Auburn University - FHWA

Intelligent Multi-Sensor Measurements to Enhance Vehicle Navigation and Safety Systems

Final Report

March 28, 2011

John Allen
allenjw@auburn.edu

Jordan Britt
brittjh@auburn.edu

Christopher Rose
rosechr@auburn.edu

(334) 844-3267

David Bevly
bevlydm@auburn.edu

Table of Contents

List of Figures

1 Introduction

Over 19,000 people died on the United States' highways due to road departures in 2008. Safety systems on vehicles, such as lane departure warning systems, can reduce the number of these deaths by warning the driver of a lane departure. These safety systems determine the lateral distance to the lane markings and warn the driver when the vehicle has left or is about to leave the lane using a single sensor such as a camera. However, current lane departure warning systems can fail due to varying conditions and environments. A multi-faceted approach using multiple sensors provides a more robust solution when solutions are not available from a single sensor. This final report describes the results of the Intelligent Multi-Sensor Measurements to Enhance Vehicle Navigation and Safety Systems project.

The lateral localization system of this project uses the following four sensors: Global Positioning System (GPS), inertial measurement unit (IMU), LiDAR (Light Detection and Ranging), and a camera. GPS, LiDAR, and the camera serve to correct the drift of the IMU using an Extended Kalman Filter. Since the LiDAR and camera both measure lateral distance with respect to the lane markings, a map of the lane markings is needed to relate the camera and LiDAR measurement to the global coordinate frame of the GPS. The map of the lane markings was made at the National Center for Asphalt Technology (NCAT) test track, and all experimental tests were conducted on this track.

Section 2 describes the camera system for determining lateral distance and provides experimental results. Section 3 explains the lateral distance algorithm using the LiDAR and shows the experimental results. Section 4 presents an overview of the integrated positioning system which uses the measurements from the camera and LiDAR to correct the inertial navigation system and provide measurements in the absence of GPS. Appendix A shows the Gantt chart for the project, Appendix B describes the procedure for determining truth data, and Appendix C describes the procedure for coordinate frame rotation and translation.

a. Publication List

[1] Allen, J. and Bevly D. "Use of Vision Sensors and Lane Maps to Aid GPS/INS under a Limited GPS Satellite Constellation." In *Proceedings of the ION GNSS*, 2009.

[2] Allen, J. and Bevly D. "Relating Local Vision Measurements to Global Navigation Satellite Systems Using Waypoint Based Maps." *IEEE PLANS*, 2010.

[3] Allen, J., Britt, J., Rose, C., and Bevly, D. "Intelligent Multi-Sensor Measurements to Enhance Vehicle Navigation and Safety Systems." In *Proceedings of the ION Technical Meeting, 2009.*

[4] Britt, J. and Bevly, D. “LiDAR Calibration Method for Vehicle Safety Systems”, *SAE World Congress 2010*, April 2010.

[5] Britt, J. and Bevly, D. “Lane Tracking using Multilayer Laser Scanner to Enhance Vehicle Navigation and Safety Systems”, *The Institute of Navigation 2009 International Technical Meeting*, January 2009.

[6] Britt, J., Broderick, D., and Hung, J. “LiDAR Attitude Estimation for Vehicle Safety Systems.” *IEEE PLANS*, 2010.

[7] Rose, C. and Bevly, D. “Camera and Inertial Measurement Unit Sensor Fusion for Lane Detection and Tracking using Polynomial Bounding Curves.” In *Proceedings of the ION GNSS*, 2009.

[8] Rose, C. and Bevly, D. “Vehicle Lane Position Estimation with Camera Vision using Bounded Polynomial Interpolated Lines.” In *Proceedings of the ION Technical Meeting, 2009*.

b. Thesis and Dissertations

Research performed in the area of sensor integration with the support of FHWA resulted in the following:

Thesis – Christopher Rose, *Lane Level Localization with Camera and Inertial Measurement Unit using an Extended Kalman Filter*, 2010

Thesis – Jordan Britt, *Lane Detection Calibration and Attitude Determination with a Multilayer LiDAR for Vehicle Safety Systems*, 2010

Thesis – John Allen, in progress expected completion May 2011

2 Camera Lane Detection

This section presents the algorithms for determining lateral lane position in the lane using a camera. The vision only algorithm for lane detection is discussed first, followed by the vision / IMU algorithm for lateral distance estimation. It is worth noting that the camera measurements sent to the integrated solution for the final project uses the vision only algorithm.

a. Vision-only Algorithm

The figure below shows a block diagram of the vision system. The system begins with an image, moves into thresholding and edge detection which is followed by the Hough transform for line extraction. Following this, lines from the line pools are selected and combined into point pools, which are then used for least squares interpolation to get the coefficients of the 2nd order polynomial model. The coefficients are then used as measurements in a Kalman filter. The coefficients states of the Kalman filter can be used to calculated the lateral distance and heading of the system, as well as for line selection.

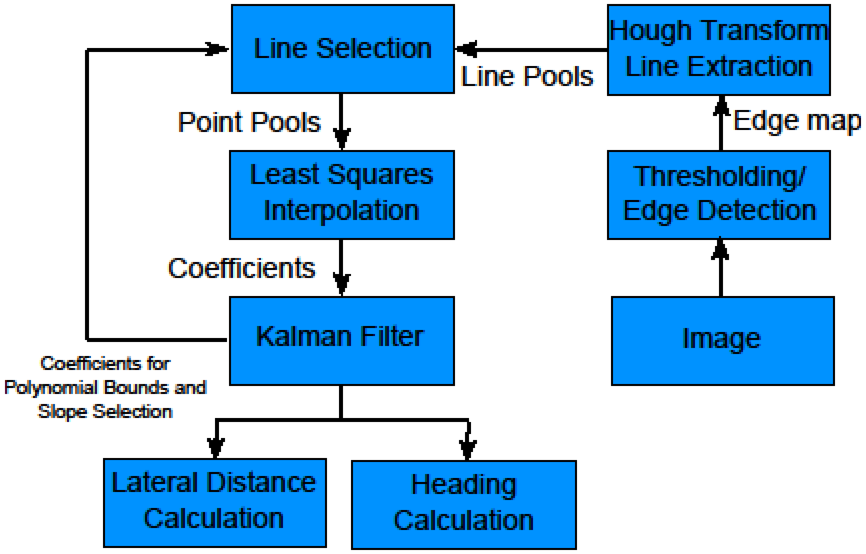
Line Selection
Hough Transform Line Extraction
Line Pools
Point Pools
Edge map
Least Squares Interpolation
Thresholding/ Edge Detection
Coefficients
Kalman Filter
Image
Coefficients for Polynomial Bounds and Slope Selection
Lateral Distance Calculation
Heading Calculation

Figure 1: Overview of Vision System

Figure 1 shows the overview of the vision system. After receiving an image from the camera, several image processing steps are conducted to detect the edges formed by the lane markings in the image. Two selection criteria are used to select which lines are valid, and each line's endpoints and midpoint is grouped into a point pool. Each lane is represented by a 2^{nd} order polynomial, and each point pool is used in least squares polynomial interpolation to find the coefficients of the 2^{nd} order polynomial lane model. Finally, each coefficient is added into a linear Kalman filter as measurements in order to reduce the effect of any erroneous lane marking estimates. The following sections delve into each step in greater detail.

1. Image Processing

The vision algorithm begins with image processing on the image obtained from the camera.

Thresholding

The threshold operation must extract the lane markings from the image and ignore extraneous features on the road. Since vehicles drive in varying environments, the threshold operation must take into account different levels of lighting and weather to extract the lane markings. Figure 2 shows the original image of the road as well as the extracted lane marking following dynamic thresholding. The lane markings are clearly visible in the thresholded image.

The figure below shows the original image at day (left) and the dynamically thresholded image (right). The dynamically thresholded image is a binary image and consists of a black background and the white lane markings, which shows the effectiveness of the lane marking extraction.

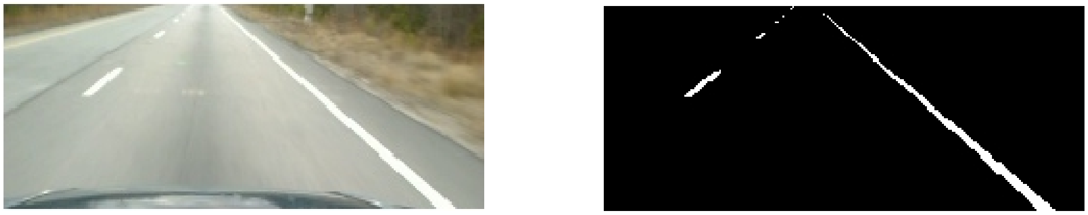

Figure 2: Original Image and Dynamically Thresholded Image

To further emphasize the need for a dynamic threshold, Figure 3 shows an image of the road at dusk and the resulting thresholded image using a constant threshold that works well for day environments. The thresholded image detects a very small portion of the lane marking.

The figure below shows the original image at twilight (left) and the constant thresholded image (right). The binary constant thresholded image only shows a sliver of the right lane marking that has been extracted, which shows the failure of the constant threshold in varying environments.

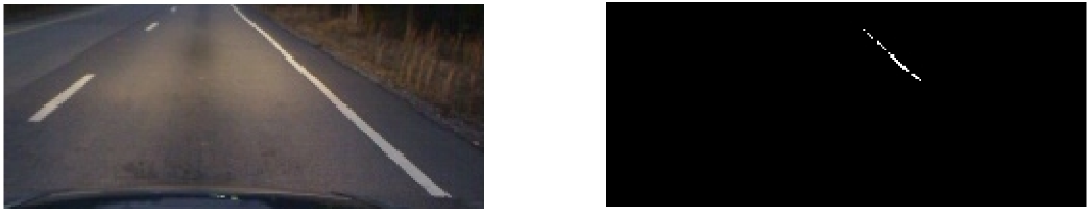

Figure 3: Darker Image and the Thresholded Image with Constant Threshold

The dynamic threshold, however, takes into account the statistics of the image to better approximate the best value for thresholding. Using the mean and standard deviation of the image, the dynamic threshold is calculated as follows:

$$T = \mu + K\sigma$$

where T is the grayscale threshold value, μ is the mean of the grayscale image, K is a measure of the noise in the system, and σ is the standard deviation of the image. Using K = 1.5, the resulting thresholded image is shown in Figure 4. The extracted lane markings are clearly visible.

The figure below shows the darker image once more (left) but with a dynamically thresholded image (right). The lane markings are clearly evident in the binary image on the right and shows the effectiveness of the dynamic threshold.

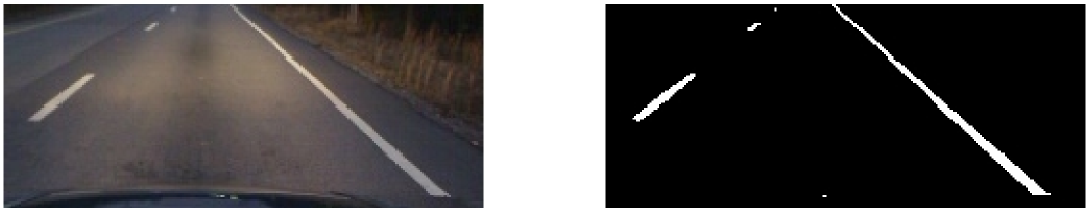

Figure 4: Darker Image and the Thresholded Image with Dynamic Threshold

Edge Detection and the Hough Transform

The edges of the thresholded image are extracted using Canny edge detection, as shown in Figure 5. The edges are broken in various places along the lane marking's edge. The Hough transform extracts the lines from the edge map while ignoring the gaps in the lines. Each line extracted from the Hough transform is considered either a left or right lane marking depending on its slope. These two groups of lines, so-called line pools, are independently processed. The following sections describe the processing on these line pools.

The figure below shows the daylight original image (left) and the edge detected image (right). The edge detected image appears as an outline of the binary thresholded image.

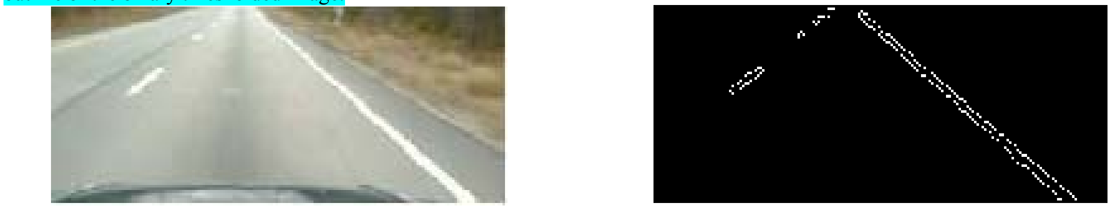

Figure 5: Canny Edge Detection

2. Line Selection

Due to erroneous features in the image such as shadows or objects on the road, extracted lines from the Hough transform may not be lines from lane markings. Two criteria must be met before a line is considered a line from a lane marking edge.

- Spatial criterion
- Slope criterion

Figure 6 shows lane marking lines that are considered to be lane marking lines (blue) and those that do not meet both criteria (red).

The figure below shows an image of the road and the extracted Hough lines. Shadows pass over the road and create erroneous lines (red). Valid lines are shown in blue.

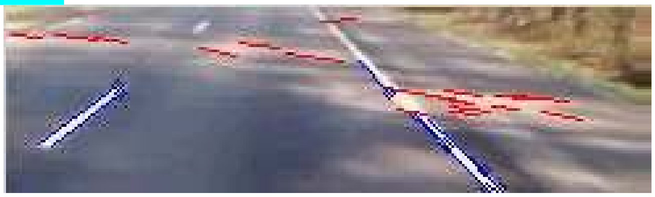

Figure 6: Lines from the Hough Transform- pass (blue) and fail (red)

The spatial criterion requires the Hough lines to be close to the last estimated 2nd order polynomial lane marking model. To achieve this, two additional 2nd order polynomial lines are setup equidistant from the last estimated 2nd order polynomial on both sides to serve as bounding curves. Hough lines outside of these bounds fail, while those that are fully within the bounds meet the criteria. These so-called polynomial bounding curves are calculated using the last estimated 2nd order polynomial model's coefficients to find three points equidistant from the last estimated 2nd order polynomial model. These three points are then used in 2nd order polynomial interpolation to extract the coefficients of the bounding lines. The endpoints of each Hough line are checked for being in between the bounding curves using the coefficients of the polynomial. The three points for the polynomial bounding curves are calculated as follows:

$$x_{rb} = x_{est} + r\sin(\tan^{-1}(2ax_{est} + b))$$
$$y_{rb} = y_{est} - r\cos(\tan^{-1}(2ax_{est} + b))$$

$$x_{lb} = x_{est} + r\sin(\tan^{-1}(2ax_{est} + b))$$
$$y_{lb} = y_{est} - r\cos(\tan^{-1}(2ax_{est} + b))$$

where x_{rb} is the x coordinate of the right bound, x_{est} is the x coordinate on the last estimated lane marking model, y_{rb} is the y coordinate of the right bound, y_{est} is the y coordinate on the last estimated lane marking model, r is the desired distance, a is the x^2 coefficient of the model, and b is the x coefficient of the model. Figure 7 shows the polynomial boundary curves (black) around the last estimated lane model (green). The valid blue Hough lines can faintly be seen around the green estimated lane model, and the rejected lines can be seen along the edge of the grass as red lines.

The figure below shows the road on a highway. Over the left and right lane markings are green lines, which represent the estimated lane marking model. On either side of the green lines are black lines, which represents the polynomial boundary lines. Along the outer edge of the green lines are blue lines, which are valid Hough lines, and yellow lines, which is the measured lane marking model prior to filtering. Red lines along the edge of the road are rejected Hough lines.

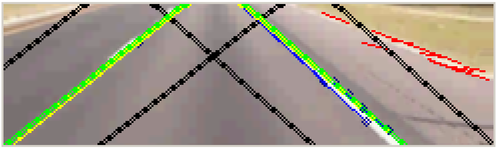

Figure 7: Polynomial Boundary Curves (black) around estimated lane model (green)

The slope criterion requires the line to have a slope approximately equal to the slope of the last estimated 2nd order polynomial lane marking model at the same horizontal position in the image. The slope of the lines from frame to frame should not change drastically, so it is expected that the slopes of the lane markings should be close to the slope of the 2nd order polynomial curve lane marking model near that lane marking line.

3. Lane Modeling and Kalman Filter

Each Hough line that passes both criteria have their endpoints and midpoints collected in point pools. These point pools then undergo least squares polynomial interpolation to determine the coefficients of the polynomial model for the lane marking. Despite the criteria, erroneous Hough lines may be considered valid lane marking edges and the resulting polynomial may not align with the lane markings seen in the image. To reduce the impact of these erroneous model estimates, the coefficients of each lane marking line are used as measurements into a linear Kalman filter. The Kalman filter consists of the following states:

$$\hat{x} = \begin{bmatrix} a_L & b_L & c_L & a_R & b_R & c_R \end{bmatrix}$$

The measurement update corrects each model depending on whether the image determines the presence of the left, right, or both lane marking models. The time update has no effect on the states.

4. Lateral Distance and Heading Calculations

Lateral Distance Calculation

With the estimate of the polynomial lane marking model, the lateral distance can be calculated. Using the coefficients of the polynomial lane model, the lateral lane distance can be calculated by employing known information from the lane markings. The following equations use the general form of the quadratic equation to determine the point on the lane model polynomial with respect to any height within the image.

$$d_r = n\left(\frac{-b+\sqrt{4ay+b^2-4ac}}{2a}\right)$$

$$d_l = n\left(\frac{-b-\sqrt{4ay+b^2-4ac}}{2a}\right)$$

where d_r is the distance to the right lane marking, n a conversion factor, a, b, and c are the coefficients of the lane marking models. The conversion factor n is predetermined by measuring the number of pixels in an image that spans the bottom row of the image from the left lane marking to the right lane marking. The bottom row of pixels is chosen as the row to measure the distance to either lane (y) since it has the most resolution with respect to the road. Under the assumption of the camera pinhole model with no skew and no camera distortion, each pixel in the row has approximately the same actual distance corresponding to its width. Figure 8 shows the row selection where the number of pixels were counted.

The figure below shows the full image of the camera image, including the hood and the sky. The measured row is labeled as a black line, and a label states the typical width of a lane. The width of the black line signifies the number of pixels to include in the conversion factor calculation.

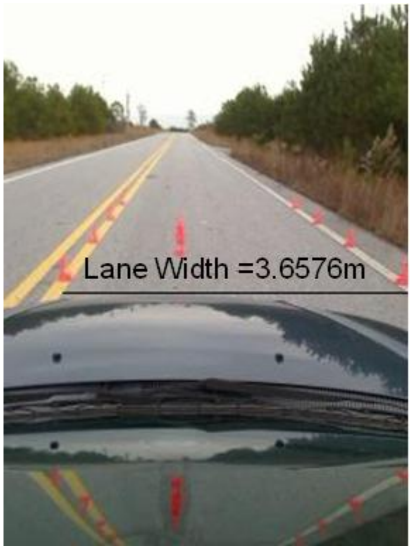

Figure 8: Conversion Factor Calculation

The hood of the vehicle is cropped from the image, and the black horizontal line becomes the lowest row in the image. Since the width of a lane is approximately 3.658 meters, the conversion factor is calculated as follows:

$$n = \frac{w_l}{p_c}$$

where w_l is the standard width of a lane or 3.658 meters and p_c is the pixel count of the lane.

Heading Calculation

The heading is determined from the camera based on the vanishing point of the measured lane markings and the vanishing point of the camera. The vanishing point of the measured lane markings is the point in the image where the lane markings meet given a straight road. Parallel lines, such as lane markings, converge to the vanishing point at the horizon. The vanishing point of the image is the location in the image where the camera is perceived to point and is constant across each image given no change in pitch. When a vehicle has a non-zero heading in the road, the vanishing point of the camera is offset from the vanishing point of the image. The amount of offset can be used to determine the heading in the image. The equation for heading is as follows:

$$\psi = \tan^{-1}\left(OP \frac{\tan\frac{\theta}{2}}{OP_2} \right)$$

where OP is the horizontal distance in pixels from the center point of the image to the measured vanishing point of the lane markings, θ is the visual angle of the camera, OP_2 is the horizontal distance from the center point of the image to the edge of the image, and ψ is the heading angle. Point O is the vanishing point of the image, and point P is the vanishing point for the lane markings, as shown in Figure 9.

The figure below shows a drawing of the road ahead for heading calculation. The lane marking can be seen along with the horizon. Vertical lines are drawn at the vanishing points of the lane marking and of the image (image center). The line formed on the horizon line by the image vanishing point and the lane marking vanishing point is labeled OP. The line formed by the image vanishing point and the end of the image is labeled OP2.

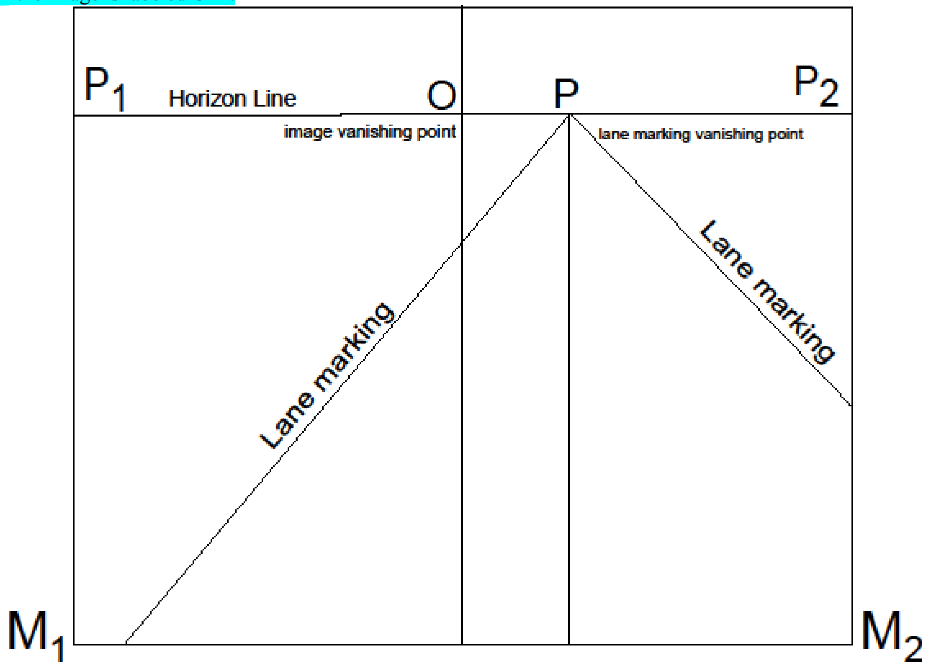

Figure 9: Heading Calculation

One drawback of the heading measurement is the requirement that the lane marking lines be fairly straight. Since the measurement relies on the assumption that the vanishing point of the lane markings corresponds to a point on the road axis with which to base the measurement of the heading, a curved road will shift the vanishing point of the lane marking away from the road frame. Future work is required to modify the algorithm to determine the heading on a curved road.

5. Experimental Results

The experimental results from the vision algorithm were conducted to determine the accuracy of the lateral distance and heading measurements from the vision algorithm. Visual inspection can reveal the validity of the coefficient state measurements and its accuracy.

Track

The experimental results from the vision algorithm were conducted to determine the accuracy of the lateral distance and heading measurements from the vision algorithm only. In this test run, a Hyundai Sonata was equipped with a forward-looking camera that was mounted on a conventional roof rack. The Sonata was driven on the right lane around the full extent of the NCAT test track to simultaneously acquire RTK GPS coordinates and frame grabs from the camera. Each distance from the camera images was compared with the truth measurements from GPS to determine the accuracy of the system. Methodology for obtaining truth data can be found in Appendix B: Truth Data. The accuracy of the coefficient states can be seen by visual inspection. Simply, if the polynomial models lie on the lane markings in the images, then the coefficients are being estimated correctly. In this test run, the quality of the lane markings is not ideal - the lane markings disappear on some segments of the track and are faint in others. As such, in several frames, no lane marking is detected. Also, at one part of the track the asphalt creates a distinct line between light colored asphalt and dark colored asphalt that is near the dashed left lane markings, and the left lane marking estimate tracks this line.

The figure below shows the calculated lateral distance and true lateral distance for the vision system. Two portions, the 50s to 100s mark and 150s to the end, show almost a bias in the system due to the curved roads.

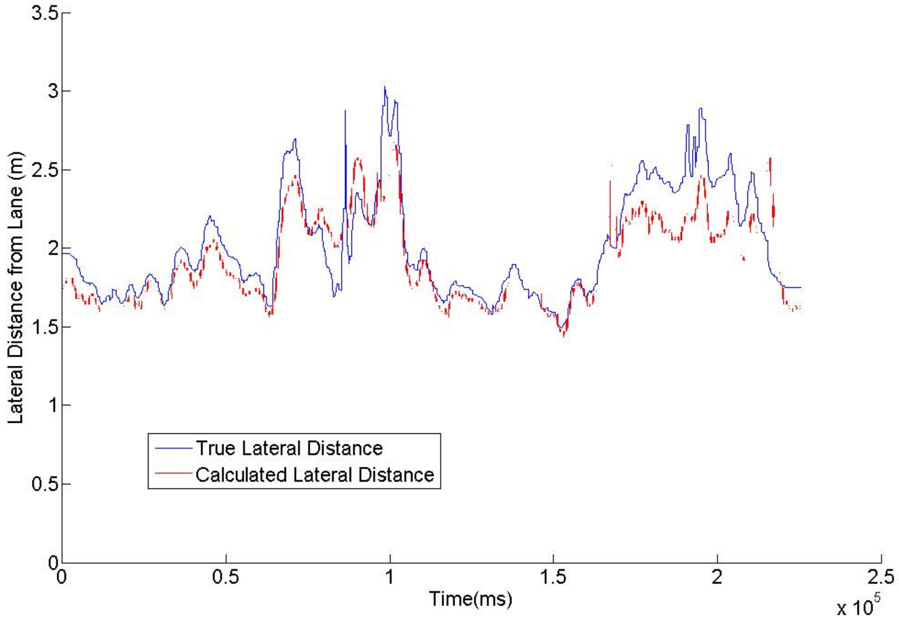

Figure 10: Calculated Lateral Distance and True Lateral Distance for Vision-only Algorithm

Figure 10 shows the truth and calculated lateral distance for the test run. From the about the 50s mark to the 100s mark and the 150s mark to the end of the test run, the lateral distance seems to increase slightly compared with the remaining sections of the graph. These sections consist of the curves of the track. While the driver drove closer to the center of the road in the turns as seen in the truth data, the error between the truth data and calculated lateral distance increased in this section as well. The lateral distance is measured at the lowest row of pixels in the image, which actually consists of a point in front of the vehicle. Due to the curvature in the lane markings, the actual lateral distance and the measured lateral distance at the point ahead of the vehicle is slightly different. Therefore, the error in the system increases due to the curvature of the lane markings.

The lateral distance error is seen in Figure 11. The maximum error in the system on a straightaway is about 20 cm. As mentioned previously, most of the error in Figure 11 arises in the turns of the track. The error is expected to increase if the vehicle maneuvered more within the lane.

The figure below shows the lateral distance error. The error has a max range from -.6m to .6m, where most of the error remaining between 0m and .2m.

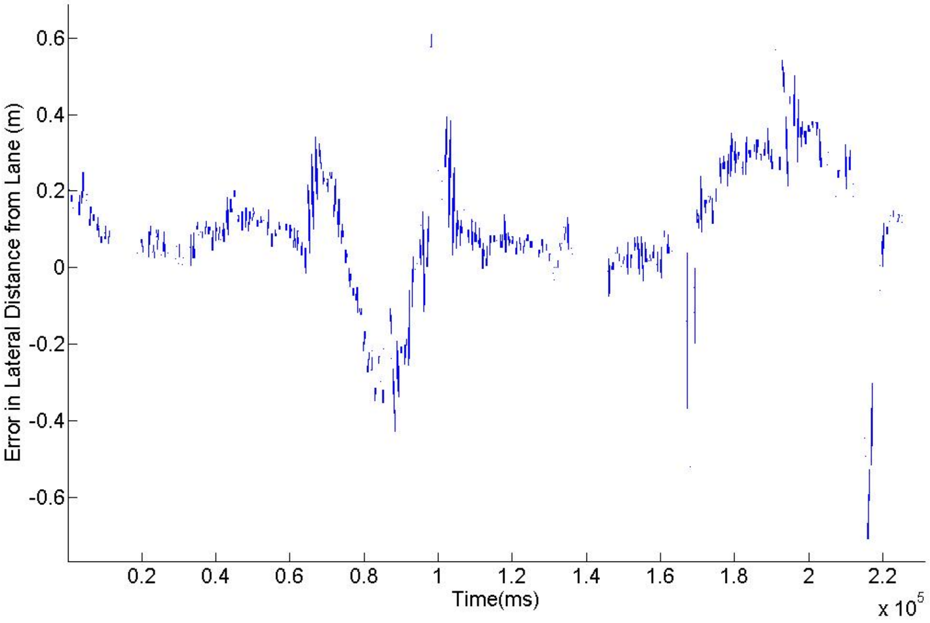

Figure 11: Lateral Distance Error

Various Environments

The performance of the vision system relies heavily on the information retained and ignored in the threshold operation of the image. The ideal threshold image contains only the lane marking lines with all other extraneous objects ignored. Different environmental factors can eliminate the lane marking lines from the image and create histograms in which a constant threshold will result in an edge map which either eliminates the lane markings from the thresholded image or eliminates the lane markings due to their inclusion as other features.

The figure below shows the track at day (left) and the constant thresholded image (right). The constant thresholded image has so-called blobs at the far range of the thresholded image due to a nonideal threshold.

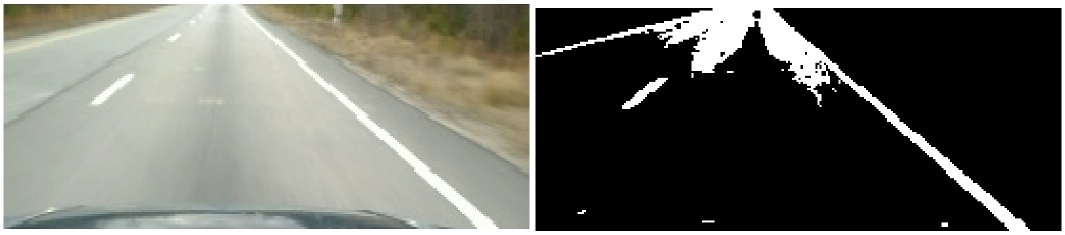

Figure 12: Track at day (left) and corresponding thresholded image (right)

Various Illumination

The histogram of images changes significantly depending on the brightness and time of day. Brightly lit scenes will eliminate the lane marking lines from the thresholded image due to pixel values around the lane markings being higher than the threshold, such as seen in the distance in Figure 12 where blobs mask the inner border lane marking.

The figure below shows the histogram for the daylight original image. The number of pixels for each grayscale value gradually increases to about 60 pixels at around 140 grayscale value. The number of pixels ramps up to 250 pixels at 150 grayscale value. The number of pixels fluctuates slightly until 170 gray level, where it ramps up to 450 pixels around 170 pixels. The number of pixels decreases significantly until the max grayscale value of 255.

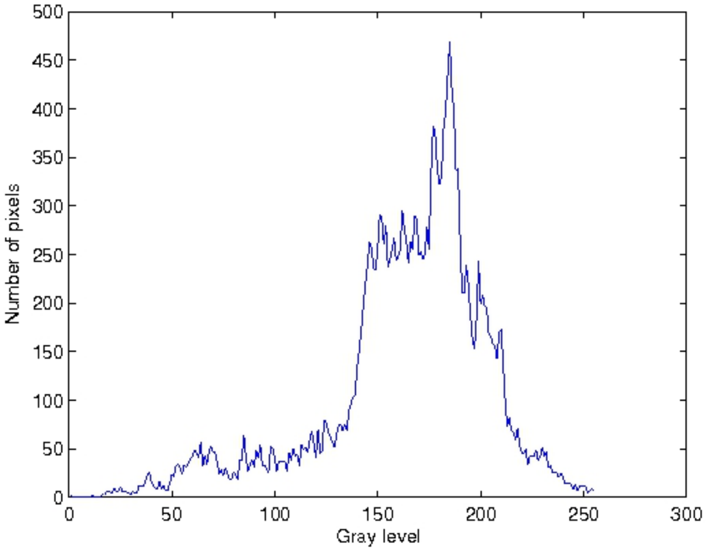

Figure 13: Histogram for the Image in Figure 12

Similarly, a poorly lit scene can eliminate the lane markings completely if the pixels of the lane marking lines have grayscale values below that of the threshold. Figure 14 is one such example, where dusk reduces the brightness of the image and the lane markings are not well illuminated except where the headlights shine, as shown in Figure 14. Figure 15 shows the histogram of the darker image in Figure 14. The grouping of pixels in the histogram are lower in grayscale value than that of the histogram of Figure 13, and a smaller fraction of pixels in the image have grayscale values above the threshold. As such, the actual lane marking grayscale values are lower as well.

The figure below shows the twilight image and the corresponding well extracted lane marking binary image.

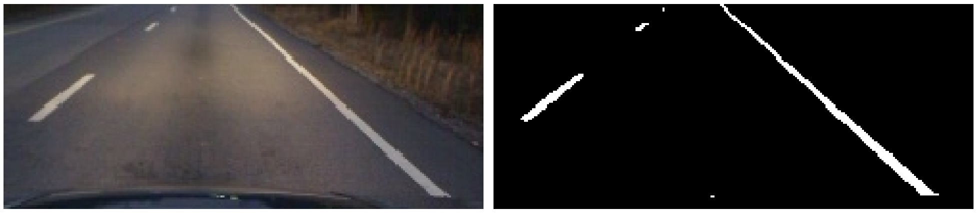

Figure 14: Track at dusk (left) and corresponding edge map (right)

The figure below shows the histogram of the dark image. The histogram has values aggregated considerably lower in the grayscale range than that of the daylight image, with a peak of 360 pixels at around 120 gray level. The graylevel stays constant around 75 pixels from 150 gray level to 190 gray level.

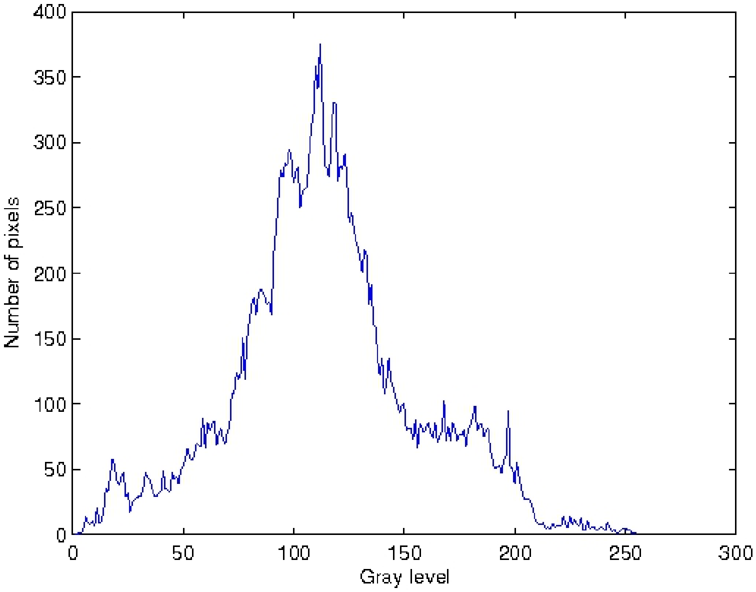

Figure 15: Histogram of image in Figure 14

Despite the dynamic threshold implemented in the system to deal with variable brightness in images, several factors can cause problems during thresholding. Images with severe intensities, such as those images with the sun as seen in Figure 16, can cause the dynamic threshold to be set higher than desired, thus eliminating the threshold from the image completely. Certainly, a new value of K in the threshold calculation would result in a better representation of the lane markings in the thresholded image, as seen in Figure 17. Although the blob from the reflection of the sun is still present in the thresholded image of Figure 17, the lane markings are present. This scenario, though rare, is one situation where a constant threshold outperforms the dynamic threshold.

Figure 16 shows the effect of the reflection of the sun on the pavement (original image left). On the right is the constant thresholded image, with a large blob between lane markings is visible. However, the lane markings are visible.

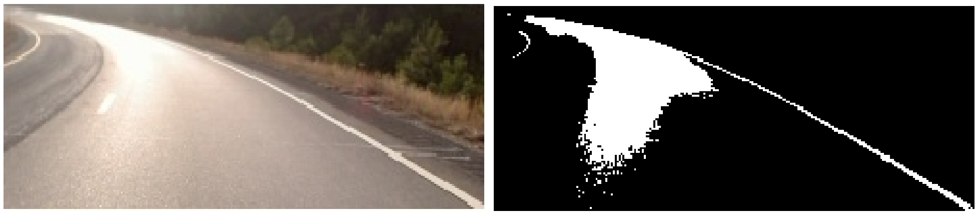

Figure 16: Reflection of sun in road (left) and corresponding edge map with constant threshold (right)

The figure below shows the corresponding edge map using the dynamic threshold. The calculated threshold fails to extract the lane markings and instead only extracts the reflection of the sun on the pavement.

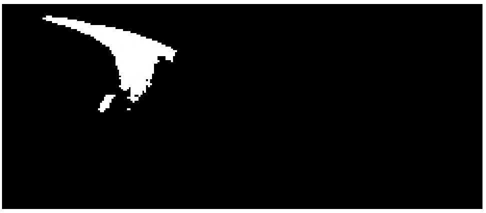

Figure 17: Corresponding edge map using dynamic threshold

Urban Environments

One such application of the vision system is lateral localization within a city to aid a navigation system when GPS fails. Figure 18 shows a typical street in a city (left) and the corresponding edge map (right). Notice that the lane markings are not detected well in the edge map and appear as a line of blobs.

The figure below shows a city street and the corresponding edge map. Three lane are visible for outgoing traffic and are seaparated by dashed lines. One lane is visible for oncoming traffic and is separated by a double yellow line. The edge map shows poorly extracted and jagged lane marking lines.

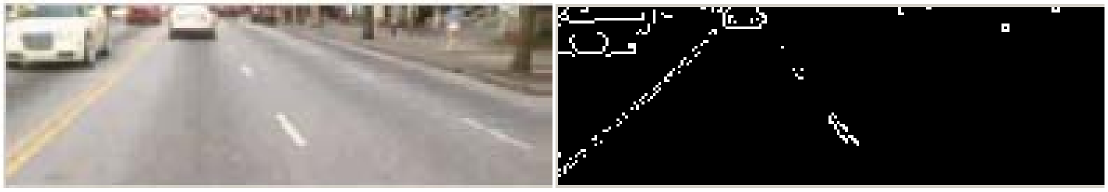

Figure 18: City street (left) and corresponding edge map (right)

A more difficult environment for lane tracking can be seen in Figure 19. In Figure 19, the vehicle is stopped at a light in heavy city traffic. The lane markings are, as expected, not detected and are often concealed in the shadows of the vehicles, as shown in the edge map of Figure 19.

The figure below shows the edge map for a stop at an intersection in heavy traffic. The corresponding edge map (right) fails to extract the lane markings, which can be seen in the shadows on the original image (left).

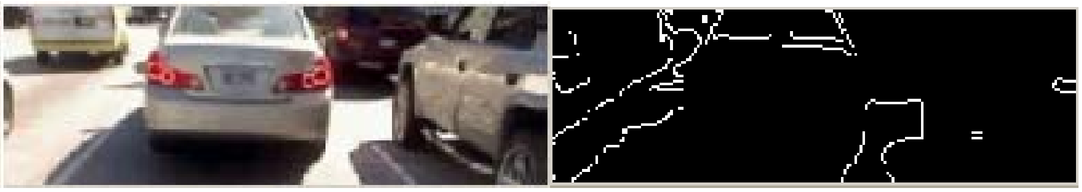

Figure 19: Stop at intersection (left) and corresponding edge map (right)

Like the sun peering through trees in Figure 20, the sun can also create these flashes of light during intersections when the sun is no longer concealed by the buildings. Figure 20 shows the edge of a shadow from a building at twilight in the city. The transition between light and dark creates the edge on the edge map, even though no actual object is present at that location in the image. Also, once the vehicle gets closer to the lit portion of the road, the threshold will exceed the 255 grayscale value due to the dynamic threshold, and the edge map will have no edges.

The figure below shows the effect of a building's shadow on the ground near dusk (left). A clearly visible edge from the shadow's edge can be seen in the edge map (right).

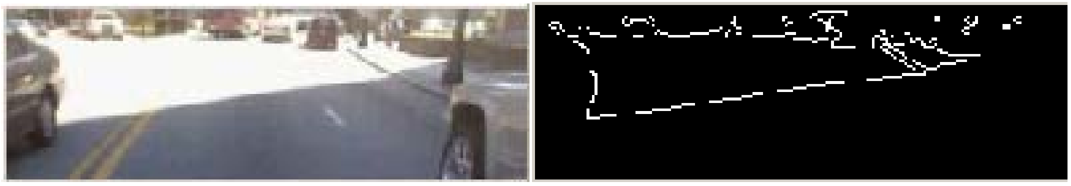

Figure 20: Sunlight peering between buildings (left) and corresponding edge map (right)

Cities are difficult environments for lane detection systems. Lane markings are often faint due to the difficulty in repainting them and lack of funds to do so. In addition, the frequent stop and go traffic can cause tracking of other objects or blockage of the lane markings by other vehicles. Fortunately, vehicle speeds in cities are usually low due to congestion, and less speed means fewer fatalities. As such, most lane departure warning systems are designed for highway scenarios and will not function below certain speeds.

Quantitative Evaluation in Varying Environments

			Rain	Rain
	Dusk	Night	Dusk	Night
Average Absolute Error (m)	0.2379	0.0307	0.0327	0.0512
RMS Error(m)	0.4214	0.0401	0.094	0.1253
std of Error	0.3526	0.0402	0.0887	0.1149
var of Error	0.1243	0.0016	0.0079	0.0132
% Detection	48.01	90	18.08	19.47

Table 1 shows the quantitative results of the vision system in four test runs. The first test run consisted of a run at dusk, where the first half of the run was facing the sun. During this run, the system failed to detect the lane markings due to the presence in the image of the sun, which pushed the threshold above the lane marking values. This impacted is reflected in the detection rate of 48%. The second test run was a night run, where the headlights illuminated the lane marking lines. During the test runs at dusk and at night in the rain, the system failed to detect lane markings for the majority of the run, and the percent detected was less than 20%. For the night, rain and dusk, and rain and night test runs, the average absolute error was less than 6 centimeters, which signifies that the data is appropriate for measurements in a Kalman filter.

	Dusk	Night	Rain Dusk	Rain Night
Average Absolute Error (m)	0.2379	0.0307	0.0327	0.0512
RMS Error(m)	0.4214	0.0401	0.094	0.1253
std of Error	0.3526	0.0402	0.0887	0.1149
var of Error	0.1243	0.0016	0.0079	0.0132
% Detection	48.01	90	18.08	19.47

Table 1: Estimated Error Statistics for Camera Given Different Weather and Lighting Conditions

b. Vision/IMU Sensor Fusion

A loosely coupled approach has been implemented which adds the inputs of an inertial measurement unit (IMU) to an extended Kalman filter (EKF) for determining the lateral distance within the lane even when vision measurements, specifically heading and lateral distance, are not available. The linear Kalman filter from the vision system is used to generate the lateral distance and heading as measurements and the IMU's longitudinal acceleration and yaw rate are used as inputs into the extended Kalman filter. An extended Kalman filter adds the longitudinal acceleration and yaw rate of an inertial measurement unit (IMU) as inputs and the longitudinal velocity of GPS and the lateral distance and heading of the camera system as measurements. This filter uses the measurements from the linear Kalman filter used in the camera-only scenario.

The figure below shows a block diagram of the vision/IMU system. On the left is a view of the vision system, with longitudinal velocity and heading added in the Kalman filter. On the right is the block diagram of the inertial system, which shows measurements from GPS for longitudinal velocity and the vision system for heading and lateral distance. In addition, the inputs for the IMU can be seen as longitudinal acceleration and yaw rate.

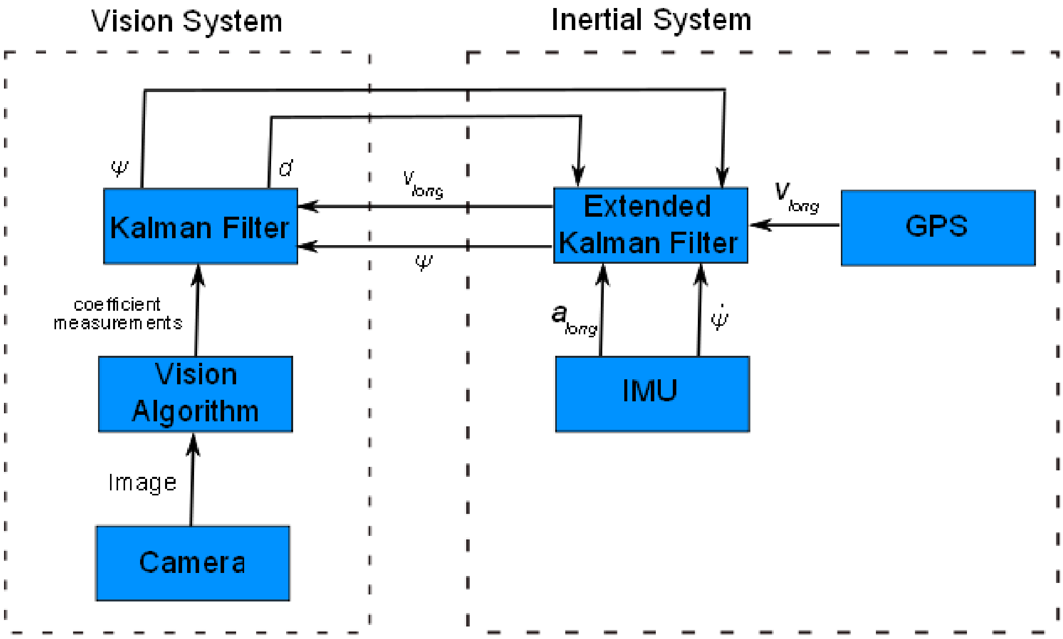

Figure 21: Overview of vision/IMU system

1. Extended Kalman Filter

An extended Kalman filter has been designed which adds the longitudinal acceleration and yaw rate of an inertial measurement unit (IMU) as inputs and the longitudinal velocity of GPS or wheel odometry and the lateral distance and heading of the camera system as measurements. This filter uses the measurements from the Kalman filter used in the camera-only algorithm.

2. Road Frame

The road frame of the system consists of the positive x-axis pointing down the road on the right lane marking, the y-axis pointing perpendicularly to the right of the direction of motion, and the z-axis pointing down and perpendicular to the road plane, as seen in Figure 22. The distance p_y is the lateral displacement from the right lane marking and is typically measured as a negative distance. As such, a positive distance measurement means that the vehicle has driven off of the road.

The figure below shows the road frame of the vision/IMU system with a drawn road and vehicle. The coordinate axes are displayed as well as the heading and lateral displacement.

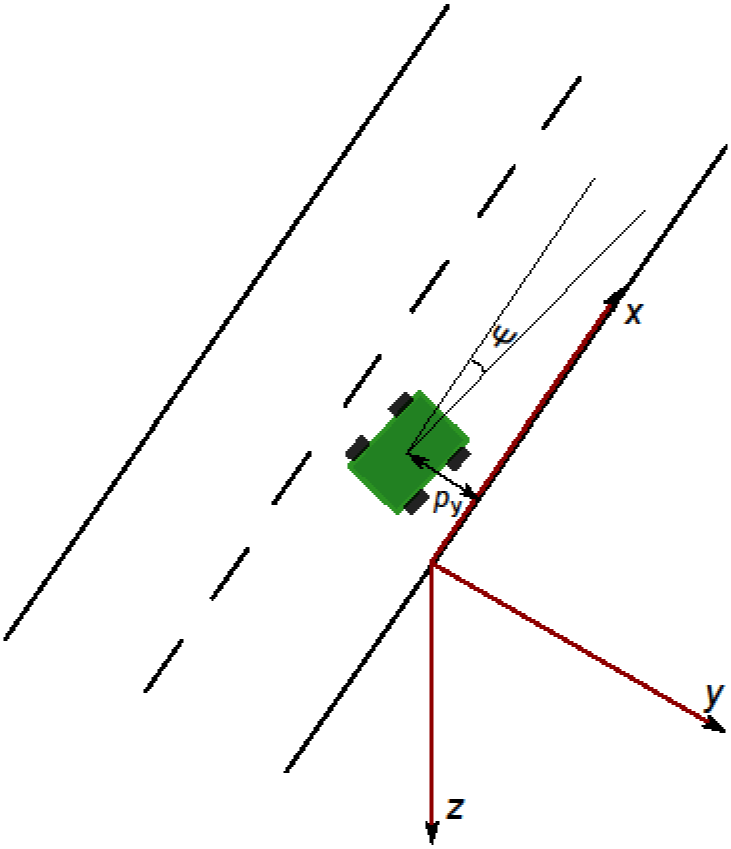

Figure 22: Road Frame

3. Filter Structure

The states for the extended Kalman filter consist of the lateral distance p_y, the longitudinal velocity v_x, the longitudinal acceleration bias b_x, the yaw ψ, and the yaw rate bias b_ψ as shown here:

$$x = \lfloor p_y \quad v_x \quad b_x \quad \psi \quad b_\psi \rfloor$$

These states were chosen to minimize the error associated with noise from the IMU outputs and to minimize integration error.

4. Time Update

The time update of the extended Kalman filter updates the vehicle states with the IMU inputs. Additionally, the lateral velocity state is sent to the vision Kalman filter for updating the lane marking model.

The mechanization equations used in the extended Kalman filter for the inertial states are as follows:

$$\begin{aligned}
\dot{p}_y &= v_x \sin(\psi) \\
\dot{v}_x &= u_2 - b_x \\
\dot{b}_x &= 0 \\
\dot{\psi} &= u_1 - b_\psi \\
\dot{b}_\psi &= 0
\end{aligned}$$

where u_1 is the yaw rate from the IMU, u_2 is the longitudinal acceleration from the IMU, and p_y, v_x, b_x, ψ, and b_ψ are the states of the filter. The bias terms b_x and b_ψ are required in order to have an unbiased signal for integration. The integration of the raw inputs can result in erroneous state estimates due to any non-zero bias. The bias terms are used to subtract out these biases to obtain a signal for integration. The longitudinal acceleration is used to obtain the longitudinal velocity, which in turn is used for determining the lateral distance in the lane. The solution of the equations of motion is conducted using the well-known Runge-Kutta method (RK4).

The time update for the vision system in the combined INS/vision system is different than that of the linear vision system alone. The discrete update of the linear Kalman filter is used to propagate the coefficient states to the next time step. The linear Kalman filter has no dynamics in the system, and the state transition matrix, A, is merely an identity matrix with the same number of rows and columns as the number of states. Knowledge of the system dynamics between images can provide an idea of the movement of the lane marking model in the image plane. The linear Kalman filter's time update can be modified to account for any changes in the lane marking model between measurement updates. This modification results in a nonlinear function which must be solved using an extended Kalman filter.

For lane marking estimates with a small x^2 coefficient, a, the equations for each coefficient were determined from the change in the x coefficient, b, of the polynomial based on the lateral velocity in the image. Small a values occur when the estimated lane marking polynomial appears straight, such as the case when driving on a straight road. Due to the slope selection criteria for determining valid lane markings, curved roads will results in high numbers of valid lane marking lines in order to capture the curve of the road. When fewer valid lane markings are present, a straight road can be assumed. For these situations, the a coefficient can be assumed to be zero, and a line model used.

The number of pixels shifted in image space is determined by:

$$m = \frac{v_x \sin(\psi) dt}{n}$$

where n is the conversion factor from real to image space, v_x is the estimated longitudinal velocity from the EKF, dt is the time step, and m is the number of pixels in the horizontal shift. Since the conversion factor n only applies to the bottom (or closest to the vehicle) row in the image, the horizontal shift is based on this bottom row. Like the lateral distance measurement from the camera, the bottom row of the image was chosen due to its projection of the nearest location in the image and as such the most precise measurement in the image.

The change in the slope of the linear lane model is a function of the shift of the road in image space m and the rate at which the slope of the straight line model changes due to perspective as that shift occurs. When the vehicle is directly over the lane marking, as is the case when $p_y = 0$, the slope of the right lane model line is infinity. As the vehicle moves laterally, the slope of the right lane model gradually decreases until the lane model aligns with the horizon. The restriction of the image size and the knowledge that the lane marking will never reach the horizon line due to other objects and the geometry of the lane itself allows for the use of a conversion factor similar to the n conversion factor used in measuring the lateral distance in the lane. The equation for the number of radians per pixel, r, is then:

$$r = \frac{2q}{w}$$

where w is the width of the image and q is the change in the slope in radians of the lane marking model from the vertical lane marking line to the point at which the lane marking line intercepts the edge of the image. This radial conversion factor can be multiplied with the shift of pixels, m, to obtain the change in slope of the system.

With the additional knowledge of the heading and the longitudinal velocity, the new coefficients for the lane marking line model is as follows:

$$b = \frac{b}{1 - brm}$$
$$c = \frac{c}{1 - brm}$$

These equations serve to give an indication of the location of the lane markings in the image when the image fails to detect the lane markings in the vision system. In addition, the tracking of the lane marking in the lane without camera measurements allows for faster convergence once the lane markings are once again visible.

5. Measurement Update

The measurement update serves to correct the states of the filter from any errors accumulated during the time updates. The vision system updates the heading ψ and lateral distance p_y states, while the velocity measurement updates the longitudinal velocity state, v_x. Bias states are not updated.

The H matrix relates the measurements to the states and is as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Each state, other than the bias terms, has a corresponding measurement. The measurement noise covariance matrix R is as follows:

$$R = \begin{bmatrix} \sigma_{px}^2 & 0 & 0 \\ 0 & \sigma_{vlong}^2 & 0 \\ 0 & 0 & \sigma_{\psi}^2 \end{bmatrix}$$

where σ_{px}^2 is the variance of the lateral position measurement, σ_{vlong}^2 is the variance of the longitudinal velocity measurement, and σ_{ψ}^2 is the variance of the heading measurement from the camera.

6. Experimental Results

Figure 23 shows the lateral distance estimate compared with the truth for lateral distance. In this test run, the lane markings fade away near the beginning of the run. About halfway through the run, the vehicle undergoes a lane change and an immediate return to the original lane. At the end of the run, the lane markings drift away at an exit. The estimated solution remains close to truth except for the following three places: the region where no lane markings were present from 10s to 20s, the lane change maneuver from 23s to 33s, and the exit to the skid pad at the NCAT test track where the lane markings disappear from 36s to the end of the test. Each of these regions resulted in no camera measurements, and dead reckoning from the IMU was used for lane estimation. As such the integration drift is seen.

The figure below shows the estimated lateral distance and true lateral distance for the experimental test run. Around 10s into the run, the estimated lateral distance diverges from the true lateral distance as the integration drift occurs without measurement updates. At 20s into the run, measurement updates corrects the drift and the estimate jumps back to the truth. From 22s to 30s into the run, the double lane change maneuver occurs and the drift occurs once more. At 30s into the run, the measurement update corrects the drift and the truth and estimate stay close for the remainder of the test run.

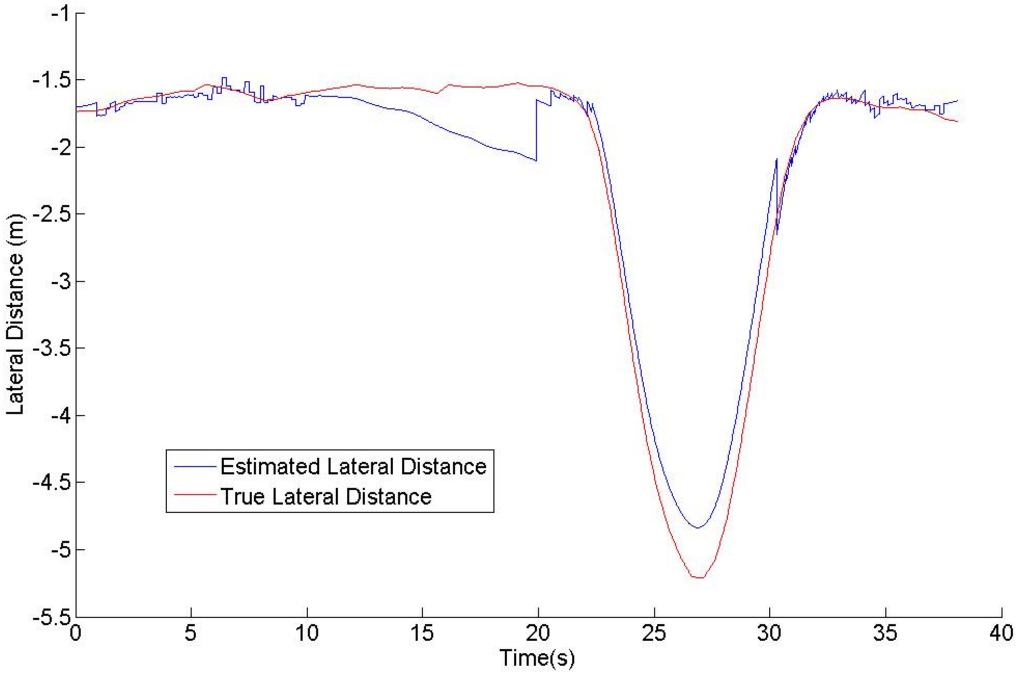

Figure 23: Estimated Lateral Distance and True Lateral Distance

In Figure 24, the error in the system with respect to truth is shown. Notice that the error grows in the three regions of dead reckoning where no measurements from the camera or GPS are presented.

The figure below shows the lateral distance error for the system. As expected, the error increases through the two regions of integration due to IMU drift.

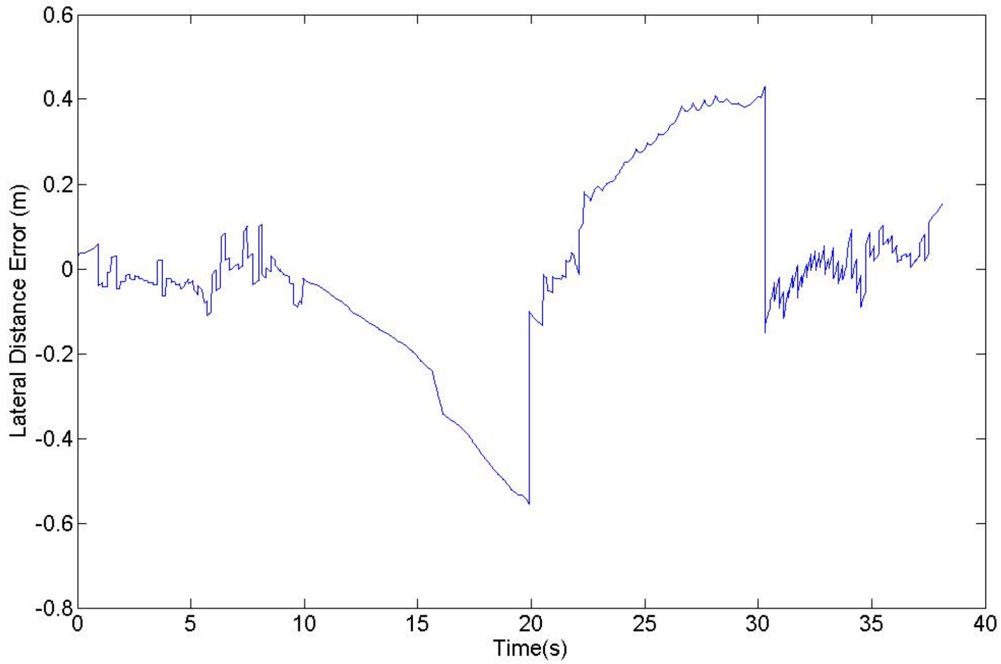

Figure 24: Lateral Distance Error

Unlike the lateral distance state, the heading state was less susceptible to integration drift. While the lateral distance state had error due to integration from both the integration of the yaw rate and the longitudinal acceleration, the heading state had integration error from only the yaw rate. In addition, the noise present in the yaw rate is smaller than that in the longitudinal acceleration, and less drift occurs, as shown by Figure 25.

The figure below shows the heading estimate and the true heading for the test run. Both values stay approximately equal throughout the test run. The double lane change maneuver is clearly evident between 20s and 35s.

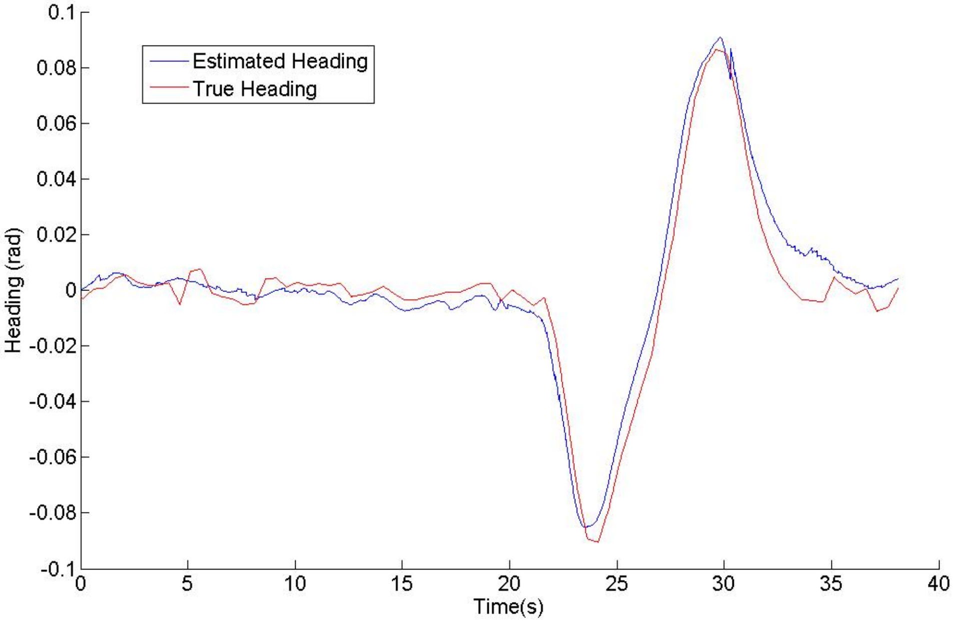

Figure 25: Estimated Heading and True Heading

Since the variance of the measurement in the R matrix of the heading measurements is less than that of the lateral distance measurement, the impact of the heading measurement is lessoned, as shown in the error plot of Figure 26. The maximum error from the heading estimate is about 0.015 rad, or less than one degree. It is worth noting that the error in the truth measurement due to sideslip was also determined to be less than one degree, which means that the error in the system could be due to the error from sideslip in the truth measurement during the last change.

The figure below shows the heading error for the test run. The error has a max range of -.015 rad to .015 rad.

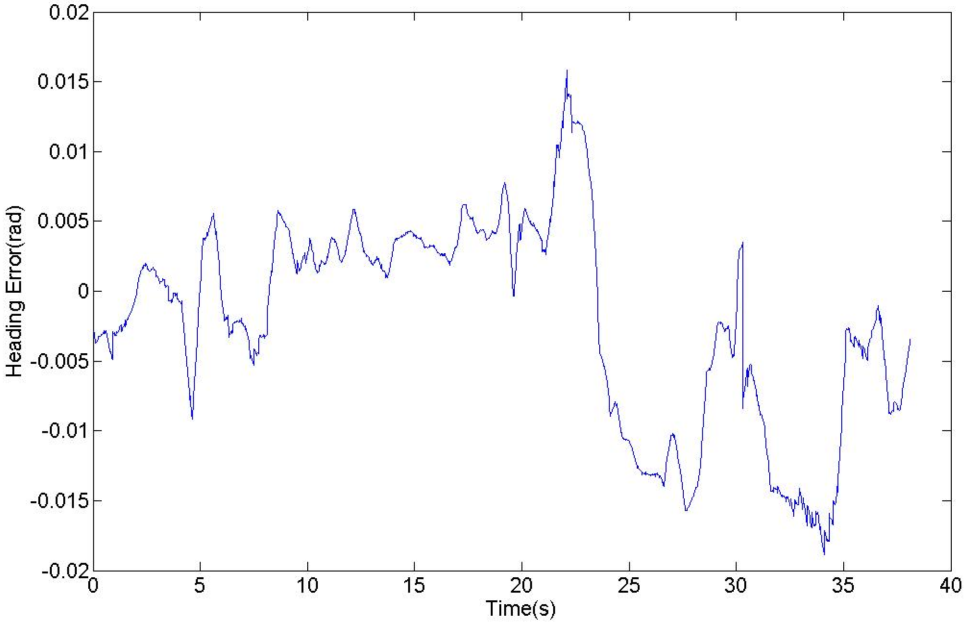

Figure 26: Heading Error

By reducing the process noise covariance on the yaw rate bias state, the error in the lateral distance during the maneuver can be decreased as the yaw rate bias decreases and better estimates the bias through the maneuver. However, this change also results in a more dynamic yaw rate bias before the dead reckoning during the lane marking outage, and the error during the lane marking outage increases.

3 LiDAR Lane Detection

A. Hardware

The hardware used for detecting the lane markings using LiDAR is an Ibeo ALASCA XT, which is a 3D automotive grade LiDAR. The LiDAR provides range measurements as well as reflectivity measurements (ehcowidth). The Ibeo has four vertically diverging beams giving the Ibeo a 3.2° field of view directly in front of the vehicle. Data from the LiDAR was taken at 10Hz with an angular resolution of approximately 0.25°.

The figure below shows the LiDAR scan area including the four scans, the coordinate axes, and the polar coordinate system used by the LiDAR to report data.

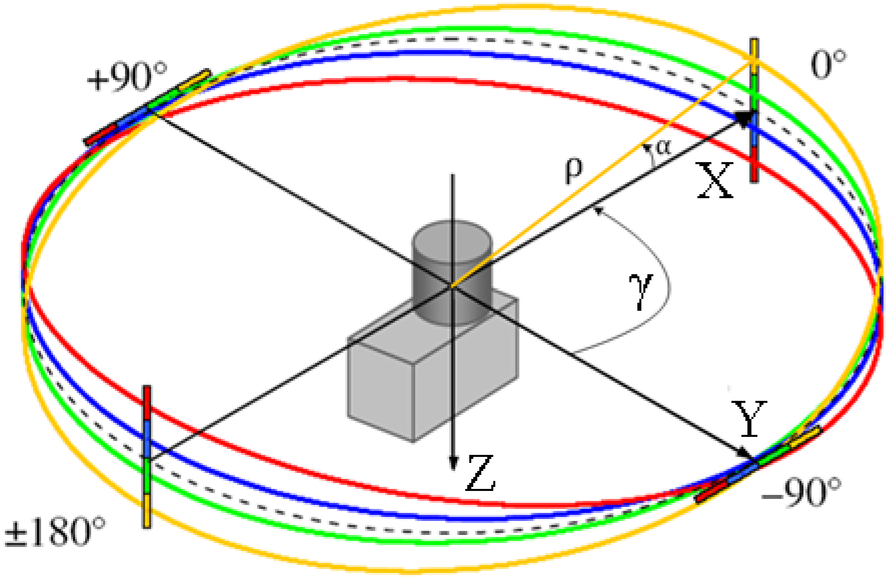

Figure 27: LiDAR Scan Area

The LiDAR was mounted to the roof rack of a vehicle. This is an ideal location for the mounting of the LiDAR because it provides a resolution of approximately 0.040m at the lane markings while scanning less than two meters in front of the vehicle. The shortened scan distance in front of the vehicle aids in the reduction of the effects of vehicle yaw, as well as prevents the scanning of other vehicles, thereby reducing any processing time that would have been otherwise dedicated to the removal of these obstructions.

B. Detection

The detection of lane markings utilizes the principle that the lane markings are more reflective than the road's surface. The detection algorithm will first bound the search area so that a lane marking is guaranteed to be scanned. Then an ideal lane model will be fit to the scanned data to detect the lane markings. Once the lane markings have been detected, the position in the lane will be determined and filtered to provide a final solution.

C. Bounding

The detection algorithm will first bound the search area so that regardless of the position of the vehicle in the lane, a lane marking will always be scanned if it is present, as seen in Figure 28. This is accomplished by assuming an ideal lane width of 12ft. Then it is assumed that the right most tires of the vehicle are touching the lane markings. The angle that is required to scan the left most lane marker is determined to be the angle bound as seen in equation below:

$$Angle\ Bound = \operatorname{atan}\left(\frac{LW - \frac{VW}{2}}{\rho}\right)$$

where LW represents the lane width, VW represents the vehicle's width, and is the distance between the LiDAR and the ground measured with zero horizontal angle.

The figure below shows a view in front of the test vehicle with the front shown. Red lines can be seen as an indication of the location of the scan from the LiDAR.

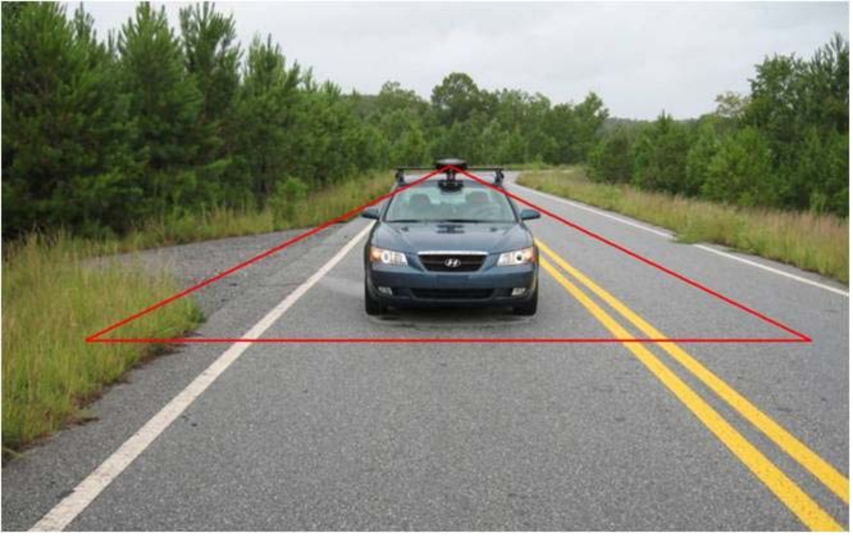

Figure 28: Scanning for Road Markings

D. Scan Matching

The heart of the detection algorithm is the matching of an ideal lane scan to an actual lane scan. The figure below shows 100 LiDAR scans averaged together. Note the distinction of the four labeled peaks and their correspondence to the image of the roadway.

The figure below shows the echo width for 100 averaged LiDAR scans. Two red lines are shown which symbolize the road edge. Four peaks can be seen, and each are labeled from 1 to 4 to denote the lane markings.

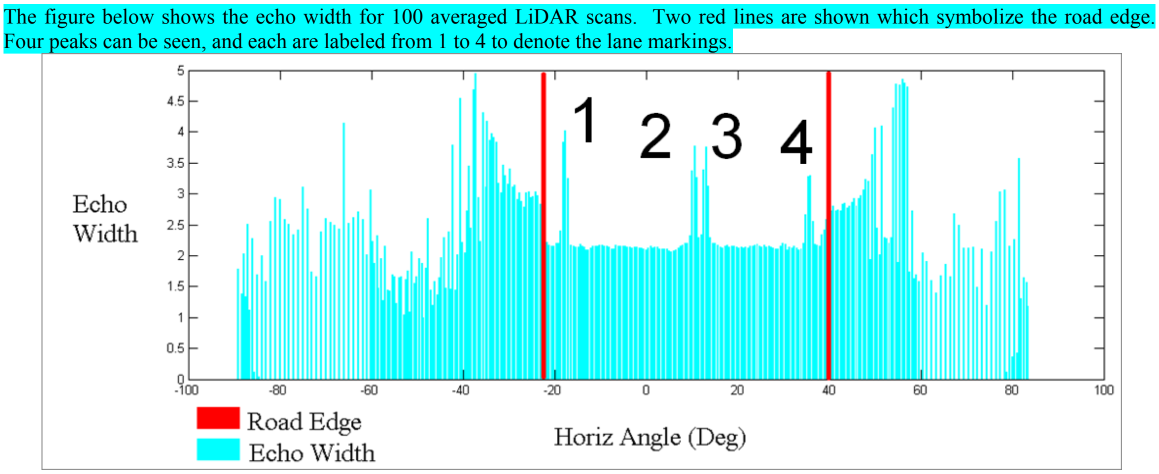

Figure 29: 100 Averaged LiDAR scans

The figure below shows the corresponding road image from the scan in Figure 29. The edge of the road are shown with a red line, and the corresponding peaks (lane markings) from Figure 29 are labeled as seen from the road.

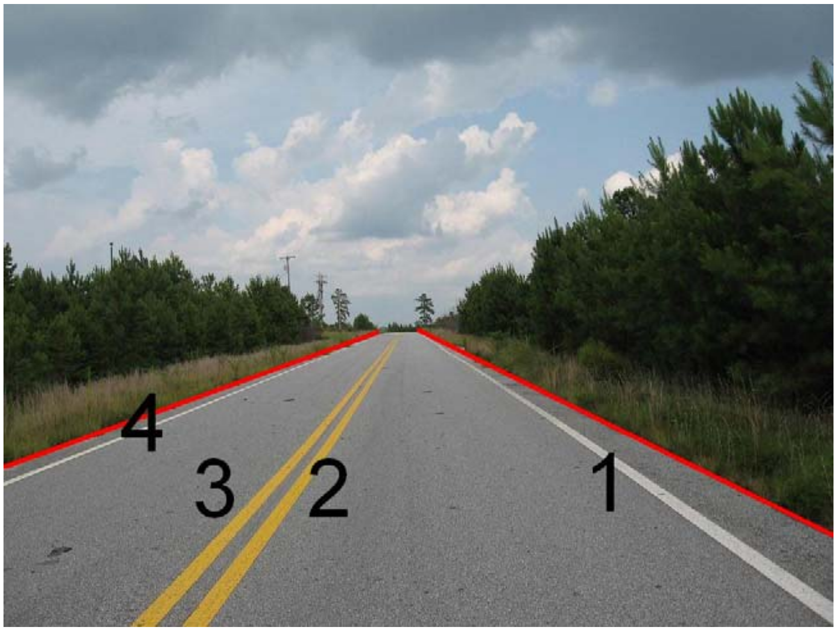

Figure 30: Lane Markings seen in LiDAR Scan

Notice in Figure 29 that the region between the spikes representing the lane markings is a relatively constant area, and as you move outside the scan bounds, there is a highly noisy region created by the environment bordering the roadway. While this averaged scan looks quite ideal, a single scan of the environment can be significantly less intuitive as seen in Figure 31.

The figure below shows a typical LiDAR scan of the road. In this graph, the peaks are much more difficult to see as the peaks are not clear.

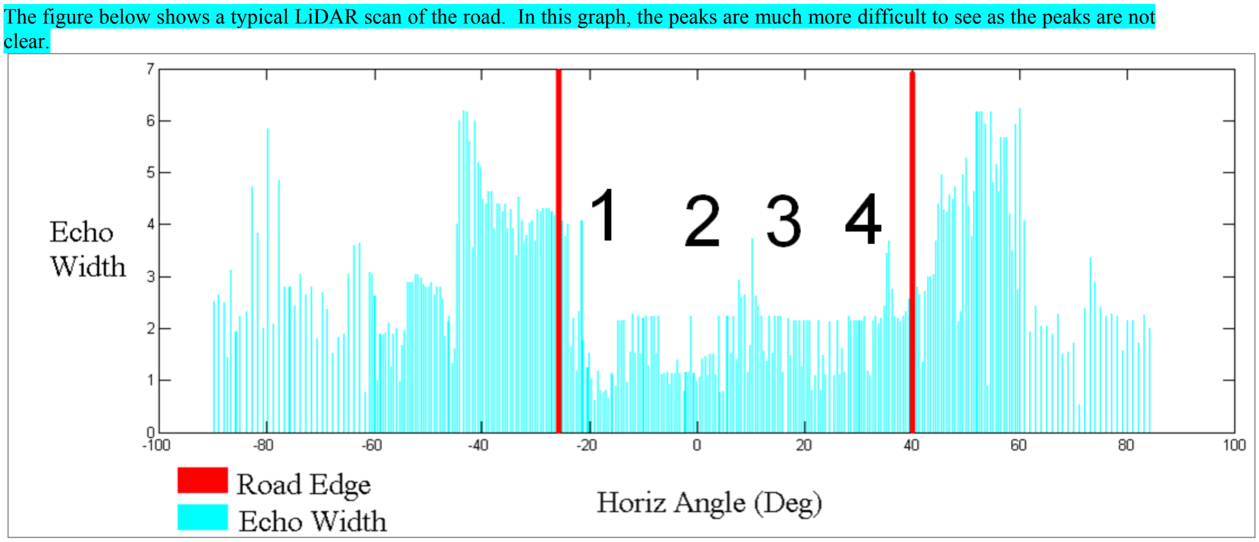

Figure 31: Typical LiDAR Scan

Therefore the algorithm must be capable of detecting lane markings in these types of non-ideal conditions where the lane markings may not be distinct. To do this an ideal lane scan is created to model the reflectivity of a lane. This ideal scan is seen in Figure 32.

The figure below shows an ideal scan. The road in front of the vehicle is represented by the constant portion of the echo width (reflectivity), and the lane markings are shown as the peaks.

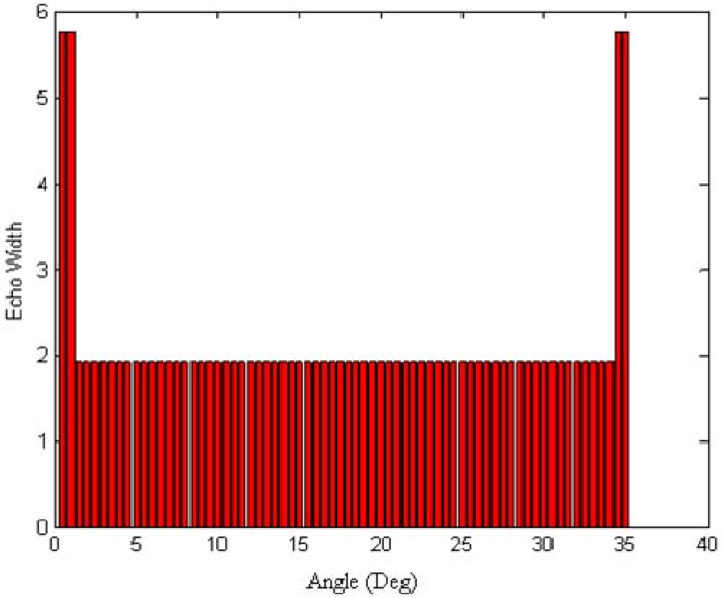

Figure 32: Ideal Scan

The ideal scan is created by first scanning the area in front of the vehicle and averaging the reflectivity. This is seen in Figure 32 as the constant area between the spikes. The spikes are generated by increasing the averaged reflectivity by 75% to represent the lane markings. Thus this ideal lane model is dynamic and changes with road and environmental conditions. Once this ideal scan is created is it compared to the actual LiDAR scan. This comparison is done over the entirety of the LiDAR scan within the previously established bounds, where the ideal LiDAR scan is varied in size from some minimum lane width in length to some maximum lane width in length until the search space is exhausted. With each comparison, the mean square error between the ideal and actual scan is calculated. The region that provides the minimum mean square error between the ideal scan and the actual scan is determined to be the location of the true lane markings. A final comparison of the reflectivity of the actual LiDAR scan to where the ideal scan's lane markings are located is analyzed. If the area of the lane markings of the extracted lane is not at least 30% above the averaged reflectivity used to model the road's surface, it is assumed that no lane marking exists. This process is repeated for each of the four scan layers. Finally, once a lane marking is detected, a window of approximately 4° is placed on the area from which the lane was detected as seen in Figure 33.

The figure below shows the bounding strategy for narrowing the search area. Instead of the large bound as seen in Figure 28, only certain portions of the road are analyzed to avoid erroneous lane detections.

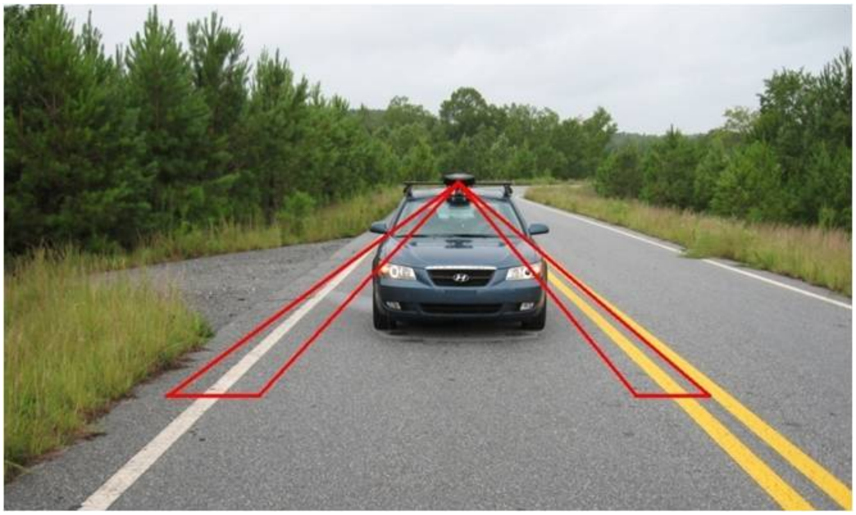

Figure 33: Bounds for Narrowed Search Area

This allows for a narrowed search space for subsequent extractions and helps mitigate any sources of error in the scan.

E. Filtering

Once the lane markings have been extracted, the position in the lane can be computed. To do this, the distance to each of the left and right lane markings is calculated and put through a weighted average to obtain a distance to the left and right lane markings. Once this weighted averaged is found, the values are subtracted to determine the offset from the center of the lane. If only the left or right lane markings are detected, the position in the lane is determined by subtracting the measured left or right distance from either an ideal lane width or a from a previously calculated lane width when both left and right measurements were available. Once this distance from the center of the lane is calculated, the result is put through a single state Kalman filter to smooth out the data.

F. Testing and Results

Tests were performed on a number of different scenarios of varied road type where no high precision GPS measurement was available; instead, an estimate of the road width was compared to the known road width as an estimate of algorithm performance. Detection rates were also recorded. Table 2 highlights these results.

Table 2: Various Scenarios for Road Type

	Avg. Lane Width Error(m)	Std of Error(m)	Detection(%)
Highway Driving (dashed lines)	0.075	0.233	94.7
Yellow & White markings	0.042	0.272	81.7
Gravel on road	0.129	0.215	97.4

Grass bordering road	0.169	0.329	76.86

Other testing included a comparison of the reported position of the LiDAR in the lane to a precision GPS position, which is compared to a survey of our test track. The results of these tests can be seen below in Figure 34 and Figure 36 along with histograms of the error in Figure 35 and Figure 37. The first test (Test 1) had a mean error 0.2293m and a variance of 0.0536m, while the second test (Test 2) has a mean error 0.1305m and a variance of 0.0484m.

The figure below shows the experimental results of the lane detection using RTK GPS as truth for lane position from center. A few spikes can be seen between the 50s and 100s marks.

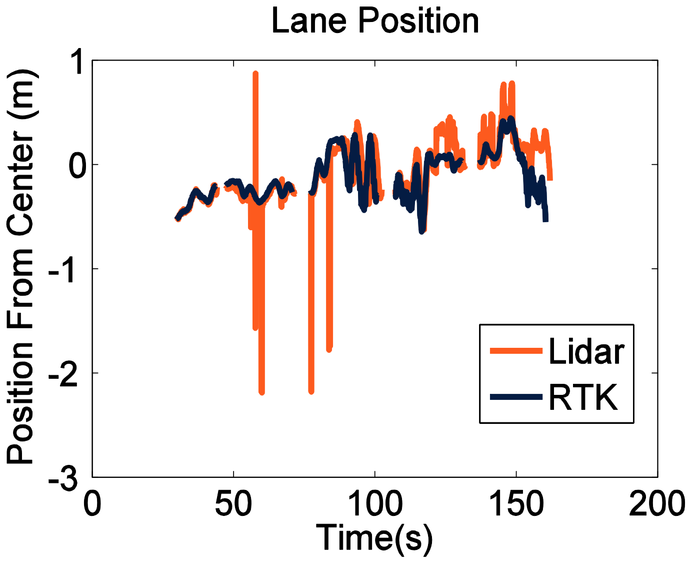

Figure 34: Lane Position LiDAR vs. RTK GPS Position (Test 1)

The figure below shows the error histogram for the first test run. The majority of the error can be seen to be close to 0 and is followed by error in the .25m range. The errors from the spikes can be seen between the -1.5m range and -2m range.

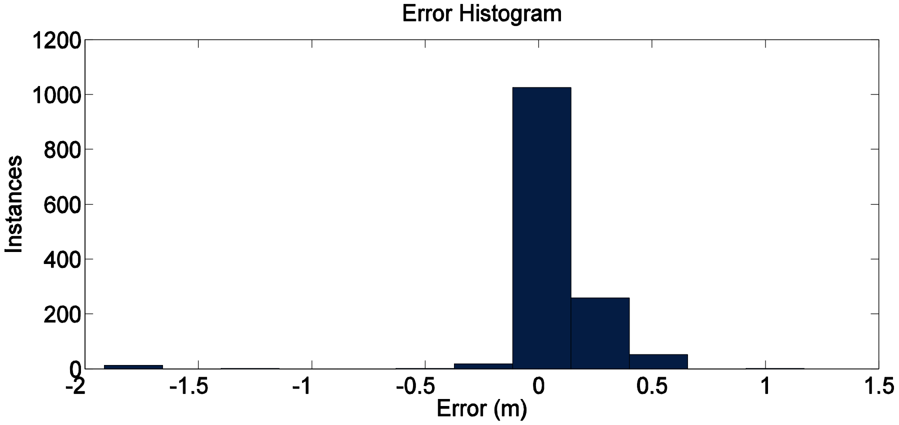

Figure 35: Error Histogram for Test 1

The figure below shows the position from center for the lane position. Spikes can once again be seen.

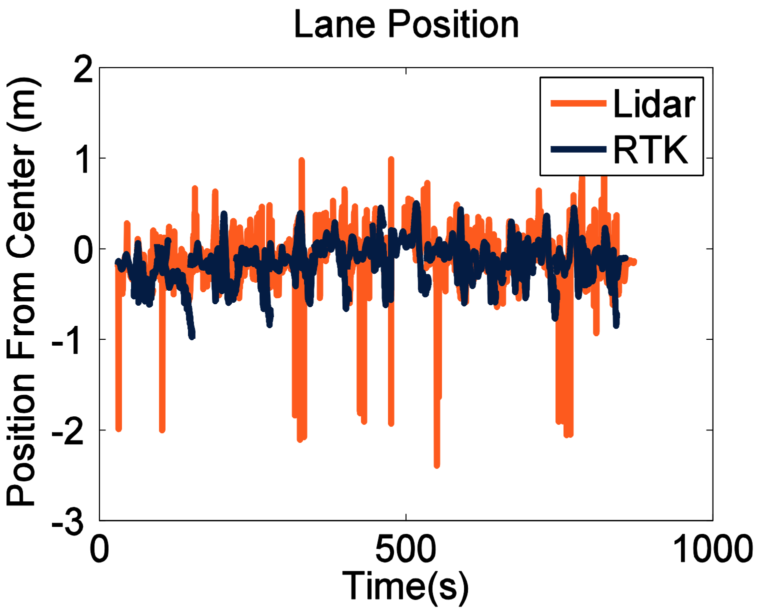

Figure 36: Lane Position LiDAR vs. RTK GPS Position (Test 2)

The figure below shows the error histogram for Test 2. The remains mostly in the 0-.25m range, with some error in the 0- -0.5m range and 0 – 0.5m range. Once more, the spikes in the image can be seen in the -1.5m and -2m range.

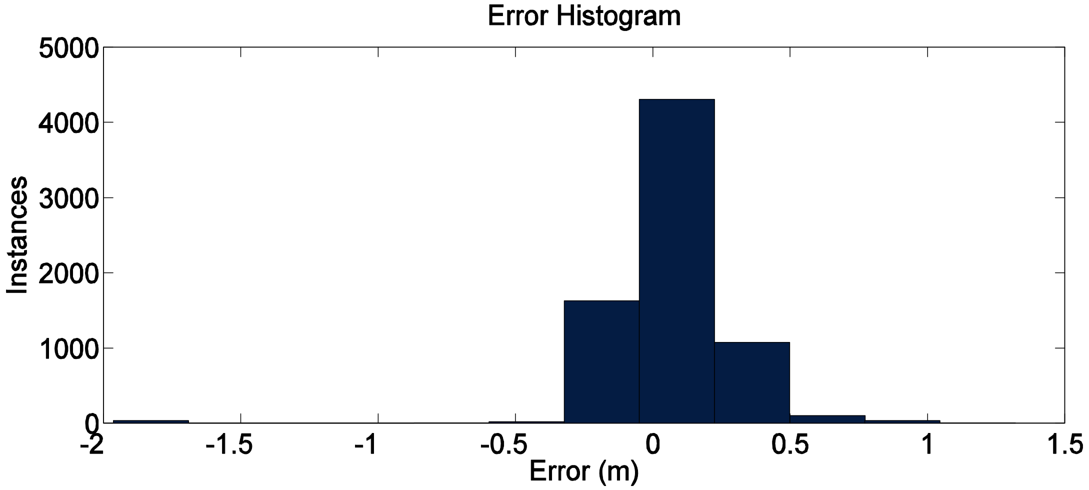

Figure 37: Error Histogram for Test 2

An additional series of tests were performed under various weather conditions, whose position results are compared to a high accuracy GPS survey in Table 3.

Table 3: LiDAR Weather Data Results

Weather Cond	Average Absolute Error (m)	RMS Error (m)	Std of Error (m)	% Detection
Dusk	0.2375	0.3559	0.3436	99
Night	0.4974	0.5413	0.2152	99
Rain Dusk	1.0327	1.1187	0.01858	18.9

Rain Night	0.9232	1.0034	0.1571	32.57

4 System Integration

This section describes the navigation filter, an extended Kalman filter, used in this project. The extended Kalman Filter is a well-known recursive solution to state estimation for nonlinear systems.

A. Six Degree of Freedom Navigation Filter

The section only covers the time update and the GPS measurement update. The vision measurements (camera and LiDAR) used in this navigation filter are presented in previous sections. There are two methods of using GPS measurements to update the filter. The first method is the loosely coupled measurement update which uses positions and velocities calculated by the GPS receiver to update the filter. The second method is the closely coupled measurement update. This method uses the GPS receiver's range and received signal frequency to each GPS satellite in order to update the navigation filter. Both types of measurement updates are covered in this section. The filter is set up to use loosely or closely coupled GPS measurement updates. This allows the filter to use a loosely coupled measurement update as long as the receiver provides a solution. If the receiver is not able to provide a solution due to insignificant GPS satellite visibility, the navigation filter can switch over to the closely coupled GPS measurement update. This allows the filter to be updated with the available GPS observations.

The navigation coordinate frame used is the earth centered earth fixed (ECEF) coordinate frame. The filter tracks the three-dimensional position and velocity of a vehicle in the ECEF coordinate frame. Measurements from an inertial measurement unit (IMU) are integrated to determine vehicle position and velocity states. Any error in the IMU measurements causes an increasing error in position and velocity estimates due to the integration. In order to eliminate this drift, measurement updates are used to update the states. This corrects the error in the position and velocity states due to integration of IMU errors. IMU measurements are also biased. Integrating biased IMU measurements will result large error growth in estimated position and velocity. The filter estimates the biases in the IMU measurements to compensate for this effect.

In order to rotate the IMU measurements (given in the body coordinate frame) to the navigation filter's coordinate frame (ECEF coordinate frame) the filter must estimate the attitude of the IMU with respect to the ECEF coordinate frame. The attitude is represented by the three Euler angles necessary to rotate between the vehicle body coordinate frame and the ECEF coordinate frame. The conventional roll, pitch, and yaw angles can be calculated using the Euler angles when the location of the vehicle is also known. More on coordinate frame rotations can be found in Appendix C. The ECEF coordinate frame is a three degree of freedom coordinate frame that is fixed to the center of the earth (denoted by sub/superscript e).

The vehicle body frame (denoted by sub/superscript b) used is the standard vehicle body coordinate frame in which the x axis points in the direction of travel, the y axis points out the passenger side, and the z axis points down.

1. States and State Equations

The following equation shows the navigation filter's state vector.

$$\vec{x} = \left[\vec{r}_{eb}^{e} \quad \vec{v}^{e} \quad \vec{\psi}^{e} \quad \vec{b}_{a}^{b} \quad \vec{b}_{g}^{b} \quad \overrightarrow{dt}_{gps}\right]$$

The state vector contains estimates of the position, velocity, and attitude of the vehicle. The state vector also contains IMU bias and GPS receiver clock bias/drift. $\vec{r}_{eb}^{e}$ is a three state vector containing the estimated three-dimensional position of the vehicle in the ECEF coordinate frame (in meters). $\vec{v}^{e}$ is a three state vector containing the estimated three-dimensional velocity of the vehicle in the ECEF coordinate frame (in meters per second). $\vec{\psi}^{e}$ is a three state vector containing the estimates for the three Euler angles (in radians) that describe the attitude of the vehicle. The attitude is expressed in terms of the three necessary rotations to rotate the ECEF coordinate frame to align with the body coordinate frame. $\vec{b}_{a}^{b}$ is a three state vector containing the estimated IMU accelerometer biases given in the IMU or body coordinate frame. $\vec{b}_{g}^{b}$ is a three state vector containing the estimated IMU gyro biases also given in the body coordinate frame. $\overrightarrow{dt}_{gps}$ is a two state vector containing clock bias and drift terms necessary to use GPS pseudorange and pseudorange rate measurements. The $\overrightarrow{dt}_{gps}$ is only necessary if these measurements are used; therefore, the state vector can consist of 15-17 states depending on the type of GPS measurements currently being used.

Equations for the change in each state are necessary to perform the time update and propagate the states forward in time. These equations are given in below.

$$\dot{\vec{x}} = \begin{bmatrix} f_1(\vec{x},\vec{u},\vec{v}) \\ \vdots \\ f_n(\vec{x},\vec{u},\vec{v}) \end{bmatrix} = \begin{bmatrix} \vec{x}_{4:6} \\ C_b^e(\vec{u}_{1:3} - \vec{x}_{10:12} - \vec{v}_{1:3}) - \Omega - g \\ C_{MECH}(\vec{u}_{4:6} - \vec{x}_{13:15} - \vec{v}_{4:6}) - [0 \quad 0 \quad \omega_e]^T \\ -\vec{v}_{7:12} \\ \vec{x}_{17} - \vec{v}_{13} \\ -\vec{v}_{14} \end{bmatrix}$$

The equations for the change in each state is a function of all the states ($\vec{x}$) and inputs ($\vec{u}$). The equations are determined by creating a kinematic model. The kinematic model describes the change in position, velocity, and attitude based off measurements of acceleration and rotation rates. IMU accelerations minus their biases are rotated into the ECEF coordinate frame using the rotation matrix (C_b^e). The rotated unbiased accelerations are corrected and integrated to determine position and velocity. Change in attitude is determined by multiplying the gyro inputs by the IMU mechanization matrix (C_{MECH}).

The first three states are the position states. The change in the position states is equal to the velocity states ($\vec{x}_{4:6}$). The next three states are the velocity states. The change in velocity is equal to the acceleration. The acceleration is an input from the IMU; however, the accelerometers provide accelerations in the body coordinate frame. The acceleration measurements are also biased. The accelerometer bias estimates ($\vec{x}_{10:12}$) are subtracted from the IMU measurements. The result is then rotated into the ECEF coordinate frame using the rotation matrix given below.

$$C_b^e = \begin{bmatrix} c_8c_9 & s_7s_8c_9 - c_7s_9 & c_7s_8c_9 + s_7s_9 \\ c_8s_9 & s_7s_8s_9 + c_7c_9 & c_7s_8s_9 - s_7c_9 \\ -s_8 & s_7c_8 & c_7c_8 \end{bmatrix}$$

The rotation matrix is constructed using the attitude estimates ($\vec{x}_{7:9}$). c_i is the cosine of the i^{th} state and s_i is the sine of the i^{th} state. More on coordinate frame rotation and translation can be found in Appendix C.

Accelerometers also measure the specific force due to gravity along with centripetal acceleration and the Coriolis effect due to the rotation of the earth. The estimated values of the forces must be subtracted from the rotated unbiased accelerations. The equation below is used to estimate the specific force caused by gravity.

$$g = GM(\vec{x}_1^2 + \vec{x}_2^2 + \vec{x}_3^2)^{-1.5}\vec{x}_{1:3}$$

The equation splits the specific force into its vector components in the ECEF coordinate frame. The norm of this vector is equal to approximately $9.81 m/s^2$. The vector is subtracted from the rotated unbiased IMU accelerations. GM is the standard gravitational parameter. The standard gravitational parameter for earth is $3.896004418x10^{14}\ m^3/s^2$. The equation below shows the estimated specific force vector cause by centripetal acceleration and the Coriolis effect (in the ECEF coordinate frame) due to the earth's rotation.

$$\Omega = \begin{bmatrix} -\omega_e^2\vec{x}_1 - 2\omega_e\vec{x}_5 \\ -\omega_e^2\vec{x}_2 + 2\omega_e\vec{x}_4 \\ 0 \end{bmatrix}$$

The centripetal acceleration and Coriolis effect are fictional forces when using the ECEF coordinate frame because this coordinate frame rotates with the earth. ω_e is the rotation rate of the earth around the z axis of the ECEF coordinate frame. The value used for this work is $7.2921159x10^{-5}\ rad/s$. This vector is subtracted from the rotated unbiased IMU accelerations to correct for the fact that the ECEF coordinate frame is a non-inertial reference frame.

The change in attitude states is equal to the product of the IMU mechanization matrix (C_{MECH}) and the unbiased gyro measurement vector.

$$C_{MECH} = \begin{bmatrix} 1 & \left(\frac{s_7 s_8}{c_8}\right) & \left(\frac{c_7 s_8}{c_8}\right) \\ 0 & c_7 & -s_7 \\ 0 & \left(\frac{s_7}{c_8}\right) & \left(\frac{c_7}{c_8}\right) \end{bmatrix}$$

The rotation rate of the earth (ω_e) is present in the gyro measurements. The rotation rate of the earth can be subtracted out after the IMU mechanization matrix is multiplied by unbiased rotation rates. The change in the bias states is set to zero. The change in GPS receiver clock bias is equal to the receiver clock drift ($\vec{x}_{17}$). The change in the GPS receiver clock drift is also set to zero.

Inputs to the system ($\vec{u}$) are measurements from an IMU. The first three inputs in the input vector ($\vec{u}_{1:3}$) are the accelerometers' measurements of acceleration (m/s^2) in the x, y, z (in that order) axis of the vehicle body coordinate frame. The fourth through sixth elements of the input vector ($\vec{u}_{1:3}$) are the gyro measurements of rotation rate (rad/s) around the x y z (in that order) axis of the vehicle body coordinate frame.

The process noise sources ($\vec{v}$) are included in the state equations. These variables are present in the equations to show the noise sources in the system. They are not actually used when updating the states. The process noise sources are only used to determine the noise input matrix (B). The first three noise sources ($\vec{v}_{1:3}$) are from the IMU accelerometers. The fourth through sixth elements of the noise source vector ($\vec{v}_{4:6}$) are from the gyro measurements. The seventh through twelfth elements of the noise source vector ($\vec{v}_{7:12}$) are artificial noise sources that are added in to allow for IMU bias estimation. These noise sources prevent the IMU bias estimation from falling asleep by ensuring a non-zero steady state gain in the Kalman filter's gain matrix. Without the added noise sources, the filter's estimated variance of the biases (contained in the state covariance matrix (P) would not increase during time updates. This will cause the estimated variance to reach some steady state value after so many measurement updates. Once this happens, the filter will not update the bias estimates. Since the biases in a IMU are know to drift over time, ideally, the estimated bias should be capable of changing over time to track the change in IMU bias. Adding artificial noise to the estimated biases will ensure that the estimated variance of the bias will also increase over time. This will allow the biases to change after measurement updates. The last two elements of the noise source vector ($\vec{v}_{13:14}$) are for the GPS receiver clock bias and drift. These values are only used when using GPS pseudorange and pseudorange rate measurements (closely coupled measurement structure).

2. Time Update

The measurements from the IMU can be used to propagate the filter's states between GPS, camera, and LiDAR measurements. The state equations are used to propagate the states forward in time. The IMU inputs ($\vec{u}$), current estimates, and state equations are used in conjunction with 4^{th} order Runge-Kutta integration to predict the value of the estimates at the next IMU input. $\vec{v}$ is the noise source vector and is ignored when integrating the states. Once the states are propagated forward in time, the state covariance matrix must be updated to reflect the new variances/covariance of the state estimates.

The equation below is used to update the state covariance matrix at each time update.

$$P = A_D P A_D^T + (BQB^T)dt$$

The IMU measurements are received at a discrete time. The equation above represents a discrete update of the state covariance matrix. The state matrix (A) must be discretized in order to use this equation. The discretization period (dt) represents the change in time between the time updates. A time update is conducted at every IMU measurement; therefore $dt = 1/f_{IMU}$ where f_{IMU} is the frequency of the IMU. The IMU used for this work had a 100Hz update rate so $dt = .01$. The noise input section of the equation (BQB^T) can be roughly discretized by multiplying each element of the product of BQB^T by dt.

The state matrix (A) is a Jacobian matrix. The state matrix is obtained by taking the partial derivative of each change in state equation with respect to each state.

$$a_{i,j} = \frac{\delta f_i(\vec{x}, \vec{u}, \vec{v})}{\delta \vec{x}_j}$$

The discrete version of the A matrix can be obtained using the equation below;

$$A_D = e^{(A)dt}$$

However, using this equation requires solving a matrix exponential at every time update. Another method of determining A_D is to discretize all of the state equations and repeat the Jacobian process using the discretized equations. This method would prevent the filter from solving a matrix exponential at every time update which greatly reduces the processing requirements.

The process noise input matrix is also a Jacobian matrix. Each element of the processes noise input matrix is obtained by taking the partial derivative of each change in state equations with respect to each noise source ($\vec{v}$).

$$b_{i,j} = \frac{\delta f_i(\vec{x},\vec{u},\vec{v})}{\delta \vec{v}_j}$$

Q is the process noise covariance matrix. The process noise covariance matrix is a diagonal matrix since the noise sources are assumed to be uncorrelated. The size of the process noise covariance is defined by the number of noise sources (12-14 depending on GPS measurement type). The diagonal elements of the process noise covariance matrix correspond to the variances of each process noise source. In order to be optimally tuned, the value of the i^{th} diagonal element of the process noise covariance should be set equal to the true variance of the i^{th} noise source in the process noise source vector ($\vec{v}$). The values for the variances of the accelerometer and gyro noise sources were estimated using static IMU data. The seventh through twelfth noise sources were added to allow for bias estimation. The values for the variance of these noise sources are used to tune the dynamics of the bias estimation. The last two noise sources are only present when GPS pseudorange and pseudorange rate measurements are being used. The values of the process noise covariance that correspond to these noise sources are used to tune the dynamics of the GPS clock errors estimates. Table 4 gives the values used for the diagonal elements of the process noise covariance matrix.

Table 4: Diagonal Elements of the Process Noise Covariance Matrix

i	Noise Source	$Q_{i,i}$	Units
1	x accelerometer	0.0005	$(m/s^2)^2$
2	y accelerometer	0.0005	$(m/s^2)^2$
3	z accelerometer	0.0005	$(m/s^2)^2$
4	x gyro	0.0000055	$(rad/s)^2$
5	y gyro	0.0000055	$(rad/s)^2$
6	z gyro	0.0000055	$(rad/s)^2$
7	x accelerometer bias	0.0001	$(m/s^2)^2$
8	y accelerometer bias	0.0001	$(m/s^2)^2$
9	z accelerometer bias	0.0001	$(m/s^2)^2$
10	x gyro bias	0.0000001	$(rad/s)^2$
11	y gyro bias	0.0000001	$(rad/s)^2$
12	z gyro bias	0.0000001	$(rad/s)^2$
13	clock bias	0.0001	m^2
14	clock drift	0.01	$(m/s)^2$

3. Measurement Update

A measurement update is conducted whenever a measurement of the states becomes available. With no measurement updates, the states would have constantly increasing error due to integration of the IMU error and biases. The measurement update uses measurements of the states to determine how the state estimates and state covariance matrix should be changed to account for the new measurement.

The extended Kalman filter measurement update is capable of using measurements that are non-linear with respect to states of the filter. In the extended Kalman filter, the Kalman gain is multiplied by the residual vector to determine the correction to the state estimates. The residual vector ($\vec{z}$) is a vector that represents the difference between the measurements vector ($\vec{y}$) and the estimated values of the measurement vector using the current states ($\hat{\vec{y}}$).

$$\vec{z} = \vec{y} - \hat{\vec{y}}$$

$\vec{y}$ is the measurement vector. The measurement vector is $mx1$ vector of measurements where m is the number of measurements. $\hat{\vec{y}}$ is the estimated measurement vector. This is a $mx1$ vector of estimated values of the measurements using the current states. Subtracting the estimated measurement vector from the measurement vector results in the residual vector ($\vec{z}$).

The estimated measurement vector is created by substituting the states into the measurement equations ($f_i(\vec{x})$).

$$\hat{\vec{y}} = \begin{bmatrix} f_1(\vec{x}) \\ \vdots \\ f_m(\vec{x}) \end{bmatrix}$$

The measurement equations are equations that are a function of the states and represents a mathematical description of the measurement. The measurement equations are used to predict the value measurement using the states of the filter. The equations are also used to construct the observation matrix (H).

K is the Kalman gain matrix. The equation below is used to compute the Kalman gain.

$$K = PH^T(HPH^T + R)^{-1}$$

The Kalman gain equation is a function of the current state noise covariance matrix (P), the observation matrix (H), and the measurement noise covariance matrix (R). H is the measurement model. It is a mxn Jacobian matrix where m is the number of measurements and n is the number of states. The equation below is used to calculate each element of the observation matrix.

$$h_{i,j} = \frac{\delta f_i(\vec{x})}{\delta \vec{x}_j}$$

R is the measurement noise covariance matrix. R is a diagonal matrix (assuming no cross correlation) that is mxm where m is the number of measurements. The diagonal element $r_{i,i}$ is set to the variance of the i^{th} measurement in the measurement matrix.

$$r_{i,i} = \sigma^2_{\vec{y}_i}$$

The Kalman gain matrix determines how much to change the states based on the variance (R) of the current measurements and the current state noise covariance matrix (P). Therefore, R acts as a weighting matrix that governs how much the states should be changed by the current set of measurements. Setting the diagonal values of the measurement noise covariance matrix to true variances of the current measurements will result in optimal performance of the extended Kalman filter.

The Kalman gain along with the observation model (H) and measurement residual vector ($\vec{z}$) are used to update the filter's states and state noise covariance matrix. The equation below is used to update the state vector.

$$\vec{x} = \vec{x} + K\vec{z}$$

The equation below is used to update the state noise covariance matrix.

$$P = (I_{nxn} - KH)P$$

Loosely Coupled GPS Measurement Update

The loosely coupled GPS measurement update uses position and velocity values provided by the GPS receiver to update the navigation filter. The GPS receiver provides estimated position and velocity in the ECEF coordinate frame. The measurement provided is a direct measurement of the first six states since the first six states of the filter are position and velocity in the ECEF coordinate frame. The receiver also provides estimated variance and covariance of both the position and velocity measurements.

The equation given below is the measurement vector for the loosely coupled GPS measurement update. The GPS position and velocity measurements are placed in this vector

$$y = \begin{bmatrix} \vec{r}^{e}_{eb} \\ \vec{v}^{e} \end{bmatrix}$$

The equation given below shows the measurement equations for the loosely coupled GPS measurement. The loosely coupled GPS measurement provides a direct measurement of the first six states of the filter.

$$\hat{\vec{y}} = \vec{x}_{1:6}$$

The equation below shows the H matrix for the loosely coupled GPS measurement. Since the measurements are direct measurements of the first six states, the H matrix is constructed with an $6x6$ identity matrix and a $6x9$ matrix of zeros.

$$H = [I_{6x6} \quad 0_{6x9}]$$

The diagonal values of the measurement noise covariance matrix (R) are set using the measurement variances provided by the GPS receiver.

$$R = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{vx}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{vy}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{vz}^2 \end{bmatrix}$$

If the GPS receiver gives the full position and velocity covariance matrices, then they can be added into the measurement noise covariance matrix,

$$R = \begin{bmatrix} R_{position} & 0_{3x3} \\ 0_{3x3} & R_{velocity} \end{bmatrix}$$

where $R_{position}$ is the position measurement covariance matrix and $R_{velocity}$ is the velocity measurement covariance matrix.

Closely Coupled GPS Measurement Update

The closely coupled GPS measurement update is a measurement update using the pseudorange and received signal frequency to all satellites in view of the receiver. The pseudorange is an estimated range to a GPS satellite. It is called a pseudorange (not a range) because the receiver's estimated range to a satellite contains a bias due to the receiver's clock error. The receiver's clock error is the difference in time between the receiver's clock's estimated time and the GPS system time. This bias is constant on all ranges; therefore, it can be estimated given enough pseudorange measurements. The estimated range rate using the received signal frequency also contains an error due to receiver clock drift, thus the estimated range rate is called pseudorange rate.

When using closely coupled GPS measurements, two extra states must be added to the state vector. The extra states are the clock bias and clock drift. The clock bias is the range bias (in meters) caused by the difference in the receiver's clock and the GPS system clock. The clock drift is the range rate bias (in meters per second) caused by the change in receiver's clock and the GPS system clock. Since these two biases are observed on all the observations, they can be estimated. However, since these values must be estimated, it takes four GPS satellites instead of just three to calculate three-dimensional position and velocity.

In order to use pseudorange and pseudorange rate measurements, the position and velocity the satellites from which observations were made must be known. The satellite's position and velocity can be computed using the satellite's ephemeris data. The ephemeris parameters are constant for several hours after first being broadcast. Therefore, the ephemeris only needs to be obtained on startup and when a new ephemeris has been issued. The satellite position and velocity for a particular observation are solved using the GPS time of the observation and ephemeris values. The ephemeris also contains a satellite clock correction term that must be multiplied by the speed of light and added to the measured pseudorange to account for errors in that particular satellite's clock. More on how the GPS ephemeris is obtained and how these values can be used to calculate satellite position, velocity, and clock correction can be found in GPS Interface Control Document available on-line.

The equation below shows the measurement vector for a measurement from one GPS satellite.

$$y = \begin{bmatrix} \rho_m + c(dt_{sv}) \\ c\dfrac{f_m}{L_1} \end{bmatrix}$$

The values in the measurement matrix are pseudorange and pseudorange rate. The GPS receiver provides an estimated pseudorange (with a satellite clock error) and received signal frequency to all the satellite signals received by the GPS receiver. The pseudorange must be corrected for the satellite clock error. The pseudorange rate can be calculated using the received signal frequency measurement.

The pseudorange must be corrected for the satellite clock errors. The satellite clock error should not be confused with the receiver clock error. The satellite clock error is the error between the satellite's clock and GPS system time. Like the satellite position and velocity, the satellite clock correction can be calculated as a function of time using the ephemeris values. The satellite clock correction is multiplied by the speed of light (meters/second) to determine the range correction (meter).

A range rate or closure speed can be calculated using the received signal frequency (f_m). The Doppler effect states that dividing the received signal frequency by the frequency of the signal at transmission (L_1) and then multiplying by the speed at which the signal travels (speed of light) results in an estimated range rate. The frequency of the L_1 GPS signal is 1.57542 GHz. Only GPS measurements from the L_1 signal were used for this work; however, if using measurements from the L_2 signal, the received signal frequency must be divided by the L_2 signal frequency (1.2276 GHz).

The equation below shows the measurement equations used to obtain the measurement prediction ($\hat{\vec{y}}$).

$$\hat{\vec{y}} = \begin{bmatrix} \hat{r} + \vec{x}_{16} \\ \hat{\dot{r}} + \vec{x}_{17} \end{bmatrix}$$

The pseudorange measurement is equal to the estimated range ($\hat{r}$) plus the receiver clock bias ($\vec{x}_{16}$). The receiver clock bias is the sixteenth state when using GPS pseudorange and pseudorange rate measurements. The clock bias is estimated in meters and represents the error (in meters) caused by the difference in the vehicle GPS receiver's clock and the GPS system time. The clock bias state can be divided by the speed of light to determine the actual clock difference in seconds. The estimated pseudorange rate measurement is equal to the estimated range rate ($\hat{\dot{r}}$) plus the receiver clock drift ($\vec{x}_{17}$). The receiver clock drift is the seventeenth state of the filter when using GPS pseudorange and pseudorange rate measurements. The clock drift estimate is estimated in meters per second and represents the rate of change of the receiver clock bias.

The equation below shows how to estimate the range to a satellite.

$$\hat{r} = \sqrt{(x_{sv} - \vec{x}_1)^2 + (y_{sv} - \vec{x}_2)^2 + (z_{sv} - \vec{x}_3)^2}$$

The equation is simply the distance formula and is a function of the position states ($\vec{x}_{1:3}$) and the position of the satellite ($[x_{sv} \quad y_{sv} \quad z_{sv}]$). The position of the satellite is calculated using the GPS time of the measurement and the ephemeris data for the satellite.

The equation below shows how to estimate the range rate to a satellite.

$$\hat{\dot{r}} = \vec{e} \begin{bmatrix} (\dot{x}_{sv} - \vec{x}_4) \\ (\dot{y}_{sv} - \vec{x}_5) \\ (\dot{z}_{sv} - \vec{x}_6) \end{bmatrix}$$

The equation is a function of the velocity states ($\vec{x}_{4:6}$) and the velocity of the satellite ($[\dot{x}_{sv} \quad \dot{y}_{sv} \quad \dot{z}_{sv}]$). The velocity of the satellite is calculated using the GPS time of the measurement and the ephemeris data for the satellite. The range rate equation is also a function of the unit vector that points from the satellite to the GPS antenna ($\vec{e}$). The difference in the three dimensional vehicle velocity and satellite velocity is rotated into the one dimensional axis that points from the satellite to the GPS antenna.

The unit vector that points from the satellite to the vehicle ($\vec{e}$) can be calculated using the state position estimates ($\vec{x}_{1:3}$) and the satellites position ($[x_{sv} \quad y_{sv} \quad z_{sv}]$). The equation below shows how to calculate the unit vector that points from the satellite to the vehicle.

$$\vec{e} = [(\vec{x}_1 - x_{sv})/\hat{r} \quad (\vec{x}_2 - y_{sv})/\hat{r} \quad (\vec{x}_3 - z_{sv})/\hat{r}]$$

Taking the partial derivative of the pseudorange measurement equation with respect to each position states results in a unit vector that points from the satellite to the vehicle. Similarly taking the partial derivative of the pseudorange rate measurement equation with respect to the velocity states results in a unit vector that points from the satellite to the vehicle. Thus, the unit vector is placed in the observation matrix (H) as shown below. The size of the observation matrix for each GPS observation is $2x17$. All the pseudorange and pseudorange rate measurements can be processed at the same time by stacking the H matrix show below.

$$H = \begin{bmatrix} \vec{e} & 0_{1x3} & 0_{2x9} & I_{2x2} \\ 0_{1x3} & \vec{e} & & \end{bmatrix}$$

$\vec{e}$ is the unit vector that points from the i^{th} satellite to the vehicle. Each observation is represented by two rows in the H matrix. The first row represents a pseudorange measurement, and the second row represents a pseudorange rate measurement. Four independent satellite observations are needed for the navigation filter to be fully observable. If more than four observations are available, the H matrix can be modified by adding two rows for each additional observation. If less than four observations are available, then the navigation filter will not be fully observable; however, a measurement update can take place using the available observations. The size of H will be $2mxn$ where m is the number of satellites observed and n is the number of states.

The equation below is used to estimate the noise present in a GPS pseudorange measurement based off the carrier to noise ratio (C/N_0) of the received GPS signal.

$$\sigma_\rho = K_\rho + \lambda_c \sqrt{\frac{4d^2 B_n}{C/N_0}\left(2(1-d) + \frac{4d}{T_s C/N_0}\right)}$$

The estimated standard deviation of the pseudorange is squared and placed into the diagonal element of the measurement noise covariance that corresponds to the pseudorange measurement. The K_ρ term is the unmodeled error term and represents all nonreceiver based error such as multipath and atmospheric delays. The remaining terms in the above equation represents the estimated error due to the receiver's delay lock loop.

The equation below is used to estimate the noise present in a GPS pseudorange rate measurement based on the carrier to noise ratio (C/N_0) of the received GPS signal.

$$\sigma_{\dot{\rho}} = \frac{f_e}{3} + \frac{\lambda}{2\pi T_s}\sqrt{\frac{4B_n}{C/N_0}\left(1 + \frac{1}{T_s C/N_0}\right)}$$

The estimated standard deviation of the pseudorange rate is squared and placed into the diagonal element of the measurement noise covariance that corresponds to the pseudorange rate measurement. The f_e term is the dynamic stress factor. The dynamic stress factor is dependent on the tracking loop bandwidth and order. The remaining terms in the equation above represents the estimated error due to the receiver's frequency lock loop. Table 5 shows the values used for equations above.

Table 5: Tracking Loop Parameters

Parameter	Value	Units
B_n, code loop noise bandwidth	2	Hz
d, correlator spacing	0.5	Chips
f_e, dynamic stress error	3	m/s
K_ρ, unmodeled range error	5	m
λ, carrier wavelength	0.1902	m
λ_c, code chip width	293.5	m
T_s, predetection integration time	0.005	s

The estimated pseudorange and pseudorange rate standard deviations are squared and placed into measurement covariance matrix R as shown below.

$$R = \begin{bmatrix} \sigma^2_{\rho,1} & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{\dot{\rho},1} & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \sigma^2_{\rho,m} & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{\dot{\rho},m} \end{bmatrix}$$

The measurement covariance matrix is a $2mx2m$ diagonal matrix where m is the number of satellite observations. R is a diagonal matrix because the noise sources are assumed to be non-correlated. This assumption is valid because the measurements are originating from different satellites; therefore, the error in one pseudorange/pseudorange rate measurement has no correlation with the error in other observations.

This section has presented the structure of the navigation filter used for this project. The navigation filter tracks the pose (position, velocity, and attitude) of the vehicle along with the three IMU accelerometer biases and the three IMU gyro biases. The filter also estimates the receiver clock bias and drift if necessary. The navigation filter is split into time updates and measurement updates. Time updates are conducted at every IMU measurement. The IMU measurements are used to propagate the states (along with the state covariance matrix) forward in time. The measurement update is conducted whenever a measurement of the states is available. GPS measurements are used to update the states (and state covariance matrix). The GPS measurements can either be loosely or closely coupled. The loosely coupled measurement is a measurement of position and velocity provided by the GPS receiver. The closely coupled measurement is a measurement pseudorange and received signal frequency from each visible GPS satellite. The filter is constructed to work with either measurement type. The filter can use loosely coupled GPS measurement until the number of visible satellite falls below four. When the number of GPS observation falls below four, the GPS receiver can no longer compute a position or velocity; however, the navigation filter can then switch over to closely coupled GPS measurements. The closely coupled measurement allows from some additional measurements when the GPS receiver can not compute a solution. The closely coupled measurement structure is also advantageous because the noise sources with the pseudorange and received signal frequency measurements are uncorrelated between GPS observations. This means the assumption that the measurement noise covariance matrix R is diagonal is more correct for the closely coupled measurement update. The report provides a method of constructing the R matrix for the loosely coupled measurement update that reflects the cross covariances between the position and velocity measurements (provided the receiver provides the cross covariance terms of the estimated position and velocity).

B. Vision Measurement Updates Using a Map

The vision measurements, either from the LiDAR or camera, give an estimated lateral position in the current lane. The road coordinate frame is the coordinate frame in which the vision measurements are given. The road coordinate frame can be approximated using a waypoint map. For this work, each lane is assumed to have its own associated lane map. The waypoints lie in the center of the mapped lane, where the distance between waypoints is defined by the road geometry. Complex road geometry will require waypoints to be close together. For example, in a turn, the waypoints will need to be close enough to capture the geometry of the road. On a straightaway, the waypoints can be very spread out due to the lack of change in road geometry.

The road coordinate frame is a three dimensional coordinate frame and is denoted by the sub/superscript r. The x-axis of the road coordinate frame points from the last waypoint passed to the next waypoint. The y-axis of the road coordinate frame is perpendicular to the x-axis. If facing the direction of the road, the y-axis points to the right. The road coordinate frame is assumed to have no bank; therefore, the y-axis is always parallel with the plane tangent to earth's reference ellipsoid. The z-axis is perpendicular to the x-y plane and points down. This type of coordinate frame was chosen because the vision measurements give a direct measurement of the y-axis position. Also, since this work is for ground vehicles, the position in the z-axis of the road coordinate frame is fixed, can be measured once, and is added in as a measurement update. The change in vertical height on any point on the vehicle is a function of the roll angle of the vehicle and the height of the point above the roll axis; however, this angle and height above the roll axis is small for normal operating conditions. Figure 38 shows a visual representation of the road coordinate frame.

The figure below shows a visual representation of the road coordinate frame.

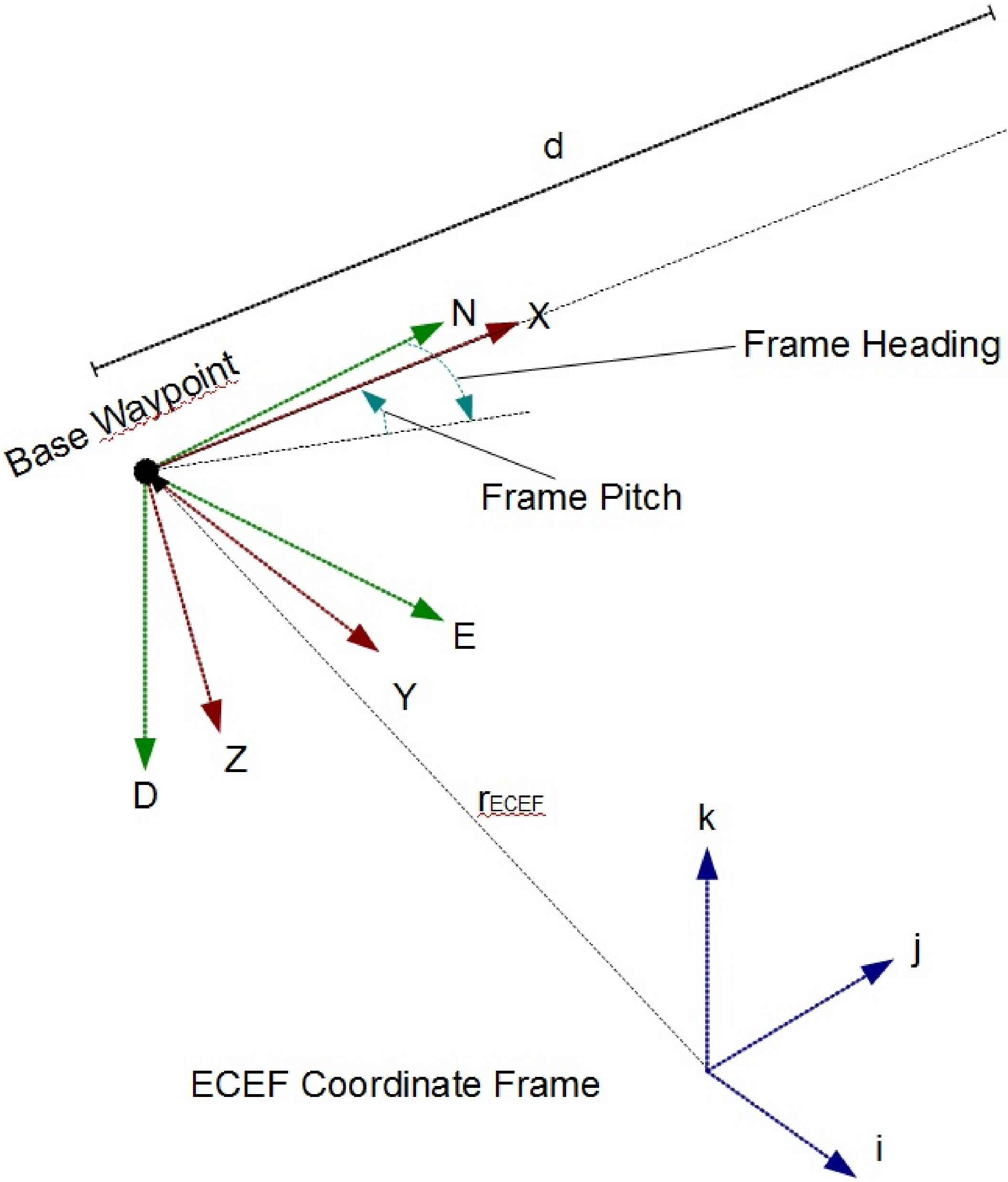

Figure 38: Road Coordinate Frame

In order to use measurements given in the road coordinate frame, the estimated position in the navigation (ECEF) coordinate frame must be mapped into the road coordinate frame. The equation below is used to map position from the navigation coordinate frame to the road coordinate frame.

$$\vec{r}^{\,r}_{rb} = C^r_e[\vec{r}^{\,e}_{eb} - \vec{r}^{\,e}_{er}]$$

$\vec{r}^{\,e}_{eb}$ is the estimated position vector in the navigation coordinate frame and $\vec{r}^{\,r}_{rb}$ is the estimated position vector in the road coordinate frame. In order to use this equation, the position of the origin of the road coordinate frame must be known ($\vec{r}^{\,e}_{er}$). The position of the origin of the road coordinate frame is the last waypoint the vehicle passed in the waypoint map. The rotation matrix from the navigation coordinate frame to the road coordinate frame (C^r_e) must also be known. In order to construct this rotation matrix, the attitude of the road coordinate frame with respect to the navigation coordinate frame must be known and is represented by three Euler angles. More information of how rotation matrix and set up and used can be found in Appendix B.

1. Waypoint Based Lane Map

The information needed to use the lane map is the position of each waypoint in the ECEF coordinate frame, the three Euler angles necessary to rotate the ECEF coordinate frame to the road coordinate frame for each waypoint, and the distance to the next waypoint for each waypoint. Therefore, the map database takes the form:

$$Map\ Database = \begin{bmatrix} \vec{r}^{\,e}_{er,1} & \vec{\varphi}_1 & d_{r,1} \\ \vdots & \vdots & \vdots \\ \vec{r}^{\,e}_{er,m} & \vec{\varphi}_m & d_{r,m} \end{bmatrix}$$

where $\vec{r}^{\,e}_{er,i}$ is the position vector of waypoint i in the ECEF coordinate frame, and $\vec{\varphi}_i$ is the attitude vector of the road coordinate frame. Like the attitude states of the vehicle, the attitude of the road coordinate frame is represented with three Euler angles. $d_{r,i}$ is the distance from waypoint i to waypoint $i+1$. This value is used to check if the vehicle has passed the next waypoint in order to keep up with the vehicles location along the map.

The form of the rotation matrix from the ECEF coordinate frame to the road coordinate frame is given below.

$$C^r_e = \begin{bmatrix} c_2c_3 & c_2s_3 & -s_2 \\ s_1s_2c_3 - c_1s_3 & s_1s_2s_3 + c_1c_3 & s_1c_2 \\ c_1s_2c_3 + s_1s_3 & c_1s_2s_3 - s_1c_3 & c_1c_2 \end{bmatrix}$$

The elements from the road coordinate frame attitude ($\vec{\varphi}_i$) are used to construct the rotation matrix. The rotation matrix from ECEF to the road coordinate frame and the position of the road coordinate frame (position of the last waypoint passed) can be used to solve for estimated position in the road coordinate frame, where c_1 is the cosine of the first attitude angle in the attitude vector and s_1 is the sine of the first attitude angle in the attitude vector. Similarly, c_2 and s_2 are the trigonometric functions of the second angle, and c_3 and s_3 are the trigonometric functions of the third angle.

The next three sections discuss how to obtain the information needed for the waypoint map. The first section shows how to obtain the waypoint positions ($\vec{r}_{er,1}^{e}$). The next section covers how to obtain the three Euler angles needed to describe the attitude of the road coordinate frame ($\vec{\varphi}_i$). The third section shows how to obtain the distance between waypoints ($d_{r,i}$).

Determining Waypoint Position

Differential GPS provides a very accurate baseline between the base station antenna and the receiver's antenna; however, global position and the accuracy of the global position are based off the base station antenna's position and the accuracy to which the base station antenna is known. For example, if the track is surveyed with differentially corrected GPS with corrections provided by a different base station than the one that will be used by GPS used to determine lane position, then a bias will be present in the lane position estimate. This bias will be a function of the errors in the base station's global position. Therefore, if both base stations are perfectly known, this bias will not be present; however, any error in either of the base stations position will result in a bias lane estimation. Surveying a specific location that can be found and surveyed at a later date will ensure that any global error in the survey base station is eliminated. Surveying this location also provides a method of using the lane survey in conjunction with a base station that is not precisely known in order to determine lane position. This map-marker can be used to determine the amount the map must be shifted to work with the different base station. The map-marker should be in a location with a good visibility of the sky, and it should be marked with something that can be easily found at a later date. This will ensure that the map can be used with any base station instead of just the base station used for the survey.

The figure below shows how the local waypoint position vectors are saved.

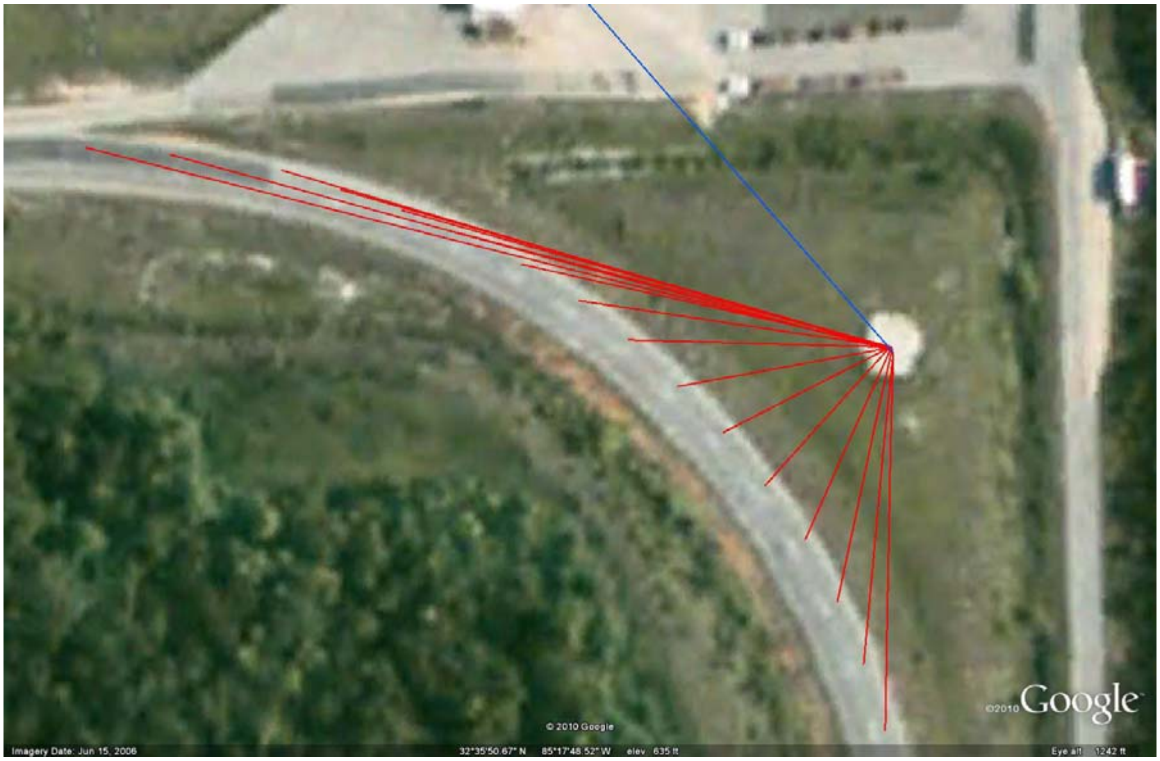
© 2010 Google
©2010 Google
Imagery Date: Jun 15, 2006
32°35'50.67" N 85°17'48.52" W elev 635 ft
Eye alt 1242 ft

Figure 39: Map Baselines

Surveying the road with differential GPS will ensure the waypoints of the lane map are very accurate with respect to the map-marker location. An accurate position vector between waypoints and the marker location can be solved by subtracting the global position of the marker location from the global positions of each lane map waypoint (represented by the red lines in Figure 39). These vectors can be saved as the actual map survey. Saving the base-line vectors eliminates any global component to the survey.

When a base station is set up to broadcast corrections used to determine lane position, the position vectors in the map survey can be added to the global solution computed at the map-marker using corrections from the new base station (represented by the blue line in Figure 39). The NCAT test track was surveyed using a survey GPS system providing accuracy on the centimeter level. The test track also has a GPS base station that provides RTK (Real Time Kinematic) corrections. The survey of the track was conducted using GPS corrections provided by a different base station than the corrections used by the vehicle's onboard GPS receiver. For this reason, the map had to be aligned to the different base stations. The shift in the map (global position of map) can be determined by surveying the map's marker position using the same corrections used by the vehicle's GPS receiver.

In order to ensure that RTK differential GPS alone can provide lane level position, the lane map must be very accurate. Any inaccuracies in the lane map will result in a biased lane position. If vision measurements are available, the filter will still produce a nonbiased lane position; however, if vision measurements become unavailable, the navigation filter will report a biased lane position. If the lane map is not accurate to the centimeter level, then any benefit from using RTK differential GPS is lost. An inaccurate lane map will cause differential GPS measurements to resemble standalone GPS measurements when compared to the lane map.

Determining Euler Angles between Global Axis and Road

The rotation matrix from the ECEF coordinate frame to the road coordinate frame:

$$C_e^r = \begin{bmatrix} c_2c_3 & c_2s_3 & -s_2 \\ s_1s_2c_3 - c_1s_3 & s_1s_2s_3 + c_1c_3 & s_1c_2 \\ c_1s_2c_3 + s_1s_3 & c_1s_2s_3 - s_1c_3 & c_1c_2 \end{bmatrix}$$

can be thought of as a sequence of two rotations.

$$C_e^r = C_n^r C_e^n$$

The first rotation is from the ECEF coordinate frame to the North, East, Down (NED) coordinate frame (C_e^n). This rotation matrix is based off two angles. ϕ_i is the latitude of the i^{th} coordinate frame, and λ_i is the longitude of the i^{th} coordinate frame. The origin of the NED coordinate frame is at the i^{th} waypoint ($\vec{r}_{er,i}^{e}$). The second rotation is from the NED coordinate frame to the road coordinate frame (C_n^r). This rotation matrix is also based off two angles; ψ_i is the heading of the i^{th} coordinate frame, and θ_i is the pitch of the i^{th} coordinate frame.

C_e^n maps coordinates in the ECEF coordinate frame to the NED coordinate frame. The longitude and latitude of the i^{th} road coordinate frame correspond to the global position of the i^{th} waypoint. The latitude and longitude can either be surveyed, or solved for using the position of the i^{th} waypoint. The equation below shows the rotation matrix from ECEF coordinate frame to the NED coordinate frame.

$$C_e^n = \begin{bmatrix} -\sin(\phi_i)\cos(\lambda_i) & -\sin(\phi_i)\sin(\lambda_i) & \cos(\phi_i) \\ -\sin(\lambda_i) & \cos(\lambda_i) & 0 \\ -\cos(\phi_i)\cos(\lambda_i) & -\cos(\phi_i)\sin(\lambda_i) & -\sin(\phi_i) \end{bmatrix}$$

C_n^r maps coordinates in the NED coordinate frame to the road coordinate frame. The pitch and heading angles can be solved by looking at the change in position between waypoint i and waypoint $i+1$. The equation below shows the rotation matrix from ECEF coordinate frame to the NED coordinate frame.

$$C_n^r = \begin{bmatrix} \cos(\theta_i)\cos(\psi_i) & \cos(\theta_i)\sin(\psi_i) & -\sin(\theta_i) \\ -\sin(\psi_i) & \cos(\psi_i) & 0 \\ \sin(\theta_i)\cos(\psi_i) & \sin(\theta_i)\sin(\psi_i) & \cos(\theta_i) \end{bmatrix}$$

The first step in solving for the road heading and pitch angles is to express the position of the $i+1$ waypoint in the NED coordinate frame based at the i^{th} waypoint. This can be done using the equation below.

$$\begin{bmatrix} N_{r,i} \\ E_{r,i} \\ D_{r,i} \end{bmatrix} = C_e^n \left[\vec{r}^{\,e}_{er,i+1} - \vec{r}^{\,e}_{er,i} \right]$$

Once the position of waypoint $i+1$ is expressed in the NED coordinate frame with the origin located at waypoint i, then the road coordinate frame heading and pitch can be calculated as fallows.

$$\psi_i = arctan2(E_{r,i}, N_{r,i})$$

$$\theta_i = arctan\left(\frac{-D_{r,i}}{\sqrt{N_{r,i}^2 + E_{r,i}^2}} \right)$$

The roll of the road coordinate frame is assumed to be zero; however, knowledge of this roll angle could be incorporated to more accurately model the road coordinate frame in areas where the road is banked.

The longitude (λ_i), latitude (ϕ_i), pitch (θ_i), and heading (ψ_i) are all substituted into their corresponding rotation matrices. This will result in a rotation matrix that maps coordinates in the ECEF coordinate frame to the road coordinate frame. The three attitude angles in the map database ($\vec{\varphi}_i$) can be extracted from this rotation matrix. The following set of equations show how to solve for the road coordinate frame attitude using the longitude, latitude, pitch, and heading angles.

$$\vec{\varphi}_{i,1} = arctan2(-\cos(\phi_i)\sin(\psi_i), \cos(\phi_i)\sin(\theta_i)\cos(\psi_i) - \sin(\phi_i)\cos(\theta_i))$$

$$\vec{\varphi}_{i,2} = arcsin(\cos(\phi_i)\cos(\theta_i)\cos(\psi_i) + \sin(\phi_i)\sin(\theta_i))$$

$$\begin{aligned} \vec{\varphi}_{i,3} = arctan2(&-\sin(\phi_i)\sin(\lambda_i)\cos(\theta_i)\cos(\psi_i) + \cos(\lambda_i)\cos(\theta_i)\sin(\psi_i) \\ &+ \cos(\phi_i)\sin(\lambda_i)\sin(\theta_i), -\sin(\phi_i)\cos(\lambda_i)\cos(\theta_i)\cos(\psi_i) \\ &- \sin(\lambda_i)\cos(\theta_i)\sin(\psi_i) + \cos(\phi_i)\cos(\lambda_i)\sin(\theta_i)) \end{aligned}$$

Determining Distance between Waypoints

The distance between the waypoints can be determined using the distance equation. This value is included in the lane map database for use by the algorithm that keeps track of the vehicles location within the lane map.

$$d_{r,i} = \sqrt{\left(\vec{r}^{e}_{er,i+1} - \vec{r}^{e}_{er,i}\right)^T \left(\vec{r}^{e}_{er,i+1} - \vec{r}^{e}_{er,i}\right)}$$

2. Vision Measurement Update

Vision Measurement Model

In order to use lateral lane position measurements, the lateral lane position with respect to the lane map must be estimated using the current states of the navigation filter. The equation below is used to find the position estimates in the road coordinate frame.

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = C^r_e\left[\vec{r}^{e}_{eb} - \vec{r}^{e}_{er,i}\right]$$

$\vec{r}^{e}_{eb}$ denotes the current position estimate in the ECEF coordinate frame. $\vec{r}^{e}_{er,i}$ denotes the position of the last waypoint passed in the ECEF coordinate frame. $\hat{y}$ is the lateral lane position estimate, $\hat{x}$ is the distance into the current road coordinate frame, and $\hat{z}$ is the vertical position in the current road coordinate frame. The z-axis of the road coordinate frame points down. C^r_e is the rotation matrix from the ECEF coordinate frame to the road coordinate frame and has the form:

$$C^r_e = \begin{bmatrix} c_2c_3 & c_2s_3 & -s_2 \\ s_1s_2c_3 - c_1s_3 & s_1s_2s_3 + c_1c_3 & s_1c_2 \\ c_1s_2c_3 + s_1s_3 & c_1s_2s_3 - s_1c_3 & c_1c_2 \end{bmatrix}$$

The three Euler angles that are substituted into C^r_e come from the map database ($\vec{\varphi}_i$) along with the position of the last waypoint passed ($\vec{r}^{e}_{er,i}$). The first three states of the filter ($\vec{x}_{1:3}$) is the position vector of the vehicle in the ECEF coordinate frame ($\vec{r}^{e}_{eb}$).

In order to perform a measurement update, three things must be known. The first is the measurement vector ($\vec{y}$). The equation below shows the measurement vector.

$$\vec{y} = \begin{bmatrix} y_{vision} \\ y_{height} \end{bmatrix}$$

The first measurement is lane position measured by the LiDAR or camera. The second measurement is the height above the lane. Since this work deals with ground vehicles, the height above the road (ground) can be measured once and assumed to be constant. This constant height is measured and included as a measurement.

The second requirement needed to perform a measurement update is the measurement equations. The measurement equations for the y and z axis of the road coordinate frame are given below.

$$\vec{\hat{y}}(\vec{x}, \vec{r}^{e}_{er}, \vec{\varphi}_i) = \begin{bmatrix} \hat{y} \\ \hat{h} \end{bmatrix}$$
$$= \begin{bmatrix} C^{r}_{e\,(2,1)}\left(\vec{r}^{e}_{eb,1} - \vec{r}^{e}_{er,i,1}\right) + C^{r}_{e\,(2,2)}\left(\vec{r}^{e}_{eb,2} - \vec{r}^{e}_{er,i,2}\right) + C^{r}_{e\,(2,3)}\left(\vec{r}^{e}_{eb,3} - \vec{r}^{e}_{er,i,3}\right) \\ -C^{r}_{e\,(3,1)}\left(\vec{r}^{e}_{eb,1} - \vec{r}^{e}_{er,i,1}\right) - C^{r}_{e\,(3,2)}\left(\vec{r}^{e}_{eb,2} - \vec{r}^{e}_{er,i,2}\right) - C^{r}_{e\,(3,3)}\left(\vec{r}^{e}_{eb,3} - \vec{r}^{e}_{er,i,3}\right) \end{bmatrix}$$

The measurement equations are a function of the states of the filter and the map parameters. The states and map parameters are plugged into these equations to estimate the measurement based off the current states. These equations are also used to determine the measurement model matrix (H). The first measurement equation is for the lateral lane position. This measurement is provided by a camera or LiDAR. The second measurement equation is the height above the road coordinate frame.

The last thing needed to perform a measurement update is the measurement model matrix (H). The measurement model is created by taking the partial derivative of the measurement equations with respect to each state. The measurement model matrix is shown below,

$$H = \begin{bmatrix} e_1 & 0_{1x3} & 0_{1x3} & 0_{1x3} & 0_{1x3} & 0_{1x3} & 0_{1x2} \\ e_2 & 0_{1x3} & 0_{1x3} & 0_{1x3} & 0_{1x3} & 0_{1x3} & 0_{1x2} \end{bmatrix}$$

where e_1 is the unit vector in the navigation coordinate frame that points in the direction of the y axis of the road coordinate frame

$$e_1 = [C^{r}_{e\,(2,1)}, C^{r}_{e\,(2,2)}, C^{r}_{e\,(2,3)}]$$

and e_2 is the unit vector in the navigation coordinate frame that points in the direction of the negative z axis of the road coordinate frame

$$e_2 = [-C^{r}_{e\,(3,1)}, -C^{r}_{e\,(3,2)}, -C^{r}_{e\,(3,3)}]$$

Both of these unit vectors are taken directly from rows of the rotation matrix from the navigation coordinate frame to the road coordinate frame.

The height and lane position measurements provide measurements in two of the three position axis. This means the system is almost fully observable just using these measurements. The only missing piece of information is position in the x axis of the road coordinate frame. If this value is measured (using something other than GPS), then the navigation filter will remain fully observable even without GPS measurements. The results also show that when using vision measurement updates, the navigation filter remains observable even if only two GPS ranges are available (instead of the typical four needed to maintain observabilty).

Map Keeping

In order to use vision updates, the navigation filter must know what map to use and where along this map the vehicle is located. Determining a vehicle's location within a certain map is easily done by searching through the map to find the closest waypoints. Determining which map a vehicle currently resides is a much more difficult problem. For example, when initializing map location on road with multiple lanes of travel in each direction, the global position used to search the maps has to be level accurate as to not cause any ambiguity as to what lane the vehicle currently resides. Standalone GPS has meter level biases; therefore, there would be no way to determine the vehicle current lane on a multi lane road. Map intersections could also cause ambiguity as to which map the vehicle is currently traveling in. Also, once the vehicle's lane is determined, the navigation filter will have to successfully track all lane changes in order to maintain the vehicle's lane map location. RTK differential GPS is considered to be accurate on the centimeter level. Therefore, if available, RTK differential GPS position can be used to initialize location on a multi-lane road with a correct guess at what lane map to use.

Determining which map the vehicle is currently residing poses an interesting problem. One method of initializing the map is to do a distance calculation from the estimated GPS position to all nearby waypoints from all nearby maps. The waypoint that is closest to the vehicle is assumed to be the last waypoint the vehicle passed, thus, the current map is set to the map containing the closest waypoint and map index is set to the index corresponding to the closest waypoint to the vehicle. This method searching every waypoint in every lane map would not be the most efficient method of finding the closest waypoint. Furthermore, there would also be an issue with how the search algorithm would continuously search for when the vehicle entered map coverage. Once entering a mapped area, accuracy of current global position becomes important in properly estimating which the lane vehicle entered at the start of a multi-lane map.

After the map and map index is selected, the distance into the current road coordinate frame is calculated using the equation below to determine if the closest waypoint is in front or behind the vehicle.

$$\widehat{x} = C^{r}_{e\,(1,1)}\left(\vec{r}^{\,e}_{eb,1} - \vec{r}^{\,e}_{er,i,1}\right) + C^{r}_{e\,(1,2)}\left(\vec{r}^{\,e}_{eb,2} - \vec{r}^{\,e}_{er,i,2}\right) + C^{r}_{e\,(1,3)}\left(\vec{r}^{\,e}_{eb,3} - \vec{r}^{\,e}_{er,i,3}\right)$$

If the closest waypoint is in front of the vehicle (the result of the above equation is negative), the map index should be shifted down until the result of the equation above is positive.

After every measurement update and time update, the states of the navigation filter will change. Every time the states of the filter are updated, the distance into the current road frame must be checked to ensure that the vehicle has not passed into the next road coordinate frame. Substituting the position states into the above equation will result in the current longitudinal distance into the road coordinate frame. If this value is larger than $d_{r,i}$ (from map database), then the vehicle has passed into the next coordinate frame.

The map index i should then be incremented by one. It may be necessary to increment the map index by more than one if the vehicle passes through a section of the map with no update. This problem becomes more likely if the waypoints are very close together and can be overcome by using a continuous check that will increment the map index until the distance estimated by the equation given above is less than $d_{r,i}$. In order to improve the robustness of the filter, the map checking system is also set up to decrease the map index if the result of the equation is less than zero.

Once the vehicle passes the last waypoint in a map, then the vehicle is assumed to have left the area covered by the current map. Once the vehicle leaves the map, it can no longer use lane position measurements to update the filter. Without the map, there is no way to incorporate lane position measurements into the global navigation filter. The vision measurements can continue to operate and estimate lane position; however, this information cannot be used to update the navigation filter's global position. As long as GPS remains available, the navigation filter will continue to operate even with no map. The position estimated by the filter can be used to determine when the vehicle has entered an area that has map coverage and reinitialize the map.

Information on what to do once the vehicle leaves the map can be saved in the map database. For example, each waypoint could also have an associated lane map for right and left lane departures. This would allow the filter to know if there was a lane to the right or left and what lane map to hand off to if the vehicle changes lanes. The map used for this work was circular. Therefore, the map tells the filter to simply go back to the start of the map once it reaches the end. For non-circular paths, the current map can contain information on what map should be used next after leaving the current map.

This section presents a method of using lane position measurements to update a global navigation filter using a waypoint based map. The map database is not considered to have any standard form. The database must contain the position of the waypoints in the navigation filter's coordinate frame. The waypoint positions is the only necessary information; however the rotation between the ECEF coordinate frame and the road coordinate frame must also be known in order to use the vision measurements. The waypoints can be used to determine the rotation between the ECEF and the road coordinate frame; however, this requires quite a bit of computational overhead which can be reduced by either saving the three Euler angles or every element of the rotation matrix (C_e^r) itself in the map database.

The method of using measurements given in a coordinate frame of the navigation coordinate frame assumes the position of the measurement coordinate frame contains no error. This means the filter assumes the lane map is perfectly accurate. If the map is not perfectly accurate, the filter will continue operate. However, lane position accuracy will be maintained and the global position estimates will be biased. This works similarly with any error in RTK GPS base station’s global position estimate. Any error in the global position of the base station's GPS antenna will cause a bias in RTK GPS measurements with respect to the lane map. This report presents a method of overcoming both of these biases given a survey marker location (assuming all waypoints and the survey marker location are centimeter accurate in relation to each other).

This report also presents a primitive method of map tracking. Using the map and vision measurements hinges on knowing the vehicle's current lane (or which lane map to use) and the vehicles current location within the lane map (determining the proper map index to use). The area of lane tracking shows the strongest potential in system improvement.

C. Results

This section presents results from the navigation filter and an evaluation of the performance of the filter. Along with showing the filter working at full capacity, various sensor outages are simulated to show how the loss of a particular sensor affects the filter's estimates. The results also show how the filter reacts when no lane map is available. Finally, the results will show how the filter performs when less than four GPS satellites are available, and the result of losing vision measurements or the lane map under a limited satellite constellation.

The figure below shows the estimated position in the north-east coordinate frame when using differential GPS and vision to update the navigation filter.

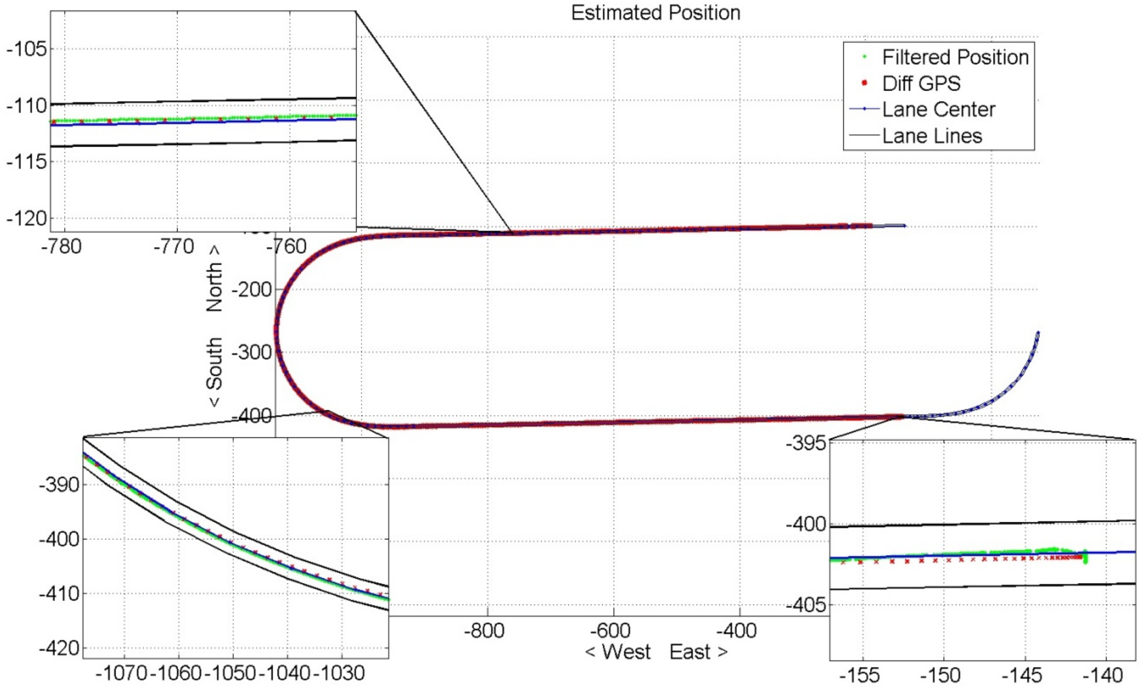

Figure 40: Estimated Vehicle Position Using Loosely Coupled RTK GPS Updates

All the error plots where created using the unfiltered RTK GPS position and velocities from the Septentrio GPS receiver. The RTK corrections were provided by an onsite base station. The close proximity to the base station ensures that RTK GPS position is accurate on a centimeter level. These RTK positions and velocities (taken at $10Hz$) were resampled to $100Hz$ to match the navigation filter's output rate. The error was estimated by differencing the resampled RTK GPS measurements and the navigation filter's estimates.

The data for the following results was collected at the NCAT test track in Opelika, Alabama. For this data set, the vehicle starts in the outside lane at the end of the front (north) straight. The test vehicle drives down the front straight at 55 mph, through a 180 degree turn, then down the back straight at 70 mph. The vehicle comes to a complete stop at the end of the back straight.

Figure 40 shows the filter's estimated position when using differential GPS measurements. The red dots represent the GPS measurements (at 10 Hz). As can be seen, the GPS measurements reside close to the center of the lane. The green dots represent the filtered solution (at 100 Hz) with differential GPS, vision, and map constraints.

The figure below shows the estimated position in the north-east coordinate frame when using standalone GPS and vision to update the navigation filter. The figure also shows the GPS measurements. The standalone GPS measurements are biased and lie outside of the lane; however, the vision measurements insure that the filter's estimated position lies inside the lane.

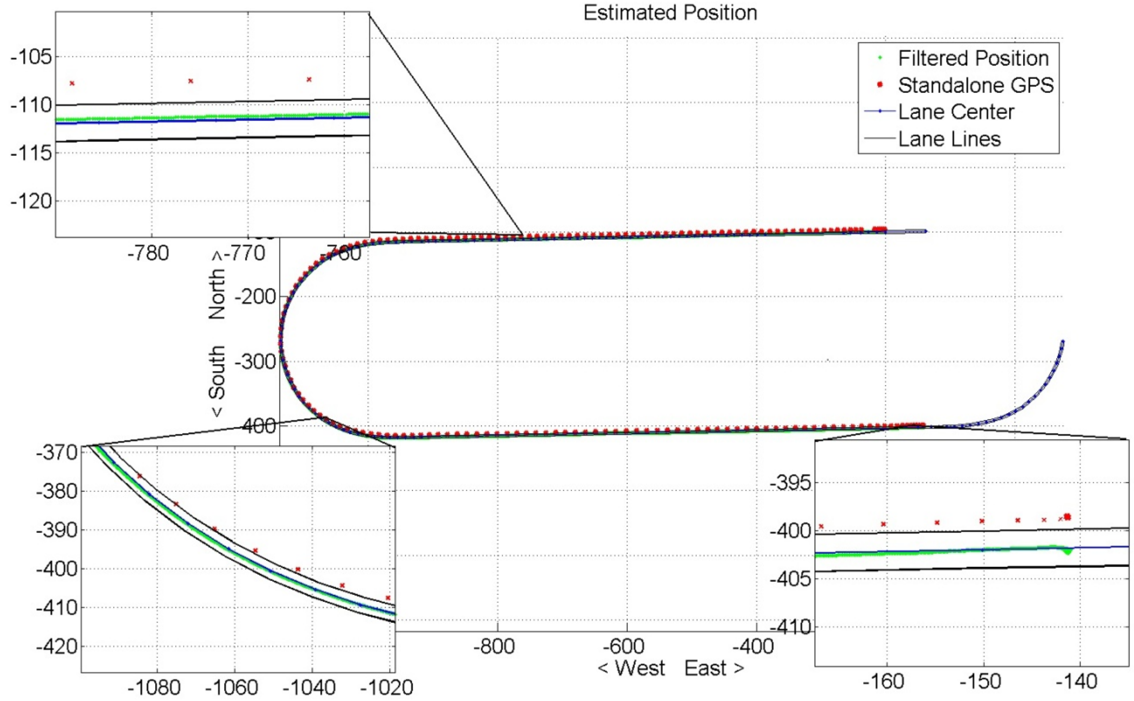

Figure 41: Estimated Vehicle Position Using Closely Coupled GPS Updates

Figure 41 shows the filter's estimated position when using standalone GPS measurements. The red dots represent the GPS measurements (at 2 Hz). As can be seen, the GPS measurements no longer reside within the lane lines. This is due to fact that standalone GPS has a drifting bias. The green dots represent the filtered solution (at 100 Hz) with standalone GPS, vision, and map constraints. The filtered solution resides within the lane lines. This is due to the fact that the vision measurement constrains the position to the map. This shows that GPS alone is not enough to determine lane position; however, fusing the GPS with vision and map constrains results in a solution that is lane level accurate. If the vision measurements are unavailable, the filtered solution will converge to the GPS measurements. This will result in a loss of lane level accuracy.

The figure below shows the filter's estimated roll and pitch, and the roll and pitch reported by the triple antenna Septentrio GPS receiver.

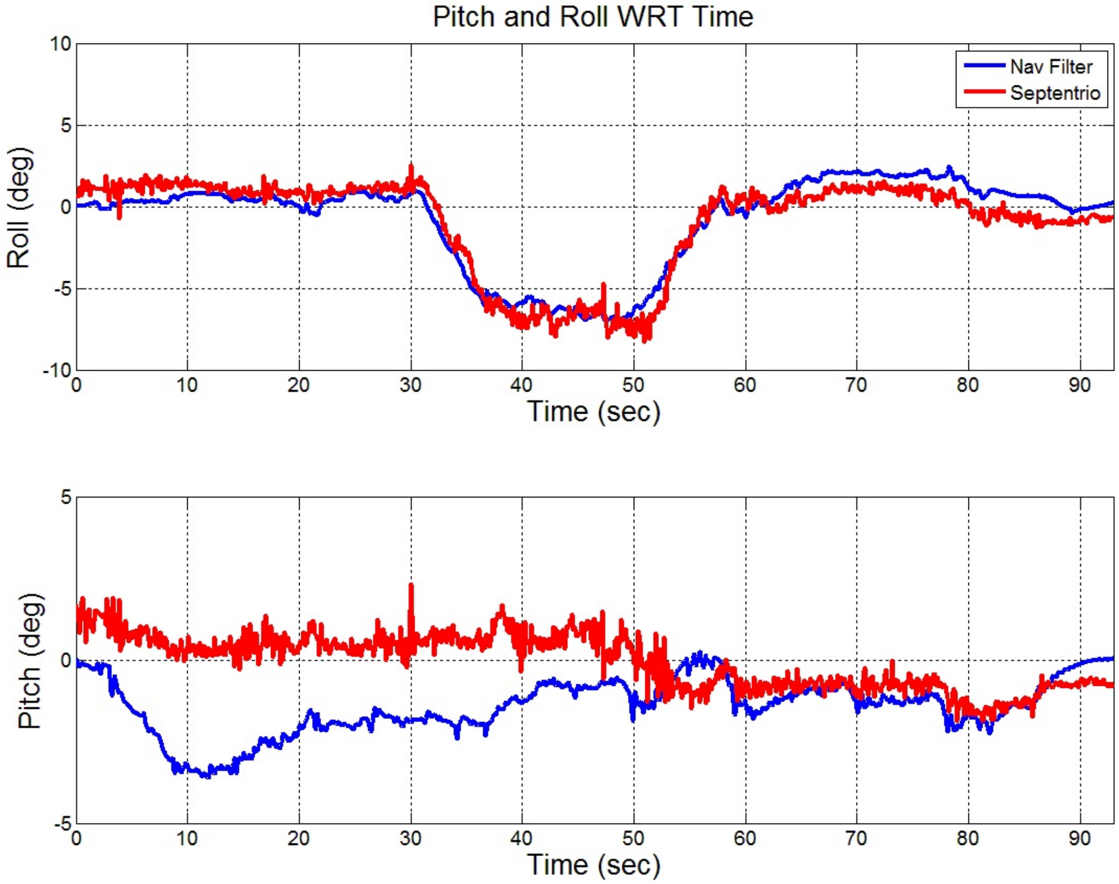

Figure 42: Estimated Vehicle Roll and Pitch

Figure 42 and Figure 43 provide the filter's estimated attitude and the Septentrio GPS receiver's estimated attitude. The Septentrio GPS receiver is a three antenna receiver that accurately measures attitude. The plot shows that the attitude estimated by the filter using a single antenna closely matches the attitude estimated by the triple antenna GPS receiver. The stability of the navigation filter is very dependent on the accuracy of the initial guess at the vehicles' attitude. For this data set, the initial pitch and roll was set to zero, and the initial heading was set at -90 degrees. The actual initial heading of the vehicle was around -92 degrees. This results in a two degree heading error until the vehicle starts to turn. Attitude estimates are unobservable without significant excitation (change in attitude). If the initial guess at the attitude is wrong, the filter can behave in an unstable manner.

The figure below shows the filter's estimated heading and the heading error. The heading error represents the difference between the filter's estimated heading and the triple antenna Septentrio GPS receiver' measured heading.

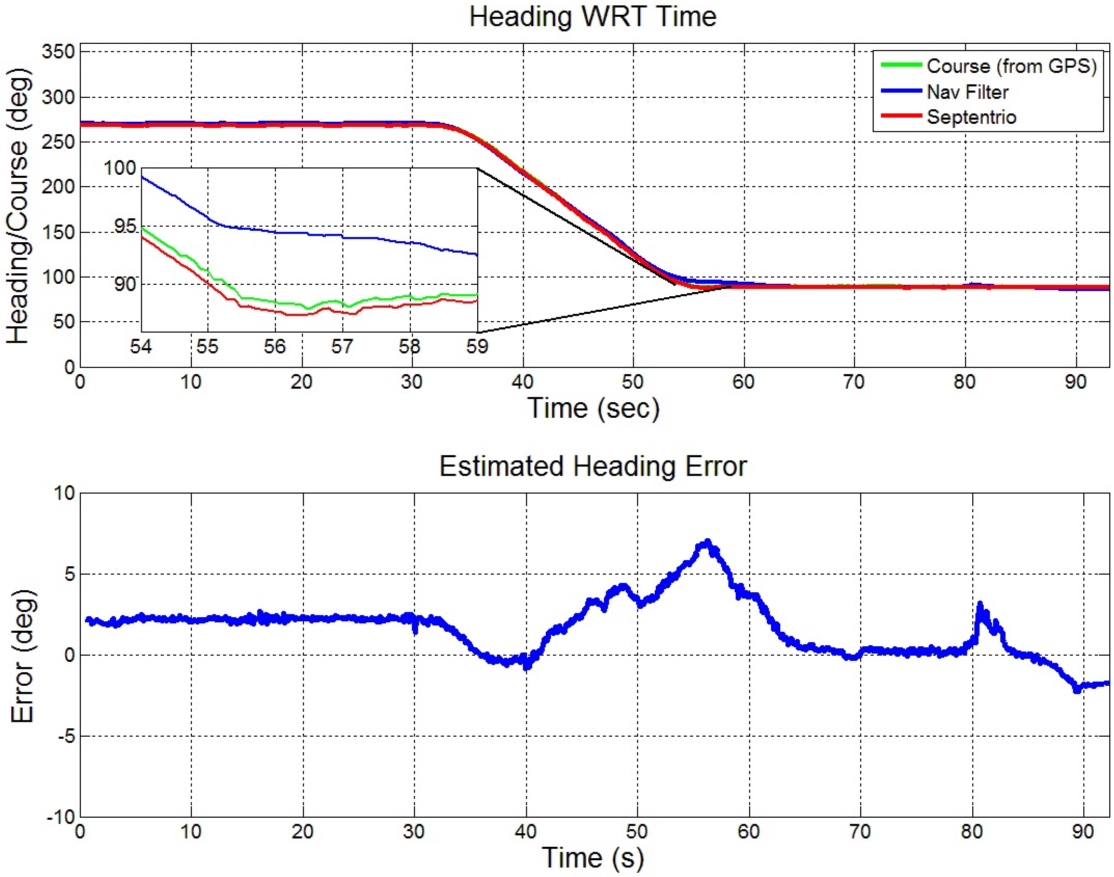

Figure 43: Estimated Vehicle Heading and Heading Error

1. Effects of Sensor Failures

The next set of figures show the effects of sensor failures on the system. At thirty seconds into the data run, different sensor outages are simulated to show what happens to the filter when certain sensors become unavailable. The simulated outage last for one minute, after which the measurements are reinstated.

The figure below shows the filter's estimated velocity and velocity error with various sensor failures. The GPS is the only sensor failure that causes a larger error in estimated velocity.

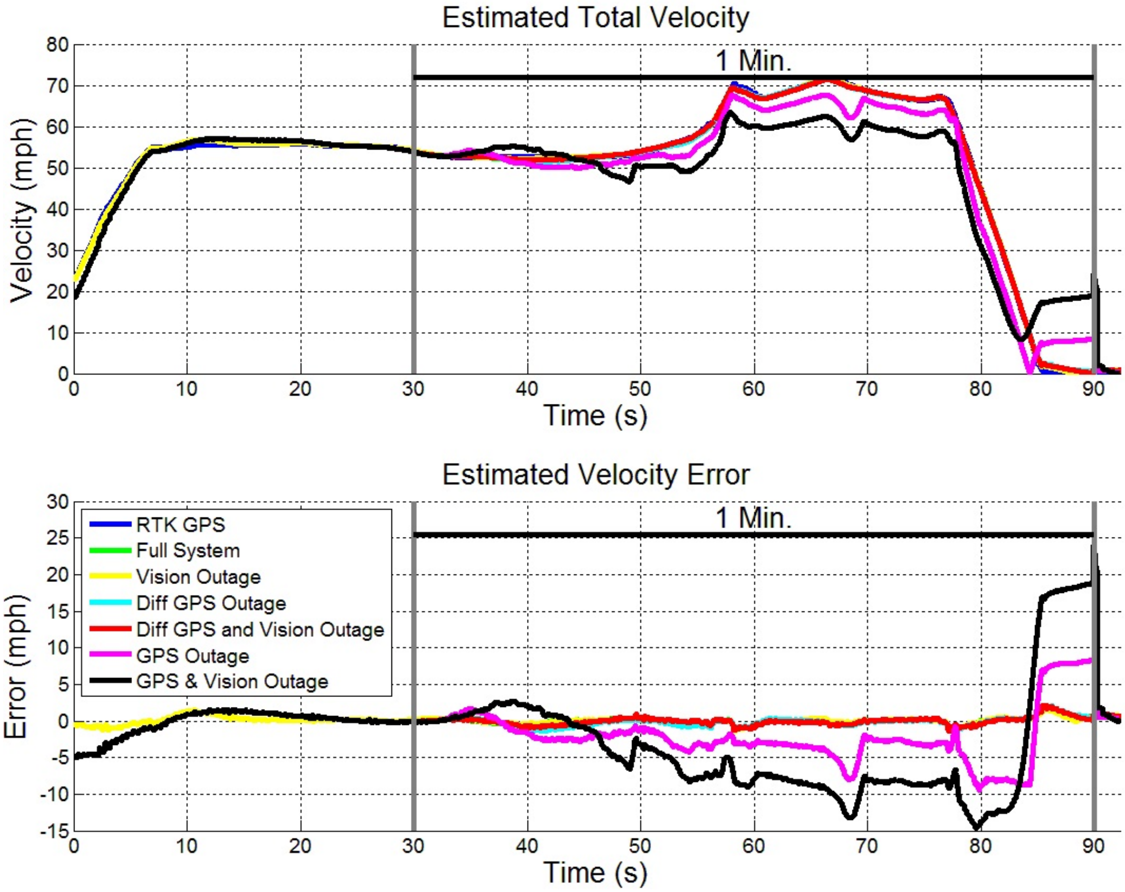

Figure 44: Estimated Velocity with Measurement Failures

Figure 44 is a plot of the estimated total velocity under several different failure conditions. The blue line represents the total velocity reported by the differential GPS alone. The green line represents the system working at full capacity. When vision measurements (yellow line), differential corrections (cyan line), or differential corrections and vision (red line) are removed, the estimated velocity closely resembles the RTK GPS estimated velocity. Only two failure situations result in a loss of velocity estimation. Both failures involve failure of all GPS measurements. Both failure of GPS (magenta line) and failure of GPS and vision (black line) result in loss of velocity estimation; however, there is a noticeable improvement when vision measurements remain operational.

The figure below shows the estimated lane position with respect to time. At 30 seconds into the run, different sets of measurements are cut off to show how the filter reacts without them. The filter still maintains an accurate lane position unless both the differential corrections and vision become unviable. This results in a bias lane position estimate. Lose of all measurements also results in a failure to estimate lane position. For the case when all measurements are lost, the error in lane position grows exponentially. After one minute of outage, the measurements are turned back on. For all cases, the filter start properly estimating lane position within a few seconds.

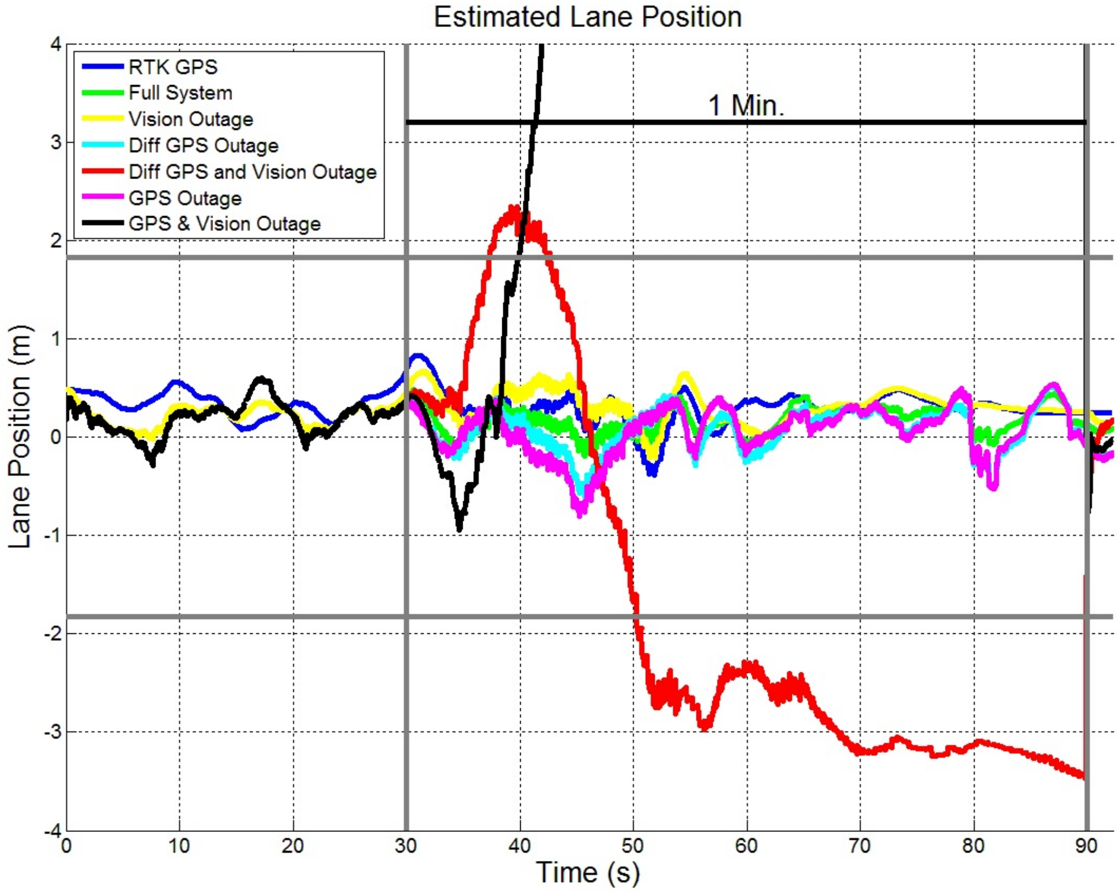

Figure 45: Estimated Lane Position with Measurement Failures

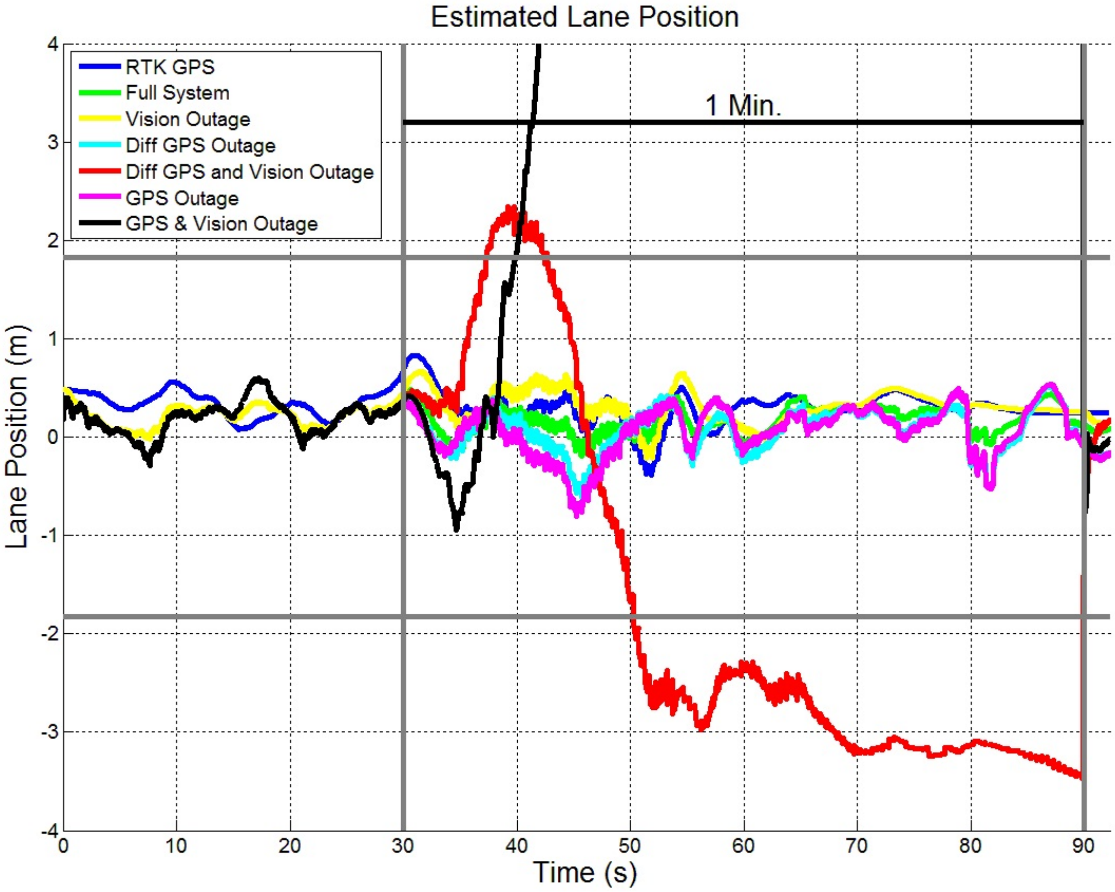

Figure 45 shows the performance of the estimated lane position under several different failure conditions. The blue line represents the lane position reported by the differential GPS alone, which was used to represent true position in the lane. The green line represents the system working at full capacity. When only the differential corrections are removed, the filter can still estimate lane position because vision measurements are still available. When GPS is completely turned off, lane position can be estimated correctly because vision measurements are still available. Only two failure situations result in a loss of accurate lane position estimation. The first case is when differential corrections and vision measurements are removed. For this situation, only standalone GPS is available which results in a biased lane position estimate. The second case is when GPS and vision are removed. This represents a loss of all measurement updates. The error in the lane position estimate will continually grow when all measurements are lost. Loss of lane position accuracy for both of this case occurs with five seconds.

The figure below shows the error with respect to time for the same data in the figure above. The case for loss of all measurements is not shown. The bias in lane position from when both differential corrections and vision is lost can clearly be seen.

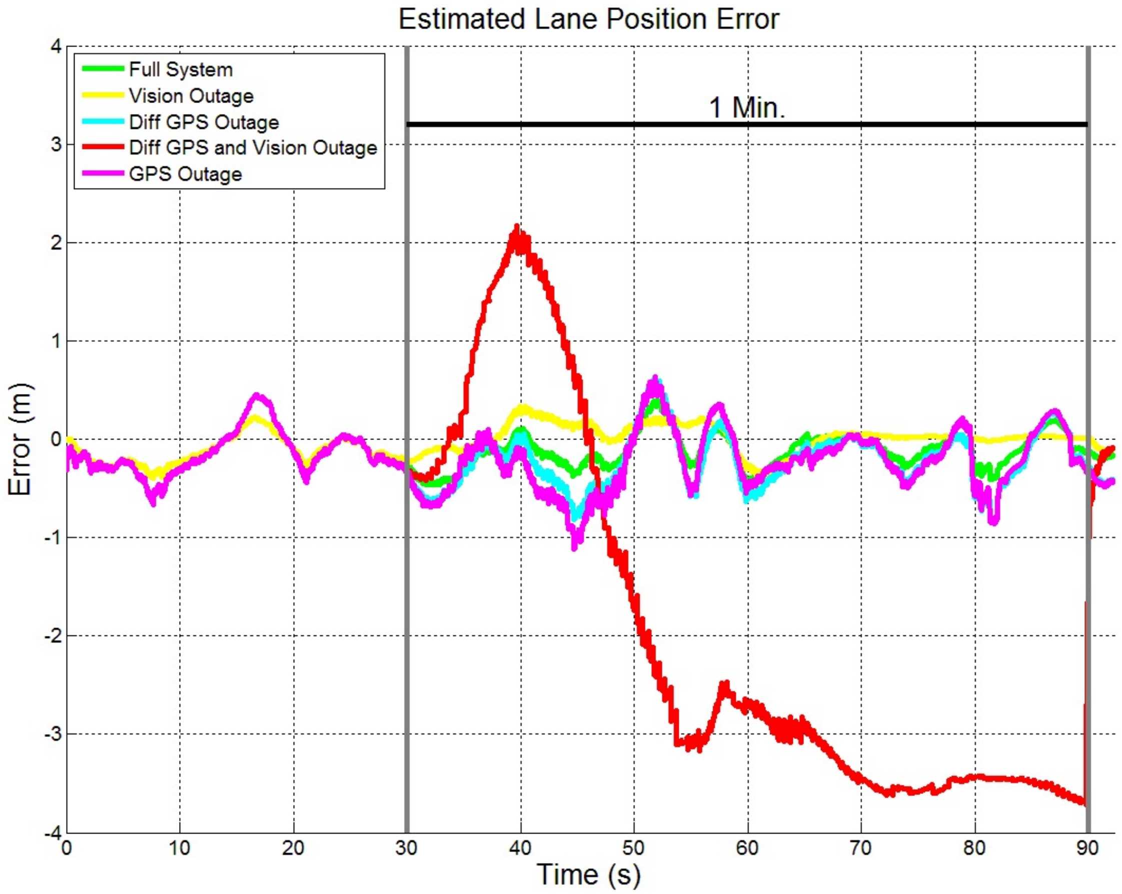

Figure 46: Estimated Lane Position Error with Measurement Failures

Figure 46 is a plot of the error of the estimated lane position. The advantage of using differential GPS can be seen when there are large errors in the vision measurements. Using differential GPS provides better filtering during periods of large vision measurement errors. For the case when only vision fails (yellow line), the filter solution converges to the GPS measurements, which in this case is differential GPS. This error plot is based off the differential GPS measurement; therefore, when differential GPS is the only measurement used (yellow line), the error is very close to zero. When differential corrections and vision are lost (red line), the filtered solution converges to the GPS measurement; however, since there are no corrections to the GPS, the measurement is biased. Ninety seconds into the data run when all measurements are reinstated, the filter corrects itself for all the failure conditions within a few seconds of reacquisition of measurements.

The figure below shows the estimated distance along the map. This represents distance traveled parallel with the direction of the road. Two of the loss of measurement situations from above causes the estimate to drift. The first is when GPS is cut off and only vision measurements are used. The second is when loss of all measurement occurs. The drift in longitudinal position is worse for the case of a loss in all measurements. Lake the data above, when the measurements are cut back on, the filter corrects itself within a few seconds.

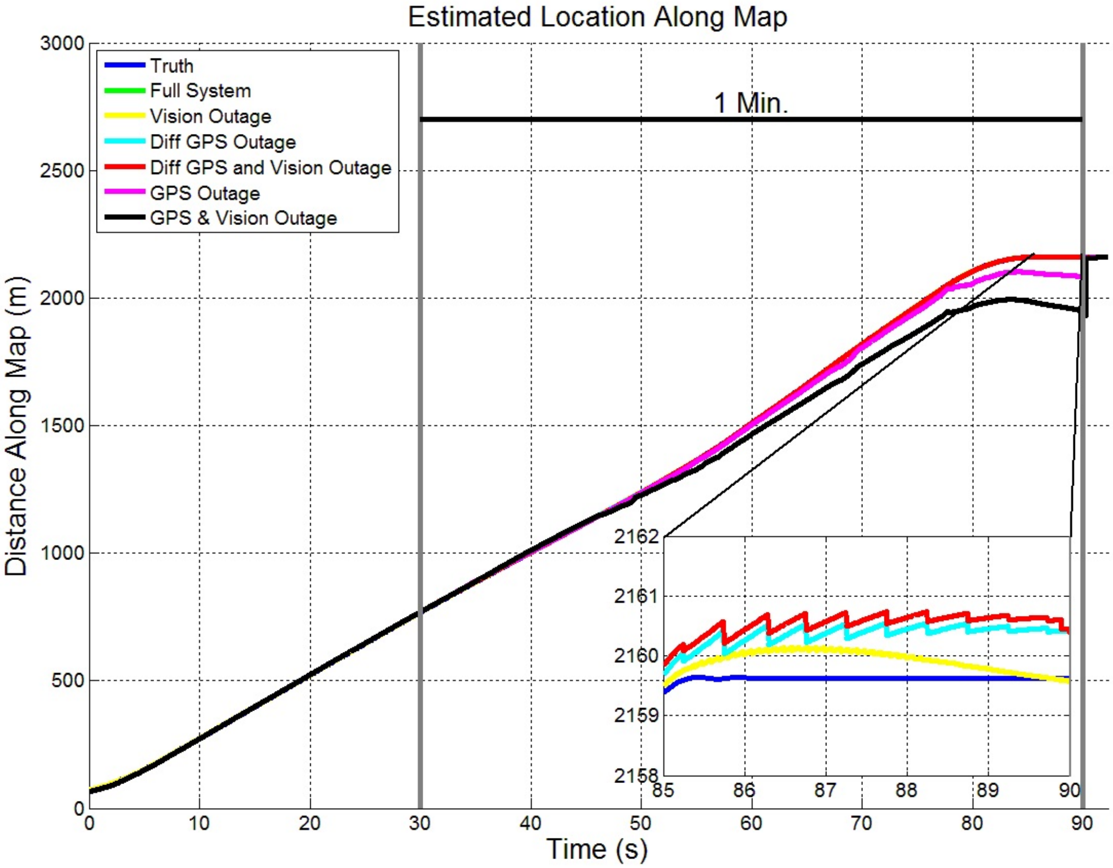

Figure 47: Estimated Longitudinal Position with Measurement Failures

Figure 47 shows the estimated longitudinal position. This represents the distance from the start of the map along the path of the map. Two failure conditions result in a loss of accuracy in the estimated distance along the map. The first is the GPS outage. For this case only vision measurements are available. As can be seen in

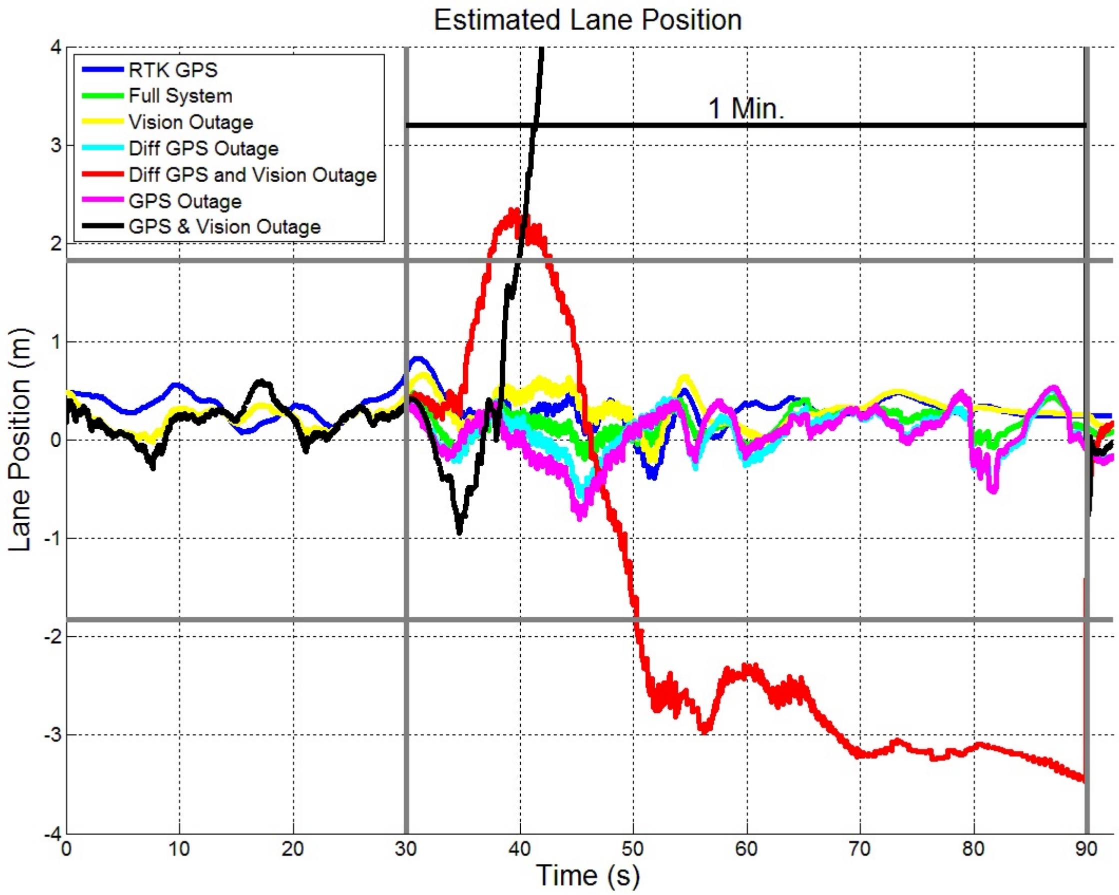

Figure 45, the lane position estimate still remains accurate for this case. However, since there is no GPS to provide position information in the axis parallel to the road, the estimated distance along the map is dead reckoned causing drift. When only vision measurements are available, the lane position estimate is still available, but the filter loses its location on the map. The second case where this estimate fails is when both GPS and vision are not available.

The figure below shows the error in the estimated longitudinal error. A drift in error can be seen for when GPS or vision and GPS are loss.

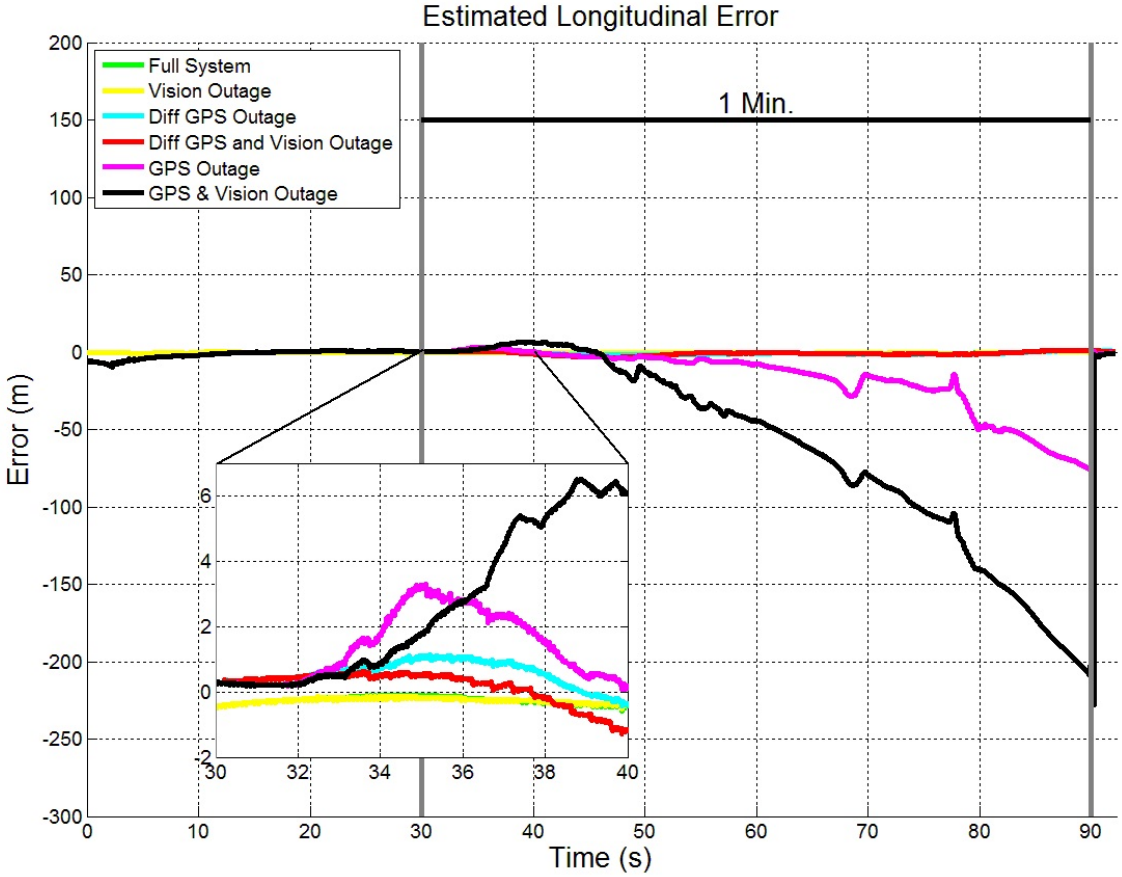

Figure 48: Estimated Longitudinal Position Error with Measurement Failures

Figure 48 is a plot of the estimated error in longitudinal position. The graph shows that the error in estimated position grows quicker for the case when no vision measurements are available. This shows that having vision does help in containing drift in longitudinal position, however, the error for both the loss of GPS (magenta line) and the loss of GPS and vision (black line) cause a continual error growth. Like the estimated lane position, the estimated longitudinal position corrects itself after measurements are reinstated.

The figure blow shows the estimated position with the various sensor failures using Google Earth. The figure also has a zoomed in portion to show the final estimated position. The red line represents loss of differential GPS and vision measurements. For this case lane position cannot be estimated due to the GPS biases and no vision measurements. The figure all shows that estimated longitudinal position cannot be properly estimated with GPS.

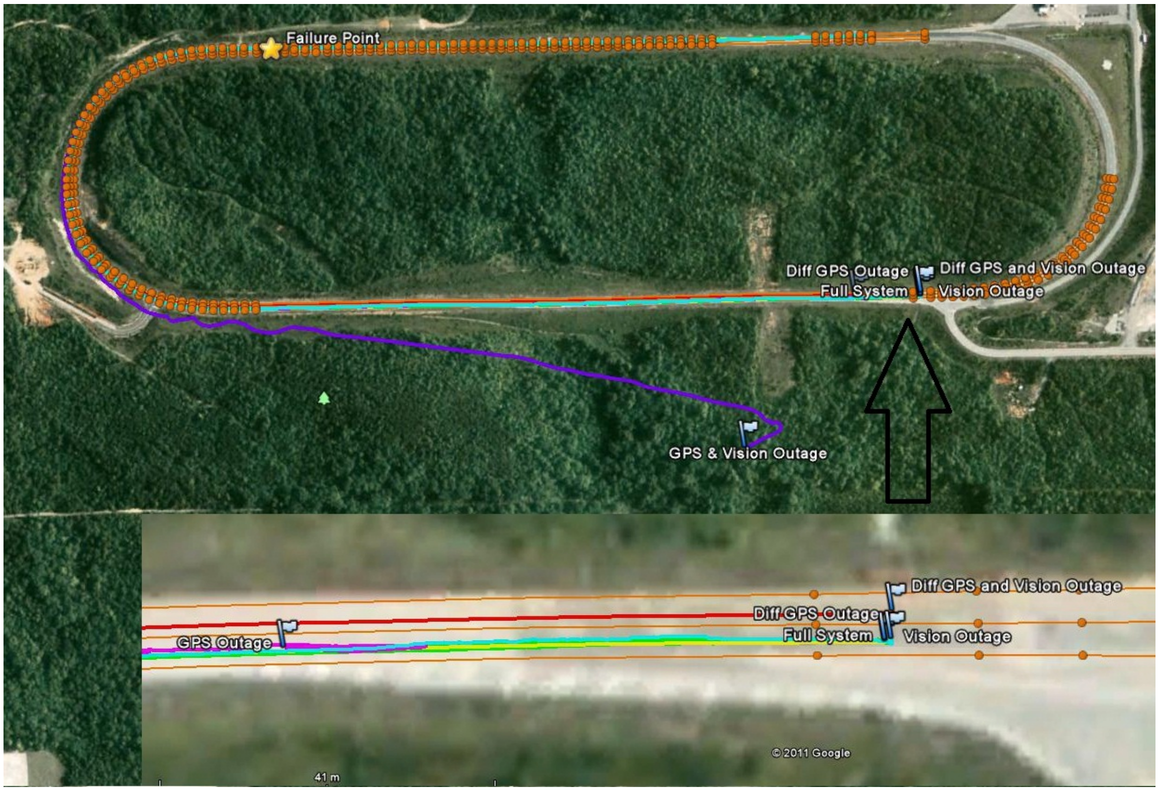

Figure 49: Estimated Position with Measurement Failures

Figure 49 is an above ground view of the estimated position during the various failures. The star represents the location of the vehicle when the measurements are removed.

The figure below shows the estimated position in each road coordinate frame axis for various lane map and vision measurement availability. The figure shows an improvement as measurements are added. The figure also shows an improvement in the longitudinal road axis position as the lane map and vision measurements become available. This is an unexpected event due to the fact the lane map and vision measurements provide no information in the longitudinal

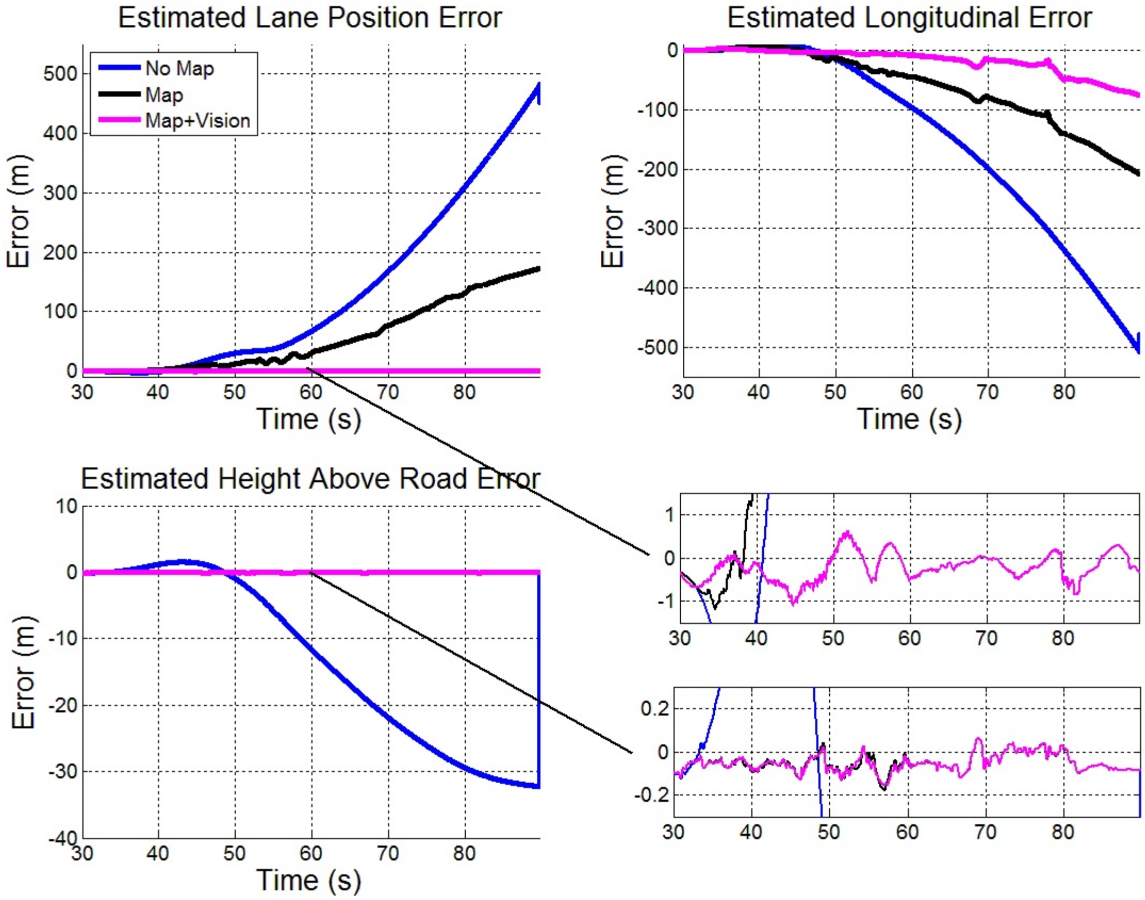

Figure 50: Estimated Error for Various Map and Vision Availability Using No GPS

Figure 50 provides the estimated error in each road axis when no GPS measurements are available. The blue line represents the error when no GPS, map, or vision measurements are available. For this case, there is no measurement updates. Therefore, there is a continual error growth in all axes due to integration of the IMU errors. The black line represents the error when no GPS or vision measurements are available. For this case, the map is available. This constrains the error in the vertical road axis. The magenta line shows the error when only GPS is not available. For this case, vision measurements are also available, constraining error in the lateral road axis. The error in the longitudinal axis is not constrained with GPS. Neither the vision measurements nor the height constraint provide full observability in the longitudinal road axis.

Table 6 shows the standard deviation of the error for each combination of available measurements. For the case when only differential GPS is used, the measurement of truth and the measurement used to update the filter are the same resulting in a low standard deviation of error.

Table 6: Error Statistics of Estimated Lane Position Given Different Measurement Availability

Failure Type	Average Absolute Error (m)	RMS Error(m)	Standard Deviation of Error (m)	Variance of Error (m^2)
None	0.1489	0.2006	0.1343	0.018
Vision	0.0267	0.1505	0.1481	0.0219
Diff GPS	0.1537	0.222	0.1601	0.0256
Diff GPS & Vision	-1.65	2.413	1.7591	3.0945
GPS	0.1502	0.2386	0.1854	0.0343
GPS & Vision	*	*	*	*
* No values computed due to a continual increase in error				

Table 7 gives the error statistics of the estimated longitudinal position for various available measurements. The error statistics of the longitudinal position matches the error statistics of the GPS measurement used. When using differential GPS (RTK GPS) the error is on the centimeter level. When standalone GPS is used the error is on the meter level. There is a continual growth of longitudinal position error if no GPS measurements are available. The growth of the error is slowed if vision measurements are available; however, the error will continue to grow until a GPS measurement becomes available.

Table 7: Error Statistics of Estimated Longitudinal Position Given Different Measurement Availability

Failure Type	Average Absolute Error (m)	RMS Error(m)	Standard Deviation of Error (m)	Variance of Error (m^2)
None	-0.2436	0.2961	0.1683	0.0283
Vision	-0.2328	0.2898	0.1726	0.02979
Diff GPS	-0.8571	1.2101	0.8541	0.7295
Diff GPS & Vision	-0.6193	1.1057	0.9159	0.8389
GPS	*	*	*	*
GPS & Vision	*	*	*	*
* No values computed due to a continual increase in error				

Table 8 gives the error statistics of the estimated lane position estimate for varying weather conditions. The error was estimated by differencing the RTK GPS reported position in the lane and the navigation filter's estimated position in the lane. The weather conditions tested were dusk, night, rain dusk, and rain night. These weather conditions are considered challenging for the vision measurements. The results for each weather condition are also split into vision measurement type used. This allows comparison of the filtered solution when using GPS and LiDAR, camera, or both LiDAR and camera measurements.

Table 8: Error Statistics of Estimated Lane Position Given Different Weather/Time Conditions

Weather/Time	Vision Type	Mean Error (m)	RMS Error(m)	Standard Deviation of Error (m)	Variance of Error (m^2)
Dusk	LiDAR	-0.3121	0.5618	0.4672	0.2183
	Camera	-0.5313	0.766	0.5517	0.3044
	Both	-0.6073	0.7811	0.4912	0.2413
Night	LiDAR	0.1284	0.9731	0.9647	0.9306
	Camera	0.3246	1.0194	0.9664	0.9339
	Both	0.1511	0.9605	0.9485	0.8997
Rain Dusk	LiDAR	0.0002	0.4354	0.4354	0.1896
	Camera	0.2484	0.8252	0.787	0.6194
	Both	0.218	1.1338	1.1127	1.238
Rain Night	LiDAR	-0.1351	0.375	0.3498	0.1224
	Camera	0.1135	0.4079	0.3918	0.1535
	Both	-0.0256	0.3944	0.3935	0.154

2. Limited Satellite Results

Urban and rural environments both pose different problems for each of the various algorithms. Navigation within a city has proved to be an interesting problem for GPS aided navigation systems due to the blockage of satellites by buildings. Similarly, trees on rural roads can block GPS satellites. However, the use of multiple sensors can provide a solution even with poor GPS satellite visibility. A typical navigation filter based off GPS measurement updates requires at least four GPS observations to remain operational. Each of the four observations is needed to solve an unknown. Three-dimensional position makes up three of the unknowns. The fourth unknown is the clock bias (or the difference in the GPS receiver's clock and the GPS system time).

The filter's vision measurements can measure position in one axis; the axis perpendicular to the road. Since the road map contains vertical information, position in the axis vertical to the road can be constrained. The height of the vehicle above the road is assumed to be constant. Therefore, the only non-constrained axis is the axis parallel to the road. Since position is only unconstrained in one axis, only using two GPS observations will result in a fully operational system. One GPS observation is needed for the unknown axis, and the other is needed for receiver clock errors.

Table 9: Satellite Azimuth and Elevation Angles

Satellite	Azimuth Angle	Elevation Angle	Carrier-to-Noise-Ratio

	(deg)	(deg)	(dB)
SV 5	63.15	19.49	48
SV 9	150.98	15.74	48
SV 15	100.48	79.8	54
SV 18	-79.82	37.84	53
SV 21	-38.55	51.92	52
SV 26	46.6	43.53	48
SV 27	128.5	24.82	42
SV 29	-148.28	48.83	51

The figure below shows a top down view of the track from Google earth. Red lines are drawn to show the azimuth angle to all the GPS visible satellites for this particular data run.

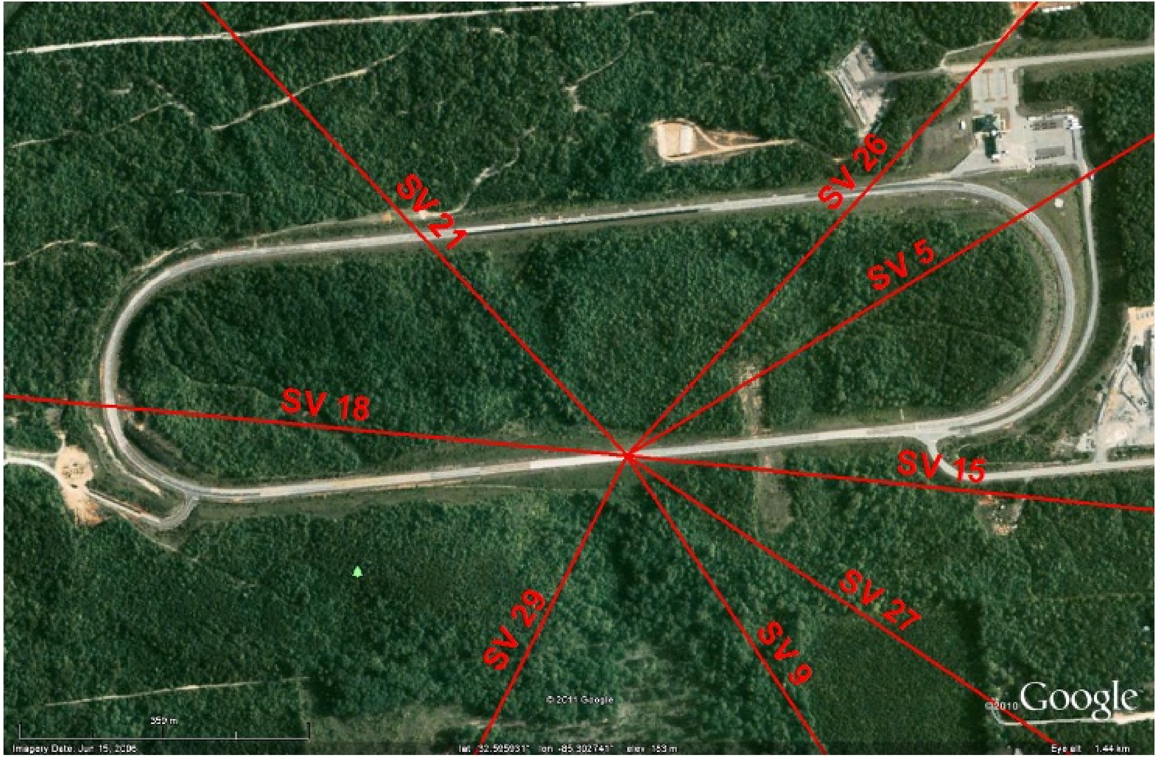

Figure 51: Satellite Line of Sight Vectors

Figure 51 shows the direction from which all the GPS observations are originating for the data run discussed in the previous section. This figure can be used as a reference for the plots below. The plots below show the data run from the last section; however, at thirty seconds, all but a few selected GPS observations are turned off. After one minute, all the GPS observations are turned back on. The simulated satellite outages will show how the filter reacts when less than four GPS observations are available and the effects of satellite geometry and measurement quality on the filter.

Like the measurement failure results, these results were created by looking at the effects of measurement failures on the same dataset. Thirty seconds into the run, all satellite observations except the satellite observations listed in the legend are turned off. Figure 52 shows the estimated velocity and velocity error for various limited satellite constellations.

The figure below shows the filter's estimated velocity and velocity error with various satellite constellations. The plot shows an increase in performance when the number of available GPS satellites is increased.

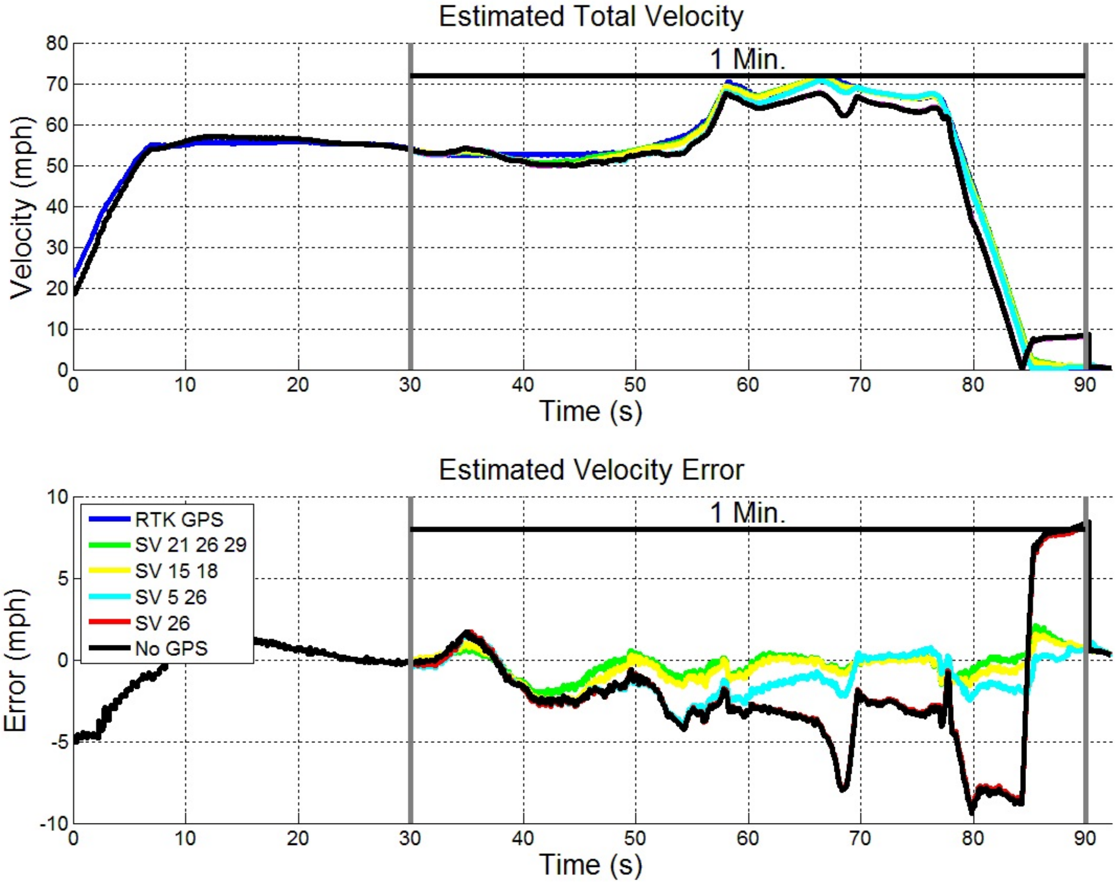

Figure 52: Estimated Velocity with Vision and Limited Satellite Visibility

Figure 52 shows that estimated velocity degrades as the number of GPS observations decreases. The three observation case (green line) has the least error, followed closely by the SV 15 and 18 combination (yellow line). The SV 5 and 26 combination (cyan) performs worse than the three and other two observations cases; however, this combination performs considerably better than the one and no observation case. There is no noticeable difference in estimated velocity when using one GPS observation (red line) and when using no GPS observations (black line).

The figure below shows the estimated lane position. 30 seconds into the run, all the GPS satellites except the ones listed in the legend are turned off to show how the filter reacts. For all cases, the lane position measurements remain accurate due to the vision measurements.

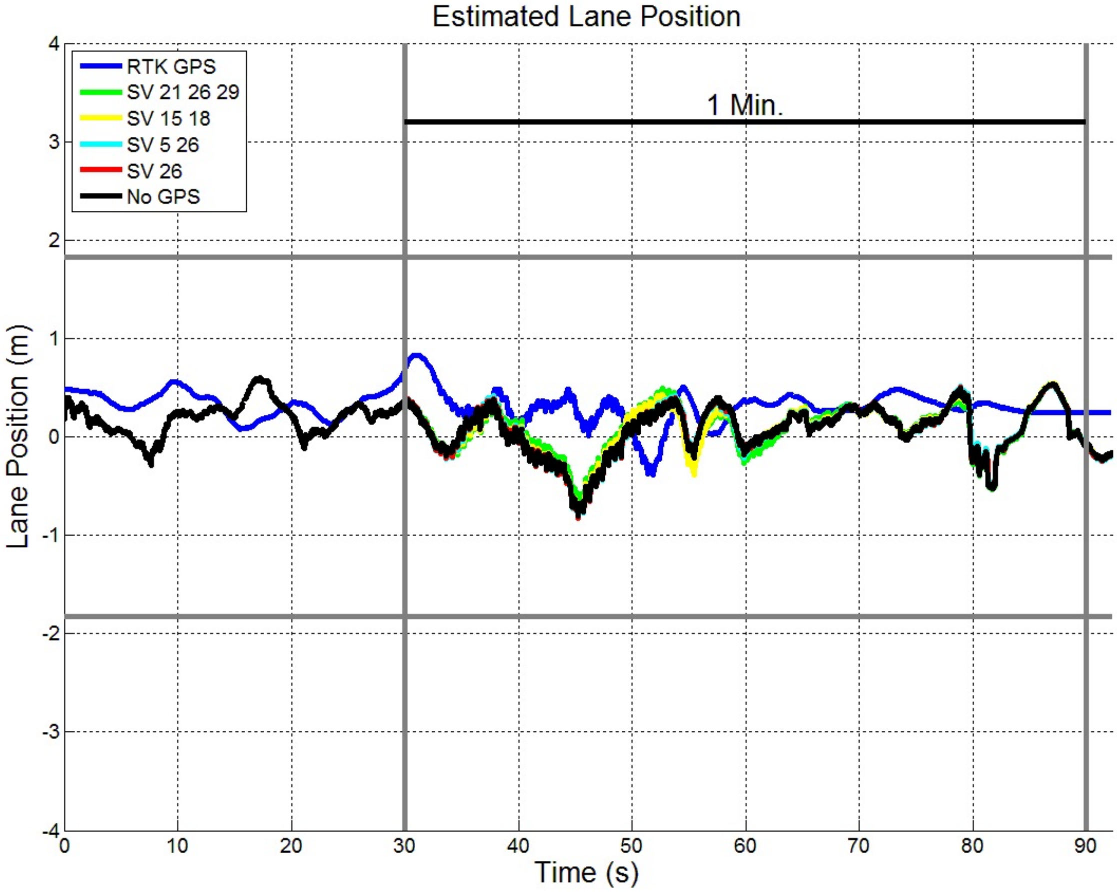

Figure 53: Estimated Lane Position with Vision and Limited Satellite Visibility

Figure 53 is a plot of the estimated lane position for the various satellite constellations. The blue line is the estimated truth using the RTK GPS measurements. The horizontal gray lines represent the lane lines assuming a twelve foot lane. When vision measurements are used, the lane position estimate remains accurate for all limited satellite constellations. Even when no GPS observations are available, the vision measurements ensure that lateral lane position accuracy is maintained. The lane position estimate reflects a filtered solution . The component of any GPS observations that lie in the lateral road axis will be ignored because the visions measurements have a much smaller associated variance. If vision measurements are not available, the navigation filter will not have the observability necessary to estimate lane position.

The figure below shows the estimated distance along the map. This represents distance traveled parallel with the direction of the road. As long as two GPS observations are available, the estimated lane position does not drift. The situation for when only one GPS observation is available results in a drifting longitudinal estimate.

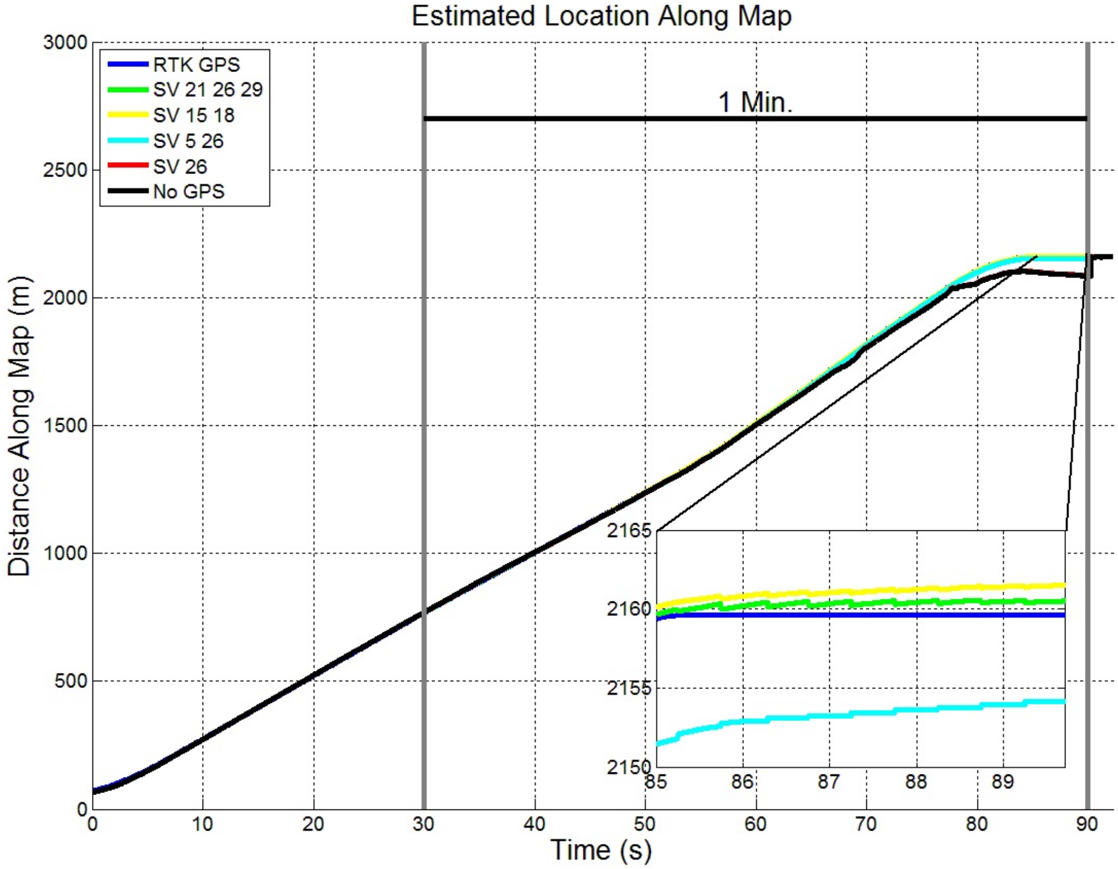

Figure 54: Estimated Longitudinal Position with Vision and Limited Satellite Visibility

Figure 54 shows the estimated longitudinal position with vision measurements. This represents the distance from the start of the map along the path of the map. This is where the use of some GPS observations shows the most benefit. The order of performance is the same as Figure 52. When there are two or more GPS observations are available, the filter is able to estimate longitudinal position; however, this value may be biased. The zoomed in section of the plot shows the biases at the end of the satellite outage. If less than two GPS observations are available, the filter no longer has the observability necessary to estimate longitudinal lane position. Also, there is no noticeable difference in drift when using one GPS observation (red line) and using no GPS observations (black line).

The figure below shows the estimated longitudinal position error. For the case when only SV 5 and 26 are available, a bias in position can be seen.

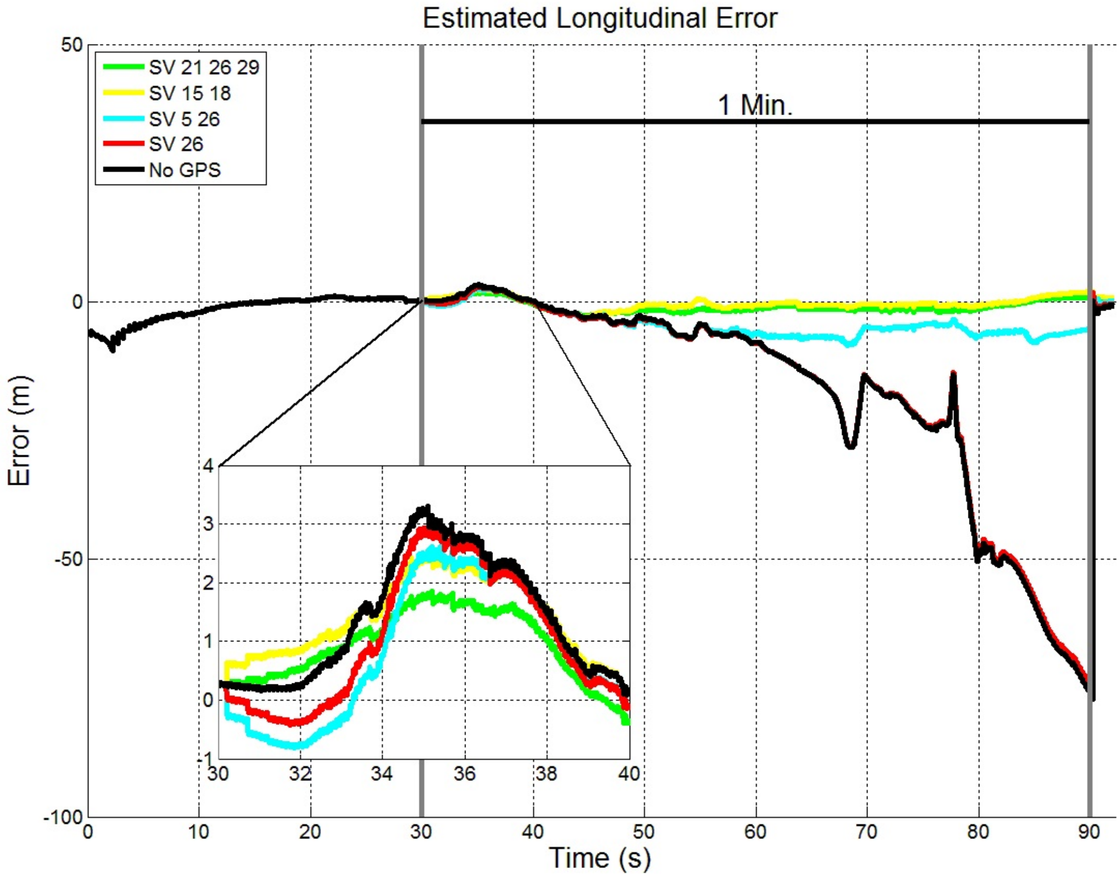

Figure 55: Estimated Longitudinal Position Error with Vision and Limited Satellite Visibility

Figure 55 is a plot of the estimated error in longitudinal position. The plot does show the estimated longitudinal position contains a large bias for the case when only SV 5 and 26 are used. This is most likely due to the lack in observability due to the fact that the azimuth angles to both of these satellites are close together. For the case when only SV 15 and 18 are used, the bias present is along the level seen when using standalone GPS with more than four observations. This is most likely due to the better observability due to the fact that the azimuth angles to the satellites are far apart. If only two observations are available, then the filter still remains observable; however, there can be bias in estimated longitudinal position. This bias is a function on the quality of the observations available and the constellation. The best results appear to come when the two observations are 180 degrees apart in azimuth angle and both of the azimuth angles line in the axis parallel with the direction of the road (one satellite in front of the vehicle and the other behind). Typically, in an urban environment, observable satellites will reside in the axis parallel with the direction of the road. The estimated longitudinal position becomes unobservable if one or no GPS observations are available.

The figure below shows the estimated position for each satellite geometry using Google Earth. For this case, both the vision measurements and lane map are available. The map is zoomed in to show the end of the data run. As long as two GPS satellites are visible, then the filter can estimated the longitudinal position in the lane to within a few meters. There is no improvement between the one satellite scenario and the no satellite scenario.

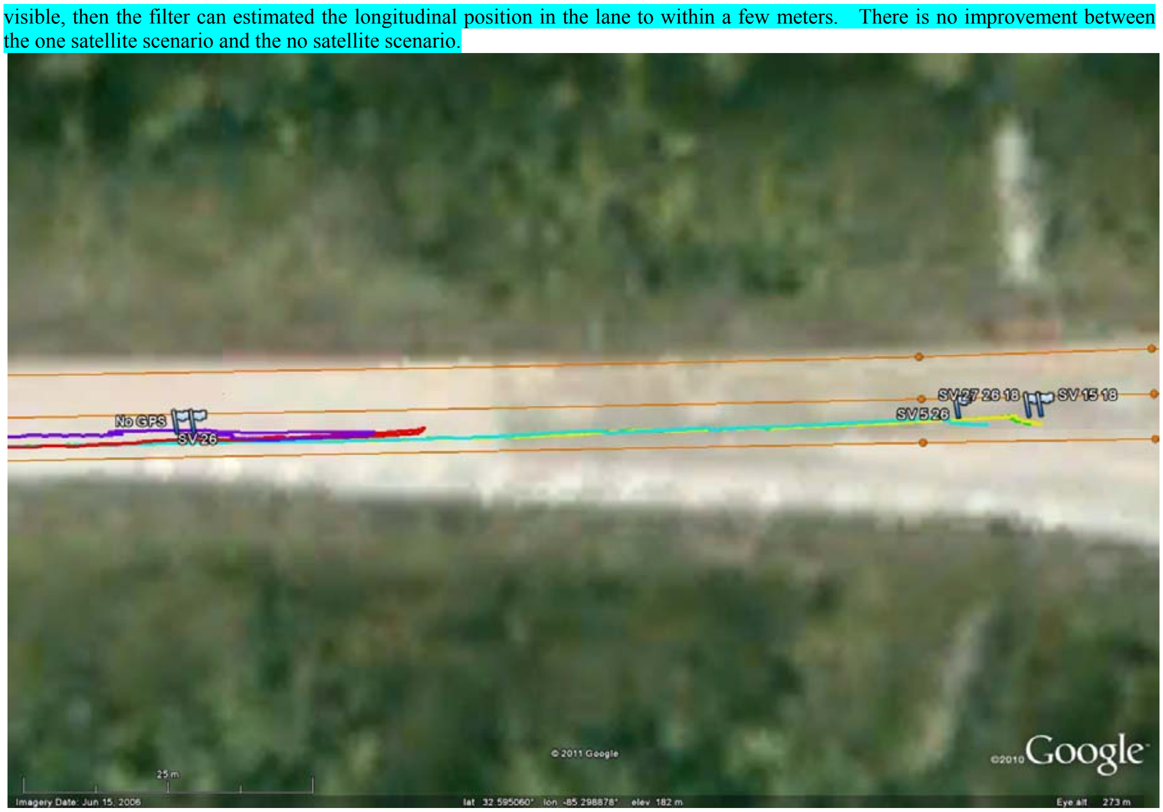

Figure 56: Estimated North-East Position with Vision Measurements, Map, and Limited Satellite Visibility

Figure 56 provides a top down view at the end of the satellite outage. At the end of the one minute outage, the car has come to a complete stop from 70 mph. For this case, both the map and vision measurements were used. The vision measurements insures that for every possible GPS satellite outage, lateral lane position is always to be maintained. The error in the longitudinal road axis is dependent on GPS observations available. As long as two observations are available, the navigation filter will remain fully observable.

The figure below shows the estimated position for each satellite geometry using Google Earth. For this figure, only GPS and the lane map where used. Without vision measurements and with the limited satellite constellation, the filter fails to provide an accurate lane position; however, there is an improvement when using the lane map over the case with no lane map or vision measurements.

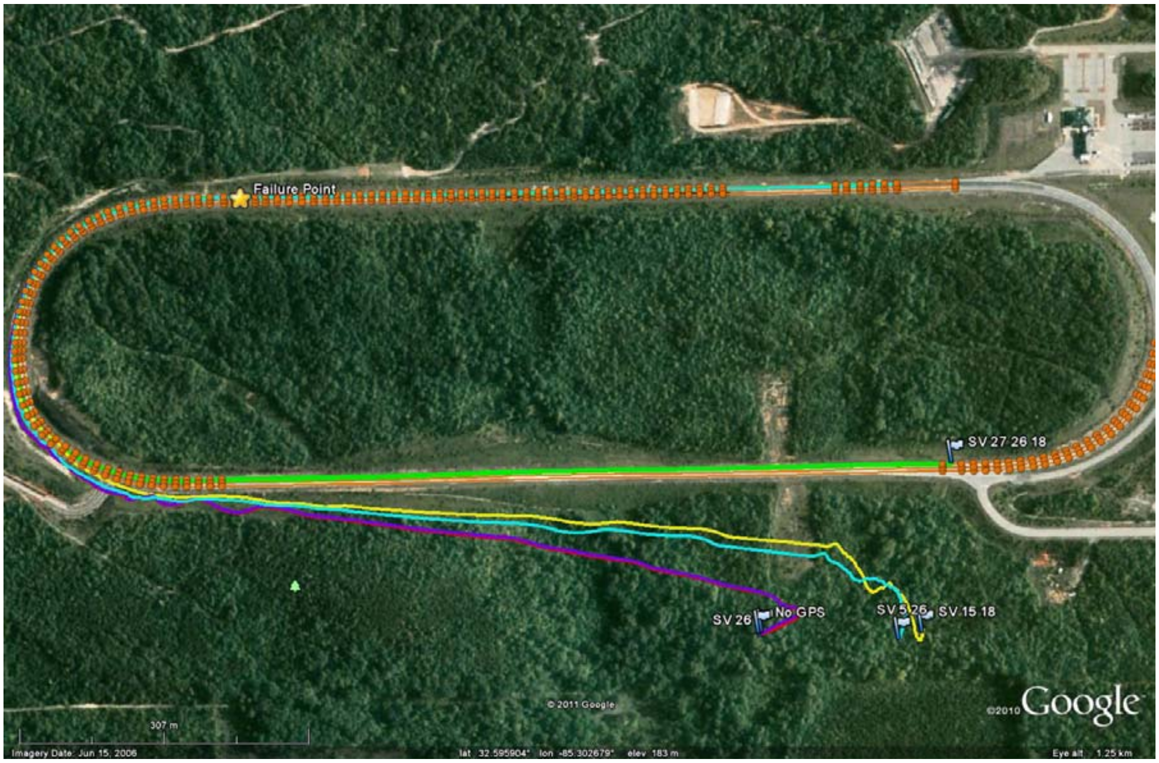

Figure 57: Estimated North-East Position with Map and Limited Satellite Visibility

Figure 57 shows the estimated position when using only the map with limited satellite visibility. When the map is available, the filter can still use height measurements to constrain the vertical position. With a constrained height, the three GPS observation case is fully observable; however, without the vision measurements, the lane position estimate will be biased. Comparing Figure 57 to Figure 58 demonstrates a clear improvement in drift when using the height measurement.

The figure below shows the estimated position for each satellite geometry using Google Earth. Without vision measurements and the lane map, and with the limited satellite constellation, the filter fails to provide an accurate lane position.

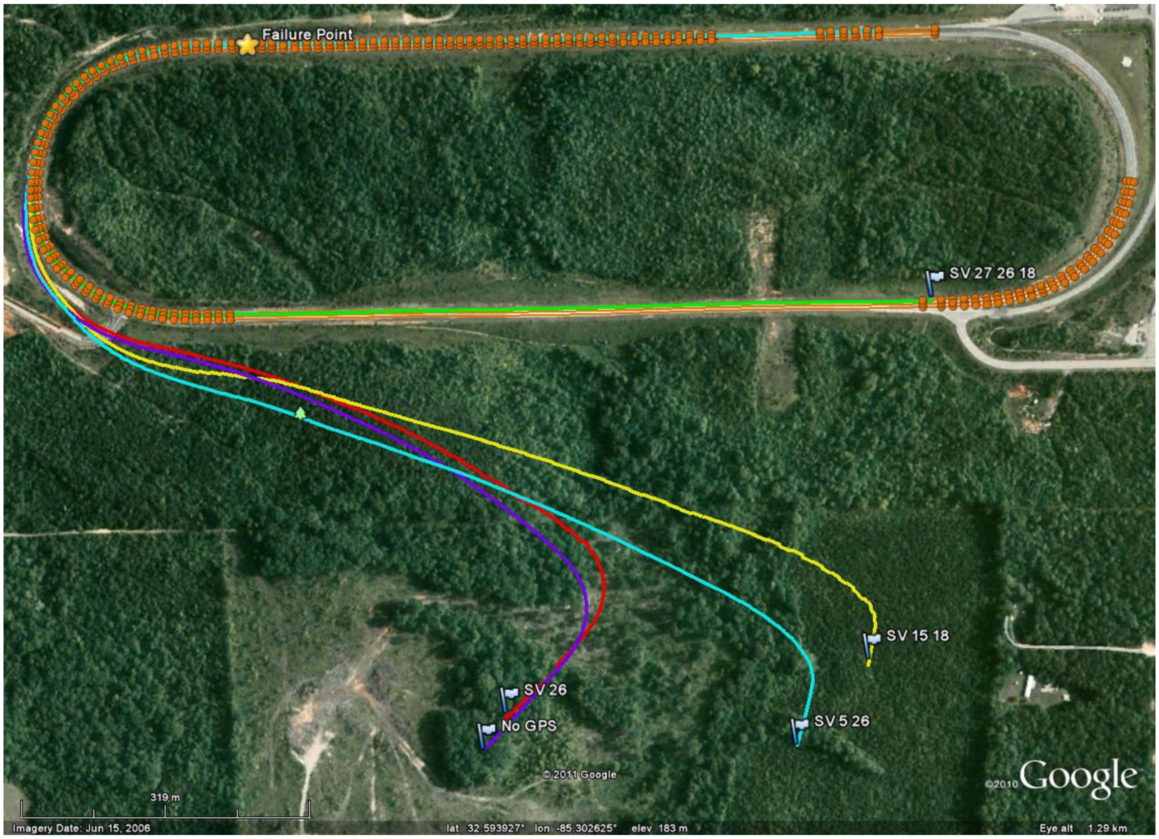

Figure 58: Estimated North-East Position with Limited Satellite Visibility

Figure 58 shows the estimated position when using no map or vision measurements. Like all the other limited satellite results, performance increases with the number of GPS observations available. The three observation case (green line) performs much better than the other cases. The system is not observable with only three GPS observations; therefore, the position will continue to degrade. With three GPS observations, the clock bias and drift is the only unobservable mode; therefore, the drift when using only three GPS observations is dependent on the stability of the clock used by the receiver. There is an improvement in drift rate when using one or two GPS observations; however, the drift for any case where less than three GPS observations are available is very large.

The figure below shows the estimated longitudinal position for each possible combination of two GPS satellite observations.

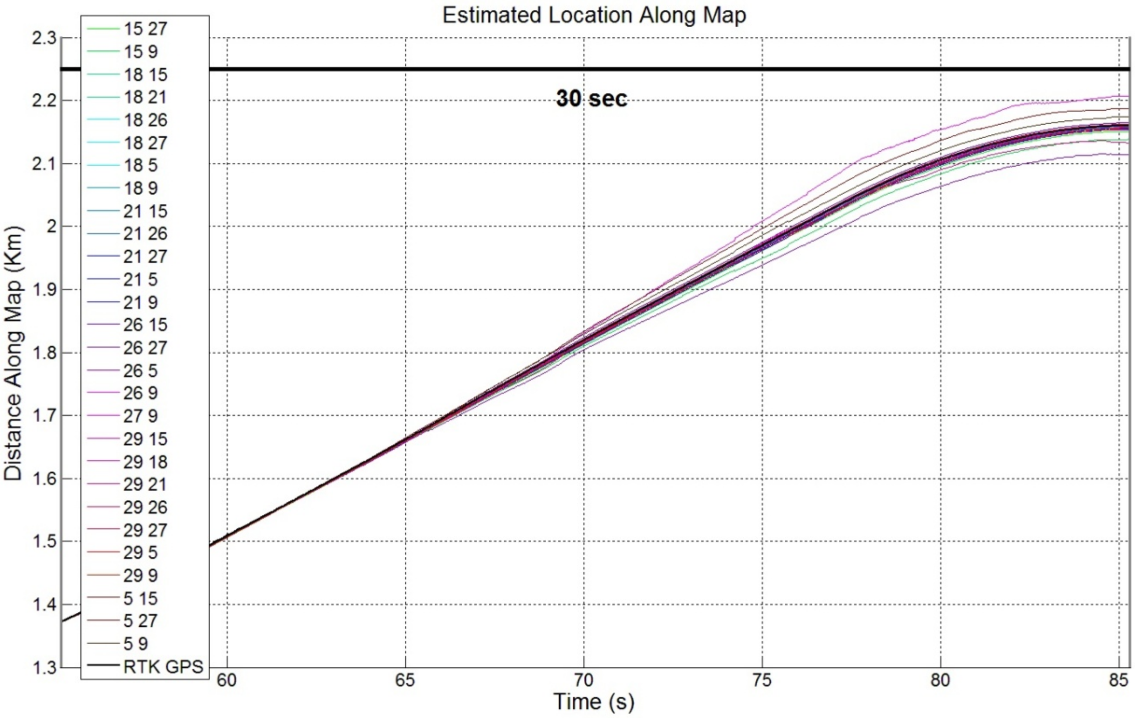

Figure 59: Estimated Longitudinal Position for All Possible SV Pairs

In order to better understand the effect of satellite constellation on the case where two GPS observations and vision measurements are used, Figure 59 shows estimated longitudinal position for every possible pair of GPS observations. For all cases, the longitudinal position is biased but observable. There are two combinations that result in what appears to be a continual error growth; however, it is not known if the estimated longitudinal position would converge to some steady state bias given enough time.

The figure below shows the estimated longitudinal position error for each possible combination of two GPS satellite observations.

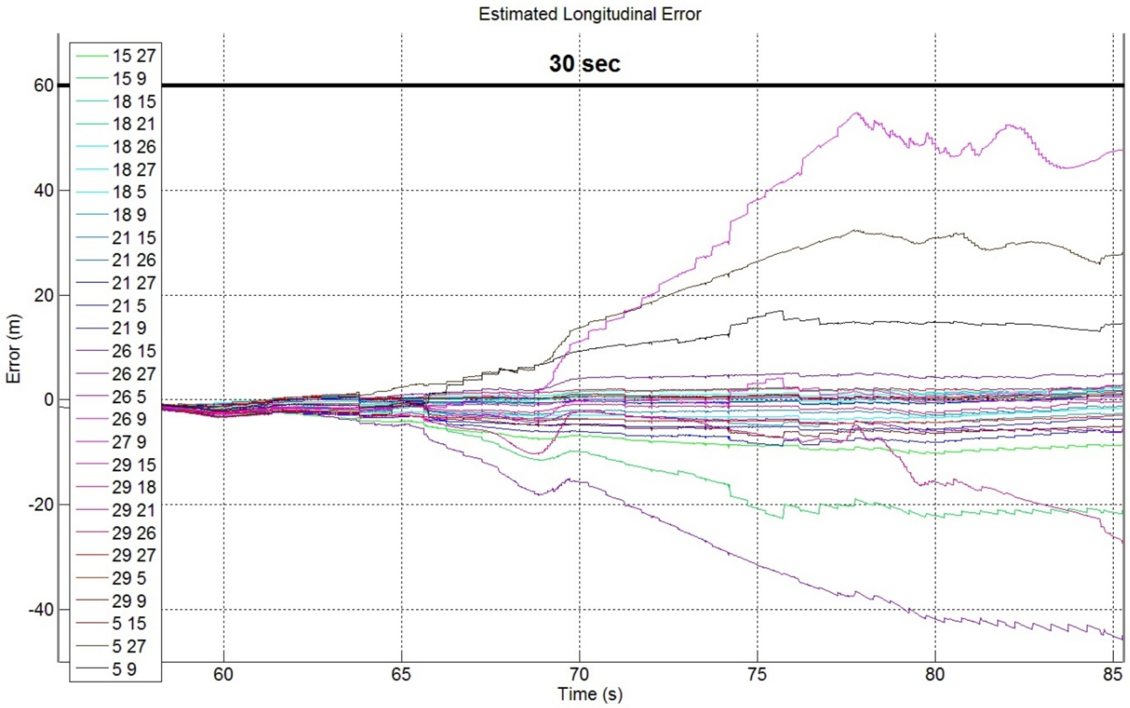

Figure 60: Estimated Longitudinal Error for All Possible SV Pairs

Table 10 shows the longitudinal position bias after one minute of navigating with only two GPS satellites for every possible pair of satellites. The satellite pairs are listed in the table from highest bias (worst performance) to lowest bias (best performance). Four of the satellite combinations resulted in less than a meter of bias. Nine of the satellite combinations resulted in less than a two meter bias. And eighteen of the twenty-eight pairs resulted in less than a 5 meter bias.

Table 10 also provides the average carrier-to-noise ratio of the two observations, and the lowest carrier-to-noise ratio of the pair. The carrier-to-noise ratio is a good indication of the noise present in an observation. A larger carrier-to-noise ratio corresponds to low measurement noise on that observation. The carrier-to-noise ratio is measured in decibels (a logarithmic scale). The data shows a correlation between carrier-to-noise ratio and longitudinal position bias. The bias tends to increase as average carrier-to-noise ratio falls. There are some outliers (SV 15 9, SV 21 9, SV 26 15); however, these outliers consists of a large carrier to noise ratio paired with a low carrier-to-noise ratio. This suggests that both observations should have good carrier-to-noise ratio to ensure a small longitudinal position bias. Ideally both observations will have a carrier-to-noise ratio larger than 50 dB. The effects of using GPS observations with less than a 40 dB carrier-to-noise ratio have not been studied.

This data does show that some satellite pairs can result in close to a fifty meter bias. The results shows that with good satellite geometry and good satellite measurements, the navigation filter will remain operational. However, if satellite geometry is bad (line of sight vectors to the satellites are close to perpendicular to the road) and measurement quality is poor (low carrier-to-noise ratio), the filter can experience large errors in longitudinal position.

Table 10: Longitudinal Position Error after 30 Seconds for Every SV Pair

SV Pair	Longitudinal Position Error (m)	Average C2N (dB)	Lowest C2N (dB)
SV 26 9	47.64	48	48
SV 26 27	-45.51	45	42
SV 5 27	27.77	45	42
SV 29 21	-26.72	5.15	51
SV 15 9	21.72	51	48
SV 5 9	14.52	48	48
SV 15 27	-8.666	48	42
SV 21 9	-6.125	50	48
SV 26 5	-6.025	48	48
SV 26 15	-5.236	51	48
SV 29 9	4.947	49.5	48
SV 21 27	-3.619	47	42
SV 29 26	2.981	49.5	48
SV 29 27	-2.846	46.5	42
SV 27 9	2.67	45	42
SV 18 9	-2.567	50.5	48
SV 18 26	2.445	50.5	48
SV 18 21	2.283	52.5	52
SV 21 26	1.953	50	48
SV 29 5	1.742	49.5	48
SV 18 5	1.682	50.5	48
SV 18 27	-1.546	47.5	42
SV 21 15	-1.19	53	52
SV 5 15	1.123	51	48
SV 21 5	0.9541	50	48
SV 18 15	0.9308	53.5	53
SV 29 15	0.5867	52.5	51
SV 29 18	-0.3918	52	51

5 Conclusions

The measurements from a camera and LiDAR were shown to produce an accurate lateral distance within the lane in various weather and lighting conditions. A waypoint map is needed in order relate these vision measurements (given in the road coordinate frame) to the global (ECEF coordinate frame). A map also provides an additional height measurement because the height of the vehicle above the road is known and constant. Provided the map is globally accurate, the vision and height measurements provide a much more unbiased measurement of global position than standalone GPS. If the map is not globally accurate, the global position estimates will be incorrect and biased. The bias will be a direct representation of the global error in the map. For example, if the map is two meters off in the east direction, the global solution when using the map and vision measurements will be biased by two meters in the east direction. The global accuracy of the map will determine the global accuracy of the estimated global position; however, the global accuracy of the map has no effect on the estimated lane position. If RTK differential GPS or vision measurements are available, the filter will always be able to estimate an unbiased lane position.

It has also been shown that vision and height measurements, along with a map, will greatly reduce the drift experienced when GPS is lost. The vision and height measurements constrain drift in two-axis; thus, the solution only drifts in the road's longitudinal axis. One GPS range could be used to observer position in the road's longitudinal axis. However, using GPS observations ads another unobservable mode (receiver clock bias and drift); therefore, two GPS observations are needed (along with vision measurements and a map) in order to maintain full observability. At least one of the GPS observations must have a comment in the road's longitudinal axis; therefore, as long as both GPS observations do not lie in the axis perpendicular to the road, full observability is maintained. It is unlikely two GPS observations will have the same azimuth angle or an exactly 180 degrees azimuth angle difference. While the navigation filter remains fully observable with two GPS observations, map, and vision measurements, the performance of positioning in the road's longitudinal axis is dependent on the constellation of the two GPS observations and the error in the observations. Satellite constellation and ranging errors can cause a large bias in the longitudinal position axis.

This report has presented a method of using lane position measurements from sensors already found on vehicles, specifically cameras, LiDARs, IMU’s, and GPS, to update a global navigation filter for pose estimates. This work has specifically focused on fusing these sensors to obtain a pose estimate even in situations where one or more of these sensors can fail. The results from this project are expected to contribute to a host of vehicle systems including navigation and especially vehicle safety.

A. Future Work

There is a large potential for future work in developing lane maps and map keeping algorithms. The information in lane maps has not been standardized. There may be extra information that should be contained in the lane map that was not covered in this work. Examples of information that can be included with each segment are the attitude of the road, distance to left and right lane marking (or lane width), and information on what map (if any) is to the left and right of the current section. The attitude of the road is needed to use the vision measurements of lane position to update the filter. The distance to the left and right lane marking can be used by the vision systems to determine position in the lane without making a constant lane width assumption. Also, setting the distance to the lane lines to zero could signify no lane marking in the current section. This would allow the filter to ignore vision measurements in sections that do not have painted lane lines (like an intersection). Adding information on what map lies to the left and the right of the current section would allow the filter to know what map and index to use when changing lanes.

The use of wheel odometry could be very effective for time when GPS is not available or when the number of the number of GPS observations is limited. Wheel odometry would provide an estimated position in the longitudinal axis to the road; however, wheel odometry consists of integrating the wheel speed sensor. This will cause a continual error growth in the odometry estimate. The drift associated with wheel odometry is small compared to the drift cause by IMU integration; therefore, there would be an improvement in the degradation of the longitudinal position, but it would not be a fully observable system with GPS. The system would however be able to maintain an accurate position for much longer when the filter is unobservable (GPS is not available).

Auburn University is currently working on a project with the FHWA that incorporates the work done for this project plus the addition of new types of measurements. The measurements include visual odometry (provided by SRI), road signature matching (provided by Penn State), and DSRC raining (provided by Kapsch). The goal of the new project (similar to this project) is to study the effects on filter robustness and performance when one navigation filter and several different types of measurements are used to determine global position and lane position for use in vehicle safety systems.

References

[1] J. Kibbel. W. Justus, and K. Furstenberg. Lane estimation and departure warning using multilayer laserscanner. In *Proc. IEEE Intelligent Transportation Systems.* Pages 607-611. September 13-15 2005.

[2] K. Furstenberg, J. Kibbel, W. Jusuts, R. Schulz, K. Dietmayer, N. Kampchen. *Advanced Microsystems for Automotive Applications 2005.* VDI-Buch. Springer Berlin Heildelberg. 2005.

[3] Alexander von Reyher et al, *A LiDAR-based approach for near range lane detection.* IEEE, June 2005.

Appendix A: The Gantt Chart

Intelligent Multi-Sensor Measurement Schedule: YEAR ONE												
	FYI 2008											
Task	Jan - 08	Feb - 08	Mar - 08	Apr - 08	May - 08	June - 08	July - 08	Aug - 08	Sept - 08	Oct - 08	Nov - 08	Dec - 08
1.0 Project Management	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
2.0 Instrument Test Vehicle	🟩	🟩	🟩									
2.1 Interface GPS/INS	🟩	🟩	🟩									
2.2 Interface LDW Camera	🟩	🟩	🟩									
2.3 Interface IBEO Lidar		🟩	🟩									
Milestone One: Test Vehicle Equipped												
3.0 Collect Sensor Data				🟩	🟩	🟩						
3.1 Define Experiments				🟩								
3.2 Data Collection					🟩	🟩						
4.0 Sensor Modeling						🟩	🟩	🟩	🟩			
4.1 GPS/ INS Model						🟩						
4.2 LDW Model							🟩					
4.3 IBEO Lidar Model								🟩				
4.4 Model Validation									🟩			
Milestone Two: Models Complete and Validated												
5.0 Sensor Processing and Data Analysis								🟩	🟩	🟩	🟩	
5.1 Develop Sensor Fusion Algorithms								🟩	🟩			
5.2 Define modified test and collect additional data									🟩	🟩		
5.3 Develop / evaluate preliminary MOEs										🟩	🟩	
6.0 Demonstration and Final Report												🟩
6.1 Demonstrate Vehicle Capability to FHWA												🟩

Intelligent Multi-Sensor Measurement Schedule: YEAR TWO												
	FYI 2009											
Task	Jan-09	Feb-09	Mar-09	Apr-09	May-09	Jun-09	Jul-09	Aug-09	Sep-09	Oct-09	Nov-09	Dec-09
1.0 Project Management	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
2.0 Re-evaluation of MOEs	🟩	🟩										
3.0 Algorithm Refinement			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Milestone One: 2nd Generation Algorithms Complete												
4.0 Quantification of Performance						🟩	🟩	🟩	🟩	🟩	🟩	🟩
Milestone Two: Performance Evaluation Experiments Complete												
5.0 Interim Report								🟩	🟩			

Intelligent Multi-Sensor Measurement Schedule: YEAR THREE							
	FYI 2010						
Task	Jan-10	Feb-10	Mar-10	Apr-10	Jun-10	Jul-10	Aug-10
1.0 Project Management	🟩	🟩	🟩	🟩	🟩	🟩	🟩
2.0 Algorithm Refinement	🟩	🟩	🟩	🟩	🟩		
Milestone One: 2nd Generation Algorithms Complete							
3.0 Quantification of Performance	🟩	🟩	🟩	🟩	🟩	🟩	
Milestone Two: Performance Evaluation Experiments Complete							
4.0 Final Report						🟩	🟩

Appendix B: Truth Data

Lane markings, when present on the track, were surveyed using RTK GPS on the outer lane marking and middle lane marking. These lane markings were nonexistent in some areas and faint in others due to the degradation of the lane markings from the trucks. The lane markings also diverged from their normal path when the road branched to other parts of the NCAT track; a situation similar to highway exits. Also, the asphalt differed at various intervals on the track due to other work being done at the NCAT track.

Due to the lack of sufficient survey points around the track, the true distance data for each run shows a noticeable jaggedness at each curve of the track. Since the lane marking between each survey point is assumed to be straight, and the lane marking between survey points is actually curved at the curves of the track, error is introduced into the truth data. As the vehicle passes one survey point and approaches another, the error increases as the actual curve in the road reaches its maximum distance from the assumed straight lane marking line. To counteract this error, a cubic spline was conducted on the survey points on the curve, which essentially provides more survey points around the curve of the track. Additional error with this strategy is introduced due to any failure to interpolate the correct curve of the lane markings. However, this error is significantly less than the error due to fewer survey points. Figure 61 shows the survey points and the interpolated points obtained through the cubic spline.

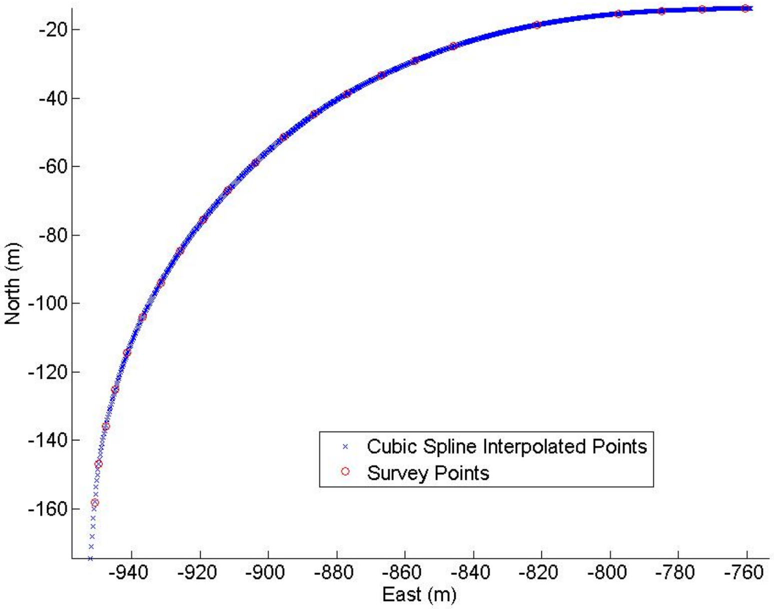

Figure 61: Cubic Spline on Curve of Track for Truth

True Lateral Distance

The lateral distance used to quantify the system performance is only as accurate as the survey points and differential GPS data taken during the test. The survey recorded the outside lane markings, the middle of the lane, and the middle lane markings (usually dashed). Position estimates from GPS during the run were timestamped and recorded in the Earth-Centered Earth-Fixed coordinate frame and were rotated into the North-East-Down (NED) coordinate frame. Each of these position estimates has standard deviations of about two centimeters.

Each GPS point from the test was compared with every outside lane survey point to determine the two survey points that are closest to the test point using the simple distance formula with the North and East coordinates. With knowledge of the nearest two survey points (labeled in Figure 62), (N_{s1},E_{s1}) and (N_{s2},E_{s2}), and the GPS test point, (N_t,E_t), the lateral distance is as determined as follows:

$$p_{truth} = \sqrt{(E_t - E_{norm})^2 + (N_t - N_{norm})^2}$$

(N_{norm},E_{norm}) is the point (blue dot in Figure 62) on the line between the survey points (N_{s1},E_{s1}) and (N_{s2},E_{s2}) at the shortest distance from the GPS point (N_t,E_t) and the line (shown in Figure 62 as the green line) between the survey points and is calculated as

$$N_{norm} = N_{s1} + u(N_{s2} - N_{s1})$$
$$E_{norm} = E_{s1} + u(E_{s2} - E_{s1)}$$

where

$$u = \frac{((E_t - E_{s1})(E_{s2} - E_{s1}) + (N_t - N_{s1})(N_{s2} - N_{s1}))}{\sqrt{(E_{s1} - E_{s2})^2 + (N_{s1} - N_{s2})^2}}$$

True Heading

True heading was calculated from the GPS survey data and the GPS point taken from the vehicle during the test run. Each GPS point from the test was compared with every outside lane survey point to determine the two survey points that are closest to the test point using the simple distance formula using the North and East coordinates. With knowledge of the nearest two survey points (shown in Figure 62 as the yellow-green line), (N_{s1},E_{s1}) and (N_{s2},E_{s2}), and the GPS test point and the last GPS test point (shown in Figure 62 by an orange line), (N_{t1},E_{t1}) and (N_{t2},E_{t2}), the lateral heading is as determined as follows:

$$\psi_{truth} = \tan^{-1}\left(\frac{m_2 - m_1}{1 + m_1 m_2}\right)$$

where

$$m_1 = \frac{N_{s2} - N_{s1}}{E_{s2} - E_{s1}}$$

$$m_2 = \frac{N_{t2} - N_{t1}}{E_{t2} - E_{t1}}$$

As a vehicle changes lane, the heading of the vehicle and the direction of motion of the vehicle may not be equal. This phenomenon is called sideslip. The presence of sideslip can cause errors in the truth measurement due to the assumption in the truth calculation for heading that the motion of the vehicle is the same as its heading. If the vehicle is sliding across the road, the GPS test data will show that the heading of the vehicle is in the direction of motion rather than the direction in which the vehicle was facing. A quick simulation in CarSim has shown that the heading error in a typical lane change due to sideslip is no more than one degree.

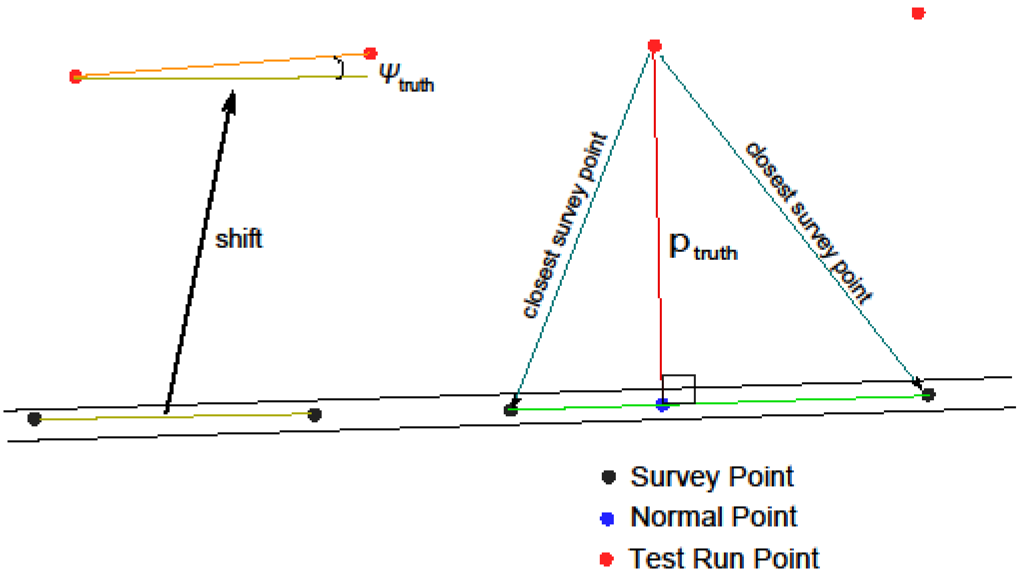

Figure 62: Determining Truth

Appendix C: Coordinate Frame Rotation and Translation

Lane position measurements are given in the road based coordinate frame. The road based coordinate frame can be approximated with a waypoint based map. A rotation matrix is need to rotate coordinates in the navigation's coordinate frame to the road coordinate frame because the navigation filter's coordinate frame is not oriented the same way as the road coordinate frame. Since the navigation coordinate frame and the road coordinate frame do not have the same origin, coordinate frame translation must also be covered.

A rotation matrix is a matrix that if multiplied by a vector of values expressed in an initial coordinate frame will result in a vector of values expressed in a new coordinate frame. The new coordinate frame has some different attitude from the initial coordinate frame. The difference in attitude will govern the values in the rotation matrix. A rotation matrix can be constructed using Euler angles. The rotational direction of Euler angles are based off the right handed coordinate system. In short, rotation about an axis that points towards the observer results in a counter-clockwise positive rotation. Rotation about an axis that points away from the observer results in a clockwise positive rotation. Rotation matrices are orthogonal; therefore, the inverse of a rotation matrix is equal to its transpose. The equation below describes this principle.

$$C_\alpha^{\beta^{-1}} = C_\alpha^{\beta^T} = C_\beta^\alpha$$

Rotation matrices are denoted by C_α^β. This rotation matrix maps coordinates in the α coordinate frame to the β coordinate frame. e denotes the ECEF coordinate frame. n denotes the North East coordinate frame. r denotes the road coordinate frame.

Two-Dimensional Rotations

The position, speed, and heading navigation filter employs a two-dimensional navigation coordinate frame. Two-Dimensional coordinate frame rotation matrices are 2x2 matrices based off one attitude angle. Figure 63 shows the principle of a two-dimensional rotation.

Figure 63 shows rotation about a single axis. A rotation matrix can be constructed to map vectors in the initial coordinate frame (i,j) to a rotated coordinate frame (i',j'). Theta is the angle of rotation. This angle is measured from the i axis to the i' axis. The right hand coordinate frame rule governs which direction of rotation is positive. The angle theta in this picture represents a positive rotation.

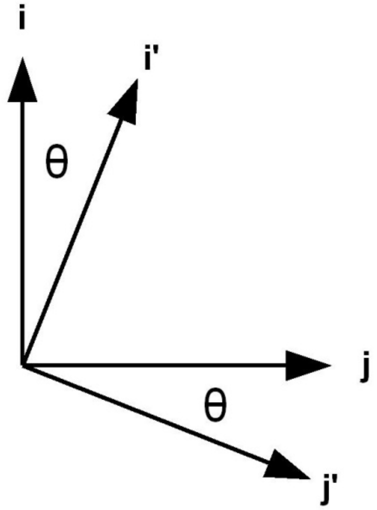

Figure 63: Two-Dimensional Rotation

The (i,j) coordinate frame is the starting coordinate frame, and the (i’,j’) coordinate frame is the final coordinate frame. For the coordinate frame in figure 2.5, the right handed coordinate frame rule defines the axis of rotation as pointing down (or away from the observer); therefore, clockwise rotation results in a positive Euler angle. The equation below shows the rotation matrix that will map coordinates in the (i,j) coordinate frame to the (i’,j’) coordinate frame. The rotation matrix used to aid the position, speed, and heading navigation filter will use a rotation matrix of the same format.

$$C_{(i,j)}^{(i',j')} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

Three-Dimensional Rotations

Three-Dimensional coordinate frame rotation matrices are 3x3 matrices based off three Euler angles. A three-dimensional coordinate frame rotation can be thought of as a series of three two-dimensional rotations. Figure 64 shows the rotation series used for this project.

Figure 64 shows a three-dimensional rotation sequence. The first rotation is about the k axis. This will result in a new coordinate frame (i’,j’,k’). The second rotation is about the j’ axis. This will result in a new coordinate frame (i’’,j’’,k’’). The third rotation is about the i’’ axis. This will result in the final rotated coordinate frame.

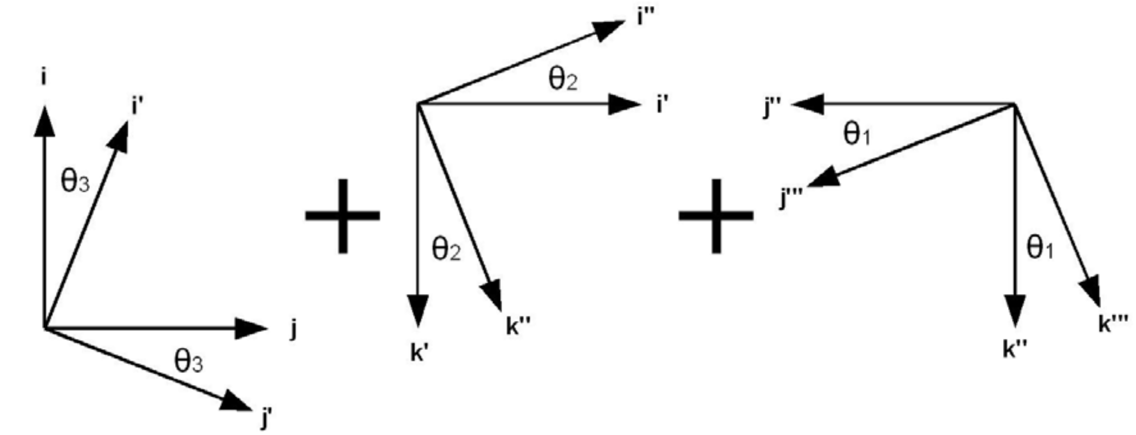

Figure 64: Three-Dimensional Rotation

The first rotation is about the z (k) axis. The second rotation is about the new y (j') axis. The third rotation is about the new x (i'') axis. The equation below shows how the three-dimensional rotation matrix is constructed. s_1 is the sine of θ_1 and c_1 is the cosine of θ_1. s_2 is the sine of θ_2 and c_2 is the cosine of θ_2. s_3 is the sine of θ_3 and c_3 is the cosine of θ_3.

$$C_{(i,j)}^{(i''',j''')} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_1 & s_1 \\ 0 & -s_1 & c_1 \end{bmatrix} \begin{bmatrix} c_2 & 0 & -s_2 \\ 0 & 1 & 0 \\ s_2 & 0 & c_2 \end{bmatrix} \begin{bmatrix} c_3 & s_3 & 0 \\ -s_3 & c_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_2c_3 & c_2s_3 & -s_2 \\ s_1s_2c_3 - c_1s_3 & s_1s_2s_3 + c_1c_3 & s_1c_2 \\ c_1s_2c_3 + s_1s_3 & c_1s_2s_3 - s_1c_3 & c_1c_2 \end{bmatrix}$$

Coordinate Frame Translation

The origin of the road based coordinate frame is not located at the same point in space as the origin of the navigation coordinate frame origin. When mapping coordinates in the navigation coordinate frame to the road coordinate frame, the coordinate frame must be moved and rotated. Figure 65 shows an example of this.

Figure 65 shows an example of a two-dimensional coordinate frame translation and rotation. In order to map a position vector in the alpha coordinate frame to the beta coordinate frame, two things must be known. The first is the position vector from the origin of the alpha coordinate frame to the origin of the beta coordinate frame expressed in the alpha coordinate frame. The second is the change in attitude from the alpha coordinate frame to the beta coordinate frame.

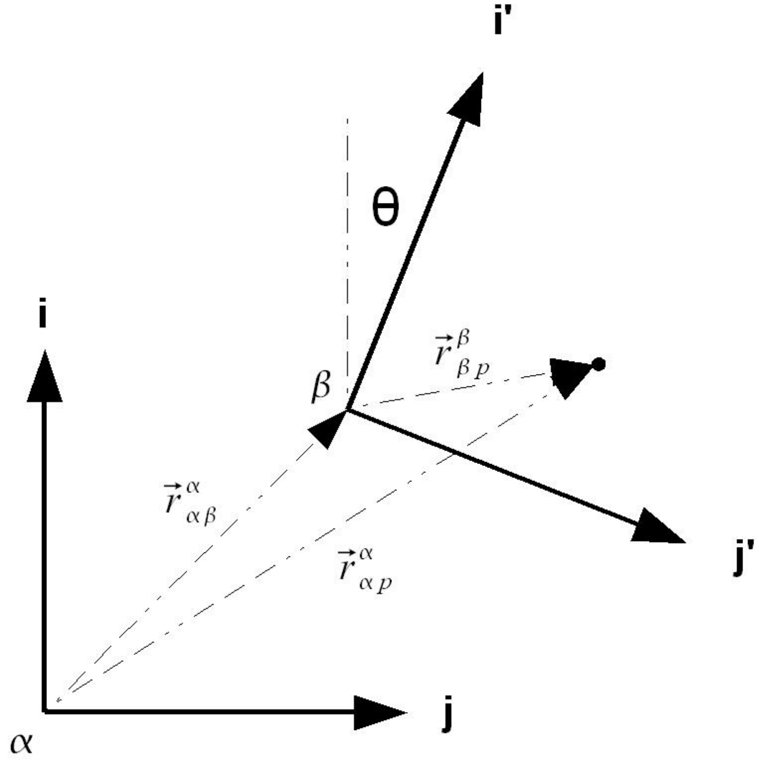
i'
θ
i
$\vec{r}^{\,\beta}_{\beta p}$
β
$\vec{r}^{\,\alpha}_{\alpha\beta}$
$\vec{r}^{\,\alpha}_{\alpha p}$
j'
α
j

Figure 65: Coordinate Frame Translation

The equation below shows how to map coordinates in coordinate frame that has been moved and rotated. $\vec{r}^{\alpha}_{\alpha p}$ is the position vector expressed in the initial coordinate frame (α), and $\vec{r}^{\beta}_{\beta p}$ is the position vector expressed in the rotated and translated coordinate frame (β). $\vec{r}^{a}_{ab}$ is a vector expressed in the initial coordinate that points from the initial coordinate frame to the new coordinate frame. C^{β}_{α} is the rotation matrix from coordinate frame alpha to coordinate frame beta.

$$\vec{r}^{\beta}_{\beta p} = C^{\beta}_{\alpha}\left[\vec{r}^{\alpha}_{\alpha p} - \vec{r}^{\alpha}_{\alpha\beta}\right]$$

Coordinates in the final coordinate frame can be mapped back to the initial coordinate frame using the equation below.

$$\vec{r}^{\alpha}_{\alpha p} = C^{\alpha}_{\beta}\vec{r}^{\beta}_{\beta p} + \vec{r}^{\alpha}_{\alpha\beta}$$

Global Coordinate Frame Rotations

Sometimes it may be necessary to map coordinates in the Earth Centered Earth Fixed (ECEF) coordinate frame to the North East Down (NED) coordinate frame. The North East plane of the NED coordinate frame is tangential to a reference ellipsoid. The reference ellipsoid is an ellipsoid that mimics the surface of the earth. Using the latitude

(ϕ) and longitude (λ) of the origin of the NED coordinate frame, a rotation matrix that maps ECEF coordinates to NED coordinates can be constructed. The equation below shows the rotation matrix that maps coordinates in the ECEF coordinate frame to the NED coordinate frame. The transpose of this matrix will map coordinates from the NED coordinate frame to the ECEF coordinate frame.

$$C_e^n = \begin{bmatrix} -\sin(\phi)\cos(\lambda) & -\sin(\phi)\sin(\lambda) & \cos(\phi) \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\phi)\cos(\lambda) & -\cos(\phi)\sin(\lambda) & -\sin(\phi) \end{bmatrix}$$

The equation below coverts ECEF position coordinates ($\vec{r}_{ep}^{e}$) to NED position coordinates ($\vec{r}_{np}^{n}$). Notice that, along with the latitude and longitude of the origin of the NED coordinate frame, the position of the origin expressed in the ECEF coordinate frame ($\vec{r}_{en}^{e}$) must be known.

$$\vec{r}_{np}^{n} = C_e^n\left[\vec{r}_{ep}^{e} - \vec{r}_{en}^{e}\right]$$

The equation below coverts NED position coordinates to ECEF position coordinates.

$$\vec{r}_{ep}^{e} = {C_e^n}^T \vec{r}_{np}^{n} + \vec{r}_{en}^{e}$$

The equation below coverts ECEF velocity coordinates to NED velocity coordinates. Notice that velocity mapping is independent of the position of the NED coordinate frame. The latitude and longitude of the NED coordinate frame is still need to construct the rotation matrix.

$$\vec{v}_n = C_e^n \vec{v}_e$$

The equation below coverts NED velocity coordinates to ECEF velocity coordinates.

$$\vec{v}_e = {C_e^n}^T \vec{v}_n$$