The Pennsylvania State University

The Graduate School

College of Engineering

VEHICLE LOCALIZATION USING LOW-ACCURACY GPS, IMU AND MAP-AIDED VISION

A Dissertation in

Mechanical Engineering

by

Vishisht Gupta

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2009

The dissertation of Vishisht Gupta was reviewed and approved* by the following:

Sean N. Brennan
Assistant Professor of Mechanical and Nuclear Engineering
Dissertation Advisor, Chair of Committee

H. Joseph Sommer
Professor of Mechanical and Nuclear Engineering

Alok Sinha
Professor of Mechanical and Nuclear Engineering

Robert T. Collins
Associate Professor of Computer Science and Engineering

Jack W. Langelaan
Assistant Professor of Aerospace Engineering

Karen A. Thole
Professor of Mechanical Engineering
Department Head of Mechanical Engineering

*Signatures are on file in the Graduate School.

Abstract

This dissertation describes the development of vehicle state estimation methods using low-cost sensors, their implementation, and comparison with highly accurate vehicle state estimators available today. This research was motivated by the problem of navigating a vehicle on a highway, where it is desirable to closely measure the vehicle's state (absolute position and orientation, rotation rates etc.) to achieve electronic stability control, collision avoidance, driver alert systems for lane departure and ultimately autonomous navigation. The focus in this thesis is to develop low-cost methods for vehicle localization. Low-cost Commercial Off-the-Shelf (COTS) sensor systems have been used to this effect.

A framework is developed to combine measurements from Global Positioning System (GPS) and Inertial Measurement Unit (IMU). Performance of a low-cost GPS receiver operating in autonomous mode integrated with a MEMS based low-cost IMU is investigated. The error sources in GPS and INS systems are characterized to choose suitable stochastic models for the error sources and to identify parameters for these models. Vehicle velocity vector is used to improve the yaw angle estimate under low yaw angle observability conditions.

To obtain an independent direct measurement of the vehicle orientation, a novel method based on terrain-aided vision is developed. This method is based on matching images captured from an on-vehicle camera to a rendered representation of the surrounding terrain obtained from an on-board map database. United States Geographical Survey Digital Elevation Maps (DEMs) were used to create a 3D topology map of the geography surrounding the vehicle. The horizon lines seen in the captured video from the vehicle are compared to the horizon lines obtained from the rendered geography, allowing absolute comparisons between rendered and actual scene in roll, pitch and yaw.

Work on terrain-aided vision based orientation estimation has been extended to use near field features like road signs and road markers. Near field features allow the measurement of vehicle position in addition to vehicle orientation. A map-aided vision algorithm is presented which registers features in the rendered images with features in real images using gradient-based minimization of sum of squared intensities. To improve the convergence properties as well as convergence time of the vision algorithm, an IMU is used to predict the location and possible variability of features in the rendered representation defining a Region-Of-Interest (ROI).

A Kalman filter framework is used to fuse the measurements from an IMU and each of the position and orientation estimation methods mentioned above. Numerical simulations are done in each case to verify the correctness of the formulation. Finally, experiments are performed at the Pennsylvania Transportation Institute (PTI) test track facility to test the performance of each method against a highly accurate GPS/IMU system.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis describes the development of vehicle state estimation methods, their implementation and comparison with exemplary vehicle state estimators found in literature. This research was motivated by the problem of navigating a vehicle on a highway, where it is desirable to measure the vehicle's absolute position and orientation closely. The position measurement is desirable to achieve collision avoidance, driver alert systems for lane departure, and ultimately autonomous navigation. The orientation and orientation rates are useful to detect vehicle instability. The key states affecting stability - and hence safety - are roll angle and sideslip. Both of these states require the measurement of very small orientation angles.

Different localization sensors have their own strengths and weaknesses. Depending on the conditions of use, the measurements given by these sensors vary in quality e.g. availability of satellites for GPS and visibility conditions for vision based sensors. As a result, there is a need to integrate various vehicle localization sensors to get better performance in state estimation. Some of the goals of this thesis are as follows.

- To develop vehicle state estimation methods using various localization sensors including Global Positioning System (GPS), Inertial Measurement Unit (IMU) and vision. A 3-D representation of the features in the environment surrounding the vehicle is also used to help in vehicle state estimation.

- To develop efficient algorithms for these estimation methods.
- To experimentally validate the estimates against high accuracy vehicle state estimators currently available.

1.1 Motivation

Vehicle state estimation is an important area of vehicle dynamics which finds applications in vehicle chassis stability control, autonomous navigation and fault detection [91, 6, 31, 99]. Vehicle stability control requires accurate estimates of vehicle states (e.g. sideslip, yaw rate, roll rate, roll angle etc.) at high update rates so that appropriate control action can be taken at an early stage of instability. Autonomous navigation requires estimates of vehicle position and attitude with respect to a fixed coordinate system. Vehicle position and attitude determination methods face challenges in dealing with high-degrees of sensor uncertainty, in the interpretation of terrain and environment observations and to make optimal localization estimates from ambiguous sensor data. Finally fault detection and isolation requires redundancy in the vehicle state estimation. The use of model-based estimators is necessary to achieve analytical redundancy, whereas hardware redundancy can be achieved by using multiple sensors to measure the same state.

Various sensors available for vehicle state estimation include inertial sensors, odometric sensors, Global Positioning System (GPS), vision and terrain sensors. If the driver's inputs are measured, the vehicle model can also serve as an additional source of state estimates. Each of these sensors have their advantages and disadvantages. If the information from each of these sensors is combined using an estimator based on a particular criteria (e.g. minimum least squares error), better estimates of vehicle states can be obtained. When there are redundant or distributed sensors, measurements from these sensors can be fused together to give an optimal estimate of vehicle states. In addition, redundant information can be used by an algorithm to achieve safe and graceful degradation of performance when there are temporary or permanent errors in some of the vehicle sensors.

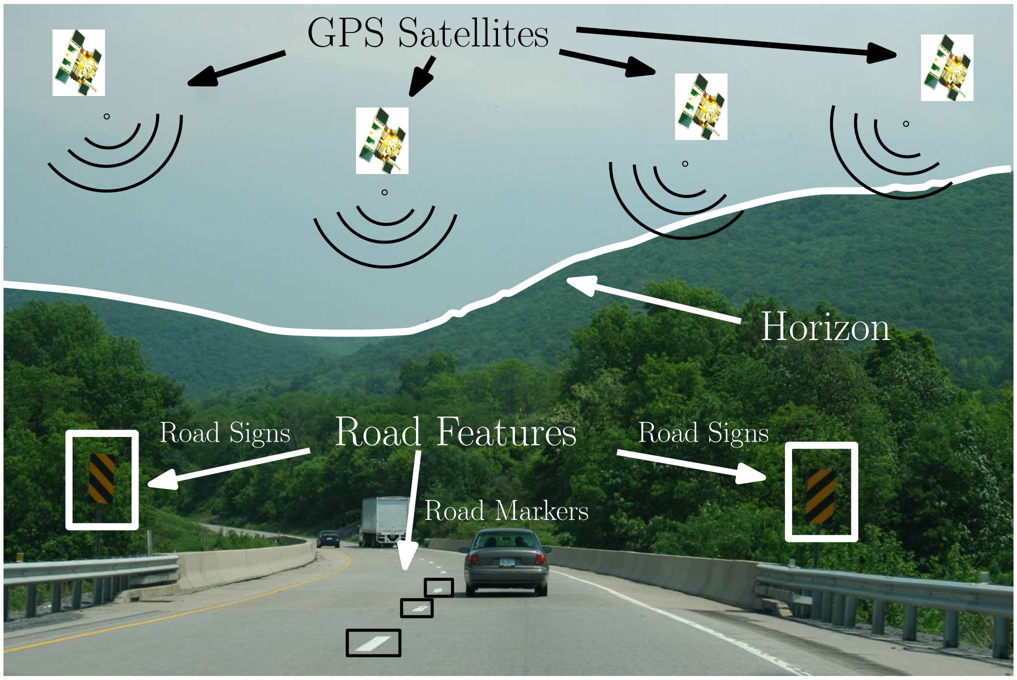


Figure 1.1. Vehicle Localization Along a Highway

This dissertation describes the development of localization techniques for a vehicle going along a highway using measurements from low-cost MEMS IMU, GPS and a monocular camera along with a map of visual features along the highway e.g. road features such as road-markers, road-signs etc. and terrain features e.g. the horizon as shown in figure 1.1.

1.2 Localization Methods

Vehicle state estimation techniques in unstructured outdoor environments can be broadly classified as shown in Figure 1.2. The figure shows the advantages and disadvantages of each sensor type as well as the most popular sensors of that type.

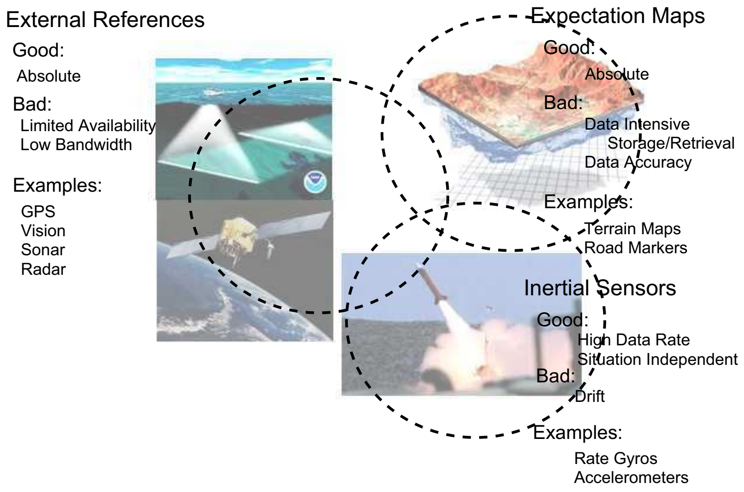


Figure 1.2. Vehicle State Estimation Methods

A brief description of each of these along with the advantages and disadvantages of using these sensors follows.

1.2.1 Inertial Sensors

Inertial sensors are commonly used in vehicle localization and vehicle stability control applications [9, 13, 61, 84]. These are passive sensors which do not rely on any external information to provide vehicle body rates. These sensors provide high data rates needed for vehicle stability control and autonomous navigation applications and are independent of outside vehicle situations. However, these inertial sensors do not directly detect position and orientation, but rather estimate these from rate information integrated over time. Because all signals have error, integration of any signal introduces drifts that grow unless external measurements of position and orientation, or assumptions about vehicle behavior, are applied.

1.2.2 Global Positioning System

GPS has become an indispensable aid in outdoor vehicle localization over the past few years. After "Selective Availability" was discontinued in 2000, a single GPS receiver became able to give a position accuracy of around 10m (rms) or better [77]. Methods have been devised to augment the standard GPS receiver to achieve greater accuracy e.g. supplying the receiver with differential corrections from a base station (Differential GPS) or by using Carrier-Phase Enhancement of GPS (real-time kinematic GPS (RTK)). Accuracies of 0.5m (Differential GPS) and 2cm (Real-Time Kinematic GPS) can be achieved using these methods. Update rates of 10-20 Hz can be obtained by using advanced GPS receivers. However, GPS systems can fail in locations where vehicles commonly drive, namely urban canyons, actual canyons, or forested road areas due to loss of line-of-sight to the satellites or due to multi-path errors.

1.2.3 Terrain-Aided Localization Sensors

If a pre-stored database of the terrain features is available, observed terrain features can be compared with the features in the stored database to localize the vehicle. This technique is called terrain-aided localization [71].

Among sensors needed to observe terrain features, vision systems are an excellent choice. The most obvious benefit is that visibility of key features within the surrounding scene can be assumed as a precondition for driving. Vision sensors readily identify features that impose stark geometric constraints on a world model, for example horizons, road edges and markers, buildings, etc. and many of these features are intentionally placed for driving localization, e.g. lane markers. Vision sensors have improved in terms of quality and data rate of the video stream and are more affordable today than ever before. Development in the field of computer vision has recently made tremendous improvements in faster image processing techniques and efficient algorithms. These improvements, when combined with the readily available processing power of modern computers, can now allow more information to be extracted from the vision stream at much higher rates compared to what was possible before. Finally, vision systems are finding increasing use in many production vehicles for lane detection or backup visibility assistance.

Terrain feature information and the technology needed to manage it is more available today than ever before. In the past decade, detailed terrain data has become publicly available in the form of Digital Elevation Models (DEM), topographical maps and 3D models of the cities, for example see the works of [30, 40] and those available from the

USGS website [20]. Edwards et al. [18] describe the generation of a high resolution, high accuracy, experimental digital terrain analysis data of a 12 km^2 test site for autonomous ground vehicle navigation. Technology for storing and retrieving such information is also undergoing revolutionary advances, motivated especially by portable and/or in-vehicle navigation aids, advanced 3D gaming, and high-speed retrieval and visualization of map databases e.g. Google Earth, Microsoft's Virtual Earth, TerraServer, NASA's WorldWind, etc.

Laser range finders have also been used extensively for robot localization in structured indoor environments where the map of the environment is supplied to the robot [108, 23, 68]. In outdoor environments, laser range finders find application in road feature detection (markers, lane, edge etc.), and obstacle detection and avoidance for autonomous navigation purposes [97, 55, 110]. The extracted road features along with a stored database of road features can be used for vehicle localization [57]. With the advancement of technology, laser range finders have reduced in size, consume less power, have high bandwidth, give better accuracy, and a longer range. This trend combined with the ability of modern day computers to process the large amount of data from range sensors in real-time, make these sensors an attractive choice as terrain sensors.

The downside of terrain sensors is that they assume that an accurate representation of the surrounding terrain is available. The accuracy of the terrain representation has a direct bearing on the estimates obtained by terrain aided localization. Also the visibility of terrain features is a requirement. Obstacles or occlusions in the path of these sensors can cause the estimator to lose track of vehicle location and orientation if redundant sensors are not available.

1.3 Sensor Fusion

Each localization sensor by itself suffers severe shortcomings, but when two or more sensors are used in coordination, outstanding results can be obtained [72]. For example, GPS when integrated with inertial sensors compensates for the slow data rate of the GPS (20 Hz), resulting in an excellent state estimator. Not only does GPS provide an accurate initial estimate for inertial estimation, but also corrects the inertial estimate periodically to remove the integration error and error due to gyro misalignment and biases [39],[96].

Commercial systems combining GPS/DGPS with tactical grade IMUs are widely available today [26, 104]. Tactical-grade IMUs have gyroscopes with bias of the order of 1 $\frac{deg}{hour}$ and accelerometers with bias of the order of 1 mg. Gyros of this quality can sense the earth's rotation rate which makes the initial alignment of INS possible. In case of a GPS outage, the INS-only localization estimate can navigate the vehicle for short periods of time of the order of minutes although the error increases over time [26]. Position accuracy of 2 m for single-point GPS and 1 cm for Real-time Kinematic GPS and orientation accuracy of the order of 0.01 degrees have been reported [26]. The cost of a typical tactical-grade IMU is \$10,000-\$20,000 which makes them unsuitable for use in commercial vehicle applications.

There are different architectures to fuse GPS and INS measurements based on the amount of information sharing between GPS and IMU estimator loops. Loose coupling is the simplest of these architectures where navigation solution is generated independently by the GPS and the INS. These independent navigation solutions are subsequently combined to generate a filtered GPS/inertial solution. Kalman filters integrating the GPS and INS at the measurement level (pseudorange and Doppler measurements) have been recently shown to result in significant performance improvements [44, 56] in environments where the satellite visibility is low e.g. urban canyons. Another approach using ultra-tight coupling between the GPS and the INS, where the inertial measurements are used to help the GPS tracking loops, has anti-jamming capability and is robust against high dynamic motion [12, 82, 52, 54].

As observed by [46, 87, 38], for a navigation system with a low-grade IMU and an single antenna GPS system, the time-invariant error dynamics model of the INS has seven unobservable modes; attitude, IMU to GPS antenna lever arm, and component of the gyro bias in the direction of the specific force. As the gyro bias is unobservable, the error in the yaw angle estimate can increase with time in the absence of vehicle accelerations. All the above mentioned unobservable modes can be made observable by maneuvering the vehicle through a series of accelerations and rotations.

Because observability is an issue in a navigation system using a low-grade IMU and a single antenna GPS system, techniques to directly measure roll, pitch and yaw angles of the vehicle have been developed using GPS measurements [65, 42, 7]. Differential Carrier Phase (DCP) measurements from multiple GPS antennas can be used to calculate the baseline vectors between antennas, which in turn are used to calculate the roll, pitch and yaw angles of the vehicle. Sub-degree accuracy in measuring roll, pitch and yaw have been reported in literature using this technique [65, 42]. This technique is very useful in providing high accuracy measurements of roll, pitch, and yaw, and hence it makes all the states of the navigation system observable under any static or dynamic condition. On the other hand, it suffers from the same disadvantages as GPS, that of the requirement of the line of sight to at least four satellites, accuracy dependence on satellite geometry, multi-path errors as well as susceptibility to jamming [109]. Also, commercial Attitude and Heading Reference Systems (AHRS) based on multiple GPS antennas and receivers are quite expensive: the approximate cost of a Novatel AHRS is around \$20,000 to \$30,000.

Another common two-sensor integration method uses terrain sensors combined with inertial sensors [89, 31, 45]. Examples of terrain sensors include vision sensors, laser range finders, radars, sonar and infrared sensors. Vision and inertial sensors naturally complement each other. In addition to providing redundancy, each of these sensors can be used to resolve the uncertainties arising from using the other sensor alone. For instance, image measurements giving absolute position and orientation can nullify the error that accumulates when integrating inertial readings, and thus this can be used to detect misalignment and estimate biases in inertial sensors. On the other hand, inertial data can resolve the uncertainty in motion estimated by camera, for example one that sees a degenerate scene such as one containing very few features. Laser range sensors or radar can also be combined with inertial sensors in a similar way.

Most localization algorithms have a structure as shown in Figure 1.3. Measurements from inertial sensors are integrated to get estimates of position, velocity and attitude of the vehicle. These estimates are also supplied to an estimation algorithm, which also takes measurements from a set of external sensors as input. The estimation algorithm generates a set of corrections and feeds them back to the inertial sensors. The output of the inertial sensors is thus adjusted to reflect information obtained from external sensors thus achieving sensor fusion.

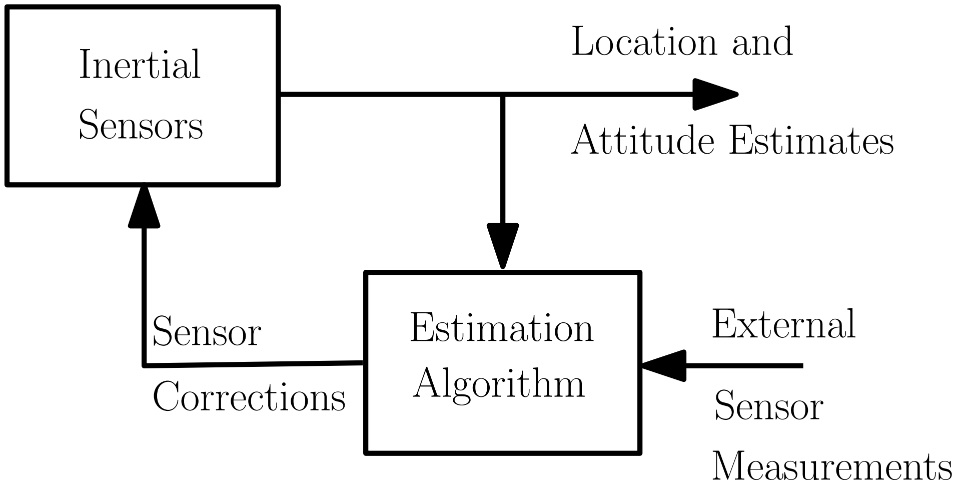


Figure 1.3. Structure of Localization Algorithms

Kalman Filtering is a popular method used in sensor fusion applications because its formulation makes it easily adaptable for sensor fusion. The Kalman filter combines the sensor measurements in such a fashion as to give an optimal estimate in the least squares sense. This combination brings out the advantages of all the sensors.

1.4 Outline of the Remaining Chapters

The remaining chapters of this dissertation are organized as follows:

Chapter 2: Related Work gives a brief review of methods of vehicle state estimation as well as methods used for sensor fusion in vehicle state estimation.

Chapter 3: Vehicle Localization Using GPS and IMU describes the basics of GPS and INS systems and develops a framework to fuse the measurements from these systems.

Chapter 4: Orientation Estimation Using Vision and Inertial Sensors describes a novel method for estimating vehicle roll, pitch and yaw using machine vision and inertial sensors employing a rendered representation of the surrounding terrain.

Chapter 5: Position and Orientation Estimation Using Near-Field Features extends the work done in chapter 4 to include near-field features for estimating vehicle position in addition to vehicle orientation.

Chapter 6: Conclusions summarize the results of this thesis and discusses possible future research directions.

Chapter 2

Literature Review

In this chapter, the literature on vehicle localization using GPS and map-aided vision is presented. Algorithms commonly used for sensor fusion are also listed with a brief discussion of their strengths and weaknesses.

2.1 GPS/INS

GPS has revolutionized outdoor localization in many different domains. GPS-aided Inertial Navigation Systems (INS) are widely used in commercial and military applications [50, 112, 25, 11]. They also find application in autonomous land vehicle systems to increase the integrity of their navigation systems [90, 101, 96, 22].

To harness the benefits of GPS/IMU combinations in commercial vehicles, much attention has been focussed in the last decade on improving the performance of Micro Electro-Mechanical Systems (MEMS) based automotive-grade IMUs with cost ranging from \$100 to \$1000. Noise levels of automotive-grade IMUs are very high which degrades the performance of GPS/IMU combinations. Along with the effort of the MEMS community to develop technologies which better the performance of gyros and accelerometers, a lot of research has been done to find better algorithms to fuse GPS and IMU measurements. For example, inertial sensor errors have been characterized using stochastic methods [1, 48, 79]. Inertial sensor error states are modeled and estimated along with position, velocity and orientation states of the vehicle which results in better state estimates as well as less error accumulation during GPS outages [32, 78].

Much work in the recent past has focussed on integrating MEMS based IMUs with accurate GPS systems (DGPS,RTK). For example, Salychev et al. [93] use Ashtech's GG24 GPS/GLONASS DGPS system with MotionPak^{TM} IMU (cost $\approx$ \$8000) to assess the feasibility of the system for navigational purposes. Mathur et al. [74] also do a similar feasibility study using the same set of IMU and DGPS. Novatel corporation has developed a MEMS IMU based navigation system using BAE sytems SiIMU01 IMU and Novatel OEM4-G2 GPS receiver running in RTK mode which is described in [24]. Crossbow MEMS IMU (AHRS-DMU-HDX) has been used with carrier phase DGPS (Ashtech Z-XII) for a marine application by Hide et al. in [44]. This IMU costs around \$4000. Brown et al. and Godha et al. [10, 36] use MEMS based Crista IMU (cost $\approx$ \$2000) along with Novatel OEM4 DGPS system for vehicle navigation. All the above mentioned studies, although providing promising results, use an IMU which is prohibitively expensive (cost \$2000 upwards) for automotive applications. Also, these studies assume that corrections from a base station are available for implementing DGPS. Although Wide Area Augmentation System (WAAS) corrections are widely available throughout U.S., the accuracy achieved using WAAS corrections is much lower than RTK or DGPS corrections [77].

GPS/IMU combinations have been used in vehicle dynamics studies where the aim is to study the variation with time of vehicle rotation rates, orientation, lateral velocity and side slip [6, 91]. Bevly [6] presents a method to measure transient changes in vehicle sideslip, roll and pitch angles using an automotive grade IMU and inexpensive single antenna GPS receiver. Simplifications have been made to INS mechanization equations and assumptions about the frequency content of the signals being estimated are used. Experimental results have been presented, although no measure of accuracy of the presented techniques has been given e.g. by comparison with a navigation-grade IMU. Ryu et al. [91] use a two antenna GPS receiver to estimate planar vehicle states, e.g. lateral velocity and yaw rate, corrected for vehicle roll and road grade. Parameters related to vehicle roll dynamics have also been estimated using GPS and INS measurements. As the vehicle roll and yaw are directly measured using a two antenna GPS system, simplifications similar to [6] have been used in the INS mechanization.

From the above discussion, it follows that the performance of a low-cost GPS receiver operating in autonomous mode along with a low-cost MEMS IMU needs to be investigated. This dissertation seeks to achieve good localization performance using low-cost GPS/IMU combinations by accurately characterizing the error sources present in these sensors. In addition to providing a measure of the performance of low-cost GPS/IMU combinations, this investigation builds a platform for the fusion of vision and inertial measurements used later in this dissertation.

2.2 Map-Aided Localization Using Vision

There is also a large body of research fusing vision and map data for vehicle or robot localization. With the public availability of United States Geographical Survey data, other approaches have used databases of stored DTMs and DEMs to aid in vehicle localization [105, 62, 89, 31, 45]. For example, Talluri and Aggarwal [105] take different views of the horizon and use them to search the underlying map for possible robot locations. Several of these horizon views are used to constrain the possible robot locations to a small area. They show robot localization results on simulated data based on a DEM. As another example, in Lerner et al. [62] the pose of the camera at two consecutive frames is derived using a DTM and the corresponding features in two frames. Each hypothesized pose of the camera gives an estimate of the depth of the features that are visible. The authors present a ray-tracing algorithm for finding the best pose of the camera at two frames which leads to displacement of the features as close as possible to the given two frames. Although novel, this algorithm is computationally expensive and results are presented for only position and yaw estimates. No experimental results are shown for roll, pitch and height. Rodriguez and Aggarwal [89] present a map-based algorithm for passive aircraft navigation system. Stereo analysis is performed on successive images to recover an elevation map which is matched to the reference digital map of the 3-D terrain to determine the position and heading of the aircraft. In Furst and Dickmanns [31], vision data is used along with a model of key landmarks (e.g. runway, buildings etc. around the airport) to aid in the positioning of the aircraft. GPS and inertial sensors are coordinated with vision to improve position and orientation estimates in aircraft navigation.

In many applications, knowledge of surrounding terrain is not known a-priori, so the goal is to build a map of the environment while simultaneously localizing the vehicle within the map, a process called Simultaneous Localization and Mapping (SLAM) [29, 14, 41]. Hoffman [45] present a SLAM based technique where semi-sparse terrain maps are generated and matched to get a vision-based state estimate. This estimate is then fused with wheel odometry using a Kalman filter to further improve the state estimate. Because the algorithm iteratively minimizes the total distance between point clouds at two frames, it is computationally quite expensive. Further, the accuracy of the system is limited because its a SLAM-based technique and there is no absolute terrain model available.

Terrain-matching localization similar to the work in this dissertation has been studied in the area of underwater vehicle localization, where x, y position and yaw orientation of the underwater vehicle are estimated by using a reference map. This reference map is either a map generated using multi-beam echo sounder as in the work of Lucido et al. [67], or from a DEM as in the work of Strauss et al. [99]. Lucido et al. [67] use a matching algorithm which locates the local depth map within the a priori larger map to determine absolute position and orientation of the underwater vehicle. They use high curvature points in reference and local maps to match the two depth profiles. The algorithm they present assumes that pitch and roll angles are known to sufficient precision. Strauss et al. [99] also make that assumption while matching underwater terrain profiles using sonar and a DEM. Vision has been used by Marks et al. [73] to determine the position of an underwater vehicle. Texture correlation and distance to a planar surface, obtained using a sonar proximity sensor, have been used to find position offsets with respect to a reference image. Visual mosaics of areas of ocean floor are generated by Richmond et al. [88] using vision and inertial sensors. In the work by Zhang et al. [116], a proof-of-concept simulation study is presented, where x,y position and yaw orientation of an underwater autonomous vehicle is estimated by fusing simulated measurements from inertial sensors and terrain matching.

It is possible to localize a vehicle without a database by fusing inertial and vision sensors, and there still remains a large body of research in this area [115, 86, 100, 49, 15]. Strelow and Singh [100] give an excellent review of vehicle state estimation from image and inertial measurements. Two algorithms (one batch and one recursive) are presented to combine vision and inertial measurements. The algorithms estimate 6 degree of freedom vehicle position, gyro and accelerometer biases and the gravity vector with respect to the world coordinate system. Sparse scene structure is also estimated where there is a provision to add/remove image features as they become visible/invisible from the image sequence. But both batch and recursive methods are susceptible to gross feature tracking errors despite the fusion of both image and inertial sensors. None of the sensors used gives an absolute estimate, so a drift in the resulting estimate is possible over time.

Langelaan et al. [60, 58, 59] present a Unscented Kalman filter (UKF) based state estimator for a UAV flying through a cluttered environment (dense forest) using monocular camera and inertial measurements. A proof-of-concept study is given for the autonomous navigation of the UAV through a forest while estimating the position, orientation and velocity of the UAV as well as positions of landmarks in the environment. An experimental study with a ground vehicle is also presented to prove the performance of the estimator. The results from this study, although promising, will accumulate correlated errors over large distances due to drift like most of the SLAM applications.

A method is presented in [34] which constrains the inertial sensor drift by tracking unknown features in the environment using a camera during GPS outages. A tightly coupled GPS/INS configuration is used to take advantage of low satellite visibility. A corner detector is used to extract features from images and a joint compatibility data association algorithm using Mahalanobis distance between the features is used for data association between successive image frames. Although the algorithm is shown to successfully contain the inertial sensor drift, the accuracy of the vision/inertial system in the absence of GPS updates is not addressed in this work.

Measurements from GPS, an odometer, a steer angle sensor and a camera along with a map of the environment are used by [57] to estimate the position and yaw angle of the vehicle in Kalman filter framework. Vehicle roll and pitch angles are not estimated in this work. Road markers are used as features and the algorithm uses a nearest neighbor criteria for object correspondence between real and virtual images. As a result, although good position accuracy is obtained, orientation estimates are not very accurate. Position and velocity vector from GPS is used in [16], lateral distance from a mapped lane and measurements from an IMU to improve lateral position estimation during GPS outages. Vehicle lateral position is the only information extracted using vision measurements in this algorithm.

Surveyed road marking information (lane lines, crosswalks, turn arrows etc.) along with the stop-lines are captured using a camera and used as measurements in a particle filter for vehicle localization in sparse GPS environments in [76]. Position estimation accuracy is shown to improve appreciably in an urban setting both with GPS and during GPS outages. The information used by the vision algorithm is the presence or absence of stop lines or lane markings and marking styles. Orientation information is not extracted using vision by this algorithm. Although this algorithm is useful in extracting global position information, the position accuracy achieved is of the order of 3 to 4 meters.

Geographically referenced aerial or satellite images are used to build a global feature map by [83]. Lane markings are used as features in the work presented which are extracted using canny edge detector and their centroid is marked as the respective feature location. The feature matching problem is treated like a standard point pattern matching problem and iterative closest point method is used to get the solution. Three variables of vehicle horizontal position and attitude are estimated. The use of lane markings as point features results in loss of information. Also, the effect of number of visible lane markers on the accuracy of the algorithm has not been addressed.

A method to fit parameterized three-dimensional models to images is presented in [63]. Curves are extracted using edge extraction techniques and perpendicular distance between the curves is minimized using gradient based methods to solve for viewpoint and model parameters. The problem with this technique is that edge extraction methods are computationally expensive and edge correspondence problem is not directly addressed.

As described in the preceding paragraphs, existing methods of map-aided localization are computationally expensive or have drift issues due to lack of an absolute reference. Therefore, the work done in this dissertation seeks to develop efficient algorithms using low-cost vision and inertial sensors along with a map of terrain features (e.g. horizons) and roadside features (e.g. road signs, road markers and buildings etc.) for vehicle localization.

2.3 Sensor Fusion

Luo et al. [70] give a tutorial on multi-sensor integration and fusion, where he lists weighted average, Kalman filter and Bayesian estimation as the methods for signal level data fusion. A weighted average method takes the weighted average of the information and uses this as a fused value. A Kalman filter is preferred over weighted average filter because, while being nearly equal in processing requirements, the Kalman filter gives out fused estimates that are optimal in a statistical sense. Bayesian estimation using consensus sensors looks for sensors that are likely to be in error and eliminates them from consideration in sensor fusion. The information from each sensor is expressed as a probability density function and a Bayesian estimator is used to maximize the likelihood of the "consensus sensors" to achieve sensor fusion.

Possible problems with creating a general methodology for multi-sensor integration and fusion are mentioned in the survey paper by Luo et al. [69]. The first requirement for sensor fusion is that the data from all the sensors should be time synchronized. This is called data alignment. The data should not only be time synchronized, it has to refer to the same features in the environment for sensor fusion to be possible. The problem of determining that each sensor is referring to the same features in the environment is called "Registration". For example, in an environment mapping application with a laser range finder and a CCD camera, texture mapping the camera data to a 3D model obtained from range data is a problem of registration. Another problem mentioned in this paper is sensor noise modeling. Sensor noise is usually assumed to be white (not correlated in space and time), Gaussian and independent to make the the analysis mathematically tractable. Correlated noise can be modeled by a shaping filter which takes white noise as input. Gaussian-ness assumption can only be justified if the noise is caused by a number of independent error sources, which is a result of central limit theorem. If the errors do not originate from within the system, the independence assumption is usually valid.

Among all the methods for sensor fusion for localization, Kalman filters are the most widely used [33]. When the system dynamics and observation models are linear, the Kalman filter can be used to compute the minimum mean squared estimate (MMSE). These filters approximate process and measurement noise by unimodal Gaussian distributions. Despite these strong assumptions, the Kalman filter has been applied with great success to many estimation problems, e.g. Watanabe [111] describes a location estimator based on the fusion of DGPS measurements with optical gyro and wheel speed sensors using a standard Kalman filter. A multi-model adaptive estimator based on Kalman filtering is presented in [106]. The main advantage of Kalman filters is their computational efficiency although that comes at the cost of restricted representational power since the Kalman filters can only represent unimodal distributions. The Gaussian assumption on the noise distributions also leads to errors in estimation. Hence, the use of Kalman filters is best when the process and measurement noise variance is not too high.

Variations of Kalman filters have been sought to increase its applicability to systems with nonlinear dynamics and measurement equations. The best known algorithm for this purpose is extended Kalman filter (EKF) [2]. This filter is based on linearizing the process and measurement model at the operating point using Taylor series expansion. Although successfully implemented for many applications [103, 3], there are many drawbacks of this filter [53]. If the time steps are not sufficiently small, the linearization can produce highly unstable filter performance. For highly non-linear systems, problems may arise even with small time steps. In this case, higher order EKF may have to be used. The calculation of Jacobian matrices used in the linearization by Taylor series expansion can be non-trivial and lead to implementation difficulties. Also, small time steps imply that computation cost involved is high, because the Jacobian matrices and predictions of state and covariance need to be calculated at each time step.

A new filter called the Unscented Kalman filter (UKF) was introduced in [53], which is based on the premise that "With a fixed number of parameters it should be easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function". The prior distribution is approximated using the minimum set of points that capture the first three moments of the prior distribution. This filter produces estimates of state and covariance which are provably more accurate than the EKF but without the need of calculating Jacobian matrices. Simulation and experimental studies have shown this filter to be give very good results in highly nonlinear estimation problems [60, 58, 59]. The limitation of UKF is that it does not apply to general non-Gaussian distributions.

Sequential Monte Carlo methods (particle filters (PFs)) allow for complete representation of posterior distribution of states, so that all the statistical estimates like mean, modes, variance and kurtosis can be calculated, enabling them to handle any nonlinearities or distributions [37, 102]. However, the number of particles required are very large when the the dimension of the state vector is high and performance depends on the choice of the function used to propagate the samples. So, particle filters don't lend themselves very well for real-time implementation for systems with a large number of states.

Chapter 3

Vehicle Localization Using GPS and IMU

This chapter discusses the fundamentals of GPS and IMU integration focussing on low-cost Commercial Off-the-Shelf (COTS) systems available in the market today. As discussed in chapter 2, previous investigations on GPS/INS integration have not focussed on the integration of low-cost COTS GPS and INS. Specifically, integration of a low-cost GPS receiver (cost $\approx$ \$100) operating in autonomous mode with a MEMS based low-cost IMU (cost $\approx$ \$100) is considered in this work. The error sources in GPS and INS systems are characterized to choose suitable stochastic models for the error sources and to identify parameters for these models. A Kalman filter framework is developed which estimates 6-DOF vehicle position along with the estimation of error sources in the IMU. As the position of the vehicle is directly observable, the performance of the system is assessed by the ability of the system to estimate vehicle orientation.

This chapter is organized as follows: The coordinate systems used in this chapter are defined in section 3.1. Section 3.2 gives an overview of the GPS system. Errors in the GPS position and velocity measurements are characterized in section 3.3. An overview of the INS system is given in section 3.4 followed by its error characterization in section 3.5. A Kalman filter framework to fuse measurements from GPS and INS systems is presented in section 3.6. Numerical simulations are done in section 3.7 to verify the correctness of the Kalman filter. Experimental results are presented in section 3.8 where the in-field performance of the low-cost GPS/IMU combination is compared to measurements obtained from a high accuracy GPS/IMU system using experiments done at the PTI test track. Finally, conclusions are drawn in section 3.9 along with the drawbacks and possible improvements of this work. A nomenclature for this chapter is listed at the end of the chapter.

3.1 Coordinate Reference Frames

Central to the process of navigation is the definition of a number of cartesian co-ordinate reference frames. For ease of computation and mathematical transformations, each of these frames is an orthogonal, right-handed co-ordinate frame. The co-ordinate frames typically used in GPS and inertial navigation systems are shown in figure 3.1 and described below [107].

The inertial frame (i-frame) has its origin at the center of the Earth and axes OX_i, OY_i, OZ_i which are non-rotating with respect to the fixed stars. Oz_i axis is coincident with the Earth's polar axis.

The Earth frame (e-frame or Earth-centered, Earth-fixed (ECEF) frame) has its origin at the center of the earth and axes OX_{ECEF}, OY_{ECEF}, OZ_{ECEF} which are fixed with respect to the Earth. O_{ECEF} axis is coincident with the Earth's polar axis and hence the axis OZ_i of the inertial frame. The intersection of the plane of Greenwich meridian with the Earth's equatorial plane defines the axis OX_{ECEF}. The e-frame rotates with respect to the i-frame at a rate Ω (The Earth rotation rate $\approx 15deg/hr$) about Oz_i axis.

The navigation frame (n-frame) is a local geographic frame with origin at the origin of the navigation system (GPS or INS, point P in figure 3.1), and axes aligned with *local* north, east and local vertical down directions which are shown as N, E and D axes in figure 3.1 respectively.

Local Level System (LLS) or east-north-up system (ENU-frame) is exactly the same as the navigation frame but with the axes switched. The X-axis points the local east, the Y-axis points the local north and the Z-axis points vertically upwards in this co-ordinate frame.

The body frame (b-frame) has its origin at the center of gravity of the vehicle and axes pointing towards roll, pitch and yaw axes of the vehicle. This coordinate frame is completely local to the vehicle and is shown in figure 3.2.

3.2 Global Positioning System

GPS is one of the many Global Navigation Satellite Systems (GNSS) which is fully operational [77]. GPS uses satellites as external references to trilaterate the position of

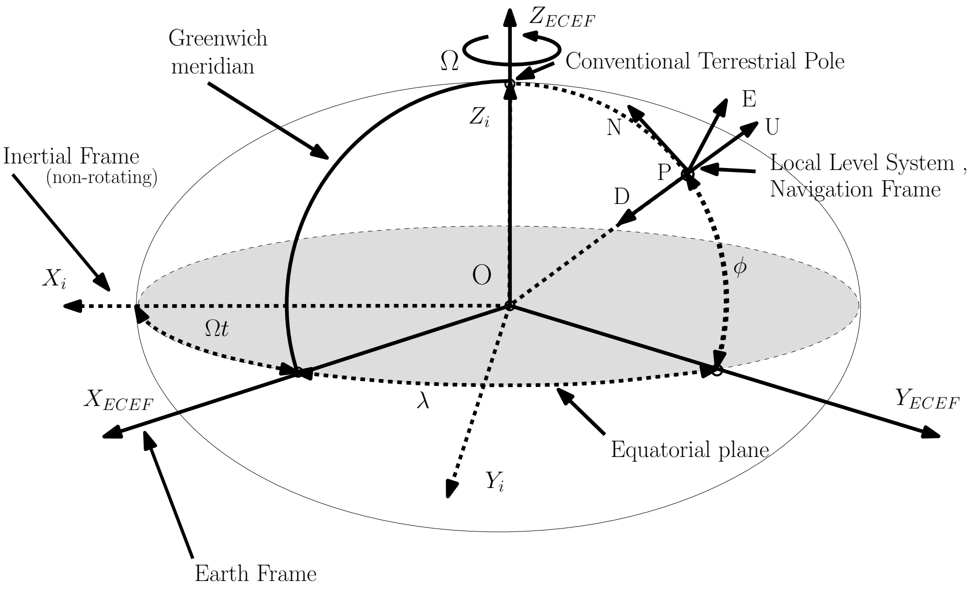


Figure 3.1. Coordinate frames of reference

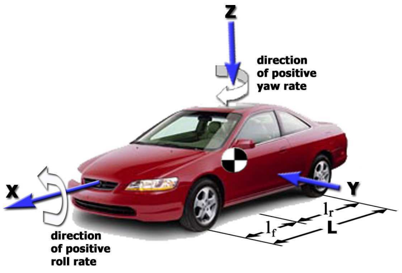
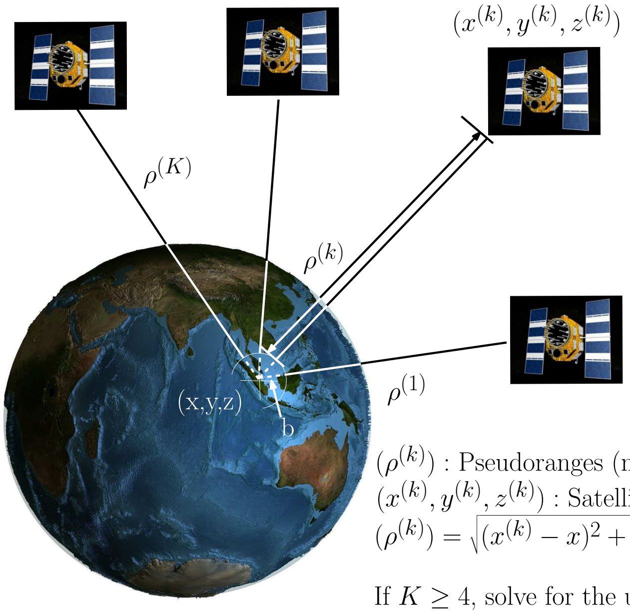


Figure 3.2. The Body Frame

the vehicle. The distance measurements from the user to the satellite are based on the transit time of the signal from the satellite to the user and are biased by a common amount. These distance measurements are called pseudoranges. Pseudoranges from at least four satellites are needed to estimate the user position.

GPS was developed by the U.S. Department of Defence (DoD) to offer the U.S. military accurate and globally available estimates of position, velocity and time. Civil users were also given access to GPS but were only provided with a limited accuracy



$(\rho^{(k)})$: Pseudoranges (measurements)
$(x^{(k)}, y^{(k)}, z^{(k)})$: Satellite Positions (known)

$$(\rho^{(k)}) = \sqrt{(x^{(k)} - x)^2 + (y^{(k)} - y)^2 + (z^{(k)} - z)^2} - b \qquad k = 1, 2, \ldots, K$$

If $K \geq 4$, solve for the user position (x,y,z)
and the receiver clock bias $t_b = \frac{b}{c}$
where c denotes the velocity of light

Figure 3.3. Principle of satellite navigation

consistent with national security considerations. GPS offers two kinds of service:

- Standard Positioning Service (SPS) for civilian use.
- Precise Positioning Service (PPS) for military and DoD authorized use.

PPS provides more accurate position estimates, and is encrypted to avoid spoofing. Each GPS satellite transmits signals at two radio frequencies referred to as Link 1 (L1: $f_{L1} = 1575.42 MHz$) and Link 2 (L2: $f_{L2} = 1227.60 MHz$), in the L-band, which covers frequencies between 1 GHz and 2 GHz. Each GPS signal consists of the following components:

- RF sinusoidal carrier signal with frequency f_{L1} and f_{L2}
- Binary codes called pseudo-random noise (PRN) codes are modulated on the carrier signals. These codes are used to measure the range from the user to the satellites

precisely. Codes associated with SPS are called coarse/acquisition codes (C/A-codes) and the codes for PPS are named precision codes (P(Y)-codes) and are encrypted. A unique C/A-code on L1 and unique P(Y)-codes on L1 and L2 are transmitted by each satellite.

- Navigation data containing satellite position and velocity (ephemeris), clock bias parameters and an almanac giving reduced precision ephemeris data on all satellites in the constellation.

For a low-cost receiver operating in autonomous mode, only L1 frequency C/A-code measurements are available. Also, the user usually doesn't have access to the raw measurements. The receiver directly outputs a least-squares solution to the user along with a measure of loss of accuracy due to the satellite geometry. The loss of accuracy due to satellite geometry is termed Dilution of Precision (DOP). The receiver generally outputs the Position Dilution Of Precision (PDOP), Horizontal Dilution Of Precision (HDOP) and Vertical Dilution Of Precision (VDOP) values. PDOP can be termed as a scaling factor which has to be multiplied by the estimated range error to the satellite to estimate the RMS 3D error in vehicle position. HDOP denotes the scaling factor for the horizontal error. Scaling factor for the vertical error is denoted by VDOP. Although these scale factor values are useful in giving an idea of overall vehicle position error assuming a roughly constant user range error, individual errors in east and north directions as well as covariances between these errors are not output by common low-cost receivers. But, the receiver does output enough information to calculate the satellite geometry matrix (defined later), which characterizes errors in east, north and vertical directions individually as well as their covariances. An analysis of positioning error is therefore presented to be able to calculate the geometry matrix.

3.2.1 Position Estimation using Pseudoranges

A brief overview of position estimation using pseudoranges is given here. A linear model for position estimation is described along with characterization of errors of this linear model. Notation follows [77] where a detailed treatment of the topic can be found.

At GPS time t, the pseudorange measurement from the kth satellite can be modeled as

$$\rho^{(k)}(t) = r^{(k)}(t, t-\tau) + c[\delta t_u(t) - \delta t^{(k)}(t-\tau)] + I^{(k)}(t) + T^{(k)}(t) + \varepsilon_\rho^{(k)}(t) \tag{3.1}$$

where $k = 1, 2, \ldots, K$. are the number of satellites in view. τ is the time it takes for the signal to travel from the satellite to the user GPS receiver. The pseudorange $\rho^{(k)}(t)$ is determined from the measured apparent transit time as:

$$\rho^{(k)}(t) = c\left[t_u(t) - t^s(t-\tau)\right] \tag{3.2}$$

where c is the speed of light. $r^{(k)}(t, t-\tau)$ is the true distance between the satellite antenna at the signal transmission time $(t-\tau)$ and the receiver antenna at the signal reception time t. $\delta t_u(t)$ and $\delta t^{(k)}$ are the receiver and satellite clock offsets, relative to a accurately maintained time reference called GPS Time (GPST). $I^{(k)}(t)$ and $T^{(k)}(t)$ are the ionospheric and tropospheric propagation delays, respectively. Modeling errors, e.g. orbital prediction error and satellite clock estimation error, and un-modeled errors, e.g.

multi-path are accounted for by the term $\varepsilon_{\rho}^{(k)}(t)$.

Ionospheric and tropospheric delays can be modeled and corrected for to a large extent, but are not always implemented in low-cost GPS receivers. The satellite clock offset estimate $\delta t^{(k)}$ is transmitted in the navigation data from the satellite and can be compensated for. Let $\rho_c^{(k)}$ denote the pseudorange obtained after accounting for the satellite clock offset and $\tilde{\varepsilon}_{\rho}^{(k)}(t)$ be the residual error created by including the ionospheric and tropospheric errors in $\varepsilon_{\rho}^{(k)}(t)$. Under these assumptions, the modified pseudorange equation can be written as:

$$\rho_c^{(k)}(t) = r^{(k)}(t, t-\tau) + c\delta t_u(t) + \tilde{\varepsilon}_{\rho}^{(k)}(t) \tag{3.3}$$

The standard deviation of $\tilde{\varepsilon_{\rho}^{(k)}}(t)$ can range from 6 to 10 m for a single frequency receiver operating in autonomous mode. Let $\mathbf{x_u} = (x_u, y_u, z_u)$ be the position of the user at the time of signal measurement and $\mathbf{x_s}^{(k)} = (x_s^{(k)}, y_s^{(k)}, z_s^{(k)})$, for $k = 1, 2, \ldots, K$, be the position of the K satellites visible to the receiver at the time of signal transmission in the ECEF frame. The geometric range from the user to the satellite is:

$$r^{(k)} = \sqrt{(x_s^{(k)} - x_u)^2 + (y_s^{(k)} - y_u)^2 + (z_s^{(k)} - z_u)^2} = \| \mathbf{x_s}^{(k)} - \mathbf{x_u} \| \tag{3.4}$$

Using equation 3.4, equation 3.3 can be re-written as:

$$\rho_c^{(k)} = \| \mathbf{x_s}^{(k)} - \mathbf{x_u} \| + b(t) + \tilde{\varepsilon}_{\rho}^{(k)} \tag{3.5}$$

where $b(t)$ has replaced the receiver clock bias term $c\delta t_u$. Pseudorange measurement from each visible satellite can be used to form a non-linear equation of the form 3.5. Each of these equations has four unknowns: Three components of the user position in the vector $\mathbf{x_u}$ and range bias due to the receiver clock error, b(t). This implies that concurrent pseudorange measurements from at least four satellites are required to estimate the user's instantaneous position.

If range measurements from K visible satellites are available, a system of K non-linear equations of the type 3.5 can be solved by using the Newton-Raphson method. In this method, the non-linear equations are linearized about an approximated user position, and solved iteratively. Let $\mathbf{x}_{u_0} = (x_{u_0}, y_{u_0}, z_{u_0})$ and b_0 be the initial estimates of user position in the e-frame and receiver clock bias, respectively. As seen in equation 3.5, the corrected pseudorange measurement from satellite k is denoted by $\rho_c^{(k)}$. The estimate of the user position and receiver clock bias also gives an approximation of the pseudorange to the satellite, given by:

$$\rho_0^{(k)} = \| \mathbf{x_s}^{(k)} - \mathbf{x_{u0}} \| + b_0 \tag{3.6}$$

Next, denote the true user position and true clock bias as $\mathbf{x_u} = \mathbf{x}_{u_0} + \delta\mathbf{x_u}$ and $b = b_0 + \delta b$, where $\delta\mathbf{x_u}$ and δb are the errors in the initial estimates of user position and receiver clock bias, respectively. In the following, a system of linear equations is developed with $\delta\mathbf{x_u}$ and δb as unknowns.

$$\delta\rho^{(k)} = \rho_c^{(k)} - \rho_0^{(k)} \tag{3.7}$$

$$= \Big(\| \mathbf{x_s}^{(k)} - \mathbf{x_u} \| + b\Big) - \Big(\| \mathbf{x_s}^{(k)} - \mathbf{x}_{u_0} \| + b_0\Big) + \tilde{\varepsilon}_\rho^{(k)}$$

$$= \| \mathbf{x_s}^{(k)} - \mathbf{x}_{u_0} - \delta\mathbf{x_u} \| - \| \mathbf{x_s}^{(k)} - \mathbf{x}_{u_0} \| + (b - b_0) + \tilde{\varepsilon}_\rho^{(k)} \tag{3.8}$$

$$\approx -\frac{(\mathbf{x_s}^{(k)} - \mathbf{x}_{u_0})}{\| \mathbf{x_s}^{(k)} - \mathbf{x}_{u_0} \|} \cdot \delta\mathbf{x_u} + \delta b + \tilde{\varepsilon}_\rho^{(k)} \tag{3.9}$$

$$= -\mathbf{1}^{(k)}(\mathbf{x_s}^{(k)}, \mathbf{x}_{u_0}) \cdot \delta\mathbf{x_u} + \delta b + \tilde{\varepsilon}_\rho^{(k)} \tag{3.10}$$

Here, $\mathbf{1}^{(k)}(\mathbf{x_s}^{(k)}, \mathbf{x}_{u_0})$ is the estimated line-of-sight unit vector from the initial estimate of the user position to satellite k, and $\mathbf{a} \cdot \mathbf{b}$ denotes the dot product of vectors $\mathbf{a}$ and $\mathbf{b}$. Use has been made of Taylor series expansion in going from equation 3.8 to 3.9. For K visible satellites, a system of K linear equations of the form 3.10 can be written as:

$$\delta\rho = \begin{bmatrix} \delta\rho^{(1)} \\ \delta\rho^{(2)} \\ \vdots \\ \delta\rho^{(K)} \end{bmatrix} = \underbrace{\begin{bmatrix} (-\mathbf{1}^{(1)}(\mathbf{x_s}^{(1)}, \mathbf{x_{u0}}))^T & 1 \\ (-\mathbf{1}^{(2)}(\mathbf{x_s}^{(2)}, \mathbf{x_{u0}}))^T & 1 \\ \vdots & \\ (-\mathbf{1}^{(K)}(\mathbf{x_s}^{(K)}, \mathbf{x_{u0}}))^T & 1 \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} \delta\mathbf{x_u} \\ \delta b \end{bmatrix} + \tilde{\varepsilon}_\rho \tag{3.11}$$

Here $\mathbf{G}$ is a $(K \times 4)$ matrix characterizing the satellite geometry for the user location and is termed the geometry matrix.

For the case of $K = 4$, four equations can be solved for four unknowns directly giving:

$$\begin{bmatrix} \delta\mathbf{x_u} \\ \delta b \end{bmatrix} = \mathbf{G}^{-1}\delta\rho \tag{3.12}$$

Here $\delta\rho$ is calculated using equation 3.7. For $K > 4$, the pseudo-inverse method from linear algebra can be used to give least-squares solution for the corrections to the initial estimates as:

$$\begin{bmatrix} \delta\hat{\mathbf{x}}_{\mathbf{u}} \\ \delta\hat{b} \end{bmatrix} = (\mathbf{G}^{\mathbf{T}}\mathbf{G})^{-1}\mathbf{G}^{\mathbf{T}}\delta\rho \tag{3.13}$$

Using the corrections from the equation above, improved estimates of the unknowns can be calculated as:

$$\begin{aligned} \hat{\mathbf{x}}_{\mathbf{u}} &= \mathbf{x}_{u_0} + \delta\hat{\mathbf{x}}_{\mathbf{u}} \\ \hat{b} &= b_0 + \delta\hat{b} \end{aligned} \tag{3.14}$$

The measurement equations can be linearized about the new estimates of unknowns and process iterated until the change in the estimates is within a certain tolerance. Analysis of this subsection shows the relation between user position error and user range error to the satellites. This relation is given by equation 3.13. In the next section, this relation is used to derive the user position error covariance in terms of user-satellite range error covariance.

3.2.2 Positioning Error

If we assume the error in the pseudoranges from different satellites as zero-mean, uncorrelated, and having the same variance, the model for measurement error in equation 3.11 becomes:

$$\begin{aligned} &E(\tilde{\varepsilon}_\rho) = \mathbf{0} \\ &Cov(\tilde{\varepsilon}_\rho) = E(\tilde{\varepsilon}_\rho \tilde{\varepsilon}_\rho^T) = \sigma_{URE}^2 \mathbf{I} \end{aligned} \tag{3.15}$$

where $E()$ and $Cov()$ denote the mean and covariance of the variable within the parentheses and σ_{URE} denotes the standard deviation of the user range error (URE) for all the satellites.

Let the error in the user's position be $\mathbf{\Delta x_u} = \mathbf{\hat{x}_u} - \mathbf{x_u}$ and $\Delta b = \hat{b} - b$, where $\mathbf{x_u}$ and b represent the true user position and error due to receiver bias, respectively and let $\mathbf{\hat{x}_u}$ and $\hat{b}$ represent the corresponding estimates. As per our assumption that the pseudorange measurements are zero mean, the position and clock bias errors are also zero mean.

$$\begin{aligned} E(\mathbf{\Delta x_u}) &= E(\mathbf{\hat{x}_u} - \mathbf{x}_u) = 0 \\ E(\Delta b) &= E(\hat{b} - b) = 0 \end{aligned} \tag{3.16}$$

Using equation 3.15, the covariance of the position and clock bias errors can be written as [77]:

$$Cov \begin{bmatrix} \delta \mathbf{\hat{x}_u} \\ \delta \hat{b} \end{bmatrix} = \sigma_{URE}^2 \underbrace{\left(\mathbf{G^T G}\right)^{-1}}_{\mathbf{H}} \tag{3.17}$$

Equations for covariance of the errors 3.17 are in ECEF coordinate frame, and are difficult to imagine relative to a local frame. So, equation 3.17 will be converted to the ENU frame to derive errors along the east, north and up directions.

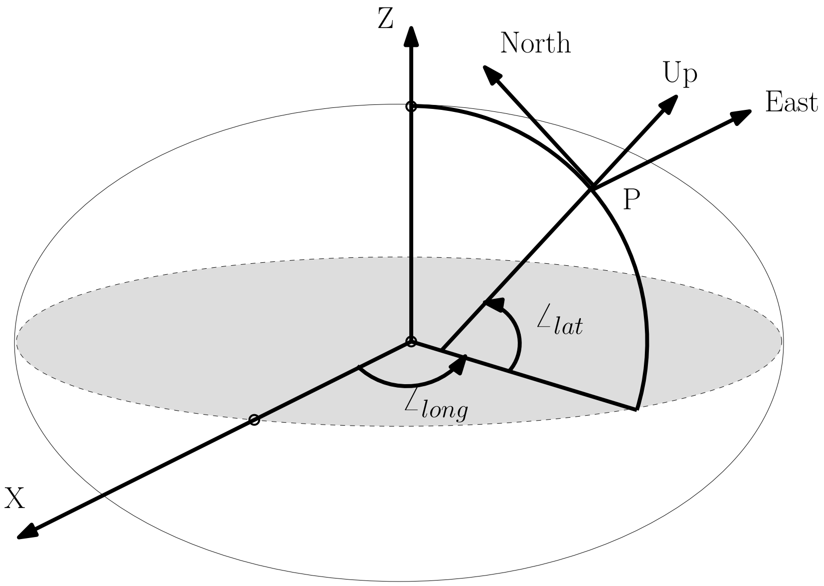


Figure 3.4. ENU frame in relation to ECEF frame

Figure 3.4 shows the relationship between ENU and ECEF coordinate frames. Defining $\mathbf{\Delta X_{u_L}} = (\Delta x_{u_E}, \Delta y_{u_N}, \Delta z_{u_U})$ as the error vector in the ENU frame, the position error vector $\mathbf{\Delta x_u}$ in the ECEF frame can be expressed in the ENU frame as [77]:

$$\mathbf{\Delta X}_{u_L} = \mathbf{R}_L \mathbf{\Delta x}_u \tag{3.18}$$

where $\mathbf{R}_L$ is given by:

$$\mathbf{R}_L = \begin{bmatrix} -\sin(\angle_{long}) & \cos(\angle_{long}) & 0 \\ -\sin(\angle_{lat})\cos(\angle_{long}) & -\sin(\angle_{lat})\sin(\angle_{long}) & \cos(\angle_{lat}) \\ \cos(\angle_{lat})\cos(\angle_{long}) & \cos(\angle_{lat})\sin(\angle_{long}) & \sin(\angle_{lat}) \end{bmatrix} \tag{3.19}$$

Here $\angle_{lat}$ and $\angle_{long}$ are the latitude and longitude at the user's position. Writing the errors in the ENU frame in terms of errors in the ECEF frame:

$$\begin{bmatrix} \mathbf{\Delta x}_{u_L} \\ \delta b \end{bmatrix} = \begin{bmatrix} \mathbf{R}_L & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{\Delta x_u} \\ \delta b \end{bmatrix} = \tilde{\mathbf{R}}_L \begin{bmatrix} \mathbf{\Delta x_u} \\ \delta b \end{bmatrix} \tag{3.20}$$

The covariance matrix in the ENU frame becomes:

$$\begin{aligned} Cov \begin{bmatrix} \mathbf{\Delta x}_{u_L} \\ \delta b \end{bmatrix} &= \tilde{\mathbf{R}}_L Cov \begin{bmatrix} \mathbf{\Delta x_u} \\ \delta b \end{bmatrix} \tilde{\mathbf{R}}_L^T \\ &= \sigma_{URE}^2 \tilde{\mathbf{R}}_L (G^T G)^{-1} \tilde{\mathbf{R}}_L^T \\ &= \sigma_{URE}^2 (\underbrace{\tilde{\mathbf{R}}_L \mathbf{G}^T}_{\tilde{\mathbf{G}}^T} \underbrace{\mathbf{G} \tilde{\mathbf{R}}_L^T}_{\tilde{\mathbf{G}}})^{-1} \\ &= \sigma_{URE}^2 \left[\tilde{\mathbf{G}}^T \tilde{\mathbf{G}} \right]^{-1} \end{aligned} \tag{3.21}$$

The coordinates of the k^{th} satellite with respect to the vehicle in the ECEF-frame are $\mathbf{x_s}^{(k)} - \mathbf{x_u}$. In the ENU-frame, these coordinates become:

$$\begin{gathered} \mathbf{x}_{s_L}^{(k)} - \mathbf{x}_{u_L} = \mathbf{R}_L . (\mathbf{x_s}^{(k)} - \mathbf{x}_u) \\ \begin{bmatrix} (x_{s_L}^{(k)} - x_{u_L}) \\ (y_{s_L}^{(k)} - y_{u_L}) \\ (z_{s_L}^{(k)} - z_{u_L}) \end{bmatrix} = \begin{bmatrix} -\sin(\angle_{long})(x_s^{(k)} - x_u) + \cos(\angle_{long})(y_s^{(k)} - y_u) \\ -\sin(\angle_{lat})\cos(\angle_{long})(x_s^{(k)} - x_u) - \sin(\angle_{lat})\sin(\angle_{long})(y_s^{(k)} - y_u) + \cos(\angle_{lat})(z_s^{(k)} - z_u) \\ \cos(\phi)\cos(\angle_{long})(x_s^{(k)} - x_u) + \cos(\angle_{lat})\sin(\angle_{long})(y_s^{(k)} - y_u) + \sin(\angle_{lat})(z_s^{(k)} - z_u) \end{bmatrix} \end{gathered} \tag{3.22}$$

Using equations 3.11, 3.19, 3.21, and 3.22, $\tilde{\mathbf{G}}$ can be written as:

$$\tilde{\mathbf{G}} = \begin{bmatrix} -\frac{(x_{s_L}^{(1)} - x_{u_L})}{\|\mathbf{x_s}^{(1)} - \mathbf{x_u}\|} & -\frac{(y_{s_L}^{(1)} - y_{u_L})}{\|\mathbf{x_s}^{(1)} - \mathbf{x_u}\|} & -\frac{(z_{s_L}^{(1)} - z_{u_L})}{\|\mathbf{x_s}^{(1)} - \mathbf{x_u}\|} & 1 \\ -\frac{(x_{s_L}^{(2)} - x_{u_L})}{\|\mathbf{x_s}^{(2)} - \mathbf{x_u}\|} & -\frac{(y_{s_L}^{(2)} - y_{u_L})}{\|\mathbf{x_s}^{(2)} - \mathbf{x_u}\|} & -\frac{(z_{s_L}^{(2)} - z_{u_L})}{\|\mathbf{x_s}^{(2)} - \mathbf{x_u}\|} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{(x_{s_L}^{(k)} - x_{u_L})}{\|\mathbf{x_s}^{(k)} - \mathbf{x_u}\|} & -\frac{(y_{s_L}^{(k)} - y_{u_L})}{\|\mathbf{x_s}^{(k)} - \mathbf{x_u}\|} & -\frac{(z_{s_L}^{(k)} - z_{u_L})}{\|\mathbf{x_s}^{(k)} - \mathbf{x_u}\|} & 1 \end{bmatrix} \tag{3.23}$$

Noting from equation 3.22, that $\mathbf{R}_L$ is a rotation matrix and hence is orthonormal, which implies that the distances are preserved:

$$\parallel \mathbf{x_s}^{(k)} - \mathbf{x_u} \parallel = \parallel \mathbf{x}_{s_L}^{(k)} - \mathbf{x}_{u_L} \parallel \tag{3.24}$$

Using the above equation, equation 3.23 can be re-written as:

$$\tilde{\mathbf{G}} = \begin{bmatrix} -\frac{(x_{s_L}^{(1)} - x_{u_L})}{\|\mathbf{x}_{s_L}^{(1)} - \mathbf{x}_{u_L}\|} & -\frac{(y_{s_L}^{(1)} - y_{u_L})}{\|\mathbf{x}_{s_L}^{(1)} - \mathbf{x}_{u_L}\|} & -\frac{(z_{s_L}^{(1)} - z_{u_L})}{\|\mathbf{x}_{s_L}^{(1)} - \mathbf{x}_{u_L}\|} & 1 \\ -\frac{(x_{s_L}^{(2)} - x_{u_L})}{\|\mathbf{x}_{s_L}^{(2)} - \mathbf{x}_{u_L}\|} & -\frac{(y_{s_L}^{(2)} - y_{u_L})}{\|\mathbf{x}_{s_L}^{(2)} - \mathbf{x}_{u_L}\|} & -\frac{(z_{s_L}^{(2)} - z_{u_L})}{\|\mathbf{x}_{s_L}^{(2)} - \mathbf{x}_{u_L}\|} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{(x_{s_L}^{(k)} - x_{u_L})}{\|\mathbf{x}_{s_L}^{(k)} - \mathbf{x}_{u_L}\|} & -\frac{(y_{s_L}^{(k)} - y_{u_L})}{\|\mathbf{x}_{s_L}^{(k)} - \mathbf{x}_{u_L}\|} & -\frac{(z_{s_L}^{(k)} - z_{u_L})}{\|\mathbf{x}_{s_L}^{(k)} - \mathbf{x}_{u_L}\|} & 1 \end{bmatrix} \tag{3.25}$$

Closely examining terms in equation 3.25 and figure 3.5, the geometry matrix $\tilde{\mathbf{G}}$ can be written in terms of elevation and azimuth angles of the visible satellites as:

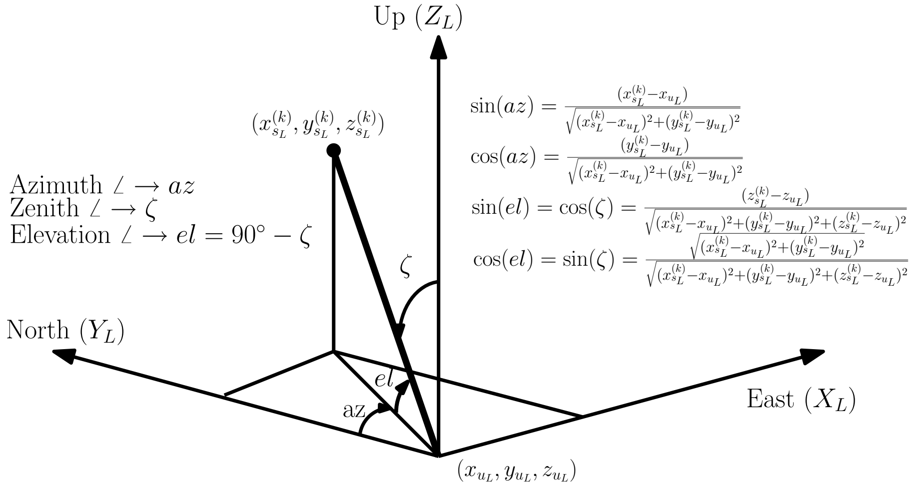


Figure 3.5. Zenith, azimuth and elevation angles defined in LLS

$$\tilde{\mathbf{G}} = \begin{bmatrix} -\cos el^{(1)} \sin az^{(1)} & -\cos el^{(1)} \cos az^{(1)} & -\sin el^{(1)} & 1 \\ -\cos el^{(2)} \sin az^{(2)} & -\cos el^{(2)} \cos az^{(2)} & -\sin el^{(2)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -\cos el^{(k)} \sin az^{(k)} & -\cos el^{(k)} \cos az^{(k)} & -\sin el^{(k)} & 1 \end{bmatrix} \tag{3.26}$$

Denoting $(\tilde{\mathbf{G}}^T \tilde{\mathbf{G}})^{-1}$ by $\tilde{\mathbf{H}}$, equation 3.21 becomes

$$Cov \begin{bmatrix} \mathbf{\Delta x}_{u_L} \\ \delta b \end{bmatrix} = \sigma_{URE}^2 \tilde{\mathbf{H}} \tag{3.27}$$

Equation 3.27 describes the covariance in the user position in terms of the variance in user-satellite range and a matrix $\tilde{\mathbf{H}}$. The matrix $\tilde{\mathbf{H}}$ provides a measure of the effect of satellite geometry on the user position error. The loss of position accuracy due to satellite geometry is termed Dilution Of Precision (DOP) [77]. The lower the DOP, the better the position accuracy. Even cheap GPS receivers provide elevation and azimuth data of visible satellites and thus the matrix $\tilde{\mathbf{H}}$ can be computed. Matrix $\tilde{\mathbf{H}}$, along with a measure of user-satellite range error can then be used to estimate the covariance of user position error.

3.2.3 Velocity Estimation

Changes in the observed frequency of the satellite signal can give information about the relative motion between the satellite and the user. User velocity can also be estimated from the relative motion as the satellite velocity can be computed from information in the GPS sentences. The Doppler shift or the range rate is biased by the receiver clock bias rate and therefore called pseudorange rate. The following analysis in the ECEF-frame shows that user's velocity estimation based on pseudorange rates is identical in structure to the user's position estimation using pseudoranges. Differentiating equation 3.1, a model for pseudorange rates is obtained as (the reference to time (t) has been dropped for brevity) [77]:

$$\dot{\rho}^{(k)} = \dot{r}^{(k)} + (\dot{b}(t) - \dot{b}^{(k)}(t)) + \dot{I}^{(k)} + \dot{T}^{(k)} + \varepsilon_{\dot{\rho}}^{(k)} \tag{3.28}$$

where $\dot{\rho}^{(k)}$ denotes the pseudorange rate which can be obtained from carrier phase measurements. The change in delays due to ionosphere and troposphere is very small over short times and hence can be neglected. The changes in the satellite clock bias $\dot{b}^{(k)}$ are negligible also. $\varepsilon_{\dot{\rho}}^{(k)}$ denotes the effect of modeling errors as well as un-modeled errors. The true range rate $\dot{r}^{(k)}$ can be written as a projection of relative velocity on the line-of-sight vector to the satellite as:

$$\dot{r}^{(k)} = (\mathbf{v_s}^{(k)} - \mathbf{v_u}) \cdot \mathbf{1}^{(k)}(\mathbf{x_s}^{(k)}, \mathbf{x}_{u0}) \tag{3.29}$$

where $\mathbf{v_s}^{(k)}$ is the satellite velocity vector, and $\mathbf{v_u}$ is the user's velocity being estimated. The line-of-sight vector to the satellite, denoted by $\mathbf{1}^{(k)}(\mathbf{x_s}^{(k)}, \mathbf{x}_{u0})$, is calculated using the calculated satellite position and an estimate of user's position. Assuming that the position of the vehicle has already been calculated from pseudoranges, the dependence of $\mathbf{1}^{(k)}$ on $(\mathbf{x_s}^{(k)}, \mathbf{x}_{u0})$ will be dropped from now on. Thus, equation 3.28 can be modified as:

$$\dot{\rho}^{(k)} = (\mathbf{v_s}^{(k)} - \mathbf{v_u}) \cdot \mathbf{1}^{(k)} + \dot{b} + \varepsilon_{\dot{\rho}}^{(k)} \tag{3.30}$$

As the satellite velocity vector is known from the navigation message from the satellite, separating the known and unknown variables in the above equations gives:

$$\underbrace{(\dot{\rho}^{(k)} - \mathbf{v_s}^{(k)})}_{\dot{\tilde{\rho}}^{(k)}} = -\mathbf{v_u} \cdot \mathbf{1}^{(k)} + \dot{b} + \varepsilon_{\dot{\rho}}^{(k)} \tag{3.31}$$

For pseudorange rate measurements from K satellites, the system of equations can be represented in matrix form as:

$$\dot{\tilde{\rho}}^{(k)} = \underbrace{\begin{bmatrix} (-\mathbf{1}^{(1)})^T & 1 \\ (-\mathbf{1}^{(2)})^T & 1 \\ \vdots & \\ (-\mathbf{1}^{(K)})^T & 1 \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} \mathbf{v_u} \\ \dot{b} \end{bmatrix} + \tilde{\varepsilon}_{\dot{\rho}} \tag{3.32}$$

which is of the same form as the linearized system of pseudorange equations used to estimate the user's position in equation 3.11. With the assumption of unbiased, uncorrelated pseudorange rate errors with equal variance equal to σ_v, a similar analysis can be used to characterize the errors in velocity estimation. The final expressions for mean and covariance of errors are given below.

$$
\begin{aligned}
E(\mathbf{\Delta v}_{u_L}) &= E(\hat{\mathbf{v}}_{u_L} - \mathbf{v}_{u_L}) = 0 \\
E(\Delta\dot{b}) &= E(\hat{\dot{b}} - \dot{b}) = 0 \qquad (3.33)
\end{aligned}
$$

$$
Cov \begin{bmatrix} \mathbf{\Delta v}_{u_L} \\ \Delta\dot{b} \end{bmatrix} = \sigma_v^2 \tilde{\mathbf{H}} \qquad (3.34)
$$

Equation 3.34 gives the relation between the covariance of user velocity error and with pseudorange rate error variance through the DOP matrix $\tilde{\mathbf{H}}$. As mentioned with position error estimation covariance analysis, the matrix $\tilde{\mathbf{H}}$ can be calculated using information from low-cost GPS receivers. This matrix, along with a measure of user pseudorange rate error, can be used to calculate vehicle velocity error covariance. Vehicle position and velocity errors are experimentally characterized in the next section for static and dynamic conditions to get estimates of the standard deviations of pseudorange range errors (σ_{URE}) and pseudorange rate errors (σ_v).

3.3 Characterization of GPS Position and Velocity Errors

A typical low-cost GPS receiver operating in autonomous mode has errors in position and velocity estimates due to a number of modeling errors as well as un-modeled effects.

In this section, these errors are characterized first in the static case, where the GPS receiver is placed at a accurately surveyed location and errors in position estimation can be directly calculated. Next, the position and velocity errors are characterized for the dynamic case, where the vehicle is driving at highway speeds and also while undergoing quick dynamic maneuvers.

3.3.1 Static Errors

A low cost GPS receiver module EB-85A, manufactured by ETEK Navigation Inc., with an update rate of 5Hz was used in this study. To characterize the GPS static positioning errors, the GPS receiver was placed at a location used for a GPS base station. The base station location was calibrated to an accuracy of within 5 cm [17]. Data was collected from the GPS receiver for one and a half hours. The dilution of precision (DOP) was also calculated using equation 3.27 and plotted alongside the positioning errors in the East, North and Up directions. Figures 3.6 through 3.8 show the positioning errors, error distributions and DOP values in East, North and Height directions. The time in these plots is expressed in Coordinated Universal Time (UTC), which is a time standard based on international atomic time and is output in GPS sentences.

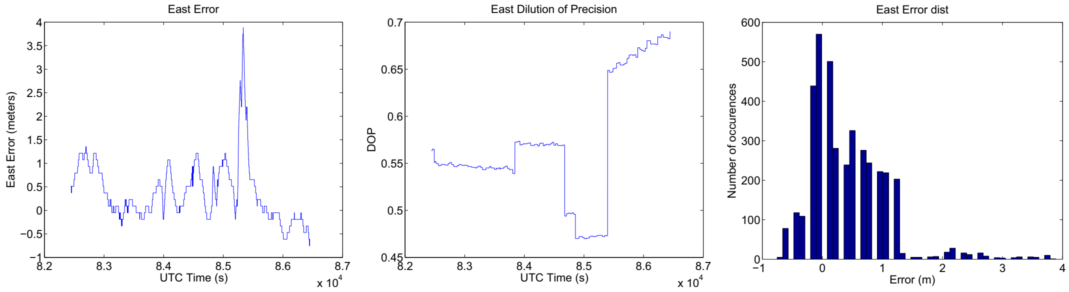


Figure 3.6. Static GPS position error in the East direction

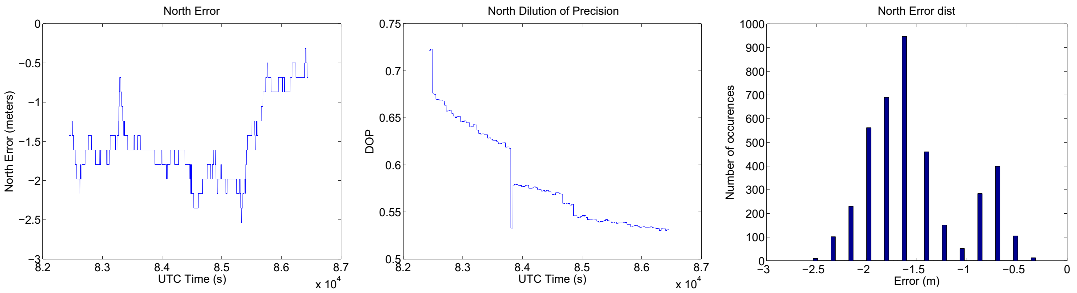


Figure 3.7. Static GPS position error in the North direction

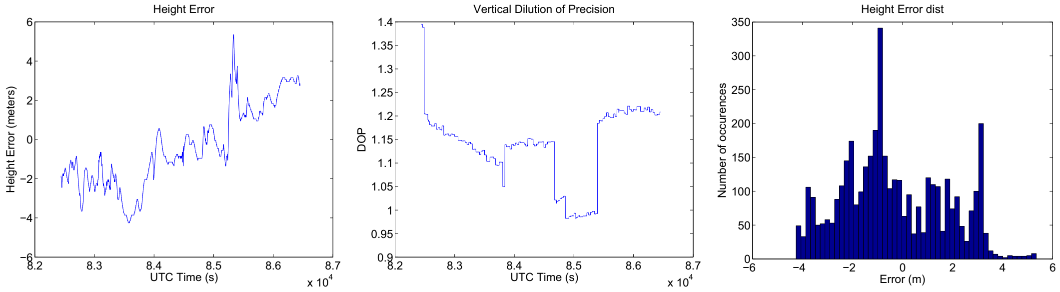


Figure 3.8. Static GPS position error in height

Figure 3.9 shows the satellites which were visible during the collection of the static data along with the sky-view showing the elevation and azimuth of the satellites. The DOP is dependent on the number of visible satellites and the elevation and azimuth of the visible satellites as given by equation 3.27. It can be clearly seen from these figures that changes in the orientation and visibility of the satellites had a direct impact on the DOP, which in turn affects the positioning errors. Looking carefully at figures 3.6 through 3.8 and using equation 3.27, the value of the user range error σ_{URE} is chosen to be equal to 5m. This value agrees with the errors and DOP values in all three directions.

3.3.2 Dynamic Errors

Dynamic positioning tests were carried out to evaluate the effect of motion of the vehicle on positioning errors. This was achieved by putting two GPS receivers, the low-cost re-

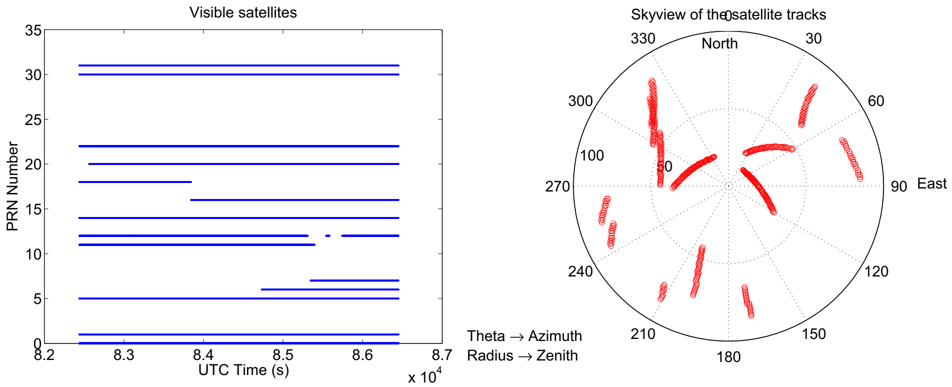


Figure 3.9. Sky-view of satellites during the collection of static data

ceiver and a Novatel DL4-Plus receiver operating in Real-Time-Kinematic (RTK) mode, side by side and comparing the position and velocity solutions while moving the vehicle around the Pennsylvania Transportation Institute test track facility doing quick dynamic maneuvers. The accuracy of the RTK solution is of the order of 2cm in position and 5cm/s in velocity and is taken as the truth value in both cases.

Figures 3.10 through 3.12 show the position in East, North and vertical directions along with the errors, error distributions and dilution of precision. It was seen in the static error analysis that the positioning errors depend on the DOP value over time. DOP values change slowly over time as the GPS satellites move through the sky and can have large changes when satellites come into view or are lost from view. The duration of the dynamic tests shown in the figures is approximately 2 minutes and the DOP values in the horizontal direction do not change appreciably during that interval. It can be seen clearly from the figures that the errors start to vary as soon as the position starts to change. It is also seen that although the errors vary, they remain well within the error ellipsoids defined by the user range error (σ_{URE}) and DOP values as given by equation 3.27. The errors are auto-correlated with a short correlation time. It is possible to model these errors as a Gauss-Markov process, but in the absence of another independent position measurement, these errors are not observable. So, these errors are assumed to be Gaussian in the present work and will be the source of errors in the final estimates.

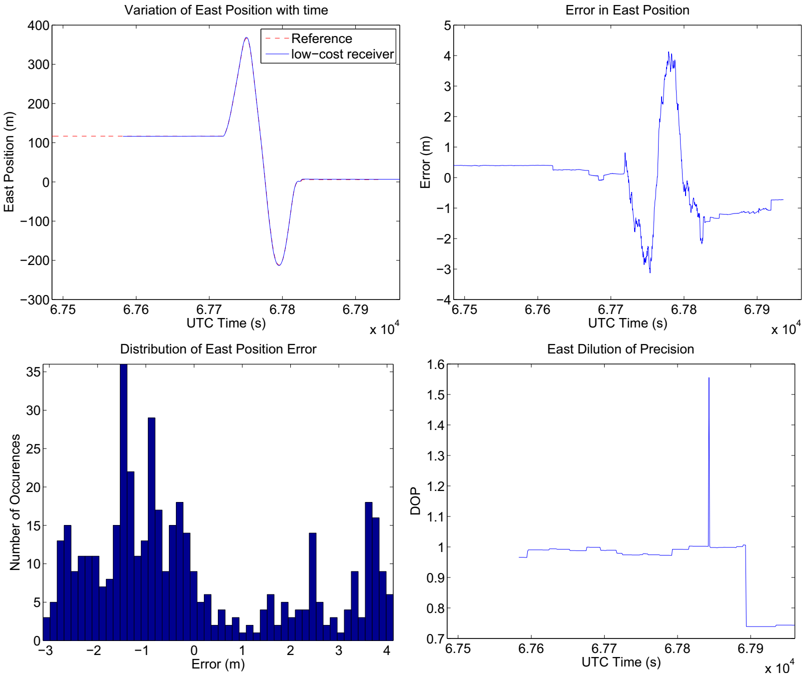


Figure 3.10. Dynamic GPS position error in the East direction

Figure 3.12 shows the variation of height error. Error in height is large relative to the horizontal position errors in this run because of the high DOP value. This can be specifically seen at between UTC time 67700 seconds and 67800 seconds, where there is a spike in the DOP value from 2.6 to 3.8 (approx.) and the error in height corresponding to this interval increases to around 12 meters. This spike in the DOP value occurs due to the temporary loss of GPS satellites as can be seen in figure 3.13. This temporary loss can be attributed to blocking of the line of sight to the satellites during vehicle motion around the test track due to tree lines and small buildings blocking satellites at

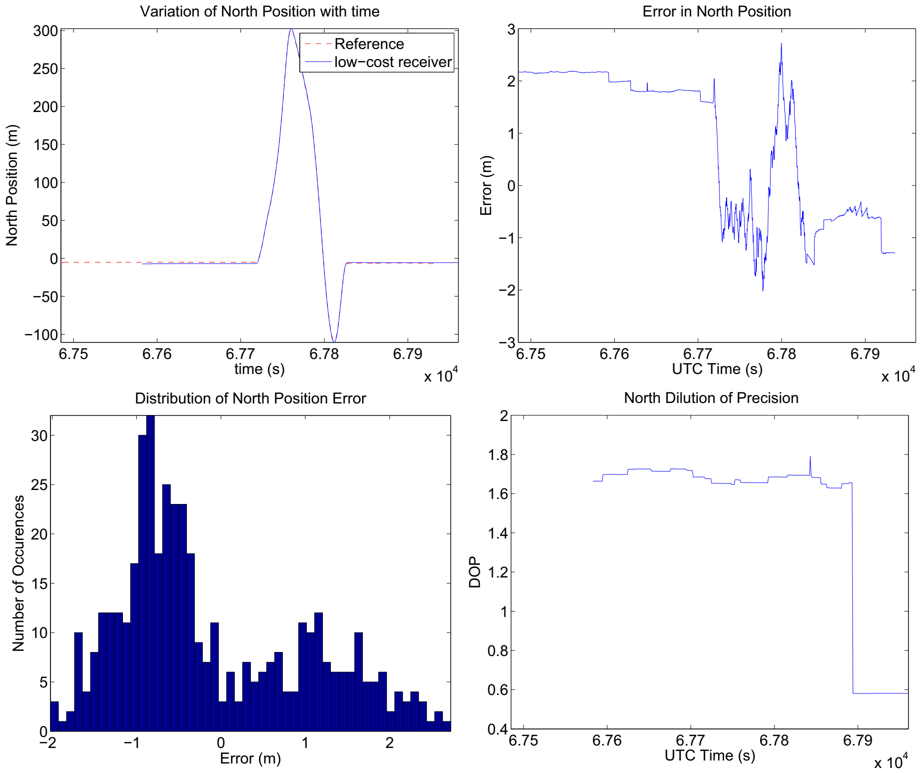


Figure 3.11. Dynamic GPS position error in the North direction

low elevation. Figure 3.13 also shows the sky-view of the satellites during the dynamic test, to give an idea of satellite geometry. As discussed before, the satellite geometry does not change appreciably during the small duration of this test.

Figures 3.14 and 3.15 show the variation of velocity, velocity errors and error distributions in North and East directions. Low-cost receivers do not output the vertical velocity values. Velocity errors are also time correlated, but will be assumed to be Gaussian. The standard deviation (σ_v in equation 3.33) of this Gaussian is chosen to be equal to 50cm/s using the error distributions.

Figure 3.16, which is a zoom in of figures 3.14 and 3.15 when the vehicle just starts to move around UTC time 6.77E4, shows the inability of the GPS to detect low velocities. As shown in the figure, the GPS only starts outputting the velocity solution when the

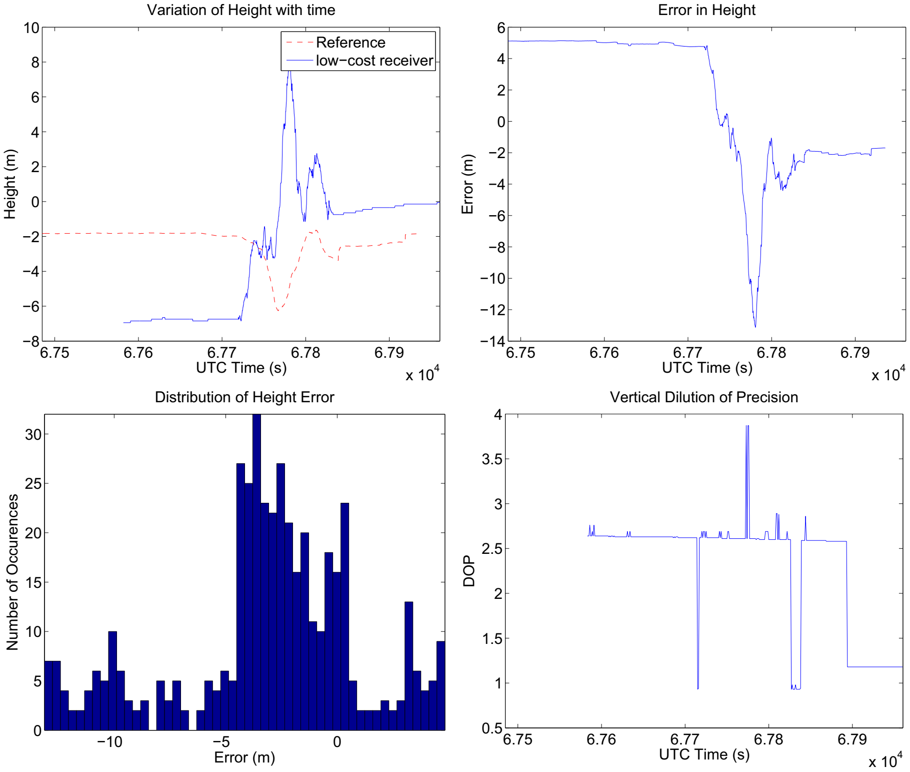


Figure 3.12. Dynamic GPS position error in height

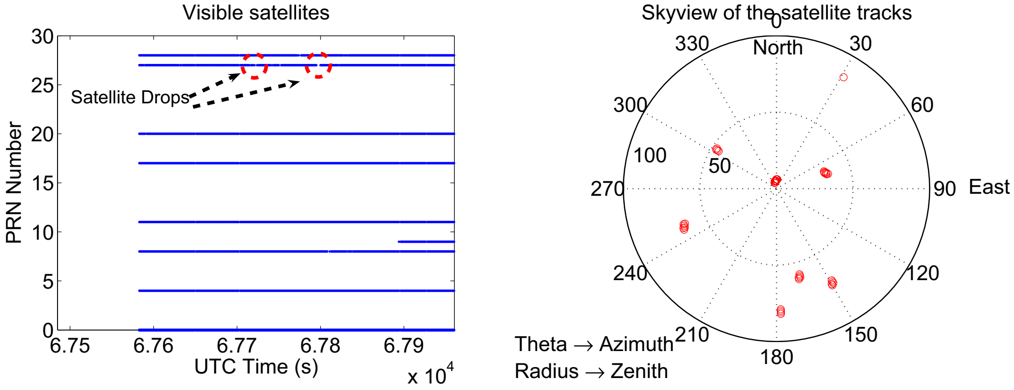


Figure 3.13. Sky-view of satellites during the collection of dynamic data

velocity reaches 1.5 m/s to 2 m/s. When starting from a stop under normal acceleration for day-to-day vehicles (1 to 3 m/s^2), it results in a delay of 1 to 2 seconds before the velocity solution from the GPS becomes useful. The velocity value of the GPS solution needs to be monitored and only used when velocity reaches a value above 2 m/s. This

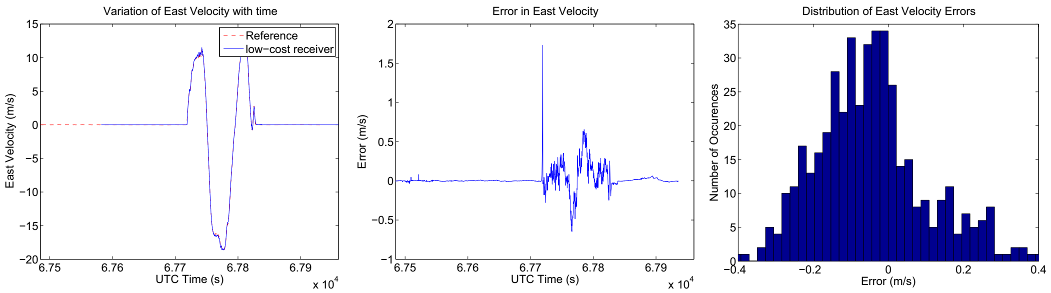


Figure 3.14. Dynamic GPS velocity error in the East direction

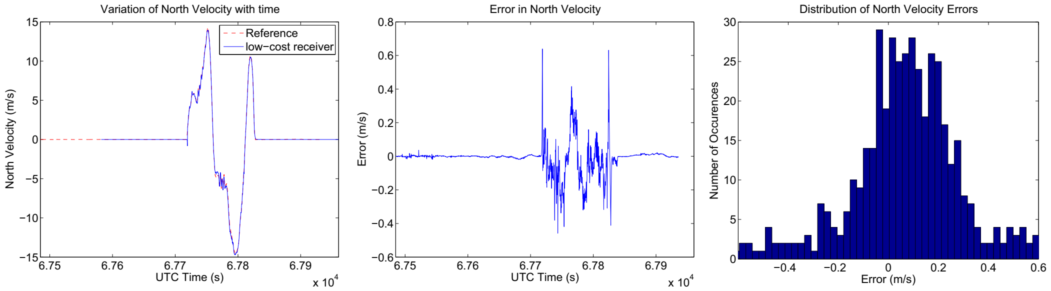


Figure 3.15. Dynamic GPS velocity error in the North direction

is important as it shows that use of a low-cost GPS alone is a very poor solution for low-speed mobile robots.

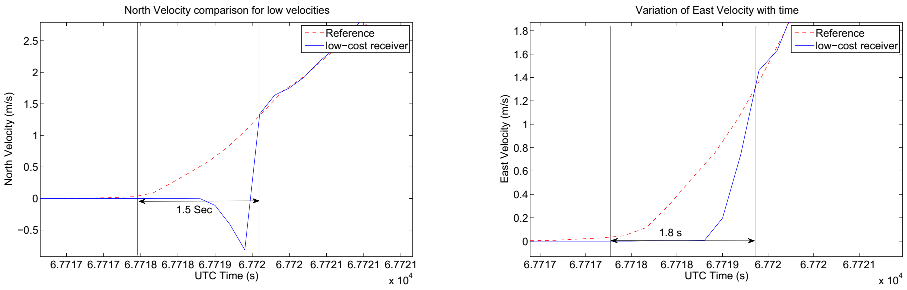


Figure 3.16. Inability of GPS to detect low velocities

3.4 Inertial Navigation System

Inertial navigation system (INS) is a self-contained system which uses the laws of classical Newtonian mechanics to determine the position of the vehicle starting from a known initial position and orientation. The position of the vehicle can be obtained by integrating the vehicle acceleration to obtain velocity and then integrating the velocity to obtain position. For calculating position in this fashion, it is necessary to keep track of the direction in which the accelerometers are pointing. Rotational motion of the vehicle can be sensed with respect to the inertial frame using gyroscopes. Gyroscopes give the rotation rates of the vehicle about the vehicle roll, pitch and yaw axes. By integrating these angular rates in a systematic manner, the orientation of the vehicle can be determined with respect to a given initial orientation. An INS usually contains three accelerometers and three gyroscopes with their sensitive axes orthogonal to each other along the roll, pitch, and yaw axes of the vehicle [9, 13, 107].

Signals from inertial sensors are inherently noisy and so integration over time results in error accumulation. In addition to these integration errors, localization using an INS is complicated by the fact that estimation of position is coupled with the estimation of orientation. Orientation angles are used to resolve the accelerations in the chosen frame of reference. A bias on a gyroscopic measurement gives rise to an error in orientation which increases linearly with time, which in turn gives rise to an error in position which increases as the cube of time. Bias on accelerometers gives rise to position errors which vary as square of time. For this reason, accuracy and noise characteristics of the inertial sensors, gyroscopes in particular, have a large impact on the navigation accuracy of the INS.

Most low-cost GPS receivers only provide the horizontal component of the vehicle velocity which is in the n-frame or ENU frame. Measurements from the IMU can also be quickly resolved from body frame to local geographic navigation frame (n-frame) or

ENU frame. Hence, the n-frame has been used as the frame of reference in this thesis.

3.4.1 Mechanization Equations

Inertial sensors measure the accelerations and angular velocities in the vehicle body frame (b-frame). In order to calculate the vehicle position in the n-frame, it is necessary to first resolve the accelerations along the n-frame axes. The transformation from the n-frame to b-frame can be achieved by three successive rotations about different axes. Following the SAE convention for orientation definition [92], the rotation sequence will be:

- rotation about reference z-axis through an angle ψ $\mathbf{C}_1 = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- rotation about new y-axis through an angle θ $\mathbf{C}_2 = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$
- rotation about new x-axis through an angle ϕ $\mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}$

Following this convention, a transformation from navigation to body axes can be written as:

$$\mathbf{C}_n^b = \mathbf{C}_3\mathbf{C}_2\mathbf{C}_1 \tag{3.35}$$

The inverse transformation from body to reference axes is the transpose of the matrix above (using the ortho-normality of the rotation matrix) and can be written as:

$$\mathbf{C}_b^n = \mathbf{C}_1^T\mathbf{C}_2^T\mathbf{C}_3^T$$

$$= \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta)\cos(\psi) & -\cos(\phi)\sin(\psi)+\sin(\phi)\sin(\theta)\cos(\psi) & \sin(\phi)\sin(\psi)+\cos(\phi)\sin(\theta)\cos(\psi) \\ \cos(\theta)\sin(\psi) & \cos(\phi)\cos(\psi)+\sin(\phi)\sin(\theta)\sin(\psi) & -\sin(\phi)\cos(\psi)+\cos(\phi)\sin(\theta)\sin(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{3.36}$$

The rotational velocities in the body frame can be used to find the rate of change of Euler angles over time using the following equations [9, 13, 107]:

$$\begin{aligned} \dot{\phi} &= (\omega_y \sin(\phi) + \omega_z \cos(\phi)) \tan(\theta) + \omega_x \\ \dot{\theta} &= \omega_y \cos(\phi) - \omega_z \sin(\phi) \\ \dot{\psi} &= (\omega_y \sin(\phi) + \omega_z \cos(\phi)) \sec(\theta) \end{aligned} \tag{3.37}$$

where ω_x, ω_y, and ω_z are the rotational velocities of the vehicle with respect to the navigation frame expressed in the body frame and are represented in the vector form as ω^b_{nb}. The rotation rates given by the gyroscopes are with respect to the inertial frame represented by the vector ω^b_{ib}. The rotation rate of the vehicle body frame with respect to the navigation frame, ω^b_{nb}, can be calculated as:

$$\omega^b_{nb} = \omega^b_{ib} - \mathbf{C}^b_n \omega^n_{in} \tag{3.38}$$

where ω_{in}^{n} is the turn rate of the navigation frame with respect to the inertial frame expressed in navigation frame. Vector ω_{in}^{n} can be computed as follows [13]:

$$\omega_{in}^{n} = \begin{bmatrix} \Omega\cos(\lambda) + \frac{v_E}{R_0+h} \\ -\frac{v_n}{R_0+h} \\ -\Omega\sin(\lambda) - \frac{v_E\tan(\lambda)}{R_0+h} \end{bmatrix} \tag{3.39}$$

Here Ω is the turn rate of the Earth, λ is the longitude at the vehicle position, v_N and v_E are the velocities of the vehicle in the north and east direction, R_0 is the radius of the Earth and h the height of the vehicle above the mean-sea level. In equation 3.38, transformation $\mathbf{C}_n^b$ is applied to convert this rotation rate vector in the body frame to be consistent throughout the equation. The magnitude of ω_{in}^{n} is small as $\Omega \approx 0.00417$deg/s and the term $\frac{v_E}{R_0+h}$ is also small as the velocity of ground vehicles is negligible compared to the radius of the Earth (≈6378.1km). The magnitude of error sources in low-cost IMUs (characterized later) is far too high to be able to sense ω_{in}^{n}. Thus while doing analysis for low-cost IMU's, this term is usually neglected in equation 3.38.

The navigation equation expressing the velocity of the vehicle with respect to Earth can be written as [13, 9]:

$$\dot{\mathbf{v}}_e^n = \mathbf{f}^n - (2\omega_{ie}^n + \omega_{en}^n) \times \mathbf{v}_e^n + \mathbf{g}_l^n \tag{3.40}$$

$$\dot{\mathbf{p}}_e^n = \mathbf{v}_e^n \tag{3.41}$$

where $\mathbf{v}_e^n$ is the velocity of the vehicle with respect to the Earth expressed in the navigation frame. In component from, it can be expressed as:

$$\mathbf{v}_e^n = \begin{bmatrix} v_n & v_e & v_D \end{bmatrix}^T \tag{3.42}$$

$\mathbf{f}^n$ is the vector representing accelerations undergone by the vehicle $\mathbf{f}^b$ resolved into the navigation frame:

$$\begin{aligned} \mathbf{f}^n &= \begin{bmatrix} f_n & f_e & f_D \end{bmatrix}^T \\ &= \mathbf{C}_b^n \mathbf{f}^b \end{aligned} \tag{3.43}$$

ω_{ie}^n represents the Earth rotation rate expressed in the navigation frame.

$$\omega_{ie}^n = \begin{bmatrix} \Omega\cos(\lambda) & 0 & -\Omega\sin(\lambda) \end{bmatrix}^T \tag{3.44}$$

ω_{en}^n, commonly referred to as transport rate [13], is the rotation rate of the navigation frame with respect to the ECEF-frame.

$$\omega_{en}^n = \begin{bmatrix} \frac{v_E}{R_0+h} & -\frac{v_n}{R_0+h} & -\frac{v_E\tan(\lambda)}{R_0+h} \end{bmatrix}^T \tag{3.45}$$

In equation 3.40, $\mathbf{g}_l^n$ is the local gravity vector, which is the sum of Earth's gravitational attraction vector and centripetal acceleration vector caused by Earth's rotation.

$$\mathbf{g}_l^n = \mathbf{g} - \omega_{ie} \times \omega_{ie} \times \mathbf{R} \tag{3.46}$$

In the component form, the navigation equations can be written as follows:

$$\dot{v}_N = f_N - 2\Omega v_E \sin(\lambda) + \frac{v_N v_D - v_E^2 \tan(\lambda)}{R_0 + h} + \xi g \tag{3.47}$$

$$\dot{v}_E = f_E + 2\Omega(v_N \sin(\lambda) + v_D \cos(\lambda)) + \frac{v_E}{R_0 + h}(v_D + v_N \tan(\lambda)) - \eta g \tag{3.48}$$

$$\dot{v}_D = f_D - 2\Omega v_E \cos(\lambda) - \frac{v_E^2 + v_N^2}{R_0 + h} + g \tag{3.49}$$

$$\dot{x_N} = v_N \tag{3.50}$$

$$\dot{x_E} = v_E \tag{3.51}$$

$$\dot{x_D} = v_D \tag{3.52}$$

Here,(x_N, x_E, x_D) are the coordinates of the vehicle in the navigation frame, and ξ and η are the angular deflections from the local vertical in the direction of the local gravity vector because of gravity anomalies. This variation has a maximum magnitude of approximately 30 arc seconds over the surface of the Earth, but will be neglected in further analysis.

3.4.2 Navigation Error Equations

In this section, the growth of errors in the attitude, velocity and position of the vehicle with time is described. This error growth model is useful for fusing the inertial measurements with an aiding sensor, e.g. GPS. Proper fusion of the error model with an aiding sensor keeps these errors small, thus allowing for the use of less computationally expensive fusion algorithms (e.g. a Kalman filter). This error model is widely used in navigation community [9, 13, 107], so only the equations in the final form are presented herein.

Let $\mathbf{\Psi} = [\delta\alpha \ \delta\beta \ \delta\gamma]^T$ be the vector of misalignment angles between true orientation and estimated orientation of the vehicle. If the estimated orientation matrix is denoted by $\tilde{\mathbf{C}}_b^n$ it can be written in terms of the true orientation matrix $\mathbf{C}_b^n$ as:

$$\tilde{\mathbf{C}}_b^n = [\mathbf{I} - \mathbf{\Upsilon}]\,\mathbf{C}_b^n \tag{3.53}$$

where $\mathbf{I}$ is a 3x3 identity matrix and $\mathbf{\Upsilon}$ is given as:

$$\mathbf{\Upsilon} = \begin{bmatrix} 0 & -\delta\gamma & \delta\beta \\ \delta\gamma & 0 & -\delta\alpha \\ -\delta\beta & \delta\alpha & 0 \end{bmatrix} \tag{3.54}$$

The differential equation describing the rate of change of the misalignment angles is [13, 107]:

$$\dot{\mathbf{\Psi}} \approx -\omega_{in}^n \times \mathbf{\Psi} + \delta\omega_{in}^n - \mathbf{C}_b^n \delta\omega_{ib}^b \tag{3.55}$$

ω_{in}^n is the error in the estimation of the vector defined by equation 3.39. $\delta\omega_{ib}^b$ is the error in the rotation rates provided by the IMU.

The error equation for velocity can be expressed as [13, 107]:

$$\delta\dot{\mathbf{v}} = -\mathbf{\Upsilon}\mathbf{C}_b^n\mathbf{f}^b + \mathbf{C}_b^n\delta\mathbf{f}^b - (2\omega_{ie}^n + \omega_{en}^n) \times \delta\mathbf{v} - (2\delta\omega_{ie}^n + \delta\omega_{en}^n) \times \mathbf{v} - \delta\mathbf{g} \tag{3.56}$$

Here, $\delta\mathbf{v} = [\delta v_N \ \delta v_E \ \delta v_D]$ is the vector of velocity estimation errors. $\delta\omega_{ie}^n$ and $\delta\omega_{en}^n$ are the estimation errors in the vectors defined by equation 3.44 and 3.45, respectively. $\delta\mathbf{g}$ is the error in the knowledge of the gravity vector. Errors due to the coriolis terms as well as the error in the gravity model are small and will be ignored henceforth. Eliminating these terms in the above equation gives:

$$\delta\dot{\mathbf{v}} = -\boldsymbol{\Upsilon}\mathbf{C}_b^n\mathbf{f}^b + \mathbf{C}_b^n\delta\mathbf{f}^b \tag{3.57}$$

Error in the vehicle position, $\delta\mathbf{p}$, is a function of the errors in the vehicle velocity and can be written as:

$$\delta\dot{\mathbf{p}} = \delta\mathbf{v} \tag{3.58}$$

3.5 Characterization of INS Errors

The MEMS sensors used in this study are low-accuracy sensors with signal errors which include wide-band noise, bias, scale factor and misalignment errors. In this section, signals from these low-accuracy sensors are first compared with signals from a tactical grade IMU. This analysis helps in determining the dominant error components. A model is chosen to represent the sensor outputs in terms of the true outputs and error states. Allan variance analysis and autocorrelation techniques are then used to find the coefficients for these error models.

3.5.1 Comparison of Low-Cost IMU with a Tactical Grade IMU

Signals from the low-cost IMU are compared to the tactical grade IMU mainly to determine the presence of scale factor errors and run-to-run bias instability. The specifications for the tactical grade IMU (HG1700-AG17) [81] used for comparison are given in table

3.1. As can be confirmed from the table, the values of the scale factor for both accelerometers and gyroscopes are negligible. The run-to-run stability of ring laser gyros is of the order of $0.002^\circ h^{-1/2}$ [95] and can be ignored for comparison purposes. As the scale factors of HG1700-AG17 tactical grade IMU are negligible, the magnitude of scale factor errors for the MEMS IMU sensors can be determined by comparison of the signals, especially at high amplitudes. Run-to-run bias instability can also be characterized the same way by keeping the low-cost IMU stationary and turning the IMU on and off at intervals of 15-20 minutes and noting the variation of the bias.

Table 3.1. Performance Specifications for HG1700-AG17 IMU

Properties	Values
Gyro Input Range	± 1000 degrees/s
Gyro Rate Bias	10 degree/hr
Gyro Rate Scale Factor	150ppm
Angular Random Walk	0.5 degrees/rt hr
Accelerometer Range	± 50 g
Accelerometer Linearity	500 ppm
Accelerometer Scale Factor	300 ppm
Accelerometer bias	3mg

Figure 3.17 shows the comparison data from a low-cost accelerometer (MMA7260Q) from Freescale Semiconductor with a high grade accelerometer used in HG1700 tactical grade IMU. As can be seen from the figure, the two signals closely follow each other and scale factor errors are not apparent even at high magnitudes of the signal. It can be safely deduced from this figure that the accelerometers used in this study do not exhibit significant scale factors. The MEMS accelerometers were turned on and off at regular intervals of 15 minutes and variance in the bias was found to be approximately 0.5 m/s^2.

Figure 3.18 shows a similar comparison between the low-cost gyroscope ADXRS150 from Analog Devices with the tactical grade gyroscope used in HG1700 IMU. A careful

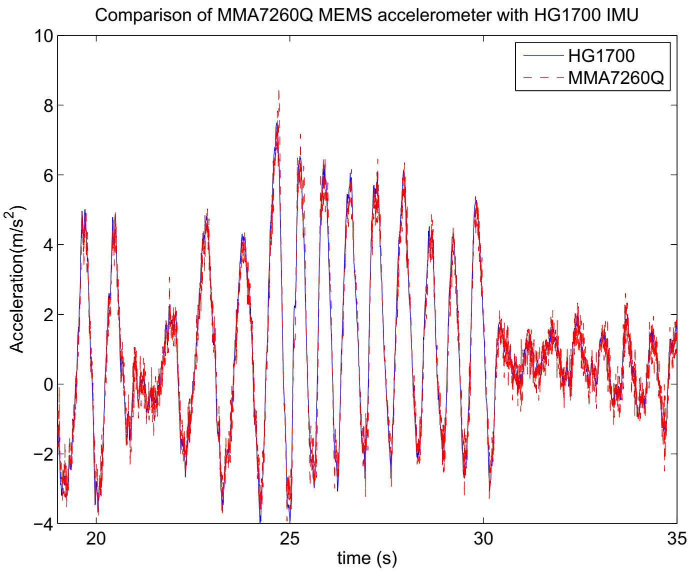


Figure 3.17. Comparison of MEMS accelerometer with high grade accelerometer

look at the figure show the presence of scale factor error. The gyro signal compensated for the scale factor (≈ 0.19)is also plotted in the figure and matches very well with the signal from the tactical grade IMU. Run-to-run bias variance for the gyro was found to be approximately 1 deg/sec.

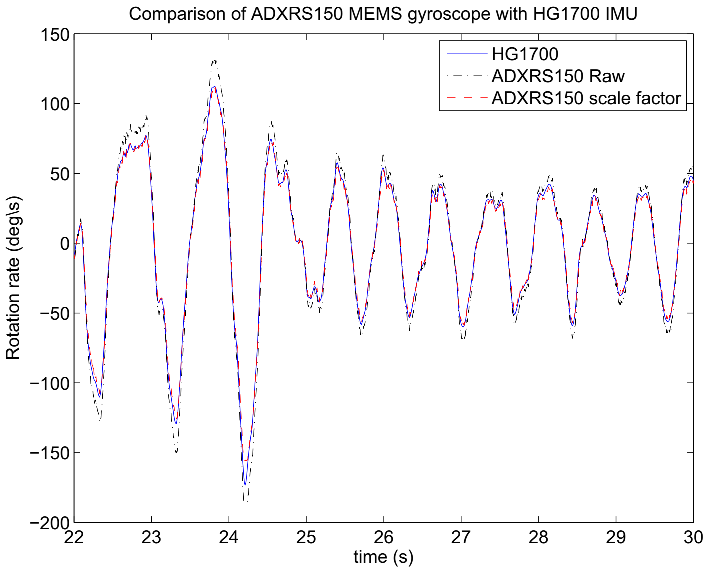


Figure 3.18. Comparison of MEMS gyro with tactical grade gyro

3.5.2 IMU Error Characterization

The Allan variance method [1, 80] was used to characterize IMU error sources. The Allan variance, when plotted on a log-log plot against the correlation time, has different slopes for different error sources, thereby leading to easy identification of an error model. Following the development in [1, 80], the model for the Allan variance, $\sigma_A(\tau)$, can be expressed as a function of correlation time, τ, as:

$$\sigma_A(\tau) = \sum_{n=-2}^{2} A_n \tau^{n/2} \tag{3.59}$$

where the coefficients A_n are obtained by curve fitting on the Allan variance plot in the least mean square sense. 1-σ measure of quantization noise (Q), angle random walk (N), and bias instability (B) can be given by the following expressions:

$$\begin{aligned} Q &= \left(\frac{\pi * 10^6}{180 * 3600 * \sqrt{3}}\right) A_{-2}(\mu rad) \\ N &= \left(\frac{A_{-1}}{60}\right)(deg/hr) \\ B &= (0.6648) A_0 (deg) \end{aligned} \tag{3.60}$$

where Allan variance has units of deg/hr and τ has units of seconds.

Figures 3.5.2 and 3.20 show the Allan variance plots for ring laser and MEMS gyros respectively. The plots are shown along with the curve fit, which yields values for parameters of interest B, N, and Q. The performance difference between the ring laser gyros and MEMS gyros can be clearly seen from the values of these parameters.

Figures 3.21 and 3.22 show the Allan variance plots for HG1700 and MEMS acprovided by a gyroscope can be expressed in terms of the true rotation rate as:

$$\tilde{\omega} = (1 + S_\omega)\omega + b_\omega + \eta_\omega \tag{3.61}$$

Here S_ω is the scale factor error, b_ω is a term representing the low frequency bias, and η_ω is the white noise. Similarly, the acceleration measurement ($\tilde{f^b}$) from the accelerometer is a function of the true acceleration as:

$$\tilde{f^b} = (1 + S_f)f^b + b_{f^b} + \eta_f \tag{3.62}$$

The wide-band (white) noise η in both noise models is assumed to be normally distributed with zero mean and a variance as given by equation 3.63. Various sets of static data from IMU sensors were plotted as histograms to observe the distribution of noise and are shown in figures 3.27 and 3.28 for HG1700 and MEMS IMU's respectively. It can seen clearly that the noise roughly follows a normal distribution. The variance of this wide-band noise can be characterized using the angle random walk noise coefficient N.

$$E[\eta^2] = N^2 \tag{3.63}$$

The low frequency bias b is modeled as an exponentially correlated random variable given by the following differential equation [33]:

$$\dot{b} = -\frac{1}{T_c}b + \eta_b \tag{3.64}$$

Here σ_0^2 as shown in figures 3.24 and 3.26 is the mean square value of the variable.

Scale factor errors are are also modeled as a exponentially correlated random variable with very long correlation time [36]. The correlation time T_{sc} is taken as 20000 seconds and the mean square value σ_{sc}^2 is taken as 1e-4.

$$\dot{S} = -\frac{1}{T_{c_s c}} S + \eta_{sc} \tag{3.66}$$

$$E[\eta_{sc}^2] = 2\frac{\sigma_S^2}{T_{c_s c}} \tag{3.67}$$

3.5.4 Mechanization Equations Revisited

Using the gyro and accelerometer model developed in the last subsection given by equations 3.61 and 3.62, true angular velocities and accelerations needed for mechanization equations 3.37 and 3.40 can be expressed in terms of raw signals from the IMU and IMU error sources. IMU error sources are estimated by fusing IMU and GPS information in a Kalman filter and are used to correct the raw angular velocities and linear acceleration data from IMU sensors for use in mechanization equations. From equation 3.61, the true angular velocity can be written as:

$$\omega = \frac{(\tilde{\omega} - b_\omega)}{(1 + S_\omega)} \tag{3.68}$$

Similarly, the true acceleration can be expressed as:

$$f^b = \frac{(\tilde{f}^b - b_{f^b})}{(1 + S_f)} \tag{3.69}$$

The specific force in navigation frame f^n needed for equation 3.40 can be computed using equation 3.43.

3.5.5 Navigation Error Equations Revisited

The navigation error equations have to be modified to accommodate IMU error sources. In this section, the navigation error equations are extended to include the errors in IMU sensor bias and scale factor estimates.

From equation 3.61, the error in rotation rate error estimate can be expressed as:

$$\begin{aligned} \delta\omega_{ib}^{b} &= \delta(\tilde{\omega} - \omega) \\ &= \omega\delta S_{\omega} + \delta b_{\omega} + \eta_{\omega} \end{aligned} \tag{3.70}$$

Similarly, using equation 3.62, the error in acceleration error estimate can be written as:

$$\begin{aligned} \delta f^{b} &= \delta(\tilde{f}^{b} - f^{b}) \\ &= f^{b}\delta S_{f} + \delta b_{f} + \eta_{f} \end{aligned} \tag{3.71}$$

Substituting the error expressions from equation 3.70 into the orientation error equation 3.55 and neglecting terms ω_{in}^{n} and $\delta\omega_{in}^{n}$ as their magnitude is small compared to the noise present in the system, an updated equation for the misalignment angles is obtained.

$$\dot{\mathbf{\Psi}} \approx -\mathbf{C}_{b}^{n}(\omega\delta\mathbf{S}_{\omega} + \delta\mathbf{b}_{\omega} + \eta_{\omega}) \tag{3.72}$$

On similar lines, equation 3.71 is substituted into equation 3.57 to give an updated equation for velocity errors.

$$\delta\dot{\mathbf{v}} = -\boldsymbol{\Upsilon} C_b^n \mathbf{f}^b + \mathbf{C}_b^n(\mathbf{f^b}\delta\mathbf{S}_f + \delta\mathbf{b}_f + \eta_f) \tag{3.73}$$

In the previous sections, GPS and INS systems have been described along with a characterization of errors in these systems. A Kalman filter framework is developed in the next section which fuses the measurements from these two systems to obtain a optimal estimate of the vehicle position in the least-squares sense.

3.6 GPS INS Integration

A Kalman filter framework is now presented that fuses the information from GPS and the INS. Vehicle orientation, velocity, position, gyro biases, accelerometer biases, gyro scale factors and accelerometer scale factors are chosen as states to be estimated. At each time step, the mechanization equations 3.37 and 3.40 are used to advance the estimates of the states using the current best estimate of errors in these states. The Kalman filter algorithm only works on the errors in the filter states. In the prediction step, the errors in the states are predicted using the governing differential equations modeling the states till the time the next GPS measurement update is available. These equations are presented shortly. The difference between the GPS measurements and the predicted value of those measurements is used as a measurement for the correction step of the Kalman filter.

3.6.1 State Update Equations

At each time step, the state of the system is updated using mechanization equations (3.37,3.40) with corrected IMU measurements (equations 3.68, 3.69) and the IMU error state differential equations (3.64,3.66). These equations are given below in the component form for all of the states.

The orientation Euler angles are updated using equation 3.37 which is reproduced below for easy reference.

$$\begin{aligned}\dot{\phi} &= (\omega_y \sin(\phi) + \omega_z \cos(\phi)) \tan(\theta) + \omega_x \\ \dot{\theta} &= \omega_y \cos(\phi) - \omega_z \sin(\phi) \\ \dot{\psi} &= (\omega_y \sin(\phi) + \omega_z \cos(\phi)) \sec(\theta) \qquad (3.74)\end{aligned}$$

where the angular velocities $\omega_x, \omega_y, \omega_z$ are calculated using equation 3.68.

The velocity and position are updated using equation 3.40. Terms ω_{ie}^n and ω_{en}^n are neglected as they are small compared to noise present in the measurements of the low-cost IMU used in this study. After neglecting these terms, the equation can be written in the component form as:

$$\dot{v}_N = f_N \qquad (3.75)$$

$$\dot{v}_E = f_E \qquad (3.76)$$

$$\dot{v}_D = f_D + g \qquad (3.77)$$

$$\dot{x}_N = v_N \qquad (3.78)$$

$$\dot{x}_E = v_E \tag{3.79}$$

$$\dot{x}_D = v_D \tag{3.80}$$

Here the specific forces are calculated using equation 3.43 reproduced here.

$$\begin{bmatrix} f_n & f_e & f_D \end{bmatrix}^T = \mathbf{C}_b^n \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T \tag{3.81}$$

where f_x, f_y, f_z are calculated using equation 3.69.

The gyroscope and accelerometer biases are updated using equation 3.64, reproduced here.

$$\dot{b}_\omega = -\frac{1}{T_{c_\omega} b_\omega} \tag{3.82}$$

$$\dot{b}_f = -\frac{1}{T_{c_f}} b_f \tag{3.83}$$

where b_ω and b_f are the biases in gyroscopes and accelerometers respectively.

The gyroscope and accelerometer scale factors are updated using equation 3.66:

$$\dot{S}_\omega = -\frac{1}{T_{sc_\omega}} S_\omega \tag{3.84}$$

$$\dot{S}_f = -\frac{1}{T_{sc_f}} S_f \tag{3.85}$$

3.6.2 State Space Model

The state space model for errors in the 21 states being estimated is presented here. The errors in orientation and velocity states are given by equations 3.72 and 3.73. Errors in position are given by equation 3.58. Errors in biases and scale factors are derived by using the δ operator on both sides of equations 3.64 and 3.66, and can be expressed as:

$$\delta \dot{b_\omega} = -\frac{1}{T_{c_\omega}} \delta b_\omega + \eta_{b_\omega} \tag{3.86}$$

$$\delta \dot{b_f} = -\frac{1}{T_{c_f}} \delta b_f + \eta_{b_f} \tag{3.87}$$

$$\delta \dot{S_\omega} = -\frac{1}{T_{sc_\omega}} \delta S_\omega + \eta_{sc_\omega} \tag{3.88}$$

$$\delta \dot{S_f} = -\frac{1}{T_{sc_f}} \delta S_f + \eta_{sc_f} \tag{3.89}$$

Using the equations given above, the state space model of the error states can be written as:

$$\underbrace{\begin{bmatrix} \dot{\mathbf{\Psi}} \\ \delta\dot{\mathbf{v}} \\ \delta\dot{\mathbf{p}} \\ \delta\dot{\mathbf{b}}_\omega \\ \delta\dot{\mathbf{b}}_\mathbf{f} \\ \delta\dot{\mathbf{S}}_\omega \\ \delta\dot{\mathbf{S}}_\mathbf{f} \end{bmatrix}}_{\delta\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & -C_b^n & \mathbf{0} & -\mathbf{\omega^b} C_b^n & \mathbf{0} \\ \mathbf{\Xi} & \mathbf{0} & \mathbf{0} & \mathbf{0} & C_b^n & \mathbf{0} & \mathbf{f^b} C_b^n \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{\mathbf{1}}{\mathbf{T_{c_\omega}}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{\mathbf{1}}{\mathbf{T_{c_f}}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{\mathbf{1}}{\mathbf{T_{csc_\omega}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{\mathbf{1}}{\mathbf{T_{csc_f}}} \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} \mathbf{\Psi} \\ \delta\mathbf{v} \\ \delta\mathbf{p} \\ \delta\mathbf{b}_\omega \\ \delta\mathbf{b}_\mathbf{f} \\ \delta\mathbf{S}_\omega \\ \delta\mathbf{S}_\mathbf{f} \end{bmatrix}}_{\delta\mathbf{x}} \tag{3.90}$$

$$+\underbrace{\begin{bmatrix} -C_b^n & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & C_b^n & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} \eta_\omega \\ \eta_{\mathbf{f}} \\ \eta_{\mathbf{b}_\omega} \\ \eta_{\mathbf{b}_\mathbf{f}} \\ \eta_{\mathbf{sc}_\omega} \\ \eta_{\mathbf{sc}_\mathbf{f}} \end{bmatrix}}_{w} \tag{3.91}$$

where $\boldsymbol{\Xi}$ is given by the following equation:

$$\boldsymbol{\Xi} = \begin{bmatrix} 0 & -f_D & f_E \\ f_D & 0 & -f_N \\ -f_E & f_N & 0 \end{bmatrix} \tag{3.92}$$

The process noise spectral density is given by the following equation:

$$\mathbf{Q(t)} = \begin{bmatrix} \sigma_\omega^\mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_\mathbf{f}^\mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_{\mathbf{b}_\omega}^\mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_{\mathbf{b}_\mathbf{f}}^\mathbf{2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_{\mathbf{sc}_\omega}^\mathbf{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_{\mathbf{sc}_\mathbf{f}}^\mathbf{2} \end{bmatrix} \tag{3.93}$$

The parameter values for this model have been identified using the characterization done earlier in this chapter, and are given in the table 3.2 for easy reference.

Table 3.2. Parameter values for MEMS IMU

Parameter	Value	Parameter	Value
σ_{ω_x}	$5.47\ deg/hr^{-1/2}$	$T_{c_{f_y}}$	1259 s
σ_{ω_y}	$6.02\ deg/hr^{-1/2}$	$\sigma_{0_{b_{f_z}}}$	$0.0039\ m/s^2$
σ_{ω_z}	$6.12\ deg/hr^{-1/2}$	$T_{c_{f_z}}$	823 s
σ_{f_x}	$0.28\ m/s/hr^{-1/2}$	$\sigma_{sc_{\omega_x}}$	0.01
σ_{f_y}	$0.25\ m/s/hr^{-1/2}$	$T_{c_{sc_{\omega_x}}}$	20000 s
σ_{f_z}	$0.27\ m/s/hr^{-1/2}$	$\sigma_{sc_{\omega_y}}$	0.01
$\sigma_{0_{b_{\omega_x}}}$	$167.06\ deg/hr$	$T_{c_{sc_{\omega_y}}}$	20000 s
$T_{c_{\omega_x}}$	1105 s	$\sigma_{sc_{\omega_z}}$	0.01
$\sigma_{0_{b_{\omega_y}}}$	$116.70\ deg/hr$	$T_{c_{sc_{\omega_z}}}$	20000 s
$T_{c_{\omega_y}}$	995 s	$\sigma_{sc_{f_x}}$	0.01
$\sigma_{0_{b_{\omega_z}}}$	$110.22\ deg/hr$	$T_{c_{sc_{f_x}}}$	20000 s
$T_{c_{\omega_z}}$	779 s	$\sigma_{sc_{f_y}}$	0.01
$\sigma_{0_{b_{f_x}}}$	$0.0205\ m/s^2$	$T_{c_{sc_{f_y}}}$	20000 s
$T_{c_{f_x}}$	1343 s	$\sigma_{sc_{f_z}}$	0.01
$\sigma_{0_{b_{f_y}}}$	$0.009\ m/s^2$	$T_{c_{sc_{f_z}}}$	20000 s

3.6.3 Measurement Model

GPS velocity and position measurements are used in the corrector step of the Kalman filter. Low-cost GPS receivers only provide horizontal components of the vehicle velocity. Therefore, only the horizontal components of the velocity along with 3D position of the vehicle are used as measurements.

Denoting the measurements from the GPS by $\mathbf{z_{GPS}}$, the measurement vector can be written in component form as:

$$\mathbf{z_{GPS}} = \begin{bmatrix} v_{N_{GPS}} & v_{E_{GPS}} & p_{N_{GPS}} & p_{E_{GPS}} & p_{D_{GPS}} \end{bmatrix}^T \tag{3.94}$$

where $v_{N_{GPS}}, v_{E_{GPS}}$ are the North and East component of the vehicle velocity and

$p_{N_{GPS}}, p_{E_{GPS}}, p_{D_{GPS}}$ are the North, East and Down coordinates of the vehicle position.

An estimate of these measurements are available from the state update step using mechanization equations denoted by $\mathbf{z_{IMU}}$:

$$\mathbf{z_{IMU}} = \begin{bmatrix} v_{N_{IMU}} & v_{E_{IMU}} & p_{N_{IMU}} & p_{E_{IMU}} & p_{D_{IMU}} \end{bmatrix}^T \tag{3.95}$$

where each term denotes the corresponding estimates of GPS measurements by the IMU mechanization equations. At each measurement update, these measurements are compared to get the measurement differences denoted by $\delta\mathbf{z}$:

$$\delta\mathbf{z} = \begin{bmatrix} v_{N_{GPS}} - v_{N_{IMU}} \\ v_{E_{GPS}} - v_{E_{IMU}} \\ p_{N_{GPS}} - p_{N_{IMU}} \\ p_{E_{GPS}} - p_{E_{IMU}} \\ p_{D_{GPS}} - p_{D_{IMU}} \end{bmatrix} = \begin{bmatrix} -\delta v_N \\ -\delta v_E \\ -\delta p_N \\ -\delta p_E \\ -\delta p_D \end{bmatrix} \tag{3.96}$$

These measurement differences can be expressed in terms of velocity and position error states of the system as:

$$\delta\mathbf{z} = \begin{bmatrix} \mathbf{0}_{2\times3} & -\mathbf{I}_{2\times2} & \mathbf{0}_{2\times1} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & -\mathbf{0}_{3\times2} & \mathbf{0}_{3\times1} & -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times12} \end{bmatrix} \delta\mathbf{x} = \mathbf{H}\delta\mathbf{x} + \eta_{\mathbf{meas}} \tag{3.97}$$

where $\eta_{\mathbf{meas}} = [\eta_{GPS_{v_N}} \; \eta_{GPS_{v_E}} \; \eta_{GPS_{p_N}} \; \eta_{GPS_{p_E}} \; \eta_{GPS_{p_D}}]$ is the measurement noise with variance vector ${\sigma_{\mathbf{GPS}}}^2$ given by equations 3.27 and 3.34 and can be written as:

$$
{\sigma_{\mathbf{GPS}}}^2 = \begin{bmatrix}
\sigma_v^2\tilde{\mathbf{H}}_{\mathbf{1\times1}} & \sigma_v^2\tilde{\mathbf{H}}_{\mathbf{1\times2}} & 0 & 0 & 0 \\
\sigma_v^2\tilde{\mathbf{H}}_{\mathbf{2\times1}} & \sigma_v^2\tilde{\mathbf{H}}_{\mathbf{2\times2}} & 0 & 0 & 0 \\
0 & 0 & \sigma_{URE}^2\tilde{\mathbf{H}}_{\mathbf{1\times1}} & \sigma_{URE}^2\tilde{\mathbf{H}}_{\mathbf{1\times2}} & \sigma_{URE}^2\tilde{\mathbf{H}}_{\mathbf{1\times3}} \\
0 & 0 & \sigma_{URE}^2\tilde{\mathbf{H}}_{\mathbf{2\times1}} & \sigma_{URE}^2\tilde{\mathbf{H}}_{\mathbf{2\times2}} & \sigma_{URE}^2\tilde{\mathbf{H}}_{\mathbf{2\times3}} \\
0 & 0 & \sigma_{URE}^2\tilde{\mathbf{H}}_{\mathbf{3\times1}} & \sigma_{URE}^2\tilde{\mathbf{H}}_{\mathbf{3\times2}} & \sigma_{URE}^2\tilde{\mathbf{H}}_{\mathbf{3\times3}}
\end{bmatrix} \tag{3.98}
$$

where notion $\tilde{\mathbf{H}}_{\mathbf{i\times j}}$ refers to the element in i_th row and j_th column of matrix $\tilde{\mathbf{H}}$ defined in equation 3.27.

The value of the measurement noise variance vector were characterized in the GPS error characterization and the values are given in table 3.3. Matrix $\tilde{\mathbf{H}}$ is calculated using data from GPS at each measurement instant using equations 3.21, 3.23 and 3.27.

Table 3.3. Parameter values for GPS Measurement Errors

Parameter	Value
σ_v	0.5 m/s
σ_{URE}	5 m

3.6.4 The Kalman Filter

A discrete Kalman filter has been used to fuse the GPS and IMU measurements. The state space model of the system given by equation 3.90 has the form:

$$
\dot{\delta}\mathbf{x} = \mathbf{F}\delta\mathbf{x} + \mathbf{G}\mathbf{w} \tag{3.99}
$$

As the data acquisition and processing happens inside the computer in the digital domain, this equation is converted into the discrete form as [27]:

$$\delta \mathbf{x_{k+1}} = \mathbf{\Phi_k} \delta \mathbf{x_k} + \mathbf{\Gamma_k w_k} \tag{3.100}$$

where $\mathbf{\Phi_k}$ is the state transition matrix given by:

$$\mathbf{\Phi_k} = e^{\mathbf{F}T} = \mathbf{I} + \mathbf{F}T + \frac{\mathbf{F^2}T^2}{2!} + \dots \tag{3.101}$$

and $\mathbf{\Gamma_k}$ is given by:

$$\mathbf{\Gamma_k} = \mathbf{G}T + \frac{\mathbf{FG}T}{2} + \cdots \tag{3.102}$$

where T is the sampling time interval and $\mathbf{I}$ is an identity matrix.

The discrete form of the process noise covariance is calculated using the process noise spectral density given by equation 3.93 as:

$$\mathbf{Q_k} = \int_0^T \mathbf{\Phi}(\tau)\mathbf{G}\mathbf{Q}(\tau)\mathbf{G}^T\mathbf{\Phi^T}(\tau)d\tau \tag{3.103}$$

For small sampling interval T, the process noise covariance given by equation 3.103 can be approximated as [27]:

$$\mathbf{Q_k} = \mathbf{\Gamma_k}\frac{\mathbf{Q}}{T}\mathbf{\Gamma_k^T} \tag{3.104}$$

The discrete form of the measurement equation 4.4 is given by:

$$\delta \mathbf{z}_k = \mathbf{H}_k \delta \mathbf{x}_k + \eta_{\mathbf{meas}} \tag{3.105}$$

where $\mathbf{H}_k = H$. The process noise variance is denoted by $\mathbf{R}_k$ and is given by:

$$\mathbf{R}_k = \sigma_{\mathbf{GPS}}{}^2 \tag{3.106}$$

where $\sigma_{\mathbf{GPS}}{}^2$ is given by equation 3.98.

In the prediction step, the error state and corresponding covariance estimate is extrapolated using the following equations [33]:

$$\delta\hat{\mathbf{x}}_k(-) = \mathbf{\Phi}_{k-1}\delta\hat{\mathbf{x}}_{k-1}(+) \tag{3.107}$$

$$\mathbf{P}_k(-) = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}(+)\mathbf{\Phi}_{k-1}^T + \mathbf{Q_{k-1}} \tag{3.108}$$

where $(-)$ notation denotes the predicted state estimates and $(+)$ notation denotes the corrected state estimates for the Kalman filter.

As the sample rate of IMU (200 Hz), is much higher than the rate at which GPS provides position and velocity information (5 Hz), the predictor step is repeated until the next GPS measurement is available.

In the corrector step, the GPS measurements are used to correct the IMU only error estimates. This is accomplished by forming innovations given by equation 3.96, which are compared with the predicted innovations using estimates from the prediction step [33].

$$\delta\hat{\mathbf{x}}_k(+) = \delta\hat{\mathbf{x}}_k(-) + \mathbf{K}_k\left[\delta\mathbf{z}_k - \mathbf{H}_k\delta\hat{\mathbf{x}}_k(-)\right] \tag{3.109}$$

$$\mathbf{P}_k(+) = \left[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right]\mathbf{P}_k(-) \tag{3.110}$$

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1} \tag{3.111}$$

3.6.5 Initial Conditions

Orientation estimates are needed for initial IMU alignment. Averaged accelerometer measurements are used to calculate the initial roll and pitch angles as follows [9, 13]:

$$\phi_0 = -\sin^{-1}\left(\frac{\bar{f}_y^b}{g}\right) \tag{3.112}$$

$$\theta_0 = -\sin^{-1}\left(\frac{\bar{f}_x^b}{g}\right) \tag{3.113}$$

where $\bar{f}_x^b$ and $\bar{f}_y^b$ are the averaged measurements from x and y axis accelerometers respectively and g is the gravity constant. Initial alignment for yaw was achieved by driving the vehicle in straight line and using the averaged vehicle velocity vector angle as the initial yaw angle of the vehicle. The vehicle is stationary at the beginning of the experiment and the position is known from GPS measurements. Run-to-run bias variance characterized in section 3.5 was used in setting the initial variance on gyroscope and accelerometer bias state estimates in the Kalman filter.

3.6.6 State Correction

After each corrector step of the error state Kalman filter, the state of the system obtained after state update step in subsection 3.6.1 is corrected using the estimates from the error state Kalman filter. For orientation, a corrected body to navigation frame transformation is calculated using equation 3.53:

$$\mathbf{C}_n^b = (\mathbf{I} - \Upsilon)^{-1}\,\tilde{\mathbf{C}}_n^b \tag{3.114}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{3.115}$$

Using equation 3.36, the corrected Euler angles for vehicle orientation are calculated as:

$$\begin{aligned} \phi(corr) &= \arctan\frac{a_{23}}{a_{33}} \\ \theta(corr) &= -\arcsin a_{13} \\ \psi(corr) &= \arctan\frac{a_{12}}{a_{11}} \end{aligned} \tag{3.116}$$

Velocity and position states are corrected as follows:

$$\mathbf{v}_e^n(corr) = \mathbf{v}_e^n - \delta\mathbf{v} \tag{3.117}$$

$$\mathbf{p}_e^n(corr) = \mathbf{p}_e^n - \delta\mathbf{p} \tag{3.118}$$

Bias and scale factor states are corrected using the following equation:

$$\mathbf{b}(corr) = \mathbf{b} + \delta\mathbf{b} \tag{3.119}$$

$$\mathbf{S}(corr) = \mathbf{S} + \delta\mathbf{S} \tag{3.120}$$

Results from numerical simulations using this Kalman filter are presented in the next section.

3.7 Numerical Simulations

To verify the correctness and performance of the Kalman filter presented above against known truth values, numerical simulations have been performed for a hypothetical vehicle trajectory. Accelerometer and gyroscope data are extracted from the trajectory and then errors are added to this data according to the IMU models given by equations 3.61 and 3.62. The performance criterion is chosen to be the ability of the filter to estimate vehicle orientation and IMU biases and scale factors.

The simulated trajectory of the vehicle is shown in figure 3.29. It consists of circular turns in opposite directions to provide sufficient excitation for all the IMU error sources to be observable. The velocity of the vehicle is also shown in the figure.

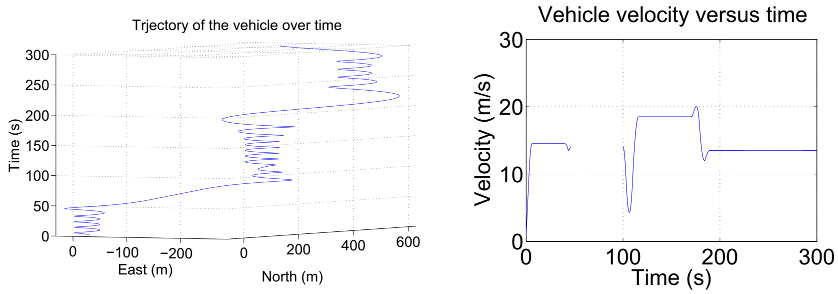


Figure 3.29. Hypothetical trajectory and velocity of the vehicle

Figure 3.30 shows the errors in roll, pitch and yaw angles of the vehicle and the corresponding 1σ error bound estimated by the Kalman filter. It can be seen from the figure that the true errors are well within the bound estimated by the filter. The yaw error plot shows that the true error as well as the estimated error of the yaw angle is high around simulation time 75 seconds and 200 seconds. Taking a look at the vehicle trajectory reveals that the vehicle is traveling straight during those times at constant speed. As the yaw angle is not observable without longitudinal or lateral acceleration [46], error accumulates in the yaw angle estimate while the vehicle is driving straight. As soon as the vehicle starts taking turns, resulting in lateral acceleration, the yaw angle again becomes observable and true and estimated values of yaw error reduce quickly.

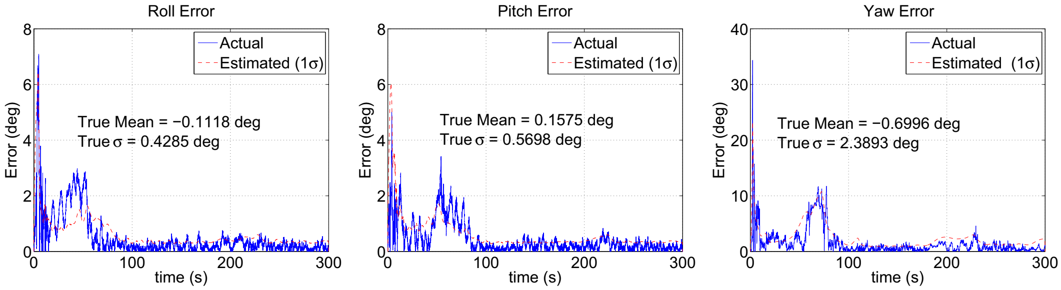


Figure 3.30. Comparison of true errors and estimated 1σ bounds

Figure 3.31 shows the bias estimates of the Kalman filter along with the true values for those biases for gyroscopes and accelerometers. It can be inferred from the figure that the bias estimates converge to the true values for all but z-axis accelerometer. The reason for the z-axis discrepancy is discussed in the next paragraph.

Figure 3.32 shows the scale factor estimation performance of the Kalman filter. It can be seen from the figure that the scale factor values converge to the true values except for the z-axis accelerometer. As the simulated vehicle trajectory did not have sufficient excitation in the z-direction, the scale factor and bias estimates for the z-axis accelerometer were not observable individually. The combination however was observable as can be seen by the fact that the error due to the scale factor estimation error (calculated by multiplying the value of z-axis acceleration (approx. 9.8 m/s^2) with the error in scale factor estimation as shown in figure 3.32) is equal and opposite to the error due to bias

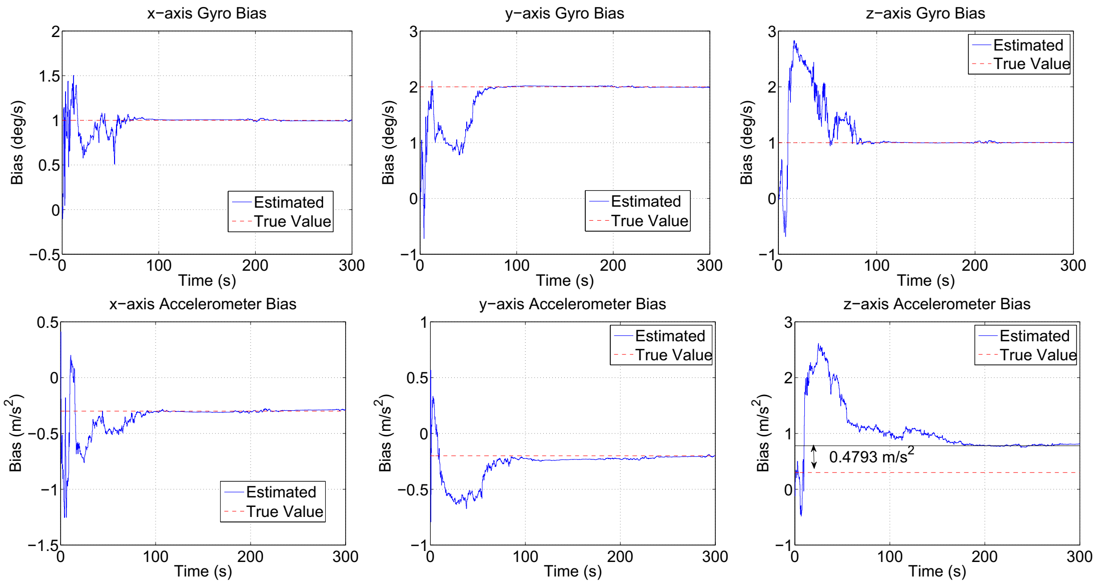


Figure 3.31. Bias estimation performance

estimation error.

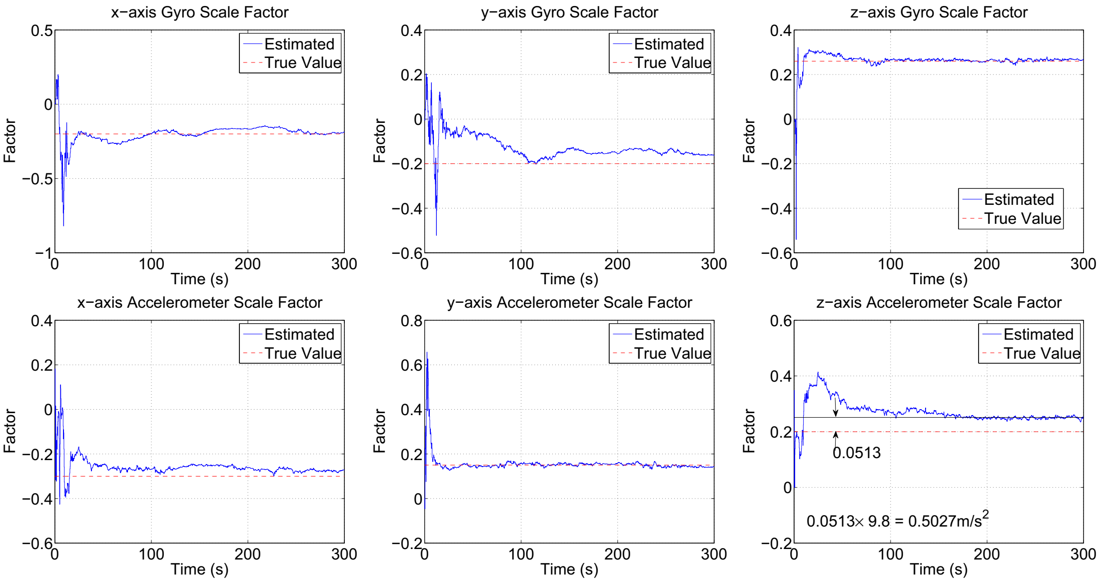


Figure 3.32. Scale factor estimation performance

3.7.1 Yaw Aiding Using GPS Horizontal Velocity Direction

As observed from figure 3.30, vehicle yaw angle is not observable without sufficient longitudinal or lateral acceleration. Low magnitudes of longitudinal and lateral velocities imply that the vehicle is traveling at slowly varying speeds along a trajectory with low radius of curvature. Under these situations, the velocity angle is a very good approximation of the vehicle yaw angle [35] and the loss of observability in yaw angle is not a severe limitation assuming that the vehicle yaw angle corresponds to the direction of travel.

Specifically, the direction of horizontal GPS velocity can be calculated as:

$$v_{GPS}\angle = \arctan(\frac{v_{E_{GPS}}}{v_{N_{GPS}}}) \tag{3.121}$$

Let $\delta v_{N_{GPS}}$ and $\delta v_{E_{GPS}}$ be the errors in the North and East GPS velocity. Also, let $v_{GPS}\tilde{\angle}$ be the velocity direction calculated using GPS velocities with errors. The expression of heading error $\delta v_{GPS}\angle$ can be derived as follows:

$$\begin{aligned}
\delta v_{GPS}\angle &= v_{GPS}\tilde{\angle} - v_{GPS}\angle \qquad (3.122)\\
&= \arctan\left(\frac{v_{E_{GPS}} + \delta v_{E_{GPS}}}{v_{N_{GPS}} + \delta v_{N_{GPS}}}\right) - \arctan\left(\frac{v_{E_{GPS}}}{v_{N_{GPS}}}\right)\\
&\approx \arctan\left(\frac{\delta v_{E_{GPS}} v_{N_{GPS}} - \delta v_{N_{GPS}} v_{E_{GPS}}}{v_{N_{GPS}}^2 + v_{E_{GPS}}^2}\right)\\
&\approx \frac{\delta v_{GPS}}{v_{GPS}} \qquad (3.123)
\end{aligned}$$

Here δv_{GPS} is the error in GPS velocity and $v_{GPS} = \sqrt{v_{N_{GPS}}^2 + v_{E_{GPS}}^2}$ is the horizontal GPS velocity. Using this equation, the standard deviation of the GPS velocity vector can be calculated as:

$$\sigma_{v_{GPS}\angle} \approx \frac{\sigma_{v_{GPS}}}{v_{GPS}} \tag{3.124}$$

When the vehicle is undergoing horizontal acceleration, the yaw angle is observable. So, a term incorporating the magnitude of horizontal vehicle acceleration has been added to the standard deviation of velocity direction to reduce the effect of yaw aiding when the vehicle is undergoing quick dynamic maneuvers or hard accelerations. A constant of multiplication is applied to the term in equation 3.124 to increase the uncertainty for low GPS velocities (under 3m/s). The changed form of 3.124 can be written as:

$$\sigma_{v_{GPS}\angle} \approx K_1 \frac{\sigma_{v_{GPS}}}{v_{GPS}} + K_2 \sqrt{f_n^2 + f_e^2} \tag{3.125}$$

where f_n and f_e are accelerations in the North and East directions. The values of K_1 and K_2 were chosen to be 0.07 and 0.1. The chosen value of K_1 gives an uncertainty (1σ) value of approximately 0.5 degrees using equation 3.125 assuming zero acceleration at a vehicle speed of 3m/s. The chosen value of K_2 adds an uncertainty (1σ) of approximately 6 degrees for every 1 m/s^2 horizontal acceleration of the vehicle.

For yaw aiding, the measurement model becomes:

$$\delta\mathbf{z} = \begin{bmatrix} \psi - v_{GPS}\tilde{\angle} \\ v_{N_{GPS}} - v_{N_{IMU}} \\ v_{E_{GPS}} - v_{E_{IMU}} \\ p_{N_{GPS}} - p_{N_{IMU}} \\ p_{E_{GPS}} - p_{E_{IMU}} \\ p_{D_{GPS}} - p_{D_{IMU}} \end{bmatrix} = \begin{bmatrix} -\delta\gamma - \delta v_N \\ -\delta v_E \\ -\delta p_N \\ -\delta p_E \\ -\delta p_D \end{bmatrix} \tag{3.126}$$

The above equation can be written in terms of error state model as:

$$\delta\mathbf{z} = \begin{bmatrix} \mathbf{0}_{1\times2} & -1 & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times1} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times1} & -\mathbf{I}_{2\times2} & \mathbf{0}_{2\times1} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times2} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times1} & -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times12} \end{bmatrix} \delta\mathbf{x} = \mathbf{H}\delta\mathbf{x} + \eta_{\mathbf{meas}} \tag{3.127}$$

The measurement noise covariance matrix can be modified as:

$$\sigma_{\mathbf{GPS}}{}^{2} = \begin{bmatrix} \sigma^2_{v_{GPS}\angle} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_v^2\tilde{\mathbf{H}}_{\mathbf{1}\times\mathbf{1}} & \sigma_v^2\tilde{\mathbf{H}}_{\mathbf{1}\times\mathbf{2}} & 0 & 0 & 0 \\ 0 & \sigma_v^2\tilde{\mathbf{H}}_{\mathbf{2}\times\mathbf{1}} & \sigma_v^2\tilde{\mathbf{H}}_{\mathbf{2}\times\mathbf{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{URE}\tilde{\mathbf{H}}_{\mathbf{1}\times\mathbf{1}} & \sigma^2_{URE}\tilde{\mathbf{H}}_{\mathbf{1}\times\mathbf{2}} & \sigma^2_{URE}\tilde{\mathbf{H}}_{\mathbf{1}\times\mathbf{3}} \\ 0 & 0 & 0 & \sigma^2_{URE}\tilde{\mathbf{H}}_{\mathbf{2}\times\mathbf{1}} & \sigma^2_{URE}\tilde{\mathbf{H}}_{\mathbf{2}\times\mathbf{2}} & \sigma^2_{URE}\tilde{\mathbf{H}}_{\mathbf{2}\times\mathbf{3}} \\ 0 & 0 & 0 & \sigma^2_{URE}\tilde{\mathbf{H}}_{\mathbf{3}\times\mathbf{1}} & \sigma^2_{URE}\tilde{\mathbf{H}}_{\mathbf{3}\times\mathbf{2}} & \sigma^2_{URE}\tilde{\mathbf{H}}_{\mathbf{3}\times\mathbf{3}} \end{bmatrix} \tag{3.128}$$

Figure 3.33 shows the performance of the Kalman filter with yaw aiding using horizontal GPS velocity direction. It can be clearly seen from the yaw angle estimate that the yaw angle errors have been considerably reduced compared to figure 3.30 where no yaw aiding was used. Vehicle yaw is well estimated even when the vehicle is traveling on approximately straight line paths with constant velocity.

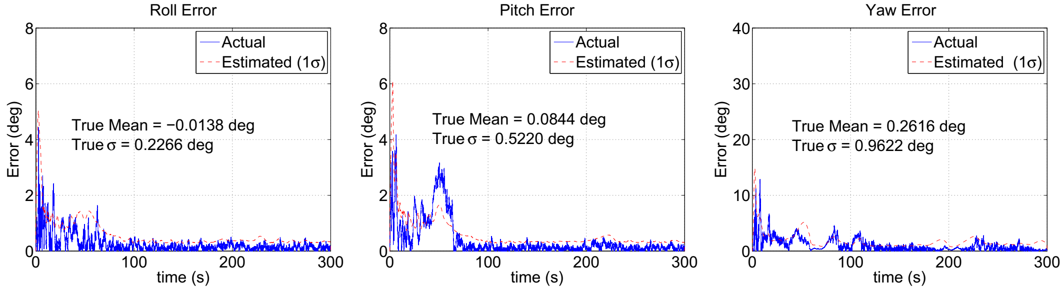


Figure 3.33. Comparison of true errors and estimated 1σ bounds with yaw aiding

In the next section, experimental data collected at the PTI using a Nissan Altima SE sedan instrumented with EB-85 low-cost GPS receiver and a MEMS IMU is processed using the Kalman filter and results compared with the Novatel SPAN system, a high-performance GPS/IMU system.

3.8 Experimental Results

Experiments were conducted at the Pennsylvania Transportation Institute's test track facility where data from the MEMS IMU as well as the low-cost GPS receiver were collected and time stamped using a Digital Signal Processor (DSP) setup. The common time stamp from the DSP makes sure that the data is aligned in time. This data was then processed off-line using the Kalman filter presented above and results compared with a high-performance GPS/IMU system placed side by side.

Data was collected at a sample rate of 200 Hz from the MEMS IMU and at 5 Hz from the low-cost GPS receiver. Results from two test runs for several minutes of data are presented below.

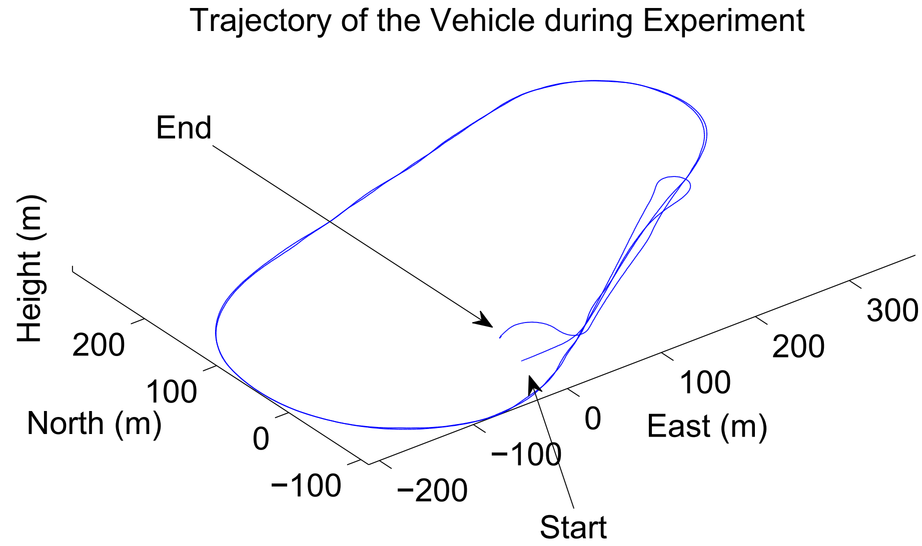


Figure 3.34. Trajectory of the vehicle at the test track: Run 1

Figure 3.34 shows the vehicle trajectory during the course of the experiment. Figure 3.35 shows the comparison of the vehicle orientation between the low-cost and high-performance GPS/IMU systems. It can be seen from the figure that there are large variations of the order of 8 degrees in roll angle, and the low-cost GPS/IMU system is able to track the roll angle profile very well. The variations in pitch angle are relatively small and are of the order of 1 to 2 degrees. Although the pitch angle estimates from the low-cost GPS/IMU system follow the general trend of the pitch angle, the errors in pitch angle as seen in figure 3.36 are of the order of 0.5 to 1 degrees. The yaw angle estimates track the large variations in yaw angle very well except the regions where the vehicle is moving straight and the errors and error estimates tend to grow. This un-observability in the yaw angle while moving straight with low accelerations was discussed in the numerical simulations section and is improved by yaw-aiding as shown later in this section.

Figure 3.36 shows the errors in the orientation angle estimates from the low-cost GPS/IMU system when the estimates from the high-performance GPS/IMU system are taken as the truth value. This assumption is valid as the error estimates from the high-performance GPS/IMU system are of the order of 0.01 degrees and are negligible as compared to the error estimates from the low-cost GPS/IMU system which are of the order of 1 to 2 degrees. The figure also shows the 1σ value of error estimates from the low-cost GPS/IMU system. As shown in the figure, the standard deviation for roll, pitch and yaw errors are 0.681 degrees, 0.465 degrees and 3.459 degrees respectively. By comparing the yaw estimates in figure 3.35 and yaw errors in figure 3.36, it can be seen that the yaw error as well as the error estimate tends to increase when the vehicle is driving straight.

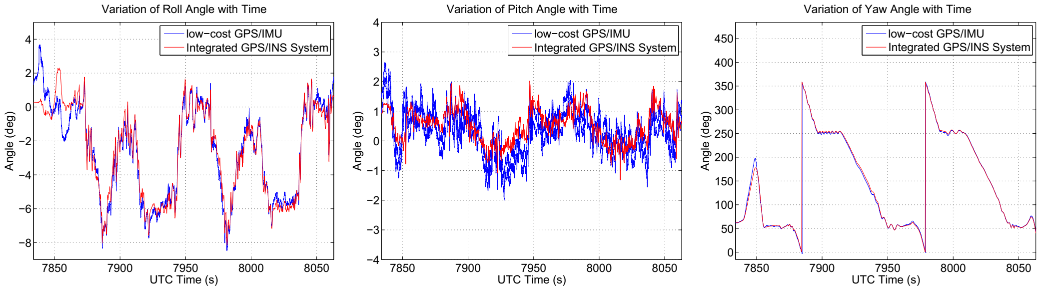


Figure 3.35. Comparison of true and estimated orientation angles: Run 1

Figures 3.37 and 3.38 show the comparison of estimated and true velocity and velocity errors. Again, the measurements from the high-performance GPS/IMU system are taken as truth values. It can be seen from the figures that the North and East velocities are

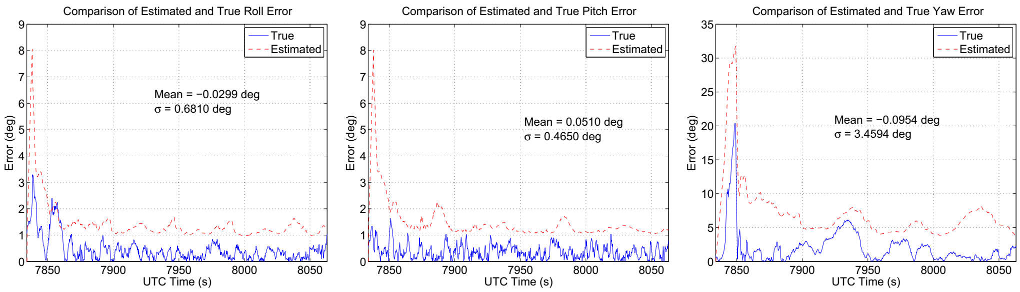


Figure 3.36. Comparison of true and estimated errors in orientation: Run 1

estimated with an accuracy of 0.3m/s and the vertical velocity is estimated with an accuracy of 1m/s approximately. It should be noted that the vertical velocity is not output by the GPS, and is an estimate resulting from the coupling of GPS and IMU systems just like the orientation angles. The 3σ error bound estimated by the Kalman filter is well above the actual error values and is approximately 1m/s for North and East velocities and approximately 2m/s for vertical velocity.

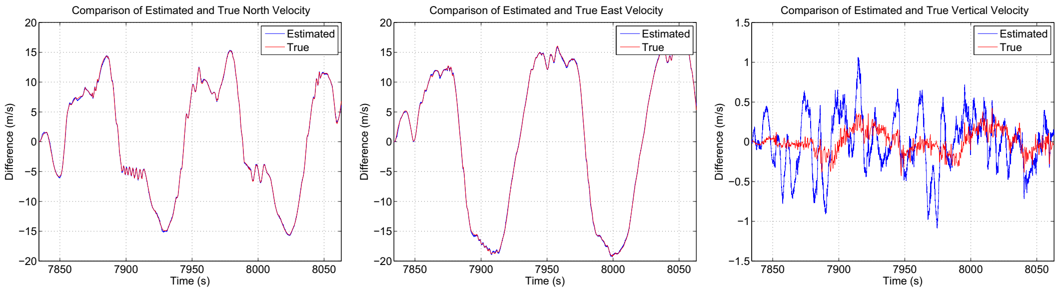


Figure 3.37. Comparison of true and estimated velocity: Run 1

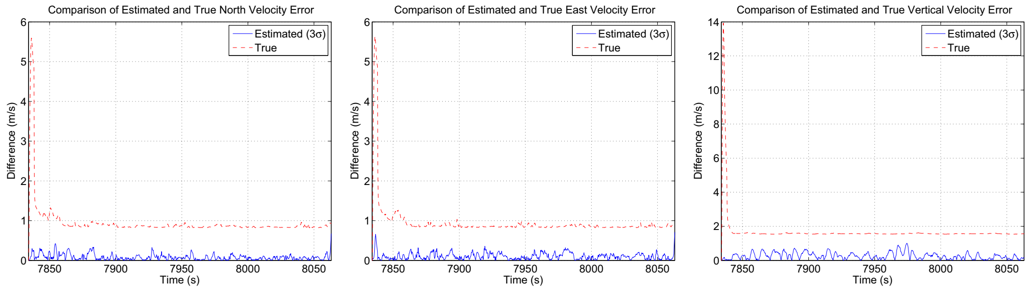


Figure 3.38. Comparison of true and estimated errors in velocity: Run 1

Figures 3.39 and 3.40 show the comparison of estimated and true position and position errors. It can be seen from the figures that the North, East and vertical positions can be calculated with an accuracy of 5m approximately. Figure 3.40 shows that the true position error exceeds the 3σ estimated error over some time intervals. This error is believed to be the result of the assumption that the GPS position and velocity errors are not auto-correlated. It was pointed out in section 3.3.2 that the errors in GPS position and velocity solution are auto-correlated and modeling them as white noise will lead to errors.

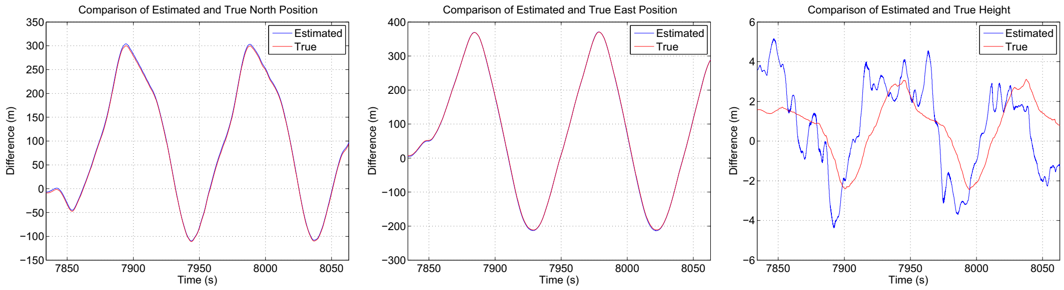


Figure 3.39. Comparison of true and estimated position: Run 1

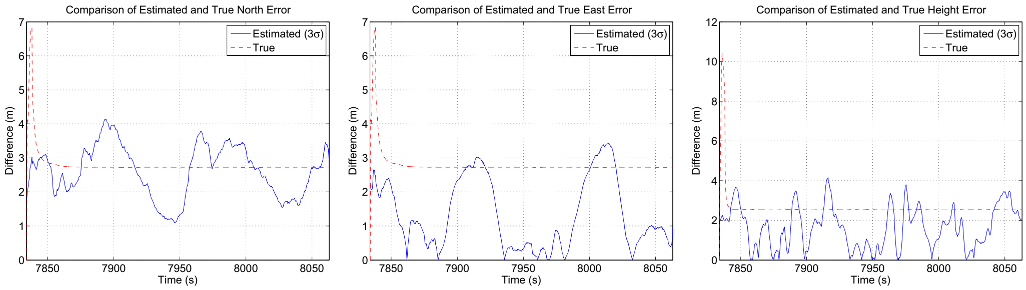


Figure 3.40. Comparison of true and estimated errors in position: Run 1

Figures 3.41 and 3.42 show the gyroscope and accelerometer bias and scale factor estimates which tend to converge to steady state values after about 100 seconds of the start of the experiment. The z-axis accelerometer shows a slight increase in value over time and is coupled with the scale factor estimate which shows a similar increase. The un-observability of z-axis accelerometer bias and scale factor individually was discussed in the numerical simulations section and is observed in experiments too.

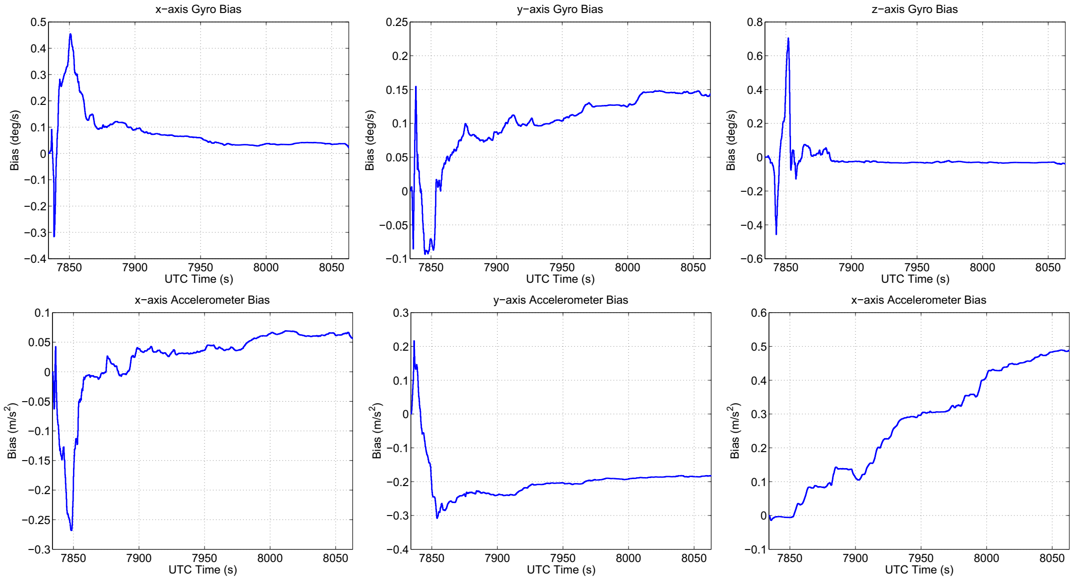


Figure 3.41. Gyroscope and accelerometer bias estimates: Run 1

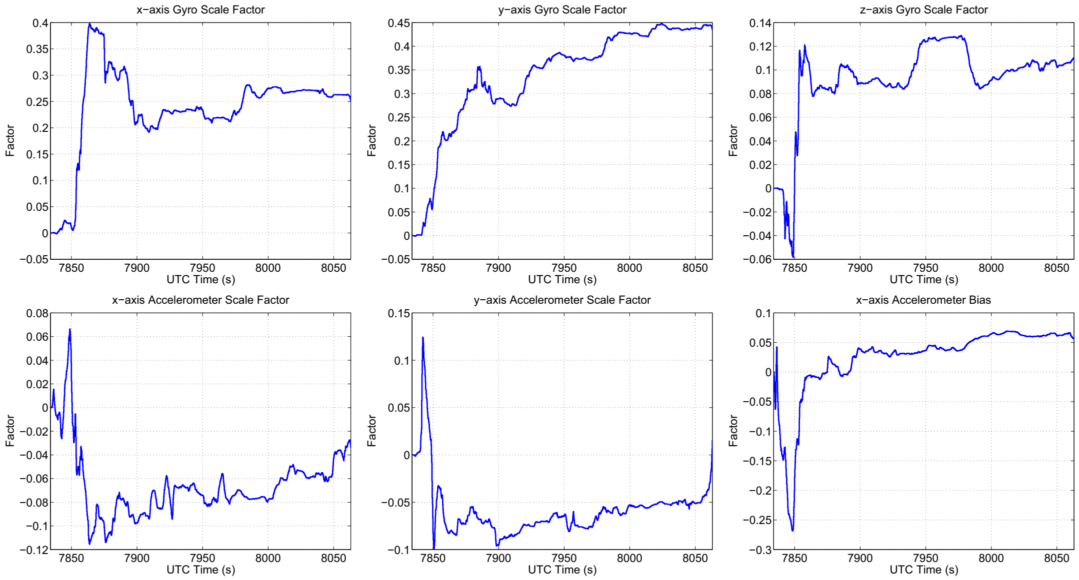


Figure 3.42. Gyroscope and accelerometer scale factor estimates: Run 1

As was discussed in the numerical simulations section, vehicle velocity direction can be used to improve the yaw angle estimates when the vehicle is undergoing low lateral and longitudinal accelerations. Yaw aiding was implemented and the results are shown in figures 3.43 and 3.44. As observed from figures, while the roll and pitch angle estimates do not change appreciably, the yaw angle estimates while the vehicle is driving straight are improved considerably. The figures are drawn to the same scale as figures 3.35 and 3.36 for purposes of comparison. The standard deviation of the yaw angle error has been reduced to 1.297 degrees.

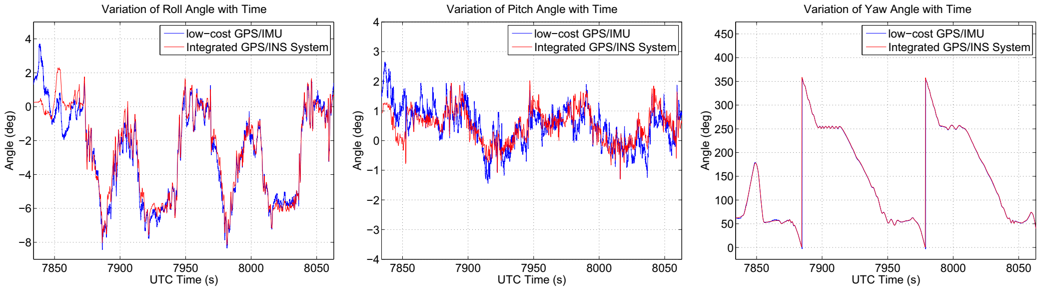


Figure 3.43. Comparison of true and estimated orientation angles with yaw aiding: Run 1

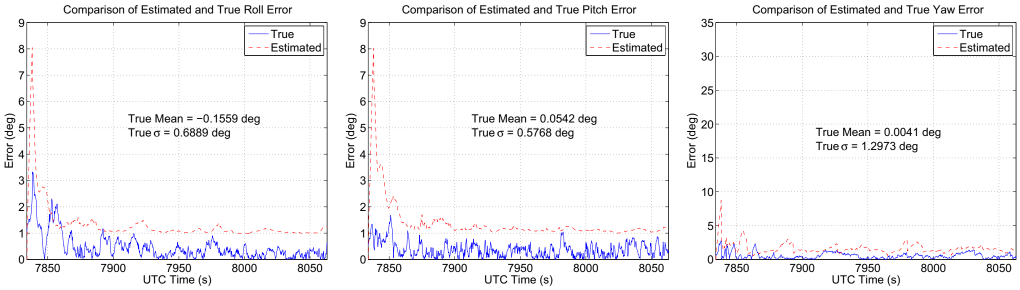


Figure 3.44. Comparison of true and estimated errors in orientation with yaw aiding: Run 1

Figures 3.45 shows the vehicle trajectory during another experiment and figures 3.46 through 3.55 show the results for this run. The improvement in yaw angle estimates for trajectories with low longitudinal and lateral accelerations is shown in figure 3.56. This figure shows a zoomed in view of yaw angle estimates in figures 3.46 and 3.54 on a stretch of vehicle trajectory when the vehicle is going straight. The improvement in the yaw angle estimates can be clearly seen in the figure.

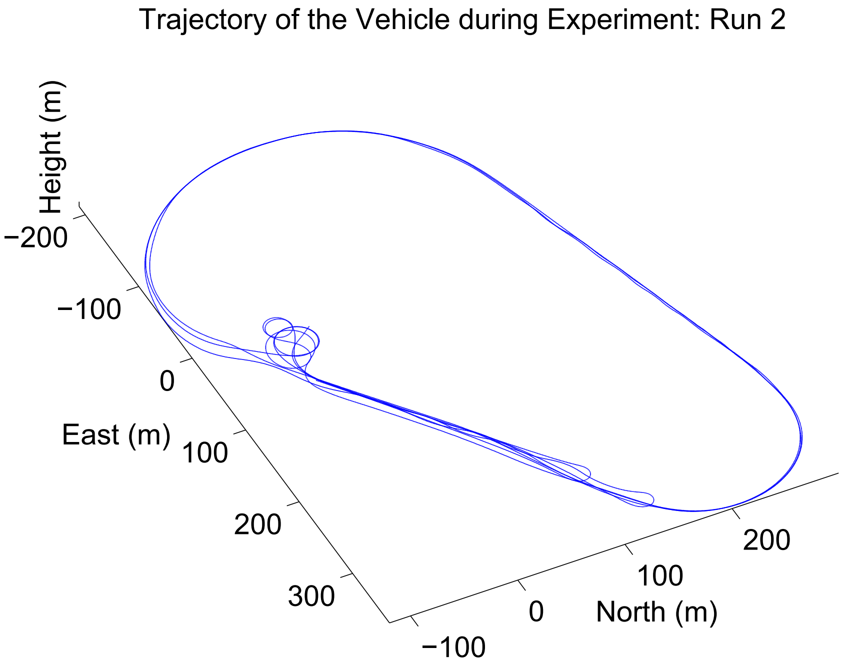


Figure 3.45. Trajectory of the vehicle at the test track: Run 2

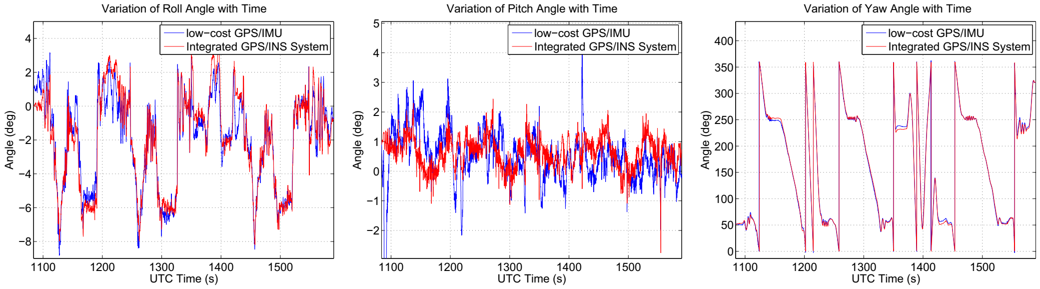


Figure 3.46. Comparison of true and estimated orientation angles: Run 2

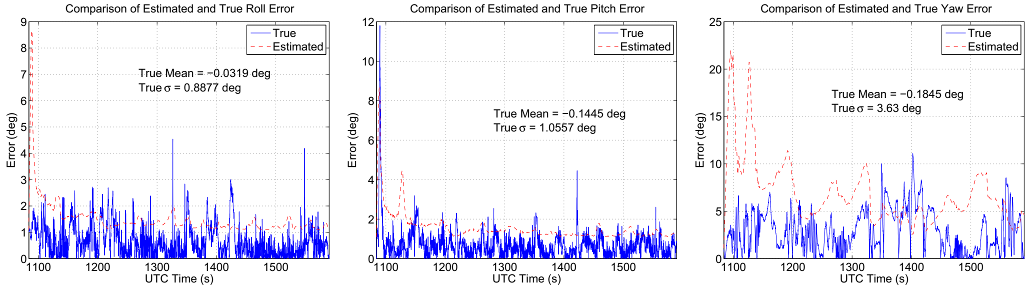


Figure 3.47. Comparison of true and estimated errors in orientation: Run 2

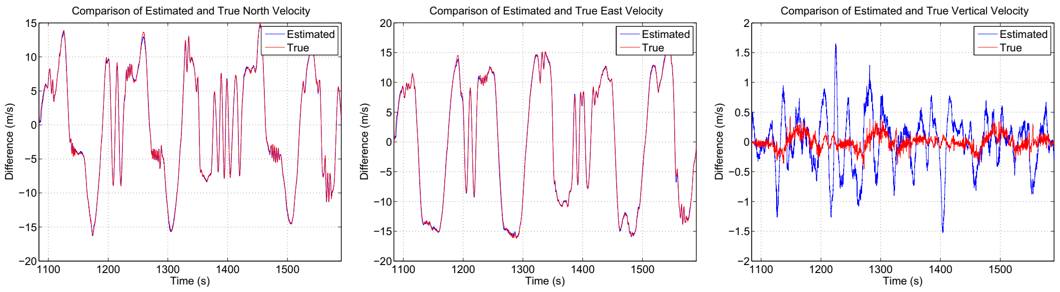


Figure 3.48. Comparison of true and estimated velocity: Run 2

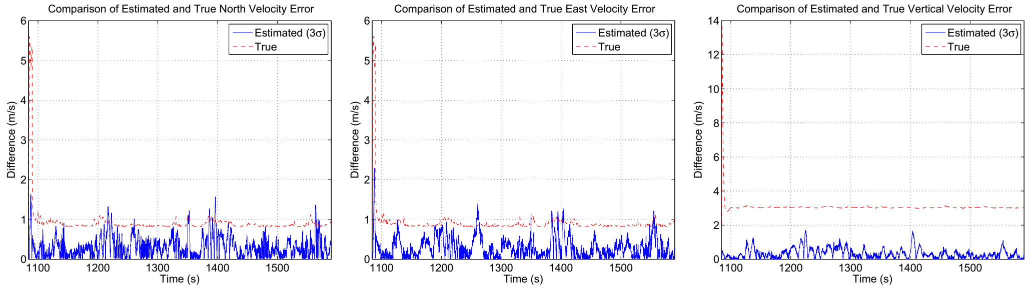


Figure 3.49. Comparison of true and estimated errors in velocity: Run 2

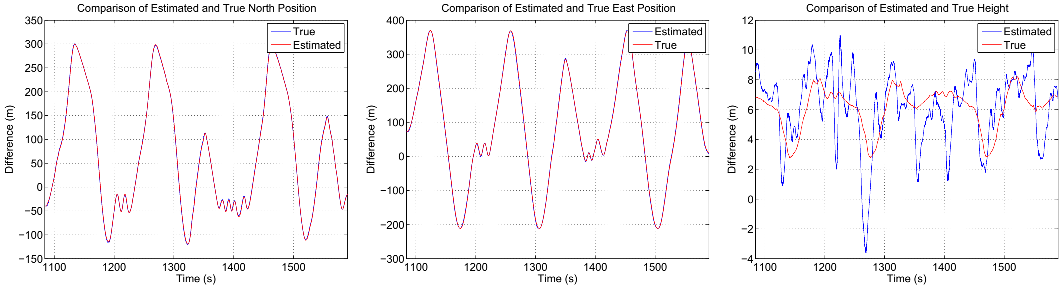


Figure 3.50. Comparison of true and estimated position: Run 2

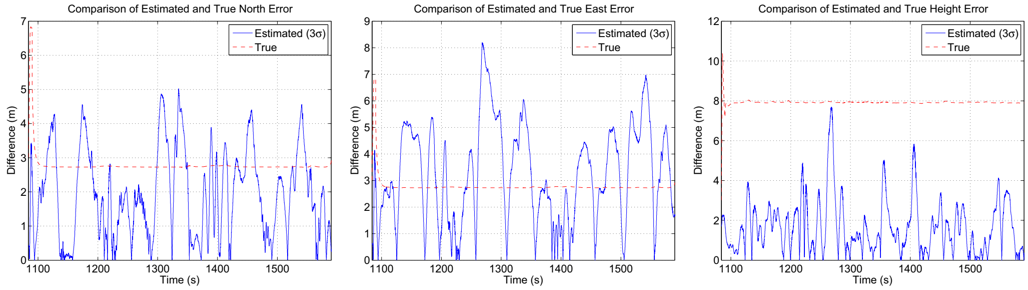


Figure 3.51. Comparison of true and estimated errors in position: Run 2

3.9 Conclusions

This chapter has demonstrated the use of a low-cost MEMS IMU and GPS receiver for vehicle localization. Angular velocity and acceleration data from the MEMS IMU

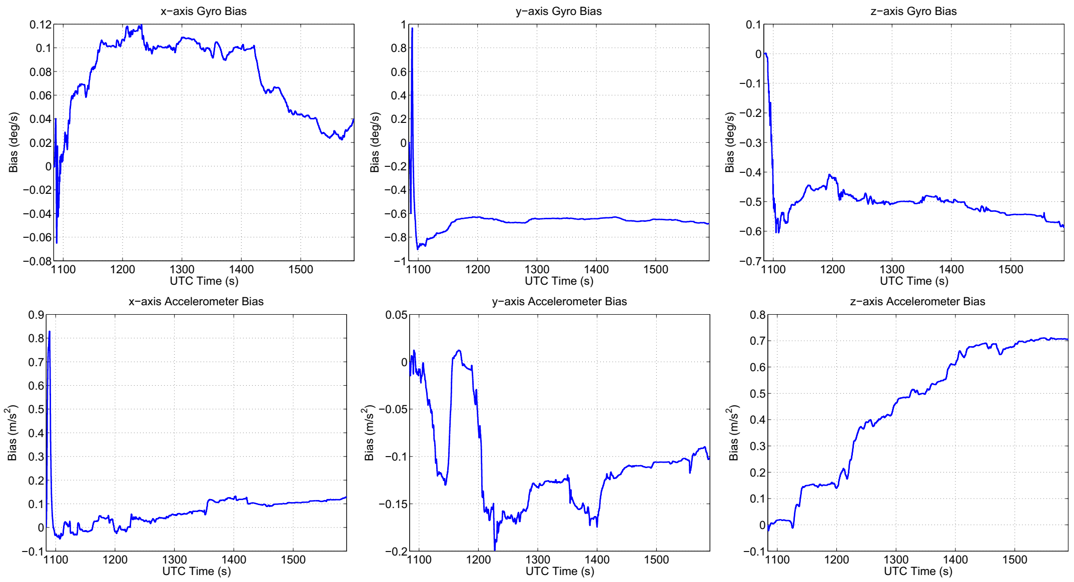


Figure 3.52. Gyroscope and accelerometer bias estimates: Run 2

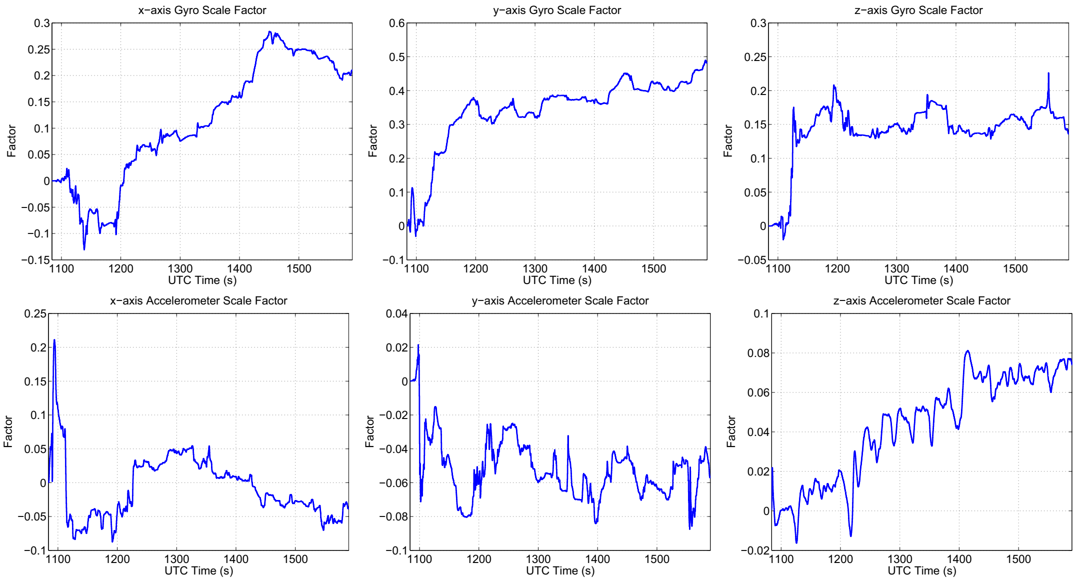


Figure 3.53. Gyroscope and accelerometer scale factor estimates: Run 2

is fused with horizontal velocity and 3D position information from the low-cost GPS receiver using a Kalman filter to estimate vehicle position, orientation and IMU error states. The velocity and position errors of a low-cost GPS receiver are characterized by comparing the measurements from the low-cost GPS receiver with measurements from

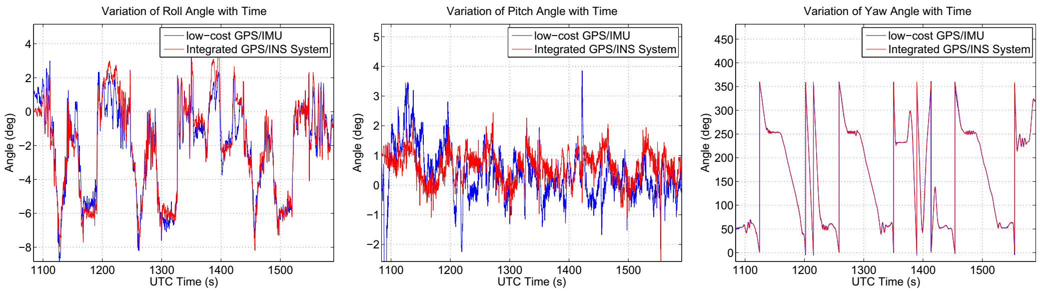


Figure 3.54. Comparison of true and estimated orientation angles with yaw aiding: Run 2

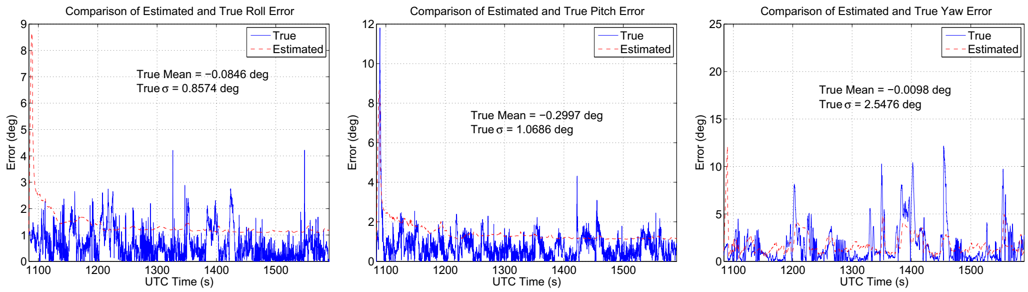


Figure 3.55. Comparison of true and estimated errors in orientation with yaw aiding: Run 2

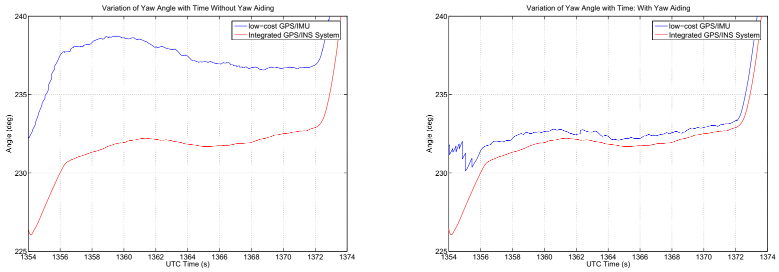


Figure 3.56. Comparison of Yaw angle estimates with and without yaw aiding: Run 2

a high-performance GPS/IMU system. The IMU errors are characterized by using Allan variance and auto-correlation techniques. As the vehicle velocity and position are directly observable, the performance criterion is chosen to be the ability of the Kalman filter to estimate vehicle orientation and IMU biases and scale factors. It was shown by conducting experiments at the Pennsylvania Transportation Institute test track that low-cost GPS/IMU Kalman filter can estimate the roll and pitch angles with 1σ error bound of approximately 1 degree and yaw angle with 1σ error bound of approximately

3.5 degrees. The yaw angle estimate was shown to be improved by using velocity direction as yaw angle measurement. This yaw aiding significantly improves the yaw angle estimates for portions of driving where the vehicle is undergoing low longitudinal and lateral accelerations e.g. straight-line driving. It was shown during experiments that the yaw angle estimate with yaw aiding has a 1σ error bound of approximately 2 degrees.

The work in this chapter shows that the accuracy of vehicle position and orientation estimates is limited by the accuracy of gyroscopes as well as the GPS position, velocity information. The position and orientation estimates can be improved significantly if direct measurements of vehicle orientation are available. As the GPS position and velocity estimates have errors of the order of 5 to 10 meters and 0.1 to 0.5 m/s respectively, performance can be improved markedly if better and independent measurement of vehicle position or velocity can be obtained. Independent measurement of vehicle position and orientation can also provide redundancy during GPS outages. One such method of generating independent orientation measurements is using terrain-aided orientation measurement using vision, and this method is discussed in the next chapter.

Notation

i-frame	The inertial reference frame, p. 26
ECEF	(e-frame)The Earth Centered Earth Fixed reference frame, p. 26
n-frame	The navigation reference frame, p. 26
ENU	East-North-Up reference frame or Local Level System (LLS), p. 26
b-frame	Vehicle body reference frame, p. 26
$\rho^{(k)}(t)$	Pseudorange measurement from the kth satellite, p. 31

c	Speed of light, p. 31
τ	Time of travel of signal from GPS satellites to user GPS receiver, p. 31
$\delta t_u(t)$	User clock offset from GPS Time, p. 31
$\delta t^{(k)}$	Satellite clock offset from GPS Time, p. 31
$I^{(k)}(t)$	Ionospheric propagation delay for the GPS signal, p. 31
$T^{(k)}(t)$	Tropospheric propagation delay for the GPS signal, p. 31
$\varepsilon_\rho^{(k)}(t)$	Modeling error in the pseudorange measurement equation 3.1, p. 31
$\rho_c^{(k)}(t)$	Pseudorange measurement for the kth satellite after correcting for satellite clock offset, p. 32
$\tilde{\varepsilon}_\rho^{(k)}(t)$	Modeling error in the pseudorange measurement equation 3.3 after including tropospheric and ionospheric errors in modeling error, p. 32
$r^{(k)}$	Geometric range from user to the satellite, p. 32
$b(t)$	$= c\delta t_u$ Pseudorange error due to receiver clock bias error, p. 32
$\mathbf{x_u}$	$= (x_u, y_u, z_u)$ Position of the satellite receiver in the ECEF frame, p. 32
$\mathbf{x_s}^{(k)}$	$= (x_s^{(k)}, y_s^{(k)}, z_s^{(k)})$ Position of the kth visible satellite in the ECEF frame, p. 32
$\rho_0^{(k)}$	Pseudorange estimate using an estimate of user position $\mathbf{x}_{u_0}$ and receiver clock bias b_0, p. 33
$\mathbf{x}_{u_0}$	Initial estimate of the GPS receiver position, p. 33
b_0	Initial estimate of the receiver clock bias, p. 33
$\mathbf{x_u}$	True position of the GPS receiver, p. 34
b	True GPS receiver clock bias, p. 34
$\delta\mathbf{x_u}$	Error in the initial estimate of user position by GPS, p. 34
δb	Error in the initial estimate of receiver clock bias, p. 34
$\mathbf{1}^{(k)}$	$= \mathbf{1}^{(k)}(\mathbf{x_s}^{(k)}, \mathbf{x}_{u_0})$ The estimated line-of-sight unit vector from the initial estimate of the receiver position to satellite k, p. 34
$\delta\rho^{(K)}$	Difference between measured and estimated pseudorange for the kth satellite, p. 34
$\tilde{\mathbf{G}}$	GPS satellite geometry matrix, p. 34
$\hat{\mathbf{x}}_\mathbf{u}$	GPS position estimate, p. 35

$\hat{b}$	GPS receiver clock bias estimate, p. 35
σ_{URE}	Standard deviation of user range error, p. 36
$\boldsymbol{\Delta}\mathbf{x}_{\mathbf{u}}$	Error in the GPS position estimate in the ECEF frame, p. 36
Δb	Error in the receiver clock bias estimate, p. 36
$\tilde{\mathbf{H}}$	Matrix depending entirely on GPS satellite geometry and having an effect on user position and velocity error covariances, p. 36
$\angle_{long}$	Longitude at GPS receiver position, p. 37
$\angle_{lat}$	Latitude at the GPS receiver position, p. 37
$\mathbf{R}_L$	Transformation from ECEF frame to ENU frame, p. 37
$\tilde{\mathbf{R}}_L$	Augmented $\mathbf{R}_L$ matrix to include identity transformation from the GPS receiver clock bias term, p. 38
$\mathbf{x}_{s_L}^{(k)}$	Position of the kth satellite in ENU frame, p. 38
$\mathbf{x}_{u_L}$	Position of the GPS receiver in ENU frame, p. 38
el	Elevation of visible GPS satellites, p. 40
az	Azimuth of visible GPS satellites, p. 40
ζ	Zenith of visible GPS satellites, p. 40
$\dot{\rho}^{(k)}$	Rate of change of pseudorange to the kth satellite, p. 41
$\dot{b}^{(k)}(t)$	Rate of change of pseudorange error due to kth satellite clock bias, p. 41
$\dot{b}(t)$	Rate of change of pseudorange error due to receiver clock bias, p. 41
$\varepsilon_{\dot{\rho}}^{(k)}$	Pseudorange rate errors due to modeling errors as well as un-modeled errors, p. 41
$\dot{r}^{(k)}$	True range rate for the kth satellite, p. 42
$\mathbf{v_s}^{(k)}$	Velocity vector for the kth satellite, p. 42
$\mathbf{v_u}$	GPS receiver velocity, p. 42
$\boldsymbol{\Delta}\mathbf{v}_{u_L}$	GPS velocity error in ENU frame, p. 43
$\dot{\Delta b}$	Rate of change of receiver clock bias estimation error, p. 43
σ_v	Standard deviation of user range rate error, p. 43
ϕ, θ, ψ	Euler angles determining vehicle orientation, p. 53

ω^b_{nb}	$= (\omega_x\ \omega_y\ \omega_z)^T$ Vector representing the rotational velocity of the vehicle with respect to the navigation frame expressed in the body frame, p. 53
ω^b_{ib}	Vector representing the rotational velocity of the vehicle with respect to the inertial frame expressed in the body frame, p. 53
ω^b_{in}	Vector representing the rotational velocity of the navigation frame with respect to the inertial frame expressed in the body frame, p. 53
$\mathbf{v}^n_e$	Vector representing the velocity of the vehicle with respect to Earth in the navigation frame. p. 54
$\mathbf{f}^n$	Accelerations experienced by the vehicle resolved into the navigation frame. p. 54
$\mathbf{f}^b$	Accelerations experienced by the vehicle in the body frame p. 54
$\mathbf{C}^n_b$	Transformation from the body frame to the navigation frame. p. 54
$\mathbf{g}^n_l$	Local gravity vector at the vehicle location p. 55
g	Average gravitational acceleration in the vertical direction at the surface of the Earth. p. 55
$\mathbf{p}^n$	$= (x_N\ x_E\ x_D)^T$ Vector representing vehicle position in the navigation frame. p. 54
$\mathbf{v}^n_e$	$= (v_N\ v_E\ v_D)^T$ Vector representing the velocity of the vehicle with respect to Earth in the navigation frame. p. 54
$\tilde{\omega}$	Raw measurement from the gyroscope, p. 65
S_ω	Gyro scale factor, p. 65
b_ω	Gyro bias, p. 65
η_ω	Gyro noise, p. 65
$\tilde{f}^b$	Raw measurement from the accelerometer, p. 65
S_f	Accelerometer scale factor, p. 65
b_{f^b}	Accelerometer bias, p. 65
η_f	Accelerometer noise, p. 65
T_{c_ω}	Time constant for gyroscope first order Gauss-Markov bias model, p. 71
T_{c_f}	Time constant for accelerometer first order Gauss-Markov bias model, p. 71
T_{sc_ω}	Time constant for gyroscope scale factor model, p. 71
T_{sc_f}	Time constant for accelerometer scale factor model, p. 71

$\mathbf{\Psi}$ $= (\delta\alpha\ \delta\beta\ \delta\gamma)^T$ Vector of misalignment angles between true orientation and estimated orientation of the vehicle, p. 57

$\delta\mathbf{v}$ $= (\delta v_N\ \delta v_E\ \delta v_D)^T$ Vector of velocity error states in North, East and Down directions, p. 58

$\delta\mathbf{p}$ $= (\delta p_N\ \delta p_E\ \delta p_D)^T$ Vector of position error states in North, East and Down directions, p. 58

$\delta\mathbf{b}_\omega$ $= [\delta b_{\omega_x}\ \delta b_{\omega_y}\ \delta b_{\omega_z}]^T$ The vector of errors in gyroscope bias estimates in the body frame, p. 72

$\delta\mathbf{b_f}$ $= [\delta b_{f_x}\ \delta b_{f_y}\ \delta b_{f_z}]^T$ Vector of errors in accelerometer bias estimates in the body frame, p. 72

$\delta\mathbf{S}_\omega$ $= [\delta S_{\omega_x}\ \delta S_{\omega_y}\ \delta S_{\omega_z}]^T$ Vector of errors in gyroscope scale factor estimates in the body frame, p. 72

$\delta\mathbf{S_f}$ $= [\delta S_{f_x}\ \delta S_{f_y}\ \delta S_{f_z}]^T$ Vector of errors in accelerometer scale factor estimates in the body frame, p. 72

$\delta\mathbf{x}$ $= (\mathbf{\Psi}\ \delta\mathbf{v}\ \delta\mathbf{p}\ \delta\mathbf{b}_\omega\ \delta\mathbf{b_f}\ \delta\mathbf{b_f}\ \delta\mathbf{S}_\omega\ \delta\mathbf{S_f})^T$ Vector of error states to be estimated using the Kalman filter, p. 72

$\mathbf{\Upsilon}$ Matrix of misalignment angles between true and estimated orientation of the vehicle, p. 69

$\delta\omega_{ib}^b$ vector denoting the error in the rotation rates given by the IMU, p. 57

σ_0 Standard deviation of the Gauss Markov process, p. 64

T_c Time constant of the Gauss Markov process, p. 64

B Bias instability coefficient for IMU sensors, p. 61

N Angle random walk coefficient for IMU sensors, p. 61

Q Quantization constant for IMU sensors, p. 61

$\omega^{\mathbf{b}}$ $= \begin{bmatrix} \omega_x & 0 & 0 \\ 0 & \omega_y & 0 \\ 0 & 0 & \omega_z \end{bmatrix}$ The matrix of corrected gyroscopes measurements calculated using equation 3.68, p. 72

$\mathbf{f^b}$ $\begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & f_z \end{bmatrix}$ Matrix of corrected accelerometer measurements calculated using equation 3.69, p. 72

$\mathbf{T_{c_\omega}}$ $\begin{bmatrix} \frac{1}{T_{c\omega_x}} & 0 & 0 \\ 0 & \frac{1}{T_{c\omega_y}} & 0 \\ 0 & 0 & \frac{1}{T_{c\omega_z}} \end{bmatrix}$ Matrix of correlation times for gyroscope biases, p. 72

$\mathbf{T_{c_f}}$ $= \begin{bmatrix} \frac{1}{T_{c_{f_x}}} & 0 & 0 \\ 0 & \frac{1}{T_{c_{f_y}}} & 0 \\ 0 & 0 & \frac{1}{T_{c_{f_z}}} \end{bmatrix}$ Matrix of correlation times for accelerometer biases, p. 72

$\mathbf{T_{csc_\omega}}$ $= \begin{bmatrix} \frac{1}{T_{csc_{\omega_x}}} & 0 & 0 \\ 0 & \frac{1}{T_{csc_{\omega_y}}} & 0 \\ 0 & 0 & \frac{1}{T_{csc_{\omega_z}}} \end{bmatrix}$ Matrix of correlation times for gyroscope scale factors, p. 72

$\mathbf{T_{csc_f}}$ $= \begin{bmatrix} \frac{1}{T_{csc_{f_x}}} & 0 & 0 \\ 0 & \frac{1}{T_{csc_{f_y}}} & 0 \\ 0 & 0 & \frac{1}{T_{csc_{f_z}}} \end{bmatrix}$ Matrix of correlation times for accelerometer scale factors, p. 72

η_ω $= [\eta_{\omega_x}\ \eta_{\omega_y}\ \eta_{\omega_z}]^T$ Gyroscope noise vector, p. 72

$\sigma^{\mathbf{2}}_\omega$ $= [\sigma^2_{\omega_x}\ \sigma^2_{\omega_y}\ \sigma^2_{\omega_z}]$ Gyroscope noise variance vector, p. 73

$\eta_{\mathbf{f}}$ $= [\eta_{f_x}\ \eta_{f_y}\ \eta_{f_z}]^T$ Accelerometer noise vector, p. 72

$\sigma^{\mathbf{2}}_{\mathbf{f}}$ $= \begin{pmatrix} \sigma^2_{f_x} & 0 & 0 \\ 0 & \sigma^2_{f_y} & 0 \\ 0 & 0 & \sigma^2_{f_z} \end{pmatrix}$ Accelerometer noise variance vector, p. 73

$\eta_{\mathbf{b}_\omega}$ $= [\eta_{b_{\omega_x}}\ \eta_{b_{\omega_y}}\ \eta_{b_{\omega_z}}]^T$ Gyroscope bias noise vector, p. 72

$\sigma^{\mathbf{2}}_{\mathbf{b}_\omega}$ $= \begin{pmatrix} \sigma^2_{b_{\omega_x}} & 0 & 0 \\ 0 & \sigma^2_{b_{\omega_y}} & 0 \\ 0 & 0 & \sigma^2_{b_{\omega_z}} \end{pmatrix}$ Gyroscope bias noise variance vector, p. 73

$\eta_{\mathbf{b_f}}$ $= [\eta_{b_{f_x}}\ \eta_{b_{f_y}}\ \eta_{b_{f_z}}]^T$ Accelerometer bias noise vector, p. 72

$\sigma^{\mathbf{2}}_{\mathbf{b_f}}$ $= \begin{pmatrix} \sigma^2_{b_{f_x}} & 0 & 0 \\ 0 & \sigma^2_{b_{f_y}} & 0 \\ 0 & 0 & \sigma^2_{b_{f_z}} \end{pmatrix}$ Accelerometer bias noise variance vector, p. 73

$\eta_{\mathbf{sc}_\omega}$ $= [\eta_{sc_{\omega_x}}\ \eta_{sc_{\omega_y}}\ \eta_{sc_{\omega_z}}]^T$ Gyroscope scale factor noise vector, p. 72

$\sigma^{\mathbf{2}}_{\mathbf{sc}_\omega}$ $= \begin{pmatrix} \sigma^2_{sc_{\omega_x}} & 0 & 0 \\ 0 & \sigma^2_{sc_{\omega_y}} & 0 \\ 0 & 0 & \sigma^2_{sc_{\omega_z}} \end{pmatrix}$ Gyroscope scale factor noise variance vector, p. 73

$\eta_{\mathbf{sc_f}}$ $= [\eta_{sc_{f_x}}\ \eta_{sc_{f_y}}\ \eta_{sc_{f_z}}]^T$ Accelerometer scale factor noise vector, p. 72

$\boldsymbol{\sigma^2_{sc_f}} = \begin{pmatrix} \sigma^2_{sc_{f_x}} & 0 & 0 \\ 0 & \sigma^2_{sc_{f_y}} & 0 \\ 0 & 0 & \sigma^2_{sc_{f_z}} \end{pmatrix}$ Accelerometer scale factor noise variance vector, p. 73

$\mathbf{z_{GPS}} = [v_{N_{GPS}}\ v_{E_{GPS}}\ p_{N_{GPS}}\ p_{E_{GPS}}\ p_{D_{GPS}}]^T$ Measurements from the GPS. North and East velocity and North, East and Down coordinates of the vehicle position, p. 74

$\mathbf{z_{IMU}} = [v_{N_{IMU}}\ v_{E_{IMU}}\ p_{N_{IMU}}\ p_{E_{IMU}}\ p_{D_{IMU}}]^T$ Estimate of the GPS measurements from the IMU measurements, p. 75

$\delta\mathbf{z} = -[\delta v_N\ \delta v_E\ \delta p_N\ \delta p_E\ \delta p_D]^T$ Measurement vector for the Kalman filter, p. 75

$\eta_{\mathbf{meas}} = [\eta_{GPS_{v_N}}\ \eta_{GPS_{v_E}}\ \eta_{GPS_{p_N}}\ \eta_{GPS_{p_E}}\ \eta_{GPS_{p_D}}]$ GPS measurement noise vector, p. 108

${\sigma_{\mathbf{GPS}}}^2$ GPS measurement noise covariance, p. 76

$\mathbf{\Phi_k}$ State transition matrix, p. 77

T Sampling time of the IMU (5 ms), p. 77

$\mathbf{Q_k}$ Process noise covariance for the Kalman filter. p. 77

$\mathbf{R}_k$ Measurement noise covariance for the Kalman filter. p. 78

$\delta\hat{\mathbf{x}}_k(-)$ Error state predicted estimates of the Kalman filter. p. 128

$\delta\hat{\mathbf{x}}_k(+)$ Error state corrected estimates of the Kalman filter. p. 78

Chapter 4

Orientation Estimation Using Vision and Inertial Sensors

The work presented in this chapter is an extension of terrain-aided localization methods applied to vehicle localization during low-speed driving. As discussed in chapter 3, orientation estimates can be greatly improved if independent measurements of vehicle orientation angles are available. A method using terrain-aided orientation estimation using vision is presented in this chapter using positions provided by the GPS. As discussed in chapter 2, previous algorithms for terrain aided localization are computationally expensive or have drift issues due to lack of an absolute reference. The method presented in this chapter resolves those issues by using efficient curve matching techniques and an absolute terrain reference model such that the orientation estimate of orientation from rate gyros is improved by fusing measurements obtained by matching horizon features. Because the rendered image is generated by using an absolute reference, a DEM, this estimate of vehicle orientation is an absolute estimate, which places an upper bound on the covariance of the error in the orientation estimate.

This chapter is organized as follows. Section 4.1 uses a perspective camera model to develop formulae to estimate the camera movement if the correspondence between image features are known. Section 4.2 details the method to generate the 3D representation of the environment. In Section 4.3, an IMU measurement model is developed and IMU error sources are characterized. Section 4.4 describes image alignment algorithms that are used to recover the error in the estimated camera position. A kinematic Kalman filter is developed using vision and inertial measurements in Section 4.5. Results from numerical simulations are presented in Section 4.6 that verify the performance of the Kalman filter. Experimental performance results are given comparing state estimates to the results obtained by using GPS/IMU Kalman filter in Section 4.7. Conclusions are drawn in the end along with a discussion of limitations and possible improvements of this work.

4.1 Obtaining Orientation Deviations from Pixel Correspondences in Two Images in Space

This section develops formulae to estimate deviations in vehicle orientation if the correspondence between features in real and rendered images is known. The transformations presented here are similar to, but simpler than, the work by [47]. One can look at this problem as trying to find the translation and rotation of the camera which is necessary to transform the view as seen in a rendered image to the one seen in the real image. This situation is again equivalent to assuming that the camera is stationary and applying a rotation and translation to the world points to emulate the camera motion. In this analysis, the world and camera coordinate systems are assumed to be right hand rectangular coordinate systems. It is also assumed that the world points have already been converted into the camera coordinate system.

More formally, if a point $p_1 = (x_1,y_1,z_1)$ in one image corresponds to a point $p_2 = (x_2,y_2,z_2)$ in another image, the following relation between p_1 and p_2 applies:

$$\begin{aligned}\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} &= \boldsymbol{R} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \boldsymbol{t} \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}\end{aligned} \tag{4.1}$$

where $\boldsymbol{R}$ represents rotation and $\boldsymbol{t}$ represents translation. The rotation matrix $\boldsymbol{R}$ can be specified in a number of equivalent ways, but we will follow a method used in vehicle dynamics [92], where $\boldsymbol{R}$ is specified as three successive rotations around z, y and x-axis (yaw, pitch and roll) by angles ψ, θ and ϕ respectively. So, as presented earlier, $\boldsymbol{R}$ can be written as the product of three separate rotations:

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

Simplifying the above equation:

$$\boldsymbol{R} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ -\sin(\psi)\cos(\phi)+\sin(\phi)\sin(\theta)\cos(\psi) & \cos(\phi)\cos(\psi)+\sin(\phi)\sin(\theta)\sin(\psi) & \sin(\phi)\cos(\theta) \\ \sin(\phi)\sin(\psi)+\sin(\theta)\cos(\phi)\cos(\psi) & -\sin(\phi)\cos(\psi)+\sin(\theta)\sin(\psi)\cos(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{4.3}$$

To present the measurement method mathematically, let $(\phi_0, \theta_0, \psi_0)$ be the initial estimate of the vehicle orientation and let $(d_\phi, d_\theta, d_\psi)$ be the orientation deviations estimated using curve matching techniques between the two different images. Then the measurement model can be written as:

$$R(\phi_m, \theta_m, \psi_m) = R(d_\phi, d_\theta, d_\psi)R(\phi_0, \theta_0, \psi_0) \tag{4.4}$$

Where $(\phi_m, \theta_m, \psi_m)$ denote the measured vehicle orientation.

The goal of this mathematical analysis is to obtain a method to estimate the deviation in orientation $(d_\phi, d_\theta, d_\psi)$ of the vehicle at each measurement time step using real and rendered images and known feature correspondences. It can be safely assumed that these angular deviations are small (frame rate ≈ 30fps). With this assumption, the approximations $\cos(d_\phi) \approx 1, \cos(d_\theta) \approx 1, \cos(d_\psi) \approx 1$, $\sin(d_\phi) \approx d_\phi, \sin(d_\theta) \approx d_\theta$ and $\sin(d_\psi) \approx d_\psi$ can be made such that, one obtains

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & d_\psi & -d_\theta \\ -d_\psi & 1 & d_\phi \\ d_\theta & -d_\phi & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{4.5}$$

Using the perspective projection model of a camera shown in Fig. 4.1, the image coordinates (u_1, v_1) of the point $p_1(x_1,y_1,z_1)$ are given by

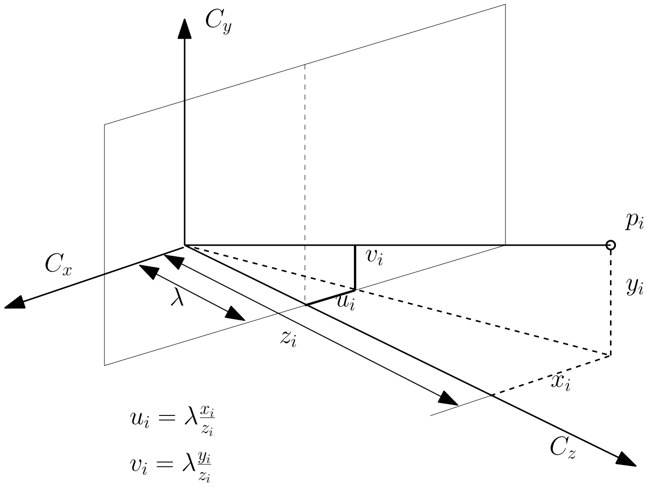


Figure 4.1. Perspective Projection Model

$$u_1 = \lambda\frac{x_1}{z_1}, \quad v = \lambda\frac{y_1}{z_1} \tag{4.6}$$

where λ is the focal length. Using the same model, the image coordinates (u_2, v_2) of the point $p_2(x_2,y_2,z_2)$ are given by

$$u_2 = \lambda\frac{x_2}{z_2}, \quad v_2 = \lambda\frac{y_2}{z_2} \tag{4.7}$$

Using values of x_2, y_2 and z_2 from equation (4.5), u_2 and v_2 can be written as:

$$\begin{aligned} u_2 &= \lambda\frac{x_1 + d_\psi y_1 - d_\theta z_1 + t_x}{d_\theta x_1 - d_\phi y_1 + z_1 + t_z} \\ v_2 &= \lambda\frac{-d_\psi x_1 + y_1 + d_\phi z_1 + t_y}{d_\theta x_1 - d_\phi y_1 + z_1 + t_z} \end{aligned} \tag{4.8}$$

Equations 4.8 are valid for the general case, for any features observed at any distance. However, in this work we are using distant horizons (far off mountains) as image features.

This choice is made because such features almost never change, and since the distances are large, it is an easier problem to solve and therefore more amenable to real-time state estimation. Also, as the contrast is usually very good between the surrounding sky and the horizon, this boundary is relatively easy to detect. To simplify equations 4.8 for the special case we are considering, we divide the numerator and denominator by z_1 and use equation (4.6):

$$\begin{aligned} u_2 &= \lambda \frac{u_1 + d_\psi v_1 - d_\theta \lambda + \lambda \frac{t_x}{z_1}}{d_\theta u_1 - d_\phi v_1 + \lambda + \lambda \frac{t_z}{z_1}} \\ v_2 &= \lambda \frac{-d_\psi u_1 + v_1 + d_\phi \lambda + \lambda \frac{t_y}{z_1}}{d_\theta u_1 - d_\phi v_1 + \lambda + \lambda \frac{t_z}{z_1}} \end{aligned} \tag{4.9}$$

Because horizon features are assumed to be very distant compared to the translational distance of the camera, we can safely assume that $\frac{t_x}{z_1} \approx 0$, $\frac{t_y}{z_1} \approx 0$ and $\frac{t_z}{z_1} \approx 0$. Even if images are acquired and processed at a very low frame rate of 10 frames per second (fps), the distance moved by the vehicle at highway speeds, 29 m/s (60 mph), per frame will be 2.9 meters per sampling interval, which is small compared to a feature a few thousand meters away, for example. With this assumption, the above equation reduces to

$$\begin{aligned} u_2 &= \lambda \frac{u_1 + d_\psi v_1 - d_\theta \lambda}{d_\theta u_1 - d_\phi v_1 + \lambda} \\ v_2 &= \lambda \frac{-d_\psi u_1 + v_1 + d_\phi \lambda}{d_\theta u_1 - d_\phi v_1 + \lambda} \end{aligned} \tag{4.10}$$

A simplified estimation formula can be obtained by dividing the above two equations,

$$\frac{u_2}{v_2} = \frac{u_1 + d_\psi v_1 - d_\theta \lambda}{-d_\psi u_1 + v_1 + d_\phi \lambda} \tag{4.11}$$

Rearranging and grouping terms associated with the camera orientation parameters $(d_\phi, d_\theta, d_\psi)$, we obtain

$$\begin{bmatrix} \lambda u_2 & \lambda v_2 & -u_2 u_1 - v_2 v_1 \end{bmatrix} \begin{bmatrix} d_\phi \\ d_\theta \\ d_\psi \end{bmatrix} = v_2 u_1 - u_2 v_1 \tag{4.12}$$

We can combine N such equations into a matrix equation (4.13).

$$A\theta = B \tag{4.13}$$

where

$$\theta = \begin{bmatrix} d_\phi \\ d_\theta \\ d_\psi \end{bmatrix}$$

$$A = \begin{bmatrix} \lambda u_2 & \lambda v_2 & -u_2 u_1 - v_2 v_1 \\ \lambda u_4 & \lambda v_4 & -u_4 u_3 - v_4 v_3 \\ \vdots & \vdots & \vdots \\ \lambda u_{2N} & \lambda v_{2N} & -u_{2N} u_{2N-1} - v_{2N} v_{2N-1} \end{bmatrix}$$

$$B = \begin{bmatrix} v_2 u_1 - u_2 v_1 \\ v_4 u_3 - u_4 v_3 \\ \\ \vdots \\ v_{2N} u_{2N-1} - u_{2N} v_{2N-1} \\ \end{bmatrix}$$

This matrix equation is a system of linear equation with more equations than unknowns, e.g. an over-determined system which can be solved using the pseudo-inverse method to get a least squares error estimate. The equation (4.13) can be solved using weighted least squares as

$$\theta = (A'WA)^{-1} A'WB \tag{4.14}$$

where

$$W = diag\,(w_1, w_2, \cdots, w_i, \cdots, w_N)$$

W is a diagonal matrix containing weights for each of the individual equations of the form 4.12 and is a measure of accuracy of pixel correspondence.

If we assume perfect knowledge of point correspondences, $(A'WA)$ will lose rank under either of the following conditions. The proof has been given in the appendix.

- $v_{2N} = a \times u_{2N}$ where a is a scalar
- $u_{2N} u_{2N-1} = -v_{2N} v_{2N-1}$

To confirm that the large-distance assumptions central to the above formulation are valid, a numerical test case was conducted. First, a grid of 2500 points forming a square (1000m x 1000m) was created. Images were generated using a perspective projection model by placing a camera at a distance of 2000m from the plane containing all the points. The camera was rotated through several sets of small rotation angles and images were generated. Normally distributed pixel noise with zero mean and variance of 5 pixels was added to the generated images. Equation (4.14) was used to estimate the roll, pitch and yaw angles from image coordinates. As shown in Fig. 4.2, the estimated roll, pitch and yaw angles agree very well with the actual roll, pitch and yaw of the camera for small rotation angles. As the formulation given by equation 4.14 is obtained after linearization based on small angle assumptions, the covariance matrix $(A'A)^{-1}$ gives estimates of error covariance which are much lower than the actual error. Therefore, the covariance matrix $(A'A)^{-1}$ cannot be used to provide bounds on error in angular deviations $(d_\phi, d_\theta, d_\psi)$.

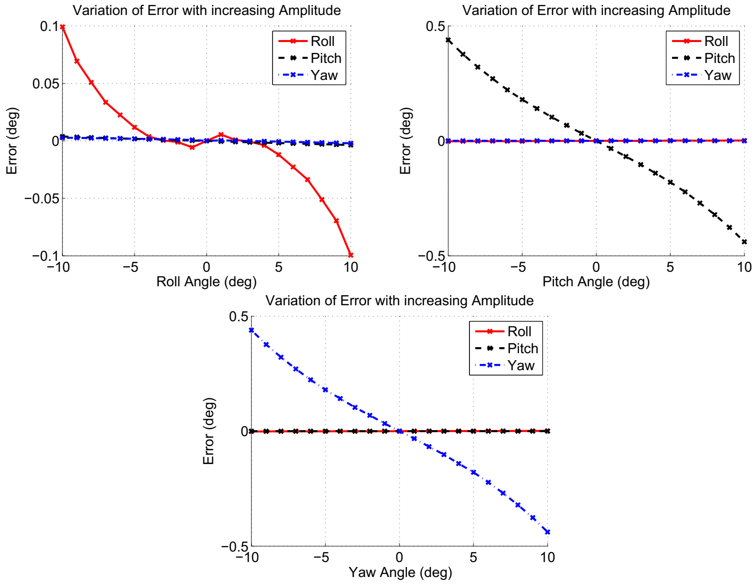


Figure 4.2. Error in roll, pitch and yaw estimation as a function of rotation angle

4.2 Terrain Representation and Rendering

To experimentally validate the vision-inertial Kalman filter, a 3-D representation of the terrain around the test-site is required. All data used in the experiments described shortly were collected at the Pennsylvania Transportation Institute (PTI) Test track. A representation of the terrain around the test-track was generated using the National Elevation Dataset (NED) [20]. The NED provides data in a seamless form, e.g. the gaps in Universal Transverse Mercator (UTM) zones have been filled using interpolation techniques with a consistent datum, elevation unit and projection. The NED has a maximum resolution of one arc-second (approximately 30 meters) for the contiguous United States. The NED is generated from Digital Elevation Models by the USGS and updated regularly to provide seamless elevation data. Digital Elevation Models are generated either by interpolating Digital Line Graphs contours or from electronic imaging sensor systems. NED is based on a bare earth model, i.e the elevation data does not take into account the height of any vegetation on the ground [20]. A more current form of elevation data which takes into account the height of the vegetation is given by Shuttle Radar Topography Mission (SRTM) elevation dataset [98]. Care has been taken to account for the fact that NED dataset uses NAD83 as a datum whereas SRTM dataset uses WGS84 as a datum. Although the difference between the two datums is negligible over North America, WGS84 has been used as a datum to calculate the UTM coordinates in this study.

Image rendering is achieved using a perspective projection model of a camera. The coordinates of all the points of the surface terrain model are transformed into the camera

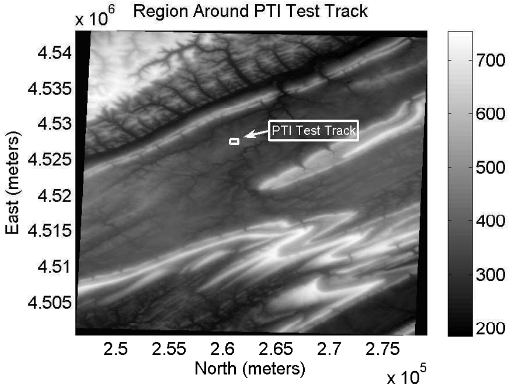


Figure 4.3. 3D Model of Area Around PTI Test Track (Colormap shows Height)

coordinate system using the roll, pitch and yaw estimates. The subset of points which are visible within the rendered image is constrained to be the same as the camera field of view. Internal parameters of the real camera were determined using the four-step camera calibration procedure as given by [43]. These points are then projected onto the image plane using the camera parameters to generate a 720x480 pixel rendered image. As an example, Fig. 4.4 shows a rendered image along with the extracted horizon feature. As of present, the 3-D representation is a surface terrain model with only height information taken from SRTM.

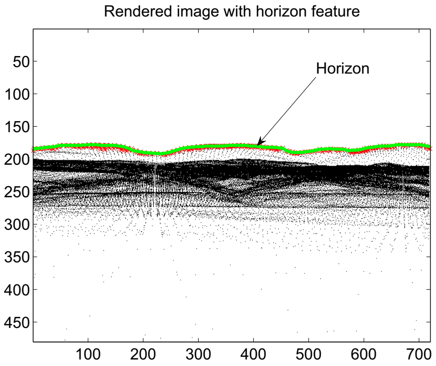


Figure 4.4. Image rendered using perspective projection model

Fig. 4.5 shows the comparison of horizon lines generated using NED and SRTM elevation datasets as viewed from the same location at the PTI test track. As can be seen clearly, the two horizon lines almost overlap each other, mostly because the height of the tree lines (one of the sources of DTM errors) is small relative to the terrain features. Although the experimental results presented in this work are not affected by the choice of elevation dataset, SRTM elevation dataset should be used in case the height of the tree lines is large or an abrupt start or end of the tree lines introduces new features in the horizon profiles e.g. cut lines due to tree harvesting. All results presented in this paper are derived using the SRTM elevation dataset.

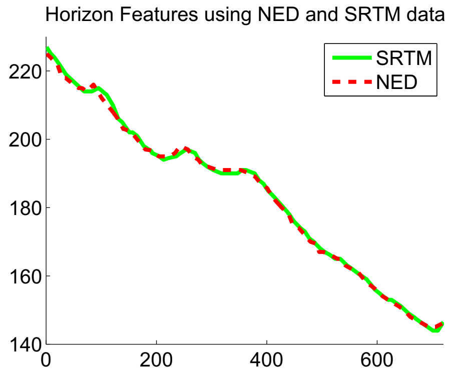


Figure 4.5. Comparison of horizon lines from NED and SRTM datasets

4.2.1 Horizon Line Extraction and Rendering

Horizon line extraction is a well studied problem [114],[5], and common methods use image enhancement, followed by edge detection and determination of uniformity of certain properties like pixel intensity, color or texture in the regions defined by the detected edges. As horizon extraction techniques are not the focus of this work, it has been assumed that a clear view of the sky is available and there is a good contrast between sky and horizon features allowing simple extraction methods. The algorithm used in this study seeks to find the horizon by searching down each pixel column of the image for an intensity change. In this study, a threshold value of 180 gray-scale units was used. The edge so extracted is smoothed using a second-order butterworth low-pass spatial filter to remove pixel noise.

4.2.2 Rendering

Rendering can be achieved using different rendering engines, e.g. Virtools or graphics libraries like OpenGL. High frame rate rendering has been demonstrated on present day gaming engines. A review of real-time terrain rendering techniques can be found in [21]. In the present study, points from the USGS DEM grid are projected using the camera perspective projection model and highest points along each column are used to form the virtual horizon feature as shown in Fig. 4.4.

4.3 The IMU Measurement Model

This work presented in this chapter focuses primarily on estimation of a vehicle's orientation, and the key component in the IMU for estimating angular positions is the gyro. The error sources observed in gyros and other rate sensors have well-known models: gyro noise is frequently modeled as having a wide band component and a bias drift component [6, 32, 11, 90].

Equation 4.15 gives gyro output r_{gyro} in terms of true vehicle rotation rate r, a bias term b_{gyro} and a wide-band noise term (white noise) (w_{gyro}). We are assuming that errors due to cross-coupling, sensor scale factor and sensor misalignment are negligible.

$$r_{gyro} = r + b_{gyro} + w_{gyro} \tag{4.15}$$

$$\dot{b}_\omega = n_{b_\omega}$$

$$E[n_{b_\omega}^2] = \sigma_b^2$$

$$E[w_{gyro}^2] = \sigma_{gyro}^2$$

$$\tag{4.16}$$

The bias drift was modeled as a random walk driven by white noise n_{b_ω}. The wide-band noise w_{gyro} is assumed to be normally distributed with zero mean and variance denoted by σ_{gyro}^2. The noise variances were characterized using Allan variance analysis.

4.4 Orientation Estimates using Vision: Horizon Matching

To complement measurements from inertial sensors, orientation measurements are also obtained by comparing real and rendered images. As shown in section 4.1, we can estimate the deviations in roll, pitch and yaw of the vehicle by comparing real and rendered images if we know the feature correspondences in the two images. To obtain feature correspondences, the horizon curves extracted from the real and rendered images are matched. Curve matching techniques have been used extensively for image/map correspondence [105, 19, 28, 89]. Refinements have also been studied that address map-specific features; for example a curve matching algorithm is described in [113, 94] which is based on matching high curvature points along the curve length. Representing the curves by characteristic strings of high curvature points essentially assigns them shape signatures which are translation and rotation invariant. These shape signatures are then compared to match the curves or to find the longest matching sub-curve between the two curves. A study of horizon matching using curvature points as shape signatures was carried out and it was found that, although the method works well while matching either rendered curves to rendered curves or real curves to real curves for small camera orientation changes, the method lacks robustness when matching real curves to rendered curves. This is due to noisy features like tree lines or mountain irregularities which are visible in real images but not captured accurately in rendered images.

To match rendered horizon curves with real horizon curves, a technique was used that we hereafter call Random SAmple Grid Search (RSAGS). Like RANSAC methods which motivate this approach, this method simplifies the iterative curve-matching procedure by avoiding the use of the entire rendered horizon-line data set. Instead, only a very small random subset of the horizon-line data is used to generate an estimate of camera orientation deviations for curve matching.

For the RSAGS implementation in this study, four points are selected uniformly and randomly from the rendered curve. A transformation given by equation 4.17 is applied to these points for a grid of parameters (t_u, t_v, α) around each curve position. In equation 4.17, t_u denotes translation of in u-direction, t_v denotes translation in v-direction and α denotes rotation of the curve in the plane of the image. W and H denote the width and height of the image in pixels respectively. (u, v) denote the image coordinates of the random points and (u_t, v_t) denote the image coordinates of the random points after the transformation. This transformation signifies a Euclidian transformation, which rotates the image pixels by α about the image center and then translates them by t_u and t_v in image u and v coordinate directions respectively.

$$\begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_u + W/2 \\ 0 & 1 & t_v + H/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -W/2 \\ 0 & 1 & -H/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{4.17}$$

For each point on the parameter grid (t_u, t_v, α), the sum of squared vertical distance is calculated between the transformed points and the points on the real curve having the same u-coordinate. In Fig. 4.6, d_1, d_2, d_3 and d_4 denote these vertical distances. The set of parameters which minimizes the sum of squared vertical distances $(d_1^2 + d_2^2 + d_3^2 + d_4^2)$ is taken as the best-fit parameter set. Using this set of best-fit parameters, the transformation given by equation 4.17 is applied to all the points in the rendered curve, and from all points, a sum of squared vertical distances from the real curve is calculated (denoted as $SSVD_i$). To be robust against the noise in image data, this entire process is repeated on several sets of four random points. The transformation giving the minimum sum of squared vertical distance $SSVD_{min}$ is thereafter assumed to be the best estimate of match parameter set $(t_{u_{best}}, t_{v_{best}}, \alpha_{best})$. This estimate provides the pixel correspondences in the two curves, and from this correspondence, equation 4.14 is used to find the estimates of roll, pitch and yaw. The inverse of the squared vertical distance has been used as the weight entry for each pixel in the weighting matrix W in equation 4.14.

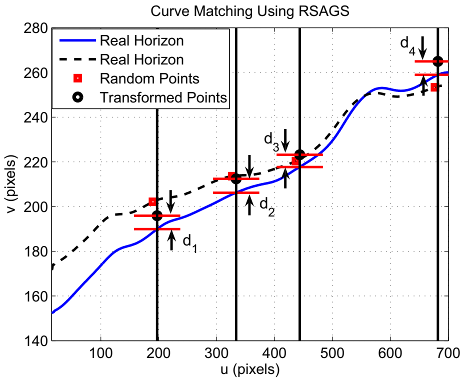


Figure 4.6. Curve Matching Using RSAGS

The size of the parameter grid to be searched for RSAGS is determined by the rate at which images are taken. The higher the frame rate, the smaller total motion of the image from frame to frame, and thus the smaller the grid size that has to be searched. The resolution of the grid determines the accuracy of pixel correspondence. In this study, a frame rate of 30 frames per second and a grid resolution of 1 pixel for t_u and t_v and 0.1 degrees for α was used. A grid size of 10 pixels for t_u and t_v and 2 degrees for α was found to be sufficient for all vehicle tests.

To confirm the validity of the RSAGS curve-fitting method, a numerical study was first conducted by comparing two rendered horizon curves. To create the two curves, we started with one horizon and moved the camera by small angles to generate another horizon by a camera perspective projection model as given by Fig. 4.1. Hence, the true deviation of the camera orientation was known. The RSAGS algorithm was used to find the best-fit translation and rotation that matches these two horizon curves, and Fig. 4.7 shows the matching results. Here $(\phi_t, \theta_t, \psi_t)$ = (1, 0.5, -0.5) denote the true camera motion parameters in roll, pitch and yaw directions, and $(\phi_m, \theta_m, \psi_m)$ = (1.0001, 0.4754, -0.4816) denote the estimated camera motion parameters in degrees. There is a good agreement between true and estimated camera motion parameters. For comparison, the figure also shows the matched horizon curve generated using estimated camera motion parameters.

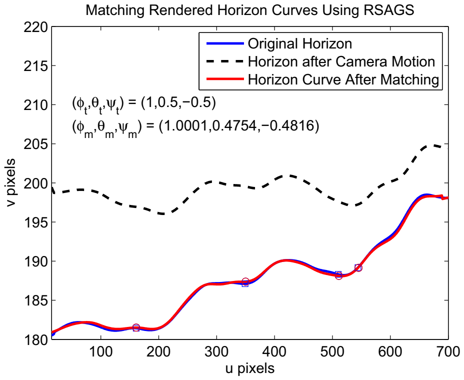


Figure 4.7. Matching Rendered Horizons Using RSAGS

As the next validation step, a real image of a horizon was captured from a stationary vehicle at the PSU Pennsylvania Transportation Institute Test Track using standard contrast threshold techniques to find the horizon boundary between sky/land. This horizon was compared to a rendered horizon curve obtained by using the local DEM and the vehicle's measured DGPS position and IMU orientation. Fig. 4.8 shows the RSAGS results. Because no truth value of camera orientation deviation in real and rendered environment is available, deviations in camera orientation $(\phi_d, \theta_d, \psi_d)$ are estimated by using RSAGS. These deviations give an estimate of error in camera orientation predicted by using IMU alone. To show a comparison of the match between rendered and measured horizon, the matched horizon curve generated by the renderer after moving the camera through the estimated deviations is also shown.

The horizon-only match does little justice to the quality of agreement between the real and rendered scene. To illustrate more fully, Fig. 4.9 shows an overlay of real and

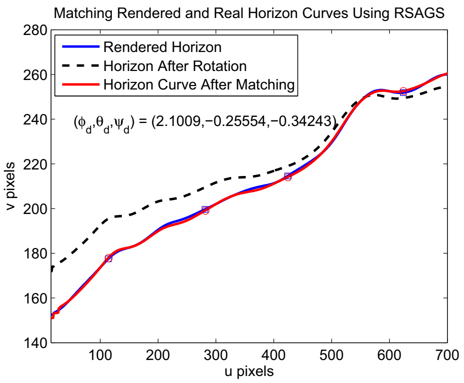


Figure 4.8. Matching Real Horizon to Rendered Horizon Using RSAGS

rendered 3-D images showing matching of the horizon feature from test-track images.

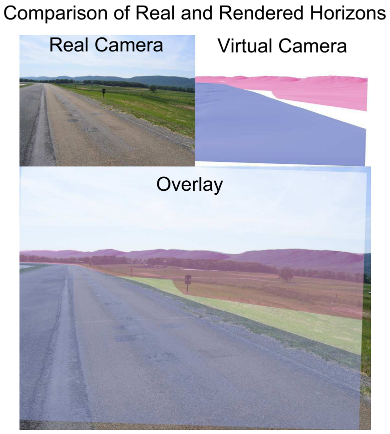


Figure 4.9. An overlay of Rendered and Real Images

4.5 Fusing Vision Measurements with Inertial Data

After obtaining vehicle orientation measurements from inertial and vision sensors, a kinematic extended Kalman filter is used to fuse this data by using vision measurements to correct inertial integration errors as well as to estimate biases in inertial sensors. Use of a kinematic estimator, as opposed to dynamic estimator, eliminates the need for recalibration when using a different vehicle (assuming the same inertial platform), as well as avoids the difficulty of wheel ground interactions.

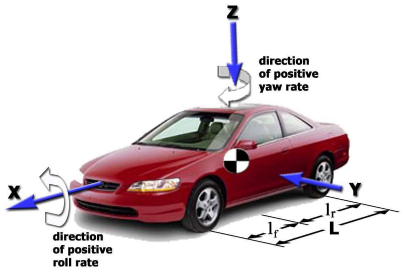


Figure 4.10. SAE Coordinate System for Vehicle

Fig. 4.10 shows the SAE vehicle coordinate system used to derive the estimator model. The notation used hereafter is as follows: The states of the system are vehicle orientation (ϕ, θ, ψ) and gyro biases (b_p, b_q, b_r). Vision measurements are denoted by $(\phi_v, \theta_v, \psi_v)$. The relationship between Euler angles and vehicle body rates $(u_\phi, u_\theta, u_\psi)$ is given by [9, 13]:

$$\begin{aligned}
\dot{\phi} &= (u_\phi - b_p) + \sin\phi \tan\theta (u_\theta - b_q) + \cos\phi \tan\theta (u_\psi - b_r) \\
\dot{\theta} &= \cos\phi (u_\theta - b_q) - \sin\phi (u_\psi - b_r) \\
\dot{\psi} &= \sin\phi \sec\theta (u_\theta - b_q) + \cos\phi \sec\theta (u_\psi - b_r)
\end{aligned} \tag{4.18}$$

Gyro biases are modeled as random walks

$$\dot{\mathbf{b}}_\omega = n_{b_\omega} \tag{4.19}$$

$$E(n_{b_\omega}^2) = \sigma_b^2$$

where $\dot{\mathbf{b}}_\omega = \begin{bmatrix} b_p & b_q & b_r \end{bmatrix}$. Defining $\mathbf{x}$ as the vector of states to be estimated

$$\mathbf{x} = \begin{bmatrix} \phi & \theta & \psi & b_p & b_q & b_r \end{bmatrix}^T \tag{4.20}$$

From equations 4.18 and 4.19, the process model can be written as

$$\dot{x} = f(x, u) + w_c \tag{4.21}$$

Here w_c is the process noise with covariance given by

$$E(w_c^2) = Q_c = \begin{bmatrix} Q_{gyro} & 0 \\ 0 & Q_{bias} \end{bmatrix} \tag{4.22}$$

where Q_{gyro} and Q_{bias} are given by

$$Q_{gyro} = \begin{bmatrix} \sigma_{gyro}^2 & 0 & 0 \\ 0 & \sigma_{gyro}^2 & 0 \\ 0 & 0 & \sigma_{gyro}^2 \end{bmatrix} \tag{4.23}$$

$$Q_{bias} = \begin{bmatrix} \sigma_b^2 & 0 & 0 \\ 0 & \sigma_b^2 & 0 \\ 0 & 0 & \sigma_b^2 \end{bmatrix}$$

Here σ_{gyro} is the gyro wide-band noise and σ_b is the noise due to bias instability as given by equation 4.15.

The measurement model can be written as

$$\begin{bmatrix} \phi_v \\ \theta_v \\ \psi_v \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \\ b_p \\ b_q \\ b_r \end{bmatrix} + v_n \tag{4.24}$$

or more succinctly as

$$z_k = H_k x_k + v_n \tag{4.25}$$

v_n in equation 4.24 is the measurement noise introduced due to the curve matching algorithm, which is limited by the grid resolution.

The measurements ϕ_v, θ_v and ψ_v are derived using equation 4.4 as follows.

$$R(\phi_v, \theta_v, \psi_v) = R(d_\phi, d_\theta, d_\psi) R(\phi_0, \theta_0, \psi_0) = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{4.26}$$

From equations 4.26 and 4.3, we have

$$\phi_v = \arctan \frac{a_{23}}{a_{33}} \tag{4.27}$$

$$\theta_v = -\arcsin a_{13}$$

$$\psi_v = \arctan \frac{a_{12}}{a_{11}}$$

Assuming that the horizon features being matched are much larger than the grid spacing, the quality of match is indicated by the final sum of vertical deviations for the whole horizon curve. Using r_ϕ, r_θ and r_ψ to denote the grid resolutions in roll, pitch and yaw directions respectively and S_v to denote the final sum of squared vertical distances, the measurement covariance can be approximated as

$$E(v_n^2) = \begin{bmatrix} r_\phi^2 + k_\phi S_v & 0 & 0 \\ 0 & r_\theta^2 + k_\theta S_v & 0 \\ 0 & 0 & r_\psi^2 + k_\psi S_v \end{bmatrix} \tag{4.28}$$

Here k_ϕ, k_θ and k_ψ are scaling constants, which are determined through implementation tests. The purpose of including S_v in the variance estimate is to recognize the uncertainty introduced by the image-matching process. Recognizing that this is an ad-hoc solution, formal techniques to bound measurement error as a function of feature error are being sought.

The covariance of the measurement noise can be written as

$$R_k = E(v_n^2) \tag{4.29}$$

A discrete extended Kalman filter is used to fuse vision and IMU measurements. In the prediction step, nonlinear equations of motion 4.18 are used as follows.

$$\hat{x}_k(-) = \hat{x}_{k-1} + f(x_{k-1}, u_k)Ts \tag{4.30}$$

$$P_k(-) = F_k P_{k-1}(+) F_k^T + Q_k$$

Here F_k is the discretized jacobian matrix of $f(x,u)$ with respect to the state vector $\mathbf{x}$, Q_k represents the discrete process noise matrix and Ts denotes the sampling time interval. Q_k is given by

$$Q_k = \frac{W_k Q_c W_k^T}{Ts} \tag{4.31}$$

where W_k is given by

$$W_k = \begin{bmatrix} G_k & 0 \\ 0 & I_{3\times3} \end{bmatrix} \tag{4.32}$$

here G_k is the discretized jacobian matrix of $f(x,u)$ with respect to the input vector $[u_\phi \; u_\theta \; u_\psi]$.

An important requirement for the Kalman filter to work is that the image and inertial data should be aligned in time. Because the inertial data is sampled much more quickly than image data, there are measurement cases where the vision measurement is not available. At these times, equation 4.30 is repeatedly used to predict the state of the system by using $x_k(+) = x_k(-)$ and $P_k(+) = P_k(-)$ for the next iteration. When a vision measurement is available, a measurement update step is applied to the states as given by:

$$
\begin{aligned}
K_k &= P_k(-)H_k^T \left[H_k P_k(-)H_k^T + R_k\right]^{-1} \\
\hat{x}_k(+) &= \hat{x}_k(-) + K_k \left[z_k - H_k \hat{x}_k(-)\right] \\
P_k(+) &= \left[I - K_k H_k\right] P_k(-)
\end{aligned}
\tag{4.33}
$$

Results from numerical simulations are presented in the next section to verify the performance of the Kalman filter.

4.6 Numerical Simulations

Numerical simulations were carried out to evaluate the performance of the Kalman filter developed in the previous section under ideal conditions. Profiles of vehicle roll, pitch and yaw angles similar to those that might be measured on a vehicle were generated and used as truth values. Gyro models given by equation 4.15 were used to simulate angular rates measured by the gyros. Two rendered images were generated, one using the predicted roll, pitch and yaw and the other using the true roll, pitch and yaw angles. Horizon lines were extracted using thresholding and then compared to find angular deviations which were used as measurements. Fig. 4.11 shows the comparison of true and estimated roll, pitch and yaw angles. The estimated and true angles overlap each other with no discernable difference.

Kalman filter estimates of gyro biases are shown in Fig 4.12. This figure shows that the estimated bias values converge to the true bias values used in the gyro models to generate the true angular rates.

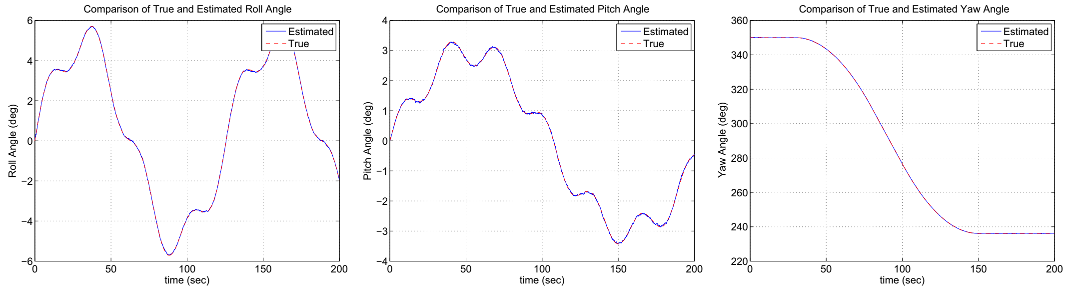


Figure 4.11. Comparison of Estimated and True Orientation Angles

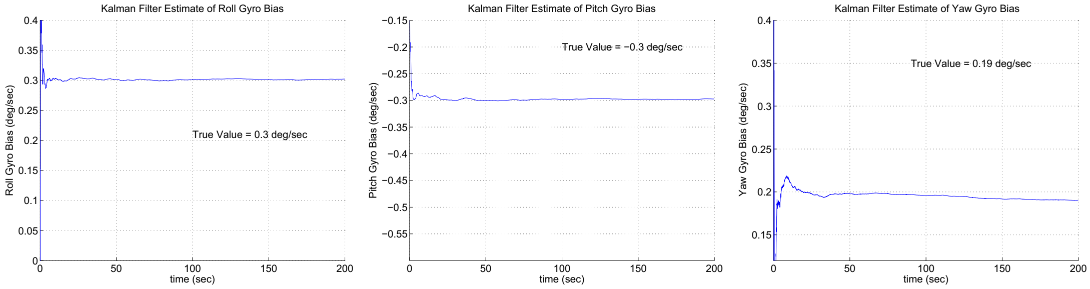


Figure 4.12. Gyro Bias Estimates

Fig. 4.13 shows the comparison of estimated and true error in roll, pitch and yaw angles. The steady state values of estimated errors reach a steady state value of around 0.04 degrees for roll, pitch and yaw. The estimated error curves in these figures shows the 1σ value of error estimated by the Kalman filter. Actual values of errors are also plotted with values of mean and standard deviation of error specified on each plot. As can be seen from the figures, the Kalman filter estimates of 1σ error (=(0.039, 0.041, 0.039) degrees) match very well with the standard deviation of actual error values (=(0.0229,0.0184,0.0272) degrees) for roll, pitch and yaw respectively.

In the next section, this Kalman filter is implemented using experimental data collected at the Pennsylvania Transportation Institute's test track facility on a Mercury Tracer station wagon vehicle instrumented with the IMU described previously and a windshield-mounted camcorder.

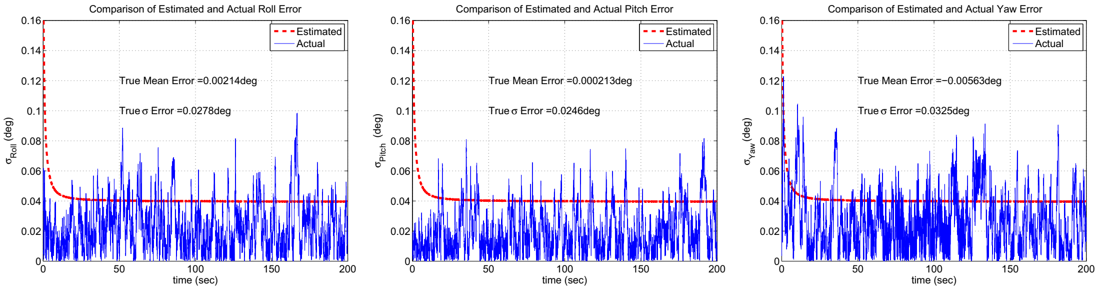


Figure 4.13. Comparison of Estimated and Actual Orientation Errors

4.7 Experimental Results

Experiments were conducted where gyro data from the IMU as well as the video of horizon features recorded from an on-vehicle camera were used off-line in a Kalman filter to predict roll, pitch and yaw angles. Fig. 4.14 shows the trajectory of the vehicle in an overhead picture of the PTI test track. Fig. 4.15 shows the real pictures from the camera at the start and end of the trajectory. The camera has a resolution of 720x480 and a 30 degree field of view.

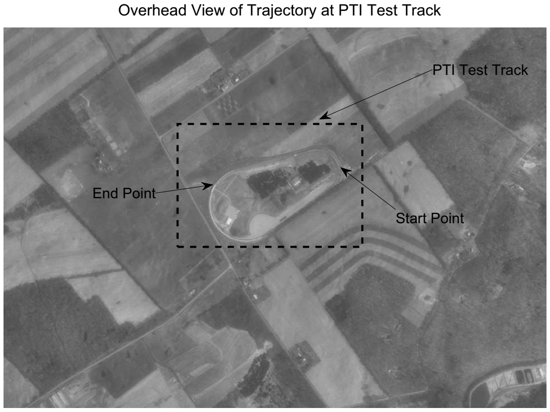


Figure 4.14. Overhead view of the PTI Test Track

Orientation information from the GPS/IMU system was also recorded as a reference. The sample rates for IMU and vision were 100 Hz and 30 Hz respectively, which is much

Figure 4.15. View from the real camera at the start and end of the trajectory

larger than the typical chassis dynamics of a vehicle (2Hz or less). Several test runs were conducted and the estimated orientation angles from the vision/inertial estimator were compared with the estimates from the GPS/IMU system. Figs. 4.16 through 4.18 show the comparison of estimates. These figures also show the deviations between the two estimates and the error estimates from the two estimators. Bias estimates obtained from the vision/inertial Kalman filter are also plotted. It can be generally seen in the test runs that there is a good agreement between the estimates of roll, pitch and yaw angles. All deviations follow a Gaussian distribution. For example, in Fig. 4.16, the roll deviation has a mean of -0.104 and a 2σ value of 0.496. The pitch and yaw deviations have a mean of -0.00766 degrees and -0.0821 degrees and a 2σ value of 0.258 degrees and 0.797 degrees respectively. It can also be seen from the Fig. 4.16 that the standard deviation of estimated error is much lower for GPS/IMU system in roll and pitch directions. We have found that the measurement resolution of RSAGS method, chosen to be 0.1 degrees (grid resolution), is the factor most limiting the error. However the standard deviation of estimated error in the yaw direction is of the same order for the vision/IMU system as in the defense-grade GPS/IMU system.

There is a small but clear discrepancy when comparing the estimates from the vision/inertial and GPS/IMU systems. The estimation error (2σ) is approximately 0.08 degrees for the vision/inertial Kalman filter, which overestimates performance compared to the experimentally measured 2σ comparison bound of approximately 0.5 degrees between vision/inertial and GPS/IMU orientation estimates. One hypothesis to explain this discrepancy is that there is an error in the orientation estimates because of individual movement of the camera and GPS antenna mounts as well as relative motion due to their different locations on the vehicle. Both of these instruments, although rigidly and carefully mounted on the car, will have some movement of their own with respect to the vehicle during vehicle motion. This is due to flexing of the vehicle's frame and the small vibration in the cantilever type mounts for the camera and GPS antenna. The error due to this vibration cannot be assumed to be white noise to add to the measurement noise as it will strongly depend on the road profile and accelerations of the vehicle. We believe that even if we assume that the estimates from the GPS/INS system are perfect, this movement can lead to a discrepancy of 0.5 degrees (2sigma) which is observed in the results. This estimate is based on the discussion of frame motion with chassis testing experts as well as the observed image vibration during the vehicle motion.

Results from three runs of the experiment are shown in Figs. 4.16 through 4.18. These tests were done at different speeds to verify that the algorithm gives estimation behavior seemingly independent of the dynamics of the vehicle.

4.7.1 Effect of Position on Orientation Estimates

Position errors were introduced in the renderer to analyze their effect on orientation estimate. A error model similar to an accelerometer with bias and random noise was

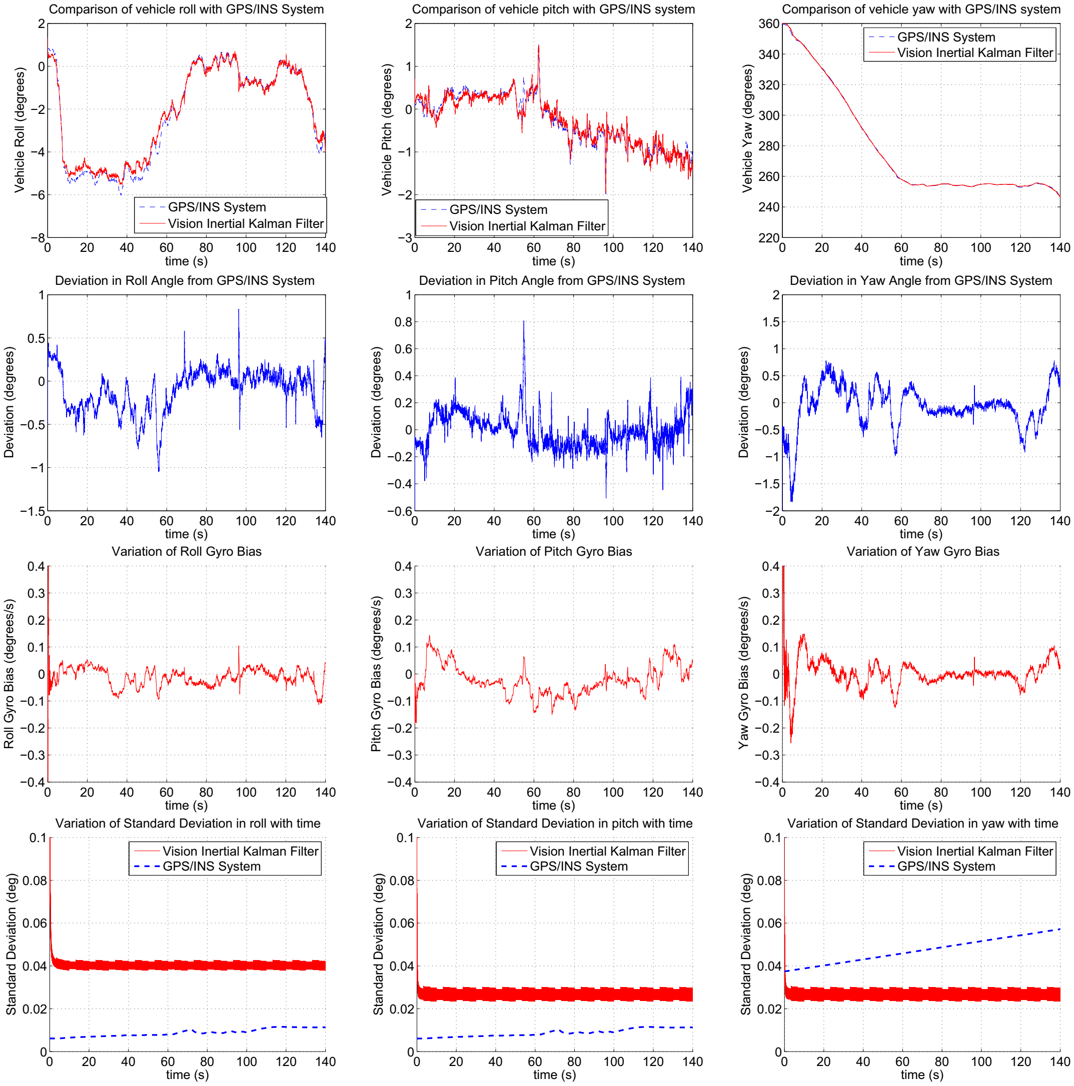


Figure 4.16. Experimental run1: Speed = 10mph

added to the exact position estimate given by the DGPS system. The position error was set to zero every 16 seconds to mimic landmark identification at regular intervals. The interval of 16 seconds was chosen to limit the error growth to within 15m in each direction after assuming typical values for accelerometer bias and noise values. A trajectory of the vehicle position during the experiment is shown in Fig. 4.19. Fig. 4.20 shows the position error in east, north and up directions. Finally, Fig. 4.21 shows the orientation estimation

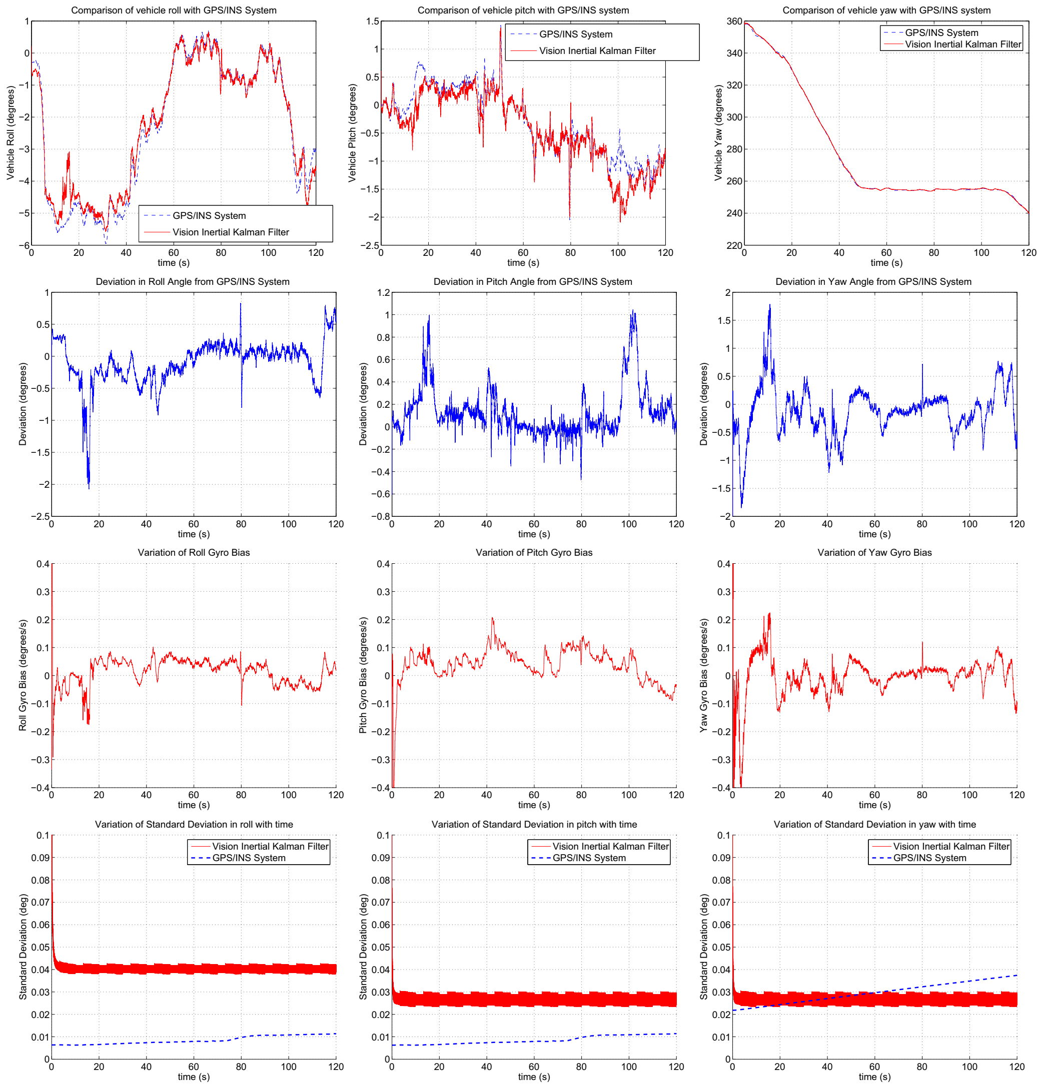


Figure 4.17. Experimental run2: Speed = 10mph

results for the first run after an error having the above mentioned characteristics was added to the position. As can be seen by comparison to Fig. 4.16, an error of the order of 10 to 15 m in each direction has little effect on the orientation estimation results. 2σ values for roll, pitch and yaw in this case are 0.499, 0.257 and 0.796 degrees respectively in this run.

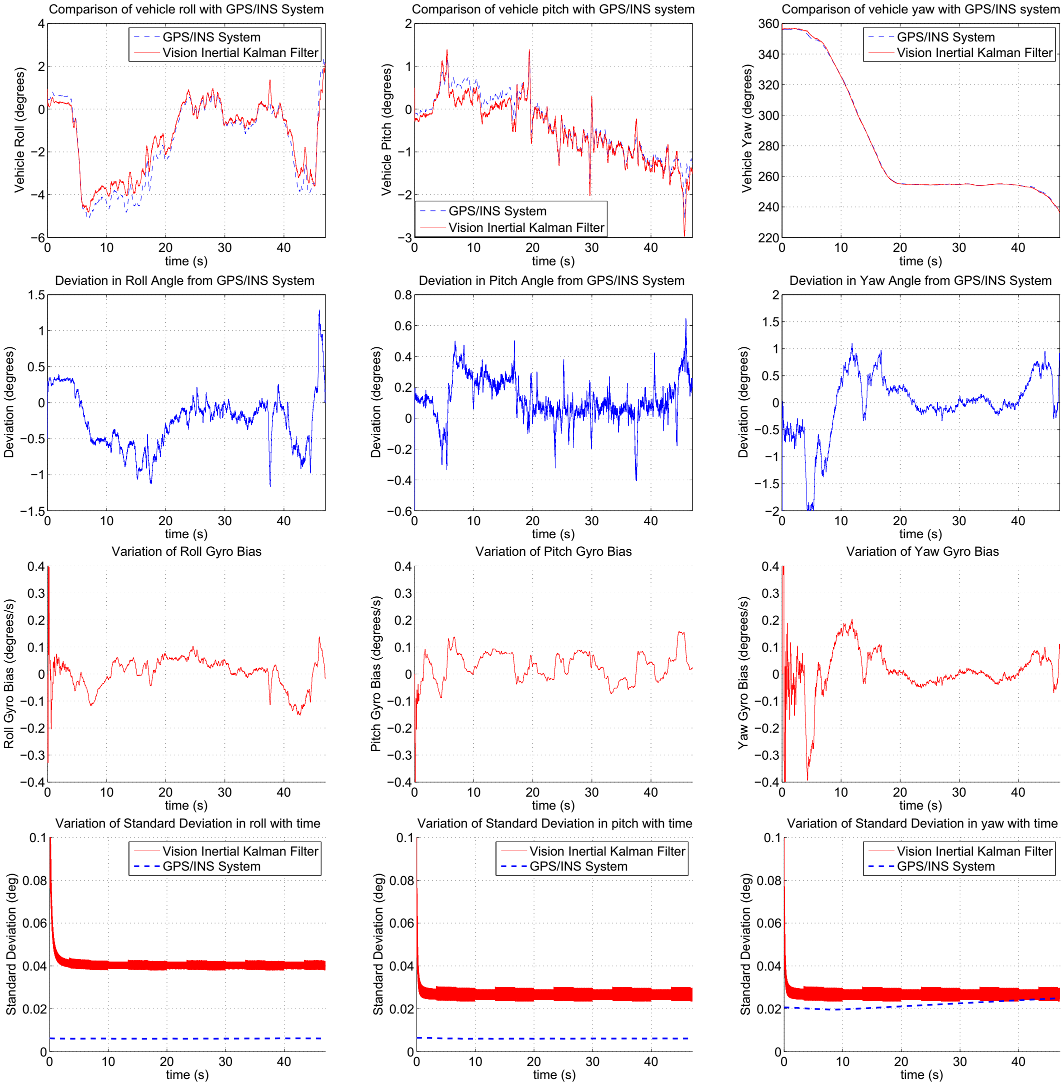


Figure 4.18. Experimental run3: Speed = 30mph

4.8 Conclusions

This chapter has demonstrated the use of horizon lines detected by a camera and DEM data stored a priori to measure vehicle roll, pitch and yaw angles. The horizon lines seen in the captured video are compared to the horizon lines generated from the rendered geography using curve matching techniques to generate the deviations of vehicle roll, pitch and yaw angles in real and rendered environments. A kinematic extended Kalman

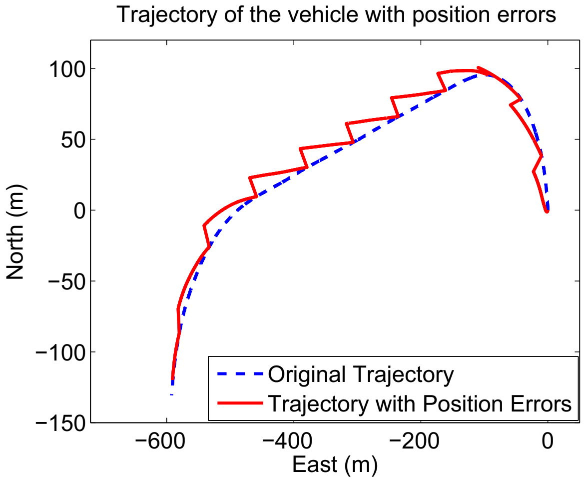


Figure 4.19. Overhead View of the Trajectory and Trajectory with Position Errors

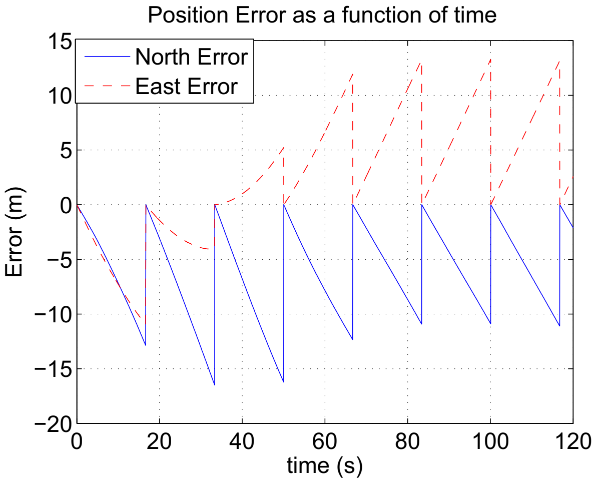


Figure 4.20. Position Error in East and North Directions

filter implemented using inertial and vision data was used to provide estimates of roll, pitch and yaw of the vehicle. Comparison of these estimates with measurements from a high quality GPS/IMU system shows close agreement in roll, pitch and yaw angles. The vision/inertial Kalman filter implemented in this paper can estimate roll, pitch and yaw angle estimates with a deviation of 2σ bound of 0.496 degrees, 0.258 degrees and 0.797 degrees respectively, when compared to the GPS/IMU system.

The investigations of this chapter suggest several future research topics that might

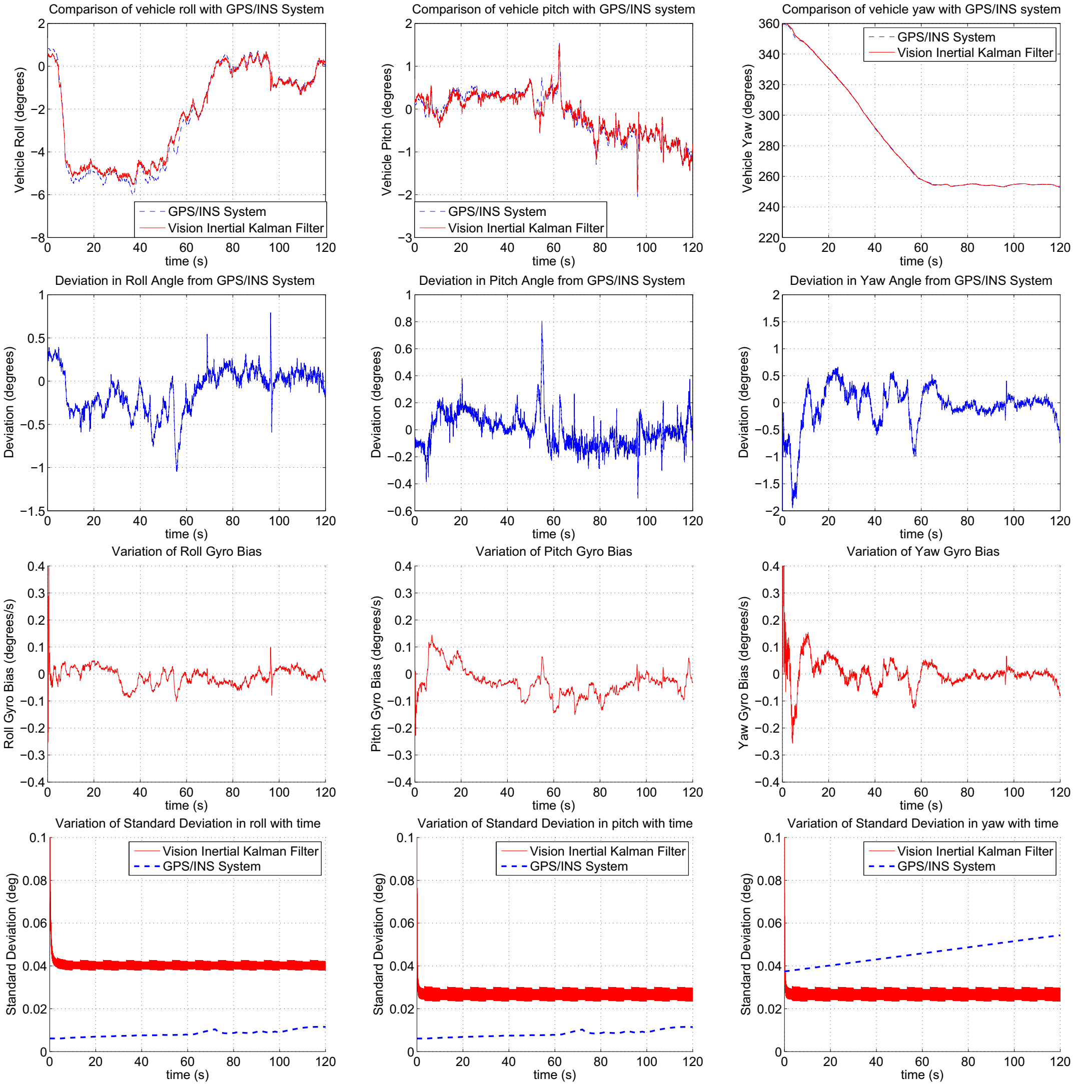


Figure 4.21. Experimental run1 with Position Error: Speed = 10mph

further improve this image-based localization method. Algorithms need to be developed which take into account the observability of roll, pitch and yaw angles from the image features. Image features, e.g. horizon contours, may lack sufficient information to resolve orientation (e.g. a Kansas flat-horizon phenomenon), particularly for yaw angle sensing. Also, the algorithm formulation can be modified such that roll and pitch measurements can be estimated even if yaw cannot be recovered.

The next stage of this work seeks to use road signs and markers for localization. This will allow estimation of local position and velocity states of a vehicle, and may additionally ease the requirement central to this work of visibility and sufficient variability of the horizon for matching.

Appendix

Rewriting equation 4.13 for convenience

$$A\theta = B \tag{4.34}$$

where

$$\theta = \begin{bmatrix} d_\phi \\ d_\theta \\ d_\psi \end{bmatrix}$$

$$A = \begin{bmatrix} \lambda u_2 & \lambda v_2 & -u_2u_1 - v_2v_1 \\ \lambda u_4 & \lambda v_4 & -u_4u_3 - v_4v_3 \\ \vdots & \vdots & \vdots \\ \lambda u_{2N} & \lambda v_{2N} & -u_{2N}u_{2N-1} - v_{2N}v_{2N-1} \end{bmatrix}$$

$$B = \begin{bmatrix} v_2u_1 - u_2v_1 \\ v_4u_3 - u_4v_3 \\ \vdots \\ v_{2N}u_{2N-1} - u_{2N}v_{2N-1} \end{bmatrix}$$

The solution is

$$\theta = (A'WA)^{-1}A'WB \tag{4.35}$$

where

$$W = diag\left(\frac{1}{s_1^2}, \frac{1}{s_2^2}, \cdots, \frac{1}{s_i^2}, \cdots, \frac{1}{s_N^2}\right)$$

$(A'WA)$ will be invertible if it has a full rank. As the rank of product of matrices is less than or equal to the constituent matrices, $(A'WA)$ will have a full rank if A has a full column rank.

The first two columns of A consist of terms λu_{2N} and λv_{2N}. If $v_{2N} = au_{2N}$, the column rank of A is clearly less than 3 and hence the first condition.

If $v_{2N} \neq au_{2N}$, the third column of A, i.e. $-u_{2N}u_{2N-1} - v_{2N}v_{2N-1}$, will be a linear combination of the first two columns if they are multiplied by $\frac{-u_{2N-1}}{\lambda}$ and $\frac{-v_{2N-1}}{\lambda}$ respectively and then added. But these two multipliers are different for each row of the matrix A assuming equation 4.34 uses distinct pixels. Hence the third column cannot be a linear combination of the first two columns except for the case when $u_{2N}u_{2N-1} = -v_{2N}v_{2N-1}$, i.e. the third column of A is identically zero. Hence the second condition.

Chapter 5

Position and Orientation Estimation Using Near Field Features

This chapter extends the work done in chapter 4 to use near field features like road signs and road markers for vehicle localization. Near field features are more visible than far-off features like mountains and can be used to estimate vehicle position in addition to orientation. Near field features also help to improve observability of vehicle orientation which was seen as a limitation of the work done in chapter 4.

The grid based RSAGS algorithm used in chapter 4 for registering real and rendered horizon features was efficient for orientation estimation as the horizon features usually have a simple geometry and are easily extracted given their contrast with the sky. Also, as the orientation angles were the only three variables to be measured, the grid-based RSAGS algorithm was computationally feasible while providing the benefit of finding the global minima. Unfortunately, near field features like road signs, buildings etc., have a more complex geometry which is relatively difficult to extract. Three additional variables of vehicle position are coupled with the variables of vehicle orientation. For example, yaw angle is coupled with lateral position of the vehicle and pitch angle with vertical position. The number of coupled variables to be estimated makes grid based algorithm like RSAGS unsuitable for this problem. Therefore, computationally efficient Newton based techniques have been used in this chapter for registering real and rendered images.

Technology is available today to create photo-realistic 3D models of cities, for example see the works of [30, 40]. Mobile mapping systems have been used to extract road features and create 3D road models [51, 75]. These models can be used to render images of the environment surrounding the vehicle through a virtual camera. A method for vehicle state estimation is proposed in this chapter which is based on registering a real image from a camera on-board the vehicle and a rendered image generated using the virtual camera. It is envisioned that after registering the real and virtual images, the virtual images can be displayed on a screen or part of the wind shield to provide vehicle navigation directions.

A gradient-based technique called Lucas-Kanade [66] and its extensions [4] have been widely used for image alignment. The goal of the Lucas-Kanade algorithm is to align a template image to a given image using a parameterized warp function. The sum of squared differences between the image intensities is used as the error function. Although the warp function can be arbitrarily complex, alignment of real and virtual cameras is neither feasible without a map of the imaged environment nor is usually the goal.

A Region-Of-Interest (ROI) based vision algorithm is proposed in this chapter which registers a rendered representation of the environmental features e.g. road signs and road markers to corresponding features in image frames coming from a real camera to determine the position and orientation of a vehicle. The Lucas-Kanade algorithm formulation [4] is extended to pose vehicle position and orientation parameters as variables using the camera perspective projection model. The algorithm registers features in the rendered images with features in real images using a gradient-based minimization of sum of squared intensities. To improve the convergence properties as well as the convergence time of the vision algorithm, an IMU is used to predict the location and possible variability of features in the rendered representation defining a ROI.

The vision algorithm used in this work is computationally efficient as it is based on fast Newton-based gradient minimization of an error function and works on well defined ROIs around the features in the environment instead of the whole image. The use of the IMU gives a very good initial estimate for the gradient minimization, making image pyramids, which are commonly used to get a good initial estimate [66], unnecessary.

Measurements from the IMU are fused with those from the vision algorithm using a Kalman filter. Features in the rendered representation provide an absolute reference for the vision algorithm, which enables the estimation of IMU errors using the Kalman filter. Numerical simulations show that the Kalman filter fusing the inertial measurements and vision measurements is successfully able to localize the vehicle as well as estimate inertial errors. Finally, results are presented from experimental test runs done at the Pennsylvania Transportation Institute test track, which show that the presented technique can estimate the position of vehicle to an accuracy of approximately 25cm and orientation of the vehicle to a sub-degree accuracy.

This chapter is organized as follows. Section 5.1 details the vision algorithm using pre-mapped features in the environment to estimate the position and orientation of the vehicle. Numerical simulations verifying the correctness of the vision algorithm are presented in section 5.2. Section 5.3 describes the technique used to generate the map used in this work as well as software used to render the map. The IMU measurement model is presented in section 5.4. An error state kinematic Kalman filter is presented in section 5.5 which fuses the measurements from the IMU and the position and orientation measurements from the vision algorithm. An efficient technique to implement the vision algorithm described in section 5.1 is presented in section 5.6 with the use of an IMU. Numerical Simulations are done in section 5.7 to verify the performance of the Kalman filter. The determination of image acquisition time delay via experiments is shown in section 5.8. Results from experiments performed at the Pennsylvania Transportation Institute test track facility are presented in section 5.9 with comparisons with a high-performance GPS/IMU system. Finally, conclusions are drawn in section 5.10. A list of symbols for this chapter can be found in the end.

5.1 Position and Orientation Estimation Using Image Intensities

Given an image generated from an estimated viewpoint of the virtual camera and a real image from the camera on-board vehicle, the goal of this section is to develop an algorithm to register the two images in order to refine the viewpoint estimate.

Assuming that the real and rendered environments are registered initially, consider a motion composed of a small change in orientation as well as position. The rendered image is denoted by I_{ren} and I_{real} denotes the the real video image after motion. This motion causes the 3D point $p_j = (X_j,Y_j,Z_j)$ with image coordinate (u_j, v_j) in the rendered image to move to $p'_j = (X'_j,Y'_j,Z'_j)$ and show up in the real image with image coordinates (u'_j, v'_j). The shift in the u and v directions for this point in the image domain is denoted by $(\delta u_j, \delta v_j)$. So, the error for pixel (u_j, v_j) is given by:

$$e_j = I_{ren}(u_j, v_j) - I_{real}(u_j + \delta u_j, v_j + \delta v_j) \tag{5.1}$$

For small δu_j and δv_j, error e_j can be rewritten as:

$$e_j \approx I_{real}(u_j, v_j) + \delta u_j I_u(u_j, v_j) + \delta v_j I_v(u_j, v_j) - I_{ren}(u_j, v_j) \tag{5.2}$$

where I_u and I_v are the gradients of the real image in u and v directions.

If the orientation change is denoted by Euler angles ϕ, θ, ψ, the rotation matrix representing the coordinate transformation due to rotation is:

$$\boldsymbol{R} = (\mathbf{C}_b^n)^T = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ -\sin(\psi)\cos(\phi) + \sin(\phi)\sin(\theta)\cos(\psi) & \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi) & \sin(\phi)\cos(\theta) \\ \sin(\phi)\sin(\psi) + \sin(\theta)\cos(\phi)\cos(\psi) & -\sin(\phi)\cos(\psi) + \sin(\theta)\sin(\psi)\cos(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{5.3}$$

Here $\mathbf{C}_b^n$ represents a transformation from body to navigation co-ordinate system. After a change in orientation and position, the following relation between p_j and p'_j applies:

$$\begin{bmatrix} X'_j \\ Y'_j \\ Z'_j \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix} + \boldsymbol{t} \tag{5.4}$$

$$= \boldsymbol{R}\begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

where $\boldsymbol{t}$ represents translation matrix consisting of translation in x, y and z directions. Assuming that the rotations in the orientation matrix $\boldsymbol{R}$ are small, a small angle approximation can be made and equation 5.4 reduces to:

$$\begin{bmatrix} X'_j \\ Y'_j \\ Z'_j \end{bmatrix} = \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix}\begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{5.5}$$

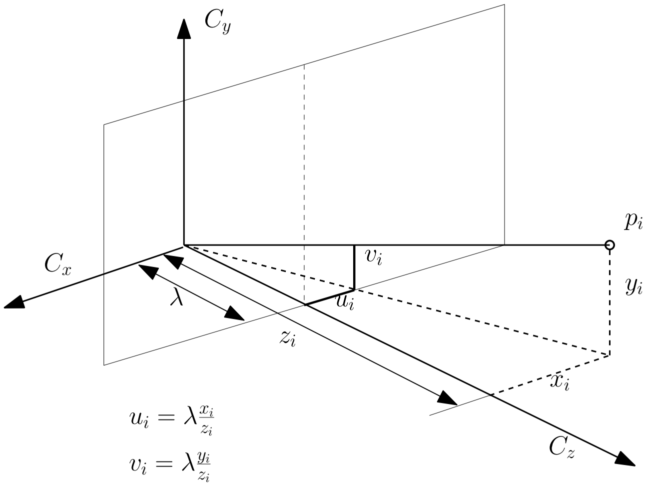


Figure 5.1. Perspective Projection Model

Using the perspective projection model of a camera shown in Fig. 5.1, the image coordinates (u_j, v_j) of the point p_j=(X_j,Y_j,Z_j) are given by:

$$u_j = \lambda\frac{X_j}{Z_j}, \quad v_j = \lambda\frac{Y_j}{Z_j} \tag{5.6}$$

where λ is the focal length. Using the same model, the image coordinates (u'_j, v'_j) of the point $p'_j(X'_j,Y'_j,Z'_j)$ are given by:

$$u'_j = \lambda \frac{X'_j}{Z'_j}, \quad v'_j = \lambda \frac{Y'_j}{Z'_j} \tag{5.7}$$

Using values of X'_j, Y'_j and Z'_j from equation (5.5), u'_j and v'_j can be written as:

$$\begin{aligned} u'_j &= \lambda \frac{X_j + \psi Y_j - \theta Z_j + t_x}{\theta X_j - \phi Y_j + Z_j + t_z} \\ v'_j &= \lambda \frac{-\psi X_j + Y_j + \phi Z_j + t_y}{\theta X_j - \phi Y_j + Z_j + t_z} \end{aligned} \tag{5.8}$$

Using equations 5.8 and 5.6, the shift in the image coordinates $(\delta u_j, \delta v_j)$ can be written as:

$$\begin{aligned} \delta u_j = u'_j - u_j &= \lambda \left[\frac{X_j + \psi Y_j - \theta Z_j + t_x}{Z_j + \theta X_j - \phi Y_j + t_z} - \frac{X_j}{Z_j} \right] \\ &= \lambda \left[\frac{X_j Y_j \phi - (X_j^2 + Z_j^2)\theta + Y_j Z_j \psi + Z_j t_x - X_j t_z}{Z_j(Z_j + \theta X_j - \phi Y_j + t_z)} \right] \end{aligned} \tag{5.9}$$

$$\begin{aligned} \delta v_j = v'_j - v_j &= \lambda \left[\frac{Y_j - \psi X_j + \phi Z_j + t_y}{Z_j + \theta X_j - \phi Y_j + t_z} - \frac{Y_j}{Z_j} \right] \\ &= \lambda \left[\frac{(Y_j^2 + Z_j^2)\phi - X_j Y_j \theta - X_j Z_j \psi + Z_j t_y - Y_j t_z}{Z_j(Z_j + \theta X_j - \phi Y_j + t_z)} \right] \end{aligned} \tag{5.10}$$

The denominator of the above equation can be simplified by observing that the z-coordinate of the point p_j is much greater than any of the motion vector components ϕ,

θ, ψ, t_x, t_y, and t_z. Simplifying the above equations:

$$\delta u_j = \lambda \left[\frac{X_j Y_j \phi - (X_j^2 + Z_j^2)\theta + Y_j Z_j \psi + Z_j t_x - X_j t_z}{Z_j^2} \right] \tag{5.11}$$

$$\delta v_j = \lambda \left[\frac{(Y_j^2 + Z_j^2)\phi - X_j Y_j \theta - X_j Z_j \psi + Z_j t_y - Y_j t_z}{Z_j^2} \right] \tag{5.12}$$

Substituting 5.11 into 5.2:

$$\begin{aligned} e_j = (I_{real}(u_j, v_j) - I_{ren}(u_j, v_j)) &+ \lambda \frac{X_j Y_j I_u(u_j, v_j) + (Y_j^2 + Z_j^2) I_v(u_j, v_j)}{Z_j^2} \phi \\ &- \lambda \frac{(X_j^2 + Z_j^2) I_u(u_j, v_j) + X_j Y_j I_v(u_j, v_j)}{Z_j^2} \theta \qquad (5.13) \\ &- \lambda \frac{X_j Z_j I_v(u_j, v_j) - Y_j Z_j I_u(u_j, v_j)}{Z_j^2} \psi \\ &+ \lambda \frac{I_u(u_j, v_j)}{Z_j} t_x + \lambda \frac{I_v(u_j, v_j)}{Z_j} t_y - \lambda \frac{X_j I_u(u_j, v_j) + Y_j I_v(u_j, v_j)}{Z_j^2} t_z \qquad (5.14) \end{aligned}$$

The above equation can be converted into matrix form and written into the following compact form:

$$e_j = (I_{real}(u_j, v_j) - I_{ren}(u_j, v_j)) + Wz \tag{5.15}$$

where z is the vector of unknowns given by:

$$z = [\phi \quad \theta \quad \psi \quad t_x \quad t_y \quad t_z]^T \tag{5.16}$$

and W is given by:

$$W^T = \begin{bmatrix} \lambda\frac{X_jY_jI_u(u_j,v_j)+(Y_j^2+Z_j^2)I_v(u_j,v_j)}{Z_j^2} \\ -\lambda\frac{(X_j^2+Z_j^2)I_u(u_j,v_j)+X_jY_jI_v(u_j,v_j)}{Z_j^2} \\ -\lambda\frac{X_jZ_jI_v(u_j,v_j)-Y_jZ_jI_u(u_j,v_j)}{Z_j^2} \\ \lambda\frac{I_u(u_j,v_j)}{Z_j} \\ \lambda\frac{I_v(u_j,v_j)}{Z_j} \\ -\lambda\frac{X_jI_u(u_j,v_j)+Y_jI_v(u_j,v_j)}{Z_j^2} \end{bmatrix}$$

The above equation can also be written in terms of image coordinates, (u_j, v_j), as:

$$W^T = \begin{bmatrix} \frac{u_jv_jI_x(x_j,y_j)+(v_j^2+\lambda^2)I_y(x_j,y_j)}{\lambda} \\ -\frac{(u_j^2+\lambda^2)I_x(x_j,y_j)+u_jv_jI_y(x_j,y_j)}{\lambda} \\ -u_jI_y(x_j,y_j)+v_jI_x(x_j,y_j) \\ \lambda\frac{I_x(x_j,y_j)}{Z_j} \\ \lambda\frac{I_y(x_j,y_j)}{Z_j} \\ -\frac{u_jI_x(x_j,y_j)+v_jI_y(x_j,y_j)}{Z_j} \end{bmatrix}$$

5.1.1 Objective Function

The goal of the objective function is to minimize the sum of squared difference between the rendered and real image in a predefined window w.

$$\begin{aligned} E &= \sum_w [I_{real} - I_{ren}]^2 \qquad (5.17) \\ &= \sum_w e_j^2 \end{aligned}$$

$$= \sum_{w} [(I_{real}(u_j, v_j) - I_{ren}(u_j, v_j)) + Wz]^2 \tag{5.18}$$

To find the minimum of the objective function, we take its derivative and set that equal to zero.

$$\frac{dE}{dz} = 0 = \sum_{w} W^T (I_{real} - I_{ren}) + \sum_{w} W^T W z \tag{5.19}$$

Rearranging the terms in the above equations gives a system of linear equations given by:

$$\sum_{w} (W^T W) z = \sum_{w} W^T (I_{ren} - I_{real}) \tag{5.20}$$

The above equation can be solved iteratively using Gauss-Newton method [85] to calculate the current absolute position and orientation of the vehicle.

5.1.2 Measurement Covariance of the Vision algorithm

The covariance of the motion estimates from equation 5.20 is given by:

$$R_{cam} = (\sum_{w} (W^T W))^{-1} \tag{5.21}$$

As this covariance estimate is based on linearization using small angle approximations, it gives conservative estimates of the measurement covariances. Other sources of error in measurement is the map error from which the rendered image is generated and error in camera parameters. For this reason, the covariance estimates given by equation 5.21 are scaled by a factor K to get an estimate of the true covariance. The modified covariance equation is given by:

$$R_{cam} = K(\sum_{w}(W^{T}W))^{-1} \tag{5.22}$$

While doing numerical simulations, map as well as camera parameters are known exactly. The value of K for numerical simulation was chosen to be 10 by comparing the actual errors to the estimated errors. Errors for the real experiments were generated by comparing the vision algorithm measurements with a high-accuracy GPS/IMU system. By comparing these errors with the estimated errors, the value of K was chosen to be 5E5. This high value of K can be explained on account of map errors, camera calibration error sources, and a really sparse feature set used for the experiments.

Equation 5.22 represents the measurement covariance R_{cam} in the camera coordinate frame. A vector X_{cam} in the camera coordinate system can be transformed to vehicle body coordinate system using the following transformation.

$$X_B = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} X_{cam} = \mathbf{C}^{b}_{cam} X_{cam} \tag{5.23}$$

Where $\mathbf{C}^{b}_{cam}$ denotes the transformation from camera to body coordinate system and X_B represents the vector in vehicle body coordinate frame.

The following equation transforms the the covariance R_{cam} from the camera coordinate system to body coordinate system.

$$R_B = \mathbf{C}^b_{cam} R_{cam} (\mathbf{C}^b_{cam})^T \tag{5.24}$$

where R_B denotes the covariance in vehicle body coordinate system.

To express the measurement covariance in navigation coordinates, position covariance and orientation-position cross-covariance needs to be transformed into the navigation coordinates. The Euler angle representation of orientation is the same in navigation and vehicle body coordinate frames and hence the orientation covariance is invariant under this transformation. To this effect, the measurement covariance matrix R_{cam} can be split into orientation and position components as shown in the following equation.

$$R_{cam} = \begin{bmatrix} R_{\Theta\Theta} & R_{\Theta X_{cam}} \\ R_{X_{cam}\Theta} & R_{X_{cam}X_{cam}} \end{bmatrix} \tag{5.25}$$

$$= \begin{bmatrix} R_{\Theta\Theta} & R_{\Theta X_B} (\mathbf{C}^b_{cam})^T \\ (\mathbf{C}^b_{cam}) R_{X_B\Theta} & (\mathbf{C}^b_{cam}) R_{X_B X_B} (\mathbf{C}^b_{cam})^T \end{bmatrix} \tag{5.26}$$

where $R_{\Theta\Theta}$ represents the orientation covariance, $R_{\Theta X_B}$ represents the orientation-position cross-covariance, $R_{X_B\Theta} = (R_{\Theta X_B})^T$ and $R_{X_B X_B}$ represents the position covariance. Observing the relation $X_N = \mathbf{C}^n_b X_B$ between position vector X_N in navigation frame and position vector X_B in the body frame, the measurement covariance in the navigation frame can be written as:

$$R_{nav} = \begin{bmatrix} R_{\Theta\Theta} & R_{\Theta X_B}(\mathbf{C}_b^n)^T \\ (\mathbf{C}_b^n)R_{X_B\Theta} & (\mathbf{C}_b^n)R_{X_BX_B}(\mathbf{C}_b^n)^T \end{bmatrix} \tag{5.27}$$

Thus, the measurement covariance R_{nav} in the navigation frame can be written in terms of the measurement covariance in camera frame R_{cam} using equations 5.27 and 5.24. Numerical simulations are done in the next section to test the performance of the vision algorithm.

5.2 Vision Algorithm Simulations

To validate the vision algorithm described in section 5.1, numerical simulations are done where a video of a hypothetical vehicle trajectory through the map as described in section 5.3 is generated using a virtual camera onboard the vehicle. This video is assumed to have been generated by the real camera, and the algorithm described in section 5.1 is used to estimate the hypothetical vehicle trajectory starting from the known initial location. Figures 5.2 and 5.3 show comparison of true and estimated orientation and position, the corresponding true and estimated errors as well as distribution of true orientation and position estimation errors.

The estimated variance in estimates is calculated using equation 5.22, where the value of K was chosen to be 10 although it is a conservative estimate for this simulated environment and over-predicts the error bounds for the orientation angles and position estimates. A careful look at the figures 5.2 and 5.3 reveals that the maximum orientation angle estimation error is 0.05 degrees approximately and maximum position estimation error is 5cm approximately The figures also show the distribution of true error for ori-

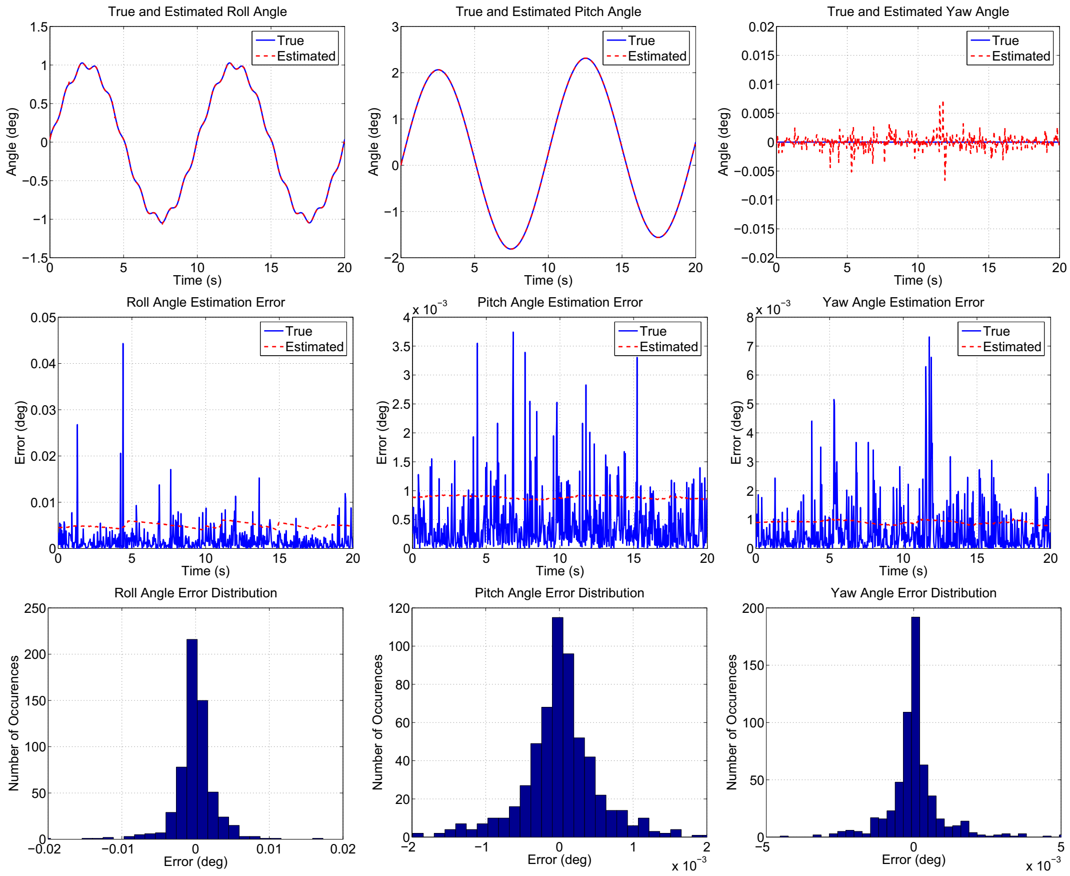


Figure 5.2. Orientation Estimates Using Vision Algorithm

entation angles and position variables and it can be clearly seen from the figures that in each case this distribution is approximately Gaussian. This shows that the vision algorithm does not inherently bias the estimates one way or the other.

5.3 Map Generation and Rendering

The map for simulations was created using 3D Studio Max software from Autodesk. A roadway was modeled along with light poles and trees on the roadside. The 3D model was exported into .3DS format and rendered using the Open Graphics Library (OpenGL). An image of the map created using 3DS Max software is shown in figure 5.4.

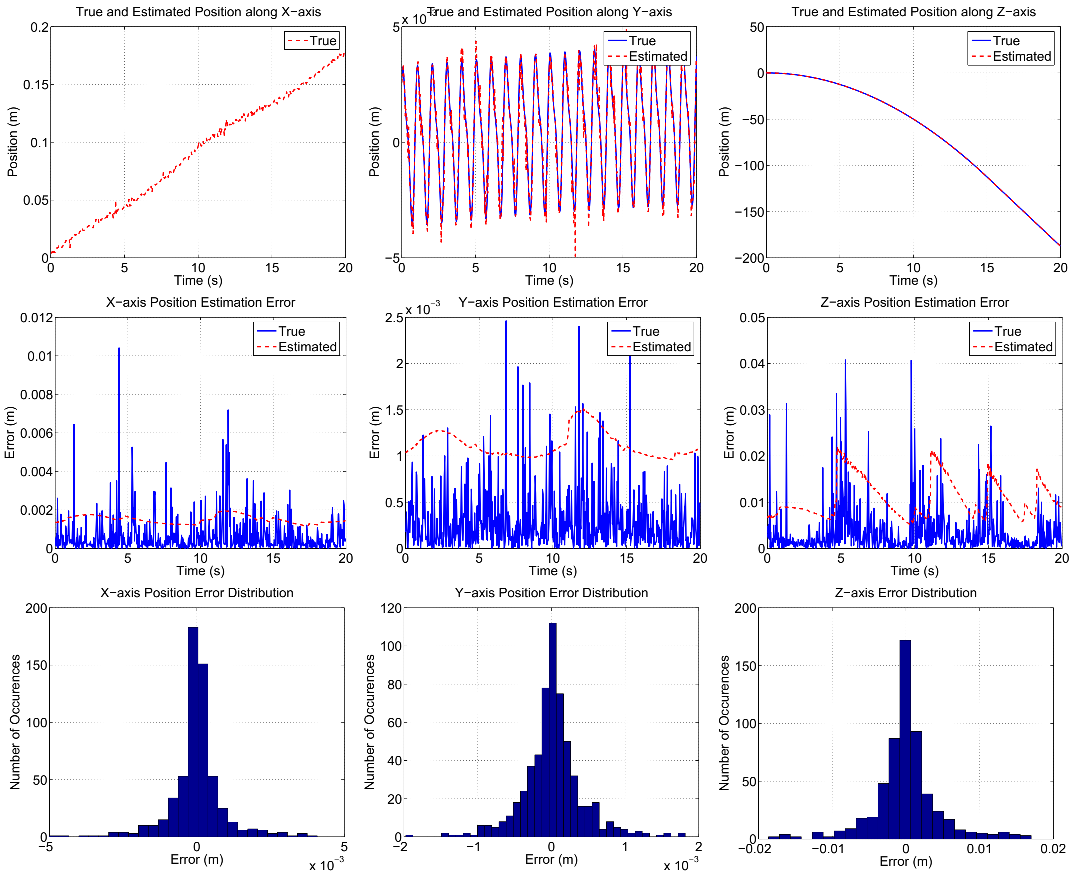


Figure 5.3. Position Estimates Using Vision Algorithm

For real experiments on the PTI test track, road markers and road signs were chosen as features. The road markers were created by gluing a reflective tape to the ground. The road markers were made 1m long with 3m distance between them to simulate the standard highway lane marker stripe, which has a size to distance ratio of 1:3.

Road signs were created using styrofoam in the shape of a stop sign. The maximum horizontal and vertical length of the road sign was chosen as 30 inches and is comparable to the size of an actual stop sign found on city roads. The coordinates of the lane markers and road signs were measured using a GPS/IMU system operating in RTK mode (position accuracy $\approx$ 2cm) and input into OpenGL as quadrilaterals to be rendered.

Map generated using 3D−Studio Max

Figure 5.4. A snapshot of the image taken using 3D Studio Max

Figure 5.5 shows an image of the map created for real experiments at the PTI test track. For image registration, the internal parameters of the real and virtual camera need to be the same. Internal parameters of the real camera were determined using the four-step camera calibration procedure as given by [43]. The camera has a resolution of 720×480 and 42° field of view.

5.4 Incorporation of an IMU and IMU Modeling

To improve the convergence properties as well as convergence time of the vision algorithm, an IMU is used to predict the location and possible variability of features in the rendered representation. Under fast dynamic conditions, features in two adjacent camera frames can have large displacements leading to little or no overlap of features in real and rendered images. This can in turn lead to poor convergence properties. Additionally,

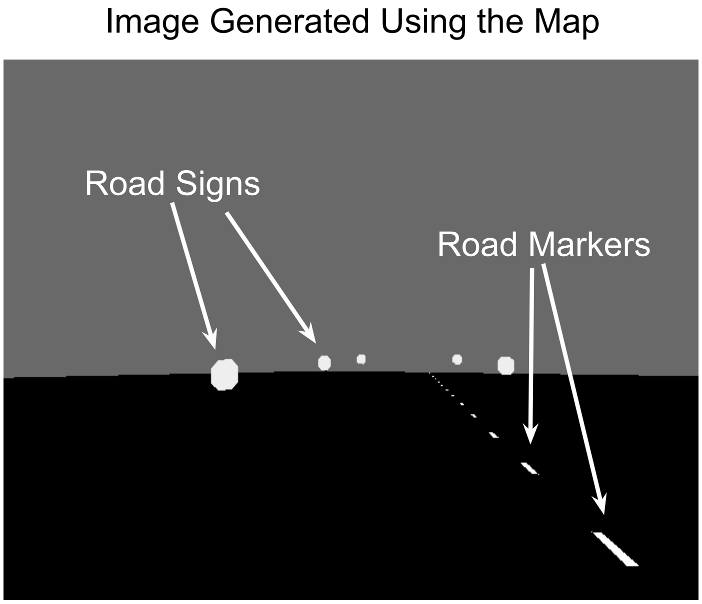


Figure 5.5. Image Generated Using the Map at PTI Test Track

the closer the features are in the real and rendered images, the less the number of iterations required for the vision algorithm to converge. As gyroscopes and accelerometers are excellent sensors to capture fast dynamic conditions, incorporation of an IMU with a high sampling rate can be used to predict the location of features in the rendered image; thereby greatly improving the convergence properties and convergence time of the vision algorithm.

To fuse the IMU measurements with those of the vision algorithm, a kinematic Kalman filter has been used. Raw IMU measurements contain errors due to sensor misalignment, scale factor, biases and noise [6, 32, 11, 90]. The presence of these errors degrades the performance of the Kalman filter. But, it is possible to model the IMU errors and estimate these in addition to the vehicle position and orientation states [6, 32]. Gyro errors are frequently modeled as having a wide band component and a bias drift component [6, 32, 11, 90].

Equation 5.28 gives gyro output $\tilde{\omega}$ in terms of true vehicle rotation rate ω, a bias term b_ω, and a wide-band noise term (white noise) (η_ω). We are assuming that errors due to cross-coupling, sensor scale factor, and sensor misalignment are negligible.

$$\begin{aligned} \tilde{\omega} &= \omega + b_\omega + \eta_\omega && (5.28) \\ \dot{b}_\omega &= \eta_{b_\omega} && (5.29) \\ E[\eta_{b_\omega}^2] &= \sigma_{b_\omega}^2 \\ E[\eta_\omega^2] &= \sigma_{gyro}^2 \end{aligned}$$

The bias drift was modeled as a random walk driven by white noise η_{b_ω}. The wide-band noise η_{gyro} is assumed to be normally distributed with zero mean and variance denoted by σ_{gyro}^2. The noise variances were characterized using Allan variance analysis [1].

Equation 5.30 gives the accelerometer output f_{acc} in terms of true vehicle acceleration f, accelerometer bias b_{acc} and a Gaussian white noise term η_f. Again, the errors due to cross-coupling, sensor scale factor and misalignment are assumed to be negligible.

$$\begin{aligned} f_{acc} &= f + b_{acc} + \eta_f && (5.30) \\ \dot{b}_f &= \eta_{b_f} && (5.31) \\ E[\eta_{b_f}^2] &= \sigma_{b_f}^2 \\ E[\eta_f^2] &= \sigma_{acc}^2 \end{aligned}$$

A white noise term $\eta_{b_{acc}}$ drives the bias drift b_f modeled as a random walk process. The wide-band noise η_f is assumed to be normally distributed with zero mean and variance denoted by σ_{acc}^2. The accelerometer error sources were also characterized using Allan variance analysis.

The IMU model developed in this section will be used in the kinematic Kalman filter developed in the next section to fuse the IMU and vision algorithm measurements.

5.5 Kalman Filter

A Kalman filter framework [33] has been used to fuse the measurements from the IMU and the vision algorithm. Vehicle orientation, velocity, position, gyroscope and accelerometer biases are the 15 chosen states to be estimated. At each time step, mechanization equations presented below are used to advance the estimates of the states using the current best estimate of these states. The Kalman filter algorithm only works on the errors in the filter states. In the prediction step, the errors in the states are predicted using the governing differential equations modeling the states until the next vision algorithm measurement update is available. The difference between the vision algorithm measurements and the predicted value of those measurements is used as a measurement for the correction step of Kalman filter. The initial location of the vehicle is assumed to be known for this work.

5.5.1 State Update

At each time step, the state of the system is updated using the equations given below in the component form for all of the states.

The orientation Euler angles are updated as [9, 107]:

$$\begin{aligned}\dot{\phi} &= (\omega_y \sin(\phi) + \omega_z \cos(\phi)) \tan(\theta) + \omega_x \\ \dot{\theta} &= \omega_y \cos(\phi) - \omega_z \sin(\phi) \\ \dot{\psi} &= (\omega_y \sin(\phi) + \omega_z \cos(\phi)) \sec(\theta) \qquad (5.32)\end{aligned}$$

Using the gyroscope model given by equation 5.28, the angular velocities $\omega_x, \omega_y, \omega_z$ are calculated using the following equation:

$$\omega = (\tilde{\omega} - b_\omega) \qquad (5.33)$$

Where $\tilde{\omega}$ denotes the raw measurements obtained from x, y and z axis gyroscopes and b_ω denotes the corresponding estimated biases.

The velocity and position are updated as:

$$\dot{v}_N = f_N \qquad (5.34)$$

$$\dot{v}_E = f_E \qquad (5.35)$$

$$\dot{v}_D = f_D + g \qquad (5.36)$$

$$\dot{x}_N = v_N \qquad (5.37)$$

$$\dot{x}_E = v_E \qquad (5.38)$$

$$\dot{x}_D = v_D \qquad (5.39)$$

Here the specific forces are given by:

$$\begin{bmatrix} f_n & f_e & f_D \end{bmatrix}^T = \mathbf{C}_b^n \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T \tag{5.40}$$

Using the accelerometer model given by equation 5.30, f_x, f_y, f_z are calculated using the following equation:

$$f^b = (\tilde{f}^b - b_{f^b}) \tag{5.41}$$

Where $\tilde{f}^b$ denotes the raw measurements obtained from x, y and z axis accelerometers and b_{f^b} denotes the corresponding estimated biases.

Gyroscope and accelerometer biases are updated as:

$$\dot{b}_\omega = 0 \tag{5.42}$$

$$\dot{b}_f = 0 \tag{5.43}$$

where b_ω and b_f are the biases in gyroscopes and accelerometers respectively.

5.5.2 Navigation Error Equations

If $\mathbf{\Psi} = [\delta\alpha \quad \delta\beta \quad \delta\gamma]^T$ be the vector of misalignment angles between true orientation and estimated orientation of the vehicle, then the differential equation governing $\mathbf{\Psi}$ is given by [9, 107]:

$$\dot{\mathbf{\Psi}} \approx -\mathbf{C}_b^n \delta\omega_{ib}^b \tag{5.44}$$

where the rotation rates given by the gyroscopes are represented by the vector ω_{ib}^b and $\delta\omega_{ib}^b$ is the error in the rotation rates provided by the IMU. $\mathbf{C}_b^n$ represents the transformation from navigation frame to body reference frame. Using equation 5.33 to substitute for the $\delta\omega_{ib}^{b}$ term, the differential equation describing the rate of change of the misalignment angles becomes:

$$\dot{\mathbf{\Psi}} \approx -\mathbf{C}_b^n(\delta\mathbf{b}_\omega + \eta_\omega) \tag{5.45}$$

The error equation for velocity can be expressed as [9, 107]:

$$\delta\dot{\mathbf{v}} = -\mathbf{\Upsilon}\mathbf{C}_b^n\mathbf{f}^b + \mathbf{C}_b^n\delta\mathbf{f}^b - \delta\mathbf{g} \tag{5.46}$$

$$\delta\dot{\mathbf{p}} = \delta\mathbf{v} \tag{5.47}$$

where $\delta\mathbf{v} = [\delta v_N \quad v_E \ \delta v_D]$ is the vector of velocity estimation errors. $\delta\mathbf{f}^b$ represents the error in the specific forces measurements provided by the IMU. $\mathbf{\Upsilon}$ is a matrix of orientation angle errors given by the following equation.

$$\mathbf{\Upsilon} = \begin{bmatrix} 0 & -\delta\gamma & \delta\beta \\ \delta\gamma & 0 & -\delta\alpha \\ -\delta\beta & \delta\alpha & 0 \end{bmatrix} \tag{5.48}$$

On similar lines, equation 5.41 is used to substitute term $\delta\mathbf{f}^b$ in equation 5.46 to give:

$$\delta\dot{\mathbf{v}} = -\mathbf{\Upsilon}C_b^n\mathbf{f}^b + \mathbf{C}_b^n(\delta\mathbf{b}_f + \eta_f) \tag{5.49}$$

Errors in biases are derived by using the δ operator on both sides of equations 5.29

and 5.31 and can be written as:

$$\delta \dot{b_\omega} = \eta_{b_\omega} \tag{5.50}$$

$$\delta \dot{b_f} = \eta_{b_f} \tag{5.51}$$

Using the equations given above, the state space model of the error states can be written as:

$$\underbrace{\begin{bmatrix} \dot{\boldsymbol{\Psi}} \\ \delta\dot{\mathbf{v}} \\ \delta\dot{\mathbf{p}} \\ \delta\dot{\mathbf{b}}_\omega \\ \delta\dot{\mathbf{b}}_\mathbf{f} \end{bmatrix}}_{\delta\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & -C_b^n & \mathbf{0} \\ \boldsymbol{\Xi} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} \boldsymbol{\Psi} \\ \delta\mathbf{v} \\ \delta\mathbf{p} \\ \delta\mathbf{b}_\omega \\ \delta\mathbf{b}_\mathbf{f} \end{bmatrix}}_{\delta\mathbf{x}} \tag{5.52}$$

$$+ \underbrace{\begin{bmatrix} -C_b^n & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & C_b^n & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} \eta_\omega \\ \eta_\mathbf{f} \\ \eta_{\mathbf{b}_\omega} \\ \eta_{\mathbf{b}_\mathbf{f}} \end{bmatrix}}_{w} \tag{5.53}$$

$\mathbf{\Xi}$ is given by the following equation:

$$\mathbf{\Xi} = \begin{bmatrix} 0 & -f_D & f_E \\ f_D & 0 & -f_N \\ -f_E & f_N & 0 \end{bmatrix} \tag{5.54}$$

The process noise spectral density is given by the following equation:

$$\mathbf{Q(t)} = \begin{bmatrix} \sigma_{\omega}^{\mathbf{2}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_{\mathbf{f}}^{\mathbf{2}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_{\mathbf{b}_{\omega}}^{\mathbf{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_{\mathbf{b_f}}^{\mathbf{2}} \end{bmatrix} \tag{5.55}$$

The parameter values for this model have been identified using Allan variance and auto-correlation techniques and are given in the table 5.1.

Table 5.1. Parameter values for MEMS IMU

Parameter	Value
σ_{ω_x}	5.47 $deg/hr^{-1/2}$
σ_{ω_y}	6.02 $deg/hr^{-1/2}$
σ_{ω_z}	6.12 $deg/hr^{-1/2}$
σ_{f_x}	0.28 $m/s/hr^{-1/2}$
σ_{f_y}	0.25 $m/s/hr^{-1/2}$
σ_{f_z}	0.27 $m/s/hr^{-1/2}$
$\sigma_{b_{\omega_x}}$	7.1 deg/hr
$\sigma_{b_{\omega_y}}$	5.23 deg/hr
$\sigma_{b_{\omega_z}}$	5.58 deg/hr
$\sigma_{b_{f_x}}$	7.9E-4 m/s^2
$\sigma_{b_{f_y}}$	3.58E-4 m/s^2
$\sigma_{b_{f_z}}$	1.9E-4 m/s^2

5.5.3 Measurement Model

Position and orientation measurements from the vision algorithm described in section 5.1 are used in the corrector step of the Kalman filter. The solution of equation 5.20 is used as a measurement vector. Denoting the measurement vector by $\mathbf{z_v}$, it can be written in component form as:

$$\mathbf{z_v} = \begin{bmatrix} \phi_v & \theta_v & \psi_v & p_{N_v} & p_{E_v} & p_{D_v} \end{bmatrix}^T \tag{5.56}$$

where ϕ_v, θ_v, ψ_v are the roll, pitch and yaw Euler angles of the vehicle orientation and $p_{N_v}, p_{E_v}, p_{D_v}$ are the North, East and Down coordinates of the vehicle position obtained from the vision algorithm.

Estimate of these measurements are available from the state update step using mechanization equations denoted by $\mathbf{z_{IMU}}$:

$$\mathbf{z_{IMU}} = \begin{bmatrix} \phi_{IMU} & \theta_{IMU} & \psi_{IMU} & p_{N_{IMU}} & p_{E_{IMU}} & p_{D_{IMU}} \end{bmatrix}^T \tag{5.57}$$

where each term denotes the corresponding estimates of vision algorithm measurements by the IMU mechanization equations.

At each measurement update, these measurements are compared to get the measurement differences denoted by $\delta\mathbf{z}$:

$$\delta\mathbf{z} = \begin{bmatrix} \phi_{IMU} - \phi_v \\ \theta_{IMU} - \theta_v \\ \psi_{IMU} - \psi_v \\ p_{N_v} - p_{N_{IMU}} \\ p_{E_v} - p_{E_{IMU}} \\ p_{D_v} - p_{D_{IMU}} \end{bmatrix} = \begin{bmatrix} -\delta\phi \\ -\delta\theta \\ -\delta\psi \\ -\delta p_N \\ -\delta p_E \\ -\delta p_D \end{bmatrix} \tag{5.58}$$

These measurement differences can be expressed in terms of orientation and position error states of the system as:

$$\delta\mathbf{z} = \begin{bmatrix} -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times6} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times6} \end{bmatrix} \delta\mathbf{x} = \mathbf{H}\delta\mathbf{x} + \eta_{\mathbf{meas}} \tag{5.59}$$

where $\eta_{\mathbf{meas}} = [\eta_{vis_\phi}\ \eta_{vis_\theta}\ \eta_{vis_\psi}\ \eta_{vis_{p_N}}\ \eta_{vis_{p_E}}\ \eta_{vis_{p_D}}]$ is the measurement noise with variance vector ${\sigma_{\mathbf{vis}}}^2$ given by equation 5.27 and can be written as:

$${\sigma_{\mathbf{vis}}}^2 = R_{nav} \tag{5.60}$$

5.5.4 Discrete Kalman Filter Equations

A discrete Kalman filter has been used to fuse the GPS and IMU measurements. The state space model of the system given by equation 5.52 has the form:

$$\dot{\delta}\mathbf{x} = \mathbf{F}\delta\mathbf{x} + \mathbf{G}\mathbf{w} \tag{5.61}$$

As the data acquisition and processing happens inside the computer in the digital domain, this equation is converted into the discrete form as [27]

$$\delta \mathbf{x_{k+1}} = \mathbf{\Phi_k} \delta \mathbf{x_k} + \mathbf{\Gamma_k} \mathbf{w_k} \tag{5.62}$$

where $\mathbf{\Phi_k}$ is the state transition matrix given by

$$\mathbf{\Phi_k} = e^{\mathbf{F}T} = \mathbf{I} + \mathbf{F}T + \frac{\mathbf{F^2}T^2}{2!} + \dots \tag{5.63}$$

and $\mathbf{\Gamma_k}$ is given by:

$$\mathbf{\Gamma_k} = \mathbf{G}T + \frac{\mathbf{FG}T}{2} + \cdots \tag{5.64}$$

where T is the sampling time interval and $\mathbf{I}$ is an identity matrix.

The discrete form of the process noise covariance is calculated using the process noise spectral density given by equation 5.55 as:

$$\mathbf{Q_k} = \int_0^T \mathbf{\Phi}(\tau)\mathbf{G}\mathbf{Q}(\tau)\mathbf{G}^T \mathbf{\Phi^T}(\tau) d\tau \tag{5.65}$$

For small sampling interval T, the process noise covariance given by equation 5.65 can be approximated as [27]:

$$\mathbf{Q_k} = \mathbf{\Gamma_k} \frac{\mathbf{Q}}{T} \mathbf{\Gamma_k^T} \tag{5.66}$$

The discrete form of the measurement equation 5.59 is given by:

$$\delta \mathbf{z}_k = \mathbf{H}_k \delta \mathbf{x}_k + \eta_{\mathbf{meas}} \tag{5.67}$$

where $\mathbf{H}_k = H$. The process noise variance is denoted by $\mathbf{R}_k$ and is given by:

$$\mathbf{R}_k = \sigma_{\mathbf{vis}}{}^2 \tag{5.68}$$

where $\sigma_{\mathbf{GPS}}{}^2$ is given by equation 5.60.

In the prediction step, the error state and corresponding covariance estimate is extrapolated using the following equations [33]:

$$\delta\hat{\mathbf{x}}_k(-) = \mathbf{\Phi}_{k-1}\delta\hat{\mathbf{x}}_{k-1}(+) \tag{5.69}$$

$$\mathbf{P}_k(-) = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}(+)\mathbf{\Phi}_{k-1}^T + \mathbf{Q_{k-1}} \tag{5.70}$$

As the sample rate of IMU (200 Hz), is much higher than the rate at which vision algorithm provides position and orientation information (30 Hz), the predictor step is repeated until the next vision algorithm measurement is available.

In the corrector step, the vision algorithm measurements are used to form innovations given by equation 5.58, which are compared with the predicted innovations using estimates from the prediction step [33].

$$\delta\hat{\mathbf{x}}_k(+) = \delta\hat{\mathbf{x}}_k(-) + \mathbf{K}_k\left[\delta\mathbf{z}_k - \mathbf{H}_k\delta\hat{\mathbf{x}}_k(-)\right] \tag{5.71}$$

$$\mathbf{P}_k(+) = \left[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right]\mathbf{P}_k(-) \tag{5.72}$$

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1} \tag{5.73}$$

5.5.5 State Correction

After each corrector step of the error state Kalman filter, the state of the system obtained after state update step in subsection 5.5.1 is corrected using the estimates from the error state Kalman filter. For orientation, the corrected body to navigation frame transformation is calculated using the following equation:

$$\mathbf{C}_n^b = (\mathbf{I} - \Upsilon)^{-1}\,\tilde{\mathbf{C}}_n^b \tag{5.74}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{5.75}$$

Using equation 5.3, the corrected Euler angles for vehicle orientation are calculated as:

$$\phi(corr) = \arctan\frac{a_{23}}{a_{33}} \tag{5.76}$$

$$\theta(corr) = -\arcsin a_{13}$$

$$\psi(corr) = \arctan\frac{a_{12}}{a_{11}}$$

Velocity and position states are corrected as follows:

$$\mathbf{v}_e^n(corr) = \mathbf{v}_e^n - \delta\mathbf{v} \tag{5.77}$$

$$\mathbf{p}_e^n(corr) = \mathbf{p}_e^n - \delta\mathbf{p} \tag{5.78}$$

Bias states are corrected using the following equation:

$$\mathbf{b}(corr) = \mathbf{b} + \delta\mathbf{b} \tag{5.79}$$

Data fusion using IMU and vision allows for an efficient implementation of the vision algorithm presented in section 5.1 which is presented in the next section.

5.6 Efficient implementation of the Vision Algorithm Using the Kalman Filter

The number of computations required the vision algorithm described in section 5.1 depends on the total number of pixels of the image included in the pre-defined window w for the solution of equation 5.20. As a map of the environment is available, assuming the position and orientation of the vehicle is known initially, the estimated location of the vehicle and its possible variability at a time in future (time of acquisition of the next vision frame) can be predicted using the prediction step of the Kalman filter. The solution of equation 5.32 gives the predicted orientation and solution of equation 5.34 gives the predicted position of the vehicle. The possible variability of the vehicle location can be calculated from the covariance matrix calculated in the predictor step of the Kalman filter given by equation 5.70.

This information of predicted vehicle location and its possible variability can be translated to predicted locations of the features around the vehicle and their possible variability in the image domain defining a Region Of Interest (ROI) around each of the feature. These ROIs can be used as the pre-defined windows for the solution of equation 5.20 in the vision algorithm. Depending on the number of features in the vehicle vicinity, the identification of ROIs reduces the number of computations required to compute the vision algorithm solution.

Equation 5.11 describes the change in the horizontal and vertical image coordinates when a point p in 3D space undergoes motion defined by vector $[\phi \ \theta \ \psi \ t_x \ t_y \ t_z]^T$ in camera coordinates. Representing this equation in terms of image coordinates:

$$\begin{aligned}
\delta u &= -\frac{uv}{\lambda}\phi + \frac{(u^2+\lambda^2)}{\lambda}\theta - v\psi + \frac{\lambda}{Z}t_x - \frac{u}{Z}t_z \\
&= c_1\phi + c_2\theta + c_3\psi + c_4 t_x + c_5 t_z \qquad (5.80) \\
\delta v &= -\frac{(v^2+\lambda^2)}{\lambda}\phi + \frac{uv}{\lambda}\theta + u\psi + \frac{\lambda}{Z}t_y - \frac{v}{Z}t_z \\
&= d_1\phi + d_2\theta + d_3\psi + d_4 t_y + d_5 t_z \qquad (5.81)
\end{aligned}$$

Here δu and δv represent the shift in u and v coordinates due to the motion vector. The possible variability of the vehicle position and orientation can be derived from the corresponding states of the covariance matrix given by equation 5.70. As the shift δu and δv can be expressed as a linear combination of vehicle motion vector states, as given by equations 5.80 and 5.81, whose co-variance can be derived from the corresponding states in equation 5.70, the variance of a pixel (u, v) along u and v axes in the image domain can be expressed as:

$$R_{\delta u \delta u} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}^T \begin{bmatrix} R_{\phi\phi} & R_{\phi\theta} & R_{\phi\psi} & R_{\phi t_x} & R_{\phi t_z} \\ R_{\theta\phi} & R_{\theta\theta} & R_{\theta\psi} & R_{\theta t_x} & R_{\theta t_z} \\ R_{\psi\phi} & R_{\psi\theta} & R_{\psi\psi} & R_{\psi t_x} & R_{\psi t_z} \\ R_{t_x\phi} & R_{t_x\theta} & R_{t_x\psi} & R_{t_x t_x} & R_{t_x t_z} \\ R_{t_z\phi} & R_{t_z\theta} & R_{t_z\psi} & R_{t_z t_y} & R_{t_z t_z} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} \tag{5.82}$$

$$R_{\delta v \delta v} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{bmatrix}^T \begin{bmatrix} R_{\phi\phi} & R_{\phi\theta} & R_{\phi\psi} & R_{\phi t_y} & R_{\phi t_z} \\ R_{\theta\phi} & R_{\theta\theta} & R_{\theta\psi} & R_{\theta t_y} & R_{\theta t_z} \\ R_{\psi\phi} & R_{\psi\theta} & R_{\psi\psi} & R_{\psi t_y} & R_{\psi t_z} \\ R_{t_x\phi} & R_{t_x\theta} & R_{t_x\psi} & R_{t_x t_y} & R_{t_x t_z} \\ R_{t_z\phi} & R_{t_z\theta} & R_{t_z\psi} & R_{t_z t_y} & R_{t_z t_z} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{bmatrix} \tag{5.83}$$

where $R_{\delta u \delta u}$ and $R_{\delta v \delta v}$ denote the variance of the pixel (u, v) in the image domain. $R_{\phi\phi}$ denotes the variance of vehicle roll, $R_{\phi\theta}$ denotes the covariance of vehicle roll and pitch variables and so on. The values of vehicle orientation covariances can be directly read from the corresponding states in equation 5.70; whereas values of vehicle position covariances as well as orientation-position cross-covariances need to be transformed from the navigation coordinate frame in equation 5.70 to camera coordinate frame. This transformation is done on similar lines as done in section 5.25 as follows.

Using equation 5.25, the predicted covariance in the navigation frame can be written in terms of predicted covariance in the vehicle body coordinate frame as follows:

$$R_{nav}(-) = \begin{bmatrix} R_{\Theta\Theta}(-) & R_{\Theta X_B}(-)(\mathbf{C}_b^n)^T \\ (\mathbf{C}_b^n)R_{X_B\Theta}(-) & (\mathbf{C}_b^n)R_{X_BX_B}(-)(\mathbf{C}_b^n)^T \end{bmatrix} \tag{5.84}$$

Using the above equation, the different components of the vehicle covariance matrix in the vehicle body coordinate frame can be derived as:

$$R_{\Theta X_B}(-) = R_{\Theta X_N}(-)(\mathbf{C}_b^n) = R_{X_B\Theta}(-) \tag{5.85}$$

$$R_{X_BX_B}(-) = (\mathbf{C}_b^n)^T R_{X_NX_N}(-)(\mathbf{C}_b^n) \tag{5.86}$$

Following the same notation as in equation 5.25, $R_{\Theta\Theta}(-)$ represents the predicted orientation covariance, $R_{\Theta X_B}$ represents the predicted orientation-position cross-covariance, $R_{X_B\Theta} = (R_{\Theta X_B})^T$ and $R_{X_BX_B}$ represents the predicted position covariance. The covariance in body coordinate frame can be transformed to covariance in camera coordinate frame using equation 5.24. The following equation shows the predicted covariance matrix in equation 5.70 broken up into the position and orientation covariance matrices described above.

$$\mathbf{P}_k(-) = \begin{bmatrix} R_{\Theta\Theta}(-)_{3\times3} & R_{\Theta V_N}(-)_{3\times3} & R_{\Theta X_N}(-)_{3\times3} & R_{\Theta X_{bias}}(-)_{3\times6} \\ R_{V_N\Theta}(-)_{3\times3} & R_{V_NV_N}(-)_{3\times3} & R_{V_NX_N}(-)_{3\times3} & R_{V_NX_{bias}}(-)_{3\times6} \\ R_{X_N\Theta}(-)_{3\times3} & R_{X_NV_N}(-)_{3\times3} & R_{X_NX_N}(-)_{3\times3} & R_{X_NX_{bias}}(-)_{3\times6} \\ R_{X_{bias}\Theta}(-)_{6\times3} & R_{X_{bias}V_N}(-)_{6\times3} & R_{X_{bias}X_N}(-)_{6\times3} & R_{X_{bias}X_{bias}}(-)_{6\times6} \end{bmatrix} \tag{5.87}$$

Where $R_{V_N V_N}$ represents the covariance of the velocity vector in the navigation frame and $R_{X_{bias} X_{bias}}$ represents the covariance of the bias state vector.

Figure 5.6 shows the procedure to determine the ROI using equations 5.82 and 5.83. The figure shows a feature in the shape of a stop sign along the road. First, a rectangular window defined by diagonal points (u_1, v_1) and (u_2, v_2) completely enclosing the feature is determined using the map. For each of the corners of this rectangular window, a 1 σ uncertainty in u and v directions is obtained using equations 5.82 and 5.83 as:

$$\sigma_{u_{u_i v_j}} = \sqrt{R_{\delta u \delta u}} \tag{5.88}$$

$$\sigma_{v_{u_i v_j}} = \sqrt{R_{\delta v \delta v}} \tag{5.89}$$

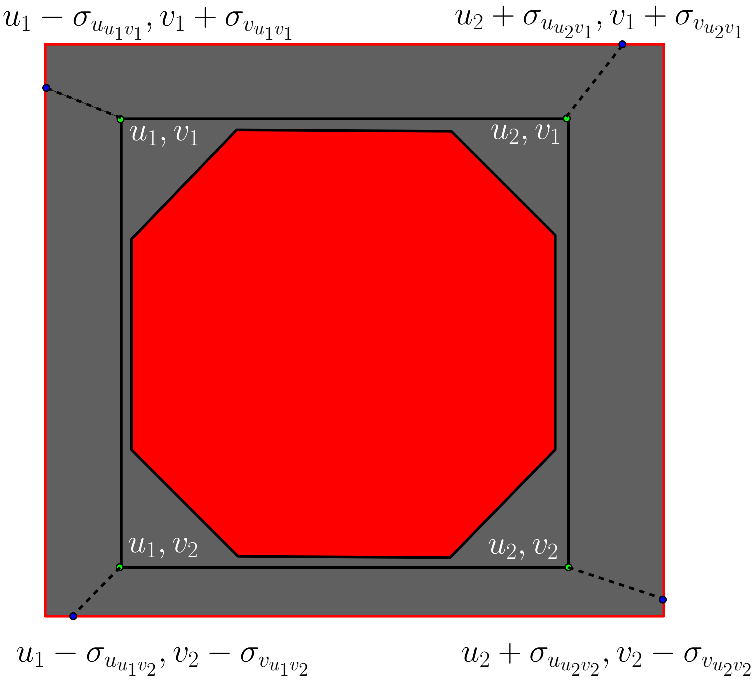


Figure 5.6. Procedure to determine the Region of Interest

Using these uncertainty values, a set of four new points are derived from the four corners of the rectangular window as $(u_1 - \sigma_{u_{u_1 v_1}}, v_1 + \sigma_{v_{u_1 v_1}})$, $(u_2 + \sigma_{u_{u_2 v_1}}, v_1 + \sigma_{v_{u_2 v_1}})$, $(u_1 - \sigma_{u_{u_1 v_2}}, v_2 - \sigma_{v_{u_1 v_2}})$ and $(u_2 + \sigma_{u_{u_2 v_2}}, v_2 - \sigma_{v_{u_2 v_2}})$. Finally, a rectangle fully enclosing these four points is chosen as the ROI as shown by the grey rectangle in figure 5.6.

To verify that the Kalman filter developed to fuse IMU and vision measurements is formulated correctly, numerical simulations are performed and results are presented in the next section.

5.7 Numerical Simulations Using Kalman Filter

To verify the performance of the Kalman filter against known truth values, numerical simulations were conducted. IMU data was extracted from a hypothetical vehicle trajectory and known errors were added to this data following the IMU model given by equations 5.28 and 5.30. The image generated from the true vehicle position was treated as the real image which was registered with the virtual image using the vision algorithm presented in section 5.1 to get the position and orientation measurements.

Figure 5.7 shows the comparison of true and estimated roll, pitch and yaw angles along with the corresponding true errors and estimated 1σ errors. The figure also shows the true and estimated roll, pitch and yaw gyroscope biases. It can be seen from the figure that the roll, pitch and yaw angle profiles overlap each other. The true error is well within the estimated 1σ bound suggesting a conservative selection of factor K in the measurement covariance equation 5.22. Also, roll, pitch and yaw gyroscope biases converge to the true values in approximately 10 seconds.

Figure 5.8 shows the comparison of true and estimated vehicle position. The corre-

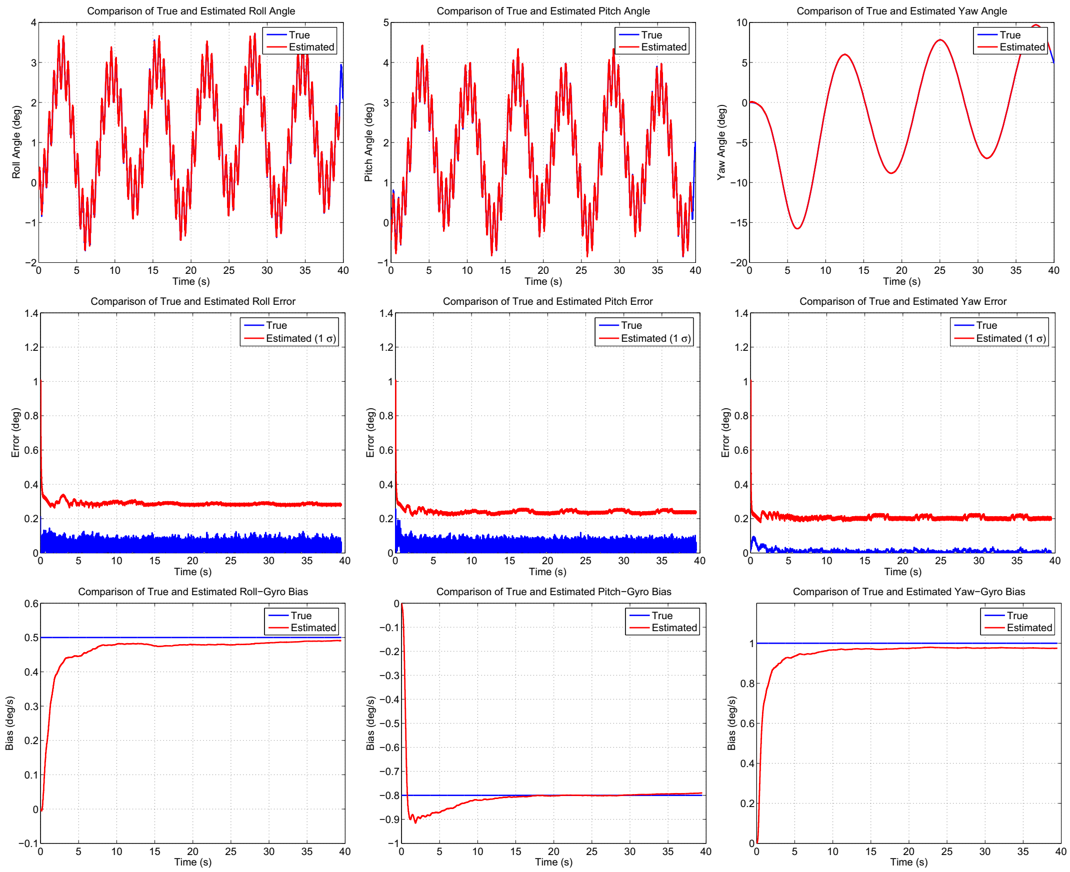
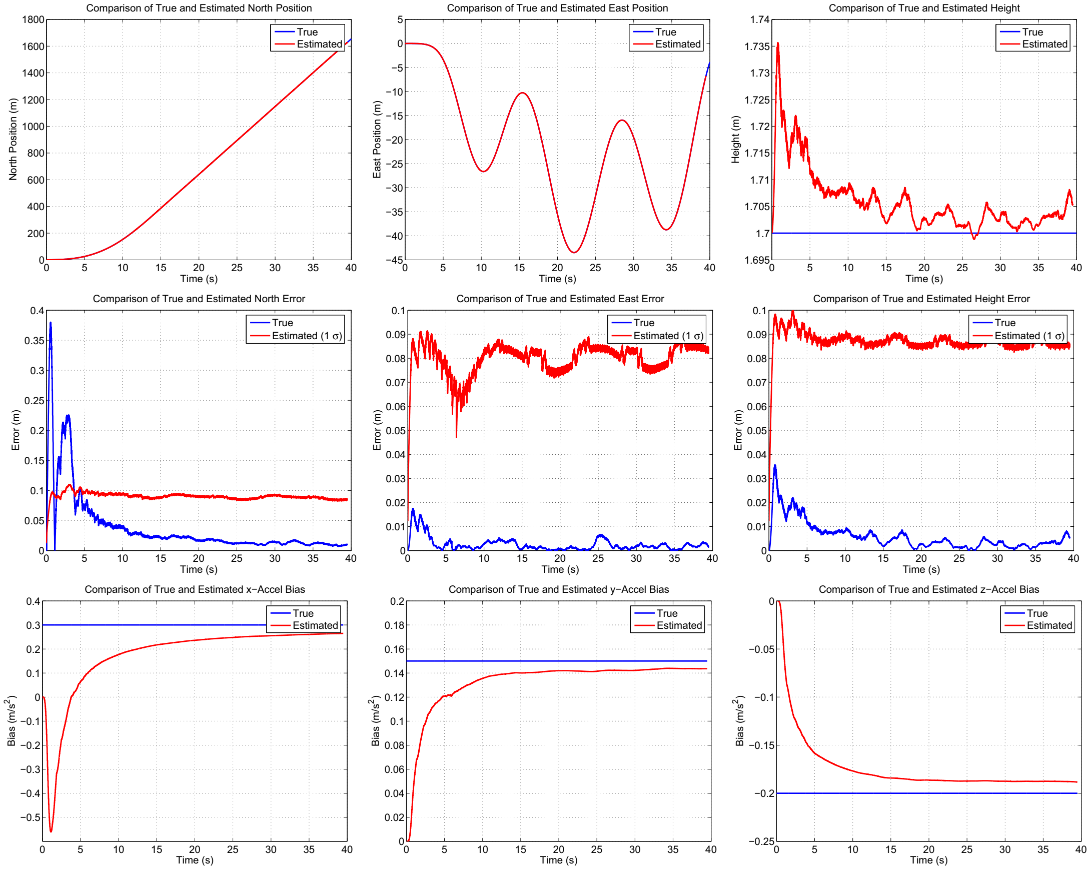


Figure 5.7. Orientation Estimates using Vision/IMU

sponding true errors and estimated 1σ errors are also shown in the figure. It can be seen from the figure that the Kalman filter provides a good estimate of the vehicle position. Also, the true estimation error is well within the estimated 1σ bound. The estimated values of x, y and z accelerometer biases converge to the true value in approximately 10-15 seconds.

5.8 Image Acquisition Time Delay Calculation

It is very important that the measurements from IMU and camera be synchronized in time for the Kalman filter formulation in section 5.5 to give accurate estimation results.


Comparison of True and Estimated North Position

Comparison of True and Estimated East Position

Comparison of True and Estimated Height

Comparison of True and Estimated North Error

Comparison of True and Estimated East Error

Comparison of True and Estimated Height Error

Comparison of True and Estimated x−Accel Bias

Comparison of True and Estimated y−Accel Bias

Comparison of True and Estimated z−Accel Bias

Figure 5.8. Position Estimates using Vision/IMU

The measurements from the IMU and the camera are time stamped immediately after acquisition. As the IMU used is an analog sensor and the A/D converter has a conversion time of the order of micro-seconds, the time stamps associated with IMU measurements can be considered accurate for the sample time step of 5 milliseconds used in this study. The time delay associated with image acquisition from the camera is unknown and needs to be determined. Experimental results showing the determination of this image acquisition delay are presented in the this section.

The image acquisition delay from the moment of image formation on the image sensor to its capture and time stamping via DSP needs to be determined for time synchronization between the IMU and the camera measurements. To calculate the time delay, a LED is toggled at known constant time intervals (1 sec in the experiments) via the DSP. The camera is focussed on this LED and images are captured over time. The difference between the known time of LED toggle and the time stamp on the image when this LED toggle event is captured is the required time delay. The response time of an LED to a signal from the DSP is of the order of nano-seconds and can be neglected [8].

Figure 5.9 illustrates the strategy used to determine this time delay. T_f denotes the sample time of image acquisition. In the present setup the frame rate is equal to $\frac{1}{29.97}$ s for NTSC analog video capture. δ_1, δ_2, δ_3 etc. denote the calculated time delay from the moment of LED toggle to the time when it is captured within an image. In case of no image acquisition delay, figure 5.9 shows that the calculated time delays δ_1, δ_2 and δ_3 are periodic. Two cases "Case 1" and "Case 2" are shown for different alignments of LED toggle period illustrated by the yellow boxes and camera frame capture period illustrated by the blue boxes. It can be seen from the figure that maximum and minimum time delays satisfy the following equation.

$$\begin{aligned} &\delta_{min}(no\ delay) \geq \epsilon,\ where\,\epsilon \ll T_f \\ &\delta_{max}(no\ delay) < T_f \end{aligned} \tag{5.90}$$

Figure 5.9 also shows the case when there is a image acquisition time delay denoted by T_d. Again, two cases "Case 3" and "Case 4" are shown for different alignments of LED toggle period and camera frame capture period. The calculated time delays δ_1, δ_2

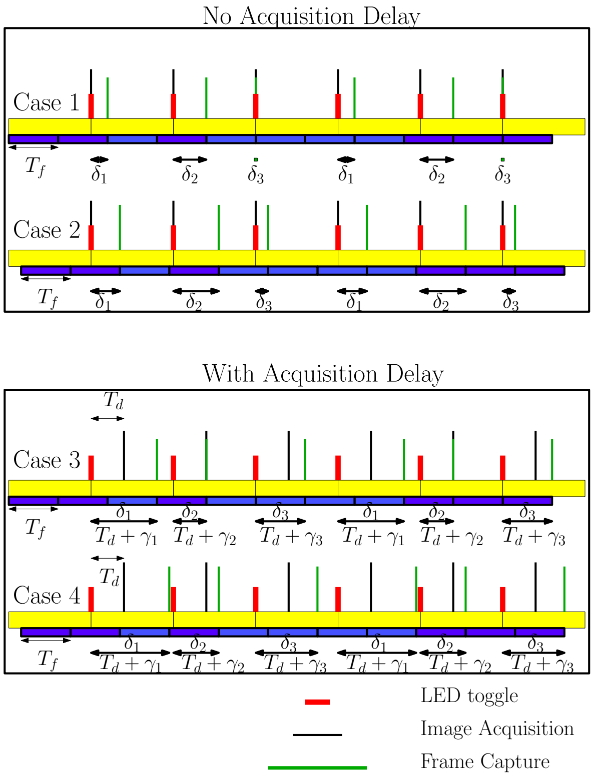


Figure 5.9. Image Acquisition Delay Calculation Strategy

and δ_3 are periodic in this case also and satisfy the following equation.

$$\delta_{min}(delay) \geq T_d + \epsilon,\ \ where\, \epsilon \ll T_f$$

$$\delta_{max}(delay) < T_d + T_f \tag{5.91}$$

Figure 5.10 shows pictures from the experiment conducted to determine the image acquisition time delay. The images show the time stamp in the title in milli-seconds. The LED is toggled at 1 sec integer time intervals. So, the exact time of LED toggle within the DSP is 1000 ms, 2000 ms, 3000 ms and so on. In figure 5.10, horizontal rows show the images around LED toggle instants of 6000 ms, 23000 ms, 24000 ms and 28000 ms.

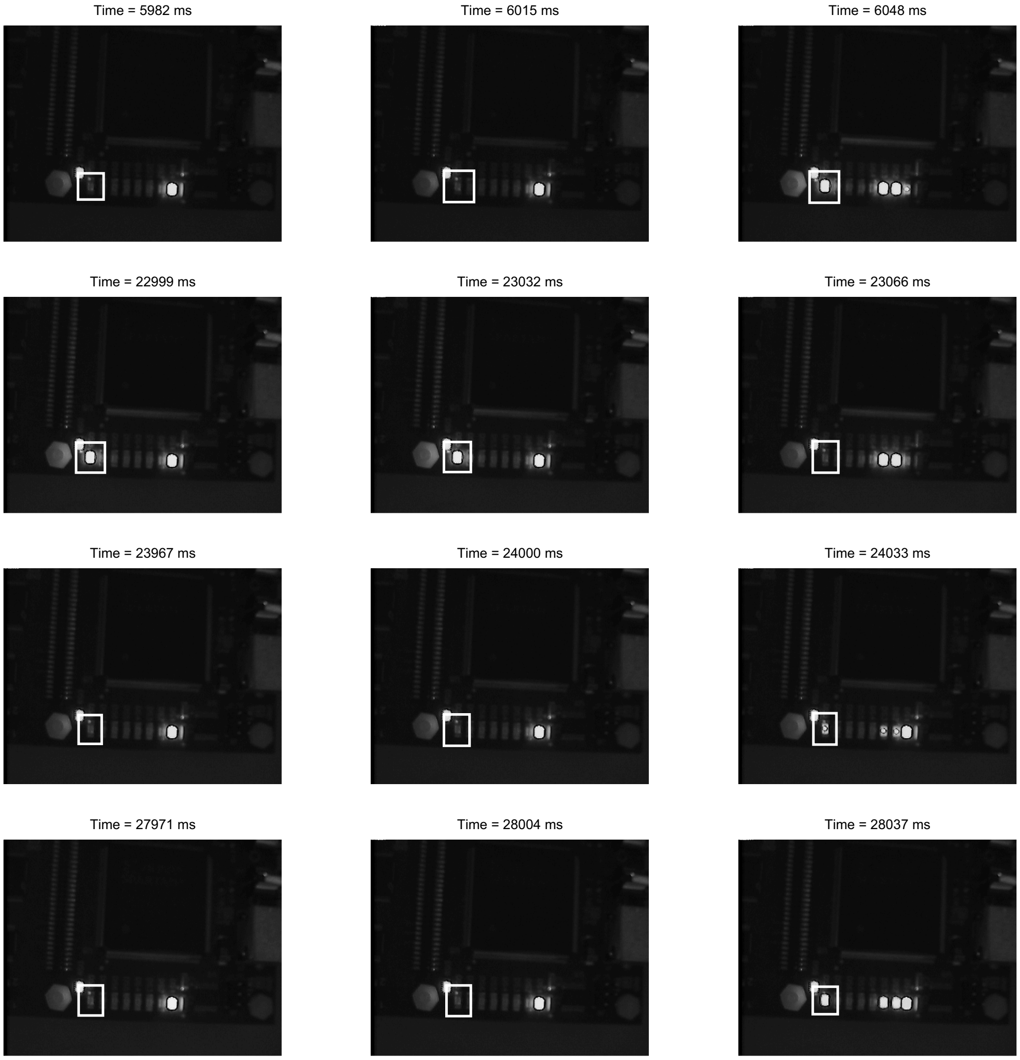


Figure 5.10. Experimental Determination of Image Acquisition Delay

Only one LED is considered (although the image contains many) and enclosed within a white rectangle for illustration. In each row of images in the figure, the right-most image captures the LED toggle event. So, the difference between the time stamp of the right-most image in each row and the integer time of LED toggle is the calculated time delay for each row. Time delay values for the figure 5.10 from top to bottom rows are 48 ms, 66 ms, 33 ms and 37 ms respectively. The measured time delay is periodic and a graph of calculated time delay versus LED toggle time is shown in figure 5.11. From the graph, the image acquisition time delay T_d is calculated to be 33 ms approximately. This image acquisition delay is taken into account while implementing the Kalman filter to synchronize IMU and camera measurements.

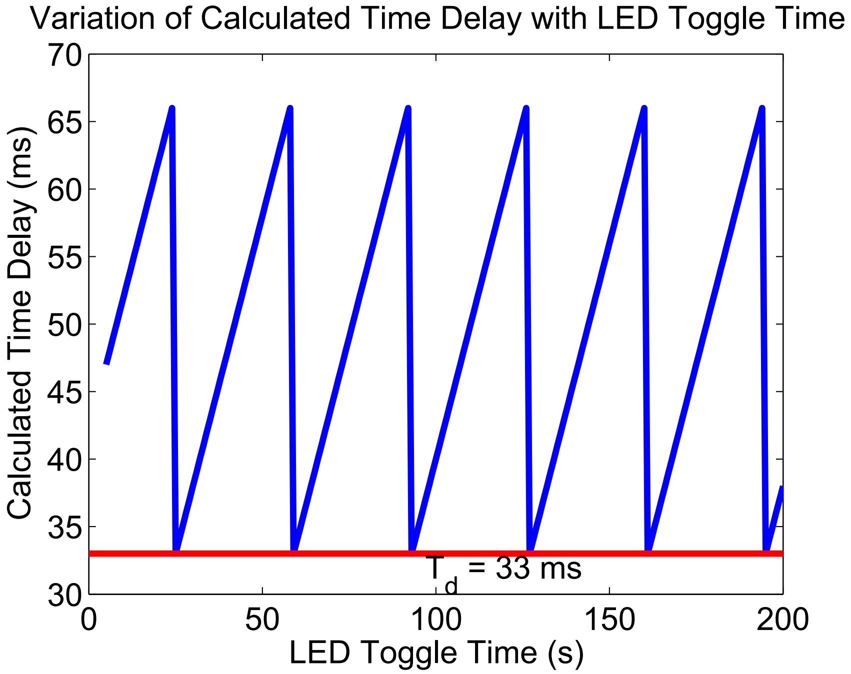


Figure 5.11. Variation of Measured Time Delay with LED Toggle Time

Experiments are conducted at the PTI test track skid pad area to experimentally verify the localization technique presented. The results are presented in the next section.

5.9 Experimental Results

Experiments were conducted at PTI test track facility to experimentally verify the vision algorithm presented in section 5.1 as well as the performance of the Kalman filter presented in section 5.5. As described in section 5.3, road markers and road signs were created to serve as features in vehicle environment. Time-stamped data is collected from an IMU and a camera at a rate of 5ms and 33.34ms (29.97 Hz frame rate) respectively. This collected data is processed off-line using the Kalman filter presented in section 5.5 to generate the estimates of vehicle position and orientation. Image acquisition delay determined in section 5.8 is compensated in the time stamps of the stored image frame data. Vehicle position and orientation estimates from a highly accurate GPS/IMU system are also recorded at the same time for comparison.

Figure 5.12 shows an image along the vehicle path, the corresponding registered rendered image as well as the overlay of the two images to illustrate the match between features in real and rendered images.

Figure 5.13 shows the variation of ROI with time as the Kalman filter converges to an approximately steady state covariance value from a initially high covariance. The selection of ROI was described in section 5.6 and the ROI windows in figure 5.13 can be seen to shrink in size with decreasing state co-variance of the Kalman filter. It can also be seen that, in this particular case, the ROI area is much smaller than the total image area and thus leads to a very large reduction in the computational effort to compute the solution using equation 5.20.

Figure 5.14 shows the estimated roll, pitch and yaw profiles of the vehicle trajectory during the experimental run. The roll, pitch and yaw estimates from the GPS/IMU system are also plotted for comparison. Although the variation in roll and pitch angles is very small (1 to 2 degrees), the trends in Vision/IMU estimates match those from the GPS/IMU system. The yaw angle profiles from the two systems match each other very closely. The figure also shows the deviation in estimates between the Vision/IMU system and the GPS/IMU system along with the 3σ estimated error bound of the Vision/IMU

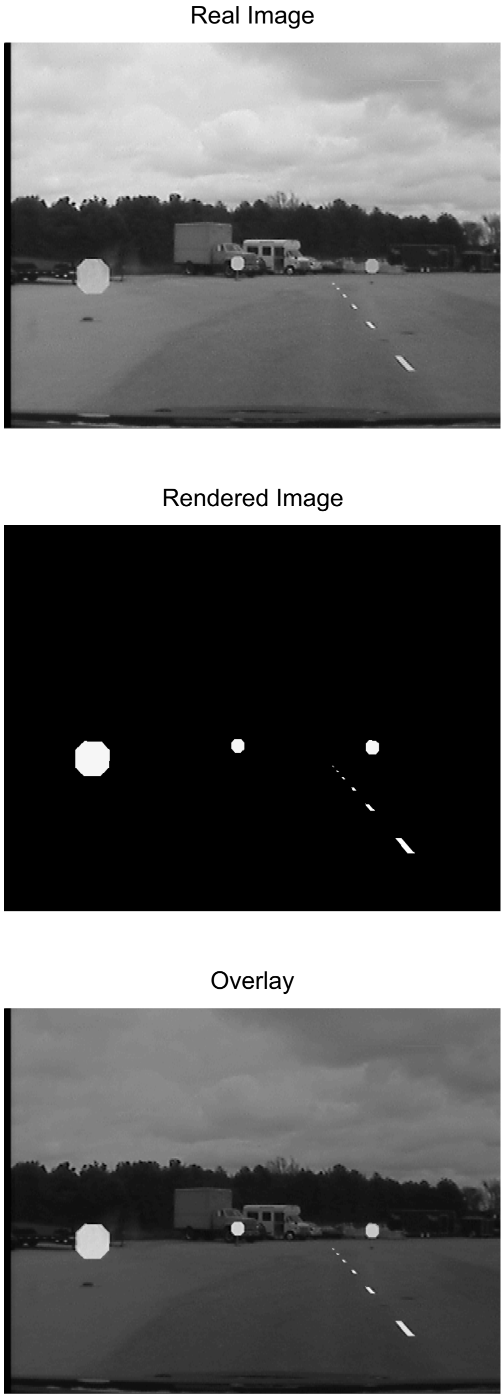


Figure 5.12. Real, Rendered and Overlaid Images

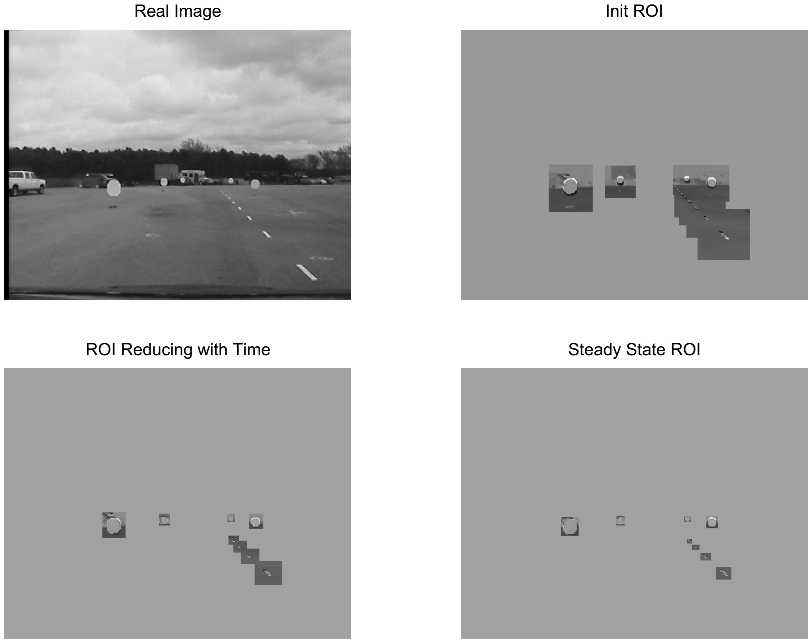


Figure 5.13. Variation of ROI over time

system. It can be seen from the figure that the magnitude of roll, pitch and yaw deviations is less than a degree throughout the duration of the experiment. The Kalman filter estimates of roll, pitch and yaw gyroscope biases are also plotted in the figure.

The estimated position of the vehicle in navigation co-ordinates is shown in figure 5.15 along with the position estimates from the GPS/IMU system for comparison. The figure shows that the position estimates from the two systems follow each other very closely. The deviation between the estimates from the two systems along with the 3σ estimated error bound of the Vision/IMU system is also shown in the figure. A careful look at the figure reveals that the magnitude of the North, East and Height deviation is less than 25 cm throughout the duration of the experiment. Between times 68 seconds and 72 seconds, the actual error exceeds the estimated error for East position estimates by about 10cm. The cause of this discrepancy is attributed to map errors, camera

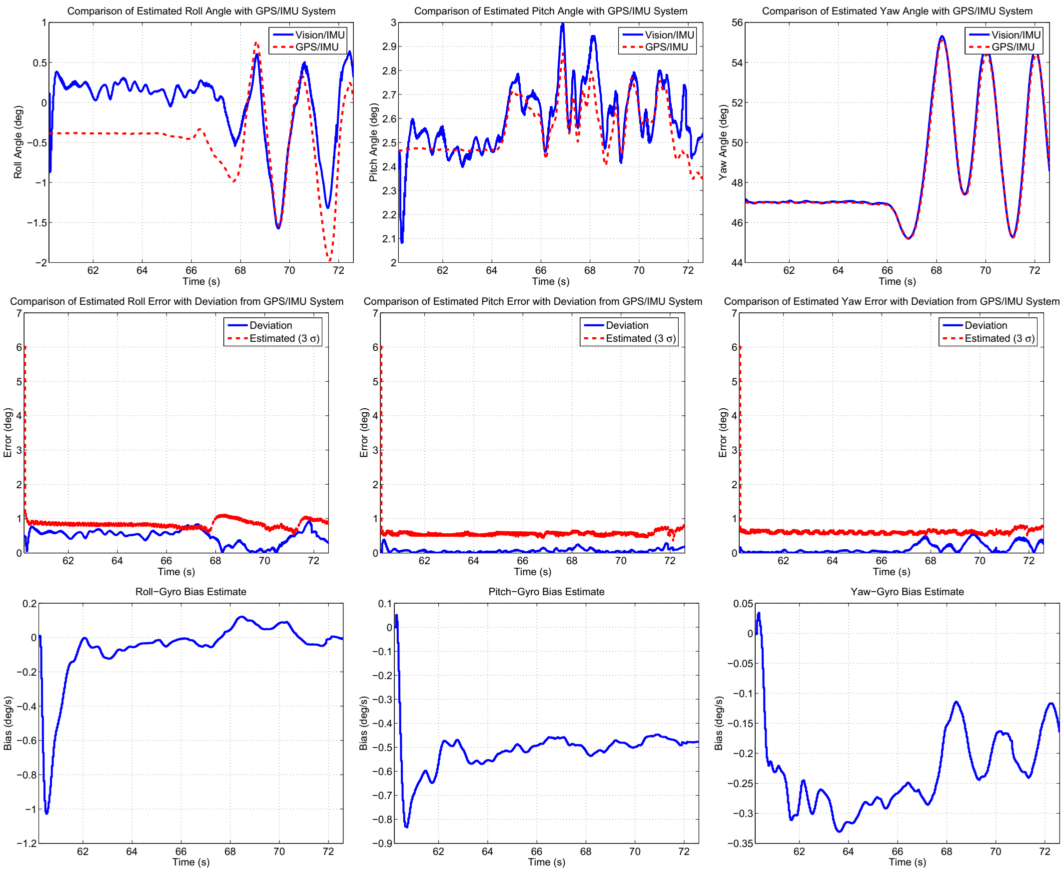


Figure 5.14. Orientation Estimates using Vision/IMU: Run1

calibration errors as well as the sparse feature set used in experiments. Vision/IMU Kalman filter estimates of accelerometer biases are also shown in the figure.

Figures 5.16 and 5.17 show the results for another run of the experiment with similar interpretations.

5.10 Conclusions

This chapter has demonstrated the use of near field features such as road markers and road signs to estimate vehicle position as well as orientation. ROI windows are selected using the map, current location and uncertainty in the current location of the vehicle

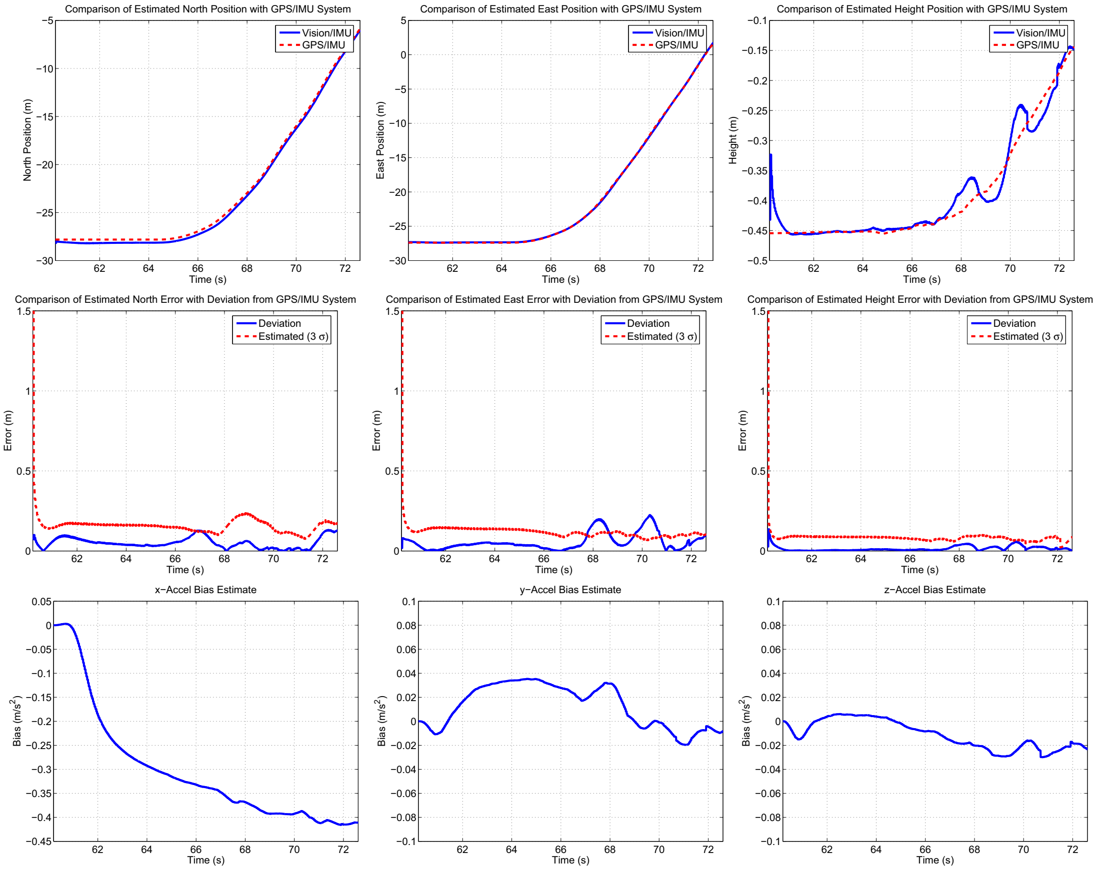


Figure 5.15. Position Estimates using Vision/IMU: Run1

and a Gauss-Newton method is applied to minimize the sum of squared image intensities between real and rendered images within the ROI. An error state kinematic Kalman filter using IMU measurements in the predictor step and vision algorithm measurements in the corrector step was used to provide estimates of position and orientation of the vehicle. Comparison of these estimates with measurements from a high quality GPS/IMU system shows close agreement. The Vision/IMU Kalman filter implemented in this paper can estimate the vehicle orientation to sub-degree accuracy and the vehicle position with an accuracy of 25cm approximately.

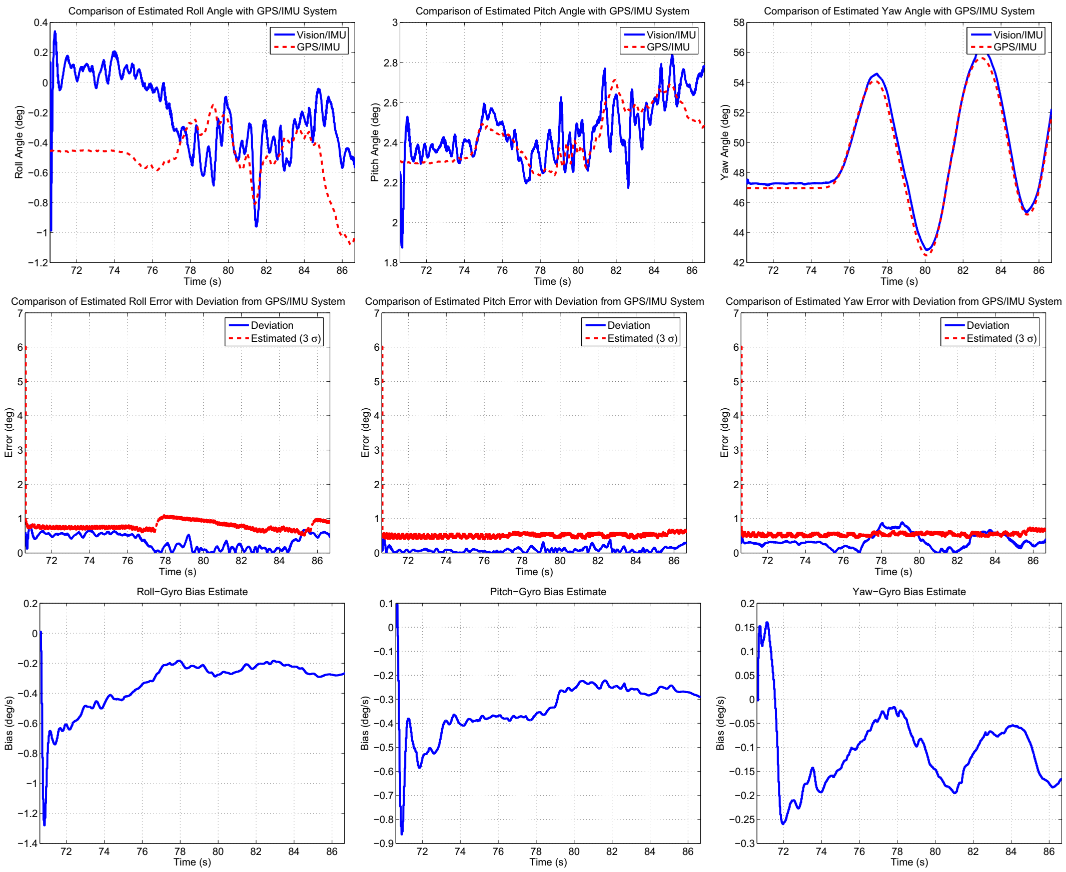


Figure 5.16. Orientation Estimates using Vision/IMU: Run2

Notation

p_j	$= (X_j, Y_j, Z_j)$ 3D point in space in camera coordinates, p. 146
(u_j, v_j)	Image coordinates of point p_j, p. 147
p'_j	$= (X'_j, Y'_j, Z'_j)$ 3D coordinates of the point p_j in camera coordinates after the camera undergoes motion $[\phi\ \theta\ \psi\ t_x\ t_y\ t_z]$, p. 146
(u'_j, v'_j)	Image coordinates of point p'_j, p. 148
$(\delta u_j, \delta v_j)$	Change in image coordinates when 3D point p_j moves to 3D point p'_j in camera coordinates, p. 148
$\boldsymbol{R}$	Rotation matrix representing coordinate transformation when camera rotates by ϕ, θ, ψ Euler angles, p. 146
λ	Focal length of the camera, p. 147

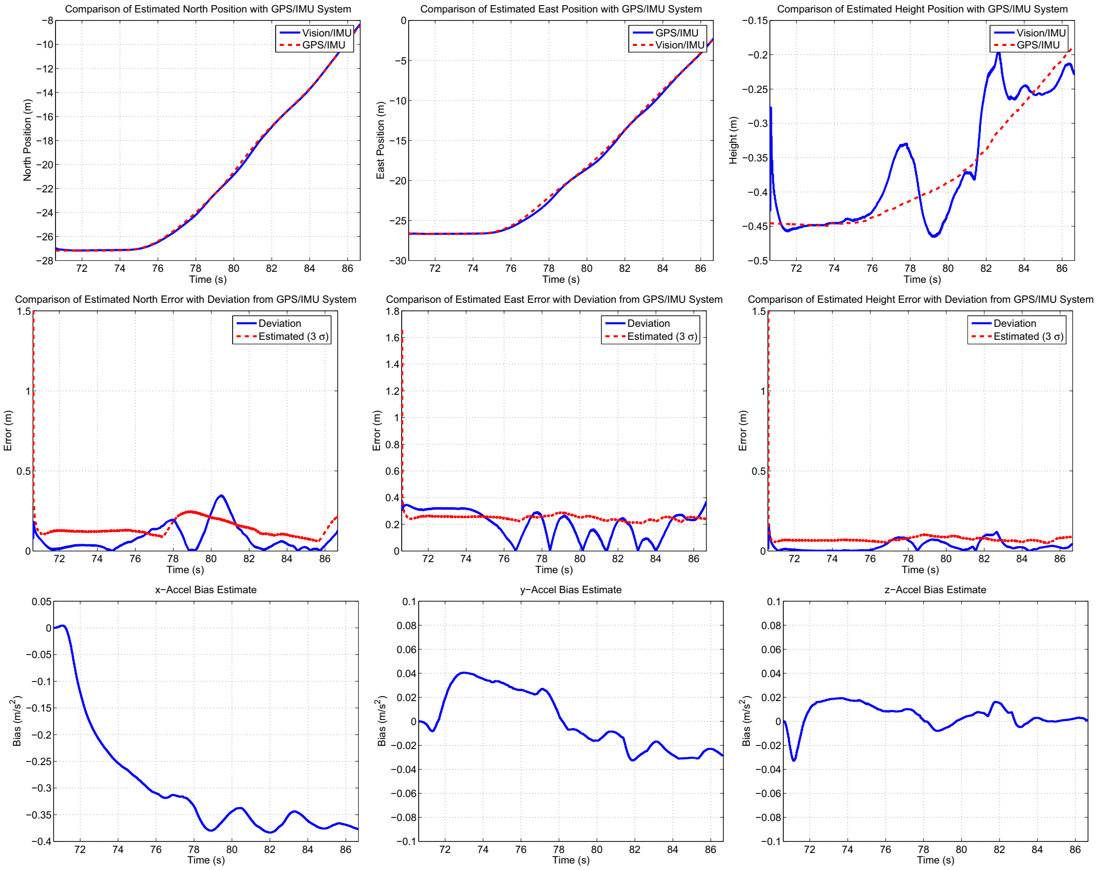


Figure 5.17. Position Estimates using Vision/IMU: Run2

$I_{ren}(u_j, v_j)$	Intensity of pixel (u_j, v_j) in the rendered image, p. 146
$I_{real}(u_j, v_j)$	Intensity of pixel (u_j, v_j) in the real image, p. 146
e_j	Difference in intensity for pixel (u_j, v_j) between real and rendered images, p. 146
z	$= [\phi\ \theta\ \psi\ t_x\ t_y\ t_z]^T$ Vector denoting camera motion, p. 149
R_{cam}	Measurement covariance of the vision algorithm, p. 152
X_{cam}	Vector in camera frame, p. 152
X_B	Vector in vehicle body frame, p. 152
$\mathbf{C}^b_{cam}$	Transformation to convert from camera frame to body frame, p. 152
$R_{\Theta\Theta}$	Covariance of the camera orientation, p. 153
$R_{\Theta X_{cam}}$	Cross-covariance of camera position and orientation in camera coordinate frame, p. 153

$R_{X_{cam}X_{cam}}$	Covariance of camera position in camera coordinate frame, p. 153
$R_{\Theta X_B}$	Cross-covariance of camera position and orientation in vehicle body coordinate frame, p. 153
$R_{X_B X_B}$	Covariance of camera position in vehicle body coordinate frame, p. 153
R_{nav}	Measurement covariance of vision algorithm in the navigation frame, p. 154
$\tilde{\omega}$	Raw measurement from the gyroscope, p. 159
ω	True rotation rate of the vehicle, p. 159
b_ω	Gyro bias, p. 159
η_ω	Gyro noise, p. 159
$\tilde{f}^b$	Raw measurement from the accelerometer, p. 159
f^b	True acceleration acting on the vehicle in the body frame, p. 159
b_{f^b}	Accelerometer bias, p. 159
η_f	Accelerometer noise, p. 159
ω_{ib}^b	Vector representing the rotational velocity of the vehicle with respect to the inertial frame expressed in the body frame, p. 161
ω_{in}^b	Vector representing the rotational velocity of the navigation frame with respect to the inertial frame expressed in the body frame, p. 161
$\mathbf{v}_e^n$	Vector representing the velocity of the vehicle with respect to Earth in the navigation frame. p. 161
$\mathbf{f}^n$	Accelerations experienced by the vehicle resolved into the navigation frame. p. 161
$\mathbf{f}^b$	Accelerations experienced by the vehicle in the body frame p. 161
$\mathbf{C}_b^n$	Transformation from the body frame to the navigation frame. p. 162
ϕ, θ, ψ	Euler angles determining vehicle orientation p. 161
g	Average gravitational acceleration in the vertical direction at the surface of the Earth. p. 161
$\mathbf{p}^n$	$= (x_N \; x_E \; x_D)^T$ Vector representing vehicle position in the navigation frame. p. 161
$\mathbf{v}_e^n$	$= (v_N \; v_E \; v_D)^T$ Vector representing the velocity of the vehicle with respect to Earth in the navigation frame. p. 161

$\boldsymbol{\Psi}$ $= (\delta\alpha\ \delta\beta\ \delta\gamma)^T$ Vector of misalignment angles between true orientation and estimated orientation of the vehicle. p. 162

$\delta\mathbf{v}$ $= (\delta v_N\ \delta v_E\ \delta v_D)^T$ Vector of velocity error states in North, East and Down directions. p. 163

$\delta\mathbf{p}$ $= (\delta p_N\ \delta p_E\ \delta p_D)^T$ Vector of position error states in North, East and Down directions. p. 163

$\delta\mathbf{b}_\omega$ $= [\delta b_{\omega_x}\ \delta b_{\omega_y}\ \delta b_{\omega_z}]^T$ The vector of errors in gyroscope bias estimates in the body frame, p. 164

$\delta\mathbf{b_f}$ $= [\delta b_{f_x}\ \delta b_{f_y}\ \delta b_{f_z}]^T$ Vector of errors in accelerometer bias estimates in the body frame, p. 164

$\delta\mathbf{x}$ $= (\boldsymbol{\Psi}\ \delta\mathbf{v}\ \delta\mathbf{p}\ \delta\mathbf{b}_\omega\ \delta\mathbf{b_f})^T$ Vector of error states to be estimated using the Kalman filter, p. 164

$\boldsymbol{\Upsilon}$ Matrix of misalignment angles between true and estimated orientation of the vehicle. p. 163

$\delta\omega_{ib}^b$ vector denoting the error in the rotation rates given by the IMU. p. 162

$\omega^{\mathbf{b}}$ $= \begin{bmatrix} \omega_x & 0 & 0 \\ 0 & \omega_y & 0 \\ 0 & 0 & \omega_z \end{bmatrix}$ The matrix of corrected gyroscopes measurements calculated using equation 5.33

$\mathbf{f^b}$ $\begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & f_z \end{bmatrix}$ Matrix of corrected accelerometer measurements calculated using equation 5.41.

η_ω $= [\eta_{\omega_x}\ \eta_{\omega_y}\ \eta_{\omega_z}]^T$ Gyroscope noise vector, p. 164

$\sigma_\omega^{\mathbf{2}}$ $= \begin{pmatrix} \sigma_{\omega_x}^2 & 0 & 0 \\ 0 & \sigma_{\omega_y}^2 & 0 \\ 0 & 0 & \sigma_{\omega_z}^2 \end{pmatrix}$ Gyroscope noise variance Matrix, p. 165

$\eta_{\mathbf{f}}$ $= [\eta_{f_x}\ \eta_{f_y}\ \eta_{f_z}]^T$ Accelerometer noise vector, p. 164

$\sigma_{\mathbf{f}}^{\mathbf{2}}$ $= \begin{pmatrix} \sigma_{f_x}^2 & 0 & 0 \\ 0 & \sigma_{f_y}^2 & 0 \\ 0 & 0 & \sigma_{f_z}^2 \end{pmatrix}$ Accelerometer noise variance vector, p. 165

$\eta_{\mathbf{b}_\omega}$ $= [\eta_{b_{\omega_x}}\ \eta_{b_{\omega_y}}\ \eta_{b_{\omega_z}}]^T$ Gyroscope bias noise vector, p. 164

$\sigma_{\mathbf{b}_\omega}^{\mathbf{2}}$ $= \begin{pmatrix} \sigma_{b_{\omega_x}}^2 & 0 & 0 \\ 0 & \sigma_{b_{\omega_y}}^2 & 0 \\ 0 & 0 & \sigma_{b_{\omega_z}}^2 \end{pmatrix}$ Gyroscope bias noise variance vector, p. 165

$\eta_{\mathbf{b_f}}$	$= [\eta_{b_{fx}}\ \eta_{b_{fy}}\ \eta_{b_{fz}}]^T$ Accelerometer bias noise vector, p. 164
$\sigma^{\mathbf{2}}_{\mathbf{b_f}}$	$= \begin{pmatrix} \sigma^2_{b_{fx}} & 0 & 0 \\ 0 & \sigma^2_{b_{fy}} & 0 \\ 0 & 0 & \sigma^2_{b_{fz}} \end{pmatrix}$ Accelerometer bias noise variance vector, p. 165
$\mathbf{z_v}$	$= [\phi_v\ \theta_v\ \psi_v\ p_{N_v}\ p_{E_v}\ p_{D_v}]^T$ Measurements from the vision algorithm. 6DOF vehicle location, p. 166
$\mathbf{z_{IMU}}$	$= [\phi_{IMU}\ \theta_{IMU}\ \psi_{IMU}\ p_{N_{IMU}}\ p_{E_{IMU}}\ p_{D_{IMU}}]^T$ Estimate of the vision algorithm measurements from the IMU measurements, p. 166
$\delta\mathbf{z}$	$= -[\delta\phi\ \delta\theta\ \delta\psi\ \delta p_N\ \delta p_E\ \delta p_D]^T$ Difference between vision algorithm measurements and the estimate of these measurements from the IMU measurements, p. 167
$\eta_{\mathbf{meas}}$	$= [\eta_{vis_\phi}\ \eta_{vis_\theta}\ \eta_{vis_\psi}\ \eta_{vis_{p_N}}\ \eta_{vis_{p_E}}\ \eta_{vis_{p_D}}]$ Vision algorithm measurement noise vector, p. 167
$\sigma_{\mathbf{vis}}{}^2$	Vision algorithm measurement noise covariance. p. 167
$\mathbf{\Phi_k}$	State transition matrix, p. 168
T	Sampling time of the IMU (5 ms), p. 168
$\mathbf{Q_k}$	Process noise covariance for the Kalman filter. p. 168
$\mathbf{R}_k$	Measurement noise covariance for the Kalman filter. p. 169
$\delta\hat{\mathbf{x}}_k(-)$	Error state predicted estimates of the Kalman filter. p. 169
$\delta\hat{\mathbf{x}}_k(+)$	Error state corrected estimates of the Kalman filter. p. 169
$R_{\delta u \delta u}$	Variance in the u-coordinate of the pixel (u, v) in the image domain derived from the variance of the solution of the Kalman filter, p. 173
$R_{\delta v \delta v}$	Variance in the v-coordinate of the pixel (u, v) in the image domain derived from the variance of the solution of the Kalman filter, p. 173
$R_{nav}(-)$	Predicted covariance in the navigation frame, p. 174
$\mathbf{P}_k(-)$	Covariance of the Kalman filter after the predictor step, p. 174

Chapter 6

Conclusions

Accurate vehicle localization is a requirement for driver assistance systems and autonomous navigation to enhance the safety and throughput of advanced highway systems. The availability, accuracy and reliability of the sensors used for vehicle localization varies depending on the sensor requirements (e.g. line of sight to satellites for GPS and visibility for vision sensors) as well as the sensitivity of the sensor to external conditions (e.g. RF jamming for GPS and weather/lighting conditions for vision sensors). This dissertation focussed on vehicle localization using an Inertial Measurement Unit (IMU), GPS and a monocular camera along with a map of the environment. Emphasis has been placed on using low-cost sensors available in the market today to investigate the feasibility of using these technologies in commercial automobiles.

The sensors used in this dissertation complement each other to enhance the accuracy of vehicle localization in addition to having different failure modes to increase the robustness of the system. For example, GPS provides the initial position required to initialize the IMU and periodically corrects the IMU solution to estimate the accelerometer and gyroscope biases whereas the IMU provides a fast data rate thus compensating for the slow data rate of the GPS. Similarly, vision sensors along with a map of the environment provide periodic corrections to the IMU solution whereas the IMU helps the vision algorithm by predicting the location and possible variability of the features in the map.

GPS provides the 3DOF position and horizontal velocity of the vehicle. This dissertation has investigated the feasibility of using a low-cost Commercial Off-The-Shelf (COTS) Micro Electro Mechanical Systems (MEMS) IMU along with a low-cost COTS GPS receiver to estimate the vehicle orientation in addition to the 3DOF position provided by the GPS. GPS and IMU error sources are characterized by comparing the measurements from low-cost GPS and IMU to measurements from a highly accurate GPS/IMU system and first order Gauss-Markov models are chosen to model these error sources. It was found that the GPS velocity measurements are not accurate for speeds below 5mph. The comparison of error sources between the low-cost IMU and a tactical grade IMU shows the magnitude of difference between the various error sources. A error state Kalman filter framework is then developed to estimate vehicle orientation, velocity, position, accelerometer and gyroscope biases and scale factors. It was seen that in the absence of sufficient excitation the bias and scale factor errors are not observable individually although the combination of the two may be observable. It was also seen that vehicle yaw was not observable in the absence of longitudinal and lateral accelerations of the vehicle, for example during straight line driving. Velocity direction was used to improve the yaw angle estimates during periods of low longitudinal and lateral accelerations. Numerical simulations were used to verify the correctness of the Kalman filter formulation. Experiments at the Pennsylvania Transportation Institute (PTI) test track facility confirm the feasibility of a low-cost GPS/IMU combination to estimate vehicle orientation in addition to vehicle position. It was observed that the accuracy of the GPS/IMU localization estimates is limited by the availability and accuracy of GPS position and velocity measurements. To this effect, alternative sensors to measure vehicle position and orientation were sought to improve the accuracy of vehicle localization as well as add redundancy to the system.

An alternative method to measure vehicle orientation was identified as terrain aided orientation estimation using vision. A terrain map was generated using National Elevation Dataset (NED) from the United States Geological Survey (USGS) website. Horizon curves extracted from virtual images generated by rendering the terrain map were registered with horizon lines from a real camera to measure the orientation of the vehicle. The rough position estimate required was obtained from a low-cost GPS. An extended Kalman filter using rate measurements from an IMU in the prediction step and measurements from the real-virtual horizon registration in the corrector step was used for vehicle orientation estimation along with the estimation of gyro biases. Numerical simulations as well as experiments at the PTI test track were used to verify this technique for vehicle orientation estimation.

The terrain aided orientation estimation method using far field features e.g. horizon was then extended to use near field features to estimate vehicle position in addition to orientation. An iterative gradient based method was used to minimize the sum of squared intensities between the rendered and real images to measure the vehicle location. A Kalman filter framework was used to fuse the measurements from an IMU and the vision algorithm. The IMU was used to predict the location of features as well as their possible variability in the next frame of the vision stream to define a Region Of Interest (ROI). The vision algorithm was only implemented on the ROI instead of the whole image; thus significantly reducing the computational effort required. Numerical simulations confirmed the validity of the Kalman filter using measurements from the IMU and the vision algorithm. Features were created in the form road markers and road signs at the PTI test track, and mapped using a highly accurate GPS/IMU system. Experiments were conducted to estimate the vehicle location and the estimates were compared to estimates from a highly accurate GPS/IMU system placed side by side to the vision/IMU system. The estimates from these two systems show a close match.

6.1 Summary of Contributions

6.1.1 Investigation of Low-Cost GPS/IMU Integration

The performance of a low-cost COTS GPS/IMU combination was investigated. The error sources present in these systems were characterized and then modeled using stochastic models. A Kalman filter framework was developed to fuse the measurements from the GPS and the IMU. The performance of the system was tested by comparison of vehicle location estimates with estimates from a highly accurate GPS/IMU system. Sub-degree accuracy was achieved for vehicle roll and pitch estimates. Yaw angle estimation accuracy of 3.5 degrees was achieved. It was found that the position can be estimated to an accuracy of 3 to 4 meters.

6.1.2 Terrain-Aided Orientation Estimation Using Far Field Features

A technique to measure vehicle orientation using far field terrain features was developed assuming that a rough estimate of vehicle position is known. USGS Digital Elevation Maps (DEM) were used to create a terrain map for the vehicle. Far field features were extracted from rendered images of the terrain as well as the real images from a camera fixed to the vehicle. Horizon curves were used as the far field features in this study. A grid based method was used to match these horizon curves to estimate vehicle orientation. Experiments were carried out to verify the performance of this technique. Sub-degree accuracy was achieved for orientation angle estimates using this technique.

6.1.3 Position and Orientation Estimation Using Near Field Features

A technique to track vehicle position and orientation was developed using a map of the near field features. An iterative gradient based method was used to register the ROI in real and rendered images. A Kalman filter framework was used again to combine the measurements from an IMU and the vision algorithm. Road markers and road signs were used as features to emulate a vehicle going along a highway. Numerical simulations as well as experiments were used to verify the performance of this technique. Sub-degree orientation accuracy and position accuracy of 25cm was obtained using this technique.

6.2 Recommendations for Future Work

The research in this dissertation suggests several future research directions. Several of these are listed below.

6.2.1 Additions/Modifications to the Vision Algorithm

6.2.1.1 Robustness in Vision Algorithm

The vision algorithm can be modified to include color information. The color information and reflective material properties can be learnt and adjusted over time. A metric to measure the difference between real and rendered images for each feature is needed. One such possible metric is normalized sum of squared intensity difference. This metric can be used to detect the presence of obstacles along a viewing direction. It can also be used to correct the map for small mapping errors as well as variation of feature locations over time.

6.2.1.2 Multiple Cameras

Multiple cameras can be used to increase the number of features in view and increase the accuracy of localization estimates as well as decouple the position and orientation estimation problems. The only requirement for using multiple cameras is that the relative position and orientation of the cameras and the IMU be known. A total station, which is an instrument used in modern surveying, can be used to determine the required relative position and orientation. For example, for the localization of a vehicle going along a highway, a down facing camera looking directly at road markers can be used measure the vehicle position very accurately, whereas a forward facing camera can converge to the orientation estimates really fast once the position is known accurately.

6.2.1.3 Performance with a photo-realistic model of the environment

Photo-realistic models of cities have been generated in the past decade. The performance and robustness of the vision algorithm needs to investigated with such photo-realistic model.

6.2.1.4 Incorporation of Landmark Identification

The vision algorithm as implemented is essentially a tracking algorithm, which tracks the vehicle position and orientation starting from a known initial location. To increase the robustness of the algorithm, landmark identification needs to be incorporated into the algorithm. Scale Invariant Feature Identification (SIFT) [64] is one such technique which can be run at a much lower rate in the background of the vision algorithm for landmark identification.

6.2.2 GPS/IMU/Vision-Map Integration

Low-cost GPS can be used to provide rough estimates of position to the vision algorithm to make it more robust. Stochastic methods can be used to model errors in the GPS measurements and vision algorithm measurements can be used to correct the GPS measurements providing estimates of GPS measurement errors. Integration of IMU, Vision and GPS can provide better estimates of vehicle location along with adding redundancy to the system providing fault detection capabilities.

6.2.3 Vehicle Model to include vehicle dynamics

If the driver inputs are measured, a vehicle model can be used to estimate some of the vehicle states. These estimates can be used to aid other localization sensors, e.g. the vision algorithm used in this research. Additionally, online parameter estimation can be done using localization estimates from other sensors to refine the parameter estimates of the vehicle model.

6.2.4 Fault Detection

Measurements from multiple sensors allow the implementation of a fault detection and diagnosis strategy. Redundant information can be used by an algorithm to achieve safe and graceful degradation of performance when there are temporary or permanent errors in some of the vehicle sensors.

Bibliography

[1] D. W. Allan. Statistics of atomic frequency standards. In *Proceedings of the IEEE*, volume 52, pages 221–230, February 1966.

[2] Brian D. O. Anderson. *Optimal Filtering.* Prentice-Hall, 1979.

[3] Immanuel A.R. Ashokaraj, Peter M.G. Silson, Antonios Tsourdos, and Brian A. White. Implementation of an adaptive EKF to multiple low cost navigation sensors in wheeled mobile robots. pages 608 – 612, Singapore, Singapore, 2002.

[4] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221 – 255, 2004.

[5] Gui-Qiu Bao, Shen-Shu Xiong, and Zhao-Ying Zhou. Vision-based horizon extraction for micro air vehicle flight control. *IEEE Transactions on Instrumentation and Measurement*, 54(3):1067–1072, June 2005.

[6] David M. Bevly. Global Positioning System (GPS): A low-cost velocity sensor for correcting inertial sensor errors on ground vehicles. *Journal of Dynamic Systems, Measurement and Control*, pages 255–264, 2004.

[7] Hongwei Bian, Zhihua Jin, and Weifeng Tian. Study on GPS attitude determination system aided INS using adaptive kalman filter. *Measurement Science and Technology*, 16(10):2072 – 2079, 2005.

[8] O. Bisi, S. U. Campisano, L. Pavesi, and F. Priolo. *Silicon-based Microphotonics.* IOS Press, 1998.

[9] K. R. Britting. *Inertial Navigation System Analysis.* John Wiley and Sons, Inc., 1971.

[10] A. Brown and Y. Lu. Peformance test results of an integrated GPS/MEMS inertial navigation package. In *Proceedings of Institute of Navigation, GPS*, pages 825–832, Long Beach, CA, 2004.

[11] Alison K. Brown. GPS/INS uses low-cost MEMS IMU. In *IEEE Aerospace and Electronic Systems Magazine*, volume 20, pages 3 – 10, 2005.

[12] J. O. Winkler C. Kreye, B. Eissfeller. Improvements of GNSS receiver performance using deeply coupled ins measurements. In *Proceedings of ION GPS*, pages 844 – 854, 2000.

[13] Averil B. Chatfield. *Fundamentals of High Accuracy Inertial Guidance*. American Institute of Aeronautics and Astronautics, Inc.,, 1997.

[14] H. Chou, M. Traonmilin, E. Ollivier, and M. Parent. A simultaneous localization and mapping algorithm based on kalman filtering. In *IEEE Intelligent Vehicles Symposium*, pages 631–635, 2004.

[15] S.G. Chroust and M. Vincze. Fusion of vision and inertial data for motion and structure estimation. *Journal of Robotics Systems*, 21(2):73–83, 2004.

[16] Joshua M. Clanton, David M. Bevly, and A. Scottedward Hodel. Highway lane tracking using GPS in conjunction with onboard IMU and vision-based lane tracking measurements. In *Proceedings of the Institute of Navigation - 19th International Technical Meeting of the Satellite Division, ION GNSS*, volume 2, pages 1076 – 1084, Fairfax, VA 22030, United States, 2006.

[17] Adam J. Dean. *Terrain Based Road Vehicle Localization Using Attitude Measurements*. PhD thesis, Pennsylvania State University, University Park, Pennsylvania, December 2008.

[18] Daniel L. Edwards, Gregory B. Desmond, and Michael W. Schoppmann. Terrain data base generation for autonomous land vehicle navigation. *Photogrammetria*, 43(2):101–107, 1988.

[19] M. D. Ernst and B. E. Flinchbaugh. Image/map correspondence using curve matching. Technical Report CSC-SIUL-89-12, Texas Instruments, 1989.

[20] Seamless data distribution system, earth resources observation and science (EROS), 2006. http://seamless.usgs.gov/website/seamless/.

[21] Ming Fan, Min Tang, and Jinxiang Dong. A review of real-time terrain rendering techniques. In *Proceedings of Computer Supported Cooperative Work in Design (CSCWD)*, volume 1, pages 685–691, Xiamen, China, May 2004.

[22] N.M. Faulkner, S.J. Cooper, and P.A. Jeary. Integrated MEMS/GPS navigation systems. In *IEEE PLANS, Position Location and Navigation Symposium*, pages 306 – 313, Palm Springs, CA, 2002.

[23] Yan Fei, Wang Wei, and Wang Sheng. Improved adaptive particle filters for mobile robot indoor localization using laser range finder. In *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, volume 2, pages 9188 – 9192, Piscataway, NJ 08855-1331, United States, 2006.

[24] T. Ford, J. Hamilton, M. Bobye, and L. Day. GPS/MEMS inertial integration methodology and results. In *Proceedings of Institute of Navigation, GNSS*, pages 1587–1597, Long Beach, CA, 2004.

[25] T. Ford, J. Neumann, and M. Bobye. OEM4 inertial: An Inertial/GPS navigation system on the OEM4 receiver. In *Proceedings of the International Symposium of Kinematic System in Geodesy, Geomatics and Navigation (KIS)*, Banff, Alberta, 2001.

[26] Tom Ford, Jason Hamilton, Mike Bobye, and Laurence Day. GPS/MEMS inertial integration methodology and results. In *Proceedings of the 17th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2004*, pages 1587 – 1597, Long Beach, CA, United States, 2004.

[27] Gene F. Franklin, J. David Powell, and Michael Workman. *Digital Control of Dynamic Systems*. Addison Wesley Longman Inc., Menlo Park, CA, third edition edition, 1998.

[28] H. Freeman and S. P. Morse. On searching a contour map for a given terrain elevation profile. *Journal of Franklin Institute*, 284:1–25, 1967.

[29] Udo Freze. A discussion on simultaneous localization and mapping. *Autonomous Robots*, pages 25–42, 2006.

[30] C. Fruh and A. Zakhor. Fast 3D model generation in urban environments. *Multisensor Fusion and Integration for Intelligent Systems*, pages 165–170, 2001.

[31] Simon Furst and Ernst-Dieter Dickmanns. A vision based navigation system for autonomous aircraft. *Robotics and Autonomous Systems*, pages 173–184, 1999.

[32] D. Gebre-Egziabher, R.C.Hayward, and J.D. Powell. Design of multi-sensor attitude determination systems. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2):627–649, 2004.

[33] A. Gelb, editor. *Applied Optimal Estimation*. M.I.T. Press, Cambridge, Mass., 1974.

[34] Michael George and Salah Sukkarieh. Camera aided inertial navigation in poor gps environments. In *IEEE Aerospace Conference Proceedings*, pages 4161345 –, Piscataway, NJ 08855-1331, United States, 2007.

[35] T. D. Gillespie. *Fundamentals of vehicle dynamics*. Society of Automotive Engineers, 1992.

[36] S. Godha and Cannon M. E. Integration of DGPS with a low cost MEMS - based inertial measurement unit (IMU) for land vehicle navigation application. In *Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS*, pages 333–345, Long Beach, California, September 2005.

[37] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings, Part F: Radar and Signal Processing*, volume 140, pages 107 – 113, 1993.

[38] D. Goshen-Meskin and I. Y. Bar-Itzhack. Observability analysis of piece-wise constant systems with application to inertial navigation. volume 2, pages 821 – 826, Honolulu, HI, USA, 1990.

[39] Mohinder S. Grewal. *Global positioning systems, inertial navigation, and integration.* John Wiley,, 2001.

[40] Brad Grinstead, Andreas Koschan, David Page, Andrei Gribok, and Mongi A. Abidi. Vehicle-borne scanning for detailed 3D terrain model generation. In *SAE Commercial Vehicle Engineering Congress*, Chicago, IL, November 2005.

[41] J.E. Guivant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, (3):242–257, 2001.

[42] Roger C. Hayward, Demoz Gebre-Egziabher, Matt Schwall, J. David Powell, and John Wilson. Inertially aided GPS based attitude heading reference system (AHRS) for general aviation aircraft. volume 1, pages 289 – 298, Kansas City, MO, USA, 1997.

[43] Janne Heikkila and Olli Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, San Juan, PR, USA, 1997.

[44] Christopher Hide, Terry Moore, and Martin Smith. Adaptive kalman filtering algorithms for integrating GPS and low cost INS. In *IEEE PLANS, Position Location and Navigation Symposium*, pages 227 – 233, Monterey, CA, United States, 2004.

[45] Brian D. Hoffman. Improved rover state estimation in challenging terrain. *Autonomous Robots*, 6:113–130, 1999.

[46] Sinpyo Hong, Man Hyung Lee, Ho Hwang Chun, Sun-Hong Kwon, and J. L. Speyer. Observability of error states in GPS/INS integration. *IEEE Transactions on Vehicular Technology*, 54(2):731–743, 2005.

[47] B.K.P. Horn. Relative orientation. *International Journal of computer Vision*, 4(1):59–78, 1990.

[48] H. Hou, A. El-Sheimy, Noureldin, and N. El-Sheimy. Inertial sensors errors modeling using allan variance. In *Proceedings of ION GPS/GNSS 2003*, pages 259 – 268, Portland, OR, United States, 2004.

[49] Zhencheng Hu and Keiichi Uchimura. Real-time data fusion on stabilizing camera pose estimation output for vision-based road navigation. In *Proceedings Stereoscopic Displays and Virtual Reality Systems*, volume 5291, pages 480–490, May 2004.

[50] Greg Hyslop, Dennis Gerth, and John Kraemer. GPS/INS integration on the standoff land attack missile (SLAM). *IEEE Aerospace and Electronic Systems Magazine*, 5(7):29 – 34, 1990.

[51] Kiichiro Ishikawa, Jun-Ichi Takiguchi, Yoshiharu Amano, and Takumi Hashizume. A mobile mapping system for road data capture based on 3d road model. In *Proceedings of the IEEE International Conference on Control Applications*, pages 638 – 643, New York, NY 10016-5997, United States, 2007.

[52] Aleksandar Jovancevic, Andrew Brown, Suman Ganguly, Joseph Noronha, and Brijesh Sirpatil. Ultra tight coupling implementation using real time software receiver. pages 1575 – 1586, Long Beach, CA, United States, 2004.

[53] Simon J. Julier, Jeffrey K. Uhlmann, and Hugh F. Durrant-Whyte. New approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, volume 3, pages 1628 – 1632, Seattle, WA, USA, 1995.

[54] Jeong Won Kim, Dong-Hwan Hwang, and Sang Jeong Lee. A deeply coupled GPS/INS integrated kalman filter design using a linearized correlator output. In *IEEE PLANS, Position Location and Navigation Symposium*, volume 2006, pages 300 – 305, San Diego, CA, United States, 2006.

[55] K.R.S. Kodagoda, W.S. Wijesoma, and A.P. Balasuriya. Road feature extraction using a 2D LMS. In *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision, ICARCV 2002*, pages 453 – 458, Singapore, Singapore, 2002.

[56] Damien Kubrak, Christophe Macabiau, Michel Monnerat, and Marie-Laure Bouchert. Vehicular navigation using a tight integration of aided-gps and low-cost MEMS sensors. In *Proceedings of the Institute of Navigation, National Technical Meeting*, volume 1, pages 149 – 158, Monterey, CA, United States, 2006.

[57] Jean Laneurit, Roland Chapuis, and Frederic Chausse. Accurate vehicle positioning on a numerical map. *International Journal of Control, Automation and Systems*, 3(1):15 – 31, 2005.

[58] Jack Langelaan and Steve Rock. Navigation of small UAVs operating in forests. In *AIAA Guidance, Navigation, and Control Conference*, volume 3, pages 2031 – 2041, Providence, RI, United States, 2004.

[59] Jack Langelaan and Steve Rock. Passive gps-free navigation for small UAVs. In *IEEE Aerospace Conference Proceedings*, volume 2005, pages 1559602 –, Big Sky, MT, United States, 2005.

[60] Jack Langelaan and Steve Rock. Towards autonomous UAV flight in forests. In *AIAA Guidance, Navigation, and Control Conference*, volume 1, pages 636 – 648, San Francisco, CA, United States, 2005.

[61] A. Lawrence. *Modern Inertial Technology*. Springer Verlag, 1993.

[62] R. Lerner, E. Rivlin, and P.H. Rotstein. Pose estimation using feature correspondences and DTM. In *International Conference on Image Processing*, pages 2603–2606, Singapore, October 2004.

[63] David G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441 – 450, 1991.

[64] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91 – 110, 2004.

[65] Gang Lu, Gerard Lachapelle, M. Elizabeth Cannon, and Peter Kielland. Attitude determination in a survey launch using multi-antenna GPS technology. pages 251 – 260, San Francisco, CA, USA, 1993.

[66] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of 7th Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[67] L. Lucido, J. Opderbecke, V. Rigaud, R. Deriche, and Z. Zhang. A terrain referenced underwater positioning using sonar bathymetric profiles and multiscale analysis. In *Oceans Conference Record (IEEE)*, volume 1, pages 417–421, Fort Lauderdale, FL, USA, September 1996.

[68] R.C. Luo, Jian-Xian Li, and G.-T. Chen. Indoor localization using line based map for autonomous mobile robot. In *IEEE International Conference on Advanced Robotics and its Social Impacts, ARSO 2008*, pages 4653586 –, Piscataway, NJ 08855-1331, United States, 2008.

[69] Ren C. Luo and Michael G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):901 – 931, 1989.

[70] Ren C. Luo and Michael G. Kay. A tutorial on multisensor integration and fusion. In *IECON Proceedings (Industrial Electronics Conference)*, volume 1, pages 707 – 722, Pacific Grove, CA, USA, 1990.

[71] Raj Madhavan. Terrain aided localization of autonomous vehicle. *Automation in Construction*, pages 83–100, 2004.

[72] J. Manyika and H. Durrant-Whyte. *Data fusion and Sensor Management : A Decentralized Information-Theoretic Approach*. Ellis Horwood, 1994.

[73] Richard L. Marks. *Experiments in Visual Sensing for Automatic Control of an Underwater Robot*. PhD thesis, Stanford University, Stanfor, CA, USA, June 1995.

[74] Navin G. Mathur. *Feasibility of Using a Low-Cost Inertial Measurement Unit with Centimeter Accuracy Differential Global Positioning System*. PhD thesis, Ohio University, Athens, Ohio, USA, November 1999.

[75] S. McLoughlin, C. Deegan, C. Mulvihill, C. Fitzgerald, and C. Markham. Mobile mapping for the automated analysis of road signage and delineation. *IET Intelligent Transport Systems*, 2(1):61 – 73, 2008.

[76] Isaac Miller and Mark Campbell. Particle filtering for map-aided localization in sparse gps environments. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1834 – 1841, Piscataway, NJ 08855-1331, United States, 2008.

[77] Pratap Misra and Per Enge. *Global Positioning System: Signals, Measurements and Performace*. Ganga-Jamuna Press, 2006.

[78] Sameh Nassar. Accurate INS/DGPS positioning using INS data de-noising and autoregressive (AR) modeling of inertial sensor errors. *Geomatica*, 59(3):283 – 294, 2005.

[79] Sameh Nassar, Klaus-Peter Schwarz, Naser El-Sheimy, and Aboelmagd Noureldin. Modeling inertial sensor errors using autoregressive (AR) models. *Navigation, Journal of the Institute of Navigation*, 51(4):259 – 268, 2004.

[80] Lawrence C. Ng. Characterization of ring laser gyro performance using the allan variance method. *Journal of Guidance, Control, and Dynamics*, 20(1):211–214, 1997.

[81] Novatel Inc., Calgary, Alberta, Canada. *$SPAN^{TM}$ Technology System: User Manual*, rev. 7 edition, December 2005.

[82] Ernest J. Ohlmeyer. Analysis of an ultra-tightly coupled GPS/INS system in jamming. volume 2006, pages 44 – 53, San Diego, CA, United States, 2006.

[83] Oliver Pink. Visual map matching and localization using a global feature map. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, pages 4563135 –, Piscataway, NJ 08855-1331, United States, 2008.

[84] J.R. Pittman. *Inertial Guidance*. John Wiley and Sons, 1962.

[85] William H. Press. *Numerical recipes in C : the art of scientific computing*. Cambridge University Press, 1997.

[86] H. Rehbinder and B.K. Ghosh. Rigid body state estimation using dynamic vision and inertial sensors. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 3, pages 2398–2403, Orlando, Florida, December 2001.

[87] Ihnsoek Rhee, Mamoun F. Abdel-Hafez, and Jason L. Speyer. Observability of an integrated GPS/INS during maneuvers. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2):526 – 535, 2004.

[88] Kristof Richmond and Stephen M. Rock. An operational real-time large-scale visual mosaicking and navigation system. pages 4099052 –, Piscataway, NJ 08855-1331, United States, 2006.

[89] Jeffrey J. Rodriguez and J. K. Aggarwal. Matching aerial images to 3-D terrain maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12), 1990.

[90] Robert M. Rogers. *Applied Mathematics in Integrated Navigation Systems.* American Institute of Aeronautics and Astronautics, 2000.

[91] C. Ryu and J. C. Gerdes. Vehicle sideslip and roll parameter estimation using GPS. In *6th Int. Symposium on Advanced Vehicle Control, AVEC*, Hiroshima, Japan, 2002.

[92] 1978. Vehicle Dynamics Terminology.

[93] O.S. Salychev, V.V. Voronov, M.E. Cannon, R. Nayak, and G. Lachapelle. Low cost INS/GPS integration: Concepts and testing. In *Proceedings of Institute of Navigation National Technical Meeting*, pages 98–105, Anheim, CA, 2000.

[94] Jacob T. Schwartz. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *International Journal of Robotics Research*, 6(2):29–44, 1987.

[95] Sidney G. Smith. Ring laser gyros. *Physics in Technology*, 18(4):165 – 169, 1987.

[96] W. Sohne, O. Heinze, and E. Groten. Integrated INS/GPS system for high precision navigation applications. In *IEEE PLANS, Position Location and Navigation Symposium*, pages 310 – 313, Las Vegas, NV, USA, 1994.

[97] J. Sparbert, K. Dietmayer, and D. Streller. Lane detection and street type classification using laser range images. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 454 – 459, 2001.

[98] Shuttle radar topography mission, 2006. http://srtm.usgs.gov/.

[99] O. Strauss, F. Comby, and M.J. Aldon. Multibeam sonar image matching for terrain-based underwater navigation. In *Oceans Conference Record (IEEE)*, volume 2, pages 882–887, Seattle, WA, 1999.

[100] Dennis Strelow and Sanjiv Singh. Motion estimation from image and inertial measurements. *The International Journal of Robotics Research*, 23(12):1157–1195, 2004.

[101] Salah Sukkarieh, Eduardo M. Nebot, and Hugh F. Durrant-Whyte. High integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*, 15(3):572 – 578, 1999.

[102] M. V. Blundell T. A. Wenzel, K. J. Burnham and R. A. Williams. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[103] M. V. Blundell T. A. Wenzel, K. J. Burnham and R. A. Williams. Dual extended kalman filter for vehicle state and parameter estimation. *Vehicle System Dynamics*, 44(2):153–171, 2006.

[104] P Fenton M Bobye J Hamilton T Ford, J Neumann. OEM4 inertial: A tightly integrated decentralised Inertial/GPS navigation system.

[105] Raj Talluri and J. K. Aggarwal. Position estimation for an autonomous mobile robot in an outdoor environment. *IEEE Transactions on Robotics and Automation*, 8(5):573–584, 1992.

[106] Yew Keong Tham, Han Wang, and Eam Khwang Teoh. Multi-sensor fusion for steerable four-wheeled industrial vehicles. *Control Engineering Practice*, 7(10):1233 – 1248, 1999.

[107] Weston J.L. Titterton, D.H. and. *Strapdown inertial navigation technology*. Reston, VA : American Institute of Aeronautics and Astronautics, 2004.

[108] William Travis, Adam T. Simmons, and David M. Bevly. Corridor navigation with a LiDAR/INS kalman filter solution. In *IEEE Intelligent Vehicles Symposium, Proceedings*, volume 2005, pages 343 – 348, Piscataway, NJ 08855-1331, United States, 2005.

[109] Chaochao Wang and Gerard Lachapelle. Development of a low-cost solution for gps/gyro attitude determination. In *Proceedings of the National Technical Meeting, Institute of Navigation*, volume 2004, pages 267 – 276, Alexandria, VA 22314, United States, 2004.

[110] Rong-Ben Wang, Bai-Yuan Gu, Li-Sheng Jin, Tian-Hong Yu, and Lie Gou. Study on curb detection method based on 3D range image by laser radar. In *IEEE Intelligent Vehicles Symposium, Proceedings*, volume 2005, pages 845 – 848, Piscataway, NJ 08855-1331, United States, 2005.

[111] Kajiro Watanabe, Kazuyuki Kobayashi, and Fumio Munekata. Multiple sensor fusion for navigation systems. pages 575 – 578, Yokohama, Japan, 1994.

[112] M. Wei and K.P. Schwarz. Testing a decentralized filter for GPS/INS integration. *IEEE PLANS, Position Location and Navigation Symposium*, pages 429 – 435, 1990.

[113] Haim J. Wolfson. On curve matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):483–489, May 1990.

[114] Ji Hwan Woo, Gwan Sung Kim, In So Kweon, and In Cheol Kim. Robust horizon and peak extraction for vision-based navigation. In *IAPR Conference on Machine Vision Applications*, Tsukuba Science City, Japan, May 2005.

[115] S. You and U. Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In *Proceedings of the IEEE Virtual Reality Conference*, pages 71–78, Yokohama, Japan, 2001.

[116] Feizhou Zhang, Xiuwan Chen, Min Sun, Ming Yan, and Dongkai Yang. Simulation design of underwater terrain matching navigation based on information fusion. In *Geoscience and Remote Sensing Symposium, IGARSS*, volume 5, pages 3114–3117, September 2004.

Vita

Vishisht Gupta

Vishisht Gupta was born and brought up in a small town of Punjab state in India. He received his Bachelor's and Master's degree in Mechanical Engineering from Indian Institute of Technology Bombay (IIT Bombay), at Mumbai, India. He went on to join Pennsylvania State University in Fall 2003 to pursue a Ph.D. in Mechanical Engineering. He is currently working with Commercial Steering Systems (CSS) group of TRW Automotive in Lafayette, Indiana.