Next Generation Vehicle Positioning Techniques for GPS-Degraded Environments to Support Vehicle Safety and Automation Systems

FHWA BAA DTFH61-09-R-00004

EXPLORATORY ADVANCED RESEARCH PROGRAM

Auburn University
Stanford Research Institute
The Pennsylvania State University
Kapsch TrafficCom Inc.

Final Report

TABLE OF CONTENTS

1. Scope

In an open environment, GPS provides a good estimation of vehicle position. Numerous improvements over the basic GPS framework have provided accuracies in the centimeter range. However, blockages of the GPS signal create significant problems for the positioning solution. In so-called “urban canyons”, GPS signals are blocked by the presence of tall buildings. Similarly, heavy foliage in forests can block line-of-sight to the satellites. Because of these problems, a broader approach is needed that does not rely exclusively on GPS. This project takes into account three key technology areas which have each been individually shown to improve positioning solutions where GPS is not available or is hampered in a shadowed environment. First, terrain-based localization can be readily used to find the vehicle’s absolute longitudinal position within a pre-mapped highway segment – compensating for drift which occurs in dead-reckoning system in long longitudinal stretches of road. Secondly, visual odometry keys upon visual landmarks at a detailed level to correlate position to a (visually) premapped road segment to find vehicle position along the roadway. Both of these preceding techniques rely on foreknowledge of road features – in essence, a feature-enhanced version of a digital map. This becomes feasible in the “connected vehicle” future, in which tomorrow’s vehicles have access to quantities of data orders of magnitude greater than today’s cars, as well as the ability to share data at high data rates. The third technology approach relies on radio frequency (RF) ranging based on DSRC radio technology. In addition to pure RF ranging with no GPS signals, information from RF ranging can be combined with GPS range measurements (which may be inadequate on their own) to generate a useful position. Based on testing and characterization of these technologies individually in a test track environment, Auburn will define a combined Integrated Positioning System (IPS) for degraded GPS environments, which will also incorporate ongoing FHWA EAR work at Auburn in fusion of GPS and on-board sensors. This integrated approach will blend the strengths of each technique for greater robustness and precision overall. This research is expected to be a major step forward towards exceptionally precise and reliable positioning by taking advantage of long-term trends in on-board computing, connected vehicles, and data sharing.

1.1 SRI Contribution

The scope of SRI’s work under Year One of this project is the evaluation of their Visual Aided Navigation System for providing highly accurate positioning for vehicles. As such there are 3 major tasks:

(1) Evaluate and provide a survey of SRI’s existing Visual Navigation results

(2) Integrate Visual Navigation system on Auburn Engineering’s G35 vehicle test platform and collect test data using the integrated system.

(3) Process and analyze the data from the tests and evaluate the performance and recommend any improvements and optimizations.

1.2 The Pennsylvania State University Contribution

The scope of Penn State's contribution focused on extending previous algorithms and analyzing their performance for use in a commercial setting. Specifically, the primary tasks under Penn State's purview were:

(1) Analysis of algorithm performance with *commercial-grade sensors*, rather than defense-grade sensors,
(2) Modification and optimization of the terrain-based localization algorithm, and exploration of alternative approaches, so that localization can take place *online* (in real-time) rather than offline, and
(3) Modification and optimization of the algorithms as well as terrain map representation, so that the localization algorithms work over a *large network of roads*, rather than a small section of a single road alone.

In other words, while the previous algorithms developed at Penn State performed well, they did so using defense-grade sensors and in an offline environment, and their performance was evaluated on single road segment, not a road network. The current study removed each of these constraints sequentially and, after necessary changes, evaluated the performance of the modified GPS-free terrain-based localization algorithms. The following sections provide an outline of the GPS-free terrain-based vehicle tracking algorithm, discuss the details of each of the aforementioned tasks and describe the associated results.

1.3 Kapsch TrafficCom Inc. Contribution

Kapsch will investigate the accuracy of close proximity calculations available from the 5.9 GHz DSRC communications channel. A great deal of information related to positioning can be inferred from the DSRC communications channel. Basic calculations may provide a location region achieved through the channel ranging calculations to more precise lane based proximity determinations through advanced analysis of the communications channel. Kapsch will research a combination of both approaches through available data defined in the IEEE 802.11p standard for 5.9 GHz communication and through scientific Radio Frequency (RF) analysis.

Kapsch will support Auburn for the characterization of the ability to utilize the 5.9 GHz DSRC communication channel for next generation non-GPS localization services. The Received Signal Strength Indication (RSSI) in-conjunction with other aspects of the DSRC communications channel will be analyzed and a method developed for signal ranging. Kapsch does not believe RSSI ranging techniques will fully meet the desired localization needs. Year 2 will yield more advanced algorithms and DSRC equipment capable of providing lane level localization from the DSRC communications channel. This task includes the following sub-tasks:

(1) System Engineering and Deployment of DSRC Infrastructure at the Auburn Test Track

(2) Lab testing of DSRC signal ranging solution

(3) On-site testing of DSRC signal ranging solution

(4) Analysis of DSRC signal ranging test results

1.4 Auburn University Contribution

In an open environment, GPS provides a good estimation of vehicle position. Numerous improvements over the basic GPS framework have provided accuracies in the centimeter range. However, blockages of the GPS signal create significant problems for the positioning solution. In so-called “urban canyons”, GPS signals are blocked by the presence of tall buildings. Similarly, heavy foliage in forests can block line-of-sight to the satellites. Because of these problems, a broader approach is needed that does not rely exclusively on GPS. Auburn will define a combined Integrated Positioning System (IPS) for degraded GPS environments, which will also incorporate ongoing FHWA EAR work at Auburn in fusion of GPS and on-board sensors. This integrated approach will blend the strengths of each technique for greater robustness and precision overall.

2 SRI – Visual Odometry

2.1 Overview of Visual Navigation Approach

SRI Sarnoff has developed a visual approach to navigation that can be deployed on a vehicle and used to determine location and pose in realtime as the vehicle moves through the environment. The first component of the system uses visual odometry algorithms to establish a relative pose and align it with the GPS solution. The second component corrects the pose, which is erroneous due to dead reckoning errors and GPS uncertainty, to enable drift correction and global positioning.

In ***visual odometry***, developed by SRI Sarnoff, distinctive scene features are tracked from frame to frame as the vehicle moves. Analysis of geometry is then performed to estimate both the 3D positions of the features in space, and the precise location and pose (6 degrees of freedom) of the vehicle itself. These relative 3D positions are combined with the IMU’s (MEMs) rotation and acceleration readings in a Kalman filter to produce a dead reckoning pose solution. This is then fused with GPS (if available) to produce an absolute pose solution. With GPS outages, the system can still recover 3D position and carry global positioning information into GPS denied areas. And IMU solution alone is subject to drift and does not provide the precision needed for this application. The IMU corrects short-term errors due to complete camera occlusion or insufficient scene texture, such as in low-light areas.

During long GPS outages, another method is needed to deal with drift due to IMU inaccuracy and dead reckoning errors. Low-bandwidth ***visual landmarks*** are generated on the fly and then matched when a location is revisited. This could be extended to use among multiple vehicles by developing the infrastructure to share landmark databases. The landmark database is represented as a collection of landmark shots, where a ***landmark shot*** is a set of visual features captured at a specific camera location and viewpoint (or camera pose), and each landmark shot acts as a basic unit during positioning based on landmark matching.

This report details the camera-based sensor packages designed for the Auburn University test vehicle, the software that was developed and evaluated during the scope of the project, and provides data analysis, using the ground truth data collected with the Auburn test vehicle.

2.1.1 Visual Navigation Software

The SRI Sarnoff's pose estimation system is shown in the diagram in Figure 1. In addition, the integration with the Auburn vehicle required GPS input from the Auburn GPS unit, plus pose output to the Auburn vehicle computer.

The figure below shows a block diagram of the SRI Sarnoff visual navigation system. The diagram has "IMU", "Front Pair", and "Bck Pair" going into the "IMU-Based Multi-Stereo Visual Odometry (Local Pose)" block. This block points to two other blocks, "Landmark Matching (Global Pose)" and "Kalman Filtering Based Global Pose Estimation". The "Landmark Matching (Global Pose)" block has Landmark Database as an input, and it also points to "Kalman Filtering Based Global Pose Estimation". Finally, the "Kalman Filtering Based Global Pose Estimation" points out for "3D Camera Pose".

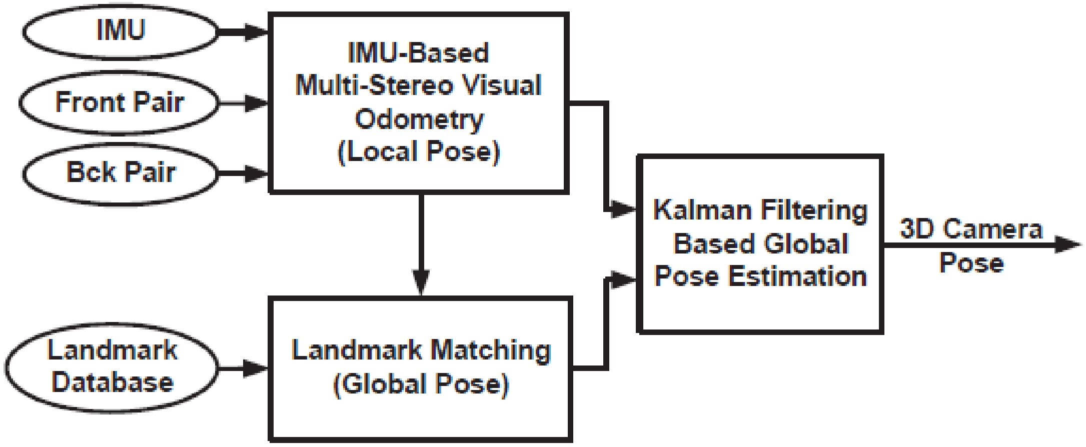


Figure 1: The full SRI Sarnoff visual navigation system

The full SRI Sarnoff visual navigation system includes a multi-stereo visual odometry module to estimate relative pose from one or more stereo camera pairs, a landmark matching module to estimate absolute position from visual features recorded during previous visits, and a Kalman filter to estimate global pose from IMU readings, landmark pose estimates, visual odometry pose estimates, and (optionally) GPS pose estimates.

During the 01/2011 trip to Auburn University, full system functionality was demonstrated including successful live interface with the Auburn vehicle computer.

- **GPS NMEA messages**: The SRI laptop was connected to the vehicle GPS receiver via serial cable and GPS data was streamed to the live visual odometry software.
- **Visual Odometry Poses output over TCP/IP:** The 6 degree-of-freedom position and bearing estimates were successfully streamed to the Auburn vehicle computer over TCP/IP Ethernet connection.

2.1.2 Extended Kalman Filter Model

An extended Kalman filter is used to fuse the measurements from visual feature tracking and a low-cost IMU. Two different formulations for the EKF have been developed at SRI, and both were evaluated using data collected in October, 2010. The first formulation is a velocity process model which specifies an explicit dynamic motion model for a given sensor platform, or assumes a constant velocity process if no dynamic model is given. The second filter representation uses an *error-state formulation*, where the filter dynamics follow from IMU error propagation equations, which evolve slowly over time and are therefore more amenable to linearization. The measurements propagated to the filter consist of the differences between the inertial navigation solution as obtained by solving the IMU mechanization equations and the external source data, which in our case is the relative pose information provided by visual odometry through feature tracking. In general, the second filter representation has proven more accurate for platforms including human-mounted systems and rough-terrain robots.

Analysis of the Kalman Filter in the visual odometry system has led to a new formulation, which provides a better linearization of the model. This is called the **error state formulation**, and it has the following features:

- Kalman filter estimates errors in the state vector, which are then fed back into the IMU mechanization block to obtain the final corrected navigation solution.
- Circumvents the need to employ platform specific dynamic process model.
- IMU accurately captures the high frequency motion, whereas the indirect Kalman filter operates on the inertial system error propagation equations which evolve smoothly and are much more adequately represented as linear.

An approach called stochastic cloning has also been added to our Kalman filter implementation for visual odometry, which provides better integration of the visual tracking measurements:

- Allows proper treatment of the relative measurements from visual tracking.
- Expresses the relative measurement in terms of motion estimates between the two clone states.
- Enables direct use of relative pose measurements
    - Does not require converting low frequency relative measurements to estimated velocities
- Augments the state vector with two copies of the state estimate, one evolving and one stationary clone
- The evolving clone is propagated by the process model (just like conventional Kalman filter framework) whereas the stationary clone is kept static and does not evolve.
- The relative measurement between the previous and current time instant is then expressed as a function of these two states and a Kalman filter update -modified to incorporate the joint covariance of the two clone states- is performed.

During testing we found a few issues with the integration of IMU and Visual Odometry in the kalman filter. During visual odometry failures (low visibility or low texture areas) the filter provides navigation solution solely based on the IMU. 6-DOF pose is derived using the IMU mechanization equations to integrate the gyro and accelerometer readings. However, this is accurate for a brief time and starts drifting very rapidly in the absence of visual odometry poses to correct the errors in the IMU solution. To account for this we have changed the filter to only provide 3-DOF orientation after a few seconds of visual odometry outage. When visual odometry becomes available again, the regular operation resumes.

Another issue we found was that in certain situations we were exceeding the specs of the gyro in the IMU and the measurements were wrapping around creating completely incorrect results. This is illustrated in the figure below.

The figure below shows a plot of corrected and uncorrected gyro measurements. The uncorrected gyro measurement wrap at 300 deg/s. The corrected gyro measurements does not contain the wrapping.

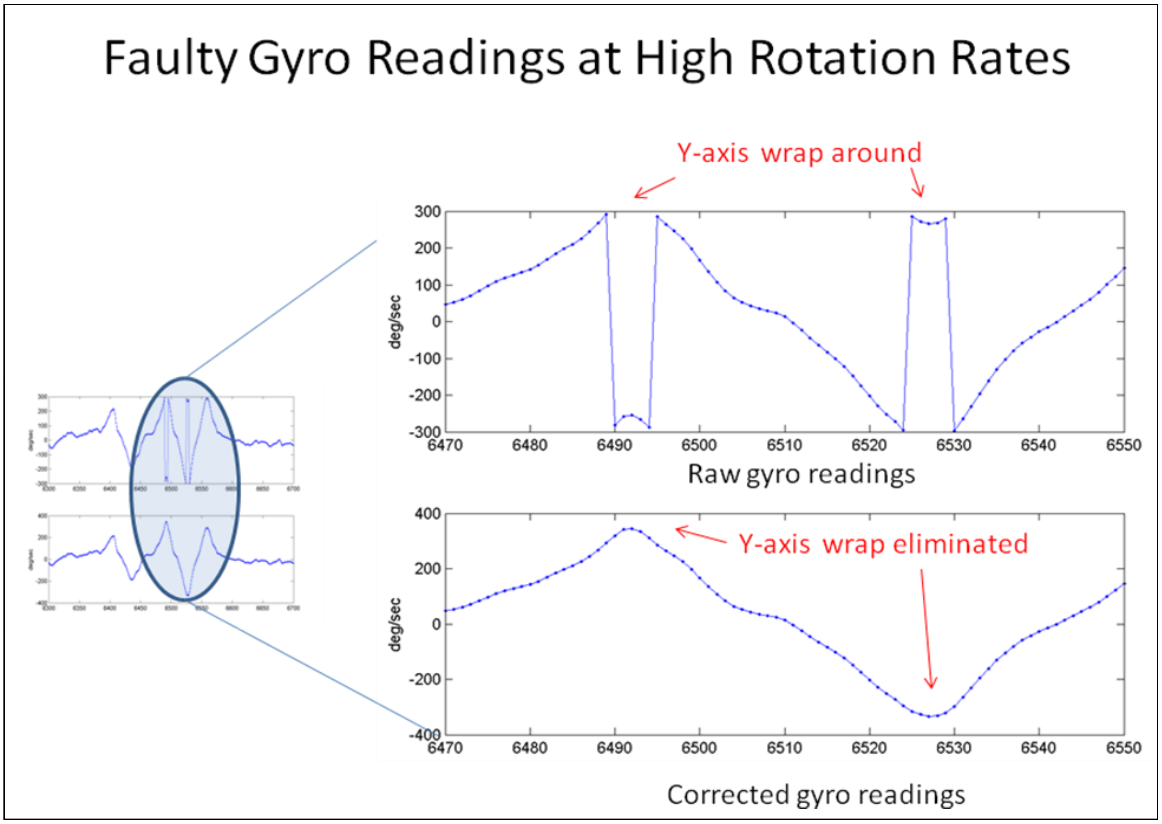


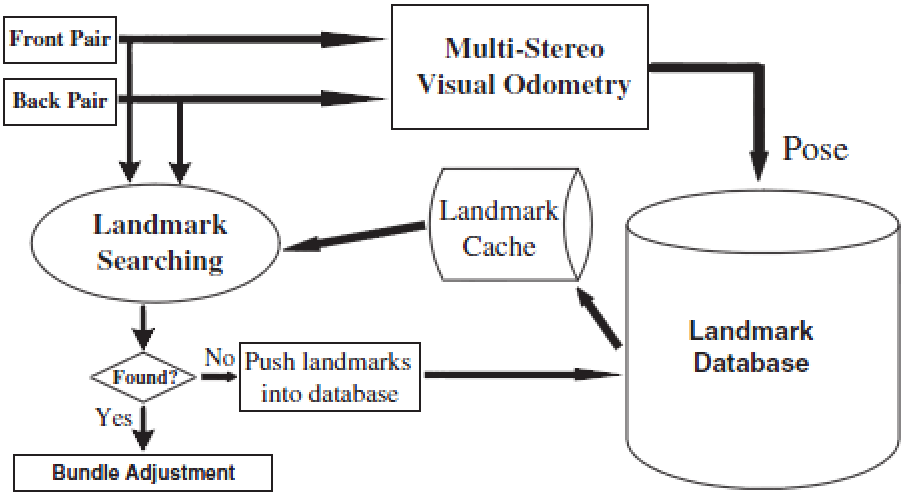
Figure 2: Demonstration of Gyro errors discovered at high rotation rates

2.1.3 Visual Odometry + Landmark Matching Development and Improvements

We have been developing a Visual Odometry + Landmark Matching system to be deployed onto Auburn University's vehicle test platform. Our Landmark Matching system usually loads all the pre-computed landmarks into memory for fastest performance. While this has been an acceptable solution for most of our previous projects it is not very scalable to very large areas, such as those that could be encountered by a vehicle. During testing over larger distances we realized that in order to maintain the scalability of our system we'd have to develop a caching mechanism for the landmark database.

Our approach relies on the fact that at any given time we only need the landmarks that could be visible from our current location. We developed a caching system based off of spatial proximity to the current location of the vehicle. The landmark database is broken up spatially into tiles and then indexed for fast searching by a tile manager. This whole process is done offline during initial landmark database computation and does not diminish the performance of our live system. When running live, the tile manager loads the tiles adjacent to our current and possible future locations based on the current movement properties of the vehicle. In between landmark matches, our visual odometry system keeps track of the movement of the vehicle. Since we know the drift characteristics of the visual odometry we can increase the radius of tiles to load based on possible drift over time and distance travelled since last landmark match. In the figure below you can see a representation of the system with Landmark Match caching.

The figure below shows a block diagram of the Landmark system. “Front Pair” and “Back Pair” point to Multi-Stereo Visual Odometry” and “Landmark Searching”. “Multi-Stereo Visual Odometry” points with Pose to “Landmark Database”. “Landmark Database” points to “Landmark Cache”, which in turn points to “Landmark Searching”. “Landmark Searching” points to the conditional “Found?”. If Yes, the conditional points to “Bundle Adjustment”. If “No”, the conditional points to “Push landmarks into database” which in turn points to “Landmark Database”.



The figure below shows a block diagram of the Landmark database and visual odometry. In an offline process, the “Landmark Database” points to “Tile Manager”, which in turn points and is pointed from “External Storage”. “Landmark Cache” also points and is pointed from “Tile Manager” with Current Pose pointing to “Tile Manager” and Tile containing the shot pointing to “Landmark Cache”. “Visual Odometry” points to “Landmark Cache” with Current Pose. Landmark Matching points to and is pointed from “Landmark Cache” with ShotReturned pointing to “Landmark Matching” and “ShotRequest” pointing to “Landmark Cache”. Finally, “Landmark Cache” points to “Cache Cleanup” with CurrentPose.

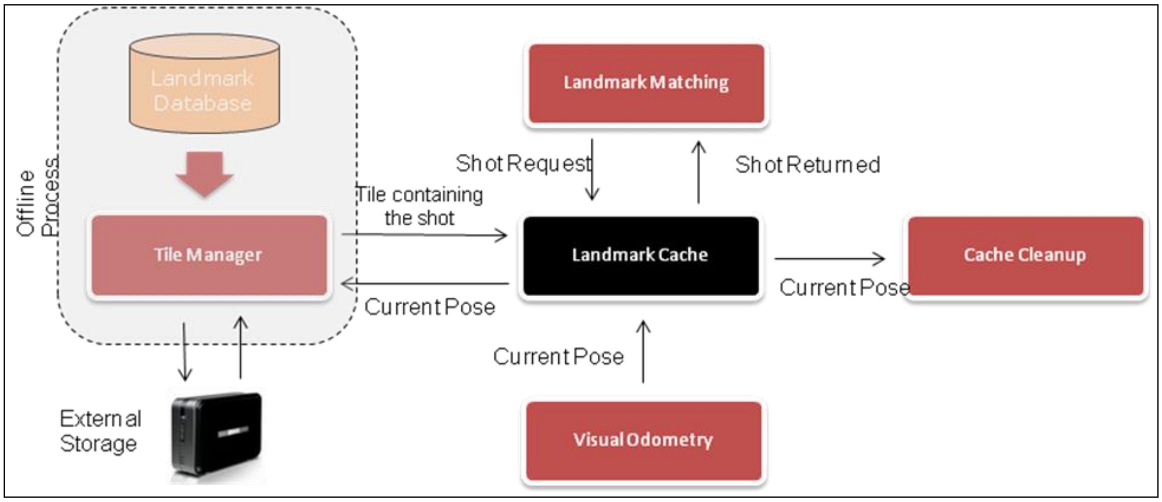


Figure 3: (Top) Flow diagram for visual odometry and landmark matching. (Bottom) System relationships between visual odometry and landmarking database using caching.

Visual landmarking gives absolute 3D positioning from landmark databases recorded and augmented on the fly.

- **Landmark image**: a constellation of HOG features, each associated with a 3D point (from stereo)
- **Landmark database**: a collection of automatically selected landmark images, referenced by the 6DOF viewing pose.
- **Landmark matching**: retrieving and recognizing a landmark image (uses vocabulary tree and spatial caching for speed), then estimating new viewing pose.

The figure below shows three images. The top left image is headed by "Dynamic Landmark Database and shows a picture of a field. The top right image points to "Dynamic Landmark Database", is headed by "Inspection Run", also points to the bottom image, and shows a similar scene (but not the same picture). The bottom picture consists of two images side by side of a scene and is headed by "Landmark Based Retrieval of Reference Image" with red and green dots representing the landmarks.

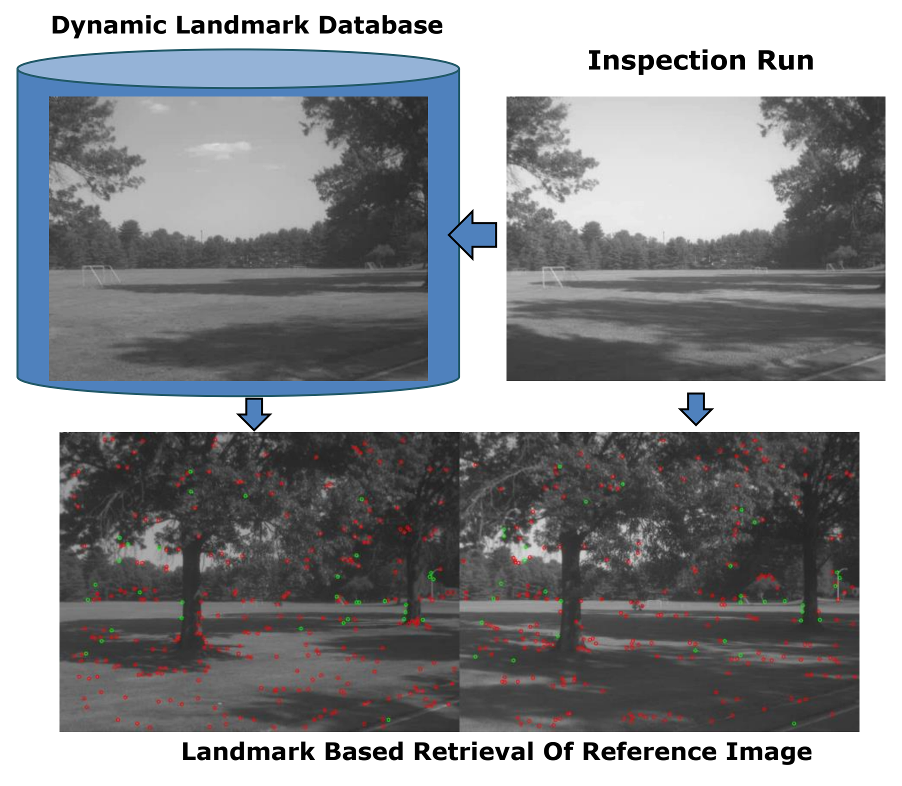


Figure 4: A landmark database is built automatically from 2D features and 3D features and view-based 6DOF poses.

Once the landmark is built, absolute positioning can be done by matching the current view to a reference image in the database and inferring the new viewpoint.

To create a visual landmark database, there are 4 major steps:

- **Feature detection and key frame selection:** key frames (landmark images) are selected in the way so that the baseline between two consecutive key frames is larger than a threshold. Harris corners are detected in each key frame and are described with HoG descriptors
- **Image retrieval based on Vocabulary Tree:** an index structure based on Vocabulary Tree is built on the fly when new landmark images are inserted into the database. For each new landmark image, the images in the existing database that can be matched with it are detected, and its initial pose is estimated
- **Feature track detection:** from the matching relationship between landmark images, long feature tracks are detected. Each feature track is associated with a 3D landmark point, and a set of 2D image measurements in multiple landmark images
- **Bundle adjustment:** the camera poses of the landmark images and the 3D locations of the landmark points are optimized with a non-linear bundle adjustment algorithm

2.2 Visual Odometry Interfacing

SRI Sarnoff designed a sensor array for mounting on, and integrating with, the Auburn University test vehicle. The purpose of the sensor array is to demonstrate visual navigation capabilities in a monitored and controlled test environment, and to allow integration with the Auburn University Integrated Positioning System in phase 2 of the FHWA program.

The basic design requires at least 2 cameras in a stereo configuration plus a low-cost, MEMS-based IMU. In addition, the sensors needed to be powered from the Auburn vehicle, and a computer needed to be configured for running data collection software and live visual positioning software. Communication with the main Auburn vehicle computer was designed to allow transmission of pose estimates and GPS data.

2.2.1 Initial Sensor and Hardware Design

SRI Sarnoff's plan for integrating a Visual Navigation system on Auburn Engineering's Vehicle Test Platform is as follows:

(1) Design 2 sensor bars, each holding 2 cameras and potentially an IMU (lost cost MEMS-based)
(2) Collaborate with Auburn Engineering to design mounting points for the sensor bars onto Vehicle Test platform such that it does not interfere with other sensors on the vehicle
(3) Build and calibrate individual sensor bars
(4) Design and build computer data collection system to be installed in vehicle and left at Auburn for duration of tests.
(5) Create software testing system and procedures
(6) Integrate, calibrate and test whole system onto Auburn's Vehicle Test Platform

Figure 5 shows the simple bar design that Sarnoff's sensor package will employ. 2 cameras and one MEMS-based IMU will be mounted on each bar. Then 2 bars will be mounted on the front and back T-slot frames on the Auburn Vehicle's roof rack. The bars will be mounted from the bottom of the T-slot frame and spaced such as to avoid interfering with the existing Auburn equipment. Cabling from each bar will follow the same cable routes as the existing equipment.

Each sensor bar will consist of 2x AlliedVision Marlin F-033B cameras and a CloudCap Crista IMU. The stereo baseline will be about 40cm. Firewire, trigger, power and serial cables will be bundled from each bar and routed inside the vehicle where the computer and trigger device will be located. Data will be able to be recorded or processed live on a Windows PC (small form factor or laptop).

The figure below shows the SRI sensor bar. The Thule rack holds up the T-slot frame, which holds the cameras in place.

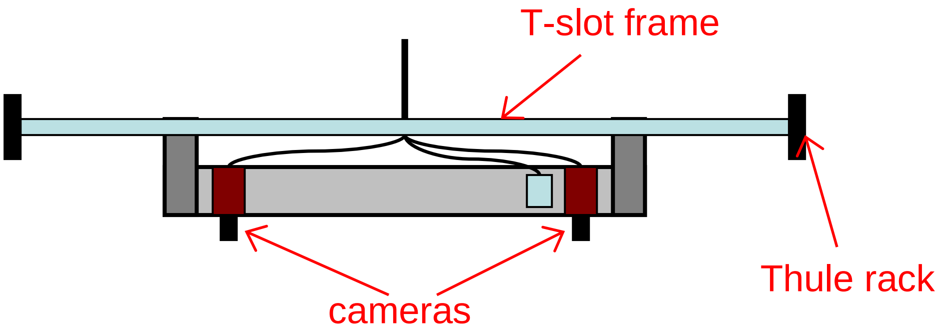


Figure 5: Basic sketch of SRI's sensor bar as attached to Auburn's Vehicle roof rack.

The figure below shows a photograph of the Thule rack and T-slot frame on Auburn's vehicle.

Figure 6: Auburn Engineering Vehicle Test Platform roof rack and mount points for SRI Sarnoff sensor bar

2.2.2 Final Sensor and Hardware Design

The final selection of sensor components, mounting hardware, and packaging was completed in September 2010. To enable high frame rate imaging and reliable long duration data collection, SRI

Sarnoff chose the Allied Vision Prosilica GC1380 cameras (see Figure 7), which use Gigabit Ethernet Interface (faster and more robust when compared to FireWire interface). These cameras will collect 30 fps at 640x480, and have very good image quality.

The figure below shows the GC1380 camera from GigEvision

Figure 7: Allied Vision Prosilica GC1380

The IMU is a low-cost (under $1K) MEMS-based unit from Cloud Cap. It has high-resolution angular rate and acceleration data which is transmitted over serial interface.

The figure below shows a photograph of the Cloudcap Crista IMU

Figure 8: CloudCap Crista IMU

To support the high frame rate and Gigabit Ethernet interface of the 4 Prosilica cameras, a Gigabit Ethernet hub was needed. We use the Netgear GS105NA, because it has jumbo frame support and 5 ports.

The visual navigation software is computationally intensive when run at high frame rate using front and back camera sets, so a computer with multiple cores that can handle Gigabit input bandwidth is necessary. We chose the Clevo Gaming Notebook, which has an Intel quadcore i7 processor and can handle high bandwidth Ethernet input.

2.2.3 Components

The components used in the sensor package include cameras, IMU, Ethernet hub, trigger circuit, cabling, and laptop computer.

- Cameras (4) - Allied Vision Prosilica GC1380
    - GigaBit Ethernet interface
    - 640x480 (after 2x2 binning)
    - 30 fps,
    - Sony ICX285 CCD, monochrome
- Lenses (4) - Kowa LM6JC
    - 6.0 mm/F1.4
- IMU (1) – CloudCap Crista
    - 100 Hz operation
    - 10x oversampling
- Ethernet hub (1) – Netgear GS105NA
    - 5 RJ45 ports
    - Jumbo frame support to 9720 bytes
- Cabling and connectors
    - Weather proof RJ45 connectors
    - Shielded CAT6 cable
    - Mil-style 10 pin connectors
- Computer (1) – AVA Direct Clevo D900F
    - Intel quadcore i7, 3.33 GHz

2.2.4 Design

The sensor payload was designed towards several criteria. The package consists of a front camera box, a rear camera box, and a power and data breakout box. The front and rear boxes need to be rigidly connected and it must be possible to remove the boxes without detaching them from this rigid coupling. The boxes must be positioned on the vehicle such that they are not occluded by, nor do they occlude, the existing sensors, racks, and the vehicle itself. In addition, the boxes must be weather proof and robust to vibration and heat variation.

The final camera box design uses a ¼” thick custom-built aluminum plate to mount the cameras and imu. This plate provides stability for the stereo camera configuration as well as functioning as a heatsink to prevent the cameras and ethernet hub from overheating in the sealed enclosure. The enclosure is an aluminum box with built-in RJ45 connectors and 10-pin connectors. Please see the wiring diagram and CAD design, Figures 9 and 10, for details.

The figure below shows the Sarnoff camera setup. The front camera enclosure shows two cameras and an IMU. Various signals are sent to the back camera enclosure, where two cameras and a gigabit hub can be seen. The power of the system comes out of a breakout box, and the laptop is connected to the remaining signals.

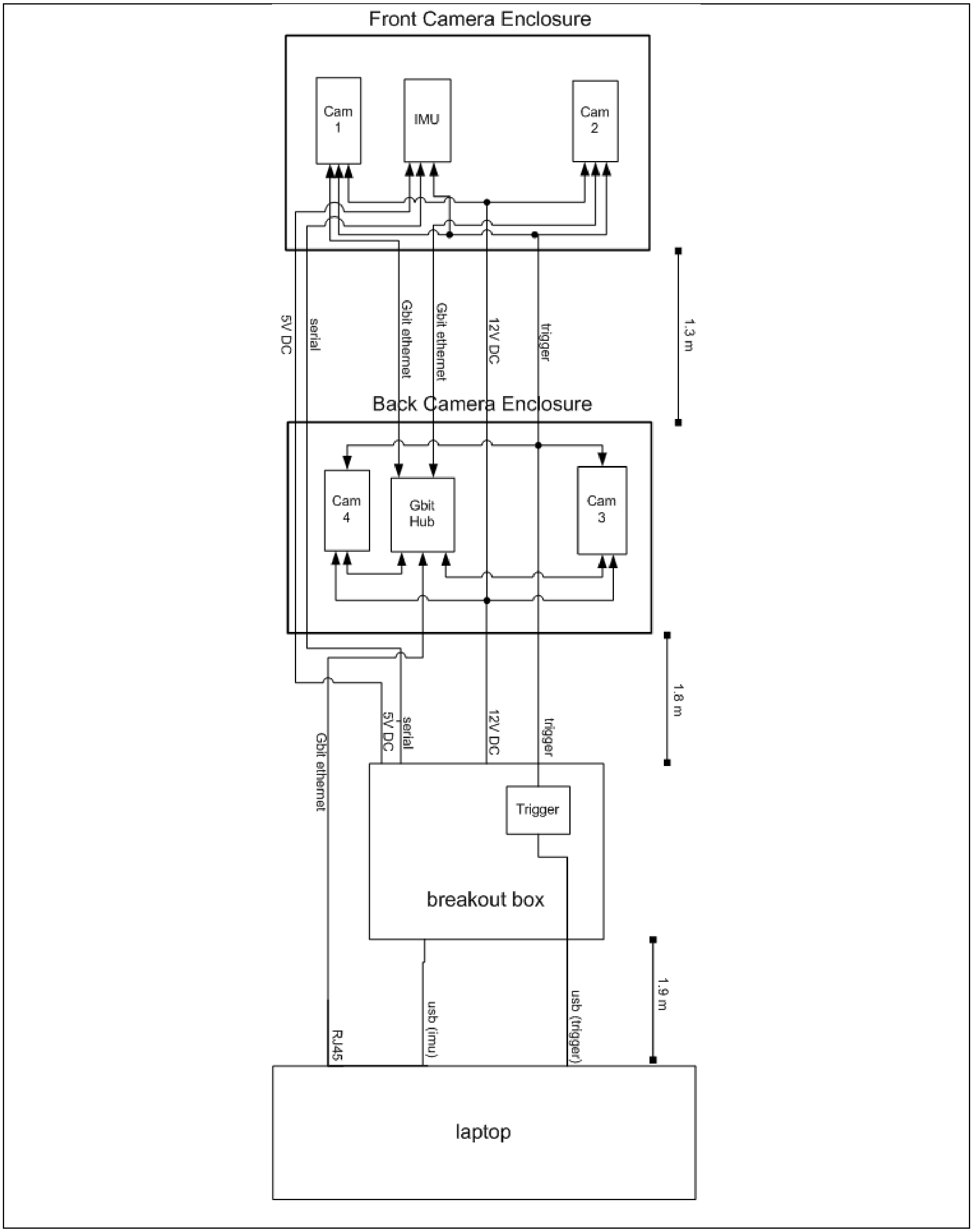
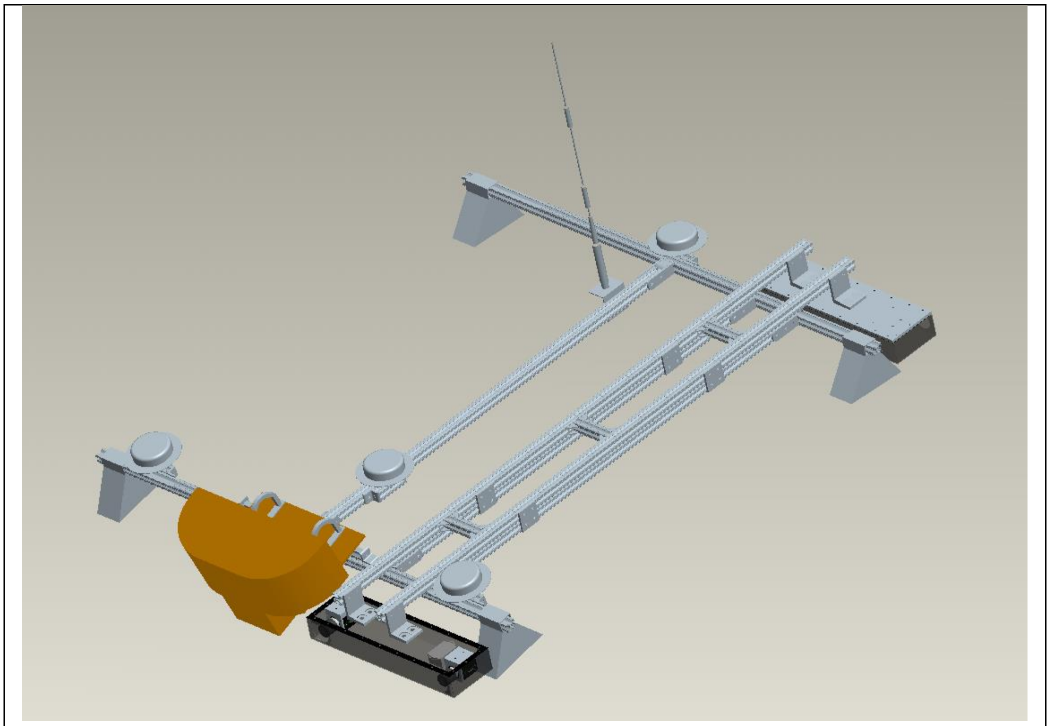

Front Camera Enclosure
Cam 1
IMU
Cam 2
5V DC
serial
Gbit ethernet
Gbit ethernet
12V DC
trigger
1.3 m
Back Camera Enclosure
Cam 4
Gbit Hub
Cam 3
1.8 m
Gbit ethernet
5V DC
serial
12V DC
trigger
Trigger
breakout box
1.9 m
RJ45
usb (imu)
usb (trigger)
laptop

Figure 9: Wiring diagram and physical layout of components in the visual navigation sensor package

The CAD model can be seen below with two images – one with an isometric view and one with a front view. For the isometric view, two bars extend parallel. A center bar joins the two bars. On the front bar, two antennas can be seen on either end, with the LiDAR visible at the middle of the front bar. The rear view bar has a Septentrio antenna at its center. Near the front on the middle bar is another Sepentrio antenna. Near the back on the middle bar is an antenna. The Sarnoff mount can be seen to the right of the middle bar. The mount extends from the front bar to the back bar with a double layer of 80x20. Three more bars keep the two bars combined at an equal distance. The cameras hang on either end of the Sarnoff bars and are pointed to the back and the front of the vehicle. For the front view, the LiDAR dominates the front center of the image. A piece of 80x20 bar stretches to either side of the image. Two antennas for the Septentrio are visible at the ends of the bar. The camera is visible as hanging down the right side of the bar below the antenna. Above the LiDAR, an antenna is visible from the back of the vehicle.



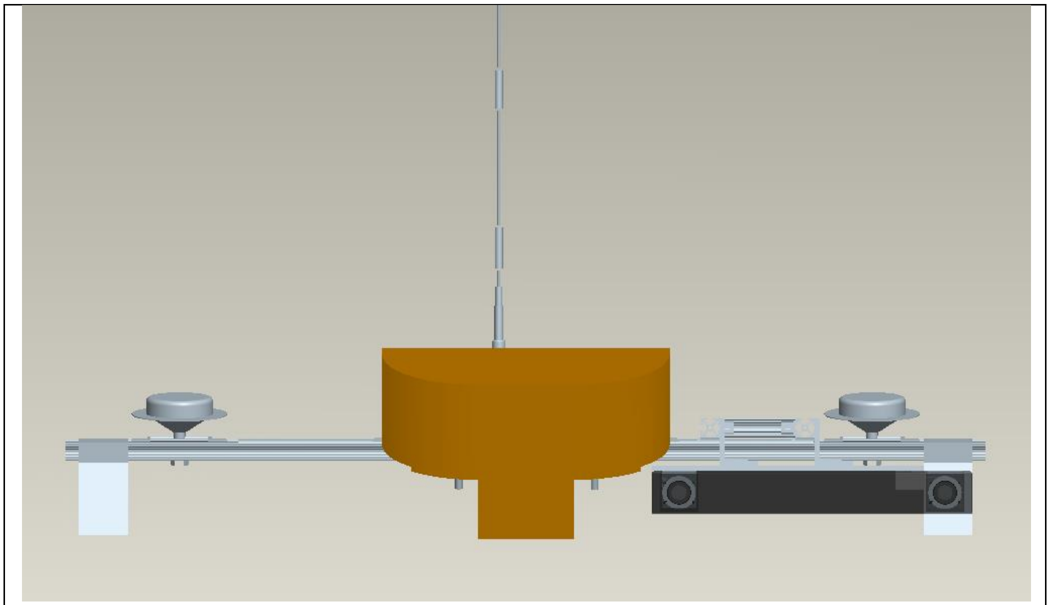


Figure 10: CAD model of the front and rear camera enclosures, shown in relation to the existing roof

The front camera enclosure sits above the front windshield, on the driver's side. The back camera enclosure sits above the rear windshield, on the driver's side. The power breakout box is mounted in the trunk. The laptop can be placed in the front passenger seat. Data and power cables must be run from front sensor box to the back, from the back sensor box to the trunk, and from the trunk to the processing laptop in the passenger compartment. The close tolerances afforded by the roof rack and the existing sensors, plus consideration to preserve an unobstructed field of view of the cameras, meant that it was difficult to ensure good placement of the boxes.

2.2.5 Installation

The visual navigation sensor package was delivered to Auburn University and installed on the test vehicle on 10/3/2010.

The sensor payload installation on the Infiniti test vehicle was begun on 10/3/2010 and finished on 10/4/2010. The existing T-slot bars needed to be replaced with longer bars (provided by Auburn University) at time of installation because the connectors to the rear sensor box were obstructed by the rear windshield. 5V and 12V power was supplied by the Auburn vehicle's power supply panel.

A full calibration was performed at time of installation. Intrinsic model parameters were calculated for each camera using the Bouguet calibration method. Extrinsic parameters (relative position and orientation in 6 degrees of freedom) were calculated between the cameras in the front stereo pair, the cameras in the back stereo pair, the front left camera and the IMU, the back left camera and the IMU, and the front left camera and the vehicle center. The extrinsic calibration between the back left camera and the IMU did not converge to a valid solution during the trip to Auburn University, so raw data was collected (video and IMU data) and the calibration was recalculated by SRI Sarnoff.

2.2.6 Modified Sensors and Hardware Design

SRI Sarnoff visited Auburn University on 1/16/2011 - 1/18/2011 for data collection and system tests. The sensor system installed at Auburn University on 10/3/2010 was found to be breached and have water damage. The rear cameras were non-responsive and the glass lens covers were fogged with internal condensation for both front and rear cameras. Both camera boxes were resealed with external application of RTV adhesive, and data was collected on the Auburn test track in rainy conditions.

The camera boxes and breakout box were shipped to SRI Princeton, where it was found that the sealant used to secure the glass lens covers had failed, letting rainwater into the boxes. The GigE network hub and one camera were damaged. The front camera box will be returned to Auburn for continued testing and inclusion in the integrated positioning system. The rear camera sensor box will not be used further.

The front camera box was rebuilt and returned to Auburn on 02/15/2011, with new cabling, network hub, and cameras installed. The configured laptop was also sent to Auburn at that time. The cameras were mounted on the vehicle and calibrated by Chris Rose at Auburn. Data was collected on the test track at varying speeds and lighting conditions and raw video, IMU, and ground truth GPS data was collected.

The figure below shows the camera diagram. In the front camera enclosure, two cameras and an IMU are present. Six signals come out and head into the breakout box block – 5V DC, Gbit Ethernet x2, 12VDC, serial, and trigger. The breakout box has Gbit Hub with RJ45 from the laptop block and the two Gbit Ethernet signals as inputs. The trigger block has the trigger and usb from the laptop connected. Finally, the serial line points to the laptop.

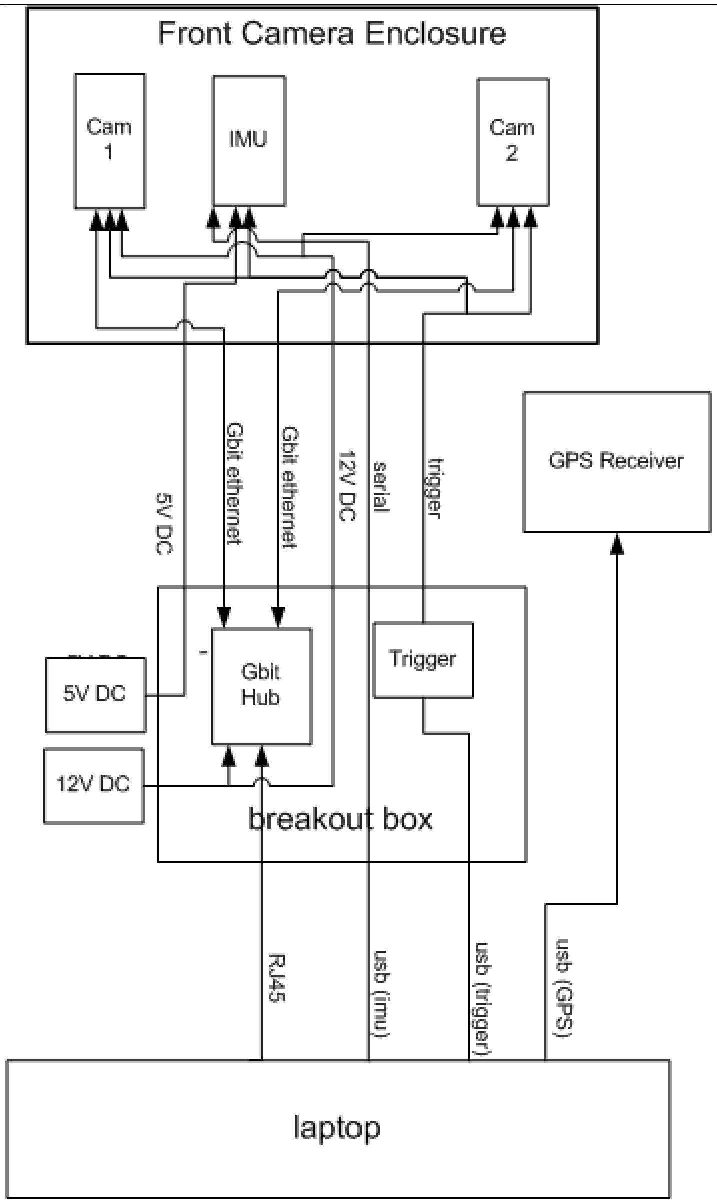


Figure 11: Wiring diagram for the front-only camera system

2.3 Visual Odometry Results

2.3.1 Data Analysis 1: Rainy conditions

Data was collected at the Auburn test track in rainy conditions on 01/17/2011, and analysis of the data is ongoing. RTK position measurements were recorded during these data collects. The data was collected from the front stereo cameras at 30Hz, from the Cloudcap Crista IMU at 100Hz, and from the Septentrio GPS at 10Hz. The rain caused significant blurring in the image (shown in Figure 12), thus decreasing the number of tracked features per frame, but this did not cause visual navigation to fail.

The figure below shows two stereo images. The top image shows the entrance to the NCAT track with water droplets clearly visible from the perspective of the cameras mounted to the vehicle. The bottom image shows the track with water droplets clearly visible from the perspective of the cameras mounted to the vehicle.

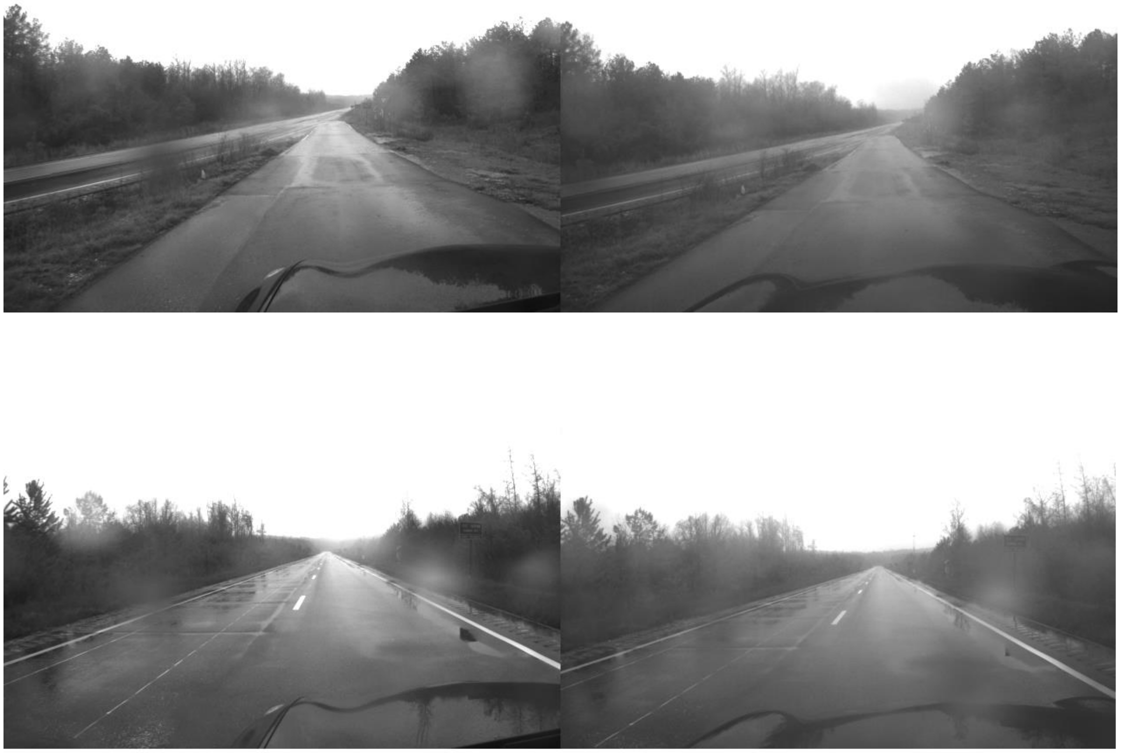


Figure 12: Image samples from front stereo pair. Rain caused image blur and specularities. Fogging of right image was caused by condensation between CCD and lens.

The figure below shows four images. The top left image shows the entrance to the NCAT track and the top right image shows the track with water droplets clearly visible from the perspective of the cameras mounted to the vehicle on both. The bottom right and bottom left images show the track with water droplets clearly visible from the perspective of the cameras mounted to the vehicle. All four images have the landmarks visible.

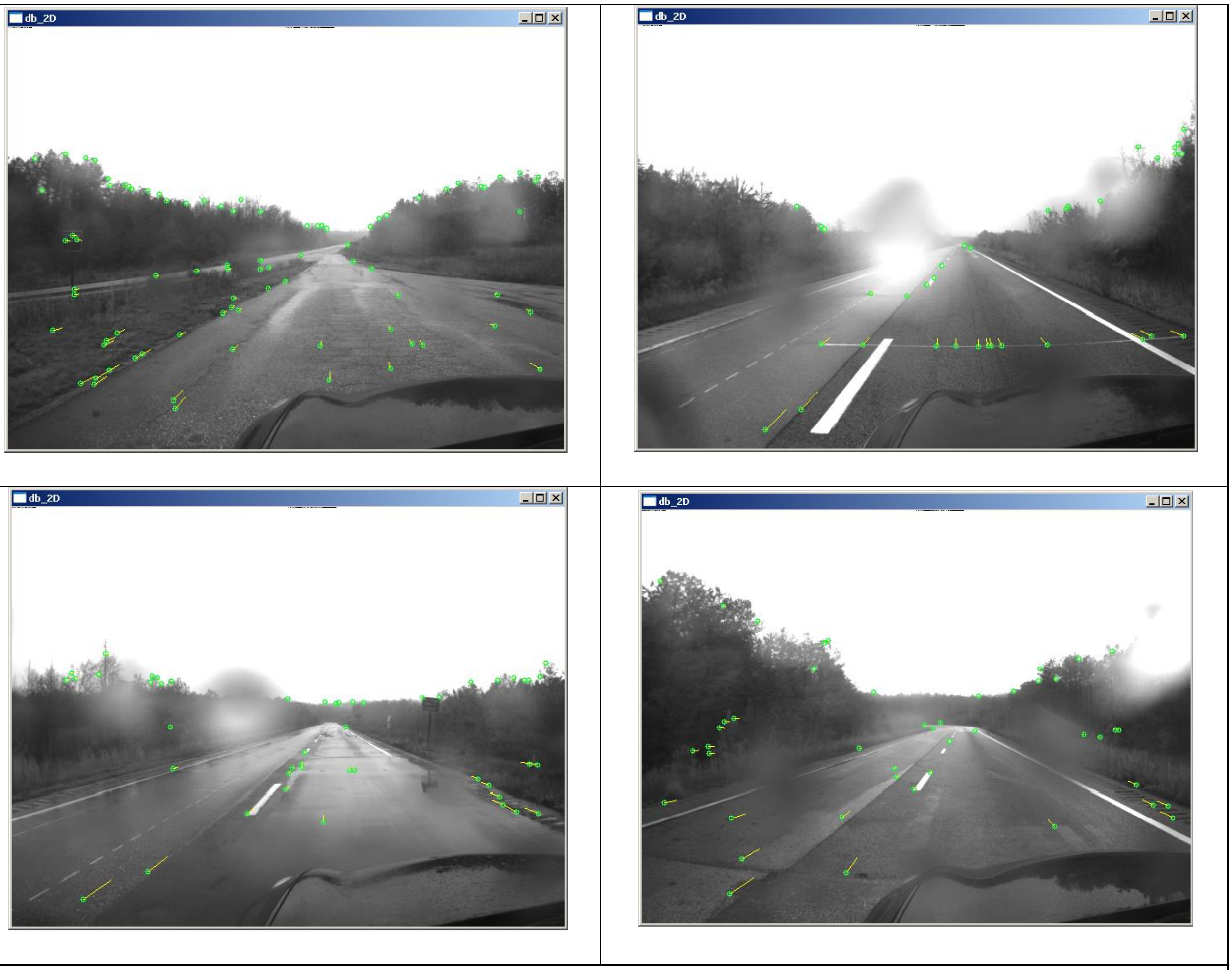


Figure 13: Feature tracking on rand-blurred image samples. Even on the most affected images, there are enough features accurately tracked to estimate relative position.

The figure below shows three graphs of the results of the positioning for the system. The top graph shows a graph ranging from -1200 to 0 in the y axis and -200 to 600 in the x axis. The start of the positions begins at (0,0) and wraps around like a stretched paperclip (due to the odometry) and ends at (400,-200). The second graph shows the system in utm with gps updates with the shape of a horizontally stretched oval with some deviations. The third plot is similar to the second plot but with fewer deviations.

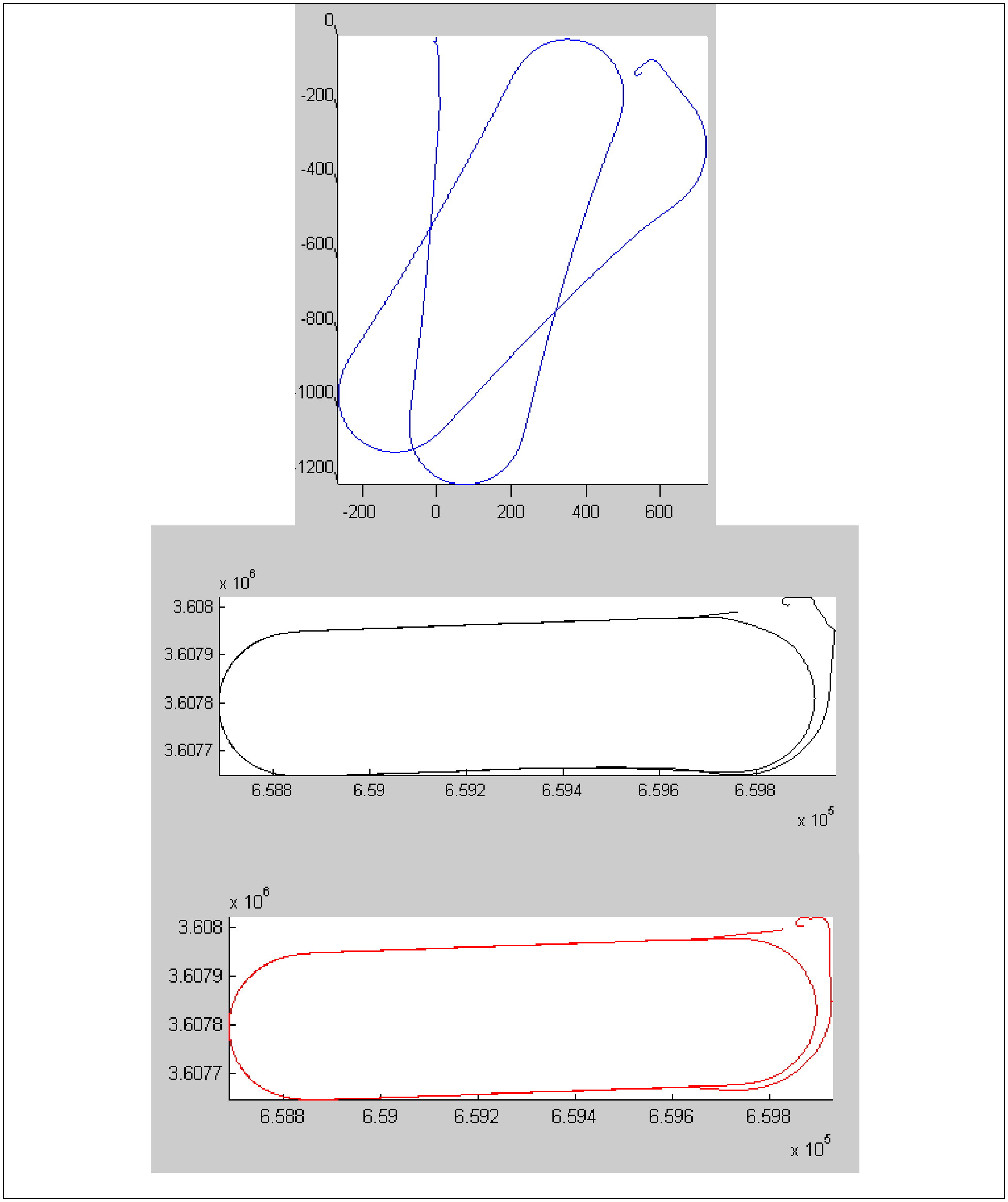

0
-200
-400
-600
-800
-1000
-1200
-200
0
200
400
600
x 10^6
3.608
3.6079
3.6078
3.6077
6.588
6.59
6.592
6.594
6.596
6.598
x 10^5
x 10^6
3.608
3.6079
3.6078
3.6077
6.588
6.59
6.592
6.594
6.596
6.598
x 10^5

Figure 14: Poses, shown in 2D, from visual odometry running while driving around track (two laps) in moderate rain.

Top: visual odometry trajectory in rain without gps input (local coordinate frame)

Center: visual odometry trajectory with gps input with simulated outages (30%, in 3 chunks)

Bottom: visual odometry trajectory with gps input (UTM coordinate frame)

2.3.2 Data Analysis 2: Visual Odometry plus Degraded GPS

The data collected by Auburn Univ on 4/18 was taken for 2 purposes. One, to assess various synchronization and configuration issues using data that includes quick starts and stops and swerves. Rapid acceleration, deceleration, and roll and pitch changes are used to locate synchronization problems between the cameras, the IMU, and the GPS data.

The remainder of the data collected on 4/18 was used to assess the accuracy of visual odometry over different speeds and different lighting conditions. Artificial degradation of the ground truth GPS data was done by adding uniform translations of 2m, 4m, and 6m, and reducing the data rate to 1Hz (from 10Hz). Visual odometry alone was also considered. Drift rates were computed for each case by comparing position of the visual odometry at each frame with ground truth position.

The figure below shows a graph of total drift over different GPS data. The 1Hz, 6m data stays below 2 meter drift over the 10 mph, 30mph, 50mph and low lighting data. The 1Hz, 4m data remains slightly above 2 meters for the 30mph, 50 mph, and low lighting. The 1Hz, 2m data is at 4m for the 10 mph run, 2m for the 30mph run, 10m for the 50 mph run, and 7m for the low lighting run. Finally, for the 1 min GPS outage, the data is at 4m for the 10mph run, 3m for the 30 mph run, 6m for the 50 mph run, and 7m for the low lighting run.

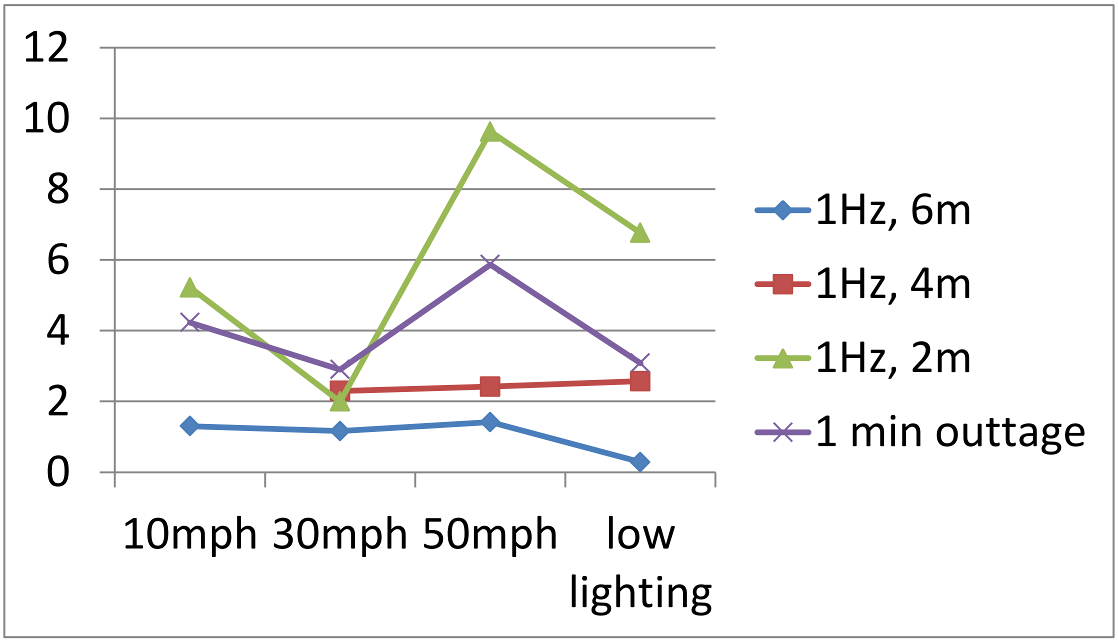
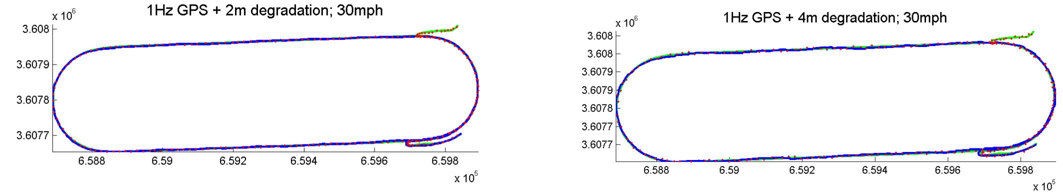


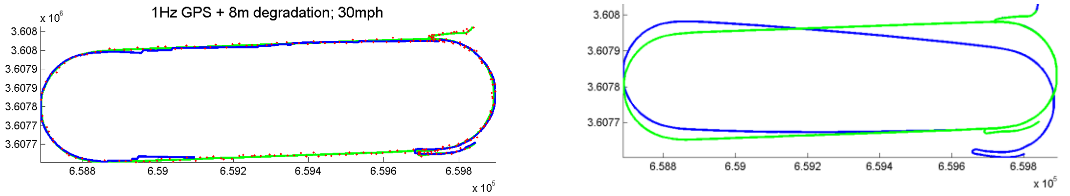
Figure 15: Comparison of drift over multiple data sequences with different GPS noise levels

Four plots are shown below which show a bird's eye view of the test track when visodo has degraded GPS at 30mph. The top left plot shows little drift for low noise. The top right plot shows similar results for the 4m degradation. More drift can be seen in the 8m degradation in the lower left plot, and finally significant drift is seen in the lower right plot for no GPS.



drift: total 1.16m, mean 2.007m
(over 4305 m traveled distance)

drift: total 2.296m, mean 3.21m
(over 4305 m traveled distance)

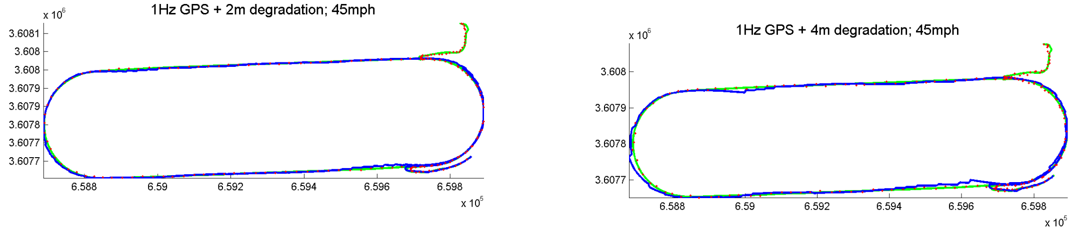


drift: total 1.995m, mean 4.572m
(over 4305 m traveled distance)

drift: total 99.95m, mean 31.201m
(over 4305 m traveled distance)

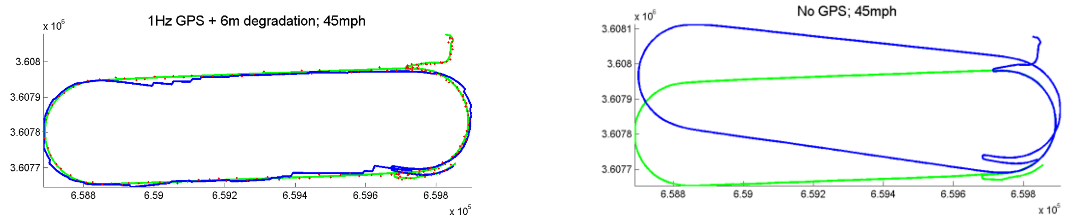
Figure 16: Results of visual odometry plus degraded GPS at 30 mph.

Four plots are shown below which show a bird's eye view of the test track when visodo has degraded GPS at 50mph. The top left plot shows little drift for low noise. The top right plot shows similar results for the 4m degradation. More drift can be seen in the 6m degradation in the lower left plot, and finally significant drift is seen in the lower right plot for no GPS.



drift: total 1.415m, mean 2.857m
(over 4305 m traveled distance)

drift: total 2.419m, mean 4.981m
(over 4305 m traveled distance)



drift: total 9.629m, mean 9.258m
(over 4305 m traveled distance)

drift: total 21.83m, mean 60.56m
(over 4305 m traveled distance)

Figure 17: Results of visual odometry plus degraded GPS at 50mph

2.3.3 Data Analysis 3: Visual Odometry plus Landmark Database Matching

In the experiments on the data collected by Auburn University on 4/18, we created a visual landmark database from the data sequence obtained at the vehicle speed of 30mph. The landmark database was tested on the rest data sequences collected at different vehicle speed (see Figure 19).

The figure below shows a block diagram of the landmark database system. The input show Input Video which goes into the "Multi-Stereo Visual Odometry" block, which points to the "Topology Inference" block. That block points to the "Bundle Adjustment" block, which finally points to the "Landmark DB".

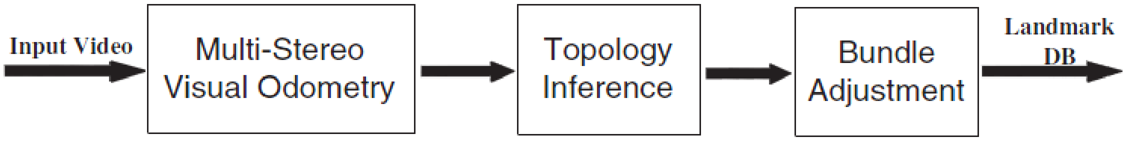
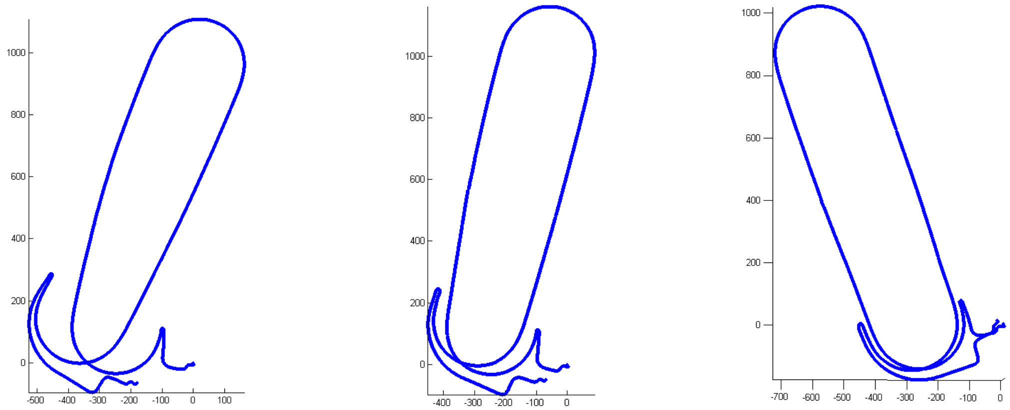


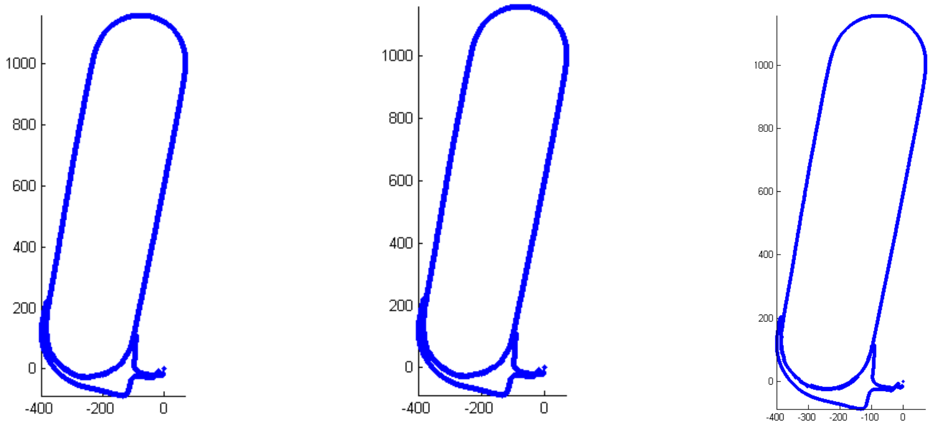
Figure 18: Diagram showing the process for creating a Landmark Database.

Figure 19 shows the trajectories generated at different vehicle speed without GPS. The trajectories at different speed without visual landmark database are displayed in the first row, and the corresponding trajectories generated with visual landmarks are shown in the second row.

The figure below shows six graphs of the positions on the track with the typical oval shape but with deviations. The first row of three graphs is without the landmark with increasing speeds. The slower speeds show greater deviations, but with the 45 mph speed inclined to the left. The second row of three graphs shows the graph with the landmark points with fewer deviations.



10 mph without landmark 30mph without landmark 45mph without landmark



10 mph with landmark 30mph with landmark 45mph with landmark

Figure 19: Trajectories generated at different vehicle speed with or without visual landmark

In Table 1, the loop closure errors are compared. We can clearly see that with visual landmarks, loop closure errors are significantly reduced.

	10mph	30mph	45mph
Without landmark	207.9 meter	112.8 meter	123.4 meter
With landmark	18.6 meter	18.6 meter	18.6 meter

Table 1: Comparison of loop closure errors at different vehicle speed with/without visual landmark

In the above experiments, we did not integrate GPS information because the current landmark coordinate system is different from the GPS coordinate system. In the future work, we will convert the visual landmark database into the GPS coordinate system, and also exploit precise GPS positioning in creating more accurate visual landmark database.

3 Penn State – Road Fingerprinting

Recent advances in vehicle safety and driver assist systems will necessitate the use of multiple localization techniques for precise positioning in order to maintain redundancy and ensure robust performance. One potential technique that may be used in degraded GPS environments or in the absence of GPS, as an input to a multi-faceted localization approach, is terrain-based localization. Terrain-based localization offers the advantages of being able to function during all times of the day and in a large variety of weather conditions, as well as being passive in nature, though it does require that a map of the terrain to be known before-hand. Penn State's GPS-free terrain-based localization algorithms (Dean & Brennan, 2009) utilize real-time in-vehicle terrain inertial measurements, such as pitch, roll, yaw, accelerations, etc., and compare these against a pre-recorded terrain map in order to determine the vehicle's current position. Prior work has shown that this approach can give offline position estimates in the direction of travel to approximately 1 meter of accuracy. This study sought to build upon the previous algorithms and assess their performance in a less-constrained and more commercially viable setting.

3.1 GPS-free Terrain-based Vehicle Tracking Algorithm

This section provides an overview of the GPS-free terrain-based vehicle tracking framework. At the heart of the framework is the idea that, as a vehicle travels over a road, attitude measurements (pitch, roll, yaw, accelerations or combinations thereof) obtained from an inertial measurement unit are sent to the tracking algorithm, which then correlates them to a pre-recorded terrain map. Specifically, the algorithm uses an Unscented Kalman Filter (UKF) to track the current position of the vehicle (Wan & van der Merwe, 2000) (Williams, Dissanayake, & Durrant-Whyte, 2001). It is assumed that the initial vehicle position estimate and covariance are available before the algorithm is initiated, presumably from a GPS signal before it fades or disappears.

The current position (or state) of the vehicle is defined as the distance of the vehicle from the last traversed intersection. The system model is given by (1), which represents the propagation of the system state, i.e. the current position of the vehicle:

$$x_{k+1} = x_k + u_k + w_k \tag{1}$$

where x_{k+1} denotes the position of the vehicle at time $k+1$, x_k denotes the position of the vehicle at time k, u_k denotes the distance moved by the vehicle as determined through odometry, and w_k denotes the zero-mean Gaussian process noise in odometry measurements. The measurement model is given by (2), which relates the current position of the vehicle with the corresponding attitude measurement.

$$y_k = f_{NL}(x_k) + v_k \tag{2}$$

where y_k denotes the attitude measurement at time k and is a nonlinear function of the vehicle position at time k, and v_k denotes the measurement noise at time k. The nonlinear function (or *terrain map*) is defined as a lookup table stored in a database that relates the position along a road to a pre-recorded attitude measurement.

An Unscented Kalman Filtering framework is used to estimate the current vehicle position in a GPS-denied or degraded GPS environment, as discussed in (Dean, Langelaan, & Brennan, 2009). A short overview of the algorithm is included here. The sigma points are initialized according to (3) as follows:

$$X_{k-1} = [x_{k-1}, x_{k-1} + C, x_{k-1} - C] \tag{3}$$

where $C = \alpha\sqrt{N_s \cdot P_{k-1}}$, α is an algorithm parameter, N_s is the number of system states, and P_{k-1} is the covariance of the position estimate at time $k-1$. A list of the parameter values used in the implementation of the algorithm during the project is included in Appendix 1. The time update (or *prediction*) step propagates the system states forward in time according to (4), (5) and (6) as follows:

$$X_k^- = X_{k-1} + u_k \tag{4}$$

$$\hat{x}_k^- = X_k^- \cdot W_m \tag{5}$$

$$P_k^- = (X_k^- - \hat{x}_k^-)W_c(X_k^- - \hat{x}_k^-)^T + Q \tag{6}$$

where X_k^- denote the *a priori* sigma points, $\hat{x}_k^-$ denotes the *a priori* position estimate, P_k^- denotes the *a priori* position covariance, and W_m, W_c and Q are algorithm parameters. For the terrain-based algorithm to function, initial estimates of position $\hat{x}_0^-$ and covariance P_0^- must be made available, typically from the last known GPS location before the GPS signal was lost.

Next, the measurement update (or *correction*) step processes the newly obtained measurement in order to provide a correction for the *a priori* position estimate and covariance according to (7), (8), (9), (10) and (11) as follows:

$$Y_k = f_{NL}(X_k^-) \tag{7}$$

$$\hat{y}_k = Y_k \cdot W_m \tag{8}$$

$$P_{yy,k} = (Y_k - \hat{y}_k)W_c(Y_k - \hat{y}_k)^T + R \tag{9}$$

$$P_{xy,k} = (X_k^- - \hat{x}_k^-)W_c(Y_k - \hat{y}_k)^T \tag{10}$$

$$K = P_{xy,k} \cdot P_{yy,k}^{-1} \tag{11}$$

Here, (7) represents the terrain map lookup step, where the *a priori* sigma points are used to recover the corresponding attitude measurements at those locations, $\hat{y}_k$ denotes the effective attitude measurement at the *a priori* position estimate $\hat{x}_k^-$, and K represents the filter gain. Finally, the *a posteriori* position and covariance estimate are given by (1) and (2) as follows:

$$\hat{x}_k = \hat{x}_k^- + K \cdot (\theta_k - \hat{y}_k) \tag{1}$$

$$P_k = P_k^- - K \cdot P_{yy.k} \cdot K^T \tag{2}$$

where θ_k denotes the attitude measurement obtained at time k. The discrete-time recursive filtering scheme represented by (3) – (2) is used to track a vehicle on a given road segment using attitude measurements, odometry and the correct terrain map. Section 4 discusses how the algorithm performance varies with sensor characteristics, and how it fares when a commercial-grade sensor is used. Section 5 details a real-time implementation and performance evaluation of the algorithm. Section 6 provides a proof-of-concept description of how the algorithm may be extended to encompass large road networks.

3.2 Algorithm Performance with Commercial-grade Sensors

One of the first tasks under Penn State's purview was to analyze the tracking performance of the existing GPS-free terrain-based localization algorithm when using commercial-grade sensors. In concomitance, Task (1) required Penn State to estimate the minimum localization accuracy achievable with a known commercial-grade sensor with given specifications. Alternatively, the problem required

Penn State to determine the minimum sensor specifications required to achieve a known level of localization accuracy.

The following subsections discuss a simulation-based analysis of the performance of the terrain-based localization algorithm using various commercial-grade inertial sensors with different sensor characteristics. The motivation behind performing a simulation-based analysis, rather than carrying out experiments with actual sensors, was that such an analysis offered a deeper insight into how different sensors and sensor characteristics affect the tracking performance of the terrain-based localization algorithm. Specifically, the following subsections not only provide a means of analyzing algorithm performance, they also provide potential future guidelines for selection and design of inertial sensors within given cost constraints and localization performance criteria.

3.2.1 The Need for Sensor Characterization

In order to simulate sensor output, it is necessary to know the characteristics of the inertial sensors employed for measuring the road disturbances such as road pitch, roll and yaw. The sensor characteristics or specifications are indicative of the various noise components, such as bias instability, angle random walk, rate ramp, quantization error, etc., that corrupt the true measurement data. Numerical values for these contributory noise terms allows us to develop simulations that produce sensor-like data streams that are representative of data obtained from actual sensors.

These simulations enable us to estimate how individual error components contribute towards measurement noise and, consequently, the localization accuracy. The same simulation-based approach also enables us to work backwards from the desired localization accuracy to determine the required sensor accuracy and sensor characteristics. Thus, sensor modeling and characterization form the first few steps towards analyzing the effect of commercial-grade sensors on the tracking performance of the terrain-based localization algorithm.

3.2.2 Sensor Modeling, Characterization and Simulation

This subsection[1] discusses the various noise sources in inertial sensor measurements. It also details the procedure used by Penn State for identifying inertial sensor characteristics through Allan variance analysis. Further, it details the models developed by Penn State for simulating a corrupted signal emanating from an inertial sensor with known characteristics.

3.2.2.1 Sensor Noise Modeling

The measurements obtained from a sensor are known to be corrupted by a variety of noise sources inherent to the sensor. For example, random flickering in the sensor's electronic components can cause

---

[1] This subsection contains excerpts from a recent publication (Jerath & Brennan, 2011) by the authors.

bias drift or bias instability in inertial sensors (IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros, 1998). The measurement error caused by these noise sources can be approximated by developing noise models for each source. *Noise modeling* is the process of specifying a functional form and a set of parameter values that represent a noise source. For example, bias instability is modeled as a first-order Gauss-Markov process (El-Sheimy, Hou, & Niu, 2008).

The primary noise sources that contribute to measurement error in inertial sensors are angle random walk (η) and bias instability (b) (Bevly, 2004). The noisy sensor measurements are calculated by adding the errors due to various noise sources to the true value, as shown in Eq. (14) :

$$\omega = \omega_{TRUE} + \eta + b \tag{14}$$

Angle random walk is modeled as a white noise applied to the angular rate measured by the gyroscope. Integration of the corrupted angular rate yields a random walk error in the angle (attitude) measurements, giving the noise source its name. The parameter used to specify angle random walk is the angle random walk coefficient (*N*) which is the square root of the noise power (IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros, 1998),

$$\mathrm{E}[\eta^2] = \mathrm{N}^2 \tag{15}$$

Bias instability is the result of the random flickering in electronic components and is modeled as a first-order Gauss-Markov process representing exponentially correlated noise, as shown in Eq. (16):

$$\dot{b} = -\beta b + \eta_b \tag{16}$$

where, $\beta = 1/T_c$, is the inverse of the correlation time, and η_b is white noise with $E\left[\eta_b^2\right] = \sigma_B^2 \approx (\beta^2 + \omega_0^2)\, B^2/\omega_0$, where B is the bias instability coefficient used to specify the noise, and ω_0 corresponds to the flicker noise cutoff frequency (IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros, 1998) (Dean, Langelaan, & Brennan, 2009). Thus, as mentioned before, given the noise parameters, measurements from a noisy sensor can be simulated using the discussed noise model. The noise parameters for inertial sensors are determined using Allan variance analysis, a procedure discussed in the following subsection.

3.2.2.2 Sensor Characterization

Sensor characterization is the process of identifying and quantifying the model parameter values of the noise sources that contribute to sensor measurement error, using actual measurements collected from a sensor. Penn State performed sensor characterization using Allan variance and autocorrelation analyses, as both are relatively simple and repeatable methods of error source identification (Godha & Cannon, 2005).

The basic premise of Allan variance analysis is that different noise sources contribute to sensor measurement error in different power spectra. Thus, their individual contributions can be quantified by observing the Allan variance in the corresponding time domain. Allan variance in the context of inertial sensors is defined as follows (Dean, Langelaan, & Brennan, 2009) (Gelb, Kasper, Nash, Price, & Sutherlan, 1974):

$$\sigma^2(\tau) = \frac{1}{2}\langle(\overline{\Omega}_{k+m} - \overline{\Omega}_k)^2\rangle \tag{17}$$

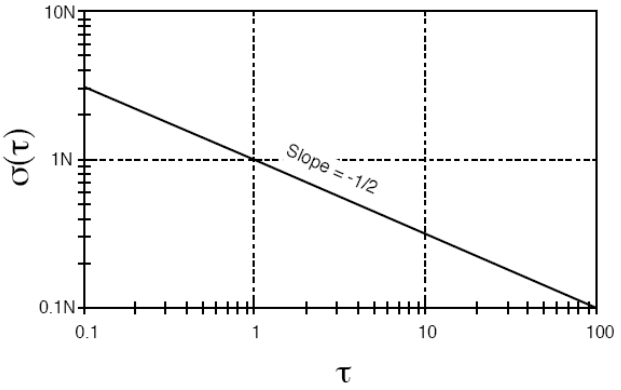
where $\overline{\Omega}_{k+m}$ represents the gyroscope mean angular rate in the k^{th} interval containing m measurements, τ denotes the correlation time and $\langle\cdot\rangle$ represents the ensemble average. The typical plots describing Allan variance for angle random walk and bias instability noise sources, as a function of correlation time, are included as Figure 20 and Figure 21, respectively. The parameters defining these noise models are obtained from the following expressions which bound their values, as detailed in (Ng & Pines, 1996):

$$N = \frac{\sigma_A\,(\tau_N)\sqrt{\tau_N}}{60} \tag{18}$$

$$B = 0.6648\,\sigma_A(\tau_B) \tag{19}$$

where τ_N corresponds to the correlation time when analyzing the Allan variance graph for angle random walk, $\sigma_A(\tau_N)$ corresponds to the Allan deviation when the correlation time is τ_N, $\sigma_A(\tau_B)$ corresponds to the Allan deviation when the correlation time is τ_B, N is referred to as the angle random walk coefficient and B is referred to as the bias instability coefficient. The other parameter in the bias instability model, the correlation time T_C, is typically determined using experimental *autocorrelation analysis*, as discussed in (Allan, 1966).

The figures below show the Allan deviation plots for the angle random walk (left) and the bias instability (right). The angle random walk graph shows a straight line with -.5 slope going from 2 N to 100 τ. The bias instability graph shows a linear increase from .02B with a slope of 1 to 8B from .01τ to 1τ. From 1τ to 10τ the slope is 0.



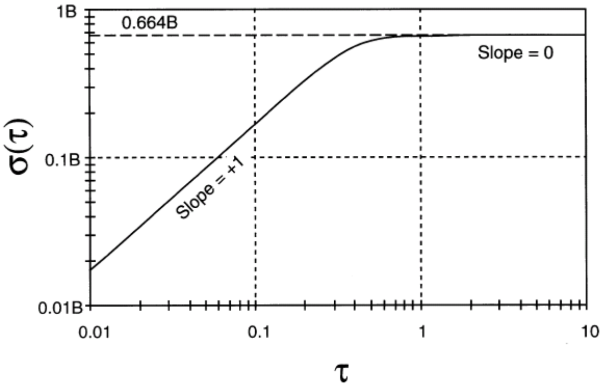


Figure 20: Allan Deviation Plot for angle random walk **Figure 21: Allan Deviation Plot for Bias Instability**

(IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros, 1998)

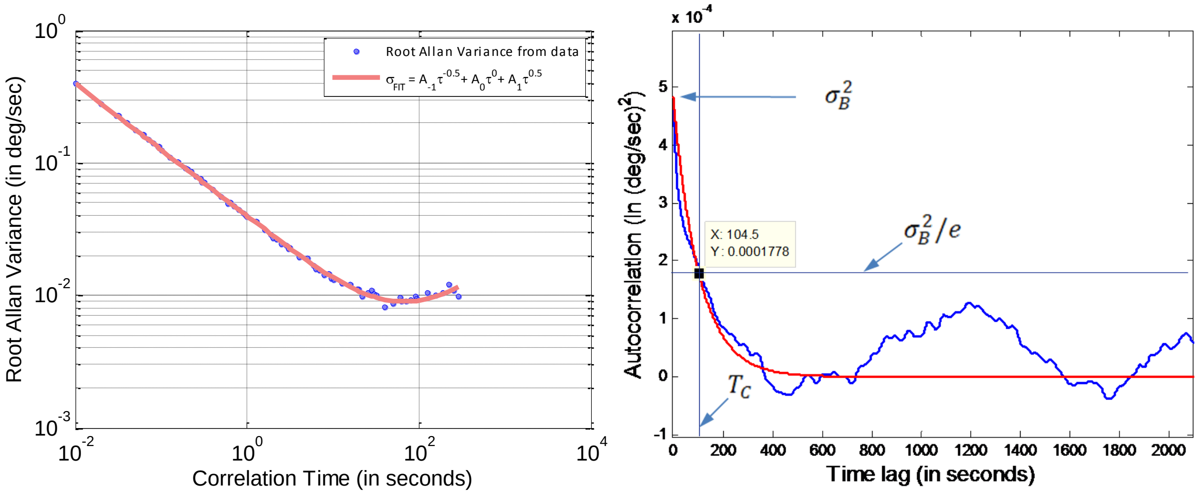
3.2.2.3 Sensor Output Simulation and Validation

Sensor simulation is the process of using the known noise models to corrupt true values of the sensed variable in order to simulate 'noisy' sensor measurements.

As part of Task (1), Penn State developed Simulink noise models for user-defined sensor specifications. Gyro noise components such as angle random walk (ARW) and angular rate random walk (RR) were modeled as white noise with known variance. Noise components such as bias instability were approximated as first-order Gauss-Markov processes (Gelb, Kasper, Nash, Price, & Sutherlan, 1974) (Han, 2009). These models were used to simulate sensor data which was then used in the localization algorithm to determine the achievable localization accuracy. Penn State also developed MATLAB scripts to characterize existing commercial-grade sensors using methods found in the literature (Han, 2009) (Xing, 2008).

A synthesis-analysis approach was used to validate the noise models and sensor characterization scripts. Specifically, Simulink noise models were used to simulate noisy data with known parameters, and the MATLAB scripts were used to recover the input sensor specifications from the noisy data. Noise sources such as angle random walk and bias instability were included in the Simulink model. Sensor data was synthesized using both angle random walk and bias instability noise models. For this example simulation, the ARW noise specification was taken as $N = 0.4\, deg/\sqrt{sec}$, and the bias instability noise specifications were taken as $B = 0.02\, deg/sec$ and correlation time, $T_C = 150\, sec$. Angle random walk noise typically has a higher variance as compared to bias instability noise and this is reflected in the choice of sensor noise characteristics. Figure 22(a) depicts the Allan variance analysis plot for the simulated data, with angle random walk, bias instability and rate random walk included in the model. While Allan variance provides a reasonable estimate of bias instability, a better estimate can be found through signal autocorrelation analysis, as described in (Xing, 2008). Autocorrelation techniques are used to establish an upper bound for the bias instability noise characteristics. Figure 22(b) indicates the autocorrelation analysis performed on filtered simulated data for recovering the bias instability noise characteristics.

The figure below shows two graphs. The graph on the right shows the Autocorrelation in $(deg/sec)^2$. The autocorrelation converges to 0 in about 400 seconds. The graph on the left shows the Root Allan variance analysis. In this graph, the Root Allan variance decreases linearly on a log-log scale to about .01 deg/sec at 100 second correlation time and then arcs upward to create a hockey stick graph.



(a) Sensor specification recovered from Allan variance

(b)Sensor specification recovered from autocorrelation analysis: $\hat{B} = 0.022\ deg/sec, \hat{T}_C = 104.5\ s$

Figure 22: Allan variance and autocorrelation analysis for noise simulated using following specifications: $\sigma_{ARW} = 0.4\ deg/sec$, $\sigma_B = 0.02\ deg/sec$, and $T_C = 150\ s$

3.2.2.4 Characterization of the Crossbow IMU 440

Penn State performed sensor characterization for the Crossbow IMU 440 using data collected at Auburn (using the protocol listed in Appendix 2) and sent to the Penn State team. The sensor specifications obtained from the analysis are included in Table 1. Figure 23 depicts the Allan variance analysis plot for the y-axis gyro of the Crossbow IMU 440. These characteristics were used to corrupt the sensor output from a defense-grade inertial sensor installed on Penn State's test vehicle and subsequently test the algorithm's performance with a simulated commercial-grade sensor.

Table 2: Sensor specifications obtained from Allan variance analysis of Crossbow IMU 440

Y-axis gyro noise specification	Value
N	0.038 deg/√sec
B	0.0057 deg/sec
T_C	357 seconds

Figure 23 shows the Root Allan Variance. In this graph, the Root Allan Variance begins at .1 deg/sec and decreases in a 2nd order parabolic fit on a log –log scale to a vertex at almost .001 deg/sec at 100 second correlation time.

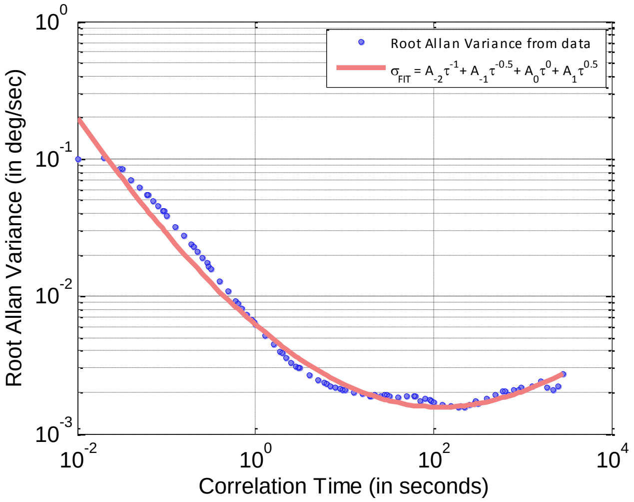


Figure 23: Allan variance analysis plot for data collected from Crossbow IMU 440

3.2.3 Algorithm performance with simulated sensors

In previous subsections, Penn State's ability to generate simulated data for any desired sensor specifications was demonstrated. Penn State also developed the capability to recover sensor characteristics from data collected using actual sensors. With these capabilities in place, Penn State proceeded to analyze the vehicle tracking accuracy achieved with the GPS-free terrain-based localization accuracy, given data corrupted by sensors with different specifications. The noise models were used to corrupt the near-truth pitch reported by a tactical-grade sensor. Figure 24 shows the simulated sensor output for a tactical-grade sensor, whereas compared to true attitude data (N = 0.001 °/√s, B = 0.0001°/s) shows the simulated sensor output for a low-cost sensor.

The figures below show the simulated sensor output for the tactical-grade inertial sensor (left) and the low-cost inertial sensor (right). The tactical-grade inertial sensor shows the simulated pitch and true pitch. Both pitches follow closely and stay close to 0 until 40 seconds, arcs up to about 1 degree at 60 seconds, and falls back to 0 around 80 seconds. The low-cost inertial sensor output shows the simulated pitch and true pitch differing significantly at around 20 seconds. The simulated pitch arcs up to about 2 degrees and falls back to 90 seconds. The true pitch stays around 0 until about 40 seconds and arcs up to about 1 degree at 60 seconds. The pitch falls down to 80 seconds to 0 degrees.

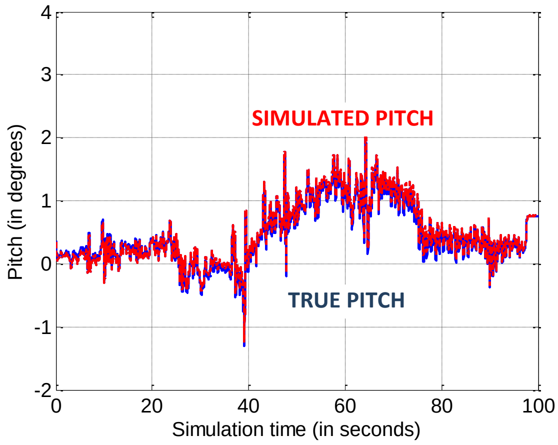


Figure 24: Simulated tactical-grade inertial sensor output compared to true attitude data (N = 0.001 °/√s, B = 0.0001°/s)

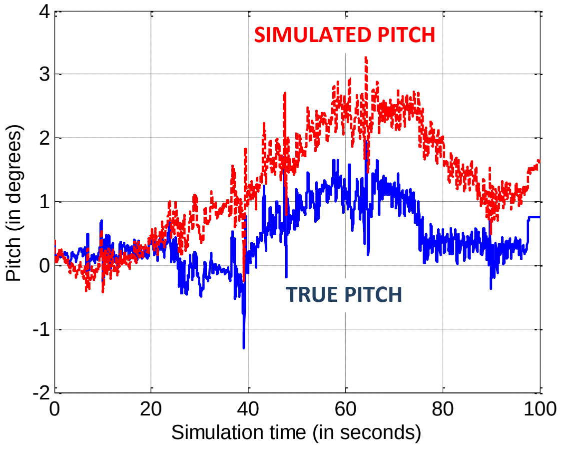


Figure 25: Simulated low-cost inertial sensor output compared to true attitude data (N = 0.01 °/√s, B = 0.01°/s)

As a preliminary example to indicate the qualitative differences in vehicle tracking accuracy, simulations were performed using sensors at the extreme ends of the sensor specification scale. Figure 26 includes the tracking error plots for the two simulated sensors with mentioned sensor characteristics. The simulated low-cost MEMS sensor had the following characteristics: $N = 0.01°/\sqrt{sec}, B = 0.01°/sec$. On the other hand, the simulated tactical-grade sensor had the following characteristics: $N = 0.001°/\sqrt{sec}, B = 0.0001°/sec$. It was observed that even a simulated low-cost MEMS sensor was able to maintain vehicle tracking albeit at slightly larger tracking error values. This observation demonstrated that the terrain-based localization algorithm has the potential to track vehicles even with low-cost sensors with inferior specifications.

The figure below shows the tactical-grade sensor and low-cost mems sensor offline tracking error. The tactical-grade sensor starts at 0 and has a peak of about 6 meters. The low-cost MEMS sensor has a peak of more than 15 meters and is greater than the tactical-grade sensor peak of 6 meters from 40 seconds to almost 80 seconds.

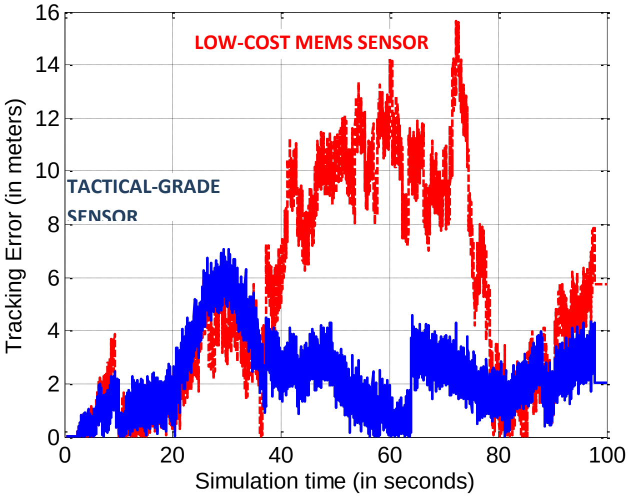


Figure 26: Offline tracking error with different simulated sensors

Next, the effects of varying angle random walk coefficients and bias instability coefficients on vehicle tracking error were analyzed. The values of the noise coefficients were chosen so as to span the range between the low-cost MEMS sensor and the tactical-grade sensor specifications. Figure 27 depicts the plot of vehicle tracking accuracy using the terrain-based tracking algorithm when the angle random walk coefficient was varied from $0.001°/\sqrt{sec}$ to $0.01°/\sqrt{sec}$. Each data point on the thin lines indicates a set of simulations run with a fixed noise model. The thin lines in Figure 27 indicate the mean tracking error as the angle random walk coefficient is varied and the bias instability coefficient is held fixed at various values viz. B = 0.0001, 0.002, 0.004, 0.006, 0.008, and $0.01°/s$. The plot indicates that the mean tracking error increased as the angle random walk was increased from the minimum to the maximum value under consideration. The thick line in Figure 27 is the average of the mean tracking error across all bias instability coefficients. It has been included as a means to emphasize the following qualitative relationship: the mean tracking error increases as angle random walk coefficient increases. Within the range under consideration, this relationship may loosely be described as being quadratic in nature.

Figure 28 indicates that the variance in the tracking error changes significantly as the angle random walk coefficient is varied from its minimum to maximum value. The variance is representative of the precision with which the vehicle was tracked. Specifically, the variance of tracking error increased by about 150% as the angle random walk coefficient increased from 0.001 to $0.01\ °/\sqrt{s}$. Further, the variation in vehicle tracking precision with angle random walk coefficient appeared to be roughly linear in nature, within the range of values under consideration.

The figure below shows the mean (left) and variance (right) of tracking error for varying angle random walk coefficients. The mean starts at about 1.75m and stays fairly flat until .006deg/r-sec, at which case it increases to .01 deg/rt-sec to 3.4m. The variance starts at 1.2m^2 and stays flat until .006 deg/rt-sec and gradually increases to 2.5m^2 at .01 deg/rt-sec.

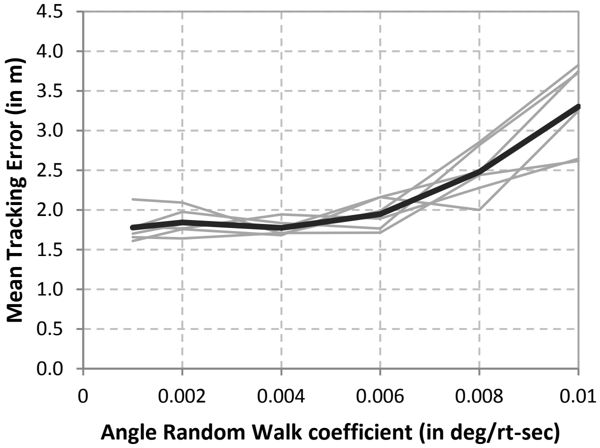


Figure 27: Mean Tracking Error for Varying Angle Random Walk Coefficients

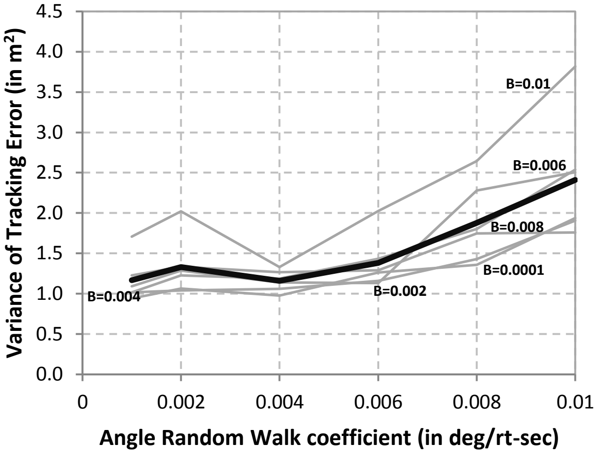


Figure 28: Variance of Tracking Error with Varying Angle Random Walk Coefficients.

A similar analysis as performed for angle random walk noise was performed for bias instability. In this case, the relationship between vehicle tracking precision and bias instability coefficients was less apparent. The bias instability coefficient was varied from $0.0001°/sec$ to $0.01°/sec$. Figure 29 includes the mean tracking error values as bias instability increased, for various fixed values of angle random walk coefficients. As before, the thick line indicates the mean tracking error averaged over the various fixed values of the angle random walk coefficients. The plot indicates that the mean tracking error increased moderately, i.e. the vehicle tracking accuracy decreased, as the bias instability noise component in the inertial sensor increased. Further, this relationship is approximately linear in nature within the range under consideration. The plot also indicates that the mean accuracy of tactical-grade sensors is slightly better than other commercial-grade or MEMS sensors, when considering vehicle tracking applications.

The figures below show the mean (left) and variance (right) of tracking error with varying bias instability coefficients. The mean tracking error starts at 2m and has a slight increase to 2.5m at .01 deg/sec. For the variance, the graph starts at 1.4 m^2 and has a slight increase to 2m^2 at .01 deg/sec.

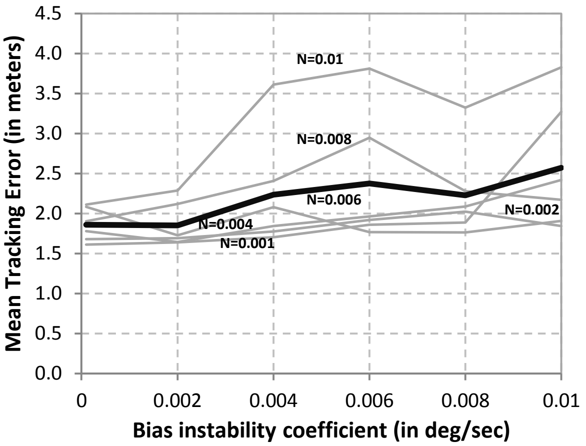


Figure 29: Mean Tracking Error for Varying Bias Instability Coefficients

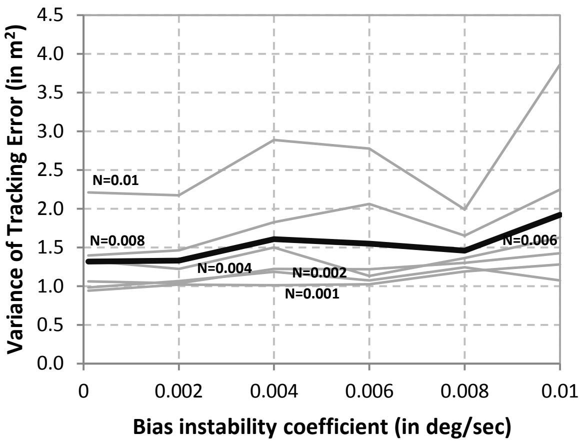


Figure 30: Variance of Tracking Error with Varying Bias Instability Coefficients

Figure 30 depicts the precision in tracking error, and indicates that the variance in tracking error increases moderately as bias instability in the sensor increases. This implies that sensors with higher bias instability coefficients have a poorer precision. Further, the simulations indicate that tactical-grade sensors, which have a bias instability of the order of $0.0001°/sec$, yield a marginally higher precision in vehicle tracking as compared to simulated low-cost MEMS inertial sensors, which have a bias instability of the order of $0.01°/sec$. In other words, the vehicle tracking performance using the GPS-free terrain-based tracking algorithm is not very sensitive to bias instability.

3.2.4 Cost vs Performance Analysis for Simulated Sensors

The next pertinent issue was to examine if the apparent advantages offered by increased precision of tactical-grade sensors, as observed from the simulations for both angle random walk and bias instability coefficients, justify the cost of these sensors. It was our viewpoint that, in a commercial setting, it would advantageous to know the cost of obtaining a desired level of accuracy and precision in vehicle tracking, even if it is not one of the primary design constraints. A few inertial sensors which were representative of their 'cost' category were selected for simulation. These sensors along with their characteristics are listed in Table 3.

Table 3: Selected Available Sensors

Sensor	Angle random walk coefficient, N $(°/\sqrt{sec})$	Bias instability coefficient, B $(°/sec)$	Approximate cost (in 2010 US dollars)
Analog Devices ADIS16367 (Analog Devices ADIS16367 Datasheet, 2011)	0.033	0.013	700

Gladiator Technologies Landmark 10 (Landmark 10 Datasheet, 2011)	0.014	0.007	3,000
Gladiator Technologies Landmark 30 (Landmark 30 Datasheet, 2011)	0.01	0.003	6,000
Honeywell HG1700 (HG 1700 Datasheet, 2011)	0.0016	0.0003	20,000

Table 3 includes the approximate price of the selected sensor plotted against the mean tracking error delivered by a simulated sensor with the same characteristics. The thick line indicates the mean tracking error achieved during vehicle tracking with the specific simulated sensor. The thin lines represent one standard deviation from the mean value, as obtained from multiple simulations. In the sensor selection process, the plot may be used as an indicator of the cost of achieving a pre-specified accuracy. Alternatively, in the sensor design process, it may be used as an indicator of the possible accuracy achieved when constrained by a pre-specified cost. From Table 3 it may be observed that the sensors may be broadly grouped into two ‘cost’ categories, the low-cost category and the higher-end category. Specifically, the plot indicates that the mean tracking error for low-cost sensors, i.e. sensors which cost less than USD 2000, is over 1 m. On the other hand, the mean tracking error for inertial sensors in the higher-end category, which cost USD 2000 or above, is less than 1 m. Further, for sensors in the higher-end category, the mean tracking error is relatively constant and appears to be independent of cost.

More importantly though, Figure 32 includes the approximate cost of the listed sensor plotted against the variance of tracking error obtained by a simulated sensor with the same characteristics. This plot is indicative of the vehicle tracking precision delivered by the inertial sensor as a function of cost. It is observed that significant improvement is obtained as cost increases from the low-cost category to the higher end category. This is evident by observing the variance of tracking error of the ADIS16367 and Landmark 10 inertial sensors.

The figures below shows the mean (left) and variance (right) of tracking error vs. price for the selected inertial sensors. Both sensors show a signficant drop after the $5000 Landmark 30, and a smoother decrease to the HG1700.

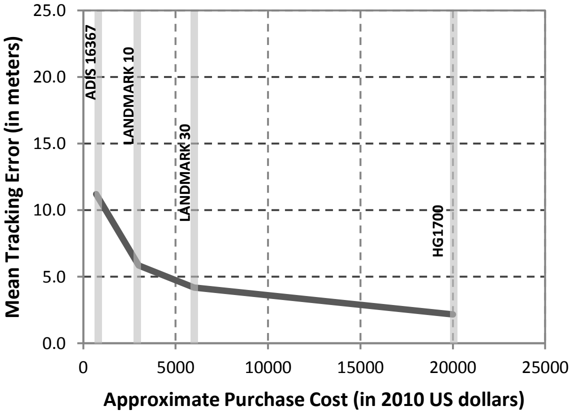


Figure 31: Mean Tracking Error vs. Price for Selected Inertial Sensors

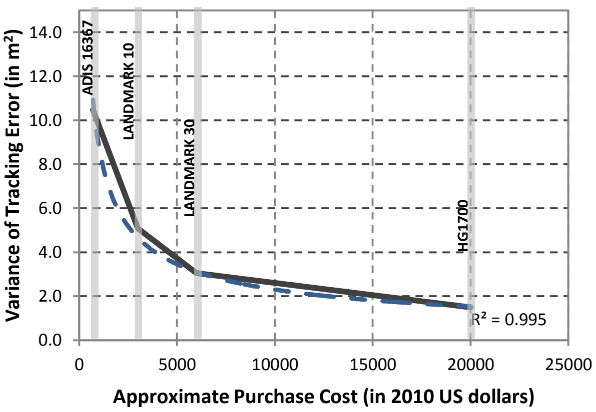


Figure 32: Variance of Tracking Error vs. Price of Selected Inertial Sensors

However, the law of diminishing returns limits the improvement in vehicle tracking precision as cost increases further in the higher-end category. In other words, it requires more and more expensive inertial sensors in order to obtain higher precision. Specifically, as shown in Figure 32, the cost of the sensor can be approximated by a power law, where the variance of tracking error approaches zero asymptotically, as the cost of the sensor approaches infinity. This represents a fundamental limitation in the sensor selection and/or design process. The associated coefficient of correlation for the fitted curve is 0.995, indicating that the power law may indeed be a good functional approximation for the relation between sensor cost and vehicle tracking error precision. In Figure 32, the relationship is described as follows:

$$\sigma^2 = A(Cost^{-B}) \tag{1}$$

where σ^2 is the tracking error variance (in m^2), A = 508.4, B = 0.586, and sensor cost is in 2010 US dollars. It may be observed that the return on investment of inertial sensors for vehicle tracking purposes reduces as cost increases, since reasonable performance is offered by even cheaper sensors in the higher-end category, such as the Landmark 10. However, the cost-to-performance ratio metric is application specific, and for several applications, such as autonomous vehicles requiring high-gain feedback, the higher cost may be justified in order to achieve higher precision.

3.3 Real-time Implementation of Terrain-based Algorithm

Task (2) entailed the development of a real-time implementation of the localization algorithm. The localization algorithm is intended to provide accurate position estimates during periods and/or regions with weak or absent GPS signals. This section discusses the algorithmic and hardware setup, the test results and the effect of terrain and sampling rate on the tracking accuracy of the GPS-free terrain-based vehicle tracking algorithm when implemented in real-time.

3.3.1 System architecture for real-time implementation

In order to facilitate the development of the system, Penn State first conceptualized the flow of information and arrived upon an appropriate system architecture for the process. The system architecture details and modularizes the various functions expected to be performed by the algorithm in real-time. Figure 33 depicts the designed system architecture. The system has been divided into four modular subsystems, allowing separation of functionality, which further facilitates individual development and testing. The four subsystems or layers in the system architecture are as follows:

1) Supervisory layer
2) Algorithmic layer
3) Sensing layer, and
4) Database layer

The system architecture block diagram is shown in Figure 33. The environment is shown as an input to the system on the left. The data is moved to the algorithm supervisor in the IPS layer, which is then used in constructing the estimator and initialization module. Various exception handling and buffers are present throughout the architecture. Information is also then sent into both an on-board and off-site database.

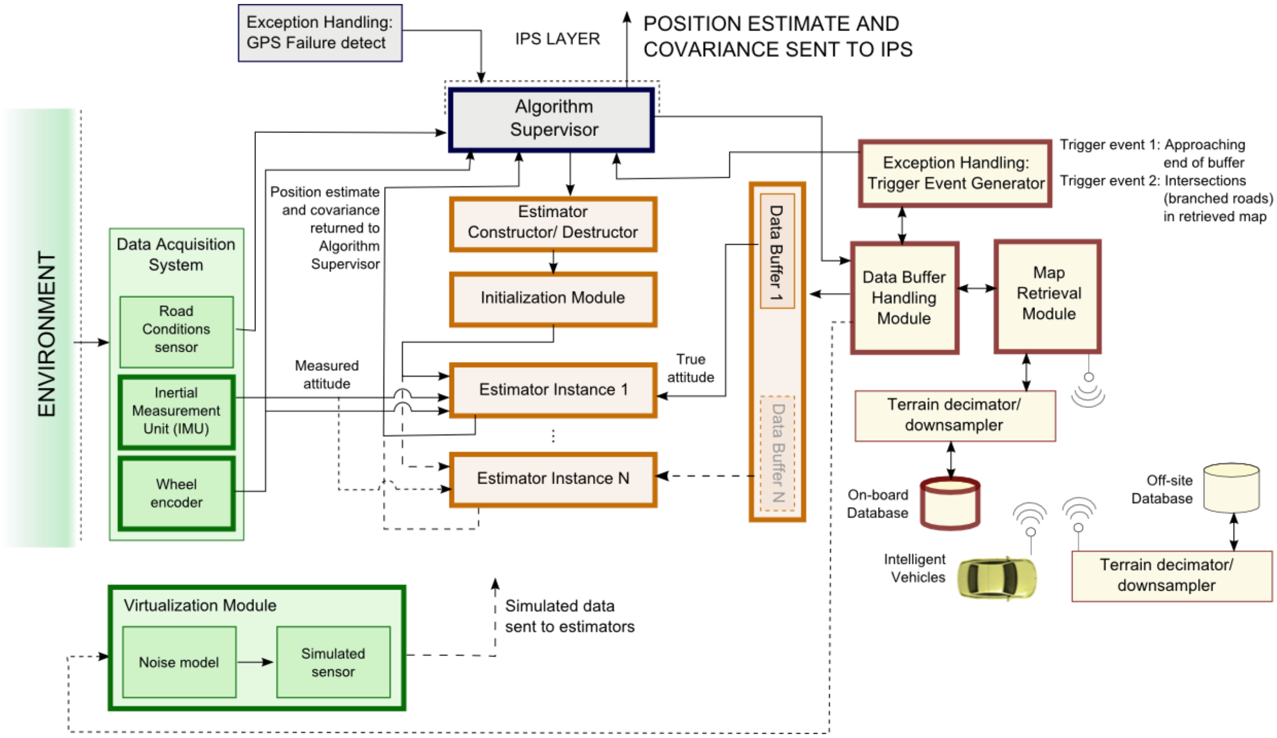


Figure 33: System architecture detailing system setup and data flow for real-time implementation. Boxes with bold outline indicate system elements built and tested by Penn State

A fifth layer that exists external to the system is the IPS (integrated Positioning System) layer that controls when and how the terrain based localization algorithm is activated. The functions and details of each of these layers are explained in detail in the following subsections.

3.3.1.1 Supervisory Layer

The function of the Supervisory layer is to maintain communication with the Integrated Positioning System central supervisor. The major functions of the Supervisory layer include receiving initialization conditions for the estimation algorithm, and transmitting current position estimates and covariance matrices to the IPS layer for data fusion.

Secondary functions of the Supervisory layer include correctly initializing the estimation algorithm and responding to internally generated interrupts such as:

(a) Interrupts related to sensor diagnostics indicating failed sensors. In this scenario, the supervisory layer is expected to discontinue sending position estimates to the IPS.
(b) Interrupts related to arrival of the vehicle at intersections. In this scenario, the Supervisory layer is expected to command the construction new instances of estimators to handle branched roads till the estimation algorithm converges to the correct position estimate. The Supervisory layer is also expected to command the destruction of estimators when it becomes evident that they are not tracking the true vehicle position.

The supervisory layer is shown in Figure 34. Various inputs, such as sensor diagnostics, exception handling, wheel encoder data, position estimates, and trigger events, are shown as inputs. Outputs include the estimator constructor/destructor commands, position estimate and covariance, and position information for the database map retrieval commands.

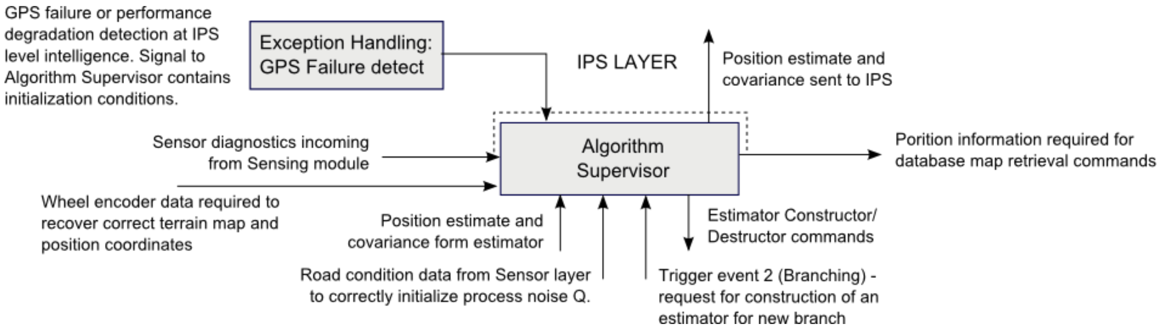


Figure 34: The Supervisory Layer

3.3.1.2 Algorithmic Layer

The primary function of the Algorithmic layer is to perform the actual state estimation using prerecorded and measured attitude data. Both particle filter and unscented Kalman filters have been shown to work for local tracking of vehicles (Dean, Langelaan, & Brennan, 2009). The Algorithmic layer receives measurement information from the Sensing layer, true terrain data from the Database layer, and commands for construction and destruction of estimators from the Supervisory layer. It sends the position estimates and covariance matrices to the Supervisory layer. When multiple estimators exist, the Algorithmic layer also maintains the data buffers for each estimator instance. Figure 35 depicts the functions and data flow in the Algorithmic layer.

Figure 35 shows the Algorithmic Layer. This stage takes commands from the Algorithm Supervisor for constructing / destroying estimator instances, initializes modules, and creates one or more estimator instances. The true attitude is used in the data buffer from the database layer for the estimator instances as well as the measured attitude values from the sensor layer.

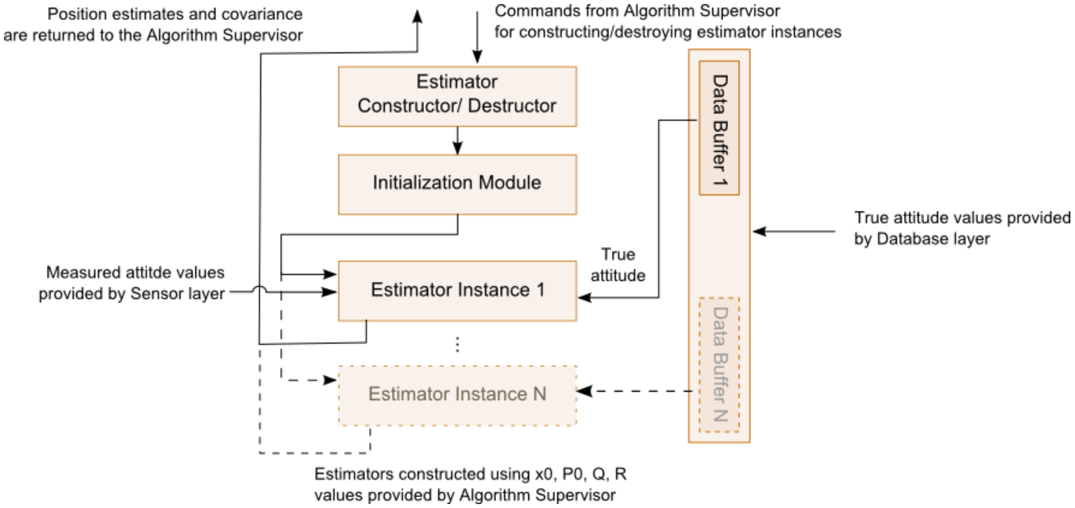


Figure 35: The Algorithmic Layer

3.3.1.3 Sensing Layer

The Sensing layer performs the functions of acquiring data from the environment, sending the data to the Algorithmic layer for processing, and monitoring sensor health. The Sensor layer consists primarily of three types of sensors – the IMU for terrain attitude measurement, wheel encoders for odometry, and in the future may contain some form of road or weather monitoring sensors in order to correctly initialize the process noise covariance, Q. The Sensing layer also contains the Virtualization module, which will be utilized to test the estimation algorithm with simulated sensor data. Figure 36 depicts the Sensing layer.

Figure 36 shows the Sensing Layer. Information from the environment is obtained through wheel encoders, IMU's, road conditions sensors in order to determine if the sensors degrade beyond acceptable limits.

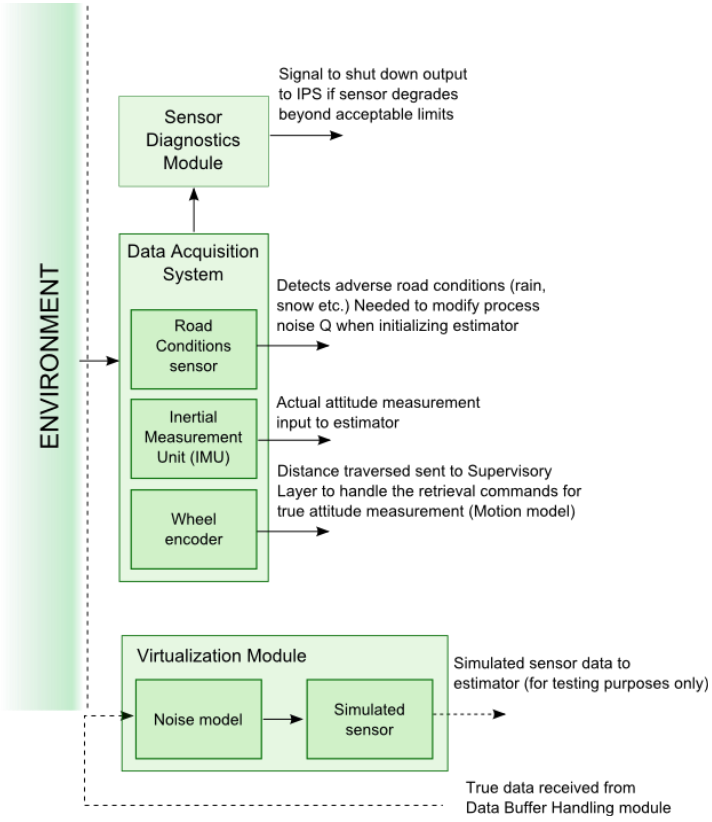


Figure 36: The Sensing Layer

3.3.1.4 Database Layer

The primary function of the Database layer is to store, retrieve and output terrain map data for the Algorithmic layer. The layer retrieves maps that may be stored on-board or at an off-site database, or may even be transferred from vehicle to vehicle. The Database layer also handles exception events such as arrival of the vehicle at an intersection, or retrieval of data for successive road segments. The layer uses the commands received from the Supervisory layer to retrieve data for the next road segment. Figure 37 depicts the Database layer.

The Database Layer takes information from the Supervisory Layer to determine if a trigger event is needed in the case of an intersection or the end of the buffer or to send data to the Algorithmic layer. Information form an on-board database, off-site database, or another vehicle helps reduce overhead by transmitting maps at a resolution appropriate for on-board sensing equipment.

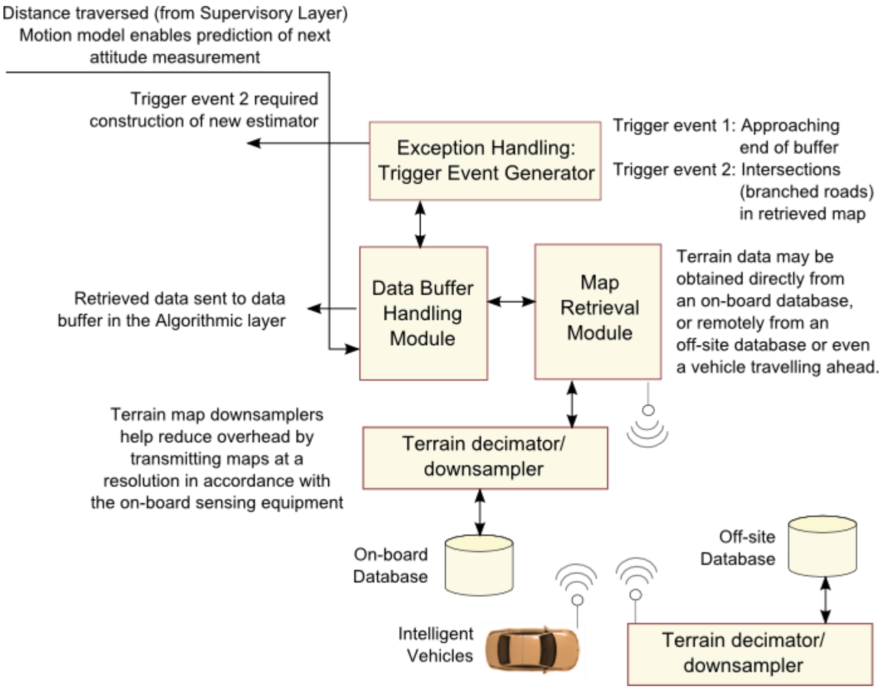


Figure 37: The Database Layer

3.3.1.5 Database Management

An important part of the real-time implementation is database management. The SPKF relies on a pre-recorded map of the terrain (containing attitude and location information) to predict the attitude at the current position estimate for use in the algorithm. In order to retrieve the required information, the estimator subsystem in the Simulink model queries the terrain database. The terrain database will typically contain information for the entire road (or road network). A segment of this information is retrieved and stored in a buffer to be used later by the algorithm. Database management refers to the handling of the buffer in an optimal manner.

The buffer (or terrain buffer) was operated as a **dynamic buffer**, wherein a small road segment is retrieved from the database based on the current location of the vehicle. As the current position estimate is updated, the terrain database is queried to retrieve information based on the updated position. Penn State incorporated the dynamic buffer configuration in the Simulink model. The bounds of the information included in the terrain buffer under the dynamic buffer configuration are shown in Figure 38.

Figure 38 shows a graph of the dynamic buffer configuration. The sigma points lines start a 0 meters and increases to 50 meters over 9 seconds. The Lower bound starts at 0. At 3 seconds the Lower bound increased from 0 meters to 45 meters over 6 seconds. The Upper bound starts at 30 meters and mirrors the Lower bound line.

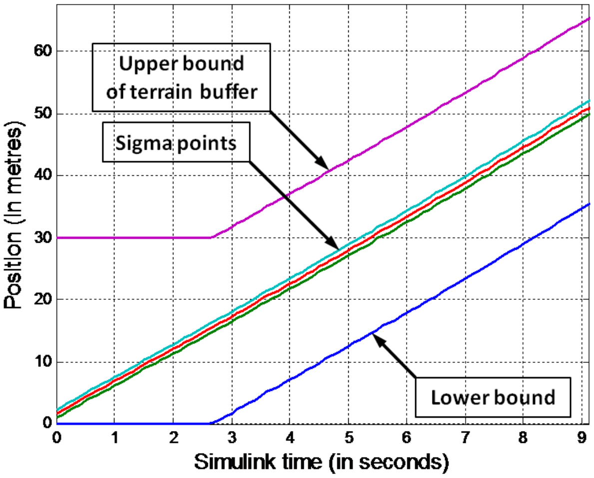


Figure 38: Dynamic buffer configuration maintains a constant window of terrain information centered on the current position estimate

3.3.2 Hardware Setup and Testing

The algorithm was implemented in real-time and tested at the Pennsylvania Transportation Institute's test track. The test vehicle was instrumented to collect odometry data through encoders mounted on the wheels, and vehicle attitude data through the HG1700 tactical grade IMU. Since one of the aims of the project was to demonstrate the ability of the algorithm to work with low-cost commercial grade sensors, the attitude data acquired from the HG1700 IMU was corrupted using a noise model representative of a Crossbow 440 IMU available at Auburn University. The test vehicle and instrumentation setup are shown in Figure 39 and Figure 40.

The lower left figure shows the Penn State test vehicle equipped with a GPS receiver on the roof and an encoder mounted on the wheel. The figure on the lower right shows the instrumentation setup including the HG1700 tactical grade IMU.

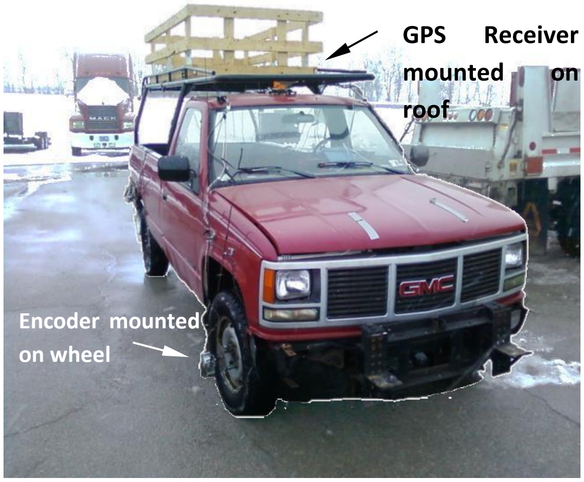


Figure 39: Test vehicle

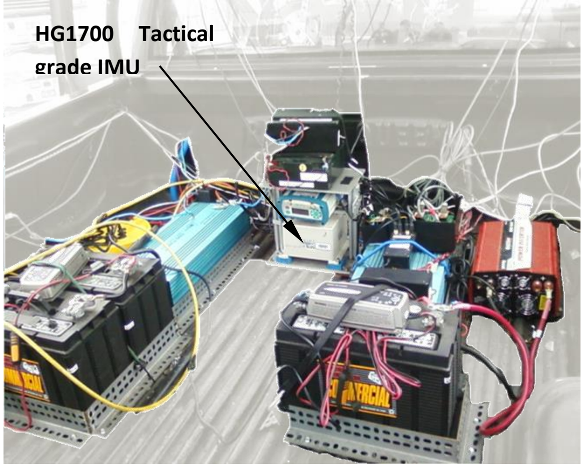


Figure 40: Instrumentation setup

During testing, the attitude measurements from the simulated Crossbow IMU 440 were found to drift from the true attitude as expected from the noise model. The 'true' attitude was defined as the attitude measured using the tactical-grade HG1700 IMU system. However, despite the drift, the algorithm continued to track the vehicle position with satisfactory performance. The results of the real-time tracking are included in Figure 41. In the test run, during which the vehicle traveled approximately 1 km, tracking accuracy hovered in the meter-level range.

Figure 41 shows the real-time tracking error. The error begins at 4 meters, peaks at 6 meters, then falls to 0 meters at 20 seconds into the run. The error immediately increases to 3 meters until it falls back to 0 meters at 40 seconds. The error then remains at 1 meter until the end of the run.

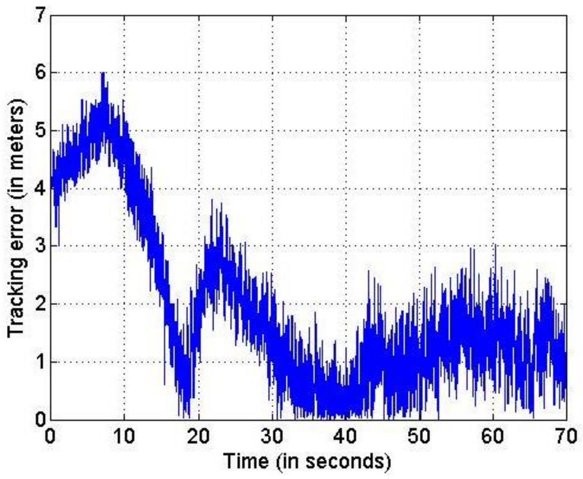


Figure 41: Real-time tracking error in absence of GPS using a simulated 'noisy' Crossbow 440 IMU. Vehicle tracking is maintained at meter level accuracy.

3.3.3 Performance Comparison in GPS-denied and Intermittent GPS Environments

Field experiments to test the algorithm were performed in two scenarios: First, with no GPS data available and second, when GPS data was only intermittently available. Figure 42 and Figure 43 show the vehicle position estimate, in terms of distance traveled along the road, and tracking error when no GPS is available, respectively. Figure 44 and Figure 45 respectively show the vehicle position estimate and tracking error when GPS data is intermittently available. As expected, it was observed that the algorithm performed better when GPS was intermittently available as compared to when GPS was unavailable.

Four plots are shown below. The top left plot shows the true position and estimated position over time based on distance traveled with no GPS. The top right plot shows the vehicle tracking error with no GPS, and the error hovers around the 10m mark with drops to 0m and peaks at 20m. The lower left plot shows the true position and estimated position once more but with intermittent GPS. The lower right plot shows the vehicle tracking error, where the error is reduced to 0 during the conditions where GPS is active.

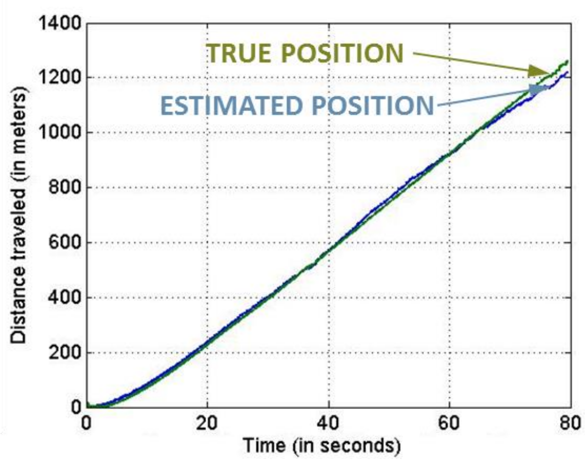


Figure 42: Vehicle tracking – Distance traveled – No GPS

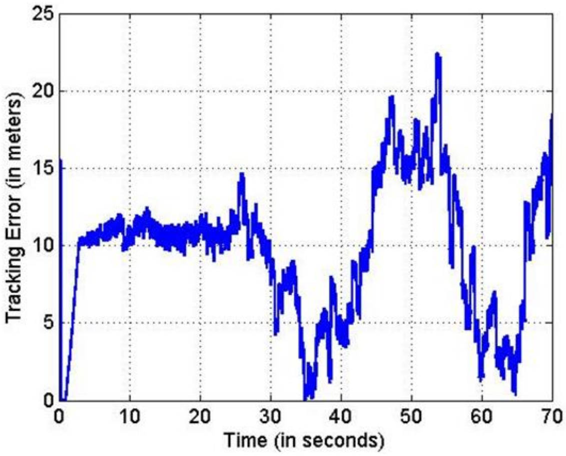


Figure 43: Vehicle tracking error – No GPS

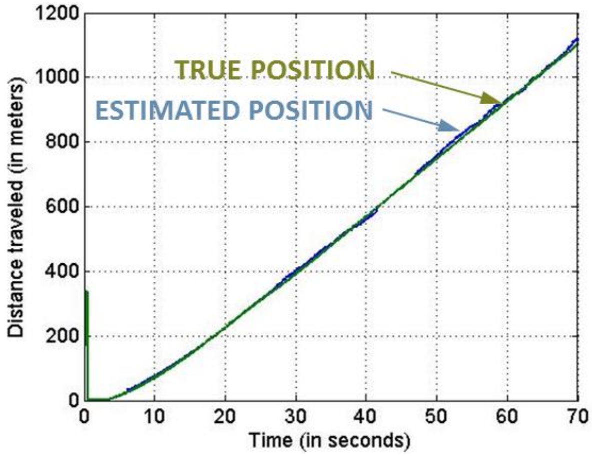


Figure 44: Vehicle tracking – Distance traveled – Intermittent GPS

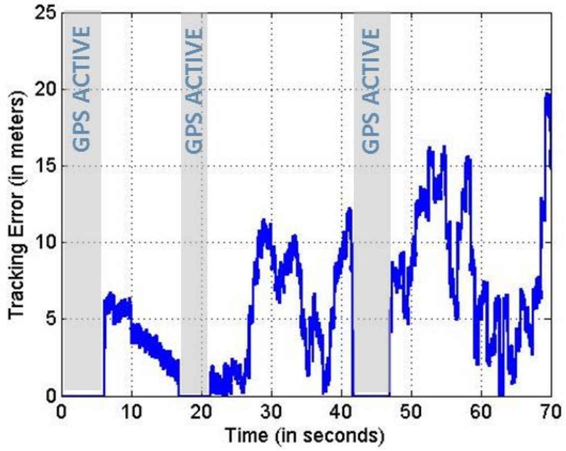


Figure 45: Vehicle tracking error – Intermittent GPS

3.3.4 Effect of Terrain and Algorithm Sampling Rate on Localization Accuracy

While testing the SPKF algorithm, it was observed that the mean accuracy of the SPKF varied significantly based on two factors, viz.

(a) Terrain, i.e. the type of road over which the vehicle was traveling. Specifically, tracking accuracy on a highway was different from that on a secondary or rural road, with all other factors remaining unchanged.
(b) Sampling rate of the SPKF algorithm, i.e. the distance after which the filter was updated using new terrain measurements.

It was found that the SPKF performed much better on secondary roads as compared to highways. This finding conforms to intuition, since it is expected that a road with large variations in terrain provides more features and information, and hence a better tracking ability, as compared to a completely flat road with no variations in terrain. As is evident from Figure 46, the terrain variations are more pronounced on secondary roads as compared to highways.

The graph below shows the pitch in degrees for Rock Road (a secondary road) and Highway 322. The Rock Road pitch changes drastically from about -2 degrees at the start to 3 degrees after around 175m. The pitch for Rock Road alternates about 2 degrees until 400 meters, at which point the pitch falls to -1.5 degrees. Once more, the pitch fluctuates about 2 degrees until 1000m, where the pitch approaches 0 degrees. The Highway 322 pitch, however, begins at -2 degrees, changes to about -1.5 degrees after about 100 meters, and slightly fluctuates until 550 meters, at which time there is a linear increase in pitch to 800 meters to slightly more than 1 degree. The pitch again stays fairly constant until 1000m and is quickly followed by a gradual decrease to 0 degrees to the end of the test run.

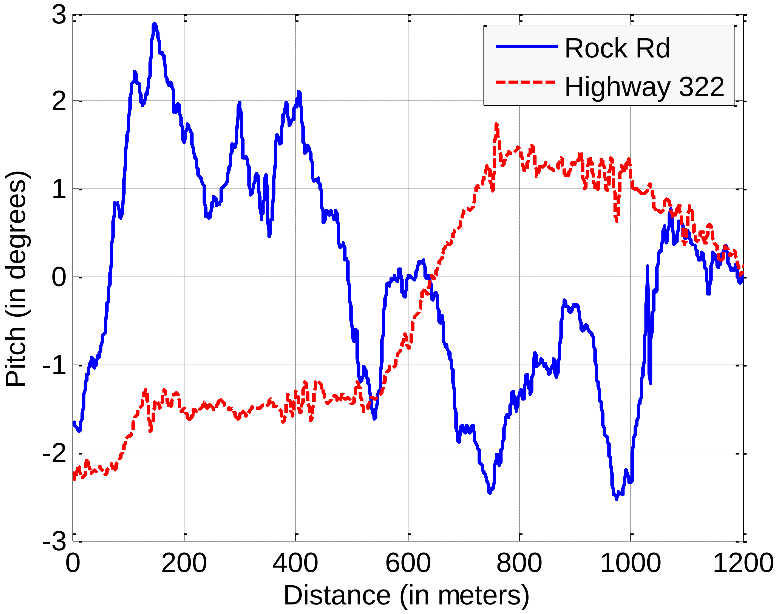


Figure 46: Pitch variations on Rock Rd (secondary road) and Highway 322. The more pronounced pitch variation on secondary roads results in greater accuracy as compared to highways, when all other factors remain unchanged.

It was also observed that the mean tracking error output was smaller for larger sampling intervals, e.g. the distance traveled by a vehicle before the SPKF is updated using new measurements. Figure 47 indicates the reduction in mean error as the sampling interval increases, for both highways and secondary roadways.

The graph in **Error! Reference source not found.** shows the mean longitudinal position error with respect to the distance update threshold for Rock Road and Highway 322, where the mean error is calculated over a randomly selected 1 km long road segment. The error for Rock Road begins at around .65m but falls to 0.3 m where the error fluctuates by about 0.1 meters until the 80m threshold, where the error increases to .4m. The Highway 322 longitudinal position error remains close to 0.6 m and fluctuates about 0.05 meters to the 85m distance update threshold.

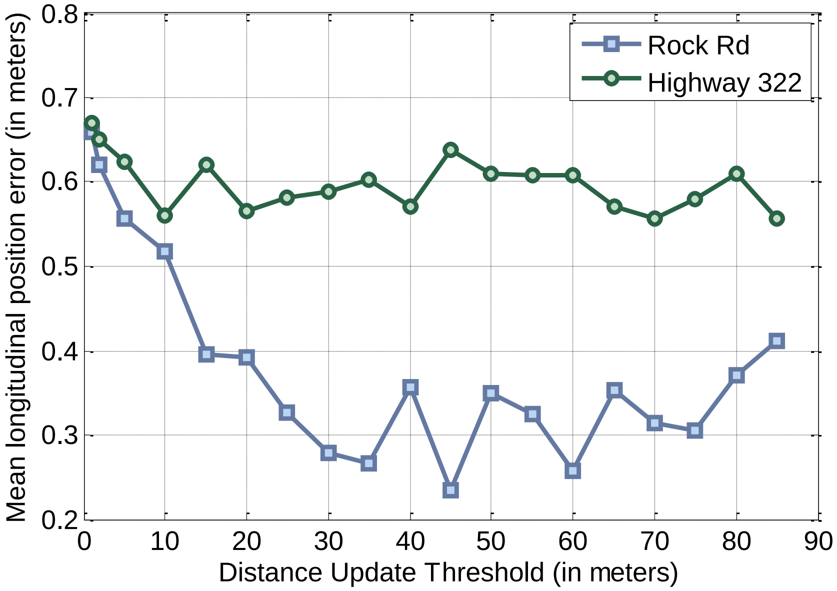


Figure 47: Variation of mean longitudinal position accuracy with varying values of Update Distance Threshold. Mean error calculated over a randomly selected 1 km long road segment using tactical-grade sensor.

The trends observed in Figure 47 are a consequence of the sensor fusion taking place in the SPKF algorithm. Specifically, incoming data from the encoders, which represent the motion model, are being fused with data from the inertial measurement unit, which represents the measurement model. At short distances (i.e. small sampling rates), the encoder dominates since there is no appreciable difference in terrain. As a consequence, IMU measurements have little contribution towards providing a correction update to the position estimate. However, at longer distances (i.e. large sampling rates), the encoder error accumulates, but the terrain also changes by a significant amount which allows larger IMU corrections of the position estimate in the subsequent “correction” or measurement update step. In addition, Figure 47 also indicates the difference in tracking accuracy of the algorithm on different roadways. Specifically, it was observed that the mean longitudinal position error on secondary roads (Rock Rd) was much smaller than the mean error on highways (Highway 322).

3.4 Road Network Implementation of Terrain-based Algorithm

Task (3) of the project aimed at expanding the scope of the terrain-based localization algorithm so that it may be used across an entire road network. As part of this task, Penn State modified the algorithm to suit a scenario where multiple road segments may be part of a vehicle's potential travel path. Specifically, data from a T-junction was used in a simulation where the vehicle could either go straight or take a right turn. This scenario is depicted in Figure 48. The corresponding terrain maps are included in Figure 49.

The figures below show the T junction simulation for the road network implementation. The figure on the left (Figure 48) shows a bird's eye view of the T-junction. The figure on the right (Figure 49) shows the terrain maps with pitches. At the point with the intersection, the terrain maps diverge.

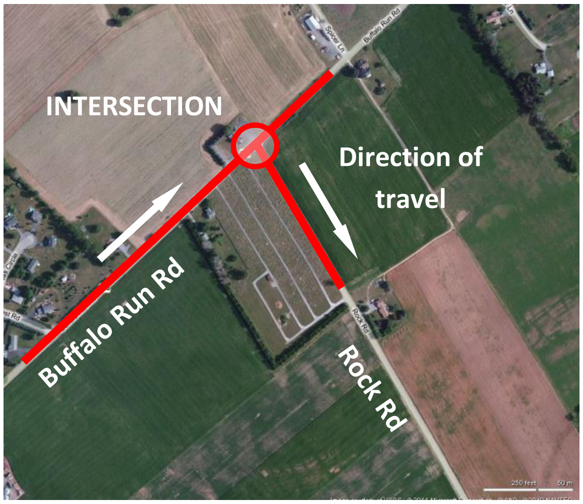


Figure 48: T junction used for simulation. Vehicle can either go straight on Buffalo Run Rd or take a right turn onto Rock Rd.

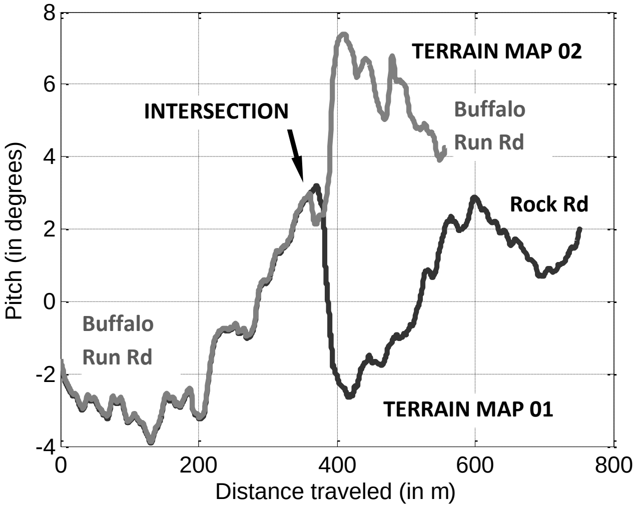


Figure 49: Terrain maps depicting pitch values for the two possible paths that the vehicle can take at this T junction.

3.4.1 Multiple Model Estimation

A multiple model estimation scheme was adopted to handle the multiple travel paths possible in this scenario. Specifically, two estimators were included in the Simulink model to provide position and covariance estimates for each of the two possible travel paths. Next, the observed and predicted pitch values were used to determine the probability of the vehicle being on either road segment according to the following equation (Zhao & Li, 2005):

$$P_k(Road\ i) = \frac{P_{k-1}(Road\ i) \cdot Likelihood(\theta_k|\hat{\theta}_k^i)}{\sum_j P_{k-1}(Road\ j) \cdot Likelihood(\theta_k|\hat{\theta}_k^j)}$$

where $P_k(Road\ i)$ represents the probability that the vehicle is on road segment i at time k, θ_k represents the pitch measurement obtained at time k, and $\hat{\theta}_k^i$ represents the predicted value of pitch at time k if the vehicle was on road segment i. The likelihood is calculated assuming a Gaussian distribution of the pitch values with mean $\hat{\theta}_k^i$. The variance of the Gaussian distribution is determined from the pitch values at the sigma points at time k (obtained from the Unscented Kalman Filter).

Experimental results indicated that the multiple model estimation scheme was able to correctly identify the road segment, and local tracking in the absence of GPS was still achieved. Figure 50 and Figure 51 depict the possible travel paths (corresponding to Terrain Maps 01 and 02). Figure 52 represents the tracking error when Estimator 1 assumed that the Terrain Map 01 was the true representation of the terrain the vehicle was traveling over. Figure 53 represents the tracking error when Estimator 2 assumed that the Terrain Map 02 was the true representation of the current terrain. It must be noted that both estimators ran simultaneously in the simulation.

The figures below show the GPS coordinates in latitude and longitude of the travel path for the intersection with the various terrain maps (Terrain Map 01 (left – Figure 50), and Terrain Map 02 (right – Figure 51)). Figure 52 shows the tracking error for Terrain Map 01, and the error stays below 3 m. Figure 53 shows the tracking error for Terrain Map 02, and the error increases dramatically. These figures show how the vehicle knows which direction the vehicle has traveled.

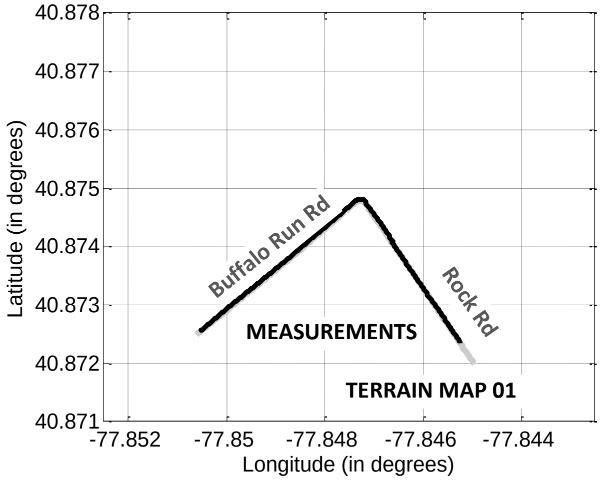


Figure 50: GPS coordinates of Terrain Map 01 (grey). The actual travel path (black) corresponds to Terrain Map 01, i.e. the vehicle took a right turn on to Rock Rd.

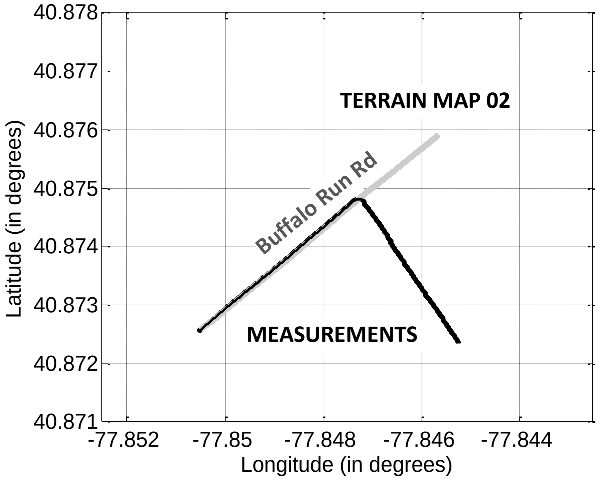


Figure 51: GPS coordinates of Terrain Map 02 (grey). The vehicle (black) did not go straight on Buffalo Run Rd.

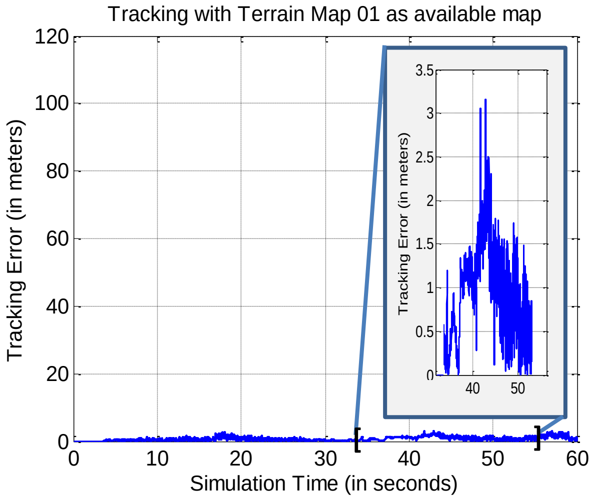


Figure 52: Tracking error with Estimator 1, assuming Terrain Map 01 is the true representation of current terrain

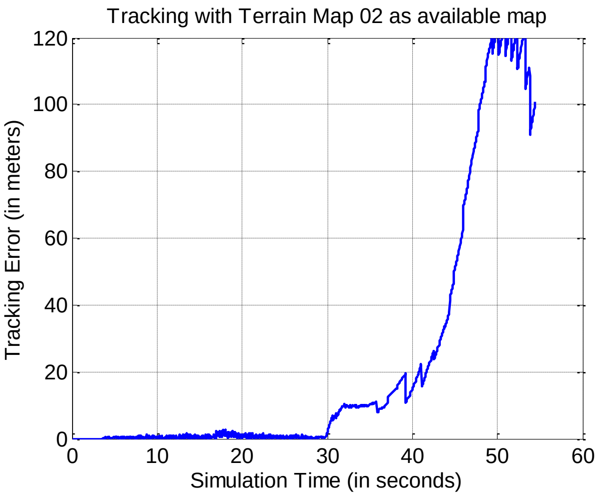


Figure 53: Tracking error with Estimator 2, assuming Terrain Map 02 is the true representation of current terrain

Figure 54 denotes the probabilities for each terrain map being the true representation of the terrain. It was observed that before reaching the intersection, the probabilities for both terrain maps were 0.5 since they represent the identical terrain as evinced by Figure 49. After reaching the intersection (at approximately 30 seconds into the run), the probability of the terrain map in Estimator 1 being correct increased to 1, whereas the probability of the terrain map in Estimator 2 being correct dropped to zero. At this juncture, the position estimate from Estimator 1 can be provided to the IPS layer for fusion with other position estimates.

The figure below shows the probability for each Terrain map. The probability stays at .5 until the point of intersection, at which time the probability increases to 1 (for Terrain Map 01) or 0 (for Terrain Map 02).

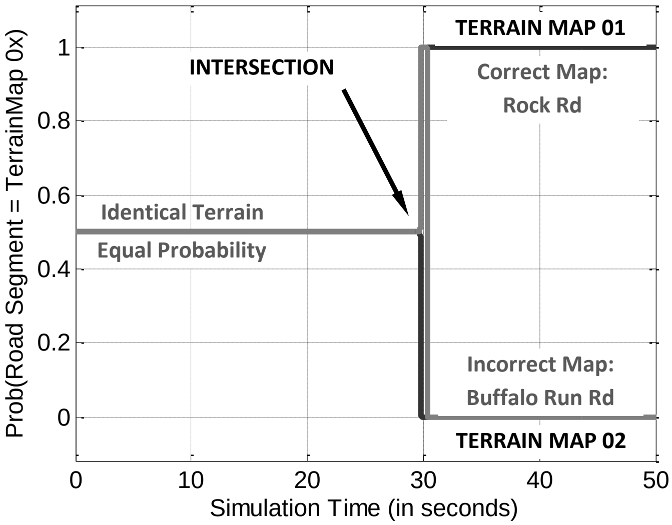


Figure 54: Probability that Terrain Map 0x is the correct map. Position estimates from road segments with high probability are chosen to be fed to the IPS

3.4.2 Server Setup

In order to provide a system that can work over an entire road network, the terrain database of such a network must be efficiently stored and quickly retrieved from disk, rather than be stored on RAM. After demonstrating the proof of concept that its GPS-free terrain-based vehicle tracking algorithms work in real time and across intersections, Penn State proceeded to setup an SQL Server to process queries and retrieve terrain map data from an ad-hoc on-board database.

Figure 55 depicts the architecture for the client-server setup. The client is the GPS-free terrain-based vehicle tracking algorithm that is running on one computer. The client sends the current position estimate requests the server to retrieve a local terrain map centered about the current position estimate. The server is hosted on-board on a second computer. The MATLAB script on the server side receives the current position estimate and queries the SQL server to retrieve a set of possible terrain maps. These local terrain maps centered about the current position estimate are then returned to the client.

The figure below shows the client/server setup block diagram. The “Ad-hoc on-board Server” block contains to sub-blocks, “Terrain DB – SQL server” and “Service handling module – MATLAB script” which point to one another. The “ad-hoc on-board Server” block points to the “Client” block with the description “Current position estimate sent to server” and is pointed to by the “Client” block with the description “Position-referenced Local Terrain Map returned to client” with “TCP/IP communication”. The “client” block has a sub-block with the description “QuaRC/Simulink running GPS-free tracking algorithm”.

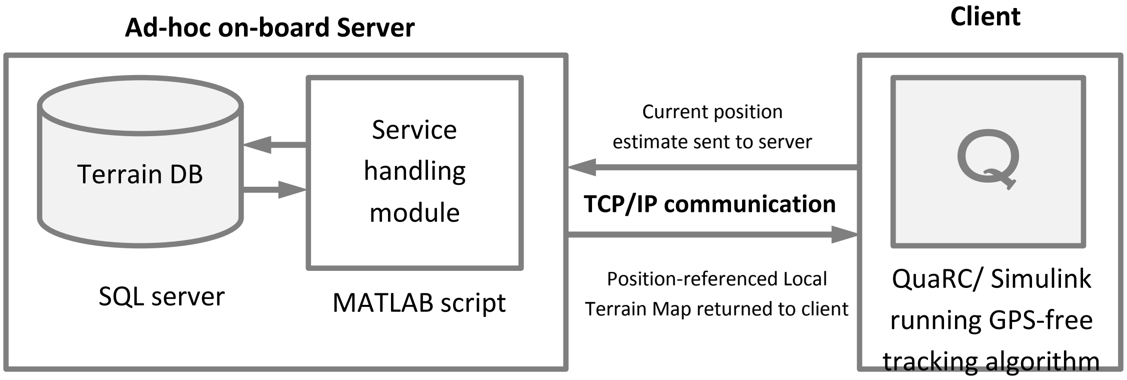


Figure 55: Client/Server setup for retrieving local terrain maps from a large road network database

At the time of completion of the project, Penn State had completed the setup of the SQL server, the querying script, and TCP/IP communication channel between the server and the QuaRC environment running the tracking algorithm. Each element of the setup was tested and was found to be functioning satisfactorily.

3.5 Road Fingerprinting Results – Auburn

In order for the Penn St. system to provide position solutions to the vehicle based off of the environment the path of the vehicle must first be surveyed. The NCAT test track was initially surveyed for Penn St. as per their instructions by logging both high and low grade IMU information, RTK GPS corrections, and wheel speed provided by the vehicle's CAN bus. This data was logged at varying speeds (30-50mph) on the outside lane of the test track. This data was then processed to provide high fidelity measurements of the vehicle's position and attitude. This filtered data was then transferred to Penn St, where pertinent data was extracted to create a map of the outside lane of the test track.

Upon delivery of the Penn St system along with the outside lane survey, the vehicle drove the same path to compare the delivered system to a truth system. The truth system consisted of logging RTK GPS data and comparing this position to the one reported by the Penn St. system. Stand-alone GPS was also recorded purely for a basis of comparison.

The figure shows solid lines representing the lane markings at the NCAT facility. The outside marking is a solid red line. The inside marking is a solid green line. The middle lane marking is represented by a dashed blue line. The positions reported by the PSU road fingerprinting method are represented by black "x" marks. Magenta circles represent the positions reported by a Novatel GPS receiver with no external aiding. The PSU markings are consistently spaced within the outer lane of the track. The standalone GPS position markings exhibit irregular spacing and occasionally leave the lane of travel.

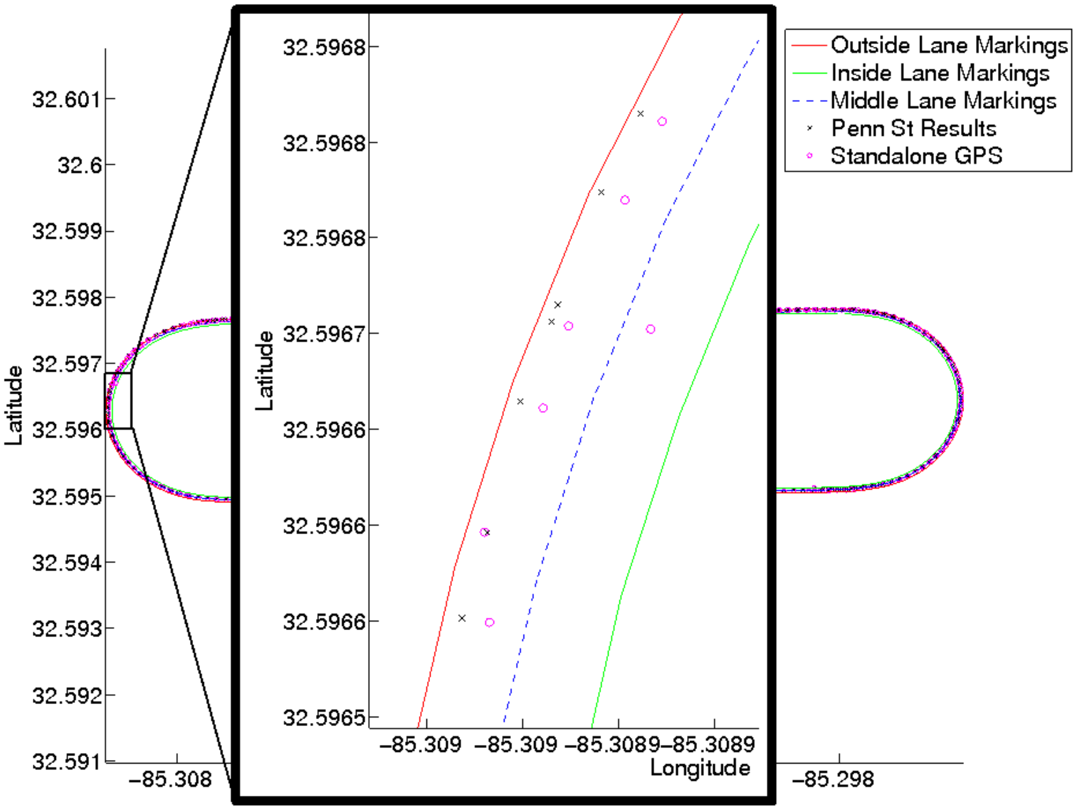


Figure 56: Position as reported by PSU road fingerprinting compared to standalone GPS at NCAT.

Note in the accompanying figure, a plot of our test track is shown with an area zoomed in on to provided additional insight. Note that at a glance there are no large jumps in the position solution from Penn St and while it often very closely represents what the vehicle is doing, there was the occasional jump in position into the adjacent lane. Based on a number of different tests, on average the Penn St. system was accurate to within 0.75 to 1m of the true position as reported by the GPS. Additionally, it is worthwhile to note that the system reported lane level accurate measurements over 80% of the time.

4 Kapsch – DSRC Ranging

4.1 DSRC Implementation

4.1.1 Received Signal Strength Indicator

The DSRC signal strength is measured in Received Signal Strength Indicator (RSSI). The RSSI is an integer ranging from 0 to 100 that is proportional to the power in the received signal. The goal of the initial testing was to find a correlation between the Received Signal Strength Indicator (RSSI) reported by the DSRC radio and range between radios. This correlation could then be used estimate range between radios using the report RSSI. In order to find a correlation between RSSI and range, both RSSI and range were logged for ranges varying between 1 and 100 meters. The location of the test was on a skid pad located at the NCAT test track in Opelika, Alabama. One DSRC radio was attached to a pole and placed on one end of the skid pad. The other radio was placed in the test vehicle. The antenna for the vehicle based DSRC radio was located on the back of the roof of the vehicle. Both antennas were placed at approximately the same height above the ground.

In order to get a variety of ranges to use to compare to RSSI, the test vehicle was driven at idling speed both towards and away from the fixed DSRC radio. This data was used to create curve fits for the data. The RSSI was fit to both a 1st and 2nd order quadratic function using least squares. The input to these functions is RSSI and the output is estimated range. There is a noticeable difference in RSSI for when the vehicle is traveling away from the fixed radio as opposed to when the vehicle is traveling towards the fixed radio; therefore, two different sets of 1st and 2nd order curve fits were computed; one for when the vehicle was traveling away from the base radio and one for when the vehicle was traveling towards the base radio.

The graph below shows RSSI a function of distance between radios. The data is split into sets groups. One set of data is for when the test vehicle is traveling away from the base DSRC radio and it is represented by red triangles. The other set of data is for when the test vehicle is traveling towards the base DSRC radio and it is represented by green circles The RSSI is noticeably higher for the case when the vehicle is traveling away from the base radio.

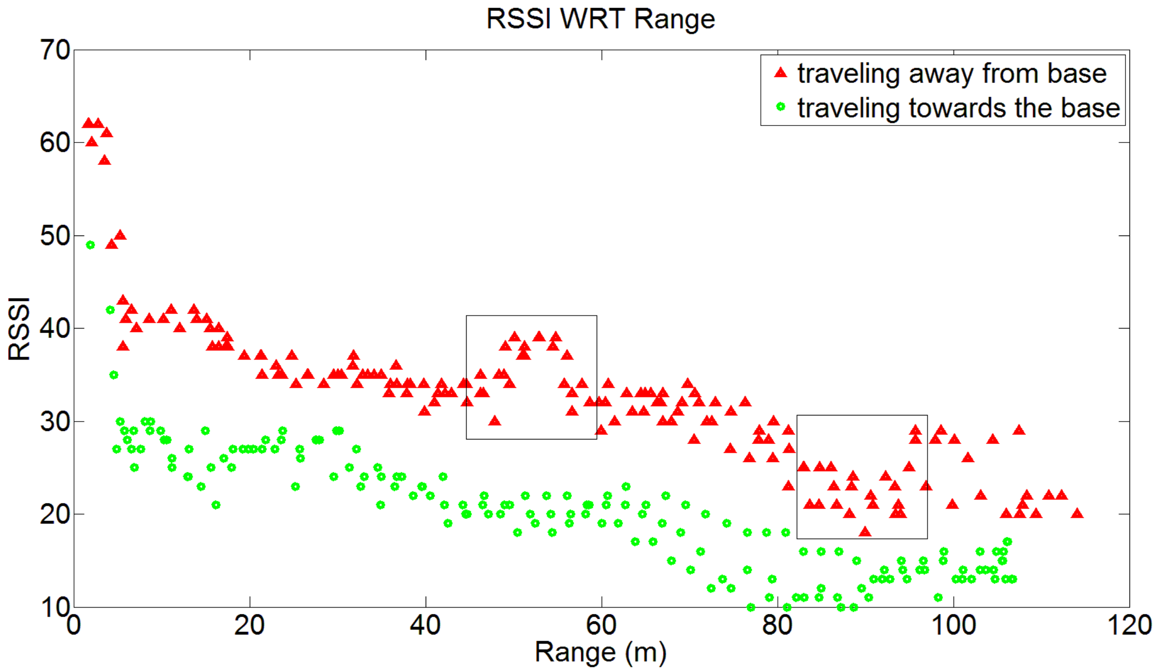


Figure 57: Plot of received signal strength indicator with respect to range between DSRC radios for when the test vehicle is traveling away and towards the base DSRC radio.

When using the curve fits to estimate range between using only RSSI, the standard deviation of the error in range is around 15 m^2. The signal strength fluctuates too much to create a strong correlation between range and signal strength. Signal strength could be used to get a general idea of the range between radios; however, range estimates based of signal strength are not accurate enough to incorporate into the current navigation filter.

The results of the experiment show that signal strength varies a great deal with respect to range between radios. One major source of fluctuation is obstructions between radios. Signal strength drops considerably when the signal travels through any solid medium. This can be seen by the difference in signal strength depending on the direction of travel from the base radio. The signal strength is significantly greater when the vehicle is traveling away from the base radio because the line on sight between the radios is clear. When the vehicle is traveling towards the base radio, signal strength is degraded due to the signal passing through the vehicle.

Another source of signal fluctuation is multi-path. Even if the line of sight between antennas is clear, signal strength can be degraded due to multiple copies of the same signal arriving at the antenna. One copy of the signal comes directly from the base DSRC radio. Ideally, this would be the only copy of the signal arriving at the user's radio. Other copies of the original signal come from the signal bouncing off a solid surfaces. The signal coming directly from the base radio and the reflected signals are combined at the antenna. The DSRC radio's measure of signal strength comes from the amplitude of the combined signal. If the direct signal and the reflected signal are in phase, then the amplitude of both waves will be added together. This will cause a rise in signal strength. If the direct signal and the reflected signal are out of phase by π radians, then the amplitude of the reflected wave will be subtracted from the amplitude of the direct wave. This will cause a fall in signal strength. The effect of multi-path can be seen in the boxed in area of Figure 57.

Some of the energy in a reflected signal will be absorbed by the surface it reflects off. The amount of energy absorbed by the surface is dependent on the material of the surface. Paved roads reflect much of the energy in signals that bounce off of it. This causes the reflected signal to have a large amplitude which will cause a large drop in signal strength when the reflected and direct signal are out of phase. Studies have shown that strategic placement of wave absorbers can significantly decrease drops in signal strength due to multi-path. Fluctuation in signal strength due the signal reflecting off the road is most prevalent when range between radios is less than 100 meters. When the range between radios is less than 100 meters, the signal strength follows a Rician fading distribution. Past 100 meters, the signal strength follows a Rayleigh fading distribution. This 100 meter critical distance assumes that the antennas are at the same height.

4.1.2 Time of flight ranging

The figure below shows the timeline for time of flight ranging. One vehicle sends a request for acknowledgement. The base radio receives the acknowledgement and sends it back. In addition to the times of flight for sending and receiving, the receive request and send acknowledgement times are included.

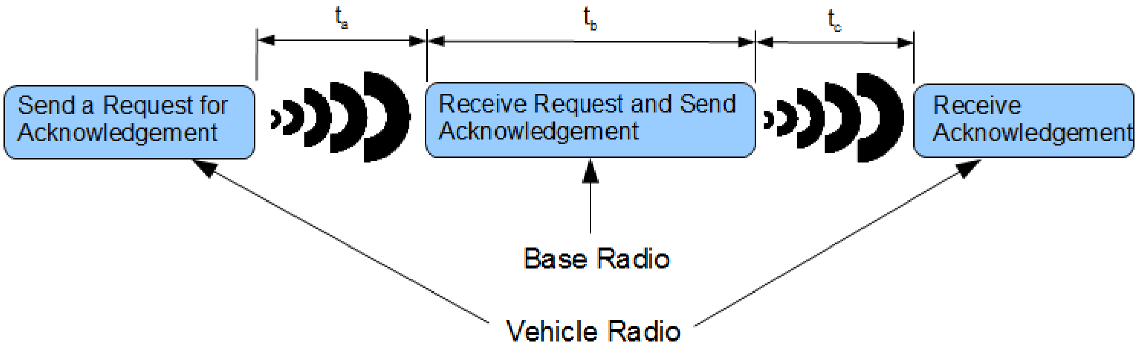


Figure 58: Timeline for time of flight ranging

4.1.2.1 DSRC Ranging Timeline

Figure 58 shows the timeline for time of flight ranging using DSRC radios. Since the clocks on the radios are not synced, a two-way time of flight method will be used to estimate range. The vehicle's radio will measure the overall time of flight and sends a message requesting acknowledgement from a base radio. The vehicle's radio must also start a precise timer when the message is sent. Ideally, this time will start as soon as the message starts to propagate from the radio's antenna. The base radio will receive the request and then send an acknowledgement. The turnaround time (t_b), or the time it takes the base radio to receive the message and send the acknowledgment, must be known in order to compute the range between the radios. Once the vehicle's radio receives the acknowledgement, it will stop the timer. Ideally the timer will stop as soon as the vehicle's antenna receives the message.

$$t_{total} = t_a + t_b + t_c$$

The vehicle's receiver measures total time of flight. The total time is the sum of the flight time from the vehicle to the base (t_a), the turnaround time at the base station (t_b), and the flight time from the base back to the vehicle (t_c). If the turnaround time (t_b) is known, the equation below can be used to estimate the range (r) between the vehicle's radio antenna and the base radio's antenna. c is the speed of light.

$$r = c\frac{(t_{total} - t_b)}{(2)}$$

There are several considerations that must be investigated when using the flight time method for ranging. Most of these concerns center on the necessary precision of the t_b and t_{total} times. Since these times are multiplied by a large number (the speed of light), an inaccuracy in the times will result in a large range estimate. An error of one microsecond (1/100000 of a second) will result in a 300 meter error in range. The signal traveling at the speed of light will travel approximately one foot every nanosecond; therefore, the total flight time (t_{total}) and the turnaround time (t_b) must be known to at least to the nanosecond to result in a range accuracy of one foot.

The DSRC radios have an onboard 1GHz processor. Theoretically, the processor should be able to measure time to the nanosecond because the cycle time of the processor is 1 nanosecond. Another issue is error in the total time due to the timer not starting and stopping precisely when the message is being sent/received at the antenna, and it may be necessary to estimate the time from when the time of flight timer is started and when the request message starts to propagate from the antenna. Similarly, it may also be necessary to estimate the time difference from when the acknowledgement is received at the antenna and when the time of flight timer is stopped. Since these time differences are based on the receiver's hardware, they should be constant.

The time it takes the base radio to receive the request and send the acknowledgement must also be known to the nanosecond in order to have an accurate range estimate. One method of determining the turnaround time is setting up the antennas at a known range and measure the time of flight. The known range can be used to estimate the turnaround time. The estimated turnaround time can be used as a constant. This method will require that the turnaround time be constant on the nanosecond level. Due to software processes involved in the turnaround time, it is unlikely that this time will be constant to the nanosecond. Another method of determining the turnaround time could involve the base radio estimating the turnaround time using its own timer. This time could then be sent in the acknowledgement or another message that is sent right after the acknowledgement. The vehicle's receiver will then have an estimated turnaround time (accurate to the nanosecond) to use in the range estimation.

Another issue will be what value is use for the speed of light. The speed of light is 299,792,458 meters per second in a vacuum; however, the speed of light is slowed when traveling through a medium like air. The refractive index of a medium is the ratio of the speed of light trough the medium and to the speed of light in a vacuum. There are methods on determining the correction to the speed of light through air based on current properties of the air like temperature and pressure.

4.2 DSRC Interfacing

The initial strategy for the DSRC ranging was as follows:

1) KTC supplied two (2) DSRC radios (MCNU R1500) to AUR. AUR setup radios on the bench.
2) KTC and Auburn agreed on the initial installation and testing strategy. One radio will be installed on the test vehicle. The other radio will be installed at the NCAT test track. Once both radio are installed and interfaced, preliminary range testing will begin.
3) KTC prepared initial version of the ranging test software.

4.2.1 Hardware Description

Each DSRC radio is enclosed in a Multiband Configurable Networking Unit (MCNU). The MCNU is a computer in a rugged enclosure. Inputs into the MCNU include 3 Ethernet connections, 1 serial port, 2 USB connectors, 4 N-type radio antenna connectors for DSCR communications, and 1 SMA connector for a GPS antenna. The MCNU has a built in ublox GPS receiver. The DSRC radio antenna for the car is magnetic for easy installation on the test vehicle. The test vehicle's DSRC antenna also contains a GPS antenna. The MCNU on the test vehicle will be interfaced using an Ethernet connection. The MCNU is connected to a router along with the existing PC in the trunk of the test vehicle. This will allow information to pass between the MCNU and the PC. Also the router will allow remote connection to both the PC and the MCNU.

Another identical MCNU will be used to simulate roadside DSRC infrastructure. The location of this MCNU has not been decided. The decision on where to set up the roadside MCNU will depend on the ranging capabilities and limitations of the unit. More than likely, a few mounting points will be chosen at the track with the ability to quickly move the MCNU between sites for different types of testing.

4.2.2 Initial Range Testing

The initial plan for testing the ranging capabilities of DSRC radios will be to simply collect range strength and "truth" range measurements. The range measurements will come from RTK GPS measurements. The ranges provided by the RTK GPS will be accurate on the centimeter level. The range measurements will be compared to the signal strength measurements to see what correlation exists between the measurements. This will give us an idea of the accuracy that can be expected when only using DSRC signal strength to estimate range between receivers.

4.2.3 Testing Software

The initial testing will utilize the test software running on both radio devices. The transmitting portion of the test software will send test messages using 5.9GHz DSRC wireless link. All messages will be transmitted with the fixed power in broadcast mode using omni-directional antenna. The receiver radio will execute the receiving portion of the test messages which will log received messages and their characteristics. For each message, the receiver log will capture signal Receive Signal Strength Indicators (RSSI), time stamp, and a message index. Also, the receive radio will log GPS information from the built-in Ublox GPS receiver.

KTC has prepared the initial version of the test software for the transmitter and receiver radios. KTC will provide the software source to AUR for further customization. Customization may include optimization of the log files and message parameters to allow for easier evaluation and comparison of the collected data to the RTK GPS data.

4.3 DSRC Results – Kapsch Testing

Kapsch implemented a modification to the 5.9 DSRC radio driver which allows to measure the turn-around time for a packet exchange between and RSE and OBU. The results indicate that the turn-around time is correlated to distances measured between vehicles.

The figure below shows two vehicles facing each other. The left vehicle is labeled as roaming, while the right vehicle is labeled as stationary.

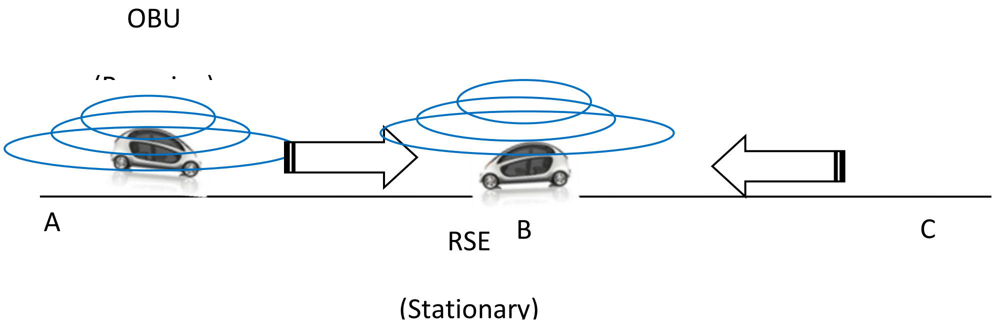


Figure 59: Field testing of the DSRC ranging

Kapsch conducted field testing of the DSRC ranging. The tests were conducted in order to verify functionality of the radio to detect and measure variations in distances between communicating radios.

4.3.1 Testing

A vehicle with an RSE MCNU R1500 was position in the middle of a street. The second vehicle had an MCU on-board unit (OBU). The new ranging software was installed on the MCU. The second vehicle was traveling between two points A and B selected at the opposite ends of the street. Ranging logs were recorded on the MCU OBU. The range between vehicles was determined using laser range meter.

4.3.2 Results

Results for two tests are shown below. In the first experiment, vehicles were initially positioned 105.3 meters apart. In the closes point, vehicles were separated by about 12 meters. The chart shows measurements of times for packet roundtrip exchange between the OBU and RSE. The chart shows that the time changed in proportion to the distance between vehicles.

The figure below shows the measurements of times for packet exchange. At A, horizontal lines extend out around the 1573-1572 range. Further horizontal lines cascade down in a stairway-like pattern until B, where the horizontal lines are once more at 1562-1564 range. The lines build up once more in a stair-like pattern until a spike, which is followed by location C.

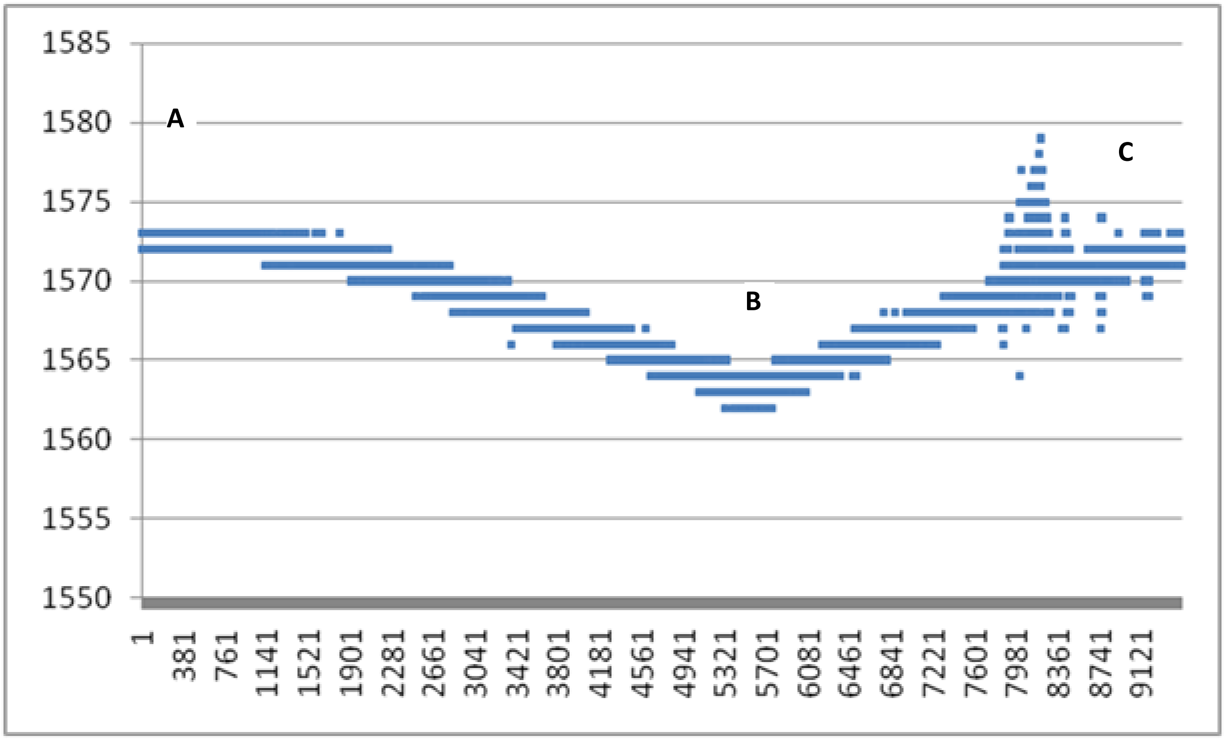


Figure 60: Measurements of times for packet roundtrip exchange

The second chart shows the second test. In this test, the initial position between the vehicles was 258 meters. The roaming vehicle travelled toward the stationary vehicle. The roaming vehicle stopped around point B and then proceeded to the point C. At point C the vehicle turned around and returned to the point B.

The figure below shows another test for measurements of times for packet roundtrip exchange. Once more there are locations labeled A, B, and C. However, in this plot the horizontal lines begin at A, drop down to B, increase to C, then drop down back to B.

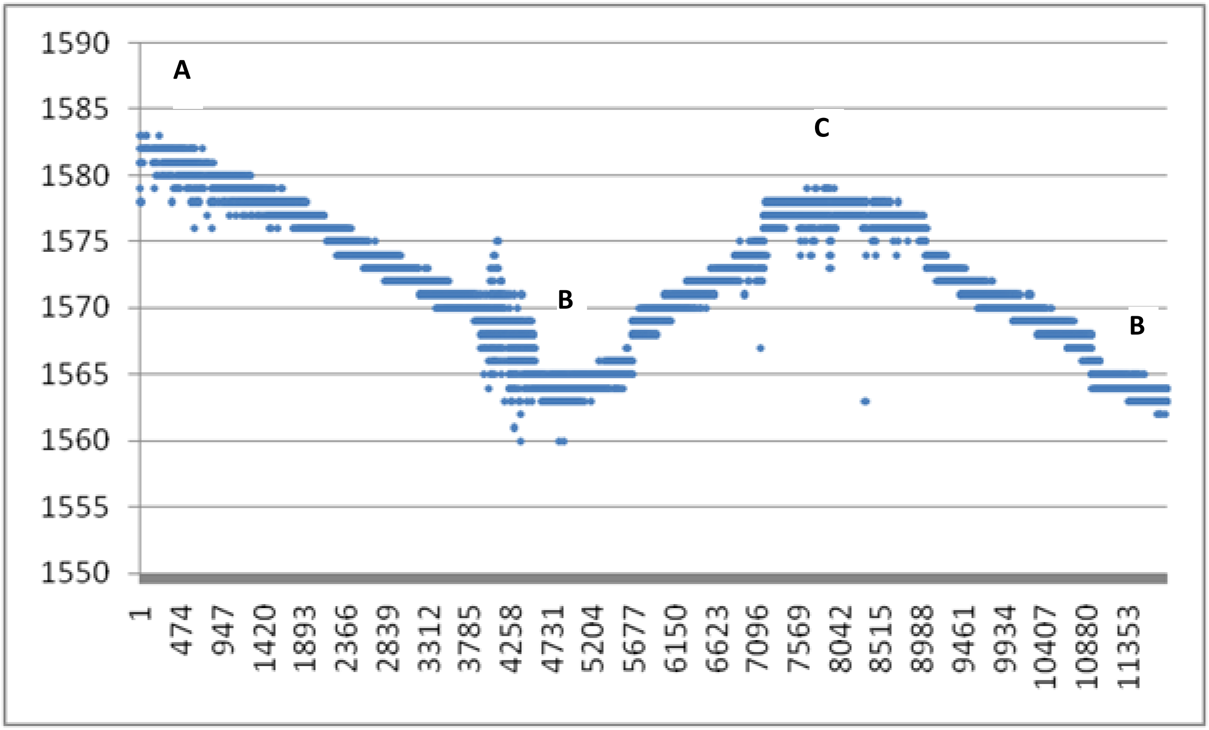


Figure 61: Second test of measurements of times for packet roundtrip exchange

4.4 DSRC Testing - Auburn Testing

The ranging test was performed using our research vehicle, which is equipped with Novatel Differential GPS and a DSRC on-board unit. The onboard GPS and DSRC antenna were placed on the roof of the vehicle. The base station DSRC antenna was placed at roughly the same height as the vehicle. Both radios were configured by following the guide that Kapsch had provided. Using the Auburn's interface to Mission Oriented Operating Suite (MOOS), the range measurement from the DSRC radio and the position data from the Novatel GPS receiver was collected and recorded. The test consisted of the static and dynamic portions. For the static tests, the vehicle was moved to the various positions around the skid pad and held stationary when the tests were running. During the dynamic tests, the vehicle was driven around the skid pad area of NCAT test track to vary the distance between the vehicle and the base station DSRC antennas. Clear line of sight for DSRC antennas was ensured for several of the test runs to avoid possible signal loss, while other tests intentionally had obstacles blocking line-of-sight. After collecting the data, the position of the base station was recorded using Novatel DGPS. Truth distance was obtained using the distance from the GPS position of the base station to the position of the receiver on the antenna. Truth data shows precision within 2 centimeters.

4.4.1 Static Testing

Static tests were also performed from various distances – 4.5m, 35m, 63m, and 72m. The results are shown in Figure 62 to Figure 69. Table 1 shows the mean and the standard deviation of the time of flight values from various distances.

The figure below shows two graphs. The top graph shows a plot ranging from 4.5 to 4.53m over a period of 80 seconds. The bottom graph shows two discrete lines between 702 and 703.

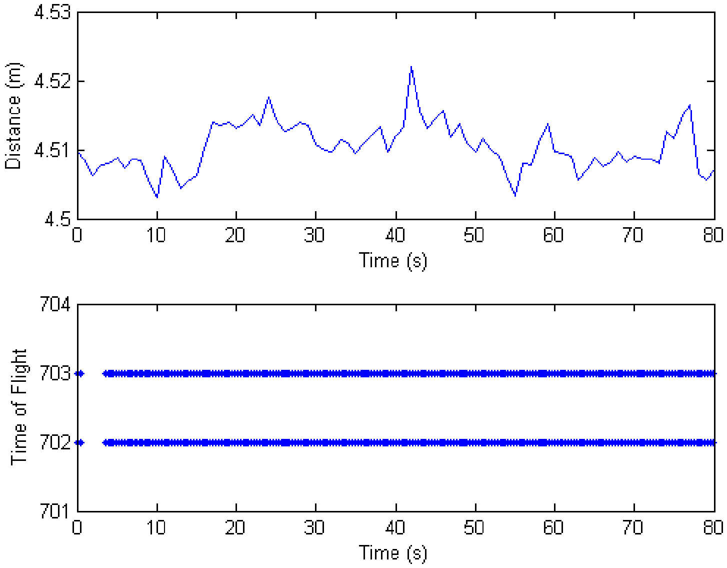


Figure 62: Close Range Static Test Result

The plot below shows a histogram of the previous plot. The 702 data point has about 7000 counts, while the 703 data point has about 26000 points.

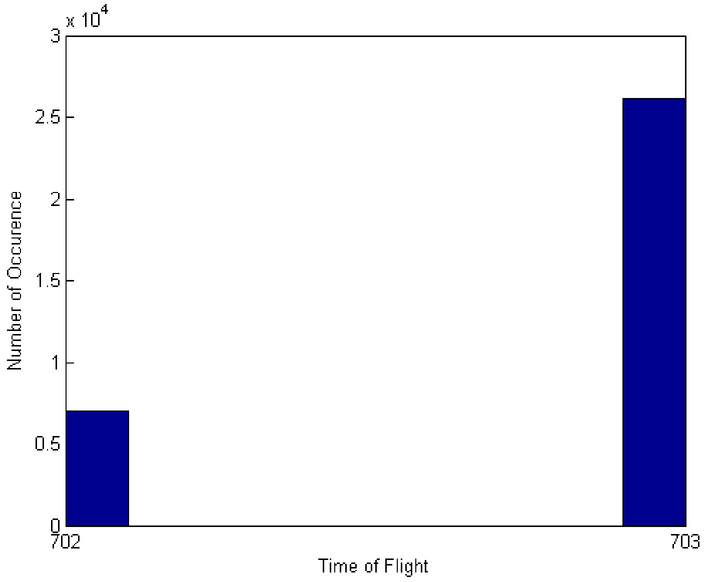


Figure 63: Close Range Static Test Histogram

The figure below shows two plots with time ranging from 0 to 160 seconds. The top plot ranges from 35.32 m to 35.335 m. The bottom plot shows a discrete plot with three values, 705, 704, and 703.

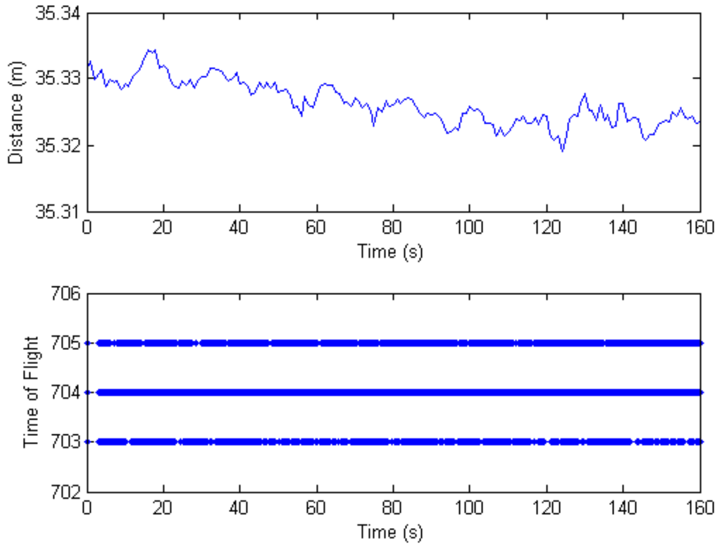


Figure 64: Medium Range Static Test Result

The figure below shows the histogram of the previous plot. The 703 data point has 5000 points, 704 has 60000 data points, and 705 has 7500 points.

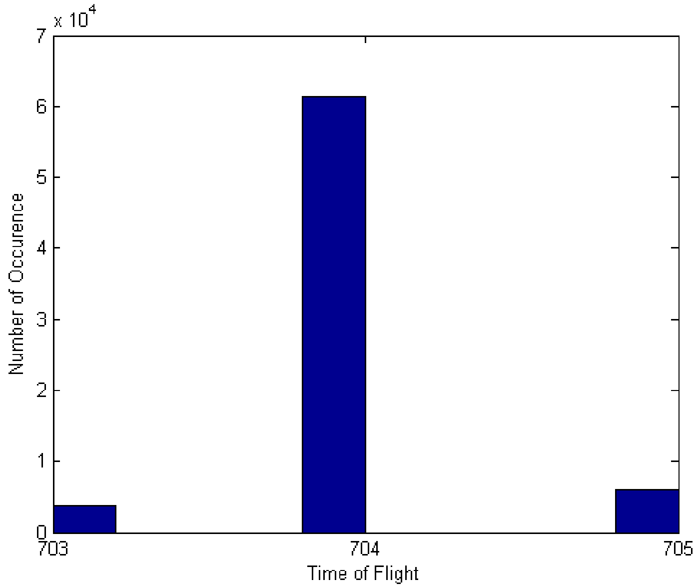


Figure 65: Medium Range Static Test Histogram

The figure below shows two plots with time ranging from 0 to 160 seconds. The top plot ranges from 35.32 m to 35.335 m. The bottom plot shows a discrete plot with six values, 706, 705, 704, 703, 702, and 701.

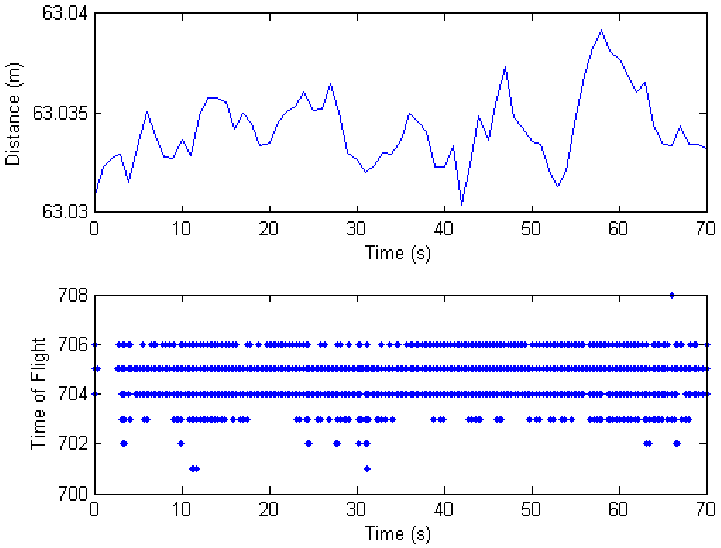


Figure 66: Long Range Static Test Result

The figure below shows the histogram of the previous plot. The 701 data point has <500 points, 702 data point has <500 points, 703 data point has 500, 704 data point has 3300 points, 705 data point has 9000 data points, 706 data point has 1200 points, and 708 has very few points.

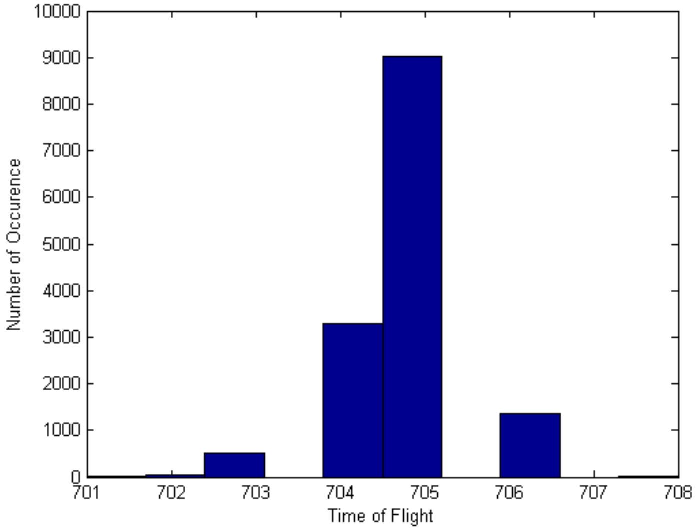


Figure 67: Long Range Static Test Histogram

The figure below shows two plots with time ranging from 0 to 160 seconds. The top plot ranges from 72.84 m to 72.85 m. The bottom plot shows a discrete plot with seven values, 701, 702, 703, 704, 705, 706, and 707.

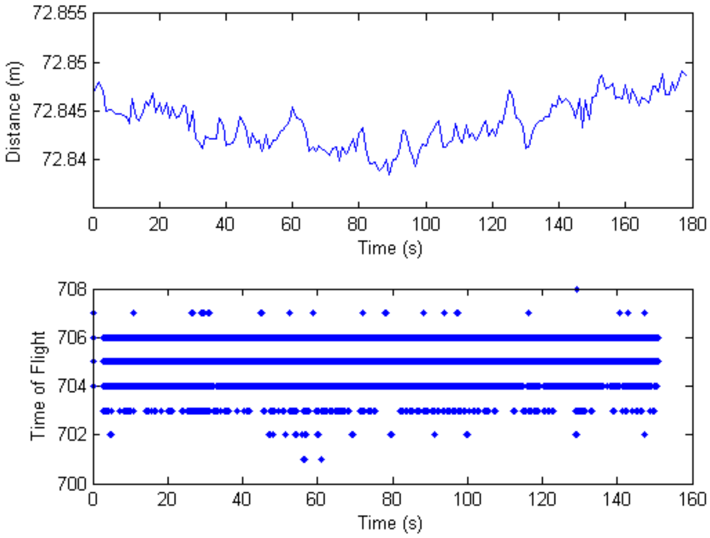


Figure 68: Maximum Range Static Test Result

The figure below shows the histogram of the previous plot. The 701 data point has <500 points, 702 data point has <500 points, 703 data point has 2000, 704 data point has 8000 points, 705 data point has 32000 data points, 706 data point has 20000 points, 707 has few data points, and 708 has very few points.

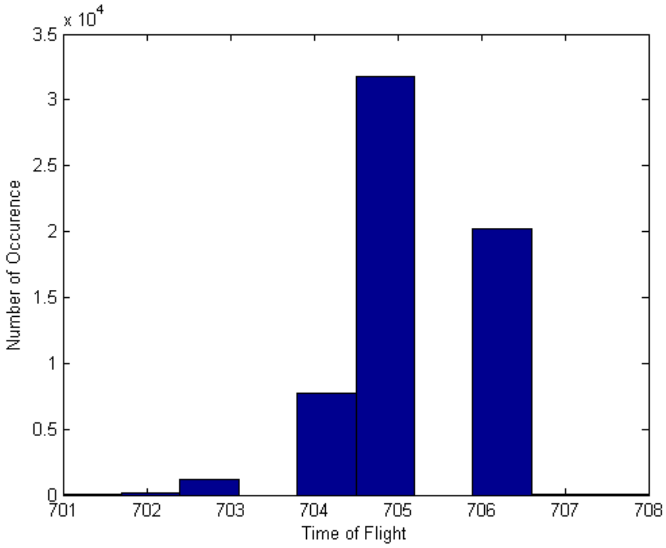


Figure 69: Maximum Range Static Test Histogram

Table 4: Means and standard deviations for the time of flight

Test Range (m)	Mean	Std. Deviation
4.5	702.7899	0.4074
35	704.0324	0.3703
63	704.7769	0.6896
72	705.1634	0.7277

4.4.2 Dynamic Testing

On the first test run, the vehicle was driven away from the base station and maintained line-of-sight. At the end of the skid pad area, the vehicle was put in reverse and driven back to near the starting position, as seen in Figure 70.

The figure below shows the GPS positions for the dynamic test. A red + signifies the position of the GPS base station at (0,0). The blue lines represent the locations of the vehicle. The locations of the vehicle goes from (20,20) goes to (95,-45), and then returns to the approximate location of the starting position.

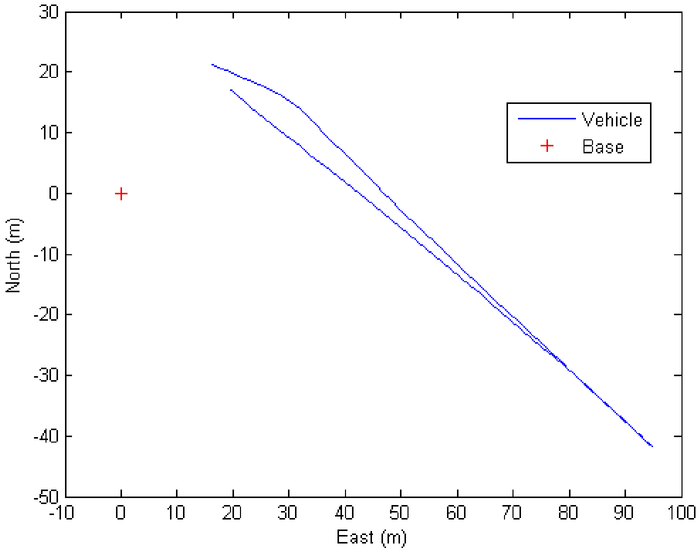


Figure 70: GPS position of the vehicle and the base station

The result of this run can be seen in Figure 71. The signal flight time increased as the distance between the antennas increased. Also, the flight time changed by 1 for about every 13 meters of distance.

The figure below shows two plots with time ranging from 0 to 140 seconds. The top plot shows a curve with the distance ranging from 25 meters at the start to 100 meters at the peak. The bottom plot shows a discrete plot which has an approximate shape of the distance curve from the top plot but with noisy distance. The time of flight ranges from 701 to 707 throughout the run.

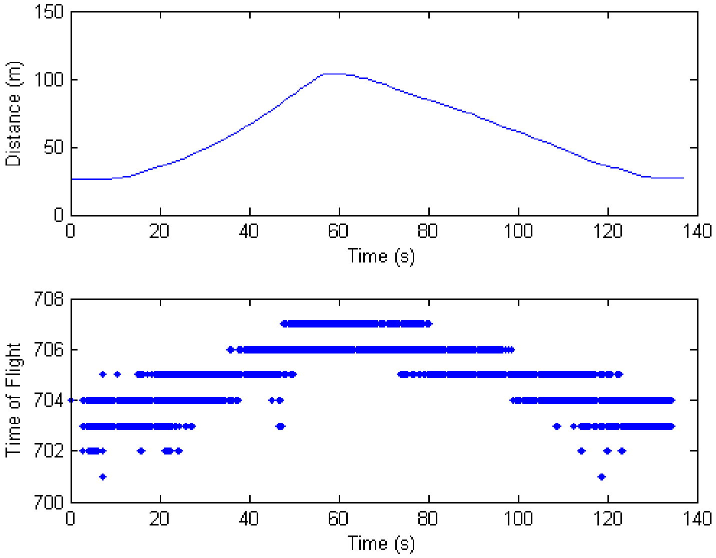


Figure 71: Distance and ranging output vs. time

On the second test, the vehicle was driven outside of the skid pad to provide a larger distance between the radios. In this test, line-of-sight was not maintained due to trees and large rock piles present on the skid pad. The vehicle path and the output result can be seen in Figure 72 and Figure 73, respectively. The vehicle began at the base station (red plus) and traveled counterclockwise to the easternmost part of the path. At this point, the car was put in reverse until the looped portion, where it was put in drive and the loop completed. Along the reversed region to the east of the loop, trees were present. Also, two rock piles blocked line of sight along the southernmost path of the vehicle.

The figure below shows the GPS positions for the second dynamic test. A red + signifies the position of the GPS base station at (0,0). The blue lines represent the locations of the vehicle. The locations of the vehicle is a loop in a small circle and with the furthest point with a straight line away.

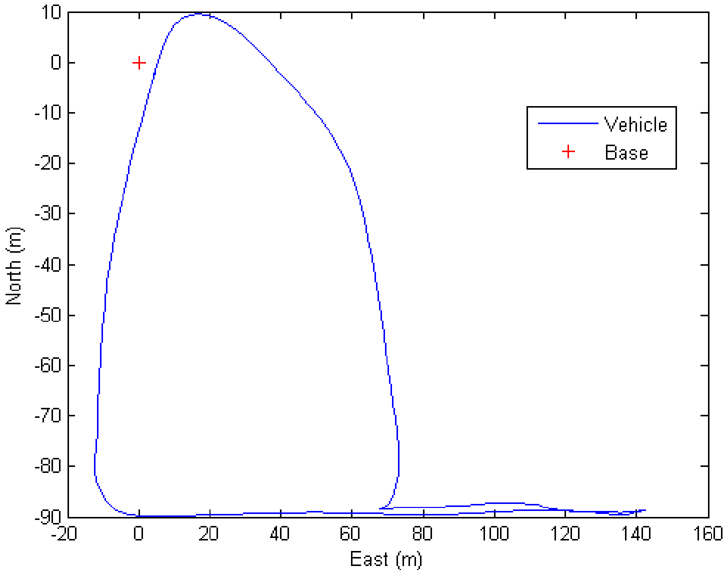


Figure 72: GPS position of the vehicle and the base station

The figure below shows two plots with time ranging from 0 to 140 seconds. The top plot shows a curve with the distance ranging from5 meters at the start to 150 meters at the peak. The bottom plot shows a discrete plot which has an approximate shape of the distance curve from the top plot but with noisy time of flights. The time of flight ranges from 701 to 710 throughout the run.

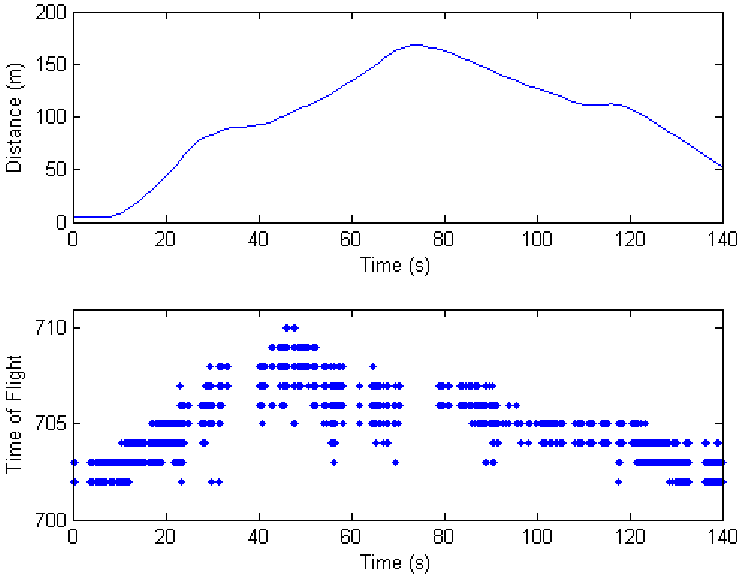


Figure 73: Distance and ranging output vs. time

Due to the rock piles blocking the line of sight of radios, the ranging measurements were lost around 30 seconds and 75 seconds into the test. The test result showed similar results to previous tests with time of flights fluctuating between 2 time of flights and 3 time of flights depending on the distance. An attempt at greater distances on the track showed that the radio was unable to receive data beyond around 710 time of flight.

5 Auburn Vision Sensors

Drawing on previous work, Auburn contributed two additional subsystems to the project - lane detection and road edge detection. Both of these subsystems were implemented with a camera and a lidar.

5.1 Lidar Lane Detection

The lidar-based lane detection algorithm is based on fitting an ideal lane model to actual road data, where the ideal lane model is updated with each lidar scan to reflect the current road conditions. Ideally a lane takes on a profile similar to the 100 averaged lidar reflectivity scans seen in the Figure 74 below, where the corresponding road is seen in the lower image. The plot is a mirror image of the photo due to the fact that the right portion of the image corresponds to negative horizontal angles. Note that this profile has a relatively constant area between the peaks, where the peaks represent the lane markings, which are typically bright and thus have higher reflectivity. The constant area is the unmarked pavement of the road surface, which is typically dark and thus has lower reflectivity. An ideal lane model is generated with each lidar scan to mimic this averaged data, where the constant portion is generated by averaging the reflectivity directly in front of the vehicle and the lane markings are generated by increasing the average road surface reflectivity by 75%. This model is then stretched over a range of some minimum expected lane width to some maximum expected lane width, and the minimum RMSE between the ideal lane and the lidar data is assumed to be the area where the lane resides.

The figure below shows an ideal echo width from reflectivity. Two vertical black lines are present at -30 degrees and 25 degrees. At -20 degrees, a "1" is listed. At 5 degrees, a "2" is listed. At 7 degrees, a "3" is listed. Finally, at 30 degrees, a "4" is listed. Peaks are present at each of these listed numbers, with the other data hovering near 2 echo width.

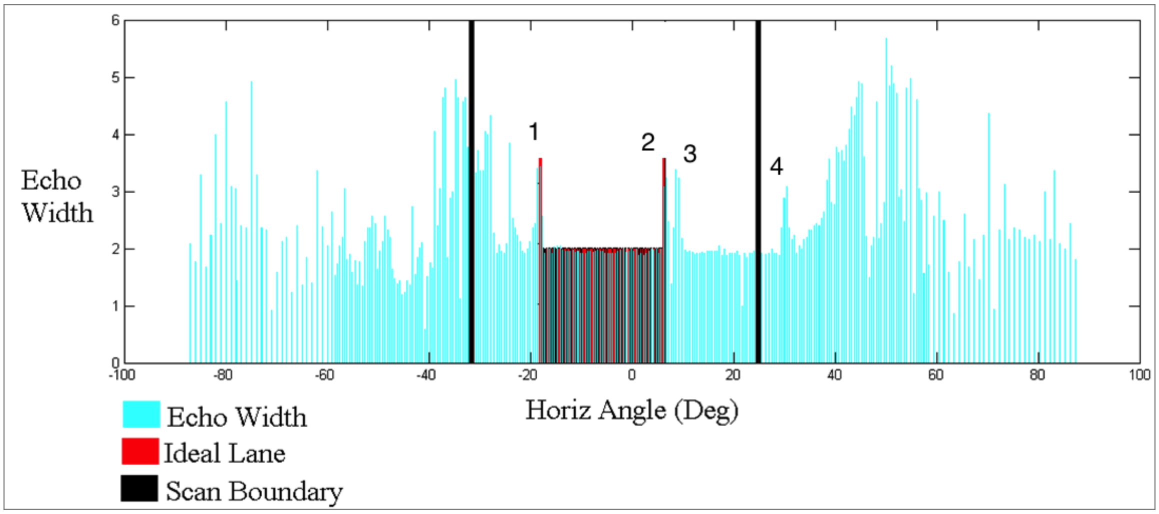


Figure 74: Echo Width from Reflectivity Scan

The figure below shows an image of the road ahead of a vehicle. Two red lines are painted on the edge of the road. A "4" is listed near the left most red line. A "3" is listed on the left side of the yellow double line at the center of the road. A "2" is listed on the right side of the yellow double line at the center of the road. A "1" is listed near the right red line.

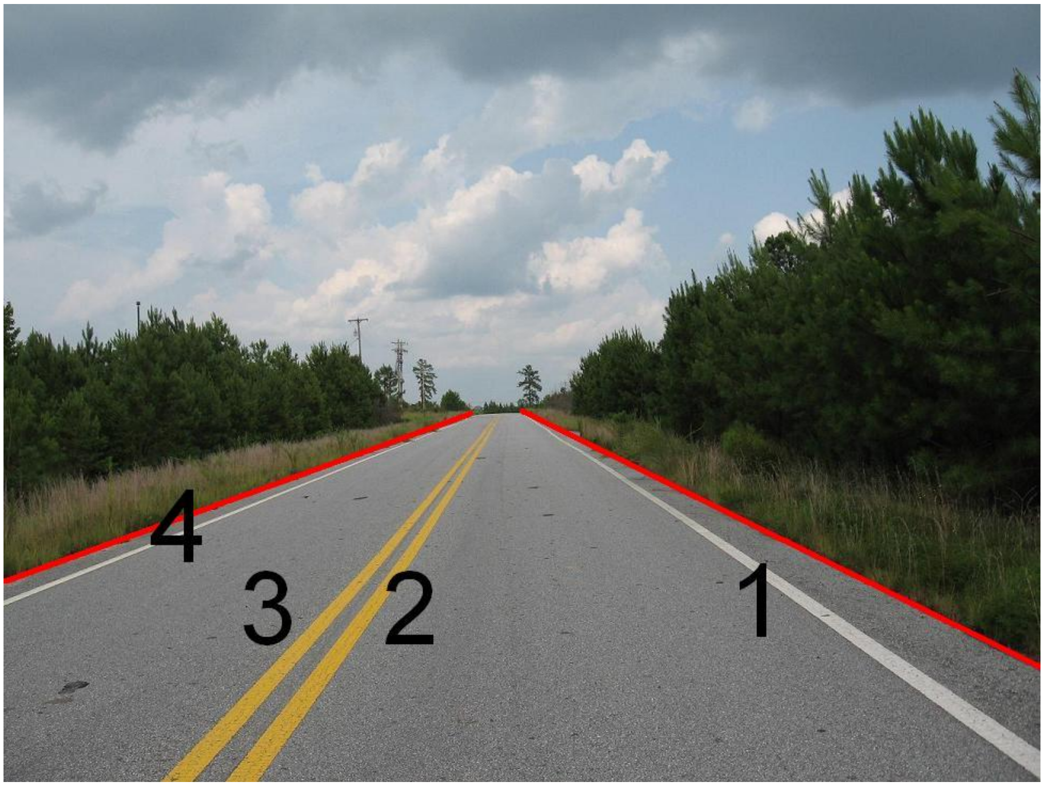


Figure 75: Road markings corresponding to Figure 74

5.2 Lidar Road Edge Detection

The lidar based road-edge detection algorithm was developed to supplement the performance of the previously developed lane detection method. Recall that the previous method of detecting lane markings incurred poor results on rural roads where the grass bordered the roadway, especially when there was no right lane marker. In this instance, the grass had a tendency to pull the lane marker detection algorithm off the road where the estimate would be reset as the error built up, obviously causing undesired performance results. To overcome problem, a road edge detection algorithm was implemented so that regardless of the presence of lane markings, the boundaries of the road could still be ascertained. Once these boundaries had been detected, the position of the vehicle in the road or lane could be computed.

5.2.1 Lidar Road Edge Detection Algorithm

The algorithm used to detect the road edges begins by first windowing the lidar data to minimize processing time. Then a series of filters are applied to both the distance and reflectivity data reported by the lidar. From this filtered data the edges of the road are extracted. This algorithm will be tested on a number of different rural roads at varying times of day to assess performance and sensitivity to outdoor conditions.

When a lidar scan of both distance and reflectivity is returned, the data is initially pre-filtered and bounded to aid in the detection process. First, the data is windowed so that regardless of where the vehicle is in the lane, the field of view will be large enough to see both road edges. Accounting for a slight shoulder and some vehicle yaw, we calculated that this comes to approximately 8m (3m to the right and 5m to the left). Additionally the distances in the lateral direction are quantized into 0.1m increments. This provides an even lateral measurement change will increase the speed at which detection algorithm is capable of running. Additionally and erroneous lidar return are removed here. These erroneous returns are typically measurements that are less than a meter in distance, and typically constitute occlusion on the lidar screen.

With the pre-filtering complete, the detection of the road edge can begin in earnest. Both distance and reflectivity data are analyzed. The premise is that we can expect a transition in material and thus reflectivity as the road transitions into gravel, grass, or some other material. Additionally the distance in the vertical direction is analyzed using the premise that at this transition there is also a slight elevation change. This is often either due to grass bordering the roadway that is higher than the roadway or alternatively, the roadway is slightly higher than the surrounding land. Therefore, with the pre-filtering complete, a derivative filter is applied to both the reflectivity and distance data (in the vertical direction). This accentuates any changes in the data from the surrounding data-points. A dynamic threshold based on current road conditions is then applied to both reflectivity and distance datasets filtered datasets. The first values that are above these filtered values along with some neighboring value also above these thresholds are selected to be the true boundaries. The output of this process can be seen in Figure 76.

The figure below shows the lidar lane detection process. Six plots are shown, the distance and echo width are shown in the top two plots, the derived filtered distance and echo width in the center image, and the quantization distance and echo width in the bottom two image.

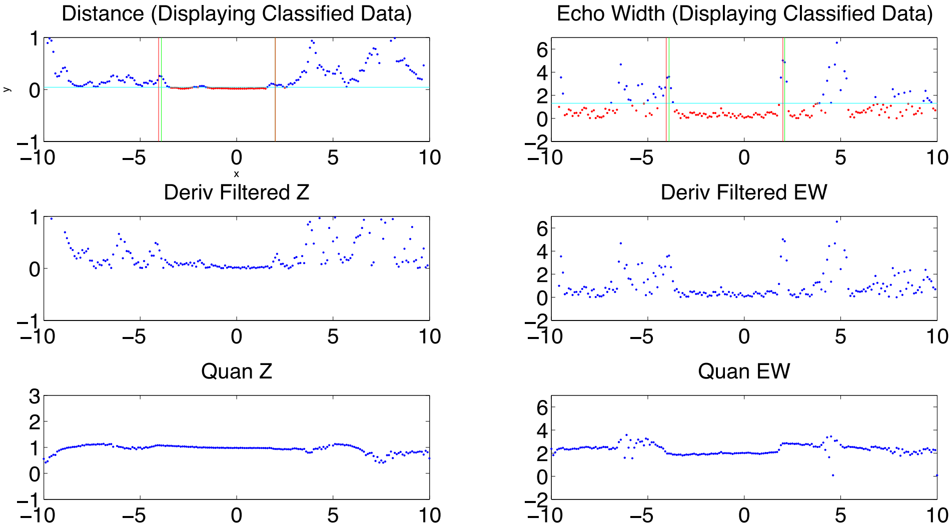


Figure 76: Lidar Lane Detection Process

The distance obtained from the reflectivity and range data are combined in a weighted average. This weighted average becomes the input to a Kalman filter with essentially acts as a low-pass filter to smooth out any spikes from erroneous detections.

5.2.2 Lidar Road Edge Detection Experimental Results

Testing was conducted by gathering lidar data on a number of rural roads outside of Auburn at various times of the day. Because we did not have an RTK survey of these roads, we chose to compare our results to the measured width of the road. It is worthwhile to denote that these results shown in Table 5 are conservative, in that often there are other driveways, intersecting roads etc, where no road edge exists. Also it was noticed that the roadway does fluctuated slightly in width. In the table below, a number of different roads are listed and results are compared on the grounds of mean absolute errors, the standard deviation of the error, and the detection rates.

Road	Mean Absolute Error (m)	Standard Deviation (m)	% Detection
Co Rd 084 (Day)	0.0876	0.1185	90.46
Co Rd 084 (Night)	0.0887	0.1198	91.26
Co Rd 188 (Day)	0.0378	0.1754	84.19
Co Rd 188 (Night)	0.0576	0.1509	90.49
MSJ Rd (Day)	0.0583	0.1297	93.66
MSJ Rd (Night)	0.0905	0.1426	93.15

Table 5: Lidar Road Edge Detection Results

5.3 Camera Lane Detection

The camera-based method for this project uses line extraction techniques from the image to detect lane markings and calculate a lateral distance from a 2^{nd} order polynomial model for the lane marking in image space. A threshold is chosen from the histogram of the image to compensate for differences in lighting, weather, or other non-ideal scenarios for extracting the lane markings. The thresholding operation converts the image into a binary image, which is followed by Canny edge detection. The Hough transform is then used to extract the lines from the image, fill in holes in the lane marking edges, and exclude erroneous edges. Using the slope of the lines, the lines are divided into left or right lane markings. Two criteria based on the assumption that the lane markings do not move significantly within the image from frame to frame are used to further exclude non-lane marking lines in the image. The first test checks that the slope of the line is within a threshold of the slope of the near region of the last frame's 2^{nd} order polynomial model. The second test uses boundary lines from the last frame's 2^{nd} order polynomial to exclude lines that are not near the current estimate of the polynomial. 2^{nd} order polynomial interpolation is used on the selected lines' midpoint and endpoints to determine the coefficients of the polynomial model, and a Kalman filter is used to filter the model to decrease the effect of erroneous polynomial coefficient estimates. Finally, the lateral distance is calculated using the polynomial model on the lowest measurable row of the image (for greater resolution) and a real-distance-to-pixel factor.

The figure below shows the road in front of a vehicle. Two green lines are painted over the white lane markings. Two parallel black lines are present around each of the green lines. Red lines are seen on the edge of the road on the road edge.

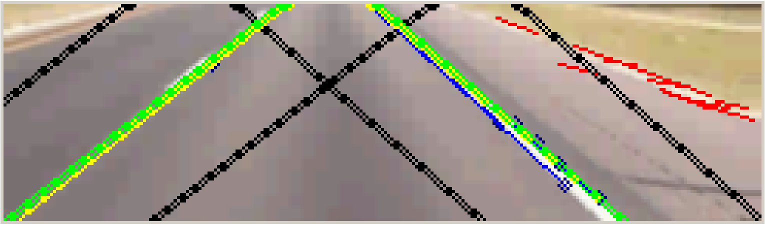


Figure 77: Camera-based lane detection (green-detected lanes, blue-extracted lane lines, red-rejected lines)

5.4 Camera Road Edge Detection

The camera road edge detection algorithm has several goals:

- Account for various road types
- Handle road obstructions including shadows
- Perform in various environments

The solution for these goals was based on one important method of navigation for humans – color. As such, a template-based algorithm using the color of the road was used to differentiate between the road and off-road.

5.4.1 Camera Road Edge Detection Algorithm

A template-based algorithm was developed to achieve these tasks. Figure 78 shows the basis of the template-based algorithm. By sampling the road in front of the vehicle, the pixels in the image corresponding to the road can be differentiated from the pixels corresponding to off-road features. At the same time, by sampling the road, the algorithm accounts for different types of road. Note that this strategy assumes that the road and off-road portions of the image have distinctive color features and that the template is assumed to be part of the road. For this reason, the template is typically taken from the region directly in front of the vehicle, which will typically be the road except for sharp turns and when the vehicle is driving off-road.

The figure below shows an image of a road in front of a vehicle which is bordered by grass and no lane markings. A square of the road is blown up, which shows the template used for determining the features of the road.

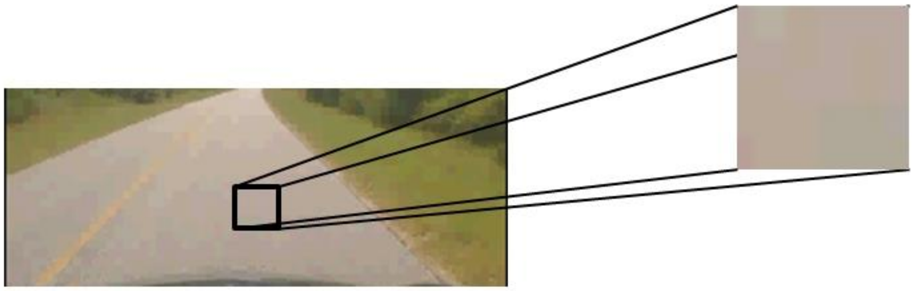


Figure 78: Template Matching for Road Edge Detection

A cross correlation matching algorithm using the template of the road is employed to extract the regions of the image that correspond to the road. Figure 79 shows the unnormalized (left) and normalized (right) cross correlation matching images (*R*) extracted from the image using the template of the road.

The figures below show the correlation matching results (R). The left view shows the unnormalized correlation, where the road is gray and the grass dark. The right view shows the normalized correlation, where the road is white and the grass dark.

Figure 79: Unnormalized Correlation Matching (left) and Normalized Correlation Matching (right)

As seen from the images, the road features are brighter than the non-road portions of the image *R*. The normalized cross correlation matching, especially, provides a better solution for varying lighting and is ultimately the correlation matching algorithm used for the algorithm. The normalized cross correlation equation to calculate the resulting correlation image *R* is given below

$$R(x,y) = \frac{\sum_{x',y'}(T(x',y') \cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}$$

where *T* is the template image, *I* is the main image, and

$$I'(x+x',y+y') = I(x+x',y+y') - \frac{1}{wh}\sum_{x'',y''} I(x+x'',y+y'')$$

where *w* is the width of the image and h is the height of the image.

A thresholding operation and edge detection is then used to extract the edges of the road. Figure 80 shows the edge map and the corresponding edge of the road (white lines). At this stage, the algorithm seeks to obtain the pixel location for the two road edges. Note that due to the mounting of the camera on the vehicle and the relatively low resolution of the camera (100x240), the measurement location for seeing both the left and right sides of the road is fairly far down the road, as indicated by the red search boxes in Figure 80. These areas limit the search of the road edges to a certain threshold from the last road edge estimate.

The figure below shows an edge map which shows the outline of the road. Two squares are present which represent the search region for the edge outline. Also, a green line is present which represents the road width constraint along with a line showing the threshold for the road width constraint.

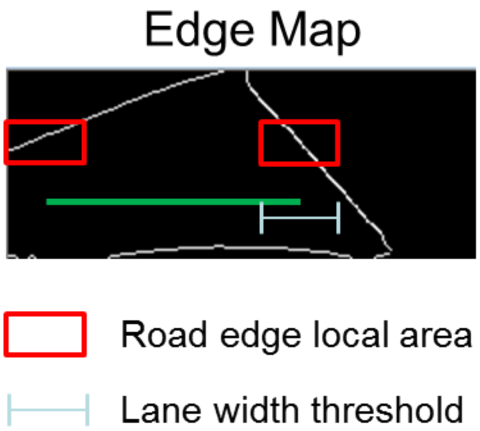


Figure 80: Edge Map with local area and lane width threshold constraints

In addition to this search region, one other constraint is used to reduce erroneous road edge detections. By employing the width of the road in the image (green) along with some threshold to account for a variable road width (cyan), the search locations for the road are further constrained so that both road edges are within a certain distance. Finally, a two state Kalman filter is used to filter erroneous detections (like that from shadows) where each state is the horizontal pixel location in the image of the road edge.

5.4.2 Camera Road Edge Detection Experimental Results

Tests were conducted on several rural roads around Auburn University's campus. Both day and night data was taken on the same stretches of road. Table 6 shows the numerical results of the tests. The width of the road was used for the test metric and was assumed to be constant. Note that, like the lidar tests which were conducted over the same stretches of road, these results are conservative and are subject to changes in road width due to intersections, driveways, and other road features that change the width of the road.

Road	Mean Absolute Error (m)	Standard Deviation (m)	% Detection
Co Rd 084 (Day)	0.0706	0.2191	77.99
Co Rd 084 (Night)	0.072	0.278	81.51
Co Rd 188 (Day)	0.1043	0.1638	86.3
Co Rd 188 (Night)	0.1384	0.2253	82.73
MSJ Rd (Day)	0.1704	0.2972	79.29
MSJ Rd (Night)	0.0667	0.1574	80.6

Table 6: Camera Road Edge Detection Results

Figure 81 shows two images from the tests. The green dot represents a detected road edge, while the red dot represents a non-detected road edge for that frame. Note that in Table 6, the Due to the use of a Kalman filter, estimates for the location of the road edge are available and often coincide with the location of the road edge due to the small dynamics inherit in driving on a straight road. The black circles are the estimate from the Kalman filter. The road itself is highlighted in red, and the small blue box is the template.

The left image of Figure 81 represents a virtual ideal road, where no obstructions are present on the road and the road edges are detected. In the right image, shadows obstruct the road and interfere with the template matching algorithm and the right road edge detection. Other situations, such as a rapidly transitioning road surface and similar road/off-road features, can create problems for the road detection as well.

The figure below shows two scenes of the view from a vehicle down a road. The first scene has two green dots along the edge of the road signifying the road edge detection. The second scene has a green dot and a red dot, signifying that the road edge was detected on one side but not the other. A shadow from the treeline is present over the red dot.

Figure 81: Road Edge Detection for Ideal (Left) and Shadowed (Right) Image

6 Auburn University – Integrated Positioning System (IPS)

Auburn University’s contribution consists of the integrated positioning system (IPS) for combining all of the information from the subsystems into a robust positioning solution.

6.1 Subsystem Integration

The IPS combines data from seven sensor systems, four subsystems, and a map database to provide ubiquitous positioning in GPS denied or degraded environments. A typical production or near production sensor suite including GPS, lidar, camera, stereo camera, inertial measurement units (IMU), wheel encoders, and DSRC radio provide inputs to either subsystems or the IPS system directly. The subsystems are shown in the figure below and include the road fingerprinting PSU system, the visual odometry SRI system, the lane departure AU system, and the DRSC Kapsch system.

The figure contains six pictures arranged in a semi-circle around a central block labeled fusion algorithm. Each of the six pictures represents one of the subsystems used in the IPS. Starting at the lower left and move clockwise the pictures show: camera LDW lateral positions, DSRC based ranging, limited GPS satellites, SRI visual odometry, PSU road fingerprinting, and lidar LDW lateral positions. Arrows connect each picture to the fusion block with the tale at the picture and the tip at the fusion block. An out arrow from the fusion block points to the IPS output: position, velocity, and attitude.

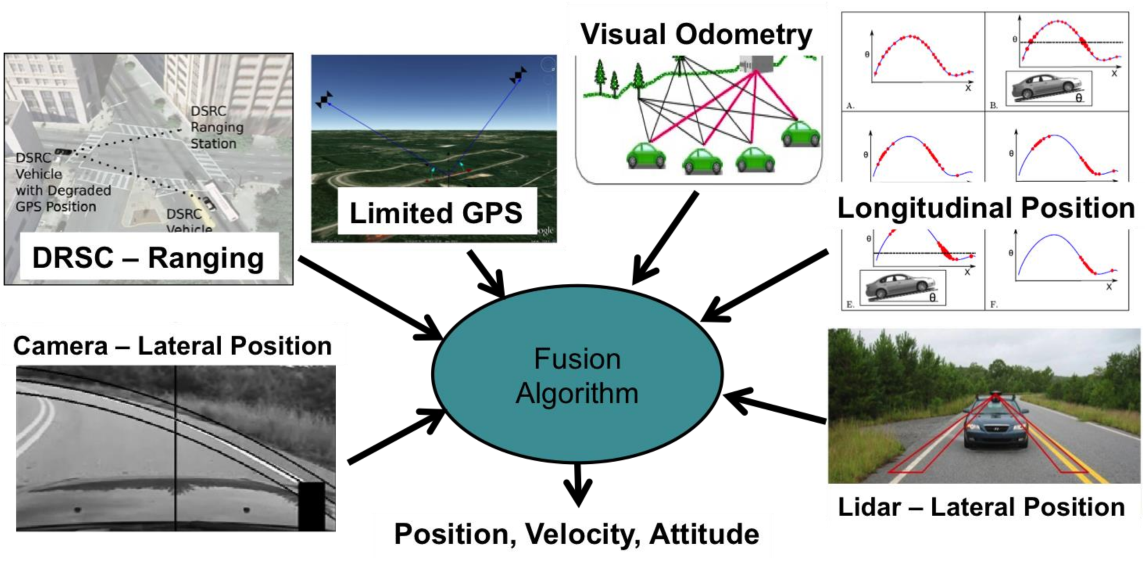
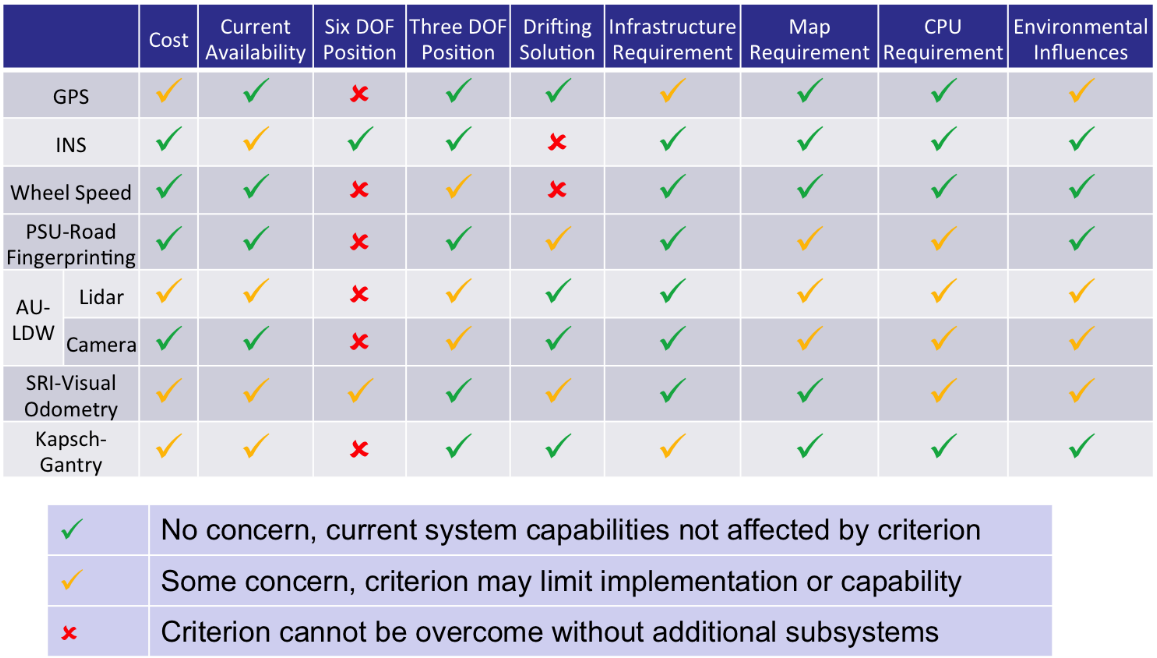


Figure 82: Visual representation of the Integrated Positioning System.

The expectation is that the combination of these systems and sensors will yield a common navigation solution that is greater than the sum of the parts. The table below illustrates the combined functionality of these systems. The capabilities matrix divides each sensor or subsystem into nine distinct criteria used to evaluate each. As seen in Figure 83, no one solution meets the highest standards for all nine criteria. The complete system combines these systems into one navigation solution, leveraging the strength of each component to provide the desired output.

The figure shows a table of IPS subsystems and the relative capabilities of each. The IPS subsystems are listed down the first column on the left in the following order: GPS, INS, wheel speed, PSU road fingerprinting, AU-LDW lidar, AU-LDW camera, SRI visual odometry, Kapsh DSRC. The capabilities criteria are listed across the first row in the following order: cost, current availability, six DOF, three DOF, drifting solution, infrastructure requirements, map requirements, cpu requirements, and environmental influences. Each cell contains either a green check mark (meaning no concern, current capabilities not effect by criteria), a yellow check mark (meaning some concern, criteria may limit implementation), or a red "x" mark (criteria cannot be overcome without additional subsystems).



		Cost	Current Availability	Six DOF Position	Three DOF Position	Drifting Solution	Infrastructure Requirement	Map Requirement	CPU Requirement	Environmental Influences
GPS		✔	✓	✗	✓	✓	✔	✓	✓	✔
INS		✓	✔	✓	✓	✗	✓	✓	✓	✓
Wheel Speed		✓	✓	✗	✔	✗	✓	✓	✓	✓
PSU-Road Fingerprinting		✓	✓	✗	✓	✔	✓	✔	✔	✓
AU-LDW	Lidar	✔	✔	✗	✔	✓	✓	✔	✔	✔
	Camera	✓	✓	✗	✔	✓	✓	✔	✔	✔
SRI-Visual Odometry		✔	✔	✔	✓	✔	✓	✓	✔	✔
Kapsch-Gantry		✔	✔	✗	✓	✓	✔	✓	✓	✓

✓	No concern, current system capabilities not affected by criterion
✔	Some concern, criterion may limit implementation or capability
✗	Criterion cannot be overcome without additional subsystems

Figure 83: Capabilities matrix showing the relative effectiveness of the subsystems of the IPS

Each of these subsystems and sensors provides data suitable for refining an estimate of one or all of the vital navigational states of a dynamic vehicle. The navigation states include the three dimensional position, three-dimensional velocity, and three dimensional attitude of the vehicle. In practice it may be satisfactory to estimate a subset of these states for a given set of sensors, such as a planar (two dimensional position and velocity with one dimension for attitude) estimation routine when a reduced order IMU is available. This is discussed in more detail in the navigation algorithms section.

The subsystem and sensor data is fused into a comprehensive navigation solution in an extended Kalman filter wherein the IMU is the foundation and the additional sensors and subsystems provide data to correct the errors of the inertial navigation solution. This is seen in the figure below. The navigation states of the vehicle are propagated forward in time using the accelerations and angular velocities from the IMU as inputs. The prediction of the vehicles pose is then corrected with the latest data from any of the available subsystems. The final output is the position, velocity, attitude, and an accurate time estimate provided by GPS.

The figure shows the components of the IPS in a block diagram. The sensor blocks are shown in red. The software subsystems (e.g PSU road fingerprinting) are shown in blue. The Kalman filter element are shown in green, and the outputs are shown in orange. Starting from the left the INS sensor feeds the INS navigation processor software block which feeds the Kalman filter state and covariance propagation step block. The propagation step block then feeds the Kalman filter state and covariance correction block along with the additional software subsystems. The output of the correction block is the system output: position, velocity, attitude and time.

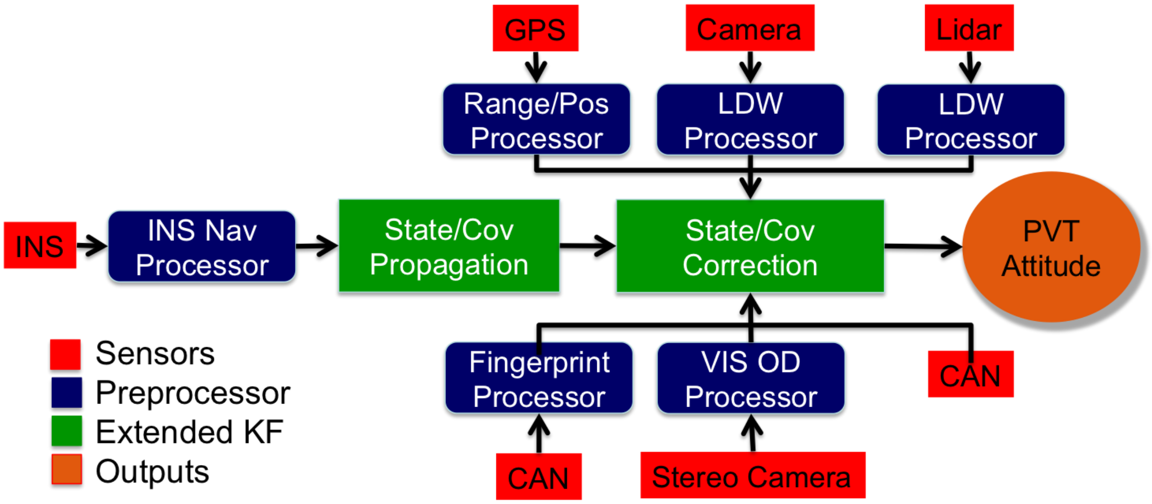


Figure 84: Block diagram of the extended Kalman filter implementation of the IPS showing inputs and outputs of subsystems and filter components.

For clarity, the components in the block diagram above are broken down by the inputs and outputs of that sensor or subsystem. The results are compiled in the table below. Notice the redundancy in the outputs and consider that failure or limited availability of any one subsystem does not cripple the navigation solution. It is important to note also that the IMU provides the basis for propagation of the vehicle states in time and therefore in the absence of IMU data alterations must be made to the filter architectures. A simple approach would be to assume the acceleration of the vehicle is negligible and uses a kinematic model to propagate the navigation states. This section provides an introduction to the integration strategy for the IPS. The step-by-step navigation algorithm is discussed in the next section.

The table shows the components of the IPS system. There are six components listed down the first column of the table. There are three column headings listed across the first row including: subsystem, inputs, and outputs. The inputs and outputs for each component are listed in the appropriate location in the table.

Table 7: The component of the IPS system and the input and output for each

Subsystems	Inputs	Outputs
INS navigation processor	Body frame accelerations and angular rates	Navigation frame accelerations and angular rates (bias corrected)
GPS processor	RF signals for SV	Range/Range Rates Positions/Velocities
Camera LDW processor	Raw image	Lateral lane position
Lidar LDW processor	Distance and reflectivity	Lateral lane position
Fingerprint processor	Pitch rate and wheel speed	Navigation frame position
Visual Odometry processor	Raw image from two cameras, internal IMU, and GPS positions	Navigation frame position

6.2 Navigation Algorithm Description

The navigation algorithm is an extended Kalman filter implementation. Three sensor combinations were examined in this work. The first combined a reduced inertial sensor system (RISS), L1 frequency GPS pseudorange and pseudorange rate, and vehicle wheel speed. The second system added L2 frequency GPS measurements. The final system used the six degree of freedom IMU instead of the reduced IMU and also added lateral lane position updates from the camera and lidar. Each implementation is discussed in detail in this section.

6.2.1 RISS

Single frequency GPS measurements were integrated with two accelerometers, a single gyroscope, and vehicle speed measurements to form the production grade navigation system for this study. The sensitive axes of the accelerometers were aligned with the longitudinal and lateral axis of the vehicle, and the sensitive axis of gyroscope was aligned with the vertical axis of the vehicle.

The states of the filter are: the position of the vehicle in geodetic coordinates (latitude (λ), longitude (ϕ), and altitude (h), the velocity in a local tangent plane (north (V_n), east (V_e), and down (V_d)), the heading (ψ) of the vehicle relative to north in the local tangent plane, two accelerometer bias states (b_{ax}, b_{ay}), one gyroscope bias state (b_{gz}), GPS receiver clock bias (Δt) and drift ($\dot{\Delta t}$), and a tire slip state (s). Kinematic relationships were used to formulate the dynamic equations for the propagation of the vehicle position states. Note that R_λ and R_φ are the meridian radius and normal radius respectively.

$$\dot{\lambda} = \frac{V_n}{R_\lambda + h}$$
$$\dot{\phi} = \frac{V_e}{\left(R_\phi + h\right)\cos(\lambda)}$$
$$\dot{h} = -V_d$$

The dynamics of the velocities were approximated by the accelerometer measurements (f_{ax} , f_{ay}), the expected measurement error, and the assumption that the vertical velocity was driven by zero mean white noise.

$$\dot{V_n} = (f_{ax} - b_{ax})\cos(\psi) - \left(f_{ay} - b_{ay}\right)\sin(\psi)$$
$$\dot{V_e} = (f_{ax} - b_{ax})\sin(\psi) + \left(f_{ay} - b_{ay}\right)\cos(\psi)$$
$$\dot{V_d} = 0$$

The heading dynamics were defined relative to the angular velocity measured by the gyroscope ω_{gz}, the measurement error b_{gz}, and the dynamics introduced by the dynamics of the Earth ω_e.

$$\dot{\psi} = \omega_{gz} - b_{gz} - \omega_e \sin(\lambda)$$

IMU biases were assumed to be first order Gauss Markov processes, as is the tire slip state. For additional information on these dynamics, see (Farrel et. al.) and (Bevly et. al.). The clock bias dynamics are driven by the drift term and the clock drift is assumed to be driven by zero mean white noise.

The Kalman filter covariance matrix was propagated using the linearized error dynamics derived from the dynamics described above (Groves). Note that in this formulation the vehicle roll and pitch were assumed to be negligible as is the lever arm between the IMU and GPS antenna.

The INS navigation solution was updated using both GPS and wheel speed measurements. GPS pseudorange and pseudorange rate measurements were used in this formulation. The INS solution was used to predict the pseudorange and pseudorange rate for each visible satellite at each GPS measurement. The pseudorange prediction model was as follows:

$$\hat{\rho} = \sqrt{(x_s - x)^2 + (y_s - y)^2 + (z_s - z)^2} + c\Delta t$$

where x_s, y_s, and z_s are the Earth Centered Earth fixed (ECEF) coordinates of the satellite, and x, y, and z are the estimates of the vehicle position in the ECEF frame. The receiver clock error is labeled $c\Delta t$. The pseudorange rate prediction is given by:

$$\hat{\dot{\rho}} = \frac{(V_{xs} - V_x)(x_s - x) + \left(V_{ys} - V_y\right)(y_s - y) + (V_{zs} - V_z)(z_s - z)}{\sqrt{(x_s - x)^2 + (y_s - y)^2 + (z_s - z)^2}} + c\Delta\dot{t}$$

where V_{xs}, V_{ys}, and V_{zs} are the velocities of the satellite in the ECEF frame, and V_x, V_y, and V_z are the estimated vehicle velocities in the ECEF frame.

The pseudorange and pseudorange rate predictions were compared to the measurements to calculate the navigation solution correction. Using the linearized geometry matrix, derived by computing the partial derivative of the pseudorange and pseudorange rate models with respect to the state vector, the correction was mapped into the state domain and applied to the navigation solution, and the error covariance matrix was updated.

An additional state update was performed using the encoder-provided wheel position measurement in terms of pulses per revolution. This measurement was converted to wheel speed using the wheel radius and the number of pulses per revolution. The two rear wheel speeds were then averaged to form one observable. This value was then compared to the wheel speed predicted by the INS solution. The model for the wheel speed prediction contains the estimated north and east velocities and the estimated tire slip.

$$\hat{v}_{ws} = (V_n^2 + V_e^2)^2(1 + s)$$

The correction to the position solution is computed by calculating the difference between the predicted and measured wheel speed and mapping that error into the state domain using the geometry matrix. Note also that the stationary condition (i.e., the measured wheel speed is zero) allows for improved estimates of the inertial biases. This is called a zero velocity update, and is further described in (Wang et. al.).

6.2.2 GPS/INS

Unlike with the reduced inertial navigation algorithm, when the full six degree of freedom IMU was used, the total attitude (i.e., roll, pitch, and heading) was estimated. Also, rather than navigating in the local tangent plane (used to minimize complexity of mapping the IMU to the navigation frame), the ECEF frame was selected as the navigation frame (minimizing the complexity of mapping the GPS to the navigation frame).

The change of navigation frames impacts the dynamic equations used in the propagation step of the Kalman filter. A rotation matrix was maintained to map accelerometer measurements from the vehicle body frame to the navigation frame. The dynamics of this rotation matrix, C_b^e were given by:

$$\dot{C_b^e} = C_b^e \Omega_{ib} - \Omega_{ie} C_b^e$$

where Ω_{ib} is the measured angular velocity of the body frame relative to an inertial frame and Ω_{ie} is the rotation rate of the Earth. Note that the gyroscope biases must be removed.

The velocity dynamics are now written in vector form as a function of the three axes accelerometer measurements accounting for the effects of gravity on the measured accelerations.

$$\dot{V}^e = C_b^e\left(f_{a_b} - b_{a_b}\right) + g_e$$

Here, f_{a_b} are the three axes accelerometer measurements, b_{a_b} are the accelerometer biases, and g_e is the local acceleration due to gravity.

Since the velocity and position are now expressed in a common navigation frame, the position dynamic equations are straightforward with P^e equal to the *x, y,* and *z* components of the position of the vehicle expressed in the ECEF frame.

$$\dot{P}^e = V^e$$

Processing the GPS pseudorange and pseudorange rate measurements follows the same procedure previously described for the RISS implementation. Note that the NovAtel GPS receiver provided measurements from both the L1 and L2 frequency carrier. This typically results in two times the number of observables at each GPS measurement epoch. The wheel speed measurement update was also performed according to the previous formulation.

6.2.3 Full Vision Integration

The full vision integration system uses six degree of freedom inertial measurements, dual frequency NovAtel GPS measurements, two estimates of lateral position within the lane – one provided by the lidar and another by the camera, and a map of the lane to estimate the pose of the vehicle. The estimation procedure followed the form described for the GPS/INS algorithm with the addition of a new update step when lateral position measurement were provided by either vision-based sensor. The lane detection and positioning methods for each vision system are described briefly below. A description of the procedure for integrating the measurements in the Kalman filter follows.

6.2.3.1 Vision Measurement Update

The described camera and lidar algorithms provide *measurements* of the lateral position of the vehicle relative to the center of the lane (Figure 85).

The figure shows a representation of a two lane road. The outer lane marking are shown are solid lines, and the middle lane marking is shown as a dashed line. Four dots are used to represent points in a map database for the right lane. The variable $\hat{r}$ is used to represent the lateral distance between the current estimate of the position of the vehicle and the line defined by the points from the map database. The variable $\tilde{r}$ is used to represent the measurement of the lateral deviation of the vehicle from the center of the lane from either the camera or lidar LDW.

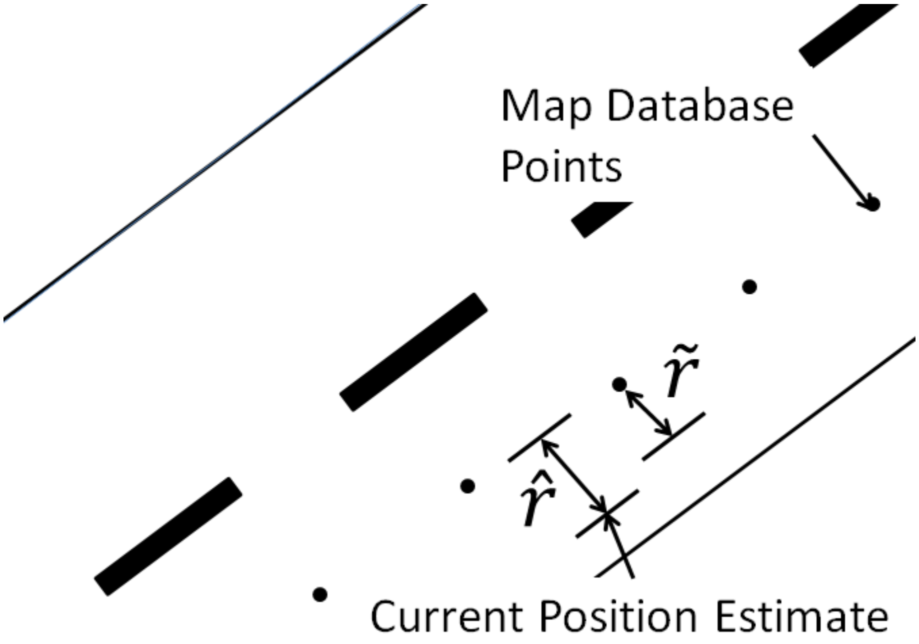


Figure 85: Visual representation of calculating the lateral position in the lane based on the current estimate of the position of the vehicle.

A *prediction* of the lateral position of the vehicle within lane is calculated using the current estimate of the vehicle position and its two closest lane map points in the direction of travel. If the estimated position is inaccurate, incorrect lane coordinates may be selected in the map. This is a limitation of this approach. The distance of the vehicle is calculated normal to the line defined by the two closest points in the north and east directions using the following equations:

$$\mu = \frac{(p_n - m_{2_n})(m_{1_n} - m_{2_n}) + (p_e - m_{1_e})(m_{1_e} - m_{2_e})}{(m_{1_n} - m_{2_n})^2 + (m_{1_e} - m_{2_e})^2}$$

$$n_n = m_{1_n} + \mu(m_{2_n} - m_{1_n})$$

$$n_e = m_{1_e} + \mu(m_{2_e} - m_{1_e})$$

$$\hat{r} = \sqrt{(p_n - n_n)^2 + (p_e - n_e)^2}$$

where m_1 and m_2 are the closest map points and $\hat{r}$ is the calculated lateral distance.

The predicted lateral distance, $\hat{r}$, is compared to the camera- or lidar-measured distance, $\tilde{r}$, to calculate correction error. The camera and lidar measurements are applied independently when either algorithm reports a lateral position. The lateral error is then mapped into the state domain using the linearized geometry matrix. Lastly the state estimates and error covariance are updated.

6.2 Hardware Integration

An Infiniti G35 sedan is equipped with the required hardware to support the navigation system complete with each subsystem described above. Sensors are mounted in location to maximize the effective range (camera, stereo camera, and lidar on the roof), to provide satisfactory performance (IMU near center of gravity), or as required for operation (wheel encoders at the wheel). The locations of the sensors and computing equipment are shown below.

Preliminary work showed the importance of vision sensing equipment locations and Phase 1 results provided a means for an easier transition into Year 2. The roof was selected as the mounting location for all vision based sensing equipment. In addition to vision based sensing equipment, the GPS antenna must be mounted on the roof of the vehicle. In an attempt to minimize the effect of the lever arm between the IMU and GPS antenna, care was taken to mount the antenna vertically above the IMU. Additional GPS antennas are seen in the figures below, but only the primary (middle) antenna is used in the IPS. There is also a radio antenna in the diagram that is used in some testing as part of the reference solution (the system used as truth for comparative purposes). This antenna is not used in the IPS. The figures below show the setup for the hardware on the roof of the vehicle.

The figures below show a computerized model of the layout of various sensors on the roof of the vehicle. The LiDAR, antennas, and mounting bars are shown, along with their dimensions in inches.

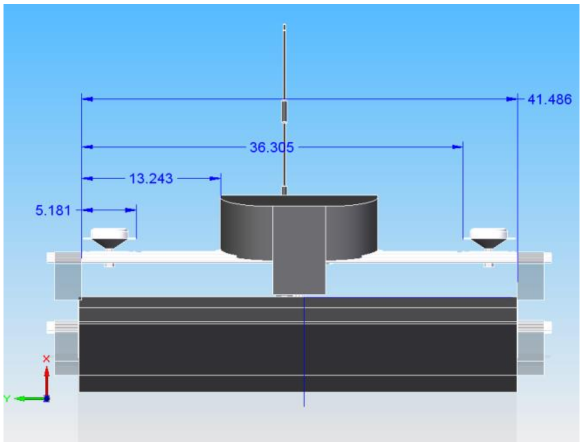


Figure 86: Detailed drawing of roof - front view

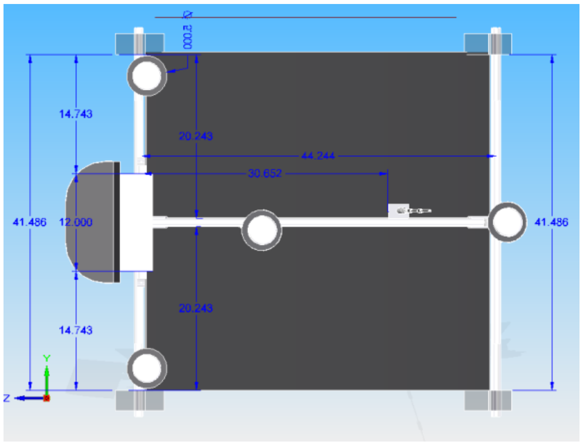


Figure 87: Detailed drawing of roof - top view

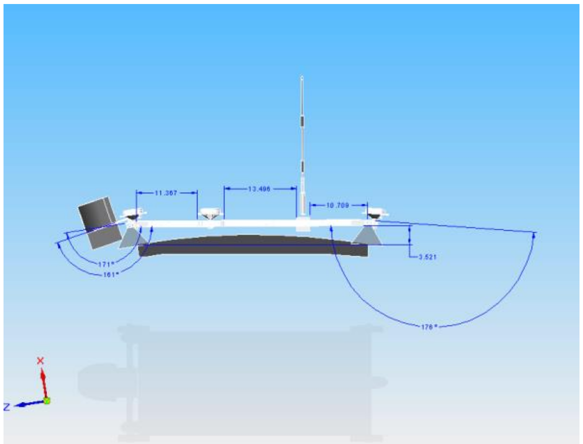


Figure 88: Detailed drawing of roof - side view

The IMU is mounted as close to the center of gravity of the vehicle as possible. As a result, the optimal location is under the armrest in the center console of vehicle. The location of the Crossbow IMU used in the IPS is shown in the figure below.

The figure is a picture of the center console of the Infiniti G35 sedan with Crossbow IMU400 mounted in the recess.

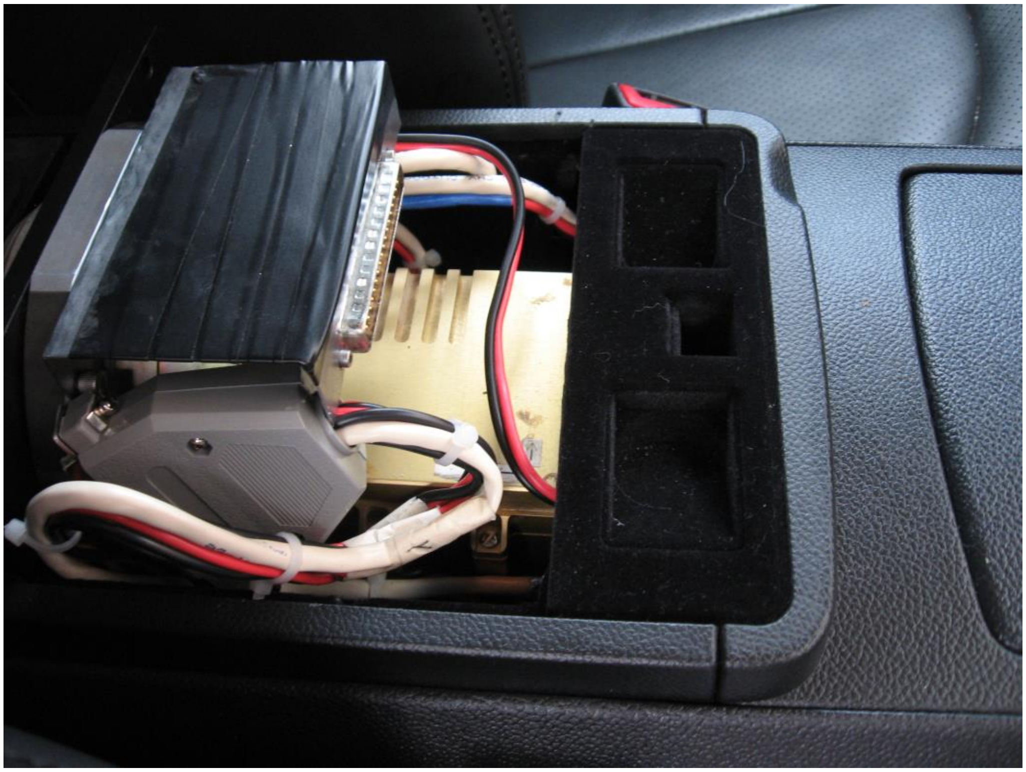


Figure 89: Picture of Crossbow IMU440 mounting location in the center console.

All of the sensors and subsystems report solutions to a central computer located in the trunk of the vehicle. The computer is mounted using a rack setup that may translate forward for access to the back of the device. The GPS receiver is also mounted on the rack in the trunk. Power for all sensors is distributed from a main junction located in the trunk, which is powered by the vehicle battery. A picture of the trunk in all its glory is provided below.

In addition to the computer in the trunk of the vehicle, laptops were employed for several subsystems. The visual odometry and road fingerprinting subsystems were implemented on dedicated laptops. The outputs from the systems were communicated to the central computer over TCP using Ethernet connections. One additional laptop was used for visualization of the IPS output. This laptop also communicated via TCP with the central computer to obtain the current navigation solution. The solution could then be visualized in real time within the vehicle as described in the next section.

The figure is a picture of the trunk of the test vehicle. It shows the rack mount used to hold the computer and GPS receiver. It also shows a portion of the power junction.

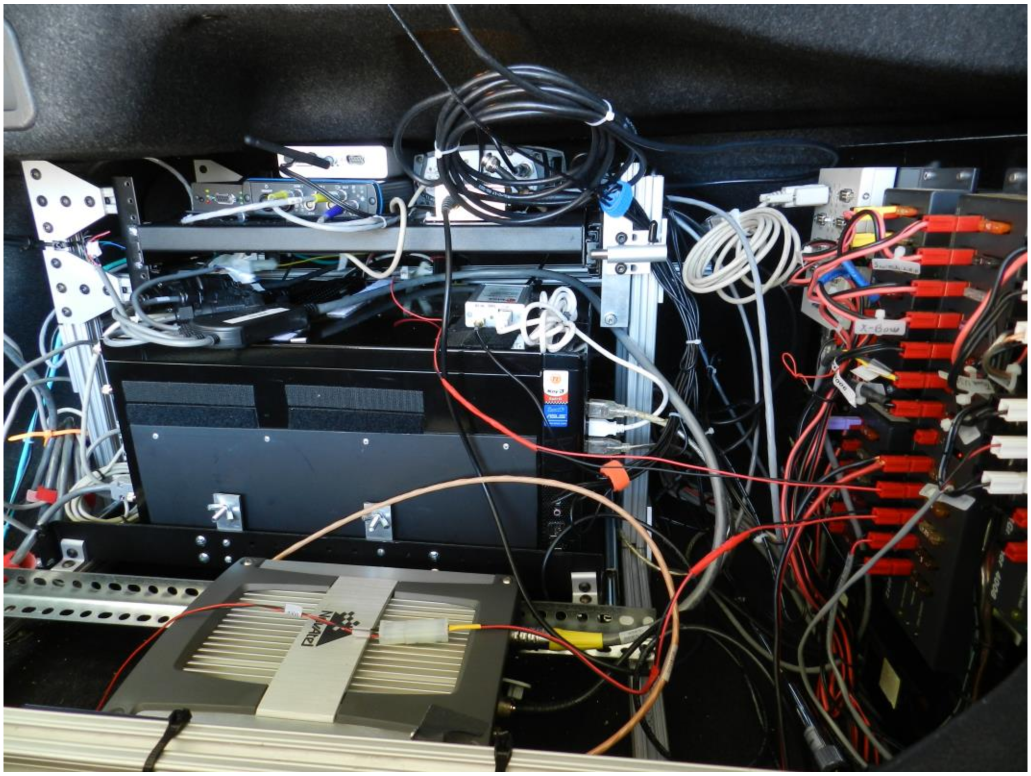


Figure 90: Picture of the trunk of the Infiniti G35 with the central computer, GPS receivers, and power components.

6.3 Software Integration

The navigation algorithms described above are implemented in software, which may operate in either real time or post process mode. The sensor and subsystem data are reported to a central database, and the data may be recorded for post process or processed on receipt for real time operation. The central database is maintained using a software package called the Mission Oriented Operating System (MOOS).

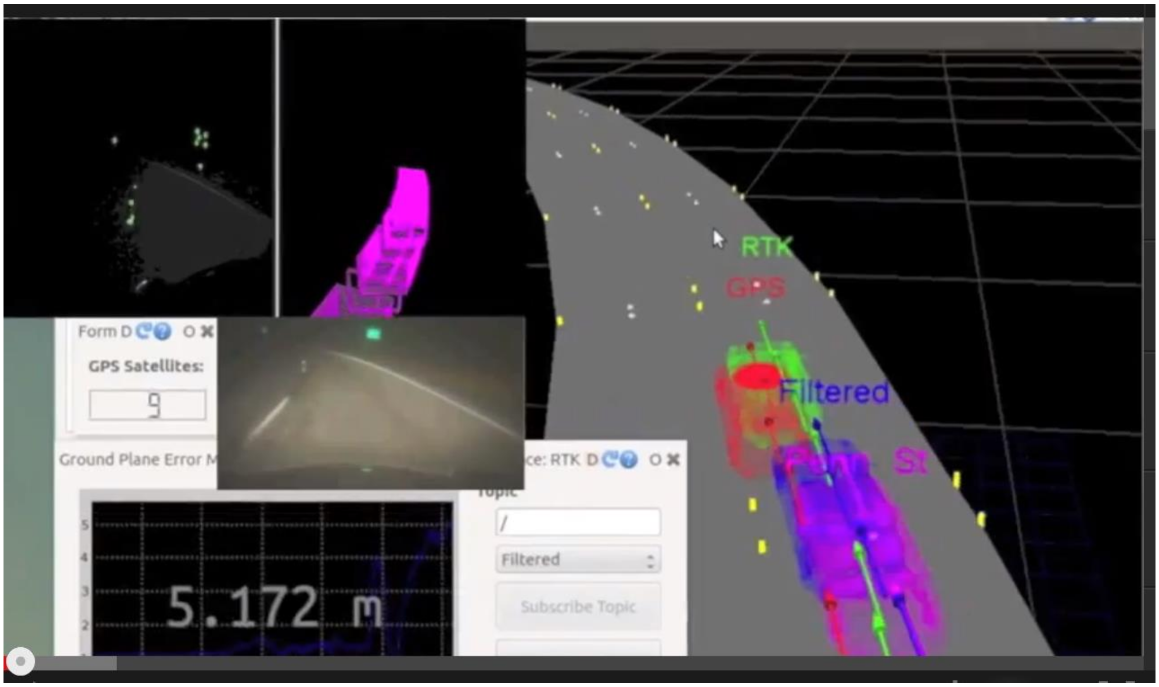
6.3.1 Mission Oriented Operating System

The MOOS was developed at Massachusetts Institute of Technology and is used primarily as software architecture for unmanned water vehicles (Newman). The software supports the central database described above, as well as some data playback tools that allow for like real time testing. In playback mode, data enters the database as it would real time allowing the user to mimic real time operation in the lab. The central database also allows multiple applications to access the same data at the same time. Each application monitors the database for changes to variables of interest, and then uses the data as prescribed. Additional MOOS tools include a graphical interface for viewing the current data in the database and data logging software for writing the database variables to a text file.

6.3.2 Real Time Visualization

In addition to the software tools available through the MOOS architecture, a graphical user interface (GUI) was developed for viewing systems outputs in real time. The GUI contained a window showing the current position of the vehicle on a surveyed map of the current surroundings (maps were developed for the National Center for Asphalt Technology and for the parking area around the Turner Fairbank facility of the FHWA). Another window shows the current horizontal error relate to a reference solution when one is available. The number of GPS satellites currently being tracked by the GPS receiver is also shown. An image of the GUI is shown in the figure below.

The figure shows a screenshot of the real time GUI for the IPS. Starting at the lower left corner and working clockwise, the image contains: a plot of horizontal error verse time for the last 10 seconds with the current error displayed numerical, a block containing the current number of visible satellites, the current camera image from the roof of the vehicle, two block from the SRI visual odometry system (discussed later), and a visual representation of the NCAT test track with four wireframe vehicle images with position derived from the standalone GPS position, RTK GPS position, the IPS position, and the PSU road fingerprinting position.


GPS Satellites: 9

Ground Plane Error M

5.172 m

Figure 91: Screenshot of the real time graphical user interface for the IPS system.

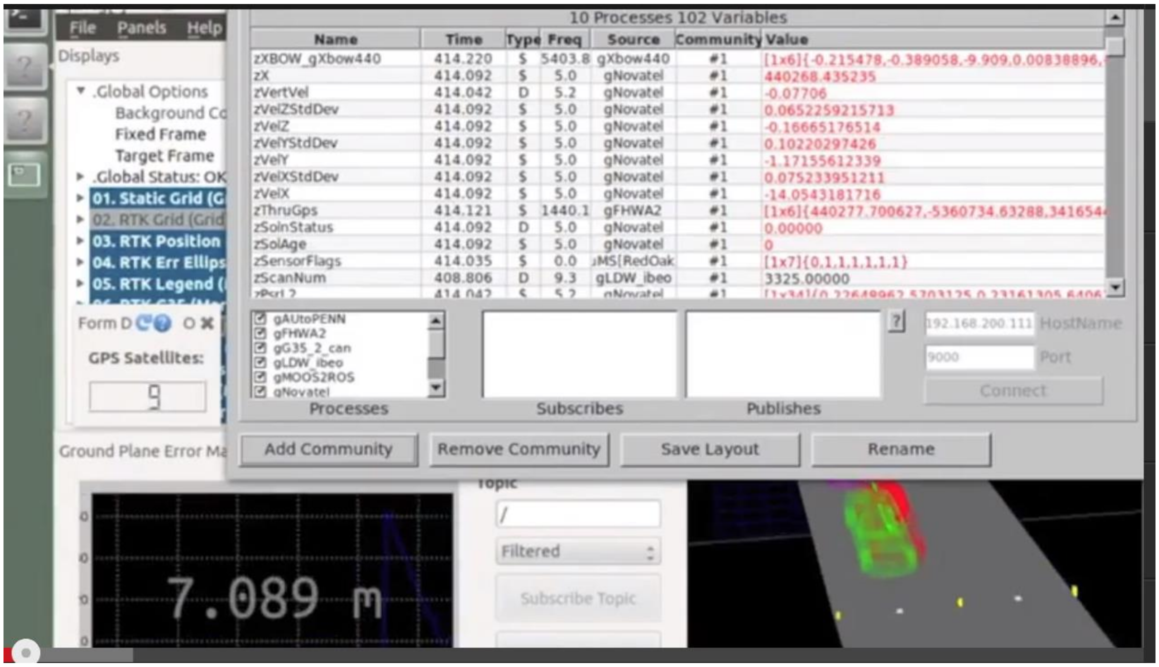
The SRI visual odometry system also has a real time visualization feature. It can be seen in the figure above as well as the next figure. Only the SRI portion of the real time visualization is shown in the figure below. In the SRI image, there are three windows: one simple shows the current camera image, one highlights the features that are being tracked in the image, and the final window shows a representation of the motion of the vehicle in the form of consecutive rectangles indicating relative pose over the last several epochs.

The figure shows a screenshot of the SRI graphical user interface. There are three images in the shoot showing from left to right: the features being tracked by the algorithm, a visual representation of the motion of the vehicle, and a raw camera image.

Figure 92: Screenshot of the SRI visual odometry real time graphical user interface.

The next image shows the GUI for the MOOS database. In the image, several variable names can be seen with the time of the data point, the data type, the frequency at which the variable is update, the sensor or subsystem from which the data came, and the current value of the variable.

The figure shows a screenshot of the MOOS database viewing tool. The database viewer is tabular with the variable names in the first column followed by time, data type, frequency, source, and value.



10 Processes 102 Variables

Name	Time	Type	Freq	Source	Community	Value
zXBOW_gXbow440	414.220	$	5403.8	gXbow440	#1	[1x6]{-0.215478,-0.389058,-9.909,0.00838896,
zX	414.092	$	5.0	gNovatel	#1	440268.435235
zVertVel	414.042	D	5.2	gNovatel	#1	-0.07706
zVelZStdDev	414.092	$	5.0	gNovatel	#1	0.0652259215713
zVelZ	414.092	$	5.0	gNovatel	#1	-0.16665176514
zVelYStdDev	414.092	$	5.0	gNovatel	#1	0.10220297426
zVelY	414.092	$	5.0	gNovatel	#1	-1.17155612339
zVelXStdDev	414.092	$	5.0	gNovatel	#1	0.075233951211
zVelX	414.092	$	5.0	gNovatel	#1	-14.0543181716
zThruGps	414.121	$	1440.1	gFHWA2	#1	[1x6]{440277.700627,-5360734.63288,341654
zSolnStatus	414.092	D	5.0	gNovatel	#1	0.00000
zSolAge	414.092	$	5.0	gNovatel	#1	0
zSensorFlags	414.035	$	0.0	MS[RedOak	#1	[1x7]{0,1,1,1,1,1,1}
zScanNum	408.806	D	9.3	gLDW_ibeo	#1	3325.00000

- ☑ gAUtoPENN
- ☑ gFHWA2
- ☑ gG35_2_can
- ☑ gLDW_ibeo
- ☑ gMOOS2ROS
- ☑ gNovatel

HostName: 192.168.200.111
Port: 9000

GPS Satellites: 9

Figure 93: Screenshot of the MOOS database viewing tool used to modify the IPS operation in real time.

One feature of the MOOS GUI that was exploited in the work was an option that allows a user to change a variable in the database by manipulating it in the GUI. Examine the GUI carefully and find the variable named “zSensorFlags”. It can be seen that the variable is a string of values taking numbers one or zero. This variable allows the user to modify the sensor suite used by the IPS. Each one or zero is associated with one of the sensors or subsystems used by the IPS. By click on the variable name, the user can change any of the values to a one (meaning use this sensor in the solution) or to a zero (meaning do not use this sensor in the solution). There are seven values corresponding respectfully to: IMU, GPS, camera LDW, lidar LDW, fingerprinting PSU, visual odometry SRI, and DSRC Kapsch. There are additional variables for re-initializing the filter and for controlling the GPS outputs; the user can select between using single or dual frequency measurements or can impose an artificial limit on the number of visible satellites. This functionality allowed for substantial real time testing in a variety of operating conditions.

6.4 Testing Results

As previously stated all testing was performed using an Infiniti G35 sedan. The hardware configuration for all test results was as described in the hardware integration section. Results that follow are divided into three testing scenarios: Detroit testing, NCAT testing, and Turner

Fairbank testing. A brief description of each test scenario is provided along with results from each test. For those test where at reference system was present and a truth solution could be calculated, errors in the positioning solution from the IPS are provided.

6.4.1 Detroit Testing

6.4.1.1 Test Procedure

Two Novatel GPS receivers were used in the test. One receiver provided inputs to the IPS; the other was used in combination with a HG1700 IMU to provide a reference position to be used as truth. Each receiver was connected to the same GPS antenna on the roof of the vehicle. Additionally, an external wheel odometer was mounted on the passenger-side rear wheel to aid the reference system. Testing occurred in Detroit, Michigan. Data was captured following a 46-mile route through the city center and surrounding suburbs to mimic driving experienced by a typical driver. In an attempt to capture the full range of GPS satellite geometry variability, the testing occurred over three days.

6.4.1.2 Test Route Selection

The test route was designed with two main aims: 1) to cover a variety of environment types that have characteristic effects on positioning performance, and 2) to be representative of typical U.S. driving in the proportioning of environment type selections. The environmental features considered important here for their influence on GNSS-based positioning accuracy and for their commonality are trees, tree canopies, mountains, overpasses, buildings, urban canyons, and tunnels. This is partially modeled after the table in (US Department of Transportation) on page 529. The test route here ensures that the Open and Urban environment types are covered.

Data from the FHWA on U.S. road use in terms of total miles traveled in each road-class category was used as the best available statistical data to indicate environment proportioning typically encountered in the U.S, even though it does not actually classify roads according to sky-visibility environments, but rather according to road size and volume. The selection of the test route was then guided by the goal of having the roads comprising the route closely match (within ±5%) the urban road-use class proportioning in the found by FHWA. The resulting 46-mile test route is shown in the figure below.

The figure shows the test route through suburban and downtown Detroit. There is a larger image of the complete route with smaller images of sections of the route to highlight the obstructions to GPS signals. Starting from the lower left and working clockwise the smaller images show: heavy foliage in suburban areas, a section of suburban Detroit where heavy foliage is present, an open freeway section, and a closer look at the Downtown section.

Figure 94: Detroit test route derived using the FHWA urban road use standards.

6.4.1.3 Test Timing

Due to satellite geometry variability over time, the test timing is a critical component of the test plan. Since the satellite configuration, as seen by the receiver on the ground, repeats every 24 hours, for repeatable results, it is desirable to have the testing span 24 hours. For procedural convenience, the 24 hour period is divided into 3 shifts. Each shift is on a different day (thus allowing a large break from test driving), and spans 10 hours, with 2-hour overlaps with respect to time of day, for a total desired span of 24 hours.

6.4.1.4 Sensors

Some sensors used here (Table 8) mimic those commonly found on modern-day passenger vehicles. Higher end sensors are also included to acknowledge the expected improvement in sensor capabilities for future production vehicles. Sensors are tested in several combinations.

Table 8: Sensor combinations for Detroit Testing

Type	**Model**	**Description**	**Rate (Hz)**
Production or Near-Production Grade			
GPS	NovAtel Propak V3	Scientific GPS receiver, but position, velocity, and L1 data used only	5
Wheel Speed	From in-vehicle CAN	Production sensor for vehicle wheel speed	50

	network		
RISS	Crossbow IMU 440, reduced	Reduced data set used for mimicking near-production grade	100
Camera	Logitech Quickcam 9000	Web-camera, production-like quality	10
Beyond Production Grade			
GPS	NovAtel Propak V3	Scientific GPS receiver gives: position, velocity, L1 and L2 pseudorange and pseudorange rate	5
IMU	Crossbow IMU 440, full	Six degree of freedom IMU, Micro-Electro-Mechanical (MEMS)	100
Lidar	Ibeo Alasca XT	marketed as automotive-grade but not typically available on production vehicles	10
Reference System			
GPS	NovAtel SPAN-SE	Scientific GPS receiver	5
IMU	Honeywell HG1700 AG58	Military-grade IMU	100
External encoder	Peiseler MT1000/r	Mounted on passenger-side rear wheel to count wheel rotations	Speed dependent
DGPS	Differential GPS solution was calculated in post-processing by differencing on-vehicle GPS data with data from local GPS reference stations to mitigate atmospheric effects. Typically, this processing limits global position errors to decimeter level.		

Results are presented in the form of a plot of satellite availability, tables showing error statistics, and plots of navigation solutions on road maps. In the figure below, the number of visible satellites and the resulting horizontal dilution of precision (HDOP) are shown for the test route over three test runs. As seen, the number of visible satellites varies from twelve to one. Accordingly, the HDOP ranges from typical values (less than 2 (Hegarty et. al.)) to extreme values when less than four satellites are visible. The GPS positioning accuracy is compromised due to the reduced GPS satellite visibility and increased multipath reflections in portions of the test route. This information is not used directly in the position estimate but provides insight to variations in the achievable accuracy. Note that the HDOP and an approximation for the one-sigma user-equivalent range error (UERE) can be used to calculate the expected one-sigma horizontal error bound. For a HDOP value of 2 and an UERE of 5.1 meters the expected one-sigma horizontal error bound is 10.2 meters (Parkinson et. al.).

The figure contains two plots with three sets of data on each. The top plot shows the number of visible satellites verse time in seconds. The bottom plot shows the horizontal dilution of precision plotted verse time in seconds. There is a legend for the time of the data runs with the blue line corresponding to a 6:00 am run, the red line corresponding a 9 am data run, and the green line corresponding to an 11 am data run.

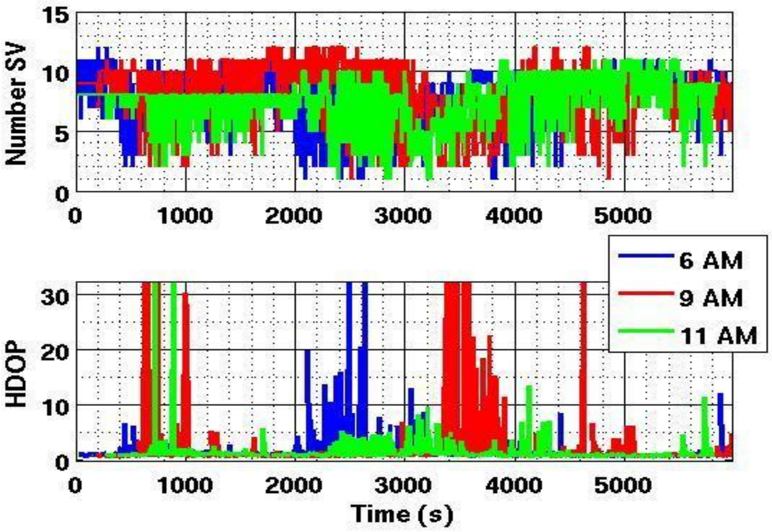


Figure 95: Number of visible satellites and horizontal dilution of position for three test run.

The results of the Novatel GPS-only solution and the GPS/INS solution combining the Novatel receiver data with the full six degree of freedom IMU440 are shown. The Novatel only solution is labeled Propak_'run number' and the GPS/INS solution is labeled GPS_INS_'run number.' Table 9 includes the mean horizontal errors and the percentage of errors less than 1.5 and 5 meters for both the standalone Novatel solution and the combined solution. The combined navigation solution exhibits improvement in all three categories with nearly 6 percentage points of improvement in the overall 5 meter error analysis.

Table 9: Errors for GPS only and GPS/INS Navigation

Run Number	Mean Absolute Horizontal Error	% < 1.5 m	% < 5 m
Propak_R1	2.3	44.1	91.4
GPS_INS_R1	2.3	41.5	94.0
Propak_R3	2.9	46.7	88.8
GPS_INS_R3	2.0	59.8	95.5
Propak_R5	2.3	38.1	86.8
GPS_INS_R5	1.9	46.5	93.2
Propak Overall	**2.6**	**55.9**	**88.4**
GPS_INS Overall	**2.2**	**63.8**	**94.3**

Error statistics are shown in Table 10 as a function of the local environment. The table gives the percentage of errors less than 1.5 meters across all test runs. As seen in the table, GPS/INS integration provided the highest improvement, by percentage in the moderate and tree environments. There was also a slight improvement in the canyon areas.

Table 10: Error Statistics Arranged by Environment Type

Device	Environment				
	Open	Ok	Trees	Canyon	All
Propak All Runs	67	49	33	14	42
GPS_INS All Runs	74	56	40	18	49
Proportion %	4	54	15	8	100

The full vision system accuracy is limited by the ability to correctly identify the closest map coordinates. A map of the test route was developed using Google Earth and analysis was performed using residual errors in GPS pseudorange predictions to verify lane selection. This method did not show statistical improvement over the GPS/INS results presented earlier. One can see qualitative improvements in the positioning estimates in the downtown area in Figure 96 when using vision based estimation updates. It is anticipated that future system improvements (such as road edge detection) will provide additional information for improved identification of the lane of travel. In order to identify the potential for improvements when the correct lane is selected, the algorithm was run assuming the lane of travel was known. Modest improvement in the percentage of errors less than 1.5 meters were seen in run 1 (45.1 %) and run 3 (61.8%). It should also be noted that the accuracy of the full vision integrated system is limited by the accuracy of the map.

The figure shows two images side by side with a legend with four variables. The red line corresponds to standalone GPS. The green line corresponds to GPS/INS. The yellow line corresponds to the full IPS solution, and the blue line corresponds to the reference solution. The image on the left shows the four position solutions as the vehicle passes under an overpass. The image on the right shows the four position solutions as the vehicle maneuvers through the downtown area.

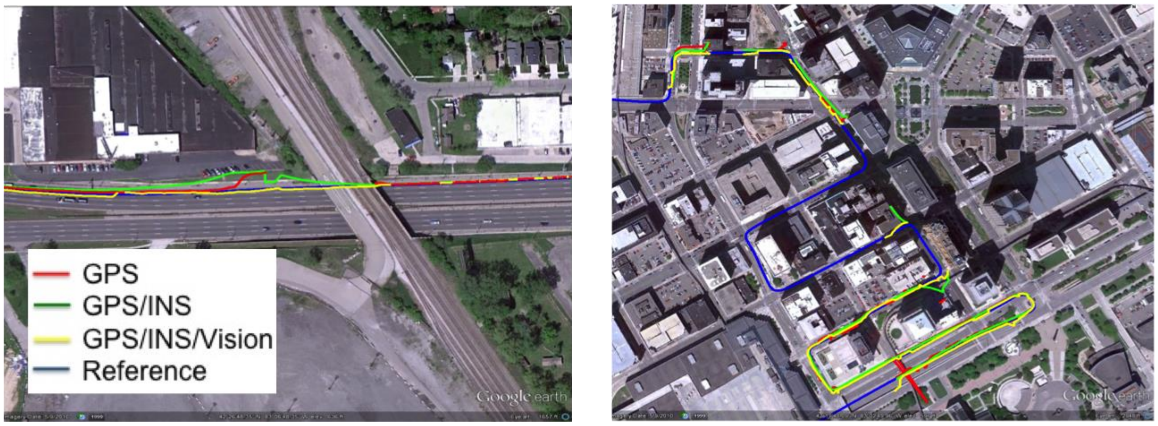


Figure 96: System performance in the downtown area of Detroit using the IPS system compared to traditional GPS and GPS/INS positioning methods.

6.4.2 NCAT Testing

Additional testing of the IPS was performed at the National Center for Asphalt Technology (NCAT) at Auburn University. The NCAT maintains a 1.7 oval test track with two lanes of travel that can be seen in the figure below. The track has banked turns and is equipped with a GPS base station that broadcast correction for RTK GPS. RTK GPS is capable of providing position estimates with errors on the centimeter level. This capability provides an accurate reference solution for comparison when testing positioning algorithm at NCAT. In addition to RTK GPS for vehicle positioning, the test track lane markings have been mapped using RTK GPS. This map of the test track is much more accurate than the map used in the Detroit testing described above.

The figure shows an aerial image of the NCAT facility including the test track and surrounding areas.

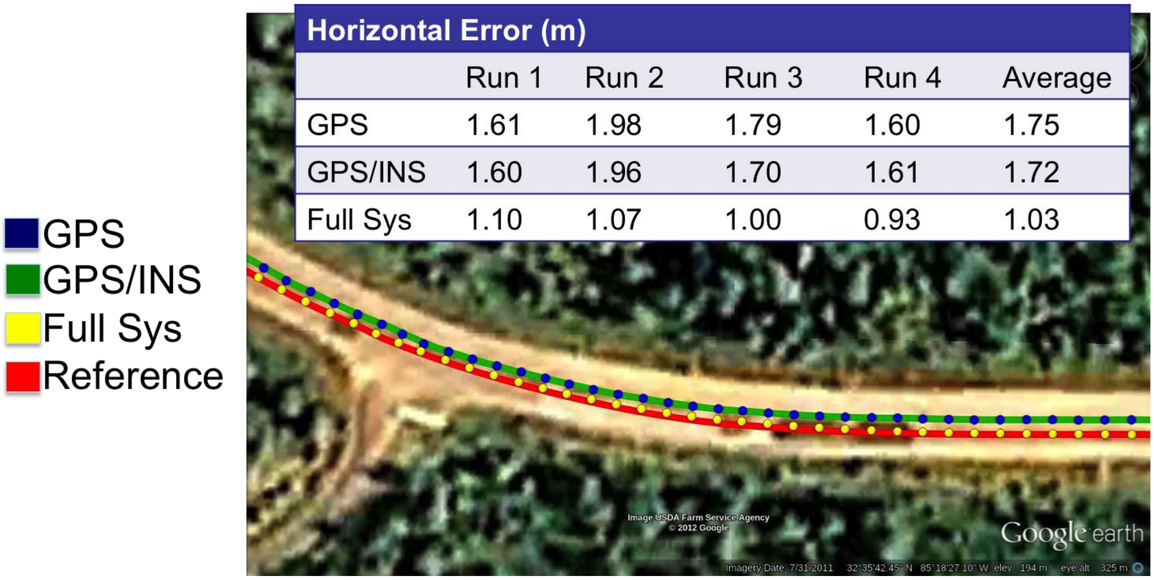
Figure 97: The test track at the National Center for Asphalt Technology.

The hardware setup for the IPS was not changed from the setup used in the Detroit testing. The reference system was modified. The tactical grade IMU and external wheel encoder where not used when calculating the reference solution. Only RTK GPS was used as the reference for the following results.

The IPS was tested in real time and in post process at the test track and the following results have been compiled. The results shown in the following figures and tables are the result of post processed data. Recording the data while running all the components of the IPS in real time was not practical as the processing load approached the computation limits of the central computer. Based on visual inspection of the real time GUI (including the real time error plotting), these results are consistent with the results seen when operating the full IPS system in real time.

In Figure 98, the mean horizontal error in the full IPS system is shown with the error in the standalone GPS, and the GPS/INS position solutions. The mean errors are shown for four test runs as well as the mean error across all runs. Clearly, the IPS outperforms the GPS and GPS/INS solutions consistently. On average the IPS mean horizontal error is three quarters of one meter less than the GPS and GPS/INS solutions- approximately a 40 percent improvement. This result is much better than the results seen in the Detroit testing due to the increased accuracy in the lane maps at the test track. There is also a visual representation of the positioning solutions seen in the figure. The full IPS is seen to track the reference solution with much higher accuracy than the GPS and GPS/INS solution through one turn on the southeast corner of the track. This corner typically has the worst GPS coverage on the track due to elevated terrain and dense tree cover on either side of the track.

The figure contains two images; one is a table with the stats described the text and the other is the visual representation of the positioning solutions. The position solutions are shown on a Google Earth image for context. The GPS, GPS/INS, full IPS, and reference solutions are color coded blue, green, yellow, and red respectively.



Horizontal Error (m)

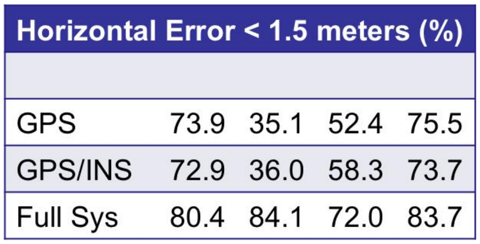
	Run 1	Run 2	Run 3	Run 4	Average
GPS	1.61	1.98	1.79	1.60	1.75
GPS/INS	1.60	1.96	1.70	1.61	1.72
Full Sys	1.10	1.07	1.00	0.93	1.03

Figure 98: Horizontal error results from NCAT for the full IPS, GPS, and GPS/INS positioning systems.

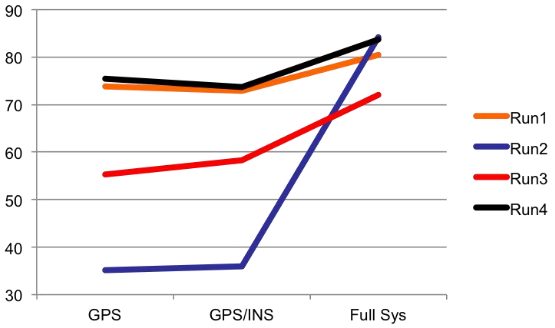
Additional numerical and visual results are seen in the next figure. The horizontal error is presented in terms of lane level accuracy, with the assumption that errors less than 1.5 meters constitute lane level accuracy. The table in the figure shows that both GPS and GPS/INS positioning provide lane level accuracy approximately 60 percent of time on average. That number is significantly higher for the full IPS position solution – 80 percent of the time on average. The visual image of the GPS and GPS/INS position solution highlights an issue that arises when combining GPS and INS data in a filter. In the image the standalone GPS position solution snaps towards the reference solution when new and more accurate becomes available. However, the GPS/INS solution gracefully approaches the reference solution as more of the improved data become available. This graceful change is due to trusting the INS information (which only measures the change in the position) and due to the inherent averaging effect of the filter. These are the same properties that make GPS/INS combination very successful over short GPS outages. When looking at the percentage of time, the error is less than 1.5 meters.

This effect will cause the GPS only solution to outperform the GPS/INS solution in some scenarios. This is seen in the first and fourth data set in the table.

The figure shows three images: a table of statistics showing the percentage of errors less than 1.5 meters for the GPS, GPS/INS, and full IPS over four data sets, the same data is shown graphical in the second image to show the consistent improvement of the IPS, and the last is a visual representation of the position solution of the GPS and GPS/INS. The solutions are shown with the reference solution with a Google Earth image of the track as a background.



Horizontal Error < 1.5 meters (%)				
GPS	73.9	35.1	52.4	75.5
GPS/INS	72.9	36.0	58.3	73.7
Full Sys	80.4	84.1	72.0	83.7



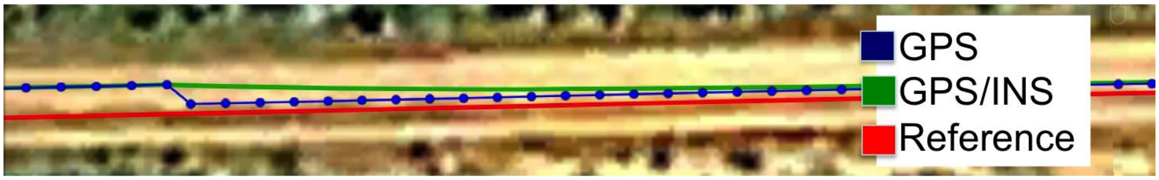


Figure 99: Horizontal Error on Track from GPS, GPS/INS, and Truth (Reference)

6.4.3 Turner Fairbank Testing

The IPS was also tested at the Turner-Fairbank facility in Virginia. An accurate reference solution was not available for comparison after efforts to calculate an RTK GPS solution failed to yield consistent results. Accordingly, only visual qualitative results are shown here. As in the previously mentioned testing, the Infiniti G35 sedan was used as the test vehicle. The hardware setup was not changed. Several test runs were completed around the parking area of the facility.

The complete test route can be seen in Figure 100. The vehicle began the route on the north side of the Turner-Fairbank building heading east. It then circled back to the west approaching the guard shack. Finally, the vehicle turned back to the east and returned to the start of the test route.

The figure shows an aerial image of the area around the Turner Fairbank facility provided by Google Earth. The area encompasses the parking area and buildings of the facility. On the image the test route is visible in the blue.

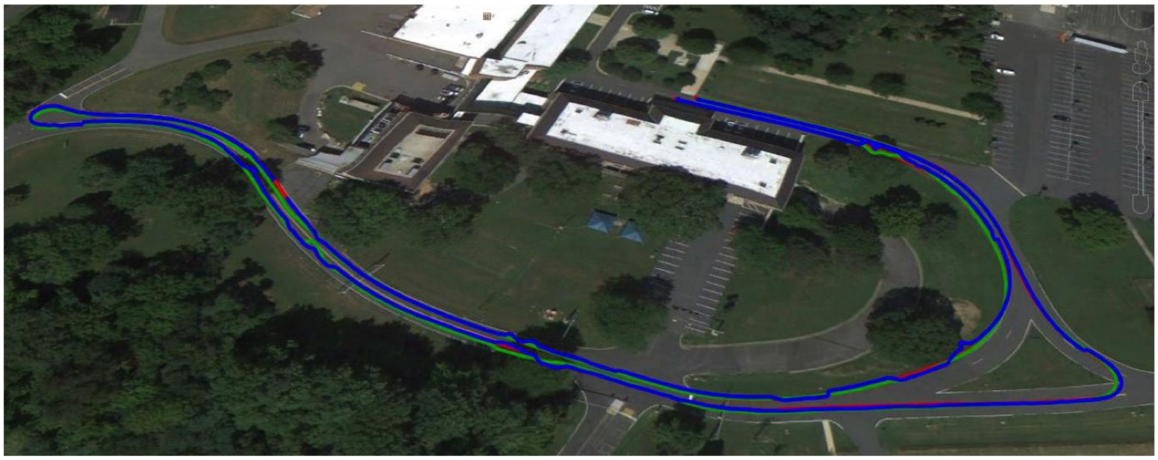


Figure 100: Turner Fairbank test site and test route.

In the next figure, the number of visible satellites is seen for one of the four data sets collected at Turner-Fairbank. Clearly, the number of satellites varies greatly over the test run. The intersection area of the test (as seen in Figure 100) had the minimum GPS coverage. The tree cover in that area largely prevent line of sight to the southern sky. As a result only three satellites were tracked through that portion of the route.

The figure is a graph showing the number of visible satellites plotted verse time in seconds. The range of visible satellites has a minimum of 3 and a maximum of 8. The graph shows approximately 150 seconds of data.

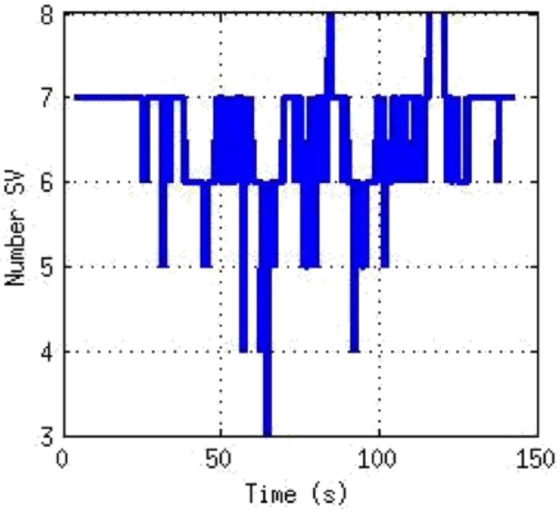


Figure 101: The number of visible satellites varies greatly over the test route from a minimum of 3 satellites to a maximum of 8.

Finally, in Figure 102, the position solution as calculated by the IPS is shown with precise maps of the test route. In the graph on the left, the curved portion on the eastern boarder of the route is seen. The IPS position solution in this route does not appear to be consistently within the lane. However, the vehicle remained in the lane during testing. The results in the image on the right are slightly better. The IPS position solution does remain in the lane as long as lane markings are available for detection by the AU LDW system. Note that the PSU road fingerprinting subsystem was not in use during this test. Since the road "signature" had not been surveyed the PSU system was not available. The graph on the right shows the results of the IPS near the intersection where the GPS visibility was low. There the IPS solution was highly dependent on the lateral lane position estimate from the camera and lidar LDW systems. That is the reason for the improvement in the solution when the lane markings are available.

There are two graphs shown in the figure. Both graphs show the lane marking of the test route with black line denoting the outer lane markings and blue lines denoting the middle lane marking. The IPS position solution is shown in green. The graph on the left shows a close up view of the results for the curved portion of the test route. The graph on the right shows a close up view of one of the straight sections of the test route near the mock intersection.

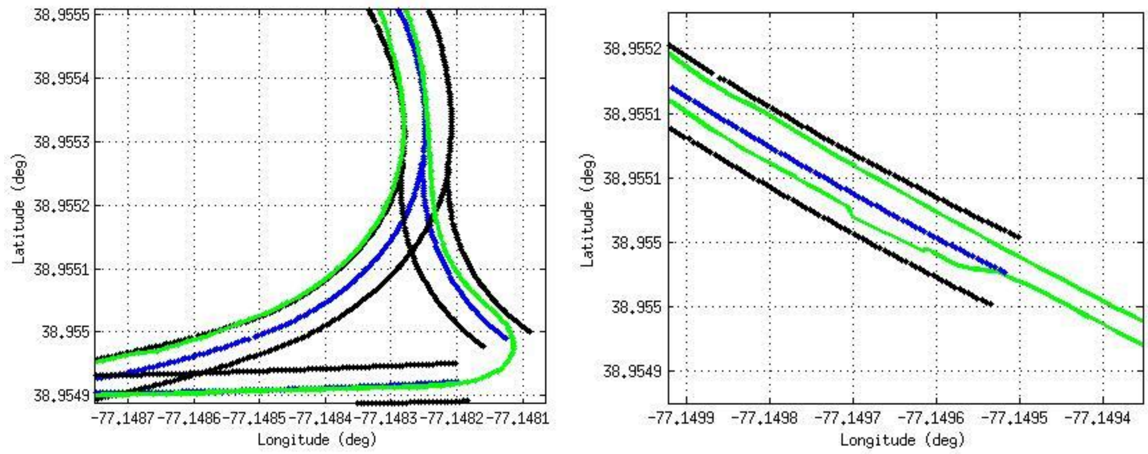


Figure 102: The full IPS worked well in the straight section of the test route where lane markings were available. The IPS did not produce the same level of accuracy in the curved sections of the route

Conclusions

This project used several positioning subsystems with complementary positioning strengths in an integrated positioning system. Each positioning subsystem was capable of providing position solutions in certain environments, but not all environments. For example, some subsystems required maps, other environments required infrastructure, and others required clear line-of-sight to satellites. The integrated positioning system, however, provided positioning solutions in many environments in which one or more of the subsystems could fail. Testing was conducted at Auburn's NCAT track, on a course in Detroit representative of America's roadways, and at Turner-Fairbank.

Auburn's lane detection vision sensors consisted of two redundant lane detection systems which provided lateral distances to the lane markings with the aid of known lane marking positions. Kapsch's DSRC distance estimation system was incapable of providing lane level positioning, but a gantry system utilizing their commercial tolling system provided a global update when the vehicle passed under a gantry. SRI's visual odometry provided lane level positioning at the cost of computational power and potential occlusions in images. Finally, Penn State's road fingerprinting system provided global positioning with the aid of maps that related the pitch of the vehicle to the global location.

Bibliography

(1998). IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros. IEEE Std. 952-1997.

Allan, D. W. (1966). Statistics of Atomic Frequency Standards. Proceedings of the IEEE, 54(2), 221-230.

Analog Devices ADIS16367 Datasheet. (2011, March). Retrieved from Analog Devices website: http://www.analog.com

Bevly, D. M. (2004). Global Positioning System (GPS): A Low-Cost Vehicle Sensor for Correcting Inertial Sensor Errors on Ground Vehicles. Journal of Dynamical Systems, Measurement and Control, 126(2), 255-264.

Bevly, D. M., and S. Cobb. GNSS for Vehicle Control. Artech House, Boston, 2010.

Britt, J., “Lane tracking using multilayer laser sacnner to enhance vehicle navigation and safety,” Proc. 2009 International Techincal Meeting of Institute of Navigation, Anaheim, 2009.

Dean, A. J., Langelaan, J., & Brennan, S. (2009). Initializing an Unscented Kalman Filter using a Particle Filter. Proceedings of the Dynamic Systems and Control Conference.

Dean, A., & Brennan, S. (2009). Terrain-based Road Vehicle Localization on Multi-Lane Highways. Proceedings of the American Control Conference, (pp. 707-712). St. Louis, MO.

Dickmanns, E.D., and B. D. Mysliwetz. “Recursive 3-d road and relative ego-state recognition”. 14(2):199-213, 1992.

Dietmayer, K., N. Kapchen, K. Furstenberg, J. Kibbel, W. Justus, and R.Shulz, Advanced Microsystems for Automotive Applications 2005. Springer Berlin Heidelberg, 2005.

El-Sheimy, N., Hou, H., & Niu, X. (2008). Analysis and Modeling of Inertial Sensors Using Allan Variance. IEEE Transactions on Instrumentation and Measurement, 57(1), 140-149.

Farrel J. and Barth, M. The Global Positioning System and Inertial Navigation. McGraw Hill, 1999.

Feng, Y., W. Rong-ben, and Z Rong-hui, “ Based on digital image lane edge detection and tracking under structure environment for autonomous vehicle,” Proc. IEEE Int. Automation and Logistics Conf., 2007, pp. 1310-1314.

Gelb, A., Kasper, J. F., Nash, R. A., Price, C., & Sutherlan, A. (1974). Applied Optimal Estimation. (A. Gelb, Ed.)

Godha, S., & Cannon, M. E. (2005). Integration of DGPS with a Low Cost MEMS - Based Inertial Measurement Unit (IMU) for Land Vehicle Navigation Application. Proceedings of the ION GPS '05. Long Beach, CA, USA. (1)

Groves, P.D. GNSS, Inertial, and Multisensor Integrated Navigation Systems. Artech House, 2008.

Han, S. W. (2009). Using Allan Variance to determine the Calibration Model of Inertial Sensors for GPS/INS Integration. Proceedings of the 6th International Symposium on Mobile Mapping Technology. Sau Paulo, Brazil.

Hegarty, C. and Kaplan, E. D., (editors) “Understanding GPS: Principles and Applications, 2nd ed.,” Artech House, Norwood, MA, 2006.

HG 1700 Datasheet. (2011, March). Retrieved from Honeywell International website: http://www.honeywell.com

Iqbal U., A. F. Okou, and A. Noureldin, “An integrated reduced inertial sensor system—RISS/GPS for land vehicle,” in Proc. IEEE/ION PLANS, Monterey, CA, May 2008, pp. 912–922.

Jerath, K., & Brennan, S. (2011). GPS-Free Terrain-based Vehicle Tracking Performance as a Function of Inertial Sensor Characteristics. Proceedings of the Dynamics Systems and Control Conference. Arlington, VA.

Jung C. R., C.R. Kelber, "A lane departure warning system based on a linear-parabolic lane model," in: Proc. of IEEE Intelligent Vehicles Symposium, Parma, Italy, 2004, pp. 891–895.

Kibbel, J., W. Justus, and K. Furstenberg, “Lane estimation and departure warning using multilayer laserscanner,” in Proc. IEEE Conf. Intell. Transp. Syst., 2005, pp. 777–781.

Landmark 10 Datasheet. (2011, March). Retrieved from Gladiator Technolgies website: http://www.gladiatortechnologies.com

Landmark 30 Datasheet. (2011, March). Retrieved from Gladiator Technologies website: http://gladiatortechnologies.com

Li, H., F. Nashashibi, and G. Toulminet, “Localization for Intelligent Vehicle by Fusing Mono-camera , Low-cost GPS and Map Data,” in Proceedings of IEEE Annual Conference on Intelligent Transportation Systems, 2010, pp. 1657–1662.

Miller, I. and M. Campbell, “Particle filtering for map-aided localization in sparse gps environments,” in Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, 2008, pp. 1834–1841.

Newman P., “MOOS – A Mission Oriented Operating Suite,” MIT Department of Ocean Engineering, Tech. Rep. OE-2003-07,2003.

Ng, L., & Pines, D. J. (1996). Characterization of Ring Laser Gyro Performance using the Allan Variance Method. Journal of Guidance, 20(1).

North, E.; Georgy, J.; Tarbouchi, M.; Iqbal, U.; Noureldin, A.; , "Enhanced mobile robot outdoor localization using INS/GPS integration," Computer Engineering & Systems, 2009. ICCES 2009. International Conference on , vol., no., pp.127-132, 14-16 Dec. 2009.

Parkinson B.W. and Spilker J.J. The Global Positioning System: Theory and Applications Volume I. Aiaa, 1996.

Rose, C., “Vehicle lane position estimation with camera vision using bounded polynomial interpolated lines,” Proc. 2009 International Technical Meeting of Institute of Navigation, Anaheim, 2009.

U.S. Department of Transportation, Federal Highway Administration, “Annual Vehicle-Miles of Travel 1980-2007, By Functional System, National Summary (Table VM-202, summary for 2007),” Jan. 2009, http://www.fhwa.dot.gov/policyinformation/statistics/vm02_summary.cfm.

von Reyher, A., Joos, A., and H. Winner, “A lidar-based approach for near range lane detection,” in Proc. IEEE Intelligent Vehicles Symp, 2005, pp. 147-152.

Wan, E., & van der Merwe, R. (2000). The Unscented Kalman Filter for Nonlinear Estimation. IEEE Adaptive Systems for Signal Processing, Communication and Control Symposium, (pp. 153-158).

Wang, J. H., and Y. Gao, “The aiding of MEMS INS/GPS integration using artificial intelligence for land vehicle navigation,” IAENG Int. J. Comput. Science, vol. 33, no. 1, pp. 33-1–33-11, 2007

Williams, S., Dissanayake, G., & Durrant-Whyte, H. (2001). Towards terrain-aided navigation for underwater robotics. Advanced Robotics, 15(5), 533-549.

Xing, Z. G.-E. (2008). Modeling and bounding low cost inertial sensor errors. IEEE/ION Position, Location and Navigation Symposium, (pp. 1122 - 1132).

Zhao, Z., & Li, X. R. (2005). The Behavior of Model Probability in Multiple Model Algorithms. Proceedings of the Seventh International Conference on Information Fusion, (pp. 331-336).

Zhao, Y., "A Map Matching Algorithm in GPS-based Car Navigation System," Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on, vol. 1, pp. 77-80, 2007.

Appendix 1: Algorithm Implementation in Project

Algorithm Parameter Values Used

With reference to the description of the algorithm in Section 3, the various parameter values used in the implementation of the algorithm are listed below:

Parameter	Value
Number of states, N_s	1
α	1
β	2
Step size (Prediction step), u_k	10 meters (default)

$\boldsymbol{W_m}$	$\begin{bmatrix} (\alpha^2-1)/\alpha^2 \\ 1/2\alpha^2 \\ 1/2\alpha^2 \end{bmatrix}$
$\boldsymbol{W_c}$	$\begin{bmatrix} \frac{\alpha^2-1}{\alpha^2}+(1-\alpha^2+\beta) & 0 & 0 \\ 0 & 1/2\alpha^2 & 0 \\ 0 & 0 & 1/2\alpha^2 \end{bmatrix}$
Process noise, $\boldsymbol{Q}$	0.01 m^2
Measurement noise, $\boldsymbol{R}$	$(0.01\ u_k)^2$ m^2 (conservative)

The parameter values can be changed to optimize the performance of the algorithm, such as when a different step size (or sampling rate) was used based on the type of road being traveled over (Section 5, highway or secondary road).

Appendix 2: Test Protocol

Test Protocol for characterizing deterministic error sources (or calibration)

The deterministic error sources under consideration are the scale factor error and the bias error. To determine the magnitude of these error sources, data for the variable of interest (in this case, the pitch of the road) has to be obtained across multiple runs. Since the variable of interest is the pitch, the sensors must ideally be co-located and otherwise be positioned along the y-axis (lateral axis) of the vehicle.

(1) Power up the apparatus (tactical-grade integrated IMU/GPS system, and other commercial-grade IMUs under consideration)
  [NOTE: The tactical-grade integrated IMU/GPS system is necessary to establish the ground truth for position, velocity and attitude measurements.]
(2) Allow 5 minutes of warm-up time
(3) Begin driving the vehicle at approximately 30 mph and initiate data logging at a sampling rate of 100 Hz. Log position, velocity and attitude data (INSPVA) from the tactical-grade IMU/GPS system and from all the commercial-grade IMUs under consideration.
(4) Log data for 10 minutes on a highway.
  [**NOTE**: Significant variations in pitch are observed at the scale of about 500 meters, so a dataset comprising between 5-10 km (or 3-6 miles) should provide an adequate set of measurements for calculating deterministic sources of error, such as scale factor. The time it takes to cover 3-6 miles at speeds of 30-60 mph is approximately 6 minutes. For ease of data collection and to guard against any erroneous data, it is recommended that the data be collected for 10 minutes.]
(5) Stop data logging and power down the system.
(6) Wait for 10 minutes and repeat steps (1) to (5) three more times, driving at approximately 40, 50 and 60 mph over the exact same stretch of highway and in the same direction.
(7) Repeat the procedure at slow speeds (say 20, 25, 30 and 35 mph) on city roads.
  [**NOTE**: Collecting data across multiple road surfaces enables the characterization of the terrain, whose impact on localization accuracy needs to be studied.]
(8) Repeat the procedure at slow speeds (say 20, 25, 30 and 35 mph) on rural roads.

Test Protocol for characterizing stochastic error sources (or noise characterization)

The stochastic error sources can be characterized from a static test. The stochastic error sources include bias instability, angle random walk, rate random walk, rate ramp, quantization noise and sinusoidal noise. All or a subset of these error sources may be present depending on the type of sensor being investigated. The following steps may be performed for collecting the data:

(1) Place the sensing apparatus in a controlled environment or a closed lab at room temperature. It is desirable that extraneous sources of error such as temperature fluctuations be minimized. However, if the sensing apparatus utilizes an integrated IMU/GPS system, place the apparatus in a stationary vehicle.

(2) Power up the apparatus (both the tactical-grade IMU/GPS system, and all other commercial-grade IMUs under consideration).

(3) Allow 5 minutes of warm-up time.

(4) Initiate data logging for all IMUs. Log position, velocity and attitude data for 5.5 hours at a sampling rate of 100 Hz. **Do not move the sensors in this duration.**

[**NOTE 1**: The lowest frequency random error source in an IMU is the rate ramp error. Thus the total duration of the test is limited by the time needed to estimate this error source. The rate ramp usually varies on the time scale of hours. Assuming that the characteristic time for rate ramp error is 1 hour and that one is willing to accept a 33% percentage error in determining this error component, the number of samples needed to estimate the rate ramp error is calculated using the formula:

$$\sigma = \frac{1}{\sqrt{2\left(\frac{N}{K} - 1\right)}}$$

where σ is the percentage error, N is the total number of data points required, and K is the number of data points per cluster. In the limiting case, there must be at least 1 unique data point per cluster. In terms of time, this translates to K = 1 hr for estimating rate ramp error. Substituting σ = 0.33, one obtains N = 5.5 hours. Thus the test should be conducted for the duration of 5.5 hours. (To provide a sense of scale, the test duration to obtain a 5% percentage error would be approximately 200 hours)]

[**NOTE 2**: The highest frequency random error source in a digital IMU is the quantization error (due to the inherent digital nature of the sensor). Since the error source is due to the digital nature of the signal itself, it has no bearing on the sampling rate. The random error source with the next highest frequency is the angle random walk. Thus, the sampling rate may be determined by:

(a) the frequency of the angle random walk error (of the order of 10 Hz), or

(b) the bandwidth of the sensor itself (in case it is an analog device).

In order to satisfy the Nyquist Theorem, the sampling rate must be at least twice as much as the highest frequency being measured, and preferably three to five times as much. A sampling rate of 100 Hz is adequate to capture the characteristics of angle random walk error. Similarly, considering a commercial analog IMU, MotionPak II, which has a bandwidth of 30 HZ, a sampling rate of 100 Hz is adequate.]

(5) Power down the system.