LOCAL SENSOR DATA FUSION AND ITS APPLICATION TO AUTONOMOUS VEHICLE NAVIGATION

By

PHIL OSTEEN

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2009

To my Mom and Dad

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Carl Crane, for introducing me to the field of autonomous robotics and giving me guidance during difficult times. The last two years at the Center for Intelligent Machines and Robotics (CIMAR) have proven invaluable to my growth as an engineer, and I thank everybody from CIMAR who has helped me over this time. The intense work leading up to the 2007 Urban Challenge and the ensuing trips to Virginia and California are experiences I will cherish throughout my career. I would finally like to thank Dr. Prabir Barooah, who taught an excellent class on data analysis and estimation that has already proven valuable for my work.

TABLE OF CONTENTS

LIST OF FIGURES

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

LOCAL SENSOR DATA FUSION AND ITS APPLICATION TO AUTONOMOUS VEHICLE NAVIGATION

By

Phil Osteen

May 2009

Chair: Carl Crane
Major: Mechanical Engineering

The rate of development of autonomous vehicles over the last five years has been remarkable. Advancements have been made at such a pace that between DARPA's first Grand Challenge in 2004 and the Urban Challenge in 2007, vehicles have gone from failing to complete basic navigation tasks to successfully navigating urban streets. Significant progress has been made in the controls and planning areas, with navigation abilities improving as a partial result of improvements in Global Positioning precision. Hardware advancements have been made as well, with both computers and sensors increasing data storage and processing capabilities. As a result, recent research has focused on combining the advanced computational abilities of computers with the increasingly relevant data provided by local sensors. This thesis deals with combining advancements in the fields of sensing and controls to improve upon existing autonomous navigation architectures. The Lane Finder Arbiter, a software component created for this research, provides an interface between raw sensing components and vehicle navigation components.

The problem statement is first described, followed by a review of similar research and a description of prior research within the Center of Intelligent Machines and Robotics (CIMAR) that led to the creation of the Lane Finder Arbiter. The Lane Finder Arbiter, the focus of the thesis, is then described. The statistical methods used for this research are then discussed, and finally the results obtained from testing are analyzed. These results are used to draw conclusions about the Lane Finder Arbiter's current strengths, as well as possible future improvements to the new navigation architecture.

CHAPTER 1
INTRODUCTION

Problem Statement

The traditional architecture of autonomous vehicles relies on local sensor data to detect, localize, and classify objects in the immediate environment. While this information is useful for decision making and obstacle avoidance, it is not directly used for vehicle localization and navigation. Rather, the majority of autonomous vehicles rely on a globally referenced sensor such as the global position system (GPS) for vehicle navigation.

A common navigation approach, the "Waypoint Navigation" method, has the vehicle navigate between globally defined goal waypoints, or "goalpoints", in a roadway via Global Position (GPos) measurement data. Depending on the distance between goalpoints, intermediate GPos "breadcrumbs" are inserted to properly define the roadway. The insertion of breadcrumbs to help properly define the roadway is illustrated in Figure 1-1. The vehicle software components are provided a-priori global roadway data from a Route Network Definition File (RNDF); therefore goalpoints and breadcrumbs can be projected ahead of the vehicle's current location.

The GPos data is determined by fusing GPS data with Internal Measurement (IMU) data, often using Kalman based filtration techniques [1][2]. The IMU measures position and orientation using data from multiple local sensors, including multi-axis accelerometers, wheel encoders, magnetic compasses and gyroscopes. The IMU inertial data corrects raw GPS data and filters large discontinuities in the GPS solution by integrating inertial information over small time spans with less frequent GPS solutions. Nevertheless, the navigation systems are constrained by the inherent limitations of GPS systems.

All sensors are somewhat dependent on their environments, but in the case of GPS, environmental effects can lead to significant errors and impair a vehicle's navigation performance. This issue shows the limitations of the GPos waypoint navigation method and identifies an area of traditional navigation architectures that can be improved. The sources of GPS measurement error include obstructions above the receiver, the orientation and number of transmitting satellites, and the global position of the receiver.

Atmospheric conditions, as well as satellite orientation with respect to the receiver, contribute to GPS measurement inaccuracies. Non-uniform atmospheric conditions lead to varying travel times for transmitting signals, which leads to unpredictable errors. This phenomenon is illustrated in Figure 1-2. The relative geometry of transmitting satellites and the receiver influences GPS measurement inaccuracies, as is shown in Figure 1-3, from [3]. Measurement error increases when satellites are clustered or when they are co-linear with respect to the receiver [4].

Other GPS measurement errors occur as a result of signals reflecting off nearby objects and arriving at the receiver at an incorrect time, called the multipath effect. The multipath effect, shown in Figure 1-4, is common in areas with tall buildings and can lead to measurement errors of up to a few meters [3]. One of the greatest sources of error in GPS systems is the occlusion of satellites by objects such as tunnels, trees, and tall buildings. Such obstructions are significant because if a large portion of the satellite constellation is blocked, the GPS solutions become invalid.

GPS systems are also susceptible to Electro Magnetic Interference (EMI) and Radio Frequency Interference (RFI), which is uncommon but still more of an issue for GPS than for localized sensors. RFI interference with GPS systems can cause total navigation failure in many

implementations of autonomous navigation systems. This was evidenced by Carnegie Mellon's GPS issues at the starting gates of the DARPA Urban Challenge, which were the apparent result of RFI caused by a nearby JumboTron television display.

One of the best commercial GPS systems is the Wide Area Augmentation System (WAAS); whose measurement errors are typically 3 meters or less [5][6]. The WAAS system uses multiple ground reference stations to calculate a differential correction to the raw GPS data received from satellites. The corrected GPS measurement offers a dramatic decrease in error compared with raw GPS data, whose error can reach up to 15 meters [5][6]. For many applications, WAAS accuracy is within the desired bounds; however precise autonomous vehicle navigation in complex environments requires that position measurement errors be less than 1 meter. While there are high-precision GPS systems available, they are still susceptible to the aforementioned issues and often carry a prohibitive price tag.

The heavy reliance on GPos data for vehicle navigation introduces the risk of unsafe vehicle behavior due to position measurement errors. This potential risk is well known, and the possible failure modes have received increased attention as the focus of autonomous vehicle research has moved into urban environments. As a result, various research groups are exploring the integration of vehicle referenced local sensor data into the GPos based navigation systems.

Urban Navigator

The test vehicle for the work presented in this thesis is CIMAR's entry in the 2007 DARPA Urban Challenge, the Urban Navigator. The Urban Navigator, show in Figure 1-5, is a Toyota Highlander hybrid which was chosen for its advanced power system as well as its efficient fuel consumption. The Urban Navigator consists of an array of twelve dual core computers as well as six SICK LMS-291 ladars, two SICK LD-LRS1000 ladars, and six Matrix

Vision BlueFox high-speed USB2.0 color cameras. The GPos system consists of a GE Aviation North-Finding-Module (NFM) combined with two GPS units and one odometer. The NFM is an IMU that uses Kalman filtering techniques to estimate the vehicle's global position and orientation, as well as angular and linear velocities.

The full vehicle sensor diagrams are shown in Figure 1-6 and Figure 1-7. The role of each sensor was determined according to three general goals for the sensing components within the vehicle architecture. The three goals are to classify the terrain, to detect and classify moving objects, and to localize the vehicle within a road lane. Some sensor components fulfill multiple roles using the same raw data set, such as the Terrain Smart Sensor (TSS). Using different classification models, the TSS determines the traversability of the environment and also estimates the center of the road lane from a single data set.

The Lane Finder Arbiter functions as an interface between the sensor components and the navigation components of the vehicle. The Lane Finder Arbiter synthesizes incoming data from three independent sensor components, and anneals this data in order to localize the vehicle in the roadway. The TSS component provides offset data at 14 and 24 meters ahead of the vehicle to the Lane Finder Arbiter and uses a SICK S-14 and a SICK S-05 laser rangefinder to provide this information. The S-14 consists of 401 laser scans per cycle (100 degree scan window at 0.25 degree scan resolution), while the S-05 rangefinders consist of 181 laser scans per cycle (90 degree scan window at 0.5 degree scan resolution). The camera Line Finder component uses two MV BlueFox 121C cameras with a resolution of 1024x768 pixels, and provides offset corrections at whichever distances it chooses. Finally, the laser Curb Finder component provides offset corrections at the origin of the vehicle reference frame coordinate system, shown in Figure 1-8, and uses a SICK S-05 laser rangefinder.

Related Work

Local Sensor Navigation

Various research topics deal with the use of local sensor data to improve vehicle navigation. A number of articles published by Baltzakis [7][8] attempt to perform vehicle navigation tasks using only local sensor data, however the experiments have been performed in controlled indoor environments. The work represents an important step toward the goal of the research of this thesis, which is to use local sensors to drive an autonomous vehicle in the absence of reliable GPS data. More advanced models and filtration algorithms are required, however, to navigate a vehicle in an uncontrolled outdoor environment.

Object Characterization

Ongoing research is devoted to making characterizations of specific objects in the environment, such as a roadway in the case of the Lane Finder Arbiter, using reliable local sensor data. The accurate characterization of environmental features improves the predictive capabilities of robots. For example, the movement patterns are predictably different between pedestrians and vehicles; therefore the vehicle can interpret these objects differently once they are identified. The research of Thorpe, et al. [9][10] focuses on identifying objects using local sensor data and fusing the data into a map of the local environment. This data is used to detect and track objects such as pedestrians, vehicles, and curbs. The sensor data is fused into a single grid, and driving decisions are made based on characterizations provided by the sensors. The fusion of multiple sensor components, each making different characterizations of the environment, is similarly a focus of this thesis.

Other research is devoted to analyzing of the density of objects to make a characterization between dense and sparse objects. Lalonde [11] develops algorithms to distinguish between

dense surfaces (buildings), sparse surfaces (bushes), and linear surfaces (light poles), when determining paths for autonomous vehicles. Statistical analysis of an increasingly large and reliable data set, including fused data sets from multiple sensors, can therefore increase the capabilities of autonomous vehicles by making important characterizations about the environment.

SLAM

Simultaneous Localization and Mapping (SLAM) algorithms are conceptually similar to the topic of this thesis—the goal of any SLAM algorithm is to use local sensor data to build a map of an unknown environment and estimate the global position of a vehicle [12][13]. Local sensors identify landmarks, which are environmental features that are distinguishable from the surrounding environment, such as curbs, vehicles, and pedestrians. The landmarks are often identified using one of two algorithms, the Spike algorithm or the Random Sampling Consensus (RANSAC) algorithm [12].

The Spike algorithm takes sensor data at each time step and classifies landmarks based on relative range differences between each laser scan. If the difference between consecutive scans exceeds a certain threshold, a landmark has been detected.

The RANSAC algorithm randomly chooses a sample of sensor data at each time step and creates a least squares curve fit to the data. The algorithm then determines how many data points lay close to the fit, and if the number of close points exceeds a threshold (consensus) then the fit defines a landmark feature.

After the landmarks are identified at each time step, the SLAM algorithm fuses the landmark data with that of the Internal Measurement Unit (IMU) to estimate the vehicle's position. If the initial position of the vehicle is accurately known, then any errors inherent in

IMU measurements will be mitigated by accurate and persistent landmark identification. The process is demonstrated in Figure1-9-Figure1-11, from [12].

The feature identification process of the SLAM algorithm is similar to that of this research; but the difference between the two processes lies in the application of landmark information. The Lane Finder Arbiter makes no attempt to estimate the position of the vehicle in a global frame, but instead provides position estimates of the vehicle relative to the center of a lane, which is done in the vehicle's local reference frame. In fact, the Lane Finder Arbiter specifically tries to reduce the reliance on global positioning (GPos) data to navigate a roadway. The offsets provided by the Lane Finder Arbiter to the vehicle navigation component, the Roadway Navigator (RN), are relative to the vehicle's current position and heading. The Arbiter does not, for example, know the distance from the vehicle to an upcoming intersection. This could be known if the sensor data was transformed into a global reference frame as with the SLAM algorithm.

Assuming there are indeed GPos and in particular IMU errors, the SLAM algorithm would be affected by these errors while the Lane Finder Arbiter would not. Also, if the sensors cannot identify any landmarks at a certain time, the SLAM algorithm must rely on IMU data. In the same situation, the Lane Finder Arbiter can project lane offsets ahead of the vehicle by using recent landmark data. Finally, the SLAM algorithm assumes that landmarks found in a given scan can be matched with landmarks found in a later scan. It is a non-trivial problem to correlate landmarks found during particular scans to those found in subsequent scans, and the SLAM algorithm fails in the case of false landmark correlations. In this case, the vehicle would update its perceived position based on a false correlation of landmarks, and the vehicle's position estimate will have an increased error. This problem is illustrated in Figure 1-12, from [13].

The effective use of local sensors to characterize a vehicle's surroundings is a topic of interest in the robotics community, and the Lane Finder architecture attempts to solve this problem using a local reference frame approach. In theory, the Lane Finder architecture could be implemented with any array of sensors that can provide offset estimations, however the current implementation on the Urban Navigator combines sensors that are well suited for roadway feature identification. The current implementation of the Lane Finder architecture, including how its data is used by the vehicle control components for roadway navigation, is described in the following chapter.

Feature Detection Algorithms

Preliminary research was performed to determine the feasibility of using a component such as the Lane Finder Arbiter to localize a vehicle within a road lane. The roadway feature identification models, which now provide input correction to the Lane Finder Arbiter, were developed and tested during this period. The laser curb identification algorithm employs a Spike approach to determine if a curb has been detected. The curb identification algorithm, demonstrated in Figure 1-13 [14], analyzes the relative angles between consecutive laser scans to identify curbs in the roadway.

The camera line finding algorithm is not a RANSAC algorithm, however it does similarly create a least squares approximation to incoming sensor data. The line finding process consists of a series of image filtrations as described below, yet the data is still dependant on lighting conditions. The cameras therefore are contained in a camera enclosure as shown in Figure 1-14, and an external filter is also attached to the camera to mitigate environmental effects.

The software filtration process, implemented by Velat, et al [15], is a combination of multiple image processing algorithms. The camera filtration process is shown in Figure 1-15- Figure 1-17, from [15]. The filtration process begins by transforming the image data to a red-

channel version of the image, which is good for maintaining yellow and white road lines while eliminating many environmental colors. Canny edge detection is performed on the new image as described in [16]. A color segmentation algorithm is then performed on the source image to classify the lines found in the edge detection algorithm as either white or yellow lines. In this manner, added information can be extracted from the camera data and interpreted in a more intelligent manner, similar to the way humans intelligently interpret road lines according to color. Finally a Hough transformation as described in [17] is applied to the source image. Lines are classified as either left lane boundary or right lane boundary lines, from which the centerline of the lane can be estimated. Road width estimates are also determined by extracting depth information from the multiple camera images using stereo vision techniques described in [18][19].

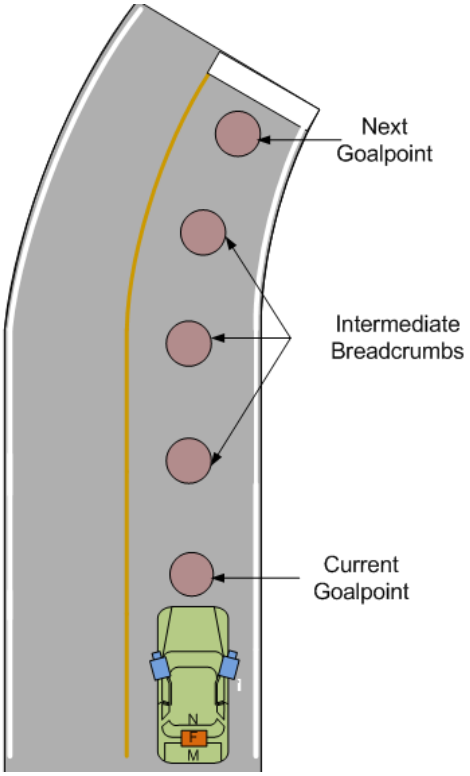

Figure 1-1. Autonomous navigation using GPos waypoints.

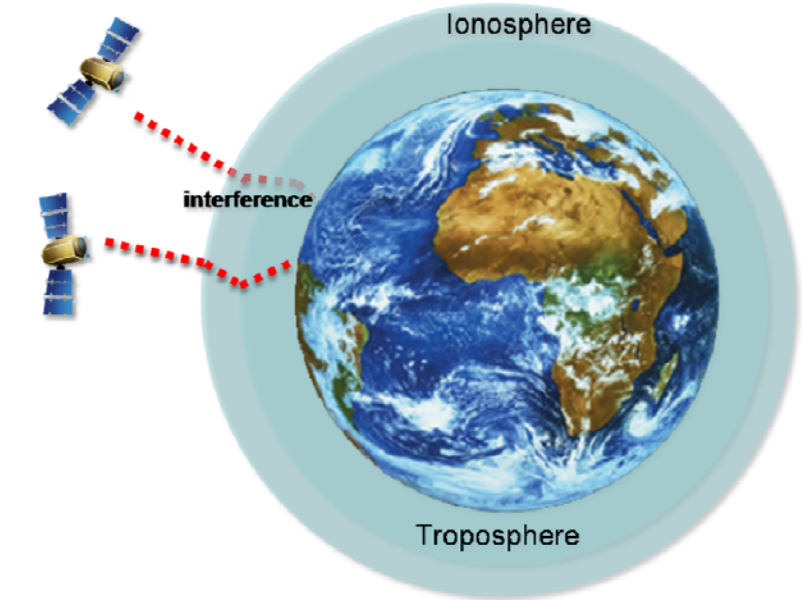

Figure 1-2. Atmospheric effects on GPS accuracy. The atmospheric interference causes latencies in the received satellite signals, which result in position estimation errors.

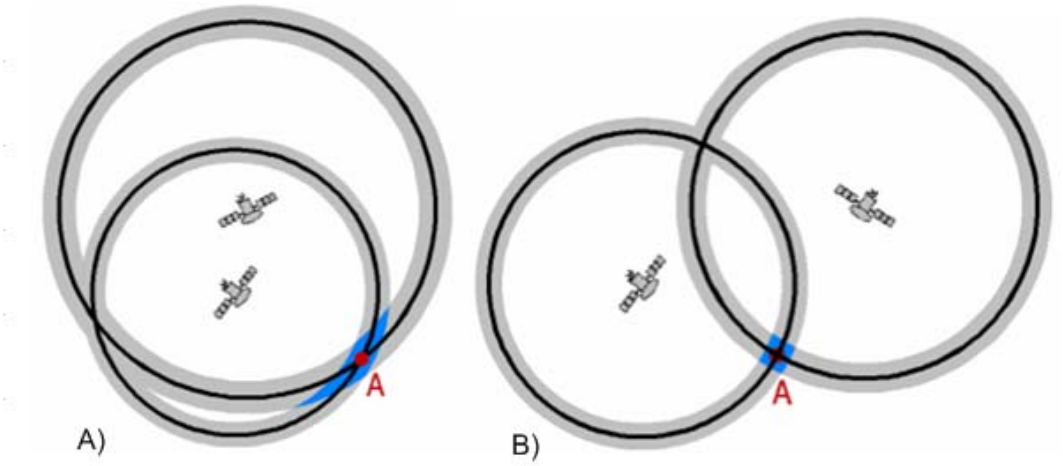

Figure 1-3. Satellite geometry effects on GPS accuracy. A) Poor satellite geometry with respect to receiver. B) Good satellite geometry with respect to receiver.

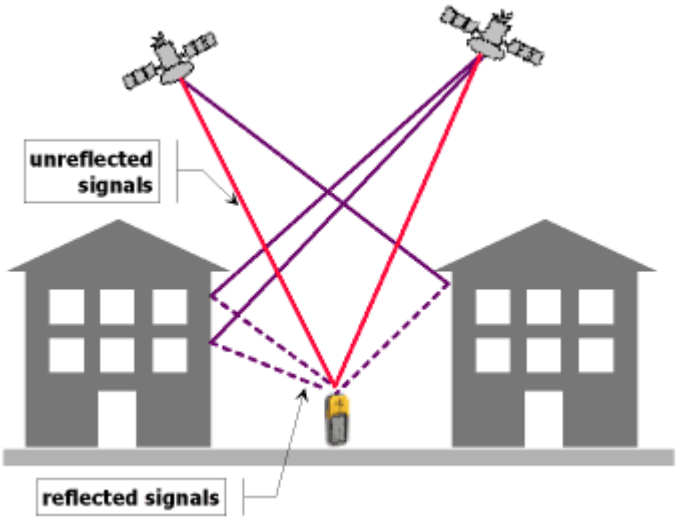

Figure 1-4. Multipath effects on GPS accuracy.

Figure 1-5. Team Gator Nation's Urban Navigator.

Figure 1-6. Front view of Urban Navigator sensor package. The Lane Finder Arbiter input data comes from the passenger vertical fan ladar, the two quarter-view cameras, and the terrain ladars.

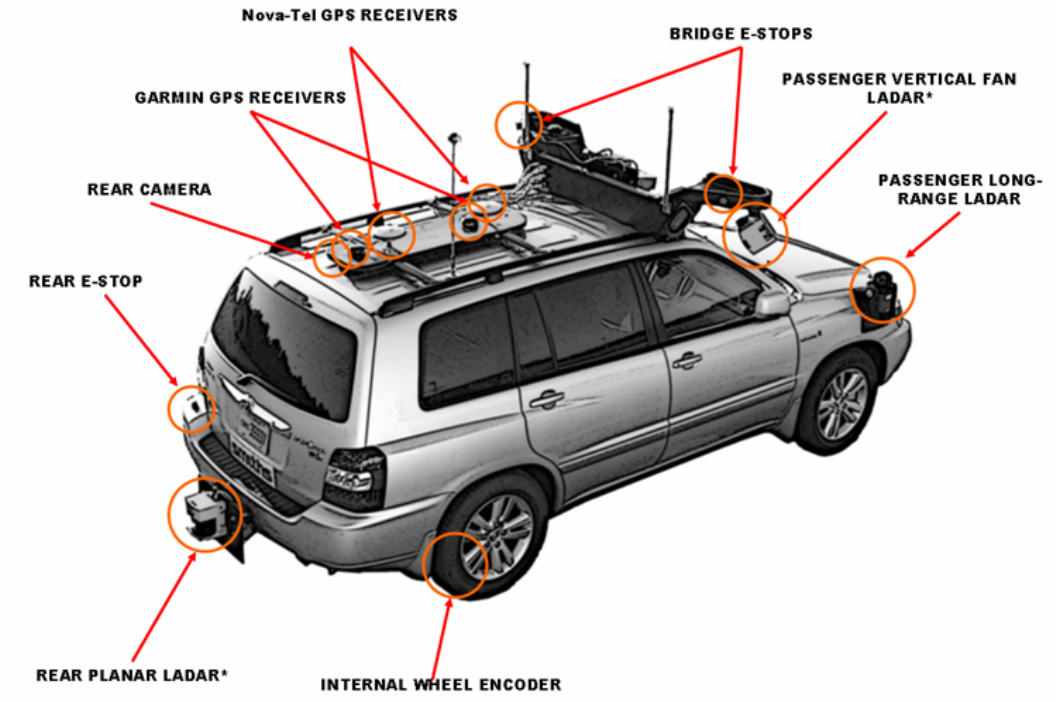

Figure 1-7. Rear view of Urban Navigator sensor package.

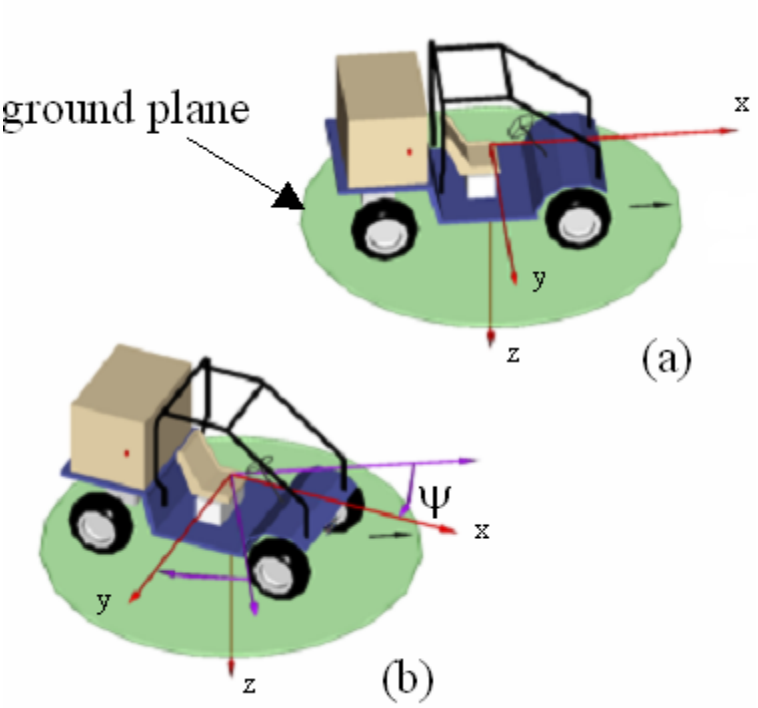

Figure 1-8. Local vehicle coordinate system. A) instantaneous coordinate system of the vehicle before movement, and B) new coordinate of the vehicle shown with previous coordinate system, illustrating change in yaw of the vehicle. For the Urban Navigator, the origin of this coordinate system intersects the centerline of the vehicle and the rear axle of the vehicle.

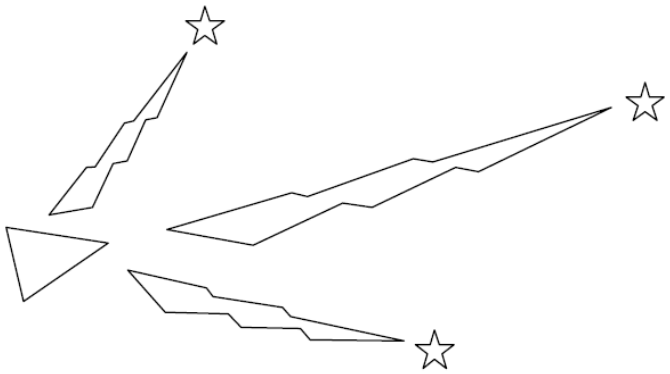

Figure 1-9. Initial condition for the SLAM process, vehicle at a known position detects three landmarks. Local sensors give relative location of landmarks with high precision.

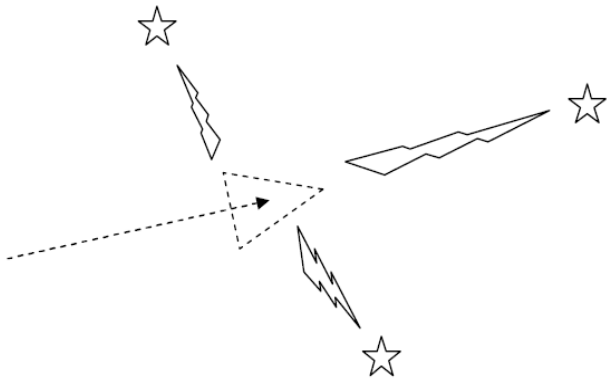

Figure 1-10. Vehicle moves from known initial position in SLAM process. Vehicle moves to new position, shown as measured by IMU, and local sensors measure the new relative location of the landmarks.

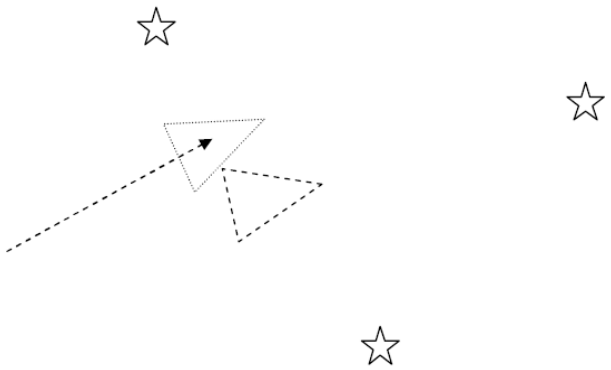

Figure 1-11. Local sensor landmark identification corrects position estimate in SLAM process. New relative locations of landmarks are used to correct the IMU position estimate.

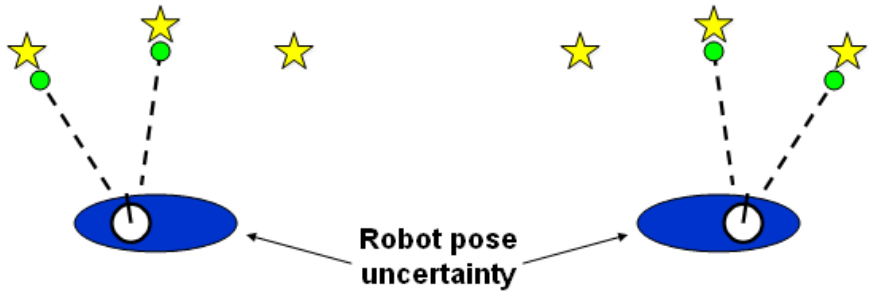

Figure 1-12. Landmark correlation failure with the SLAM algorithm. Depending on whether the landmark identification correlates the left landmarks or the right landmarks to previous landmarks, the robot position estimation will be different.

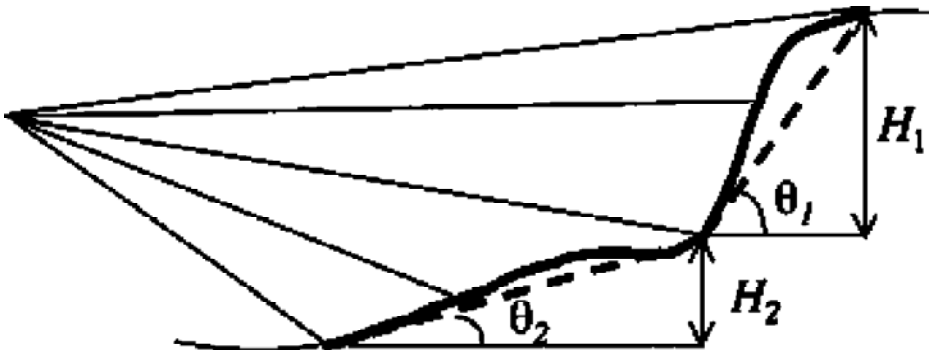

Figure 1-13. Five consecutive laser strikes on a sloping surface. H_2 represents a bump in a road while H_1 represents a curb.

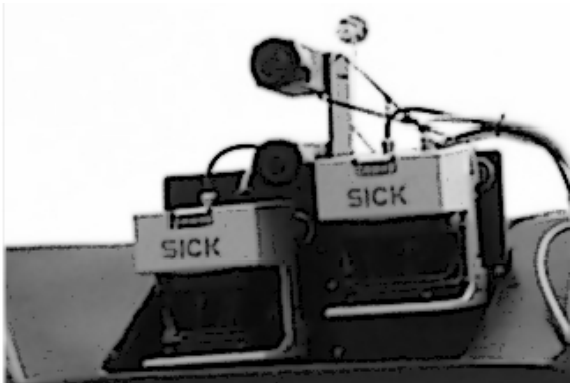

Figure 1-14. Camera and laser array atop the Urban Navigator. The cameras are mounted in opaque enclosures and have external filters to mitigate lighting effects.

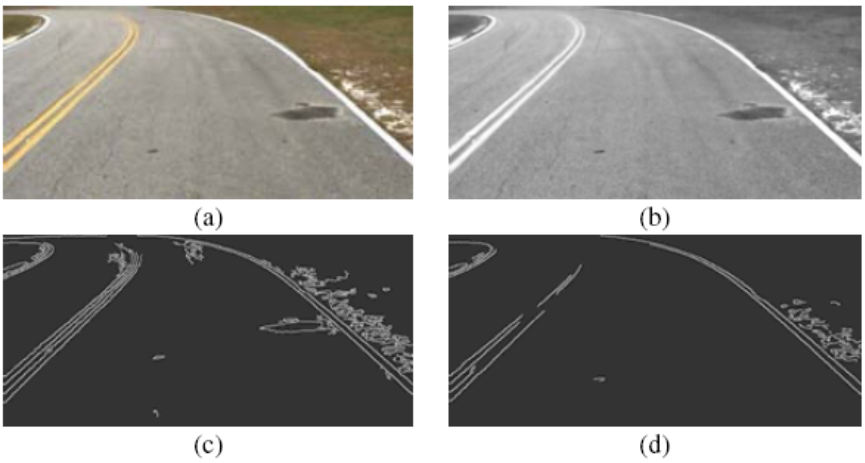

Figure 1-15. Canny edge detection process. A) Source image. B) Red channel filtered image. C) Canny edge detection with 50/200 threshold values. C) Canny edge detection with 130/200 threshold values.

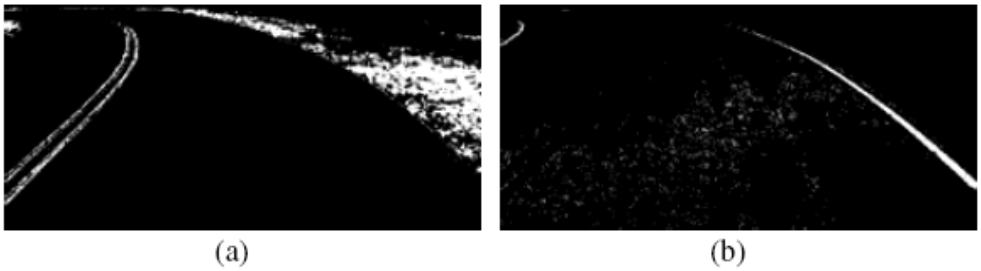

Figure 1-16. Color segmentation images. A) Yellow color filtration result. B) White color filtration result.

Figure 1-17. Post Hough Transform result image. Red lines estimate centerline location ahead of vehicle, green lines estimate road width ahead of vehicle, blue and yellow lines represent lane boundaries.

CHAPTER 2
LANE FINDER ARBITER

Vehicle System Architecture

The three basic upper level systems on most autonomous vehicles are the sensing, intelligence, and control systems. On a lower level perspective, other systems (data routing protocols, for example) are vital as well, but from an upper level perspective these are the three prevalent systems. The software component architecture of the Urban Navigator, shown in Figure 2-1, can be thought of as a control loop. The loop begins with the sensing components providing new information about the vehicle's surroundings to the intelligence components. The intelligence components make a decision about the vehicle's behavior and recommend course of action, which is executed by the control components. The vehicle then responds to the control commands, and the loop is completed as the sensing components evaluate the new surroundings.

Many in the robotics community see the largest potential for improvement in the sensing field, especially as hardware becomes cheaper and more advanced [20]. The problem of extracting as much meaningful information as possible from millions of data points per second is one of the most difficult problems in autonomous vehicle robotics. As a result, many sensing software algorithms are not intelligent—they don't use vehicle state information to decide how to interpret the raw sensor data. Increased intelligence at the sensing level, coupled with the ability to make more valuable characterizations, will improve the capabilities of autonomous vehicles. It is not as important, for example, for a camera component to try to characterize lines in the roadway during an intersection or N-point turn behavior as it is during a roadway navigation behavior. Although the sensing components themselves are not intelligent in the Lane Finder architecture, the Lane Finder Arbiter intelligently interprets the incoming sensor

data, sending lane correction information only when the vehicle is in the Roadway Navigation behavior state.

A large volume of data is used by the Lane Finder Arbiter to estimate a best fit curve to the correction data, allowing the Lane Finder Arbiter to confidently quantify the "goodness" of the fit. Figure 2-2 and Figure 2-3 show views of the Lane Finder Arbiter visualizer, and demonstrate the difference between a high confidence data fit and a low confidence fit. The white line in Figure 2-2, projected on the screen ahead of the vehicle, represents a high confidence curve fit. No line is projected in the case of a low confidence curve fit, as is shown in Figure 2-3. The Lane Finder Arbiter sends outgoing lane correction data to the vehicle navigation components only when there is high confidence in the best fit curve.

Messaging Structure

When the Lane Finder Arbiter is confident in the generated curve fit, it sends correction information ahead of the vehicle to the two intelligence components responsible for safe navigation, the Roadway Navigator and the Local World Model. The two components use the correction information differently, and both of the applications are integrated into the system architecture to improve navigation performance. The Local World Model uses the correction information to evaluate the accuracy of current GPos data, while the Roadway Navigator uses the information to improve the performance of the search algorithm.

The Roadway Navigator receives breadcrumbs ahead of the vehicle from the Local World Model and from the Lane Finder Arbiter, which are used to determine how to navigate the vehicle. This process is illustrated in Figure2-4, with the Lane Finder Arbiter breadcrumbs being used during Roadway Navigation behavior. At each time step the Roadway Navigator performs an A* search, shown in Figure 2-5, ahead of the vehicle to determine the least cost path. The

Lane Finder and GPS target points constrain the search window area, thus making the search process more efficient.

Difficulties arise with this system in the case of consistent GPS measurement errors. The A* search does not yield a consistent result at consecutive time steps and as a result, the driving route changes slightly at each time step, resulting in poor steering performance. To account for this impaired navigation, and to prevent more dangerous situations that result from GPS error, the Lane Finder Arbiter also sends a correction to the Local World Model.

The Local World Model is a component that fuses a-priori roadway data, provided by GPos surveying, with a persisting list of obstacles to generate a global map of the vehicle's surroundings. An issue with this process is that transforming local sensor obstacle data to a global reference frame causes the global positioning errors to project onto the previously accurate local sensor data. The projection of GPos errors onto obstacle locations is an issue of great concern, as it could lead to unsafe vehicle behavior.

The Local World Model uses a global gridmap, shown in Figure 2-6, and inserts obstacles into this map before sending navigation recommendations such as speed and steering to the Roadway Navigator. To avoid errant obstacle locations in the global gridmap, GPos errors must be resolved when using the gridmap data for navigation purposes. The errors are resolved using the Lane Finder Arbiter correction—if the Local World Model perceives that GPos data consistently disagrees with Lane Finder correction data, the gridmap will be shifted to account for this inconsistency. The new gridmap data is then sent to the Roadway Navigator, and the new navigation target points provide a consistent path to follow, solving the problem of impaired steering performance. This process is illustrated in Figure 2-7. The gridmap shift therefore

solves both the problem of impaired navigation performance and obstacle location error, under the assumption that Lane Finder correction data is reliable.

Point Transformations

Incoming corrections from the sensor components are inserted into a persistent lane correction list. To avoid saturating the list with corrections while the vehicle is stopped, input corrections are only added to the list while the vehicle is moving. Since data filtering takes place at each sensor component before a correction is sent to the Lane Finder Arbiter, the corrections and their weights are inserted into the correction list without further filtration. The weight of each new correction depends on the confidence assigned to the correction by the given sensor component. The list stores each correction until the weight value of a correction decays beyond a threshold and the correction is deleted.

Before new corrections are added to the list, the existing list corrections must be decayed and transformed into the new vehicle coordinate frame. The correction locations are transformed according to inertial data provided by the Velocity Smart Sensor (VSS). The VSS component provides speed information (meters/second) as well as the yaw rate (radians/second) to various components, including the Lane Finder Arbiter. The transformation is performed using the matrix techniques discussed in Duffy and Crane [21].

For the coordinate system given by Figure 1-8, the transformation is modeled as a pure rotation about the z-axis, followed by a pure translation in the x-y plane. For high data update rates, such as the VSS messaging rates of 30 Hz, the change in yaw ($\Delta\psi$) and the change in velocities (Δx , Δy) are assumed to be linear. The resulting transformation matrix is given by

Equation 2-1. After the transformation of the existing list corrections is completed, the new input corrections are added to the list and confidence in the list data is determined.

$$ {}^{t}_{t+\Delta t}T = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\Delta\psi) & -\sin(\Delta\psi) & 0 & 0 \\ \sin(\Delta\psi) & \cos(\Delta\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2\text{-}1) $$

Regression Techniques

Least square regression techniques are used to determine whether the best curve fit to the correction list data is reliable enough to aid in vehicle navigation. When the list is relatively small, with fewer than 20 data points, the list is deemed too small to achieve a reliable curve fit. As the list grows to contain more than 20 corrections, a least squares regression is performed on the list data to determine the best curve fit.

With a minimum parameter of 20 points, the Lane Finder Arbiter generally takes less than 1 second to begin sending corrections to the control components. It is also likely that multiple sensor components will have contributed at least one correction over this time span, providing the list with correction data from independent sources. Given that, on average, over 100 points are persisted in the lane correction list at a given time, the 20 point minimum affects only two situations. The first situation is during the component's startup state, and the second is when the vehicle has switched from a specialized behavior into Roadway Navigation behavior. In both cases, testing confirmed that false data was not being sent to control components as a result of premature curve fitting.

Curve Fitting

In order to properly evaluate incoming data and determine confidence in the outgoing data, a characterization of the nature of the data set must be made. The offset data in the Lane Finder Arbiter's correction list is assumed to be given according to Equation 2-2. The known offset values (contained in Y) are therefore assumed to be a linear function of the known range values (contained in H) and deterministic coefficient values (contained in θ), subject to random errors (contained in e). The error values are assumed to be normally distributed, with zero mean and variance σ^2; and are also assumed to be uncorrelated with one another. The offset values are also assumed to be normally distributed, with mean μ and variance σ^2. This presents a basic parameter identification problem, with a goal of finding an optimal estimate of the deterministic parameter θ.

$$Y = H\theta + e \tag{2-2}$$

Least Squares Approach

Assuming that the data takes the form of Equation 2-2, the goal of the least square regression is to estimate the deterministic fitting parameter θ given N observations of range (x) and offset (y). The fitting parameter θ represents the coefficients of the least squares curve fit to the data contained in the list. Depending on whether a first or second order curve fit is performed, θ will either contain the coefficients of Equation 2-3 or Equation 2-4.

$$y = c_1 x + c_0 \tag{2-3}$$

$$y = c_2 x^2 + c_1 x + c_0 \tag{2-4}$$

The sample mean and sample variance of the correction list data are first determined to decide whether a first or second order curve fit should be performed. For a random variable Y that is normally distributed with mean μ and variance σ^2, the sample mean $\hat{\mu}$ and sample variance $\hat{\sigma}^2$ can be expressed by Equation 2-5 and Equation 2-6, where $y_1 \ldots y_N$ are the N observations of Y.

$$\hat{\mu} = \left(\frac{1}{N}\right)\sum_{i=1}^{N} y_i \tag{2-5}$$

$$\hat{\sigma}^2 = \left(\frac{1}{N-1}\right)\sum_{i=1}^{N} (y_i - \hat{\mu})^2 \tag{2-6}$$

If the sample variance of the correction list data is within a threshold, a first order least squares regression is performed to the data; otherwise a second order regression is performed. The least square regression estimates, as well as the Chi-Squared analysis derived later in the section, are performed using the GNU Scientific Library (GSL) linear algebra toolbox [22].

The symmetric definite weight matrix, given by Equation 2-7, is comprised of the individual weights of each correction point. As described earlier, the weight of each newly inserted correction is equal to the confidence value assigned by the sensor components, which is a decimal value between 0 and 1. The weights are decayed based on both the elapsed time since the correction was new and the current speed of the vehicle. When the weight of a correction falls below a minimum threshold, the correction is discarded from the list.

$$W = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & w_N \end{bmatrix} \tag{2-7}$$

The deterministic parameters for a first order approach are given by Equation 2-8 and Equation 2-9; those for a second order approach are given by Equation 2-10 and Equation 2-11. The general solution to the least squares problem, for a first or second order regression, is given by Equation 2-12.

$$H = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix} \tag{2-8}$$

$$\theta = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} \tag{2-9}$$

$$H = \begin{pmatrix} 1 & x_1 & x^2{}_1 \\ \vdots & \vdots & \vdots \\ 1 & x_N & x^2{}_N \end{pmatrix} \tag{2-10}$$

$$\theta = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} \tag{2-11}$$

$$\left(H^T W H\right)\hat{\theta}^* = H^T W y \tag{2-12}$$

$$\hat{Y} = H\hat{\theta}^* \tag{2-13}$$

The reduced Chi-Squared analysis described in Laub [23] and Taylor [24] is then applied. The estimate of the observation of Y, $\hat{Y}$, is given by Equation 2-13 after the least squares estimate $\hat{\theta}^*$ of the fitting parameter θ is determined. Then the accuracy of the estimate $\hat{Y}$ is found by performing a Chi-Squared analysis. The difference between the predicted observation $\hat{y}_i$ and the actual observations y_i is the residual ε_i. The Chi-Squared statistic, χ, is a function of the residual and is given by Equation 2-14, with σ^2 being the true variance of the correction data.

$$\chi^2 = \sum_{i=1}^{N} w_i \left\{ \frac{1}{\sigma^2} [\varepsilon_i]^2 \right\} \tag{2-14}$$

The reduced Chi-Squared statistic, ϑ, is given by Equation 2-15. The parameter ν is given by Equation 2-16, with N being the number of data points to fit and p being the number of fitting parameters. If the reduced Chi-Squared statistic falls between 0.5 and 1.5, then the residuals are small enough that high confidence can be associated with the least squares regression.

$$\vartheta = \chi^2 / \nu \tag{2-15}$$

$$\nu = N - p - 1 \tag{2-16}$$

The reduced Chi-Squared statistic compares the estimated variance, given in the numerator of Equation 2-14, to the actual variance of the data, given in the denominator of Equation 2-14. The reduced Chi-Squared statistic approximates the ratio of the variance of the data from the best curve fit and the true variance of the data; therefore a value of 1 for the statistic is the optimal result [23][24]. A value that is too low implies that the data variance from the estimated fit is much lower than the true variance of the data, which indicates that the nature of the data is poorly understood or that the weight values of the corrections are too small. A value that is too high indicates that the regression produced a poor fit to the data, since the estimated variance far exceeds the true variance.

Recursive Least Squares Approach

A recursive least squares approach was implemented according to the following methodology and tested against the general least squares approach. Given new data points, a

weighted recursive least squares algorithm can be used to predict the upcoming state of the parameter θ. Assuming white noise error that is uncorrelated with previous errors, an initial condition state estimate is sufficient to perform a recursive least squares analysis. The symmetric positive definite weight matrix W is defined according to the current matrix and the old weight matrix as shown in Equation 2-17, and is given in terms of the state k.

$$W = \begin{bmatrix} W_k & 0 \\ 0 & W_{k+1} \end{bmatrix} \tag{2-17}$$

The choice of weights in the new and old weight matrices is important to the accuracy of the parameter estimation, as it determines how the fit reacts to the dynamically changing incoming data. The weights in this application are the same as those for the non-recursive least squares analysis. The initialization of the recursive least squares approach is based on a batch processing technique as described by Malik [25], with the recursion being delayed until initial conditions are properly estimated. The initial condition for the correction matrix P is found after an initial data gathering period, and is defined by Equation 2-18.

$$P_0 = \left(H_0 W_0 H_0\right)^{-1} \tag{2-18}$$

Since the weight matrices are symmetric positive definite, and assuming that the sampling period is long enough such that H_0 is full column rank, the term $H_0 W_0 H_0$ is also symmetric positive definite and therefore invertible. From this point, the algorithm recursively adapts to the dynamically changing input data, using Equation 2-19 and Equation 2-20.

$$P_{k+1} = P_k - P_k H_{k+1}^T \left(H_{k+1} P_k H_{k+1}^T + W_{k+1}^{-1}\right)^{-1} H_{k+1} P_k \tag{2-19}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + P_{k+1} H_{k+1}^T W_{k+1} \left(Y_{k+1} - H_{k+1} \hat{\theta}_k\right) \tag{2-20}$$

The recursive least squares approach was tested for possible use on the Lane Finder Arbiter, with results of simulated testing yielding similar results to the non-recursive least squares approach. Also, the most recent estimate of the parameter θ, $\hat{\theta}_k$, must be changed at each new time step (as well as the correction list data) according to the transformation procedure previously described. Since the recursive least squares approach yields a similar result to the non-recursive least squares approach and requires more computations, the non-recursive least squares approach was adopted for the Lane Finder Arbiter.

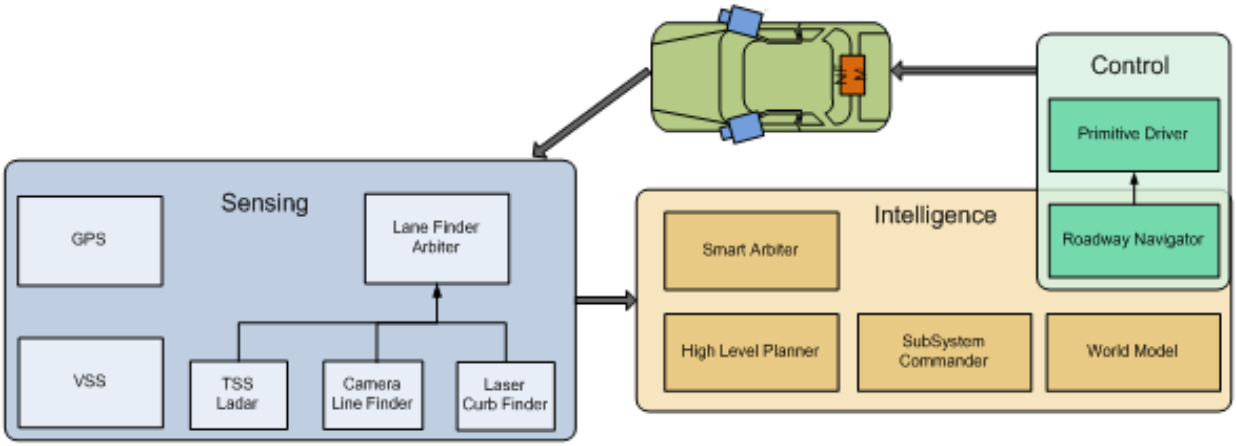

Figure 2-1. Logical diagram of component architecture of the Urban Navigator.

Figure 2-2. Lane Finder Arbiter visualization screen with a confident fit. The white line ahead of the vehicle represents a confident fit as described by the reduced Chi-Squared statistic analysis. Decayed offset data pixels are shown behind the vehicle.

Figure 2-3. Lane Finder Arbiter visualization screen without a confident fit. No first or second order curve fit of the data shown satisfy the criterion for a confident fit, therefore no correction data is communicated with the navigation components.

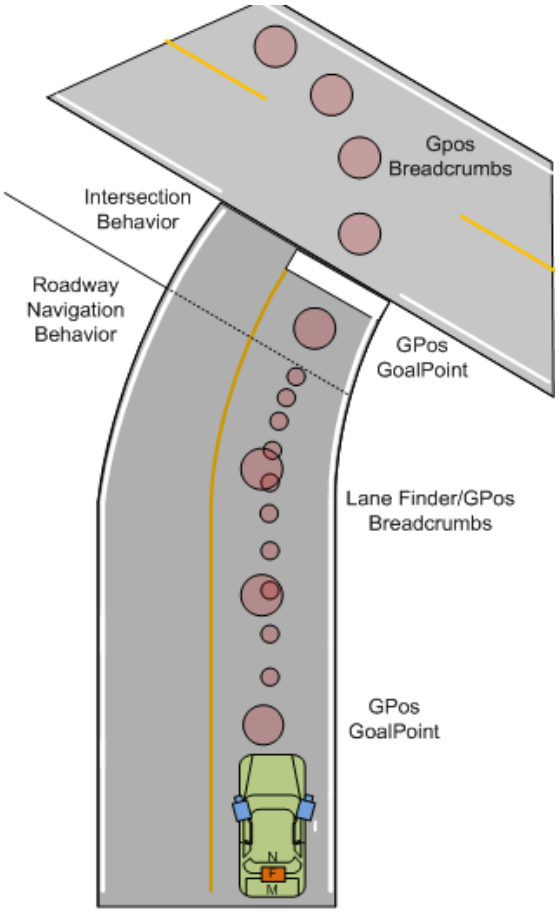

Figure 2-4. Lane Finder and GPos breadcrumb insertion. Depending on the behavior of the vehicle, Lane Finder breadcrumbs are inserted between GPos goalpoints by the Roadway Navigator.

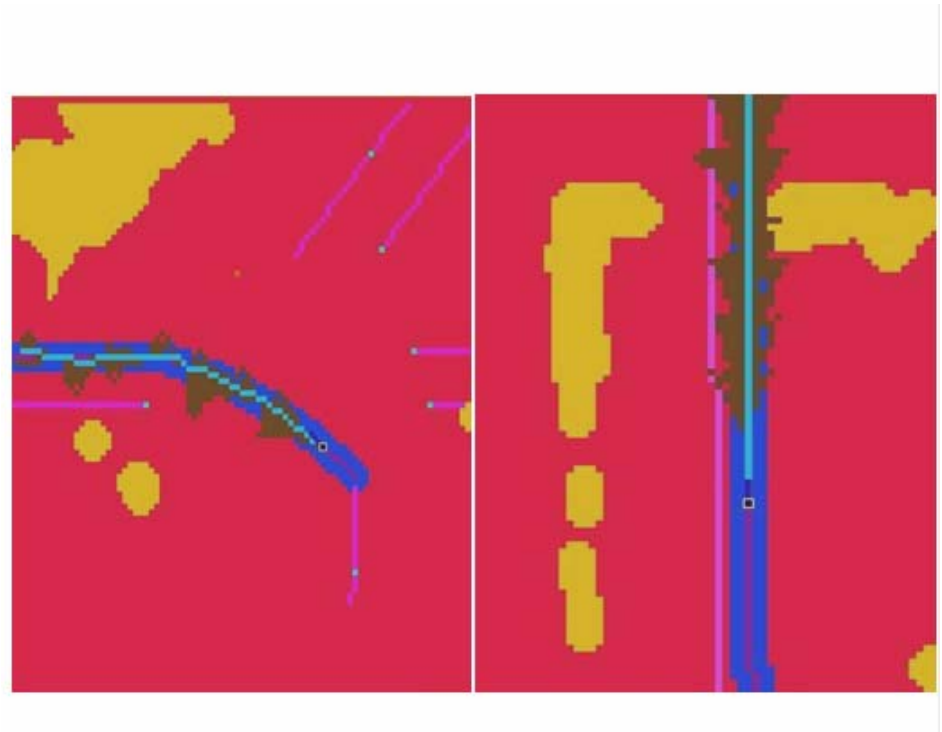

Figure 2-5. Roadway Navigator traversability grid. A* search algorithm nodes are shown in brown, the least cost path is shown as a light green line, and obstacles are shown in orange.

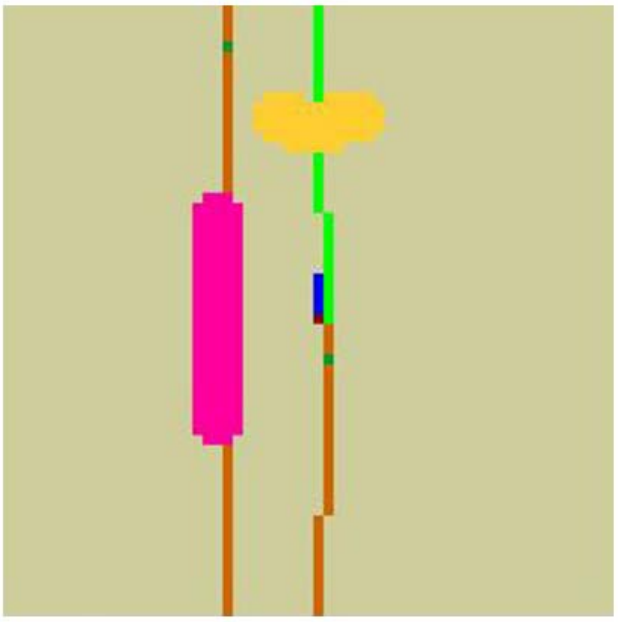

Figure 2-6. Local World Model global reference frame gridmap. A stationary obstacle is shown in yellow, and a moving obstacle is shown in pink.

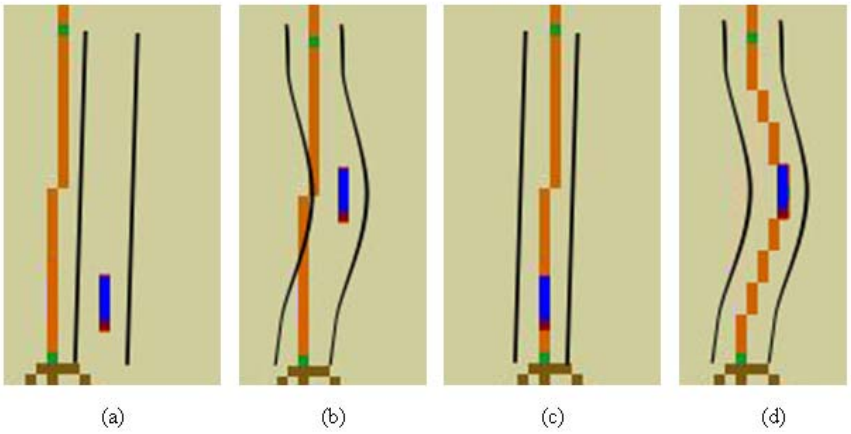

Figure 2-7. Local World Model resolution of discrepancy between Lane Finder Arbiter data and GPos data. Two scenarios are presented. A) The Lane Finder Arbiter perceived lane, shown by black lines, disagrees consistently with the GPos perceived lane. B) The Lane Finder Arbiter lane disagrees temporarily with the shape of the GPos lane. C) A global map shift causes the new GPos lane to agree with the Lane Finder lane. D) A temporary GPos lane shift is performed, but the global map is not shifted.

CHAPTER 3
RESULTS

Static Testing

Sensors provide important and accurate traversability information to the navigation components, and in this respect they are involved in vehicle navigation. The traversability information, however, is used by the Roadway Navigator to determine where not to drive, which is less valuable for navigation than information determining where to drive. The traversability information is nonetheless important to the vehicle architecture, and has the added benefit of failing safe. False positive results of a model that determines where the vehicle should not drive will keep the vehicle at a stop; while those of a model determining where the vehicle should drive might cause an unsafe dynamic behavior. Therefore, when using models to characterize where to drive in a roadway, thorough testing must be performed to ensure that there are no false positives.

In the case of uncertainty in the curve fit of incoming correction data, the Lane Finder Arbiter will not send a correction report to the navigation components. Instead, the outgoing messages will communicate the fact that there is no confidence in the curve fit of the current data. The sensor components also look at their own lane center estimates and decide whether or not they are confident enough to provide input data to the Lane Finder Arbiter. In this respect, there are two layers of safeguarding against false positives; though testing must be performed to confirm the effectiveness of the system.

Figure 3-1 – Figure 3-3 show the results of a series of static tests of each sensor component output, with the vehicle parked in a road lane. The results shown are those of the roadway characterization models, with the measured data being the actual inputs of the Lane Finder Arbiter. As expected, the camera correction data has a higher variance than the ladar correction

data. The variance of the camera data, however, is low enough for autonomous navigation purposes to be considered as reliable as ladar data. Therefore, the confidences in the camera and laser sensors remain equal in the Lane Finder Arbiter. If the variance of the camera output data was sufficiently large (> 0.25 m), then an additional sensor component confidence value would be implemented within the Lane Finder Arbiter. The total ranges of all the data sets are less than a half meter, which is an improvement on the potential 3 meter inaccuracy of GPS. The true offset values ahead of the vehicle are difficult to precisely measure, since the offset measurements are based on the projection of the centerline of the vehicle at exact distances ahead of the vehicle. Testing showed a bias of less than six inches for the laser Curb Finder component and less than a foot for the camera Line Finding component.

Testing also yielded the desirable result that no obvious false positives were sent to the Lane Finder Arbiter; which would be indicated by large outliers in the sensor data. The models did, however, fail to identify the lane center 100% of the time, with an identification rate of 86% for the camera and at least 90 % for the ladar sensing components.

Dynamic Testing

A Lane Finder Arbiter simulator was created to test the Lane Finder architecture and evaluating the effectiveness of the curve fitting algorithms. Sensor data was collected and synchronized with high precision GPos data, and this data was used to generate a simulated environment in which the Lane Finder Arbiter was tested. The complete messaging system in the Lane Finder architecture was included in the simulation, as the Roadway Navigator and Local World Model received correction messages from the Lane Finder Arbiter. After the Lane Finder Arbiter's curve fitting algorithm was completed, data was gathered to determine the

accuracy of the Lane Finder Arbiter offset values with respect to high precision GPos offset values. The results of a dynamic test of the Lane Finder Arbiter are shown in Figure 3-4.

The test run shown in Figure 3-4 consisted of a human driver driving the Urban Navigator while Lane Finder Arbiter offset data and GPos data were recorded. During the run, the driver intentionally swerved while remaining within the lane; this is seen near the midpoint of the data set shown in Figure 3-4. The GPos data was recorded while the GPos system was in High Precision mode, whose RMS error is within one meter. This reliable GPos data can therefore be used to test the validity of the Lane Finder Arbiter correction data.

The GPos data of the roadway has previously been surveyed, and the result is shown in Figure 3-5. Before the Urban Navigator can become fully operational, even during simulation, a pre-defined global map of the roadway must be provided to the mission route planning component, the High Level Planner. Therefore, the center of each lane on the roadway has been surveyed with high precision in the global coordinate frame. The logged GPos coordinates and the surveyed GPos coordinates are transformed from latitude / longitude form to Universal Transverse Mercator (UTM) form, which is given in meters. The vehicle offsets as measured by GPos can then be measured and compared to the offsets measured by the Lane Finder Arbiter. Since the Lane Finder Arbiter offset data is independent of the vehicle's global position, such dynamic tests can test the accuracy of the Lane Finder Arbiter output corrections compared to high precision GPos corrections. The results of the dynamic testing show that the correction differences between the GPos measurements and the Lane Finder Arbiter's measurements were within 0.5 meters. These results indicate that the Lane Finder Arbiter has achieved the goal of providing accurate correction information for vehicle navigation purposes.

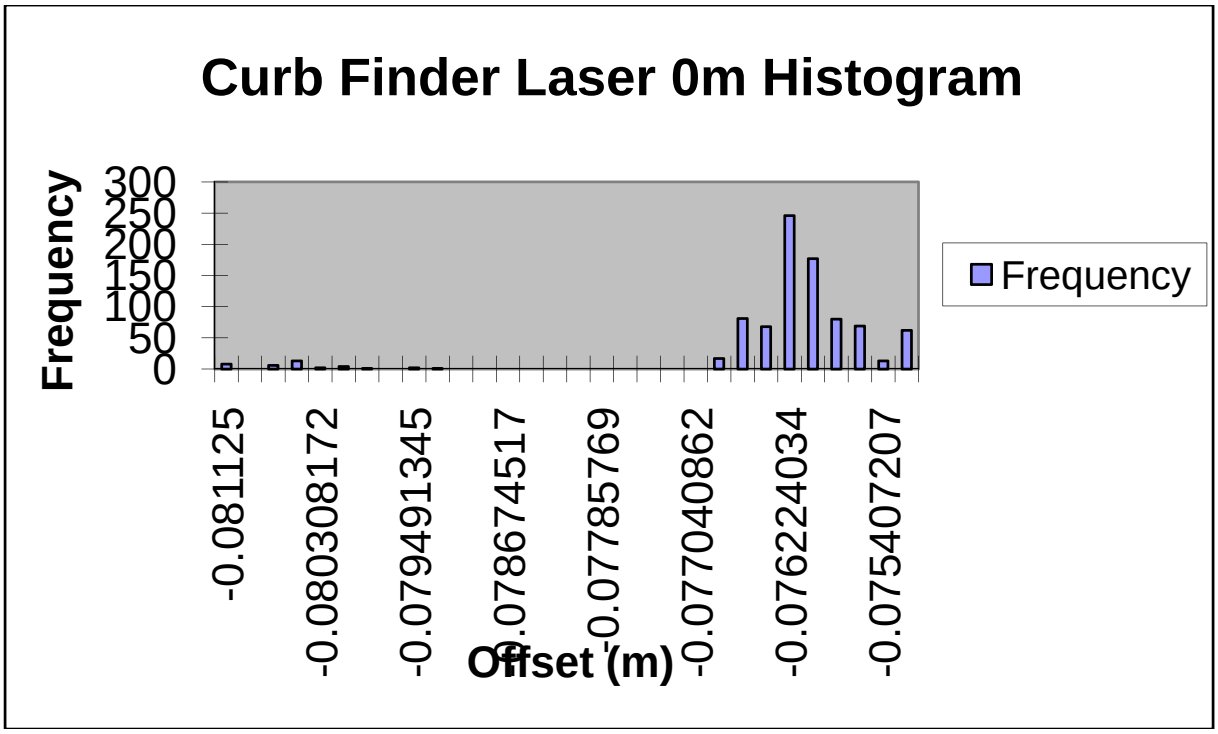

Figure 3-1. Histogram of curb offset values during a static test. The range is 0.006 m, with 850 total measurements.

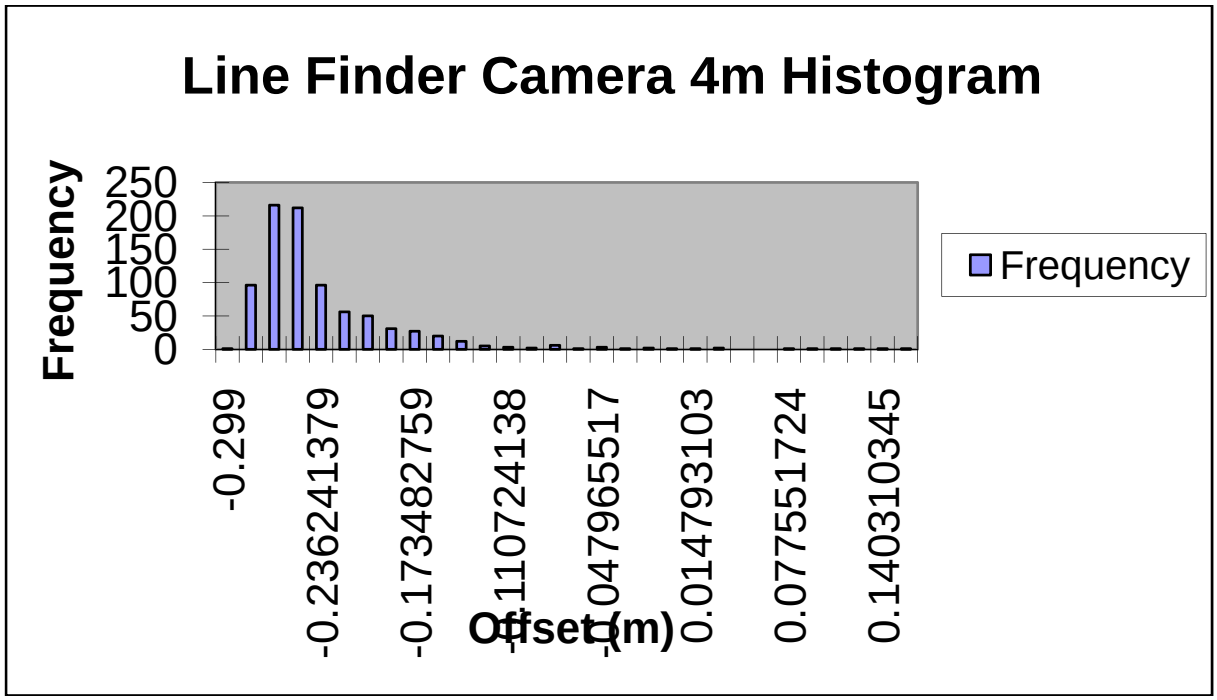

Figure 3-2. Histogram of line offset values during a static test. The range is 0.439 m, with 850 total measurements.

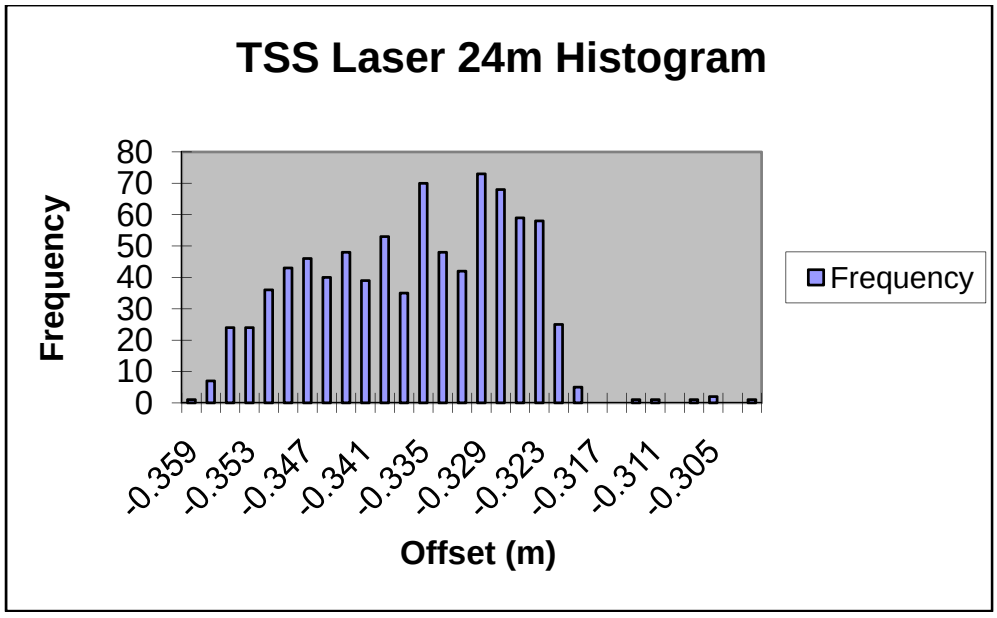

Figure 3-3. Histogram of long range curb offset values during a static test. The range is 0.056m, with 850 total measurements.

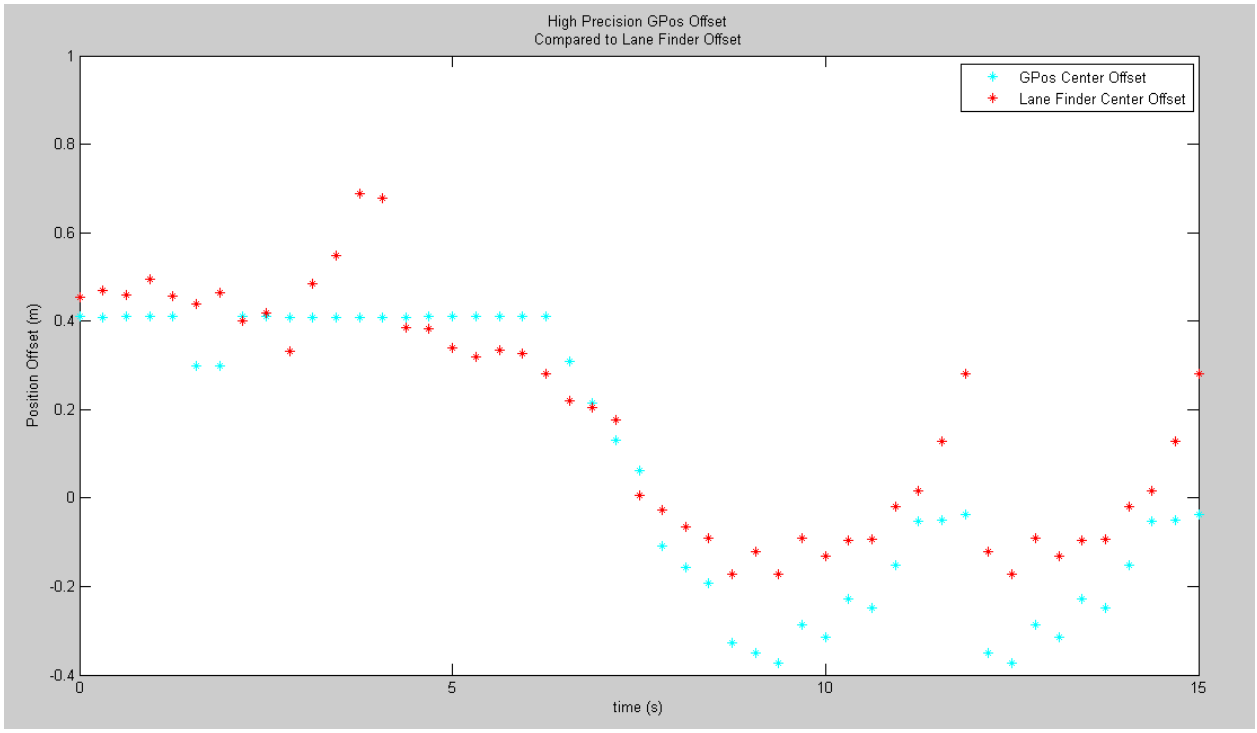

Figure 3-4. Results of dynamic testing with high precision GPos correction data and Lane Finder Arbiter lane correction data.

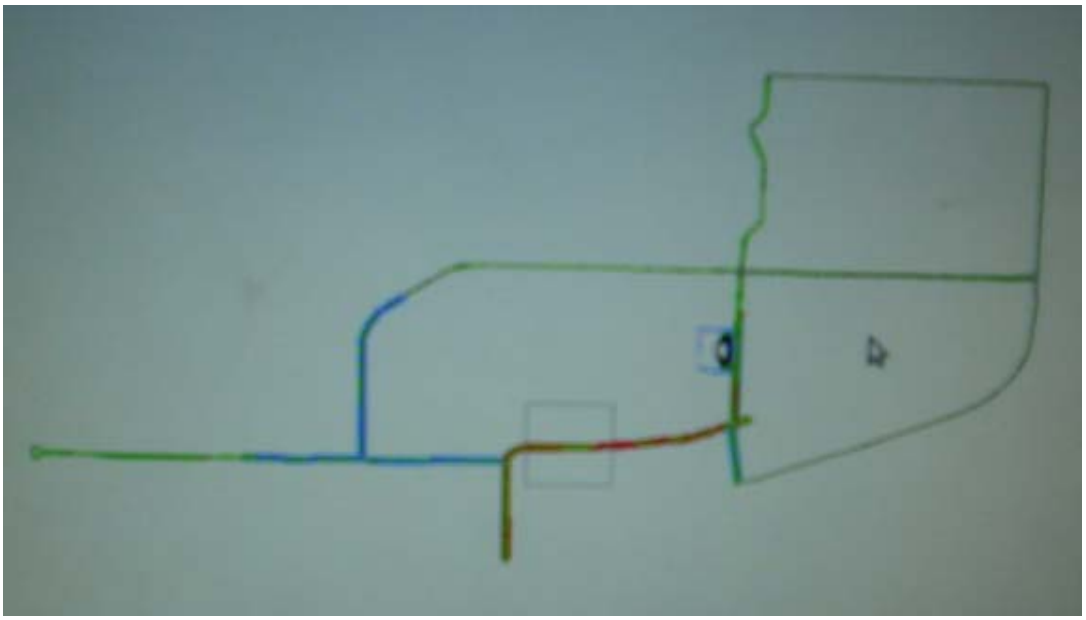

Figure 3-5. High Level Planner map. Vehicle is shown within a box on the red segment of a previously surveyed roadway. This surveying process provides high precision GPos reference points for the center of each lane in the roadway.

CHAPTER 4
CONCLUSION

The Lane Finder Arbiter provides a portable software component which can interpret local sensor data and use the data to aid in autonomous vehicle navigation. The new navigation architecture uses GPos data points for general navigation purposes and the Lane Finder Arbiter correction points for instantaneous navigation. Ideally, the vehicle would even be able to drive through a roadway where there is no GPS at all, using the local sensors to safely drive the vehicle.

An advantage of the new architecture is that multiple sensors must fail for the vehicle to be rendered useless, and in the absence of Lane Finder correction data the vehicle can still drive using traditional GPos waypoint navigation methods. Another advantage of this architecture is that the same data from the local sensors is used in multiple ways on the vehicle, making the vehicle itself more efficient. The only requirements on the Lane Finder architecture are that robust characterizations of the roadway are made at the sensor level, and that adequate data processing techniques take place at the sensor level to determine confidence.

More complex filtration techniques, including predictive Kalman-based techniques, represent an alternative solution approach that requires further research and testing. Traditional navigation systems could also be improved with the incorporation of a neural network, which would store and recall static objects along the roadway. Therefore the vehicle could localize its position based on surrounding landmarks as humans sometimes do, or it could choose optimized navigation routes based on previous knowledge of the local environment. This is another valuable research goal that, if achieved, would greatly improve the intelligence and capabilities of autonomous vehicles.

Finally, while an idealized goal, the achievement of human-like driving performance is a valid metric from which to measure the progress of autonomous vehicle development. The traditional navigation scheme is similar to a driver who drives through streets looking only at his GPS system, with a passenger whose only purpose is to alert the driver of any oncoming obstacles. Even with advanced GPos systems, which are used on most autonomous vehicles, this driving style is not as reactive as one which uses local sensors to navigate the surroundings. If the design goal of autonomous vehicles remains to emulate human driving performance while eliminating the unpredictability of human drivers, the most natural navigation system would utilize a local system such as the Lane Finder Arbiter.

LIST OF REFERENCES

1. Caron, F., Duflos, E., Pomorski, D., and Vanheeghe, P., "GPS/IMU data fusion using multisensor Kalman filtering: Introduction of contextual aspects", *Informative Fusion*, Vol. 7, Issue 2, June 2006, pp. 221-230.

2. Bell, J., and Stol, K., "Tuning a GPS/IMU Kalman Filter for a Robot Driver", *Proceedings, Australian Conference on Robotics and Automation*, Auckland, New Zealand, 6-8 December 2006.

3. Wöbner, M., "Sources of error in GPS," *Kowoma*, Schweiz, Germany, 2004, *http://www.kowoma.de/en/gps/errors.htm#top.*

4. Allred, B., Daniels, J., and Ehsani, M. R., "Handbook of Agricultural Geophysics", *CRC Press*, Boca Raton, FL, 2008.

5. "Global Positioning System Wide Area Augmentation System (WAAS) Performance Standard", *Federal Aviation Administration and Department of Transportation*, Washington, DC, 31 October 2008.

6. Rife, J., Pullen, S., Enge, P., and Pervan, B., "Paired Overbounding for Nonideal LAAS and WAAS Error Distributions", *IEEE Transactions on Aerosapce and Electronic Systems*, Vol. 42, Issue 4, October 2006.

7. Baltzakis, H., and Trahanias, P., "A Hybrid Framework for Mobile Robot Localization: Formulation Using Switching State-Space Models", *Autonomous Robots*, Vol.15, Issue 2, New York, New York, September 2003, pp. 169-191.

8. Baltzakis, H., Argyros, A., and Trahanias, P., "Fusion of laser and visual data for robot motion planning and collision avoidance". *Machine Vision and Applications*, Vol. 15, Secaucus, NJ, December 2003, pp.92-100.

9. Thorpe, C., Carlson, J., Duggins, D., Gowdy, J., Maclachlan, R., Mertz, C., Suppe, A., and Wang, B., "Safe Robot Driving in Cluttered Environments", *IEEE Proceedings, 11th International Symposium of Robotics Research*, Siena, Italy, 19-22 October 2003.

10. Thorpe, C., Clatz, O., Duggins, D., Gowdy, J., Maclachlan, R., Ryan, J., Mertz, C., Siegel, M., Wang, C., and Yata, T., "Dependable Perception for Robots", *IEEE Proceedings, International Advanced Robotics Programme*, Seoul, South Korea, 21-22 May 2001.

11. Lalonde, J.F., Vandapel, N., Huber, D., and Hebert, M., "Natural terrain classification using three-dimensional ladar data for ground robot mobility", *Journal of Field Robotics*, Vol. 23, Issue 10, Walden, Massachusetts, October 2006, pp.839-861.

12. Riisgard, S., and Blas, M.R., "A tutorial approach to Simultaneous Localization and Mapping", *Massachusetts Institute of Technology*, Cambridge, MA, 2005, http://ocw.mit.edu/NR/rdonlyres/Aeronautics-and-Astronautics/16-412JSpring-2005/9D8DB59F-24EC-4B75-BA7A-F0916BAB2440/0/1aslam_blas_repo.pdf.

13. Thrun, S.; Burgard, W.; Fox, D. "Probabilistic Robotics," *MIT Press*, Cambridge, MA, 2005.

14. Manduchi, R., Castano, A., Talukder, A., and Matthies, L., "Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation", *Autonomous Robots*, Vol.18, New York, New York, 2005, pp. 81-102.

15. Velat, S., Lee, J., Johnson, N, and Crane, C., "Vision Based Vehicle Localization for Autonomous Navigation", *IEEE Proceedings, International Symposium of Computational Intelligence in Robotics*, Jacksonville, FL, 20–23 June 2007, pp. 528-533.

16. Canny, J., "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, Issue 6, Washington, DC, Nov 1986, pp. 679-698.

17. Yu, B., Jain, A., "Lane Boundary Detection Using a Multiresolution Hough Transform", *IEEE Proceedings, International Conference on Image Processing*, Vol. 2, Santa Barbara, California, 26-29 October1997, pp. 748-751.

18. Nedevchi, S., Danescu, R., Frentiu, D., Marita, T., Oniga, F., Pocol, C., Schmidt, R., and Graf, T., "High Accuracy Stereo Vision System for Far Distance Obstacle Detection", *IEEE Proceedings, Intelligent Vehicles Symposium*, Parma, Italy, 14-17 June 2004, pp.292-297.

19. Williamson, T., "A High-Performance Stereo Vision System for Obstacle Detection", *Robotics Institute, Carnegie Mellon University*, Pittsburgh, Pennsylvania, September 1998.

20. Stentz, A., Kelly, A., Rander, P., Herman, H., Amidi, O., Mandelbaum, R., Salgian, G., Pederson, J., "Real-time, multi-perspective perception for unmanned ground vehicles", *AUVSI Proceedings, Unmanned Systems Symposium*, July 2003.

21. Crane, C. and Duffy, J. "Kinematic analysis of robot manipulators," *Cambridge University Press*, New York, New York, 1998.

22. Gough, B. "GNU Scientific Library Reference Manual – 2nd Edition", *Network Theory Ltd.*, Bristol, United Kingdom, 2003.

23. Laub, C., and Kuhl, T., "How Bad is Good? A critical look at the fitting of reflectivity models using the reduced chi-square statistic", *Science Poster Abstracts, SNS-HFIR User Meeting*, Oak Ridge, TN, 11-13 October 2005.

24. Taylor, J. R., “An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements”, *University Science Books*, Herndon Virginia, 1997.

25. Malik, M. B., “State-Space Recursive Least-Squares”, *National University of Science and Technology*, Rawalpindi, Pakistan, 2004.

BIOGRAPHICAL SKETCH

Philip R. Osteen was born in Gainesville, FL, in 1984. He graduated from the University of Florida in 2006 with a BS in aerospace engineering, and will receive a MS in mechanical engineering in 2009. He joined the Center for Intelligent Machines and Robotics at the University of Florida his senior undergraduate year, and has worked as a graduate student since. He is a member of the American Association for the Advancement of Science and of Tau Beta Pi engineering honor society, and he participated in the 2007 DARPA Urban Challenge as a member of Team Gator Nation.