Autonomous Navigation Technologies Developed to Support the 2007 DARPA Urban Challenge[1]

Team Gator Nation - #B131

David Armstrong, Antoin Baker, Carl Crane, Doug Dankel, Greg Garcia, Nicholas Johnson, Jaesang Lee, Shannon Ridgeway, Eric Thorn, Steve Velat, and Ji Hyun Yoon

Center for Intelligent Machines and Robotics
University of Florida, Gainesville, FL

John Washburn
Smiths Aerospace, Grand Rapids, MI

submitted 1 June 2007

Executive Summary

This paper describes the system design developed by Team Gator Nation in preparation for the 2007 DARPA Urban Challenge. A hybrid Toyota Highlander (see Figure 1) has been automated and instrumented with pose estimation (GPS and inertial) and object detection (vision and ladar) sensors. The control architecture consists of four primary elements, i.e. Planning Element, Perception Element, Intelligence Element, and Control Element. The architecture is implemented on a system distributed over ten single-board computers that intercommunicate via the Joint Architecture for Unmanned Systems (JAUS) version 3.2 protocol.

The primary contribution of this work is that related to addressing the technical challenges of (a) the reconciliation of differences in estimated global pose, a priori data, and sensed information, (b) the determination of the appropriate behavior mode, and (c) the smooth transition of vehicle control between behavior modes. The processes that perform these tasks as well as the other necessary processes that perform perception, data integration, planning, and control are described in detail together with their design rationale. Finally, testing results accomplished to date are presented.

Figure 1: Team Gator Nation NaviGATOR

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

1. Introduction and Overview

1.1. Team Overview

Team Gator Nation is comprised of twenty five faculty, students, and engineers from the University of Florida Departments of Mechanical and Aerospace Engineering, Electrical and Computer Engineering, and Computer and Information Science and Engineering together with engineers from Smiths Aerospace. The personnel bring together a broad range of expertise in the area of autonomous unmanned systems. The team participated in both the 2004 and 2005 DARPA Grand Challenge events under the name Team CIMAR.

1.2. Problem to be Solved

In DARPA's vision, "The Urban Challenge features autonomous ground vehicles maneuvering in a mock city environment, executing simulated military supply missions while merging into moving traffic, navigating traffic circles, negotiating busy intersections, and avoiding obstacles." Moving the challenge into an urban setting adds structure and complexity to the Grand Challenge problem. Previous success relied on a single mode of operation, without interaction with the environment beyond simple traversal. Success in the Urban Challenge will require numerous modes of operation and complex interaction with the environment. It is expected that the urban environment will also hamper the use of GPS for localization, further complicating the challenge.

The specific problem to be solved is detailed in the Urban Challenge Technical Evaluation Criteria document [1]. Here the problem is organized into four categories, i.e. Basic Navigation, Basic Traffic, Advanced Navigation, and Advanced Traffic, each of which is more complex than the previous. Upon reviewing this document, the authors identified the following set of technical challenges:

1. pavement (road) detection and lane detection
2. detection of static obstacles
3. detection and classification of dynamic objects
4. environment data representation and sensor integration
5. localization
6. reconciliation of differences in estimated global pose, a priori data, and sensed information
7. high level mission planning
8. determination of appropriate behavior mode
9. smooth transition of vehicle control between behavior modes
10. interprocess communication and coordination of multiple threads on multiple computers
11. fault tolerance

This paper documents the design choices that have been made to address these challenges.

1.3. Prior Work and Impact of Current Effort

Much work has been done in the past twenty years to address many of the specific technical challenges that are listed in the previous section. References [2]-[7] provide excellent summaries of the advancements made by other teams competing in the 2005 DARPA Grand

Challenge. References [8] and [9] present the authors' work related to the 2005 event. Numerous references can be cited for each of the important technical challenges, but are not presented here due to space limitations.

The authors believe that the approach presented here makes new contributions primarily with respect to items 6, 8, and 9 in the preceding list of technical challenges. Traditional approaches, such as for example vision processing algorithms to identify lane markings in an image, are modified as needed and integrated into the system. However, the work related to (a) the reconciliation of differences in estimated global pose, a priori data, and sensed information, (b) the determination of the appropriate behavior mode, and (c) the smooth transition of vehicle control between behavior modes is identified as the major contribution of this effort.

1.4. Summary of Approach and Concept of Operation

The main function of the autonomous architecture is to generate control commands for the vehicle that result in expected and desired behaviors in response to its mission and the surrounding environment. The approach chosen allows for the behaviors required for the Urban Challenge to be implemented without changing the control command generation methodology. A challenge in vehicle control is maintaining consistency in the control commands thorough time. If an approach requires changing the method of command generation as different behaviors are required, much care is needed to ensure continuity in the control commands.

The overall approach is briefly summarized as follows:

(1) An off-line path planning program generates a desired motion path based on the Route Network Definition File (RNDF) and the Mission Data File (MDF).

(2) A tessellated Local World Model (300m × 300m grid with 0.5m resolution) is generated based on a priori road network data and the planned motion path. The center point of the Local World Model is located at the current location of the vehicle as determined from sensor positioning data.

(3) Data from ladar and vision sensors, which identify static obstacles, dynamic objects, smooth terrain, and road lane regions, is integrated as a layer into the Local World Model.

(4) Based on the a priori data and sensed data stored in the Local World Model, software components referred to as *Situation Assessment Specialists* focus on making specific findings (one simple example is the specialist that reports if the lane to the left, or right, is clear of other vehicles or obstacles).

(5) Six software components referred to as *Behavior Specialists* then make an assessment of whether their corresponding behavior mode is appropriate at this moment. The six behavior modes are Roadway Navigation, Open Area Navigation, Charge Lane, Reverse Direction, Intersection Traversal, and Parking.

(6) A software component referred to as the *Decision Broker* selects the behavior mode for the system based on the recommendations of the Behavior Specialists.

(7) Based on the behavior mode, a software component called the *Smart Arbiter* then generates a 60m × 60m traversability grid that is formed to elicit a specific response from the vehicle (change lanes is an example).

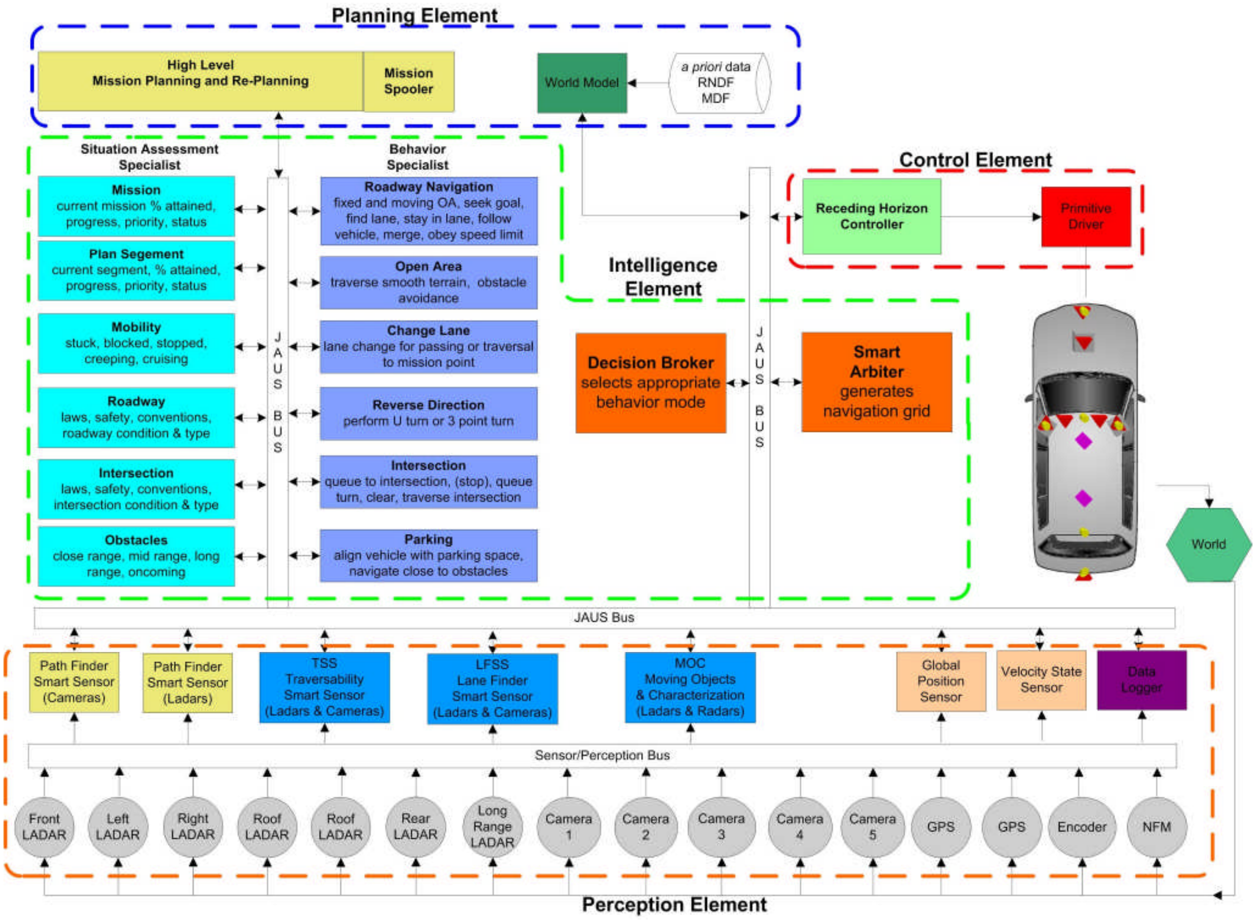


Figure 2: System Architecture

(8) Finally, the *Receding Horizon Controller* component plans a suitable path through the grid that was output by the Smart Arbiter. Steering, throttle, and braking commands are generated to execute the planned path.

The design and implementation of this control approach is detailed in Section 2 of this paper. The system components are shown in Figure 2, however the figure itself cannot convey the nature of the interactions that occur between components as was presented in the list in the previous paragraph.

1.5. System Architecture and Framework

The system architecture is a natural extension of the Joint Architecture for Unmanned Systems (JAUS) Reference Architecture, Version 3.2, which defines a set of reusable components and their interfaces. The actual core software to support the JAUS messaging system was developed and extensively tested for the previous Grand Challenge and supports the current effort with little or no modification required.

At the highest level, the architecture consists of four basic elements, which are depicted in Figure 2. The Planning Element contains the components that act as a repository for a priori data such as the RNDF and the MDF. This element will also perform the high level route planning and re-planning based on that data plus real-time information provided by the rest of the system. The Control Element contains the Primitive Driver that performs closed-loop control on vehicle actuators to keep the vehicle on a specified path. The Perception Element contains the components that perform the sensing tasks required to determine the vehicle's position, to find a road, to find the lanes on a paved road, to locate both static and dynamic obstacles, and to evaluate the smoothness of terrain. Finally, the Intelligence Element contains the components that work together to determine the best course of action to navigate the vehicle in a complex environment based on the current mission and situation.

2. Analysis and Design

The detailed system design and design rationale are presented in this section. The description is divided into ten sections, i.e. vehicle design, computer architecture, localization, high level planning, local world model, perception, adaptive planning framework, behaviors, smart arbiter, and motion execution.

2.1. Vehicle Design

A survey of COTS vehicles resulted in the consideration of hybrid type SUV's resulting in the selection of a Toyota Highlander Hybrid. This vehicle is a full hybrid, capable of running on the electric power train and/or the V6 internal combustion engine. The vehicle has a wheel base of 2.715 m, a width of 1.825 m and a height of 1.735 m. Vehicle weight as delivered is 1850 kg (4070 lbs). The Toyota Highlander Hybrid's chassis has undergone NHTSA crash testing. Toyota equips the vehicle with anti-lock braking, traction control, skid control, and electronic power steering. It was desired that the automation design not bypass these Toyota stability features.

The nature of a full hybrid requires that the throttle and brake be drive-by-wire. Automation of these controls is effected via an interface with the brake and throttle sensors to allow computer input of control signals to the vehicle control architecture. A servo-motor has been installed on the steering column to replace the human input to allow drive by wire behavior of the steering subsystem. The shift lever is automated via a servo actuator. The wiring harness has been interfaced with to allow turn signals and brake lights to function as needed. The reverse indicator lamps function as usual when the shift actuator places the vehicle in reverse. The drive by wire implementation is controlled by a touch-screen tablet located in the front seat. This computer handles the low level interface with the vehicle.

The Toyota Highlander Hybrid has an electrical system suitable for powering the automation and compute resource requirements for the DARPA Urban Challenge. The high voltage DC system includes a battery bank based on nickel-metal hydride cells. The bank has high availability, able to provide 45 kw for short durations. The nominal voltage of the bank is 288 VDC. This battery system sources a 12 VDC output converter that replaces a conventional alternator. The autonomous and compute power systems are sourced from both the high voltage DC system and the 12 VDC system.

Safety protocols are implemented via in-car and on-car e-stop switches and an off-car safety radio. The safety radio mimics the functionality of the Omnitech DGCSR(RX) safety radio. Emergency stop is affected via a mechanical system that applies braking effort.

2.2. Computer Architecture

Compute resources are designed and specified to allow flexibility and expansion as needed. Power system design for efficiency is a major concern because power availability is the driving limit on the maximum compute resources deployable.

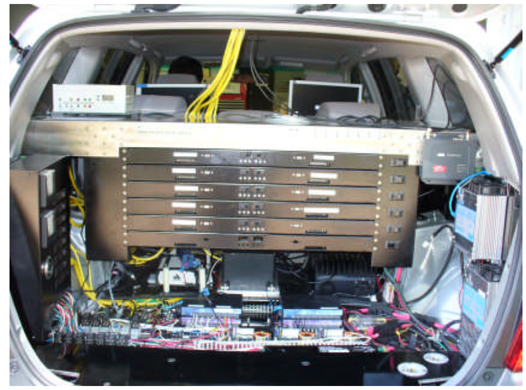


Figure 3: Computer Rack and Power Conditioning Equipment

The compute resources system designed for use in the Urban Navigator can consist of up to 12 ATX form factor motherboards with two available Gig-E network switches. The resources are housed in a custom designed and fabricated enclosure mounted in place of the third row seat of the vehicle as shown in Figure 3. Each motherboard is powered by a nominal 12 VDC sourced ATX power supply. The power supplies are capable of maintaining operation down to ~8 VDC. The compute resources currently deployed consist of 10 ATX server type socket AM2 motherboards with dual Gig-E network controllers built in. The CPUs deployed are AMD X2 4600 EE's. Booting and storage resources are 4 Gigabyte compact flash cards and 80 Gigabyte laptop hard drives. Operating systems deployed are Windows XP and Linux Fedora Core 6. I/O connectivity for sensors has been implemented via USB or Ethernet, to minimize complexity. The currently deployed system consumes between 25 and 45 amps at 14 VDC, depending on system load. Again, the particular computer resources were selected to minimize power consumption and to provide reliable operation.

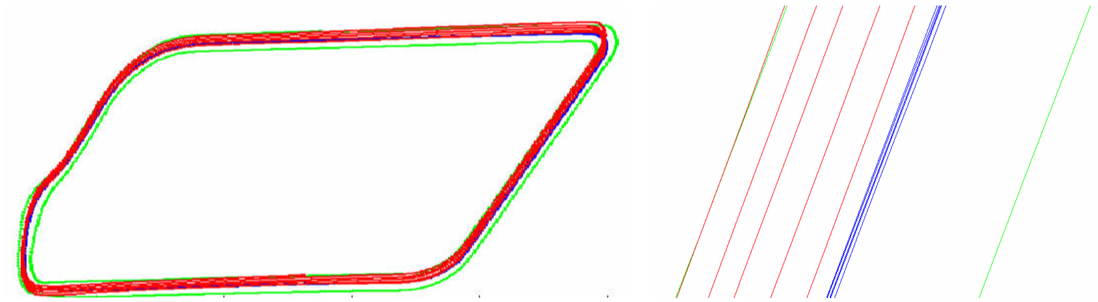
In-car development is facilitated via a dual head KVM connected to all motherboards. A rear facing monitor is mounted to the headrest of each front seat. Keyboards and pointing devices are located in the second row seats. Off-car monitoring and development is affected by an 802.11 b wireless bridge with a minimum 250 meter range.

2.3. Localization

Geo-localization is achieved using a Smiths Aerospace North-Finding-Module (NFM) combined with four GPS units and two odometers. The NFM is an inertial navigation system that maintains Kalman Filter estimates of the vehicle's global position and orientation as well as angular and linear velocities. The overall system (referred to as GPOS) is shown in Figure 4.

Figure 4: GPOS Components

The system design is predicated on redundancy and relative accuracy when GPS is lost. The GPS signal provided to the NFM will come from one of the four onboard GPS units. They include two Novatel Propak, V3-HP units with Omnistar subscription service, and two Garmin WAAS Enabled GPS 16 units. An onboard computer simultaneously parses data from the four GPS units and routes the best-determined signal to the NFM. This is done to maintain the best available GPS information to the NFM at all times. The best GPS solution is determined by evaluating each signal with respect to its unit type, mode of operation, HDOP,



(a) (b)

Figure 5: (a) GPOS Repeatability Data (b) Magnified View

RMS, number of satellites, and duration of uninterrupted signal among other criteria. The NFM has been programmed to use a different set of tuning parameters in its Kalman Filter depending on what type of GPS signal it is receiving.

In the event that all GPS units lose track of satellites, as seen during GPS outages such as when the vehicle is in a tunnel, the NFM will maintain localization estimates based on inertial and odometer data. This allows the vehicle to continue on course for a period of time; however, the solution will gradually drift and the accuracy of GPOS will steadily decrease as long as the GPS outage continues. Under ideal conditions the GPOS system typically maintains Global position accuracies and repeatability in the range of 0.1 to 0.5 meters. Figure 5 shows five laps around a 0.6 mile test track with GPS (blue lines) and five laps with no GPS (red lines). The vehicle was driven as close as possible in the center of the road (road edges are 28' apart and are marked by green lines) for every lap. Without GPS, the NFM was using only the encoder signals to damp the velocity errors. Under these conditions the GPOS system maintains Global position accuracies less than 5 meters for a distance traveled of approximately 3 miles without GPS.

2.4. High Level Planning

The High-Level Planner (HLP) provides overall guidance to the Urban NaviGator. Its high-level goals are to:

1. Create a representation of the RNDF that readily allows for efficient data manipulation during route planning,
2. Use the MDF to plan a route through the RNDF representation using an A* algorithm [10],
3. Periodically communicate waypoints to the Local World Model, so it has an accurate record of the immediate planning space,
4. Re-plan a route when obstacles are encountered, and
5. Collect data from the sensors about the domain as it is explored and populate the RNDF representation with these data so it contains a more accurate representation of the domain.

One problem with the RNDF is that it provides a rough representation of the entire domain. Figure 6a shows four waypoints from the provided, sample RNDF. The light blue line details the actual roads that should be followed while the dark blue line details the naïve, "as the crow flies" path between these points. Initially, every road segment within the RNDF is flagged as unexplored. When traversing a segment for the first time, "breadcrumbs" (additional waypoints) are created to fill in details about the segment, and the segment is marked as explored (see Figure 6b). The distance between these "breadcrumbs" will vary depending on characteristics (e.g., curves and road surface) of the segment. These "bread-crumbs" allow the representation of additional details about the segment that can be used as heuristic information by the A* algorithm in planning later missions.

(a) (b)

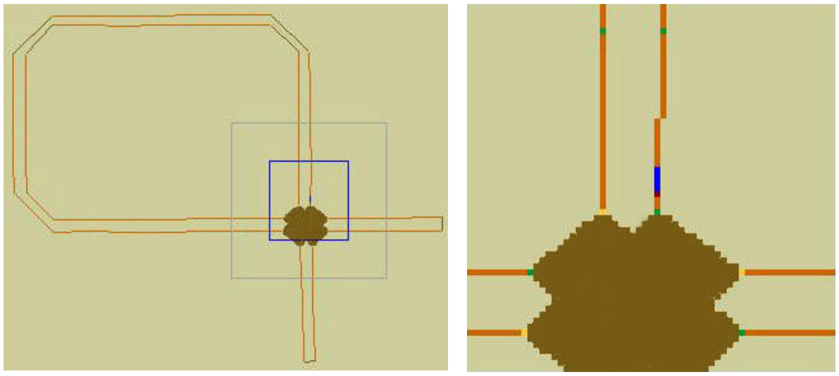
Figure 6: (a) Sparse RNDF Data ; (b) RNDF Data Augmented with History Information.

While the HLP maintains the overall A* generated mission path, The Local World Model works on a much finer (300m × 300m) scale. Periodically, the HLP determines and transfers to the Local World Model a local long-range (150 meter ahead) view of the domain. The Local World Model uses this information for localized path planning. As the Urban NaviGator advances along the local path generated by the Local World Model, any generated "breadcrumbs" and sensor data valuable to future path planning are communicated back to the HLP so they can be added to its RNDF representation.

2.5. Local World Model

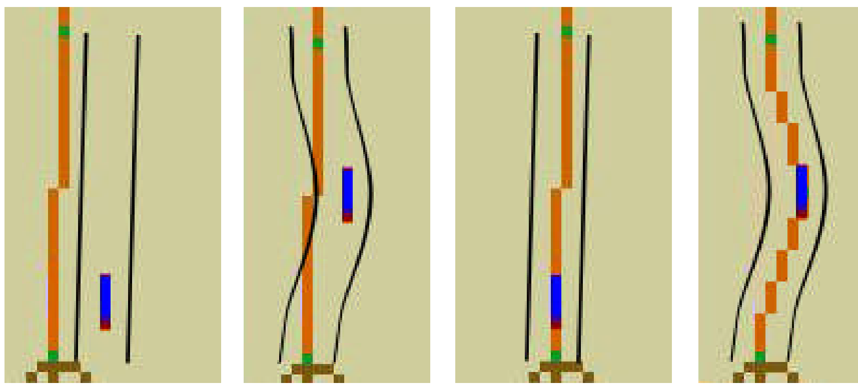
The Local World Model has multiple roles within the system architecture. First, it generates a model of the world based on the a priori RNDF. It receives a subset of the RNDF waypoints within a 300m × 300m area of the vehicle from the High Level Planner (HLP) and draws an estimated picture of the world into a rasterized grid using a resolution of 0.5m. This raster based approach was chosen because the output from the Local World Model can then be easily incorporated into other system components. The grid resolution of 0.5m was chosen from experience in the 2005 DARPA Grand Challenge, but can be varied depending on the mission. For example, it is anticipated that a finer grid will be needed when maneuvering in a parking lot scenario. Figure 7a shows an example of the 300m × 300m grid. Other components, such as the perception components, which are discussed in the next section, work with a smaller 60m × 60m grid. Any needed information is extracted from the 300m × 300m grid and can be transmitted to any necessary components. Figure 7b shows such a sub-sampled grid.

After the initial estimate of the world is created from the RNDF, the Local World Model will localize the vehicle position in the world using data from the GPOS component as well as lane finding and path finding sensors. The lane finding and path finding sensors are incorporated to account for possible discrepancies between the RNDF and the GPOS data. The Local World Model takes the information about the sensed world such as the number of lanes detected and the position of the center of the sensed lane, and adjusts the a priori world map to fit the sensed world. Figure 8 gives examples where adjustment is necessary. In this figure, the black lines represent the actual road as it is sensed relative to the vehicle, the brown lines are based on the RNDF, and the blue rectangle signifies the vehicle position which is based on GPOS. In (a) either GPOS is incorrect or the RNDF points are not in the lane. In (b) the waypoints do not



(a) (b)

Figure 7: (a) 300m × 300m Raster Local World Model ;
(b) Sub-sampled 60m × 60m Grid.



(a) (b) (c) (d)

Figure 8: Arbitration of Discrepancy in GPOS Data,
RNDF Data, and Sensed Data

describe the road accurately. Using data from the lane and path finding sensors the Local World Model accounts for these errors. In (c) the RNDF map has been shifted to align the RNDF road and the sensed world. In (d), the Local World Model has added additional waypoints to correct for the discrepancy between the RNDF and the real road.

Next, The Local World Model is responsible for characterizing, predicting, and injecting dynamic information into the world model, which can then be propagated throughout the system. A list of objects is received from the Moving Object and Classification sensor which provides the position, velocity, and size of the detected objects. The Local World Model overlays these objects onto the world map and estimates their future position based on velocity and direction of travel. A probabilistic determination of the future position is used to give regions where it is highly likely the vehicle will be. The addition of the moving obstacle information allows the Urban NaviGator to have a better understanding of what is happening in the world. Figure 9 shows the Local World Model output with a moving obstacle shown in blue and its estimated future position shown in red.

Finally, the Local World Model dynamically spools waypoints to the Receding Contro. After the HLP has planned a path that completes the mission, it provides a rough plan to the Local World Model that contains only the checkpoints, entry points, and exit points that need to be traversed. The Local World Model then takes the rough plan and fills in the intermediate waypoints that need to be traversed to travel from one HLP point to another. This provides the flexibility to modify the waypoints that need to be traversed based upon the current operating behavior without re-planning the overall mission. Figure 10 shows the change in the mission waypoints based upon a change in the operating behavior. In (a) all the mission points sent by the HLP are shown. This mission involves making a loop around the course and coming to a stop at the segment at the bottom. In (b) a number of intermediate points have been filled in to be sent to the RN. All points up to a set distance away from the vehicle are sent. In (c) the mission points have been shifted to the other lane in order to execute a change lane behavior due to the obstacle (in blue) detected in the same lane.

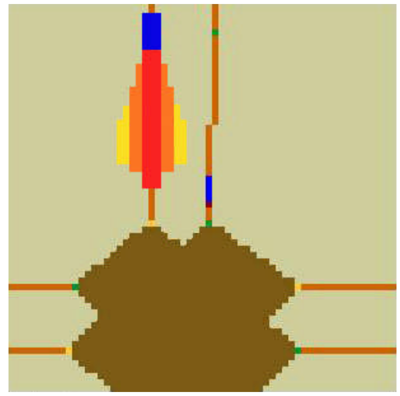
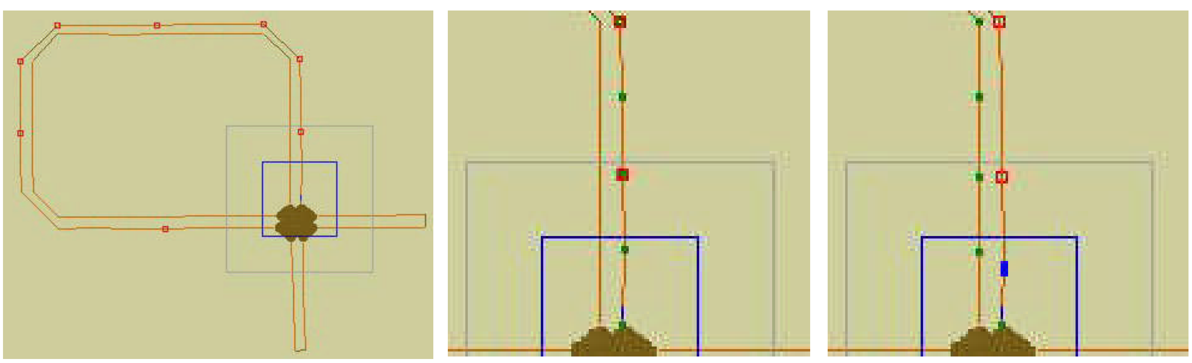


Figure 9: Moving Obstacle Depicted in Local World Model

In summary, the Local World Model provides a detailed 300m × 300m representation of the environment around the vehicle. It receives a priori roadway data from the High Level Planner as well as static and dynamic obstacle and lane information from the perception system. The Local World Model constantly estimates any discrepancies between the a priori and sensed data by calculating a net offset that can be applied to the a priori data in the event that sensed data is momentarily lost. Lastly, the Local World Model maintains a list of mission goal points



(a) (b) (c)

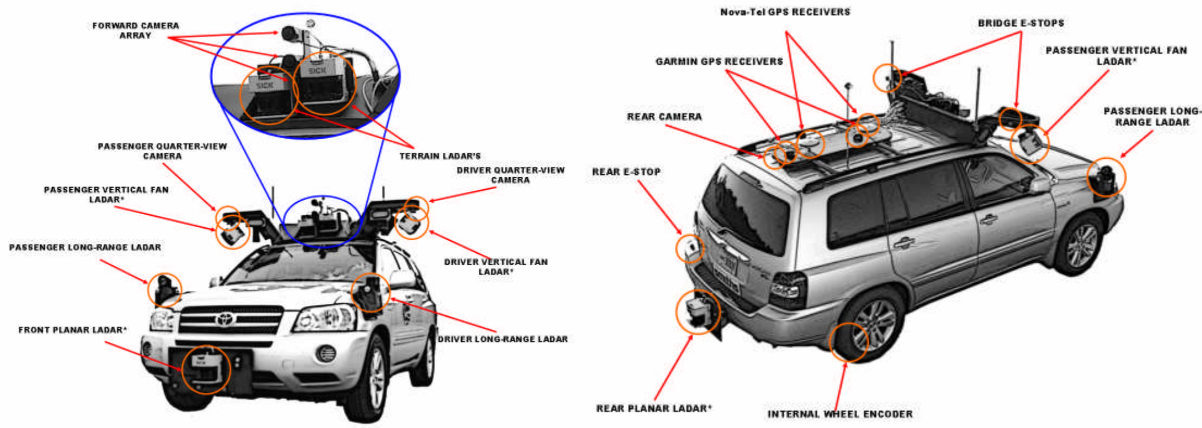
Figure 10: Dynamic Update of Intermediate Points

that identify the correct lane of travel. This information is transmitted to the Roadway Navigator (discussed subsequently) for motion execution.

2.6. Perception

2.6.1. Sensors

The sensor packaged deployed on the vehicle includes an array of LADAR and vision sensors. These include six of the SICK LMS-291 type LADARs, two of the SICK LD-LRS1000 long range LADARs, six Matrix Vision BlueFox high-speed USB2.0 color cameras, and an additional BlueFox camera configured to see in the near IR band. Moreover, many of the



(a) (b)

Figure 11: Sensors

sensors deployed on the vehicle are articulated with one degree of freedom. Figure 11 depicts the sensor configuration.

The LADAR sensing package is designed to provide high-rate obstacle and terrain evaluation. To do so, three categories of sensing components were designed to meet the needs of obstacle avoidance and terrain characterization. These categories include long-range obstacle detection, medium/short-range obstacle detection, and terrain classification.

For long-range detection two SICK LD-LRS1000 LADARs were mounted on both driver and passenger front quarter-panels. These sensors provide effective range information to 300 meters at rates of up-to 15 Hz with resolution at distance to within one meter. Moreover, the sensors are vertically offset to allow for the perception of differential height obstacles and to prevent the inadvertent flashing of each LADAR by the other.

Medium/short-range obstacle detection is accomplished by considering both obstacles detected by the long-range sensors in addition to those detected by any of the six LMS-291 type sensors. Of these, two are oriented in a planar fashion on articulated mounts, located on the front and rear bumpers, which allow the sensor to be pitched from 90° below to 20°above the plane of the vehicle. This feature is used to accomplish both enhanced terrain classification in tight-spaces such as parking lots as well as to maintain parallelism with the horizon of the road surface during roadway navigation behaviors. Figure 12 illustrates the front actuated planar LADAR.

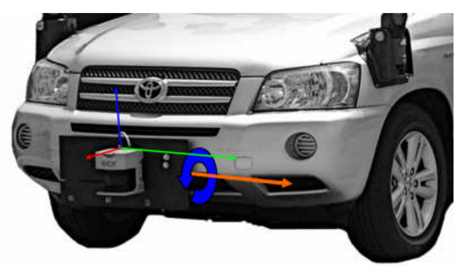


Figure 12: Front Actuated Planar LADAR

Negative obstacle and terrain classification is accomplished primarily through the use of an additional two LMS-291 LADARs mounted on the forward of the roof of the vehicle in a downward-pitched orientation with their focal points set at 20 and 30 meters respectively. These sensors are complemented by another two LMS-291 sensors located on the overhanging

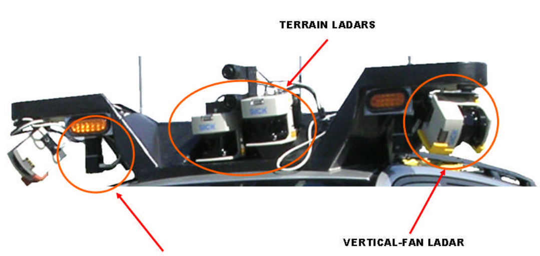


Figure 13: Sensor Bridge Components

wings of the sensor bridge on the vehicle roof. These sensors are also articulated and are configured in a vertical-fan orientation. The articulation of the vertical fan sensors provides active measurement of the environment surrounding the vehicle and plays a major role in vehicle behaviors including changing lanes, parking, and intersection behavior. Figure 13 depicts the sensor bridge located on the top of the vehicle.

2.6.2. Common Services

The sensor architecture deployed on the vehicle was designed to have common services and interfaces. These common services include the distribution of acquired sensor data from various sources (serial interfaces, Ethernet, USB, etc.) to other computing nodes who are requesting the information. All such data-traffic is handled on a managed L2 Gigabit Ethernet switch which is dedicated to sensor traffic. The services provided include streaming computer vision information, LADAR range data, and actuator feedback information. This information is consumed by various computing nodes known as *Smart Sensors* and used to generate high level information about environment traversability, obstacle detection, and vehicle localization.

2.6.3. Smart Sensor Concept

The smart sensor concept unifies sensor data and results into a generic format. To do so, a standardized data format and representation was designed which serves as the common data structure for the generation, transfer, and analysis of sensor information. This representation, known as a *Traversability Grid*, consists of a tessellated grid which is tied to a geo-spatial location at the time of generation. The grid, utilized in the previous Grand Challenges, has been expanded to provide a finer resolution of both traversability (the measure of cost for traversing a spatial location) and an expanded set of reserved values [8]. Moreover, the information represented in the grid can easily be used by arbitration components to generate higher order information and implement behavioral changes to drive planning and control components to successfully reach their goals. Figure 14 depicts three example traversability grids and the result of sensor fusion by an arbitration component.

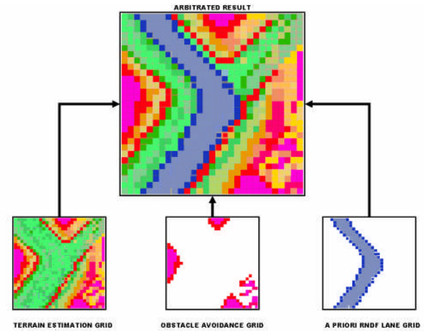


Figure 14: Smart Sensor Traversability Grid Fusion

By utilizing a common data representation, developers can work independently of arbitration components. Moreover, the Smart Sensing components can operate asynchronously at a wide variety of rates and grid resolutions due to the spatially located nature of the traversability grid [11]. This is possible due to the spatial mapping of each grid as it is fused with the other available sensor information. Thus, the arbitration process takes into account the geo-spatial offsets between the various Smart Sensor traversability grids when fusing their information such that the resulting representation is consistent as the vehicle moves regardless of speed, orientation, or position.

2.6.4. Traversability Smart Sensor

The traversability smart sensor was the primary sensor component that was used by the authors in the 2005 DARPA Grand Challenge. It is incorporated here because it identifies safe regions of travel, i.e. areas where there are not obstacles to impact and areas where the terrain is relatively smooth. Additional smart sensors will be discussed subsequently that address the specific problems of road and lane detection and dynamic object detection which are critical to success in the Urban Challenge.

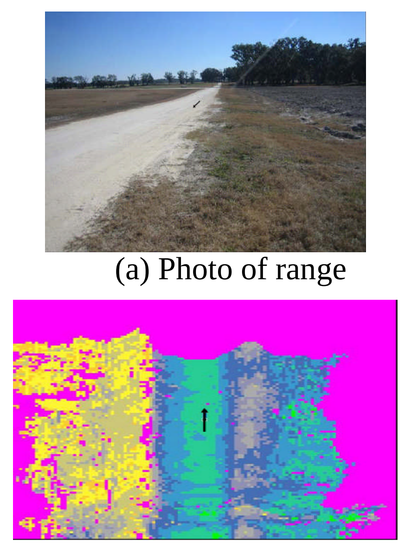
The traversability smart sensor component utilizes the eight LADAR based sensors on the vehicle to perform analysis for static obstacle characterization, terrain evaluation, and negative obstacle detection. These analyses are accomplished by utilizing the raw range and orientation information provided by each of the LADAR sensors to generate a point-cloud representation of the data. The general point information then undergoes a series of spatial transformations such that all data is mapped relative to the vehicle's inertial corrected global position and orientation (GPOS).

i) Vertical Obstruction Detection and Localization

The process of detecting vertical obstacles surrounding the vehicle is driven primarily by information obtained from the front and rear planar LADARs. This data provides consistent information about obstructions which are taller than the vehicle's bumper height and represent regions of the environment which are non-traversable. The two LMS291-S05 sensors provide range estimates in 0.5° increments over a 180° arc at 76 Hz. The articulated sensors attempt to maintain the plane of their measurements parallel to the ground plane with an offset equal to the 0.5 meter mounting height of the sensors. The globally mapped data points generated by the sensors are then mapped into the corresponding traversability grid cell in which they are bound. By maintaining statistics regarding the number of hits in a given cell, a traversability value is assigned. Moreover, the area between cell hits and the origin of the vehicle is designated as *free-space*. This is accomplished by the use of Bresenham lines drawn from the location of a hit cell to the sensor's origin. The cells returned by the Bresenham lines then have their number of hits reduced, and their traversability re-evaluated.

ii) Terrain Estimation and Characterization

Terrain estimation is accomplished by considering data generated by both the articulated vertical fan ladars located on the sensor bridge wings and the two forward facing pitched ladars on the center of the sensor bridge. By first mapping the globally referenced data points into the appropriate grid cell locations, the data is then analyzed for the variance, slope, absolute height, and mean height to generate an estimate of the best fitting plane within each given grid cell [12]. By comparing the slope and height of each plane in both an absolute sense and relative to the planes in neighboring cells an estimate of traversability is assigned to the given cell. Figure 15 depicts the output of the terrain estimation process.



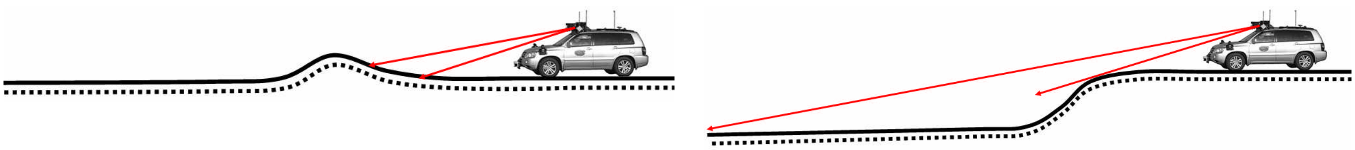
(a) Photo of range

(b) Terrain Estimation Grid

Figure 15: Terrain Estimation

iii) Negative Obstacle Detection and Localization

The detection and localization of negative obstacles is completed by considering the same data set used for terrain estimation. Particularly, the two forward facing terrain LADARs provide the most critical information regarding negative obstacles in the immediate path of the vehicle. To this end, the slope of the forward facing terrain ladars was tuned through testing with various sizes of negative obstacles such as pot-holes, ditches, and cliff-like changes in grade. Through this testing which has spanned the previous two Grand Challenges, an optimal declination for each LADAR was found to be 10° and 20° respectively. This declination results in the scan-lines of the LADARs hitting at roughly 32 meters and 20 meters ahead of the vehicle while on a relatively level road. The process of evaluating the information provided by the sensors for negative obstacle detection is similar in nature to that of terrain estimation. However, the focus of the negative obstacle evaluation is to look for sudden and drastic changes in range along the scan line. Such dramatic changes in range indicate that a given scan-line or region of a scan-line has crested over the threshold of a drop-off or hill and indicate a possibly hazardous part of the environment. Figure 16 displays two such cases where terrain and negative obstacle generate



(a) (b)

Figure 16: Traversability Smart Sensor (a) Terrain Estimation (b) Negative Obstacle Detection

obstacle information.

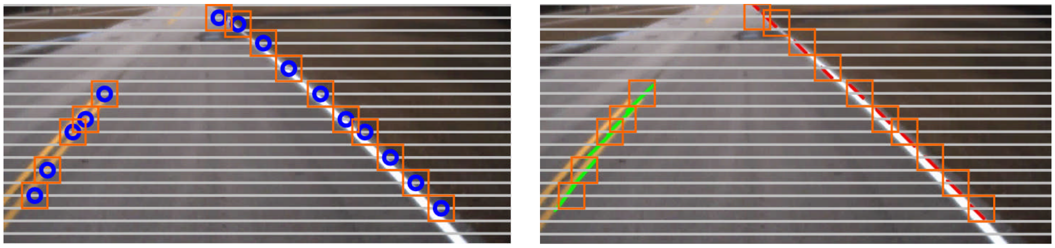
2.6.5. Lane Finding Smart Sensor

The Lane Finding Smart Sensor (LFSS) is a vision based sensing component which generates traversability information corresponding to the perceived locations of lane demarcations. Through a combination of analyses, images captured from each of the three forward facing and one rear facing BlueFox cameras are processed to yield geometric representations of significant painted demarcations and natural boundaries [13]. These entities are then analyzed for color content based on an adaptively tuned set of tolerances and classified appropriately. The result of the lane detection is a consistent and robust 2nd-order estimate of the location and orientation of perceived lane demarcations.

i) Lane Demarcation Detection and Localization

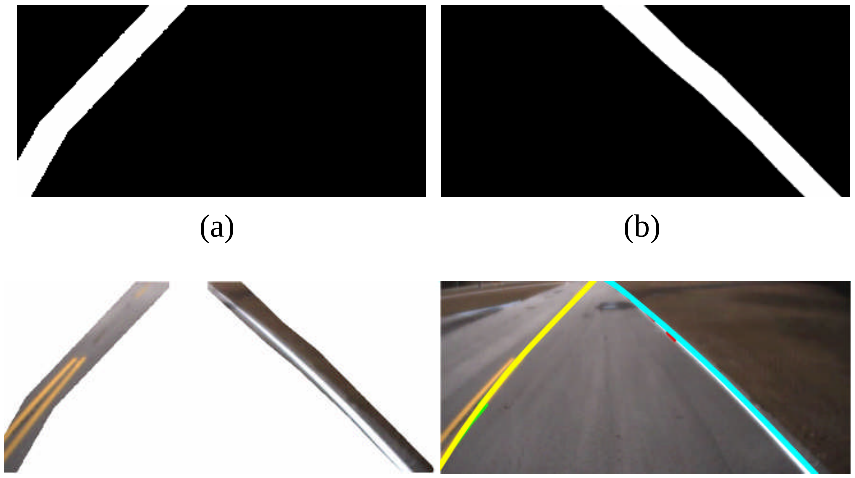
Detection of painted and natural lane/road demarcations is accomplished by considering multiple environmental cues. These cues include intensity, color content, shape, and orientation. To begin, the component takes a captured image and performs a row based analysis to detect significant points of interest. These points of interest are then used to extract sub-images from the source image which will then be searched for any dominant linear elements. This process is depicted in Figure 17.

The extraction of linear elements is performed by using the standard Hough transformation. The resulting sequence of linear elements is then sorted to find the most dominant linear element (if any) in a given sub-image. After extracting all significant line elements from the image, the resulting lines are first added to an optimized binary heap for sorting according to angular orientation in a polar reference frame with the origin at the upper left of the source image [14]. This sort aids in the rapid clustering of related elements into groups who are spatially related. Again, by utilizing a binary heap sorted on angular orientation, a first order approximation of the curve the cluster of line segments represents can be generated. This approximation is then used to generate a mask image as shown in Figure 18.



(a) (b)

Figure 17: Line Finder Image Analysis (a) Region of Interest Extraction (b) Linear Element Extraction.



(a) (b)

(c) (d)

Figure 18: Line Finder Image Masking (a) First Mask (b) Second Mask (c) Image from Masks (d) Least Squares Curves

The resulting masks are applied to the source image sequentially to analyze the color content in the masked region. This process serves both to identify the color and type of the line entity. The color analysis process is discussed in detail in the following section. When the analysis is complete, each cluster's point data is used to solve a second order approximation by use of the Least Squares method where the optimization function and the resulting partial derivatives are of the form [15]

$$E(A,B,C)=\sum_{k=1}^{N}\left(Ax_k^2+Bx_k+C-y_k\right)^2 \tag{1}$$

$$\frac{\partial E}{\partial A}=2\sum_{k=1}^{N}\left(Ax_k^2+Bx_k+C-y_k\right)\left(x_k^2\right),$$
$$\frac{\partial E}{\partial B}=2\sum_{k=1}^{N}\left(Ax_k^2+Bx_k+C-y_k\right)\left(x_k\right),$$
$$\frac{\partial E}{\partial C}=2\sum_{k=1}^{N}\left(Ax_k^2+Bx_k+C-y_k\right) \tag{2}$$

By solving the subsequent partial differential equations and for each clusters data set, a series of quadratic approximations are generated. These approximations have to date proven fairly robust given the cluster generation process.

ii) Color Classification and Adaptive Tuning

The color classification of the masked regions of the source image is performed in two processes: k-means clustering for adaptive tuning and color segmentation. The first process begins by taking sub-sampled training regions from the masked image. The RGB content of the cells in the masked regions are then used to populate a point distribution where the red, green, and blue values of each pixel in the region are mapped to the x, y, and z axes respectively [16]. A sample training region and distribution is presented in Figure 19.

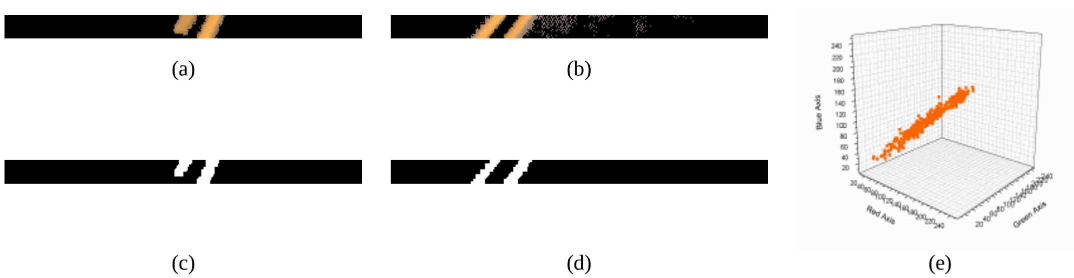


Figure 19: Color Distribution of Training Region Data

The color distribution depicted above is then processed using the k-means clustering algorithm to minimize the average deviation from the dominant line which is fit to the clustered pixel information. The k-means algorithm is expressed as

$$\sum_{j=1}^{k} \sum_{i \in S_j} \left| x^i - \mu_j \right|^2 \qquad (3)$$

where x is the RGB pixel vector, μ is the RGB K-mean, k is the number of clusters, and i is the number of pixels. With the color distribution generated from the training region, each mask undergoes color segmentation to evaluate whether a given pixel is likely white, yellow, or otherwise. By then analyzing the resulting set of segmented pixels, an estimate of the line's color content is generated as a normalized percentage of yellow and/or white content. By using multiple such training regions through the various masks, the color thresholds used in the segmentation process become more robust to dynamic changes in lighting or transient elements in the image such as glare and shadow. Figure 20 depicts the color segmented masks for the yellow and white channels of a given source image.

Figure 20: Color Segmented of Masked Regions (a) Yellow Channel (b) White Channel

From the above results, statistical information is generated including the number of color classified pixels, the total pixels in the mask, and the variance of the pixel content within the mask. Finally, a normalized percentage representation of how "yellow" or "white" is generated. The results of the normalized comparison of the number of pixels found to be of a given color to the total number of pixels found in the segment provide a estimate of the likelihood that a given cluster of line segments represent a given color of line. Moreover, the lack of significant color content to a region which has significant line content can be discerned to imply a natural boundary or transition from road to the surrounding environment.

iii) Stop-Line Detection and Localization

In addition to the statistics generated by the color segmentation process and the element clustering process, an estimate of the *general-lane-orientation* is produced by averaging the linear elements of the most significant clusters in the image. The result is a reasonable estimate of the orientation of the perceived road. From this information each cluster's average slope is then compared to that of the overall average in an effort to find any significant deviations. Such deviations indicate that a cluster is not only significant but also generally normal to the direction of the road. Such segments are inferred to have potential to be *stop-lines*. Figure 21 shows an example of an element cluster which has been determined to have significant divergence from the general orientation of the road and the sub-image which is extracted for further analysis.

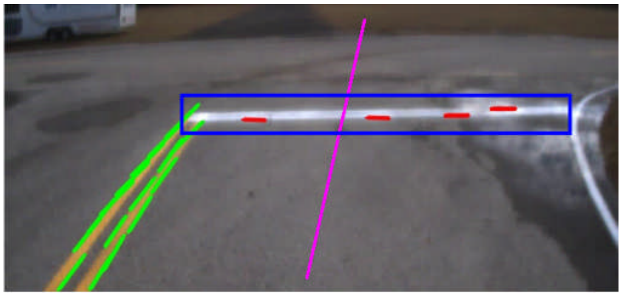


Figure 21: Stop-Line Detection

By generating an estimate of the location of the perceived stop-line, the smart sensor can aid in both safe roadway navigation and correction of a priori mapping data contained within the RNDF. These processes are discussed further in the section on Metadata generation.

iv) Traversability Grid Mapping and Generation

Thus far, the results of the various image processing techniques has only been discussed in terms of the perspective image space provided by the camera. However, to utilize the results of these analyses, the information must be mapped into the standard traversability grid. This process is accomplished through the use of a pre-calibrated transformation matrix and rotation matrix to project the pixel information from image-space to grid-space. The transformation matrix is generated from camera calibration data and then used to project the calculated line entities onto the traversability grid. However, whereas in the past Challenges, the entire image was mapped, the improvements in the most recent vision components and the geometric data they extract make it possible to project only a select set of points along each element. From there, the elements are reconstructed in the grid-space as a series of connected linear elements which approximate the original curve(s). Figure 22 depicts a sample lane-finder traversability grid with current, left, and right lanes painted.

(a) (b)

Figure 22: Lane Finder Traversability Grid (a) with lane boundaries and center estimation (b) with painted lanes

v) Metadata Generation

In addition to the basic grid representation of the perceived lanes and stop-lines, the Lane Finding Smart Sensor provides abstracted information in the form of metadata. This metadata currently contains information regarding the number of lanes perceived, the type of boundaries of the lane the vehicle currently occupies, and estimates of lane width at various distances ahead of the vehicle. Moreover, the metadata contains the estimated relative position of any perceived stop-lines and the vehicle's relative-pose within the lane. The later of the metadata is critical in both a priori correction and learning, and is also important in driving numerous behaviors.

2.6.6. Moving Object Detection and Localization

One of the most critical smart sensing components in development is the Moving Obstacle Smart Sensor (MOSS). The MOSS is responsible for the detection, localization, and classification of both moving and static obstacles which are perceived to either be in motion or have the potential to move. It should be noted that the choice of sensors for long-range detection was driven by the need to perceive moving obstacles to provide the necessary time response. In other words, the range of the chosen SICK LD-LRS1000 LADARS is a function of the minimum safe detection radius to provide a ten second decision and action period given a moving obstacle at a speed of 30 mph while assuming a closing speed of 60 mph. From these design criteria it was determined that the vehicle must be able to detect moving obstacles at a minimum range of 270 meters.

The detection process is accomplished via a fusion of sensing technologies. Foremost is the use of the long-range LADAR array. The raw range data returned by the array of planar-oriented LADARs is evaluated for vehicle sized obstructions. From this analysis a list of potential dynamic obstacles is generated. The list is then used to drive a vision based classification process which utilizes Haar classification. Next, the raw range data provided by the LADAR is mapped to an image and processed using the Hough transform to extract linear elements [17]. From this map, the relative orientation and location of the candidate obstacles is generated. Haar classification is then completed on the regions of the image which have been flagged as likely containing a vehicle. The classification then confirms or refutes the assertion that the obstacle in the given region is a vehicle. It is important to note that detecting stopped vehicle is as important as detecting moving ones as a stopped vehicle's future actions are inherently unpredictable. The result of these processes is the generation of a standardized vector representation of all candidate moving obstacles [18]. Information regarding the obstacle's location, velocity, classification, and eventually intent (turning, braking, changing lanes) is then provided to higher level planning elements to take appropriate action. Figure 23 depicts the sample output of the Moving Obstacle Smart Sensor.

(a) (b)

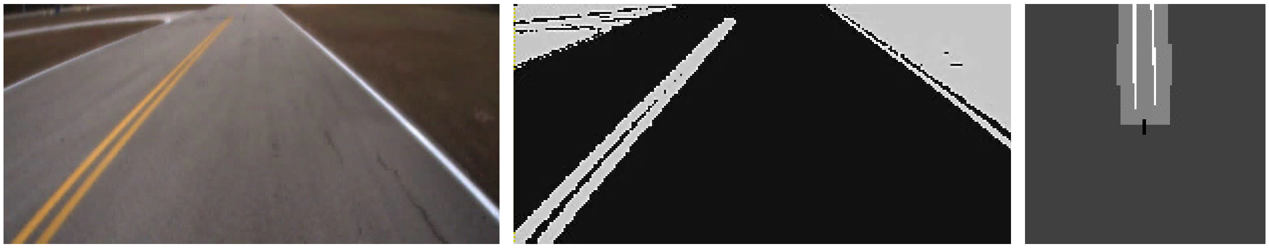
Figure 23: Moving Obstacle Smart Sensor (a) Original Scene (b) Point Data with Hough Lines

2.6.7. Path Finding Smart Sensor

The objective of the path finding smart sensor is to detect pavement regions. This is important in cases where a road has no lane markings or the lane markings are in poor condition or obscured. Vision based and ladar based components are integrated to accomplish this task.

i) Vision Based Path Finding

The Vision Path Finding Smart Sensor (VPFSS) utilizes the same color segmentation process employed in the Lane Finding Smart Sensor component with the exception that it is trained to search for asphalt and other such road materials and textures. By applying the segmentation to the captured images the component provides a simple yet powerful approximation of where the road is located and adds value to the arbitration process. Figure 24 depicts the output of the VPFSS.

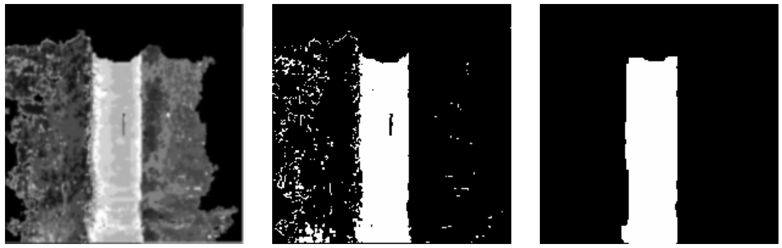


(a) (b) (c)

Figure 24: Vision Path Finding Smart Sensor (a) Original Scene (b) Segmented Image (c) Traversability Grid

ii) LADAR Based Path Finding

The LADAR based Path Finding Smart Sensor (LPFSS) functions in much the same manor as the VPFSS in that it relies on a combination of image processing techniques to extract a simplified estimate of the location of smooth, drivable terrain. By generating a topographic image of the terrain from the terrain estimation LADAR data, the component generates a virtual picture of the terrain. This picture is then processed through a series of thresholds, erosions, and other enhancement filters to isolate the flat and smooth regions of the image. Figure 25 depicts this process and the resulting traversability grid.



(a) (b) (c)

Figure 25: LADAR Path Finding Smart Sensor (a) Topographic (b) Threshold (c) Traversability Grid

The resulting traversability grid is then processed for line content to determine the right most dominant linear bound of the path which in turn is used to maintain a right bias while operating in an unstructured environment.

2.7. Adaptive Planning Framework

Team Gator Nation has developed and deployed the Adaptive Planning Framework [19] to address the issues associated with behavior mode selection. This concept is presented in this section. The specific behaviors and arbitration strategy associated with the Urban Challenge problem is presented in the subsequent two sections.

In the Adaptive Planning Framework, the system is assumed to be able to operate in a finite number of behavior modes. These behavior modes govern how the vehicle operates under various driving conditions. The framework is predominantly used to make intelligent decisions pertaining to these behaviors. The framework is scalable to systems of varying complexity and size and is compatible with existing architectures such as JAUS RA-3.2, NIST 4D/RCS, and others. The Adaptive Planning Framework is composed of three principle elements tasked with assessing the situation, determining the suitability and viability of all possible solutions, and executing the most suitable of all recommended solutions.

2.7.1. Situation Assessment Specialists

Dynamic environment information, originating from any array of sensors is monitored and managed by the Situation Assessment Specialists. Each specialist design is tailored to the sensor or collection of sensors whose data it will be analyzing. These specialists can, but are not required to, “live” on the same computing node that directly receives the sensor input. While the inputs to the specialist can come from any data source, the output or “finding” must adhere to specific guidelines outlined by the framework. Findings can be in the form of conditions, state, or events. A condition may have a value of present or absent only. All conditions are by default absent and must be proven present at each iteration. A finding classified as a state can only exhibit one of many a priori states. The event category is reserved for findings whose occurrence at some point in time is of significance even after the initial finding has passed. Once the findings have been generated the information is disseminated to all other components that might need it.

An example of a situation assessment specialist would be a software component whose sole function was to determine if it is safe to move to the adjacent lane. This component would monitor sensor data as reported by the Traversability Smart Sensor and reach a Boolean conclusion which would be stored as metadata for use by other processes. A second example would be a software component whose sole function was to determine if it is ‘legal’ to move to an adjacent lane. Here ‘legal’ is defined as not crossing a yellow line or not changing lanes when near approaching an intersection.

2.7.2. Behavior Specialist

The findings rendered by the Situation Assessment Specialists are consumed by the behavior specialists. There is a one-to-one mapping of each behavior with a behavior specialist. The role of the specialist is to monitor the findings and evaluate the suitability of its behavior under the current perceived operating conditions. As with the specialist findings, the default recommendation is unsuitable and must be proven appropriate at every iteration of the program to ensure truth of the results and operating safety. This specialist does not possess the ability to activate or deactivate its associated behavior; such authority is only given to the Decision Broker.

2.7.3. Decision Broker

At the highest level of the framework lies the Decision Broker. Its role is to monitor all Behavior specialist recommendations. It assumes ultimate authority over how the Urban NaviGator will operate while in autonomous mode. Like the other entities within the framework, the Decision Broker can base its conclusions on not only the recommendations and findings of other specialists, but it may also look at data from any other pertinent source. Team Gator Nation’s implementation of the Adaptive Planning Framework centralizes all the Decision Broker functionality within the JAUS Subsystem Commander and has the added responsibility of selecting which component receives control of the vehicle’s JAUS Primitive Driver. The framework architecture employs an asynchronous, iterative, forward chaining reasoning approach to decision making.

2.7.4. Data Marshalling and Metadata

The Adaptive Planning Framework does not specify a specific means for data marshalling nor does it restrict the system architect to a particular data type or structure for distribution. The Urban NaviGator deployed by Team Gator Nation is JAUS RA3.2 compliant. Consistent with the team’s existing JAUS implementation, the Adaptive Planning Framework data transport uses a publish/subscribe model. This allows for an arm’s length transaction between parties and eliminates a step in latency associated with a centralized blackboard approach. Truth maintenance and transport is handled by way of a JAUS Node Manager with added Metadata Management capability. A component may simply subscribe to a publisher’s “mailing list” and they will automatically receive an update every time that publisher has an updated finding or whenever there is a periodic synchronization pulse. Furthermore the subscribers will have knowledge of publisher’s state and will be aware of any abhorrent behavior that may lend itself to misinformed findings.

2.8. Behaviors

The Urban NaviGator is programmed with six behavior modes. The corresponding behavior specialist constantly evaluates the appropriateness of its behavior mode and the decision broker determines which mode will have operation of the vehicle.

2.8.1. Roadway Navigation

The Roadway Navigation behavior is the primary driving behavior deriving commands to be sent to the vehicle actuators while the objective is lane following. This behavior will allow the vehicle to navigate the roadway within the lines of its desired lane and maintain a safe following distance behind any vehicles ahead.

2.8.2. Open Area Navigation

Open area navigation is a behavior that should only be needed in special circumstances during the Urban Challenge event. This behavior allows the vehicle to move towards a goal location without striking any object and while avoiding any rough terrain. This is in effect the only behavior mode that was required in the 2005 DARPA Urban Challenge. It will be useful in the Urban Challenge when the vehicle is in an open area such as a parking lot prior to performing an actual parking maneuver.

2.8.3. Change Lane Maneuver

The change lane maneuver will be used in passing situations or in cases where the vehicle must change lanes in a multi-lane road in order to pass through a mission goal point. The behavior will constrain the vehicle to remain within the lane boundaries of the new lane.

2.8.4. Reverse Direction

This behavior is called whenever it is determined that the current lane is blocked and there is no alternate clear lane available for passing. It will also be applicable in cases where the vehicle has traversed into a 'dead end' road in order to reach a mission goal point.

2.8.5. Intersection Traversal

The intersection traversal behavior will be applicable when the vehicle enters the vicinity of an intersection. This is one of the most complicated behavior modes in that the system must rely on a series of situation assessment specialists to safely navigate the intersection. This behavior mode must handle queuing, stopping at the stop line, determining right of way, and ultimately traveling through the intersection while avoiding other vehicles.

2.8.6. Parking Lot

This behavior must deal with the problems that arise in the parking lot scenario where precise motion is necessary. When the vehicle approaches the vicinity of an assigned parking space, precise path planning will be initiated to align the vehicle as required. Situation assessment specialists will be monitoring the near surroundings of the vehicle to center the vehicle in its parking space while avoiding any static or dynamic objects.

2.9. Smart Arbiter

The purpose of the Smart Arbiter component is to generate a 60m × 60m traversability grid, centered at the vehicle's current position, which is used to implement a desired behavior. Motion execution, which is discussed in the next section, is accomplished via an A* search through this grid to determine the least cost path. In most cases, the least cost path will be obvious as the grid has been constructed to accomplish a desired action. <u>An important feature of this entire approach is that specific behavior modes can be changed with smooth continual control of the vehicle</u>.

The Smart Arbiter will attain inputs from the Terrain Smart Sensor, the Lane Finding Smart Sensor, the Path Finding Smart Sensor, and the Local World Model and will build its grid based on the current behavior mode of the system. For example, if the system is in the Roadway Navigation behavior, then the grid cells corresponding to the positions of the line on the edge of the lane as identified by the Lane Finding Smart Sensor will be marked as non-traversable regions in the Smart Arbiter grid. The cells corresponding to the road lane will be marked as highly traversable. This will prevent the planner from planning outside the current lane.

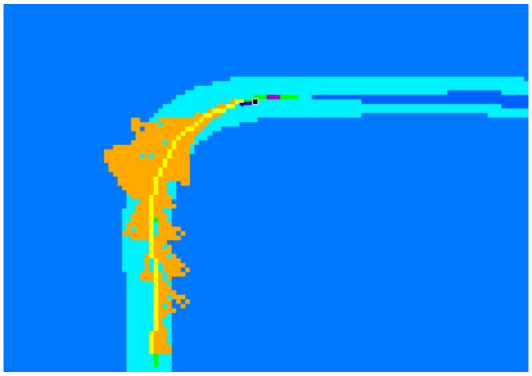


Figure 26: Path Search Algorithm

2.10. Motion Execution

Real time motion planning is accomplished via a receding horizon controller. Receding horizon is a form of model predictive control (MPC) used to solve complex and constrained optimization problems [20]. The application of suboptimal MPC to nonlinear systems such as the Urban NaviGator is given in [21]. In this case, receding horizon control is used to optimize a trajectory through the traversable space the vehicle will encounter. Inputs to the control component include the sensed, cumulative 60m × 60m traversability grid which is assembled by the Smart Arbiter component.

Local path planning is accomplished by means of an A* search algorithm [22]. The goal is to optimize the cost of the trajectory from the current vehicle position to a goal position, and thereby find a set of open loop actuator commands that minimize the cost of traversal. The search finds different trajectories by generating possible input commands and extrapolating them through a vehicle kinematics model. The model used is that of a front-wheel steered, rear-wheel drive vehicle. The costs of the different trajectories are based on distance traveled and the traversability grid values encountered. Figure 26 shows path segments that were searched and the final determined path. Tests with the algorithm have shown that it is able to calculate an optimal path at an update rate over 30 Hz.

Closed loop control is achieved by repeatedly running the optimization algorithm as traversability grid data and vehicle state information are updated. Steering actuator commands are optimized through the A* search, while throttle and brake commands used to control speed are handled using a simple PID controller.

3. Results and Performance

The previous section presented the ten primary elements that comprise Team Gator Nation's NaviGator vehicle system. Although much of the discussion focused on *how* each element works, information was also presented about *why* each element was designed as it was. Implementation examples and results associated with most of the ten elements were presented in the previous section for clarification. This section will focus on results and performance associated with integration and implementation. It must be noted that the development process is ongoing so that complete results are not presently available.

Figure 27 shows the results from a recent Roadway Navigation test. The roadway is marked by the green lines and has a nominal width of 28’. Figure 27b shows a magnified view of one of the corners. Lane tracking and traversal have been successfully demonstrated as shown in the figures. Although this is a simple result, it does also demonstrate that the necessary components such as the Local World Model and the Smart Arbiter have been developed and integrated with the Receding Horizon controller and the Primitive Driver. These are the important building blocks that must first be in place before the more complicated behavior modes can be introduced.

Team Gator Nation is working hard to implement all Basic Navigation and Basic Traffic behaviors in June 2007. This will of course be followed by the advanced functions that are required for the Urban Challenge.

4. Conclusion

The performance requirements identified in the Urban Challenge Technical Evaluation Criteria are challenging. The system must be able to detect and model its environment and then plan and execute appropriate actions in real time.

The approach described in this paper was generated after careful consideration of the design requirements. The central concept is the integration of a priori and sensed information in a raster format in the Local World Model. Based on this information, an appropriate behavior is selected via arbitration. The behavior is executed by generation of a Roadway Navigation grid coupled with metadata.

The primary new contribution of this approach is that related to solving the technical challenges of (a) the reconciliation of differences in estimated global pose, a priori data, and sensed information, (b) the determination of the appropriate behavior mode, and (c) the smooth transition of vehicle control between behavior modes. These particular developments are being implemented and integrated with the necessary hardware and software components to address the requirements of the DARPA Urban Challenge.

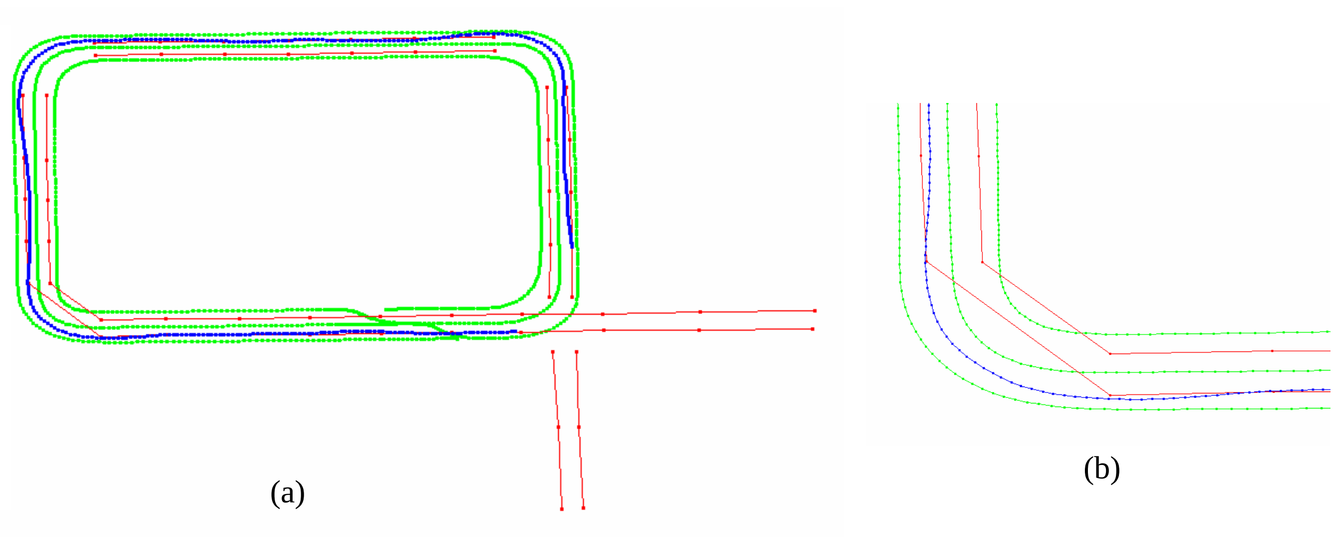


Figure 27: Data from Autonomous Roadway Navigation

5. References

[1] DARPA Urban Challenge Technical Evaluation Criteria, 16 Mar 2006.

[2] Urmson, C., et al., "A Robust Approach to High-Speed Navigation for Unrehearsed Dessert Terrain," Journal of Field Robotics, Vol. 23, Issue 8, Aug 2006.

[3] Trepagnier, P., et al., "KAT-5: Robust Systems for Autonomous Vehicle Navigation in Challenging and Unknown Terrain," Journal of Field Robotics, Vol. 23, Issue 8, Aug 2006.

[4] Mason, R., "The Golum Group/ University of California at Los Angeles Autonomous Ground Vehicle in the DARPA Grand Challenge," Journal of Field Robotics, Vol. 23, Issue 8, Aug 2006.

[5] Lakhotia, A., et al., "CajunBot: Architecture and Algorithms," Journal of Field Robotics, Vol. 23, Issue 8, Aug 2006.

[6] Travis, W., et al., "SciAutonics-Auburn Engineering's Low-Cost High-Speed ATV for the 2005 DARPA Grand Challenge," Journal of Field Robotics, Vol. 23, Issue 8, Aug 2006.

[7] Miller, I., et al., "Cornell University's 2005 DARPA Grand Challenge Entry," Journal of Field Robotics, Vol. 23, Issue 8, Aug 2006.

[8] Crane, C., et al., "Team CIMAR's NaviGATOR: An Unmanned Ground Vehicle for the 2005 DARPA Grand Challenge," Journal of Field Robotics, Vol. 23, Issue 8, Aug 2006.

[9] Touchton, R., Galluzzo, T., Kent, D., and Crane, C., "Perception and Planning Architecture for Autonomous Ground Vehicles," IEEE Computer, Dec. 2006, pp. 40 – 47.

[10] Coppin, Ben, Artificial Intelligence Illuminated, Jones and Bartlett Publishers, Sudbury, MA, 2004, ISBN 0-7637-3230-3, pp. 108-109.

[11] Touchton, R., et. al, "Planning and modeling extensions to the Joint Architecture for Unmanned Systems (JAUS) for application to unmanned ground vehicles," SPIE Defense and Security Symposium, Orlando, Mar 2005.

[12] Solanki, S., "Development of Sensor Component for Terrain Evaluation and Obstacle Detection for an Unmanned Off-Road Autonomous Vehicle," Ph.D. Disertation, Univ. of Florida, 2007.

[13] Nanning, Q., and Cheng, H.. "Springbot: a Prototype Autonomous Vehicle and Its Algorithms for Lane Detection." IEEE Trans. on Intelligent Transportation Systems 5, (2004).

[14] Odagiri, K., Kobayashi, K., Watanabe, K., Umino, H., and Nakaho, N., "Development of Active Contour Extraction for Autonomous Vehicle Lane Detection at Outdoor Environment." Society of Automotive Engineers, 2001.

[15] Matthews, John H., Numerical Methods for Computer Science, Engineering and Math, Englewood Cliffs, New Jersey: Prentice-Hall, 1987, 224-252.

[16] Lee, J., Crane, C., Kim, S., and Kim, J., "Road Following in an Unstructured Desert Environment using Monocular Color Vision as Applied to the DARPA Grand Challenge," Proceedings of the International Conference on Control, Automation, and Systems (ICCAS 2005), Gyeong Gi, Korea, June 2005

[17] Gonzalez, R.C. and Woods, R.E., Digital Image Processing, Prentice Hall, 2002.

[18] Viola and Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," Proceedings IEEE Conf. on Computer Vision and Pattern Recognition. 2001.

[19] Touchton, R. "An Adaptive Planning Framework for Situation Assessment and Decision-Making on an Autonomous Ground Vehicle," Ph.D. Dissertation, University of Florida, 2006.

[20] Mayne, D. Q., Rawlings, J. B., Rao, C. V., Scokaert, P. O. M., "Constrained model predictive control: Stability and optimality," Automatica, Volume 36, Issue 6, June 2000, pp. 789-814.

[21] Scokaert, P.O.M, Mayne, D.Q., Rawlings, J.B., "Suboptimal Model Predictive Control (Feasibility Implies Stability)," IEEE Transactions on Automatic Control, Volume 44, Issue 3, March 1999, pp. 648-654.

[22] Hart, P.E., Nilsson, N.J., Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, SSC4 (2), (1968), pp. 100-107.