Online Rotational Self-Calibration of LiDAR Sensors when Mounted on a Ground Vehicle

by

Stephanie W. Meyer

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 7, 2021

Keywords: LiDAR, calibration, autonomous vehicle, robot, sensors, attitude

Approved by

David Bevly Chair, Professor of Mechanical Engineering
Scott Martin Co-chair, Assistant Research Professor of Mechanical Engineering
Thaddeus Roppel, Associate Professor of Electrical and Computer Engineering
Howard Chen, Assistant Research Professor of Mechanical Engineering

Abstract

This thesis presents a method for online extrinsic rotational calibration of a LiDAR sensor with minimal manual intervention. The approach leverages the expected geometry of structured environments common to ground vehicle applications to correct for sensor-to-vehicle rotational misalignment without the need of calibration targets, additional sensors, or an accurate model of vehicle kinematics or dynamics. Once an extrinsic calibration transform is estimated by this approach, it can be applied to the raw LiDAR data to transform it into a frame with known relation to other calibrated sensors and points of interest on the vehicle body, such as a control point or vehicle frame origin. This approach may be used in applications that involve the vehicle following the trajectory of a distinct, straight section of a road or pathway, and where the sensor collection initiated with the vehicle at rest on an extended level surface. To estimate the yaw offset, the road trajectory is detected, and for the roll and pitch offsets, the orientation of the ground plane in front of the vehicle is estimated. This thesis also proposes a novel road edge and lane line marking detection algorithm capable of detecting the edges and markings at arbitrary orientations and locations as required for estimating the LiDAR yaw offset, as well as details an approach for estimating the ground orientations. The approach estimates roll and pitch offsets up to 90 degrees, and yaw offsets of up to 45 degrees with respect to a level LiDAR with x-axis aligned with the forward x-axis of the vehicle it is mounted to. In testing, the estimated calibration was able to correct for dynamic roll and pitch estimations of the ground plane to a root mean square error of within 1.20 and 2.11 degrees in roll and pitch respectively over all tested scenarios, and was within 0.099 and 0.119 degrees in roll and pitch when the ground orientation remained constant with respect to gravity. The road edge orientation detection was able to detect road lines in 80% of tested frames with a root mean square error of 0.47 degrees in detected line orientation.

Acknowledgments

I have many individuals and groups of people to thank for their support during the development of this thesis and throughout the course of my Master’s Degree at Auburn. Firstly are my advisors Dr. David Bevly, Dr. Scott Martin, and Dr. Howard Chen, for allowing me the opportunity to study and research in the GPS and Vehicle Dynamics Laboratory, and for guiding, supporting, and challenging me along the way, and similarly my committee member Dr. Thaddeus Roppel for support, kindness, and robotics inspiration. I am particularly grateful to Howard for his great friendship and unfaltering dedication to paper reviews, methods and analysis advice, presentation practices, and so much more, not only for myself but for so many other members of the GAVLab alongside me. I certainly owe what accomplishments I have made to the work of the giants who came before me, from big, historic names to GAVLabers who studied, developed, coded, and wrote before me. I am lucky for all the enriching friendships I have made along the way, from new and old GAVLabers to co-workers on projects and internships, and for old friends who support me through the distance and through too-long gaps of study- and work-silence. For Giuseppe for encouragement and enthusiasm in what I do, for love and friendship, and for allowing me time and space to do what needs to be done. Of course my family are due heaps of gratitude as well, for getting me here to begin with, for being proud of me and for telling me so, for understanding when some trips home were spent with me on the computer more than any of us would prefer, and for making sure I stop sometimes, if only for a little walk around the bottom or for happy hour with Dee.

Table of Contents

Table of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Summary

In any sensor-equipped system, calibration is a step that is necessary to minimize errors and distortions from the sensor data, and to root the sensor measurements into a frame that is meaningful to the system. Specifically, a calibration method is any process that determines the mathematical relationships needed to adjust the data from its given domain or form into a desired domain. This relationship is usually provided in the form of a transformation matrix involving rotations, translations, scaling, and skewing. Examples of sensor-dependent systems that benefit significantly from sensor calibration include both stationary and mobile robots as well as advanced driver assistance systems (ADAS).

Calibration comes in many forms, and can generally be broken into two categories, intrinsic and extrinsic. Intrinsic calibration is the determination of how the data is distorted or offset with respect to the coordinate frame of the sensor body in Euclidean space, typically considered a Cartesian frame, or what static errors it may incur. Sensors such as RADAR and LiDAR are often intrinsically calibrated by the manufacturer of the sensor units prior to shipment. The intrinsic transforms for these sensors are applied within the embedded sensor electronics or within manufacturer-provided sensor driver software, allowing the data accessed by the sensor user to be pre-calibrated intrinsically and already aligned with the sensor’s Cartesian frame and converted to Euclidean geometry.

Extrinsic calibration is the measurement or estimation of the rotational and translational offsets between the sensor Cartesian frame and another, known Cartesian frame. Calibration transforms are most often expected to be static, and as such they are usually defined between two objects on the same rigid body. If extrinsic calibration has not been applied, sensor data is typically capable of providing information only with respect to the sensor itself, and not with respect to the system or body that the sensor is mounted to. Perfect alignment between the sensor and the mounted body is sometimes assumed, though mounting can rarely be achieved with the alignment precision required for most applications. Further, the assumption of perfect alignment does not allow for the sensors to be mounted with an intentional offset, which may allow the sensor to better capture data of interest. As such, extrinsic calibration allows for imperfect mounting and intentional mounting offsets to be accounted for.

While there is a wide range of applications that benefit from intrinsic and extrinsic calibration, and within these applications there are many different types of sensors that need calibrating, this thesis limits its scope to a detailed focus on extrinsic rotational calibration of three-dimensional LiDAR sensors on automated and semi-automated ground vehicles. Though developed for this purpose, the application of this approach may be extended to other ground vehicles so long as they meet the required operational qualities.

LiDAR is a perception sensor commonly included on smaller ground robot systems, mobile mapping vehicle systems, and on research automated vehicles. Though not as common in commercial automated vehicles due to the restrictively high prices of LiDAR units, the sensors are beginning to see commercial release on these platforms as well [1]. LiDAR provides relatively high-accuracy (commonly around +/ 3 cm) ranging and geometric samples of surfaces surrounding the sensor. In robotic and vehicle applications, LiDAR-based algorithms such as map-based localization, object detection and tracking, and road and lane detection are often conducted using either two-dimensional or three-dimensional LiDAR, with the results being robust and information-rich if three-dimensional LiDAR data is used.

Extrinsic calibration of LiDAR sensors on ground vehicles commonly requires a special calibration period, potentially involving calibration targets, trained personnel to execute the test, and often cannot be conducted on just any given set of data that did not undergo calibration prior to recording. This thesis proposes an approach where a ground vehicle system may automatically determine the extrinsic calibration of any LiDAR sensor rigidly mounted to its body given only the data from the LiDAR itself.

1.2 Calibration Background

Extrinsic calibration of a LiDAR sensor is a common practice that is undergoing continuing research for improving the usability, resource requirements, and accuracy of the available approaches. As such, there are many existing approaches for acquiring an extrinsic calibration transform for a LiDAR. These approaches differ in the requisite level of human intervention, resources, and restrictive scenarios.

1.2.1 Direct Measurement Calibration

The least technical calibration approach is to directly measure the differences between the sensor and the reference point on the vehicle or system. This may be done with a variety of instruments, including the rudimentary tape measure and the much more reliable laser-surveying machine commonly known as a total station. When using a total station, the relative range, azimuthal angle, and vertical angle to a series of manually-selected points on the sensors and vehicle may be measured, and the translational and rotational calibration offsets directly calculated from these 3D points. It is generally accepted that the lever arms of a calibration transform may be acquired from direct measurements with adequate accuracy. The rotational offsets, however, prove somewhat more difficult to precisely measure [2].

1.2.2 Externally Comparative Calibration Methods

Alternatively, many LiDAR calibration approaches involve algorithmic comparison between the LiDAR data and data from another sensor or source to determine an optimal alignment transform between the two data sets. The data that is compared may be raw point clouds, as when comparing a LiDAR to another LiDAR or other point cloud data, or may be information which has been extracted from the LiDAR data using various processing approaches. The type of extracted data used in comparative calibration approaches can be generalized into two categories: object or feature information and pose or motion information.

Object or feature position estimates can be used when calibrating the LiDAR to a camera, RADAR, or other perception sensor. This requires that the same features or objects be visible and detectable in both sensors. The objects or features may be naturally present in the calibration environment, such as corner features or distinct objects such as traffic cones, poles, and road signs, or may be designated targets placed in the environment for calibration [3-4]. First, the objects or features must be detected in both data sets. Next, associations must be determined between object pairs across the data sources (ie. a specific object detected from the LiDAR data must be matched with the same object as detected from the other data set). Once same objects are paired across the sensors, the optimal transform which best aligns all object poses from the LiDAR to the corresponding poses in the other data set is the estimated calibration transform between the two units. The resultant calibration is subject to compounded errors from the object position detection algorithm, the association, and the optimal alignment.

Pose or motion estimates may be used to calibrate a LiDAR to a GNSS sensor, inertial measurement unit (IMU), to a control point using wheel or steering encoders through a motion model of the vehicle, or to camera or other LiDAR using visual or point cloud odometry approaches [5-6]. These approaches require the estimation of the vehicle pose from the LiDAR and the corresponding pose source and have their limitations as well. The calibration of the LiDAR is dependent on the accuracy and availability of the comparison pose estimation (the motion estimates coming from another sensor or source, such as an IMU) and its associated calibration. Additionally, LiDAR-based pose estimation approaches are highly dependent on the dominance of environmental geometry over sampling geometry in the LiDAR data, uniquely-identifiable features, and the lack of independently dynamic objects within the field of view of the LiDAR. Feature uniqueness and lack of dynamic objects are qualities often not met on the standard roadway, where pedestrians and other vehicles are often present as dynamic objects, and one scan of the roadway often appears nearly identical to the next scan. Certain approaches to LiDAR odometry enjoy more success in roadway conditions with relatively sparse sampling and uniform sampling geometries than others, most notably point-to-plane approaches. These methods develop upon the standard scan matching approach, iterative closest point (ICP), by pairing points in one point cloud with the closest plane in the other point cloud, rather than associating closest points between the two clouds [7]. The relative success of point-to-plane ICP is shown in [8], though there is still a strict requirement for the scenes to be non-uniform as the vehicle traverses the environment. These limitations in the success of pose and motion estimation from LiDAR data communicate into limitations in the success of calibration estimation from pose- and motion-comparison.

1.2.3 Internally (Geometric) Comparative Calibration Methods

The externally comparative calibration approaches require that another data set in addition to the data from the LiDAR being calibrated be available. The calibration directly solved for from externally comparative methods is also a relative transform between the two data sources, rarely directly calibrating the LiDAR frame to a vehicle frame of interest. An approach that removes the requirement of additional data sets is to use the geometry of the scene within the LiDAR data itself. One category of geometry-based calibration approach involves the use of specific target objects. These targets are visible, unique, and distinct from the environment within the LiDAR data, with known geometric extent. Examples of targets include the cylindrical target used to calibrate a manipulator end to a laser scanner in [9], or a set of circular, 2D reflective targets placed on a wall in [2]. While these target-based approaches tend to yield high-accuracy lever arm and orientation calibration, the tools, objects, or scenarios these methods require may not always be available, and further require trained personnel capable of accurately executing the calibration procedure or target setup.

A second geometry-based calibration approach avoids the need for target objects by leveraging assumptions about the content of the environment visible to the LiDAR, and the expected behavior of the vehicle body with respect to the environment. For example, Britt et al. [10] used the visible geometry within the LiDAR's field of view during common operational scenarios. In that work, the authors were able to achieve around 0.8 degree pitch and 0.31 degree roll accuracy with this approach, but the processes involved hundreds of iterations of trigonometric math executions per frame that was processed, resulting in a solution with high computational demands and low solution time.

1.2.4 Considerations for the Proposed Approach

To avoid the limitations of the previous calibration approaches, and to operate with a self-imposed goal of using only the data from the LiDAR being calibrated to account for instances when corroborative data is not available for calibration assistance, the approach proposed in this thesis builds upon the approaches outlined by Britt et al and utilizes a targetless geometry-based calibration approach. This thesis attempts to expand the calibration estimate provided in [10] from a rotation-only calibration to a full 6-degree-of-freedom estimation, and further reduces some of the computations required in [10] to achieve steps such as roll and pitch estimation from a ground plane. Further, the assumption made in [10] is made more robust by applying a yaw-calibration correction to the LiDAR data before ground fitting, as this and other approaches establish static regions of interest for the ground plane points. The static region of interest is more readily ensured to be free of conflicting object points if the relationship of the area to the vehicle is known. For instance, a side-of-vehicle area of interest may need to be significantly narrower than a forward-facing area of interest to avoid capturing point returns from passing traffic or off-path objects. The approach outlined in this thesis allows the yaw calibration to be conducted prior to ground estimation, unlike [10] where roll and pitch must be calibrated first.

1.3 Definition of Coordinate Frames

Given a ground frame g, a measurement frame m, and a body frame b, an extrinsic calibration of a sensor (measurement unit) with frame m to align it on a body with frame b solves for the transformation T_b^m from the body frame to the measurement frame. The global frame is presented here to provide a stationary frame with which to reference the orientations of the measurement and body frames as they move, for conceptual clarity. The orientation of the body in the ground frame is equal to the transformation from the global frame to the body frame T_g^b. Similarly, the orientation of the measurement frame in the global frame is T_g^m. A generic visualization for these three frames is given in Figure 1.1.

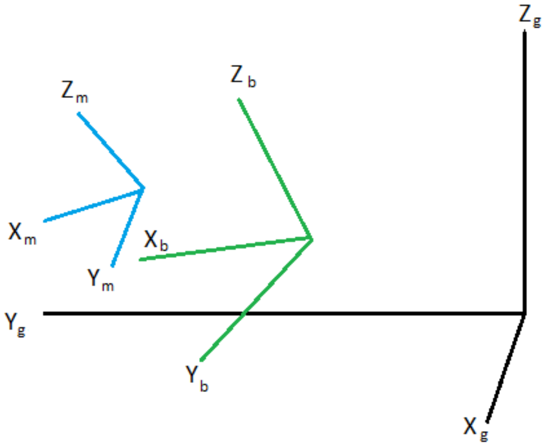


Figure 1.1: Generic Coordinate Frames for a Sensor (m), a Body (b), and a Global Reference (g)

For the purposes of extrinsically calibrating the orientation of a LiDAR to a vehicle that it is mounted to, the sensor frame m is set to be that of the LiDAR, the body frame b is a frame aligned with the body of the vehicle, and the global frame g is defined such that the plane formed by the x- and y-axes is level according to gravity. The reference frame defined for the vehicle may have one of various orientations and points of origin options, depending on the application. Commonly in automated vehicle applications, a control point is defined on the vehicle, often at the center of the rear axle the center of gravity of the vehicle. If the center of the rear axel control point is chosen as the origin for the vehicle body frame, a translational calibration of the LiDAR frame to the vehicle frame would align the LiDAR origin to this control point. The x-axis of the vehicle frame is aligned to the forward direction of the vehicle, and the y- and z-axes can either be positive left and up respectively, or right and down respectively depending on the application.

From a LiDAR perspective, it is unlikely that the height difference between the ground and the rear axle would be visible within the LiDAR data. Because of this, the thesis will focus on a body frame that has origin at the projection of the center of the rear axle onto the ground, and orientation such that the reference frame is level with the ground when the vehicle is at-rest on level ground. To align with an intuitive "up is positive" perspective, a z-up frame is selected. If the at-rest height of the vehicle axes from the frame is known, an estimated or static parameter height can be used to project the ground-projected vehicle body frame back to the height of the axles.

In Figure 1.2, the frames defined for this calibration approach are displayed as "LiDAR Frame" for frame m, "Vehicle Frame" for a height-projected frame b, and "Base Footprint Frame" for the on-ground frame b. Note that it is not required that the LiDAR be mounted on the hood of the vehicle as shown here. It is only necessary that the LiDAR have view of the ground roughly in front of the vehicle, roll and pitch mounting offsets of less than 90 degrees, and a yaw mounting offset of less than 45 degrees.

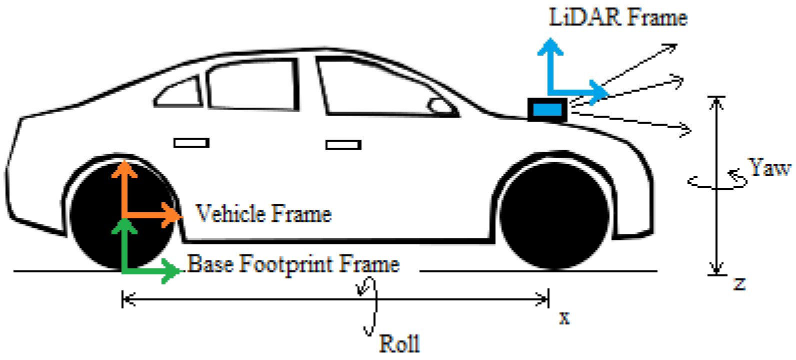


Figure 1.2: Sensor and reference coordinate systems for a standard Ackermann-style sedan passenger vehicle with arbitrarily-mounted LiDAR. LiDAR (blue), vehicle (orange), and base footprint (green) coordinate axes

The use of a reference frame projected to the ground beneath the vehicle warrants further discussion. For a calibration transform to be meaningful, it must represent the current rotational and translational differences between the two frames being calibrated. Typically, a calibration transform is expected to be static, and so is only valid when the two frames are on the same rigid body or otherwise do not experience a change in relative orientation or position over the period of application. The choice of a reference frame on the ground breaks this static-transform requirement, as the suspension elements of the vehicle and motion inherent in the tires result in significant translational and rotational changes between the vehicle body to which the sensor is attached and the ground when the vehicle is traveling through space. This difference arises from turning, linear accelerations and decelerations, shifting of load, changes in the bank and grade angles of the ground, or presence of obstacles on the ground that the vehicle engages with. A static transform can be approximated by defining the calibration as the *at-rest* rotational and translational differences between the sensor frame and the ground, acknowledging that the reference or vehicle frame will not align with the true ground at all points in time.

In certain applications, the instantaneous difference between the sensor frame and the ground frame may be useful, such as when "levelling" LiDAR or camera data to the ground plane. Though not directly covered in this thesis, the calibration estimation approach outlined here may be easily extended to provide continuously updating sensor-to-ground roll and pitch estimates.

1.4 Contributions

This thesis proposes an approach where a ground vehicle system may automatically determine the extrinsic rotational calibration of any LiDAR sensor that is rigidly mounted to its body given only the data from the LiDAR itself. The specific contributions of this thesis include the following:

- A yaw-independent road edge detection algorithm for LiDAR with experimental results

- An efficient plane-fit and ground orientation algorithm for LiDAR with experimental results
- Overview and real-world experimental results for a rotational 3-degree-of-freedom online extrinsic calibration algorithm for LiDAR on a ground vehicle using only the geometry available within the field of view of the LiDAR itself

The proposed calibration approach is conducted in four major steps. The yaw calibration is first estimated through detection of a visible road edge. The roll and pitch calibrations are subsequently determined through LiDAR-to-Ground orientation estimation. During the yaw estimation, the approach assumes that the vehicle is on a roadway with a recognizable edge, either distinguished by height such as when there is a curb or distinct ditch along the edge of the road, or by intensity, such as with reflective paint lines. Further, it assumes that the vehicle is closely following the trajectory of this road edge, as is generally the case during roadway travel. The LiDAR-to-Ground orientation estimate assumes that the ground in front of the vehicle is planar and that the vehicle is level with the ground, or has an effectively static orientation offset to the ground.

The primary data type utilized throughout the approach is the point cloud $\mathrm{P_i}\ (\mathrm{x_j}, \mathrm{y_j}, \mathrm{z_j}, \mathrm{I_j})\ (\mathrm{i} = 0, \dots, m)$, which consists of m three dimensional points $(\mathrm{x_j}, \mathrm{y_j}, \mathrm{z_j})$ with intensity values $\mathrm{I_j}$. This point cloud of sampled environment surfaces is reported by the LiDAR with respect to the measurement unit frame m, also known as the LiDAR frame. An intensity value is an indication of the reflectivity in the LiDAR pulse spectrum of the reflected point surface Intensity values are reported by most three-dimensional LiDAR sensors.

The calibration estimation is carried out on a frame-by-frame basis over n frames, where $\mathrm{i} = 0, \ldots, n$. It is not necessary for the frames to be consecutive. Individual calibration transform results for each frame are combined to produce a more representative and smoothed calibration estimate either through simple averaging or through a probabilistic filter.

1.5 Thesis Outline

The remainder of this thesis will first review supporting knowledge of LiDAR sensors, LiDAR measurement processing, calibration transforms, and various calibration approaches in Chapter 2. A detailed description of the different algorithms involved in this automatic online LiDAR calibration is given in the following chapters, starting with yaw calibration from road trajectory detection in Chapter 3 and followed by roll and pitch calibration from ground plane orientation detection in Chapter 4. Several experiments are summarized in which the proposed calibration method was utilized on two ground vehicles in Chapter 5, and the thesis concludes in Chapter 6 with a discussion on the results of the study and potential future work.

Chapter 2

Preliminary/Supporting Theoretical Background

This chapter contains various prior work and supporting theories that aid in the understanding of the rest of the thesis. First is a discussion on LiDAR as a sensor for robotic and automotive applications, including data characteristics, operational concepts, LiDAR types, and strengths and weaknesses of the sensor. The theory and techniques involved with the processing of LiDAR point cloud data is discussed in the Point Cloud Processing Background section, with a comparison of point cloud and image processing approaches. Transforms and their different mathematical representations and properties are given in the Transforms Background. An introduction to methods for solving the least squares problem is given next, followed by an overview of how to find the root mean square error, which is used repeatedly throughout the remainder of the thesis.

2.1 LiDAR Background

LiDAR is a laser-ranging sensor that utilizes time-of-flight principals with laser pulse emissions to determine the distance between the sensor and surfaces or objects in the environment. In essence, the sensor measures the amount of elapsed time from the emission of a pulse to the return of the same pulse to the laser receivers on the sensor. This time will vary based on the distance the laser pulse was able to travel before encountering an infrared-reflective surface and being reflected back to the sensor. Therefore, the elapsed time can be used to determine that distance. More sophisticated distancing techniques may utilize modulation and phase shifts of the laser returns to determine distance, rather than relying on a count of elapsed time.

These laser ranging sensors provide relatively high range accuracy (usually centimeter-level) and have visible range up to hundreds of meters for automotive grade sensors. For any mobile platform, LiDAR can be used for object detection, mapping, localization, and semantic scene understanding. For automotive uses, this often includes road space and road marking detection and identification, detection of other road users and their behaviors and motion, and detection of objects such as road signage.

LiDAR come in many different forms, differing in the dimensionality of the outputted measurements and the geometry of the spatio-sampling patterns. LiDARs can be found in one, two, and three dimensions. A one-dimensional (1D) LiDAR emits a single laser pulse at a time and involves no internal moving parts. Unless externally moved, the 1D LiDAR will continue to measure the range to the same point in space over time – the point that is linearly in front of the laser emission point.

Two-dimensional (2D) LiDARs commonly utilize a single laser emission-receiver unit (or channel) similar to the 1D LiDAR, but involve internal moving parts. An example 2D LiDAR is a single-channel mechanically scanning LiDAR, which rotates its single laser emission-receiver unit in a single direction about the sensor’s up- or z-axis as laser pulses are emitted and received. Each laser pulse is marked with an azimuthal angle representing the location in the rotation that the measurement was acquired from, resulting in a set of range-azimuth pairs. This set of range-azimuth measurements is called a point cloud, and is a two-dimensional sampling of a circular “slice” of the environment. Such rotational LiDARs are capable of a horizontal field-of-view of up to 360 degrees within a single measurement cycle, also known as a single scan.

Three-dimensional (3D) LiDARs commonly utilize both internal motion and multiple laser emission-receiver units, or multiple channels. The laser unit sets commonly come in powers of 2, such as 4, 8, 16, 32, and 64. Even larger numbers of channels are becoming available as new sensor technology is developed [11]. These sets of laser units may be stacked vertically with a set vertical angle offset between each unit, or may be stacked horizontally. Vertical-stacked units are often mechanically rotated such that several 2D slices of the environment are sampled at different vertical angles simultaneously as the units are rotated about the sensor's z-axis (or mirrors are rotated about the vertical stack to the same effect). Similar to their 2D counterparts, these mechanically rotating 3D LiDAR are capable of up to 360 degree horizontal field-of-view within a single scan, with vertical field-of-view determined by the number of laser units and the vertical angle separation between them. Alternatively, the horizontal-stacked units typically each undergo independent motion enacted by micro-electromechanical systems (MEMS), or more commonly, mirror systems around the laser units are independently mechanized by MEMS. This MEMS motion achieves a sometimes user-configurable sampling pattern over the environment, often with raster-path sampling geometry. These types of 3D LiDARs typically provide much higher density point clouds with fewer laser units than the 3D rotational LiDAR, but have reduced horizontal fields-of-view. Figure 2.1 demonstrates the different scan geometries of mechanically scanning and MEMS 3D LiDARs.

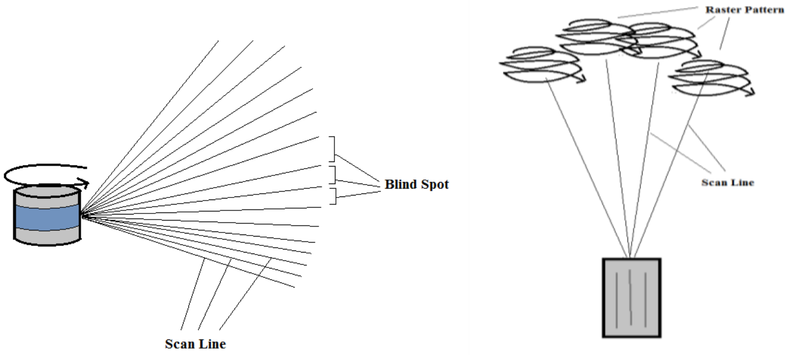


Figure 2.1: Left: Mechanically Rotating 3D LiDAR with 16 Channels, Right: MEMs 3D LiDAR with 4 Channels

Along with the distance to points on surrounding surfaces, LiDAR sensors will often provide an indirect measure of the reflectivity of the reflecting surface, a relative value provided in a 8-bit format termed “intensity.” This is not the true reflectivity of the surface and is generated from the strength of the reflected pulse upon receipt at the sensor [12]. Intensity provides a measure for distinguishing different surfaces or objects within a point cloud similar to color differences in pixels in a camera image, and is useful for such applications as road sign and road marking detection and identification.

LiDARs as sensors in mobile robotics and automated vehicles applications have several strengths over other perception sensor options. LiDAR data is significantly more structured than an automotive RADAR point return, such as that of the Aptiv Electronically Scanning RADAR (ESR) [13], which provides a set number of high-likelihood return points with no set sampling geometry. LiDAR point clouds also typically offer a much higher density sampling of the environment over RADAR sensors, though certain high-density RADAR are beginning to offer densities similar to those of lower-density LiDAR. LiDAR sensors have superior range accuracy and consistency to depth-from-stereo imaging approaches, as the LiDAR sensor directly measures the range rather than inferring it from further processing. As active sensors which do not rely on reflected ambient light to generate measurements, LiDARs are also significantly less affected by lighting conditions than cameras.

There are weaknesses to the LiDAR sensor as well. To begin with, the range accuracy of LiDAR returns is dependent on the strength of the reflected signal. As such, as the travel distance of the laser pulse is increased, the accuracy of the detected range will decrease, though this reduction in accuracy is typically only significant at very large distances. Similarly, as the reflectivity of the encountered object decreases, the strength of the returned signal decreases and the accuracy of the detectable range decreases. Within the operational ranges of a LiDAR, the range error incurred due to decreased reflectivity is typically negligible to most applications. The operable range for the LiDAR is defined by a minimum as well as maximum distance within which the LiDAR is able to reliably measure surfaces above a certain low reflectivity threshold. Outside of the desired sensing range, the signal-to-noise ratio on the point returns are too low to be reliably meaningful, and are often filtered out by the embedded sensor logic rather than sent to the user. The minimum sensible range for 3D automotive LiDAR is typically anywhere between 0.1 and 0.3 meters. Automotive LiDAR often have maximum visible range reaching to anywhere from 100 to 300 meters [14 - 18]. Unlike RADAR data, LiDAR data typically includes no indication of target radial speed. While less subject to occlusion and scattering due to weather and environment effects such as rain, dust, and fog than camera data is, LiDAR are also more vulnerable to these elements than is lower-frequency RADAR data [19].

2.2 Point Cloud Processing Background

As both cameras and 3D LiDAR are perception sensors providing measurements that represent a "snapshot" of the environment around the sensor, there are many parallels in the data structures and processing techniques which can be applied to or seen in their output. 3D LiDAR measurement point clouds have the structure $\mathrm{P_i}\ (\mathrm{x_j}, \mathrm{y_j}, \mathrm{z_j}\ \mathrm{I_j})\ (\mathrm{i} = 0, \dots, m)$, which consists of m three dimensional points $(\mathrm{x_j}, \mathrm{y_j}, \mathrm{z_j})$ with intensity values $\mathrm{I_j}$. This is similar to the output of a camera, which is a structured (or ordered) set of m n-dimensional points where $n = 1$ for monochrome cameras and $n = 3$ for color cameras $(\mathrm{ri}, \mathrm{gi}, \mathrm{bi}$, for example). As opposed to LiDAR point sets, camera point, or pixel, sets are arranged such that the set can quickly be formatted as a two-dimensional matrix of pixels Pxlij. Due to the sampling geometry of LiDAR sensors and the lack of perspective projections in the generation of the data, consecutive points in the set of LiDAR points are often spatially consecutive (or neighboring) in the environment represented by the set, but the mapping from the linear list of points to a geometrically-significant structured matrix is not as straight-forward or uniform as that of the camera point sets.

Many of the processing techniques used for 2D image processing can be applied to LiDAR data as well, given minor re-structuring of the LiDAR point cloud. The three-dimensional version of an image is typically termed a voxel grid, and is a uniform grid in three dimensions, or is a uniform three-dimensional matrix. When sorted into a voxel grid data structure, the LiDAR data will typically yield a large amount of non-occupied grid cells, as only the first-occlusion surfaces are represented by non-zero values. Therefore, an environment with no surfaces would yield an empty voxel grid, and an environment with multiple consecutive vertical surfaces radially from the sensor would result in empty in space behind the first surface, as the LiDAR is incapable of measuring beyond occlusions. Further, unlike camera images, which are presented in dense format where every cell represents a measured value, a 3D voxel grid is not a dense representation of a LiDAR, so any blind spots due to raster pattern for MEMS LiDAR or vertical angle dead-space in mechanically rotating LiDAR will appear as empty cells in the resultant grid. See Figure 2.1 for an example MEMs raster pattern. If a 3D voxel grid is generated from LiDAR data, three dimensional complements of image approaches may be applied, such as a three-dimensional Fourier transform [20].

Alternatively, image processing approaches may be used as designed for camera images if the LiDAR data is transformed into two-dimensional space. This may be done by first selecting a perspective of the LiDAR data (a top-down, or birds-eye-view is a common perspective choice), and then either directly parsing into a scaled 2D grid or transforming into a non-Cartesian representation more similar to that of camera images (such as through a perspective projection mapping). In [21], LiDAR point clouds were converted to camera-style 2D images to allow for the use of typical image feature detection processes.

Birds-eye-view perspective with scaled Cartesian mapping is a straight-forward 3D-to-2D option which requires only one pass through each of the n LiDAR points for each point cloud. The grid cell value, or pixel value, may represent one of several values, with common options including binary occupancy indication, intensity measurement, or height of the tallest point within the cell. This birds-eye-view option compresses or completely discards height information from the LiDAR data, but directly preserves lateral and longitudinal spatial relationships between points, or between occupied cells. This is often a desirable trade-off, as the extent of the vertical field of view may limit the completeness of available height data for discrete entities such as buildings. Further, mobile ground robots (which are typically constrained to remain grounded on pseudo-planar surfaces) are often more affected by the longitudinal and lateral layout of entities in their environments than by the height or vertical layout. For example, in an object detection algorithm which attempts to determine the number and arrangement of other vehicles on the roadway surrounding a passenger vehicle, clustering from a birds-eye-view image from LiDAR data can directly yield the (x, y) coordinates of the edges or centers of the other vehicles in the sensor frame. Birds-eye-view 3D-to-2D mapping is used in [22], where the LiDAR point cloud is parsed into a 2D intensity grid before image processing techniques are applied to detect multiple lane lines across a marked roadway.

There are also some approaches which seek to determine a dense structured format to sort LiDAR data into. This is perhaps most straightforward for mechanically scanning LiDAR data, which can be formatted into a dense spherical grid with relative ease and without losing much spatial resolution. This dense data structure is used in [31], where the LiDAR point cloud from a mechanically scanning LiDAR (Velodyne HDL-64E [23]) is projected into spherical space for a "dense, grid-based representation" before it is inputted into a convolutional neural network for semantic segmentation point labeling. Spherical projection can also be used to transform data from LiDARs with other sampling patterns, though the less grid-like rastering pattern of MEMS LiDAR doesn't lend itself as efficiently or as perfectly to a dense spherical representation.

Other LiDAR processing approaches are designed to work directly on the point cloud as it is delivered from the LiDAR driver software, taking advantage of either the processing savings inherent to skipping a re-ordering step or any advantages that may be inherent to the point cloud as-is. For example, [10] takes advantage of the horizontally uniform scanline geometry of a mechanically rotating LiDAR to aid in lane marking detection. In this approach, a graph of intensity values vs azimuthal angle is formed over a single channel from a 3D point cloud. Traversing the graph bins consecutively, the first peak in intensity values is marked as one edge of the road, and the next peak is searched for at approximately an estimated or hard-coded road with later. The existing cross-sectional ordering of the points within the raw point cloud removes any need to sort prior to searching for intensity peaks within this example. Common categories of point cloud processing include the following:

- Point Cloud Registration (ie scan matching or map matching) – Localization, motion estimation [7, 24]
- Clustering – object detection and bounding [25]
- Feature detection – extended features such as lane markings, or simple features such as corners/planes [21]
- Parameterization – linear or curvature fitting, plane-fitting, model-fitting [26-29]

- Artificial Intelligence, Machine Learning, and Neural Networks – semantic segmentation, object detection and labelling [30-31]

Within the calibration approach outlined in this thesis, operations will be performed both directly onto the raw point cloud, and onto an image created from the point cloud in birds-eye-view format. Linear feature recognition from image processing and parameterization in the form of simple linear least squares solutions are also utilized. Linear least squares solutions are covered in more detail in Chapter 4, while the creation of a birds-eye-view image from a point cloud is covered in Chapter 3.

2.4 Transforms Background

Transformations and orientations between frames can be defined in many ways. Three common representations of rotational transforms (or orientations) include Euler angles, direct cosine matrices (DCMs), and unit quaternions [32]. Euler angles are roll, pitch, and yaw values that represent the rotational amounts which must be sequentially applied to a body to align it to a new orientation. A common sequence is yaw, pitch, roll (YPR) applied to a global frame to rotate it into a body frame, in which a vector in the global frame is first rotated by ψ radians about the Z_g axis, resulting in intermediate axes X_g', Y_g', and $Z_g' = Z_g$ as an application of the yaw rotation. The pitch rotation of θ radians is then applied about the Y_g' axis, resulting in the X axis aligning to the body frame, with axes $Y_g'' = Y_g'$, Z_g'', and $X_g'' = X_b$. Finally, the roll of ϕ radians is applied to the X_g'' axis, bringing the remaining y- and z-axes into alignment with the body frame. Equations (2.1 – 2.3)

$$x_b = \left(x_g \cos(\psi) + y_g \sin(\psi)\right) \cos(\phi) - \left(z_g\right) \sin(\theta) \tag{2.1}$$

$$\begin{aligned} y_b = {} & \left(y_g \cos(\psi) - x_g \sin(\psi)\right) \cos(\phi) \\ & - \left(z_g \cos(\theta) + \left(x_g \cos(\psi) + y_g \sin(\psi)\right) \sin(\theta)\right) \sin(\phi) \end{aligned} \tag{2.2}$$

$$\begin{aligned} z_b = {} & \left(z_g \cos(\theta) + \left(x_g \cos(\psi) + y_g \sin(\psi)\right) \sin(\theta)\right) \cos(\phi) \\ & + \left(y_g \cos(\psi) - x_g \sin(\psi)\right) \sin(\phi) \end{aligned} \tag{2.3}$$

show this process mathematically, with a rotation of vector $v_g = \left[x_g, y_g, z_g\right]$ into vector $v_b = [x_b, y_b, z_b]$. These equations can be represented in matrix form, creating a DCM in YPR sequence that can be called R_g^b. This DCM is given in Equation (2.4),

$$\begin{aligned} & R_g^b(\psi, \theta, \phi) \\ & = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ -\cos(\phi)\sin(\psi) - \sin(\phi)\sin(\theta)\cos(\psi) & \cos(\phi)\cos(\psi) - \sin(\phi)\sin(\theta)\sin(\psi) & -\sin(\phi)\cos(\theta) \\ -\sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi) & \sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi) & \cos(\phi)\cos(\theta) \end{bmatrix} \end{aligned} \tag{2.4}$$

and the DCM-form of the rotation application is shown in Equation (2.5).

$$v_b = R_g^b(\psi, \theta, \phi) * \ v_g \tag{2.5}$$

For a derivation of the YPR DCM, see Appendix A.

Given the orientation of two frames *b* and *c* defined with respect to a third frame a in DCM form R_a^c and R_a^b, the rotational difference R_b^c between the two orientations can be solved using Equation (2.6),

$$R_b^c = R_a^c * R_a^{b\prime} \tag{2.6}$$

where $\prime$ indicates a matrix transpose [33].

Alternatively, these DCMs can be represented in the form of the 4-member complex number known as a quaternion. Quaternions have the advantage over DCM representations in that they do not experience singularities at a pitch of 90 degrees (or any other operation that aligns two axes), and they are independent of angle sequence in their representation of orientation and rotational differences [33]. The equations for applying quaternion rotations to a vector as given in Equation (2.7)

$$v' = q \otimes v \otimes q' \tag{2.7}$$

and for solving for rotational differences as given in Equation (2.8)

$$q_b^c = \ q_a^c \otimes (q_a^b)^* \tag{2.8}$$

parallel those for the same operations with DCMs.

In Equation (2.7), a vector is converted into a “pure quaternion” v, such that $v = [x, y, z, 0\ ]$ and the real component of the quaternion is equal to zero. The quaternion used to rotate the vector is q, and the result is another pure quaternion v' , which may be converted into a vector by using the x, y, and z complex components of v' as the x, y, and z vector components [34]. Here, $\otimes$ represents quaternion multiplication which differs from vector and matrix multiplication, and * represents a quaternion conjugate. For more detail on these quaternion operations, see Appendix A.

Note that the transforms provided to this point have involved rotation only. They may be used to rotate two frames into alignment from an orientation perspective, but are not capable of removing translational offsets. To allow for defining a complete rotational and translational transform in a single entity, an affine matrix transform T_a^b between two frames a and b can be defined as a DCM appended with a translation vector and lofted into 4-D space as given in Equation (2.9).

$$T_a^b = \begin{bmatrix} R_{11} & R_{12} & R_{13} & x \\ R_{21} & R_{22} & R_{23} & y \\ R_{31} & R_{32} & R_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

T_g^b can then be used to bring a vector representing the frame g into complete rotational and translational alignment with a vector representing frame b according to Equation (2.10).

$$v_b = T_g^b * v_g \tag{2.10}$$

Within this discussion of representations of rotational and translational offsets, it is prudent to also discuss in more detail the representation of orientation by roll, pitch, and yaw angles. Namely, there is some confusion between the representation of roll and pitch angles as the instantaneous angle between a plane and a vector in space and the representation of it as a sequence of rotational corrections which may be applied to align two vectors. The first of which may be termed “instantaneous angles,” while the second are the Euler angles as mentioned before.

Instantaneous angles are a relatively intuitive means for defining static orientations of planes or bodies. An infinite projection of any given non-horizontal plane slices the axes of a reference coordinate frame so as to produce two right triangles of interest, one formed by the plane with the z and x axes of the reference frame and one by the plane with the z and y axes of the reference plane as shown in Figure 2.2.

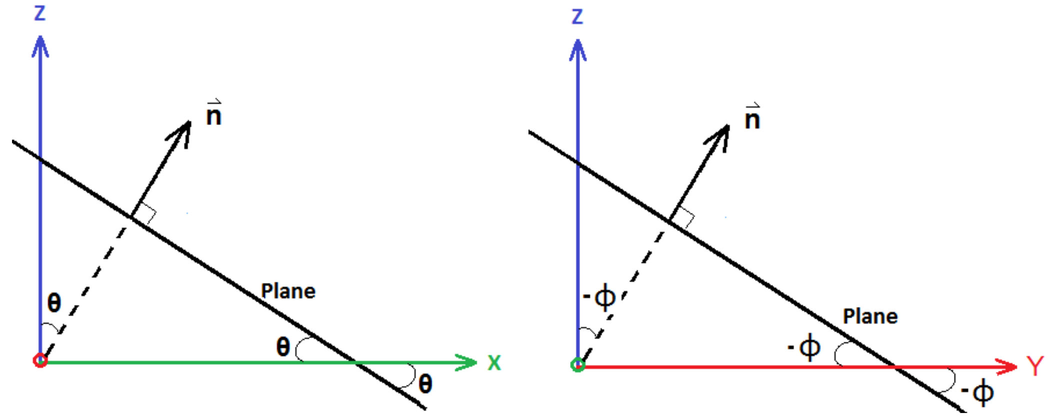


Figure 2.2: Instantaneous Angles Visualization. Left: Instantaneous Pitch, Right: Instantaneous Roll, with planar normal vectors $\vec{\boldsymbol{n}}$

The instantaneous roll and pitch can be calculated using Equations (2.11-2.12),

$$\phi = -\,atan\left(\frac{y}{z}\right) \tag{2.11}$$

$$\theta = atan\left(\frac{x}{z}\right) \tag{2.12}$$

where x, y, and z are the components of the normalized normal vector of the plane. These angles can also be obtained after a projection of the normal vector of the plane onto the x-z and y-z planes of the reference coordinate axes, as is done in [26].

While intuitive for stating orientations, instantaneous angles cannot be directly used to rotate data from one frame into another, and therefore are poorly suited for rotations which are intended to be used to align data or frames to one another. Euler angles, which can be converted readily into DCM or quaternion form, make much stronger calibration and alignment orientation candidates due to the direct application of quaternions and DCMs in rotating data and frames.

Euler angles, a DCM, or a quaternion representation of a plane's orientation may be obtained from the plane normal using rotational difference calculation between the plane normal vector and a vector representing the normal vector of the x-y plane of the reference frame, or [0,0,1]. Alternatively, the Euler angles in YPR sequence from the reference frame to the plane frame may be solved for directly using Equations (2.13-2.15) sequentially.

$$\phi = -atan\left(\frac{y}{z}\right) \tag{2.13}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.14}$$

$$\theta = atan2\left(\frac{x'}{z'}\right) \tag{2.15}$$

where $[x, y, z]$ is the normalized normal vector of the plane. The vector $[x', y', z']$ is an intermediate normal vector of the plane after the Euler roll is applied to the plane, but before a pitch rotation brings it into alignment with the reference frame.

In this thesis, several forms of the rotational and translational transforms will be utilized in different aspects of the solution. The final calibration transform is provided as a four-dimensional affine matrix as shown above, with a rotation transform in the form of a YPR DCM from reference frame to sensor frame. For further reading on transformations, the representations of translations and rotations, and the properties thereof, see [32-35].

2.6 Least Squares Solutions

Linear equations are any equation that comes in the form of Equation (2.16),

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{2.16}$$

where **x** is a value or set of values, and **A** is the slope or set of slopes that the unknown matrix **x** is conditioned by to produce **b**. In matrix form, **A** is any linear transformation which may be applied on a vector or matrix of values **x** to map them onto the vector or matrix of values **b**. Typically, **A** and **b** are known and solutions for **x** are desired.

When the values for A and b are singular, simple division may be used to solve for the likewise singular value of x. Similarly, when the matrix **A** is square and invertible, the matrix analog of single-value division may be leveraged. In many cases, the solution is over- or under-determined thereby creating a non-square **A**, and a somewhat more sophisticated approach must be used to solve for **x**. A relatively simple alternate solution is found in the form least squares approaches. The least squares problem is an optimization problem, where the solution for **x** given Equation (2.16) must have minimized squared error when evaluated with Equation (2.17) [36].

$$\left|\left|\boldsymbol{Ax}-\boldsymbol{b}\right|\right|^2 \tag{2.17}$$

The least squares problem is well-studied, and there are several methods for reaching a least squares solutions. The basic least squares solution is given in Equation (2.19).

$$\widehat{\boldsymbol{x}}=(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{2.19}$$

The implementation of least squares used within this thesis leverages the singular value decomposition (SVD). An SVD can be applied to a matrix such as **A** to decompose it into characteristic singular matrices **U**, **Σ**, and **V** in the format of Equation (2.19),

$$\boldsymbol{A}=\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\boldsymbol{V}}^* \tag{2.19}$$

where * is the complex conjugate transpose [36]. According to [36], SVD methods are “mature and simple”, and it is common to “abstract away the details underlying the SVD computation” by using functions from existing computer language libraries. In this thesis, the jacobisvd approach provided by the Eigen library has been utilized used to conduct the SVD of matrix **A,** and is designed to be computationally efficient over large matrices (such as an **A** matrix composed of a large point set). The Jacobi method was originally posited by C. G. J. Jacobi in 1846 in [37], and reiterated and analyzed in such papers as [38-39].

The solution for **A** as shown in Equation (2.19) can be substituted into Equation (2.17). In the overdetermined case where **x** has fewer rows than **A** and **b**, the least squares estimation of **x** is then given as $\tilde{\boldsymbol{x}}$ in Equation (2.20) [36].

$$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-\mathbf{1}}\tilde{\boldsymbol{U}}^{*}\boldsymbol{b} \tag{2.20}$$

Chapter 3

Yaw Calibration from Road Trajectory

3.1 Introduction

The proposed extrinsic calibration routine begins with estimation of the yaw-component of the calibration transform. This thesis seeks to derive the calibration of a LiDAR to a vehicle frame using only the data from the LiDAR itself, and as such must leverage the geometries present within the sensor-generated point cloud. Calibration using in-scan geometries requires the definition of a feature or object from which to derive the calibration. This feature must be capable of providing a value that can be compared to a known or expected value to determine the calibration offset, and it must be reasonable to expect the feature to be present within the LiDAR data during normal use.

To determine what feature or object is suitable for geometry-based yaw calibration, consider the following example. A LiDAR with a yaw-offset between its frame and the vehicle's reference frame will be rotated about the vehicle's z-axis such that the x- and y- axes of the vehicle do not point in the same directions, as shown in Figure 3.1. This rotational offset will express itself as misalignment between the forward (or positive x) direction for the LiDAR and the forward direction for the vehicle.

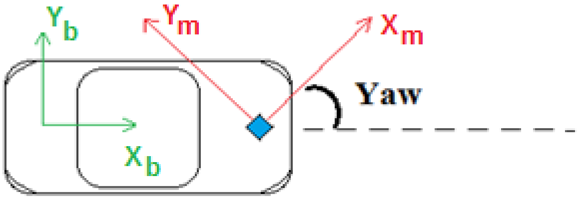


Figure 3.1: Illustration of a LiDAR with Axes X_m, Y_m Mounted with a Yaw Offset to the Vehicle with Axes X_b and Y_b.

This suggests that the simplest choice for the comparative object for yaw calibration purposes is the forward direction or x-axis of the vehicle with respect to the LiDAR, which is expected to be aligned to the LiDAR x-direction. However, the forward direction of the vehicle is not easily observed within the LiDAR frame. It might be observable if a straight or flat surface on the car that was confirmed to be parallel or perpendicular to the vehicle forward direction was visible to the LiDAR. However, this is often not the case, as LiDAR sensors are typically mounted to view the scene around the vehicle and not the vehicle itself.

In place of a direct measurement on the vehicle, an external object or feature must be chosen as a stand-in. When considering scenarios in which a vehicle will be aligned to some external structure, travel along a distinct roadway comes to mind. During normal travel along a roadway, when no turns or lane-change maneuvers are being made, the vehicle may be expected to closely follow the trajectory of the roadway. During straight travel, the direction of travel is parallel to the forward direction or x-axis of the vehicle. If the road trajectory is observable by the LiDAR, the heading offset of this trajectory from the LiDAR x-axis will be approximately equal to the yaw offset between the LiDAR and vehicle frames.

The detection of road edges, lane markings, and road space with LiDAR data is well-researched. All of these detections inherently yield a measure of the local trajectory of the road, and so a version of one of these detection approaches would provide the comparative object data that is needed for this calibration. The majority of the existing road detection approaches, discussed in more detail in the following section, require that the LiDAR calibration is known or the mounting offsets are negligible, which complicates the use of road edge detection approaches for yaw calibration. This chapter proposes a novel road edge detection approach based on the Hough transform which allows for the detection of arbitrarily located and oriented road lines with respect to the LiDAR point-of-view. This road edge detection approach is then used to determine the yaw calibration offset between the vehicle and LiDAR.

3.2 Road Edge and Road Trajectory Background

This approach requires that the road edge be detected and parameterized. Much progress has been made on road edge detection, specifically intensity-based lane line detection using LiDAR point clouds. From a LiDAR perspective, road or pathway edges may be defined by a variety of qualities, including distinct ground roughness change at the path edge, height change at the path edge (such as a curb), and painted or material markings with significant intensity difference to the surface of the roadway (such as painted lane lines on most paved roadways).

There are various approaches for detecting road edges or road lane markings from LiDAR. These approaches may be roughly broken down into four categories: neural network and machine learning approaches, utilization of histograms of intensity values, direct operation on 3D point clouds, and operation on 3D-to-2D mappings of the LiDAR data.

3.2.1 Neural Network and Machine Learning Approaches

One of the more under-researched categories of LiDAR-based lane line, road edge, or road space detection is the category of artificial intelligence approaches such as those involving neural networks or other machine learning approaches. Not much documented progress can be found in this area, and [40-42] are some of the few available resources on the topic. Though their 3D lane-line marking results are promising, neither [40] nor [42] were designed to work with LiDAR data alone, and instead use the LiDAR data as augmentative input to primarily image-based detection networks. LiDAR data is used as the sole input in [41] in which a fully convolutional neural network is used to semantically label the pixels of a 2D, multi-channel birds-eye-view image mapping of LiDAR data. The authors show promising success, but are effective only in determining the pixel-wise boundaries of the expected road space, with strange road geometries arising around non-ego vehicles on the roadway. While it may be possible to conduct further processing on these labeled pixels to extract the road edges or road trajectory, this approach does not directly provide parameterized curves describing the roadway, as is needed in the calibration approach for this thesis.

As with many other applications of neural networks to the field of vehicle and robotic perception, much more significant development has gone into artificial intelligence for detecting road lines from camera images than from LiDAR point clouds. While it is theoretically possible to apply these well-developed networks and studies to images created from LiDAR data, no documented study on such an approach was found in literature.

3.2.2 Histogram Approaches

The use of histograms to aid in LiDAR lane marking detection is well-documented. Example approaches leveraging histograms include [10, 43-44]. Kibbel, et al. in [43] convert 3D point clouds into Cartesian histograms that span laterally across the roadway. This lateral Cartesian histogram has bins which each cover 0.1 m across the roadway, with bins accumulating counts of points that have intensities higher than a set threshold and occupy that lateral region in space assigned to the bin. The bins with the highest number of high-intensity points are detected as likely lane marking locations. As an alternative to a Cartesian lateral histogram, [44] parses its intensity-thresholded points into a 2D polar histogram, where the left-to-right bins represent azimuthal angle increments and bottom-to-top bins represent distance from the sensor. This polar histogram allows the approach from [43] to be cognizant of the distance of point returns from the sensor, enabling distant points with lower intensity returns than closer points to be thresholded differently than their close counterparts. The polar arrangement allows same-row bins to be equi-distant from the sensor origin, allowing different intensity expectations to be applied at different rows or distances. In [10], a cross-road histogram is compared to a histogram of an ideal road scan, and the RMSE of bins is used to detect lane line positions.

3.2.3 Direct Point Cloud Approaches

Examples of approaches working directly on point cloud data include [45-46]. In [45], an intensity filter and angular region-of-interest are used to filter lane marking points on either side of the vehicle prior to applying parameter calculation of azimuth and curvature of the point lines. In [46], similarly, an adaptive region-of-interest is used to filter the point cloud prior to intensity-filtering. Then, a parabolic equation is fit to the detected point sets, and trajectory parameters such as vehicle yaw offset and road curvature are calculated directly from the parabolic fit parameters.

3.2.4 Mapping to 2D Approaches

Approaches which first convert the 3D LiDAR point cloud into a 2D image prior to processing include [22, 31]. In both of these examples, the point cloud is intensity-filtered and then parsed into monochrome birds-eye-view Cartesian 2D grid images. After conversion into 2D, classical image processing approaches for lane marking detection may be applied. In [22], the points are region-of-interest filtered and the selected points subjected to a parabolic line fit. In [49], a more typical image processing approach is applied in the form of a Hough transform, which searches through the image for strong linear feature candidates in parameter space.

3.2.5 Drawbacks to Common Road Trajectory Detection

Three common sub-steps used in many LiDAR road edge and lane marking detection approaches are intensity filtering, the aggregation of several point cloud frames over time using an ego motion estimates, and the use of windows- or regions-of-interest. Intensity filtering works for painted road marking detection by removing lower-intensity road surface points that are not painted, and leaving the higher-intensity points returned from the painted surfaces. Aggregation of multiple frames of point clouds over time works to increase the density of the point cloud data prior to processing, thus increasing the detectability of the linear road features [46]. However, aggregation requires a reasonable estimate of the motion incurred by the vehicle between each assimilated frame, such that the frames can be transformed and properly registered together. Regions-of-interest serve to reduce the search space the detection algorithm has to work with, and can be used to remove false positives and noise point data. However, these regions of interest often make inflexible assumptions about the positioning of the linear features relative to the LiDAR frame which may not always hold true.

Many approaches to lane-line and road edge detection utilize regions-of-interest. In [45], angular regions-of-interest where the lane lines should be are paired with intensity filtering to acquire road marking points every 0.5 meters directly from the point data. In [22], the point data is transformed into birds-eye-view 2D images before the expected lane width is used to apply regions-of-interest in the image over expected lane marking positions. In [10], a pre-determined ideal scan which assumes the road width is known and constant is used to detect subsequent live lane markings by thresholding the difference between the current real scan and the ideal one. This effectively sets windows of interest located at the lane marking positions from the ideal scan. Other approaches do not rely on a pre-set region-of-interest, but rather take an initialized expected position and estimate the useful window positions over time. In [47], an intensity filter is applied before using an initialized-then-estimated road width to determine expected lane line positions.

Regardless of the type of approach, the algorithms typically assume that the LiDAR-to-vehicle calibration is known, and that regions-of-interest or ideal scans can be designed utilizing lane markings or road edges entirely to the left or entirely to the right of the point cloud. A vehicle traveling along a pathway with distinctive edges will generally see one edge to its left and one edge to its right when the perception sensor is aligned with the forward trajectory of the vehicle. This assumption does not hold well when significant yaw offsets are present and unaccounted for between the LiDAR sensor mounting and the frame of the vehicle, complicating the use of road edge detection approaches for yaw calibration.

3.2.6 The Haugh Transform Method

To overcome the minimal yaw-misalignment requirement, an algorithm that is capable of conducting an efficient search over the entire intensity-filtered point cloud is needed. A Hough transform provides one such search algorithm to detect linear features. Hough transforms are common in image-based road edge and lane line detection, and can be extended to 3D point cloud lane detection. In [49], a 3D point cloud is intensity filtered and converted into a birds-eye-view

2D intensity image. Once the 2D intensity image has been created, a Hough transform is used to determine lane line candidates from strong linear feature agreeance. To select the best lane line candidates from the Hough output, the line candidate with the maximum number of parallel candidates is selected along with all parallel candidates. Prior to filtering and conversion into 2D, several frames of LiDAR data were assimilated using ego motion prediction to increase the density and therefore linear strength of the lane marking points in the intensity image. To acquire ego motion estimates, either a separate sensor providing accurate ego motion states or a successful implementation of LiDAR odometry are required. The limitations inherent to LiDAR odometry approaches (e.g. the seldom-met requirements that environmental and object geometry be stronger than scan acquisition geometry across the entire point cloud, and that no objects move with respect to the environment be observed by the LiDAR) together with a drive for greater computational efficiency motivate an alternative approach. In this thesis, an approach which does not require aggregation of multiple frames is developed.

3.2.7 Considerations for the Proposed Road Trajectory Detection Approach

A novel road edge detector based on the Hough transform lane detection method is proposed and subsequently applied for determining the yaw offset between the vehicle and the LiDAR. The Hough transform is leveraged for its ability to detect arbitrarily oriented linear features, and therefore can detect linear road trajectory indicators in uncalibrated LiDAR data. The proposed approach differs from [43] primarily in that our approach is designed and tested for sparse LiDAR data so as not to require ego motion estimation.

3.3 Assumptions Required for the Proposed Yaw Calibration Method

There are several assumptions that make this yaw calibration from road trajectory detection possible:

1. The vehicle is traveling on a path or road
2. Path or road edges are distinct in height, intensity, or texture/roughness
3. Path edges are visible to the LiDAR
4. The vehicle closely follows road trajectory
5. A straight section of the roadway is traversed and is detectably straight
6. The LiDAR is mounted with a yaw offset magnitude of no more than ± 45 degrees

The first assumption is that the vehicle the LiDAR is mounted to (and being calibrated to) is traveling along a distinct pathway, such as a paved public road. Given the second assumption, that the pathway has well-defined edges, a LiDAR-based algorithm is capable of detecting the edges. It is further assumed that the vehicle is traveling parallel to the roadway it is on. A vehicle which travels on a well-defined physical pathway will typically try to match the trajectory of the pathway unless departing from a lane or the entire pathway. Lastly, it is assumed that there is a straight section of roadway that is traversed during the application in which this calibration approach is applied. During a straight section of roadway, the trajectory of the vehicle will be parallel to both the forward direction or x-axis of the vehicle and the trajectory of the roadway. As such, the vehicle trajectory is indirectly observable by the LiDAR without the need for motion estimation. The sixth assumption that the LiDAR yaw offset be no larger than 45 degrees in either direction is due to the minimum-yaw selection criteria which allows the Hough transform to select for arbitrarily-oriented road edges or lane marking lines given sparse LiDAR data.

3.4 Details of the Proposed Yaw Calibration Method

The starting point for the proposed online autocalibration routine is the estimation of a yaw calibration transform. When the vehicle is traveling parallel to the trajectory of the roadway, the roadway within the LiDAR data will appear parallel to the x-axis of the LiDAR when the LiDAR is yaw-aligned to the vehicle frame. Conversely, when the LiDAR is not yaw-aligned to the vehicle, the road edge will appear diagonal with respect to the LiDAR x-axis from the LiDAR perspective. The degree to which the roadway is diagonal relative to the LiDAR x-axis is approximately the yaw offset between the LiDAR frame and the vehicle frame. The x-axis of the LiDAR frame is defined by the LiDAR manufacturer, and is expected to coincide with the forward direction of the sensor, as intrinsic calibration has already been conducted by the manufacturer. As such, the yaw calibration step requires that the road edge be detectable, and detectably straight. An example of a LiDAR with a distinct yaw calibration offset is illustrated in Figure 3.2, where the assumed parallel nature of the vehicle and roadway trajectories during the straight section of roadway is present.

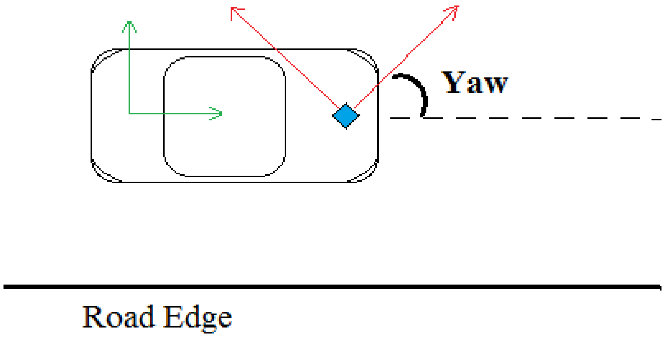


Figure 3.2: Yaw Calibration Offset from Road Trajectory Offset

Prior to executing the yaw calibration routine, a road edge or lane marking must be detected. If the road edge trajectory is estimated to be linear at a given instant (i.e. the road is not currently curving), the yaw calibration procedure can be executed. The true detection of whether the road is curving or straight most explicitly requires a road edge detection algorithm that is capable of estimating curvature. The detection routine utilized for this calibration step is capable only of approximating the current road edges as straight, linear features. To bypass the need for a more complex road edge detection at this time, it is possible to leverage the fact that the straight edge detection method provides noticeably unreliable results during curved sections. As such, a straight section of roadway is assumed to be present whenever consistent edge detection results are available from the detection algorithm. The yaw calibration is therefore only conducted when the road edge detection has been valid for *n* frames, in this thesis for 10 consecutive frames.

The road edges are detected by first filtering the LiDAR point cloud by an intensity threshold. In this thesis, any intensity value over a pre-determined threshold of 35 out of a range of 0 to 256 is included in the output cloud, and any lower-than-35 intensity points are excluded. This follows from the principle that asphalt is a low-reflective surface in the IR spectrum, while the paints used to draw lane lines are highly reflective in the IR spectrum. This juxtaposition of low-reflectivity ground and high-reflectivity edge markings makes it possible to define a threshold that maintains painted marking lines while removing general ground points. It should be recognized that a height-based filtering is also an option in the event that the road or path edges are distinctive in height, such as when the roadway is bordered by curbs or barriers. In roadways with curbs, the user should define a threshold to remove all points below a given height. The particular value of the intensity or height threshold will vary depending on the qualities of the LiDAR used and environment the system is operating in. Two examples of point clouds before and after intensity filtering are shown in Figure 3.3. The data shown in this figure was collected using a Velodyne VLP-16 Puck LiDAR [15] on Highway 280 in Alabama.

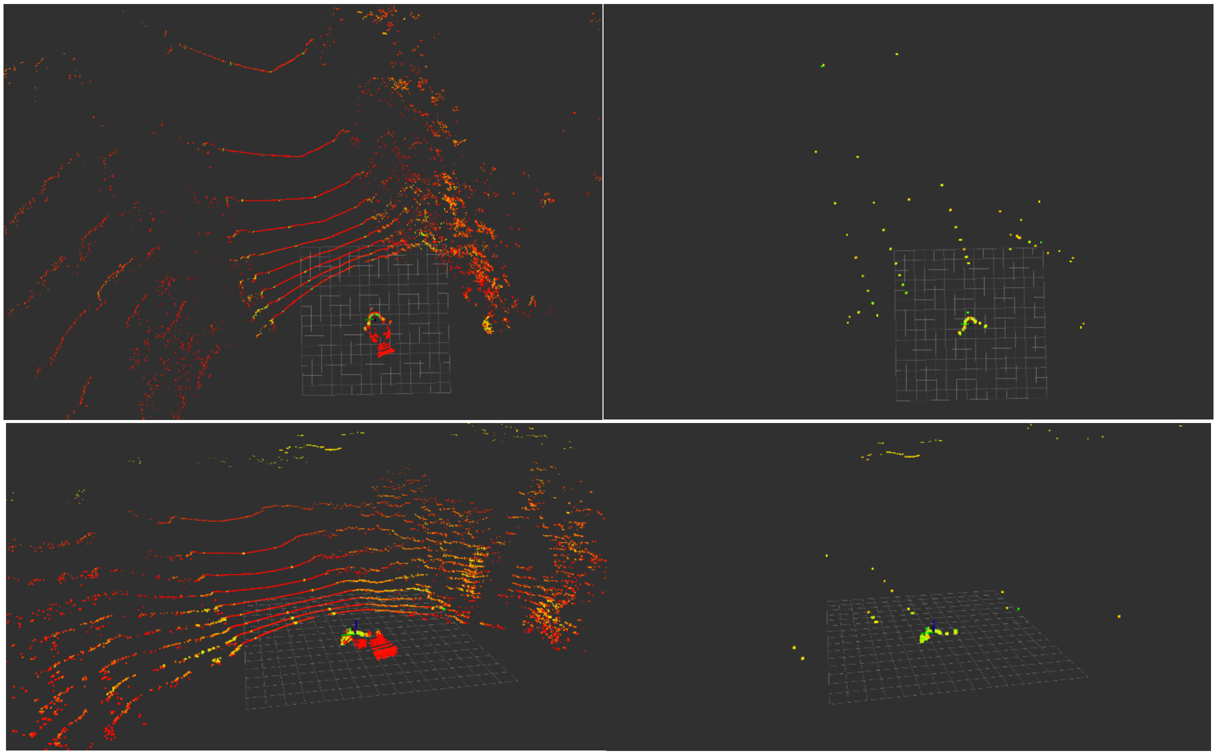


Figure 3.3: Top: Top-Down View of 3D Point Cloud (Left: Full cloud, Right: Intensity-filtered cloud); Bottom: Arbitrary Perspective of 3D Point Cloud (Left: Full cloud, Right: Intensity-filtered cloud)

A pass-through filter which filters by both point height and intensity is given in Equation (3.1),

$$P_i' = P_i\begin{pmatrix} x_j \\ y_j \\ z_{low} \leq z_j \leq z_{hi} \\ I_{low} \leq I_j \leq I_{hi} \end{pmatrix}, j = 0, \dots, m \tag{3.1}$$

where P_i' is the resultant filtered cloud. The upper and lower bounds for each value are determined by the user prior to implementing the algorithm, and are designated here by the $_{low}$ and $_{hi}$ subscripts for minimum and maximum bounds respectively. The pass-through filter step serves to remove ground points which do not represent the road edges as well as many object and noise points. Even well-tuned positional and intensity bounds will not successfully remove all non-road-edge points. Road clutter points as well as some intensity values from objects such as vehicles and road signs will remain.

After thresholding the intensity or the height, the filtered LiDAR point cloud is converted into monochrome birds-eye-view images. The pixel values used in the images are either zero to indicate no point presence, or 255 to indicate point presence. These values are valid for an image with 8-bit pixel encodings, and appear as black and white pixels respectively. This black-where-empty, white-where-occupied image is also known as a binary occupancy grid. The 3D-to-2D mapping here takes the form of P_i' being parsed into a 2D grid image with image size and a resolution ν in meters/pixel as set by the user. In this thesis, a resolution of ν = 0.05 meters was used over 100 x 100 meters in y and 98 meters in x. Grid parsing is conducted as a forward mapping between the point cloud and the resultant image, where all points from the filtered cloud will be present in the image, but not all image pixels will be updated by a point. This is accomplished by converting each $(x, y)_j$ point into a set of image coordinates $(k, l)_j$ using Equation (3.2).

$$k_j = \frac{x_j - x_{min}}{\nu},\ l_j = \frac{y_j - y_{min}}{\nu},\ j = 0, \dots, m \tag{3.2}$$

The image C_i is initially set to have all zero (black) pixels. The image coordinates $(k, l)_j$ are used to directly index into the image array, and the pixel value is updated at that location to be 255 (white) to reflect the presence of a point from the filtered cloud as shown in Equation (3.3).

$$C_i[k_j, l_j] = 255, \qquad j = 0, \dots, m \tag{3.3}$$

The result at this point is an image of primarily black pixels with white pixels where the road edge points and some noise points are present, as seen in Figure 3.4.

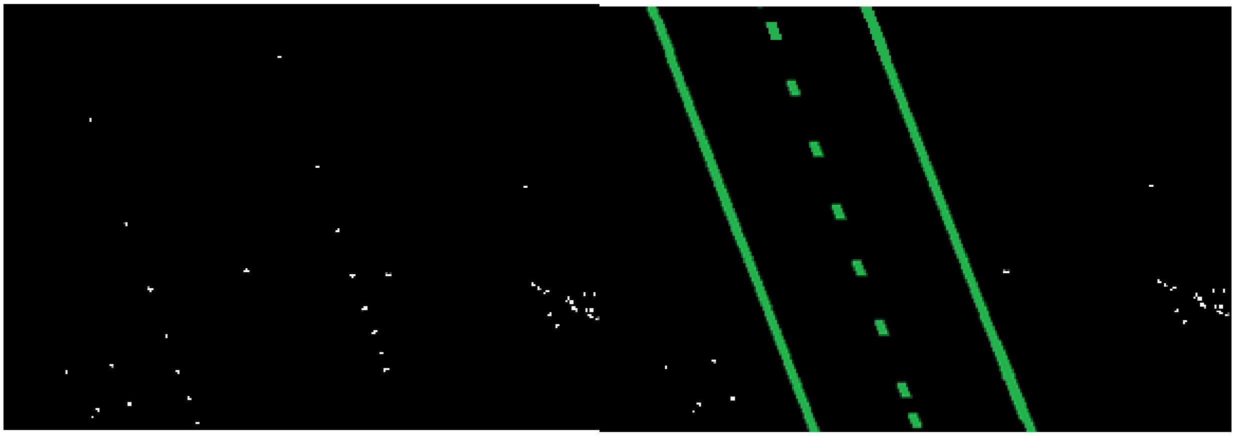


Figure 3.4: Left: Example birds-eye-view image of an intensity-filtered LiDAR point cloud, where the point cloud was collected on a roadway with painted lane and edge lines. Sparse LiDAR points (white) and non-road-edge space (black). Right: Indication of where the lane markings are located relative to the image (manually labelled).

The filtered images are then subjected to a progressive probabilistic Hough transform to extract linear candidates [48]. A Hough transform detects linear features within images, and yields a set of parameters for the line results. The basis of the algorithm is a search through parameter space to identify lines with a large number of “votes.” Parameter space is defined by the parameters (a, b) of a standard 2D line equation, as shown in Equation (3.4),

$$y = ax + b \tag{3.4}$$

where the parameters a and b represent slope and intercept, respectively, and x, y are coordinates of points on the line.

The count of votes for each parameter coordinate (a, b) is set to the number of pixels which fall within a threshold of the line represented by those parameters when drawn onto the black and white point image. Line parameter sets with a large number of votes are reported as detected lines. An example of a small parameter space search is shown in Figure 3.5. The most likely line detected in this example has parameters a = 5 and b = 3. Probabilistic Hough transforms increase algorithm efficiency by selecting a subset of input pixels on which to perform the transform. The progressive probabilistic Hough transform differs from other probabilistic Hough transforms in that it attempts to qualify voting points as noise or not based on an adaptive threshold on the number of votes each point has cast, where the threshold is updated as more votes are cast throughout the image to better qualify the noise-point vote count. The more lines a point falls on or votes for relative to the other points in the image, the more likely the point is noise. Essentially, the progressive probabilistic Hough transform is designed to be more robust to noise points than the Hough transform alternatives. A convenient implementation of the progressive probabilistic Hough transform can be found in the OpenCV library in the *HoughLinesP* function.

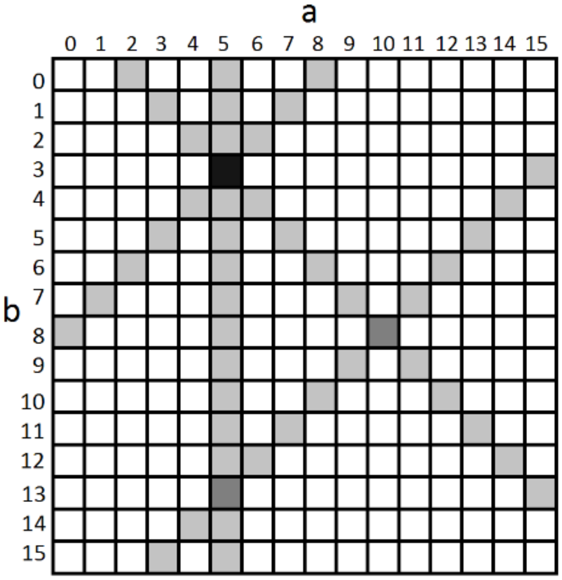


Figure 3.5: Example simplified Hough parameter space search result over 256 line options, with 4 voting points. The more likely parameter combinations are shown with a darker cell value (white – no votes, light gray – 1 vote, dark gray – 2 votes, black – 3 votes)

The parameters which result from the Hough transform are either naturally in the form of *a* and *b* from Equation (3.4), or can be easily converted into this form, depending on the implementation of the Hough transform being used. The yaw angle of each line candidate l_k can be solved for from its parameters $(a, b)_k$ using Equation (3.5),

$$\psi_k = \operatorname{atan}\left(\frac{1}{a_k}\right), k = 0 \ldots K \tag{3.5}$$

where ψ_k is the yaw angle in radians of the line from the image x direction or vertical, a_k is the line's slope parameter from Equation (3.4), and $k = 0,\ldots, K$ where there are K lines in L_i.

From the set of Hough result lines, the road edges are selected for based on a minimum-yaw-calibration and thresholded parallel criteria shown in Equation (3.6).

$$L_i' = L_i(l_k \; s.t. \; (\psi_k - \psi_{i\,min}) < \Delta\psi), \qquad k = 0, \ldots, K \tag{3.6}$$

Here, L_i' is the list of selected road edge lines for frame i, L_i is the total set of Hough line results, and l_k is a single line result $(a, b)_k$. $\psi_{i\,min}$ is the minimum yaw angle over all of the lines in L_i, and $\Delta\psi$ is a user-determined threshold for how different the selected road edge yaw angles may be from the minimum yaw line and still be considered parallel to it. In this thesis, the yaw threshold used was 8.6 degrees. The average angle-from-vertical of the selected road edge lines in L_i' is reported as the current road trajectory angle, which is an inverted estimate of the yaw calibration offset.

This differs from the selection criteria used in [49] due to the single-frame LiDAR data utilized in this thesis. Since ego motion has not been used to aggregate several frames of consecutive data together, the intensity images used in this thesis are much sparser than were leveraged in the approach in [49]. These sparse images have much more empty space between the pixels indicating lane marking or edge presence, which generates fewer Hough votes and thus makes the lines less prominent to the detector. The strength of lines drawn from noise and non-road-edge and non-lane-line points is increased with respect to the true detections, and thus many semi-parallel false line detections arise from even a well-tuned Hough transform. As such, the max parallel criteria leveraged in [49] is no longer best suited to detecting lane lines or road edges. Instead, an assumption is made that the LiDAR is not tilted more than 45 degrees in the yaw direction to either side of the roadway, and as such the best candidate lane lines will be those with minimum angle from vertical in the intensity image. Further detail on the custom Hough transform with birds-eye-view image approach is given in the following section.

The Hough line results and selected road edge line selection are visualized for a single frame in Figure 3.6. This approach avoids drawing regions of interest to the left and right of the sensor, or otherwise operating under the assumption that the road edges should be left or right of the sensor or separated by pre-set or estimated road or lane widths. Though highly dependent on the resolution of the LiDAR and the minimal presence of other high-intensity features in the field of view, this approach allows for the road edge to be detected at an arbitrary orientation relative the sensor, which is necessary when using these detections to estimate a potentially large yaw calibration offset (up to 45 degrees offset for most LiDARs) between the sensor and the vehicle body.

Figure 3.6: Left: Hough line detection results, Middle: parallel thresholded minimum yaw lines, Right: selected left road edge as projected onto the full 3D point cloud

3.5 Experiment and Results

An experiment was conducted to assess the performance of the road trajectory detection algorithm as presented in the previous section. In this section, the experimental methods are first outlined, including the equipment and setup used, procedures enacted, and the theory and expected learnings to be gained from the experiment. Following the overview of the methods, the experimental results are given and discussed.

3.5.1 Experimental Methods

A test using real-world highway data was conducted to evaluate the ability of the road edge and trajectory algorithm to properly estimate the yaw angle of the road trajectory. The data was collected using a Peterbilt 579 tractor-trailer with a mechanically rotating, 16-channel, 360-degree Velodyne VLP 16 LiDAR [15] mounted on the hood of the truck. The LiDAR was mounted with an unmeasured yaw offset and a roll and pitch according to the slope of the hood at the mounting position (also unmeasured). The truck was then driven on a section of US Route 280 east of Opelika, Alabama and the LiDAR data was collected.

During the drive, painted lane markings across the roadway in front of the vehicle were visible within the field of view of the LiDAR. In post-processing, the proposed lane detection algorithm was applied and the road trajectory-to-LiDAR-forward yaw offset was estimated for each frame of LiDAR data. This provided a set of algorithm result data in the form of yaw angles in radians.

Truth data was generated for a set of 20 frames through manual segmenting and labeling methods. To generate this truth for each frame, the raw LiDAR point cloud was saved in x, y, z, intensity format using comma separated value (csv) files. The point files were then opened in the CloudCompare point cloud editing software for editing and visualization. Using the CloudCompare value filter, the clouds were thresholded using hand-selected minimum intensity values (which varied ± 5 about 35 depending on the best quality of results per-frame from a human perspective) until the lane marking points represented the majority of the remaining cloud points, without parsing out many of the marking points with somewhat lower intensity. After applying the intensity filter, points belonging to each lane marking was manually selected and saved as a separate file for each frame. These lane truth points were then each processed with a simple linear fit algorithm, and the yaw parameter calculated from the linear fit as shown in Equation (3.5). Figure 3.7 shows an example of a left and right lane marking truth line generated from the truth lane marking points intercepts and yaw offsets. Also displayed on this plot are lines representing the average yaw offset for the road trajectory provided by the algorithm, and the average yaw offset for the truth road trajectory points. These average yaw offset visualization lines are plotted with intercept at the origin of the LiDAR frame.

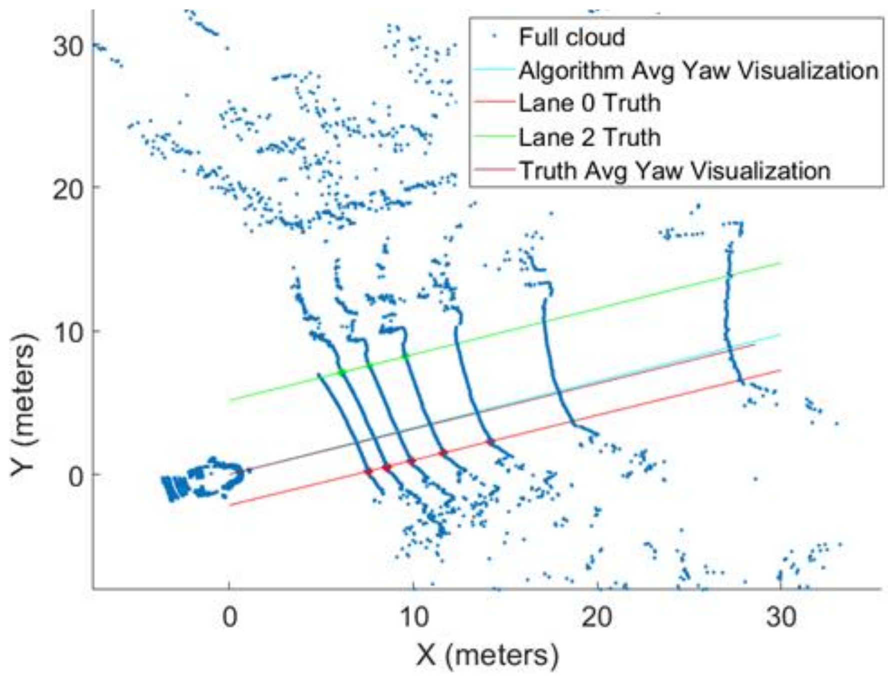


Figure 3.7: Road trajectory result (light blue), annotated truth lane lines (red, green), and road trajectory truth (purple). Road trajectory result and truth are nearly overlapping.

After algorithm results and truth data are both available in the form of road trajectory yaw offsets in the LiDAR frame, a direct comparison can be made for each frame of data. Direct comparison at this point will yield a plot where the yaw error magnitude is less than 30 degree for most frames, and far exceeds 30 degrees for a smaller portion of the frames. This is a result of the inability of the road edge detection algorithm to provide a road edge result within certain frames, such as when the road edge points are not visible within the filtered point cloud image, or the non-road-edge points are dominant over the road edge points throughout the algorithm. It is more meaningful to acknowledge these missed detections separately from the frames where road edge was detected than it is to accumulate all errors equally. Therefore, this test considers two statistics to qualify the road edge and trajectory detection algorithm: 1) percentage of frames in which the true road edge features were detected and 2) the accuracy in degrees of the estimated road trajectory yaw from successful detections. For the first metric, failed detections were defined as frames with the road trajectory yaw error above a certain threshold, in this case 30 degrees. The error from the failed detections are not included in the overall accuracy rating for valid detections.

During live operations, it is not possible to separate the true detections from the false detections using error-to-truth, as the truth would not be known. Instead, either all non-zero yaw values are assumed to be valid, the error of the yaw value from one frame against the yaw from the previous frame may be thresholded (i.e. removing large changes in yaw offset), or a filtering mechanism may be put in place. During the calibration routine conducted during further experiments in Chapter 6, the first 10 consecutive frames with non-zero, closely-valued yaw values from the road edge detection were used to estimate the yaw calibration offset.

3.4.2 Error Calculations

The error in this experimental analysis is reported as root mean square error (RMSE). This is calculated according to Equation (3.7),

$$RMSE = \sqrt{\sum_{i=0}^{n} e_i} \tag{3.7}$$

where n is the number of data elements and is equal to the number of frames processed, and e is the error, or difference, between a truth value and a value estimated by the LiDAR and LiDAR algorithms. In this experiment, the error is the difference between the estimated yaw offset of the detected road trajectory and that of the true road trajectory with respect to the LiDAR for each frame or scan of LiDAR data as shown in Equation (3.8).

$$e_i = \hat{\phi}_i - \phi_i \tag{3.8}$$

3.5.3 Experimental Results

Both the full error plot and the error plot with "non detection" frame errors removed can be seen in Figure 3.8. As can be seen by the relatively low number of negative peaks in the complete error line, the algorithm was able to generate a value for the road trajectory yaw offset, and therefore was able to detect a road edge, more often than it failed to do so. Further, by examining the fixed error line with non-detection errors removed, it is clear that the error, not considering missed detections, was far less than 20 degrees.

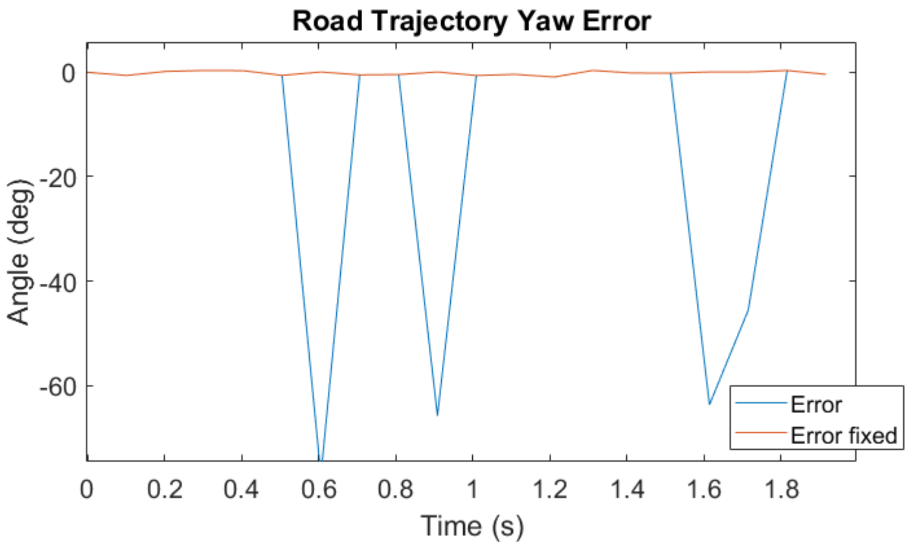


Figure 3.8: Error between LiDAR-estimated road trajectory yaw and truth road trajectory yaw. Full error results (blue) and error results with failed detection instances removed (orange)

Quantitatively, the road edge and trajectory algorithm was able to successfully detected valid road edges in 80% of frames. Within the cases that a valid road edge was detected, this approach was able to estimate the road edge orientation, or the road trajectory angle, to 0.47 degrees RMSE. A closer view of the fixed error plot is available in Figure 3.9, where it is apparent that the error ranges from about -1 degree to less than 0.5 degree. It should be noted that the data plotted here contained only nominal, continuous roadways, and did not involve such edge cases as lane splits or merges, exit or entrance ramps, or intersections, which would likely increase the mis-detection percentage.

Figure 3.9: Close up of error between LiDAR-estimated road trajectory yaw offset and truth road trajectory yaw offset. Full error results (blue) and error results with failed detection instances removed (orange)

Chapter 4

Pitch and Roll Calibration from Ground Detection

4.1 Introduction

4.1.1 Selection of Comparative Geometry

The second step of the proposed extrinsic calibration approach involves the estimation of the roll and pitch calibration offsets. Determining calibration offsets using geometries present within the LiDAR data alone, as is pursued in this thesis, requires that comparative data be generated from objects in the environment. For the yaw calibration discussed in Chapter 3, the comparative object was the visible road edges which provide a direct measurement of the LiDAR yaw offset. For roll and pitch calibration, a similar direct offset measurement may be used if a plane of interest can be identified that can be aligned to the desired reference frame x-y plane.

In on-road applications with approximately front-facing LiDAR sensors, the roadway provides the constant presence of a reasonably planar surface within the LiDAR view during most if not all operations. This makes it a favorable geometry choice for use in determining roll and pitch offsets.

Characterization of the ground plane generally yields estimates of the roll and pitch orientation of the plane, while the yaw orientation of the plane remains ambiguous and unobservable unless certain features are also observed, such as road edges, lane markings, or orientations of other road markings such as arrows. Such yaw-from-features calculations were covered in Chapter 3 and remain independent of plane characterization.

An issue in the roll and pitch calibration arises from the dual requirement that the LiDAR be yaw-aligned to the vehicle frame (such that the ground in front of the LiDAR may be assumed to be road- or path-surface, and therefore approximately planar) and that the vehicle be at-rest when the ground plane estimation and calibration be conducted. The yaw-calibration approach must therefore be conducted and the estimated yaw correction applied prior to conducting the roll and pitch calibration. However, the yaw calibration approach utilized here requires that the vehicle be aligned to a straight section of roadway with distinct edges or markings such that the trajectory of the roadway is parallel or nearly parallel to the forward direction of the vehicle. This assumption is most readily held when the vehicle is traveling along straight sections of roadway, and therefore the vehicle must have started its road-traversing application prior to yaw calibration. This roll and pitch calibration approach expects that the vehicle be at-rest on a level section of ground for a few seconds when it is conducted, which is most likely to be valid prior to the start of an application before the vehicle begins to move. To overcome these conflicting calibration scenarios, the data from the start of an application may be saved such that the prior at-rest point clouds may be used in the roll and pitch calibration process after the vehicle moves and the yaw calibration is possible.

4.1.2 Further Discussion on the On-Ground Reference Plane

When a sedan-type passenger vehicle is sitting stationary on a planar surface, the vehicle body x-y plane as defined in Figure 1.2 will sit at a static roll and pitch offset from the ground, determined by the at-rest suspension deflection and mounting offsets between the vehicle and the x-y plane of the axles. As the vehicle remains stationary and static, the axles themselves have a static roll and pitch offset to the road surface determined by the mounting and deflection of the tires. For other vehicles and mobile robots, similar overall static deflections between body and ground planes will likely be present.

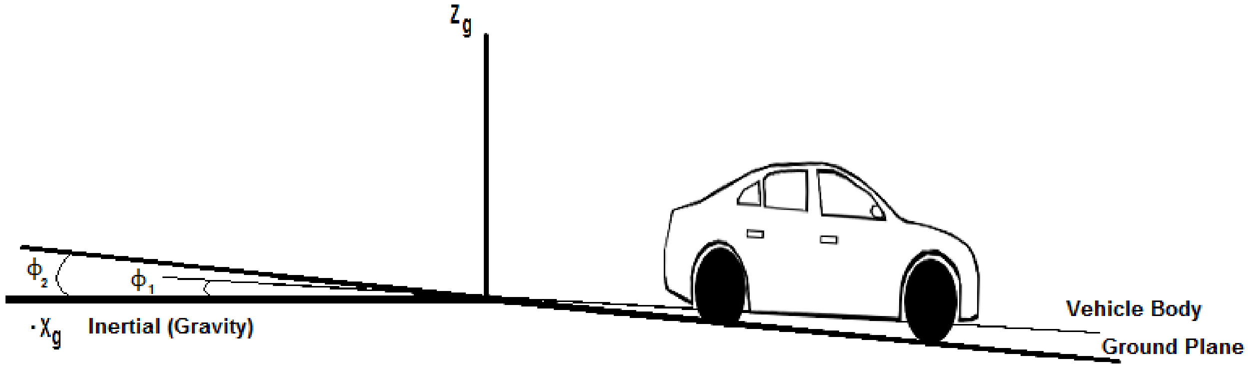


Figure 4.1: Sedan Vehicle At-Rest, With Ground-to-Gravity (ϕ_1) and Vehicle Body-to-Gravity ϕ_2 Roll Offsets

When the vehicle is at rest, the combination of the static roll and pitch offsets for the body-to-axles and axles-to-ground transforms result in a single static vehicle-to-ground offset. The reference frame used in this calibration is defined as the on-ground projection of a frame aligned to the vehicle body with the origin at the vehicle control point, as visualized in Figure 1.2. The ground plane may not be parallel to the x-y plane of the vehicle body at this point, as the suspension elements between the vehicle and ground do not guarantee zero-pitch and zero-roll when at rest, which can be seen illustrated in Figure 4.1. However, it is expected that the vehicle will generally express the same static offsets when coming to rest on any two ground planes, given that the two ground planes have the same grade and bank angles. The static at-rest offsets are expected to vary as ground plane orientations with respect to gravity vary. Consequentially, when calibration offsets are estimated using the LiDAR-to-ground observations, there will always be an unmeasured rotational offset between the on-ground reference frame and the true vehicle body frame, variable by the grade and bank of the ground at the time the calibration offsets were estimated and by the characteristics of the vehicle suspension system. Unless the vehicle suspension system is modeled and the ground grade and bank are known and available to the calibration procedure, this rotational offset will remain unknown, and is equal in magnitude to $\phi_2 - \phi_1$ as shown in Figure 4.1.

An IMU mounted to and aligned to the vehicle body can measure the offset ϕ_1 of the vehicle body with respect to gravity, neglecting IMU error. If both this value and the ground bank angle are known (or assumed to be zero, in the case of a level plane), the roll offset between the vehicle and reference frame on the ground may be used to correct the calibration estimate such that the reference plane is parallel to the vehicle body. The pitch offset between vehicle body and reference frame may similarly be corrected with the IMU-measured pitch and ground grade angle. For certain applications, such as those requiring ground plane detection and removal from LiDAR data, it may be useful to correct the LiDAR data such that the ground plane is always approximately level in the LiDAR perspective. To do this, continuous estimation of the dynamic roll and pitch offsets from the LiDAR to the instantaneous ground plane may be calculated at every frame and used to correct the LiDAR data for roll and pitch instead of a static roll and pitch calibration correction.

4.2 Ground Plane Orientation Detection Background

Though not as common as other object and feature detection and mapping, there are several examples of road- or ground-orientation detection from LiDAR data, both within vehicle and mobile robot applications as well as other fields such as surveying and road analysis. These approaches generally involve two main steps. First, the LiDAR data is sectioned such that the points on the plane of interest are isolated from the rest of the scan, and second, the isolated plane points are characterized to yield either a plane equation or immediately parsed into planar orientation.

4.2.1 Plane Characterization Approaches

Many of the approaches to ground- or road-orientation estimation from LiDAR data leverage characterization of a set of points into a plane equation [26-29]. These approaches rely upon the estimation of plane-equation parameters or normal vectors that describe a best-fit plane given a set of ground points. For example, in [27], the parameters a, b, and d to $ax + by + z + d = 0$ are solved for given a series of points (x, y, z). From this equation, a, b, and c are used to compose the "directional" or normal vector of the underlying plane. In [29], least squares as a method for determining plane parameters is avoided in favor of RANSAC plane finding for its outlier-rejection properties.

Similarly, an approach using a regression-based plane fit was used by Zhang and Frey in [28]. The intent of the approach in [28] was to determine road grade and bank from aerial LiDAR scans, with the intention of characterizing vehicle emissions with relation to parameters such as road grade. Prior to fitting a parameterized plane to the ground points, the LiDAR data was rotated according to the aerial vehicle's orientation as reported by the on-board IMU, in an attempt to align the LiDAR data with a global inertial frame. The error of the LiDAR data used in this paper was reported as having range error of 7.7 to 25 cm. The density or number of points used in each plane fit regression instance was not reported. This approach reported road grade errors as much as 3% grade or 1.71 degrees during dynamic sections where the road grade was changing, and less than 0.1% grade or 0.057 degrees where the grade was steady, when compared to NCDOT road grade survey data. In [28], Zhang and Frey utilized a liner regression fit to find the parameters a, b, and c of a plane equation $z = ax + by + c$ given a set of points (x, y, z). Here, z is a set of LiDAR road point elevations, x is the corresponding set of distances along the centerline of the road segment, and y is the corresponding set of distances from the centerline of the road segment.

The coefficients a, b, and c are plane equation parameters. It is claimed within the paper that the plane-equation coefficients a and b are the road grade and road bank, and c is a constant with distance measurement units consistent with the units used for the x, y, and z values respectively.

Similar to the approach outlined in this thesis, ground plane characterization was used to extrinsically calibrate a LiDAR to a vehicle reference frame in [26]. To solve for a yaw calibration estimate, optimal alignment between local grid maps, or map matching, was employed. Following the yaw calibration, a least squares approach was used to directly calculate the normal vector of a plane underlying a set of mean-centered (x, y, z) points. In [26], a matrix U is composed of the x, y, z points offset by the mean of the points. The least-squares decomposition of the matrix yields a set of eigenvectors and eigenvalues. From these results, the eigenvector corresponding to the smallest eigenvalue is used as the normal vector indicating the orientation of the plane underlying the set of points. Following plane normal estimation, the roll and pitch are estimated by projecting the normal vector onto theoretical level y-z and x-z planes. In [26], the authors also suggest a real-time ground points identification method which identifies sets of high-likelihood ground points, and is robust to outliers and non-planar objects [50].

3.2.2 Non-parametric Orientation Estimation

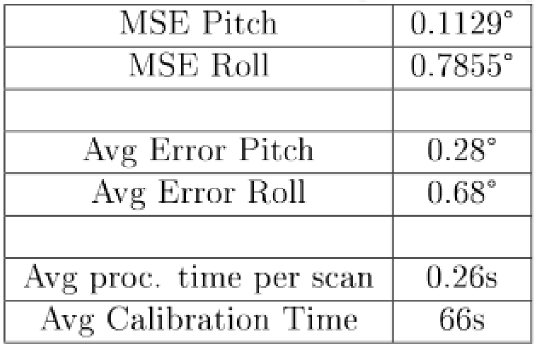
Examples of ground orientation estimation that doesn't employ explicit parameterization of a plane given a set of points are outlined by J. Britt in [10]. Britt introduced two different partial calibration methods involving ground plane characterization. The first of the proposed calibration approaches solved for the relative offset between the sensor and the ground in terms of pitch and height through singular points selected along the x-axis of the LiDAR across multiple scan lines using a rotationally scanning 3D LiDAR. The second solves for the relative roll and pitch between the sensor and ground over a series of 3 paired ground points from within a region of interest, and subsequently solves for the yaw orientation of the sensor relative to the vehicle given a pure pitch excitation when stationary. In both proposed calibration approaches, the estimated offsets were then used to calibrate the LiDAR data to the reference frame. The reference frame used in [10] was implicitly defined by the methods as a having an x-y plane parallel to the ground at the starting position of the vehicle and the y-axis aligned with the axis about which the rotational excitation was “pure,” presumably the y-axis of the vehicle.

In the first calibration approach in [10], road pitch and height with respect to the sensor was found from two selected, non-co-channel points in a LiDAR scan. This approach assumes that roll and yaw of the sensor with respect to the road surface and trajectory are negligible, and recommends that one-hundred frames of the LiDAR data be averaged together to reduce error due to noise in the data. Further, it is designed to work with mechanically scanning LiDAR only, though the approach has potential to be extended to LiDAR data that does not have an inherent channel-based structure. Relatively simple trigonometry is used to solve for first the pitch and then the height from selected target points in the averaged scans.

In a calibration approach also outlined in [10], first the roll, then pitch, and then the yaw of a LiDAR on a vehicle are determined from selected ground plane points, assuming that the ground plane is level during calibration. The yaw be determined when the vehicle is known to be experiencing a pure pitch rotation using the amount of false roll angle that is detected during this maneuver, assuming that the sensor is roll- and pitch-calibrated. A roll, pitch, yaw Euler rotation sequence is solved over a single scan using 3 selected target points in two trigonometric equations. Further, more involved trigonometric equations are used to determine the vehicle (sensor) yaw and pitch from the ground post-calibration, assuming that the ground plane is no longer level. These equations are lengthy, but equate to the removal of the calibration offsets from the above calculated values when applied post-calibration using the calibration rotations and three selected point measurements.

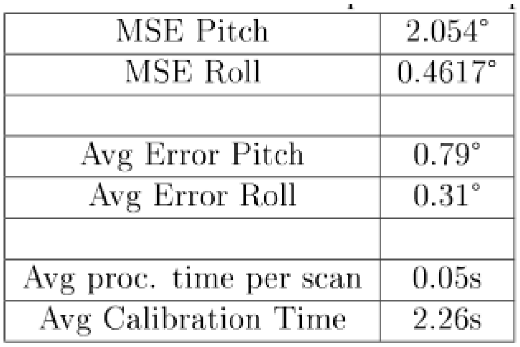
The data in [10] was processed to determine the pitch and roll for single, unaveraged scans, by comparing the roll and pitch calculations from n sets of 3 points selected from a set of m points from the scan. For a static test, it was reported that $m \cong 8$, and for the dynamic test, m is anywhere from about 4 to about 8, as the range of angles from which the points were selected was halved in size from the static test, but the approximate number of points within this range was not reported as before. The accuracies of this approach as found over several scans are given in Table 4.1 and Table 4.2. For the approach proposed by Britt, it was assumed that all selected points belonged to a planar ground.

Table 4.1: Error When Vehicle Is Static. Table 5.1 from [10]



MSE Pitch	0.1129°
MSE Roll	0.7855°
Avg Error Pitch	0.28°
Avg Error Roll	0.68°
Avg proc. time per scan	0.26s
Avg Calibration Time	66s

Table 4.2: Error When Vehicle is Dynamic. Table 5.2 from [10]



MSE Pitch	2.054°
MSE Roll	0.4617°
Avg Error Pitch	0.79°
Avg Error Roll	0.31°
Avg proc. time per scan	0.05s
Avg Calibration Time	2.26s

3.2.3 Considerations for the Proposed Ground Plane Orientation Detection Approach

A plane-fit approach as was popular in many of the other approaches is expected to offer an efficiency increase over the approach from [10] when calculating roll and pitch of a plane for calibration purposes, as it achieves an averaging effect to reduce measurement noise by performing one regression instance over a cluster of points, rather than by iterating the same calculations several times using different point selections.

4.3 Assumptions Required for the Proposed Roll and Pitch Calibration Method

The estimation of the roll and pitch offsets from the ground plane in front of the vehicle using the method developed in this thesis require the following assumptions:

1. The ground under and in front of the vehicle are approximated by a single infinite, level plane
2. The vehicle and ground orientations are static (vehicle is at-rest)
3. The ground is visible to the LiDAR
4. The yaw calibration offset of the LiDAR to the vehicle frame has been determined

The first assumption, that the ground in front of and that underneath the vehicle are essentially the same infinite plane, makes it so that the relative orientation estimates for the ground in front of the vehicle as visible to the LiDAR closely approximates the relative orientations present between the vehicle body and the ground directly underneath it. This allows the ground-in-front to stand in for the reference frame defined as a projection of the vehicle frame onto the ground. The assumption that this effectively infinite plane is level indicates that the ground has zero-valued grade and bank angles with respect to gravity. This condition is not strictly necessary, but simplifies the correction of the ground reference plane to be parallel to the vehicle body to require only the presence of a well-aligned IMU or INS on the vehicle.

The second assumption, that the vehicle be at-rest, arises from the dynamic nature of the vehicle-to-ground orientations when the vehicle is in motion due to the non-rigid suspension and tire elements separating them. The suspension state when the vehicle is at-rest is assumed to be repeatable, providing a more meaningful reference frame even in the case that a vehicle-to-ground offset is not known independently of the LiDAR-to-ground offset and the reference frame cannot be fully aligned to the vehicle body.

The third assumption simply ensures that the LiDAR is capable of observing the chosen comparison geometry object, the ground surface, and therefore has the necessary input data to perform the pitch and roll calibration algorithm. If no ground is visible, the calibration simply cannot be completed. The fourth assumption makes it likely that the ground in front of the LiDAR will be a presumably flat drivable surface, as this surface is likely to be in front of the vehicle as it prepares to drive. If the LiDAR is not yaw-calibrated and is mounted with a significant yaw offset such that its forward direction views the environment to the side of the vehicle, it is likely that non-planar surfaces such as unpaved roadside, foliage, or various objects may be directly in front of the LiDAR where it is expecting the clear ground view.

4.4 Details of the Proposed Roll and Pitch Calibration Method

In this calibration approach, roll and pitch calibration components are estimated by detecting the planar orientation of the ground surface just in front of the vehicle. This step utilizes a static window of interest to isolate the ground points, and so assumes that the surface within the designated region 1) is planar and 2) represents the planar surface the vehicle is currently on top of. These two assumptions hold well for most road surfaces within a short distance of the front of the vehicle. To ensure that the window of interest captures the road in front of the vehicle, the yaw calibration of the LiDAR must first be estimated and applied such that the calibrated forward direction of the LiDAR point cloud corresponds to the forward direction of the vehicle. Once the planar orientation of the road surface in front of the vehicle has been estimated, the LiDAR-to-Ground roll and pitch of the plane are used as the roll and pitch calibration offsets of the vehicle to the ground-projected vehicle reference frame.

Visualizations of one example configuration for the roll, pitch, and height offsets of the LiDAR to the reference frame are given in Figure 4.2 and Figure 4.3. It should be noted that this algorithm does not require that the LiDAR be mounted on the hood of a vehicle, only that the ground in front of the vehicle be visible to the LiDAR. It should also be noted that these diagrams most explicitly display instantaneous orientation angles, or the roll and pitch angular offsets of the LiDAR plane normal from the y-z and x-z planes of the reference frame respectively. Though this definition of orientation offset may be desirable in some applications, it is more common to solve for calibration offsets in terms of Euler angles, which may be translated to or from a rotation matrix or quaternion. This allows the direct application of the solved offsets as a rotational transform to align the sensor frame with the reference frame. As such, the roll angle may be first solved for, the ground plane normal as seen by the LiDAR then rotated by the inverse of the solved roll, and then the pitch solved for from the rotated ground plane normal. This yields the inverse of the roll and pitch as compatible with a roll, pitch, yaw sequence from LiDAR to ground plane (reference frame), or in yaw, pitch, roll when inverted to transform from reference to LiDAR.

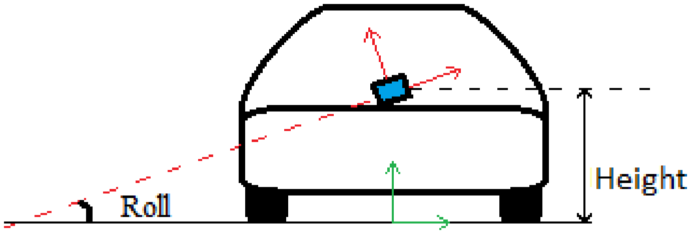


Figure 4.2: Visualization of an example roll and height calibration offset

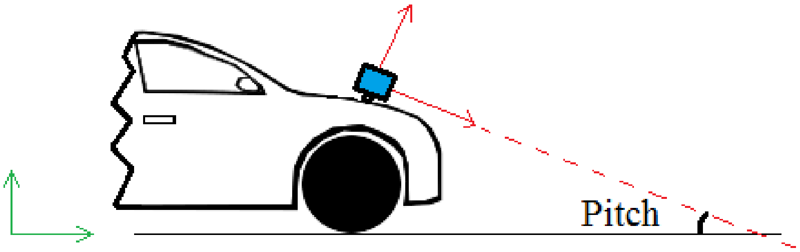


Figure 4.3: Visualization of an example pitch orientation offset

The planar orientation estimation approach begins with a pass-through-filter selection of points within a region-of-interest from the yaw-calibrated LiDAR point cloud. Currently, the spatial bounds of this pass-through filter are static and pre-determined by the user of the algorithm to bound a region-of-interest that is likely to be planar, reasonably close to the vehicle, often free of non-planar obstacles and features. For this thesis, the bounds varied depending on the test scenario, and are specified in the appropriate experimental sections. An example pass-through filter is shown in Equation (3.1), where z and intensity values were bounded. For this algorithm, the primary bounds are applied to x and y, though z and intensity bounds may be added at the user's discretion to generate Equation (4.2).

$$P_i' = P_i\begin{pmatrix} x_{low} \leq x_j \leq x_{hi} \\ y_{low} \leq y_j \leq y_{hi} \\ z_{low} \leq z_j \leq z_{hi} \\ I_{low} \leq I_j \leq I_{hi} \end{pmatrix}, j = 0, \dots, m \tag{4.1}$$

This window of interest should be designed to be relatively square and to remain within the surface of the roadway, and on a multi-lane road, to remain within the ego lane of the vehicle. The larger the window of interest is in the x direction, generally, the less noisy and more accurate the roll result is, and similarly the pitch result is a function of the size of the window in the y-direction. The accuracy and spread of the orientation estimates are also dependent on the number of points within the window of interest, increasing in accuracy as the number of points is increased.

A standard-form three-dimensional plane equation is shown in Equation (4.2)

$$a_i^0 x_j + b_i^0 y_j + c_i^0 z_j = d_i^0 \tag{4.2}$$

where x_j, y_j, z_j represent points on the plane, and $j = 0 \dots m$ where *m* is the number of points used in the plane fit. A re-arranged matrix format of (4.2) designed to be solvable with a least squares approach is shown in Equation (4.3).

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_m & y_m & z_m \end{bmatrix}_i \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{4.3}$$

The new *a*, *b*, and *c* parameters are equivalent to the original line parameters a^0, b^0, and c^0 divided by the original d^0. Similar plane fit approaches to estimating relative ground orientation angles can be seen in [26, 29].

Using least squares to solve the matrix line equation as shown in Equation (4.3), alternate forms of the coefficients a^0, b^0, and c^0 of a plane equation as shown in Equation (4.2) are solved for over the filtered set of ground points. Mapping Equation (4.3) onto Equation (2.16) yields $\boldsymbol{A} = \boldsymbol{P_i}$ (without intensity values included), $\boldsymbol{b} = \mathbf{1},$ and **x** containing the unknown plane parameters for the frame $(a, b, c)_{\boldsymbol{i}}$. As discussed in the background on Least Squares Solutions (Section 2.6), the SVD of the matrix **A** is again here used here to solve the least squares problem. In the case posited in this section, the desired solution **x** has fewer columns (3x1) than **A** and **b** have rows (*m*x3 and *m*x1 respectively), so long as the point matrix **A** is composed from more than one point $\mathbf{p_j}$. As such, the linear matrix equation shown in Equation (4.3) is overdetermined, and the least squares estimation of **x** is then given previously in Equation (2.20) as $\widetilde{\boldsymbol{x}}$.

The components of $\widetilde{\boldsymbol{x}}$, or the alternate-form parameters a, b, and c, represent the normal vector of the underlying plane for the frame of points used to solve for the equations. This normal vector form is shown in Equation (4.4),

$$\overrightarrow{n_i} = [a_i \quad b_i \quad c_i] \tag{4.4}$$

where *a* represents the x component, *b* the y component, and *c* the z component. This normal vector can be used to directly solve for the roll and pitch of the plane in the sensor coordinate system. For convention, the angles are solved in YPR order, and represent the rotation between the LiDAR body and the ground plane, agnostic of the undefined planar yaw. To solve for pitch and roll angles in YPR from global to local Euler order, the normal vector must be rotated about the x axis by the inverse of the roll solved for in Equation (4.5).

$$\phi_i = -atan\left(\frac{b_i}{c_i}\right) \tag{4.5}$$

This is rotation is achieved in Equation (4.6),

$$\begin{bmatrix} a_i' \\ b_i' \\ c_i' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_i) & -\sin(\phi_i) \\ 0 & \sin(\phi_i) & \cos(\phi_i) \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} \tag{4.6}$$

where the resultant rotated normal vector represented by a', b', and c'. This rotated normal vector is used to solve for the pitch orientation. ϕ_i and θ_i are roll and pitch angles, respectively, between the sensor reference frame and the detected ground plane in Equation (4.7).

$$\theta_i = atan\left(\frac{a_i'}{c_i'}\right) \tag{4.7}$$

Equations (4.6-4.7) can be simplified to yield Equation (4.8).

$$\theta = \operatorname{atan}\left(\frac{a}{-\mathrm{b}\sin(\phi) + c\cos(\phi)}\right) \tag{4.8}$$

In the case when a few non-planar (noise or object) points are captured within the window of interest and subjected to the plane fit, the resulting plane orientation will be skewed by the outlier points. To handle this situation and make the approach more robust to such cases, iterative plane fit with outlier removal may be utilized. To perform the outlier removal, the plane fit is first conducted on the original points within the window of interest. Following the fit, the distance from each input point to the fitted plane is calculated and checked against a threshold for inlier-outlier differentiation. If a point is further away from the plane than the threshold allows, it is removed from the point set. After removing all distant points, the plane fit algorithm is repeated on the reduced point set, and the process repeated until either a set minimum number of outlier points were removed in the previous iteration, or a set maximum number of iterations has been executed.

This iterative outlier removal is capable of removing noise and non-planar points so long as the initial skew of the plane fit was reasonably close to the major trend of true plane points. In other words, if the outlier is too far from the true points, or if a large number of outliers weigh the fit away from the true points too far, either noise and non-planar points will not exceed the distance threshold to the original fitted plane, or the true plane points will be themselves removed as outliers. This outlier removal method is capable of removing a small number of non-planar object points and noise, but cannot recover from the intrusion of a densely-sampled non-planar object fully into the window of interest. In the ground orientation data shown throughout this thesis, a maximum of 10 iterative fit intervals were used with a threshold of 0.03 for outlier removal.

An example plane-fit result is shown in Figure 4.4. The LiDAR point cloud can be seen in yellow and orange across the ground of the scene, as well as somewhat on the hood of the truck used as the platform to collect this data. A line drawing of the truck and roof-mounted LiDAR are drawn in to add perspective to the image, and the algorithm-fitted plane is displayed over the *x*, y bounds of the window of interest on the ground in light green.

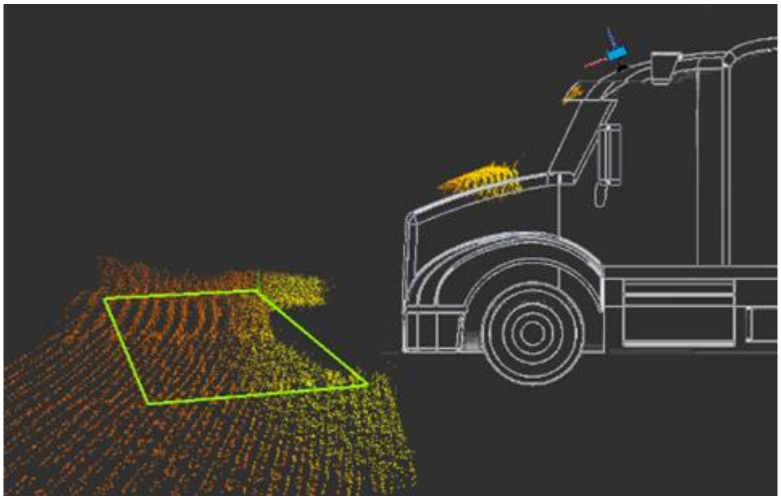


Figure 4.4: Plane fit result (green box) displayed on 3D intensity point cloud, with truck line drawing for mounting and perspective reference

4.5 Experiment and Results

An experiment was conducted to assess the performance of the ground-orientation algorithm as presented in the previous section. In this section, the experimental methods are first outlined, including the equipment and setup used, procedures enacted, and the theory and expected learnings to be gained from the experiment. Mathematical background on post-processing data alignment and details on the calculation of error for this experiment are then given, followed by the experimental results discussion.

4.5.1 Experimental Methods

This experiment tested the ground orientation algorithm for nominal performance. An Innoviz Pro MEMS LiDAR [16] served as the data sensor. This LiDAR has 0.2 degrees horizontal resolution and 0.45 degrees vertical resolution, and scans the environment in a raster pattern over a 73 degree horizontal by 20 degree vertical field of view. An XSENS MTI 200-2A8G4 IMU was used as a comparison sensor, as an IMU is capable of providing orientation measurements for itself or a body it is aligned and mounted to. This IMU was manually aligned and mounted to the top of the LiDAR. For truth data, the orientation outputs of a rigid body tracking feature within a camera-based Optical Motion System (MoCap) was used, where the rigid body being tracked was a marker set consisting of 4 reflective points which was attached to the top of the LiDAR sensor next to the IMU, as seen in the left image of Figure 4.5. The cameras used in the MoCap system were Optitrack Prime 13 cameras [51].

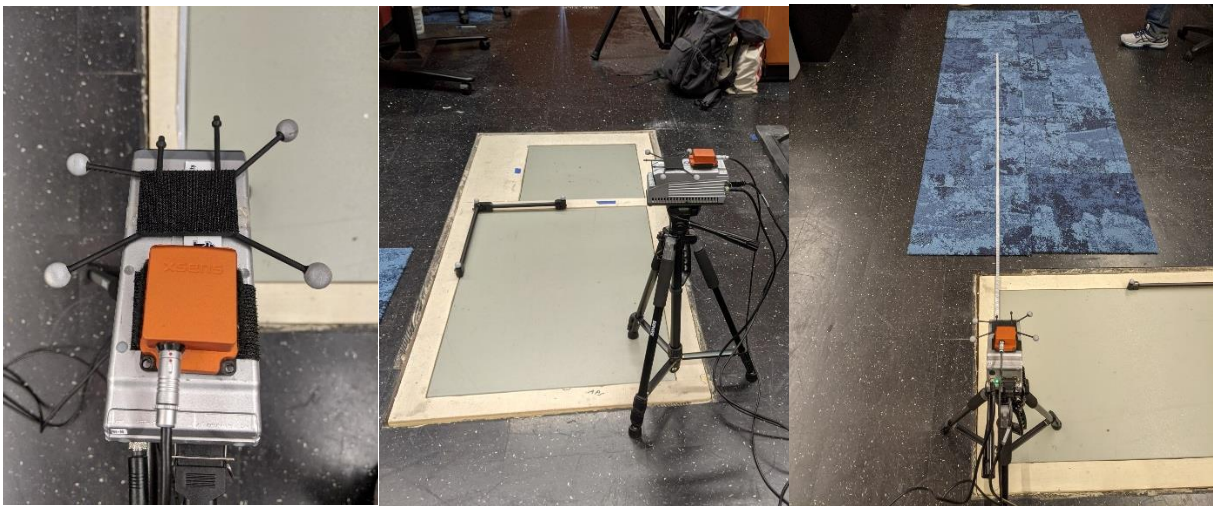


Figure 4.5: Left: Innoviz Pro + XSENS MTI + Reflective Market Rigid Body Co-Mounted, Middle: Adjustable Platform Setup with Reflective Axes for the Motion Capture System, Right: LiDAR-Visible Plane

The LiDAR-IMU-marker assembly was itself mounted on a camera tripod which was capable of being excited in the roll and pitch directions, as seen in the middle image of Figure 4.5. The MoCap reference frame was defined by the L-shaped set of reflectors as seen on the floor in the middle image, where the x-axis points to the rear of the LiDAR, the z-axis points toward the left of the LiDAR, and the y-axis points upward. The LiDAR was placed with an open planar surface within its field of view. The dark floor paneling of the motion capture laboratory were not highly reflective in the IR spectrum utilized by the LiDAR for its measurements. As such, the floor was generally invisible to the LiDAR except at very sharp angles, where the floor would appear as noisy, low-intensity points. This reflectivity issue was resolved by placing carpeting tiles, which were highly visible to the LiDAR, within the area where we intended to characterize the planar orientations.

The LiDAR assembly was then given a series of pitch and roll excitations, with orientation measurements being generated for the motion by the LiDAR algorithm, the IMU, and the MoCap. Each of the different sensors measured the orientation with different references and assumptions. The LiDAR measures the relative pitch and roll between a set of points sampling a portion of the floor in front of the LiDAR, with static bounds on the floor section within the LiDAR frame (and therefore not necessarily static bounds on the floor itself). In this test, bounds of 0 to 3.2 meters in x, -0.6 to 0.2 meters in y, and -10 to 10 meters in z were used. It is assumed that the section of floor points that are characterized into a planar orientation measurement is representative of the floor directly beneath the LiDAR body. If this assumption holds, the orientation measurements generated from the LiDAR data represent the orientation of the LiDAR body relative to the floor beneath it.

The IMU, on the other hand, measures the relative orientation between the IMU body (here mounted to the LiDAR body such that the two bodies move simultaneously) and a theoretical reference plane that is level with respect to gravity (ie. a plane that has a normal vector parallel to the vector of acceleration due to gravity at this location). If the floor underneath the assembly is parallel to this theoretical level plane, and if the IMU frame is well-aligned to the body and frame of the LiDAR, then the measurements of the IMU are expected to be equal to the relative orientations between the LiDAR body and the floor beneath it. While the floor is essentially level from a human perspective, when studying single-digit and sub-degree orientation measurements, it is unlikely that the floor will be exactly parallel to a plane which is level with respect to gravity, due to limitations in construction and manufacturing precision. To account for this, assuming that the orientation offset between the ground plane and the level reference plane is static as the floor is not changing orientation with respect to gravity, a static rotational offset may be approximated and removed to align the IMU measurements to the LiDAR measurements. Further, if the IMU frame is not well-aligned to the LiDAR frame, alignment adjustment approaches conducted in post-processing on the two data sets may be used to account for and remove this assumedly static offset as well.

The motion capture system reports the relative orientation between the rigid reflective marker assembly, mounted to the LiDAR body, and a reference axes established through the use of another rigid reflective marker assembly placed on the floor of the room about a meter away from the LiDAR assembly. The marker assembly attached to the LiDAR was given an origin or zero-degree orientation when the LiDAR body was approximately parallel to the floor of the room. Under certain conditions, the orientation measurements from the MoCap system should equal those from the LiDAR data. These conditions are summarized by the following three assumptions. First, the tracked frame of the marker assembly must be well-aligned to the LiDAR body. Second, the portion of the floor that the MoCap reference frame is defined from must be parallel to the portion of the floor that the LiDAR assembly rests on and the portion of the floor that the LiDAR captures, and finally enough of the cameras comprising the MoCap system must maintain good view of the marker assembly throughout the duration of the test. To adjust for marker frame misalignment to the marker body, additional alignment adjustments may be conducted between the LiDAR data and the MoCap data in post processing.

4.5.2 Post-Processing Alignment Adjustments

There are many approaches that may be applied in attempt to optimally align two sets of data. In the case of alignment between two sets of orientation measurements, this consists of essentially estimating and applying the rotational transform which results in the best alignment between the two data sets. The alignment approach employed for the analysis of the test results discussed here is a solution to the matrix equation as shown in Equation (4.9) [52-54].

$$\boldsymbol{AX} = \boldsymbol{YB} \tag{4.9}$$

Before solving for the alignment, the data in this experiment must be set up so as to fit the **AX**=**YB** form. **A** and **B** are rotation matrices defined from orientation measurements, where **A** is the orientations from the LiDAR to the ground as measured by the LiDAR, and **B** is comprised of either the orientations between the IMU and a theoretical internal reference or the orientations between the marker assembly attached to the LiDAR and the MoCap reference frame. **Y** and **X** are rotational transform matrices relating the frames that **A** is a measurement between to the frames that **B** is a measurement between, and are assumed to be static. Namely, **X** is the transform representing the orientational offsets between the LiDAR frame and either the IMU or the marker array that are mounted on top of the LiDAR body, and **Y** is the transform representing the orientational offsets between either the inertially flat plane referenced by the IMU or the floor-based reference frame established by the MoCap system. The presumed frames for comparing

LiDAR data to data from the IMU can be seen in Figure 4.6, while the frames for comparing LiDAR data to the MoCap data can be seen in Figure 4.7. It should be noted that the **AX=YB** solution is not always constrained to solving for the **X** and **Y** transforms that equate to the diagrams below.

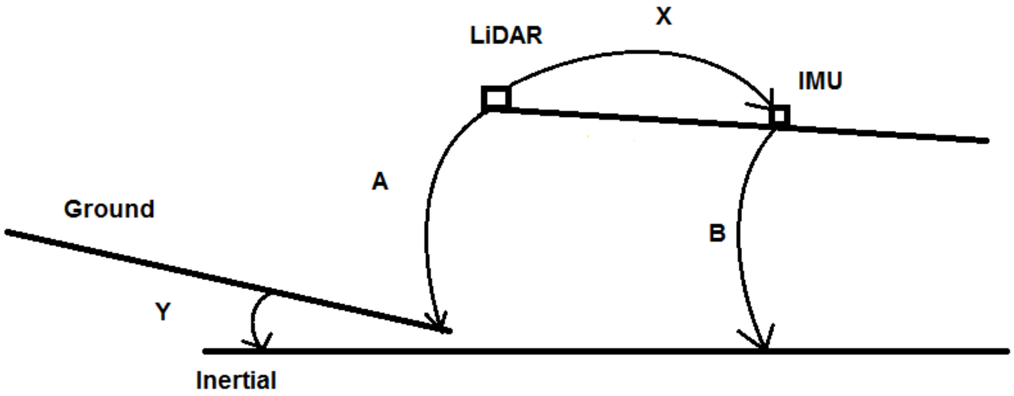


Figure 4.6: AX = YB Frames for a LiDAR and IMU

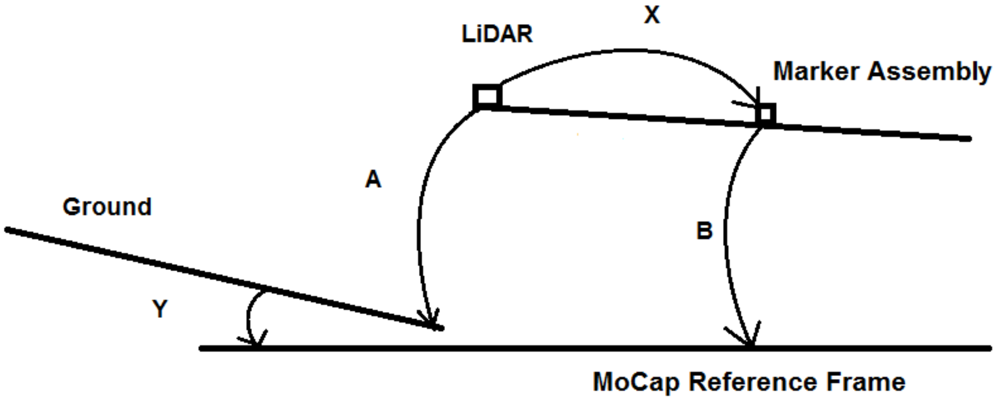


Figure 4.7: AX = YB Frames for the LiDAR and MoCap System (Note: Y is approximately Identity)

In the analysis in this thesis, the closed-form approach from [53] is used to solve Equation (4.9). This begins with a conversion of data sets **A** and **B** into a series of quaternions $\boldsymbol{q_{Ai}}$ and $\boldsymbol{q_{Bi}}$, if not already in quaternion form. For each frame, these data quaternions are used to form matrices **Q$_i$** and **W$_i$** as shown in Equations (4.10-4.11).

$$\boldsymbol{Q_i}(\boldsymbol{q_{Ai}}) = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix} \tag{4.10}$$

$$\boldsymbol{W_i}(\boldsymbol{q_{Bi}}) = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix} \tag{4.11}$$

Each consecutive **Q$_i$** and **W$_i$** are used to iteratively form matrix **C** according to Equation (4.12).

$$\boldsymbol{C_i} = \boldsymbol{C_{i-1}} - \boldsymbol{Q_i'} * W_i, i = 0 \dots n \tag{4.12}$$

The solutions for **X** and **Y** can be determined from the singular value decomposition of matrix **C**, where **C** = **C$_n$**. As detailed in the Least Squares Solutions section of Chapter 2, SVD yields matrices **U**, **Σ**, and **V** according to Equation (2.19). **X** and **Y** are the columns of **U**, **V** corresponding to the smallest member on the diagonal of **s**, similar to a selection of an eigenvector corresponding with smallest eigenvalue in eigenvalue decomposition [54]. Once solved for, the estimated **X** and **Y** transforms may be applied to transform **B** into alignment with **A** according to Equation (4.13).

$$\boldsymbol{B'} = \boldsymbol{YBX'} \cong \boldsymbol{A} \tag{4.13}$$

4.5.3 Error Calculations

The error used for analysis of these experimental results comes in the form of the frame-by-frame rotational difference between the motion capture roll and pitch results and the roll and pitch results from the LiDAR. It is calculated according to the rotational difference between two rotation matrices in Equation (2.6). Using $\boldsymbol{R_{ei}}$ as the rotation matrix representing the error for frame i, $\boldsymbol{R_{ai}}$ as the rotation matrix representing the algorithm data for frame i, and $\boldsymbol{R_{ti}}$ as the rotation matrix representing the truth for this frame, Equation (2.6) can be re-written as Equation (4.14).

$$\boldsymbol{R_{ei}} = \boldsymbol{R_{ai}}\boldsymbol{R_{ti}}' \tag{4.14}$$

The errors **e**$_i$ for roll and pitch can be solved for by decomposing $\boldsymbol{R_{ei}}$ into roll and pitch values as defined in Equation (4.16) (see Appendix A for decomposition into Euler angles).

The error in this experimental analysis is reported and analyzed as root mean square error (RMSE). This is calculated according to Equation (4.15),

$$\boldsymbol{RMSE} = \sqrt{\sum_{i=0}^{n} \boldsymbol{e}_i} = \begin{bmatrix} \theta_{RMSE} \\ \phi_{RMSE} \end{bmatrix} \tag{4.15}$$

where *n* is the number of data elements and is iterates to the number of frames processed, and *e* is the error, or difference, between a truth value and a value estimated by the LiDAR and LiDAR algorithms. In this equation, the square root function in the RMSE is conducted element-wise over the sum vector, and the resulting RMSE result is a vector containing both pitch and roll RMSE results. In this experiment, the error is a vector containing the roll and pitch errors as decomposed from $\boldsymbol{R_{ei}}$ as shown in Equation (4.16).

$$\boldsymbol{e}_i = euler(\boldsymbol{R_{ei}})_{\phi,\theta} = \begin{bmatrix} \theta_{ei} \\ \phi_{ei} \end{bmatrix} \tag{4.16}$$

4.5.4 Experiment Results

The results for the test as described in the previous section are shown in full in Figure 4.8. In these results, it can be seen that the system underwent a series of manual step changes in pitch from about 10 seconds to about 35 seconds, and underwent an approximate step change in roll from about 40 to 65 seconds. While this test was intended to assess the best-case performance of the LiDAR plane detection and orientation algorithm in terms of accuracy, the experiment revealed more about the limitations of the algorithm than it revealed about the best-case performance of the algorithm. Initially, the primary issue with these first full plots involves the manner in which roll excitation was enacted. This action involved tilting the entire assembly about one or two of the tripod legs, and was highly imprecise. This excitation did not properly constrain the motion to a pure roll excitation with respect to the assembly, which can be seen as additional error in the pitch plot during the roll excitation in Figure 4.10, from about 45 seconds to 60 seconds during the roll excitation period. This addition of alignment error during the roll excitation rendered counterproductive the post-processing alignment adjustments conducted on this data, resulting in the nearly constant bias between the LiDAR data and MoCap data during the majority of the test, as seen in the figure. To account for this roll execution error, limited sections of the data were utilized to analyze the pitch excitation and then the roll excitation independently.

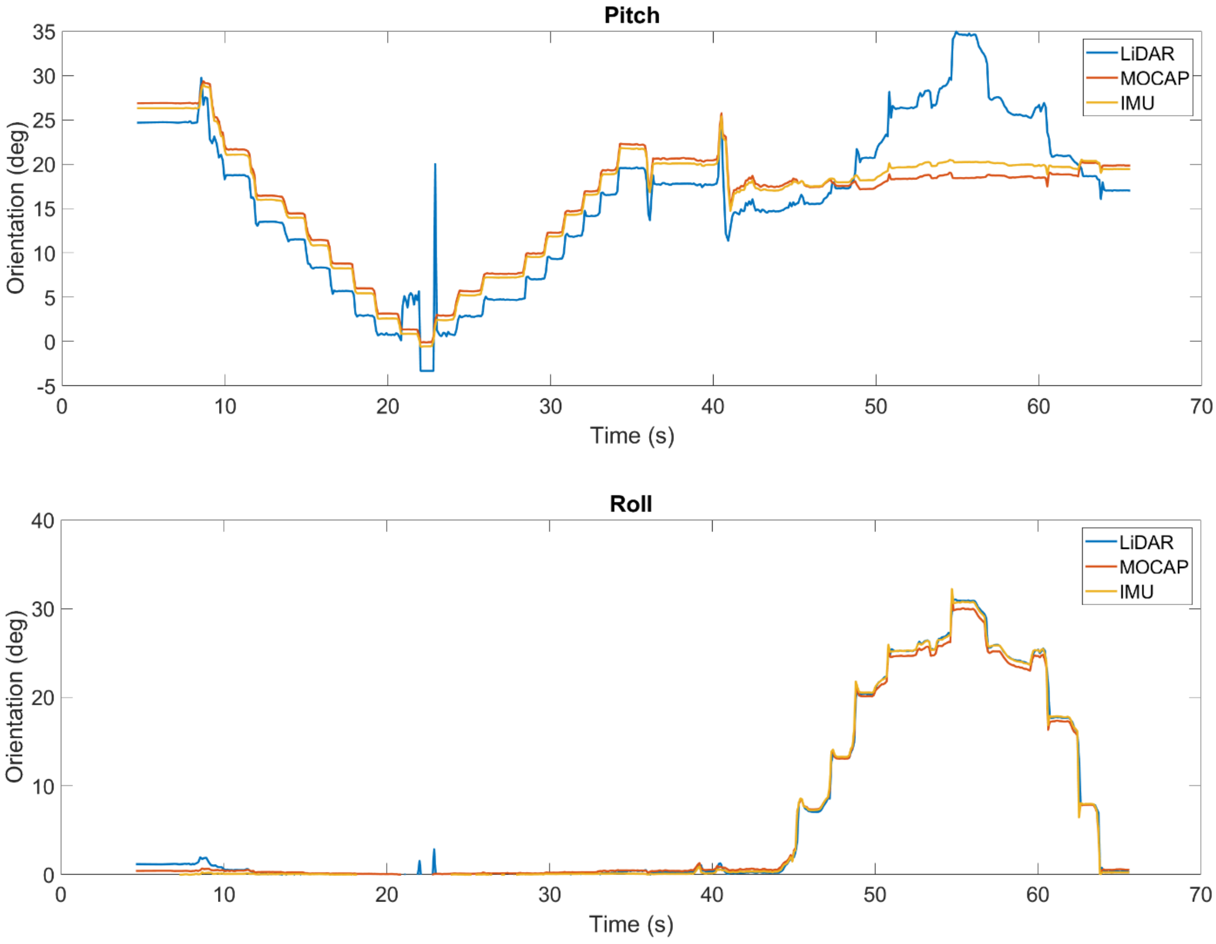


Figure 4.8: Full Ground Orientation Test (Blue – LiDAR Results, Orange – Motion Capture (Truth), Yellow- IMU Data) (Top: Pitch Values, Bottom: Roll Values)

These smaller-section data results with a focus on the pitch and roll excitation periods for the respective pitch and roll measurement plots are shown in Figure 4.9 and in Figure 4.10, respectively. As can be seen when comparing this plot to the full data results, it is clear that the alignment adjustment algorithms were better able to correct for misalignment in the data when some of the more faulty data sections were removed.

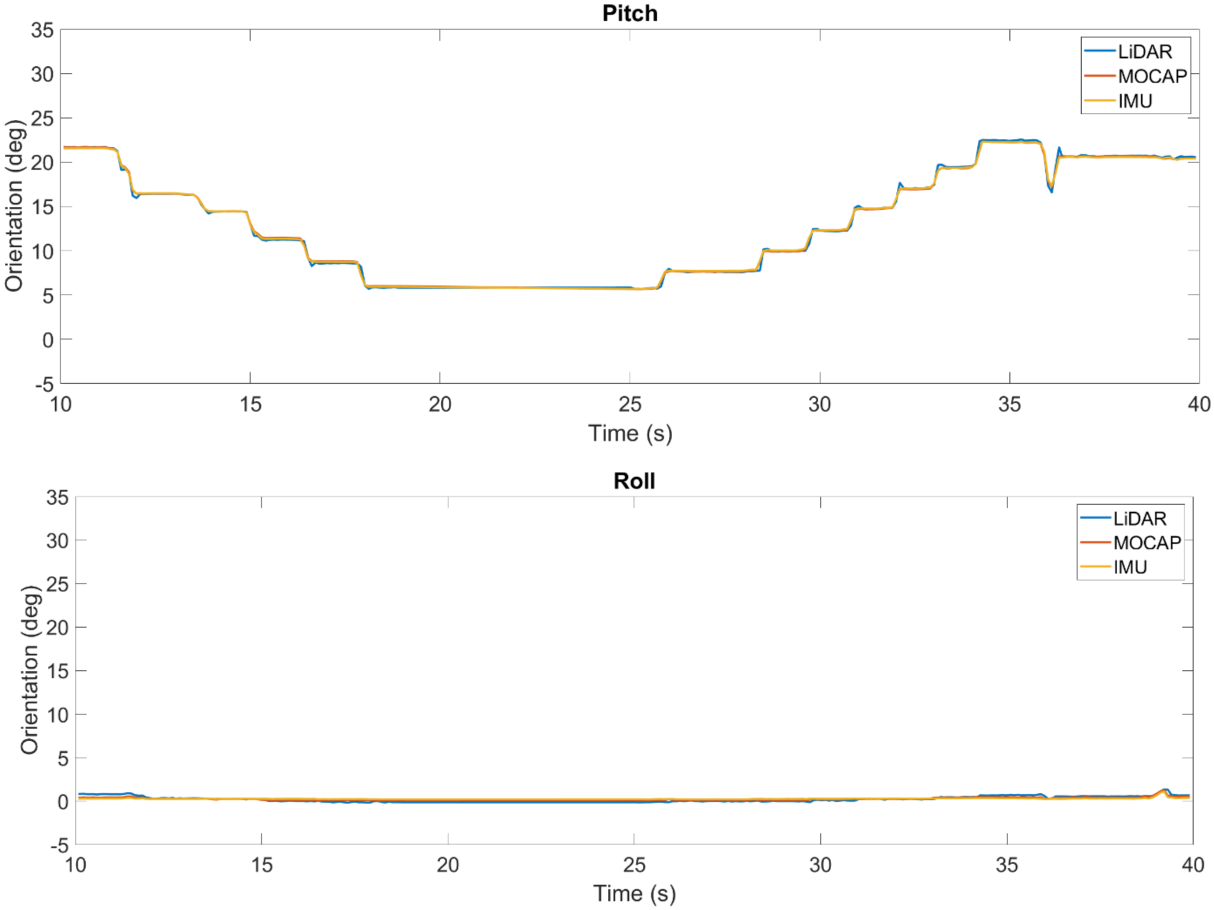


Figure 4.9: Pitch Excitation Period: Ground Orientation Test (Blue – LiDAR Results, Orange – Motion Capture (Truth), Yellow- IMU Data) (Top: Pitch Values, Bottom: Roll Values)

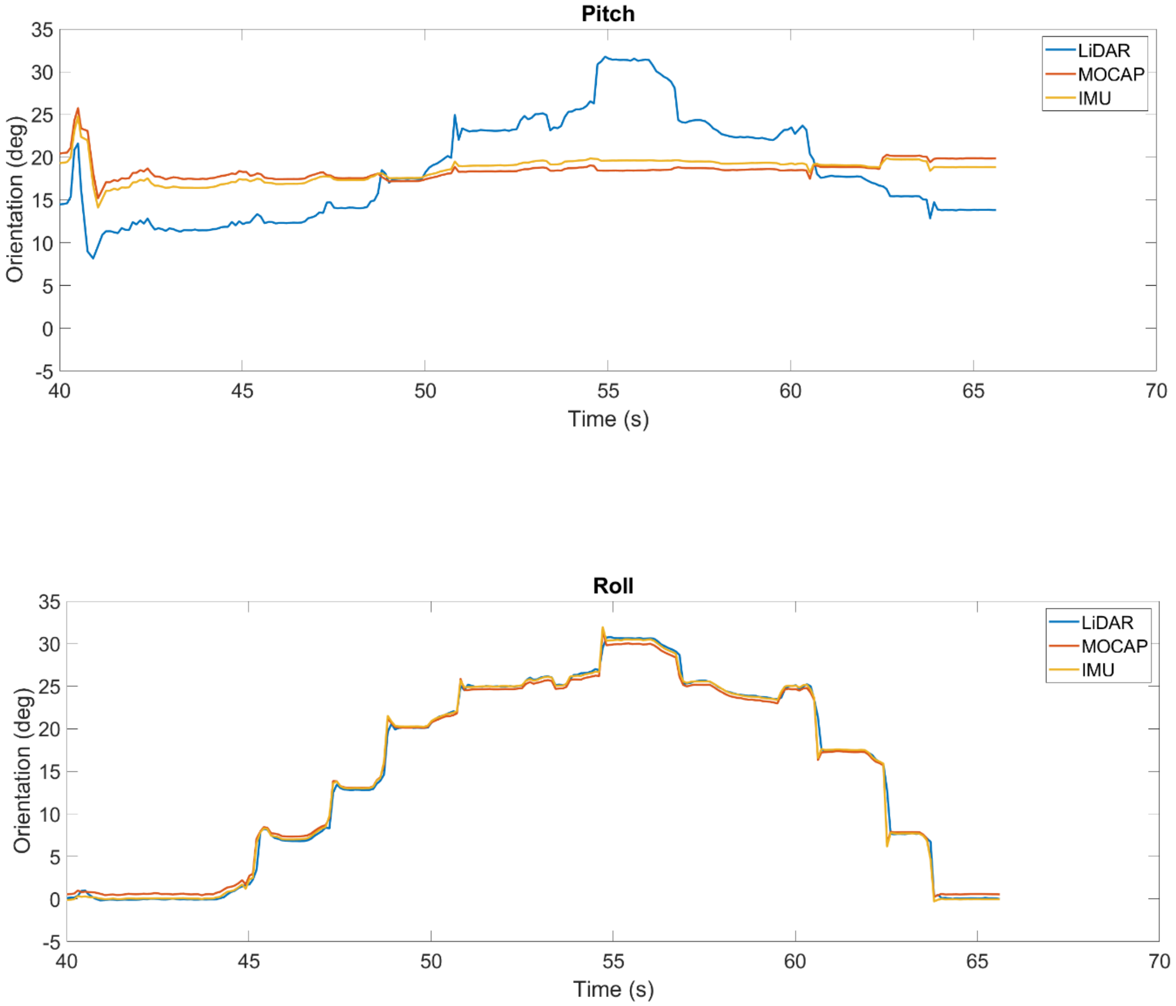


Figure 4.10: Roll Excitation Period: Ground Orientation Test (Blue – LiDAR Results, Orange – Motion Capture (Truth), Yellow- IMU Data) (Top: Pitch, Bottom: Roll)

These better-aligned data sets were able to reveal further challenges to the test methods. For example, the close up plot of the pitch values in Figure 4.11 and Figure 4.12 shows a temporary increase in the error of about 0.5 degrees between LiDAR results and truth at the end of every step angle change. This is most likely due to user error and platform mis-design, as starting and stopping a pitch angle change was erratic and difficult for a human to control. The effort required to start a change in angle compared to the effort to stop one was large, and the step change was achieved over a very short period from a human perspective. This hard-to-control motion may have resulted in jostling and disruption to the system and the LiDAR sensor.

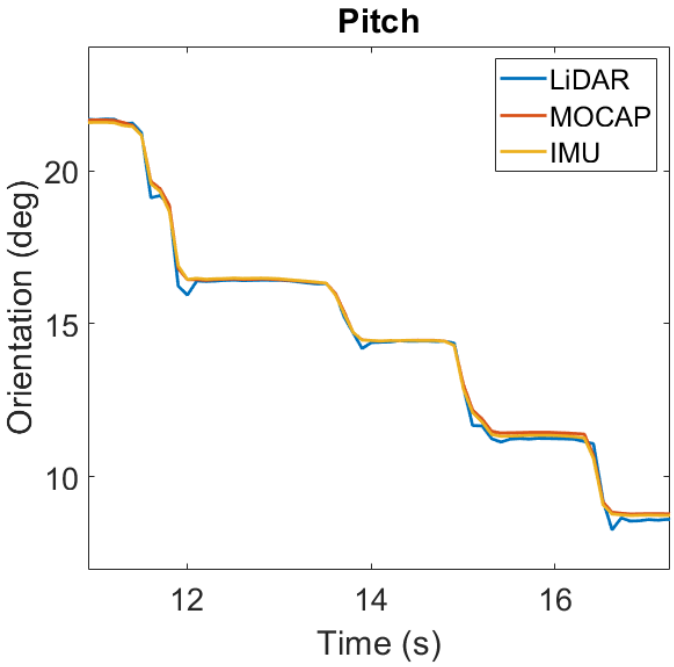


Figure 4.11: Close-up of Pitch Plot at Peak of Dynamic Pitch Motion

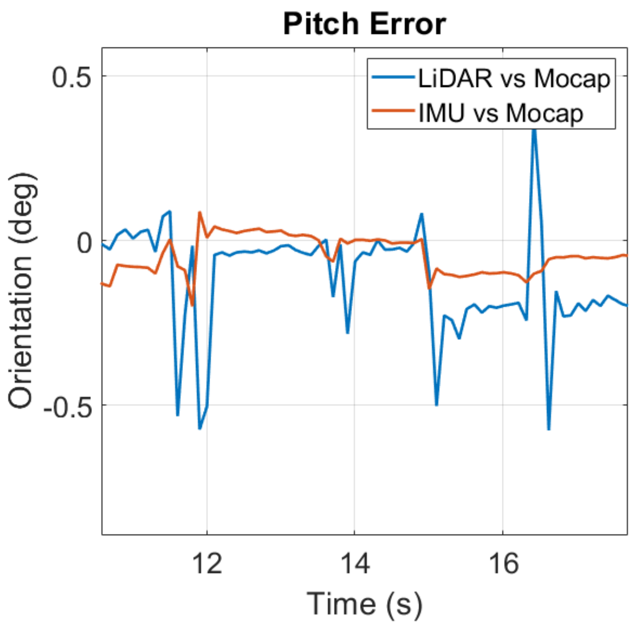


Figure 4.12: Close-up of Error Pitch Plot at Peak of Dynamic Pitch Motion

A further issue with the test setup involved the narrow field of view available to the LiDAR, limiting it to a bounding box of 0.8 meters wide and 3.2 meters long, further exacerbated by sections of the floor which were not reflective in the spectrum used by the LiDAR and thereby were not visible to the LiDAR. Both this narrow bounding box and some of the invisible floor sections can be seen in Figure 4.13. Also in Figure 4.13, another issue is illustrated, which arises from large pitch excitations to the LiDAR. As the LiDAR body orientation becomes more sharp with respect to (or more parallel with) the floor of the room, fewer points on the floor fall within the LiDAR's field of view and the designated bounding box. As the accuracy and noisiness of the plane-fit result are dependent on the density and spread of points available for the plane fit, the accuracy of the LiDAR plane orientation results decreases at pitch angles closer to zero. Further, as the pitch angle of the LiDAR body is altered, the global bounds of the section of the floor from which the roll and pitch measurements are calculated also change. The floor is assumed to be entirely planar, such that at all points on its surface it expresses the same orientation with respect to gravity. However, this may not be the case due to construction defects, settling, and wear. Therefore, as the LiDAR view changes given different angles of excitation, it is likely that the true average orientation of the floor within its sampling space does change as the bounded portion of the floor is altered. The truth orientation does not reflected this change in sampled floor orientation to correspond with the change in the LiDAR data, as the reference plane for the orientation of the assembly for the MoCap measurement is consistent and static.

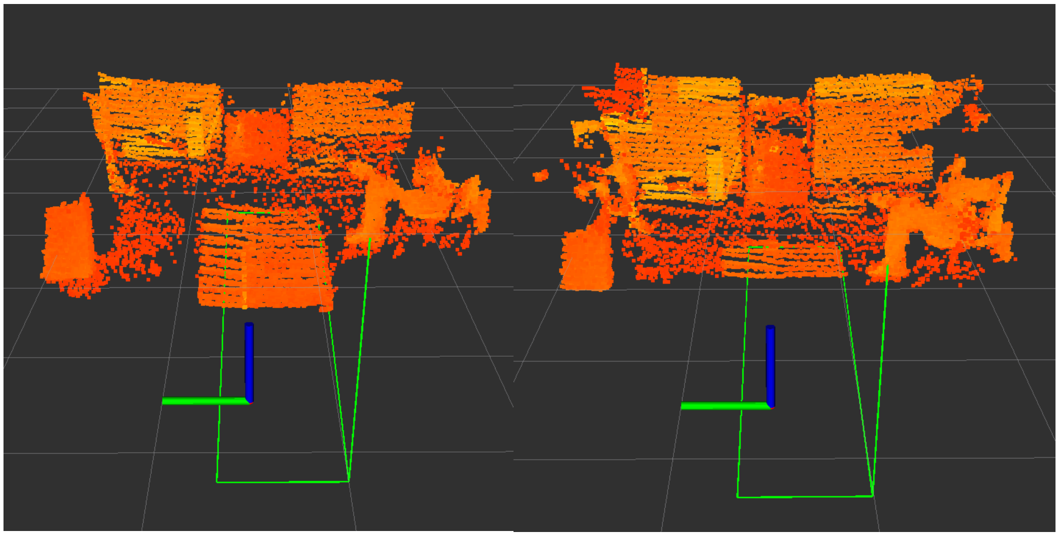


Figure 4.13: Left: LiDAR Data With Green Plane Fit Result, Right: Same view a few frames later into a negative pitch excitation

Table 4.3: RMSE Results for Roll and Pitch during Simple Proof of Ground Orientation Algorithm

Test	Pitch RMSE (deg)	Roll RMSE (deg)
Full	5.04	0.645
Pitch Only	0.229	0.176
Roll Only	5.96	0.784

The RMSE for these three sections of the test are shown in Table 4.3. Overall, from this test, the algorithm may be capable of performing as well as 0.023 and 0.1768 degrees RMSE pitch and roll respectively under less-than-ideal circumstances, as shown during the pitch-only tests. Under severely unideal circumstances, as vital assumptions begin to be invalidated, the algorithm performance may drop to or as poorly as 6.0 or 0.8 degrees RMSE in pitch and roll as seen during the roll-only tests. It has also been revealed that the assessment of the performance of this algorithm is highly dependent on the assumption that the ground is equal in orientation with respect to gravity throughout the useful field of view of the LiDAR throughout the test. Further, as the presumably static transform between the two sets of compared data is shifted, the errors between them after being corrected for alignment will increase dramatically. It is recommended that this ground orientation estimation approach be used in circumstances with a large field of view on the plane of interest. Additionally, if comparing to data from another sensor or other source, the calibration offsets between the two sources should either estimated if static, or continuously estimated if dynamic.

Chapter 5

Proposed Rotational Calibration Estimation Experiments

5.1 Experimental Methods

In this chapter, several tests across multiple scenarios using two vehicle platforms are examined with the intent of assessing the performance of the full proposed calibration approach. The chapter begins with an overview of the goals and methods involved in the experiments, followed by a description of the vehicles and sensors used. This overview of the experimental setup is followed by sections which explain the nature of the truth data used in the studies, methods used to combine data results over several frames, methods for error propagation analysis and variance/covariance quantification, and the methods used to calculate the error statistics for this study. Finally, the chapter presents the experimental results and a brief discussion of the presented data.

5.1.1 Experiment Overview

The accuracy of the calibration estimates was assessed through analysis of the errors of ground orientation estimates as generated from calibrated LiDAR data. This approach was chosen due to limitations arising from a lack of confident truth values for the calibration transforms themselves. In place of direct assessment of the calibration transforms, the accuracy of the calibration may be determined through the analysis of the accuracy of a secondary data result which is dependent on the accuracy of the calibration estimates. Such an approach was seen in [10] as well, where calibration accuracies were analyzed through ground orientation estimates.

To carry out an assessment of calibration in this manner, several elements must be present in the experiment design. First, both a LiDAR sensor capable of viewing a significant planar surface and a truth sensor capable of measuring system orientation values must be available. Secondly, the LiDAR must be rigidly mounted to a system capable of traversing a roadway with distinct edges such that the trajectory of the roadway approximates the trajectory of the system at any given point. This limits the system of choice to be a mobile robot or vehicle. For the experiments conducted in this thesis, two vehicles were chosen: a Lincoln MKZ sedan passenger vehicle, and a Peterbilt 579 tractor-trailer without the trailer attached. These two platforms represent common platforms among autonomous vehicle applications with both commercial and academic interest. The two platforms share many commonalities as far as function and behavior, though the tractor-trailer system is a more complex system of multiple rigid and non-rigid bodies attached by various suspension elements, while the sedan vehicle can be summarized as a two-body system, body and chassis, connected by relatively simple suspension elements.

For both vehicles, a front-facing LiDAR and an inertial sensor, or inertial navigation system (INS) were mounted rigidly to the body or cab. This INS sensor is practically equivalent to an IMU for the purposes of this study. The vehicles were then used to conduct various maneuvers, allowing for the rotational calibration offsets to be estimated for the LiDAR, and subsequently allowing for various roll and pitch dynamics to be recorded by the LiDAR and inertial sensor. Following the experiments, the roll and pitch as observed by the LiDAR ground orientation algorithm was compared to the roll and pitch as observed by the inertial sensor to determine bounds for the calibration accuracy. Further, the calibration results themselves were statistically compared to alternative alignment approach solutions, namely an **AX**=**YB** alignment as described in Chapter 4.

5.1.2 Platform and Sensor Descriptions

As mentioned previously, the first platform was a sedan-style passenger vehicle, a Lincoln MKZ. The sedan vehicle was outfitted with an iXblue Atlans A7 [55] GNSS/INS sensor and accompanying antenna, though only the INS orientations were utilized for this study. The INS sensor was mounted in the trunk of the vehicle to a rigid frame which was itself aligned and rigidly mounted to the floor of the vehicle. This sensor is assumed to be parallel to the vehicle frame. This INS sensor provides the truth orientation data for this platform. In addition to the INS sensor, an Innoviz Pro MEMs LiDAR [16] was mounted to the front hood of the vehicle, at various offset angles and positions throughout the experiments.

During the analysis shown in this thesis, the LiDAR was initially closely aligned with respect to the vehicle body, facing roughly forward and mounted in the center of the car laterally. This scenario is considered to have a relatively small misalignment between the LiDAR and the vehicle frame, and the tests involving this mounting configuration are thus labeled “small misalignment.” In a second scenario involving the sedan vehicle, a more exaggerated mounting offset was created by introducing a significant yaw (still less than 45 degrees) from its original orientation between the LiDAR and the vehicle frame. Accompanying this yaw adjustment were minor changes in the pitch and roll mounting angles due to a shift in the relative slopes on the hood of the car. This mounting scenario is considered to have a “large misalignment” between the LiDAR and vehicle, and is labeled accordingly. The large misalignment scenario with an extra lateral offset (unused in this study) on this vehicle is shown in Figure 5.1.

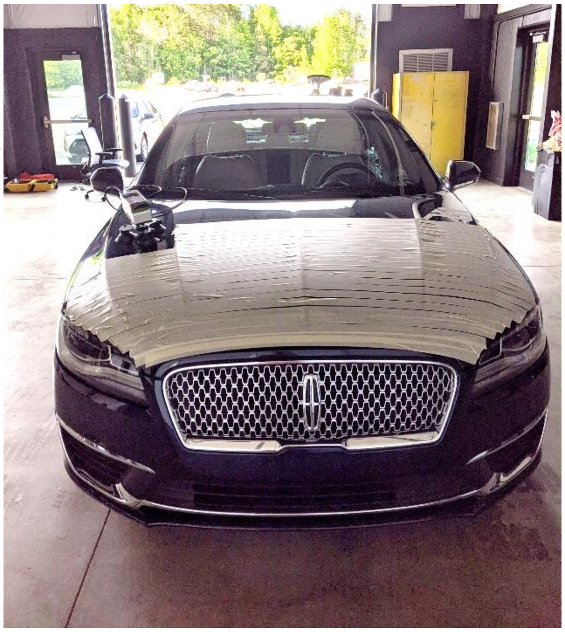


Figure 5.1: LiDAR Mounted on Lincoln MKZ (In “Large Misalignment” Position with Additional Translational Offsets)

For both mounting scenarios, the same maneuvers were conducted using the vehicle. Two of these maneuvers are studied here for orientation accuracy – a stationary roll and pitch excitation, and a figure-eight maneuver. During these maneuvers, the INS data as well as the vehicle-to-road roll and pitch angles from the LiDAR ground orientation algorithm were captured. For both alignment scenarios, region-of-interest bounds for the ground orientation algorithm were set to 3 to 6 meters in x, -1.5 to 1.5 meters in y, and -10 to 10 meters in z. A third scenario, a drive along a clearly marked roadway, was used for the yaw calibration estimation but is not studied in depth here as the INS data used for comparison is not expected to represent truth for the LiDAR orientations along the roadway, where the road grade and slope vary significantly along the path. The stationary excitation and figure-eight maneuvers were conducted on a skid pad, or a relatively flat, paved, and open parking-lot type structure at the National Center for Asphalt Technology

(NCAT) outside of Opelika, Alabama. During the stationary excitation maneuver, the vehicle remained in one place on the skid pad and was given first sinusoidal roll and then sinusoidal pitch excitations through the application of external force on the vehicle body.

The second platform utilized in these experiments was a Peterbilt 579 tractor-trailer tractor. On this vehicle, two high-fidelity proprietary GNSS/INS receiver-antenna pairs were mounted at differing locations. The first of these receivers was mounted to the floor of the cab, aligned to the back wall of the vehicle. The second receiver was mounted to a flat surface on the chassis just behind the cab, and was similarly aligned to the vehicle frame. In addition to the GNSS/INS sensors, piston potentiometers were mounted between the chassis and wheel axles at the front and rear of the vehicle on both sides. Together, these three sensor sets were used to provide truth comparisons for the roll and pitch orientations as measured by the LiDAR algorithm. The Innoviz Pro LiDAR was again used in this platform, this time mounted to the roof of the vehicle and pitched significantly downward so as to capture points on the roadway in front of the vehicle. This mounting and the tractor-trailer platform can be seen in Figure 5.2.

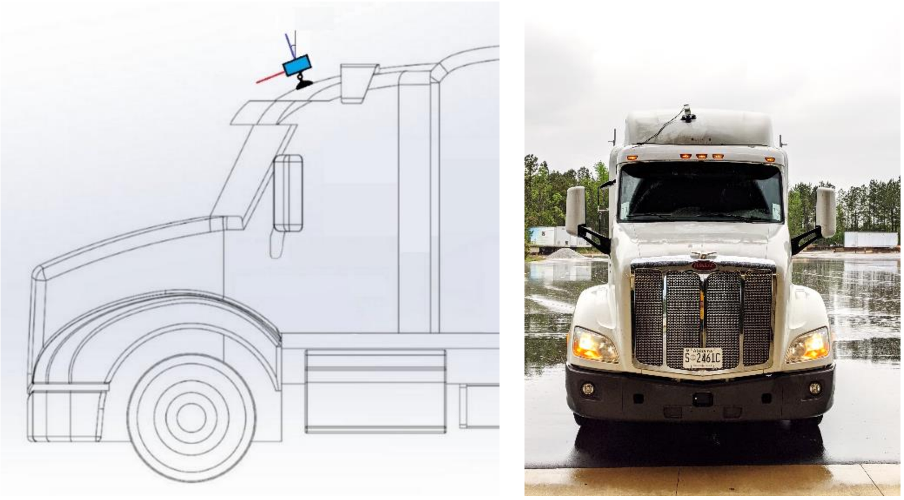


Figure 5.2: LiDAR Mounting on Tractor-Trailer. Left: Mounting Concept Diagram, Right: Actual Mounting

This tractor-trailer platform underwent similar maneuvers to the sedan vehicle, including stationary roll and pitch excitation in the same manner as described before, three laps around the NCAT track as a dynamic test, and driving on the same well-marked roadway as before, Highway 280 outside Opelika, Alabama. During the non-road-drive maneuvers, the INS data as well as the vehicle-to-road roll and pitch angles from the LiDAR ground orientation algorithm were again captured. For the tractor-trailer scenarios, region-of-interest bounds for the ground orientation algorithm were set to 5 to 8 meters in x, -2 to 1 meters in y, and -10 to 10 meters in z, with the same size bounds as the MKZ placed farther from the front of the vehicle. Again, the stationary and track maneuvers were analyzed for calibration and roll and pitch accuracy, but the road drive was only used to estimate the yaw calibration offset.

5.1.3 Truth Data Selection and Generation

For both platforms and all experiments, INS sensors are used to provide truth data. It should be noted here that an INS is an inertial sensor, and as such, it provides measurements to what can be considered an "inertial" global reference frame. This reference frame has a z-axis, or normal vector for its x-y plane, parallel to the direction of acceleration due to gravity. This reference plane can be said to be "theoretically level with respect to gravity", and may be alternatively referred to as the gravity reference plane. Assuming that the INS is aligned to the vehicle frame, this INS will produce measurements for the rotational offsets between the vehicle body and the inertial (gravity) reference frame, ϕ_1 as shown in Figure 4.1.

The LiDAR, in conjunction with the ground orientation algorithm, measures the orientation of the LiDAR with respect to a ground plane. This ground plane is not guaranteed to be level with respect to gravity, and rarely will be. Not only is the ground not often “level,” but it is also not uniformly oriented across all sections of itself. One might be tempted to assume that the ground the vehicle is on can be represented by an infinite plane. In reality, the ground is a non-planar, undulating surface, and at points undulates at a low spatial frequency so as to appear locally level. Note that “level” is used here to refer to parallelness of the planar normal vector to the direction of acceleration due to gravity, while “flat” refers to a situation where an area of ground has very similar or identical orientation to the sections of ground around it, or is at least approximately planar over a significant area.

When the same section of ground is observed over multiple frames, it is assumed that the average inertial orientation of the section, or the ground slopes or road grade and bank, will remain constant throughout the duration of the observation (i.e., the ground orientation at any given location is static with respect to gravity). The error observed between the ground-to-LiDAR measurements and the inertial-to-vehicle measurements is a combination of the mounting offsets between the two sensors and the ground orientation angles with respect to the gravity frame, which are unobservable by the LiDAR algorithm. The mounting offsets are always assumed to be rigid, and when the vehicle is not moving, the LiDAR is assumed to always have in its field of view a static section of ground that does not change its grade and bank angles throughout the test. From these assumptions it follows that the offsets between the LiDAR data and the INS data should be static and easily removable so long as the vehicle does not move across the ground, or is stationary. This allows the INS data to be used as direct truth for the LiDAR roll and pitch during the stationary excitation periods.

It may be assumed that the skid pad structure used in these tests is approximately flat, if not necessarily level. From this it follows that any location on the skid pad should express approximately the same grade and slope as any other location from a global reference perspective. Locally, its grade and slope depend on the observer's local yaw orientation, as this affects the axes about which pitch and roll are measured or experienced. As such, the INS is used as merely approximate truth for the LiDAR data for any maneuvers conducted entirely on the skid pad, such as the figure-eight maneuver conducted by the sedan platform.

For surfaces such as a common roadway or the NCAT test track, it is well-known that the road grade and bank are not constant along most sections of the pathways. For this reason, an alternate truth is needed above simply comparing the INS data to the LiDAR orientation data. For the Peterbilt system, a secondary INS was mounted to the chassis of the vehicle, providing relative orientation between the chassis and theoretical level, while the primary INS was mounted to the cab, the same rigid body the LiDAR was attached to, and thus measured relative orientation between the cab and relative level. Subtracting the chassis orientations from the cab orientations yields a measurement of the orientation between the cab and chassis. Deflection, or suspension, sensing elements were attached between the chassis and the wheel axles. Additionally, the wheels were assumed to be stiff such that the deflection measured by these sensors approximates the offset between the chassis and the roadway. Adding orientations calculated from these suspension deflections to the difference between the cab and chassis INS sensors results in an estimate of the rotational offsets between the cab frame and the roadway, which is congruent to the cab-to-roadway measurements of the LiDAR. As such, the truth for the dynamic track test with the Peterbilt platform was calculated according to the logical arrangement in Equation (5.1).

$$(INS_{cab} - INS_{chassis}) + D \cong Lidar \tag{5.1}$$

LiDAR is simply the roll and pitch rotational offset as estimated from the LiDAR data, between the cab and the road surface. INS_{cab} and $INS_{chassis}$ refer to the cab- and chassis-to-inertial-frame rotational offsets, respectively. D is the rotational offset as calculated from the suspension sensors, which is used to account for the rotational difference between the chassis and the ground plane beneath the vehicle. Using quaternions, the relationship shown in Equation (5.1) is expressed explicitly in Equation (5.2),

$$\boldsymbol{q_{chassis}} \otimes \ (\boldsymbol{q_{cab}})^{*} \otimes \boldsymbol{q_D} = \boldsymbol{q_{Lidar}} \tag{5.2}$$

where $\boldsymbol{q_{chassis}}$, $\boldsymbol{q_{cab}}$, $\boldsymbol{q_D}$, and $\boldsymbol{q_{lidar}}$ refer to the quaternion-style rotational transforms of $\mathrm{INS}_{\mathrm{chassis}}$, $\mathrm{INS}_{\mathrm{cab}}$, D, and LiDAR respectively, $\otimes$ is quaternion multiplication, and * is quaternion conjugation. A process for acquiring roll and pitch angles from suspension deflections can be found in Appendix C.

In an attempt to generate values that approximate direct truth for the calibration estimates, alternate alignment approaches may be utilized. The alignment method of choice here is $\boldsymbol{AX} = \boldsymbol{YB}$ from [52-53], as discussed previously in Section 4.5.2. This approach estimates $\mathbf{X}$ and $\mathbf{Y}$, theoretically equivalent to the LiDAR-to-INS offset and the ground slopes, respectively. When used to rotate a set of orientation measurements to bring them into alignment with another set of orientation measurements, $\mathbf{X}$ and $\mathbf{Y}$ act as a single static transform between the two data sets. This alignment approach effectively accounts for many sources of misalignment between the two data sets, including nonrigidity, drift, value error, noise, and misalignment between various frames, by seeking an optimal reduction of the difference between the two data sets regardless of the source of those differences. It is here hypothesized that the dominant source of misalignment between the LiDAR and INS data tends to be a static calibration offset between the two sensors, and as such the $\boldsymbol{AX} = \boldsymbol{YB}$ alignment solution may be used as an imprecise point of comparison for the calibration estimates.

A single rotational offset is calculated from the **X** and **Y** rotational results as follows. First, the calculated **X** and **Y** are used to rotate the data into approximate alignment with the truth data according to Equation (4.13). Then, a single rotational difference is found between the rotated data and its original form, and this difference, whether a quaternion or rotation matrix, is decomposed into YPR Euler roll and pitch angles. These angles will represent the rotational alignment offsets as estimated with the $\boldsymbol{AX} = \boldsymbol{YB}$ approach.

5.1.4 Multi-frame Statistics

In estimating the calibration values, a separate yaw, pitch, and roll offset is calculated for each frame independently. Due to inherent error in the process and data, a single frame is not enough to confidently determine the most likely or mean offset. As such, some sort of combination of data over multiple frames would allow for noise-filtering. One simple approach is collect a certain number of single-frame offset results, and then batch-average the values to determine the mean offset. This simple batch average is illustrated in Equation (5.3).

$$\vec{r}_{avg} = \sum_{i=0}^{n} \vec{r}_i, \quad \vec{r}_i = \begin{bmatrix} \psi_i \\ \theta_i \\ \phi_i \end{bmatrix}, \qquad i = 0, \dots, n \tag{5.3}$$

The calibration is averaged over n frames, where the vector $\vec{r}_i$ contains the estimated yaw (ψ_i), pitch (θ_i), and roll (ϕ_i) calibration values from each frame i. In implementation, the yaw may be estimated and therefore averaged separately from the pitch and roll without interfering with the algorithm results. For these experiments, the number of averaged frames n was always 10.

To improve the precision of the estimated ground orientation from LiDAR data, a de-noising or smoothing filter was utilized in some of the tests (namely, the dynamic or track tractor-trailer test). This de-noising filter comes in the form of a simple Kalman filter, where the process update step is skipped through use of identity process transition and process covariance matrices. An identity measurement matrix is used to directly propagate roll and pitch estimates from the LiDAR into the filter without alteration, and a non-identity measurement covariance matrix is used to conduct the smoothing operation. The measurement covariance matrix was a diagonal matrix with values manually tuned for each application. More detailed Kalman Filter equations are given in Appendix B.

5.1.5 Covariance Methods

One advantage of a least squares approach is the relative ease with which the covariance of the results may be assessed. Propagation of error from the inputted **A** and **b** matrices (as previously introduced in Equation (2.16)) to the $\hat{\boldsymbol{x}}$ solution matrix can be conducted as follows. A version of Equation (2.16) which acknowledges the presence of a disturbance or error $\boldsymbol{\epsilon}$ is given in Equation (5.4).

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon} \tag{5.4}$$

Given the existence of the overdetermined least squares solution $\hat{\boldsymbol{x}}$ as discussed in Chapter 2, the covariance of $\hat{\boldsymbol{x}}$ can be determined according to

$$cov(\hat{\boldsymbol{x}}\,) = \sigma^2 \boldsymbol{A}^T \boldsymbol{A}^{-1} \tag{5.5}$$

where σ^2 is a scalar value representing the variance of the error defined according to the relationship shown in Equation (5.6) [57].

$$cov(\boldsymbol{\epsilon}) = \sigma^2 * \boldsymbol{I} \tag{5.6}$$

This relationship requires that all of the measurements that compose **A** and **b** have a uniform error as defined by the common error variance σ^2. For any given LiDAR, it is common for the manufacturer to specify range error in a radial direction with respect to the sensor origin, and it is possible for a range within which the vertical and azimuth angle measurements are expected to fall to be provided as well. Rather than propagate the spherical errors into the Cartesian points used to compose **A**, σ^2 may be approximated according to

$$\sigma^2 = (\sigma_{\bar{x}})^2 * n \tag{5.7}$$

where *n* is the sample size or number of fitted points, and $\sigma_{\bar{x}}$ is the standard error of the fitted point set to the fitted plane, assuming that the fitted plane is the true underlying plane. The standard error can be found using

$$\sigma_{\bar{x}} \cong \frac{\sigma_x}{\sqrt{n}} \tag{5.8}$$

where σ_x is the standard deviation of the residuals, **e**, over the *n* points used to compose **A**. The residuals are defined according to Equation (5.9) [57]

$$\boldsymbol{e} = \hat{\boldsymbol{\epsilon}} = \boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{b} \tag{5.9}$$

while the standard deviation can be found using Equation (5.10).

$$\sigma_x = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(e_i - \bar{e})^2} \tag{5.10}$$

When least squares is used to solve for the unknown plane parameters a, b, and c from a set of *n* three-dimensional approximately planar points where *n* is greater than one as done in this thesis, $cov(\boldsymbol{\hat{x}})$ is a 3x3 matrix which specifies the covariance of the estimated plane parameters. Since the plane parameter estimation is in this thesis a sub-step of the ground orientation estimation algorithm, the plane parameter covariance as solved from these least squares relationships must be propagated into covariance for the roll and pitch measurements. For a linear transformation as generalized in

$$\boldsymbol{y} = \boldsymbol{Gx} \tag{5.11}$$

the covariance propagation can be calculated according to Equation (5.12).

$$cov(\boldsymbol{y}) = \boldsymbol{G}\, cov(\boldsymbol{x})\, \boldsymbol{G^T} \tag{5.12}$$

In this thesis, the plane parameters $\boldsymbol{\hat{x}}$ are transformed into roll ϕ and pitch θ according to Equations (4.5, 4.8), as previously discussed. Linearizing these roll and pitch equations to yield a matrix **G** is a non-trivial task. As an alternative, the inverse relationship may be considered. The planar normal, whose x, y, and z components coincide with the plane parameters a, b, and c, may be determined from any given roll and pitch orientation values. The normal vector can be determined by rotating a “level” basis vector by a rotation matrix representing the given roll and pitch, as shown in

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \boldsymbol{R}(\phi, \theta) * \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.13}$$

where $\boldsymbol{R}(\phi, \theta)$ is a YPR rotation matrix as presented in Equation (2.4) with zero-valued yaw. Equation (5.13) can be simplified to Equation (5.14),

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -\sin(\theta) \\ -\sin(\phi)\cos(\theta) \\ \cos(\phi)\cos(\theta) \end{bmatrix} \tag{5.14}$$

which can be more readily linearized than can Equations (4.5, 4.8). Using small-angle approximation, a linearization of Equation (5.14) is shown below.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \cong \begin{bmatrix} -\theta \\ -\phi \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \phi \\ 1 \end{bmatrix} \tag{5.15}$$

Observing that this equation follows the linear transform form shown in Equation (5.11), the relationship between the plane parameter covariance $cov(\hat{\boldsymbol{x}})$ and the covariance of the vector $[\theta \quad \phi \quad 1]^T$ can be approximated by plugging in the matrix values from Equation (5.15) into Equation (5.12), which yields Equation (5.16).

$$cov(\hat{\boldsymbol{x}}) \cong \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\theta^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \tag{5.16}$$

This equation simplifies to

$$cov(\widehat{\boldsymbol{x}}) \cong \begin{bmatrix} \sigma_\theta^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.17}$$

from which an approximate, simplified equality between the variance for the *a* parameter and that of the pitch estimate and the variance for the *b* parameter and that of the roll estimate can be assumed, as shown in Equations (5.18-5.19)

$$\sigma_a^2 \cong \sigma_\theta^2 \tag{5.18}$$

$$\sigma_b^2 \cong \sigma_\phi^2 \tag{5.19}$$

where σ_a^2 is the element in the $cov(\widehat{\boldsymbol{x}})$ matrix at row one column 1, and σ_b^2 is the element in the $cov(\widehat{\boldsymbol{x}})$ matrix at row two, column 2.

5.1.6 Error Calculations

The error between the LiDAR roll and pitch and the INS roll and pitch were calculated in the same manner as was the error in Chapter 4. It takes the form of the frame-by-frame rotational difference between the planes represented by the LiDAR roll and pitch estimates and those represented by the INS roll and pitch measurements.

The error used for analysis of these experimental results comes in the form of the frame-by-frame rotational difference between the INS roll and pitch and the roll and pitch data from the LiDAR. It is calculated as shown previously in Equation (2.6), with $\boldsymbol{R_{ei}}$ as the rotation matrix representing the error for frame i, $\boldsymbol{R_{Li}}$ as the rotation matrix representing the LiDAR data for frame i, and $\boldsymbol{R_{IMUi}}$ as the rotation matrix representing the INS orientations for this frame. Equation (2.6) can thus be re-written as Equation (5.20).

$$\boldsymbol{R_{ei}} = \boldsymbol{R_{Li}}\boldsymbol{R_{IMUi}}' \tag{5.20}$$

See Appendix A for decomposition into Euler angles. The error in this chapter is reported and analyzed as root mean square error (RMSE). This is calculated according to Equation (5.21)(**4.16**),

$$\boldsymbol{RMSE} = \sqrt{\sum_{i=0}^{n} \boldsymbol{e}_i} = \begin{bmatrix} \theta_{RMSE} \\ \phi_{RMSE} \end{bmatrix} \tag{5.21}$$

where *n* is the number of data elements and is iterates to the number of frames processed, and *e* is the error, or difference, between a truth value and a value estimated by the LiDAR and LiDAR algorithms. In this equation, the square root function in the RMSE is conducted element-wise over the sum vector, and the resulting RMSE result is a vector containing both pitch and roll RMSE results.

In this experiment, the error is a vector containing the roll and pitch errors as decomposed from $\boldsymbol{R_{ei}}$ as shown in Equation (5.22).

$$\boldsymbol{e}_i = euler(\boldsymbol{R_{ei}})_{\phi,\theta} = \begin{bmatrix} \theta_{ei} \\ \phi_{ei} \end{bmatrix} \tag{5.22}$$

5.2 Experiment and Results

In this section, the more meaningful plots and data from the experiments are displayed and briefly discussed. The section begins with an overview of the roll and pitch errors as calculated from the LiDAR roll and pitch and the INS roll and pitch over all tests. Next, a closer look at the calibration values estimated for the different tests is given, followed by a more thorough discussion of the roll and pitch results from individual tests, beginning with the sedan “small misalignment” and “large misalignment” scenarios followed by the tractor-trailer scenario. Finally, the section is concluded with a discussion on the validity of the truth data used.

5.2.1 Overview of Results

Table 5.1 provides an overview of the roll and pitch error between the LiDAR and the truth values for the different tests, in the form of RMSE, error range, and error standard deviation. The data can be seen plotted in Figure 5.3 and Figure 5.4. The first six tests all involved stationary excitation, with the platform vehicle sitting still on the skid pad while being externally roll and pitch excited. These RMS errors and standard deviations are smaller for all stationary tests than for the dynamic tests in both roll and pitch. This indicates that the estimated calibration more accurately accounts for the misalignment between the LiDAR and INS data during the static case, and has accuracy bounded by the errors seen here. The calibration is less representative for the dynamic tests, including figure-eight tests for the sedan platform and NCAT track laps for the tractor-trailer. This increase in RMSE and standard deviation during the dynamic tests may indicate an introduction of additional misalignments on top of the calibration offsets during motion, such as a changing ground slope breaking the assumption of congruency between the two data sets. Alternatively, the increase in error could indicate additional inaccuracies in the calibration that express themselves only during dynamic scenarios. The average RMSE, max error, and standard deviation is summarized across the stationary and dynamic tests in Table 5.2, for an average stationary RMSE of 0.070 and 0.069 degrees in roll and pitch, respectively, for the calibrated data, and for the dynamic tests, an average RMSE of 0.956 and 1.77 degrees in roll and pitch. The combined overall results are given in Table 5.3. Here, the combined average RMSE is 0.513 degrees in roll for calibrated data, and 0.922 degrees pitch.

Table 5.1: RMSE, Error Range, and Standard Deviation (Std Dev) for Roll and Pitch Values Estimated from Calibrated LiDAR Data During Different Tests (values in deg)

Test Type			**Raw**			**Auto Calibrated**			**AX=YB**		
			RMSE	**Range**	**Std Dev**	**RMSE**	**Range**	**Std Dev**	**RMSE**	**Range**	**Std Dev**
Stationary	**Sedan: Small Misalignment**	**Roll**	1.31	1.47	0.033	0.035	0.186	0.030	0.033	0.196	0.033
		Pitch	15.9	16.1	0.032	0.037	0.287	0.034	0.031	0.263	0.031
	Sedan: Large Misalignment	**Roll**	2.04	2.55	0.095	0.099	0.488	0.088	0.091	0.481	0.091
		Pitch	13.9	14.1	0.051	0.051	0.388	0.041	0.046	0.397	0.046
	Tractor-Trailer	**Roll**	9.60	10.3	0.121	0.077	0.365	0.077	0.130	0.735	0.130
		Pitch	34.0	34.5	0.141	0.119	0.972	0.115	0.104	0.899	0.104
Dynamic	**Sedan: Small Misalignment**	**Roll**	1.18	2.90	0.744	0.843	1.93	0.746	0.746	2.05	0.746
		Pitch	15.0	17.2	1.31	1.56	3.63	1.206	1.19	2.61	1.19
	Sedan: Large Misalignment	**Roll**	2.93	5.67	1.00	1.20	3.05	0.928	1.00	3.01	1.00
		Pitch	15.2	17.5	1.09	1.65	3.47	1.09	1.07	2.96	1.07
	Tractor-Trailer	**Roll**	9.18	10.4	0.404	0.824	2.32	0.441	0.443	1.81	0.443
		Pitch	35.9	37.7	0.500	2.11	5.50	0.535	0.312	1.67	0.312

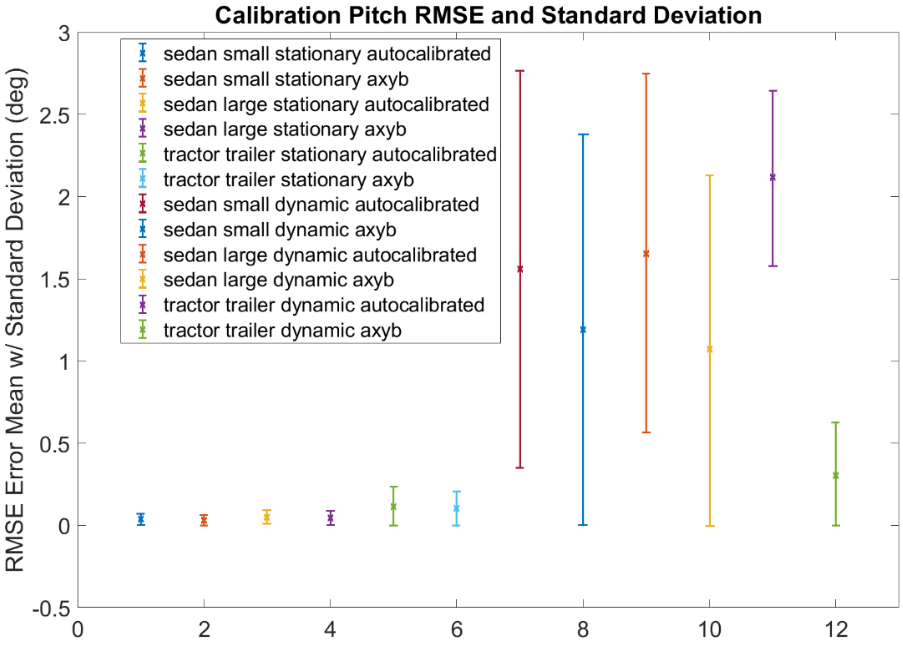


Figure 5.3: RMSE and Standard Deviation Plot for Pitch Estimates (and by extension Calibration Pitch Offsets) for Different Test Scenarios.

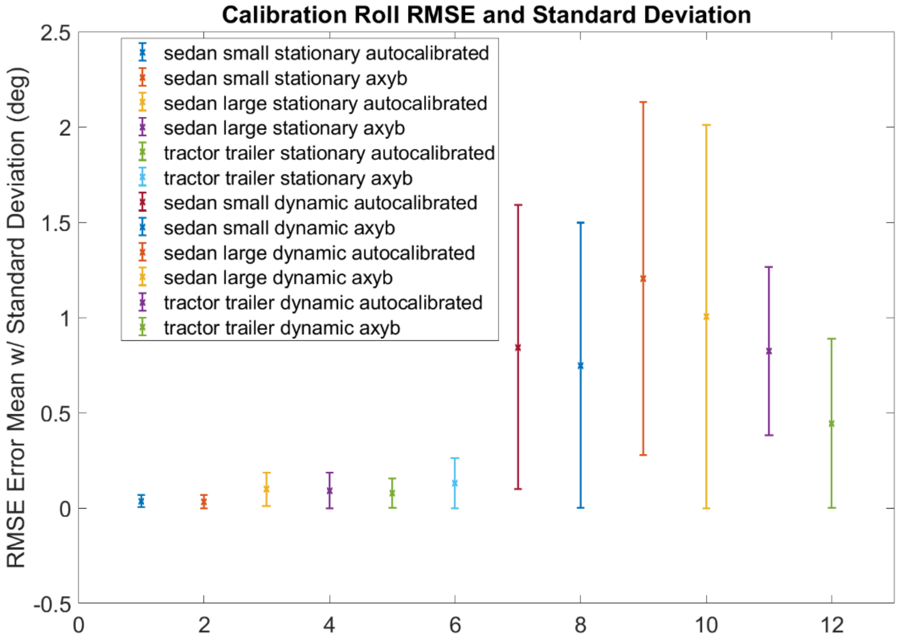


Figure 5.4: RMSE and Standard Deviation Plot for Roll Estimates (and by extension Calibration Roll Offsets) for Different Test Scenarios

Table 5.2: Average RMSE, Max Error, and Standard Deviation for Stationary and Dynamic Tests (values in deg)

		Raw			**Auto Calibrated**			**AX=YB**		
		RMSE	**Range**	**Std Dev**	**RMSE**	**Range**	**Std Dev**	**RMSE**	**Range**	**Std Dev**
Stationary	**Roll**	4.32	4.77	0.083	0.070	0.346	0.065	0.085	0.471	0.085
	Pitch	21.3	21.6	0.075	0.069	0.549	0.063	0.060	0.520	0.060
Dynamic	**Roll**	4.43	6.32	0.716	0.956	2.43	0.705	0.730	2.29	0.730
	Pitch	22.0	24.1	0.967	1.77	4.20	0.944	0.857	2.41	0.857

Table 5.3: Average RMSE, Max Error, and Standard Deviation Overall (values in deg)

	Raw			**Auto Calibrated**			**AX=YB**		
	RMSE	**Range**	**Std Dev**	**RMSE**	**Range**	**Std Dev**	**RMSE**	**Range**	**Std Dev**
Roll	4.37	5.55	0.400	0.513	1.39	0.385	0.407	1.38	0.407
Pitch	21.7	22.9	0.521	0.922	2.37	0.504	0.459	1.45	0.459

These averaged test results may be compared to the state-of-the-art in environmental geometric object calibration using only LiDAR data [10]. Such a comparison is given in

Table 5.4 and Table 5.5. This approach can be seen with a 91.5% decrease in the roll RMSE seen in the state-of-the-art and a decrease of 39% in pitch RMSE. The dynamic test results yielded a 107% increase in RMSE in roll and a 14% decrease in pitch RMSE. Overall, the proposed algorithm displayed a decrease in roll RMSE of 18% in comparison to the state-of-the-art and 15% reduction in pitch RMSE.

Table 5.4: Average RMSE for Stationary and Dynamic Test Scenarios from the Proposed Approach and the State of the Art

		RMSE (deg)	Range (deg)	Std Dev (deg)	Prior Art [10] RMSE (deg)
Stationary	**Roll**	0.070	0.346	0.065	0.786
	Pitch	0.069	0.549	0.063	0.113
Dynamic	**Roll**	0.956	2.43	0.705	0.462
	Pitch	1.77	4.20	0.944	2.05

Table 5.5: Average Overal RMSE from the Proposed Approach and the State of the Art

	RMSE (deg)	Range (deg)	Std Dev (deg)	Prior Art [10] RMSE (deg)
Roll	0.513	1.39	0.385	0.624
Pitch	0.922	2.37	0.504	1.08

5.2.2 Sedan Passenger Vehicle Results

In this section, the tests with the MKZ sedan with LiDAR and high-fidelity GNSS/INS sensor mounted to its body are analyzed. The estimated calibration rotational offsets are given in Table 5.6 along with **AX**=**YB** offsets which approximate the calibrations and offer a point of comparison. It should be noted that there are two values of **AX**=**YB** offsets shown in each respective cell. The first value was calculated to align the uncalibrated LiDAR data to the IMU data in the stationary maneuvers, and the second value was calculated to align the two data sets in the dynamic case. As more non-misalignment errors are present in the dynamic cases, these **AX**=**YB** offset results are presumably less likely to accurately represent a static calibration transform between the two data sources. Across these results, the stationary **AX=YB** offsets have direct magnitude differences with their auto-calibration counterparts for roll and pitch of 0.023 to 0.305 degrees, respectively. It should be noted that the yaw calibrations estimated by the **AX=YB** algorithm are not meaningful here in terms of calibration comparison, as the truth yaw was set to zero so as to not interfere with the alignment of the observed roll and pitch when trying to align to the zero-valued unobserved LiDAR yaw used in the approach. Additionally, the yaw offset is not observable to the **AX=YB** approach given LiDAR orientation estimates in roll and pitch only.

Table 5.6: Sedan Estimated Calibration Offsets

	Small Misalignment		**Large Misalignment**	
	Autocalibration	**AX = YB**	**Autocalibration**	**AX = YB**
Roll (deg)	-1.33	-1.31 or -0.913	-1.73	-2.04 or -2.76
Pitch (deg)	15.9	-15.9 or -15.0	14.0	-13.9 or -15.2
Yaw (deg)	0.769	0.010 or -0.005	-13.7	-0.073 or -0.021

In Table 5.7-5.8, the iterative process of calculating the roll and pitch calibration estimates is shown for the small and large misalignment cases. In each case, the roll and pitch offsets were simultaneously estimated over 4 iterative corrections. In each iteration, the first ten saved frames of available roll and pitch estimates from the ground orientation algorithm were collected and averaged. The resulting average roll and pitch offsets were then used as calibration offsets, and used to correct the saved LiDAR point clouds. The saved and now corrected LiDAR data from the beginning of the stationary tests were then used to re-calculate roll and pitch. The new roll and pitch results were then compared to the target roll and pitch (INS roll and pitch values here), and the error used to update the estimated calibration offsets. This process was repeated for four iterations to yield initial errors of less than 0.01 degree in roll and pitch for the small misalignment case, but 6 iterations were required to reach similarly low levels of initial LiDAR-to-INS errors in the large misalignment case, suggesting that larger misalignments are more difficult to ascertain with this calibration approach and require more data to accurately estimate the calibration offsets. The iterative calibrations and errors can be seen to converge in Figure 5.5-5.6, where the calibrations converge to real values and the errors converge toward zero.

Table 5.7: Small Misalignment Iterative Roll and Pitch Calibration Estimation

	Roll Calibration (deg)	**Pitch Calibration (deg)**	**Roll Error (to target IMU roll) (deg)**	**Pitch Error (to target IMU pitch) (deg)**
Iteration 1	0	0	1.52929655	15.9299456
Iteration 2	-1.5292965	15.9299456	-0.1903234	-0.0261117
Iteration 3	-1.3389731	15.9038339	-0.0068341	-0.0025349
Iteration 4	-1.332139	15.901299	-0.0009855	-0.000214

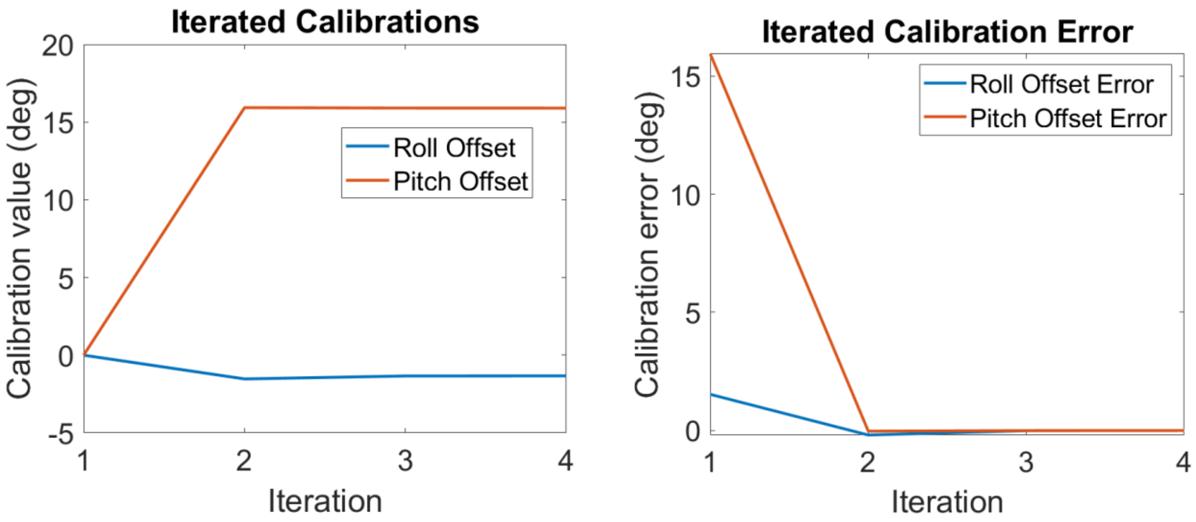


Figure 5.5: Sedan Small Misalignment Iterative Calibration Plots

Table 5.8: Large Misalignment Iterative Roll and Pitch Calibration Estimation

	Roll Calibration (deg)	**Pitch Calibration (deg)**	**Roll Error (to target IMU roll) (deg)**	**Pitch Error (to target IMU pitch) (deg)**
Iteration 1	0	0	-1.72	14.1
Iteration 2	1.72	14.1	3.25	0.775
Iteration 3	-1.53	14.9	0.407	-0.797
Iteration 4	-1.93	14.1	-0.163	-0.114
Iteration 5	-1.77	14.0	-0.038	0.035
Iteration 6	-1.73	14.0	0.006	0.010

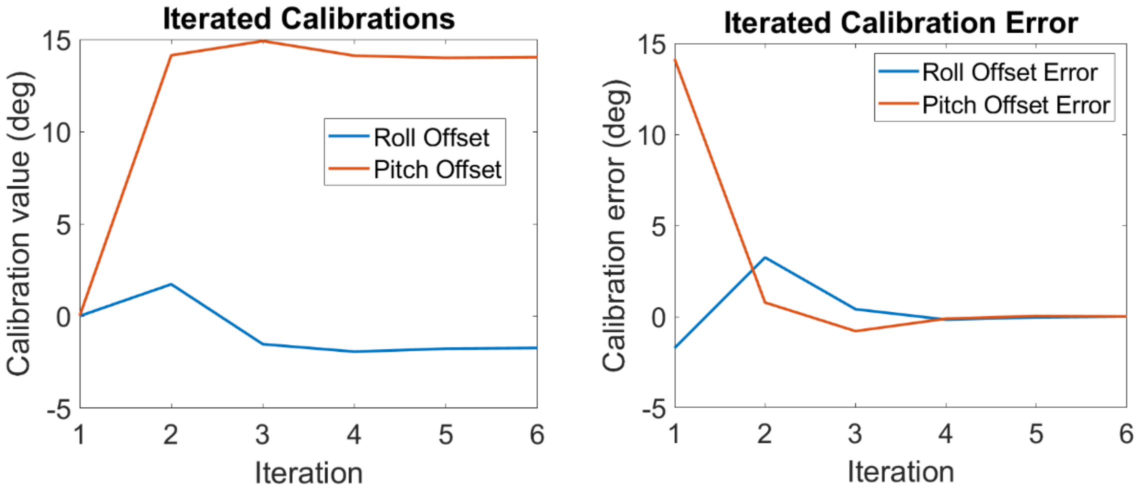


Figure 5.6: Sedan Large Misalignment Iterative Calibration Plots

Promising trend-matching can be seen in the plots for the stationary roll and pitch excitations for the sedan platform when only an autocalibration has been applied, as shown in Figure 5.7 for the small misalignment. A close up view of the roll and pitch excitation sections are shown in Figure 5.8, indicating good time-alignment and sinusoidal trend-matching, with some unexplained overshoot in the LiDAR data in comparison to the truth data. The error plots for this data are provided in Figure 5.9. From the error plots, it is clear that during the completely static portions of the test, the error stays within -0.1 and 0.1 degrees at all times. These long static portions with low error likely result in a reduction in the overall RMSE results for this stationary data. These error plots also show that the error increases during the dynamic excitation, which is further reflected by the increase in RMSE error during the tests that are primarily dynamic rather than primarily static. Some of the dynamic error may be due to sampling discrepancies between the two sensors, as they do not sample orientations at precisely the same instance in time, and no interpolation based on time stamp matching has been attempted here. It is also potentially due in part to the change in what ground section falls into the LiDAR's window of interest during the motion. Additionally, the ground is not guaranteed to be flat at all points in space, and the LiDAR is sampling different points in space during dynamic motion, thus breaking the static ground slope angle assumption that allows it to be compared to the INS data. Similar results can be seen in the large misalignment values in Figure 5.10, close up on excitations in Figure 5.11, and error plots in Figure 5.12.

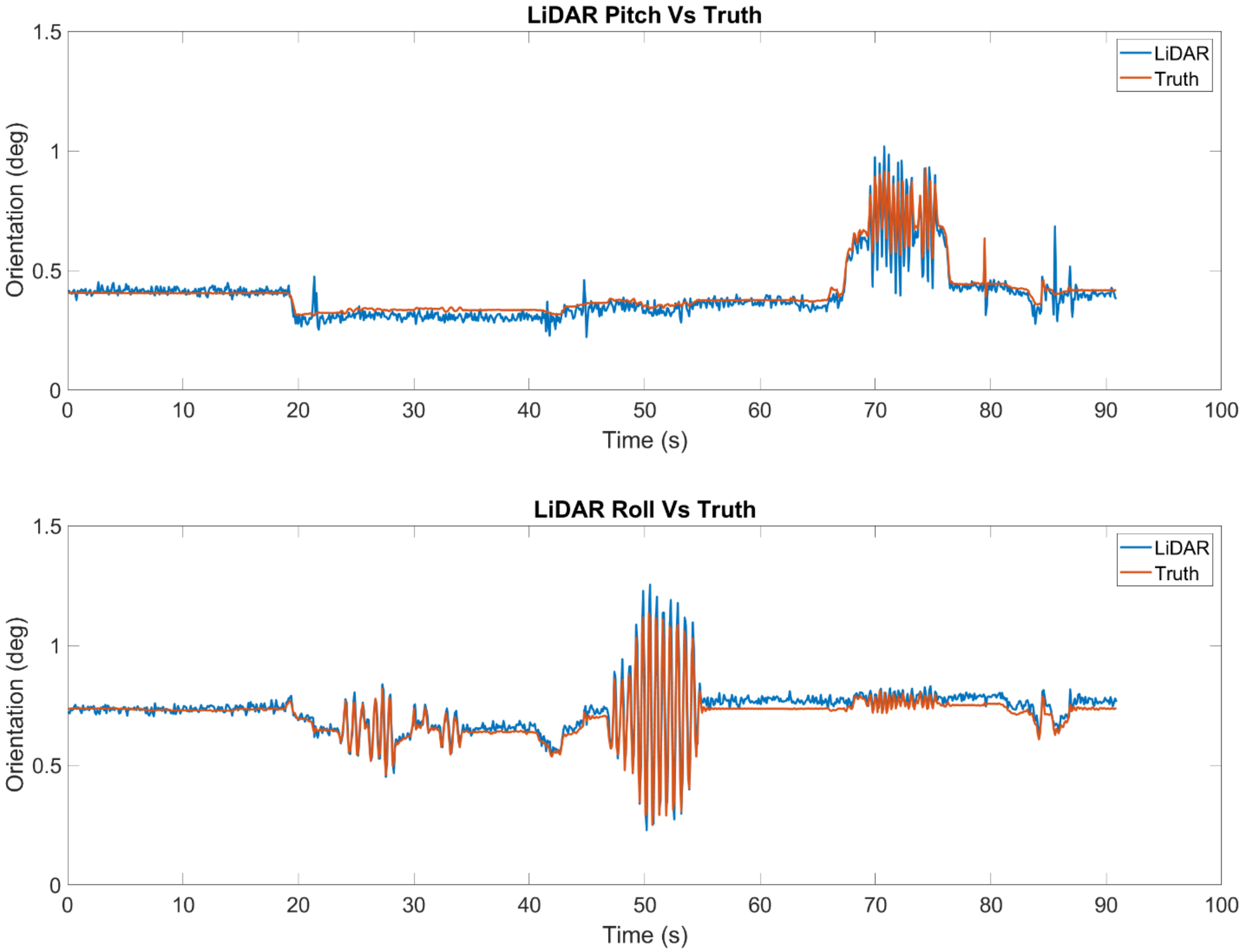


Figure 5.7: Sedan Small Misalignment Stationary Excitation Orientations with Autocalibrated LiDAR Data. (Top: Pitch Values. Bottom: Roll Values)

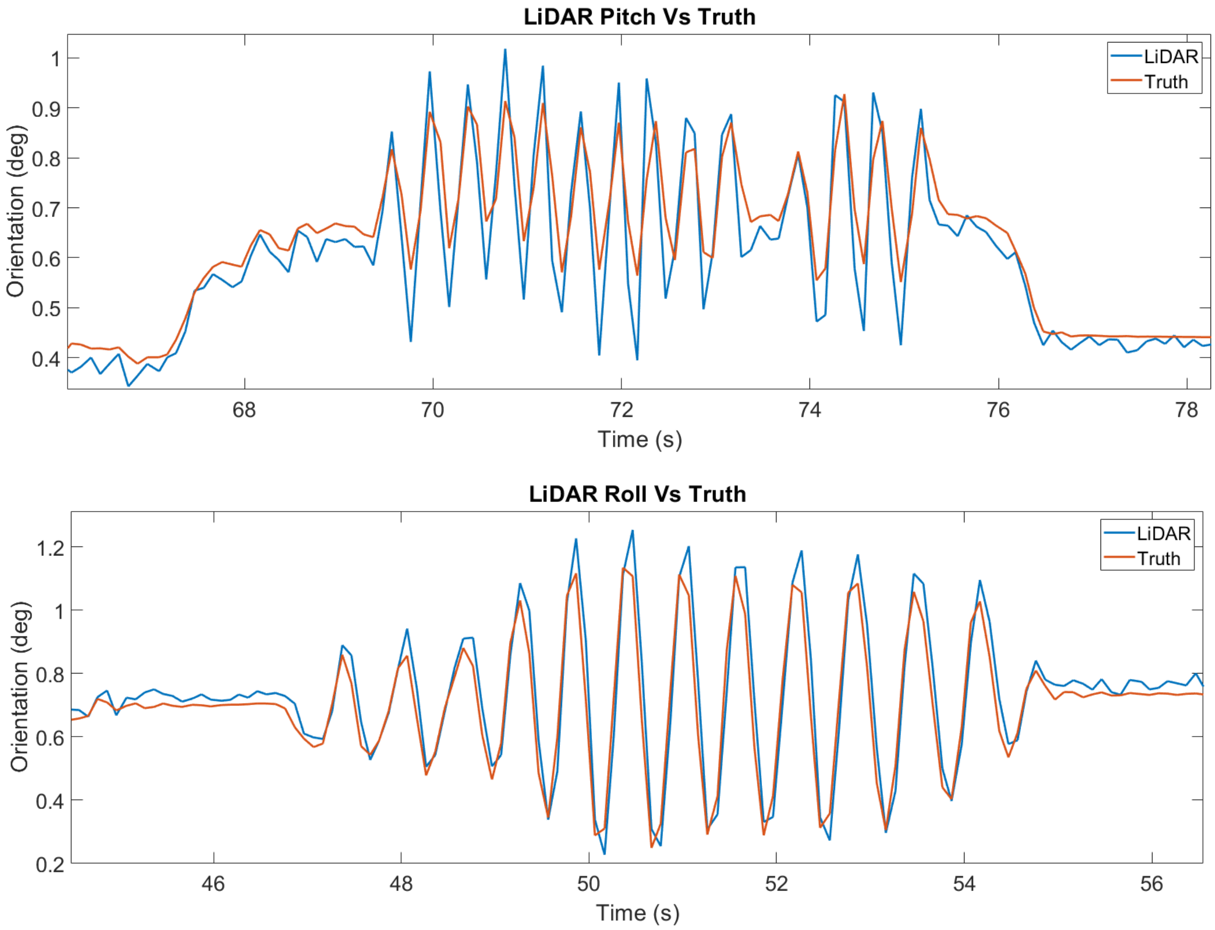


Figure 5.8: Close up on Roll and Pitch Excitations from Sedan Small Misalignment Stationary Excitation Test with Autocalibrated LiDAR Data. (Top: Pitch Values. Bottom: Roll Values)

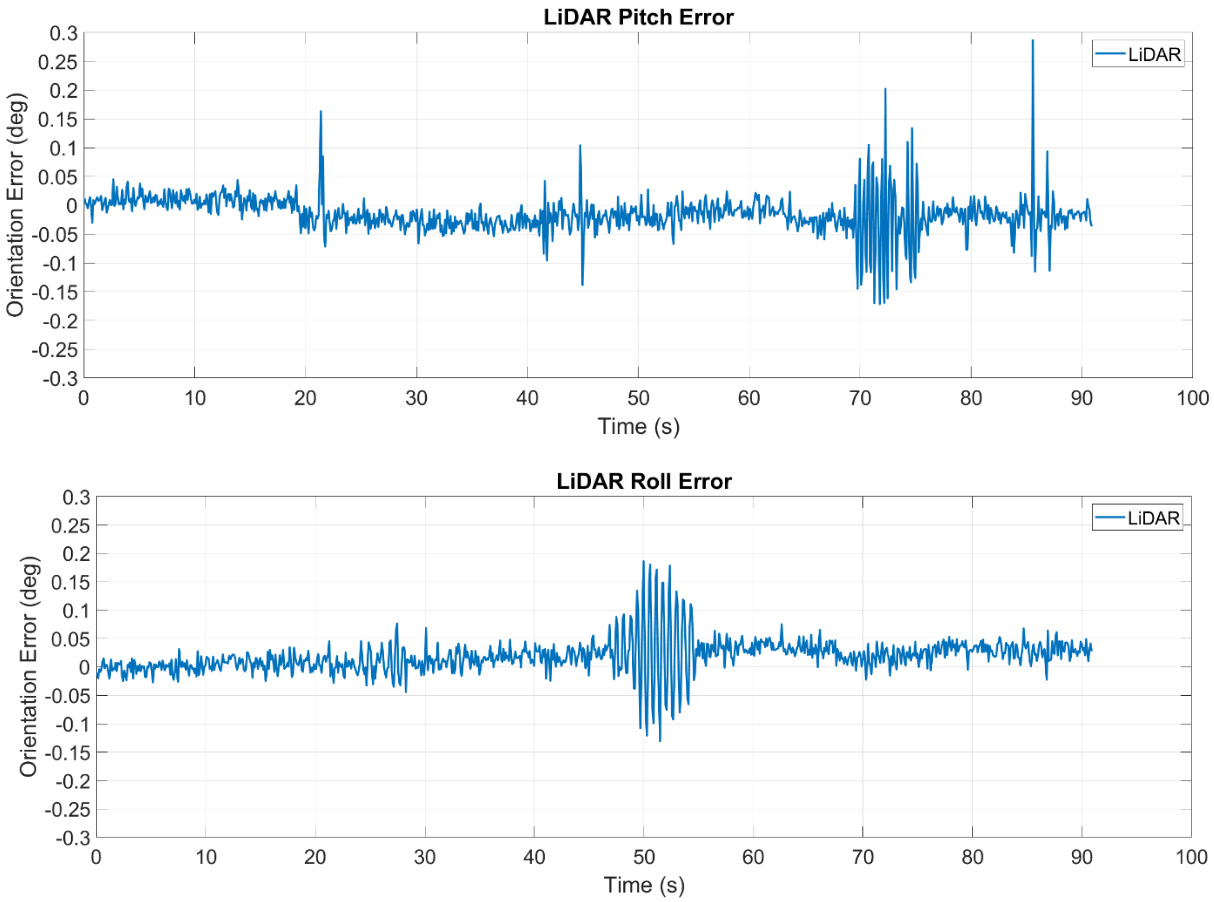


Figure 5.9: Error Plots of the Sedan Small Misalignment Stationary Excitation Orientations with Autocalibrated LiDAR Data. (Top: Pitch Values. Bottom: Roll Values)

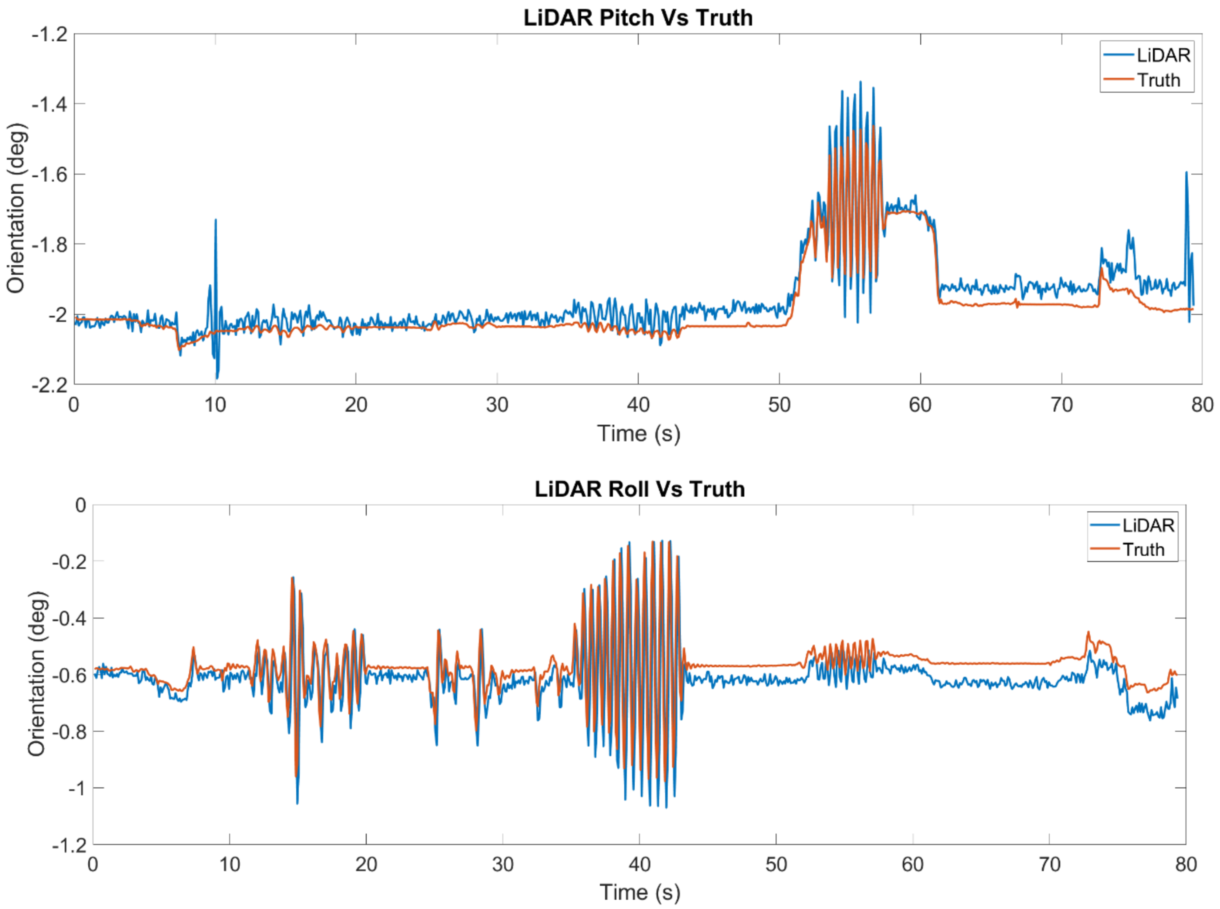


Figure 5.10: Sedan Large Misalignment Stationary Excitation Orientations with Autocalibrated LiDAR Data. (Top: Pitch Values. Bottom: Roll Values)

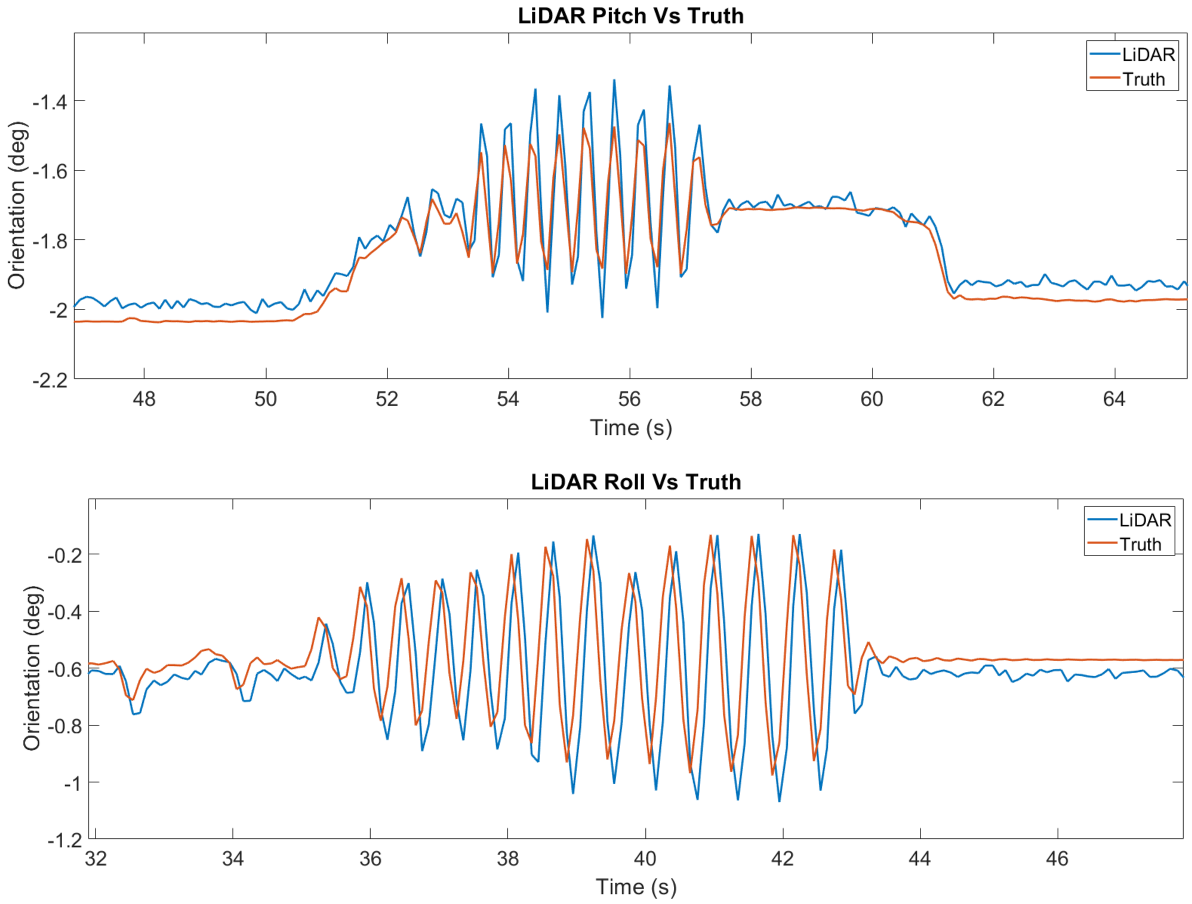


Figure 5.11: Close up on Roll and Pitch Excitations from Sedan Large Misalignment Stationary Excitation Test with Autocalibrated LiDAR Data. (Top: Pitch Values. Bottom: Roll Values)

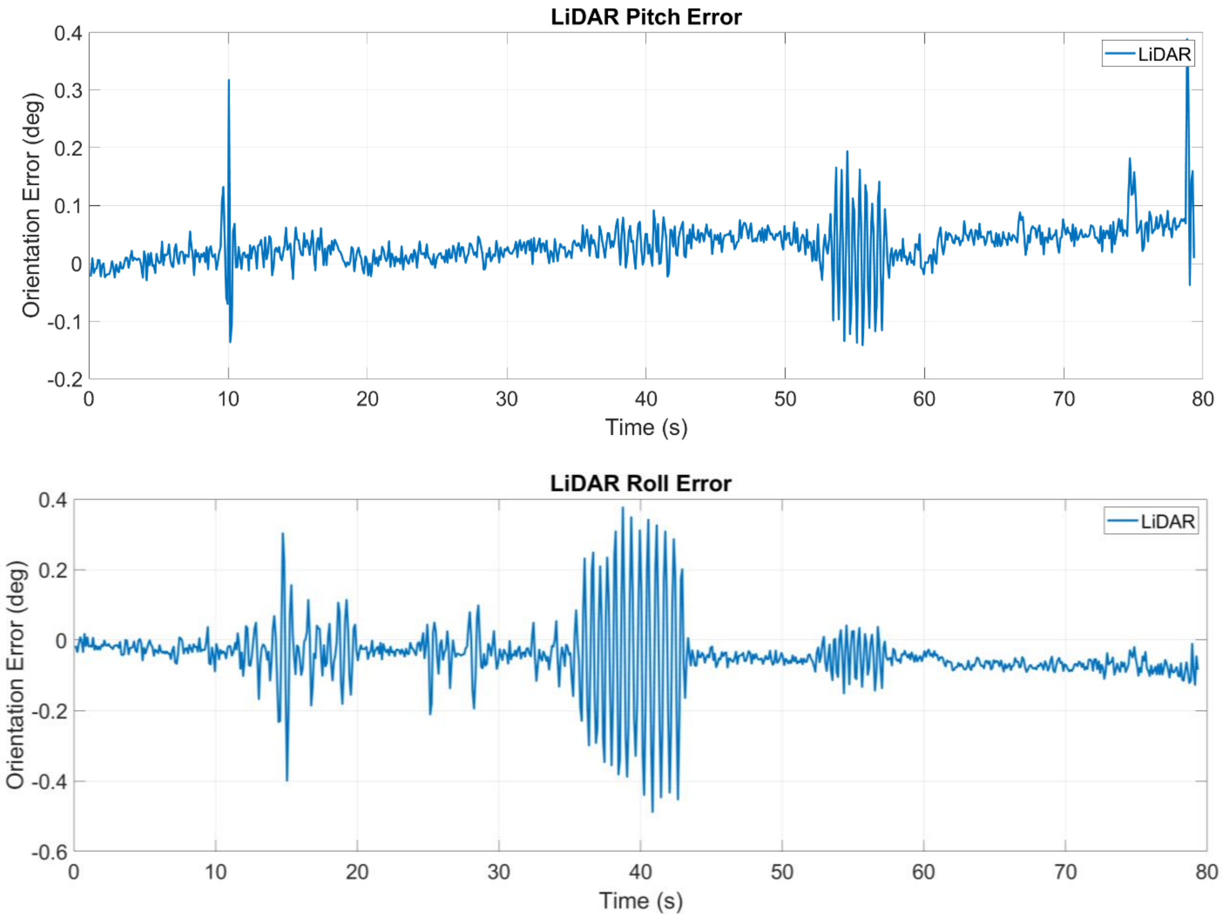
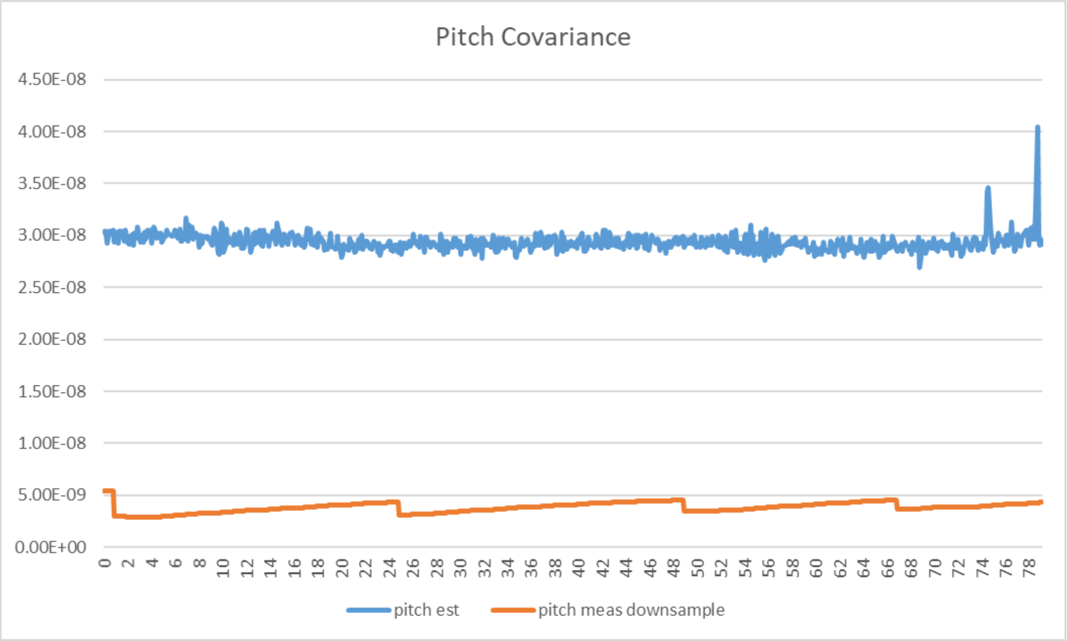


Figure 5.12: Error Plots of the Sedan Large Misalignment Stationary Excitation Orientations with Autocalibrated LiDAR Data. (Top: Pitch Values. Bottom: Roll Values)

The estimated variance values for the roll and pitch estimates from the LiDAR data are shown in Figure 5.13 for the sedan large misalignment stationary test, in comparison to the variance values reported by the truth sensor for the truth roll and pitch measurements. These plots show the variance values for a plane fit over approximately 6,000 points per frame. It is clear that the plane fit variance values as estimated are more steady than the INS variance, which express a saw tooth pattern over time. The variance between the two data sources are on similar orders of magnitude, though the pitch variance shown here for the LiDAR data is nearly an order of magnitude larger than the INS variance.



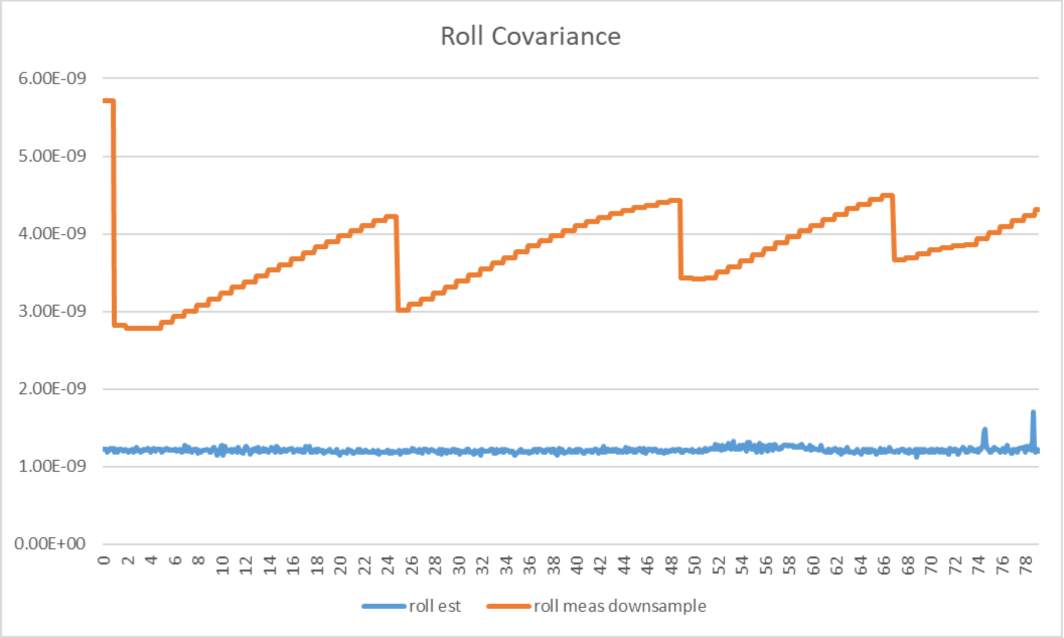


Figure 5.13: Variance comparison between LiDAR estimates (blue) and INS sensor measurements (orange) for the sedan stationary test with large misalignment (Top: Pitch, Bottom: Roll)

5.2.3 Tractor-Trailer Vehicle Results

This section discusses the results using the second test platform, a tractor-trailer with no trailer attached. The tractor-trailer utilized the same LiDAR as in the MKZ tests, an Innoviz Pro, and utilized the proprietary, high-grade GNSS/INS sensors similar to (but slightly higher-accuracy than) the iXblue sensor. The test results for the tractor-trailer were similar to the sedan test results, with an overall increase in RMSE of the dynamic tests over the stationary tests for both platforms. The tractor-trailer data had generally higher error to the small misalignment sedan test with similar mounting offsets, most likely due to the increased complexity (and therefore increased opportunities to not meet the required assumptions, and for more sources of misalignment to be introduced over simple calibration offsets). The estimated calibration offsets can be seen in comparison to the stationary and dynamic **AX=YB** offsets in Table 5.9, with roll and pitch direct magnitude differences being approximately 0.284 degrees in roll and 1.55 degrees in pitch, which is somewhat worse than the errors for the stationary sedan calibration comparison (but still with less than 2 degrees of difference).

Table 5.9: Tractor-Trailer Estimated Calibration Offsets

	Autocalibration	**AX = YB**
Roll (deg)	9.89	9.60 or 9.17
Pitch (deg)	32.4	-34.0 or -35.9
Yaw (deg)	-0.01	0.097 or -0.083

As seen before with the sedan tests, the roll and pitch calibration were calculated using iterative averaging of a set of 10 frames which were saved from the beginning of the stationary data set, and calibrated after a single average of 10 yaw offsets was applied for yaw correction. These iterative results are shown in Table 5.10. Similar to the small misalignment sedan case, only 4 iterations were needed to estimate the roll and pitch calibrations such that the initial LiDAR-to-IMU errors were less than 0.01 degrees. The consecutively-estimated calibrations and associated errors are plotted in Figure 5.14, and can easily be seen to converge.

Table 5.10: Tractor-Trailer Iterative Roll and Pitch Calibration Estimation

	Roll Calibration (deg)	**Pitch Calibration (deg)**	**Roll Error (to target ins roll) (deg)**	**Pitch Error (to target ins pitch) (deg)**
Iteration 1	0	0	-9.3547507	33.965935
Iteration 2	9.35475068	33.965935	-0.4861654	-1.5645145
Iteration 3	9.84091608	32.4014205	-0.0463647	0.00331341
Iteration 4	9.88728075	32.4047339	-0.0070099	-0.000797

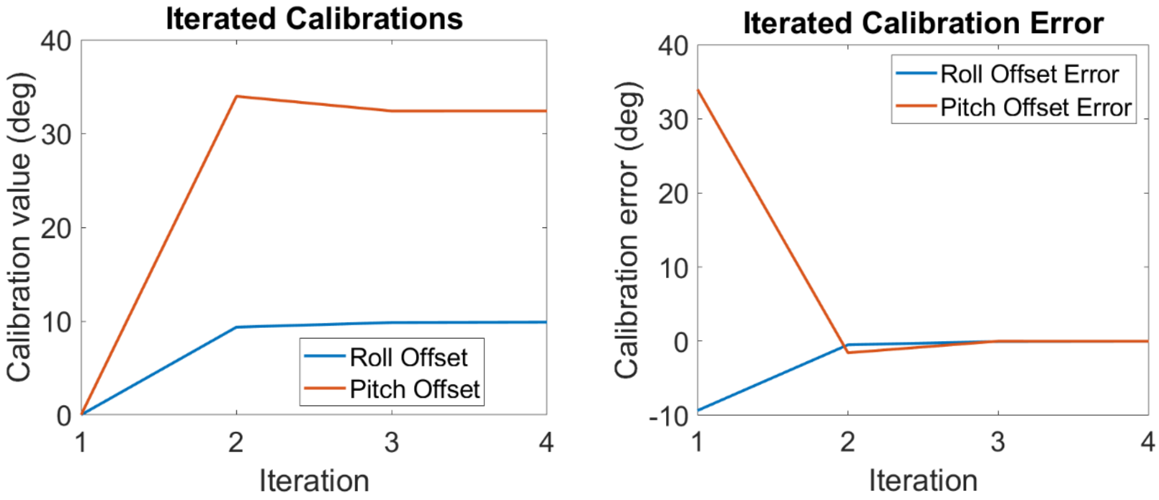


Figure 5.14: Tractor-Trailer Iterative Calibration Plots

For the tractor-trailer platform, significant noise was present on the LiDAR data during dynamic maneuvers such as the laps on the NCAT track (potentially due to the relatively low intensity or relatively high roughness of the road surfaces leading to an increase in noise in the LiDAR plane fit results). As such, it was found useful to filter the data according to the method presented in Section 6.1.3, in this case with a static value of 0.09 on the covariance matrix diagonal for both roll and pitch. The difference between unfiltered (Figure 5.15) and filtered (Figure 5.16) data is shown below, with a significant improvement in the visibility of the shape of the LiDAR data trend apparent in the filtered data. Also shown here is a significant static offset between the

LiDAR and INS values, due to different ground slopes being experienced during this test than were used to calibrate the LiDAR initially. These errors are undoubtedly exacerbating the RMSE and other error statistics used to assess the calibration during this step, as it is clear that the INS and LiDAR data are not well-aligned by the chosen calibration reference frames.

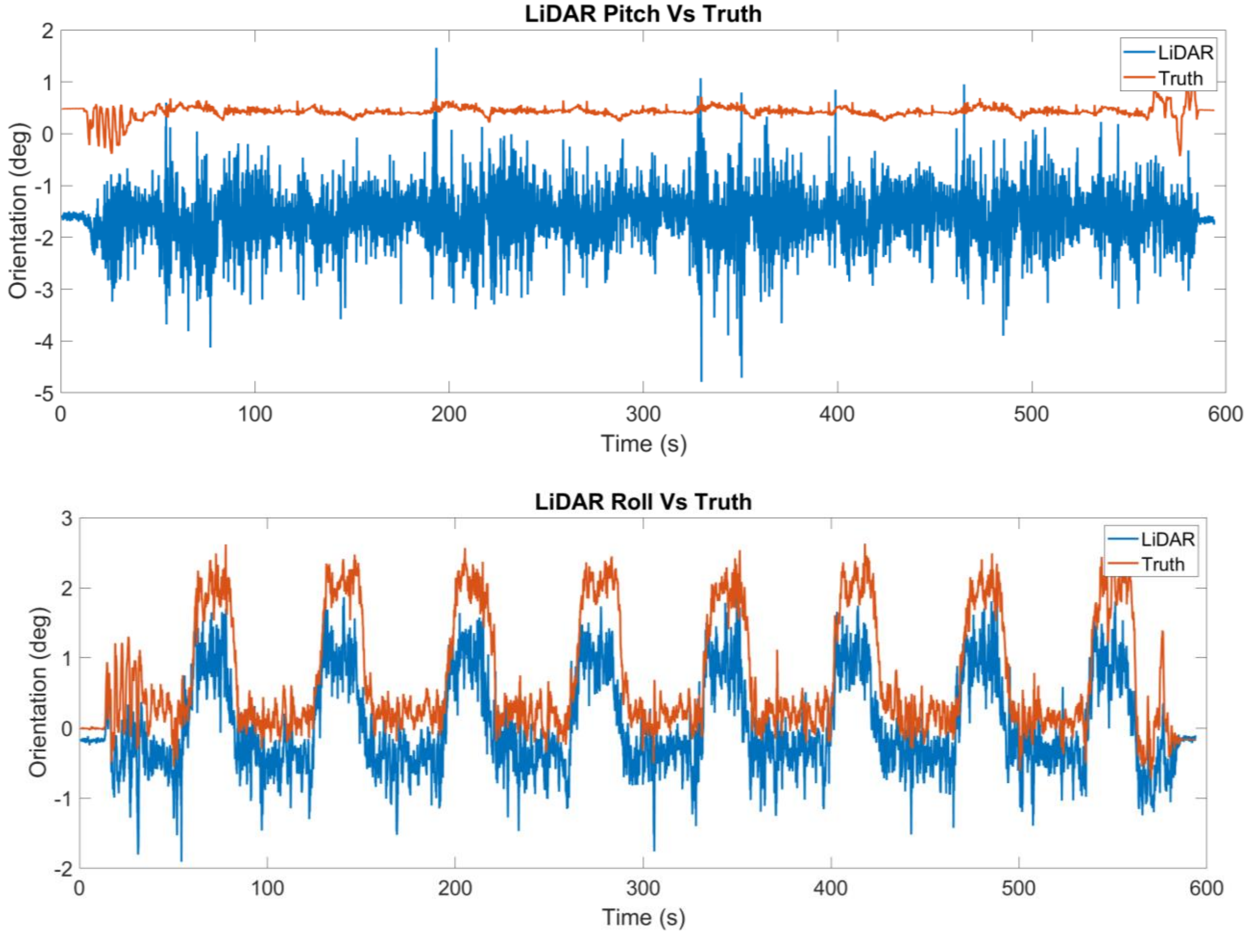


Figure 5.15: Unfiltered Tractor-Trailer Dynamic Laps Test with Autocalibrated LiDAR (Top: Pitch Values, Bottom: Roll Values)

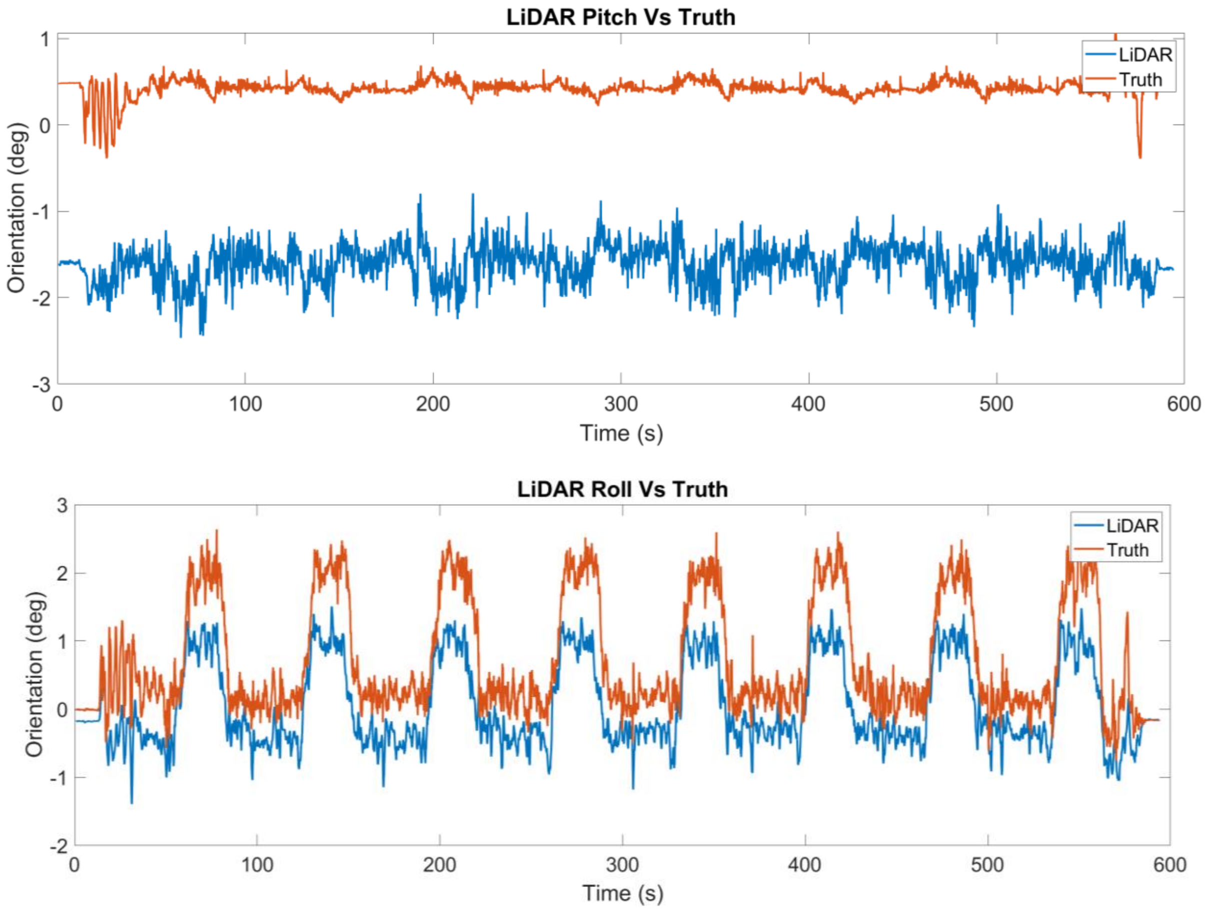


Figure 5.16: Filtered Tractor-Trailer Dynamic Laps Test with Autocalibrated LiDAR (Top: Pitch Values, Bottom: Roll Values) (Covariance Diagonal: 0.09)

Similar again to the sedan test case, promising results can be seen in the stationary excitation autocalibration results for the tractor-trailer. The results are shown in Figure 5.17, with close up of the excitation sections shown in Figure 5.18. These plots again show good time alignment in roll and again some overshoot of the LiDAR over the INS data. There does appear to be potential minor time misalignment in the pitch data, though it is hard to determine due to the noisiness of the data during the excitation. In Figure 5.19, the truly static portions are within -0.1 and 0.1 degrees of error, but those errors increase to ± 1.0 degrees for pitch and ± 0.4 for roll degrees during the dynamic portions.

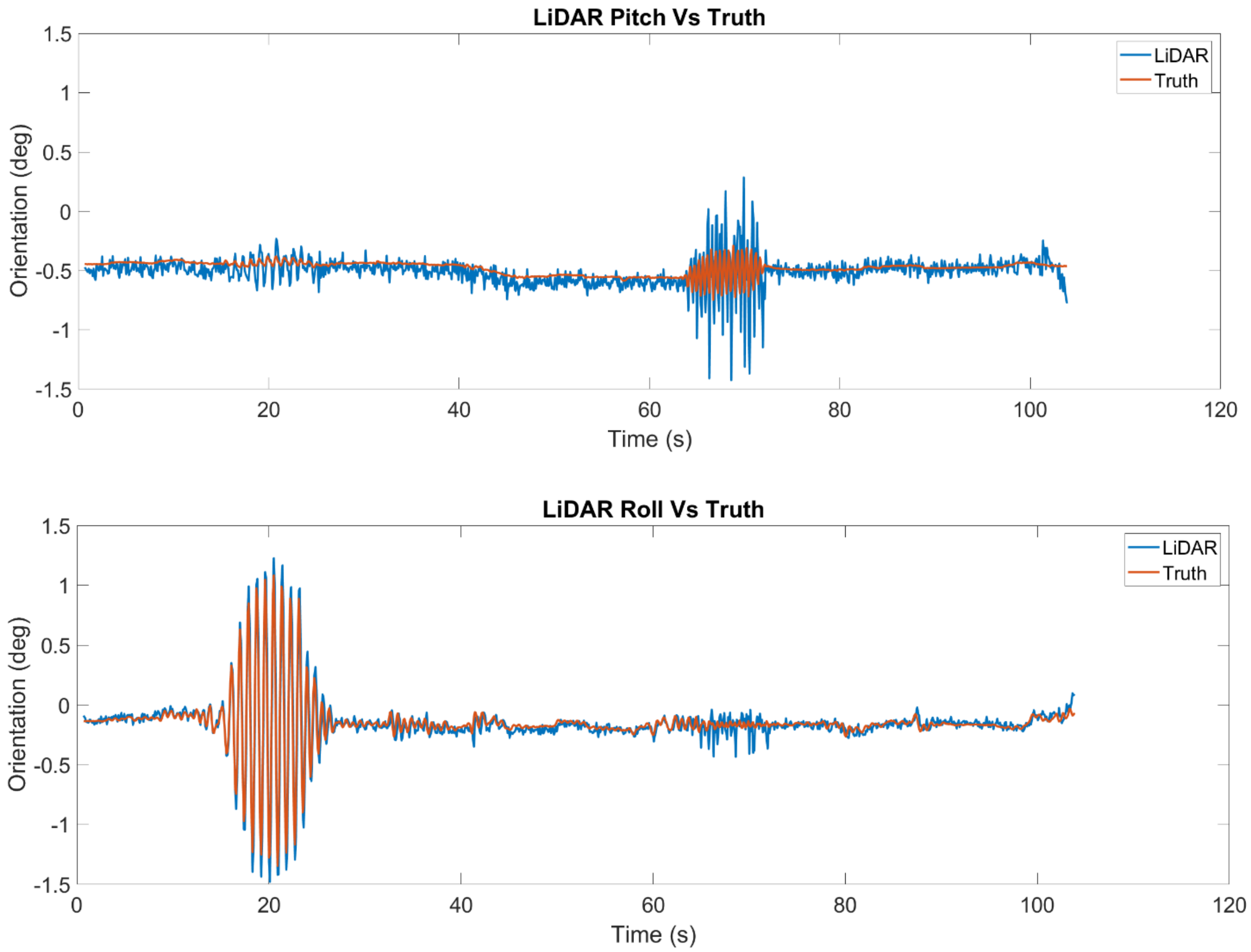


Figure 5.17: Tractor-Trailer Stationary Excitation Orientations with Autocalibrated LiDAR Data. (Top: Pitch Values. Bottom: Roll Values)

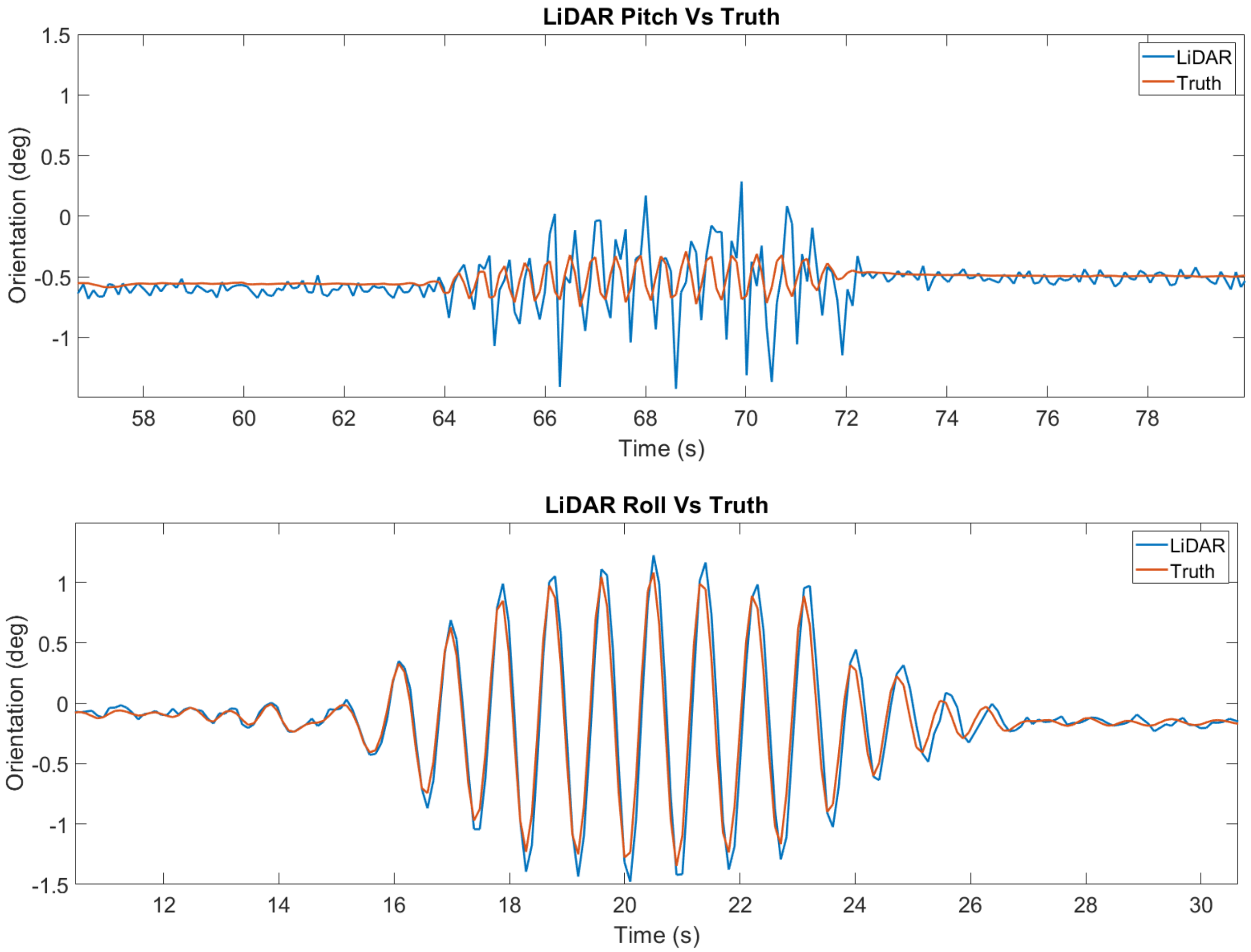


Figure 5.18: Close Up on Roll and Pitch Excitations From Tractor-Trailer Stationary Excitation Test with Autocalibrated LiDAR Data. (Top: Pitch Values. Bottom: Roll Values)

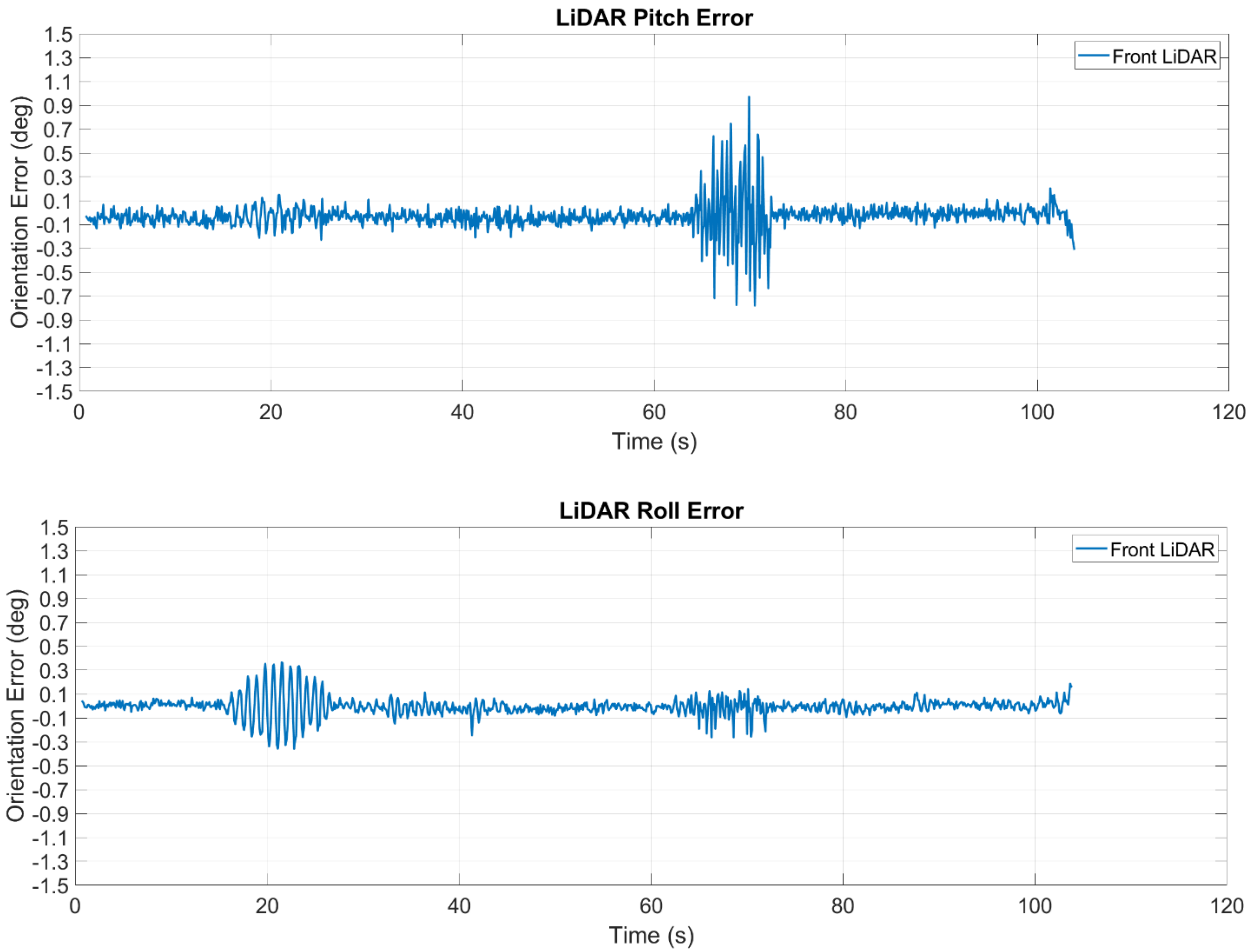
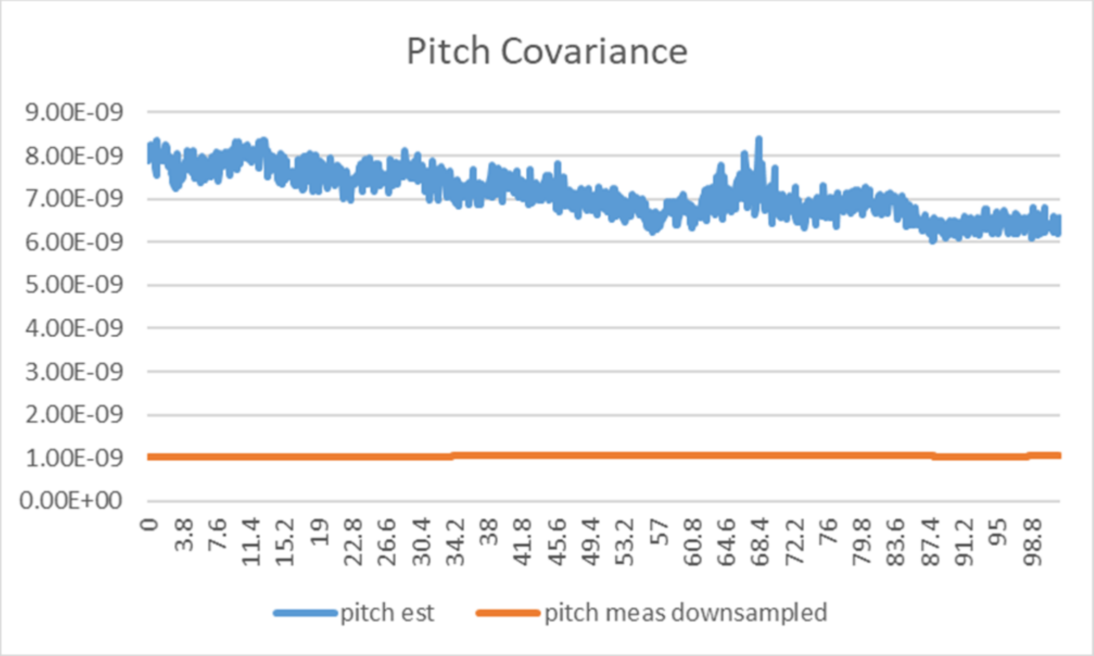


Figure 5.19: Error Plots for Tractor-Trailer Stationary Excitation Orientations with Autocalibrated LiDAR Data. (Top: Pitch Values. Bottom: Roll Values)

The variance for the stationary tractor-trailer test is shown in Figure 5.20. The LiDAR ground orientation detection algorithm used for this data included around 3,000 points per plane-fit (per frame), approximately half the points used in the sedan tests. This reduction in points is due primarily to the increase in distance from the sensor that the designated region-of-interest plane was sampled at, which were uniform in x and y bounds but radially further away in the tractor-trailer tests due to both an offset x bound as well as increased height due to the top-of-the-cab mount point. For the truth sensor used in these tests, the sensor variance does not show a distinct saw tooth pattern, and instead appears to be consistent across the test. In both roll and pitch, the LiDAR variance is around 7.00e-09 more than the sensor variance. Of note is the increase in roll variance from the sedan tests and slight decrease in pitch variance. Overall, it is expected that the variance for this test should be larger than for the sedan tests due to the decrease in sampled points.



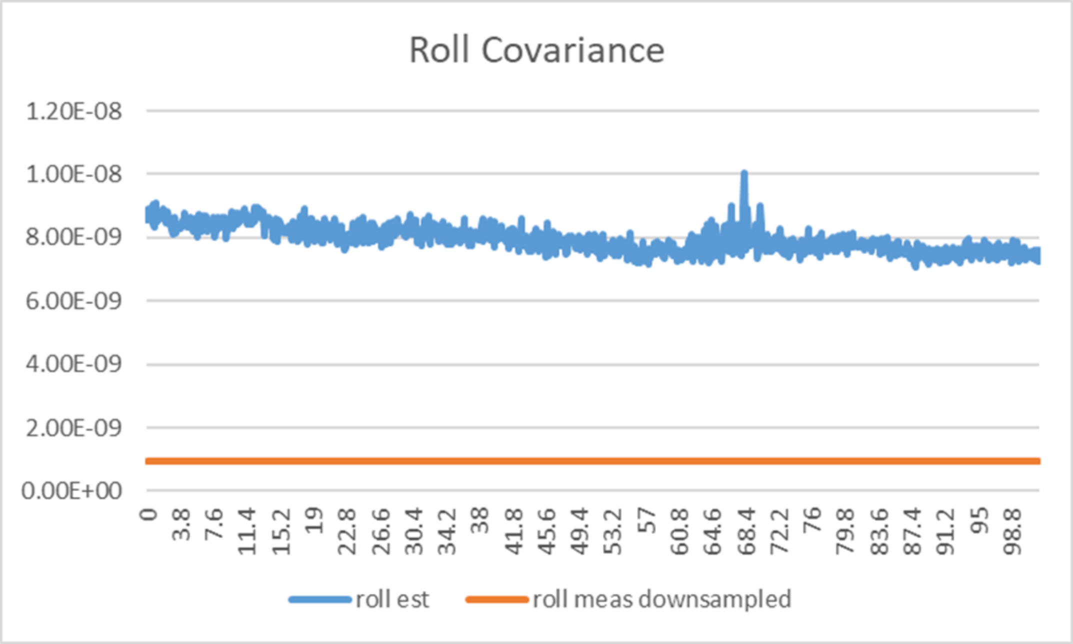


Figure 5.20: Variance comparison between LiDAR estimates (blue) and INS sensor measurements (orange) for the tractor-trailer stationary test (Top: Pitch, Bottom: Roll)

5.2.4 Discussion of Truth Validity

The assumption that the offsets between the INS truth data and the LiDAR orientations are static is sometimes problematic. This issue is apparent in the roll and pitch measurements during the small-misalignment sedan road drive test, as shown in Figure 5.21. Particularly along the stretch of data between 50 seconds and 200 seconds, the truth data can be seen deviating from around zero degrees with peak magnitudes up to 2 or 6 degrees, while the LiDAR data remains relatively close to 0 degrees. This is presumably due to hills and valleys in the roadway. As the vehicle climbs a hill, its pitch with respect to a level gravity plane becomes more negative (i.e. its nose tilts up with respect to the theoretically level gravity plane). However, the roadway itself has a pitch slope (or grade angle) which causes the vehicle to pitch its nose upward, and as such the vehicle-to-road angle which is visible to the LiDAR during these climbs is much smaller in magnitude than the INS sensor angles. A similar phenomenon occurs for downhill motion. This example highlights the incongruence between the frames measured by the LiDAR and those measured by the inertial sensors used here as “truth.” The LiDAR and an inertial truth may only be compared directly to one another when the ground grade and bank are static and accounted for, which is not always the case even when the vehicle is stationary, and is almost never the case when the vehicle is moving across a surface such as a roadway or parking lot. As an aside, it is also interesting to see within the road data that at points in time, the LiDAR and inertial angles did trend together, in places where the pitch excitation seen in the INS sensor is presumably due to dynamics of the vehicle independent of the current road grade. It may be loosely stated that the LiDAR is providing only measurements of highly dynamic motion (instances where the vehicle is shifting relative to the ground orientation), while the inertial sensor is providing road-slope-following rotational motion with the higher dynamics seen by the LiDAR expressed as disturbances.

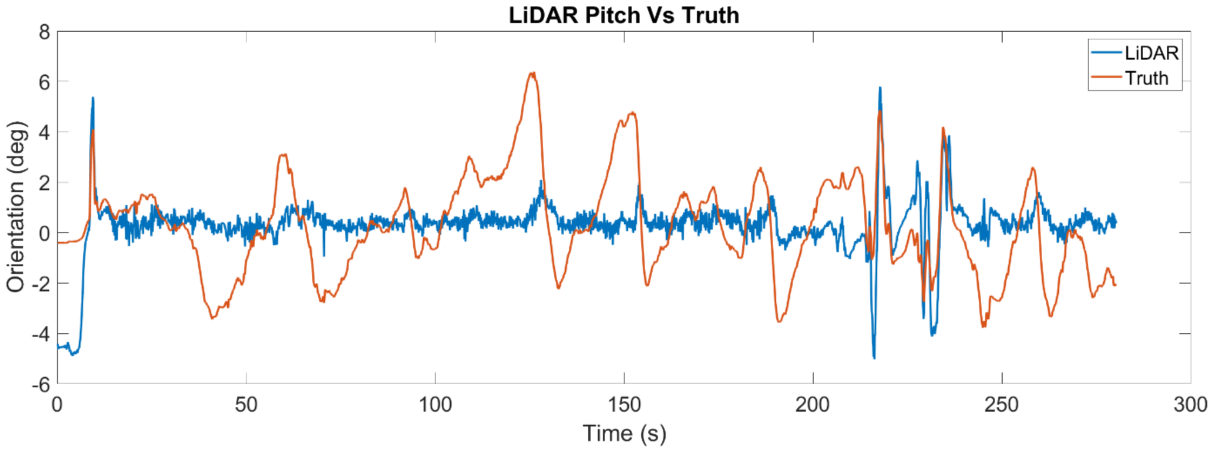


Figure 5.21: Calibrated Pitch Data from Sedan Road Drive

The surface of the skid pad, assumed to be a uniformly-sloped plane, showed variations in inertial slope at different points along its surface, indicated by the 4 degree difference in pitch between the truth and LiDAR data at the beginning of the data in Figure 5.21 (which started from a location on the skid pad). Varying values of initial offsets can also be seen in other plots for the MKZ sedan tests, excluding the stationary tests as the initial frames from this type of test were used to estimate the roll and pitch calibration offsets from a level gravity plane reference. These errors in IMU data, which is used as the truth source, should be taken into consideration when viewing the experimental LiDAR algorithm results shown here.

In Figure 5.22, the orientation results for the dynamic figure-eight sedan test when the LiDAR was mounted with a small yaw offset to the vehicle frame are shown. Here, it can be seen in both the pitch and roll plots that the different starting location on the skid pad from the previous test (to which the LiDAR was calibrated) result in static offset angles prior to motion. Note that the inertial truth values for the pitch from this test switch between about ± 1 degree throughout the motion. It is expected that the vehicle experiences significantly more roll than pitch excitation during directional changes in the figure-eight maneuver, but the trend in Figure 5.20 does not meet this expectation. It is possible that the inertial angles change sign whenever the vehicle changes direction according to the major sloping orientation of the skid pad (i.e. there is a significant north-to-south slope of a few degrees). As the vehicle goes up the major slope, the vehicle experiences a negative inertial pitch, and as it goes down a slope it sees a positive inertial pitch. The ground remains approximately level with respect to the LiDAR, however, and the LiDAR data does not express this sign change, leading to the expressed incongruence between the sign of the LiDAR data with respect to the truth data during certain time intervals.

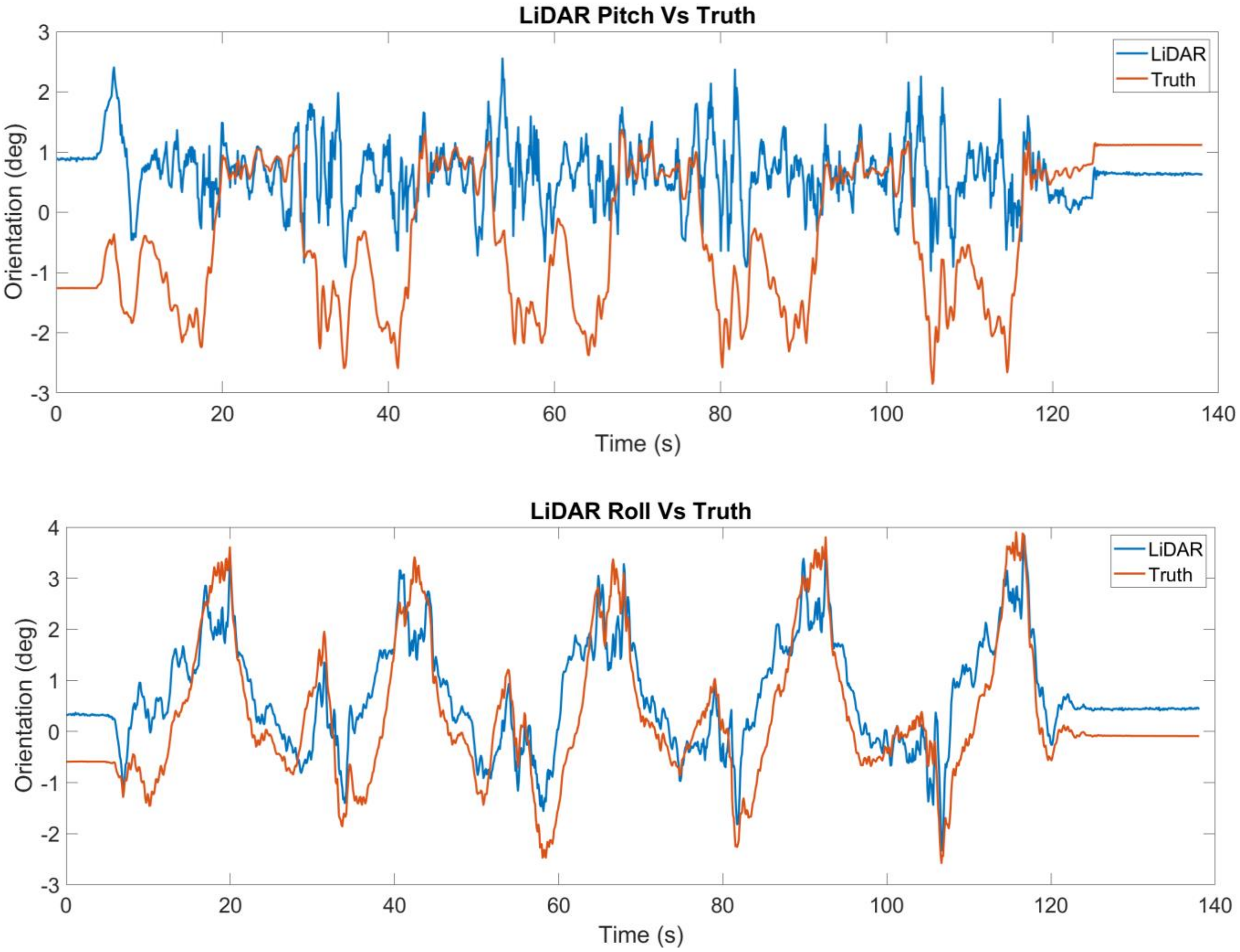


Figure 5.22: Sedan Small Misalignment Autocalibrated LiDAR Results for a Figure 8 (dynamic) Test. (Top: Pitch results, Bottom: Roll results)

Chapter 6

Conclusions and Future Work

LiDAR devices are common sensors for the perception suite of ground vehicles and mobile robots. The extrinsic calibration of these sensors reduces the error of data and estimates generated from the LiDAR data, as well as increases its usability in cases of large mounting offsets between the LiDAR and the vehicle it is equipped on. Extrinsic calibration of LiDAR often requires manual procedures, target objects to be placed in the view of the LiDAR, or companion sensors such as cameras or inertial measurement units (IMU). Few sources in literature describe a fully self-dependent, algorithmic calibration of a LiDAR which also can be incorporated into normal operational scenarios. This thesis proposed a new approach for such online self-calibration for the rotational calibration components, such that only a short static period on a relatively level surface and travel along a straight section of distinctly-edged roadway or pathway are required to achieve the calibration.

Specifically, this thesis made three major contributions. The primary contribution was the development and testing of the proposed rotational extrinsic calibration routine on two vehicle platforms. In addition, the thesis also contributed the development and testing of a road edge and road marking detection scheme for LiDAR, leveraging the Hough transform on sparse birds-eye-view point cloud images with a minimum angle selection criteria. The third major contribution was a ground roll and pitch estimation routine using a parameterized plane-fit approach with static window of interest and iterative outlier removal.

Through testing on a simple stationary platform with a 4-channel MEMS LiDAR and optical motion capture tracked orientations as truth, the ground orientation algorithm was tested to perform with at least 0.18 to 0.78 degrees RMSE in roll and pitch, though the test situation revealed unmet assumptions (e.g. that the ground should be uniformly oriented at all sample locations), and the errors during this test were exacerbated by experimental error.

The road edge and lane marking detection approach was tested on a roadway with painted lane markings using a 16-channel rotationally scanning LiDAR and compared to parameterized line fit results from manually annotated lane points from each frame as truth. This analysis revealed that when the algorithm is incapable of providing a valid lane line result, the outputted orientation of detected line has a large error to truth, and as such has a large difference to the preceding and following valid lane line detections. The algorithm was able to detect valid lane lines in 80% of tested frames, and had RMSE of 0.47 degrees in detected line orientation.

The full calibration was tested indirectly by estimating and applying the rotational calibration corrections to data collected form a 4-channel MEMS LiDAR with various offsets to the attached vehicle, and subsequently conducting ground orientation estimation using the calibrated data. In these tests, the MEMs LiDAR was used to generate all three roll, pitch, and yaw calibration offset estimates. Two different vehicle platforms were used – a sedan passenger vehicle and a tractor-trailer without attached trailer. The roll and pitch results from the ground orientation algorithm applied to calibrated LiDAR data were compared to orientation data collected from an IMU aligned to and rigidly attached to the same vehicle body as the LiDAR during tests in which the vehicle did not travel across the ground. For dynamic test cases on the tractor-trailer, the LiDAR data was compared to a conglomerate truth consisting of two IMU outputs and suspension deflection measurements. Across these tests, the calibrated roll and pitch were found to have an average RMSE of 0.070 and 0.069 degrees respectively for cases where the vehicles experienced rotational excitation while stationary. Due in part to “truth” inaccuracies, the RMSE during dynamic tests (e.g. road travel) for the calibrated orientation results were increased to 0.956 and 1.77 degrees in roll and pitch respectively.

This study has revealed opportunities for potential improvement and future work. In testing improvements, if a confident truth value for the calibration transforms can be generated, the accuracy of the calibration values estimated with this approach may be directly assessed. From prior art on extrinsic LiDAR calibration, it is not common to utilize truth values for the calibration estimates during experimentation and performance assessment. However, a truth value may be generated through use of a more laborious calibration approach such as a thorough target-based maneuver scenario or total station surveying, or one which utilizes companion sensors and has been tested with good results. The direct assessment of calibration truth would allow the algorithm to be tested in a wider variety of scenarios, as it would no longer be limited by the restrictive assumptions that allow the IMU orientations to be compared to the ground orientation results.

Algorithmic improvements may involve a more robust non-planar object resilience in the ground orientation algorithm, and the capability of detecting curved road sections and reporting the road curvature from the road edge and lane line detection algorithm. Rather than batch averaging the calibration results from several selected frames, a more continuous fileting scheme such as a Kalman filter could be tested for a quicker, potentially more robust convergence to the calibration estimate.

Finally, this rotational calibration estimation may be augmented with a lever arm estimation routine. Height lever arm calibration from the LiDAR to the base footprint frame may be estimated from ground plane detection in conjunction with the estimation of the roll and pitch offsets. Calibration may be attempted for longitudinal lever arm with curved road trajectory detection. Longitudinal lever arm from road curvature assumes that the origin of the reference frame is a point on the vehicle that closely tracks the trajectory of the roadway during the curve, and therefore will be offset by any present sideslip in the vehicle motion. This calibration may be corrected if sideslip estimates are available. Lateral lever arm estimation is more difficult, with one potential solution being the utilization of observable ego-vehicle symmetry in the LiDAR field of view.

Bibliography

[1] Volvo. Guided by laser: how autonomous cars see.

[2] Z. Koppanyi and C. K. Toth (2018). Experiences with acquiring highly redundant spatial data to support driverless vehicle technologies. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume IV-2, 2018 ISPRS TC II Mid-term Symposium "Towards Photogrammetry 2020".*

[3] Z. Pusztai, L. Hajder E2017). Accurate Calibration of LiDAR-Camera Systems Using Ordinary Boxes. *Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 394-402.*

[4] X. Gong, Y. Lin, and J. Liu (2013). 3D LIDAR-Camera Extrinsic Calibration Using an Arbitrary Trihedron. *Sensors 2013, 13(2).*

[5] Z. J. Taylor and J. I. Nieto (2015). Motion-based calibration of multimodal sensor arrays. *IEEE International Conference on Robotics and Automation.*

[6] J. Lyu, J. Xu, X. Zuo, and Y. Liu (2019). An Efficient LiDAR-IMU Calibration Method Based on Continuous-Time Trajectory. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*

[7] Y. Chen and G. Medioni (1991). Object Modeling by Registration of Multiple Range Images *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*.

[8] J. Britt (2016). Simultaneous Localization and Auto-Calibration and Mapping of Ground Vehicles. *Dissertation, Auburn University.*

[9] J. Alberts, S. P. Kleinschmidt, and B. Wagner (2019). Robust calibration procedure of a manipulator and a 2D laser scanner using a 1D calibration target. *17th International Conference on Informatics in Control, Automation and Robotics At: Prague, Czech Republic.*

[10] J. H. Britt (2010). Lane detection, calibration, and attitude determination with a multi-layer lidar for vehicle safety systems. *Thesis, Auburn University.*

[11] Velodyne LIDAR (2017). Velodyne LiDAR Launches VLS-128, The World's Highest Resolution LiDAR for Autonomous Vehicles.

[12] ArcMap. What is lidar intensity data? https://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/what-is-intensity-data-.htm

[13] Autonomous Stuff. Aptive ESR 2.5. https://autonomoustuff.com/products/aptiv-esr-2-5-24v

[14] Autonomous Stuff. Quanergy M8 LiDAR Sensor. https://www.egps.net/datasheets/quanergy-m8-datasheet.pdf

[15] Velodyne (2018). Velodyne LiDAR Puck. Real-Time 3D LiDAR Sensor. *Velodyne LiDAR, Inc,* https://www.mapix.com/wp-content/uploads/2018/07/63-9229_Rev-H_Puck-_Datasheet_Web-1.pdf

[16] Innoviz Technologies Ltd. (2019). Innoviz Pro. High Performance Solid-State LiDAR. https://innoviz.tech/wp-content/uploads/2018.12.25-InnovizPro-Datasheet.pdf

[17] Ouster. Product Website. https://ouster.com/

[18] Hexagon/Autonomous Stuff. Luminar Hydra. https://autonomoustuff.com/products/luminar-hydra

[19] G. Brooker, R. Hennessey, C. Lobsey, M. Bishop, and E. Widzyk-Capehart (2007). Seeing through dust and water vapour: Millimeter wave radar sensors for mining applications. *Journal of Field Robotics, V. 24, Issue 7 p. 527-557.*

[20] S. Zhang, H. Wang, J. Gao, C. Xing. (2019). Frequency domain point cloud registration based on the Fourier transform. *Journal of Visual Communication and Image Representation, V. 61 May 2019 p. 170 – 177.*

[21] Y. Li and E. B. Olson (2010). Extracting general-purpose features from LIDAR data. *2010 IEEE International Conference on Robotics and Automation, 2010, pp. 1388-1393, doi: 10.1109/ROBOT.2010.5509690.*

[22] L. de Paula Veronese, A. Ismail, V. Narayan, and M. Schultze (2019). An accurate and computational efficient system for detecting and classifying ego and sides lanes Using LiDAR. *(Unpublished) Visteon***.**

[23] Velodyne LiDAR. HDL-64E: High Definition Real-Time 3D Lidar. https://velodynelidar.com/products/hdl-64e/

[24] P. Pfaff, W. Burgard, and D. Fox (2006). Robust Monte-Carlo Localization using Adaptive Likelihood Models. *H. Christiensen, editor, European Robotics Symposium 2006, volume 22 of springer star advanced, pages 181-194. Springer-Verlag Berlin Heidelberg, Germany.*

[25] M. Himmelsbach, A. Müller, T. Lüttel, and H. J. Wünsche. (2008). LIDAR-based 3D objet perception. *Proceedings of 1st International Workshop on Cognition for Technical Systems.*

[26] J. M. Maroli, Ü. Özgüner, K. Redmill, and A. Kurt (2017). Automated rotational calibration of multiple 3D LIDAR units for intelligent vehicles. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC).*

[27] Y. D. Chen, J. Ni, and S. M. Wu (1993). Dynamic calibration and compensation of a 3D laser radar scanning system. *Proceedings IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 1993, pp. 652-658 vol.3, doi: 10.1109/ROBOT.1993.291831.*

[28] K. Zhang and H. C. Frey (2005). Road Grade Estimation for On-Road vehicle Emissions Modeling Using LiDAR Data. *Proceedings, Annual Meeting of the Air & Waste Management Association, June 20-23, 2005, Minneapolis, MN.*

[29] J. Lam, K. Kusevic, P. Mrstik, R. Harrap, and M. Greenspan (2010). Urban scene extraction from mobile ground based lidar data. *Proceedings of the 5th International Symposium on 3D Data, Processing, Visualization & Transmission.*

[30] C. R. Qi, H. Su, K. Mo, and L. J. Guibas (2017). PointNet: Deep Learning on Point sets for 3D Classification and Segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[31] B. Wu, A. Wan, X. Yue, and K. Keutzer (2018). SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. *2018 IEEE International Conference on Robotics and Automation (ICRA).*

[32] H. Chen. The Effets of Movement Speeds and Magnetic Disturbance on Inertial Measurement Unit Accuracy: The Implications of Sensor Fusion Algorithms in Occupational Ergonomics Applications. *Dissertation, University of Iowa.*

[33] G. S. Faber, C. Chang, P. Rizun, J. T. Dennerlein (2013). A novel method for assessing the 3-D orientation accuracy of inertial/magnetic sensors. *J Biomech 2013 doi: 10.1016/j.jbiomech.2013.07.029*

[34] Stevens, Brian L., Frank L. Lewis. Aircraft Control and Simulation, 2nd Edition. *Hoboken, NJ: John Wiley & Sons, 2003.*

[35] J. Diebel (2006). Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. *Stanford University, Stanford, California.*

[36] S. L. Brunton and J. N. Kutz (2019). Data driven science & engineering: machine learning, dynamical systems, and control. *Ch 1 Sec. 4 Cambridge University Press.*

[37] C. G. J. Jacobi (1846). Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkom-menden Gleichungen numerisch aufzulösen, Crelle's Journal für reine und angew. *Math.,30 (1846), pp. 51–95.*

[38] Demmel, J. and Veselić, K. (1992). Jacobi's method is more accurate than QR. *SIAM Journal on Matrix Analysis and Applications 13(4) p1204-1245.*

[39] Z. Drmac and K. Veselic. New Fast and Accurate Jacobi SVD Algorithm: II. Lapack working Note 170.

[40] M. Bai, G. Mattyus, N. Homayounfar, S. Wang, S. K. Lakshmikanth, R. Urtasun (2018). Deep Multi-Snesor Lane Detection. *IEEE International Conference on Intelligent Robots and Systems (IROS) 2018*.

[41] L. Caltagirone, S. Scheidegger, L. Svensson, M. Wahde (2017). Fast LIDAR-based Road Detection Using Fully Convolutional Neural Networks. *2017 IEEE Intelligent Vehicles Symposium (IV).*

[42] A. S. Huang, D. Moore, M. Antone, E. Olson, S. Teller (2009). Finding Multiple Lanes in Urban Road Networks with Vision and LiDAR. *Autonomous Robots 26(2-3)*.

[43] J. Kibbel, W. Justus, K. Furstenberg (2005). Lane estimation and departure warning using multilayer laserscanner. *Proceedings 2005 IEEE Intelligent Transportation Systems*.

[44] P. Lindner, E. Richter, G. Wanielik, K. Takagi and A. Isogai (2009). Multi-channel lidar processing for lane detection and estimation. *12th International IEEE Conference on Intelligent Transportation Systems, 2009, pp. 1-6, doi: 10.1109/ITSC.2009.5309704.*

[45] Alberto Holgado-Barco, Diego Gonzalo-Aguilera, Pedro Arias-Sanchez, Joq uin Martinez-Sanchez. Semiautomatic Extraction of Road Horizontal Alignment from a Mobile LiDAR System. *Computer-Aided Civil and Infrastructural Engineering.*

[46] T. Ogawa and K. Takagi (2006). Lane recognition using on-vehicle LiDAR. *IEEE Intelligent Vehicle Symposium, 2006.*

[47] J. Jung and S. Bae (2018). Real-time road lane detection in urban areas using LiDAR data. *Electronics V7.*

[48] J. Matas, C. Galambos, and J. Kittler (2000). Robust detection of lines using the progressive probabilistic Hough transform. *Computer Vision and Image Understanding, 78(1):119–137*.

[49] F. Ghallabi, F. Nashashibi, G. El-Haj-Shhade, M. Mittet (2018). LIDAR-Based Lane Marking Detection for Vehicle Positioning in an HD Map. *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*.

[50] M. Himmelsbach, F.V. Hundelshausen and H.-J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles", *IEEE Inteligent Vehicles Symposium*, 2010.

[51] OptiTrack Cameras. Prime 13. https://www.optitrack.com/cameras/prime-13/

[52] H. Zhuang, Z. S. Roth, and R. Sudhakar. Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form AX = YB. *IEEE Transactions on Robotics and Automation*, vol. 10, *no. 4, pp. 549-554, Aug. 1994, doi: 10.1109/70.313105*.

[53] F. Dornaika and R. Horaud. Simultaneous Robot-World and Hand-Eye Calibration. *IEEE Transactions on Robotics and Automation, Institute of Electronics Engineers (IEEE).*

[54] M. I. Shah, R. D. Eastman, and T. H. Hong (2012). An Overview of Robot-Sensor Calibration Methods for Evaluation of Perception Sensors. *2012 Performance Metrics for Intelligent Systems Workshop*.

[55] iXblue. Atlans Series. https://www.ixblue.com/products/atlans-series

[56] T. Lacey. Tutorial: The Kalman Filter. *Kalman Filter ch. 11 p. 134-140, MIT.*

[57] W. H. Greene (2002). Econometric Analysis, fifth edition. *Prentice Hall, New Jersey*. p. 16, 19, 46

Appendix A

Rotational Transforms

A.1 Roll, Pitch, and Yaw Rotations

The following single-axis DCM transforms are defined between generic global and body frames. A roll angle is indicated by ϕ, a pitch angle by θ, and a yaw angle by ψ.

$$R_g^b(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$R_g^b(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$R_g^b(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

A.2 Derivation of YPR Rotation Matrix

A YPR DCM can be derived by a sequential multiplication of the single-axis yaw, pitch, and roll rotation matrices.

$$R_g^b(\psi, \theta, \phi) = R_g^b(\phi) * \ R_g^b(\theta) * R_g^b(\psi)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} * \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} * \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

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

A.3 Euler Angles to and From Quaternion and DCM [32]

Euler to Quaternion:

$$\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{A}} = \boldsymbol{q}_{\boldsymbol{z}}(\psi) \otimes \boldsymbol{q}_{\boldsymbol{y}}(\theta) \otimes \boldsymbol{q}_{\boldsymbol{x}}(\phi) = \begin{bmatrix} \cos(\psi/2)\cos(\theta/2)\cos(\phi\backslash 2) + \sin(\psi/2)\,sin(\theta\backslash 2)sin(\phi/2) \\ \cos(\psi/2)\cos(\theta/2)\sin(\phi\backslash 2) - \sin(\psi/2)\,sin(\theta\backslash 2)cos(\phi/2) \\ \cos(\psi/2)\sin(\theta/2)\cos(\phi\backslash 2) + \sin(\psi/2)\,cos(\theta\backslash 2)sin(\phi/2) \\ \sin(\psi/2)\cos(\theta/2)\cos(\phi\backslash 2) - \sin(\psi/2)\,sin(\theta\backslash 2)cos(\phi/2) \end{bmatrix}$$

Quaternion to Euler:

$$\boldsymbol{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

$$\begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \tan^{-1}(2(q_0 q_3 + q_1 q_2)/\,(q_0^2 + q_1^2 - q_2^2 - q_3^2)\,) \\ \sin^{-1}(2(q_o q_2 - q_1 q_3) \\ \tan^{-1}(2(q_0 q_1 + q_2 q_3)/\,(q_0^2 - q_1^2 - q_2^2 + q_3^2)\,) \end{bmatrix}$$

YPR DCM to YRP Euler:

$$R_b^g(\psi,\theta,\phi) = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} =$$

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

$$\psi = \tan^{-1}(R_{21}/R_{11})$$

$$\theta = -\sin^{-1}(R_{31})$$

$$\phi = \tan^{-1}(R_{32}/R_{33})$$

A.4 Quaternion Operations

Skew Symmetric cross product matrix function [35]:

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$C(\boldsymbol{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

Quaternion Multiplication [32, 35]:

$$\boldsymbol{q_1} = \begin{bmatrix} q_{10} \\ \vec{\boldsymbol{q}}_{\boldsymbol{1}} \end{bmatrix} = \begin{bmatrix} q_{10} \\ q_{11} \\ q_{12} \\ q_{13} \end{bmatrix}$$

$$\boldsymbol{q_2} = \begin{bmatrix} q_{20} \\ \vec{\boldsymbol{q}}_{\boldsymbol{2}} \end{bmatrix} = \begin{bmatrix} q_{20} \\ q_{21} \\ q_{22} \\ q_{23} \end{bmatrix}$$

$$\boldsymbol{q_1} \otimes \boldsymbol{q_2} = \begin{bmatrix} q_{10} \\ \vec{\boldsymbol{q}}_{\boldsymbol{1}} \end{bmatrix} \otimes \begin{bmatrix} q_{20} \\ \vec{\boldsymbol{q}}_{\boldsymbol{2}} \end{bmatrix} = \begin{bmatrix} q_{20} & -\vec{\boldsymbol{q}}_{\boldsymbol{2}}^{\boldsymbol{T}} \\ \vec{\boldsymbol{q}}_{\boldsymbol{2}} & q_{20} I_{3x3} - C(\vec{\boldsymbol{q}}_{\boldsymbol{2}}) \end{bmatrix} \begin{bmatrix} q_{10} \\ \vec{\boldsymbol{q}}_{\boldsymbol{1}} \end{bmatrix} = \begin{bmatrix} q_{10} & -\vec{\boldsymbol{q}}_{\boldsymbol{1}}^{\boldsymbol{T}} \\ \vec{\boldsymbol{q}}_{\boldsymbol{1}} & q_{10} I_{3x3} - C(\vec{\boldsymbol{q}}_{\boldsymbol{1}}) \end{bmatrix} \begin{bmatrix} q_{20} \\ \vec{\boldsymbol{q}}_{\boldsymbol{2}} \end{bmatrix}$$

Quaternion Conjugate [32]:

$$\boldsymbol{q_a^b} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

$$\boldsymbol{q_a^b} = \boldsymbol{q_b^{a}}^{*} = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix}$$

Appendix B

Kalman Filter Equations

B.1 Kalman Filter Definitions

The Kalman Filter involves 8 matrices and vectors:

Process Transition Matrix:

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_{11} & \cdots & \phi_{1m} \\ \vdots & \ddots & \vdots \\ \phi_{n1} & \cdots & \phi_{nm} \end{bmatrix}$$

Process Noise Covariance Matrix:

$$\boldsymbol{Q} = \begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{bmatrix}$$

Measurement Transition Matrix:

$$\boldsymbol{H} = \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mn} \end{bmatrix}$$

Measurement Noise Covariance Matrix:

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{bmatrix}$$

Kalman Gain Matrix at Time k:

$$\boldsymbol{K_k} = \begin{bmatrix} k_{11} & \cdots & k_{1m} \\ \vdots & \ddots & \vdots \\ k_{n1} & \cdots & k_{nm} \end{bmatrix}$$

Error Covariance Matrix at Time k:

$$\boldsymbol{P_k} = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix}$$

State Vector at Time k:

$$\boldsymbol{x_k} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Measurement Vector at Time k:

$$\boldsymbol{y_k} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

The Kalman Filter is comprised of four steps, shown below [56]:

Process Update:

$$\boldsymbol{\hat{x}^{+}}_{k+1} = \boldsymbol{\Phi}\widehat{\boldsymbol{x_k}}$$

$$\boldsymbol{P}_{k+1}^{+} = \boldsymbol{\Phi}\boldsymbol{P}_k\boldsymbol{\Phi}^T + \boldsymbol{Q}$$

$$\boldsymbol{k} = \boldsymbol{k} + \boldsymbol{1}$$

Update Kalman Gain:

$$\boldsymbol{K}_k = \boldsymbol{P_k^{+}}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P_k^{+}}\boldsymbol{H}^T + \boldsymbol{R})^{-\boldsymbol{1}}$$

Measurement Update:

$$\boldsymbol{\hat{x}}_k = \boldsymbol{\hat{x}^{+}}_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{H}\boldsymbol{\hat{x}^{+}}_k)$$

Update Covariance:

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_k^{+}$$

B.2 Values Used in this Thesis

In this thesis, the following values were used to filter the plane roll and pitch from the ground orientation algorithm.

$$\boldsymbol{\Phi} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

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

$$\boldsymbol{R_{dynamic}} = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.9 \end{bmatrix}$$

$$\boldsymbol{R_{static}} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

$$\boldsymbol{K_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\boldsymbol{P_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Appendix C

Suspension Deflection Angles

Equations and process provided by Trenton Hilyer (Auburn University)

Given deflections for approximately the front left, front right, back left, and back right corners of the vehicle between body/chassis and axels s_1, s_2, s_3, s_4, and distances between the suspension deflection sensors front lateral (b_f), back lateral (b_b), left longitudinal (b_l), and right longitudinal (b_r), the roll and pitch of the body with respect to the axels can be found.

$$\phi_{front} = \operatorname{asin}\left(\frac{(s_1 - s_2)}{b_f}\right)$$

$$\phi_{\ back} = \operatorname{asin}\left(\frac{(s_3 - s_4)}{b_b}\right)$$

$$\theta_{right} = \operatorname{asin}\left(\frac{(s_1 - s_3)}{b_r}\right)$$

$$\theta_{left} = \operatorname{asin}\left(\frac{(s_2 - s_4)}{b_l}\right)$$

The overall orientation angles can be approximated by the average of the angles at the right and left and front and back:

$$\phi = (\phi_{front} + \phi_{back})/2$$
$$\theta = (\theta_{right} + \theta_{left})/2$$