

8-2018

## System Architecture, Calibration, and Control for LiDAR systems Onboard Unmanned Vehicles

Tamer Shamseldin  
*Purdue University*

Follow this and additional works at: [https://docs.lib.purdue.edu/open\\_access\\_dissertations](https://docs.lib.purdue.edu/open_access_dissertations)

---

### Recommended Citation

Shamseldin, Tamer, "System Architecture, Calibration, and Control for LiDAR systems Onboard Unmanned Vehicles" (2018). *Open Access Dissertations*. 2067.  
[https://docs.lib.purdue.edu/open\\_access\\_dissertations/2067](https://docs.lib.purdue.edu/open_access_dissertations/2067)

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.  
Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

SYSTEM ARCHITECTURE, CALIBRATION, AND CONTROL FOR LIDAR  
SYSTEMS ONBOARD UNMANNED VEHICLES

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Tamer Shamseldin

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2018

Purdue University

West Lafayette, Indiana

**THE PURDUE UNIVERSITY GRADUATE SCHOOL**  
**STATEMENT OF DISSERTATION APPROVAL**

Dr. Ayman Habib, Chair

Lyles School of Civil Engineering

Dr. Melba Crawford

Department of Agronomy, Schools of Civil Engineering and Electrical and  
Computer Engineering

Dr. James Bethel

Lyles School of Civil Engineering

Dr. Inseok Hwang

School of Aeronautics and Astronautics

**Approved by:**

Dr. Dulcy Abraham

Head of the School Graduate Program

TO my wife, kids, and my parents.



## ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Ayman Habib for all his invaluable guidance, encouragement, and support during my Ph.D. studies. He has inspired me to become an independent researcher and helped me to understand the power of critical reasoning. Without his valuable suggestions on my research and providing excellent study environment, this dissertation would not have been possible. Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Melba Crawford, Dr. James Bethel, and Dr. Inseok Hwang, for their insightful comments and encouragement, but also for the hard question which incited me to widen my research from various perspectives.

My gratitude also goes to my colleagues at the Digital Photogrammetry Research Group (DPRG), Magdy Elbahnasawy, Yun-Jou Lin, Mohammed Aldosari, Dr. Eui Myoung Kim, Ronald Benziger, Dr. Fang He, Dr. Rodrigo Gallis, Weifeng Xiong, Radhika Ravi, Dr. Zhang Quan, Megan Miller, Seyyed Meghdad, Tian Zhou, and Lisa Laforest. I would also like to appreciate all the help from my friends, Ankit Manerikar, Ali Masjedi, and Behrokh Nazery.

I deeply thank my dear parents for their unconditional trust, timely encouragement, and endless patience. Please keep praying for me. I love you and thank you for everything. Furthermore, I would like to thank my wife, Nermeen Kassem and my two sons Youssef and Omar. You are the best thing happen in my life, may ALLAH bless you all.

Finally I would like to thank my country, EGYPT, for funding and supporting my research, specially the Egyptian Armed Forces, my sponsor, for the unwavering support.

# TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	x
ABBREVIATIONS . . . . .	xvii
ABSTRACT . . . . .	xix
1 INTRODUCTION . . . . .	1
1.1 Background . . . . .	1
1.2 Motivation . . . . .	3
1.3 Problem Statement and Research Challenges . . . . .	6
1.4 Research Objectives . . . . .	7
1.5 Dissertation Outline . . . . .	9
2 RELATED WORK . . . . .	10
2.1 Overview . . . . .	10
2.2 System Integration of MMS Framework . . . . .	10
2.2.1 Unmanned Aerial Vehicles (UAVs) . . . . .	11
2.2.2 Onboard Mapping Sensors (Laser Scanners) . . . . .	13
2.2.3 UAV-based LiDAR systems . . . . .	19
2.3 LiDAR System Calibration . . . . .	25
2.3.1 Different Calibration Techniques for Eliminating the Impact of the Systematic Error in the LiDAR System Parameters . . . . .	26
2.3.2 The Necessary Flight and Target Configuration Requirements for LiDAR System Calibration . . . . .	31
2.4 SLAM-Assisted Coverage Path Planning and Implementation of Pseudo GNSS/INS Localization System for GNSS-denied Environment . . . . .	34
3 DEVELOPMENT OF SYSTEM ARCHITECTURE FOR A LIDAR-BASED AIRBORNE MOBILE MAPPING SYSTEM . . . . .	40

	Page
3.1 Introduction . . . . .	40
3.2 UAV Mobile Mapping System Framework . . . . .	42
3.2.1 Mapping System Platform . . . . .	43
3.2.2 Direct Geo- referencing Module . . . . .	45
3.2.3 Active Sensor (Laser Scanner) . . . . .	49
3.3 System Integration of UAV-based LiDAR system . . . . .	52
3.3.1 Synchronization Process . . . . .	54
3.3.2 Data Storage Unit . . . . .	57
3.3.3 System Interference . . . . .	58
3.4 System Integration of Wheel-based LiDAR system . . . . .	62
3.5 Point-Cloud Reconstruction . . . . .	70
3.6 Summary . . . . .	74
4 BIAS IMPACT ANALYSIS AND LiDAR SYSTEM CALIBRATION . . . .	76
4.1 Introduction . . . . .	76
4.2 Bias Impact Analysis . . . . .	78
4.2.1 Mathematical Model for a UAV-based LiDAR Point Positioning	78
4.2.2 Bias Impact Analysis for a Spinning Multi-Beam Laser Scanner	80
4.2.3 Optimal Flight Line Configuration for Calibration Process . . .	92
4.2.4 General Mathematical Model for LiDAR System Calibration . .	93
4.3 Feature Extraction . . . . .	95
4.4 Pseudo-Conjugate Points in Overlapping Strips and Weight Modifica- tion Procedure . . . . .	98
4.5 Experimental Results . . . . .	101
4.5.1 Single Flying Height . . . . .	103
4.5.2 Multiple Flying Heights . . . . .	106
4.6 Summary . . . . .	111
5 SLAM-ASSISTED COVERAGE PATH PLANNING AND IMPLEMEN- TATION OF PSEUDO-GNSS/INS LOCALIZATION SYSTEM FOR IN- DOOR LIDAR MMS . . . . .	112

	Page
5.1 Introduction . . . . .	112
5.2 SLAM-assisted Coverage Path Planning . . . . .	112
5.2.1 Overview . . . . .	112
5.2.2 Hybrid Approach Implementation . . . . .	113
5.2.3 Offline Planning - CPP Strategy . . . . .	115
5.2.4 Real-time SLAM-assisted CPP . . . . .	119
5.2.5 System Implementation . . . . .	120
5.2.6 Experimental Results . . . . .	121
5.3 Implementation of Pseudo-GNSS/INS Localization System for Indoor LiDAR MMS . . . . .	124
5.3.1 SLAM-based MMS for GNSS-denied Environments . . . . .	125
5.3.2 Proposed Methodology of a Pseudo-GNSS/INS Module . . . . .	126
5.3.3 Hardware System Implementation . . . . .	131
5.3.4 Operational Strategy . . . . .	132
5.3.5 Post-Processing Enhancement . . . . .	133
5.3.6 Experimental Results . . . . .	141
5.3.7 Summary . . . . .	163
6 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH	165
6.1 Summary of Contributions . . . . .	165
6.2 Recommendations for Future Work . . . . .	167
REFERENCES . . . . .	169
VITA . . . . .	179

## LIST OF TABLES

Table	Page
3.1 The DJI S1000+ UAV Specifications . . . . .	44
3.2 The DJI Matrice Pro (M600) UAV Specifications . . . . .	45
3.3 Trimble APX-15 UAV (V2) Specifications . . . . .	47
3.4 Novatel SPAN-IGM-S1 Specifications . . . . .	48
3.5 The Specifications of the Velodyne HDL-32E . . . . .	50
3.6 The Specifications of the Velodyne VLP-16 Puck/ Puck Hi-Res . . . . .	51
3.7 The Trimble POSLV-125 Specifications . . . . .	63
3.8 The Novatel SPAN-CPT Specifications . . . . .	66
3.9 The Trimble POSLV-220 Specifications . . . . .	68
4.1 Impact of bias in each of the mounting parameters on 3D point coordinates	83
4.2 Mounting parameters of VLP-16 before and after calibration test 1 . . . .	104
4.3 Correlation matrix of mounting parameter estimates from calibration test	1104
4.4 Mounting parameters of VLP-16 before and after calibration test 2 . . . .	105
4.5 Correlation matrix of mounting parameter estimates from calibration test	2105
4.6 Mounting parameters of VLP-16 before and after calibration test 3 . . . .	106
4.7 Correlation matrix of mounting parameter estimates from calibration test	3107
4.8 Calibration of the UAV system: Square root of the a-posteriori variance factor . . . . .	108
4.9 RMSE of plane/line fitting for different configurations (Ground Patches, Rooftop, Building Facade) . . . . .	108
4.10 RMSE of plane/line fitting for different configurations (Reflective Boards)	109
4.11 RMSE of plane/line fitting for different configurations (Hut Surfaces) . .	110
4.12 RMSE of plane/line fitting for different configurations (Hut Ridges) . . .	110
5.1 The details for the data acquisition for the two locations . . . . .	141

Table	Page
5.2 The quantitative analysis (RMSE and ND) of plane (1) . . . . .	162
5.3 The quantitative analysis (RMSE and ND) of plane (2) . . . . .	162
5.4 The quantitative analysis (RMSE and ND) of plane (3) . . . . .	162
5.5 The quantitative analysis (RMSE and ND) of plane (4) . . . . .	163

## LIST OF FIGURES

Figure	Page
2.1 Examples of different UAVs platforms: (a) eBee SenseFly SA (an example of fixed-wing aircraft), and (b) DJI Inspire 2 (an example of rotorcraft) . .	12
2.2 Examples of multi-beam laser scanner sensors, (a) Velodyne VLP-16 Puck, and (b) Quanergy M8-1 LiDAR . . . . .	16
2.3 ASC's Tiger-Eye 3D Flash LiDAR Camera [24] . . . . .	17
2.4 Examples of different laser scanning systems: (a) Psomas Mobile Mapper system (an example of terrestrial mobile laser system), and (b) RIEGL VQ-1560-DW (an example of airborne laser system) [34,35] . . . . .	18
2.5 Scanfly Airborne mapping system . . . . .	21
2.6 NEXUS 800 UAV mapping system . . . . .	22
2.7 MiniVUX-1UAV mapping system . . . . .	22
2.8 Routescene airborne mapping System . . . . .	23
2.9 Sky-Scanner LiDAR/image data collection system . . . . .	23
2.10 YellowScan mobile mapping System . . . . .	24
2.11 The recommended flight configuration for LiDAR system calibration by Burman [61] . . . . .	32
2.12 The recommended flight configuration for LiDAR system calibration by Morin [62] . . . . .	33
2.13 The optimal flight and target configuration for airborne LiDAR system calibration by Kersting et al. [65] . . . . .	33
2.14 The boustrophedon motion . . . . .	35
2.15 The square spiral motion [80] . . . . .	35
2.16 The trapezoidal cellular decomposition of the target environment [10] . . .	37
2.17 The boustrophedon cellular decomposition of the target environment [10] .	37
3.1 A functional block diagram of a GNSS/INS-based MMS framework with N sensors . . . . .	40

Figure	Page
3.2 The DJI S1000+ platform for mapping applications . . . . .	43
3.3 The DJI Matrice 600 Pro (M600) platform for mapping applications . . . .	44
3.4 Direct Geo-referencing of LiDAR data using a GNSS/INS module . . . . .	46
3.5 Trimble APX-15 UAV (V2) GNSS/INS unit . . . . .	46
3.6 Novatel SPAN-IGM-S1 GNSS/INS unit . . . . .	48
3.7 The Velodyne HDL-32E sensor . . . . .	49
3.8 The Velodyne VLP-16 Puck/ Puck Hi-Res sensor . . . . .	51
3.9 The DJI M600 Pro based LiDAR mapping system (Velodyne VLP-16-Puck Hi-Res) . . . . .	52
3.10 The DJI S1000+ based LiDAR mapping systems – (a) The Velodyne VLP- 16 Puck -(b) The Velodyne HDL-32E . . . . .	53
3.11 Synchronization process of LiDAR-based mapping system . . . . .	54
3.12 Integration scheme for the DJI M600 Pro UAV-based system . . . . .	55
3.13 The Velodyne webpage application for checking the reception of the PPS signal and the GNRMC message . . . . .	56
3.14 The Wireshark software to ensure the reception of GPRMC message . . . .	56
3.15 Data storage unit onboard UAV-LiDAR system . . . . .	57
3.16 Physical installation of several sensors onboard DJI S1000+ (a) Side view- (b) Top view . . . . .	59
3.17 Physical installation of several sensors onboard DJI M600 Pro (a) Side view- (b) Top view . . . . .	60
3.18 The GNSS antenna mast design (DJI S1000+) – (a) Initial design -(b) New design . . . . .	61
3.19 DJI M600 Pro based LiDAR mapping system – (a) Old installation -(b) New housing box . . . . .	62
3.20 The Trimble POSLV-125 GNSS/INS Module . . . . .	63
3.21 PhenoRover-based MMS-(a)Old boom design-(b)New boom design . . . .	64
3.22 The car-mount mapping system (Portable system) . . . . .	65
3.23 The Novatel SPAN-CPT GNSS/INS unit . . . . .	65
3.24 Purdue Wheel-based mobile mapping System (PWMMS) . . . . .	67



Figure	Page
3.25 The Trimble POSLV-220 GNSS/INS Module . . . . .	67
3.26 Integration scheme for the wheel-based LiDAR systems . . . . .	69
3.27 Integration of multi-LiDAR sensors to single PC through multiple Ethernet ports . . . . .	69
3.28 Integration of multiple-LiDAR sensors with GNSS/INS module through embedded system electronic kits . . . . .	70
3.29 Illustration of the GNSS/INS data interpolation . . . . .	71
3.30 Spherical Linear Interpolation Method . . . . .	72
3.31 Interpolated quaternion $q_i$ computation based on the $C_1$ derivation . . . .	73
3.32 Interpolated quaternion $q_i$ computation based on the $C_2$ derivation . . . .	74
4.1 Conceptual basis of the LiDAR system calibration approach . . . . .	77
4.2 Illustration of point positioning of a Light Detection and Ranging (LiDAR) system . . . . .	79
4.3 Illustration of a spinning multi-beam LiDAR unit (VLP-16 Puck Hi-Res) .	79
4.4 Coordinate systems for DJI M600 Pro-based LiDAR system . . . . .	81
4.5 Schematic diagram illustrating the symbolic notations used for a UAV-based LiDAR system calibration . . . . .	82
4.6 Impact of bias in lever-arm component across the flying direction ( $\Delta X$ ) . .	84
4.7 Impact of bias in lever-arm component along the flying direction ( $\Delta Y$ ) . .	85
4.8 Impact of bias in lever-arm component in the vertical direction ( $\Delta Z$ ) . . .	85
4.9 Impact of bias in boresight pitch ( $\Delta\omega$ ) . . . . .	86
4.10 Impact of bias in boresight roll ( $\Delta\phi$ ) . . . . .	88
4.11 Relationship between $x'$ -coordinates for: (a) two tracks in the same direction, and (b) two tracks in opposite directions. . . . .	89
4.12 Impact of bias in boresight roll across the flying direction ( $\Delta\kappa$ ) . . . . .	90
4.13 Impact of bias in boresight roll along the flying direction ( $\Delta\kappa$ ) . . . . .	91
4.14 Optimal/Minimal flight configuration for mounting parameters estimation for a UAV-based LiDAR system. . . . .	93
4.15 Optimal feature configuration for mounting parameters estimation for a UAV-based LiDAR system . . . . .	97

Figure	Page
4.16 Intensity data from a point cloud that includes highly reflective boards highlighted by the zoomed-in areas . . . . .	98
4.17 Discrepancy vector between non-conjugate points along corresponding (a) planar, and (b) linear features . . . . .	99
4.18 Illustration of local coordinate systems for (a) planar and (b) linear features	100
4.19 Configuration of flight lines and target primitives used for calibration as visible in 3D point clouds and RGB orthophoto . . . . .	102
4.20 Actual alignment of coordinate systems for the UAV LiDAR system used in this study . . . . .	103
4.21 Qualitative evaluation of calibration targets before and after calibration: (a) Checkerboard, (b) hut-shaped target, (c) building facade, and (d) ground patch . . . . .	107
5.1 Functional block diagram for coverage-based SLAM implementation . . .	114
5.2 Example of a simulated data: (a) Desired area , (b) Polygon triangulation technique, and (c) Optimum Polygon triangulation . . . . .	116
5.3 Boustrophedon Pattern for Coverage . . . . .	117
5.4 System Implementation using Roomba Create 2 . . . . .	121
5.5 Results for SLAM-assisted CPP framework: (a) Original Area, (b) Total Coverage Path Generated, and (c) Robot Path during Navigation . . . .	122
5.6 Loop Closure Error with SLAM implementation . . . . .	123
5.7 Loop Closure Error without SLAM implementation (Odometer) . . . . .	123
5.8 Functional block diagram of the proposed Pseudo-GNSS/INS framework for an Indoor MMS . . . . .	127
5.9 Real-time SLAM for MMS positioning, extraction of a 2D-scan from the full 3D-LiDAR point cloud . . . . .	129
5.10 Visualization of the windowing operation for extracting the 2D-point cloud	130
5.11 Pseudo-GNSS/INS framework geo-referencing signals generation . . . . .	131
5.12 Implemented Indoor MMS with Pseudo-GNSS/INS framework . . . . .	132
5.13 The demonstration of ICP registration technique [135] . . . . .	134
5.14 The relative translation and orientation parameters between each two epochs through the segment . . . . .	136

Figure	Page
5.15 The derivation of stationary scan locations using concatenation of motion estimation . . . . .	137
5.16 The derivation of the backward trajectory . . . . .	138
5.17 The computation of the time difference between the time of the trusted location and the time of any epoch through the trajectory either forward or backward trajectory . . . . .	139
5.18 The determination of orientation of the smoothed trajectory by using spherical linear interpolation . . . . .	140
5.19 The derivation of the smoothed trajectory . . . . .	140
5.20 The results for MMS Mapping Operation for location I (a) Robot trajectory obtained from the Pseudo-GNSS/INS position data, (b) 3D point cloud obtained for the entire scans, and (c) Final reconstructed map after applying ICP registration for the stationary scan locations . . . . .	142
5.21 The results for MMS mapping operation for location II (a) Original map for Mann Hall and (b) Robot trajectory obtained from the Pseudo-GNSS/INS position data . . . . .	143
5.22 The results for MMS mapping operation for location II (a) 3D-point cloud obtained for the entire scans, (Range = 70 m) (b) 3D-point cloud obtained for the entire scans, (Range = 5 m), (c) Final reconstructed map after applying ICP registration for the stationary scan locations, and (d) Reconstructed map over original map . . . . .	144
5.23 The position of the UGV platform through different trajectories for the first segment . . . . .	145
5.24 The position of the UGV platform through different trajectories for the second segment . . . . .	146
5.25 The position of the UGV platform through different trajectories for the third segment . . . . .	146
5.26 The heading of the UGV platform through different trajectories for the first segment . . . . .	147
5.27 The pitch of the UGV platform through different trajectories for the first segment . . . . .	147
5.28 The roll of the UGV platform through different trajectories for the first segment . . . . .	148
5.29 The heading of the UGV platform through different trajectories for the second segment . . . . .	148

Figure	Page
5.30 The pitch of the UGV platform through different trajectories for the second segment . . . . .	149
5.31 The roll of the UGV platform through different trajectories for the second segment . . . . .	149
5.32 The heading of the UGV platform through different trajectories for the third segment . . . . .	150
5.33 The pitch of the UGV platform through different trajectories for the third segment . . . . .	150
5.34 The roll of the UGV platform through different trajectories for the third segment . . . . .	151
5.35 The difference in x-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the first segment . . . . .	152
5.36 The difference in y-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the first segment . . . . .	152
5.37 The difference in the pitch angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the first segment . . . . .	153
5.38 The difference in the roll angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the first segment . . . . .	153
5.39 The difference in the heading angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the first segment . . . . .	154
5.40 The difference in x-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the second segment . . . . .	154
5.41 The difference in y-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the second segment . . . . .	155
5.42 The difference in the pitch angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the second segment . . . . .	155

Figure	Page
5.43 The difference in the roll angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the second segment . . . . .	156
5.44 The difference in the heading angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the second segment . . . . .	156
5.45 The difference in x-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the third segment . . . . .	157
5.46 The difference in y-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the third segment . . . . .	157
5.47 The difference in the pitch angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the third segment . . . . .	158
5.48 The difference in the roll angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the third segment . . . . .	158
5.49 The difference in the heading angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the third segment . . . . .	159
5.50 The 3D-point cloud reconstruction (only three segments of the entire trajectory) using the real-time SLAM-based trajectory . . . . .	160
5.51 The 3D-point cloud reconstruction (only three segments of the entire trajectory) using the generated smoothed trajectory . . . . .	161
5.52 The 3D-point cloud reconstruction (only three segments of the entire trajectory) using the ICPP-based stationary scan locations . . . . .	161

## ABBREVIATIONS

ALS	Airborne Laser System
BCD	Boustrophedon Cellular Decomposition
CPP	Coverage Path Planning
CW	Continuous Wave
DBM	Digital Building Model
DMI	Distance Measurement Instrument
DPRG	Digital Photogrammetry Research Group
EMC	Electro-Magnetic Compatibility
FOV	Field of View
GCP	Ground Control Point
GNSS	Global Navigation Satellite System
GPRMC	Recommended minimum specific GPS/Transit data
ICPP	Iterative Closest Projected Point
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LiDAR	Light Detection and Ranging
LRF	Laser Range Finder
LSA	Least Squares Adjustment
MMS	Mobile Mapping System
MSA	Minimum Sum of Altitudes
N-S	North to South
PC	Personal Computer
PPS	Pulse Per Second
PWMMS	Purdue Wheel based-Mobile Mapping System

RMSE	Root-Mean-Square Error
ROS	Robot Operating System
RSS	Remote Sensing Sensors
RTK	Real Time Kinematic
SLAM	Simultaneous Localization And Mapping
S-N	South to North
TIN	Triangular Irregular Network
TMLS	Terrestrial Mobile Laser Scanner
TOF	Time Of Flight
UDP	User Datagram Protocol
UGV	Unmanned Ground Vehicle
UAV	Unmanned Aerial Vehicle

## ABSTRACT

Shamseldin, Tamer Ph.D., Purdue University, August 2018. System Architecture, Calibration, and Control for LiDAR systems Onboard Unmanned Vehicles. Major Professor: Ayman Habib.

The development of Unmanned Aerial Vehicles (UAVs) as a mobile platform for deploying portable systems has benefited several applications in the civilian and military fields in the past few decades. Also, the parallel advances in peripheral technology such as the enhancement of GNSS/INS modules have resulted in a remarkable development of both UAVs as well as UAVs applications. Such development leads to the establishment of a system architecture for UAV design that will be the basis of all discussion in this dissertation. Centrally, this dissertation introduces a generic framework for UAVs equipped with a GNSS/INS positioning and orientation module as well as low-cost LiDAR sensors for targeted mapping and monitoring applications. An essential aspect of this research proposes a LiDAR system calibration procedure for a mobile airborne platform. Such a calibration procedure can directly estimate the mounting parameters relating the laser scanners to the onboard GNSS/INS unit, i.e., the lever-arm and boresight angles for a LiDAR unit through an outdoor calibration procedure. This approach is based on the use of conjugate planar/linear features in overlapping point clouds derived from different flight lines. Furthermore, an optimal configuration of target primitives and flight lines is determined by analyzing the potential impact of bias in mounting parameters of a LiDAR unit. To add a degree of autonomy to this integrated framework, a Coverage Path Planning (CPP) approach is proposed. Such approach is performed to achieve complete coverage of the area of interest in a minimum time with the aid of real-time Simultaneous Localization and Mapping (SLAM) technique. The successful implementation of SLAM with this



integrated framework furthermore offers insight into extending the system in conditions where one of the subsystems may not function properly. For example, in GNSS-denied environments, the GNSS/INS modules fail to work correctly due to the absence of consistent GNSS signals. This dissertation introduces a Pseudo-GNSS/INS integrated framework that is implemented using probabilistic SLAM techniques. Such a framework allows for the extension of the operation of such systems for GNSS-denied environments and hence is a significant contribution towards increasing robustness and autonomy in terrestrial/aerial mapping systems.

# 1. INTRODUCTION

## 1.1 Background

Over the last decade, the mapping community has witnessed several developments that had a significant impact on facilitating comprehensive 3D mapping of our environment with better accuracy, and at lower cost such as: (a) Significant improvement in the direct geo-referencing technology and the ensuing ability of substantial reduction of control requirements, (b) emergence of low-cost digital cameras as a viable mapping tool in airborne and close range Photogrammetry, (c) widespread acceptance and adoption of Light Detection and Ranging (LiDAR) systems on-board terrestrial and airborne platforms for direct acquisition of surface information, and d) integration of image and LiDAR data for 3D modeling and visualization applications. Coupled with these technical advances, there has been an expansion in the user sector of mapping products from provincial, state, and Federal organizations who are mainly interested in large-area mapping (e.g., National mapping, forest inventory, Glaciology, and 3D city modeling), to new users. Such new users are interested in more-detailed large-scale mapping (e.g., Archaeological documentation, pipeline inspection, landslide hazard analysis, 3D modeling of individual buildings and objects, infrastructure inventory and monitoring, indoor mapping for building information management, and open pit mining). It has become evident that non-traditional mapping platforms are needed to address the requirements of this broad user base. In response to these needs, Mobile Mapping Systems (MMS) have emerged as a viable tool, and they have changed the paradigm of the mapping process.

The MMS is a multi-task system that usually comprises: (i) a platform and power supply; (ii) a control module; (iii) an imaging module; (iv) a positioning and orientation module; and (v) a data processing module. The kinematic platform can be a land

vehicle, a backpack carried by a human operator, an air vehicle, or a marine vehicle, either manned or unmanned, that provides sufficient power supply for mission operation. The control module is responsible for data acquisition based on time or distance interval. The imaging module could include video cameras, digital cameras, and/or laser scanners. The positioning and orientation module is the most expensive component and most crucial for the determination of the geographic location of the ground objects. It encompasses a Global Navigation Satellite System (GNSS) receiver, an inertial measurement unit (IMU), and/or a distance measurement instrument (DMI).

One should note that there is a significant impact of MMS on manned terrestrial and airborne platforms regarding the quality of the mapping product in the absence of traditional ground control points. However, the initial investment, mobilization cost, and the demand for the end users to have a high level of technical expertise are preventing the widespread adoption of this technology by potential individuals who might benefit from such technology. Coupled with such issues, the need for an accurate 3D reconstruction of our environment has become essential for non-traditional mapping applications and cannot be satisfied by traditional mapping that is based on dedicated data acquisition systems which are designed for mapping purposes. Recent advances in hardware and software development have made it possible to conduct accurate 3D mapping without using costly and high-end data acquisition systems. For example, low-cost laser scanners and navigation systems can provide accurate mapping if they are adequately integrated at the hardware and software levels. Moreover, the impressive developments in the mobile mapping technology (i.e., mapping using acquired data from sensors onboard a mobile platform) have made accurate 3D mapping more feasible whenever and wherever needed. In this regard, UAV-based mapping systems proved to be capable of providing high-quality mapping products while bridging a significant gap, regarding the extent of the area to be mapped as well as accessibility constraints between traditional terrestrial and airborne applications [1]. More specifically, UAV-based mapping can be used for a wide variety of applications while combining the advantages of both traditional airborne and terres-

trial mapping systems. Therefore, this dissertation will focus on the development of system architecture for a LiDAR-based UAV mapping system and deriving accurate geospatial information from LiDAR data by finding out an accurate and practical LiDAR system calibration technique as well as performing a 3D reconstruction of a GNSS-denied environment based on real-time Simultaneous Localization and Mapping (SLAM) technique using the onboard 3D-LiDAR sensor.

## 1.2 Motivation

In the past few decades, it was evident that traditional mapping activities could not meet the demand of emerging mapping applications due to limited financial and technical resources. The development of terrestrial and airborne MMS has been mainly motivated by the needs of traditional and new applications. However, MMS are quite expensive and cannot be rapidly and efficiently deployed. Fortunately, UAVs are evolving as a promising geospatial data acquisition system that could satisfy the needs of the same mapping applications and overcome MMS limitations [2,3]. This promise is mainly attributed to advances in low-cost direct geo-referencing systems as well as imaging sensors operating at different portions of the electromagnetic spectrum. Despite their military origins, UAVs are now seeing expanded use in commercial and civil applications, including precision agriculture, air quality monitoring, pipeline inspection, utility management, mapping, surveillance, and hazard assessment. Compared to human-operated systems, the main advantages of UAVs include the following facts: (a) they can be cost-effectively stored and deployed, which make them optimal for rapid response applications; (b) they can fly at lower elevation and slower speed than manned aircrafts, thus providing high-quality spatial data; (c) they can perform missions and acquire data autonomously so that human interaction is minimized, thus reducing pilot's exposure to risk; (d) they are highly maneuverable, which is ideal for low altitude flying and complex environments; and (e) they can operate in dangerous environments. These characteristics make UAVs an optimal

platform for affordable rapid-response mapping applications [4, 5]. The realization of these benefits is the main reason behind the current and future growth in UAV production and applications.

UAVs, which are intended for mapping purposes, are equipped with passive sensors (i.e., medium-format digital cameras) and/or active sensors such as LiDAR units. One should note that a great deal of research has been geared towards the development of camera-based UAV mapping systems. However, LiDAR-based UAV mapping is still a new area of research. LiDAR sensors onboard airborne and terrestrial platforms have been established as a proven technology for the derivation of dense point clouds with high positional accuracy. The main factors behind the widespread use of LiDAR systems include the ever-continuous improvement in GNSS/INS direct geo-referencing technology as well as enhanced performance and reduced size and cost of LiDAR units. The ability to derive accurate geospatial information from UAVs is contingent on having a precise position and orientation of the platform, which is usually based on an integrated GNSS/INS unit. It is worth noting that UAV payload restrictions might enforce the use of consumer-grade sensors, which in turn will negatively affect the quality of the final product. Therefore, research efforts are needed to address the challenges arising from the use of lower-quality mapping sensors while maintaining the quality of the mapping outcome. The balance between the UAV-payload capacity and the required accuracy for the final product is considered one of the key challenges for the UAV-based mapping system.

A single/multi-unit LiDAR system calibration technique is needed to perform in order to derive point clouds with high positional accuracy. Such calibration procedures can be used to directly estimate the mounting parameters relating a LiDAR sensor to the onboard GNSS/INS unit. To be more specific, a LiDAR system calibration entails the estimation of the intrinsic parameters of the individual scanners as well as the mounting parameters that define the spatial relationship between such sensors (e.g., the mounting parameters relating the laser scanners to each other as well as the mounting parameters relating these sensors to the onboard navigation

unit). Therefore, a LiDAR system calibration is considered an essential and critical task to ensure the attainment of the prospective accuracy.

LiDAR system calibration process is usually carried out in several steps: (i) laboratory calibration, (ii) platform calibration, and (iii) in-situ calibration. Laboratory calibration is performed by the system manufacturer during which all the individual system components are calibrated (e.g., range offset, mirror angle scale, or relative alignment of multi-laser systems) [6]. Then, the platform calibration which is conducted by the data provider determines the mounting parameters between the LiDAR reference point and the GNSS antenna. Since the parameters determined in the laboratory and platform calibration steps might be biased and/or not stable over time, an in-situ calibration should be carried out to refine such parameters.

As far as deriving accurate geospatial information from the LiDAR data is concerned, reliable LiDAR system calibration procedures need to distinguish the suitable primitives. More specifically, calibration primitives are considered the features that will be used for identifying the discrepancies between overlapping LiDAR strips and the control surface. Due to the irregular nature of the LiDAR points, the identification of distinct points is quite difficult and not reliable [7]. Therefore, a new calibration procedure based on the use of different types of geometric features (e.g., linear and planar) should be used. In this regard, the focus of the LiDAR system calibration is to simultaneously estimate the mounting parameters relating the different system components by minimizing the discrepancy between conjugate linear and/or planar features in overlapping point clouds derived from different flight lines. Such methodology is considered one of the objectives of this dissertation.

As an extension to the research area, for some mapping applications, there is a significant need to survey a particular area of interest [8, 9]. Then, the complete area coverage should be obtained in a minimum time due to UAV's constraints (i.e., payload restriction, endurance limit, robustness, and weather circumstances). Therefore, designing an intelligent path planning by performing CPP techniques can help to figure out the optimal coverage path. Such a path will attain the maximum area

coverage in a minimum time since there are many applications which hold this constraint in higher regard [10, 11]. Using such techniques for indoor/outdoor mapping are massively sensitive to odometry errors. So, the heading angle deviation error is unavoidable and can lead to an accumulating drift in navigation measurements. Therefore, utilizing one of the probabilistic techniques such as SLAM can achieve significant improvement in the system operation. It is worth noting that such technique is considered an efficient choice for some applications where the availability of a GNSS signal is scarce or non-existent.

It is quite challenging to extend the MMS to GNSS-denied areas, such as indoor environments. The deployment of these systems is restricted to applications and environments where a consistent availability of GNSS signals is assured. One should note that the positioning and orientation module is considered to be a crucial component of the MMS skeleton. Therefore, MMS usage for applications within GNSS-denied areas necessitates the development of an alternative module [12]. Such a module can act as a viable substitute to the GNSS/INS unit for system operation without having to resort to an exhaustive modification of the current MMS to operate in GNSS-denied locations. The implementation of such a framework is another objective of this dissertation.

### **1.3 Problem Statement and Research Challenges**

To date, a great deal of research has been conducted on 3D mapping systems that have been traditionally established using passive imaging systems onboard UAVs. The utilization of active sensors such as laser scanners for UAVs-based mapping is considered a new research area. However, to satisfy the needs of various mapping applications, some challenges have to be addressed. The main challenges that are faced by the research community in this regard and hence are a topic of substantial interest include:

i) Optimization of sensor selection and placement: limitations on the UAV payload enforce constraints on the selection of both the GNSS/INS positioning and orientation system as well as the LiDAR unit.

ii) LiDAR system calibration and bias impact analysis: calibrating a directly georeferenced laser scanning unit to ensure the accuracy of the generated point cloud is a major concern for the UAV-based LiDAR system.

iii) Robust system control: development of a reliable control technique for the platform is desirable for applications that require continuous monitoring of specific objects (i.e., system control to maintain the desired object within the field of view of the LiDAR unit).

iv) 3D-map reconstruction in GNSS-denied environment: development of a Pseudo-GNSS/INS framework is needed to act as a viable substitute to GNSS/INS units for MMS operation without having to resort to a comprehensive modification of the current system integration in order to work in GNSS-denied areas.

## 1.4 Research Objectives

As discussed above, the primary objective of this dissertation is to develop a generic framework for unmanned vehicles that are equipped with a GNSS/INS positioning and orientation module as well as low-cost laser scanners for targeted mapping and monitoring applications. Furthermore, in this research work, a Pseudo GNSS/INS integrated framework onboard Unmanned Ground Vehicle (UGV) will be developed to allow for operation within GNSS-denied environments. The following sub-objectives are proposed to accomplish the overall objective and establish the new framework:

- Propose the development of a system architecture for a LiDAR-based UAV mapping system as follows:
  - selects the appropriate platforms as well as the sensors for the mapping purpose, and



- performs the time synchronization process between the proposed sensors, since such process plays a prominent role in creating accurate geo-referencing information.
- Propose a LiDAR system calibration and bias impact analysis, which:
  - analyzes the potential impact of bias in mounting parameters of a LiDAR unit on the resultant point cloud for different orientations of target primitives and different flight lines configurations to increase the accuracy of the estimated mounting parameters, and
  - develops an accurate system calibration, which entails the estimation of the intrinsic parameters/characteristics of the scanner as well as the mounting parameters that relate the scanner to the GNSS/INS unit.
- Propose a coverage path planning method, which:
  - establishes a degree of autonomy in the system to allow for autonomous path generation and navigation for a given area to be mapped, and
  - performs a probabilistic SLAM framework to allow for dynamic update of the map as well as the platform position and heading during the mission operation.
- Propose an implementation of a SLAM-based Pseudo-GNSS/INS localization system for indoor LiDAR MMS framework, which:
  - generates real-time robot pose estimates based on online SLAM technique, and
  - performs the entire operation of the MMS by using a single 3D LiDAR sensor. Such operation incorporates the positioning module and the mapping sensor.

## 1.5 Dissertation Outline

The remainder of this dissertation explains the proposed strategies in more detail:

- Existing pertinent literature for both 3D reconstruction using active sensors in GNSS-affluent/-denied environment and LiDAR system calibration is reviewed in Chapter 2.
- The proposed strategy for the development of the system architecture for a LiDAR-based UAV mapping system is introduced in Chapter 3.
- The proposed strategy for LiDAR system calibration and bias impact analysis is explained in Chapter 4.
- The proposed strategy for performing the maximum area coverage using CPP techniques as well as the implementation of SLAM-based Pseudo-GNSS/INS integrated framework to operate in GNSS-denied environments is discussed in Chapter 5.
- Finally, the key contributions of the dissertation are summarized, and recommendations for future work are provided in Chapter 6.

## 2. RELATED WORK

### 2.1 Overview

As it has been introduced in Chapter 1, the primary objective of this research is to develop a generic framework for unmanned vehicles-based MMS. Such MMS are equipped with either single or multiple LiDAR sensors integrated with a GNSS/INS positioning and orientation module to derive an accurate 3D reconstruction of the area of interest. Furthermore, in this research work, a pseudo GNSS/INS integrated framework will be developed to extend the MMS framework for operation within GNSS-denied environments. Therefore, existing pertinent literature for laser scanning principles and the existing approaches for both 3D reconstruction using active sensors and LiDAR system calibration are discussed in this chapter. Specifically, a review of the existing mapping systems for LiDAR-based 3D reconstruction using single/multi-LiDAR systems are presented in Section 2.2. Then, several research activities related to LiDAR system calibration for airborne/terrestrial mapping systems are introduced in Section 2.3. Also, the coverage path planning techniques for unmanned vehicles and SLAM-based mapping for GNSS-denied environments are explained in Section 2.4 and Section 2.5, respectively.

### 2.2 System Integration of MMS Framework

The hardware system implementation of the MMS framework which concerns the platforms and sensors selection as well as the synchronization process between the mapping sensors is considered one of the necessary steps to build a reliable MMS. Therefore, this part of the dissertation will discuss the literature review of the platforms, onboard mapping sensors, and the development of MMS.

### 2.2.1 Unmanned Aerial Vehicles (UAVs)

To date, with the numerous technological developments in MMS, several platforms such as mobile sensor systems onboard tractors, tethered balloons, and manned aircraft have become attractive options [13]. However, some financial and technical constraints (e.g., the initial investment, mobilization cost, and the required technical expertise of end users) prevent the widespread adoption of such human-operated systems. In response to that, UAVs have recently emerged as a promising platform for mapping and remote-sensing data acquisition to fulfill the needs of diverse applications [14–16].

UAVs are motorized aircraft that can be autonomously flown or remotely controlled from the ground and are designed to fly without a human pilot onboard. UAVs were initially motivated by military goals and applications such as surveillance, reconnaissance, and unmanned inspection [17–19]. One should note that UAVs can operate in inaccessible areas which are affected by natural disasters, such as volcanic eruptions and earthquakes in addition to dangerous areas like mountainous locations, floodplains, and desert areas. UAVs classification depends on their range, endurance, and payload capacities. Based on such critical factors, one can generally classify UAVs into two main types, fixed-wing aircraft and rotorcraft as illustrated in Fig. 2.1(a) and Fig. 2.1(b) [20, 21]. Each type is adequate for particular missions and applications [22, 23]. Fixed-wing systems allow for more speed, which gives it a significant advantage to be the first choice for large area or long-distance missions as well as providing the user with a longer operation time due to the better battery economy. Also worth considering is the fact that such systems are able to carry considerable payloads for longer distances. On the other hand, fixed-wing systems cannot effectively work in remote areas where little or no infrastructure is present. For example, such platforms need more space for take-off and landing. In the same manner, fixed-wing systems need more space for turns, thereby increasing the path length and decreasing efficiency. Furthermore, by using fixed-wing platforms, the data density from LiDAR

sensing perspective is reduced, and such platforms need high-performance sensors to reach a specific point density due to the higher speed of fixed-wing platforms. It is important to note that such platforms require air moving over their wings to generate lift, thereby it needs to stay in a constant forward motion. Therefore, fixed-wing systems are not the best choice for stationary applications such as inspection work since such systems cannot stay stationary.



(a)



(b)

Fig. 2.1.: Examples of different UAVs platforms: (a) eBee SenseFly SA (an example of fixed-wing aircraft), and (b) DJI Inspire 2 (an example of rotorcraft)

In response to such restrictions, rotorcraft platforms come into play and become more favorable for small-area surveys with complex terrain than fixed-wing platforms. Regarding rotorcraft platforms, there is no need for an airstrip for take-off and landing. Also, such platforms offer excellent maneuverability within the flight mission.

Compared to the fixed-wing platform, the rotorcraft platform delivers more stability in the flight mission (less affected by wind). In fact, rotorcraft platforms deliver several advantages which make the UAV-based LiDAR system accessible to a broader user community [24]. However, rotorcraft platforms need a sophisticated maintenance process since such platforms involve complex mechanical and electronic components. As a result, such a process might decrease the operational time of the platform.

### 2.2.2 Onboard Mapping Sensors (Laser Scanners)

Light Detection and Ranging (LiDAR) systems onboard static and mobile platforms have emerged as a prominent tool for the direct derivation of accurate point clouds along object surfaces with high density. The widespread adoption of LiDAR systems is motivated by recent advances in laser ranging and scanning technologies as well as direct geo-referencing systems which can provide accurate position and orientation of the platform at high frequency. The central concept of laser ranging is facilitating the estimation of the distance between the laser beam firing point and its footprint. Range measurements can be achieved by utilizing Continuous Wave (CW) laser systems or pulsed laser systems. The range ( $\rho$ ) is determined by measuring the phase difference between the transmitted and the received signal in CW laser systems, which are commonly applied in terrestrial LiDAR systems [25]. In pulsed laser systems, the range measurement is based on the Time of Flight (TOF) of the laser pulse (i.e., the time delay between the emitted and received laser pulses). One should note that the pulsed laser systems are usually used in most available airborne LiDAR systems [26, 27]. Equation 2.1 illustrates the range measurement which is based on the TOF of the laser pulse. In this equation,  $\rho$  refers to the range of laser scanner,  $C$  denotes the speed of light, and  $t$  represents the measured time interval.

$$\rho = \frac{Ct}{2} \quad (2.1)$$

Regarding scanning systems, the laser beam is steered by a mirror that either rotates in a single direction (i.e., linear laser scanners) or in two directions (e.g.,

elliptical laser scanners). The steering mirror when either coupled with an internal rotation of the scanning unit or motion of the carrying platform would allow for the generation of dense point cloud along surrounding objects. The former mechanism is used for static scanning while the latter is used for mobile systems. Multi-beam laser scanners such as Velodyne products, have several laser beams that are pointing at different directions. A rotational mechanism allows for  $360^\circ$  coverage across the axis of rotation. The field of view along the rotational axis is controlled by the set-up of the laser beams. Modern laser systems are capable of providing up to a million pulses per second. This capability allows for the derivation of highly dense point clouds.

For the derivation of the mathematical relationship between the sensor measurements and the object coordinates of the point cloud, one should start by establishing the different coordinate systems associated with a LiDAR unit. The vector and matrix notations used in this dissertation are as follows:

- $r_a^b$  denotes the coordinates of point ‘a’ relative to point ‘b’ in the coordinate system associated with point ‘b’.
- $R_a^b$  denotes the rotation matrix that transforms a vector defined relative to the coordinate system ‘a’ into a vector defined relative to the coordinate system ‘b’.

For coordinate systems associated with a single LiDAR unit, the rotation matrix relating the different components will be denoted as the boresight matrix while the spatial offset relating them will be denoted as the lever arm. One should note that for a GNSS/INS assisted mobile LiDAR unit, the coordinates of a given point  $I$  relative to the mapping reference frame can be derived through a vector summation process as can be seen in Equation 2.2 [6, 28–31]. In this equation,  $r_I^m$  is defined the ground coordinates of the laser beam footprint relative to the mapping frame,  $r_b^m(t) \& R_b^m(t)$  are defined the interpolated position and orientation of the Inertial Measurement Unit (IMU) body frame relative to the mapping frame. Also,  $r_{lu}^b \& R_{lu}^b$  are defined the laser unit location and boresight matrix relative to the body frame,  $R_{lb}^{lu}(t)$  is defined the

rotation matrix related the laser beam to the laser unit, and  $r_I^{lb}(t)$  is represented the position of a point  $I$  with respect to the laser beam.

$$r_I^m = r_b^m(t) + R_b^m(t)r_{lu}^b + R_b^m(t)R_{lu}^b R_{lb}^{lu}(t)r_I^{lb}(t) \quad (2.2)$$

One should note that scanning units can be classified according to the laser range into close range and long-range scanning systems based on their intended applications. For example, close-range laser scanning systems (i.e., ranges between 0 and 200 meters) are used in Terrestrial Laser Scanners (TLS), Terrestrial Mobile Laser Scanners (TMLS), and Airborne Laser Scanner (ALS). Otherwise, the long-range laser scanning systems have the range between several hundred meters up to several kilometers. Such systems are used in many mapping applications related to ALS. Most of the UAV-mapping applications depend on close-range scanning systems because of their light weight and low cost compared with long-range scanning units.

It is worth noting that, there are various industrial LiDAR units which have been developed with affordable sensors suitable for a LiDAR-based UAV mapping system. Most of such sensors are mainly designed for industrial applications and robotics such as Hokuyo UTM-30LX-EW and Sick LMS-291 LiDAR sensors. The main advantages of such industrial sensors are their compact size, durability, and low cost. However, the performance of these sensors might have limitations on the maximum range and ranging accuracy. Due to such limitations, these types of sensors are not always the suitable choice for some mapping applications which require a considerable range with significant accuracy. However, such sensors are sufficient for many other research applications. In recent years, with the rapid improvement of LiDAR sensors, the automotive industry has delivered multi-beam laser scanner sensors such as Velodyne VLP-16 Puck and Quanergy M8-1 as shown in Fig. 2.2. Such sensors have several laser beams that are pointing at different directions as mentioned before [24]. Multi-beam laser scanner sensors introduce high data rate scanning options at a reasonable price.





Fig. 2.2.: Examples of multi-beam laser scanner sensors, (a) Velodyne VLP-16 Puck, and (b) Quanergy M8-1 LiDAR

In addition to that, there are particular sensors which are mainly designed for airborne operations, and their specifications make them more suitable for advanced use. However, the development of such sensor designs has a significant effect on the cost of these sensors as well as a negative impact on exploiting their capabilities in the mapping community. Riegl VQ-480-U and VUX-1 variants are examples of such sensors, which allow for operations at high altitude, especially for a manned aircraft. On the other hand, Hokuyo UXM-30LXH-EWA and Velodyne series (e.g., HDL-32E, VLP-16, and VLP-16 High Resolution) are considered types of sensors which are suitable for low-altitude missions, especially for an unmanned aircraft. Such sensors can operate at flight altitudes of up to 100 meters, but in practice, typical operation altitudes are 40-70 meters [32].

It is important to note that the LiDAR-based mapping and remote sensing research area is exposed to continuous advancement on various fronts due to several research innovations and technological developments that have been exerted towards such area of research. As a result, VLP-32C is a long-range LiDAR sensor that is recently delivered by Velodyne [33]. Such a sensor has the advantage of using 32 laser beams which are aligned over the range of  $+15$  degrees to  $-25$  degrees using the same design of VLP-16. One should mention that VLP-32C will be a popular choice due to its compact size and light weight (0.925 Kg). Furthermore, it can scan

up to 1,200,000 points per second with a range of 200 meters and typical accuracy of  $\pm 3$  cm. In addition to that, flash LiDAR technology (e.g., ASC TigerCub) is considered to be a promising progression for the UAV-based LiDAR system since the three-dimensional flash LiDAR operates like a 2D camera [24]. The 3D focal plane array of such a camera has rows and columns, similar to 2D cameras, but with extra capability of having 3D "depth" and intensity as shown in Fig. 2.3.



Fig. 2.3.: ASC's Tiger-Eye 3D Flash LiDAR Camera [24]

More specifically, the time that the camera's laser flash pulse needs to reach the object and bounce back to the camera's focal plane is recorded by each pixel. One should note that each pixel has independent triggers and counters to record the time-of-flight of the laser light pulse to the object. The physical range of the objects in front of the camera is calculated, and a 3D point cloud frame is generated at video rates. It is important to note that 3D flash LiDAR technology has been adopted in a variety of applications such as collision avoidance, object identification, restricted area event alerts, terrain mapping, and hazardous material detection. Due to the military origin of such technology, only limited knowledge is available in the civil domain. Furthermore, flash LiDAR sensors are currently very expensive. It is expected that when fully developed and operational, these types of sensors will alter the paradigm of the UAV-based LiDAR research field and have an apparent significant improvement in the upcoming years.

As mentioned previously, LiDAR units use laser beams to measure ranges and generate precise 3D information about the scanned area. Such units can be set up

onboard airborne platforms (such as UAVs) which are known as ALS, or terrestrial platforms (such as cars or trucks) that are cognized as TMLS. Fig. 2.4(a) and Fig. 2.4(b) show some examples of the laser scanning systems that are commonly used by the mapping community [34,35]. These systems have constituted a prominent tool for the direct acquisition of accurate point clouds along object surfaces with high density. The ideal platform for conducting a survey depends on the desired application. For instance, an airborne LiDAR system is more suitable for precision agriculture, DBM generation, or accident-scene reconstruction. However, a mobile terrestrial LiDAR system would be convenient for other applications such as infrastructure monitoring, and geometric documentation of transportation corridor.



(a)



(b)

Fig. 2.4.: Examples of different laser scanning systems: (a) Psomas Mobile Mapper system (an example of terrestrial mobile laser system), and (b) RIEGL VQ-1560-DW (an example of airborne laser system) [34,35]

### 2.2.3 UAV-based LiDAR systems

The emergence of MMS has set a marked paradigm in the photogrammetric and mapping community, motivated by the technical advancements in consumer-grade navigation sensor (GNSS/INS) technology. Furthermore, the development of MMS is affected by the emergence of low-cost digital cameras, and widespread adoption of LiDAR sensors onboard airborne/terrestrial platforms for direct acquisition of surface information. The integration between UAVs and a LiDAR unit is a relatively new area of research and applications. Also, such integration is considered as one of the most impressive developments in the surveying market in the sense of providing a mapping platform that is equipped with LiDAR systems. Regarding this new area of research, a remotely controlled helicopter equipped with navigation sensors and a laser range finder (altimeter) was one of the first starting points in 2006. Such a platform could work for topographic surveys. The extensive use of UAVs-based LiDAR systems have been studied for almost a decade now. Therefore, there is an opportunity to have many discussions about the feasibility and possibilities of utilizing such systems in the mapping industry.

As mentioned before, UAVs, which are intended for mapping purposes, are equipped with passive and/or active sensors together with a direct geo-referencing unit (GNSS/INS) to determine the platform's position and orientation. It is important to note that the incorporation of GNSS and INS is considered a fundamental trend for positioning and navigation projects to overcome the shortages of the individual use of GNSS and INS. In other words, GNSS can provide only position information of the platform. However, INS, although it provides the position and orientation changes of the platform, suffers from drift errors over time of standalone usage. Such GNSS/INS integration can ensure precise position, velocity, and attitude estimation of the mapping platform. Furthermore, GNSS/INS integration can overcome some of the problems related to tricky areas such as forests and urban canyons, where GNSS cannot accurately perform the duty alone due to the absence of GNSS signal. Therefore, the GNSS/INS

module is considered to be a crucial component of MMS since the integration of such module with the mapping sensors can provide high-quality maps.

To date, motivated by the availability of consumer-grade navigation systems, several research efforts have been exerted towards UAV-based 3D reconstruction. Such research takes advantage of the integration between the direct geo-referencing unit and LiDAR unit(s) onboard UAV platforms [1, 36, 37]. Nagai et al. [38] integrated an inexpensive SICK LMS-291 LiDAR unit with direct geo-referencing module onboard a helicopter model RPH2. Also, Yi Lin et al. [39] developed a mini-UAV-based LiDAR system. Such system comprises a Ibeo Lux laser scanner, Sick LiDAR unit, and NovAtel GNSS/INS module (SPAN-CPT) onboard a helicopter model Align T-Rex 600E. Guo et al. [40] developed a UAV-based LiDAR system for ecosystem-wide biodiversity studies. Such a system consists of a Velodyne LiDAR unit (VLP-16 Puck Lite), NovAtel GNSS/INS module (SPAN-IGM-S1), and a micro-computer. It is important to note that utilizing lower-grade GNSS/INS cannot provide sufficient accurate geo-referencing data. For large UAV platforms, higher-grade GNSS/INS modules are used; however, such modules increase the system weight and cost [41]. The utilization of Micro-Electro-Mechanical Systems (MEMS) technology allows for significant reduction of weight, cost, and power requirements. Recent development of MEMS technology produces MEMS GNSS/INS modules. Such MEMS GNSS/INS units are suitable for small UAV platform [42, 43]. Therefore, a GNSS/INS module is considered a promising technology for LiDAR-based UAV MMS.

The implementation of LiDAR systems onboard UAVs, supported by GNSS/INS units, has prospered not only for military applications but also for commercial use. The development of some innovative techniques in UAVs-based LiDAR systems will be explained ahead. Scanfly is a new product which is set up by 3D Target in 2016 and is developed mainly for UAV platforms as shown in Fig. 2.5. The system consists of a Velodyne LiDAR unit (VLP-16 Puck Lite), accurate survey-grade GNSS/INS module with dual antenna GNSS receiver, and an integrated board which manages the data capture and synchronization process. The utilized navigation system provides an

accuracy of  $\pm 2$  cm in position and  $\pm 0.05$  degree in attitude. Furthermore, this system is flexible to install an additional image sensor, such as a panoramic camera to enable photographic documentation of the surroundings. It is worth noting that the Scanfly system is developed to use SLAM techniques which supplement the integrated GNSS/INS unit to achieve the best accuracy in case of GNSS satellite outage [44].

NEXUS 800 UAV is considered another example for a new airborne mapping system which is developed through the cooperation between four companies (Hypack, Infinite Jib, Velodyne, and SBG) in the UAV market as depicted in Fig. 2.6. Such system consists of a Velodyne LiDAR unit (VLP-16 Puck Lite) and an SBG GNSS/INS module which provides an accuracy of  $\pm 2$  cm in position and  $\pm 0.1$ - $0.2$  degrees in attitude. In addition to that, the platform has onboard windows Personal Computer (PC) that provides a real-time view of the quality of the acquired data [45].



Fig. 2.5.: Scanfly Airborne mapping system

In recent years, RIEGL has several research efforts which have been exerted towards mobile mapping community. As a result, RIEGL proposed its latest production called entry-level miniVUX-1 UAV LiDAR system as depicted in Fig. 2.7. The system comprises a RIEGL miniVUX-1 LiDAR sensor, an integrated GNSS/INS unit, and an optional RGB camera. The advantages of using such a system include its compact size, lightweight, and robustness. In addition, such sensors can be mounted on several categories of UAVs platforms such as a fixed wing, rotorcraft, and multi-rotor UAVs. Furthermore, online waveform processing for data acquisition is also provided [46].



Fig. 2.6.: NEXUS 800 UAV mapping system



Fig. 2.7.: MiniVUX-1UAV mapping system

The UAV LidarPod system is another integrated mobile mapping product which is developed by Routescene in collaboration with Hanseatic Aviation Solutions and Mapix Technologies [47]. The system provides a rotorcraft as a platform equipped with a Velodyne LiDAR unit (HDL-32E), and an integrated RTK GNSS/INS unit as illustrated below in Fig. 2.8. Furthermore, this system entails a ground station which ensures the transmission of RTK GNSS corrections to the Lidarpod system.

Moreover, Siteco company has launched its new mobile mapping product that is called Sky-Scanner system for LiDAR/image data collection onboard rotorcraft UAVs [48]. Sky-Scanner system is considered a high-performance airborne mapping system with low-cost sensors onboard both DJI-S1000+ and DJI Matrice-600 Pro

UAVs platforms as depicted in Fig. 2.9. This system is comprised of a Velodyne LiDAR unit (VLP-16 Puck Lite), a GNSS/INS module, and an RGB camera. The utilized navigation system is Trimble AP15 which provides an accuracy of  $\pm 2\text{-}5\text{ cm}$  in position and  $\pm 0.025\text{-}0.08$  degrees in attitude.



Fig. 2.8.: Rutescene airborne mapping System



Fig. 2.9.: Sky-Scanner LiDAR/image data collection system

Furthermore, YellowScan already established a new LiDAR onboard UAV surveying solution with the highest accuracy and dense georeferenced point cloud data in February 2016 called YellowScan Surveyor as depicted in Fig. 2.10. This system consists of a Velodyne LiDAR unit (VLP-16 Puck Lite) and a GNSS/INS module. The utilized navigation system is Trimble APX-15 UAV which provides an accuracy of  $\pm 2\text{-}5\text{ cm}$  in position and  $\pm 0.025\text{-}0.08$  degrees in attitude. Due to the robust design of



the system, YellowScan Surveyor can meet the needs of several mapping applications and such a system can be easily adapted to any UAV platform [49].

Although there have been many developments regarding UAV-based LiDAR systems, there is still a significant need to have a purpose-built system in contrast to using a commercially-available mapping system for two main reasons: system flexibility and interchangeability, and cost. One should note that most of the commercial-mapping systems provide platforms which are equipped with the required set of the mapping sensors for the intended application. More specifically, offering a fixed set of the mapping sensors which are difficult to manipulate, will lead to an obvious limitation in the applications of MMS. Moreover, it is important to note that establishing a purpose-built system is less expensive than purchasing a commercial one. In other words, the implementation of system integration of the individual mapping sensors is performed instead of using the commercial-mapping system which is more general and costly. Therefore, this dissertation aims to provide a turn-key solution for implementing a low-cost UAV-based LiDAR system which is flexible enough to deal with several mapping applications.



Fig. 2.10.: YellowScan mobile mapping System

### 2.3 LiDAR System Calibration

LiDAR system calibration is considered the foremost and critical task to ensure deriving accurate geospatial information. LiDAR system calibration entails the estimation of the intrinsic parameters (i.e., laser ranging and scanning unit) of the LiDAR unit as well as the mounting parameters (i.e., lever arm offset and boresight angles) that relate the LiDAR unit to the GNSS/INS module. The lever arm is often measured using classical surveying methods. However approximate values for the boresight angles are estimated from the sensor's alignment, and these initial mounting parameters are not accurate.

LiDAR system calibration process is usually carried through several steps: (i) laboratory calibration, (ii) platform calibration, and (iii) in-situ calibration [50]. Laboratory calibration is conducted by the system manufacturer during which all the individual system components are calibrated (e.g., range offset, mirror angle scale, or relative alignment of multi-beam laser system). For a GNSS/INS-assisted mobile systems, the lever arm offset and boresight angles between the laser unit and the IMU coordinate systems, as well as the lever arm offset between the IMU and the sensor reference point are determined. In the platform calibration, the lever arm offset between the sensor reference point and the GNSS antenna is determined. An in-situ calibration is often required to refine the parameters that are determined at both the laboratory and the platform calibrations since such parameters might be biased and/or not stable over time. Due to the non-transparent and sometimes empirical calibration procedures, many systematic discrepancies between conjugate surface elements in overlapping point clouds have been observed in the collected LiDAR data. In recent years, there is extensive research that has been exerted towards developing methods to eliminate or reduce the impact of systematic errors in system parameters on the derived point cloud.

### 2.3.1 Different Calibration Techniques for Eliminating the Impact of the Systematic Error in the LiDAR System Parameters

The existing approaches for mitigating the impact of system parameters on the derived point cloud can be classified into two main categories: system-driven (calibration) and data-driven (strip adjustment) procedures. This categorization is mainly based on the nature of the utilized data and mathematical model. System-driven calibration procedures are based on the physical sensor model relating system measurements/parameters to ground coordinates of LiDAR points. These procedures incorporate the system's raw measurements or at least the trajectory and time-tagged point cloud for estimating biases in the system parameters with the help of the LiDAR point positioning equation. However, data-driven methods only utilize the XYZ coordinates of the LiDAR point cloud. One should note that system-driven methods are considered the accurate way to mitigate the impact of systematic errors in the LiDAR system parameters on the derived point cloud. Since access to the system's raw measurements might be restricted to LiDAR system manufacturers, several research efforts are exerted towards the development of data-driven methods [6, 51–54]. Since data-driven approaches aim at improving the compatibility between overlapping strips by estimating local transformation parameters between the laser strips coordinate system and the reference one, such methods are also known as strip adjustment procedures. Kilian et al. [51] introduced an adjustment procedure like the photogrammetric strip adjustment for detecting discrepancies and refining the compatibility between overlapping strips. The dependence on distinct points to relate control surfaces and overlapping LiDAR strips is considered the main drawback of such an approach. The identification of distinct points is quite challenging because of the irregular nature of the LiDAR points. Kager [55] suggested a suitable primitives, where planar features are used in the strip adjustment procedure. Maas [53] proposed a least-square adjustment method to derive the correspondence between discrete points in one LiDAR strip and TIN patches in the other one. Such method

depends on detecting the discrepancies between conjugate surface elements instead of improving the compatibility between neighboring strips or analyzing the detected discrepancies. The major drawback of data-driven methods is the mathematical model employed to relate the LiDAR point cloud and the reference frame. The effects of systematic errors in the system parameters are usually modeled by an arbitrary transformation function between the LiDAR point cloud and reference frame coordinate systems. Depending on the nature of the inherent biases in the LiDAR system parameters, the utilized transformation function might not be appropriate. Habib et al. [56] developed a data-driven method by simplifying the LiDAR point positioning equation. Due to the absence of the system's raw measurements, an estimate of biases in the system parameters is derived by imposing constraints on the data acquisition configuration.

System-driven techniques can be classified to rigorous or quasi-rigorous approaches. The rigorous approach incorporates the system's measurements [57–60]. However, the quasi-rigorous approach incorporates at least the trajectory and time-tagged point cloud for estimating biases in the system parameters with the help of the LiDAR point positioning equation [6, 61]. Filin [58] proposed a method to determine the system calibration parameters by using control surfaces. To be more specific, such parameters are estimated by constraining the LiDAR points to the control surfaces they belong to. The main drawback of this method is that it can only work if the control surfaces are available in the environment. The control surfaces are considered one of the suitable primitives that can deal with the irregular nature of the LiDAR point cloud. In Skaloud and Lichti [57] the calibration parameters are estimated by enforcing a group of points to lie on a common plane. The utilized planes are selected manually and its parameters are determined along with the calibration parameters. However, such an approach depends on the availability of large planar patches with varying topography (i.e., surfaces with varying slope and aspect values) which can be available over urban areas. Furthermore, the number of unknowns changes with the number of planes used in the calibration procedure. Morin [62] proposed a method to

establish the correspondence between overlapping strips by utilizing point primitives. As mentioned before, the identification of distinct points is quite challenging due to the non-selective nature of the LiDAR points. Also, this method depends on the assumption that the average of the coordinates of tie points in overlapping strips corresponds approximately to the ground truth. Burman [61] and Toth [63] proposed a calibration procedure that considers only biases in the boresight angles. Moreover, in Burman [61], the amount of unknown parameters changes with the extent of the area or the number of primitives being used in the calibration method since the surface model is also considered as an unknown. Toth [63] estimated the boresight angles by utilizing identified discrepancies between conjugate surface elements in overlapping LiDAR strips. Such discrepancies are obtained via a matching procedure that works on interpolated regions. However, the determined planimetric offsets have low accuracy because of the weak reliability of the matching outcome. Habib et al. [6] and Bang [64] proposed the quasi-rigorous method to overcome such limitations. Such a method deals with a linear scanner with an assumption that the laser unit is nearly vertical (i.e., small pitch and roll angles). Such an assumption leads to a more relaxed data requirement. In other words, only the trajectory position and time-tagged point cloud coordinates are required. However, such an approach is not optimal for datasets captured by unsteady platforms (e.g., helicopters). More specifically, such datasets have significant pitch and roll angles that might negatively affect the quality of the estimated parameters. Kersting et al. [65] proposed a rigorous calibration method which is considered a more flexible calibration procedure since such a method can be performed without strict requirements (e.g., flight, terrain coverage, control, and pre-processing requirements). The overlapping strip pairs and the regions between the overlapping strip pairs are manually selected in this method and then are used in a LiDAR system calibration procedure. However, some problems could arise when using these manually-selected pairs/regions such as non-uniform balance of the distribution of the slope and aspect values within the selected regions, redundant slope and aspect values, and dependence on the experience of the operator. Therefore, the

implementation of a LiDAR calibration procedure based on the rigorous approach to deal with multi-beam laser scanner sensors is one of the objectives of this research work.

Over the past few years, a great deal of research has been devoted to modeling the inherent systematic errors in Velodyne laser scanners as well as the calibration of LiDAR systems [66–68]. The cost-effective Velodyne laser scanner, which is a multi-beam laser scanner unit and can rapidly capture a high volume of data, has been used in many mobile mapping systems and robotics applications [69–71]. Underwood et al. [72] calibrated the extrinsic parameters relating a SICK LMS-291 to a Novatel Synchronized Position Attitude Navigation (SPAN) system by minimizing the discrepancy between sensed data and a known structure (i.e., a vertical pole and relatively flat ground). Muhammad and Lacroix [73] performed calibration of a rotating multi-beam LiDAR with the objective to align the scan data as close as possible to a ground truth environment. He et al. [74] used pairwise multi-type 3D geometric features (i.e., point, line, plane) to derive the mounting parameters between 2D LiDAR and GPS/IMU. First, the points are segmented into different features, and their quality is evaluated to compute weights to be used in the minimization of normal distance between conjugate features. However, when the initial parameters are considerably inaccurate, the segments and derived weights may not be reliable.

Chan and Lichti [75] introduced an intrinsic parameters calibration for Velodyne HDL-32E based on static stations and also analyzed the temporal stability of range measurements which indicated an approximate warm-up time of 2000 sec for most laser beams. Glennie et al. [68] performed a geometric calibration with stationary VLP-16 to marginally improve the accuracy of the point clouds by approximately 20%. Moreover, they also investigated the range accuracy of VLP-16, which is quoted to have an RMSE value between 22 to 27 mm in the factory supplied calibration certificate. However, it was observed that some of the laser beams have worse range accuracy than the others. Although many LiDAR system calibration procedures have been developed in the past, outdoor calibration of integrated GNSS/INS and multi-

unit 3D laser scanners is still an active area of research. Due to the irregular nature of the LiDAR point cloud which needs suitable primitives and mechanism for using them, ALS, and TMLS calibration is considered a more challenging task compared to the photogrammetric system calibration.

It is important to mention that a reliable LiDAR system calibration procedure needs to settle the primitives that will be used for figuring the discrepancies between overlapping LiDAR strips and control surface. Distinct points have been used as a primitive in photogrammetric data for a long time. However, due to the non-selective nature of the LiDAR scanning process (i.e., inability to force the laser beam to scan a specific point), points cannot be directly used as the calibration primitives. In other words, it is not possible to reliably identify common points in overlapping point clouds and control surfaces [7]. Higher level features (e.g., linear and planar features) can be used as calibration primitives since such features can be reliably derived from overlapping point clouds and established as tie and control features. However, those features are not explicitly available in a LiDAR point cloud. Thus, extraction of such features has to be preceded by a data processing stage (e.g., segmentation of planar and pole-like features) [57, 76]. Furthermore, there is no guarantee that linear and planar features will always be available within the LiDAR point cloud (i.e., such features will be mainly available in urban and semi-urban environments) which is another real challenge that needs to be addressed.

Due to the extensive adoption of LiDAR systems and several efforts in evolving standards for the delivery of the LiDAR data, the access to the system raw measurements is expected to be not the only requirement to have a rigorous calibration procedure. Furthermore, a precise analysis to design an optimal flight and target configuration for calibration is the first and foremost step in order to ensure the most accurate estimates of mounting parameters. One should note that few research efforts have investigated the necessary flight and control configuration for LiDAR system calibration. Therefore, existing pertinent literature for such configurations are discussed in the upcoming section.

### 2.3.2 The Necessary Flight and Target Configuration Requirements for LiDAR System Calibration

Few research efforts have explored the required flight and control configuration to perform a rigorous LiDAR system calibration procedure as mentioned before. An analytical analysis of the recoverability of the elevation and intensity values at the interpolated grid cells, datum shifts, and the boresight angles using different configurations was introduced by Burman [61]. The following configurations were investigated: one LiDAR strip, two LiDAR strips flown in opposite directions, and three LiDAR strips (i.e., two strips in opposite directions and one strip perpendicular to them). Such analysis is executed with and without elevation and intensity gradients and with control information. For each configuration, the possibility of the estimation of each of the investigated parameters is analyzed. Burman [61] recommended a flight configuration which comprises four strips flown in opposite and in cross direction along with control information as depicted in Fig. 2.11. It is important to note that the recommended flight configuration can provide sufficient redundancy.

Burman [61] suggested to utilize a calibration site with sloped terrain as well as to select the regions close to the edges of the strips for the calibration procedure. Due to the nature of the suggested calibration procedure (the primitives are considered unknowns), a considerable correlation between the parameters was still observed (e.g., the vertical datum shift and the elevation values at the grid cells). The recoverability of the lever arm offset and systematic errors in the measured range and scan angle is not investigated.

Morin [62] recommended a flight configuration which was designed to recover the boresight angles and the scale factor of mirror scanning angle while considering a flat calibration site. Such configuration assumes that control points can be identified in overlapping strips. Fig. 2.12 illustrates the recommended flight configuration by Morin [62]. In order to recover the boresight pitch angle, different flying heights were recommended. At different flying heights, a bias in the boresight pitch angle



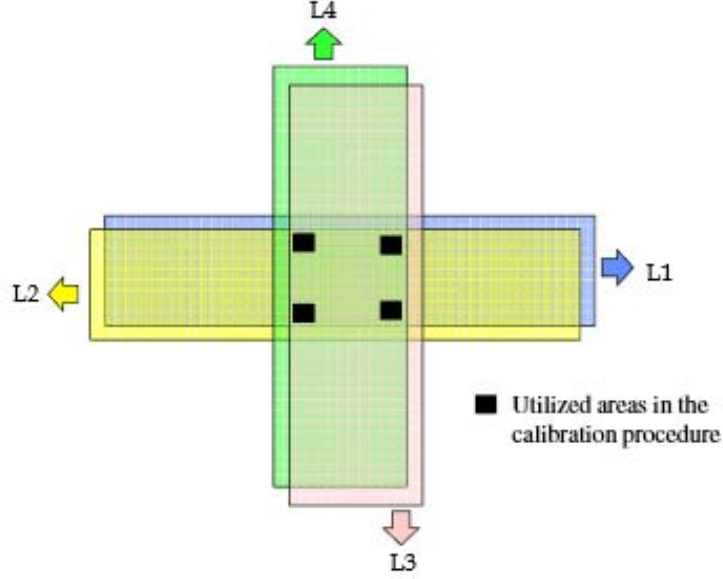


Fig. 2.11.: The recommended flight configuration for LiDAR system calibration by Burman [61]

( $\Delta\delta\omega$ ) causes a vertical discrepancy ( $\Delta h$ ) between overlapping strips as illustrated in Equation 2.3. It is worth noting that for small values of boresight pitch angle, the vertical discrepancy will be very small, which might influence the accuracy of the estimated parameter.

$$\Delta h = (H_1 - H_2) \left( \frac{1}{\cos \Delta\delta\omega} - 1 \right) \quad (2.3)$$

The optimal flight and target configuration for accurate estimation of the system parameters (i.e., the planimetric lever arm offset components, the boresight angles, the range bias, and the mirror angle scale) that was proposed by Kersting et al. [65], comprises three side lap cases and one vertical control point. The optimal flight configuration entails four LiDAR strips which are captured from two different flying heights in opposite directions with 100% side lap, and two LiDAR strips, which are flown in the same direction with the least side lap possible (while having sufficient conjugate surface elements between the strips) as depicted in Fig. 2.13. However, utilizing manually-selected pairs/regions in such a method will cause some problems

as mentioned before. Therefore, such problems might influence the accuracy of the estimated parameters during the calibration procedure.

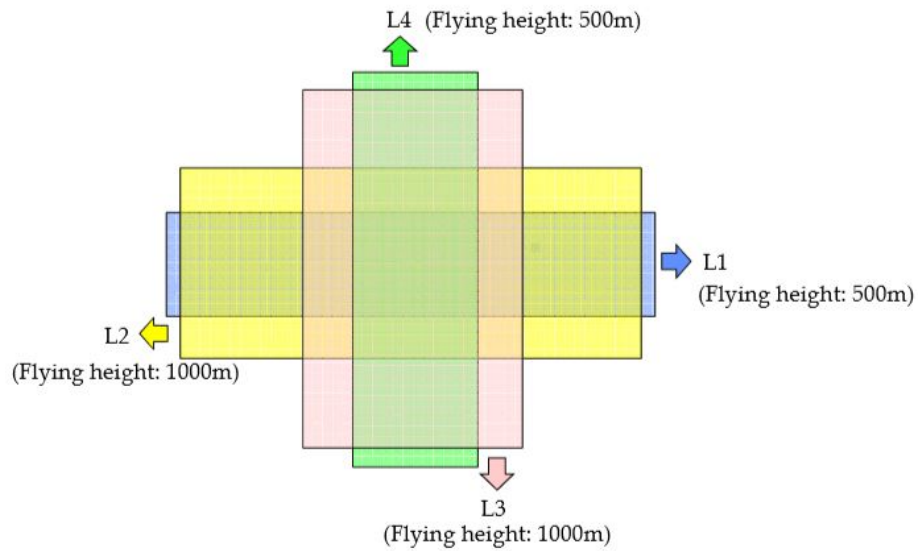


Fig. 2.12.: The recommended flight configuration for LiDAR system calibration by Morin [62]

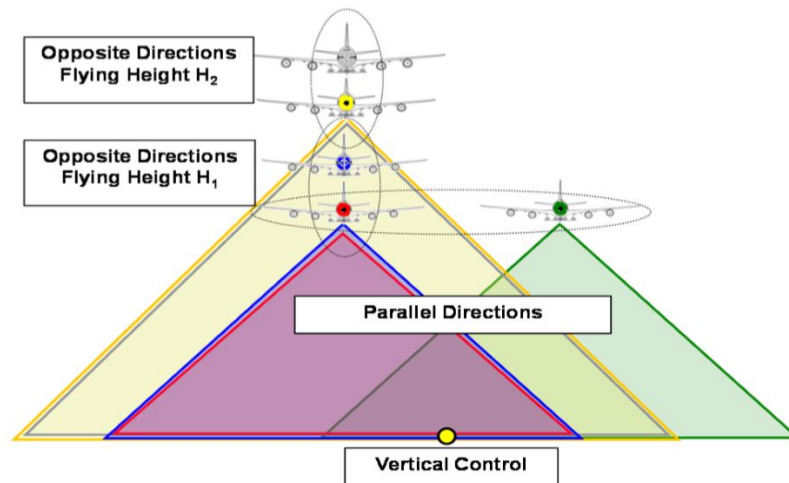


Fig. 2.13.: The optimal flight and target configuration for airborne LiDAR system calibration by Kersting et al. [65]

## 2.4 SLAM-Assisted Coverage Path Planning and Implementation of Pseudo GNSS/INS Localization System for GNSS-denied Environment

Over the past decades, there is an extensive necessity for autonomous unmanned vehicles to perform special missions such as rescue and routine operation, exploration in hazardous environments, and to reach inaccessible areas such as nuclear power plants. To reduce human casualties, various efforts have been exerted towards upgrading the unmanned vehicles to be more intelligent and compact. Coverage Path Planning (CPP) problem is considered one of the main challenges for intelligent unmanned systems. This problem plays a pivotal role to improve the viability and mission ability of unmanned systems. One should note that CPP problems have been investigated in the robotics community. To be more specific, CPP is the problem of performing a path that can go through all the points of the environment of interest while avoiding either static or dynamic obstacles [77].

Choset [78] discussed that the CPP algorithms are classified as offline and online algorithms. For offline CPP algorithms, the environment is known in advance. One should note that there are different techniques to address the CPP problem such as cellular decomposition, genetic algorithms, neural networks, spiral filling paths, spanning trees, and ant colony method [79]. However, it might be unrealistic to have a prior knowledge of the area of interest for some cases. Therefore, the online algorithms use real-time sensor observations to sweep the entire area of interest. Such algorithms are also called sensor-based coverage algorithms. For a CPP problem, the coverage of area of interest is performed through two standard basic motions: (i) the boustrophedon (back-and-forth) motion, and (ii) the square spiral motion as shown in Fig. 2.14 and Fig. 2.15, respectively. The idea behind using such motions is the ability to cover the region of any shape as well as the possibility of movements especially in an environment full of obstacles. One should note that a CPP is considered a robust algorithm if the unmanned platform completely executes the coverage task with non-overlapping areas in finite time [80].

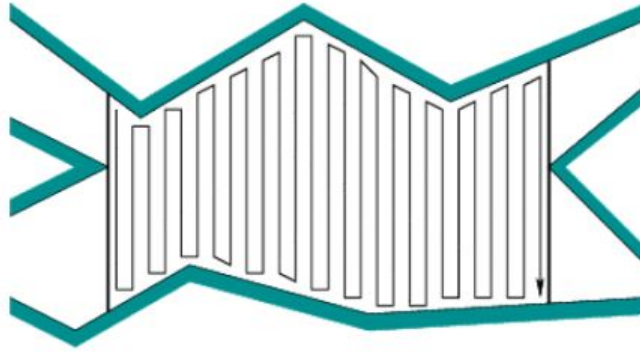


Fig. 2.14.: The boustrophedon motion

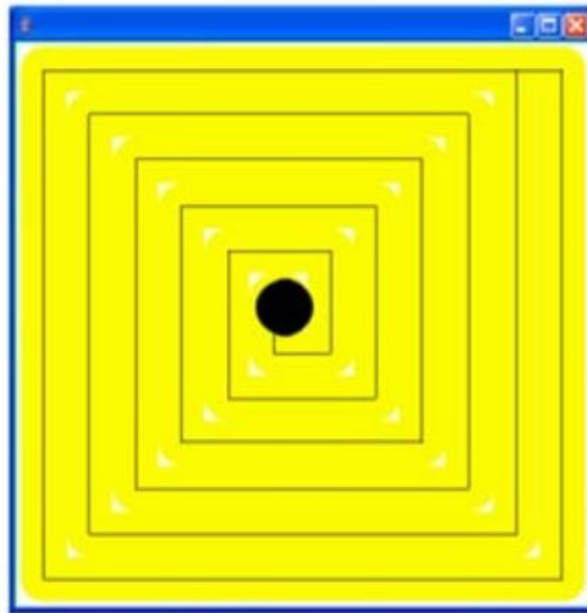


Fig. 2.15.: The square spiral motion [80]

For area coverage applications, CPP strategies form a class of motion planning methods that is disparate from generic motion planning techniques. This is mainly because the use of CPP strategies addresses area complete coverage problems. Such problems encompass applications which are ranging from aerial surveying and robotic demining [81, 82] to autonomous lawn mowing and vacuum cleaning [83, 84]. This distinction of CPP can be highlighted by definition provided by Galceran and Carreras

[10]. Therefore, while a general motion planning problem is concerned with generating an optimal trajectory between a source and destination, a CPP problem focuses on generating a path that maximizes area coverage. Since a CPP technique forms a basic framework for many path planning problems, this technique is addressed in literature for many robotic applications such as vacuum cleaning robots, autonomous harvesting, and underwater/aerial surveying [85–87].

Among the many approaches to solving the CPP problem, this dissertation focuses on exact cellular decomposition techniques which ensure complete coverage of the target area with energy/time-optimality. The trapezoidal decomposition and the boustrophedon decomposition are considered two of the most popular offline (environment assumed to be known a priori) cellular decomposition approaches that were discussed by Latombe [77]. The trapezoidal decomposition is considered a simple offline technique that can only handle planar and polygon areas [88]. Such technique divides the free space of the target area into trapezoidal cells as depicted in Fig. 2.16. Then, each cell that has two parallel sides, can be covered by simple back and forth motions parallel to either side. Therefore, complete coverage is ensured by visiting each cell in the adjacency graph. The major shortcoming of such a method is that it generates numerous trapezoidal cells that require too many redundant back and forth motions to guarantee complete coverage. To overcome such limitation, the boustrophedon decomposition technique was proposed by Choset and Pignon [89]. Such method compensates for the redundant movements by merging the cells that do not contribute to change in connectivity of the nodes in the adjacency graph as shown in Fig. 2.17. This merging technique reduces the number of cells in the decomposition. Thereby, the overall number of back and forth motions is reduced. However, while boustrophedon patterns used in this technique allow for complete area coverage, such a method does not guarantee distance/energy optimality for the entire path traced by the robot. One should mention that all such cellular decomposition methods rely on maximizing the area coverage without considering the time and energy spent. The Minimum-Sum-of-Altitudes (MSA) method described by Huang [90] provided a cost



Recent advances in SLAM techniques have greatly improved robot pose (position and orientation) estimation. The advent of SLAM techniques necessitated further modifications to the CPP problem. The accurate pose and map estimation provided by SLAM can be utilized conveniently for a navigation problem. However, for a CPP technique, the map/position update from SLAM needs to influence the unmanned platform controls to maximize area coverage [92, 93]. It is worth noting that the problem of improving the pose of the platform while performing area coverage has been addressed in only few research efforts. SLAM implementations for coverage applications have addressed such issues as dynamic coverage planning and optimal path tracing [94, 95]. In this dissertation, a new hybrid strategy is developed to use an area coverage function allowing dynamic changes to the coverage path from any map/pose updates provided by real-time SLAM method. It is important to note that the inception of the concept of SLAM techniques is considered the primary motivation behind the development of MMS frameworks and their operation in GNSS-denied areas [96].

Just as for the case of GNSS/INS-based MMS for GNSS-affluent areas, there has been comprehensive research and development for MMS frameworks in GNSS-denied areas. The concept of SLAM primarily establishes a probabilistic Bayesian framework that simultaneously estimates the pose of an MMS as well as landmarks/key points that form the environment map. Both feature-based, as well as volumetric, SLAM techniques have been implemented for terrestrial/airborne MMS frameworks [97, 98] and even underwater MMS [99]. Offline reconstruction techniques have been developed for MMS-based LiDAR systems such as LiDAR Odometry and Mapping (LOAM) [100]. Such techniques can provide high-quality maps from 3D-LiDAR sensors (i.e., Hokuyo UTM-30LX laser scanner). Considerable progress has also been made in Visual SLAM methods using monocular [101, 102] and RGB-D cameras [103] for MMS frameworks.

Implementation of such SLAM-based MMS frameworks has exhibited positive results for many applications. Corso and Zakhor [104] developed an indoor human

operated MMS framework using SLAM techniques for localization. This framework is equipped with five Laser Range Finders (LRF), two fish-eye cameras, and an IMU. Similarly, a feature-based SLAM system has been implemented in [105] for an MMS-based LiDAR. Zhao et al. [106] implemented SLAM frameworks for outdoor environments using incorporation between two laser scanner sensors and an IMU unit. More comprehensive integration has been carried out through incorporation of multi-sensor onboard platforms by [107, 108] allowing the use of both cameras and LiDAR sensors onboard the MMS. The indoor MMS implementation described in [109–111] demonstrates how high-quality indoor maps using multiple sensors can be built successfully on a SLAM-based framework in GNSS-denied environments. Nuchter et al. [112] illustrates an inexpensive mobile mapping solution that can be mounted on a backpack equipped with 2D & 3D laser scanner since such a system is implemented for IMU-free MMS by performing SLAM.

One should mention that most of these frameworks propose the implementation of an MMS based on SLAM for GNSS-denied environments without considering its flexibility to be converted into a GNSS/INS framework for operation in a GNSS-affluent area. The SLAM-based MMS framework has a different structure compared to a GNSS/INS-based MMS for two main reasons: (i) the position data generated by SLAM is dependent on the mapping sensor measurements, and (ii) the sensor data logged onboard the MMS is not tagged with any geo-referencing signals. Due to such distinctions, it is difficult to incorporate a SLAM-based MMS framework into a GNSS/INS-based MMS framework directly, without extensive modifications to the system. In this dissertation, a new framework is proposed to mitigate this problem by implementing the SLAM-based pseudo-GNSS/INS to operate like a GNSS/INS module. This pseudo-GNSS/INS framework introduces a novel approach for implementing the MMS framework for flexible operation in GNSS-denied/GNSS-affluent areas. Furthermore, such a framework has the advantages of implementing the entire operation, which incorporates the positioning module and the mapping sensor, utilizing single 3D-mapping sensor.



### 3. DEVELOPMENT OF SYSTEM ARCHITECTURE FOR A LIDAR-BASED AIRBORNE MOBILE MAPPING SYSTEM

#### 3.1 Introduction

MMS have emerged as a viable tool in many mapping applications and have been implemented on a range of airborne/terrestrial platforms. However, the general skeleton that describes the operating framework of the MMS remains somewhat common across all platforms and applications. This structure is shown in Fig. 3.1 for the case of a GNSS/INS-based MMS framework and is also explained ahead for each of its operational blocks.

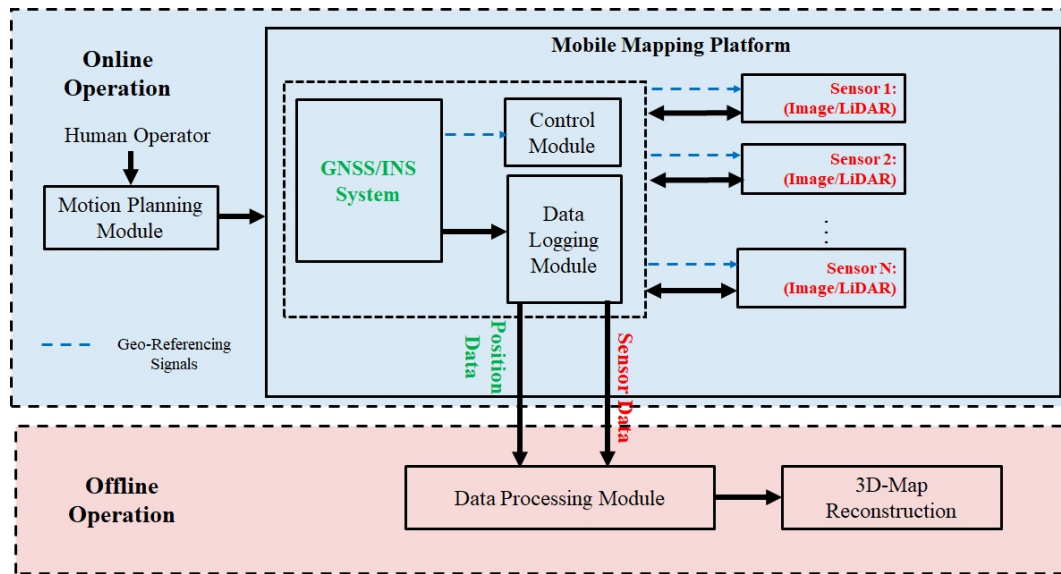


Fig. 3.1.: A functional block diagram of a GNSS/INS-based MMS framework with N sensors

While this block diagram describes the MMS, which use a GNSS/INS module for positioning, the block structure still highlights the essential features of a generic MMS. Fig. 3.1 allows for a visualization of the definition of the MMS in several blocks, for the mobile platform which describes a kinematic platform that can be a backpack carried by a human operator, airborne/terrestrial vehicle, either manned or unmanned, on top of which the mapping sensors are mounted and houses sufficient power supply for the mission operation. The MMS are mostly human-operated although several attempts at autonomous navigation of such MMS have also been successful. For the positioning and orientation (GNSS/INS) module, it is the most expensive component for the MMS and the most crucial for the appropriate determination of the geographic location of the map objects. Regarding the sensors module, it includes the network of mapping sensors that can range from monocular/stereo cameras, hyperspectral cameras, and/or 2D/3D LiDAR units. These sensors provide the data stream for map construction and need to be calibrated thoroughly for proper reconstruction. The data logging module is responsible for storing the sensor data in an appropriate format for precise offline reconstruction; however, the control module is responsible for managing the data acquisition based on time or a distance interval as a system block. Finally, the data processing module is responsible for gathering data to reconstruct the map of the surrounding environment. Owing to the high volume of data, this is mainly done off-line. Operation of the MMS yields two outputs for map reconstruction - the position data, which contains the trajectory traced by MMS during navigation and the raw sensor data which entails points used for reconstructing the maps.

This phase of research aims at developing a system architecture for a low-cost UAV mapping system using directly geo-referenced active optical ranging systems, while considering the challenges posed by using consumer-grade sensors, platform payload restrictions, endurance capabilities, and the diversity of potential users/applications. In this regard, a comprehensive investigation for ensuring the selection of the suitable platforms as well as the sensors for the mapping purpose is proposed. Also, a system integration is developed by performing the synchronization process between a direct

geo-referencing unit and an active optical ranging sensor onboard the proposed UAV platform. The appropriate data storage unit is investigated to meet the requirements of UAV-payload demands. Furthermore, the impact of system interference on the performance of a UAV platform as well as the GNSS/INS trajectory is investigated. In addition to that, the development of a system architecture for a LiDAR-based wheel mapping system is considered as another objective of this portion of research. One should note that the system integration of a LiDAR-based wheel mapping system is performed in a way that is similar to a LiDAR-based UAV system integration.

### **3.2 UAV Mobile Mapping System Framework**

In spite of the proven potential of UAV-based mapping using passive and active optical imaging systems, more in-depth analysis is still needed for these mapping systems. For example, a well-designed system architecture that considers payload restrictions and the specifications of the utilized direct geo-referencing component as well as the imaging and ranging systems in light of the required mapping accuracy and intended application is still lacking. In response to that, considerable research efforts are exerted to move from a mapping system that exhibits potential to a mapping platform that is recognized by the mapping industry, regulatory organizations, and end-user community. In order to address the above limitations, this dissertation develops a system architecture as well as a framework for the effective integration of a low-cost direct geo-referencing unit together with active optical ranging systems (e.g., laser scanners) for accurate 3D mapping environments.

To ensure that the UAV platform will be capable of satisfying the needs of the desired mapping and monitoring activity, this phase of the dissertation will investigate the specifications of the necessary components that are commensurate with payload restrictions, the extent of the area to be mapped, and required accuracy. This system which is mainly for outdoor applications includes a direct geo-referencing unit based on an integrated GNSS and INS receiver board, and a minimum of one active ranging

unit. All these components are rigidly fixed within the UAV platform and will be introduced in the next sections.

### 3.2.1 Mapping System Platform

Considering the implementation at hand, the DJI S1000+ and DJI Matrice M600 Pro can meet the mapping applications needs because of payload capabilities, endurance, and robustness. The DJI S1000+ platform is designed for professional aerial mapping applications, and provides some advantages such as safety, stability, and ease of use as shown in Fig. 3.2. This platform is a vertical take-off and landing UAV that weighs approximately 4.4 kg without the batteries. With a maximum takeoff weight of about 11 Kg, such a platform can easily fly with the installed sensors and batteries for up to 15 minutes, which means that time increases as the weight decreases. The DJI S1000+ platform allows around 4 to 5 Kg payload of sensors/equipment (including batteries) to be efficiently installed onboard. The specifications of the DJI S1000+ are presented in Table 3.1 [113].



Fig. 3.2.: The DJI S1000+ platform for mapping applications

Regarding the DJI Matrice 600 Pro (M600 Pro), it is a six-rotor flying platform that is designed for mapping and industrial applications as shown in Fig. 3.3. The DJI M600 Pro uses two sets of six intelligent flight batteries (TB47S and TB48S

batteries) with different capacity depending on the planned flight mission time. A new battery management system is utilized to provide safety, more endurance, and power system reliability. With a maximum takeoff weight of 15.5 kg, such a drone can be able to meet the particular needs of several applications such as precision agriculture and crash scene reconstruction. The detailed specifications of the DJI M600 Pro are introduced in Table 3.2 [114].

Table 3.1.: The DJI S1000+ UAV Specifications

Feature	DJI S1000+
Total Weight	4.4 kg without batteries
Takeoff Weight	13.2 lb /6 kg to 24.2 lb /11 kg
Max Flight Speed	13-20 m/s
Motor Stator Size	41x14mm
Motor Max Power	500W
Motor Weight (with Cooling Fan)	158g
Foldable Propeller Material	High strength plastics
Foldable Propeller Size	15x5.2inch
Foldable Propeller Weight	13g
Power Battery	LiPo (6S,10000mAh 20000mAh)
Max Power Consumption	1500W (@9.5Kg Takeoff Weight)
Hover Time	15min (9.5Kg Takeoff Weight)
Working Environment Temperature	-10 ° C to +40 ° C



Fig. 3.3.: The DJI Matrice 600 Pro (M600) platform for mapping applications

Table 3.2.: The DJI Matrice Pro (M600) UAV Specifications

Feature	DJI M600 Pro
Total Weight (with six TB47S batteries)	9.5 kg
Total Weight (with six TB48S batteries)	10 kg
Max Takeoff Weight Recommended	15.5 kg
Standard Battery (Model: TB47S)	Capacity: 4500 mAh
Optional Battery (Model: TB48S)	Capacity: 5700 mAh
Hovering Time (with six TB47S batteries)	6 kg payload: 16 min
Hovering Time (with six TB48S batteries)	5.5 kg payload: 18 min
Working Environment Temperature	-10 ° C to +40 ° C

### 3.2.2 Direct Geo- referencing Module

The derivation of the ranging data requires the geo-referencing of the mapping platform, which entails the determination of the position and the orientation of the individual sensors relative to a user-defined coordinate system. To be more specific, the platform's position and orientation can be directly established using an integrated GNSS/INS module, which is known as a direct geo-referencing unit as shown in Fig. 3.4. The main advantage of such a unit is the reduction or even elimination of the ground control requirement, which is quite useful in mapping remote and inaccessible areas, as well as reducing the cost of the overall mapping process. One should note that establishing ground control is the most expensive mapping task next to the deployment of the data acquisition platform. Therefore, the incorporation of a GNSS/INS position and orientation module has become the default for the majority of UAV-based mapping activities. In this dissertation, comparative performance of the Novatel SPAN-IGM-S1 and the Trimble APX-15 UAV (V2) GNSS/INS units will be investigated ahead.

For this investigation, the Trimble APX-15 UAV (V2) is considered due to its low weight, compact size, and robust positioning and orientation information as shown in

Fig. 3.5. The POSPac MMS 8.1 Differential GNSS Inertial post-processing software from Applanix is used for post-processing of the raw GNSS/INS data. The accuracy attained after post-processing is  $\pm 0.025^\circ$  for pitch/roll and  $\pm 0.08^\circ$  for heading (yaw), and the position accuracy is  $\pm 0.02\text{-}0.05\text{ m}$ . The specifications of the Trimble APX-15 UAV (V2) are illustrated in Table 3.3 [115].

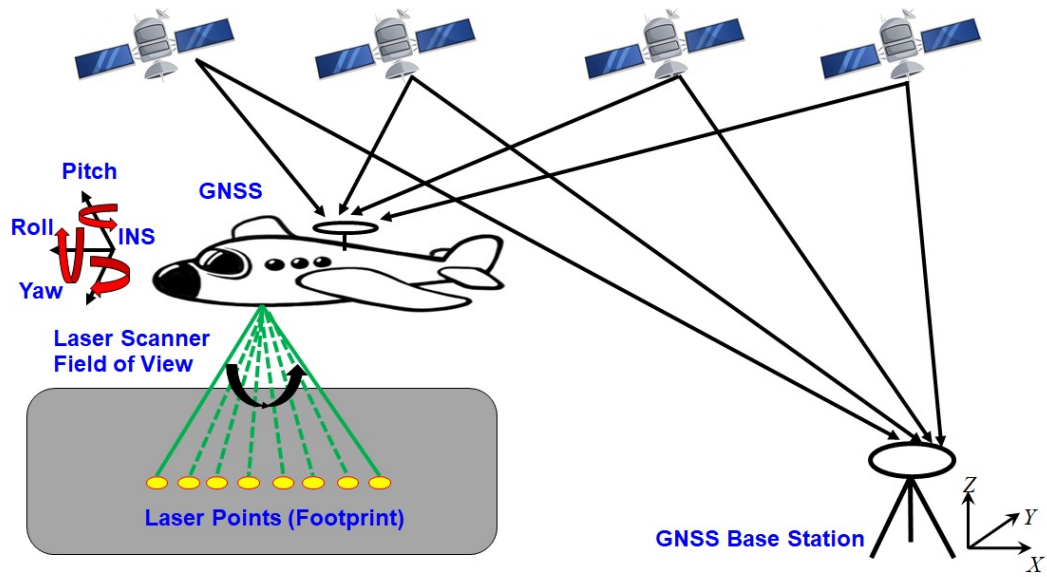


Fig. 3.4.: Direct Geo-referencing of LiDAR data using a GNSS/INS module



Fig. 3.5.: Trimble APX-15 UAV (V2) GNSS/INS unit

Table 3.3.: Trimble APX-15 UAV (V2) Specifications

Feature	Trimble APX-15 UAV (V2)
Weight (without cables)	60 grams
Size	67 L x 60 W x 15 H mm (nominal)
Voltage Input	9-30 V DC
Power Consumption	3.5 watt at room temperature
Position Accuracy (post processing)	$\pm 0.02 - 0.05$ (m)
Velocity Accuracy (post processing)	0.015 (m/sec)
Roll & Pitch Accuracy (post processing)	$\pm 0.025$ (deg)
True Heading Accuracy (post processing)	$\pm 0.080$ (deg)
IMU data rate	200 Hz
GNSS data rate	5 Hz
Working Environment Temperature	-40 ° C to +75 ° C

Also, the Novatel SPAN-IGM-S1 navigation system is considered in this phase of research for UAV-based mapping activities and is shown in Fig. 3.6. For such a system, the GNSS position collection rate is 20 Hz, and IMU measurement rate is 125 Hz. The Inertial Explorer Differential GNSS Inertial post-processing software from Novatel is used for post-processing of the raw GNSS/INS data. The accuracy attained after post-processing is  $\pm 0.015^\circ$  for pitch/roll and  $\pm 0.08^\circ$  for heading (yaw), and the position accuracy is  $\pm 0.02$  m. The detailed specifications of the Novatel SPAN-IGM-S1 are illustrated in Table 3.4 [116].

According to the previous investigations, the Trimble APX-15 UAV (V2) module will be used in the system architecture instead of the Novatel SPAN-IGM-S1 unit due to its low weight, compact size, power consumption, and cost. These specifications are commensurate with the requirements of the UAV-payload restrictions and the endurance. One should note that although the position accuracy of the Novatel



SPAN-IGM-S1 module is a bit better than the Trimble APX-15 UAV (V2) unit, the position accuracy of this unit can still satisfy the needs of the intended mapping applications.



Fig. 3.6.: Novatel SPAN-IGM-S1 GNSS/INS unit

Table 3.4.: Novatel SPAN-IGM-S1 Specifications

Feature	SPAN-IGM-S1
Position Accuracy (post processing)	$\pm 0.02$
Roll & Pitch Accuracy (post processing)	$\pm 0.015^\circ$
True Heading Accuracy (post processing)	$\pm 0.08^\circ$
Voltage Input	+10 to +30 VDC
Power Consumption	6 watt
Weight	0.54 Kg
Dimensions	152 L x 142 W x 51 H mm (nominal)
Data Logging media	USB or Serial connections
Number of GNSS Receivers	1
Distance Measurement Indicator (DMI)	No Support for optional DMI input
Working Environment Temperature	-40 ° C to +65 ° C

### 3.2.3 Active Sensor (Laser Scanner)

In the past few years, different types of LiDAR sensors are introduced with different scanning mechanisms, number of laser beams, and geometric configurations. The Velodyne LiDAR sensors are gaining attention in UAV-based mapping applications, specifically the Velodyne HDL-32E as shown in Fig. 3.7 and Velodyne VLP-16 Puck & Puck Hi-Res. Regarding Velodyne HDL-32E, its compact size and light weight (1.3 Kg including cables) make it the suitable choice for its specifications. Such a sensor has the advantage of using 32 laser beams which are aligned over the range of  $+10.67$  degrees to  $-30.67$  degrees that provide the vertical Field of View (FOV), and its patent-pending rotating head design delivers a 360-degree horizontal FOV ( $\alpha$ ). Furthermore, it can scan up to 700,000 points per second with a range of 100 meters and typical accuracy of  $\pm 2$  cm. The output from HDL-32E is a high definition 3D-point cloud that provides mobile mapping applications with more useful environmental data than conventional LiDAR sensors. The detailed specifications of the Velodyne HDL-32E are introduced in Table 3.5 [117].



Fig. 3.7.: The Velodyne HDL-32E sensor

The Velodyne VLP-16 Puck is smaller than the Velodyne HDL-32E since it only has 16 channels which are aligned over the range of +15.00 degrees to -15.00 degrees. Such channels deliver a 360-degree horizontal FOV ( $\alpha$ ). The VLP-16 Puck does not have visible rotating parts when compared with Velodyne HDL-32E as represented in Fig. 3.8. Also, it can scan up to 300,000 points per second with a range of 100 meters and typical accuracy of  $\pm 3$  cm. The VLP-16 Puck Hi-Res is considered a higher resolution version of VLP-16 Puck and is mainly used for several activities which necessitate a high-resolution point cloud. It is worth mentioning that although both sensors have identical technical specifications, there is only one significant difference between such sensors. The vertical FOV of VLP-16 Puck Hi-Res is 20 degrees (+10.00 degrees to -10.00 degrees) instead of 30 degrees for the Puck unit as mentioned previously. The specifications of both sensors are illustrated in Table 3.6 [32].

Table 3.5.: The Specifications of the Velodyne HDL-32E

<b>Feature</b>	<b>Velodyne HDL-32E</b>
Number of channels	32 channels
Range	70-100 m
Scan rate	5-20 Hz
Number of pulses/second	Up to 700,000 points/second
Motor RPM	300-1200 rpm (user selectable)
Horizontal Field of View (FOV)	360°
Vertical Field of View (FOV)	40°
Angular Resolution (Vertical)	1.33°
Voltage Input	9-32 VDC (31.4 watt)
Dimensions	149.86mm height x 85.3mm diameter
Weight (including cables)	1.3 kg
Working Environment Temperature	-10 ° C to +60 ° C



Fig. 3.8.: The Velodyne VLP-16 Puck/ Puck Hi-Res sensor

Table 3.6.: The Specifications of the Velodyne VLP-16 Puck/ Puck Hi-Res

Feature	Velodyne VLP-16 Puck/ Puck Hi-Res
Number of channels	16 channels
Range	Up to 100 m
Scan rate	5-20 Hz
Number of pulses/second	Up to 300,000 points/second
Motor RPM	300-1200 rpm (user selectable)
Horizontal Field of View (FOV)	360°
Vertical Field of View (FOV)	30° for Puck & 20° for Puck Hi-Res
Angular Resolution (Vertical)	1.33°
Voltage Input	9-18 VDC (8 watt)
Dimensions	103 mm Diameter x 72 mm Height
Weight (without cables)	0.83 kg
Working Environment Temperature	-10 ° C to +60 ° C

According to the previous investigations, the Velodyne VLP-16 Puck Hi-Res unit will be used in the system architecture instead of the Velodyne HDL-32E sensor due to its low weight, compact size, power consumption, and cost. These specifications are commensurate with the requirements of the UAV-payload restrictions and endurance. Furthermore, the Velodyne VLP-16 Puck Hi-Res is utilized instead of the Velodyne VLP-16 Puck due to its vertical FOV which allows it to provide a highly-dense point cloud as mentioned previously.

### 3.3 System Integration of UAV-based LiDAR system

The demand for professional UAV-based mapping solutions is growing worldwide. In this regard, LiDAR-based system integration is considered one of the fundamental factors that have a significant effect on the ability to provide accurate geospatial information. As it has been mentioned before, the proposed UAV-based LiDAR system consists of an industrial-grade autonomously flown UAV, either the DJI S1000+ or DJI M600 Pro, equipped with a LiDAR sensor coupled with a GNSS/INS navigation unit as well as a management system. The management system is concerned with several tasks, including the power system distribution and has a processing unit which is responsible for setting the data-acquisition parameters, data storage for the generated real-time LiDAR data, data logging, and time synchronization. All these components are rigidly fixed within the DJI M600 Pro and DJI S1000+ as shown in Fig. 3.9, Fig. 3.10(a), and Fig. 3.10(b) respectively. In Fig. 3.10(b), one should mention that the DJI S1000+ based LiDAR mapping system using the Velodyne HDL-32E sensor is still not operational at this stage. However, it is straightforward to modify the system installation to receive this sensor.



Fig. 3.9.: The DJI M600 Pro based LiDAR mapping system (Velodyne VLP-16-Puck Hi-Res)



**LiDAR Unit  
(VLP-16 Puck)**



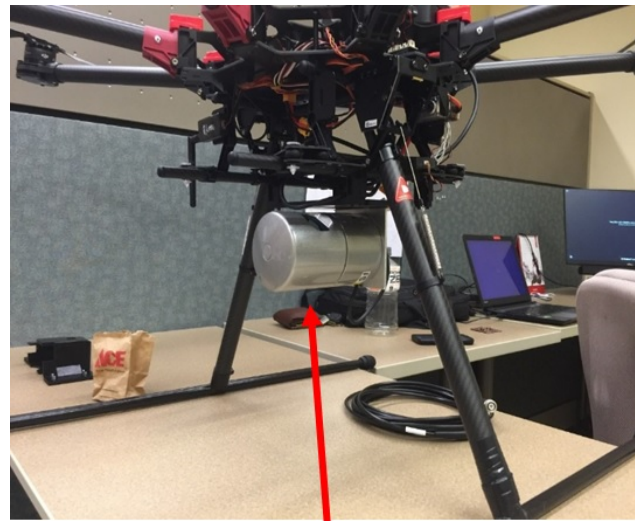
**GNSS/INS  
Module**



**Raspberry  
Pi**



(a)



**LiDAR Unit (HDL-32E)**



(b)

Fig. 3.10.: The DJI S1000+ based LiDAR mapping systems – (a) The Velodyne VLP-16 Puck -(b) The Velodyne HDL-32E

### 3.3.1 Synchronization Process

The time synchronization process is the most critical and key part of the whole system integration architecture for the LiDAR-based mapping system. Such a process has a considerable impact on the generated data in the sense of ensuring that the captured data will have the necessary characteristics (i.e., accurate geo-referencing information for the captured frames). In order to derive direct geo-referencing data, the integrated GNSS/INS module supplies sequentially precise time pulses, known as Pulse-Per-Second (PPS) signals. Such signals give the ability to generate a time-tagged point cloud. In this regard, synchronization to GNSS-PPS signals gives the ability to determine the exact firing time of each laser pulse which should be received by the interface box of the LiDAR sensor. Furthermore, the navigation unit provides a navigation message, also known as Recommended minimum specific GPS/Transit data (GPRMC message). This navigation message includes information regarding the real-time position, rotation, and GPS time. It is worth noting that the reception of the GPRMC message must conclude less than 500ms after the rising edge of the PPS signal as is clearly depicted in Fig. 3.11.

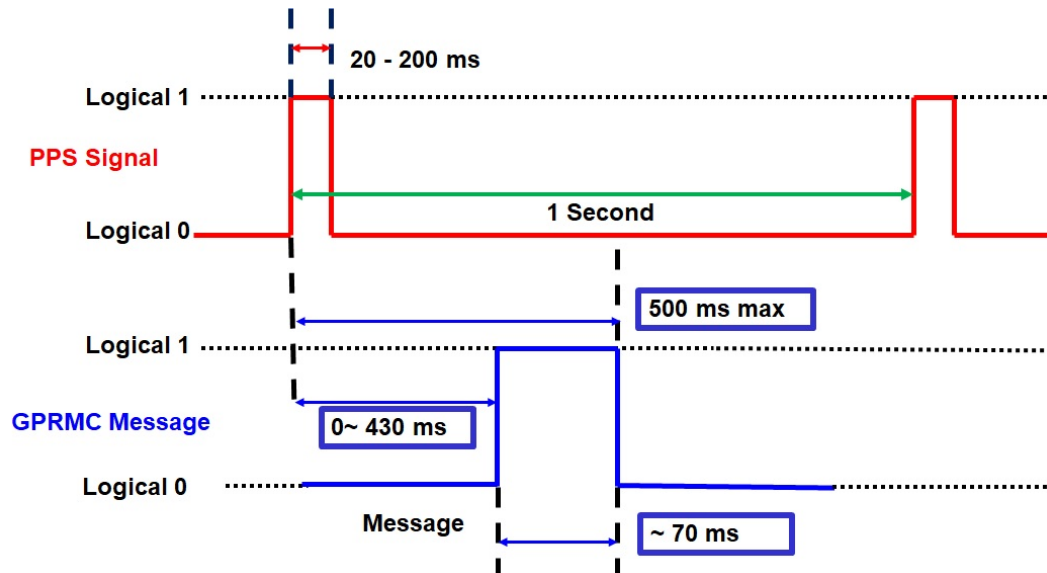


Fig. 3.11.: Synchronization process of LiDAR-based mapping system

Currently, the GPRMC message is recorded over a dedicated RS-232 serial port and received by the LiDAR unit via the interface box in the form of serial data, which should be at 9600 baud rate. Fig. 3.12 shows the block diagram of the DJI M600 Pro UAV-based MMS. Such a block diagram indicates triggering signals, feedback signals, and communication wires/ports between sensors and power connections.

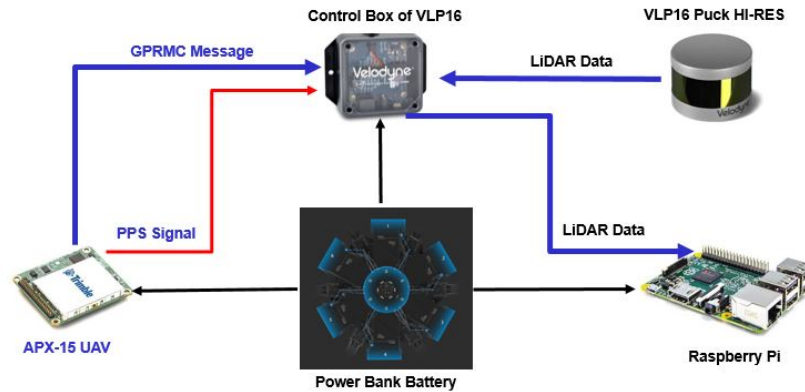


Fig. 3.12.: Integration scheme for the DJI M600 Pro UAV-based system

It is important to explain that some procedures should be considered before data acquisition using a UAV-based LiDAR system such as the effectiveness of the synchronization process. Such a procedure is highlighted to ensure the stability of the system integration architecture between different sensors onboard the UAV platform. Since the synchronization process between the GNSS/INS unit and LiDAR sensor is very crucial as previously mentioned, it is highly recommended to check the process via the webpage application of a LiDAR unit. Through this webpage, one should note a full demonstration related to the reception of the PPS signal and the GPRMC message which includes the real-time navigation solution from the GNSS/INS unit. The synchronization process validation is illustrated in Fig. 3.13.

Furthermore, Wireshark software is another tool to ensure the synchronization between the GNSS/INS unit and LiDAR sensor [118]. Such software is mainly used for recording the raw data of LiDAR unit. So, one should ensure the reception of



GPRMC message by checking the position packet of LiDAR's raw data as shown below in Fig. 3.14.

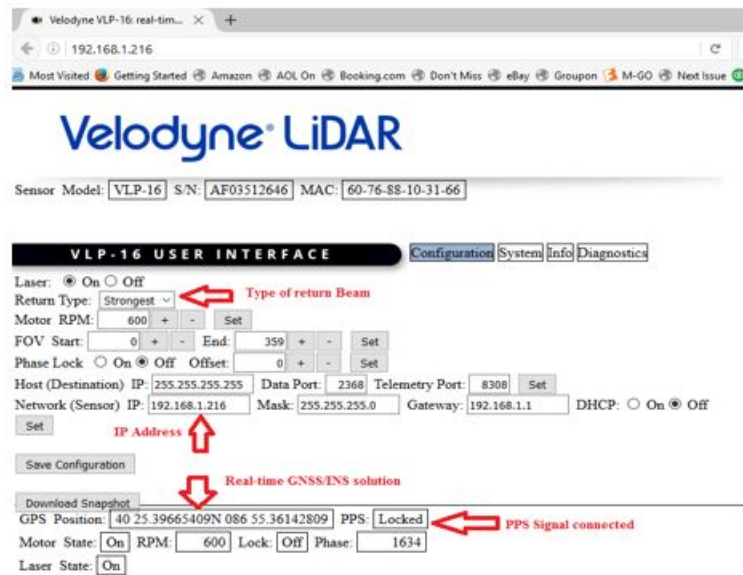


Fig. 3.13.: The Velodyne webpage application for checking the reception of the PPS signal and the GNRMC message

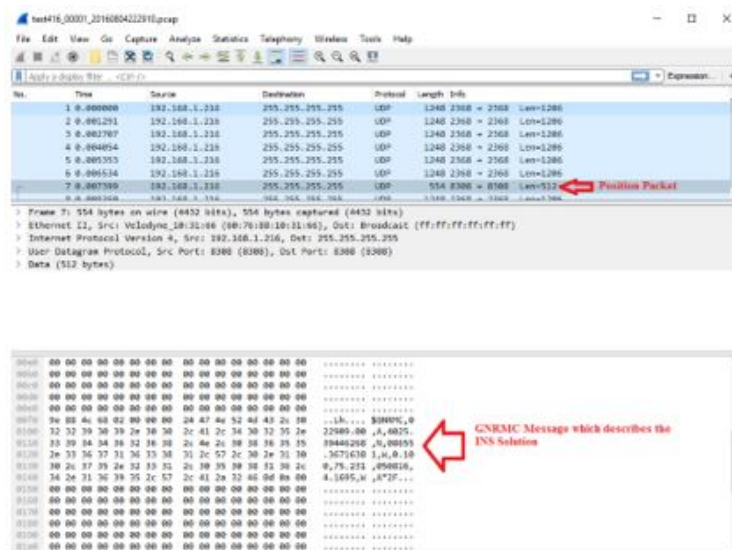


Fig. 3.14.: The Wireshark software to ensure the reception of GPRMC message

### 3.3.2 Data Storage Unit

Regarding raw data, the Velodyne HDL-32E and VLP-16 Puck/Puck-Hi-Res have about 8 Megabits/Second data rate. Therefore, LiDAR sensors need an appropriate media for storing raw data. Minicomputers are investigated for this purpose, but the standard weight is about 500 gm, which means almost 10% of the UAV payload is used. As a result, a Raspberry Pi 3 (weighing about 50 gm) with 1.2 GHz 64-bit quad-core ARMv8 CPU is used [119]. Its small size and light weight make for a simple installation on the UAV and it is also cost effective. This choice saved around 90% of the weight compared with a mini-computer. Fig. 3.15 shows the data storage unit onboard UAV-LiDAR system.

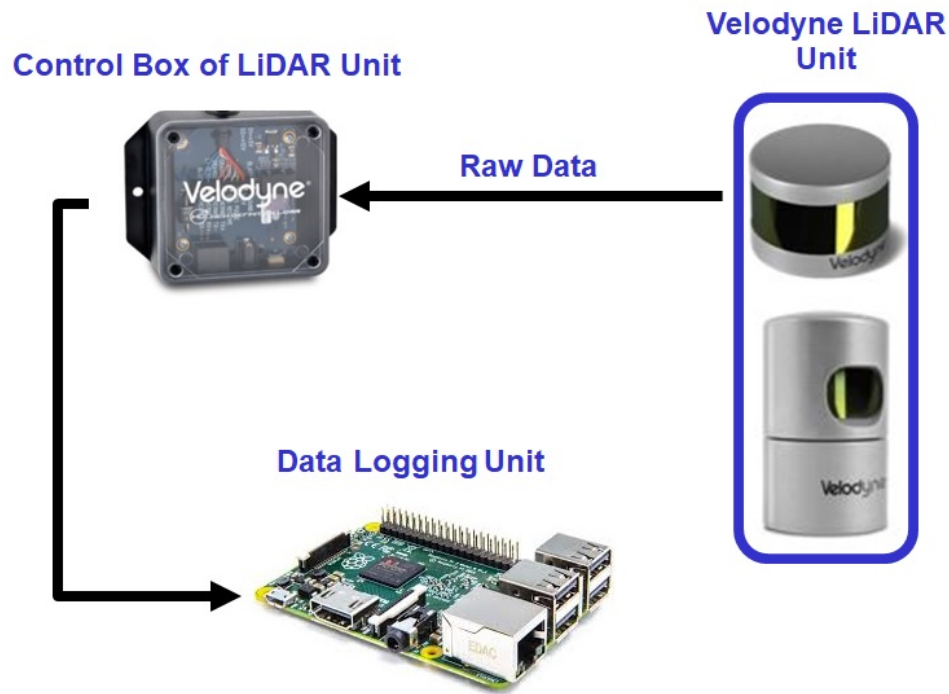
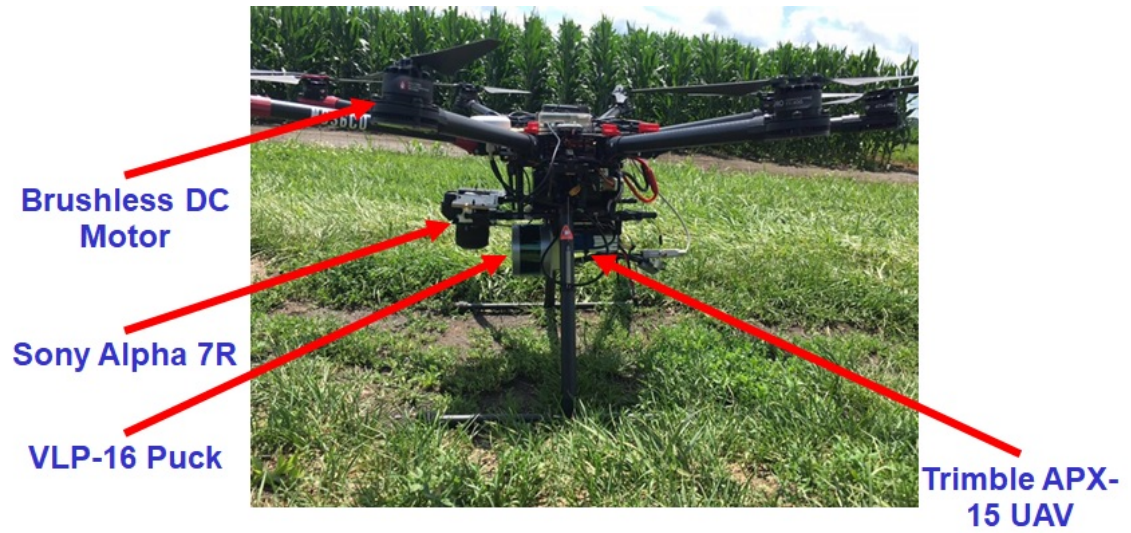


Fig. 3.15.: Data storage unit onboard UAV-LiDAR system

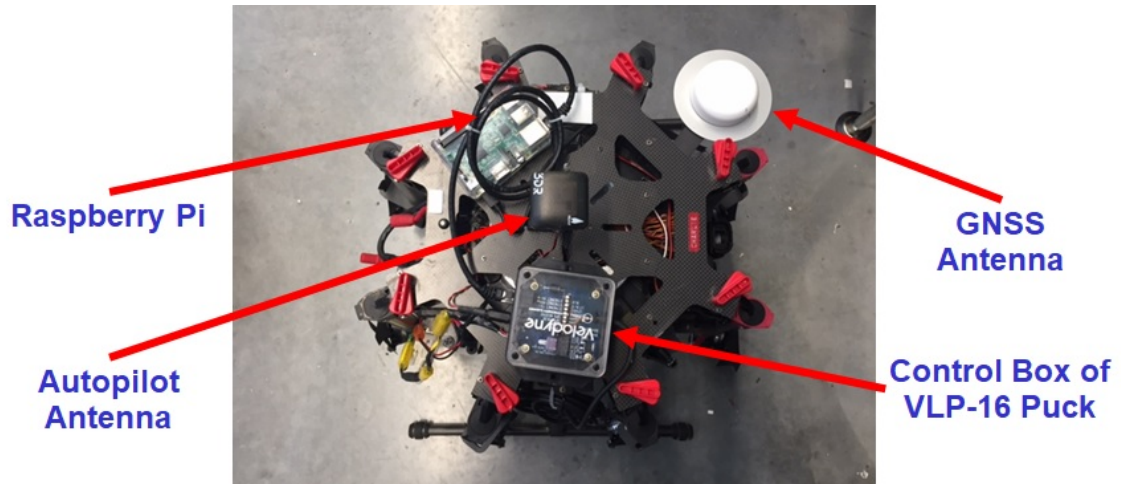
### 3.3.3 System Interference

System interference is considered to be one of the most important challenges that is encountered in the UAV community. Such an issue has a significant effect on the performance of the aerial platform as well as the generated trajectory from the direct geo-referenced unit (GNSS/INS). The GNSS sensor is a pivotal part of the UAV platform which is used to guide the UAV and ensure following the pre-defined flight path. For autonomously controlled systems, the UAV can be stabilized by using inertial sensors in case there is an absence of a GNSS signal, but it still cannot navigate to the landing spot without the involvement of a pilot. The majority of aerial mapping platforms use dual frequency (L1/L2) Real Time Kinematic (RTK) receivers due to the ability to provide a reliable position accuracy. To achieve a high precision navigation, the phase based techniques are required. Phase-based techniques need good signal quality and signal availability, and it is considered a challenging task with UAV platform [120].

The system interference phenomena can be classified into internal interference and external interference. For the internal interference, the UAV platform itself is considered one of the obvious sources of interference due to limited space available for the installation of several sensors such as the Velodyne VLP-16 Puck & Puck Hi-Res, the Sony Alpha 7R (RGB camera), and the Trimble APX-15 UAV coupled with different technical specifications and operations of such sensors. This limited space causes the GNSS antenna to be physically close to the onboard sensors and different electrical circuits as depicted in Fig. 3.16(a), Fig. 3.16(b), Fig. 3.17(a), and Fig. 3.17(b). Therefore, the probability of exposure to the Electro-Magnetic Compatibility (EMC) increases and has a negative impact on the GNSS receiver performance.

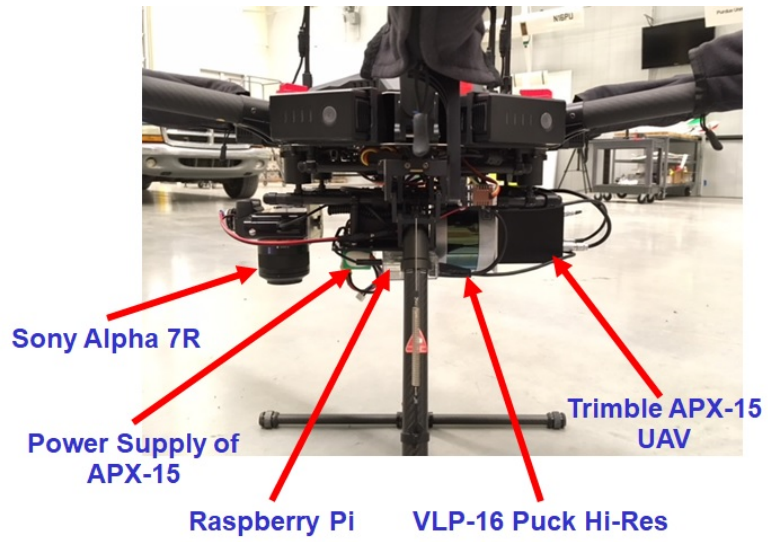


(a)

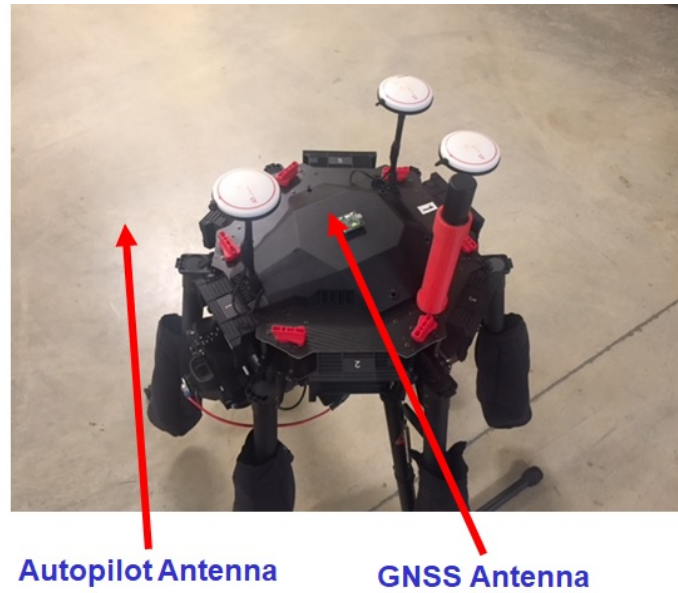


(b)

Fig. 3.16.: Physical installation of several sensors onboard DJI S1000+ (a) Side view- (b) Top view



(a)



(b)

Fig. 3.17.: Physical installation of several sensors onboard DJI M600 Pro (a) Side view- (b) Top view

In addition, the rotorcraft brushless DC motor is also one of the main sources in the sense of internal interference, which causes an interference effect on the GNSS



antenna when the motor is powered up. In terms of external interference, there is a suspicion of terrestrial vehicle GPS jammers which are initially designed to disable GPS reception and might be an additional source of interference [120]. During the UAV flight mission, if a vehicle equipped with a GPS jammer crosses the flight track, it could affect the navigation process. In response to that, there are two modifications considered to avoid the impact of the EMC on the GNSS receiver. First, there is an initial design of the GNSS antenna mast that keeps the antenna close to the onboard sensors. Therefore, a new design of that mast is implemented to be much taller than the initial design for keeping the GNSS antenna physically away from the sensors onboard the UAV platform as shown in Fig. 3.18. Second, the unreasonable wiring or incorrect wire connection between the onboard sensors as well as the electronic circuits increases the impact of the EMC and has a negative effect on the performance of such sensors. Thus, the Trimble APX-15 UAV unit has been fitted in a housing box to accommodate a wiring harness as well as the necessary electronic circuits with reasonable wiring connections as shown in Fig. 3.19.

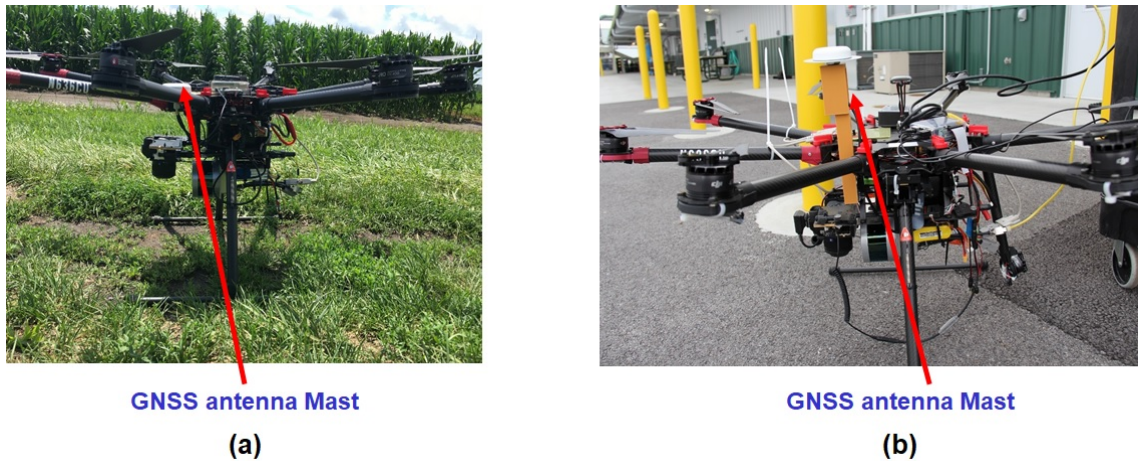


Fig. 3.18.: The GNSS antenna mast design (DJI S1000+) – (a) Initial design -(b) New design



Fig. 3.19.: DJI M600 Pro based LiDAR mapping system – (a) Old installation –(b)  
New housing box

### 3.4 System Integration of Wheel-based LiDAR system

In this dissertation, two examples of the wheel-based LiDAR systems will be discussed. The high clearance tractor (denoted here forth as a PhenoRover-based system) which is used for agriculture management and the car mount system which is used for transportation corridor monitoring applications. Both systems are used to collect LiDAR and photogrammetric data for 3D-point cloud generation. The MMS onboard the PhenoRover as well as the car mount system are developed in the same way as the UAV-based LiDAR system but with minor differences regarding the choice of the sensors that can work effectively for a terrestrial mapping system. Furthermore, unlike the UAV-based MMS, both MMS have no payload restrictions, such as sensor weight, size or power consumption. Such advantage enables the installation of a more accurate GNSS/INS unit, irrespective of its weight or size. Moreover, a computer is used instead of Raspberry Pi to store the LiDAR and image data, which in turn allows more extended missions. The mapping system onboard the PhenoRover has two Velodyne HDL-32E laser scanners, which are directly georeferenced by the Trimble POS LV-125 unit as depicted in Fig. 3.20. For the POS LV-125, the post-processing

accuracy in position can be  $\pm 2\text{-}5\text{ cm}$ , and the achieved accuracy for the roll/pitch and heading can be  $\pm 0.025^\circ$  and  $0.06^\circ$ , respectively. In case of 60 seconds GNSS outages, the post-processing accuracy in position can be  $\pm 0.2\text{-}0.8\text{ m}$ , and the achievable accuracy for the roll/pitch and heading can be  $\pm 0.05^\circ$  and  $0.2^\circ$ , respectively. The detailed specifications of the Trimble POS LV-125 are illustrated in Table 3.7 [121].



Fig. 3.20.: The Trimble POSLV-125 GNSS/INS Module

Table 3.7.: The Trimble POSLV-125 Specifications

Feature	POSLV-125
Position Accuracy (post processing)	$\pm 0.02\text{-}0.05$
Roll & Pitch Accuracy (post processing)	$\pm 0.025^\circ$
True Heading Accuracy (post processing)	$\pm 0.06^\circ$
Voltage Input	+9 to +34 VDC
Power Consumption	60 watt
Weight	1.3 Kg
Dimensions	164 L x 160 W x 66 H mm (nominal)
Data Logging media	Internally, or via Ethernet.
Number of GNSS Receivers	2
Distance Measurement Indicator (DMI)	Support for optional DMI input
Working Environment Temperature	$-40^\circ\text{ C}$ to $+60^\circ\text{ C}$



In addition to that, the mapping system onboard the PhenoRover is mainly developed for the agriculture management applications as mentioned before. It is worth noting that agriculture management has gained tremendous interest from researchers in plant science as well as other fields due to the global challenge of ensuring crop yield for food and fuel generation. In this dissertation, a PhenoRover-based MMS is mainly focusing on accelerating energy crop development for the production of renewable transportation fuels. An initial boom design is implemented to install the mapping sensors onboard for such application. In order to have a better data acquisition geometry, a new boom design is implemented since it can provide more height and distance between the mapping sensors and the plants as depicted in Fig. 3.21.



Fig. 3.21.: PhenoRover-based MMS-(a)Old boom design-(b)New boom design

The car mount system is considered another terrestrial mapping system that is mainly designed for collecting 3D spatial data geo-referenced to a global reference frame, and is used for several applications such as transportation corridor monitoring, and infrastructure monitoring as mentioned before. In this regard, a compact portable system design is implemented. The portable system can be easily installed on any vehicle with a smooth roof top and gives the ability not to be restricted for any particular vehicle as shown in Fig. 3.22.



Fig. 3.22.: The car-mount mapping system (Portable system)

The portable system has two Velodyne HDL-32E laser scanners and two FLIR Flea-2G camera, which are directly georeferenced by the Novatel Span-CPT unit. Such unit is depicted in Fig. 3.23. For the Novatel Span-CPT, the post-processing accuracy in position can be less than  $\pm 2$  cm, and the achieved accuracy for the roll/pitch and heading can be  $\pm 0.008^\circ$  and  $0.035^\circ$ , respectively. In case of 60 seconds GNSS outages, the post-processing accuracy in position can be  $\pm 0.23$  m, and the achievable accuracy for the roll/pitch and heading can be  $\pm 0.013^\circ$  and  $0.038^\circ$ , respectively. The detailed specifications of the Span-CPT are illustrated in Table 3.8 [122].



Fig. 3.23.: The Novatel SPAN-CPT GNSS/INS unit

Table 3.8.: The Novatel SPAN-CPT Specifications

Feature	SPAN-CPT
Position Accuracy (post processing)	$\pm 0.02$
Roll & Pitch Accuracy (post processing)	$\pm 0.008^\circ$
True Heading Accuracy (post processing)	$\pm 0.035^\circ$
Voltage Input	+9 to +18 VDC
Power Consumption	16 watt
Weight	2.28 Kg
Dimensions	152 L x 168 W x 89 H mm (nominal)
Data Logging media	USB or Serial connections.
Number of GNSS Receivers	1
Distance Measurement Indicator (DMI)	No Support for optional DMI input
Working Environment Temperature	$-40^\circ\text{C}$ to $+65^\circ\text{C}$

As far as durability is concerned, a new rack design of a dedicated platform is implemented to install several mapping sensors onboard as well as to provide flexibility for adjusting the mounting of such sensors in different manners. To be more specific, such mounting flexibility ensures the ability of the dedicated platform-based MMS to meet the requirements of different mapping applications with respect to having the useful and appropriate data for each activity. Fig. 3.24 shows the Purdue Wheel-based MMS (PWMMS) which is considered the dedicated platform-based MMS.

The dedicated platform-based MMS has two Velodyne HDL-32E laser scanners and three FLIR Flea-2G camera, which are directly georeferenced by the Trimble POS LV-220 unit. Such unit is depicted in Fig. 3.25. For the POS LV-220, the post-processing accuracy in position can be less than  $\pm 2$  cm and the achieved accuracy for the roll/pitch and heading can be  $\pm 0.02^\circ$  and  $0.025^\circ$ , respectively. In case of 60 seconds GNSS outages, the post-processing accuracy in position can be  $\pm 0.24$  m,

and the achievable accuracy for the roll/pitch and heading can be  $\pm 0.06^\circ$  and  $0.03^\circ$ , respectively. The detailed specifications of the Trimble POS LV-220 are illustrated in Table 3.9 [121].



Fig. 3.24.: Purdue Wheel-based mobile mapping System (PWMMS)



Fig. 3.25.: The Trimble POSLV-220 GNSS/INS Module

One should note that there is a similarity between the system integration developed for the wheel-based MMS framework and UAV platforms. Such similarity in the system integration methodology allows for the implementation of the same technique explained above for the wheel-based LiDAR systems. The system integration has been developed in a way which demonstrates the flexibility of the proposed work. To be more specific, the developed system integration can be used for not only airborne

mapping systems but also for other systems such as the wheel-based LiDAR systems, with similar mapping sensor structure and techniques. Fig. 3.26 illustrates the block diagram of the wheel-based LiDAR system (the PhenoRover-based MMS and the dedicated platform-based MMS). Such block indicates triggering signals, feedback signals, and power connections.

Table 3.9.: The Trimble POSLV-220 Specifications

<b>Feature</b>	<b>POSLV-220</b>
Position Accuracy (post processing)	$\pm 0.02$
Roll & Pitch Accuracy (post processing)	$\pm 0.02^\circ$
True Heading Accuracy (post processing)	$\pm 0.025^\circ$
Voltage Input	+10 to +34 VDC
Power Consumption	60 watt
Weight	2.4 Kg
Dimensions	167 L x 185 W x 68 H mm (nominal)
Data Logging media	Internally, via USB or Ethernet.
Number of GNSS Receivers	2
Distance Measurement Indicator (DMI)	Support for optional DMI input
Working Environment Temperature	$-40^\circ\text{C}$ to $+60^\circ\text{C}$

In this dissertation, a multi-sensor integration system is developed onboard the wheel-based LiDAR systems. Since the connection of multi-LiDAR sensors (either HDL-32E or VLP-16 Puck/Puck Hi-Res) causes data interference between the connecting LiDAR units and the LiDAR data could be lost. Therefore, the connection of multi-LiDAR sensors to a single personal computer (PC) is considered quite problematic. The LiDAR sensors usually connect to the PC for recording the data through RJ45 port (Ethernet port) with a particular IP address by using User Datagram Pro-

toocol (UDP). The recorded data can interfere with each other and cannot be extracted properly due to network activity from another LiDAR unit. To be more specific, having two sensors sending data to a broadcast address will cause data loss. In order to use multi-LiDAR sensors on a single PC, multiple Ethernet ports should be used (one for each sensor) as depicted below in Fig. 3.27. Furthermore, each Ethernet port should be set to a different IP address. One should mention that the Ethernet ports should be configured in a way to use the matching IP address of each sensor particularly.

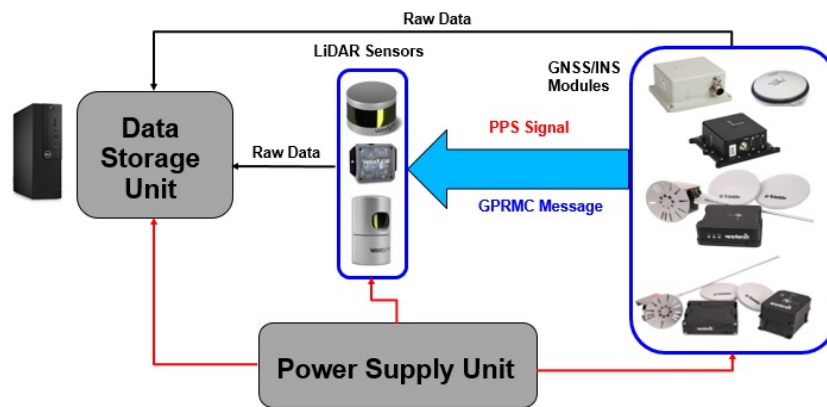


Fig. 3.26.: Integration scheme for the wheel-based LiDAR systems

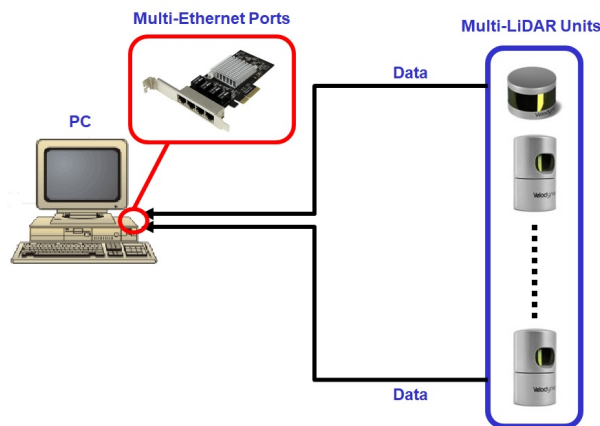


Fig. 3.27.: Integration of multi-LiDAR sensors to single PC through multiple Ethernet ports

It is worth noting that the integration stability between multi-LiDAR sensors and a GNSS/INS unit is considered an important part of this research. One should mention that the mapping system cannot rely on directly splitting the PPS signal to feed multi-LiDAR sensors to perform the synchronization process due to the insufficient current rating. Therefore, the embedded system modules such as the Raspberry Pi kit or the Arduino kit could be used to split the PPS signal in a way that keeps the current rating of such signals to successfully synchronize the LiDAR sensors as depicted in Fig. 3.28.

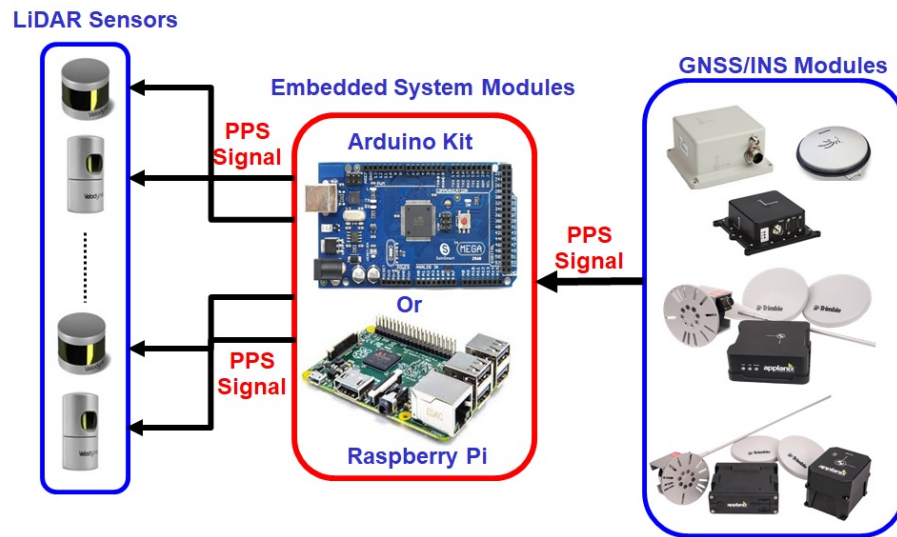


Fig. 3.28.: Integration of multiple-LiDAR sensors with GNSS/INS module through embedded system electronic kits

### 3.5 Point-Cloud Reconstruction

As mentioned earlier, the integration between the direct geo-referencing unit and the laser scanning sensors is performed for the LiDAR mapping system onboard an airborne/terrestrial platform. However, there is a distinct data rate difference between the integrated mapping sensors. More specifically, the GNSS/INS module determines

the position and orientation of the IMU body frame at 100 Hz to 200 Hz data rate while the data rate of the LiDAR sensor is from 300KHz to 1MHz. Due to the higher rate of LiDAR data compared to the derived data from GNSS/INS, an interpolation process should be performed to derive the position and orientation of the platform for each laser pulse as illustrated in Fig. 3.29, where  $t_i$  denotes the time of LiDAR data acquisition epoch.

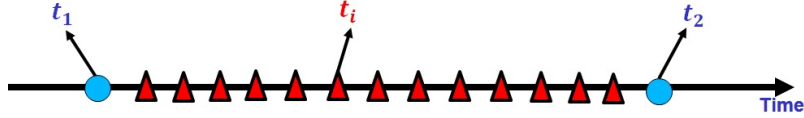


Fig. 3.29.: Illustration of the GNSS/INS data interpolation

For position interpolation, a linear interpolation is performed between every two points in the derived trajectory where the LiDAR point of interest lies in between as shown in Equation 3.1. The  $r_b^m(t_1)$  and  $r_b^m(t_2)$  denote the position of the IMU body frame relative to the mapping reference frame at GPS time  $t_1$  and GPS time  $t_2$  respectively, and  $r_b^m(t_i)$  represents the position of the IMU body frame relative to the mapping reference at the desired point timestamp ( $t_i$ ).

$$r_b^m(t_i) = \frac{t_2 - t_i}{t_2 - t_1} * r_b^m(t_1) + \frac{t_i - t_1}{t_2 - t_1} * r_b^m(t_2) \quad (3.1)$$

For deriving rotation matrices, a spherical linear interpolation method is performed using quaternion representation as depicted in Fig. 3.30. A quaternion is a 4D-vector that has one real and three imaginary parts as described in Equation 3.2. The  $i$ ,  $j$ , and  $k$  denote the terms of imaginary parts of a quaternion vector. Equations 3.3 - 3.6 show the properties of the quaternion vector. One should note that the real part for the pure quaternion is zero. Furthermore, the relationship between the quaternions and the rotation matrices is illustrated in Equations 3.7 - 3.15 [123].

$$q = q_0 + q_x i + q_y j + q_z k \quad (3.2)$$



$$i^2 = j^2 = k^2 = ijk = -1 \quad (3.3)$$

$$i = jk = -kj \quad (3.4)$$

$$j = ki = -ik \quad (3.5)$$

$$k = ij = -ji \quad (3.6)$$

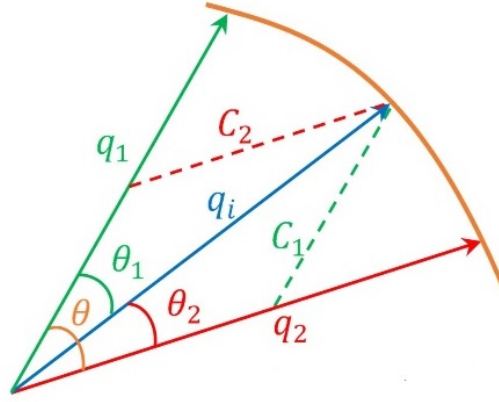


Fig. 3.30.: Spherical Linear Interpolation Method

$$r_{11} = q_x^2 + q_0^2 - q_z^2 - q_y^2 \quad (3.7)$$

$$r_{12} = 2q_xq_y - 2q_0q_z \quad (3.8)$$

$$r_{13} = 2q_xq_z + 2q_0q_y \quad (3.9)$$

$$r_{21} = 2q_xq_y + 2q_0q_z \quad (3.10)$$

$$r_{22} = q_y^2 - q_z^2 + q_0^2 - q_x^2 \quad (3.11)$$

$$r_{23} = 2q_yq_z - 2q_0q_x \quad (3.12)$$

$$r_{31} = 2q_xq_z - 2q_0q_y \quad (3.13)$$

$$r_{32} = 2q_yq_z + 2q_0q_x \quad (3.14)$$

$$r_{33} = q_z^2 - q_y^2 - q_x^2 + q_0^2 \quad (3.15)$$

In the spherical linear interpolation method, the interpolated quaternion rotation  $q_i$  needs to be evaluated based on the known rotations represented by  $q_1$  and  $q_2$ , whose angular deviation is  $\theta$ . The angular deviations of the interpolated quaternion relative to  $q_1$  and  $q_2$  are  $\theta_1$  and  $\theta_2$ , respectively. The angular deviation  $\theta$  can be derived using Equation 3.16 and Equation 3.17. The angular deviations  $\theta_1$  and  $\theta_2$  should be defined based on the corresponding derived GPS times  $t_1$  and  $t_2$  from GNSS/INS trajectory at the required interpolated time ( $t_i$ ) as illustrated in Equation 3.18 and Equation 3.19 respectively. To compute  $q_i$  which denotes the interpolated quaternion,  $C_1$  and  $C_2$  as illustrated in Fig. 3.31 and Fig. 3.32 should be derived first through Equation 3.20 and Equation 3.21, respectively. Then, the interpolated quaternion  $q_i$  can be determined through Equation 3.22 and Equation 3.23, respectively.

$$q_1 \cdot q_2 = \|q_1\| * \|q_2\| \cos(\theta) \quad (3.16)$$

$$q_1 \cdot q_2 = \cos(\theta) \quad (3.17)$$

$$\theta_1 = \frac{t_i - t_1}{t_2 - t_1} \theta \quad (3.18)$$

$$\theta_2 = \frac{t_2 - t_i}{t_2 - t_1} \theta \quad (3.19)$$

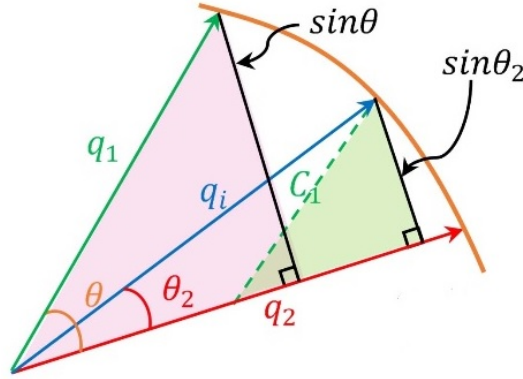


Fig. 3.31.: Interpolated quaternion  $q_i$  computation based on the  $C_1$  derivation

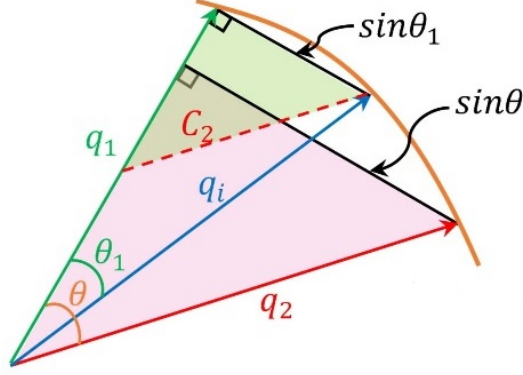


Fig. 3.32.: Interpolated quaternion  $q_i$  computation based on the  $C_2$  derivation

$$C_1 = \frac{\sin \theta_2}{\sin \theta} \quad (3.20)$$

$$C_2 = \frac{\sin \theta_1}{\sin \theta} \quad (3.21)$$

$$q_i = C_1 q_1 + C_2 q_2 \quad (3.22)$$

$$q_i = \frac{\sin \theta_2}{\sin \theta} q_1 + \frac{\sin \theta_1}{\sin \theta} q_2 \quad (3.23)$$

One should note that there is no considerable difference between the spherical linear interpolation method and the conventional linear interpolation technique in case of having a smooth motion of the utilized platform. However, in case that such a platform maneuvers while in motion, the conventional linear interpolation method cannot feed accurate results compared to the spherical linear interpolation technique.

### 3.6 Summary

In this Chapter, a system architecture for a low-cost UAV mapping system with price range less than \$15,000 using directly geo-referenced active ranging systems is developed, while considering the challenges posed by using consumer-grade sensors with accuracy less than 5 cm, platform payload restrictions, and endurance. Therefore, the specifications of the necessary components that are commensurate with the challenges previously mentioned are investigated to meet the needs of the

non-traditional mapping community. Also, a system integration is developed by performing the synchronization process between a direct geo-referencing unit and an active optical ranging sensor onboard the proposed UAV platform. Furthermore, the impact of system interference on the performance of a UAV platform as well as the GNSS/INS trajectory is investigated. In addition to that, the development of a system architecture for a LiDAR-based wheel mapping system is developed in a way that is similar to a LiDAR-based UAV system integration.

## 4. BIAS IMPACT ANALYSIS AND LIDAR SYSTEM CALIBRATION \*

### 4.1 Introduction

Recent advances in hardware and software development have made it possible to conduct accurate 3D-mapping without using costly and high-end data acquisition systems. For example, low-cost laser scanners, and navigation systems can provide accurate mapping if they are effectively integrated at the hardware and software levels. Moreover, ongoing developments in mobile mapping technology have made accurate 3D mapping more feasible, specifically UAVs that are emerging as an economical and practical mobile mapping platform. To derive point clouds with high positional accuracy, estimation of mounting parameters relating the laser scanners to the onboard GNSS/INS unit is the most crucial step.

This phase of research proposes a LiDAR system calibration approach for a UAV-based MMS. The purpose of system calibration is to simultaneously estimate the mounting parameters (i.e., the lever-arm and boresight angles) relating the different system components through an outdoor calibration procedure. Such parameters minimize discrepancies between derived surfaces from multiple flight lines while reducing ground control requirements as depicted in Fig. 4.1. This approach is based on the use of conjugate planar/linear features in overlapping point clouds derived from different flight lines. Designing an optimal/minimal flight and feature configuration for

---

\*THIS CHAPTER IS LARGELY BASED ON THE JOURNAL PAPERS(RAVI, RADHIKA, TAMER SHAMSELDIN, MAGDY ELBAHNASAWY, YUN-JOU LIN, AND AYMAN HABIB. "BIAS IMPACT ANALYSIS AND CALIBRATION OF UAV-BASED MOBILE LIDAR SYSTEM WITH SPINNING MULTI-BEAM LASER SCANNER." APPLIED SCIENCES 8, NO. 2 (2018): 297. AND RAVI, RADHIKA, YUN-JOU LIN, MAGDY ELBAHNASAWY, TAMER SHAMSELDIN, AND AYMAN HABIB. "BIAS IMPACT ANALYSIS AND CALIBRATION OF TERRESTRIAL MOBILE LIDAR SYSTEM WITH SEVERAL SPINNING MULTIBEAM LASER SCANNERS." IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (2018).

calibration is the first and foremost step in order to ensure the most accurate estimates of mounting parameters. In this regard, optimal denotes decoupling (removing any significant correlation between) various components of the mounting parameters. Such configuration is achieved by conducting an in-depth rigorous theoretical analysis of the potential impact of bias in mounting parameters of a LiDAR unit on the resultant 3D point cloud.

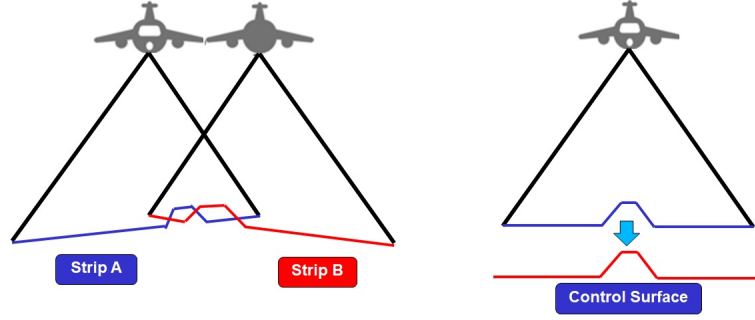


Fig. 4.1.: Conceptual basis of the LiDAR system calibration approach

In this regard, a detailed bias impact analysis facilitates the design of an optimal configuration of target primitives and flight lines for ensuring accurate calibration results. Habib et al. [124] discussed the bias impact analysis in detail for airborne linear scanners while describing the simplified and quasi-rigorous approaches for calibration, whereas, in this research, the bias impact analysis is conducted for a spinning multi-beam laser scanner starting from the 3D- point-positioning equation. The optimal target primitive configuration is devised by studying the impact of biases on planes oriented in different directions and the optimal flight line configuration is determined based on the effect of biases arising from flight lines with different directions and lateral separation on planes with varying orientation. Finally, the proposed analysis and calibration strategy are validated by calibrating a UAV-based LiDAR system using two different datasets—one acquired with flight lines at a single flying height and the other with flight lines at two different flying heights. The calibration performance is evaluated by analyzing correlation between the estimated system parameters, the

a-posteriori variance factor of the Least Squares Adjustment (LSA) procedure, and the quality of fit of the adjusted point cloud to planar/linear features before and after the calibration process.

## 4.2 Bias Impact Analysis

### 4.2.1 Mathematical Model for a UAV-based LiDAR Point Positioning

For conducting the impact analysis of biases in the mounting parameters on the 3D mapping frame coordinates, the first and foremost step is to establish the mathematical relation between these entities. Such entities can further be used to derive the partial derivatives of 3D coordinates with respect to the mounting parameters, thus quantifying the impact of biases in the mounting parameters. A UAV-based LiDAR system consisting of a spinning multi-beam laser scanner involves three coordinate systems – mapping frame, IMU body frame, and laser unit frame – as shown in Fig. 4.2.

The GNSS/INS integration provides the time dependent position,  $r_b^m(t)$ , and rotation,  $R_b^m(t)$ , relating the mapping frame and IMU body frame coordinate systems, according to the optimized solution from the available GNSS and inertial measurements. The laser unit (lu) is related to the IMU body frame by a rigidly defined lever arm,  $r_{lu}^b$ , and boresight matrix,  $R_{lu}^b$ . A point,  $I$ , acquired from the system can be reconstructed in the mapping coordinate system using Equation 4.1. For the laser unit frame, the origin is defined at the laser beams firing point, and the z-axis is along the axis of rotation of the laser unit. For a spinning multi-beam laser unit, each laser beam is fired at a fixed vertical angle,  $\beta_j$  where  $j = 1, 2, \dots, 16$ ; the  $\alpha$  denotes the angle along the axis of rotation; and the range,  $\rho$  is defined by the distance between firing point and its footprint, as shown in Fig. 4.3. So, the coordinates of a 3D point relative to the laser unit coordinate system,  $r_I^{lu}(t)$ , is defined by Equation 4.2.

$$r_I^m = r_b^m(t) + R_b^m(t)r_{lu}^b + R_b^m(t)R_{lu}^b r_I^{lu}(t) \quad (4.1)$$

$$r_I^{lu}(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \rho(t) \cos \beta_j \cos \alpha(t) \\ \rho(t) \cos \beta_j \sin \alpha(t) \\ \rho(t) \sin \beta_j \end{bmatrix} \quad (4.2)$$

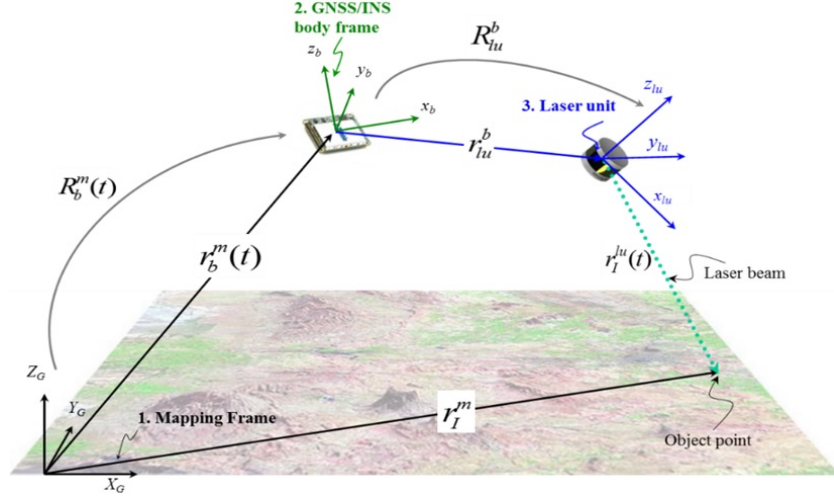


Fig. 4.2.: Illustration of point positioning of a Light Detection and Ranging (LiDAR) system

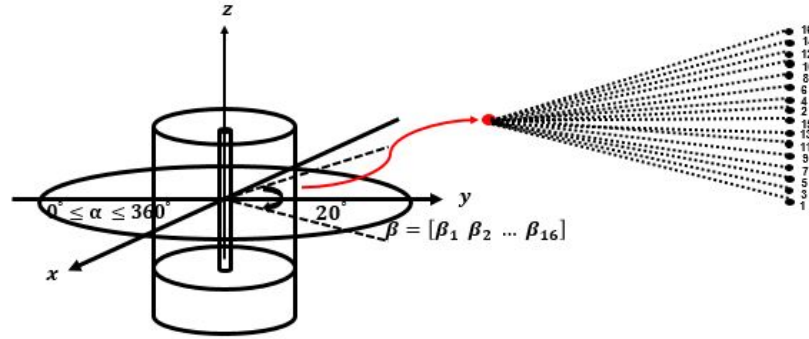


Fig. 4.3.: Illustration of a spinning multi-beam LiDAR unit (VLP-16 Puck Hi-Res)



#### 4.2.2 Bias Impact Analysis for a Spinning Multi-Beam Laser Scanner

The objective of this section is to derive a mathematical formulation that shows the impact of biases in the mounting parameters on the coordinates of points along planar features with different orientations. This analysis can further aid the development of an optimal/minimal flight and target configuration for calibration. Note that planar features are used specifically for this analysis as they facilitate the observation of positional deformations in one direction at a time, i.e., the effect in the direction normal to the plane. For simplifying the bias impact analysis without any loss of generality, several assumptions are made. Firstly, the IMU is mounted on the UAV with its X- and Y-axes aligned along the starboard and flight line directions, respectively. The IMU is assumed to be perfectly vertical, i.e., the z-axis of the IMU body frame is assumed to be perfectly aligned with the vertical direction of the mapping frame. Furthermore, the flight line directions are assumed to be either from South-to-North ( $\kappa = 0^\circ$ ) or from North-to-South ( $\kappa = 180^\circ$ ). These assumptions facilitate the decision as to whether the impact is along/across the flight line and vertical directions. As a result, the rotation matrix  $R_b^m(t)$  would be given by Equation 4.3, where the top and bottom signs are for S-N and N-S flight line directions, respectively. In order to generalize the analysis regardless of the orientation of the LiDAR unit relative to the IMU body frame, Equation 4.1 is slightly modified by introducing a virtual LiDAR unit frame,  $lu'$ , which is almost aligned with the IMU body frame. Moreover, the use of a virtual LiDAR unit frame also prevents gimbal lock in the mounting parameter estimation. This modification is implemented by expressing the term  $R_{lu}^b$  in Equation 4.1 as:  $R_{lu}^b = R_{lu'}^b R_{lu'}^{lu'}$ , where  $R_{lu'}^{lu'}$  is defined according to the laser scanner unit alignment relative to the IMU. The modified LiDAR point positioning is given by Equation 4.4. The LiDAR unit coordinate system alignment on the UAV platform used in this system and the assumed IMU body frame coordinate system (with the X, Y, Z axes along starboard, forward, and up directions, respectively) are shown in Fig. 4.4.

$$R_b^m(t) = \begin{bmatrix} \pm 1 & 0 & 0 \\ 0 & \pm 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

$$r_I^m = r_b^m(t) + R_b^m(t)r_{lu}^b + R_b^m(t)R_{lu}^b R_{lu}^{lu'} r_I^{lu}(t) \quad (4.4)$$

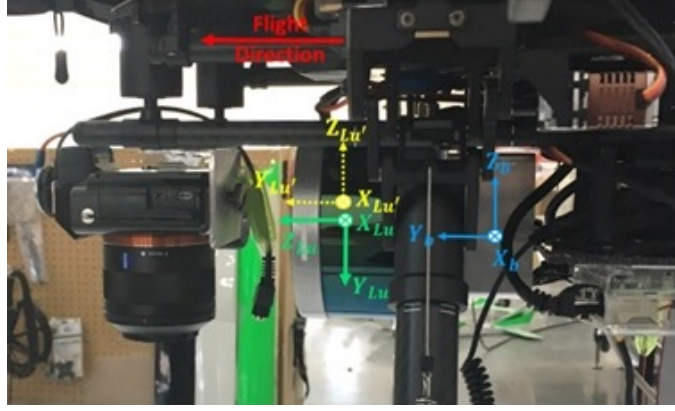


Fig. 4.4.: Coordinate systems for DJI M600 Pro-based LiDAR system

Since the virtual LiDAR unit frame is almost aligned with the IMU body frame, it results in small values for the differential angular boresight parameters  $(\Delta\omega, \Delta\phi, \Delta\kappa)$  relating the two frames. So, the matrix  $R_{lu}^b$  can be written as shown in Equation 4.5, using the small angle approximations. Here,  $\Delta\omega$ ,  $\Delta\phi$ , and  $\Delta\kappa$  denote the rotation around the X, Y, Z-axes of the IMU body frame (i.e., across flight, along flight, and vertical directions), respectively. Hence, these parameters denote the boresight pitch, roll, and heading angles, respectively. The point coordinates relative to the virtual LiDAR unit frame according to Fig. 4.4, are given by Equation 4.6 and Equation 4.7 in terms of the flying height,  $H \pm \Delta h$ , and scan angles ( $\alpha$  and  $\beta$ ). The  $H$  denotes the flying height above average ground elevation, and  $\Delta h$  denotes the variation in target height. The virtual LiDAR unit frame is almost parallel to the IMU body frame. The schematic illustration of such symbolic notations is depicted in Fig. 4.5 for a UAV-based LiDAR system. Substituting Equation 4.5, Equation 4.6, and Equation 4.7 in

Equation 4.4, the revised form of the LiDAR point positioning equation is derived, as given in Equation 4.8, where  $\Delta X$ ,  $\Delta Y$ ,  $\Delta Z$  are the lever-arm offset parameters of the LiDAR unit frame relative to the IMU body frame. Now, the impact of the presence of bias in the system mounting parameters can be analyzed by differentiating Equation 4.8 with respect to the system mounting parameters and this is given by Equation 4.9.

$$R_{lu'}^b = \begin{bmatrix} 1 & -\Delta\kappa & \Delta\phi \\ \Delta\kappa & 1 & -\Delta\omega \\ -\Delta\phi & \Delta\omega & 1 \end{bmatrix} \quad (4.5)$$

$$r_I^{lu'}(t) = R_{lu'}^{lu} r_I^{lu}(t) = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_{lu'}^{lu} \begin{bmatrix} \rho \cos \beta \sin \alpha \\ \rho \cos \beta \cos \alpha \\ \rho \sin \beta \end{bmatrix} = \begin{bmatrix} \rho \cos \beta \sin \alpha \\ \rho \sin \beta \\ -\rho \cos \beta \cos \alpha \end{bmatrix} \quad (4.6)$$

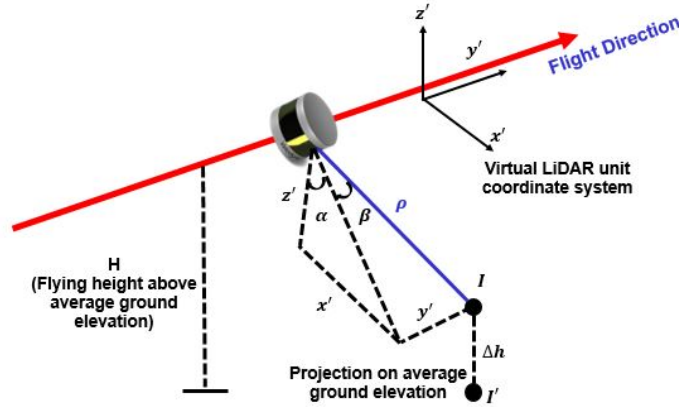


Fig. 4.5.: Schematic diagram illustrating the symbolic notations used for a UAV-based LiDAR system calibration

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = (H \pm \Delta h) \begin{bmatrix} \tan \alpha \\ \frac{\tan \beta}{\cos \alpha} \\ -1 \end{bmatrix} \quad (z' = -\rho \cos \beta \cos \alpha = -(H \pm \Delta h)) \quad (4.7)$$

$$r_I^m = r_b^m(t) + R_b^m(t) \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} + R_b^m(t) \begin{bmatrix} 1 & -\Delta\kappa & \Delta\phi \\ \Delta\kappa & 1 & -\Delta\omega \\ -\Delta\phi & \Delta\omega & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (4.8)$$

$$\delta r_I^m(\delta\Delta X, \delta\Delta Y, \delta\Delta Z, \delta\Delta\omega, \delta\Delta\phi, \delta\Delta\kappa) = \begin{bmatrix} \pm\delta\Delta X \\ \pm\delta\Delta Y \\ \delta\Delta Z \end{bmatrix} + \begin{bmatrix} \pm z' \delta\Delta\phi \mp y' \delta\Delta\kappa \\ \mp z' \delta\Delta\omega \pm x' \delta\Delta\kappa \\ y' \delta\Delta\omega - x' \delta\Delta\phi \end{bmatrix} \quad (4.9)$$

The bias impact can be analyzed thoroughly by isolating the terms in Equation 4.9 corresponding to the impact of bias in each of the mounting parameters for each of the mapping frame coordinates  $X_m$ ,  $Y_m$ , and  $Z_m$ , representing the coordinates across flying direction, along flying direction, and vertical direction, respectively – as given in Table 4.1.

Table 4.1.: Impact of bias in each of the mounting parameters on 3D point coordinates

	$\delta X_m$	$\delta Y_m$	$\delta Z_m$
$\delta\Delta X$	$\pm\delta\Delta X$	0	0
$\delta\Delta Y$	0	$\pm\delta\Delta Y$	0
$\delta\Delta Z$	0	0	$\delta\Delta Z$
$\delta\Delta\omega$	0	$\mp z' \delta\Delta\omega$	$y' \delta\Delta\omega$
$\delta\Delta\phi$	$\pm z' \delta\Delta\phi$	0	$-x' \delta\Delta\phi$
$\delta\Delta\kappa$	$\mp y' \delta\Delta\kappa$	$\pm x' \delta\Delta\kappa$	0

Table 4.1 can now be used to assess the impact of each bias for planar surfaces in different orientations – vertical planes parallel to flight direction, vertical planes perpendicular to flight direction, and horizontal planes) – thus indicating the impact across flight direction, along flight direction, and vertical direction, respectively.

### Impact of Bias in Lever-arm component across the flying direction ( $\Delta X$ )

A bias in this component ( $\delta\Delta X$ ) will introduce a constant shift ( $\pm\delta\Delta X$ ) across the flying direction as depicted in Fig. 4.6. The introduced shift is flying direction dependent and does not depend on the location of the point in question relative to the virtual laser unit coordinate system. As a result, its impact will be visible in case of having vertical planes parallel to the flying direction scanned from two flight lines in opposite directions.

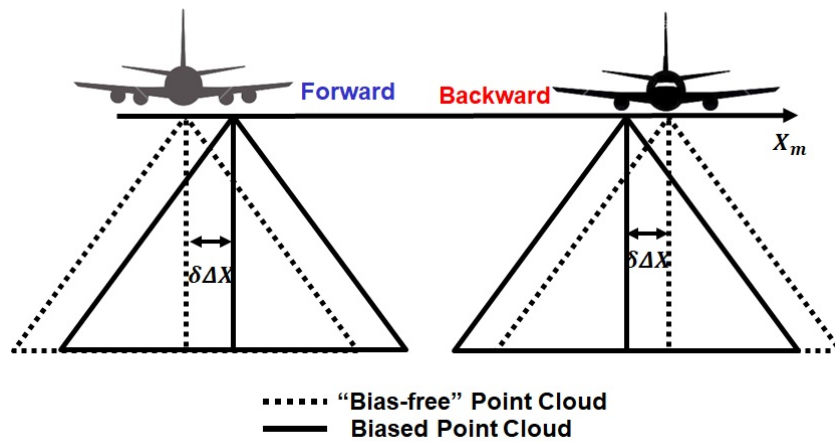


Fig. 4.6.: Impact of bias in lever-arm component across the flying direction ( $\Delta X$ )

### Impact of Bias in Lever-arm component along the flying direction ( $\Delta Y$ )

A bias in this component ( $\delta\Delta Y$ ) will introduce a constant shift ( $\pm\delta\Delta Y$ ) along the flying direction as shown in Fig. 4.7. The introduced shift is flying direction dependent and does not depend on the location of the point in question relative to the virtual laser unit coordinate system. Therefore, it would impact vertical planes perpendicular to the flying direction scanned from two flight lines in opposite directions.

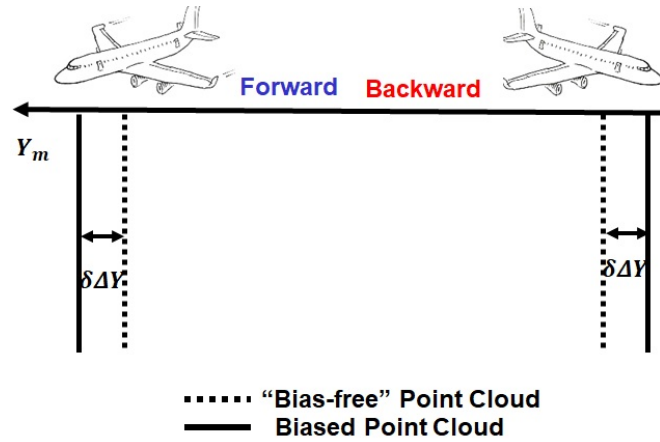


Fig. 4.7.: Impact of bias in lever-arm component along the flying direction ( $\Delta Y$ )

#### Impact of Bias in Lever-arm component in the vertical direction ( $\Delta Z$ )

A bias in this component ( $\delta\Delta Z$ ) will introduce a constant shift ( $\delta\Delta Z$ ) in the vertical direction as shown in Fig. 4.8. The introduced shift is flying direction independent and does not depend on the location of the point in question relative to the virtual laser unit coordinate system. As a result, the entire point cloud would be shifted in the vertical direction by the same amount. So, this bias would not affect planes at any orientation for any flight line configuration.

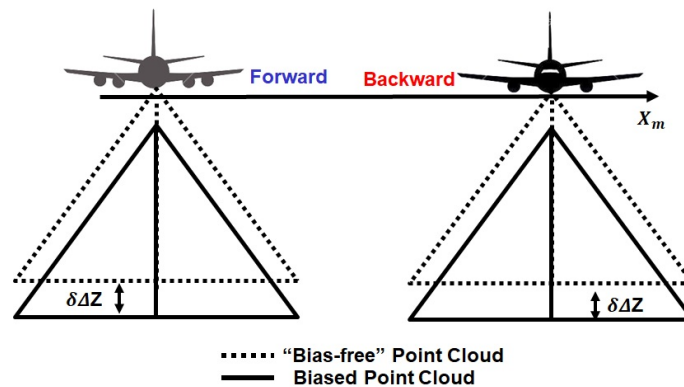


Fig. 4.8.: Impact of bias in lever-arm component in the vertical direction ( $\Delta Z$ )

### Impact of Bias in Boresight Pitch ( $\Delta\omega$ )

A bias in this component ( $\delta\Delta\omega$ ) will cause shifts along the flying direction as well as in the vertical direction as depicted in Fig. 4.9. The impact of boresight pitch bias along the flying direction ( $\mp z'\delta\Delta\omega = \pm(H \pm \Delta h)\delta\Delta\omega$ ) is flying direction dependent and its magnitude depends on the height ( $z'$ ) of the point in question relative to the virtual laser unit coordinate system. This impact would be visible in case of a planar feature perpendicular to the flying direction being scanned by flight lines in opposite directions.

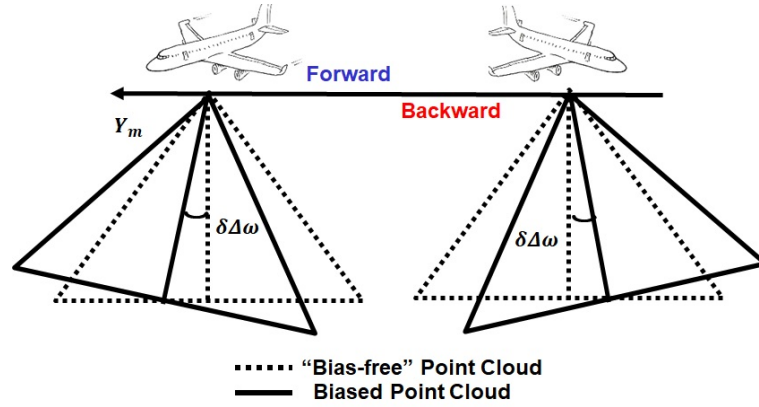


Fig. 4.9.: Impact of bias in boresight pitch ( $\Delta\omega$ )

The impact of boresight pitch bias in the vertical direction ( $y'\delta\Delta\omega$ ) would be manifested in horizontal planes. The magnitude of this impact depends on the  $y'$ -coordinate of the point in question. So, this impact would be visible even in case of a single flight line capturing a horizontal planar feature as long as there is a significant variation in the  $y'$ -coordinate at a given location within such a plane, i.e., if the same portion of the plane is scanned by the laser unit while being at different locations.

Since the bias in lever arm component along the flying direction ( $\Delta Y$ ) also causes shifts along the flying direction, there is a need to decouple the impacts so as to estimate both these biases accurately. The boresight pitch bias has an impact in the vertical direction in addition to the impact along flying direction; therefore, it

would aid in naturally decoupling  $\delta\Delta\omega$  and  $\delta\Delta Y$ , provided there is a significant  $y'$ -coordinate variability. But, the  $y'$ -coordinate variation could be reduced depending on the nature of the targets used or the sensor configuration. For the VLP-16 Puck Hi-Res, the  $y'$ -coordinate variability is limited by the relatively narrow vertical FOV of the LiDAR unit ( $\pm 10^\circ$ ), thus making it insufficient to eliminate the correlation. In such cases, there is a need to have a significant variation in the value of  $(H \pm \Delta h)$  in order to decouple the impacts of  $\delta\Delta Y$  and  $\delta\Delta\omega$  so as to estimate both these biases accurately. This can be achieved by one of the following ways:

- Two different flying heights: The shift caused along the flight direction by the bias  $\delta\Delta\omega$  will vary depending on the flying height, whereas the shift due to the bias  $\delta\Delta Y$  will be constant for any flying height. Thus, the two biases can be derived accurately using flight lines at different flying heights.
- Variation in target height w.r.t. flying height: In case of flight lines at the same flying height, a variation in the height of points along a target primitive would result in varying shifts. The amount of variation required depends on the flying height, i.e.,  $H \pm \Delta h = H(1 \pm \frac{\Delta h}{H})$ . So, the higher the value of  $\frac{\Delta h}{H}$  for a given flying height, the better the estimation accuracy of  $\Delta Y$  and  $\Delta\omega$  will be. A high variation in  $\frac{\Delta h}{H}$  can be achieved by having either vertical planes at different heights or with a significant variation in the heights along given targets.

### **Impact of Bias in Boresight Roll ( $\Delta\phi$ )**

A bias in this component ( $\delta\Delta\phi$ ) will cause shifts across the flying direction as well as in the vertical direction as shown in Fig. 4.10. The impact of this bias across the flying direction ( $\pm z' \delta\Delta\phi = \mp (H \pm \Delta h) \delta\Delta\phi$ ) is flying direction dependent and its magnitude depends on the height ( $z'$ ) of the point in question relative to the virtual laser unit coordinate system. This bias would impact vertical planes parallel to the flying direction scanned from two flight lines in opposite directions.



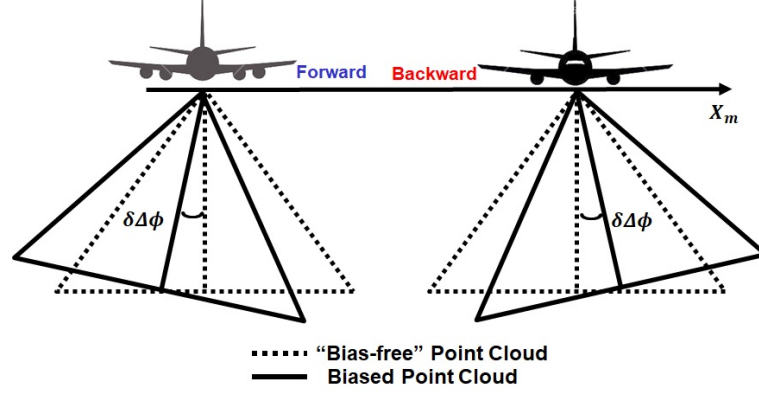


Fig. 4.10.: Impact of bias in boresight roll ( $\Delta\phi$ )

The impact of this bias in the vertical direction ( $-x'\delta\Delta\phi$ ) is flying direction dependent (since  $x'$ -coordinates will change signs depending on the flying direction, as shown in Fig. 4.5) and its magnitude depends on the  $x'$ -coordinate of the point in question. The resultant discrepancy in the Z-coordinate on combining two tracks in the same and opposite directions are given by Equation 4.10 and Equation 4.11, respectively, according to Fig. 4.11. Here,  $D_{AB}$  denotes the lateral distance between the two tracks and  $X$  denotes the distance of the point in question from the line bisecting the lateral distance between the two flight lines. Such analysis reveals that this bias would cause a discrepancy for horizontal planes scanned from two flight lines in the same direction depending on the lateral distance between the tracks. On the other hand, for two tracks in opposite directions, the discrepancy would depend on the lateral location of the planar patch of interest relative to the bisecting direction between the tracks.

$$\delta Z_{mA} - \delta Z_{mB} = (-x'_A + x'_B)\delta\Delta\phi = -D_{AB}\delta\Delta\phi \quad (4.10)$$

$$\delta Z_{mA} - \delta Z_{mB} = (-x'_A + x'_B)\delta\Delta\phi = -2X\delta\Delta\phi \quad (4.11)$$

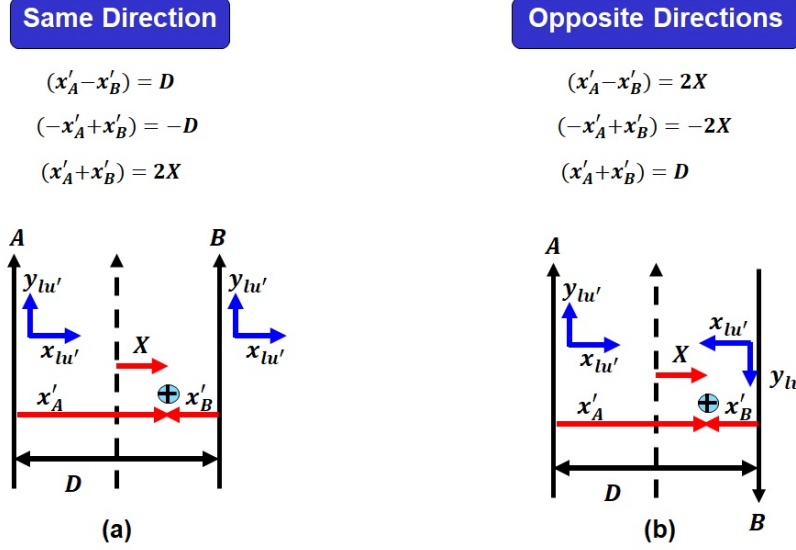


Fig. 4.11.: Relationship between  $x'$ -coordinates for: (a) two tracks in the same direction, and (b) two tracks in opposite directions.

Since a bias in lever arm  $\delta X$  also causes shifts across the flying direction, there is a need to decouple the impacts of  $\delta\Delta X$  and  $\delta\Delta\phi$  so as to estimate both these biases accurately. Due to the impact of boresight roll bias in the vertical direction in addition to the impact in the across flying direction, it would aid in naturally decoupling  $\delta\Delta\phi$  and  $\delta\Delta X$ , provided there are planar patches scanned from flight lines in the same direction with sufficient lateral separation. Also, such impact would aid some planar patches scanned from flight lines in the opposite direction and located at a significantly high lateral distance from the flight lines. However, in case of an unavailability of such planar patches located at a high lateral distance, the decoupling of the two parameters can be achieved by ensuring a significant variation in the value of  $(H \pm \Delta h)$ . This can be achieved by one of the following ways:

- Two different flying heights: The shift caused across the flying direction by the bias  $\delta\Delta\phi$  will vary depending on the flying height, whereas the shift due to the bias  $\delta\Delta X$  will be constant for any flying height. Thus, the two biases can be derived accurately using flight lines at different flying heights.

- Variation in target height w.r.t. flying height: In case of flight lines at the same flying height, a variation in the height of points along a target primitive would result in varying shifts. The amount of variation required depends on the flying height, i.e.,  $H \pm \Delta h = H(1 \pm \frac{\Delta h}{H})$ . So, the higher the value of  $\frac{\Delta h}{H}$  for a given flying height, the better the estimation accuracy of  $\Delta X$  and  $\Delta \phi$  will be.

### Impact of Bias in Boresight Heading ( $\Delta \kappa$ )

A bias in this component ( $\delta \Delta \kappa$ ) will cause shifts across and along the flying direction as depicted in Fig. 4.12 and Fig. 4.13, respectively. The impact of this bias across the flying direction ( $\mp y' \delta \Delta \kappa$ ) is dependent on the  $y'$ -coordinate variability. So, this would cause a discrepancy for vertical planes parallel to the flying direction for a single track. Moreover, the discrepancy on combining two tracks in same or opposite directions would depend on the  $\pm y'$  variability within the points comprising such vertical planes.

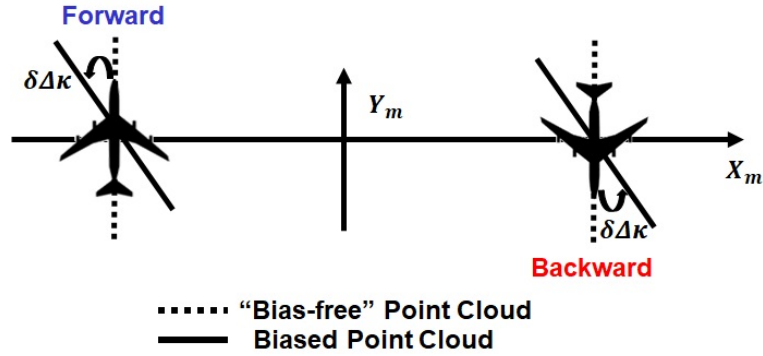


Fig. 4.12.: Impact of bias in boresight roll across the flying direction ( $\Delta \kappa$ )

The impact of this bias along the flying direction ( $\pm x' \delta \Delta \kappa$ ) is flying direction independent since the sign change of  $x'$ -coordinate on flight direction change is nullified by the presence of dual sign in the term. Also, the magnitude of this impact is  $x'$ -coordinate dependent. This bias would induce a discrepancy in case of vertical planes perpendicular to the flying direction scanned from two flight lines in

the same/opposite directions depending on the lateral distance between the tracks, as given by Equation 4.12 and Equation 4.13. For the UAV system used in this study, the impact along flying direction will be more pronounced than the impact in the across flying direction since the LiDAR unit is scanning with the laser beams rotating  $360^\circ$  around the  $y'$ -axis, thus resulting in a high  $x'$ -coordinate variability. However, the  $y'$ -coordinate variability is limited by the total vertical FOV of  $\pm 10^\circ$  of the Velodyne VLP-16 Puck Hi-Res unit.

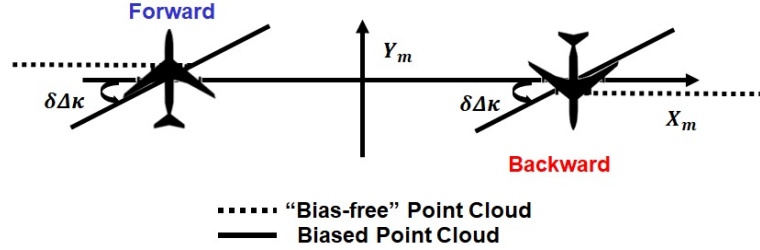


Fig. 4.13.: Impact of bias in boresight roll along the flying direction ( $\Delta\kappa$ )

$$\delta Y_{mA} - \delta Y_{mB} = (x'_A - x'_B)\delta\Delta\kappa = D_{AB}\delta\Delta\kappa \quad (4.12)$$

$$\delta Y_{mA} - \delta Y_{mB} = (x'_A + x'_B)\delta\Delta\kappa = D_{AB}\delta\Delta\kappa \quad (4.13)$$

Throughout the previous discussion, a system where the LiDAR unit coordinate system is not aligned with IMU body frame was proposed by using a virtual laser unit frame. However, the X, Y, Z-axes of the IMU body frame were assumed to be aligned along the starboard, forward and up directions. For other generic situations where the IMU body frame is not aligned in such a manner, a virtual IMU body frame is introduced. For such cases, the LiDAR equation will be modified to result in Equation 4.14. Here,  $R_{y'}^b$  is a fixed rotation depending on the alignment of the actual IMU body frame relative to the UAV vehicle frame. Hence, this modification renders the current bias impact analysis indifferent to the LiDAR unit and IMU body frame alignment within the platform.

$$r_I^m = r_b^m(t) + R_b^m(t)R_b^b r_{lu}^{b'} + R_b^m(t)R_b^b R_{lu}^{b'} R_{lu}^{lu'} r_I^{lu}(t) \quad (4.14)$$

### 4.2.3 Optimal Flight Line Configuration for Calibration Process

Based on the above discussion, the minimum/optimal flight and control configuration for the estimation of the mounting parameters should be comprised of five flight lines as illustrated in Fig. 4.14. To be more specific, the following comments can be made regarding an optimal flight configuration for calibration:

- The lever arm  $\Delta X$  can be estimated using opposite flight lines while scanning vertical planar features parallel to the flight direction.
- The lever arm  $\Delta Y$  can be estimated using opposite flight lines while scanning vertical planar features perpendicular to the flight direction.
- The lever arm  $\Delta Z$  for a given spinning multi-beam laser scanner can be estimated only using vertical control, which can be in the form of horizontal planar patches.
- The boresight pitch  $\Delta\omega$  can be estimated using opposite flight lines along with another flight line at a different height while scanning horizontal planar features and vertical planar features perpendicular to the flight direction. Another alternative for having a flight line at different flying height is to have vertical planar features whose extent in the vertical direction is significant w.r.t. the flying height or having vertical planar patches at different heights.
- The boresight roll  $\Delta\phi$  can be estimated using opposite flight lines along with another flight line at a different height while scanning horizontal planar features and vertical planar features parallel to the flight direction. Another alternative for having a flight line at different flying height is to have vertical planar features with significant height variation w.r.t. the flying height or having vertical

planar patches at different heights. Additionally, increasing the lateral distance between the tracks and between horizontal patches and the tracks would improve the boresight roll estimation.

- The boresight heading  $\Delta\kappa$  can be estimated by scanning vertical planes from two flight lines in the same direction with a significant lateral separation between them. This configuration would eliminate any discrepancies caused by lever-arm as well as boresight pitch and roll components.

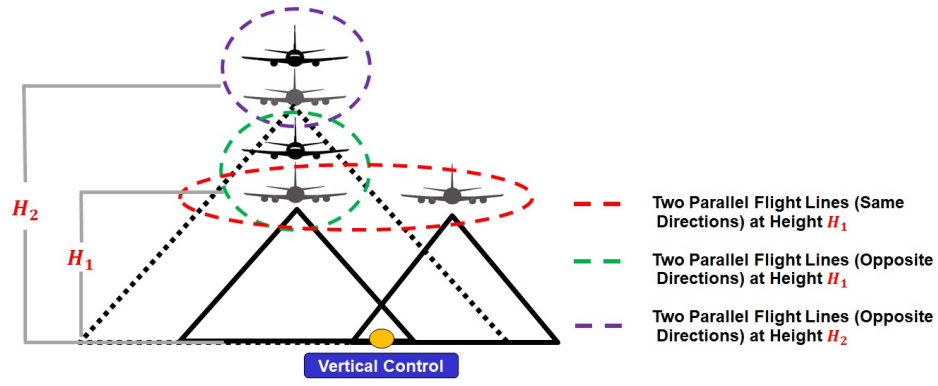


Fig. 4.14.: Optimal/Minimal flight configuration for mounting parameters estimation for a UAV-based LiDAR system.

#### 4.2.4 General Mathematical Model for LiDAR System Calibration

The mathematical model of LiDAR system calibration that could be adopted for estimating the desired system parameters while using overlapping point clouds and control surfaces is introduced. In this phase of research, there will be an assumption that the conjugate distinct points in the overlapping LiDAR and control surfaces can be identified. The mathematical formula for LiDAR point positioning is represented in a symbolic form in Equation 4.15, where  $x$  represents the unknown system parameters,  $y$  represents the system measurements,  $e$  represents the noise contaminating

the system measurements ( $e \sim (0, \Sigma)$ ), and  $\Sigma = \sigma_0^2 P_{XYZ}^{-1}$  represents the variance-covariance matrix with a-priori variance factor  $\sigma_0^2$  and weight matrix  $P_{XYZ}$  of that noise vector. The two overlapping strips are referred to as a strip-pair, and the two flight lines within a strip-pair are denoted by subscripts A and B. Equation 4.16 states that conjugate points in overlapping strips should have identical coordinates after removing the noise impact ( $e_A$  and  $e_B$ ) and using the true values for the system parameters  $x$  [31].

$$r_I^m = f(y - e, x) \quad (4.15)$$

$$r_{I_A}^m - r_{I_B}^m = f(y_A - e_A, x) - f(y_B - e_B, x) = 0 \quad (4.16)$$

In order to use the Least Square Adjustment (LSA) formula, Equation 4.15 must be linearized using Tayler's series expansion which is shown on the right side of Equation 4.17. The terms  $f_A$  and  $f_B$  represent the predicted point cloud coordinates reconstructed using the noise-contaminated measurements and the approximate values for the system parameters, for flight lines A and B, respectively; the terms  $J_{x_A}$  and  $J_{x_B}$  represent the Jacobian matrices relative to the system parameters for the A and B flight lines, respectively; and the terms  $J_{y_A}$  and  $J_{y_B}$  represent the Jacobian matrices relative to the system measurements for the A and B flight lines, respectively.

$$r_{I_A}^m - r_{I_B}^m \approx f_A - f_B + J_{x_A} \delta x - J_{y_A} e_A - J_{x_B} \delta x + J_{y_B} e_B \quad (4.17)$$

The Jacobian matrices are evaluated using the available measurements and the approximate system parameter values. The theoretical basis of the proposed LiDAR calibration algorithm here is that the derived point clouds from different flight lines are compared and use the observed discrepancies to estimate the unknown system parameters. Therefore, we take the predicted coordinates,  $f_A$  and  $f_B$ , and put them on the left side leaving the unknowns on the right side. The final representation of the calibration math model is illustrated in Equation 4.18.

$$\begin{aligned}
f_A - f_B &= (J_{x_B} - J_{x_A})\delta x + (J_{y_A}e_A - J_{y_B}e_B) \\
(J_{y_A}e_A - J_{y_B}e_B) &\sim (0, J_{y_A}\Sigma_A J_{y_A}^T, J_{y_B}\Sigma_B J_{y_B}^T)
\end{aligned} \tag{4.18}$$

The above discussion is concerned with conjugate points that can be identified in geometric tie features in overlapping strips. When it comes to conjugate points that can be identified in a control surface, and LiDAR point cloud from flight line A, similar set of equations could be derived as seen as seen in Equations 4.19 - 4.21. The noise-free control point is represented as  $r_{IC}^m$ , and the noise-contaminated control point is represented as  $r_{IC_0}^m$ . The noise contaminating the control point is represented as  $e_C$  ( $e_C \sim (0, \Sigma_C)$ ), and its variance-covariance matrix is represented as  $\Sigma_C$ . Equations 4.16 - 4.21 are simultaneously used in LSA to solve for the unknown system parameters [125]. The corrections to the approximate values of unknown parameters  $\delta x$  are estimated within the LSA to minimize the sum of squares of weighted residuals  $e^T P'_{XYZ} e$ , where  $P'_{XYZ}$  is the modified weight matrix of the noise vector.

$$r_{IA}^m - r_{IC}^m = f(y_A - e_A, x) - (r_{IC_0}^m - e_C) = 0 \tag{4.19}$$

$$r_{IA}^m - r_{IC}^m \approx f_A - r_{IC_0}^m + J_{x_A}\delta x - J_{y_A}e_A + e_C \tag{4.20}$$

$$f_A - r_{IC_0}^m = -J_{x_A}\delta x + (J_{y_A}e_A - e_C) \tag{4.21}$$

$$(J_{y_A}e_A - e_C) \sim (0, J_{y_A}\Sigma_A J_{y_A}^T + \Sigma_C)$$

### 4.3 Feature Extraction

In this part of research, a calibration strategy is proposed to estimate the mounting parameters of the LiDAR unit with respect to the onboard GNSS/INS unit using geometric tie features (e.g., planar, and linear/cylindrical features). After collecting data from several flight lines, a 3D point cloud relative to a global reference frame will be derived using the system measurements (i.e., the GNSS/INS unit position



and orientation, laser range, and the mirror scan angles) and initial estimates for the mounting parameters. Then, conjugate features are identified and extracted from the reconstructed point cloud. Finally, an iterative LiDAR system calibration with weight modification is proposed to derive the mounting parameters based on the minimization of normal distances between conjugate features [126].

Owing to the non-selective nature of the LiDAR scanning process (i.e., inability to force the laser beam to scan a specific point), points cannot be directly used as the calibration primitives. In other words, it is not possible to reliably identify common points in overlapping point clouds and control surfaces. Therefore, higher level features (e.g., planar and linear features), such as building façades, ground patches, light poles, and lane markers, are used and these can be directly extracted from overlapping areas within the flight lines. However, conjugate feature extraction from several flight lines could be time-consuming and inefficient, especially when the initial estimates for mounting parameters used to reconstruct the 3D point cloud are considerably inaccurate. To facilitate automated identification of conjugate features in such cases, specifically designed calibration boards covered by highly reflective surfaces, and could be easily deployed and set up in outdoor environments, are used in this study. More specifically, various traffic signs (75 cm wide Stop signs, 90 cm x 60 cm Wrong Way signs, and 60 cm x 60 cm checkerboard targets) are used as highly reflective boards. As mentioned previously, in order to enable an optimal feature configuration, there is a need to have an overlapping area with varying topography which means having linear/planar features with slopes as well as different aspects as depicted in Fig. 4.15.

The highly reflective boards can be easily identified from intensity data, as shown in Fig. 4.16, where the points belonging to these boards exhibit higher intensity values compared to other LiDAR points. Firstly, a pre-defined threshold is set to extract the high-intensity points. To avoid the extraction of high-intensity points belonging to objects other than these boards, an approximate pre-set region is manually set as seed points for each board. Then, a distance-based region growing technique is

adopted to group the high intensity boards. Finally, a plane-fitting is done for these points, and the points lying within a normal distance threshold from the best-fitting plane are extracted. Other planar features, such as ground patches or wall patches, can be extracted by defining two diagonally opposite corners. A bounding box is constructed around the planar feature of interest by adding a buffer value (in X, Y, and Z directions) to the coordinates of diagonally opposite corners. Again, a plane-fitting is done for the points contained inside the box, and the ones lying within a normal distance threshold from the best-fitting plane are extracted. One should note that there is a challenge when using such high-level features; the link between the features and sensor model as represented by the LiDAR equation is lost (i.e., the LiDAR equation is a point positioning equation). In other words, a direct link between the calibration primitives and the direct system parameters can only be achieved when using discrete points. One possible approach is to use non-conjugate points that belong to corresponding features (e.g., planar features) within the calibration procedure. These points will be denoted here forth as pseudo conjugate points.

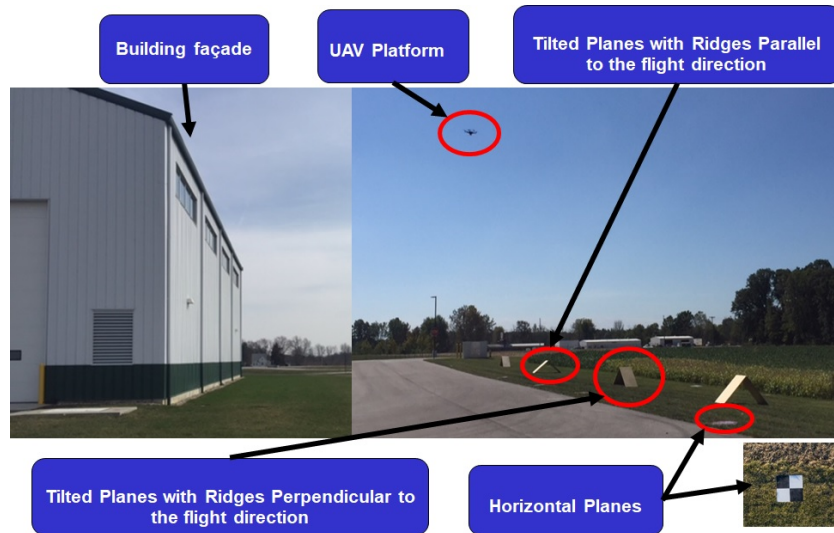


Fig. 4.15.: Optimal feature configuration for mounting parameters estimation for a UAV-based LiDAR system

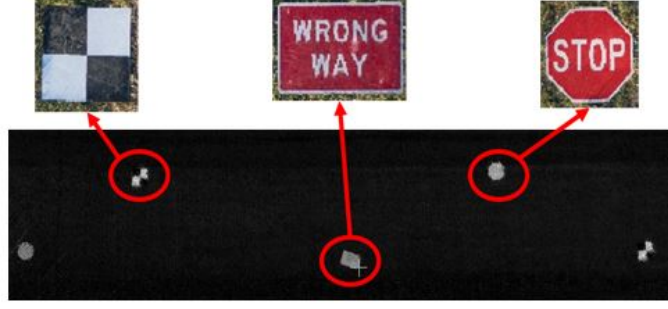


Fig. 4.16.: Intensity data from a point cloud that includes highly reflective boards highlighted by the zoomed-in areas

#### 4.4 Pseudo-Conjugate Points in Overlapping Strips and Weight Modification Procedure

In the proposed calibration method, conjugate features are extracted from the point clouds of LiDAR sensors and several flight lines. The mounting parameters of each sensor are derived by minimizing the discrepancies among conjugate features (points/lines/planes) in overlapping flight lines. Each pairing between conjugate features will result in a misclosure vector, which would be random ( $\vec{e}$ ) in case of a conjugate point pair, as given by Equation 4.22. However, a pairing between non-conjugate points along corresponding planar or linear/cylindrical features would additionally introduce a non-random component ( $\vec{D}$ ) in the misclosure vector, as given by Equation 4.23. Such points will be denoted here forth as pseudo conjugate points. The non-random component  $\vec{D}$  would lie along the planar surface or along the linear feature/axis of cylinder, respectively, as illustrated in Fig. 4.17.

$$r_I^m(A) - r_I^m(B) = \vec{e} \quad (4.22)$$

$$r_I^m(A) - r_I^m(B) = \vec{D} + \vec{e} \quad (4.23)$$

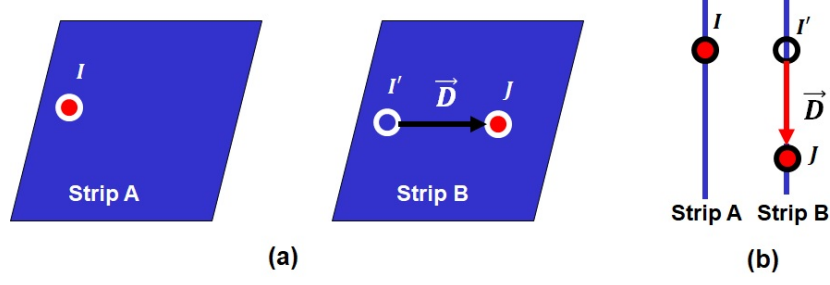


Fig. 4.17.: Discrepancy vector between non-conjugate points along corresponding  
(a) planar, and (b) linear features

Therefore, a modified weight matrix,  $P'$ , is introduced to eliminate the non-random component of the misclosure vector,  $\vec{D}$ , from the LSA cost function, as given by Equation 4.24 [126]. To derive this matrix, a local coordinate system (UVW) is established first. For linear/cylindrical features, the U axis is aligned along the line/axis of cylinder and V and W axes are arbitrarily chosen to satisfy the orthogonality of the UVW triad. For planar features, W axis is aligned along the normal vector of the plane in question, and U and V axes are arbitrarily chosen along the planar feature. An illustration of the local coordinate systems for the two types of features is shown in Fig. 4.18. Then, a rotation matrix,  $R_{XYZ}^{UVW}$ , relating the local and mapping coordinate systems is derived according to the components of the vectors, U, V, and W relative to the mapping frame. The weight matrix,  $P_{XYZ}$ , in the mapping coordinate system is transformed to a weight matrix,  $P_{UVW}$ , in the local coordinate system according to the law of error propagation (Equation 4.25). The weight matrix,  $P_{UVW}$ , is modified by assigning a zero weight to the elements corresponding to the direction of  $\vec{D}$ . More specifically, the non-random component of the misclosure vector ( $\vec{D}$ ) can be eliminated from the LSA minimization target function by setting a zero weight in the corresponding direction. The direction of ( $\vec{D}$ ) for a linear/cylindrical feature is along the U axis. Therefore, the modified weight matrix,  $P'_{UVW}$ , has zero weight in all the elements pertaining to the U axis (Equation 4.26). Similarly, the direction of ( $\vec{D}$ ) for a planar feature is along the U and V axes. So, all the elements

pertaining to the U and V axes are assigned a zero weight (Equation 4.27). The modified weight matrix,  $P'_{XYZ}$ , in the mapping coordinate system is derived using Equation 4.28. Finally, the obtained modified weight matrix,  $P'_{XYZ}$ , is applied to the condition in Equation 4.23 to account for the use of pseudo conjugate points along corresponding features within overlapping flight lines.

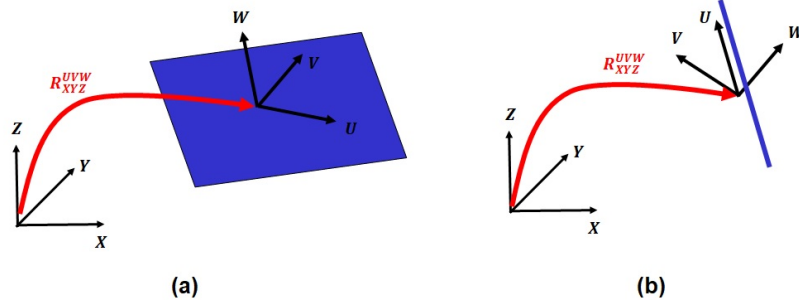


Fig. 4.18.: Illustration of local coordinate systems for (a) planar and (b) linear features

$$P' \vec{D} = P' \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = 0 \quad (4.24)$$

$$P_{UVW} = R_{XYZ}^{UVW} P_{XYZ} R_{XYZ}^{UVW^T} = \begin{bmatrix} P_U & P_{UV} & P_{UW} \\ P_{VU} & P_V & P_{VW} \\ P_{WU} & P_{WV} & P_W \end{bmatrix} \quad (4.25)$$

$$P'_{UVW} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_V & P_{VW} \\ 0 & P_{WV} & P_W \end{bmatrix} \quad (4.26)$$

$$P'_{UVW} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & P_W \end{bmatrix} \quad (4.27)$$

$$P'_{XYZ} = R_{XYZ}^{UVW^T} P'_{UVW} R_{XYZ}^{UVW} \quad (4.28)$$

Finally, the mounting parameters for the laser unit can be derived by minimizing the discrepancies among the conjugate linear and/or planar features obtained from different flight lines. However, when the initial estimate of the mounting parameters is inaccurate, the estimated modified weight matrix would be imprecise which would affect the accuracy of the derived mounting parameters. Hence, this research proposes an iterative calibration procedure. Firstly, the discrepancy among extracted features is minimized to derive mounting parameters through the weight modification process. Then, the points along the extracted features are re-generated using the newly estimated mounting parameters and the discrepancy among conjugate features is minimized again using a newly defined modified weight matrix. The above steps are repeated until the change in the estimates of the mounting parameters is below a predefined threshold.

#### 4.5 Experimental Results

A comprehensive analysis of the impact of biases in the different mounting parameters of a UAV-based LiDAR system has been conducted to devise an optimal flight and target configuration and proposed a calibration strategy. Several experiments are conducted to validate the feasibility of the proposed strategy and quality of calibration, followed by an evaluation of the devised optimal configuration based on the standard deviation and correlation matrix for the estimated mounting parameters for two different datasets. One dataset is acquired with flight lines at a single flying height and the other with flight lines at two different flying heights. For the UAV system consisting of a Velodyne VLP-16 Puck Hi-Res LiDAR unit and an APX-15 GNSS/INS unit, a LiDAR Error Propagation Calculator developed by Habib et al. [127] is used to compute the expected accuracy of point positioning for this system. Such calculator enables the user to determine the accuracy of the ground coordinates

for a certain point derived from a LiDAR system, given the values of the LiDAR input parameters and their accuracies. For the proposed UAV mapping system, the calculator suggests an accuracy of 5-6 cm for a flying height of 15-25 m. In this research, the UAV-based LiDAR system is flown with six tracks each at a flying height of 15 m and 25 m with a speed of 1.5 m/s over sixteen specially designed highly reflective boards (75 cm wide Stop signs, 90 cm x 60 cm Wrong Way signs, and 60 cm x 60 cm checkerboard targets) and five hut-shaped targets (with two 60 cm x 120 cm planar boards) with their ridges oriented parallel and perpendicular to the flying direction. The average lateral separation between the tracks is 6 m. The two surfaces corresponding to each of these huts are used as planar features for calibration, and their ridges are used as conjugate linear features. Additional planar features, such as ground patches, rooftops, and building facades, are also used for calibration. The configuration of the tracks and the target primitives (in pink) are shown in Fig. 4.19. For the UAV system used in this study, the X, Y, Z-axes of the IMU body frame are not aligned along the starboard, forward, and up directions. Instead, the coordinate frames are aligned as shown in Fig. 4.20. So, a virtual IMU body frame defined as per Equation 4.12 is used. The virtual LiDAR unit frame is derived using the nominal value for  $R_{lu}^{lu'} = R_\omega(-90^\circ)R_\phi(0^\circ)R_\kappa(0^\circ)$  according to the system setup as illustrated before in Equation 4.5. Therefore, the mounting parameters relating the virtual IMU and virtual laser unit coordinate systems are estimated.

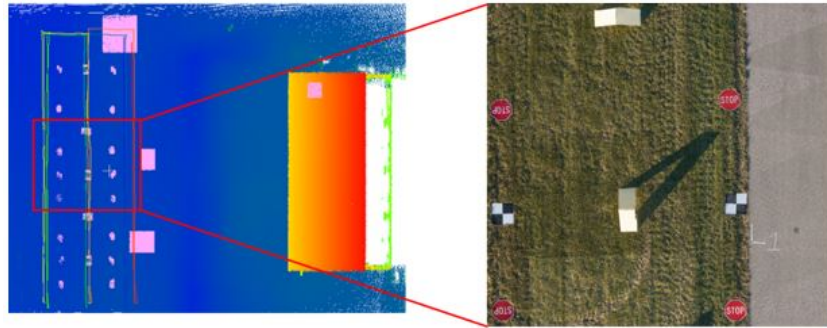


Fig. 4.19.: Configuration of flight lines and target primitives used for calibration as visible in 3D point clouds and RGB orthophoto

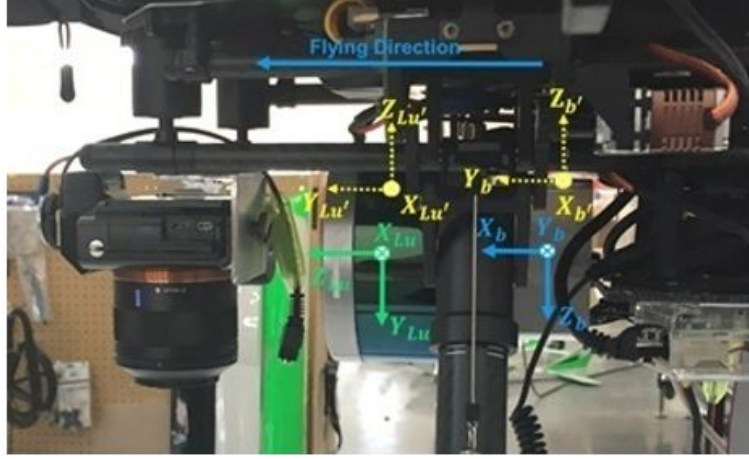


Fig. 4.20.: Actual alignment of coordinate systems for the UAV LiDAR system used in this study

#### 4.5.1 Single Flying Height

First, a sub-optimal configuration is used to evaluate the theoretical bias impact analysis for the estimation of system parameters. In this case, six flight lines at a single flying height of 15 m are used along with the target primitives lying exactly below the flight lines (the sixteen boards and five hut-shaped targets) and with no significant  $X'$  and  $Z'$  -coordinate variation. The initial approximations of these mounting parameters and the final results (along with their standard deviations) from the proposed calibration procedure are listed in Table 4.2. One should note that the lever arm  $\Delta Z$  for the laser unit is fixed during the calibration procedure. The correlation matrix for the estimated mounting parameters of the laser unit is listed in Table 4.3, which indicates that  $\Delta Y$  is highly correlated with  $\Delta \omega$  (0.9905). The average accuracy after calibration can be quantified by the square root of the a-posteriori variance factor ( $\hat{\sigma}_0$ ), which is 3.12 cm in this case. This is better than the expected accuracy of around 5-6 cm according to the accuracies of the hardware involved and an error propagation calculation.



Table 4.2.: Mounting parameters of VLP-16 before and after calibration test 1

VLP-16 LiDAR Unit Mounting Parameters ( $r_{lu'}^{b'}$ ) ( $R_{lu'}^{b'}$ )						
	$\Delta X(\text{m})$	$\Delta Y(\text{m})$	$\Delta Z(\text{m})$	$\Delta\omega (^{\circ})$	$\Delta\phi (^{\circ})$	$\Delta\kappa (^{\circ})$
Initial	0	0.02	0	0	0	0
Final	0.0084	-0.0037	0	0.1676	-0.7666	-0.2825
Standard Deviation	0.0168	0.0381	Fixed	0.0797	0.0323	0.0542

Table 4.3.: Correlation matrix of mounting parameter estimates from calibration test 1

	$\Delta X$	$\Delta Y$	$\Delta Z$	$\Delta\omega$	$\Delta\phi$	$\Delta\kappa$
$\Delta X$	1.0000	0.1258	0.0000	0.1303	0.6647	0.0231
$\Delta Y$	0.1258	1.0000	0.0000	0.9905	0.0922	0.0296
$\Delta Z$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\Delta\omega$	0.1303	0.9905	0.0000	1.0000	0.0916	0.0263
$\Delta\phi$	0.6647	0.0922	0.0000	0.0916	1.0000	0.0351
$\Delta\kappa$	0.0231	0.0296	0.0000	0.0263	0.0351	1.0000

The high correlation between the system parameters ( $\Delta Y$  and  $\Delta\omega$ ) renders the calibration results unreliable. So, we incorporate planar features located at a significant lateral separation from the flight lines (ground patches, rooftops, and building facade) while still considering flight lines at a single flying height of 15 m. The corresponding mounting parameters and correlation matrix are listed in Table 4.4 and Table 4.5, respectively. The standard deviation of all the estimated parameters can be seen to have reduced as compared to the previous case. Moreover, the correla-

tion matrix indicates a reduction in the correlation between  $\Delta Y$  and  $\Delta\omega$  from 0.9905 to 0.9032 and the correlation between  $\Delta X$  and  $\Delta\phi$  is also reduced from 0.6647 to 0.1025. These reductions can be attributed to the variation in  $z'$  and  $x'$ -coordinates, as derived in the theoretical bias impact analysis. The correlation between all the other parameters has also decreased, thus proving the improvement in calibration results. The average accuracy after calibration as quantified by the square root of the a-posteriori variance factor ( $\hat{\sigma}_0$ ) is 2.22 cm in this case, which is also less than the previous case of 3.12 cm.

Table 4.4.: Mounting parameters of VLP-16 before and after calibration test 2

VLP-16 LiDAR Unit Mounting Parameters ( $r_{lu'}^{b'}$ ) ( $R_{lu'}^{b'}$ )						
	$\Delta X(\mathbf{m})$	$\Delta Y(\mathbf{m})$	$\Delta Z(\mathbf{m})$	$\Delta\omega$ ( $^\circ$ )	$\Delta\phi$ ( $^\circ$ )	$\Delta\kappa$ ( $^\circ$ )
Initial	0	0.02	0	0	0	0
Final	0.0180	-0.0067	0	0.1598	-0.6942	-0.2538
Standard Deviation	0.0118	0.0181	Fixed	0.0361	0.0139	0.0438

Table 4.5.: Correlation matrix of mounting parameter estimates from calibration test 2

	$\Delta X$	$\Delta Y$	$\Delta Z$	$\Delta\omega$	$\Delta\phi$	$\Delta\kappa$
$\Delta X$	1.0000	0.0240	0.0000	0.0141	0.1025	0.0298
$\Delta Y$	0.0240	1.0000	0.0000	0.9032	0.1390	0.0171
$\Delta Z$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\Delta\omega$	0.0141	0.9032	0.0000	1.0000	0.1775	0.0221
$\Delta\phi$	0.1025	0.1390	0.0000	0.1775	1.0000	0.2456
$\Delta\kappa$	0.0298	0.0171	0.0000	0.0221	0.2456	1.0000

### 4.5.2 Multiple Flying Heights

Finally, all the flight lines of this experiment (six at a height of 15 m and six at a height of 25 m) are used with all the target primitives described previously. As suggested by the theoretical analysis, this is the most optimal configuration for calibration. The qualitative analysis of some of the target primitives (a checkerboard target, a hut-shaped target, a building facade, and a ground patch) is shown in Fig. 4.21, which indicates a major improvement in the alignment for each of the calibration targets compared to such targets before calibration. The mounting parameters obtained in this case and the corresponding correlation matrix are given in Table 4.6 and Table 4.7, respectively. The average accuracy after calibration as quantified by the square root of the a-posteriori variance factor ( $\hat{\sigma}_0$ ) is 2.24 cm in this case, which is 0.2 mm worse than the second case (where  $\hat{\sigma}_0$  was 2.22 cm). This perceived deterioration is a result of the higher error propagation in the case of points captured using flight lines at an altitude of 25 m (as compared to 15 m in the second case) as illustrated in Table 4.8.

Table 4.6.: Mounting parameters of VLP-16 before and after calibration test 3

VLP-16 LiDAR Unit Mounting Parameters ( $r_{lu'}^{b'}$ ) ( $R_{lu'}^{b'}$ )						
	$\Delta X(\text{m})$	$\Delta Y(\text{m})$	$\Delta Z(\text{m})$	$\Delta\omega$ (°)	$\Delta\phi$ (°)	$\Delta\kappa$ (°)
Initial	0	0.02	0	0	0	0
Final	0.0189	0.0086	0	0.0427	-0.7051	-0.3381
Standard Deviation	0.0103	0.0145	Fixed	0.0263	0.0118	0.0377

Table 4.7.: Correlation matrix of mounting parameter estimates from calibration  
test 3

	$\Delta X$	$\Delta Y$	$\Delta Z$	$\Delta\omega$	$\Delta\phi$	$\Delta\kappa$
$\Delta X$	1.0000	0.0215	0.0000	0.0187	0.0908	0.0778
$\Delta Y$	0.0215	1.0000	0.0000	0.8438	0.0794	0.0022
$\Delta Z$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\Delta\omega$	0.0187	0.8438	0.0000	1.0000	0.1414	0.0133
$\Delta\phi$	0.0908	0.0794	0.0000	0.1414	1.0000	0.2458
$\Delta\kappa$	0.0778	0.0022	0.0000	0.0133	0.2458	1.0000

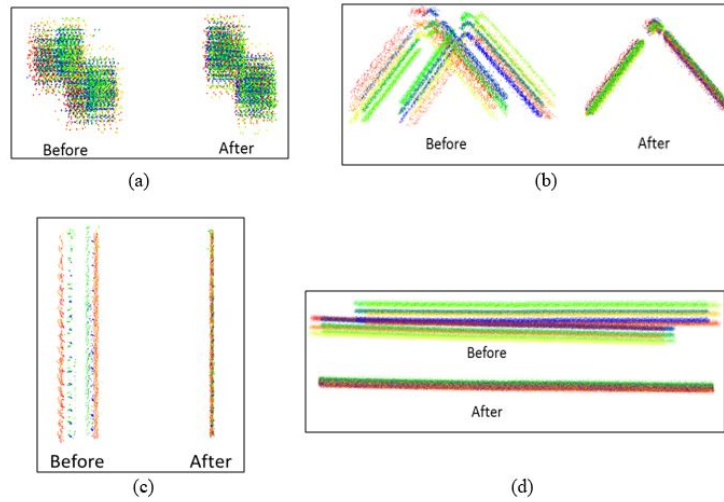


Fig. 4.21.: Qualitative evaluation of calibration targets before and after calibration: (a) Checkerboard, (b) hut-shaped target, (c) building facade, and (d) ground patch

The number of LiDAR points along each feature and the RMSE of normal distance of points from best-fitting plane/line for all the extracted features before calibration and after each of the three cases of calibration are listed in Tables 4.9 - 4.12, which indicates a significant improvement of point clouds after calibration and also that

the best results are achieved in the third case, which consists of an optimal target primitive and flight line configuration where there are multiple flight lines at different flying heights and planar/linear features oriented in different directions with sufficient variation in lateral distance from the flight lines.

Table 4.8.: Calibration of the UAV system: Square root of the a-posteriori variance factor

	<b>Before Calibration</b>	<b>After Calibration Case 1</b>	<b>After Calibration Case 2</b>	<b>After Calibration Case 3</b>
$H(\text{m})$		15 m	15 m	15 m & 25 m
$\hat{\sigma}_0 \text{ (m)}$		0.0312	0.0222	0.0224

Table 4.9.: RMSE of plane/line fitting for different configurations (Ground Patches, Rooftop, Building Facade)

<b>Feature ID</b>	<b>Before Calib. RMSE (m)</b>	<b>Case 1 No. of points</b>	<b>Case 1 RMSE (m)</b>	<b>Case 2 No. of points</b>	<b>Case 2 RMSE (m)</b>	<b>Case 3 No. of points</b>	<b>Case 3 RMSE (m)</b>
<b>Ground Patches, Rooftop, Building Facade</b>							
<b>Ground 0</b>	0.1178	0	NA	123011	0.0192	218355	0.0207
<b>Ground 1</b>	0.0940	0	NA	135840	0.0233	230256	0.0233
<b>Ground 2</b>	0.1220	0	NA	63474	0.0190	113103	0.0198
<b>Rooftop</b>	0.1466	0	NA	1173	0.0117	3055	0.0131
<b>Building Facade</b>	0.3461	0	NA	9035	0.0274	22470	0.0257

Table 4.10.: RMSE of plane/line fitting for different configurations (Reflective Boards)

<b>Feature ID</b>	<b>Before Calib. RMSE (m)</b>	<b>Case 1 No. of points</b>	<b>Case 1 RMSE (m)</b>	<b>Case 2 No. of points</b>	<b>Case 2 RMSE (m)</b>	<b>Case 3 No. of points</b>	<b>Case 3 RMSE (m)</b>
<b>Reflective Boards</b>							
<b>Board 0</b>	0.0813	3006	0.0202	3006	0.0232	4568	0.0252
<b>Board 1</b>	0.0676	2987	0.0215	2987	0.0198	4596	0.0217
<b>Board 2</b>	0.0928	4683	0.0188	4683	0.0221	7397	0.0244
<b>Board 3</b>	0.0762	4735	0.0198	4735	0.0189	7462	0.0211
<b>Board 4</b>	0.0923	4851	0.0169	4851	0.0205	7578	0.0229
<b>Board 5</b>	0.0791	4456	0.0196	4456	0.0193	7062	0.0205
<b>Board 6</b>	0.0794	2736	0.0167	2736	0.0199	4320	0.0208
<b>Board 7</b>	0.0712	2736	0.0196	2736	0.0189	4361	0.0204
<b>Board 8</b>	0.0842	4566	0.0190	4566	0.0221	7140	0.0231
<b>Board 9</b>	0.0759	4492	0.0187	4492	0.0183	7045	0.0194
<b>Board 10</b>	0.0872	4934	0.0186	4934	0.0217	7829	0.0230
<b>Board 11</b>	0.0740	4976	0.0206	4976	0.0198	8031	0.0210
<b>Board 12</b>	0.0791	2996	0.0170	2996	0.0193	4628	0.0213
<b>Board 13</b>	0.0694	3198	0.0199	3198	0.0193	4999	0.0208
<b>Board 14</b>	0.0823	4420	0.0170	4420	0.0196	6562	0.0228
<b>Board 15</b>	0.0763	4907	0.0173	4907	0.0175	7304	0.0197

Table 4.11.: RMSE of plane/line fitting for different configurations (Hut Surfaces)

Feature ID	Before Calib. RMSE (m)	Case 1 No. of points	Case 1 RMSE (m)	Case 2 No. of points	Case 2 RMSE (m)	Case 3 No. of points	Case 3 RMSE (m)
<b>Hut Surfaces</b>							
<b>Surface 0</b>	0.1097	3359	0.0734	3359	0.0736	5292	0.0895
<b>Surface 1</b>	0.1019	3613	0.0708	3613	0.0720	5610	0.0880
<b>Surface 2</b>	0.1488	3879	0.0146	3879	0.0173	6181	0.0259
<b>Surface 3</b>	0.1305	4576	0.0184	4576	0.0176	7586	0.0210
<b>Surface 4</b>	0.1082	3139	0.0710	3139	0.0713	5179	0.0880
<b>Surface 5</b>	0.1071	3918	0.0738	3918	0.0749	6461	0.0924
<b>Surface 6</b>	0.1469	4149	0.0164	4149	0.0186	6543	0.0237
<b>Surface 7</b>	0.1365	4654	0.0190	4654	0.0181	7689	0.0208
<b>Surface 8</b>	0.1018	4158	0.0708	4158	0.0722	6567	0.0868
<b>Surface 9</b>	0.1123	3356	0.0733	3356	0.0736	5457	0.0906

Table 4.12.: RMSE of plane/line fitting for different configurations (Hut Ridges)

Feature ID	Before Calib. RMSE (m)	Case 1 No. of points	Case 1 RMSE (m)	Case 2 No. of points	Case 2 RMSE (m)	Case 3 No. of points	Case 3 RMSE (m)
<b>Hut Ridges</b>							
<b>Ridge 0</b>	0.0568	817	0.0255	817	0.0270	1006	0.0434
<b>Ridge 1</b>	0.1120	1044	0.0167	1044	0.0176	1723	0.0212
<b>Ridge 2</b>	0.0523	722	0.0298	722	0.0313	937	0.0417
<b>Ridge 3</b>	0.1129	1034	0.0179	1034	0.0186	1694	0.0212
<b>Ridge 4</b>	0.0614	813	0.0262	813	0.0270	1071	0.0461

## 4.6 Summary

In this phase of research, a LiDAR system calibration strategy for a UAV-based MMS is proposed to directly estimate the mounting parameters for spinning multi-beam laser scanners through an outdoor calibration procedure. This approach is based on the use of conjugate planar/linear features in overlapping point clouds derived from different flight lines. Designing an optimal configuration for calibration is the first and foremost step in order to ensure the most accurate estimates of mounting parameters. This is achieved by conducting a rigorous theoretical analysis of the potential impact of bias in mounting parameters of a LiDAR unit on the resultant point cloud. The dependency of the impact on the orientation of target primitives and relative flight line configuration would help in deducing the configuration that would maximize as well as decouple the impact of bias in each mounting parameter so as to ensure their accurate estimation. Finally, the proposed analysis and calibration strategy are validated by calibrating a UAV-based LiDAR system using two different datasets – one acquired with flight lines at a single flying height and the other with flight lines at two different flying heights. The calibration performance is evaluated by analyzing correlation between the estimated system parameters, the a-posteriori variance factor of the Least Squares Adjustment (LSA) procedure, and the quality of fit of the adjusted point cloud to planar/linear features before and after the calibration process.



## 5. SLAM-ASSISTED COVERAGE PATH PLANNING AND IMPLEMENTATION OF PSEUDO-GNSS/INS LOCALIZATION SYSTEM FOR INDOOR LIDAR MMS\*

### 5.1 Introduction

In this chapter, the SLAM-assisted coverage path planning (CPP) approach for unmanned vehicles and SLAM-based mapping for a GNSS-denied environment are explained in Section 5.2 and Section 5.3, respectively. For the SLAM-assisted CPP approach, the offline CPP algorithms which include convex decomposition methods as well as the suitable way to obtain the optimal boustrophedon pattern are introduced. In addition to that, the utilization of the real-time SLAM technique in a way that can assist the CPP algorithms to ensure maximum area coverage is discussed. Regarding the SLAM-based mapping for a GNSS-denied environment, the proposed pseudo-GNSS/INS framework is introduced. Also, the post-processing trajectory enhancement techniques which include the Iterative Closest Projected Point (ICPP) algorithm and the implementation of the smoothed trajectory are explained in detail. Finally, the experimental results are quantitatively and qualitatively illustrated.

### 5.2 SLAM-assisted Coverage Path Planning

#### 5.2.1 Overview

For some mapping applications, occasionally, there is an essential need to only survey a specific area of interest rather than surveying the whole field while ensuring

---

\*THIS CHAPTER IS LARGELY BASED ON A CONFERENCE PAPER(SHAMSELDIN, TAMER, ET AL. "SLAM-BASED PSEUDO-GNSS/INS LOCALIZATION SYSTEM FOR INDOOR LIDAR MOBILE MAPPING SYSTEMS." POSITION, LOCATION AND NAVIGATION SYMPOSIUM (PLANS), 2018 IEEE/ION. IEEE, 2018.

maximum area coverage. This phase of research proposes the CPP algorithms which is considered the task of determining the path for unmanned platforms to pass over all the points of an area of interest without any repetition. In this regard, a convex cellular decomposition algorithm is performed for the polygonal area of interest. By considering odometry errors, a real-time SLAM technique is used to enhance the pre-planned path rather than relying on the unrealistic assumption of an idealized path execution. More specifically, due to the heading angle deviation error, such a stochastic technique is utilized to reshape the nominal path for achieving the maximum area coverage. One should note that such a SLAM algorithm runs onboard the unmanned platform in real-time while performing the preplanned mission. The path adaptation is based on the observation measurements provided by a 2D-LiDAR unit.

### 5.2.2 Hybrid Approach Implementation

In this dissertation, a new hybrid approach that uses an area coverage function allowing dynamic changes to the coverage path through the map updates by performing SLAM technique is developed. This hybrid approach is considered the key contribution of this part of the research. This section provides a functional description of different blocks of the implemented LiDAR mapping system which is illustrated in Fig. 5.1. The block diagram demonstrates the system bifurcation into an offline CPP that generates the motion path for navigation, and a real-time SLAM system that uses control input from this path to update for dynamic changes during path traversal. The block diagram is briefly described ahead while the theory behind each of the blocks is elaborated on in latter sections:

#### Block Diagram Description

For the offline planning, the input to the mapping system is an initial coverage map estimate which can be updated for iterative operations as follows:

- **Polygonal Approximation:** The proposed CPP planner requires a polygonal approximation of the area boundary for complete coverage. Hence, the initial map is fed as input in the form of counter-clockwise vertex coordinates of the polygonal approximation of the area to be covered. The starting point for the coverage path is also provided. If the provided map is not in polygonal form, it is converted to such form by using Douglas-Peucker Polyline Simplification [128].

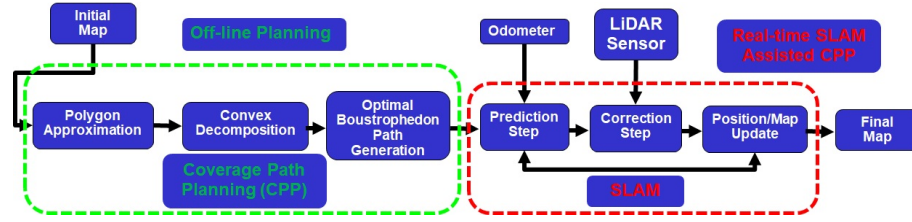


Fig. 5.1.: Functional block diagram for coverage-based SLAM implementation

- **Convex Decomposition:** From the polygonal area provided, the first step is to partition the entire area into a disjoint set of convex cells. This is done through convex decomposition by polygon triangulation using the Ear-Splitting Algorithm [129]. The criterion for convex decomposition involves factors that maintain coverage path optimality by performing the Minimum Sum of Altitude (MSA) technique within each cell [90].
- **Optimal Boustrophedon Pattern Generation:** A boustrophedon coverage pattern is then generated for each of these convex cells while the optimality of coverage in terms of path length is achieved by using the MSA criterion for selecting the optimal sweep direction.

Regarding the real-time SLAM-assisted CPP, the motion path generated by the offline planner is fed to the online system, which uses these control inputs for SLAM operation as explained ahead. For this implementation, a G-Mapping SLAM strategy

is used for real-time operation using 2D-LiDAR as a mapping sensor to perform localization and mapping [130]. The real-time SLAM-assisted CPP steps are described as follows:

- **Prediction Step (SLAM):** The prediction step is identical to a generic SLAM prediction step. Since the coverage path generated by the optimal boustrophedon pattern generation block feeds control inputs to the robot for path traversal. As these inputs are incrementally fed to the robot, the robot pose estimate is updated accordingly.
- **Correction Step (SLAM):** The correction step for SLAM uses the 2D-LiDAR point cloud to update the occupancy grid for the environment map.
- **Position/Map Update:** The current robot position and the occupancy grid map are updated continuously while the robot traces the entire coverage path. This operation is carried out upon two main events. First, when the robot trajectory is observed to deviate from the preplanned coverage path. Second, if the occupancy grid observes a new dynamic change in the map dimensions requiring a new coverage path to be generated from the offline planner for the new area.

### 5.2.3 Offline Planning - CPP Strategy

This section describes in detail the blocks implemented in the offline planner as well as how they are modified for efficient operation. More specifically, the convex decomposition algorithm which can deal with either a convex or concave polygon will be introduced. Furthermore, the best way to achieve an optimal boustrophedon pattern with minimum path length as well as a minimum number of turns will be discussed.

## Convex Decomposition Algorithm

One should note that the system in Fig. 5.1 implements a convex decomposition for CPP using the polygon triangulation method. The conceptual basis of such a method states that any polygon with  $n$  vertices can always be triangulated and always have a  $n - 2$  triangle. Since the triangle is always convex (the interior angle at vertices is smaller than  $\pi$  radians), there is a guarantee that the convex decomposition will be achieved [131].

The idea of polygon triangulation is based on the Ear-Clipping Algorithm which performs the triangulation of the desired polygon [129]. For a generic Ear-clipping algorithm, the adjacent vertices of the current chosen point are connected successively so that the polygon is partitioned into a disjoint set of triangles. For convex decomposition, the algorithm can be modified to also recursively merge the triangles formed by ear-clipping into the prior cell if such a merged cell is found to be convex. In this manner, a convenient convex decomposition of a monotonic polygon can be carried out through the Ear-Clipping algorithm. Fig. 5.2 shows a simulated data which illustrates how such an algorithm is performed over a desired polygon area.

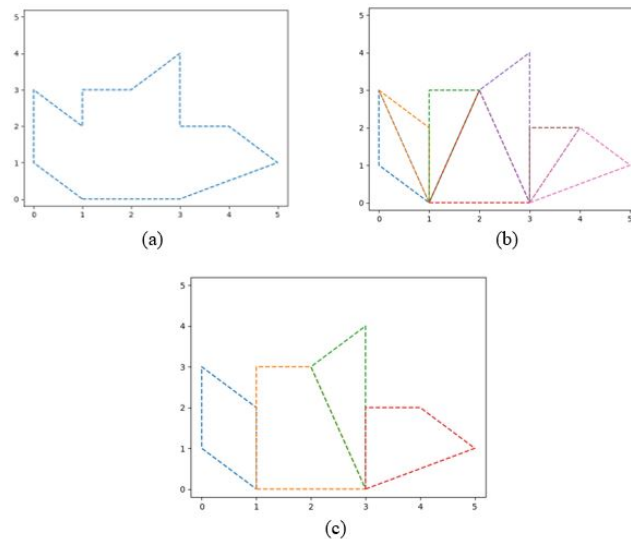


Fig. 5.2.: Example of a simulated data: (a) Desired area , (b) Polygon triangulation technique, and (c) Optimum Polygon triangulation

## Optimal Boustrophedon Pattern Generation

The whole idea of the BCD algorithm is to minimize the number of turns for any convex polygon which can be performed by figuring the optimal sweep direction and to address this case, the MSA technique is utilized. The generic boustrophedon pattern for a convex polygon, shown in Fig. 5.3, best describes its nature. Such a pattern consists of parallel zig-zag patterns equally spaced from each other at a distance equal to the robot coverage radius with each line extending till the polygon boundary. The total length of the boustrophedon path  $L_{tot}$  is a function of the sweep lengths,  $L_{i,k}$  and the turn lengths  $L_{turn,k}$  as expressed by Equation 5.1. Since the coverage radius  $r$  is much smaller than the polygon width, minimizing  $L_{tot}$  is possible through minimizing the number of turns,  $n$  as shown in Equation 5.2, where  $n$  denotes total number of turns and  $i$  represents the sweep direction.

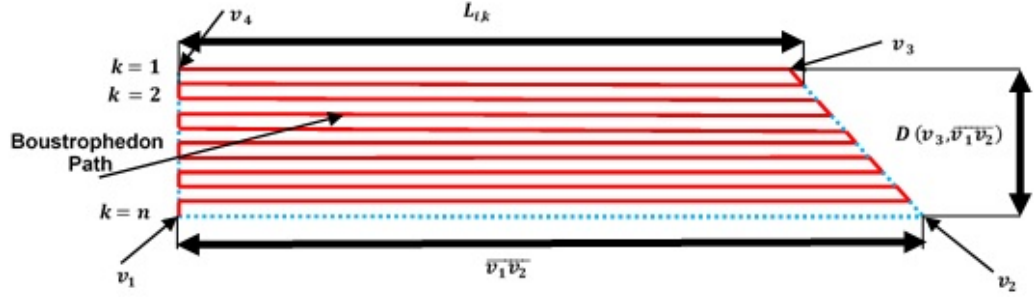


Fig. 5.3.: Boustrophedon Pattern for Coverage

$$L_{tot} = \sum_{k=1}^n L_{i,k} + n \cdot L_{turn,k} \quad (5.1)$$

$$\min |L_{tot}| = \min \left| \sum_{k=1}^n L_{i,k} + n \cdot L_{turn,k} \right| \quad (5.2)$$

One should note that the energy of turning is more than that of straight line motion. Thus minimizing the number of turns provides a metric of achieving distance optimality [132]. Therefore, the optimality for a coverage path is more directly

achieved through the MSA criterion proposed by Huang [90] via finding the optimal sweep direction  $\theta_{MSA}$  that solves the cost function in Equation 5.3.

$$\theta_{MSA} = \min_i \{ \max_m [D(v_m, \overline{v_j v_{j+1}})] \} \quad (5.3)$$

$$j = 1, 2, \dots, n \quad m \neq j, j + 1$$

Here,  $D$  represents the span defined by the edge  $(\overline{v_j v_{j+1}})$ . In other words,  $D$  is the length of the longest altitude from the edge  $(\overline{v_j v_{j+1}})$  to a vertex  $v_m$  while  $\theta_{MSA}$  is the direction of the optimal edge. The above cost function implies that for a convex polygon, the optimal direction giving the minimum number of turns is the direction of the edge that has the minimum sum of altitudes and therefore the minimum span [84]. Using this function, it is thus possible to find the correct direction optimizing the number of turns and the total path length as long as the polygon is convex.

Regarding the Ear-Clipping algorithm, it is important to note that there is a parameter that can be added to this algorithm which also calculates the MSA cost of the cells formed by splitting the polygon. The algorithm considers all the adjacent vertices of the current point and determines the MSA cost as well as the convexity of the cells formed by merging the ear-clipped triangles. Then, the convex cell which gives the least MSA cost is selected. If this MSA cost is found to be larger than the sum of the MSA cost of the unmerged cells, then the algorithm will discard the merged polygon and will proceed to the next triangle. Otherwise, the merged cell is selected to remain in the set of partitioned cells as the algorithm moves to the next point. The decision to merge a cell depending upon the convexity and MSA optimality makes sure that every convex cell within the partitioned polygon is MSA optimal.

#### 5.2.4 Real-time SLAM-assisted CPP

Applications involving autonomous navigation and planning of mobile agents can benefit greatly by employing real-time SLAM techniques. SLAM techniques allow for a robust estimation of robot position as well as the interest points in the map simultaneously through Bayesian inference. Generic SLAM techniques provide a navigation solution with minimal deviation from the preplanned trajectory of the robot. In addition, the objective function of CPP algorithm is to achieve the maximum area coverage and this target function cannot be performed without a reasonable navigation solution. To ensure obtaining an acceptable navigation information, a SLAM technique should be implemented for the CPP problem. In this phase of the research, a CPP-based SLAM approach is proposed to allow for dynamic modification of the CPP path with successive detection of map features as well as achieving a robust mapping application.

#### SLAM technique

SLAM in mobile robotics refers to the process of developing a map of an unknown environment by a mobile robot while concurrently pinpointing the position of the robot within this map [133]. To be more specific, such a technique estimates the robot poses and landmarks at the same time and is classified as a classic chicken-or-egg problem. SLAM is an essential method to a range of indoor, outdoor, air, and underwater applications for both manned and unmanned vehicles. In other words, SLAM in GNSS-denied environments is considered a new major challenge in the mapping community. By utilizing SLAM techniques based on the probabilistic Bayesian framework, the unmanned platform moves through an unknown environment and performs feature extraction using the onboard mapping sensor at certain time intervals. Both navigation and landmark detection are estimated using platform sensor data. The expression for the Bayesian estimate, which contains the pose and landmark vectors of robot as given by Equation 5.4, is derived from alternating steps of



prediction and correction for each observation of the platform. In this equation,  $x_t$  denotes the current state at time  $t$ ,  $z_{1:t}$  represents the sensor observations till  $t$ ,  $bel(x_t)$  represents the belief at time  $t$ ,  $u_{1:t}$  represents the control inputs till  $t$ , and  $\eta$  denotes the normalizing constant.

$$\begin{aligned} bel(x_t) &= p(x_t \mid z_{1:t}, u_{1:t}) \\ &= \eta p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1} \end{aligned} \quad (5.4)$$

The primary belief equation for SLAM,  $bel(x_t)$ , describing the current state of the platform can be broken down into the prediction and correction steps as illustrated in Equations 5.5 and 5.6, respectively [134].

$$\overline{bel}(x_t) = \int p(x_t \mid x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1} \quad (5.5)$$

$$bel(x_t) = \eta p(z_t \mid x_t) \overline{bel}(x_t) \quad (5.6)$$

### 5.2.5 System Implementation

At this phase of research, the proposed SLAM-assisted CPP approach is implemented for a UGV platform for indoor MMS. Roomba Create2 is utilized as the unmanned platform which is equipped with a 2D-LiDAR unit (RPLiDAR-A2) as shown in Fig. 5.4. Roomba Create2 is a programmable robot that is easy to program and control through the use of several programmable kits (i.e., Raspberry pi and Arduino kit). Due to the compact design of the programmable robot, it is necessary to make some design modifications by constructing extra plates which hold more sensors. The RPLiDAR-A2 is a low cost a 360-degree 2D-LiDAR which can generate 4,000 pulses per second with high rotation speed as well as performing 360-degree scan at a 6-meter range. This system is implemented in a Robot Operating System (ROS) framework using G-Mapping SLAM. It is important to note that the implementation

of SLAM algorithm is based on Rao-Blackwellized particle filters which have been introduced as effective numerical methods to solve the SLAM problem. Specifically, the particle filter is mainly used for estimation problem of non-linear motion model and non-linear Gaussian state space models. This approach uses a particle filter in which each particle carries an individual map of the environment. Each particle represents a possible robot trajectory and a map. The framework of particle filter has been extended for solving the SLAM problem with a landmarks map [134].

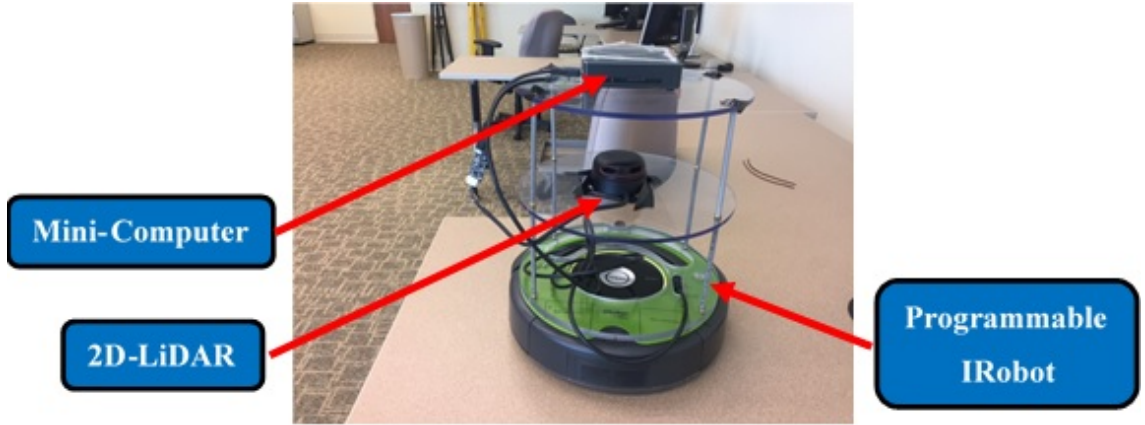


Fig. 5.4.: System Implementation using Roomba Create 2

### 5.2.6 Experimental Results

In order to test the developed CPP algorithm, a test benchmark is proposed that generates randomized polygons with an increasing number of sides. When testing the entire system comprising of the offline CPP algorithm as well as the real-time SLAM assisted CPP, a standard L-shaped area benchmark is used [77]. Such benchmark provides a 2D-area that can check the cellular decomposition-based path planning algorithm. To be more specific, the concavity of the L-shaped polygon checks for its ability to perform cellular decomposition using the Ear-Clipping algorithm.

The above system implementation has been used for executing the SLAM-assisted CPP frameworks for performance and analysis, the results of which are described in this section. Fig. 5.5 (a-c) shows the step-by-step operation of the combined approach between the offline CPP and real-time SLAM technique on a target area with the implemented system. Fig. 5.5 (a) and Fig. 5.5 (b), respectively, show the original area of interest that needs to be covered and the generated total coverage path. Fig. 5.5 (c) shows the actual trajectory of the robot during online navigation.

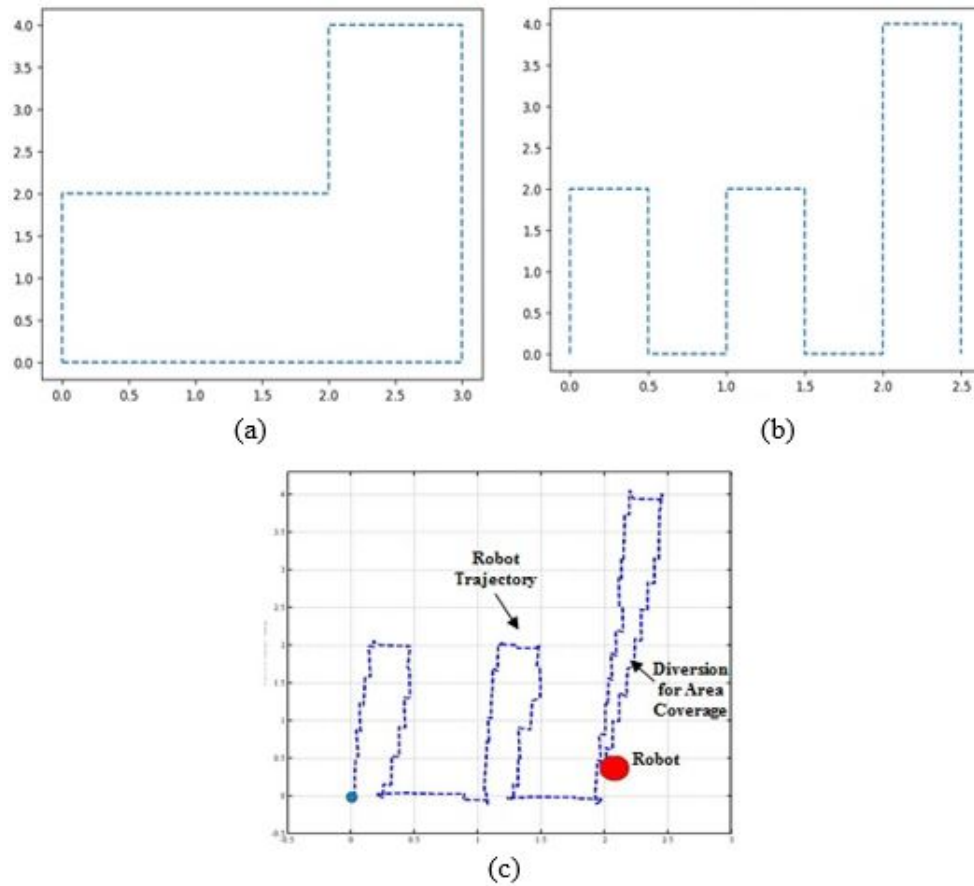


Fig. 5.5.: Results for SLAM-assisted CPP framework: (a) Original Area, (b) Total Coverage Path Generated, and (c) Robot Path during Navigation

To evaluate the results, a qualitative analysis has been done to check the loop closure error which is defined as the difference between the first and last point with

respect to total distance covered. Therefore, nineteen experiments are performed using several number of particles to see if there is any improvement of the loop closure error. One can observe that the loop closure error (%) when using a real-time SLAM algorithm (the mean value=2.15) is much less than the loop closure error (%) when only using the odometer of the UGV without SLAM algorithm (the mean value=14.01) as shown in Fig. 5.6 and Fig. 5.7 respectively. In addition, it is noticed that the loop closure error (%) is independent of the number of particles.

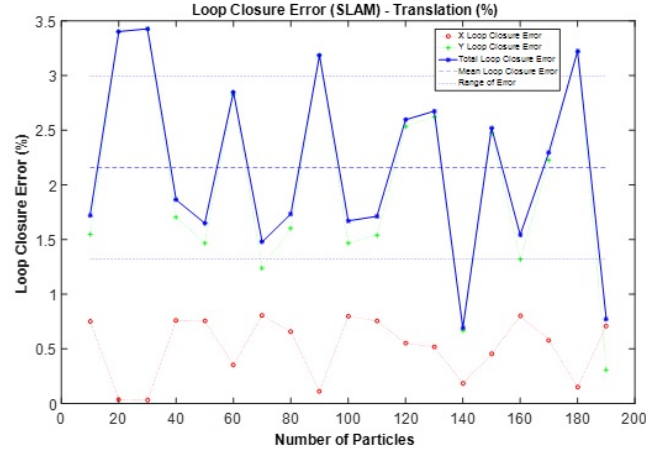


Fig. 5.6.: Loop Closure Error with SLAM implementation

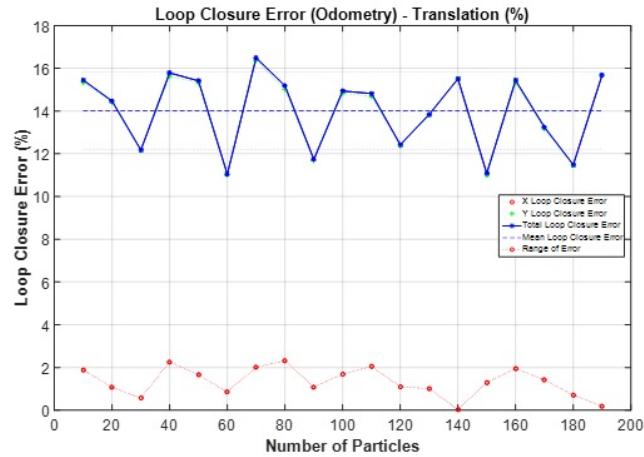


Fig. 5.7.: Loop Closure Error without SLAM implementation (Odometry)

### 5.3 Implementation of Pseudo-GNSS/INS Localization System for Indoor LiDAR MMS

As an extension to this research work, extending the MMS to GNSS-denied areas is quite challenging and necessitates the development of a feasible substitute to the GNSS/INS module for system operation within GNSS-denied areas. It is important to note that such an alternative module should be implemented while keeping the main skeleton of the MMS. This research has been motivated by attempts at robust integration of different components onboard MMS. Such MMS achieve the purpose of high precision and accurate mapping previously mentioned in Chapter 3. The integration of GNSS/INS module and mapping sensors amplifies both the advantages and drawbacks of a GNSS/INS unit. Since the generated geo-referenced map is only confined to regions where there is a consistent availability of GNSS signals, this stands out as an obvious limitation of the GNSS/INS module. Therefore, MMS are unsuitable for several applications in GNSS-denied environments such as indoor surveying and underground navigation. For such applications where the availability of GNSS signal is scarce or non-existent, the MMS framework requires a positioning system other than a GNSS/INS module. In such a case, use of probabilistic techniques such as the SLAM method have allowed for the development of a MMS capable of operating in GNSS-denied areas. While the performance of such systems does closely match that of a GNSS/INS-based MMS in general, the two implemented frameworks are completely different in structure from each other. As a result, it is difficult to combine the two systems without making comprehensive changes within the framework implementation.

To solve this problem, this phase of research proposes the development of a Pseudo-GNSS/INS module that serves as a convenient substitute to a GNSS/INS-based MMS framework for the purpose of operation within GNSS-denied environments. While this Pseudo-GNSS/INS module uses probabilistic SLAM techniques for estimating the MMS pose, such a module is implemented to behave identically

to a GNSS/INS unit. Therefore, the MMS framework can simply switch between the real GNSS/INS unit and the Pseudo-GNSS/INS module for seamless operation. The major notions that bring about this implementation include: (i) utilizing frame transformation to extract 2D scan from the entire 3D-point cloud for efficiently carrying out real-time SLAM, (ii) generating of the position data from these real-time SLAM pose estimates, (iii) providing the necessary geo-referencing signals identical to a GNSS/INS module and such signals are utilized in the MMS operation, and (iv) incorporating these SLAM pose estimates with LiDAR data to reconstruct a complete geo-referenced laser point cloud in the post-processing mode.

### 5.3.1 SLAM-based MMS for GNSS-denied Environments

The main limitation of a GNSS/INS positioning module is its dependency on the availability of a GNSS signal. Hence, for operation in GNSS-denied areas, the position and orientation data for the MMS needs to be generated using alternative methods. The concept of SLAM allows this position and orientation data to be generated from the sensor data itself. Therefore, it is possible to estimate the robot trajectory simultaneously with the map using a Bayesian framework described by SLAM. Such an MMS framework is thus capable of operating without the need of a GNSS/INS module. Generally, a SLAM-based MMS framework stands out from the GNSS/INS-based MMS framework previously mentioned in Chapter 3, because of the following distinction:

- The position data generated by SLAM is not independent from the sensor data but is simultaneously generated with the observation of landmarks for map reconstruction.
- The sensor data logged onboard the MMS is not tagged with any geo-referencing signals.

- Pose estimation of the MMS from data association of landmarks in successive data streams is computationally very expensive. Hence, it is very difficult to implement real-time SLAM for pose estimation without the aid of costly parallel processors onboard the MMS.

Due to such distinctions, it is difficult to incorporate a SLAM-based MMS framework into a GNSS/INS- based MMS framework directly, without extensive modifications to the system. The proposed framework in this research introduces the implementation of a SLAM-based Pseudo-GNSS/INS system that virtually mimics a GNSS/INS unit. Therefore, the proposed framework can be operated with or without a GNSS/INS module in GNSS-affluent as well as GNSS-denied environments without requiring any modification in structure. This proposed framework will be explained in detail in the next section.

### 5.3.2 Proposed Methodology of a Pseudo-GNSS/INS Module

The proposed pseudo-GNSS/INS framework which utilizes the SLAM techniques to allow operation of an MMS in GNSS-denied areas is illustrated in Fig. 5.8. Such framework is done in a way that a physical GNSS/INS module can be added to the system without making any changes to the MMS framework. One should note that a pseudo-GNSS/INS module onboard MMS behaves like a GNSS/INS unit, however, such module is based on SLAM for pose estimation. The block description in Fig. 5.8 also shows other novel features of the pseudo-GNSS/INS module such as it generates the necessary geo-referencing signals for the mapping sensors. Such a module generates position data based on the sensor data in real-time and performs the entire operation using a single 3D-mapping sensor (i.e., LiDAR unit).

The main concepts that allow the implementation of such system are enumerated as follows:

- 2D Scan Extraction: The idea behind carrying out a computationally efficient real-time SLAM for the pseudo-GNSS/INS module is that if the focus of SLAM

is to derive a pose estimation for the MMS then the SLAM operation can be carried out using only a fraction of the sensor data. One should note that a simplified 2D-planar motion is considered an assumption in this part of research. This is because a highly dense sensor data maybe required for robust map reconstruction but may not be mandatory for the localization of the MMS as is done by the Pseudo-GNSS/INS framework. Such extraction is formally explained in the next sections for the case of an indoor 3D-LiDAR MMS.

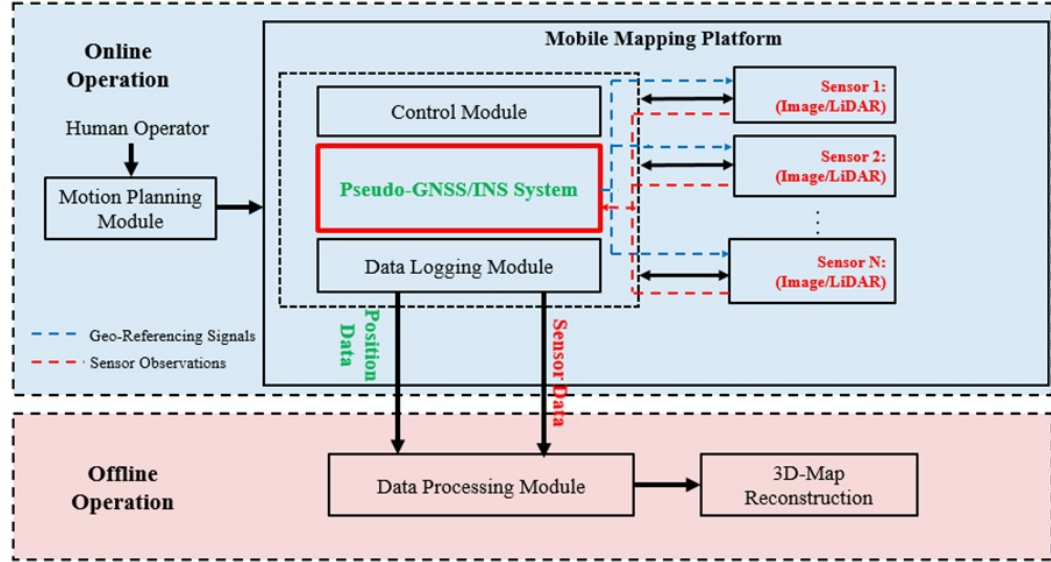


Fig. 5.8.: Functional block diagram of the proposed Pseudo-GNSS/INS framework for an Indoor MMS

- **Single Sensor Localization Operation:** Because of the 2D-scan generation, the powerful advantage of the pseudo GNSS/INS framework is that the positioning module and the sensor network in the MMS get incorporated into a single unit comprised of one 3D-mapping sensor. This is considered a very important feature of the pseudo-GNSS/INS based MMS and not only simplifies the system integration but also reduces the payload of the MMS.



- **Independent Logging of Position/Body-Orientation Data:** Another important consequence of the real-time SLAM operation is that the position/body-orientation data can now be logged from SLAM pose estimation separately from the sensor, however, such data is generated dependently on the sensor information. This provides a substitute data to the MMS in place of the GNSS/INS data making it possible to operate in GNSS-denied environments.
- **Pseudo-GNSS/INS Signal Generation:** Since the SLAM-based pseudo-GNSS/INS framework illustrated in Fig. 5.8 can generate pose estimates in real-time. Such framework can also provide geo-referencing signals for the sensors in real-time. Therefore, the sensor data has the advantage of being time-tagged for map reconstruction as it would be for a real GNSS/INS module.

It is important to note that the above points will be elaborated on in the upcoming sections for a pseudo-GNSS/INS based MMS framework for the case of an indoor mapping system using a single 3D-LiDAR unit.

### **Real-Time SLAM for MMS Positioning**

The significant point of the implementation of the pseudo-GNSS/INS framework is the fact that only a 2D-scan of the entire point cloud in the sensor data stream is required for the purpose of robust pose estimation. Therefore, if robust pose estimation is the primary aim for SLAM operation, then a good estimate can still be obtained by confining the landmarks to the plane of motion of the robot after making an assumption of 2D-motion. This gives a provision to extract a 2D-point cloud from the 3D-sensor data for the purpose of localization of the robot that include the position and orientation of the platform. Such strategy is used in this research for generating real-time pseudo-GNSS/INS data. Fig. 5.9 depicts the extraction of a 2D-scan from the entire 3D-LiDAR point cloud for an indoor MMS.

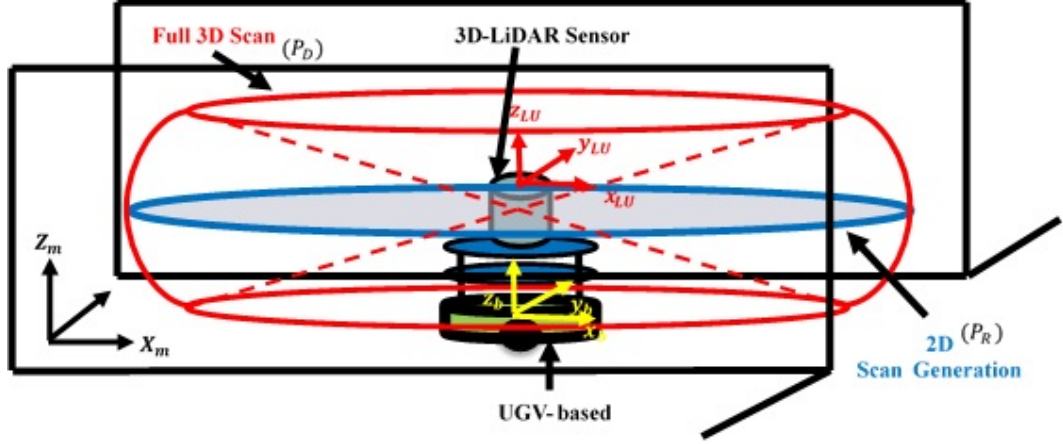


Fig. 5.9.: Real-time SLAM for MMS positioning, extraction of a 2D-scan from the full 3D-LiDAR point cloud

Fig. 5.9 shows that the body-coordinate systems and the mapping sensor-coordinate systems are almost aligned. Since the motion of the robot is constrained to only 2D-planar motion and rotation around a vertical axis, the SLAM operation can be efficiently carried out by extracting a 2D-point cloud corresponding to the plane parallel to the horizontal plane of motion of the robot. However, if the body-coordinate systems and the mapping sensor-coordinate systems are not aligned, an arbitrary orientation is considered between both coordinate systems. Therefore, a 2D-scan can be generated by performing a spatial filtering operation of the 3D-point cloud as illustrated in Equation 5.7 and Equation 5.8, respectively.  $P_R$  denotes the reduced 2D-point cloud,  $P_D$  is the original 3D-point cloud,  $h(P_D, \beta, \Delta\theta)$  is a windowing filter,  $\beta$  denotes the angle formed by the point w.r.t sensor frame,  $\Delta\theta$  is the angle difference between the body and sensor frames, and  $\delta\theta$  is the user-defined threshold for the filter. This operation can be easily visualized for a 3D-point cloud as depicted in Fig. 5.10.

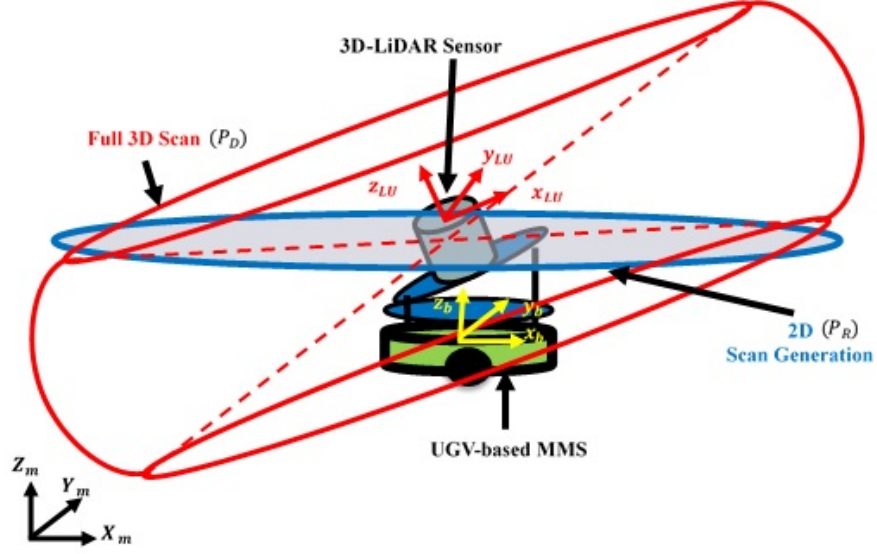


Fig. 5.10.: Visualization of the windowing operation for extracting the 2D-point cloud

$$P_R = h(P_D, \beta, \Delta\theta) \cdot P_D \quad (5.7)$$

$$h(P_D, \beta, \Delta\theta) = \begin{cases} 1 & \text{for } |\beta + \Delta\theta| \leq \delta\theta \\ 0 & \text{Otherwise} \end{cases} \quad (5.8)$$

### Pseudo-GNSS/INS Signal Generation

Another important step towards the implementation of the framework in Fig. 5.8 is the generation of the time tags signals to extract the corresponding position and orientation information of the platform. For example, in order to derive direct georeferencing data, a GNSS/INS unit supplies sequentially precise time pulses, known as a PPS signal, which gives the ability to generate a time-tagged point cloud. For the pseudo-GNSS/INS framework, the same step is carried out by generating sequential time pulses using an embedded system kit (i.e., Raspberry pi). Furthermore,

the pseudo-GNSS/INS framework provides the navigation message (GPRMC message) in the same format (NMEA-format). Such a navigation message includes the information about position, rotation, and GPS time. For the system implementation described in this phase of research, such a message is recorded over a dedicated RS-232 serial port and received by the LiDAR unit (i.e., Velodyne VLP-16 Puck) via the interface box in the form of serial data. Such system implementation which describes the geo-referencing signals generation is shown in Fig. 5.11.

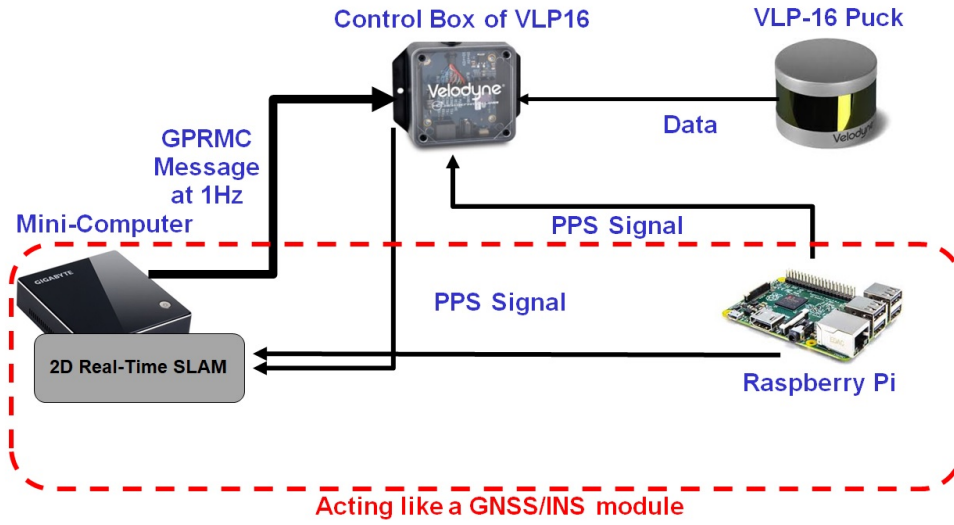


Fig. 5.11.: Pseudo-GNSS/INS framework geo-referencing signals generation

### 5.3.3 Hardware System Implementation

The proposed pseudo-GNSS/INS framework has been implemented onboard UGV for an indoor mapping application. It is important to note that the structure and blocks for such a framework are the same as those for a GNSS/INS-based MMS. The same system can be used with a real GNSS/INS module without any modifications required. The UGV platform utilized is the programmable robot, Roomba iCreate2

as mentioned before. Such platform comprises 3D-LiDAR sensor, a mini-computer, power supply unit and a Raspberry Pi 3 as shown in Fig. 5.12.

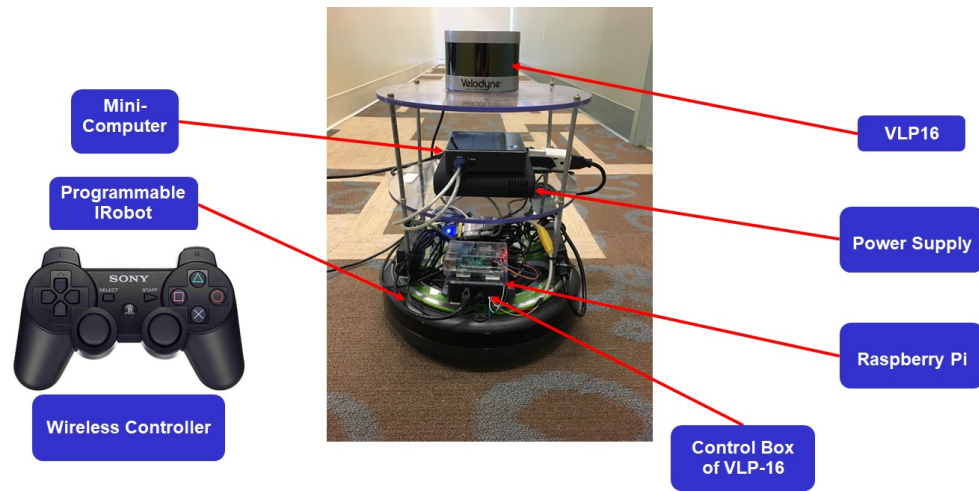


Fig. 5.12.: Implemented Indoor MMS with Pseudo-GNSS/INS framework

The mapping sensor used for such MMS implementation is the Velodyne VLP-16 Puck which is already discussed in Chapter 3. The overall operation of the system is controlled by a mini-computer which contains the data logging module, the control module, the user interface module, and most importantly, the pseudo-GNSS/INS framework. All the system blocks in this mini-computer are implemented in a ROS environment. The UGV is human operated via a Bluetooth controller that interfaces with a computer. A hardware implementation is utilized to generate synchronization pulses from the Pseudo-GNSS/INS framework to be connected to the sensor. A Raspberry Pi 3 is used in the system configuration since such a unit is responsible for sequentially generating the PPS signal in order to derive direct geo-referencing data.

#### 5.3.4 Operational Strategy

Since the 3D-LiDAR sensor onboard the MMS is crucial for both positioning and map construction for the MMS, the presence of any sensor noise has an obvious

effect on the sensor measurements. One should note that the sensor noise distinctly increases through the motion. Therefore, such noise negatively affects the generated trajectory that is basically based on real-time SLAM technique. Such a situation creates another challenge for the proposed system that needs to be addressed. This can be resolved by developing a data collection strategy that has multi-stationary scans through the whole dataset thus ensuring an overlap between successive stations. The idea behind such a strategy is to avoid any accumulated error which is generated through motion. In terms of off-line operation, all the points are reconstructed by incorporating the generated trajectory based on real-time SLAM with the LiDAR data in order to produce geo-referenced 3D reconstruction. To achieve an improved accuracy and better inference of the 3D-indoor environment, the ICPP algorithm is utilized to register the generated time-tagged point cloud from the stationary scans. It is important to note that those stationary scans will become trusted locations after applying ICPP technique to allow for a trajectory enhancement between every two successive locations.

### **5.3.5 Post-Processing Enhancement**

#### **ICPP Registration Technique**

The proposed framework generates the SLAM-based position and orientation data from only a 2D-scan point cloud. However, as far as the final map reconstruction is concerned, the result can still be further improved by utilizing the entire 3D point cloud in a post-processing mode for pose correction. This is considered one motivation for carrying out post-processing 3D-registration for the stationary scans since the proposed method for generating a trajectory of the mapping platform is completely based on only a small fraction of 3D-LiDAR unit measurements. Therefore, a registration approach is performed by applying the ICPP algorithm on the stationary scans of the entire trajectory for the simultaneous registration of multiple overlapping point clouds. Such a technique can allow proper enhancement of the reconstructed

map since the registration procedure gives the opportunity for the determination of inconsistencies between the generated surfaces from the stationary scans.

The conceptual basis of the ICPP method is briefly summarized below [135]:

- First a point to triangular patch which includes the closest three points match is established by examining if the point locates within the triangular di-pyramid, which has the three triangular patch points as a base and a user-chosen normal distance as the height to set the two peaks as depicted in Fig. 5.13.

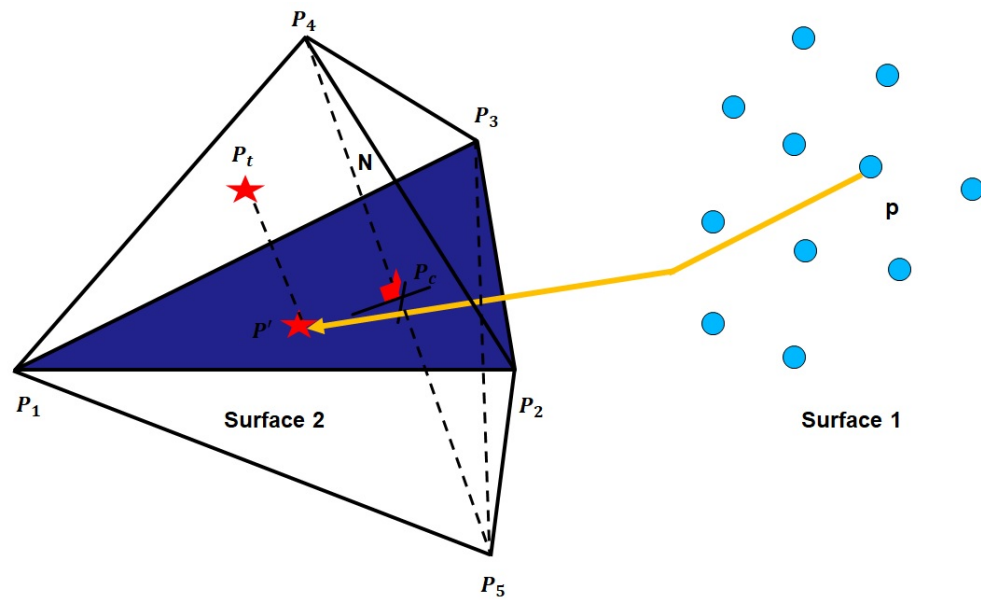


Fig. 5.13.: The demonstration of ICPP registration technique [135]

- Then, the point is projected onto the patch surface, and its projection is then used as a match for the original point.
- This process is carried out for all valid three points combinations within the point cloud.

## Trajectory Improvement using Stationary Scans

As mentioned before, there is a significant need to enhance the real-time SLAM-based trajectory to improve the generated 3D-reconstruction of the mapping environment of interest. Therefore, a smoothing approach is proposed based on the registered stationary scans. It is worth noting that the registered stationary scans are derived after applying ICPP method for such locations along the entire trajectory. Therefore, the ICPP-based stationary scans are considered as trusted locations. In this phase of research, a smoothing technique is implemented through averaging between the forward and backward trajectories. The forward trajectory is generated based on real-time SLAM technique at 1Hz data rate. However, the backward trajectory is derived based on the ICPP-based stations while considering the translation and orientation parameters of the epochs (which refers to the instance where the position and orientation of the platform is obtained from the forward trajectory) along the forward trajectory.

The entire forward trajectory is divided into several segments, each segment is established between each two ICPP-based stationary scan locations. Then, the relative translation ( $r_{i+1}^i(F)$ ) and the rotation matrix ( $R_{i+1}^i(F)$ ) between each two epochs (1Hz data rate) are computed through the segment in the forward trajectory as illustrated in Equation 5.9 and Equation 5.10, respectively. It is important to note that such a computation process is based on successive starting and ending stationary scan locations as shown in Fig. 5.14. In this figure,  $r_i^m(F)$  denotes the position of epoch  $i$  with respect to the mapping frame in the forward trajectory,  $R_i^m(F)$  denotes the orientation of epoch  $i$  with respect to the mapping frame in the forward trajectory, and  $i, i + 1, \dots, N$  represents the number of epochs.

$$r_{i+1}^i(F) = R_m^i(F)[r_{i+1}^m(F) - r_i^m(F)] \quad (5.9)$$

$$R_{i+1}^i(F) = R_m^i(F)R_{i+1}^m(F) \quad (5.10)$$



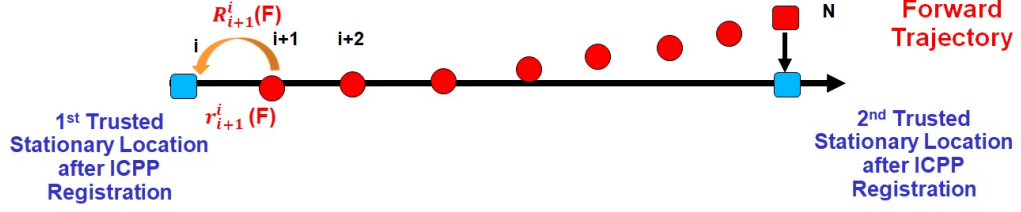


Fig. 5.14.: The relative translation and orientation parameters between each two epochs through the segment

Then, the ICPP procedure is applied for the stationary scan locations to derive the trusted locations at the beginning and ending of each segment. For the 1<sup>st</sup> segment, the updated position and orientation of the stationary location relative to the mapping frame ( $r_{b(t_2)}^{m(t_1)}$  &  $R_{b(t_2)}^{m(t_1)}$ ) given the initial forward trajectory ( $r_{b(t_2)}^{m(t_2)}$  (F) &  $R_{b(t_2)}^{m(t_2)}$  (F)) and the outcome from the ICPP registration between the 1<sup>st</sup> stationary station and the 2<sup>nd</sup> stationary station through the 1<sup>st</sup> segment is illustrated in Equation 5.11 and Equation 5.12. The drift in the 1<sup>st</sup> segment will be modeled as incremental translation and orientation parameters between mapping frames at different times ( $r_{m(t_2)}^{m(t_1)}$  &  $R_{m(t_2)}^{m(t_1)}$ ). Therefore, the ICPP algorithm focuses on the estimation of such parameters. For the 2<sup>nd</sup> and 3<sup>rd</sup> segments, the adjusted position and orientation of the stationary scan locations relative to the mapping frame are derived by using concatenation of motion estimation as shown Fig. 5.15. Equations 5.13-5.16 illustrate the derivation of the updated stationary scan locations for the 2<sup>nd</sup> segment and 3<sup>rd</sup> segment, respectively.  $r_{m(t_3)}^{m(t_2)}$  and  $R_{m(t_3)}^{m(t_2)}$  denote the outcome from ICPP registration between the 3<sup>rd</sup> stationary station and the 2<sup>nd</sup> one through the 2<sup>nd</sup> segment, and  $r_{m(t_4)}^{m(t_3)}$  and  $R_{m(t_4)}^{m(t_3)}$  represent the outcome from ICPP registration between the 4<sup>th</sup> stationary station and the 3<sup>rd</sup> one through the 3<sup>rd</sup> segment.

$$r_{b(t_2)}^{m(t_1)} = r_{m(t_2)}^{m(t_1)} + R_{m(t_2)}^{m(t_1)} r_{b(t_2)}^{m(t_2)}(F) \quad (5.11)$$

$$R_{b(t_2)}^{m(t_1)} = R_{m(t_2)}^{m(t_1)} R_{b(t_2)}^{m(t_2)}(F) \quad (5.12)$$

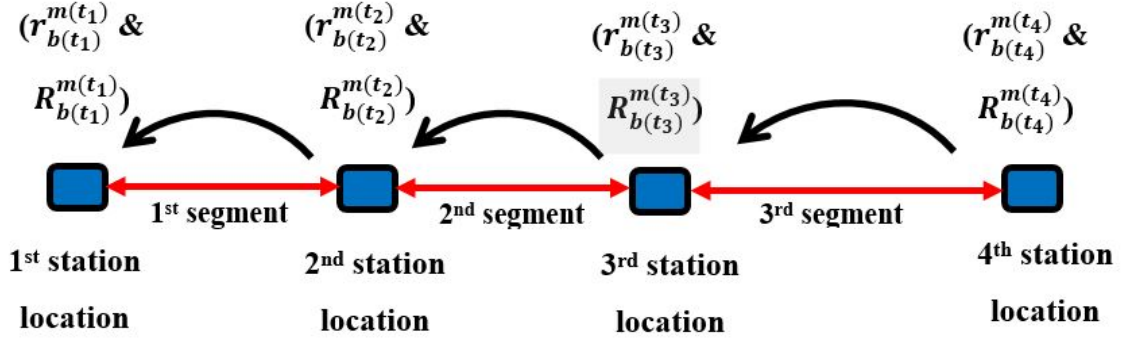


Fig. 5.15.: The derivation of stationary scan locations using concatenation of motion estimation

$$r_{b(t_3)}^{m(t_1)} = r_{m(t_2)}^{m(t_1)} + R_{m(t_2)}^{m(t_1)}[r_{m(t_3)}^{m(t_2)} + R_{m(t_3)}^{m(t_2)}r_{b(t_3)}^{m(t_3)}(F)] \quad (5.13)$$

$$R_{b(t_3)}^{m(t_1)} = R_{m(t_2)}^{m(t_1)}R_{m(t_3)}^{m(t_2)}R_{b(t_3)}^{m(t_3)}(F) \quad (5.14)$$

$$r_{b(t_4)}^{m(t_1)} = r_{m(t_2)}^{m(t_1)} + R_{m(t_2)}^{m(t_1)}[r_{m(t_3)}^{m(t_2)} + R_{m(t_3)}^{m(t_2)}[r_{m(t_4)}^{m(t_3)} + R_{m(t_4)}^{m(t_3)}r_{b(t_4)}^{m(t_4)}(F)]] \quad (5.15)$$

$$R_{b(t_4)}^{m(t_1)} = R_{m(t_2)}^{m(t_1)}R_{m(t_3)}^{m(t_2)}R_{m(t_4)}^{m(t_3)}R_{b(t_4)}^{m(t_4)}(F) \quad (5.16)$$

After computing the relative translation and orientation information between each two epochs through the segment and deriving the updated stationary location, the backward trajectory is derived as illustrated in Equation 5.17 and Equation 5.18, respectively. Fig. 5.16 shows the derivation of the backward trajectory via the segment.  $r_{i+1}^m(B)$  and  $R_{i+1}^m(B)$  denote the updated stationary location after applying ICPP procedure.

$$r_i^m(B) = r_{i+1}^m(B) + R_{i+1}^m(B)r_i^{i+1}(F) \quad (5.17)$$

$$R_i^m(B) = R_{i+1}^m(B)R_i^{i+1}(F) \quad (5.18)$$

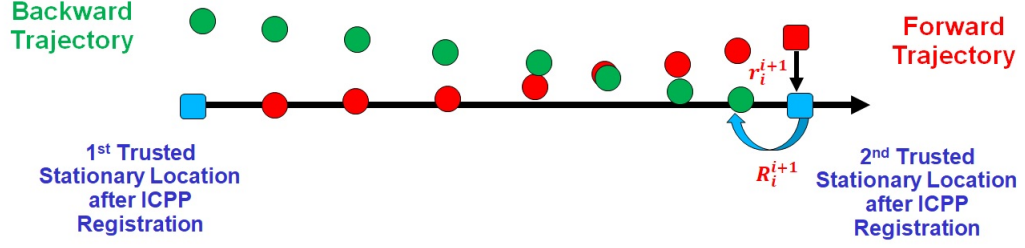


Fig. 5.16.: The derivation of the backward trajectory

A smoothed trajectory is derived by combining both forward and backward trajectories. It is important to note that such trajectories will start and end in trusted successive scan locations. Therefore, an interpolation method should be performed to derive the position and orientation information of the smoothed trajectory. For position interpolation, a linear interpolation should be performed between every two epochs in both forward and backward trajectories through computing the weight factors for both trajectories as illustrated in Equation 5.19. In this equation,  $t_F$  denotes the difference between the time of the trusted location in the beginning of the forward trajectory ( $t_1$ ) and the time of any epoch ( $t_S$ ) through such trajectory. However,  $t_B$  denotes the difference between the time of the trusted location in the beginning of the backward trajectory ( $t_2$ ) and the time of any epoch ( $t_S$ ) through such trajectory as depicted in Fig. 5.17. Also,  $W_F$  and  $W_B$  denote the weight factors of the forward and backward trajectories, respectively.

$$r_i^m(\text{Smoothed}) = W_F r_i^m(F) + W_B r_i^m(B) \quad (5.19)$$

$$\begin{aligned} t_F &= t_s - t_1 & t_B &= t_2 - t_s \\ W_F &= \frac{t_B}{t_F + t_B} & W_B &= \frac{t_F}{t_F + t_B} \end{aligned}$$

As mentioned previously, the rotation interpolation can be performed by utilizing a spherical linear interpolation since it is considered the best way to do the rotation

interpolation by using quaternion representation as depicted in Fig. 5.18. The rotation matrices of both forward and backward trajectories are represented by  $q_{R_i^m(F)}$  and  $q_{R_i^m(B)}$ , respectively, whose angular deviation is  $\theta$ . The angular deviation  $\theta$  can be derived as illustrated in Equation 5.20. Also, the angular deviations of the interpolated quaternion  $q_{R_i^m(Smoothed)}$  to  $q_{R_i^m(F)}$  and  $q_{R_i^m(B)}$  are  $\theta_F$  and  $\theta_B$ , respectively and can be derived as shown in Equation 5.21 and Equation 5.22, respectively.

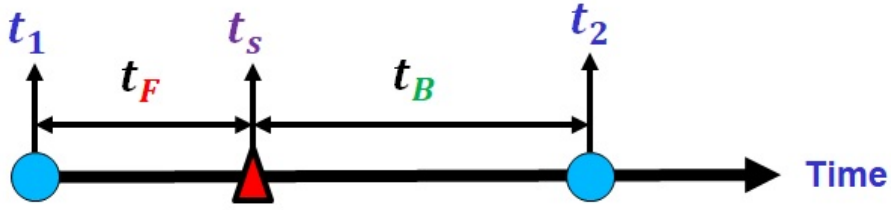


Fig. 5.17.: The computation of the time difference between the time of the trusted location and the time of any epoch through the trajectory either forward or backward trajectory

$$\theta = \cos^{-1}(q_{R_i^m(F)} \cdot q_{R_i^m(B)}) \quad (5.20)$$

$$\theta_F = \frac{t_F}{t_F + t_B} \theta \quad (5.21)$$

$$\theta_B = \frac{t_B}{t_F + t_B} \theta \quad (5.22)$$

To compute  $q_{R_i^m(Smoothed)}$  which denotes the interpolated quaternion,  $C_F$  and  $C_B$  should be derived first through Equation 5.23 and Equation 5.24, respectively. Then, the interpolated quaternion  $q_{R_i^m(Smoothed)}$  can be determined through Equation 5.25. Fig. 5.19 shows the derivation of the smoothing trajectory through the segment.

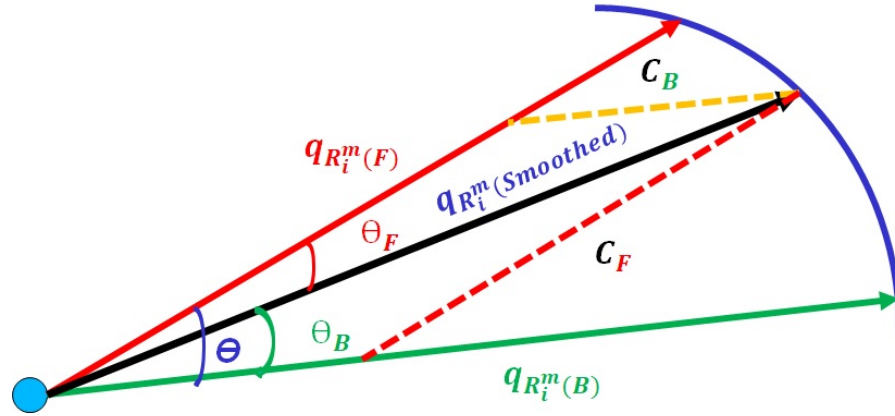


Fig. 5.18.: The determination of orientation of the smoothed trajectory by using spherical linear interpolation

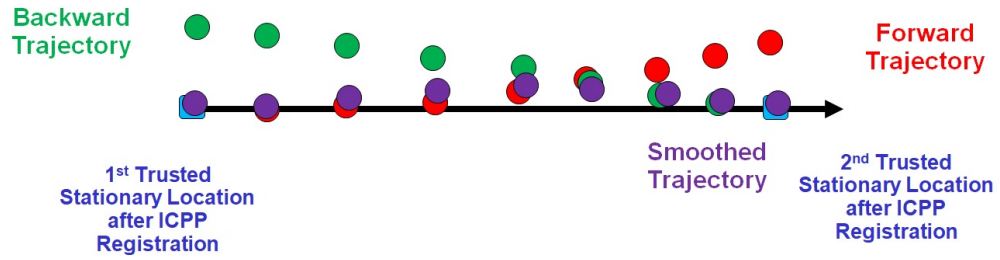


Fig. 5.19.: The derivation of the smoothed trajectory

$$C_F = \frac{\sin \theta_B}{\sin \theta} \quad (5.23)$$

$$C_B = \frac{\sin \theta_F}{\sin \theta} \quad (5.24)$$

$$q_{R_i^m}(Smoothed) = C_F q_{R_i^m}(F) + C_B q_{R_i^m}(B) \quad (5.25)$$

### 5.3.6 Experimental Results

To test the performance of the proposed pseudo-GNSS/INS-based MMS framework, an indoor mapping application was considered (since an indoor environment is essentially denied of GNSS signals). The MMS described above, consisting of a 3D-LiDAR unit as the mapping sensor as well as the onboard pseudo-GNSS/INS module was deployed for two locations. The first location is the Digital Photogrammetry Research Group (DPRG) Lab at Purdue University and is comprised of a single room. The second location is the entire First Level of Mann Hall at Discovery Park, Purdue University and contains a number of corridors and turns. The datasets generated by the pseudo-GNSS/INS-based MMS for both locations show the flexibility of the MMS which can operate in a different environments. Details of the data acquisition for these locations are enumerated in Table 5.1.

Table 5.1.: The details for the data acquisition for the two locations

	<b>Area Description</b>	<b>Stationary Stations</b>	<b>Distance Covered (m)</b>
<b>Location I</b>	<b>Mann Hall 153</b>	2	16.58
<b>Location II</b>	<b>Mann Hall Corridor</b>	17	166.60

The results of the operation of the MMS for the two locations are qualitatively illustrated in Fig. 5.20, Fig. 5.21 and Fig. 5.22, respectively. For location I, Fig. 5.20 (a) shows the platform trajectory which is derived from the pseudo-GNSS/INS position data. Fig. 5.20 (b) shows the 3D-point cloud reconstruction derived from the entire scans through the whole trajectory. Due to the sensor noise and the SLAM algorithm drift error, the generated trajectory is not accurate enough to provide a reasonable 3D-point cloud reconstruction as depicted in Fig. 5.20 (b). However, Fig. 5.20 (c) shows an acceptable 3D-point cloud reconstruction after applying the

ICPP registration for the stationary scan locations through the entire trajectory. For location II, Fig. 5.21 (a) shows the original map for the 1<sup>st</sup> floor of Mann Hall and the planned MMS platform path. Fig. 5.21 (b) shows the platform real-time trajectory which is derived from the pseudo-GNSS/INS position data.

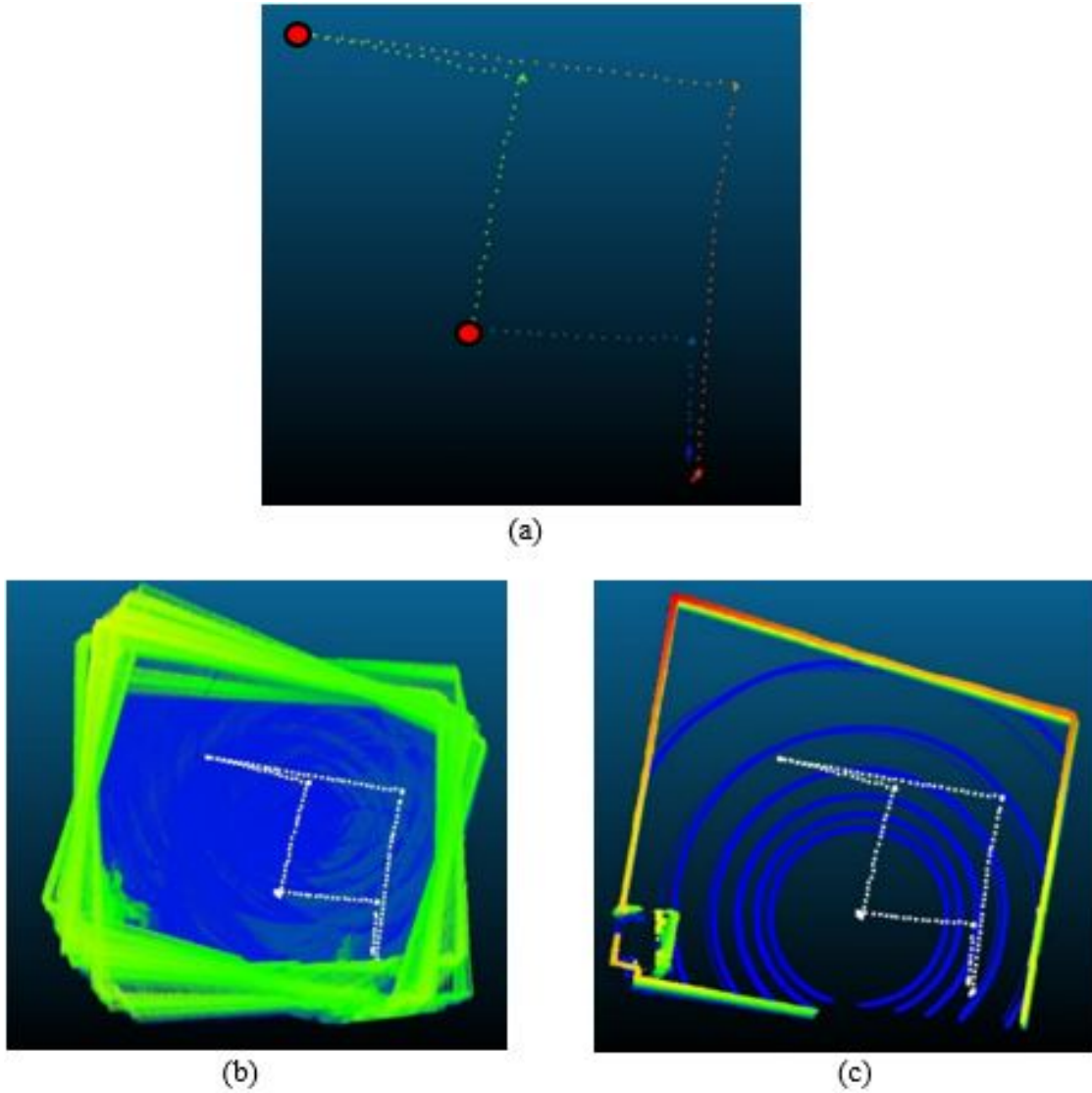


Fig. 5.20.: The results for MMS Mapping Operation for location I (a) Robot trajectory obtained from the Pseudo-GNSS/INS position data, (b) 3D point cloud obtained for the entire scans, and (c) Final reconstructed map after applying ICPP registration for the stationary scan locations



Fig. 5.21.: The results for MMS mapping operation for location II (a) Original map for Mann Hall and (b) Robot trajectory obtained from the Pseudo-GNSS/INS position data

Furthermore, another experiment was performed to test the effect of the laser range of the mapping sensor on the quality of map construction. For location II, this experiment was done by performing the mapping operation on the same targeted area (location II) and execute the 3D-point cloud reconstruction for laser ranges of 70 m and 5 m. It is obvious that the sensor noise contaminating the sensor measurements and the algorithm drift error have an impact on the derived 3D-point cloud reconstruction as shown in Fig. 5.22 (a) and Fig. 5.22 (b), respectively. However, the qualitative results show that reducing the laser range of the LiDAR unit produces a reasonable 3D-point cloud reconstruction even without applying ICP registration. Performing ICP registration for the point cloud has a significant improvement in the quality of the map construction as depicted in Fig. 5.22 (c). After applying ICP registration procedure for the stationary scan locations through the generated trajectory, the effect of such procedures is evident from the superimposed map of location II on the original map as depicted in Fig. 5.22 (d).



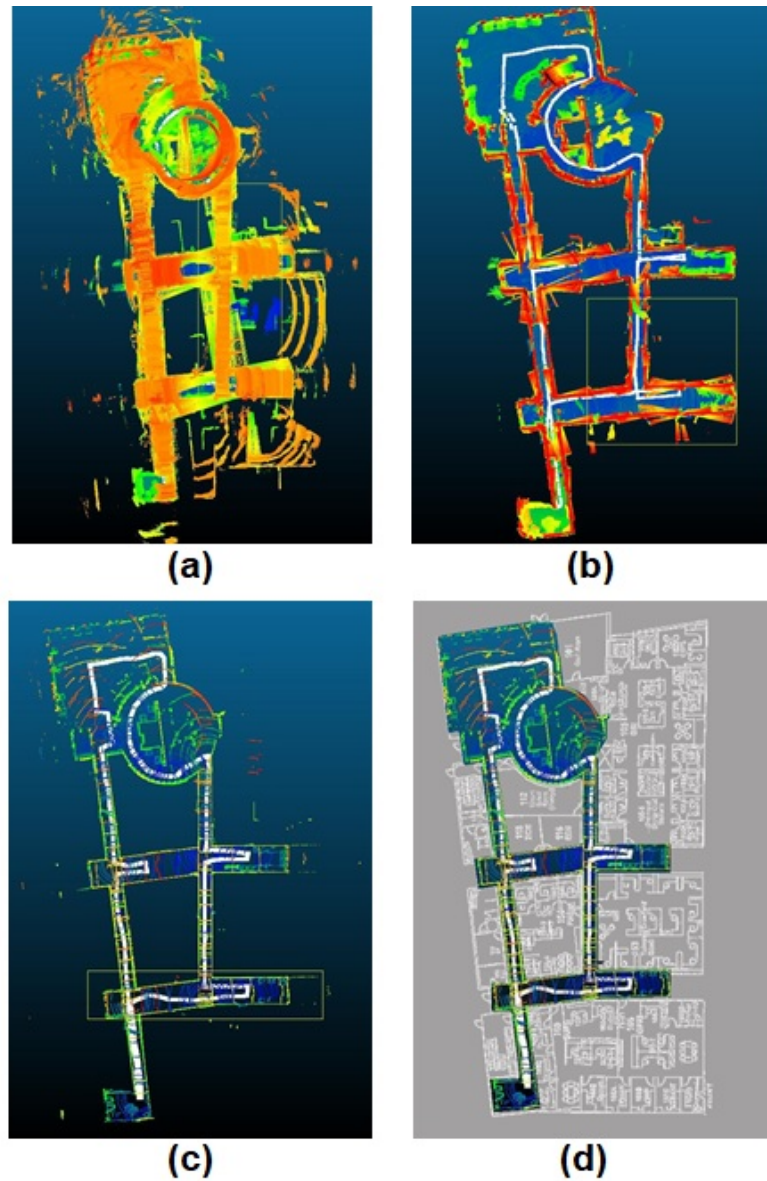


Fig. 5.22.: The results for MMS mapping operation for location II (a) 3D-point cloud obtained for the entire scans, (Range = 70 m) (b) 3D-point cloud obtained for the entire scans, (Range = 5 m), (c) Final reconstructed map after applying ICPP registration for the stationary scan locations, and (d) Reconstructed map over original map

For location II, the smoothing approach for trajectory enhancement is investigated by conducting this technique on three different segments of the entire trajectory. As mentioned before, each segment is established between two successive starting and ending stationary scan locations (i.e., ICPP-based stationary scans). The results of the smoothing approach for the three different segments are qualitatively illustrated from Fig. 5.23 until Fig. 5.49. For the three segments of location II, Fig. 5.23, Fig. 5.24, and Fig. 5.25 show the position of the UGV through the three trajectories (i.e., forward, backward, and smoothed trajectories). It is obvious that the generated smoothed trajectory (green) is very close to the forward trajectory (red) at the beginning of the segments, however, it starts to become closer to the backward trajectory (blue) at the ending of such segments. More specifically, the smoothed trajectory starts and ends at the trusted locations which are derived after applying the ICPP registration technique. Fig. 5.26 - Fig. 5.34 illustrate the heading, pitch, and roll angles of the UGV through the three trajectories and such angles of the smoothed trajectory has the same behavior as the position of such trajectory.

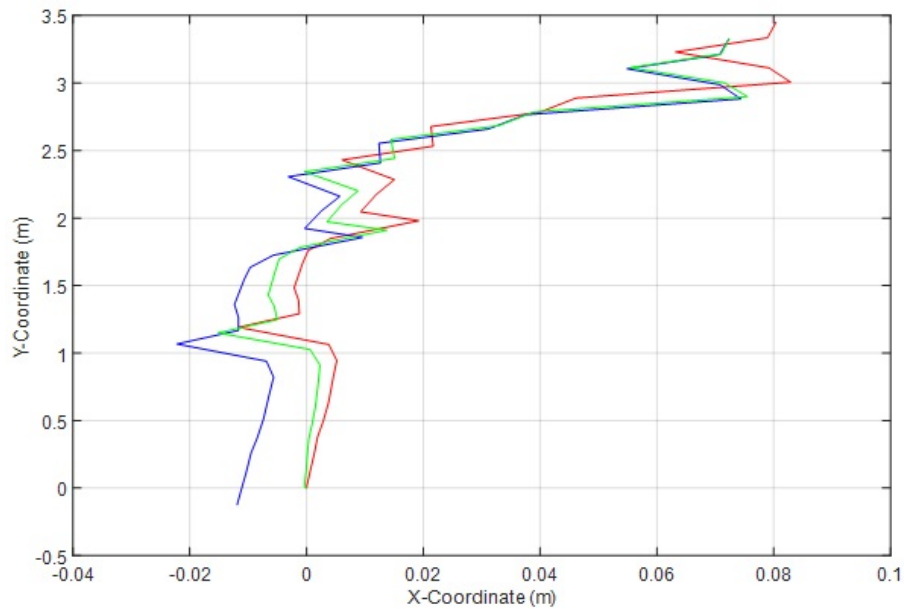


Fig. 5.23.: The position of the UGV platform through different trajectories for the first segment

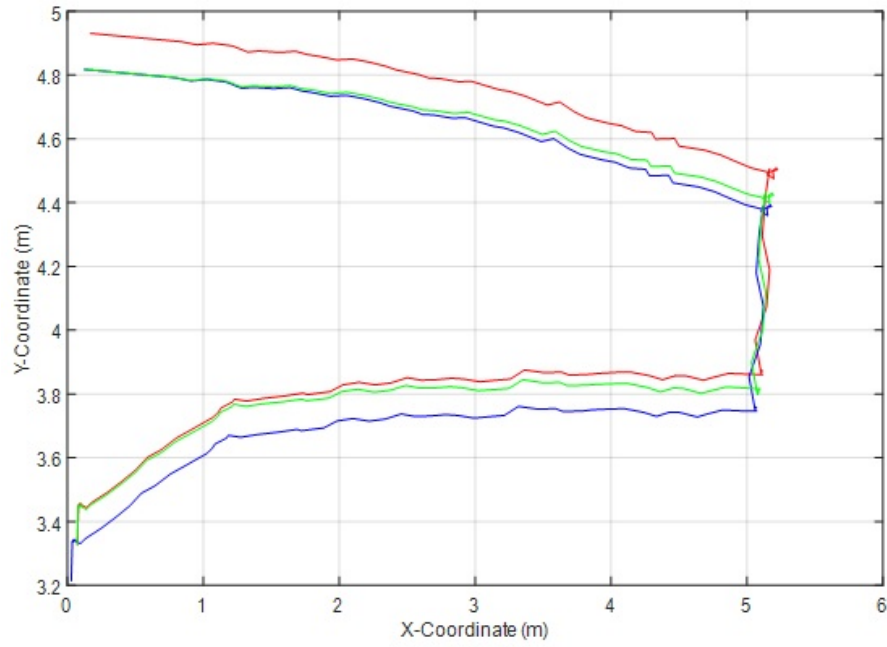


Fig. 5.24.: The position of the UGV platform through different trajectories for the second segment

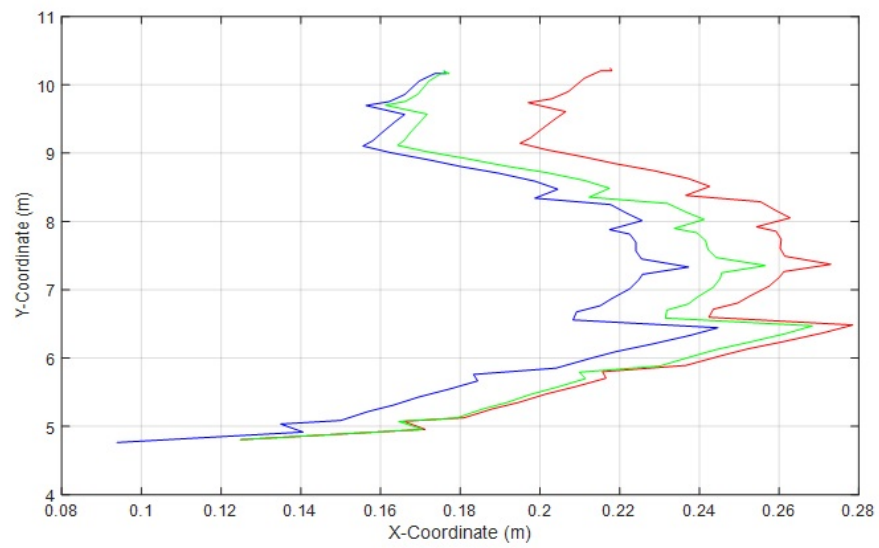


Fig. 5.25.: The position of the UGV platform through different trajectories for the third segment

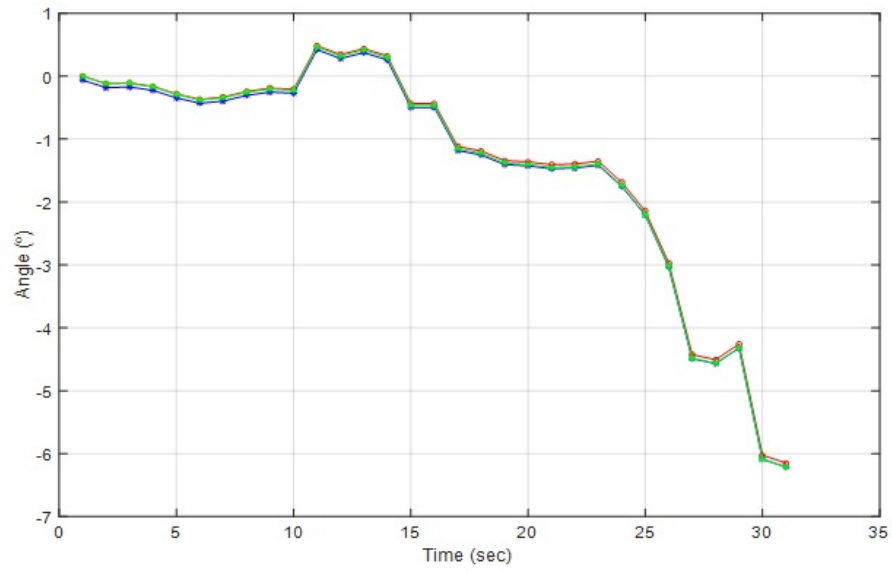


Fig. 5.26.: The heading of the UGV platform through different trajectories for the first segment

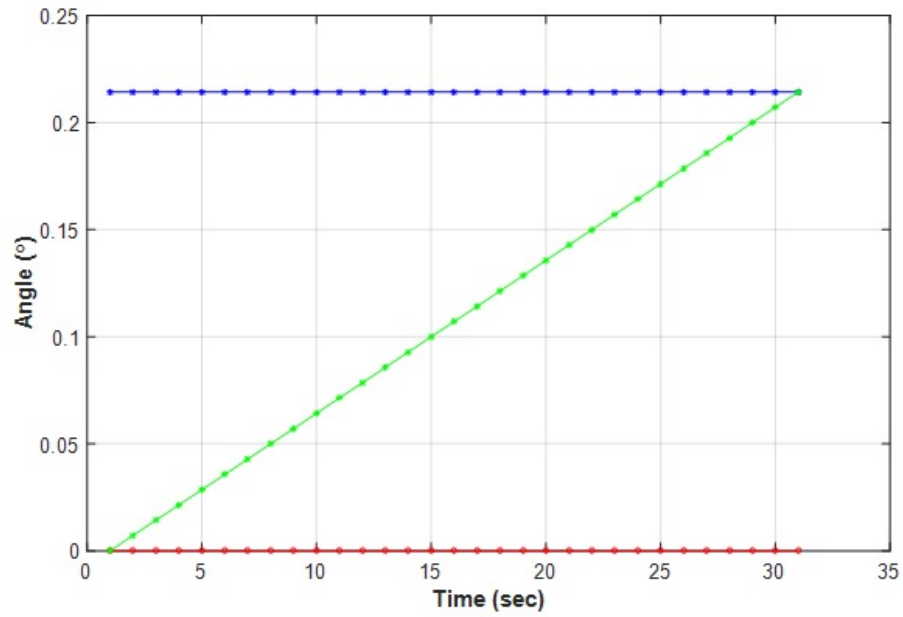


Fig. 5.27.: The pitch of the UGV platform through different trajectories for the first segment

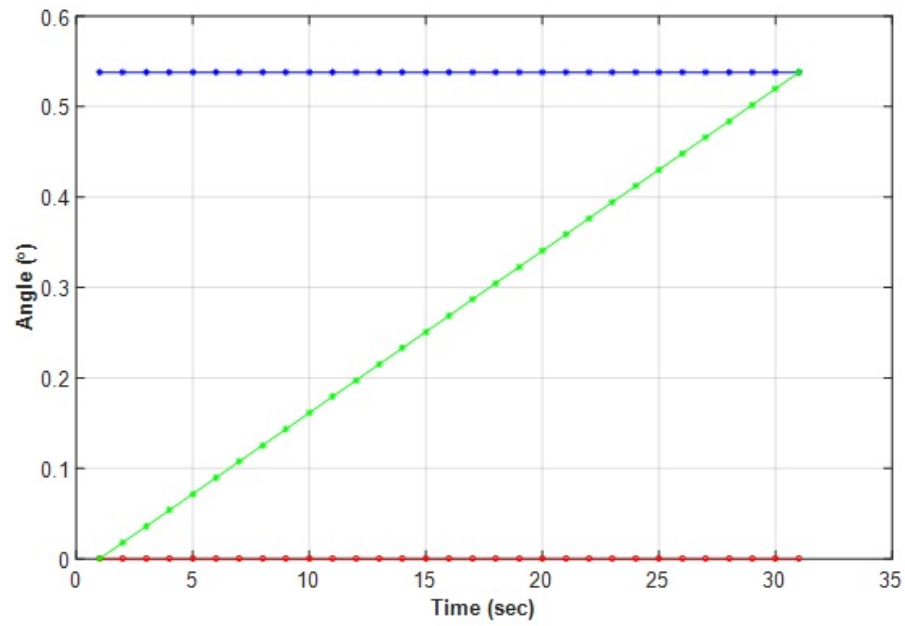


Fig. 5.28.: The roll of the UGV platform through different trajectories for the first segment

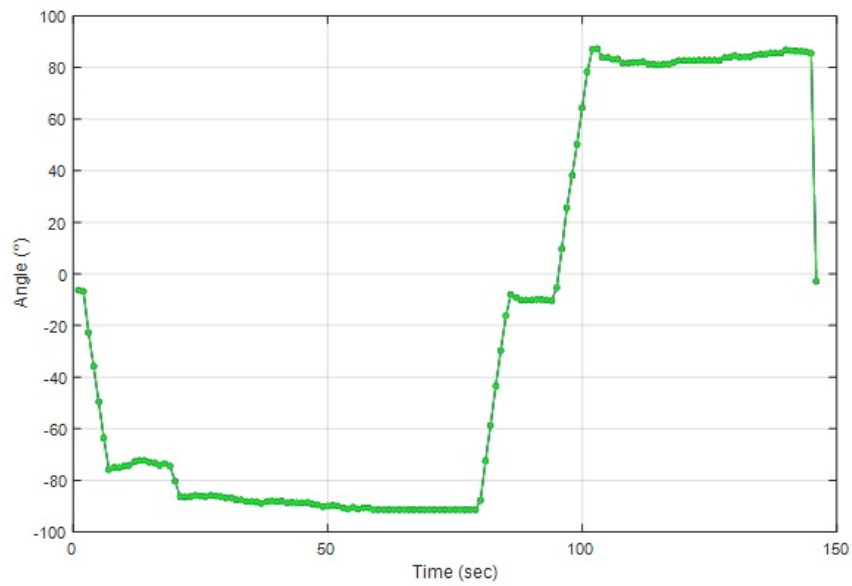


Fig. 5.29.: The heading of the UGV platform through different trajectories for the second segment

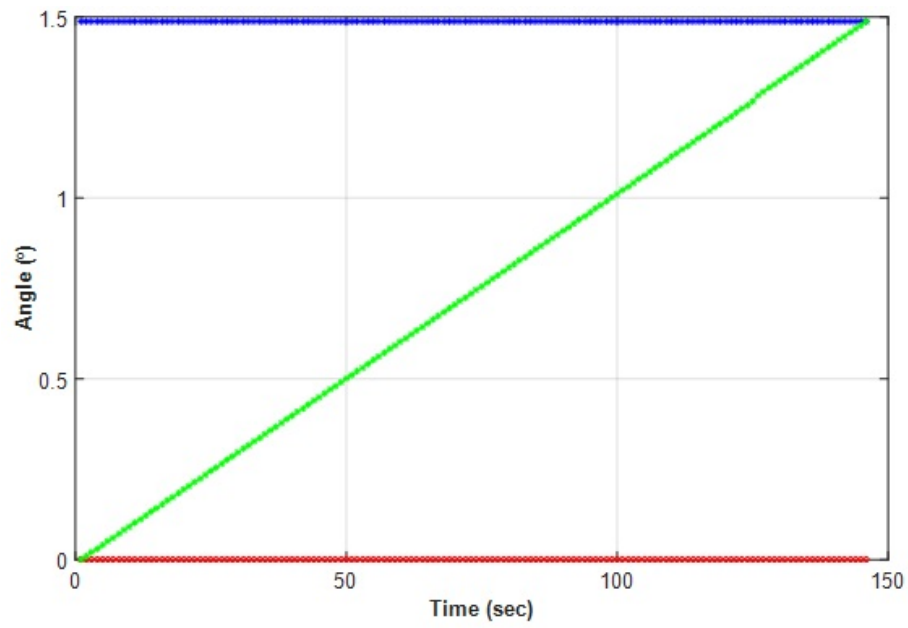


Fig. 5.30.: The pitch of the UGV platform through different trajectories for the second segment

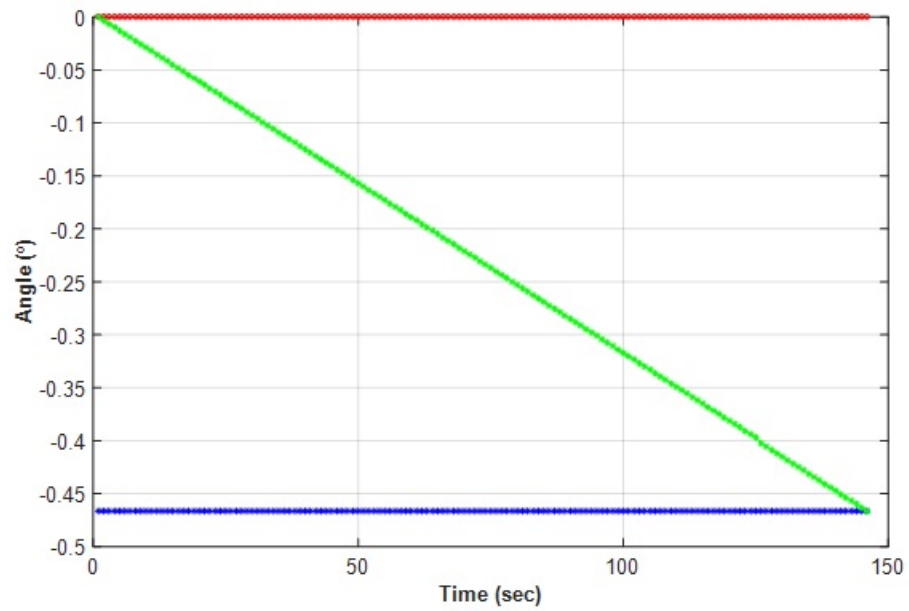


Fig. 5.31.: The roll of the UGV platform through different trajectories for the second segment

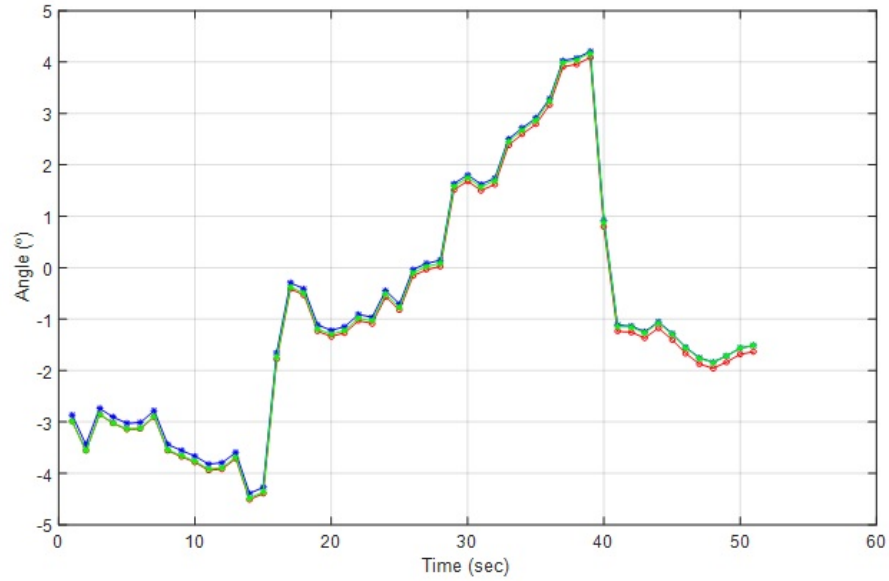


Fig. 5.32.: The heading of the UGV platform through different trajectories for the third segment

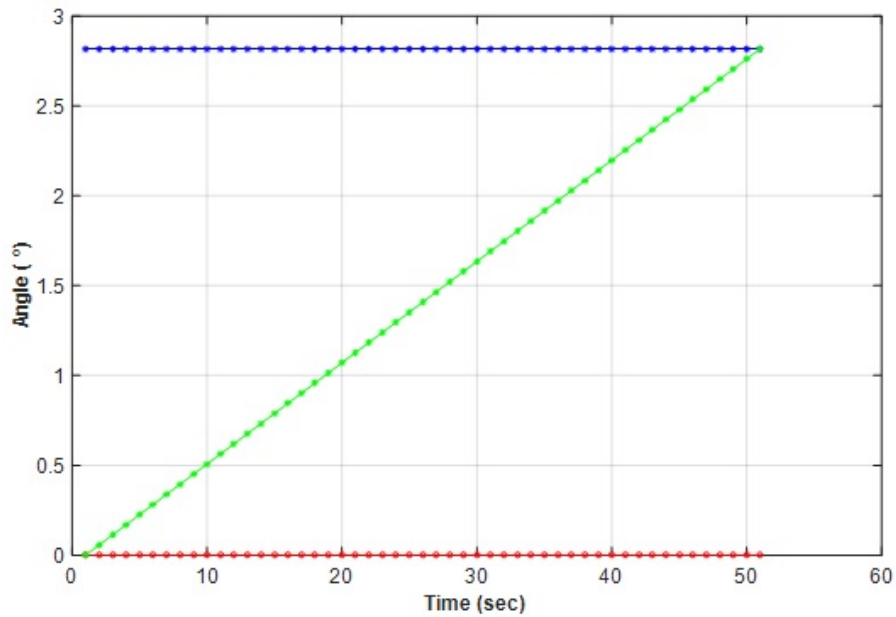


Fig. 5.33.: The pitch of the UGV platform through different trajectories for the third segment

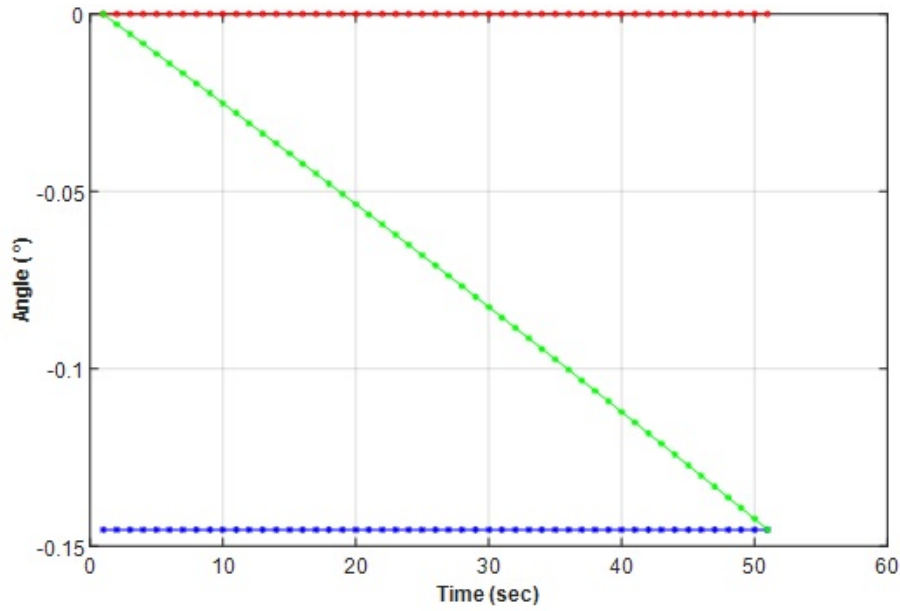


Fig. 5.34.: The roll of the UGV platform through different trajectories for the third segment

For closer investigation, the difference in x-y-directions as well as the pitch, roll, and heading angles between the smoothed trajectory and the forward trajectory (red) and backward trajectory (blue) for the three different segments of location II are depicted from Fig. 5.35 until Fig. 5.49. One should note that the intersection point (a) between the two curves denotes the maximum error at the middle time of the trajectory and the point (b) represents the zero-smoothed error at the trusted station (i.e., the stationary scan location at the beginning of the forward trajectory). Furthermore, the point (c) denotes the zero-smoothed error at the trusted station (i.e., the stationary scan location at the beginning of the backward trajectory).



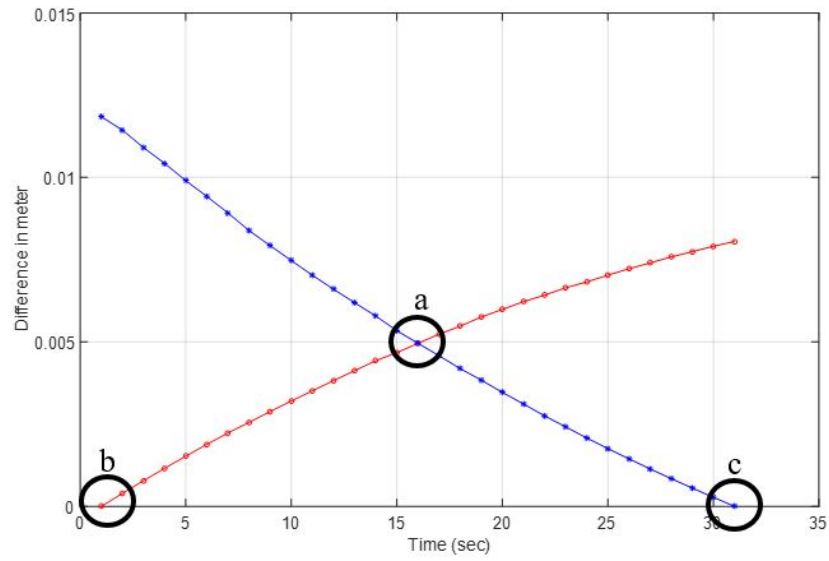


Fig. 5.35.: The difference in x-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the first segment

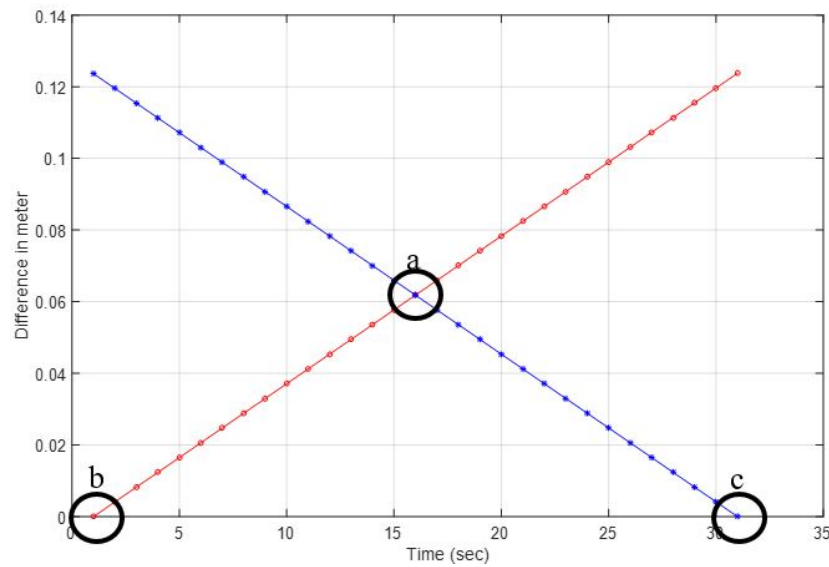


Fig. 5.36.: The difference in y-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the first segment

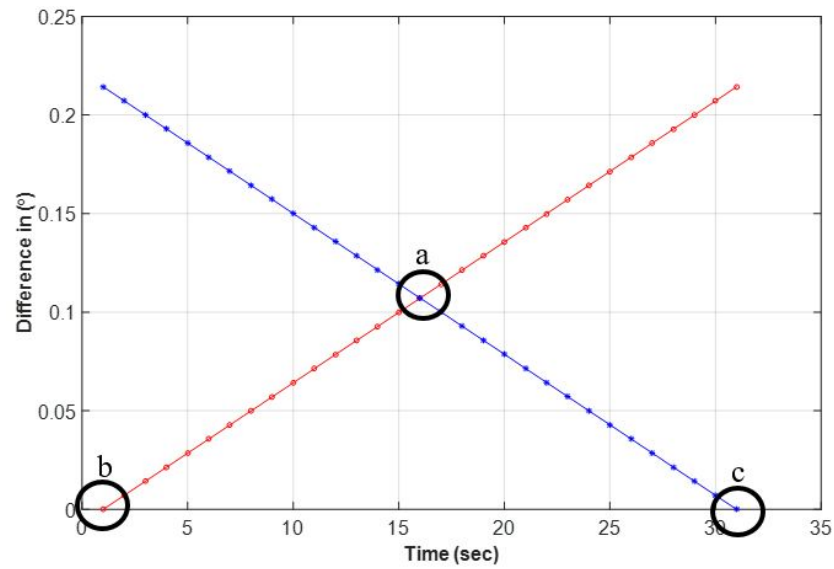


Fig. 5.37.: The difference in the pitch angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the first segment

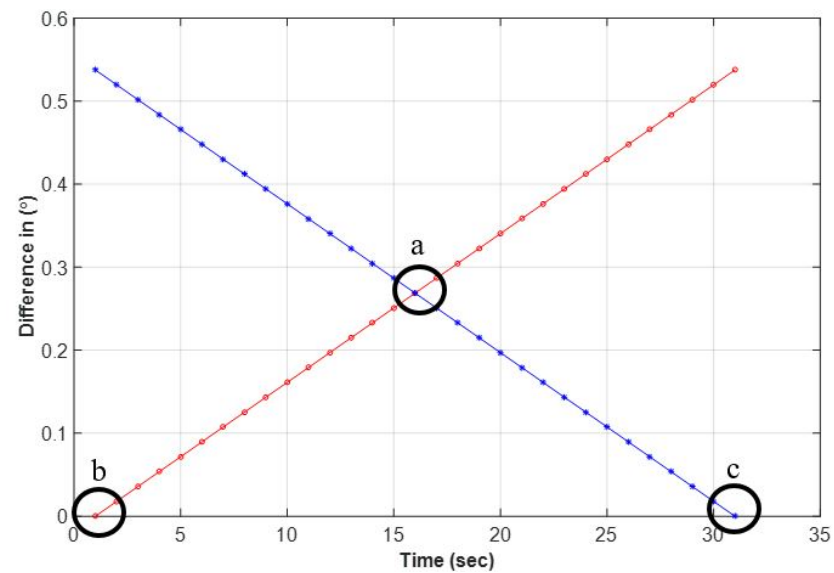


Fig. 5.38.: The difference in the roll angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the first segment

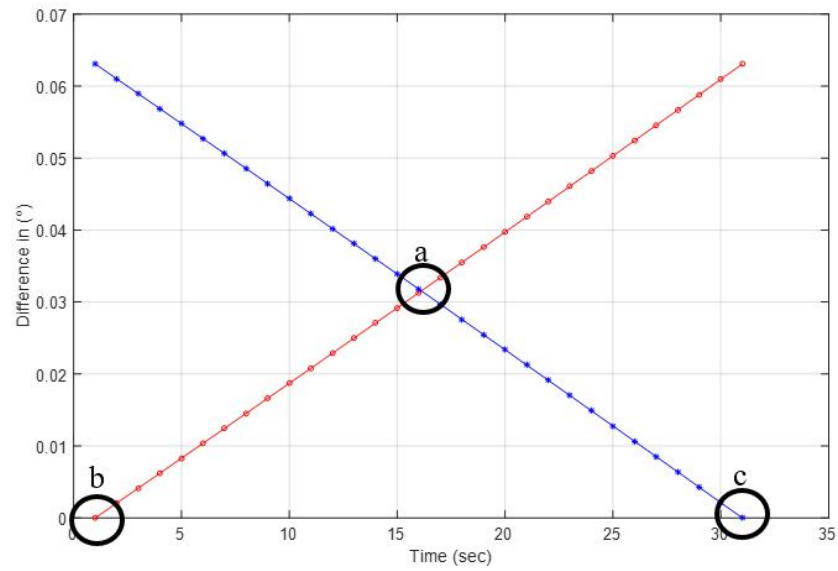


Fig. 5.39.: The difference in the heading angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the first segment

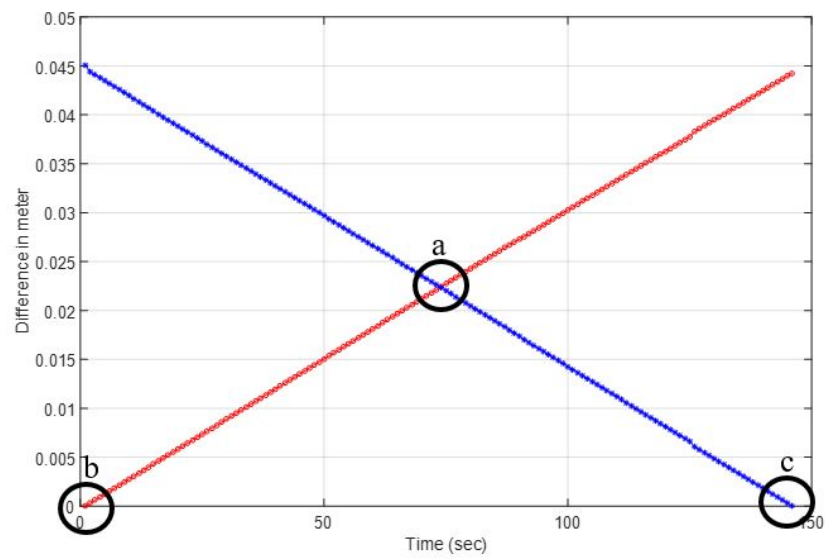


Fig. 5.40.: The difference in x-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the second segment

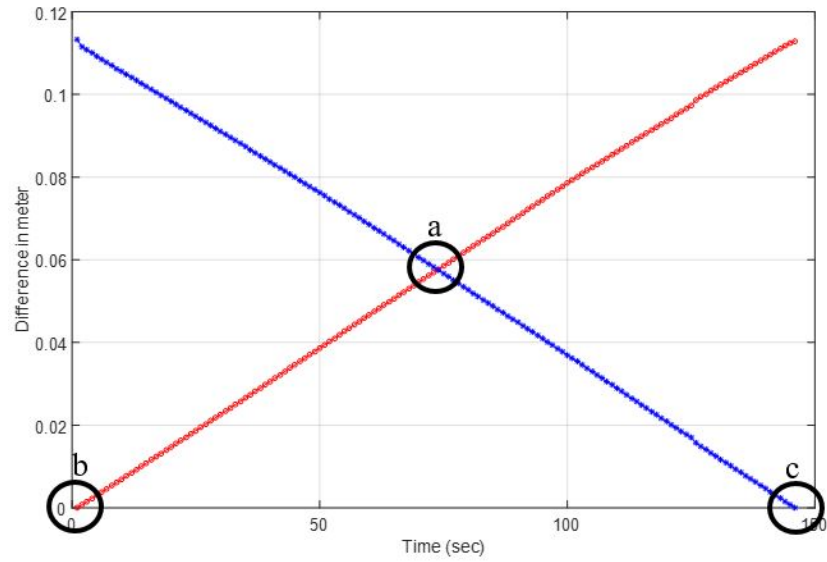


Fig. 5.41.: The difference in y-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the second segment

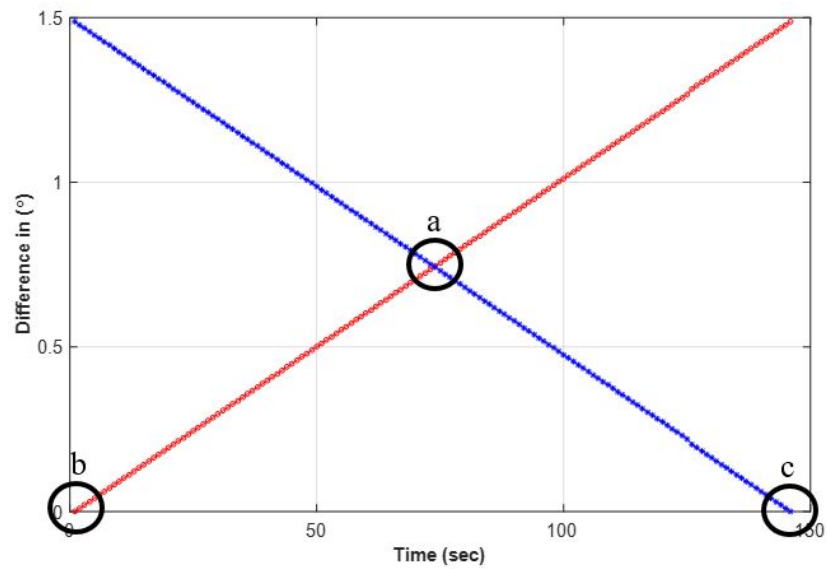


Fig. 5.42.: The difference in the pitch angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the second segment

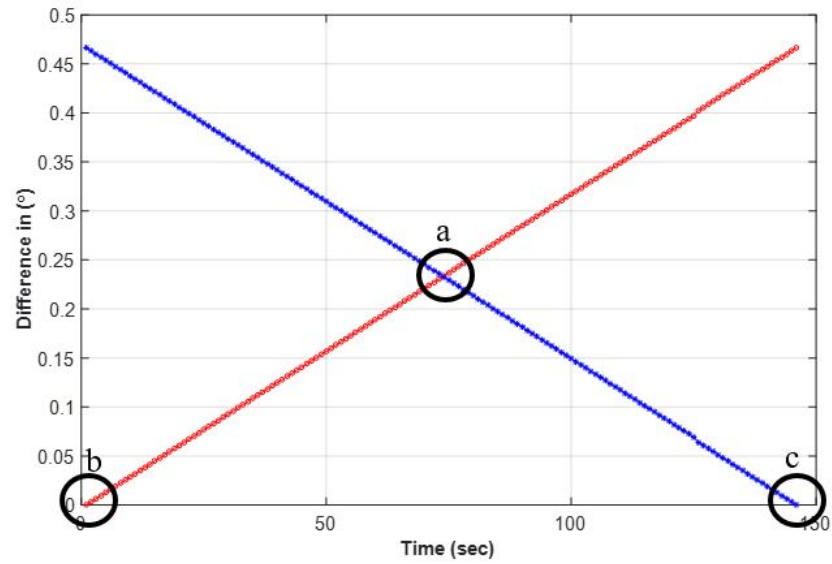


Fig. 5.43.: The difference in the roll angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the second segment

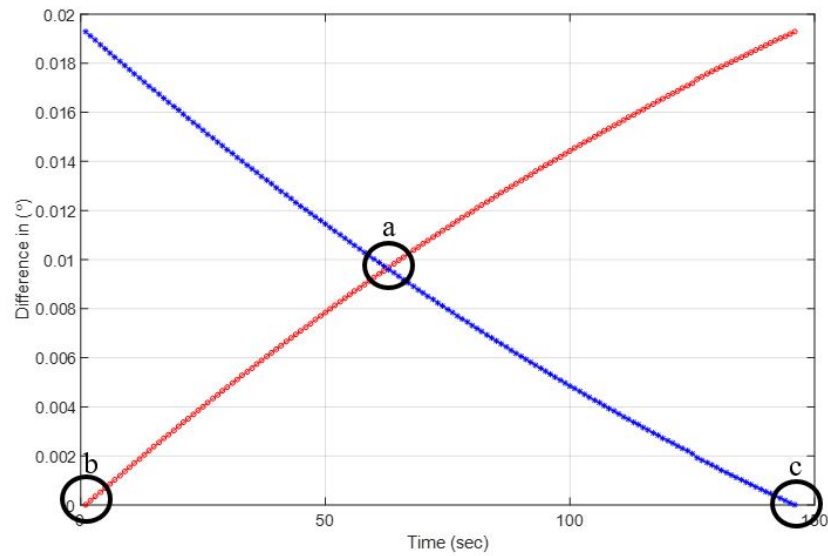


Fig. 5.44.: The difference in the heading angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the second segment

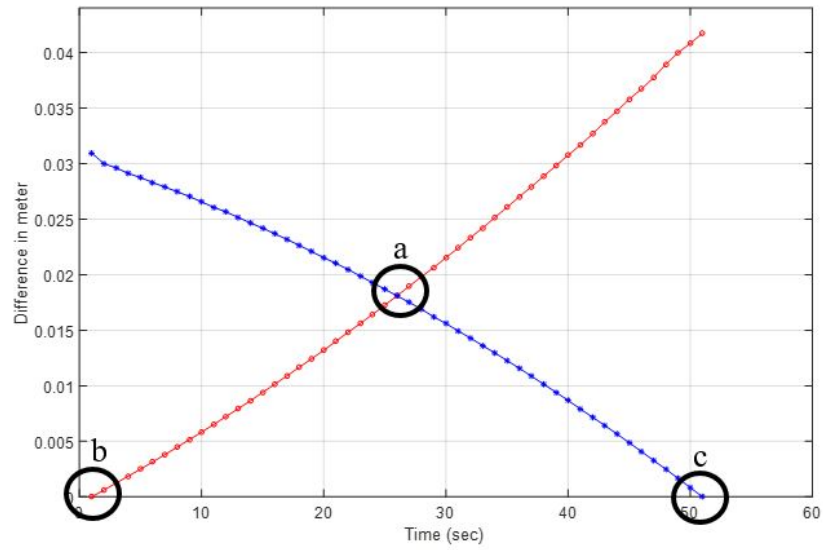


Fig. 5.45.: The difference in x-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the third segment

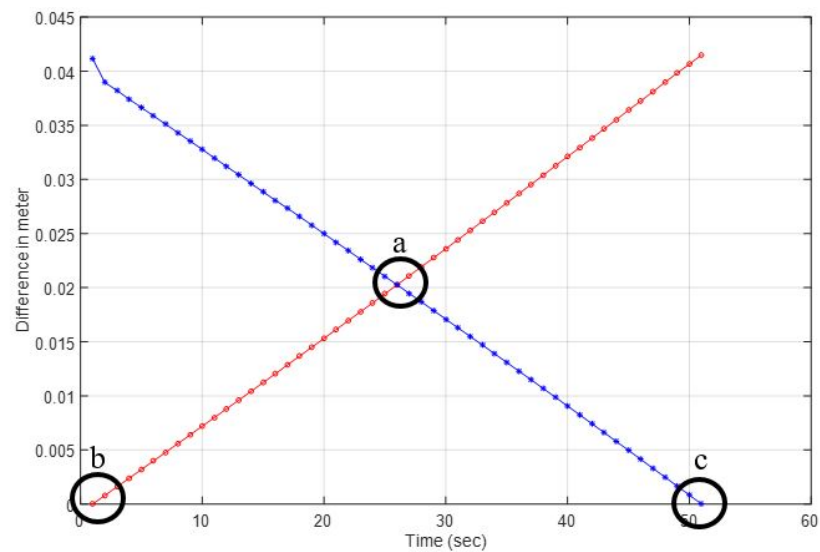


Fig. 5.46.: The difference in y-direction between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the third segment

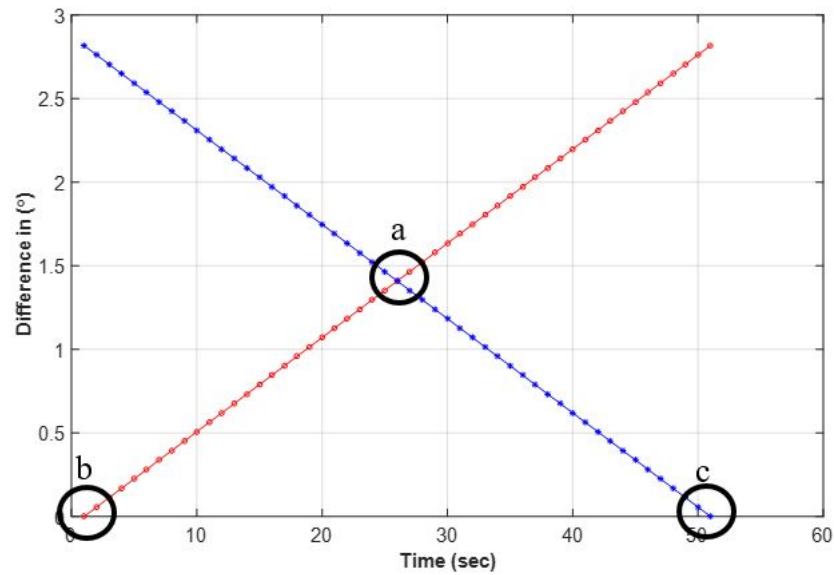


Fig. 5.47.: The difference in the pitch angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the third segment

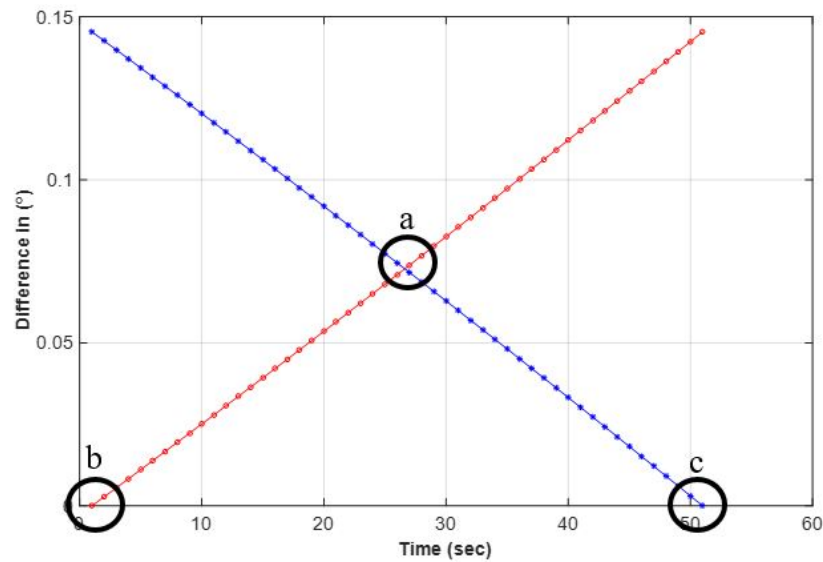


Fig. 5.48.: The difference in the roll angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the third segment

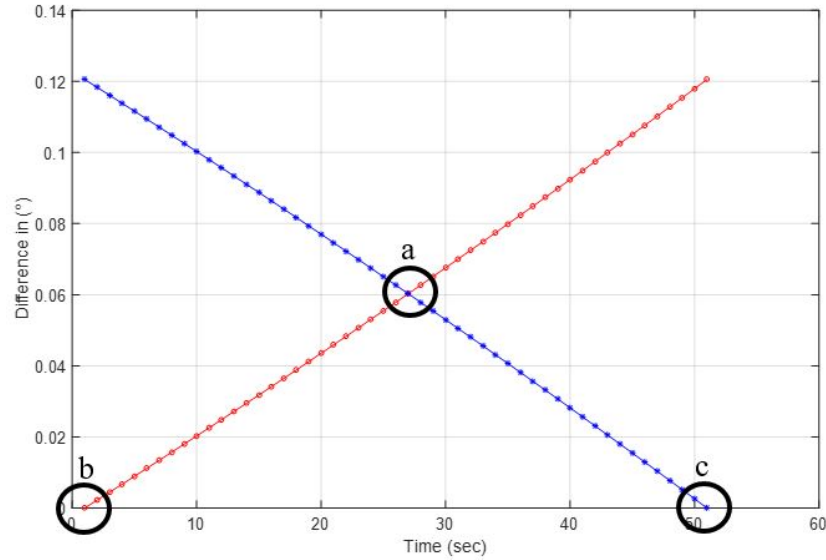


Fig. 5.49.: The difference in the heading angle between the smoothed trajectory and the forward trajectory (red) as well as the backward trajectory (blue) for the third segment

In this part of the research, both qualitative and quantitative evaluation are introduced to evaluate the performance of the proposed smoothing process. More specifically, qualitative analysis is performed by deriving a 3D-point cloud using the generated smoothed trajectory for the three different segments. Then, the results will be compared to the original 3D-point cloud generated by using the real-time SLAM-based trajectory. Based on the visual inspection, there is no significant improvement while considering the smoothed trajectory for the 3D-point cloud derivation of the three segments as depicted in Fig. 5.50 and Fig. 5.51. One should note that the derived map reconstruction while considering the ICP-based stationary scan locations has a significant enhancement for the 3D-point cloud reconstruction as shown in Fig. 5.52. For quantitative evaluation, a planar surface is extracted from three datasets. Such datasets are derived by utilizing three different ways (i.e., real-time SLAM trajectory, ICP-based stationary scans approach, and smoothed trajectory). One should note that the 3D-point cloud derived using ICP-based stationary scans



is considered the reference dataset. The RMSE of the normal distance of points belonging to a planar feature from its best-fitting plane are listed in Tables 5.2 - 5.5. Furthermore, the normal distance (ND) of the derived planar features using the real-time SLAM trajectory as well as the smoothed trajectory with respect to the reference dataset is reported in Tables 5.2 - 5.5. One can conclude that a LiDAR point cloud gets closer to the ICPP-based surface registration from stationary scan locations (accuracy) by using the smoothed trajectory. However, the internal noise level gets worse (precision).

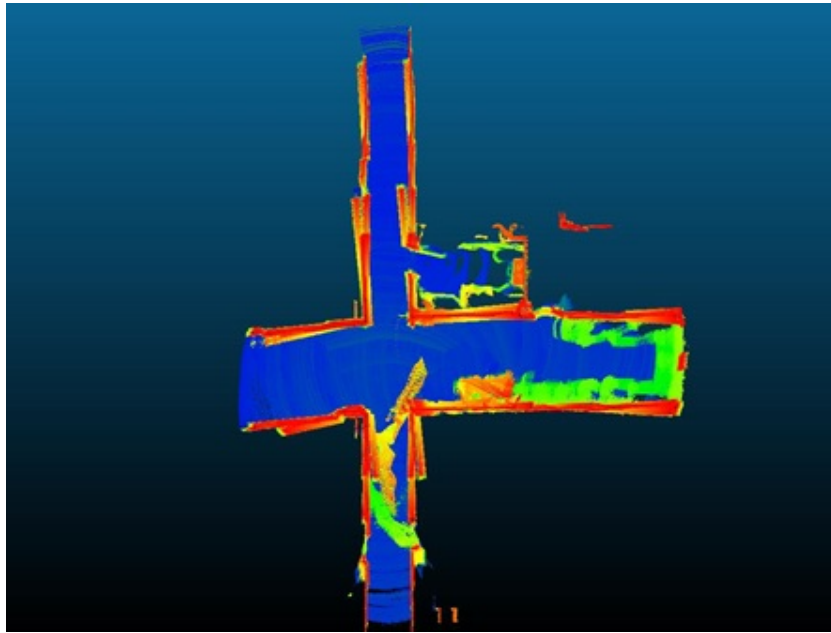


Fig. 5.50.: The 3D-point cloud reconstruction (only three segments of the entire trajectory) using the real-time SLAM-based trajectory

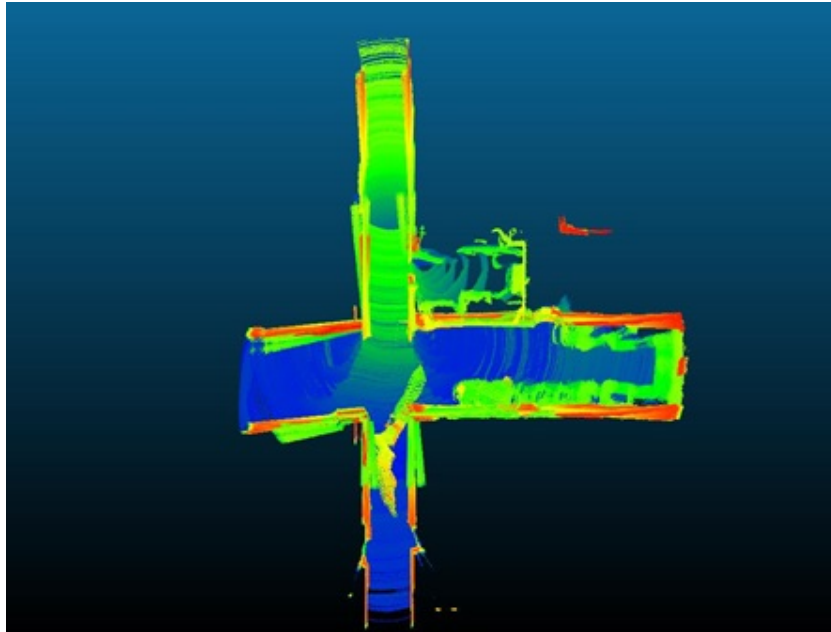


Fig. 5.51.: The 3D-point cloud reconstruction (only three segments of the entire trajectory) using the generated smoothed trajectory

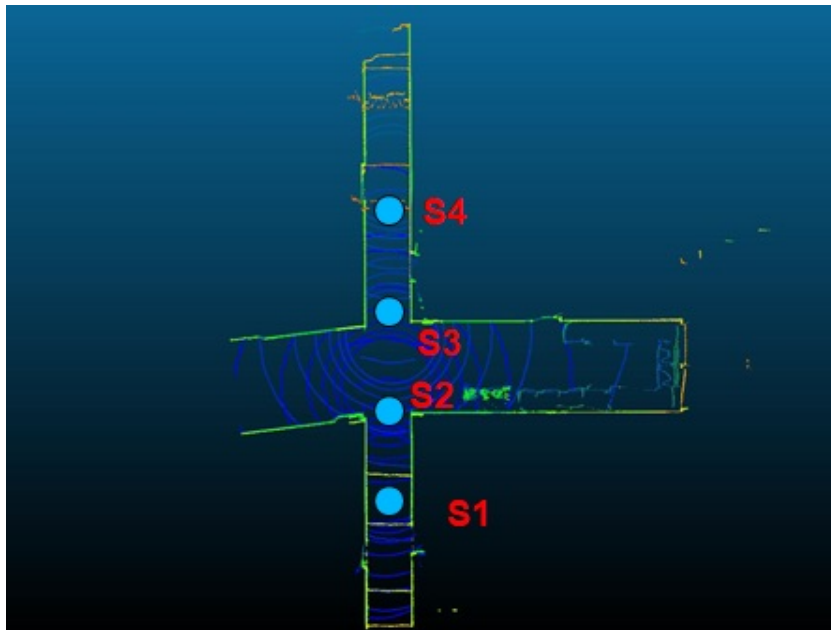


Fig. 5.52.: The 3D-point cloud reconstruction (only three segments of the entire trajectory) using the ICPP-based stationary scan locations

Table 5.2.: The quantitative analysis (RMSE and ND) of plane (1)

<b>Plane</b> (1)	<b>Number of points</b>	<b>RMSE (m)</b>	<b>ND (m)</b>
<b>Dataset(SLAM)</b>	1,329,822	0.031	0.114
<b>Dataset(ICPP)</b>	17,951	0.006	0
<b>Dataset(Smoothing)</b>	980,715	0.067	0.056

Table 5.3.: The quantitative analysis (RMSE and ND) of plane (2)

<b>Plane</b> (2)	<b>Number of points</b>	<b>RMSE (m)</b>	<b>ND (m)</b>
<b>Dataset(SLAM)</b>	1,820,926	0.019	0.112
<b>Dataset(ICPP)</b>	24,847	0.008	0
<b>Dataset(Smoothing)</b>	720,531	0.034	0.021

Table 5.4.: The quantitative analysis (RMSE and ND) of plane (3)

<b>Plane</b> (3)	<b>Number of points</b>	<b>RMSE (m)</b>	<b>ND (m)</b>
<b>Dataset(SLAM)</b>	1,012,652	0.033	0.121
<b>Dataset(ICPP)</b>	9,556	0.006	0
<b>Dataset(Smoothing)</b>	667,829	0.051	0.056

Table 5.5.: The quantitative analysis (RMSE and ND) of plane (4)

<b>Plane</b> (4)	<b>Number of points</b>	<b>RMSE (m)</b>	<b>ND (m)</b>
<b>Dataset(SLAM)</b>	637,995	0.063	0.112
<b>Dataset(ICPP)</b>	15,060	0.005	0
<b>Dataset(Smoothing)</b>	529,096	0.078	0.069

### 5.3.7 Summary

In this phase of research, a new hybrid system is considered for a LiDAR-based indoor mapping system which introduces a 2D-coverage path planning approach that is implemented along with online SLAM technique. The offline CPP can be altered for use with the aid of online SLAM by proposing two procedures: (i) perform a convex cellular decomposition of the polygonal coverage area while still tracing the shortest coverage path and (ii) apply the SLAM operation to suit the CPP strategy and evaluates the navigation errors in terms of an area coverage cost function. The implementation results show how the SLAM-assisted CPP strategies allow for an improvement in the total area coverage and perform a robust operation.

Furthermore, the implementation of an indoor MMS using a 3D-laser scanner onboard UGV has been considered for the task of generating high density maps of GNSS-denied environments. To mitigate the impact the absence of GNSS data has on the mapping process, this part of the research proposes a pseudo-GNSS/INS integrated framework which utilizes probabilistic SLAM techniques to estimate the platform pose and heading from 3D-laser scanner data. This proposed framework has been implemented based on three major notions: (i) utilizing frame transformation to extract 2D scan from the entire 3D-point cloud for efficiently carrying out real-time SLAM, (ii) generating the position data from these real-time SLAM pose estimates,

and (iii) performing the entire operation through use of a single 3D-mapping sensor. The final geo-referenced point cloud can then be derived through post-processing after applying the ICP registration procedure at the stationary scan locations along the entire trajectory. Also, a smoothing approach based on the registered stationary locations is proposed to enhance the real-time SLAM-based trajectory to improve the generated 3D-reconstruction of the mapping environment of interest. However, the generated smoothed trajectory has no significant improvement on the generated 3D-point cloud reconstruction. The implementation, performance, and results of the proposed MMS framework demonstrate the ability of this Pseudo-GNSS/INS framework to operate in GNSS-denied areas as well as to simply switch between the real GNSS/INS unit and Pseudo-GNSS/INS module for desired operation.

## 6. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

### 6.1 Summary of Contributions

This research focused on developing a generic framework, including system setup and calibration for UAVs equipped with a GNSS/INS positioning and orientation module as well as low-cost LiDAR sensors for targeted mapping and monitoring applications. Furthermore, a Pseudo GNSS/INS integrated framework onboard a UGV is developed to allow for operation within GNSS-denied environments. Several strategies are proposed to establish such framework. First, a system architecture for a low-cost UAV mapping system using directly geo-referenced active ranging systems was developed.

In addition, a LiDAR system calibration strategy for a UAV-based MMS that can directly estimate the mounting parameters was proposed through an outdoor calibration procedure. Finally, a Pseudo-GNSS/INS module that serves as a convenient substitute to a GNSS/INS-based MMS framework for the purpose of operation within GNSS-denied environments was developed. The contributions of each proposed strategies are summarized as follows:

#### System Architecture For UAV-Based Mapping System

The proposed system architecture for UAV mapping system has several contributions:

- A system architecture for low-cost UAV mapping using directly geo-referenced active ranging optical systems is developed while considering the challenges

posed by using consumer-grade sensors, platform payload restrictions, and endurance capabilities.

- The system integration has been developed in a way which demonstrates the flexibility of the proposed work. More specifically, the developed system integration can be used for not only airborne mapping systems but also for other systems such as the wheel-based LiDAR systems, with similar mapping sensor structure and techniques.

### **LiDAR System Calibration Strategy For A UAV-Based MMS**

The main contributions of the proposed LiDAR system calibration for a UAV-based MMS can be summarized as follows:

- A comprehensive bias impact analysis is conducted for a UAV-based LiDAR system consisting of a spinning multi-beam laser scanner.
- Based on this analysis, an optimal target primitive setup and flight line configuration was devised for calibrating a UAV LiDAR system.
- An iterative calibration strategy is proposed for deriving the system parameters using different types of conjugate features (i.e., planar, linear/cylindrical) at the same time.

### **SLAM-Assisted CPP And Implementation Of Pseudo-GNSS/INS Framework**

The proposed SLAM-assisted CPP and the Pseudo-GNSS/INS framework have the following contributions:

- A new hybrid system is developed for a LiDAR-based indoor mapping system that introduced a 2D-CPP problem implemented along with real-time SLAM technique.

- A Pseudo-GNSS/INS framework is developed that demonstrates an effective technique for implementing a flexible MMS framework that can be deployed into GNSS-denied as well as GNSS-affluent areas without exhaustive modifications to the system.
- To act like a GNSS/INS module, the Pseudo-GNSS/INS can supply geo-referencing signals (which includes PPS signal and GPRMC message) to successfully synchronize the LiDAR sensor for ensuring time-tagged point cloud generation which is considered an innovative implementation.
- The positioning module and the mapping sensor in the proposed framework get incorporated into a single unit comprised of single 3D- mapping sensor. This is considered a very important feature of the Pseudo-GNSS/INS-based MMS framework.

## 6.2 Recommendations for Future Work

Recommendations/suggestions for future work related to the proposed strategies are presented as follows:

- The system architecture for UAV- based mapping system will be focusing on the system integration of high-quality mapping sensors to meet the needs of particular mapping applications such as pipeline inspection and infrastructure inventory and monitoring. One should note that high-end mapping sensors can ensure the desired quality of the final product to satisfy the requirements of such applications.
- A LiDAR system calibration will focus on combining the mounting parameters (i.e., extrinsic parameters) and sensor parameters (i.e., intrinsic parameters) to obtain a comprehensive calibration leading to even more accurate point clouds. The obtained LiDAR-based 3D point cloud can be combined with information from other sensors, such as RGB cameras and hyperspectral sensors, to



extract valuable information related to different applications. Furthermore, a fully automated procedure should be developed for the extraction of calibration primitives.

- The Pseudo-GNSS/INS framework implementation will be focusing on improving the proposed smoothing approach with the aid of Kalman filter implementation to enhance the real-time SLAM trajectory generated by such framework. Also, the incorporation of cameras with such framework can provide an improvement for the trajectory estimation of indoor environment. Moreover, the future work will focus on performing the 3D-SLAM algorithm in order to provide a robust trajectory estimation.

## REFERENCES

## REFERENCES

- [1] G. Jozkow, C. Toth, and D. Grejner-Brzezinska, "Uas topographic mapping with velodyne lidar sensor," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, p. 201, 2016.
- [2] U. Niethammer, M. James, S. Rothmund, J. Travelletti, and M. Joswig, "Uav-based remote sensing of the super-sauze landslide: Evaluation and results," *Engineering Geology*, vol. 128, pp. 2–11, 2012.
- [3] K. Anderson and K. J. Gaston, "Lightweight unmanned aerial vehicles will revolutionize spatial ecology," *Frontiers in Ecology and the Environment*, vol. 11, no. 3, pp. 138–146, 2013.
- [4] F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, and D. Sarazzi, "Uav photogrammetry for mapping and 3d modeling—current status and future perspectives," *International archives of the photogrammetry, remote sensing and spatial information sciences*, vol. 38, no. 1, p. C22, 2011.
- [5] A. Fritz, T. Kattenborn, and B. Koch, "Uav-based photogrammetric point clouds—tree stem mapping in open stands in comparison to terrestrial laser scanner point clouds," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 40, pp. 141–146, 2013.
- [6] A. Habib, K. I. Bang, A. P. Kersting, and J. Chow, "Alternative methodologies for lidar system calibration," *Remote Sensing*, vol. 2, no. 3, pp. 874–907, 2010.
- [7] F. Ackermann, "Airborne laser scanning—present status and future expectations," *ISPRS Journal of Photogrammetry and Remote sensing*, vol. 54, no. 2-3, pp. 64–67, 1999.
- [8] I. A. Hameed, "Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3-4, pp. 965–983, 2014.
- [9] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 718–724, 2004.
- [10] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [11] R. N. De Carvalho, H. Vidal, P. Vieira, and M. Ribeiro, "Complete coverage path planning and guidance for cleaning robots," in *Industrial Electronics, 1997. ISIE'97., Proceedings of the IEEE International Symposium on*, vol. 2. IEEE, 1997, pp. 677–682.

- [12] S. Yun, Y. J. Lee, and S. Sung, "Imu/vision/lidar integrated navigation system in gnss denied environments," in *Aerospace Conference, 2013 IEEE*. IEEE, 2013, pp. 1–10.
- [13] L. Busemeyer, D. Mentrup, K. Möller, E. Wunder, K. Alheit, V. Hahn, H. P. Maurer, J. C. Reif, T. Würschum, J. Müller *et al.*, "Breedvision—a multi-sensor platform for non-destructive field-based phenotyping in plant breeding," *Sensors*, vol. 13, no. 3, pp. 2830–2847, 2013.
- [14] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 92, pp. 79–97, 2014.
- [15] J. Everaerts *et al.*, "The use of unmanned aerial vehicles (uavs) for remote sensing and mapping," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, no. 2008, pp. 1187–1192, 2008.
- [16] S. Nebiker, A. Annen, M. Scherrer, and D. Oesch, "A light-weight multispectral sensor for micro uav—opportunities for very high resolution airborne remote sensing," *The international archives of the photogrammetry, remote sensing and spatial information sciences*, vol. 37, no. B1, pp. 1193–1199, 2008.
- [17] F. Nex and F. Remondino, "Uav for 3d mapping applications: a review," *Applied geomatics*, vol. 6, no. 1, pp. 1–15, 2014.
- [18] N. Heinze, M. Esswein, W. Krüger, and G. Saur, "Image exploitation algorithms for reconnaissance and surveillance with uav," in *Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications VII*, vol. 7668. International Society for Optics and Photonics, 2010, p. 76680U.
- [19] A. W. N. Ibrahim, P. W. Ching, G. G. Seet, W. M. Lau, and W. Czajewski, "Moving objects detection and tracking framework for uav-based surveillance," in *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*. IEEE, 2010, pp. 456–461.
- [20] Inspire, *Series2*, 2017. [Online]. Available: <https://www.dji.com/inspire-2/info>
- [21] SenseFly, *eBee*, 2017. [Online]. Available: <https://www.sensefly.com/drone/ebec-mapping-drone/>
- [22] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, "Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use," *Remote Sensing*, vol. 4, no. 6, pp. 1671–1692, 2012.
- [23] S. G. Gupta, M. M. Ghonge, and P. Jawandhiya, "Review of unmanned aircraft system (uas)," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 4, pp. pp-1646, 2013.
- [24] A. Kukko, *GIM*, 2016. [Online]. Available: <https://www.gim-international.com/content/article/the-current-state-of-the-art-in-uas-based-laser-scanning>
- [25] Faro, *Scanner*, 2007. [Online]. Available: <https://www.faro.com/products/construction-bim-cim/faro-focus/>

- [26] H. Geosystems, *Leica*, 2007. [Online]. Available: <https://leica-geosystems.com/en-us/products/laser-scanners>
- [27] RIEGL, *RIEGL*, 2007. [Online]. Available: <http://www.riegl.com/nc/products/airborne-scanning/>
- [28] C. Vaughn, J. Button, W. Krabill, and D. Rabine, “Georeferencing of airborne laser altimeter measurements,” *International Journal of Remote Sensing*, vol. 17, no. 11, pp. 2185–2200, 1996.
- [29] E. P. Baltsavias, “Airborne laser scanning: existing systems and firms and other resources,” *ISPRS Journal of Photogrammetry and Remote sensing*, vol. 54, no. 2-3, pp. 164–198, 1999.
- [30] N. El-Sheimy, C. Valeo, and A. Habib, *Digital Terrain Modeling: Acquisition, Manipulation and Applications (Artech House Remote Sensing Library)*. Norwood, MA: Artech House, 2005.
- [31] A. Habib, Y. Lin, and Z. Lari, “Geometric processing: Active sensor modeling and calibration (lidar),” 2018.
- [32] Velodyne, *VLP-16*, 2017. [Online]. Available: <http://velodynelidar.com/vlp-16.html>
- [33] ———, *VLP-32C*, 2018. [Online]. Available: <http://velodynelidar.com/vlp-32c.html>
- [34] PSOMAS, *PSOMAS Mobile Mapper System*, 2017. [Online]. Available: <https://psomas.com/expertise/services/surveying-mapping/>
- [35] RIEGL, *RIEGL Laser Measurement Systems*, 2017. [Online]. Available: <http://www.riegl.com/>
- [36] L. Wallace, A. Lucieer, C. Watson, and D. Turner, “Development of a uav-lidar system with application to forest inventory,” *Remote Sensing*, vol. 4, no. 6, pp. 1519–1543, 2012.
- [37] L. Wallace, A. Lucieer, and C. S. Watson, “Evaluating tree detection and segmentation routines on very high resolution uav lidar data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7619–7628, 2014.
- [38] M. Nagai, T. Chen, R. Shibasaki, H. Kumagai, and A. Ahmed, “Uav-borne 3-d mapping system by multisensor integration,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 701–708, 2009.
- [39] Y. Lin, J. Hyypä, and A. Jaakkola, “Mini-uav-borne lidar for fine-scale mapping,” *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 3, pp. 426–430, 2011.
- [40] Q. Guo, Y. Su, T. Hu, X. Zhao, F. Wu, Y. Li, J. Liu, L. Chen, G. Xu, G. Lin *et al.*, “An integrated uav-borne lidar system for 3d habitat mapping in three forest ecosystems across china,” *International journal of remote sensing*, vol. 38, no. 8-10, pp. 2954–2972, 2017.

- [41] B. Yang and C. Chen, "Automatic registration of uav-borne sequent images and lidar data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 101, pp. 262–274, 2015.
- [42] M. Hirose, Y. Xiao, Z. Zuo, V. R. Kamat, D. Zekkos, and J. Lynch, "Implementation of uav localization methods for a mobile post-earthquake monitoring system," in *Environmental, Energy and Structural Monitoring Systems (EESMS), 2015 IEEE Workshop on*. IEEE, 2015, pp. 66–71.
- [43] H. M. Tulldahl and H. Larsson, "Lidar on small uav for 3d mapping," in *Electro-Optical Remote Sensing, Photonic Technologies, and Applications VIII; and Military Applications in Hyperspectral Imaging and High Spatial Resolution Sensing II*, vol. 9250. International Society for Optics and Photonics, 2014, p. 925009.
- [44] GEOZONE, *LiDAR Scanfly*, 2016. [Online]. Available: <http://www.geozone.com/index.php/en/lidar-scanfly>
- [45] HYPACK, *NEXUS-800*, 2016. [Online]. Available: <http://www.hypack.com/products/NEXUS800>
- [46] RIEGL, *miniVUX-1UAV*, 2017. [Online]. Available: <http://www.riegl.com/products/unmanned-scanning/riegl-minivux-1uav/>
- [47] Routsene, *LidarPod*, 2017. [Online]. Available: <https://www.routsene.com/the-3d-mapping-solution/uav-lidar-system/>
- [48] SITECOINFORMATICA, *Sky-Scanner System*, 2017. [Online]. Available: <http://www.sitecoinf.it/en/solutions/sky-scanner>
- [49] YellowScan, *YellowScan Surveyor*, 2016. [Online]. Available: <http://www.yellowscan.fr/products/yellowscan-surveyor>
- [50] P. Kumari, *A curvature based method for systematic errors adjustment in airborne laser scanning data*. University of Florida, 2011.
- [51] J. Kilian, N. Haala, M. Englich *et al.*, "Capture and evaluation of airborne laser scanner data," *International Archives of Photogrammetry and Remote Sensing*, vol. 31, pp. 383–388, 1996.
- [52] M. Crombags, R. Brügemann, and E. J. de Min, "On the adjustment of overlapping strips of laser altimeter height data," *International Archives of Photogrammetry and Remote Sensing*, vol. 33, no. B3/1, pp. 230–237, 2000.
- [53] H.-G. Maas, "Methods for measuring height and planimetry discrepancies in airborne laserscanner data," *Photogrammetric Engineering and Remote Sensing*, vol. 68, no. 9, pp. 933–940, 2002.
- [54] S. Filin and G. Vosselman, "Adjustment of airborne laser altimetry strips," 2004.
- [55] H. Kager, "Discrepancies between overlapping laser scanner strips—simultaneous fitting of aerial laser scanner strips," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, no. B1, pp. 555–560, 2004.

- [56] A. Habib, K. I. Bang, A. P. Kersting, and D.-C. Lee, "Error budget of lidar systems and quality control of the derived data," *Photogrammetric Engineering & Remote Sensing*, vol. 75, no. 9, pp. 1093–1108, 2009.
- [57] J. Skaloud and D. Lichti, "Rigorous approach to bore-sight self-calibration in airborne laser scanning," *ISPRS journal of photogrammetry and remote sensing*, vol. 61, no. 1, pp. 47–59, 2006.
- [58] S. Filin, "Calibration of airborne and spaceborne laser altimeters using natural surfaces," Ph.D. dissertation, The Ohio State University, 2001.
- [59] P. Friess, "Toward a rigorous methodology for airborne laser mapping," *Proceedings EuroCOW*, pp. 25–27, 2006.
- [60] S. Lindenthal, V. Ussyshkin, J. Wang, and M. Pokorny, "Airborne lidar: A fully-automated self-calibration procedure," *ISPRS Achieves, ISPRS Calgary 2011 Workshop*, pp. 29–31, 2011.
- [61] H. Burman, "Calibration and orientation of airborne image and laser scanner data using gps and ins," Ph.D. dissertation, Institutionen för geodesi och fotogrammetri, 2000.
- [62] K. W. Morin, "Calibration of airborne laser scanners," *University of Calgary, Calgary, Canada*, 2002.
- [63] C. Toth, "Calibrating airborne lidar systems," 2002.
- [64] K. I. Bang, "Alternative methodologies for lidar system calibration," Ph.D. dissertation, University of Calgary, 2010.
- [65] A. P. Kersting, A. Habib, K.-I. Bang, and J. Skaloud, "Automated approach for rigorous light detection and ranging system calibration without preprocessing and strict terrain coverage requirements," *Optical Engineering*, vol. 51, no. 7, p. 076201, 2012.
- [66] G. Atanacio-Jiménez, J.-J. González-Barbosa, J. B. Hurtado-Ramos, F. J. Ornelas-Rodríguez, H. Jiménez-Hernández, T. García-Ramírez, and R. González-Barbosa, "Lidar velodyne hdl-64e calibration using pattern planes," *International Journal of Advanced Robotic Systems*, vol. 8, no. 5, p. 59, 2011.
- [67] C. Glennie, B. Brooks, T. Ericksen, D. Hauser, K. Hudnut, J. Foster, and J. Avery, "Compact multipurpose mobile laser scanning system—initial tests and results," *Remote Sensing*, vol. 5, no. 2, pp. 521–538, 2013.
- [68] C. Glennie, A. Kusari, and A. Facchin, "Calibration and stability analysis of the vlp-16 laser scanner," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2016.
- [69] B. Schwarz, "Lidar: Mapping the world in 3d," *Nature Photonics*, vol. 4, no. 7, p. 429, 2010.
- [70] J. Choi, "Hybrid map-based slam using a velodyne laser scanner," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 3082–3087.

- [71] B. Vallet, W. Xiao, and M. Brédif, “Extracting mobile objects in images using a velodyne lidar point cloud,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, no. 3, p. 247, 2015.
- [72] J. Underwood, A. Hill, and S. Scheduling, “Calibration of range sensor pose on mobile platforms,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 3866–3871.
- [73] N. Muhammad and S. Lacroix, “Calibration of a rotating multi-beam lidar,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 5648–5653.
- [74] M. He, H. Zhao, F. Davoine, J. Cui, and H. Zha, “Pairwise lidar calibration using multi-type 3d geometric features in natural scene,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1828–1835.
- [75] T. O. Chan and D. D. Lichti, “Feature-based self-calibration of velodyne hdl-32e lidar for terrestrial mobile mapping applications,” in *The 8th International Symposium on Mobile Mapping Technology, Tainan, Taiwan*, 2013, pp. 1–3.
- [76] A. F. Habib, K. I. Bang, S.-W. Shin, and E. Mitishita, “Lidar system self-calibration using planar patches from photogrammetric data,” in *The 5th Int. Symp. Mobile Mapping Technology*, 2007.
- [77] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.
- [78] H. Choset, “Coverage for robotics—a survey of recent results,” *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.
- [79] A. Ntawumenyikizaba, H. H. Viet, and T. Chung, “An online complete coverage algorithm for cleaning robots based on boustrophedon motions and a\* search,” in *Information Science and Digital Content Technology (ICIDT), 2012 8th International Conference on*, vol. 2. IEEE, 2012, pp. 401–405.
- [80] H. H. Viet, V.-H. Dang, S. Choi, and T. C. Chung, “Bob: an online coverage approach for multi-robot systems,” *Applied Intelligence*, vol. 42, no. 2, pp. 157–173, 2015.
- [81] E. Acar, Y. Zhang, H. Choset, M. Schervish, A. G. Costa, R. Melamud, D. C. Lean, and A. Graveline, “Path planning for robotic demining and development of a test platform,” in *International Conference on Field and Service Robotics*, vol. 1, 2001, pp. 161–168.
- [82] Y. Li, H. Chen, M. J. Er, and X. Wang, “Coverage path planning for uavs based on enhanced exact cellular decomposition method,” *Mechatronics*, vol. 21, no. 5, pp. 876–885, 2011.
- [83] J. Jin and L. Tang, “Optimal coverage path planning for arable farming on 2d surfaces,” *Transactions of the ASABE*, vol. 53, no. 1, pp. 283–295, 2010.
- [84] Y. Liu, X. Lin, and S. Zhu, “Combined coverage path planning for autonomous cleaning robots in unstructured environments,” in *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*. IEEE, 2008, pp. 8271–8276.



- [85] F. Yasutomi, M. Yamada, and K. Tsukamoto, "Cleaning robot control," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on.* IEEE, 1988, pp. 1839–1841.
- [86] S. Hert, S. Tiwari, and V. Lumelsky, "A terrain-covering algorithm for an auv," in *Underwater Robots.* Springer, 1996, pp. 17–45.
- [87] M. Ollis and A. Stentz, "Vision-based perception for an automated harvester," in *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, vol. 3. IEEE, 1997, pp. 1838–1844.
- [88] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of robot motion: theory, algorithms, and implementation.* MIT press, 2005.
- [89] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and service robotics.* Springer, 1998, pp. 203–209.
- [90] W. H. Huang, "The minimal sum of altitudes decomposition for coverage algorithms," *Rensselaer Polytechnic Institute Computer Science Technical Report*, vol. 6, 2000.
- [91] M. Mazo Jr and K. H. Johansson, "Robust area coverage using hybrid control," in *TELEC, Santiago de Cuba, Cuba*, 2004, pp. 1–8.
- [92] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.
- [93] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *ICRA*, 2003, pp. 1985–1991.
- [94] A. Kim and R. M. Eustice, "Perception-driven navigation: Active visual slam for robotic area coverage," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* IEEE, 2013, pp. 3196–3203.
- [95] E. Galceran, R. Campos, N. Palomeras, M. Carreras, and P. Ridao, "Coverage path planning with realtime replanning for inspection of 3d underwater structures," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on.* IEEE, 2014, pp. 6586–6591.
- [96] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics.* Springer, 2008, pp. 871–889.
- [97] J. Artieda, J. M. Sebastian, P. Campoy, J. F. Correa, I. F. Mondragón, C. Martínez, and M. Olivares, "Visual 3-d slam from uavs," *Journal of Intelligent and Robotic Systems*, vol. 55, no. 4-5, p. 299, 2009.
- [98] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Vision-based odometry and slam for medium and high altitude flying uavs," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1-3, pp. 137–161, 2009.
- [99] D. Ribas, P. Ridao, J. D. Tardós, and J. Neira, "Underwater slam in man-made structured environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 898–921, 2008.

- [100] J. Zhang and S. Singh, “Loam: Lidar odometry and mapping in real-time.” in *Robotics: Science and Systems*, vol. 2, 2014.
- [101] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [102] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.
- [103] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An evaluation of the rgb-d slam system,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1691–1696.
- [104] N. Corso and A. Zakhori, “Indoor localization algorithms for an ambulatory human operated 3d mobile mapping system,” *Remote Sensing*, vol. 5, no. 12, pp. 6611–6646, 2013.
- [105] C. Brenner, “Vehicle localization using landmarks obtained by a lidar mobile mapping system,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences: [PCV 2010-Photogrammetric Computer Vision And Image Analysis, Pt I] 38 (2010), Nr. Part 3A*, vol. 38, no. Part 3A, pp. 139–144, 2010.
- [106] H. Zhao, M. Chiba, R. Shibasaki, X. Shao, J. Cui, and H. Zha, “Slam in a dynamic large outdoor environment using a laser scanner,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 1455–1462.
- [107] F. Keller and H. Sternberg, “Multi-sensor platform for indoor mobile mapping: system calibration and using a total station for indoor applications,” *Remote sensing*, vol. 5, no. 11, pp. 5805–5824, 2013.
- [108] C. Wen, L. Qin, Q. Zhu, C. Wang, and J. Li, “Three-dimensional indoor mobile mapping with fusion of two-dimensional laser scanner and rgb-d camera data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 4, pp. 843–847, 2014.
- [109] K. M. A. Yousef, J. Park, and A. C. Kak, “An approach-path independent framework for place recognition and mobile robot localization in interior hallways,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2669–2676.
- [110] H. Kwon, K. M. A. Yousef, and A. C. Kak, “Building 3d visual maps of interior space with a new hierarchical sensor fusion architecture,” *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 749–767, 2013.
- [111] K. A. Yousef, “Hypothesize-and-verify based solutions for place recognition and mobile robot self-localization in interior hallways,” Ph.D. dissertation, Purdue University, 2013.
- [112] A. Nüchter, D. Borrmann, P. Koch, M. Kühn, and S. May, “A man-portable, imu-free mobile mapping system.” *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 2, 2015.

- [113] DJI, *S1000+*, 2016. [Online]. Available: <https://www.dji.com/spreading-wings-s1000-plus>
- [114] —, *M600 Pro*, 2017. [Online]. Available: <https://www.dji.com/matrice600-pro>
- [115] Trimble, *APX-15 UAV*, 2017. [Online]. Available: <https://www.applanix.com/products/dg-uavs.htm>
- [116] Novatel, *IGM-S1*, 2016. [Online]. Available: <https://www.applanix.com/products/dg-uavs.htm>
- [117] Velodyne, *HDL-32E*, 2016. [Online]. Available: <http://velodynelidar.com/404.html>
- [118] Wireshark, *Ethernet*, 2017. [Online]. Available: <https://www.wireshark.org/>
- [119] Raspberry-Pi, *Embedded-System*, 2016. [Online]. Available: <https://www.raspberrypi.org/>
- [120] A. Jafarnia Jahromi, “Gnss signal authenticity verification in the presence of structural interference,” Ph.D. dissertation, University of Calgary, 2013.
- [121] Trimble, *POSLV-125*, 2016. [Online]. Available: <https://www.applanix.com/products/poslv.htm>
- [122] Novatel, *Span-CPT*, 2015. [Online]. Available: <https://www.novatel.com/products/span-gnss-inertial-systems/span-combined-systems/span-cpt/>
- [123] H. Fourati and D. E. C. Belkhiat, *Multisensor Attitude Estimation: Fundamental Concepts and Applications*. CRC Press, 2016.
- [124] A. F. Habib, A. P. Kersting, A. Shaker, and W.-Y. Yan, “Geometric calibration and radiometric correction of lidar data and their impact on the quality of derived products,” *Sensors*, vol. 11, no. 9, pp. 9069–9097, 2011.
- [125] E. M. Mikhail and F. E. Ackermann, “Observations and least squares,” 1976.
- [126] E. Renaudin, A. Habib, and A. P. Kersting, “Featured-based registration of terrestrial laser scans with minimum overlap using photogrammetric data,” *Etri Journal*, vol. 33, no. 4, pp. 517–527, 2011.
- [127] A. Habib, J. Lay, and C. Wong, “Specifications for the quality assurance and quality control of lidar systems,” *Submitted to the Base Mapping and Geomatic Services of British Columbia*, 2006.
- [128] A. Saalfeld, “Topologically consistent line simplification with the douglas-peucker algorithm,” *Cartography and Geographic Information Science*, vol. 26, no. 1, pp. 7–18, 1999.
- [129] G. Mei, J. C. Tipper, and N. Xu, “Ear-clipping based algorithms of generating high-quality polygon triangulation,” in *Proceedings of the 2012 International Conference on Information Technology and Software Engineering*. Springer, 2013, pp. 979–988.

- [130] M. Montemerlo and S. Thrun, *FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics*. Springer, 2007, vol. 27.
- [131] A. Narkhede and D. Manocha, “Fast polygon triangulation based on seidel’s algorithm,” in *Graphics Gems V*. Elsevier, 1995, pp. 394–397.
- [132] F. Morbidi, R. Cano, and D. Lara, “Minimum-energy path generation for a quadrotor uav,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1492–1498.
- [133] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (slam): Part ii,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [134] G. Grisetti, C. Stachniss, and W. Burgard, “Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling,” in *ICRA*, 2005, pp. 2432–2437.
- [135] M. Al-Durgham, I. Datchev, and A. Habib, “Analysis of two triangle-based multi-surface registration algorithms of irregular point clouds,” *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3812, pp. 61–66, 2011.

VITA

## VITA

Tamer Shamseldin received the B.S. and M.S degrees in Electrical Engineering from Military Technical College (MTC), EGYPT, in 2004 and 2012, respectively. He joined the Ph.D. program in Interdisciplinary Program "Geo-Electrical" between Geomatics at the Lyles school of Civil Engineering and Electrical and Computer Engineering at Purdue University, West Lafayette, IN, USA., in August 2015. His current research deals with the development of the system architecture for a LiDAR-based Unmanned Aerial Vehicles (UAVs) and terrestrial mapping system, single and multi-LiDAR system calibration, SLAM-assisted coverage path planning for indoor LiDAR mapping system, and the implementation of Pseudo-GNSS/INS Localization System for Indoor LiDAR MMS.