also identifies situations that might require changes to the current maneuvers being executed. Further challenges are to robustly distinguish between static obstacles, moving vehicles, and threats. Current TRL of 1 will be raised to TRL 3 by 12/31/06, to TRL 5 by 3/30/07, and TRL 7 by 4/30/07.

The situational planner finds a kino-dynamically feasible vehicle trajectory that moves towards the RNDF waypoint selected by the mission planner, while obeying traffic rules. The planner uses a closed-loop system model of the vehicle under the action of the low-level controller to predict its future behavior, so the generated path is always (i) consistent with the vehicle capabilities, and (ii) robust with respect to environmental disturbances and uncertainties in the system. To avoid obstacles and stay in the drivable road surface, a motion planning algorithm based on Rapidly-exploring Random Trees (RRT) will be applied [3]. This RRT-like approach was successfully demonstrated in the 2005 Darpa Grand Challenge race by the UCLA team, which was co-led by Prof. Frazzoli. The RRT-like planner used by the UCLA team was based on a static environment, which was consistent with the characteristics of that race. Thus, we assess the TRL of our planner *in a static environment* as 5.

In a more constrained (e.g., urban) and dynamic environment, characterized by traffic rules and interaction with other moving vehicles, the on-line computation load of the currently-available algorithm may become excessive. We will address this issue by embedding our a priori knowledge of the environment's structure, the traffic rules, and the current selected vehicle maneuver in the sampling strategy of the planner. This will add a deterministic bias to the RRT-like planner, significantly increasing its efficiency and reliability.

Safe trajectory planning requires a prediction of future behavior of obstacles, with explicit accounting for uncertainty. Safety is accomplished within the planning algorithm by ensuring the existence of a feasible safety maneuver for various types of threats so the vehicle always has the option of interrupting execution of the nominal plan to switch to a contingency plan [8]. The situational planner also designs these safety maneuvers in real-time to remain in the high confidence region of the local map and minimize the overall collision risk. The primary task here is to identify reactive maneuvers that can be executed given the perceived threat. These maneuvers can be designed off-line, but the selection of which to execute will depend on the current vehicle state and the situational awareness.

The development schedule is such that the planner will reach a TRL of 6 for static environments by 12/31/06, a TRL of 7 for static environments and TRL of 3 for dynamic environments by 1/31/07, and finally it will reach full operational capability (TRL of 7) by 4/30/07.

Low-level Control The low-level controller has inputs of the vehicle's state estimate (attitude quaternion and position information), and generates the steering, throttle, and brake commands to follow the trajectory specified by the situational planner. Though autonomous vehicle control is a mature field, the new urban challenge requires much tighter performance envelope then previous work has demonstrated. Specifically, an urban driving environment requires the vehicle be able to execute three-point turns, park in a parking lot, follow a path in reverse, come to a stop at a specified location, smoothly accelerate from a stop, as well as execute lane change and passing maneuvers.

Our approach to the steering control problem is to develop a model of the system performance which facilitates the implementation of a variant of the pure pursuit algorithm. For coordinating brake and throttle, we will develop a hybrid controller that allows the system to switch between either a linear proportional-integral controller for the throttle or a feedforward-integral controller for the brake. The key step here is to ensure fast and relatively smooth transitions from throttle on/off and braking to reduce mechanical wear, ease the comfort of any technicians, and ensure that the vehicle follows the velocity profile with sufficient accuracy.

Additionally, the controller must have an embedded safety system that continuously ensures the vehicle is not entering a sliding or rollover situation. This will be achieved by using a simplified car model to predict the vehicle's performance before the implementation of any control signal.

This work is currently at a TRL of 3. Though there is a rich body of research to draw from in this field, many different approaches will be evaluated to ensure that the vehicle can meet the performance requirements. Given the importance of the low-level controller to all other tests, by 12/31/06 this work will be at a TRL of 5 and by 1/31/07 this algorithm will be at a TRL of 7.

2.4 Software

Software reliability is of paramount importance. We continually perform regression testing on our software suite, on both simulated and real data to guarantee that bugs are detected early. Each major software build undergoes multiple levels of testing, ranging from fast and simple to complex and thorough. This ensures that the most basic errors are quickly caught (for example, before any code is actually deployed) without compromising our ability to detect the most subtle errors, which are not always found by perfunctory checks.

We also have procedures in place for code review; the goal to ensure that every line of code is examined by several team members to check for reliability and robustness.

Software modules are made as independent as possible. Each module is an independent process, and different modules may run on different CPUs. Most modules communicate via UDP multicast over Ethernet, which is fast, reliable, and scales well. Modules that have stricter packet delivery guarantees establish individual TCP connections. If a module should fail (due to hardware or software problems), other modules will be largely unaffected.

A well defined message-passing protocol allows certain modules to serve as drop-in replacements for other modules without requiring configuration or code changes in any other module. For example, we are able to test different mission planning techniques by simply swapping out the mission planner, while leaving the other modules untouched. Similarly, we are able to test in simulation by replacing only the perception and actuation modules.

Separating modules into individual processes also allows us to choose the best programming language for a particular task. Latency-sensitive modules where speed is of the essence, such as the low-level controller, are written in C. More complex modules that do not have stringent timing demands, such as the mission planner, are written in Java. By taking this approach, we gain the benefits of a high-level byte-compiled language with automatic memory management and an extensive standard library of data structures and algorithms, without sacrificing low-level control of modules where it is needed.

A unified testing framework is able to test all components at once, and a and a framework for autogenerating parameter marshalling and transmission code from message templates has virtually eliminated all bugs from the message-passing system. A highly configurable process manager is able to quickly load and run an arbitrary arrangement of software modules, depending on the mission at hand. The process manager also serves as a monitor, and is able to detect when modules are behaving erratically (for example, by not transmitting messages at an expected rate) or have failed entirely.

Chapter 3

Management Plan

This section provides an overview of our team composition and management plan.

3.1 Team Composition

We have assembled a diverse team spanning four academic departments and several laboratories at MIT and Olin College. Our team leader is Prof. John Leonard (Mechanical Engineering Department), who will oversee the development and implementation of navigation and mapping algorithms. He is joined by three Co-PI's: Prof. Jonathan How (Aeronautics/Astronautics Department), focusing on planning, guidance, and control; Prof. Seth Teller (Electrical Engineering and Computer Science Department), focusing on sensing and perception; and Prof. David Barrett (Olin College), focusing on vehicle mechanical, electrical, and safety systems.

Charles Stark Draper Laboratories has assigned a group of personnel full-time to our team for the duration of the Urban Challenge effort. Leading the effort from Draper is Chris Sanders, a core team member who leads the scheduling and project management aspect of our overall team.

Two experienced EECS graduate students, Ed Olson and David Moore, are Technical Student Leads; they complete our "core team" of seven people (with Leonard, How, Teller, Barrett, and Sanders).

Additional team members holding faculty or research staff positions include Prof. Trevor Darrell (vision), Prof. Emilio Frazzoli (planning and control), Prof. Bill Freeman (vision), Prof. Berthold Horn (vision), Dr. Karl Iagnemma (hazard avoidance and control), Prof. Nicholas Roy (planning under uncertainty), Prof. Daniela Rus (motion), and Prof. Russ Tedrake (learning and control). Two additional Draper staff are part of the team as well: Dane Richter, focusing on vehicle state estimation, and Troy Jones, focusing on integration and testing.

Several graduate students from EECS, MechE, and Aero/Astro, and a number of undergraduate students from MIT, Olin College, and Wellesley College round out the team.

Each team member is part of one or more sub-teams focusing on safety, integration/testing, navigation, perception, planning, and control. Figure 3.1 shows the mapping of lead team members to each component of the system architecture. This chart plays a vital role in our management approach, providing the basis for defining interfaces, accountability, and testing (TRL level

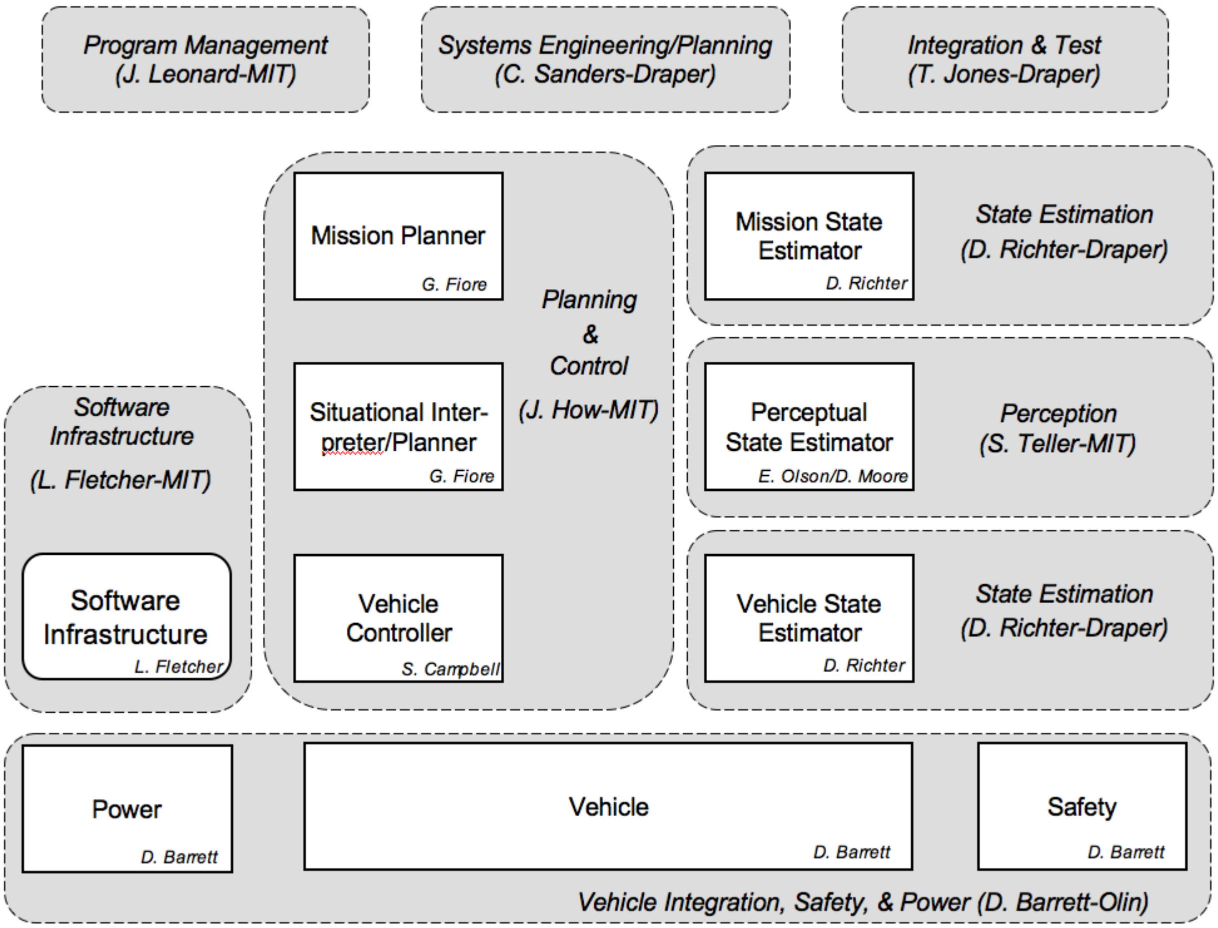

Figure 3.1: Work breakdown schedule, showing the assignment of each element of the system architecture to its lead member of the team.

achievement).

3.2 Management Structure

Team MIT's management strategy has six major facets:

1. A capability ladder with integration and testing milestones;
2. Core team goals meetings;
3. All-hands meetings run by the core team;
4. Sub-team meetings run by sub-team leads;
5. Bullpen development for cross-subteam work; and

6. Safety, integration and testing efforts.

Our capability ladder and integration and test milestones are the major drivers of our activities. The capability ladder is a collection of “rungs,” each representing a vehicle capability of increasing sophistication. The rungs are sequenced so as to span all Urban Challenge milestone and competition capabilities. We perform frequent, usually daily, integration and testing to assess our progress on the ladder.

Core team goals meetings involve only the seven members of the core team. At these meetings we come to consensus about what must be accomplished during the next one to two weeks, while simultaneously tracking our progress with respect to our long-term goals and milestones. Each goals meeting ends with a clear statement of what we have decided, along with concrete responsibilities for each core team member and associated sub-team(s).

All-hands meetings occur roughly every two or three weeks, and include every available team member. These meetings are run by a core team member, according to focus (scheduling, technical area, safety, etc.). They often consist of round-the-table status updates, bottleneck identification, and individual acknowledgements of accountability in front of the group. We also use these meetings for morale-building and for discussion of issues such as logo design.

Sub-team meetings occur much more frequently, at least weekly, and are run by the core team member leading the sub-team in question. These meetings serve to keep sub-team members focused on goals derived from the core team goals meetings.

Bullpen development occurs around the clock in two locations, at MIT CSAIL and at MIT Aero/Astro, while the race vehicles are in the CSAIL parking garage. This is not satisfactory. Another major contribution from Draper Laboratories is heated high-bay space, available in late October 2006, which will enable us to merge our two bullpen areas with live storage for the vehicles themselves, along with exhaust venting, work tables and chairs, power, network, and an indoor GPS repeater.

Safety, integration, and testing form the sixth component of our management structure. One core team member, David Barrett, oversees the safety aspects of the vehicle’s mechanical and power systems. Our team wiki includes a safety plan, consisting of a description of the varying levels of autonomy (from none to full) available for vehicle operation, the options available to human operators under a variety of failure modes (from non-dangerous to dangerous), and the operating conditions that must be met in order to run a vehicle test at each autonomy level. The safety plan is summarized briefly elsewhere in this document.

Chapter 4

Schedule and Milestones

The project development schedule is defined ultimately by the qualification and finally the Urban Challenge contest. However, Team MIT can only reach this goal by dividing the project into a set of manageable tasks and milestones that will culminate in a competition worthy autonomous vehicle. The following sections give a brief overview of the current team organization and snapshot of the development schedule. As with any project of this nature, the schedule and planning will evolve during the year and the challenge for the management team will be keeping focus on the ultimate goals.

4.1 Spiral Development Concepts

The development model for the project is loosely based on the Spiral concept of design and build (See Figure 4.1). Currently, the team is building a complete Spiral 1 Vehicle based on a Ford Escape which was purchased early in the program. This car will serve as an active development testbed for the team and lessons learned will be applied to the Spiral 2 vehicle - the Land Rover LR3. The Escape will continue throughout the project as the proving ground for technologies to be integrated into the LR3. The Spiral 2 vehicle effort should begin in January when the LR3 returns from conversion at EMC.

4.2 Integrated Development & Testing Schedule

The focus of this project is development of a system with a broad range of performance requirements. The team has defined Internal Milestones to achieve incremental levels of capability, leading up to each DARPA milestone. The milestone schedule is as follows:

- Milestone A (12/18/2006): GPS based waypoint navigating car with basic LIDAR obstacle avoidance.
- Milestone B (1/20/2007): GPS based waypoint navigating with LIDAR obstacle avoidance and Vision based road/obstacle identification.

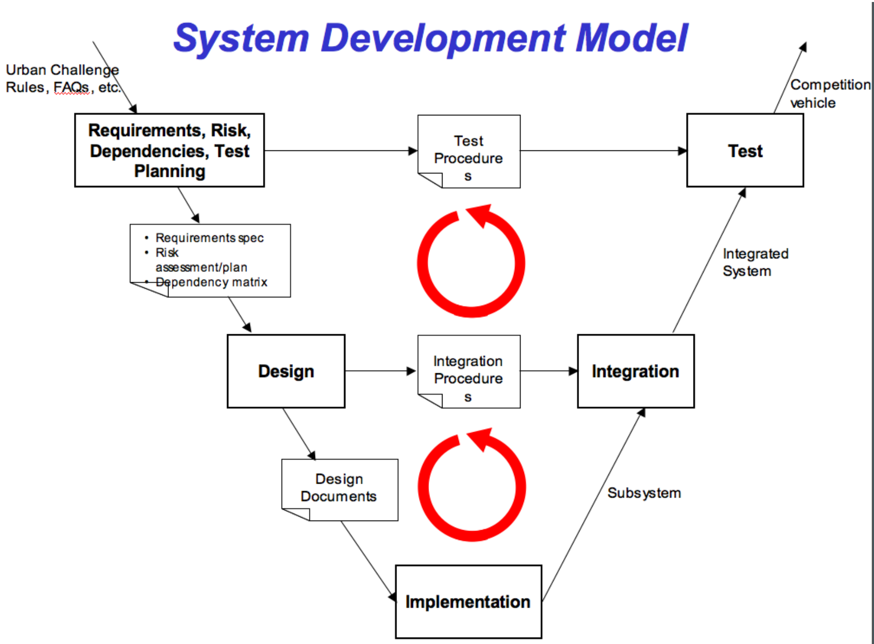

Figure 4.1: Spiral design philosophy.

- Milestone C (2/24/2007): Integrated road relative and GPS navigating car with fused LIDAR/Vision and moving obstacle avoidance and tracking.
- Milestone D (3/24/2007): Robustly navigating and planning in a representative Urban Environment (some GPS outage) with stationary and moving obstacles.
- Milestone E (4/21/2007): Add remotely driven traffic vehicle to opposing lane, perform intersection crossing, endurance for cumulative behaviors.
- Milestone 2 (6/2007): Site visit demonstration of Basic Traffic, Basic Navigation on Urban Course.
- Milestone F (7/21/2007): Expand endurance, add remote intersection traffic vehicle and merging.

  Milestone G (8/25/2007): Verify ALL required behaviors, perform representative full mission.

4.2.1 Subsystem Development

The program schedule reflects the current breakdown of the Subsystem teams. Each subsystem is assigned a project defined TRL level which increases over time. Also note that each team is responsible for generating Interface Control Documents (ICD's) which are needed to reduce errors when interfacing the software and hardware for each team. In general, the teams are slated to reach a new TRL level every 1-2 months - a rapid cycle is required given the compressed schedule.

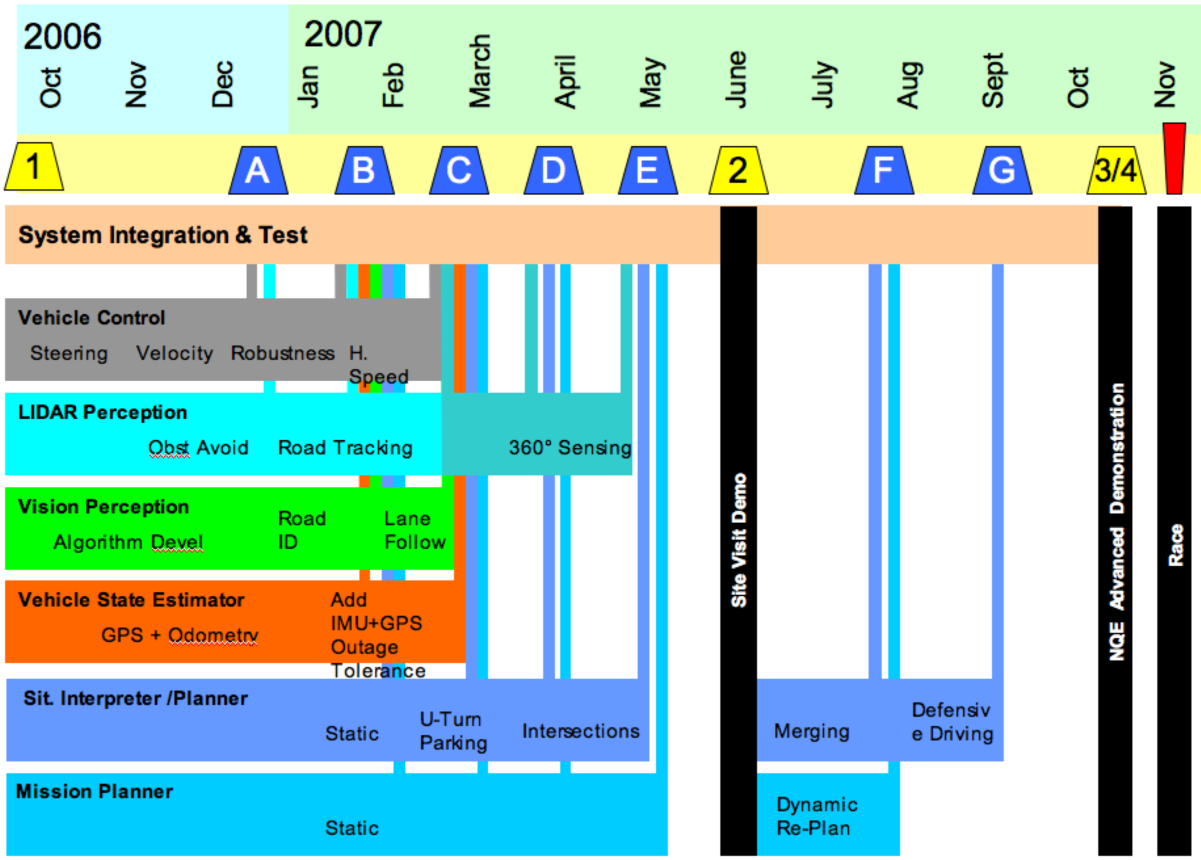

Figure 4.2: Schedule summary.

Perception Subsystem

Development of the Perception system began earliest in the project and must achieve a high level of robustness by February 2007. Other subsystem teams, such as Planning and Control use Perception inputs to make decisions about the vehicle's path and speed. Perception is divided into two parts: LIDAR based and Vision based. The implementation of basic LIDAR obstacle avoidance will happen quickly and provide valuable inputs to the design of the LIDAR system. Whereas the Vision system is working towards overall algorithm design and performance metrics. In February the Perception system will be generating the first fused data sets for testing.

Vehicle Control Subsystem

The earliest subsystem to reach maturity must be the low level Vehicle Control. A robust and fault tolerant control is critical to operating the vehicle during closed-loop (autonomous) operations. Therefore, priority is given to tests which benefit the Vehicle Controller development. Since the Vehicle Controller comes first, so must its corresponding documentation to define the interface between it and the Planning system.

Situational Interpreter & Planning Subsystem

Development of the Situational Interpreter & Planner (SIP) will lag the other subsystems simply due to the overall complexity of its job. The SIP is responsible for assimilating the available sensor data, checking for route requirements from Mission Planner and making most decisions about vehicle driving (such as intersection precedence behavior and passing decisions. Building up considerable experience, the SIP team will first implement a ”Static” SIP which will make decisions based on a static external environment. Development of the ”Dynamic” SIP will continue in earnest until the end of March and then it will be up to testing to discover flaws that were not caught in simulation.

Mission Planning Subsystem

The mission planner (MP) is in charge of the overall route and conveys this information to the SIP. Mission planning for a fixed course will be tested in January when planning the simple missions around the ”Range 1” test track. The teams plans on a fully functional Mission Planner by the end of March ready for intensive testing.

Vehicle State Estimator

Wheel odometry, GPS and a coupled IMU system will provide and estimate of the vehicle position in the route. This capability will be used in December to test GPS based waypoint following for Milestone A. To reach maturity in January, the estimator must appropriately compensate for expected errors, like GPS multipath, and handle complete loss of GPS without causing erratic maneuvers.

4.2.2 Test Planning

The remaining two months before the DARPA Milestone 2 Site Visit are slated for intensive testing on a representative Urban Test Range - which is currently under investigation. Each of the Internal (and DARPA) Milestones currently have basic test plans and are in refinement. By developing our test plans early, the team hopes to understand basic needs - such as test facility requirements - early and mitigate the risks of a poorly controlled test environment. This early test planning phase also allows us to identify the safety requirements and procedures for each test - as indicated by the Safety Levels defined in the Safety and Testing Plan.

Range 0 Testing

The very first tests performed were targeted to acquiring ”training” data sets for the software developers and in the schedule are labeled as ”Range 0” tests - meaning tests performed in a non-surveyed environment. Range 0 tests also include operations such as acquiring vehicle dynamics related data for various steering and speed conditions.

Range 1 Testing

The ”Range 1” test area is currently in design and will be located on a flat paved parking lot on or near the Olin College campus. This test range will include a oval track of road boundary lines which has been precisely surveyed to allow the generation of a RNDF for the course. The site will be off-limits to public traffic and suitable for many low speed (max 10 MPH) tests.

Range 2 Testing

The ideal ”Range 2” would have all the main features of the DARPA Challenge site - including intersections, varying road types, and buildings. The team is researching several potential candidate sites, such as non-used military bases in order to obtain the proper permissions as early as possible.

Chapter 5

Safety and Testing Plan

Maintaining a high level of safety is a central goal. This section describes the safety capabilities available to car operators and bystanders. It also lays out requirements for those operating the vehicle before performing any testing; these requirements (once ratified by the team PIs) must be adhered to at all times.

Figure 5.1 summarizes our design approach with respect to safety. The idea is to have multiple levels of redundancy throughout the system, from the lowest (hardware) level to the highest (software), with a nested set of fallbacks to provide robust safe operation despite failure of individual components.

The vehicle can be operated in varying degrees of autonomy and with different mission plans. The safety requirements vary accordingly. Before proceeding a test, the first step is to determine which testing category applies. This then determines the safety procedures required.

5.1 Definitions

Operator: Person who sits in the driver's seat of the vehicle. This person's primary responsibility is safety, and shall not be involved in manipulation of software or algorithms. During non-autonomous or partially autonomous modes, the operator will be manipulating gas, brake, and/or steering. During fully autonomous modes, the operator's only responsibility is safety: operating an E-stop and observing the road. In safety levels 1-4, the operator may use a hands-free headset to communicate with a range safety officer outside the car. In safety levels 5 and above, the operator should not use a headset.

Vehicle Safety Officer (VSO): Person who sits in the front passenger seat during some modes of operation. Like the operator, this person's primary responsibility is safety. The VSO is only required in safety levels 5 and above. In these safety levels, the VSO communicates with a range safety officer outside the car using a hands-free headset and also can operate the E-stop.

Range Safety Officer (RSO): Person outside the vehicle who monitors the ingress and egress points of the test area and ensures that no pedestrians or other vehicles are present inside the course. If an unsafe situation arises, the RSO notifies the operator or VSO via radio for them to stop the vehicle.

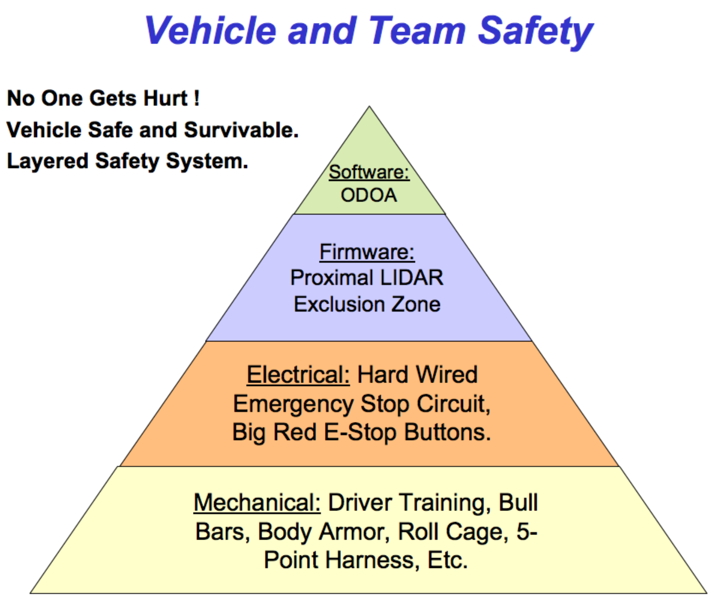

Figure 5.1: Summary of multi-layered safety pyramid.

5.2 Safety Mechanism Overview

This section describes the various safety mechanisms designed into the vehicle. We intend to test and validate each safety mechanism in isolation, and in actual operation, before relying upon it during situations in which other vehicle capabilities are tested.

Cancel button: This button is located in the cabin accessible to both front seats. It disconnects the computer from the vehicle's drive-by-wire system and applies a modest braking force. It is intended for non-emergency stopping in the event of software malfunction (i.e., when the operators detect an error but the situation is not hazardous).

E-Stop Button: The e-stop button is located at all exterior corners of the vehicle and in the cabin. The e-stop applies maximum braking force and is intended for emergencies only, as the rapid braking itself can be hazardous.

Engine Kill Button: The engine kill button is accessible to both driver and front passenger and kills the engine and the vehicle's power electronics simultaneously. The brakes are not automatically depressed since the entire system is powered down. This stop serves as a final option should both of the other stop mechanisms prove inoperable.

LIDAR Time-to-Collision (TTC): In later autonomous operation, the LIDAR system will have an independent "time to collision" estimator which will have independent authority to apply braking force. This provides additional protection should the more complex sensing+perception+planning systems malfunction.

Remote Pause and Kill (RPK): Upon receipt of the DARPA-certified wireless pause and kill device, there will be the capability to stop the vehicle remotely. Until then, the occupants of the vehicle are solely responsible for stopping the vehicle.

Vehicle Danger Zone (VDZ):

The vehicle's danger zone is the area near the vehicle in the current direction of travel. Anything within 2 second of travel time (e.g., 30 feet at 10mph, 60 feet at 20mph) is in the danger zone.

The operator must stop the vehicle if a human comes within the VDZ.

We say that an object is "collision-friendly" if the vehicle can come in contact with the object, for testing purposes, without causing damage or injury. Example collision-friendly objects include traffic cones and lightweight plastic barrels.

Many testing levels require the vehicle to stop if an object that is not collision-friendly is detected within the VDZ.

5.3 Testing Levels

This section describes testing levels 1 through 8.

5.3.1 LEVEL 1: Manned, non-autonomous operation

When relocating the vehicle under complete human control (i.e., when ferrying the vehicle from one location to another), the vehicle may be operated by a single driver. This driver must be on a list of insured drivers maintained and controlled by the PI's.

5.3.2 LEVEL 2: Low-speed operation; idle gas; steering autonomy

Description:

- The gas/steer controller is set to an idling position via a software mechanism that applies no gas.
- Speeds are limited to 10mph.
- Note that only a software restriction prevents the vehicle from asserting gas; the vehicle operator must be prepared to cancel the problem should the gas pedal ever be actuated.

Requirements:

- The test operator may not have any safety-related duties (e.g., vehicle safety or range safety). The test operator may be positioned inside or outside of the car.
- The vehicle operator is seated in the driver's seat and maintains control over gas, brake and transmission. The vehicle operator has no other duties (but we recognize that the operator should always be looking out of the window to observe the vehicle's vicinity).
- A separate range safety officer is required if the range cannot be readily secured against bystander incursions. Constant communication between RSO and vehicle operators.

- Three-point restraints are required for vehicle occupants.
- Only collision-friendly objects are permitted within the vehicle's danger zone [1].
- Vehicle may be operated only on smooth, paved roads.
- Emergency plan (i.e., ensure: that 911 connectivity exists; that location of nearest hospital is known; and that operators have the ability to report location).

5.3.3 LEVEL 3: Low-speed operation; gas autonomy; manual steering

Description:

- The mechanical steering interlock is disconnected, providing absolute operator control over steering.
- Speeds are limited to 10mph on small courses (25 meters of open "buffer space" surrounding the testing area); or 15mph on large test courses (40 meters of open "buffer space" surrounding the testing area).

Requirements:

- Same as Level 2

5.3.4 LEVEL 4: Low-speed operation; gas autonomy, steering autonomy

Description:

- Gas and steering are under software control.
- Speeds are limited to 10mph.

Requirements:

- All requirements of level 2, plus:
- The vehicle operator mans the Cancel and E-Stop buttons.

5.3.5 LEVEL 5: High-speed gas/brake testing

Description:

- Higher-speed testing of gas/brake commands.
- Gas and brake are under autonomous control.
- Vehicle operator maintains control over steering.

- Speeds limited to 30mph.

Requirements:

- Successful, stable performance at test levels 1-4 inclusive.
- All requirements of level 2, plus:
- Driver maintains control over steering (only).
- A second vehicle safety officer (VSO) is required to man the internal E-Stop (only).
- The driver and VSO must coordinate to ensure that, at all times, at least one person is observing the vicinity of the vehicle.
- A third range safety officer is required outside the vehicle. Additional safety officers are required if one officer cannot see the entire test course from one position.
- Five-point harnesses and low-speed-rated helmets worn by all vehicle occupants.
- Formal test plan and explicit PI approval are required prior to testing.

5.3.6 LEVEL 6: Low-speed close-quarters maneuvering

Description:

- Gas and steering are under software control
- Speeds are limited to 15mph
- Collision-unfriendly obstacles (e.g., K-rails) are permitted.

Requirements:

- Successful, stable performance at test levels 1-4.
- All requirements of level 5, plus:
- Automatic LIDAR "time-to-collision" safety enabled.
- Formal test plan and explicit PI approval are required prior to testing.

5.3.7 LEVEL 7: Desert Grand Challenge 2005

Description:

- Follow a course similar to the last Grand Challenge at speeds no greater than 50 mph
- Hazardous terrain
- No moving obstacles

Requirements:

- Successful, stable performance at test levels 1-6.
- All requirements of level 6, plus:
- Full crash helmets
- Chase car

5.3.8 LEVEL 8: Close-quarters maneuvering with moving obstacles

Description:

- Urban Challenge 2007 conditions

Requirements:

- All requirements of level 7, plus:
- Additional to-be-determined requirements for operators of nearby vehicles.

5.3.9 Simulation

The control algorithms will be tested in a simulation environment before being used on the vehicle. The simulation testing will help to reveal potential hazardous vehicle behavior without endangering personnel or the vehicle. Simulation will also allow us to concretely understand how the vehicle will interact with other vehicles in an environment without adverse effects from collisions. The goal is to primarily simulate our situational planner and mission planner.

Chapter 6

Budget Plan

This section summarizes the team's past and future expenditures of funds obtained from the DARPA Track A contract and from other sources.

6.1 Resources

Our team has worked hard to raise funds to help maximize its chances for success in this project. The resources obtained thus far are as follows:

Amount	Source
$125K	MIT Dean of Engineering
$60K	MIT Aero/Astro (12 month RA-ship)
$60K	MIT Mechanical Engineering (12 month RA-ship)
$25K	MIT EECS (one semester TA-ship)
$45K	MIT CSAIL
$20K + $40K	Ford-MIT Alliance (Land Rover LR3)
$200K	Quanta Computer (Blade server racks)
$1M	Draper (three staff engineers plus facilities)
$1M	DARPA Track A proposal funds
2,575K	Total (cash + personnel + equipment)

Our effort was launched with a $125,000 donation of discretionary funds from the Dean of the MIT School of Engineering. This has been matched by $45,000 from the Director of MIT CSAIL. In addition, our team has been provided with donations of salary and tuition support for support of several graduate students, including one 12 month research assistantship each from the ME and Aero/Astro departments (cash equivalent approximately $60,000 each) and one semester teaching assistantship for an urban challenge-affiliated undergraduate and graduate subject (cash equivalent approximately $25,000).

Our team has also received two valuable in-kind donations of equipment: 1) a Land Rover LR3 vehicle (cash value approximately $40,000) made available for our use by the Land Rover Brand, Ford Motor Company through the actions of the Ford-MIT Alliance, and 2) several rack-mount blade server computer systems (cash value approximately $200,000), to be provided by Quanta Computer. The Ford-MIT Alliance is also providing a $20,000 of research funding towards the cost of the LR3 fly-by-wire conversion.

Our team is committed to ongoing fund-raising efforts with a goal of raising $500,000 additional cash funds to support contingencies, to permit experimentation with alternative navigation and sensor suites, and to facilitate more extensive off-site testing in remote field locations.

6.2 Expenditure Plan

Our expenditures are organized in the following categories:

- phase 1: prior to Track A project start
- phase 2: site-visit preparation phase
- phase 3: NQE preparation phase
- phase 4: Urban Challenge competition preparation phase

In phase 1, the primary goal of our initial expenditures has been to obtain a street-legal, fly-by-wire vehicle as soon as possible, to initiate our spiral development process. Accordingly, our team spent approximately $20,000 to purchase a Ford Escape and an additional $50,000 to have it converted to fly-by-wire operation by Electronic Mobility Controls in Baton Rouge, LA. An additional $70,000 of funds is allocated for conversion of our Land Rover LR3.

In phase 2, the DARPA track A funds will be used for support of two Phd students and two postdoctoral fellows at MIT, a group of undergraduate students at Olin College, and for acquisition of various sensors, computing, and power subsystems for use in race vehicles.

Funds obtained in phases 3 and 4 will be used for continued support of PhD students and postdoctoral fellows, for acquisition and conversion of a second LR3 race vehicle, and for acquisition of a second set of sensors & power systems for our vehicles.

References

[1] ALIGHANBARI, M., BERTUCCELLI, L. F., AND HOW, J. Filter-Embedded UAV Task Assignment Algorithms For Dynamic Environments. In *Proceedings of the AIAA Guidance, Navigation and Control Conference* (Aug 2004), no. AIAA-2004-5251.

[2] BOSSE, M., NEWMAN, P., LEONARD, J., AND TELLER, S. Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *The International Journal of Robotic Research 23* (2004), 1113–1139.

[3] FRAZZOLI. *Robust Hybrid Control for Autonomous Vehicle Motion Planning*. PhD thesis, MIT, June ’01.

[4] NEGAHDARIPOUR, S., AND HORN, B. A direct method for locating the focus of expansion. *Computer Vision, Graphics, and Image Processing 46*, 3 (June 1989), 303–326.

[5] OLSON, E., LEONARD, J., AND TELLER, S. Fast iterative optimization of pose graphs with poor initial estimates.

[6] OLSON, E., WALTER, M., LEONARD, J., AND TELLER, S. Single cluster graph partitioning for robotics applications. In *Proceedings of Robotics Science and Systems* (2005), pp. 265–272.

[7] SAND, P., AND TELLER, S. Particle video: Long-range motion estimation using point trajectories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (June 2006).

[8] T. SCHOUWENAARS, M. VALENTI, E. F., AND J. P. HOW, E. R. Linear Programming and Language Processing for Human-Unmanned Aerial-Vehicle Team Mission Planning and Execution. *AIAA Journal of Guidance, Control, and Dynamics 29*, 2 (March-April 2006), 303–313.

[9] TIN, C. Robust Multi-UAV Planning in Dynamic and Uncertain Environments. Master’s thesis, MIT, Aug. ’04.