HAL
open science

Laser based sensor fusion and control for the tele-operation of minidrones

Carlos Viña

To cite this version:

Carlos Viña. Laser based sensor fusion and control for the tele-operation of minidrones. Robotics [cs.RO]. Université Pierre et Marie Curie - Paris VI, 2017. English. NNT : 2017PA066525 . tel-01882687

HAL Id: tel-01882687

https://theses.hal.science/tel-01882687v1

Submitted on 27 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Pierre et Marie Curie
ED 391 : Sciences mécaniques, acoustique, électronique & robotique
Institut des Systèmes Intelligents et de Robotique (ISIR)

Thesis

Laser-based sensor fusion and control for the tele-operation of mini-drones

Ph.D thesis prepared at ISIR by
Carlos Viña

supervised by
Pascal Morin

Thesis committee members:

Mme.	Isabelle Fantoni	Reviewer
M.	Simon Lacroix	Reviewer
M.	Bruno Gas	Chairman
M.	Julien Marzat	Examiner
M.	Vincent Frémont	Examiner

October 24, 2017

Abstract

Aerial robotics is a prominent field of research that has seen great commercial success during the last years. This has been driven by technological advances, such as batteries with increased power storage, accurate light-weight sensing and higher on-board processing capabilities, that have allowed the development of highly efficient and affordable small-sized airborne platforms, commonly referred to as mini-drones, miniature unmanned aerial vehicles (mini-UAVs) or micro aerial vehicles (MAVs). This has opened the way to promising new applications in surveillance and inspection tasks, which have seen an increasing demand for MAV platforms with improved autonomous capabilities. In recent years, this has been a key subject of research in the power industry, where power utilities are subject to deterioration due to atmospheric conditions, and require extensive monitoring programs. In this sense, aerial surveys, based on remote sensing mounted on large piloted aircraft, have provided a mean of covering extensive areas in relatively short periods of time. Building on this, MAVs have the potential of fully automating the inspection process, further reducing costs and inspection times. In this context, this thesis addresses autonomous electric tower inspections with MAVs. Namely, self-localization, the first step in a long series of tasks towards achieving fully autonomous capabilities, is the main focus of this work. We explore how 2D laser scanners, which have become one of the most popular remote sensing technologies in mobile robot navigation, can be coupled with commonly available sensors to obtain real-time estimates of a MAV's 6 degree of freedom pose, using uniquely on-board sensing and processing capabilities. Numerous topics are developed in this thesis, from classic scan matching algorithms, such as the iterative closest point (ICP) algorithm and proposed adaptations to the electric tower scene, to sensor fusion and feed-back control, whose designs used throughout this work are briefly introduced in the final chapter. Validations based on experimental flights and extensive simulations are presented.

Contents

Contents i

List of Figures iv

Introduction 1

1 Related works on MAV state estimation and autonomous navigation with laser scanners 6

1.1 Overview . . . 6

1.2 Pose tracking . . . 7

1.2.1 2D laser odometry . . . 8

1.2.1.1 Deterministic approach . . . 8

1.2.1.2 Probabilistic approach . . . 10

1.2.2 Height estimation on MAVs . . . 12

1.2.2.1 Laser-based height estimation . . . 12

1.2.2.2 Barometer-based height estimation . . . 13

1.2.3 Attitude estimation on MAVs . . . 13

1.3 Sensor fusion . . . 14

1.3.1 Fusing relative state measurements . . . 14

Stochastic Cloning Kalman filter (SC-KF). . . . 15

1.4 Towards a completely autonomous MAV . . . 15

1.4.1 Global localization . . . 16

Monte Carlo localization (MCL). . . . 17

1.4.2 Simultaneous localization and mapping (SLAM) . . . 17

1.4.2.1 Particle filter based SLAM . . . 18

GMapping. . . . 18

1.4.2.2 Graph-based SLAM . . . 19

Graph construction. . . . 19

Graph optimization. 20
1.4.3 Fusing relative and absolute state measurements 21
1.4.3.1 Recent developments on MAV multi-sensor fusion 22
Filtering techniques. 22
Smoothing techniques. 23
1.4.4 Achieving high-level navigation tasks 23
Obstacle avoidance. 23
Path planning. 24
Exploration. 24
1.5 Conclusions 25

2 Laser-based pose estimation for planar environments 26
2.1 Overview 26
2.2 The planar environment assumption 27
2.3 The Iterative Closest Point (ICP) algorithm 28
2.3.1 The Metric-based Iterative Closest Point (MbICP) algorithm 30
2.3.2 Limitations of ICP-based approaches for state estimation 32
2.4 Proposed pose tracking approach 32
2.4.1 2D laser odometry 32
2.4.2 Attitude estimation 34
2.4.3 Height estimation 36
2.5 Experimental results 36
2.6 Conclusions 42

3 Laser-based pose estimation for electric tower inspection: Feature-based approach 44
3.1 Overview 44
3.2 Characteristics of the laser range measurements on an electric tower 45
3.2.1 Notable features 47
3.3 Tracking the cross-sections of the electric tower 48
3.3.1 Scan segmentation 49
3.3.1.1 Extracting the front line segment 50
3.3.1.2 Extracting the lateral sides 50
3.3.2 Geometric fitting 50
3.3.3 Calculating the position and orientation 52
3.4 Results 53
3.4.1 Simulation results 53

3.4.1.1 First case: flight in front of the tower. 54
3.4.1.2 Second case: flight around the tower 56
3.4.2 Experimental results 58
3.4.2.1 Modelling the electric tower. 58
3.4.3 Limitations 59
3.5 Conclusions 61

4 Laser-based pose estimation for electric tower inspection: Model-based approach 62
4.1 Overview 62
4.2 Adapting the ICP algorithm to the electric tower scene 63
4.2.1 First proposed approach: using the tower point cloud reconstruction 64
4.2.2 Second proposed approach: using the tower planar representation . 65
4.2.3 Limitations 66
4.3 Proposed pose tracking approach 66
4.3.1 Laser odometry 67
4.3.2 Height estimation 67
4.3.2.1 First stage: Vertical velocity observer 68
4.3.2.2 Second stage: Fusing barometer and laser odometry measurements 68
4.3.3 Attitude estimation 70
4.3.3.1 Gain scheduling 70
4.4 Simulation results 71
4.4.1 Attitude estimation results 72
4.4.2 Laser odometry results 73
4.4.2.1 Laser odometry without the altitude observer 75
4.4.2.2 Laser odometry with the aid of the altitude observer . . . 77
4.4.3 Height estimation results 80
4.4.3.1 Vertical velocity estimation 80
4.4.3.2 Fusing barometer and laser odometry measurements . . . 80
4.5 Conclusions 82

5 MAV sensor fusion and feedback control 84
5.1 Overview 84
5.2 MAV rigid body dynamics 85
5.3 Sensor fusion 86
5.3.1 Rotation subsystem 86

5.3.2 Translation subsystem 87
5.4 Control 88
5.4.1 Attitude control 88
5.4.2 Position control 90
5.5 Results 90
5.5.1 Indoor scene experimental results 91
5.5.1.1 Sensor fusion 92
5.5.1.2 Feedback control 93
5.5.2 Electric tower scene simulation results 94
5.5.2.1 Sensor fusion 95
5.5.2.2 Feedback control 97
5.6 Conclusions 101

Conclusions and future work **102**

Appendices **105**

A Stability analysis and gain tuning of the altitude observer **107**

Bibliography **111**

List of Figures

1 A common high voltage transmission line. 3

2.1 Determining the search region for the correspondence search. 34

2.2 The indoor experimental setup. **(a)** Quadrotor developed at ISIR, equipped with a Hokuyo URG-30LX 2D LiDAR, an MPU6000 3 axis accelerometer/gyrometer unit and an SF10/A laser altimeter from Lightware Optoelectronics. **(b)** The quadrotor follows a set of waypoints relying exclusively on the on-board estimates and position controller. 37

2.3 A sample of the observed laser scans obtained from the indoor flight shown in Fig. 2.2b. The current laser scan (in red) is aligned to the reference scan (in blue), recovering transformed laser scan (in green). 38

2.4 For an indoor test flight, following a set of waypoints (as in Fig. 2.2b): **(a-c)** Comparing the 2D laser odometry estimates with the ground truth. **(d-f)** Absolute estimation with respect to the ground truth. 40

2.5 Roll and pitch angle estimation results, fusing accelerometer and gyrometer measurements with the attitude observer from Eq. (2.18). **(a-b)** Comparing the attitude estimates against the MoCap ground truth. **(c-d)** The absolute estimation errors with respect to the ground truth. 41

2.6 Height estimation results using the laser altimeter measurements and a second-order low-pass filter, as described in Sec. 2.4.3, for an indoors flight. **(a)** Comparing the altitude estimates against the MoCap ground truth. **(b)** Absolute estimation errors with respect to the ground truth. 41

3.1 The quadrotor from Fig. 2.2a was manually flown in front of an electric tower from a 60 kV distribution line, to register the laser range measurements. 45

3.2 Laser range measurements acquired on the tower from Fig. 3.1: **(a)** All sides are visible (the best case scenario). **(b)-(c)** Occlusions sometimes block the lateral and backsides from view. **(d)** Only the front side is visible (the worst case scenario). This happens when horizontal bars on the tower block the lateral and back sides from view. 46
3.3 Based on observations from Fig. 3.2: **(a)** Notable cross-section features captured in the 2D laser scans. **(b)** Proposed parametrization of the electric tower's cross-section. 48
3.4 Illustrating the proposed scan segmentation process: **(a)** Detecting the front side. **(b)** Detecting the left and right sides. 49
3.5 The simulation setup in the Gazebo simulation environment. A CAD model of an electric tower was used, whose dimensions roughly correspond to those of the tower from Fig. 3.1. 53
3.6 The two simulated flights. The blue line indicates the trajectory followed by the quadrotor. An example of a tracked cross-section is visible on the laser scan (red points), with corresponding estimated front $\mathcal{F}$ and center frames $\mathcal{C}$. **(a)** First case study for a flight in front of the tower, at different heights. **(b)** Second case study for a flight around the tower, at a fixed height. 54
3.7 For a flight in front of the simulated electric tower (from Fig. 3.6a): **(a-c)** Comparing the tracked position $\boldsymbol{\xi}_{\mathcal{C}} = (x_{\mathcal{C}}, y_{\mathcal{C}})^{\intercal}$ and orientation $\psi_{\mathcal{C}}$ of the tower's center with the simulation ground truth. **(d-f)** The absolute tracking errors, with respect to the simulation ground truth. **(g)** The MAV's elevation with respect to the ground level, obtained from the ground truth. 55
3.8 Processing time (in ms) of the tracking method for the results shown in Fig. 3.7. The proposed tracking method processes laser scans at an average time of 0.3 ms. 56
3.9 For a flight around the simulated electric tower (from Fig. 3.6b). **(a-c)** Comparing the tracked position $\boldsymbol{\xi}_{\mathcal{C}} = (x_{\mathcal{C}}, y_{\mathcal{C}})^{\intercal}$ and orientation $\psi_{\mathcal{C}}$ of the tower's center with the simulation ground truth. **(d-f)** The absolute tracking errors, with respect to the simulation ground truth. The tracking method fails (starting at t=16 s) when the MAV transitions from one side of the tower to another. 57
3.10 The estimated depth and width as a function of the height for the electric tower from Fig. 3.1, fitted with straight lines. 59

3.11 Partial 3D point cloud reconstruction of the electric tower from Fig. 3.1, for a vertical flight in front of the tower. The laser scans are aligned using the tracked cross-section's center, the quadrotor's altitude (from the laser altimeter) and attitude (from the IMU measurements). 60

4.1 An example of the proposed gain scheduling approach for the attitude observer from Eq. (2.18), according to Eq. (4.15), for different values of α. This parameter determines the steepness of the transitions between the nominal gains k_L and k_H. As $\tilde{a}_m$ increases, the gains k_γ decrease from k_L towards k_H. For higher values of α the transitions are faster. 70

4.2 The simulated flight around the tower. The blue line indicates the trajectory. Throughout the flight the quadrotor was oriented towards the center of the tower. 72

4.3 For a portion of the flight from Fig. 4.2: **(a)** The deviations of the accelerometer readings from gravity according to Eq. (4.14). **(b)** The resulting scheduled gains for different values of α. The gains become more reactive for larger values of α. 73

4.4 For the attitude observer from Eq. (2.18) and different values of α: **(a)** Absolute roll estimation error. **(b)** Absolute pitch estimation error. **(c)** The corresponding $\tilde{a}_m$. As α increases, the errors caused by peaks in $\tilde{a}_m$ are reduced. 74

4.5 The two models used in the simulations as references for the ICP algorithm to align the laser scans: **(a)** Point cloud reconstruction. **(b)** Planar model. 75

4.6 Laser odometry results for the simulated flight from Fig. 4.2. For both ICP implementations: **(a)** Comparing the laser odometry outputs with the ground truth. The point cloud approach fails before finishing the flight. **(b)-(e)** Absolute estimation errors with respect to the simulation ground truth. 76

4.7 Comparing the computation time for the laser scan registrations. The planar model approach (in red) is approximately ten times faster. The significant speed gains are the result of the projection-based matching strategy adopted in Sec. 4.2.2. 77

4.8 Comparing the laser odometry altitude estimates without the aid of the altitude observer (in red) from Fig. 4.6e and with the aid of the altitude observer (in blue). **(a)** In the planar model approach spurious peaks are reduced. **(b)** In the point cloud approach, large altitude errors are avoided (at $t = 30$ s) and the approach no longer fails. 78

4.9 Laser odometry results after introducing the altitude observer estimates in the scan registration initializations. For both ICP implementations: **(a)** Comparing the laser odometry outputs with the ground truth. In contrast to Fig. 4.6, the point cloud approach now finishes the flight successfully. **(b)-(e)** Absolute estimation errors with respect to the simulation ground truth. Both approaches now achieve similar accuracy. 79
4.10 Comparing the computation time for the scan registrations when using the altitude observer estimates in the initializations. There are no significant changes with respect to Fig. 4.7. The planar model approach (in red) remains significantly faster than the point cloud approach (in blue). 80
4.11 Vertical velocity estimates (Vz) obtained by fusing barometer and IMU measurements. **(a)** Without barometer drift. **(b)** With barometer drift. . . 81
4.12 The absolute altitude errors for ICP without the aid of the altitude observer, the barometer measurements with drift, and the altitude observer for: **(a)** ICP with planar model. **(b)** ICP with point cloud reconstruction. 82
4.13 For the altitude observer and both ICP implementations: (Top) Comparing the barometer drift estimates with the ground truth. (Bottom) Absolute estimation errors. The observer succeeds in both cases. 83

5.1 The 2D laser odometry and resulting sensor fusion outputs for an indoor flight from (see Fig. 2.2b). The MAV follows a set of pre-defined waypoints, at a fixed height of 1 m, using on-board processing and sensing. The ground truth shows the actual path followed by the MAV. 91
5.2 The results from fusing the 2D laser odometry with accelerometer readings (Eq. (5.10)) and gyrometer readings (Eq. (5.6)) for the flight from Fig. 5.1. **(left)** Comparing the laser odometry and sensor fusion outputs to the MoCap ground truth for the **(a)** x, **(b)** y and **(c)** ψ components. **(right)** The corresponding absolute estimation errors. The sensor fusion closely follows the laser odometry for all states. 92
5.3 The horizontal velocity estimation results from fusing laser odometry and accelerometer readings with the observer proposed in Eq. (5.10). **(a)** The estimated horizontal velocities follow the ground truth closely thanks to the choice of observer gains and high laser odometry rates (near 40 Hz). **(b)** The absolute estimation errors, where the larger peaks correspond directly to transitions between waypoints (when compared to Fig. 5.2a and Fig. 5.2b). 93

5.4 The resulting closed-loop control of $\{x, y, \psi\}$ with the MbICP laser odometry, for the duration of the flight from Fig. 5.1. **(a)** Time response of the control system for different desired locations. The ground truth pose from the MoCap system shows that all states converge to the desired location. **(b)** In all cases the tracking errors converge asymptotically towards zero. . 94

5.5 Testing the sensor fusion (Sec. 5.3) and feedback control (Sec. 5.4) designs in the electric tower scene. In separate simulations, the laser odometry relied on the ICP adaptations from Sec. 4.2, using **(a)** the tower point cloud reconstruction (see Fig. 4.5a) and **(b)** the planar tower representation (see Fig. 4.5b) for the scan registrations. In both simulations, the MAV follows a set of waypoints using state estimates from on-board sensing: 2D laser scanner, gyrometer, accelerometer and barometer. 95

5.6 The absolute estimation errors of **(a)** x, **(b)** y, **(c)** z and **(d)** ψ, for the simulated flights from Fig. 5.5. **(left)** Comparing the absolute errors of the fused state estimates to the ICP-based laser odometry with the point cloud model (from Fig. 5.5a). **(right)** Comparing the absolute errors of the fused state estimates to the ICP-based laser odometry with the planar tower model (from Fig. 5.5b). 96

5.7 The horizontal velocity estimates recovered from fusing the laser odometry and accelerometer readings (see Eq. (5.10)). **(a)** Comparing the V_x estimates for the point cloud approach (top), and the planar model approach (bottom). **(b)** Making the same comparison for V_y. In general, the velocity estimation errors for the planar case are smaller. However, both approaches converge rapidly towards the ground truth. 98

5.8 The resulting closed-loop control of $\{x, y, z, \psi\}$ for the laser odometry with the point cloud approach. **(a)** Time response of the control system for a 70 s portion of the simulated flight and different desired locations. The ground truth pose of the MAV shows that all states converge to the desired location. **(b)** The tracking errors converge asymptotically towards zero for all states. 99

5.9 The resulting closed-loop control of $\{x, y, z, \psi\}$ for the laser odometry with the planar representation approach. **(a)** Time response of the control system for a 70 s portion of the simulated flight and different desired locations. The ground truth pose of the MAV shows that all states converge to the desired location. **(b)** The tracking errors converge asymptotically towards zero for all states. 100

Introduction

Aerial robotics is a prominent field of research that has seen great commercial success during the last years. Technological advances, such as batteries with increased power storage, accurate light-weight sensing and higher on-board processing capabilities, have allowed the development of highly efficient and affordable small-sized airborne platforms. Mini-drones, miniature unmanned aerial vehicles (mini-UAVs) or micro aerial vehicles (MAVs) are terminologies used to refer to these platforms, with weights ranging from hundreds of grams to a few kilograms [1]. Their payload and flexibility, which allows them to carry a wide array of light-weight perception sensors, together with their easy maintenance and safe operation are some of the characteristics that make them attractive research test beds. Researchers in this field delve into new opportunities for UAVs in a great number of indoor and outdoor applications, solving challenges related to design, control and autonomous navigation.

Following this trend, the *Autonomous Mini-UAVs* research chair, under which this doctoral thesis was developed, was established with the sponsorship and collaboration of the French company Réseau de Transport d'Électricité (RTE), the French transmission system operator. The research activity of this chair focused on improving the autonomy and robustness of mini-UAVs through the use of embarked sensors, the on-board processing of the sensor data and the design of efficient control algorithms, for applications that ranged from full-autonomy to tele-operation. Of particular interest were surveillance and inspection applications, where aerial robotic platforms have seen an increasing demand. In this context, electric tower inspections with mini-UAVs was identified as a potentially interesting application.

Power utilities, such as electric towers, are subject to deterioration due to the atmospheric conditions to which they are exposed. Ensuring their integrity and avoiding network downtime require extensive monitoring programs, that traditionally rely on close, visual inspections by human operators. This procedure is time-consuming, exhausting and

dangerous. Aerial surveys have thus gained increasing popularity as they allow covering vast areas in relatively short periods of time, by relying on thermal imaging, aerial imaging, optical satellites, among other remote sensing technologies (see [2] for a survey). In particular, airborne laser scanning (ALS) technologies have recently attracted a large attention due to their capability of achieving high quality 3D models of infrastructure with high spatial resolution [3,4]. In ALS applications, powerful 3D light detection and ranging (LiDAR) sensors are mounted on manned aircraft, such as helicopters [2, 4, 5], then data acquisition is typically carried out using a GPS sensor and an inertial measurement unit (IMU) to keep track of the aircraft's position and orientation. The geo-referenced range readings are then processed offline for a wide variety of classification or reconstruction tasks such as detecting power lines [5,6], vegetation management [3] and making 3D models of the electric towers [7]. Nonetheless,the high operational costs of piloted aircraft has remained a major set-back for the development of these applications.

The automation of inspection tasks has thus become a key subject of research in the power industry [4]. In this context, MAVs present an attractive solution, since they provide an affordable and flexible mean of gathering spatial information [8–10]. In power utility inspections tasks, global positioning systems (GPS) remain the predominant choice of sensing for autonomous MAV navigation purposes [8]. However, a GPS signal is not always accurate, can be perturbed by the strong electromagnetic fields in the proximity of the power lines [11] and provides no perception of the surroundings environment. As a result, a safe, collision-free flight cannot be achieved relying on GPS measurements uniquely, which is instead limited to waypoint navigation at large distances from the inspected objects [2,8,10]. Such tasks require a proper spatial awareness of the surroundings, which can be achieved with on-board remote sensing.

On the one hand, a great body of work has been dedicated to vision-based solutions (stereo and monocular) for power utility inspection tasks with aerial platforms [8, 9]. These sensors have low-power consumption, are light-weight and affordable, suitable for MAVs [12]. These works have largely focused on power-line inspections [10,12–15], where, in some cases, the visual feedback is used by the MAV platform to track and autonomously follow the power lines [12, 14, 15]. In certain works, the vision-based navigation system is complemented with GPS readings [10, 15]. In others, the additional remote sensing has allowed achieving high-level tasks such as obstacle avoidance [14, 16, 17] and trajectory planning [12, 17], and more specific tasks, such as measuring the distance of the power lines to surrounding vegetation [18]. Despite these results, vision-based solutions suffer from notable drawbacks, such as a high computational cost, making real-time processing a challenge, and a high sensitivity to textures and luminosity [4], which is particularly

Figure 1: A common high voltage transmission line.

problematic in outdoor scenarios.

On the other hand, recent years have seen great advances in light-weight 2D LiDARs, in terms of size reduction, accuracy and measurement frequencies, characteristics which have made them an appealing alternative for MAVs. While performance and precision remain far from their 3D counterparts, which are too heavy to be carried on small aerial robots, they can be effectively used for autonomous MAV navigation, as has been demonstrated numerous times for indoor scenarios [19–23]. Behind the success of these sensors lies the fact that they excel when navigating in cluttered environments, as they directly measure the distance to surrounding objects, with high precision, and naturally open the way for sense-and-avoid functionalities required for safe flights. Extending their use to power utility inspection tasks results interesting, as they can be used for basic and affordable ALS applications, where initial results have been shown for power line monitoring with MAVs [24] and climbing robots [11], while allowing MAVs to achieve higher levels of autonomy and close-up inspections, which is hard to accomplish with other sensors.

Problem statement

The main focus of this work was to explore how 2D LiDARs, coupled with commonly available sensors, could be used to obtain real-time estimates of a MAV's 6 degree of freedom (DoF) pose, using uniquely on-board sensing and processing capabilities. Thus,

throughout this work, the complete sensor setup was as follows:

- **2D laser scanner:** Since odometric sensors to measure raw displacements aren't available for MAVs, an alternative approach is to infer motion from range sensing. In this work, laser range measurements from 2D LiDARs were used for this purpose.
- **Inertial measurement unit (IMU):** This device commonly uses a combination of a three-axis accelerometer, a three-axis rate gyroscope, and a magnetometer [25], and is found at the heart of all MAV platforms. In this work, magnetometers weren't used as they are highly sensitive to magnetic interference, and are very unreliable in the proximity of the power lines. Only an accelerometer and a gyrometer were used for inertial measurements.
- **Altitude sensor:** Two types of altitude sensors were considered: laser altimeters and barometers. On the one hand, laser altimeter measure directly the distance to the ground and are a popular choice indoors. On the other hand, barometers measure changes in atmospheric pressure to determine height and are more common for outdoor navigation.

Pose estimation from range sensing depends greatly on the structure of the surrounding environment. The methodology developed in this work was aimed at electric tower inspection scenes, namely, scenes with steel lattice towers made up of rectangular cross-sections commonly used to support high-voltage transmission lines, such as illustrated in Fig. 1. Throughout this study, particular attention was given to the tower's body, which makes up the largest portion of the structure. The tower heads have a more complex structure that require a separate analysis [7, 26], and were not considered in this work. The long-term aim of this work is to achieve autonomous inspection capabilities of electric towers with MAVs. Before treating the electric tower case, indoor scenes are briefly revisited.

Thesis outline

The rest of this thesis is organized as follows:

- **Chapter 1:** This chapter introduces relevant concepts related to state estimation and autonomous navigation, ranging from pose tracking and sensor fusion, addressed throughout this work, to path planning and exploration. The discussion is focused

on aspects considered in previous works on MAVs equipped with laser scanners. For each topic, examples and results from recent applications are provided.

- **Chapter 2:** This chapter concerns laser-based pose estimation in typical indoor scenes, which are characterized by planar and vertically-flat objects. As such, classic laser scan matching techniques are studied, namely, variants of the popular Iterative Closest Point (ICP) algorithm, and how they are adapted to the MAV's 3D motion. While this provides estimates of the MAV's 2D pose, complete 3D pose estimation is also addressed in this chapter, including attitude estimation from fused gyrometer and accelerometer readings, and height estimation from a laser altimeter. Results based on experimental indoor flights are shown. The purpose of this part was to serve as an initial experience for the more complex electric tower scene, considered in the subsequent chapters.

- **Chapter 3:** This chapter formally introduces the electric tower inspection scene, the main challenges and differences with respect to typical indoor scenes. A method for tracking the tower's cross-sections in 2D is presented, which relies on extracting notable features directly from the 2D laser scans. The advantages and disadvantages of the proposed method are presented based on results from simulations and experimental flights.

- **Chapter 4:** Continuing the study of electric tower inspection scenes, this chapter seeks to develop a 3D pose estimation approach from the laser scans, considering a prior modelling step of the electric tower. The idea is to build on results from Chapter 3 and extend classic scan matching techniques that have been successful indoors, as presented in Chapter 1, to the electric tower case. Here, scan registrations are aided by attitude estimates from fused gyrometer and accelerometer readings. A gain-scheduling approach is presented, to improve attitude estimation performance under changing MAV dynamic states. Finally, a two stage altitude observer is proposed for laser-barometer fusion. Results are presented based on simulated flights.

- **Chapter 5:** This final chapter briefly presents our sensor fusion and closed-loop control designs for the two scenes studied throughout this work, indoors and electric tower inspections, relying on our complete sensor setup and the pose estimation approaches developed in previous chapters. Results based on experimental and simulated flights are presented.

Contributions

Part of the content presented in these chapters was the subject of the following publications:

C. Viña and P. Morin, "Model-based pose estimation on-board MAVs equipped with 2D laser scanners for the automatic inspection of electric towers," *2017 11th International Workshop on Robot Motion and Control (RoMoCo)*, Wasowo, 2017, pp. 78-84.

C. Viña and P. Morin, "Feature-based pose estimation on-board MAVs equipped with 2D laser scanners for the automatic inspection of electric towers," *2017 11th International Workshop on Robot Motion and Control (RoMoCo)*, Wasowo, 2017, pp. 71-77.

C. Viña and P. Morin, "MAV local pose estimation with a 2D laser scanner: A case study for electric tower inspection," *International Journal of Micro Air Vehicles*. Submitted for publication.

CHAPTER 1

Related works on MAV state estimation and autonomous navigation with laser scanners

Contents

1.1 Overview **6**

1.2 Pose tracking **7**

1.2.1 2D laser odometry 8

1.2.2 Height estimation on MAVs 12

1.2.3 Attitude estimation on MAVs 13

1.3 Sensor fusion **14**

1.3.1 Fusing relative state measurements 14

1.4 Towards a completely autonomous MAV **15**

1.4.1 Global localization 16

1.4.2 Simultaneous localization and mapping (SLAM) 17

1.4.3 Fusing relative and absolute state measurements 21

1.4.4 Achieving high-level navigation tasks 23

1.5 Conclusions **25**

1.1 Overview

Recent years have seen an increasing demand for micro air vehicle (MAV) platforms with improved autonomous capabilities. This has been driven by promising applications

ranging from inspection and surveillance, to search and rescue operations in disaster areas. Underlying most of these applications is one of the fundamental problems of mobile robots: autonomous navigation. This is a robot's capability of reaching a desired location without human guidance. While great advances have been made on MAVs, work still needs to be done to provide the levels of reliability and safeness that will allow this technology to proliferate on a larger scale.

This chapter addresses several topics related to MAV autonomous navigation and is divided in two parts. The first part is dedicated to the on-board and real-time aspects of state estimation, which were the main focus of the work and contributions of this doctoral thesis. We present several key concepts on the topics of pose tracking (Sec. 1.2) and sensor fusion (Sec. 1.3), and highlight notable examples of successful applications on MAVs, focusing on works that use 2D laser scanners for range sensing. Together with control, which will be addressed in the final chapter of this dissertation, these three tasks constitute the first basic steps towards achieving autonomous flight capabilities.

Building on the first part, the second part of this chapter, presented in Sec. 1.4, provides an insight into the remaining set of tools required for achieving truly autonomous capabilities on MAVs equipped with laser scanners. This covers topics from global localization in Sec. 1.4.1, to simultaneous localization and mapping (SLAM) in Sec. 1.4.2. Then, in Sec. 1.4.3 we discuss recent developments in multi-sensor fusion with aims to achieving reliable and stable closed-loop control in prolonged flights and large-scale environments. Lastly, Sec. 1.4.4 presents a brief survey of previous works on high-level navigation tasks, that can be achieved relying on the previous steps, such as obstacle avoidance, path planning and exploration. Overall, in Sec. 1.4 we provide a brief theoretical overview of each of these topics and examples of direct applications on MAVs with laser scanners. While this second part of the chapter covers subjects that were not treated throughout this work, it gives a clear understanding of the complexity of tasks that can be achieved on MAVs using simple 2D LiDARs, and serves as a guideline into possible directions and challenges that will be faced in continuations of this work.

1.2 Pose tracking

The first problem to address in autonomous navigation is estimating the robot's relative pose with respect to the immediate surroundings, also known as *pose tracking* or *local pose estimation*. This stage mainly concerns the online and real-time aspects of pose estimation, and is thus essential to ensure stable and fast navigation [27]. In mobile

robots, these techniques are usually the front-end of a complete navigation pipeline, and as such they must be computationally efficient and accurate, as this affects all subsequent steps of more complex tasks such as mapping, planning and exploration.

Pose tracking suffers from two main limitations. On the one hand, the initial location must be known in advance, since pose tracking is traditionally the result of incremental pose updates or local observations of the environment. This, however, is a key enabling factor for achieving real-time capabilities, since a precise initial pose allows focusing estimation in a reduced (local) state space. On the other hand, most tracking techniques are naturally incapable of recovering from localization failures. Rectifying both issues is part of the *global localization* problem, where a robot has to determine its location under complete uncertainty. This will be the subject of discussion in Sec. 1.4.1.

A MAV's pose in 3D space is described by 6 degrees of freedom (DoF) corresponding to 3 translations and 3 rotations. Tracking the complete 6 DoF pose depends on the on-board sensor layout. As previously described, we consider a MAV equipped with a 2D laser scanner, an IMU (accelerometer and gyrometer), and an altimeter (laser-based or barometer-based). We now give a brief overview of how previous works have recovered pose information from each of these sensors.

1.2.1 2D laser odometry

We start with the simplified problem of 2D pose estimation: translations in the X-Y plane and rotations about the Z axis (the yaw angle). For many years, this has been at the heart of research for pose estimation on ground robots, with typically planar motion. For these platforms, odometry is readily available from embarked sensors such as wheel encoders. MAVs, on the other hand, don't provide a similar possibility. Pose estimation by integrating accelerometer and gyroscope measurements leads to large drifts in short periods of time, and is instead considered as part of the sensor fusion procedure (state prediction). This will be discussed in Sec. 1.3. At a local scale, a MAV's 2D motion must instead be inferred from range sensing, such as laser scanners.

This brings us to the discussion of pair-wise *registration* techniques, which seek to associate pairs of data sets into a common coordinate system by minimizing the alignment error. The registration of free-form 3D shapes (i.e., point sets, line sets, parametric curves and parametric surfaces) is a key problem in computer vision, with a wide range of applications such as object reconstruction and facial recognition [28]. The extensions of registration techniques to laser scans, which are essentially a set of points (i.e., a

point cloud), is referred to as *scan registration* or *scan matching*. With the introduction of laser scanners in mobile robots, following the popularity of the SICK LMS-200 and the subsequent development of small-sized Hokuyo laser scanners [29], scan matching techniques found an application in robot pose estimation.

The problem of pose estimation from laser scan matching, or *laser odometry*, is formulated as follows: given two measurements taken at two different positions, corresponding to an arbitrary known initial pose and the current position, find the rigid body transformation that best aligns the two point sets (i.e., two laser scans, or a scan and a map). From the pose estimation point of view, this transformation corresponds to the robot's relative displacement between measurements. This concept was popularized by Lu and Milios in their landmark study for 2D pose estimation from 2D range scans [30]. In subsequent years, laser odometry saw great success on ground robots as an alternative or complement of wheel odometry. More recently, 2D laser odometry has also been extended to MAVs [19–23], however, with more limited success due to the particular constraints imposed by these platforms. This will be discussed at the end of this section.

In this section, we are interested in how 2D laser range sensing has been used as a source of on-board odometry, particularly for MAV platforms. As the body of work available in this subject is very dense, we start with a general classification of the techniques into deterministic (Sec. 1.2.1.1) and probabilistic (Sec. 1.2.1.2) approaches, and focus the discussion on methods that have provided successful results on-board MAVs.

1.2.1.1 Deterministic approach

This family of algorithms treats the registration problem from a purely geometric perspective. The physical properties of the sensors involved, and the uncertainties in the measurements and registration process are not taken into account. Instead, the focus is to reduce the alignment error between a pair of shapes. Laser depth measurements provide a discrete approximation of the surrounding shapes in terms of a set of points. Different registration approaches can be classified based on how these points are associated.

On the one hand, *feature-based* approaches seek to extract geometric primitives from the smooth areas covered by the raw measurements. These primitives are simple shapes that can be easily parametrized, such as key-points (e.g., corners, edges), lines, planes and curves. Extracting features reduces the size of the registration problem, and feature-based registration algorithms can be computationally efficient. On the downside, these approaches require the surrounding environment to be structured and information is lost

in feature extraction process. Popular variants include Hough Scan Matching (HSM) [31], Iterative Closest Line (ICL) [32] and Point-to-line Iterative Closest Point (PLICP) [33].

On the other hand, *point-to-point* matching techniques use directly the raw range measurements. These techniques don't require assumptions about the surrounding geometry nor the existence of predefined features. As such, they can be very robust and flexible as they can work for structured and unstructured scenes. However, these approaches can be time-consuming, since the computational cost scales rapidly with number of points used in the registration process. Popular algorithms include Iterative Closest Point (ICP) [34], Iterative Dual Correspondence (IDC) [30] and the Polar Scan Matcher (PSM) [35].

A common drawback of deterministic approaches is their sensitivity to outliers, which are typically the product of occlusions, moving objects and sensor noise. Since the physical properties of the sensors aren't taken into account, heuristics have to be used instead to deal with outliers. This includes setting distance thresholds for point correspondences, rank filters or more advanced methods based on robust statistics [36, 37]. The presence of outliers can easily degrade the quality of the registrations, particularly in iterative methods, and an outlier rejection strategy must always be considered. Another common drawback is that most of these approaches only guarantee convergence to a local minima and require a good initial guess to perform correctly. A bad initialization may lead the algorithm to converge to a local minimum far from the optimal solution.

In any case, the registration task usually follows a similar procedure: establishing point correspondences and minimizing the alignment error (in the least squares sense). Following these two steps, the ICP algorithm [34] is by far the most widely used technique for pair-wise registration, due to its simplicity and efficiency. The baseline ICP algorithm establishes point correspondences based on a closest-point rule and the Euclidean distance, then computes the rigid body transformation that minimizes the alignment error (the sum of squared Euclidean distances) between point correspondences. These two steps of the registration process are carried out iteratively, refining the alignment at each iteration, until a local minima is reached. Different variants seek to improve the baseline ICP algorithm in terms of convergence rates and accuracy by using different sampling strategies, distance metrics, weighting, correspondence rejection and optimization techniques [37]. Properties of the local structure, such as normals or curvatures, can be extracted from the raw points to improve the data association and error minimization [38, 39]. A more detailed discussion on the ICP algorithm, its variants and limitations will be given in Chapter 2.

Applications on MAVs

Laser odometry from variants of the ICP algorithm has been employed in numerous works with MAVs [20, 22, 23]. In [23], Sa et al. rely on the Point-to-line Iterative Closest Point (PLICP) variant [33], which uses a point-to-line distance metric known to lead to faster convergence rates. They carry out experimental validations on a low-cost MikroKopter platform, using a Hokuyo URG-04LX 2D laser range finder with 10 Hz measurement rate, but all laser processing is performed off-board. Works that have achieved on-board laser odometry based on ICP include [20, 22]. In [20], Shen et al. employ the ICP algorithm in its basic form to recover 2D pose estimates at 20 Hz on-board. Dryanovski et al [22] instead use the PLICP algorithm, however, they achieve 2D pose estimation at 30 Hz. Both works rely on an Astec Pelican platform with Hokuyo UTM-30LX 2D laser scanner (40 Hz rate), an on-board 1.6 GHz Atom processor and 1 Gb of RAM.

1.2.1.2 Probabilistic approach

A common probabilistic formulation is to treat the registration task as a maximum likelihood problem [40, 41]. The goal is to find the rigid body transformation that maximizes a scan's likelihood, that is, the probability of obtaining a laser measurement, given an initial guess (i.e, the current pose) and an environment "model", which can be a previous scan, for incremental scan matching, or a previously obtained map of the surroundings. In this case, the underlying probability distribution is referred to as the *sensor model*, as it captures the uncertainties and physical properties of the sensor.

A well-known sensor model for laser scanners is the *mixture beam-model* [42], which seeks to approximate the physical causes of the individual measurements, such as reflections from known objects, reflections from dynamic objects, failures due to transparency, maximum range readings, among others. This model considers the fact that laser beams cannot go through objects, and allows dealing with occlusions without needing special heuristics, unlike deterministic approaches. Note that the maximum sensor range, which is often ignored in deterministic approaches, also has a probabilistic interpretation and is exploited in this sensor model. A probabilistic scan registration algorithm based on this model was proposed in [41].

However, computing the mixture beam-model requires using ray-tracing techniques that can be very time consuming. As an alternative, *likelihood fields* only consider the beam's endpoints and the distance to the nearest object in the environment model (map).

The distribution itself is usually a mixture of a Gaussian, centered at the distance to the nearest obstacle, and uniform distributions, to account for random measurements and maximum sensor range. Assuming that the map is static, a look-up 2D grid table can be pre-computed, often in terms of log-likelihoods, to allow for significant speed gains [43].

In either case, the laser beams in a scan are considered conditionally independent (given the environment model) and distributed according to the sensor model. Then, finding the maximum likelihood estimate consists in searching through the entire state space, which can't be done exhaustively due to its continuous nature. On the one hand, several works use iterative gradient-based methods to simplify this task, such as hill climbing [41]. These techniques are fast, but are prone to local minima and require a good initialization. On the other hand, grid searches discretize the state space and find an optimal solution through extensive sampling. These techniques are more robust than gradient-based methods but inherently slow, depending on the resolution and grid size. In [43], Olson et al. propose the Correlative Scan Matcher (CSM), and a multi-resolution approach to implement grid searches in real-time. They also recover covariance matrices as a measure of the uncertainty of the registrations, which is useful for robotics applications.

Many more formulations of probabilistic registrations exist. Some works seek to adapt the correspondence search and error minimization steps of the ICP algorithm to a probabilistic framework, such as the Probabilistic Iterative Correspondences (pIC) algorithm [44] and the Generalized-ICP (GICP) algorithm [45]. Other works rely instead on particle filter formulations, which can approximate arbitrary probability distributions, and don't require restrictive Gaussian assumptions [46, 47]. In any case, probabilistic methods overall share the same advantages and disadvantages with respect to deterministic approaches. By considering the sources of uncertainty in the registration process, probabilistic methods tend to be more robust to initialization errors, noise and outliers. This, however, comes at a high computational cost that makes attaining real-time capabilities with limited computational resources a challenge, even in situations where deterministic approaches can achieve this seamlessly.

Applications on MAVs

Several notable applications of probabilistic registration methods on MAVs can be cited. In [21], Grzonka et al. present an autonomous indoor MikroKopter quadrotor equipped with a Hokuyo URG-04LX sensor (10 Hz), and use a variant of the CSM algorithm to estimate the MAV's incremental 2D motion. They maintain a history of previous laser

scans and pose estimates, and use likelihood fields and a multi-resolution grid search approach, as proposed in [43], to recover the most likely pose. Their scan registrations take an average of 5 ms in indoor tests. However, all laser processing is performed off-board in a ground station computer, which introduces significant delays (up to 120 ms) in the estimation process.

In [19], Bachrach et al. present an AsTec Pelican quadrotor equipped with a Hokuyo UTM-30LX 2D laser scanner (40 Hz), a 1.6 GHz Intel Atom processor and a 1 Gb of RAM, capable of navigating autonomously in unknown GPS-denied environments. For their map representation, they assume that the scans measure planar surfaces (as indoors and urban scenes), and polyline contours are extracted from a history of previous laser scans and used to generate a 2D map of likelihood contours. Then, for the alignment search, instead of using an exhaustive grid search, as in the CSM algorithm [43], they use a faster hill climbing technique. This is justified by the fast measurement rates (40 Hz). Unlike [21], the scan registrations are performed on-board. Their scan registrations take an average of 12.5 ms indoors and in an urban canyon scene.

Discussion

A common characteristic of the cited works on 2D laser odometry applied to MAVs (probabilistic and deterministic) is that they have focused heavily on indoor (structured) scenarios [19–23]. This has primarily two causes. On the one hand, MAVs navigate in 3D environments and the 2D laser scans can capture different objects at different heights. Attempting to align pairs of 2D scans without considering the MAV's 3D motion will lead to false correspondences and poor performances. Typically, this is handled by assuming that surrounding structures are planar and invariant to height, an assumption that primarily holds for indoors and urban scenes. This is the topic of discussion of Chapter 2.

On the other hand, 2D laser scans only capture a planar slice of the environment. In structured scenes, such as indoors, this is sufficient to recover well-defined contours of the surroundings that enables using pair-wise scan registrations techniques. However, in unstructured 3D environments, overlap between pairs of 2D scans can be very limited. This compromises the quality of the scan registrations, as noted in [19], where their platform was incapable of operating in densely vegetated areas and featureless scenes, such as wide open spaces and long corridors. These scenes represent the inherent limitations of using 2D laser scanners as the only source of range sensing. Recent works address these limitations using multiple range sensing modalities. This will be discussed in Sec. 1.4.3.

1.2.2 Height estimation on MAVs

There exists a wide variety of sensors for determining a MAV's altitude. The choice depends on the type of environment. In this section we are interested in laser altimeters and barometers. On the one hand, laser altimeters measure directly the distance to the ground and are a popular choice for indoor navigation. Different techniques adopted in previous studies will be presented in Sec. 1.2.2.1. On the other hand, barometers measure the change in atmospheric pressure to determine the height, and are a popular commercial solution for outdoors navigation. Barometer-based height estimation will be discussed in Sec. 1.2.2.2.

1.2.2.1 Laser-based height estimation

On MAVs equipped with 2D LiDARs, numerous works estimate altitude by placing mirrors to reflect multiple laser rays downwards and directly measuring the distance to the ground [19–23]. This allows avoiding the costs and power-consumption of using additional sensors. This approach proves reliable when navigating over flat surfaces. However, any sudden change in the floor elevation produced by objects in the scene leads to sharp jumps in the height estimates, which can compromise the position controller. Several solutions have been proposed to account for this situation, by assuming that the ground elevation is piecewise constant and trying to detect the discontinuities.

In [22], Dryanovski et al. present a robust approach which relies on altitude histograms from 20 deflected laser rays to estimate the MAV's altitude, while tracking the floor's elevation. First, the roll and pitch angles estimated from IMU measurements are taken into account to correct the range measurements. Then, they create the altitude histograms with bin sizes of 2 cm, and they average the measurements that fall on the bin that contains the peak of the histogram. This allows them to identify any discontinuities, which are assumed to be produced by edges on the floor.

In [21], Grzonka et al. simultaneously track the MAV's absolute height and the elevation of the ground. They create multi-level grid maps of the floor, where 2D grids are grouped into levels that correspond to a constant height with respect to a fixed frame. First, given the laser altitude measurements and current 2D pose (estimated from a SLAM module), they estimate the MAV's altitude from the current multi-level map. A Kalman filter is then used to fuse this estimate with inertial measurements to obtain the robot's vertical speed. Then, this information is used to update the ground elevation map and

add any new levels that may have been detected in the range measurements. A second set of Kalman filters track the elevation of each level separately. In their experimental validation, they correctly estimate the MAV's height in an office environment, while detecting changes in elevation due to chairs and tables, but provide no insight into the computational cost of their approach.

In [19], Bachrach et al. also deflect 20 laser beams. They first average the measurements, then directly differentiate the averaged value to obtain the vertical velocity. This allows detecting discontinuities on the floor, as any sudden change of elevation is translated into large peaks in the velocity estimates, which are detected based on the maximum expected acceleration of the platform. Hence, they assume that the MAV flies over a flat floor, and any change in elevation is due to small objects and treated as a brief local disturbance. They rely on a complementary filtering approach to obtain smooth altitude estimates, which considers the distance travelled after detecting a discontinuity. They achieve an altitude RMS error of 2 cm and vertical velocity RMS error of 0.2 m/s.

1.2.2.2 Barometer-based height estimation

While laser altimeters have proven to be effective when navigating indoors, performance remains highly dependent on the floor's layout, which can be very irregular in typical outdoors environments. In these scenarios, barometric sensors are a popular choice among commercial MAVs. These sensors estimate the absolute or relative height of an object by measuring the atmospheric pressure. However, fluctuations in pressure due to weather conditions cause these height measurements to drift over time. Sensor fusion techniques are thus used to estimate and compensate this drift by using additional sources such as GPS [48], and IMUs [49, 50]. More recently, differential barometry has been gaining popularity [51, 52]. In this configuration, a second barometer is set stationary on the ground and used as a reference measurement to track changes in local pressure, effectively reducing drift and increasing accuracy. While recent works have obtained impressive results with differential barometry [51, 52], the focus of this work was using on-board sensing only, and differential barometry was not considered.

1.2.3 Attitude estimation on MAVs

Fast and accurate attitude estimates are an essential part of any MAV platform. Absolute attitude information can be recovered from magnetometers and accelerometers [25,53,54].

On the one hand, magnetometers provide measurements of the surrounding magnetic field in the body attached frame, and allow deducing the MAV's heading [25, 55]. However, they are very sensitive to local magnetic fields and measurements can be noisy. On the other hand, accelerometers measure the so-called *specific acceleration*. When the linear acceleration is small, this sensor directly measures the gravity vector, thus acting as an inclinometer and providing direct observations of the roll and pitch angles. This is a common assumption applied in attitude estimation [25, 54, 56], which has shown to work well in practice. On the downside, accelerometers are highly sensitive to vibrations induced by the propellers and require significant filtering to be useful [53]. This, in exchange, can introduce important latencies in the estimations. Thus, complementary attitude information is commonly obtained from gyrometers, which measure the angular velocity along the three rotational axis in the body attached frame. These sensors are less sensitive to vibrations and are very reliable. Absolute attitude can be recovered for the three rotation axis by integrating the measured angular rates, however, this causes the estimation error to grow without bound [53].

Hence, sensor fusion techniques are used to combine the information from all three sensors to tackle drift and noise issues, and to obtain more accurate attitude estimates. In literature, the use of linear stochastic filters, such as Kalman filters [53] or Extended Kalman filters (EKFs) [57, 58], as the means to fuse inertial measurements is very common. While these filters have been successful in certain applications, they can have an unpredictable behaviour when applied to non-linear systems [59]. An alternative is to use non-linear observer design techniques, which present strong robustness properties and guaranteed exponential convergence [25, 59]. Numerous recent works have shown successful results in obtaining accurate attitude estimates from noisy and biased measurements using low-cost IMUs [59, 60]. In this work we adopt a non-linear observer formulation to obtain attitude estimates which will be presented in Chapter 2.

1.3 Sensor fusion

So far we have described how previous works have tracked a MAV's pose in 3D space and in real-time given our sensor setup. The question now is how to recover the accurate velocity estimates required for the MAV's control loop. Directly differentiating the position estimates is avoided in practice, as this greatly amplifies any underlying noise, leading to unreliable results [22, 23]. Instead, this is achieved through sensor fusion techniques, by exploiting the MAV's dynamics and the multiple on-board sensors.

Sensor fusion was slightly addressed in Sec. 1.2.2 and Sec. 1.2.3, as it was required to recover several of the MAV's states. For the purposes of this section, however, we view the sensor fusion problem as the intermediary step between localization and control. From this point of view, the pose estimates derived in the previous section are seen as "measurements" of their corresponding states, and the outputs of this fusion step are filtered pose estimates and velocity estimates. To facilitate the discussion, the state measurements are classified as *relative* or *absolute* (global), according to the nature of the underlying sensor. Absolute measurements provide direct observations of a state in a unique, consistent frame (e.g., GPS, magnetometer). Relative measurements reflect changes of a state in a time interval, such as a robot's displacement between two sensor samples. This is usually the case when inferring motion from range data (e.g., laser odometry, visual odometry). In both cases, the measurements are used for state correction (update). Motion derived from inertial measurements is instead used for state prediction (propagation).

1.3.1 Fusing relative state measurements

In this case, the main concern is obtaining pose and velocity estimates in real-time. Literature regarding this topic is very vast, and is linked to the type of sensing used on-board. On MAVs equipped with 2D LiDARs, the goal is to fuse the laser odometry measurements (Sec. 1.2.1) with the inertial measurements. Stochastic filters, such as EKFs, are predominantly used for this purpose [19, 20], while simpler complementary filters have also provided satisfying results [23]. Other works focus on using cascades of filters for further noise reduction. Dryanovski et al. [22], first use an alpha-beta filter to obtain rough initial velocity estimates from the laser position estimates, which are then used as a correction in a Kalman filter which includes inertial measurements. Shen et al. [20] propose a cascade of two separate EKFs to achieve accurate results and high rates.

It is important to note that state estimation from relative state measurements is in reality a complex data fusion problem. Since relative measurements reflect changes of a state between two time instants, they have a direct dependence on the current and a previous state of the system. This violates a basic assumption of stochastic filters, such as Kalman filters and its variants, which require the measurements to be independent from any previous filter states. This issue is often overlooked in works with MAVs, which instead adopt a simplified solution by treating the relative measurements as *pseudo-absolute* state measurements: The current relative measurement is applied to the previous state estimate

and directly used to update the filter [19, 20, 22, 23]. While this approach is sufficient for simple applications, ignoring the state dependencies that arise from relative measurements leads to sub-optimal estimates of the robot's states and covariances [61,62]. Better results can be obtained by taking them into account.

Stochastic Cloning Kalman filter (SC-KF). Due to their recursive nature, standard Kalman filter frameworks do not provide a direct way to determine dependencies between states estimated at different times. Therefore, taking into account the correlations introduced by relative state measurements require reformulating the filtering problem. To deal with this, the Stochastic Cloning Kalman Filter (SC-KF) [61] is a variant of the EKF that seeks to transform the dependence of relative measurements on previous states, into a dependence on the current filter states. This is done by augmenting the state vector to include two copies (clones) of the state resulting from the last filter update. The first copy evolves as usual with filter predictions (from proprioceptive measurements), while the second copy remains unchanged. As a result, the robot states affected by the relative measurement are both represented explicitly in the filter, and the classic EKF framework can be applied. Experimental validations for laser and wheel odometry fusion demonstrate that drift is greatly reduced, compared to a standard EKF. Only recently have adaptations of the SC-KF been tested on MAVs, for operations in large-scale environments [58, 63]. This will be discussed in Sec. 1.4.3.

Discussion

Data fusion relying uniquely on relative measurements provides the real-time capabilities required for a stable control loop. However, small estimation errors accumulate over time, leading to drift in the state estimates. Steps can be taken to mitigate this issue, such as using the SC-KF framework, which can efficiently increase the time that a platform can navigate with acceptable levels of error [61]. Nonetheless, without direct state observations, drift is unavoidable when covering large distances or in prolonged flights. Further complications arise from requiring an initial known pose and the incapacity of recovering from large errors. Consequently, the effectiveness of this kind of approach is limited to flights of short durations or small-scale environments. For more general applications it is necessary to introduce absolute state measurements from sources such as GPS or SLAM algorithms to tackle these issues [58, 63]. This is an essential aspect for achieving truly autonomous capabilities and will be discussed in Sec. 1.4.3.

1.4 Towards a completely autonomous MAV

A truly autonomous robotic platform must be capable of reaching a desired location in an unsupervised way. The first two steps towards achieving this are pose estimation (localization) and control. Pose estimation at a local scale (pose tracking) and in real-time was discussed in Sec. 1.2. MAV feedback control will be discussed in the final chapter of this dissertation, however, this problem requires accurate pose and velocity estimates in real-time, which are obtained through an intermediate step that combines information from the multiple on-board sensors through data fusion techniques, as discussed in Sec. 1.3. Building on these basic steps, in this section we describe the remaining steps of a complete navigation pipeline. In the following discussions, we focus on methods that have been employed on MAVs equipped with 2D laser scanners.

Pose tracking as described in Sec. 1.2 is perhaps one of the most studied problems in mobile robots [64]. It has also been the main focus of this work, as the aim is to achieve real-time on-board pose estimation. However, local pose estimation requires a known initial pose and typically can't recover from large tracking errors [42]. This proves troublesome for navigating safely during long sustained flights and in large-scale environments. A truly autonomous platform must be capable of addressing these issues, which requires introducing absolute (global) state measurements in the estimation process. In this section, we describe how to obtain this information from laser range measurements through global localization techniques (Sec. 1.4.1), in known environments, and SLAM algorithms (Sec. 1.4.2), in unknown environments. Then, we present how these measurements can be used for reliable state estimation in prolonged flights and large-scale environments through multi-sensor fusion techniques (Sec. 1.4.3). This section ends with a brief insight into high level tasks previously achieved with MAVs and 2D laser scanners. As previously mentioned, these topics were not treated throughout this work, but give an overall view of the type of tasks that can be achieved on MAVs using simple 2D LiDARs, and provide insights into possible directions for continuations of this work.

1.4.1 Global localization

This problem focuses on determining a robot's pose with respect to a unique, globally consistent frame, without knowledge of the initial position. It is also assumed that the robot is navigating in a known environment, that is, that a map of the surroundings is available. Probabilistic frameworks have become the main choice for global localization

tasks. In this case, the robot's pose is considered to be a time-varying random variable, whose posterior distribution (also referred to as the *belief* [42]) is conditioned by the data known up to that time: the previous state estimates, observations (range measurements), actions (control inputs or odometry readings) and a map (assumed to be static).

As the robot's location is unknown, it must instead be *inferred* from the available data. Most popular localisation algorithms rely on the Bayes filter, a probabilistic inference technique, to estimate a robot's belief [42, 64]. The key of this formulation is the Markov assumption, which implies that future and past data (measurements) are independent if one knows the current state, in this case the robot's pose. On the one hand, this allows keeping track of the belief recursively, i.e., using only the previous estimate (referred to as the *prior*) and most recent data, which is convenient for online robotic applications. On the other hand, further derivations of the Bayes filter allow characterizing the belief by two conditional distributions known as the *motion model* (or transition model) and the *sensor model*, assumed to be time-invariant. First, the motion model is a probabilistic approximation of a robot's kinematics. Second, the sensor model represents the range sensor's physical properties (as previously described for scan registrations in Sec.1.2.1.2). This characterization allows estimating the belief in a simple two-step prediction correction scheme, depending on the sensory input: odometry readings or control inputs drive the motion model, used for prediction, and the range measurements are comprised in the sensor model, used for correction.

A key characteristic of global localization problems is that they require representing situations in which a robot maintains multiple and distinct guesses of its location. This implies that the underlying probability distributions are complex and multi-modal. Algorithms that make simplifications (e.g., Kalman filters which assume uni-modal Gaussian distributions) are not capable of handling these types of applications. Therefore, global localization algorithms seek to estimate the complete posterior distribution, without making restrictive assumptions about its nature [42]. With such a broad knowledge of the localization uncertainties, the initial pose does not have to be known in advance. Furthermore, handling multi-modal distributions opens the way to recovering from sudden tracking errors, formally known as the *kidnapped robot problem* [42].

Monte Carlo localization (MCL). This popular algorithm for global localization was proposed by Fox et al. in [64, 65]. This algorithm relies on particle filters which can approximate arbitrary probability distributions. The idea is to use a set of samples (particles), which represent a guess of the robot's pose, with corresponding weights (referred

to as *importance factors*), to establish a discrete approximation of the desired posterior distribution. As the robot moves, new particles are generated by sampling poses from the current belief, and predicting the following pose using the robot's motion model. The weights of each particle are then re-calculated using the measurement model [66]. As particle filters focus drawing samples from regions with high probability, they use available computational resources efficiently, and are suitable for robotics applications [65,66].

Applications on MAVs

Variants of the MCL algorithm have been previously used on MAVs for 2D pose estimation from 2D laser range measurements. In [21], Grzonka et al. use MCL to align the laser scans to a previously obtained grid-map. An incremental scan matcher based on the CSM algorithm was used as an odometry input, and likelihood fields to represent the sensor measurements (as described in Sec. 1.2.1.2). For their experimental validation, the map was obtained from a ground robot, and the test flights were carried out at a constant height. However, laser processing was performed on an off-board computer. On the other hand, Dryanovski et al. in [22] also provide brief experimental results based on MCL. They perform estimation on-board and successfully localize a MAV from an unknown position given a known map. However, no details are given on estimation errors and computation time. In both cases, experiments were carried out indoors.

1.4.2 Simultaneous localization and mapping (SLAM)

As was seen in the previous section, there exists efficient solutions for determining the robot's pose when a map of the environment is known in advance. However, many practical applications require navigating in unknown scenarios. This leads to a more complex problem, as the map and robot's location must be determined at the same time, while the robot moves. This is the problematic addressed by simultaneous localization and mapping (SLAM) algorithms. Here, uncertainty and sensor noise are prevailing factors and most SLAM techniques are cast in a probabilistic framework. In fact, the Bayes filter formulation described in Sec. 1.4.1 underlies most state-of-the art algorithms in probabilistic robotics, including SLAM. Unlike the global localization case, however, here the map is an unknown state of the system which must also be inferred from the sensor measurements.

There exists two alternative formulations to SLAM. The first, called the *full SLAM problem*, seeks to estimate the posterior over the entire robot trajectory and the map,

and is mainly used for offline batch processing. The second, called the *online SLAM problem*, only estimates the posterior over the current pose and the map, and is mostly used for incorporating data incrementally. In either case, the high-dimensionality of the problem is a major concern for practical implementations, and finding computationally efficient solutions is one of the main motivations behind the large diversity of approaches to the SLAM problem. For a comprehensive overview of this topic, the reader is referred to [67–69]. The following discussions present a brief survey of SLAM techniques that have been applied on MAVs equipped with 2D laser scanners.

1.4.2.1 Particle filter based SLAM

The particle filter formulation that has been successfully implemented in global localization problems (e.g., the MCL algorithm from Sec. 1.4.1) can also be extended to SLAM applications. However, standard particle filters are very inefficient in high-dimensional spaces [65]. This is the case for both formulations of the SLAM problem. The strategy for tackling this issue revolves around reducing the dimension of the space that requires sampling.

While filtering approaches to SLAM usually rely on the online SLAM formulation, interesting conditional independence properties between maps and trajectories of the full SLAM problem can be exploited to derive an efficient formulation. In particular, if a robot's trajectory is known, then the full posterior distribution can be expressed in a factored form, where the map and trajectory posteriors are separated [67,68]. Then, the map posterior can be solved analytically, while the trajectory posterior, of greatly reduced size, can be approximated, e.g., through particle filtering. This technique is formally known as *Rao-Blackwellisation*, and the resulting estimator is the Rao-Blackwellized particle filter [70].

In essence, the Rao-Blackwellized particle filter separates the full SLAM problem into a localization problem, through particle filtering, and a mapping problem with known poses, that is solved analytically [71]. In this case, each particle contains a possible trajectory and a map. This idea was developed by Murphy in [72], and popularized by Montemerlo et al. with the FastSLAM algorithm [71] for landmark-based mapping, where each particle represents a guess of the robot's trajectory with corresponding landmarks, which are individually tracked with Kalman filters.

GMapping. More recently, Grisetti et al. proposed the GMapping algorithm [73], which uses Rao-Blackwellized particle filters to efficiently learn grid maps. While this map representation has higher memory requirements than landmark maps, as used in FastSLAM, it can represent arbitrary unstructured environments. Moreover, they propose an adaptive re-sampling technique, limiting the risk of deleting good samples from the filter (which can lead to particle depletion [73]). Also, they propose an improved distribution model based on laser scan matching for drawing samples more accurately. This greatly reduces the number of particles required by the filter (by approximately one order of magnitude when compared to previous methods [73]) resulting in a very computationally efficient solution. These characteristics have made GMapping one of the most widely used SLAM algorithms in robotics applications [74].

Applications on MAVs

On MAVs, particle filter based SLAM techniques have been used in several works. In [75], Bachrach et al. use an adaptation of GMapping to obtain 2D grid maps of unknown indoors environments. They use a motion model based on laser scan matching (as described in Sec. 1.2.1) instead of the standard wheel odometry for ground robots. They also modify the 2D grid-map representation to account for changes in height and attitude. In the experimental results, processing the laser scans takes up to 1-2 seconds on an off-board computer. In [22], Dryanovski et al. also use GMapping for indoors 2D SLAM. All laser processing is achieved on-board, however no details are given on estimation errors or processing time.

1.4.2.2 Graph-based SLAM

A key drawback of filter-based SLAM approaches, such as EKF or particle filter formulations, is that information is discarded once it has been processed. Situations that lead to large uncertainties, such as large loop closures or nested loops, can't be handled efficiently as they require revisiting past data and correcting past errors. Furthermore, in the case of particle filters, the choice of the number of particles and how it scales with the size of the environment is a poorly understood problem.

An interesting alternative are graph-based SLAM algorithms [69]. Here, the idea is to construct a pose graph, where each *node* represents a robot pose with corresponding observations of the environment (raw range measurements). In certain cases, landmark (feature) locations, which are extracted from the measurements, are also included in the

graph. Observations and landmarks contain spatial information that relate neighbouring nodes and allow establishing constraints between them, referred to as *edges*. Then, the node configuration that best satisfies the entire set of constraints defines the most likely map and trajectory for the given measurements, in other words, it's a solution to the full SLAM problem [69].

Since the pose graph is constructed for the complete history of robot poses and sensor measurements, finding the optimal node configuration requires solving a very large optimization problem. Nonetheless, the graph representation of the full SLAM problem has an underlying sparse nature, and recent advances in sparse linear algebra have allowed solving these large optimization problems in a computationally efficient way [69,76]. This allows highlighting one of the main advantages of the graph-based SLAM formulation: high-dimensional problems resulting from large-scale environments can be handled efficiently. In effect, some of the largest maps constructed from SLAM algorithms have been the result of graph-based techniques [68].

Graph-based SLAM methods can be decomposed into two different tasks, which are commonly carried out in an alternating way: *graph construction* and *graph optimization*. We now give a brief insight into each of these tasks.

Graph construction. This first task depends heavily on the type of sensing used, as it consists in determining the constraints between nodes from the raw measurements. This can be achieved on-line and is commonly referred to as the SLAM *front-end*. Most constraints are derived from incremental motion estimation (such as wheel encoders or incremental scan matching) between successive nodes, referred to as *odometry edges*. However, when navigating in an unknown terrain, the uncertainty in SLAM becomes increasingly large as errors are accumulated along the estimated trajectories. These errors have no bounds and can only be corrected once a previously explored place has been revisited (i.e., closing the loop). Constraints derived from recognizing previously explored areas are referred to as *loop closure edges*. In general, constraints are non-linear as they typically encode rigid body transformations.

The main challenge in the graph construction is solving the so-called *data association problem*, that is, determining if measurements taken at different positions correspond to the same object. This can be difficult due to eventual ambiguities or symmetries in the environment. Establishing associations between landmarks can be achieved with sophisticated approaches such as branch and bound, spectral clustering, among others [69]. If landmarks aren't considered, this is often the result of aligning range measurements

between neighbouring nodes through scan matching as in [77, 78].

Graph optimization. This second task aims at finding the optimal node configuration given the graph constraints. This stage is referred to as the SLAM *back-end*, and doesn't require specific knowledge of the sensor measurements, but the constraints must be known in advance from a SLAM front-end. A common assumption is that graph constraints are independent of each other and normally distributed [69, 76, 77], which allows developing the posterior distribution of the full SLAM problem into a convenient quadratic equation. This leads to a non-linear least squares problem that can be solved with standard iterative solvers such as Gauss-Newton or Levenberg-Marquardt. However, recent works seek to build upon algorithms that deal with large-scale optimization problems efficiently by exploiting the sparse nature of the graphs. Popular SLAM back-ends include Graph-SLAM [76], general graph optimization (g2o) [79] and incremental smoothing and mapping (iSAM) [77].

Applications on MAVs

In recent years, graph-based SLAM has become the state-of-the-art approach in terms of accuracy and robustness. The shift towards this family of algorithms is noticeable in works with MAVs and LiDARs. Bachrach et al., who previously used particle-filter based GMapping in [75], rely on pose graph SLAM in more recent studies [19]. Here, they construct a pose-graph using their scan matcher (described in Sec. 1.2.1.2) for incremental pose estimation and loop closure detection. Whenever a loop closure is detected, the graph is optimized with the incremental sampling and smoothing (iSAM) algorithm [77], which is a popular SLAM back-end that exploits sparse QR factorizations to achieve online capabilities. In their experiments, their SLAM module provides pose updates approximately every 2 seconds, however, all mapping was performed off-board. Nonetheless, their MAV platform successfully creates 2D maps autonomously from laser range measurements in indoors and urban scenes, without any previous information about the environment.

Further interesting results were obtained by Grzonka et al. in [21]. They similarly rely on their scan matching algorithm to construct the pose graph and detect loop closures. Graph optimization is instead carried out with stochastic gradient descent, as described in [80], which performs gradient descent on individual randomly selected constraints, achieving quick convergence and increased robustness to local minima. In their experiments, they are capable of mapping an indoor scene with multiple nested loops. While SLAM is also performed in 2D, they provide an extension to multi-floor SLAM,

as described in Sec. 1.2.2.1. Mapping is also performed off-board, and no details on the computation time of mapping are given.

Graph-based SLAM using exclusively on-board processing was achieved by Shen et al. [20]. They rely on an Iterated Extended Kalman Filter (IEKF) to optimize the pose graph. One particularity of their work is that the optimization is carried out on the MAV's complete 6 DoF. As a result, their SLAM module directly provides multi-floor maps, which was validated experimentally for a large-scale indoor scene. However, they require a camera for loop closure detection, by extracting and matching SURF features from the images.

1.4.3 Fusing relative and absolute state measurements

The interest is now to derive the velocity estimates necessary for the control loop, which, as for local state estimation (Sec. 1.2), are obtained through sensor fusion techniques. The SLAM module described in the previous section provides absolute state measurements that can be exploited for this purpose. However, it is important to note that SLAM algorithms require extensive computational resources. On MAVs, they are commonly performed off-board [19, 21] and only a handful of studies achieve on-board capabilities, but at very low rates (2-10Hz) [20, 22]. Therefore, in this case, real-time pose and velocity estimates can't be recovered from SLAM and a stable feedback position control can't be achieved from SLAM alone.

In Sec. 1.3.1 it was discussed how pose and velocity estimates can be recovered through sensor fusion in a local scale from relative state measurements (e.g., laser odometry, visual odometry). Despite providing the necessary real-time capabilities for a stable control loop, prolonged flights lead to drift in the state estimates when sensor fusion relies uniquely on relative state measurements. Properly dealing with drift issues requires corrections from absolute state measurements, such as those provided by SLAM. By fusing relative and absolute measurements it is possible to leverage their respective advantages. On the one hand, this can provide the real-time capabilities required for achieving a stable and robust control. On the other hand, this also brings the global consistency required for navigating safely in large-scale environments.

Applications on MAVs

Absolute and relative state information can be combined in a simple way by performing SLAM and sensor fusion separately. On one side, relative state measurements are fused as described in Sec. 1.3.1 to obtain real-time local state estimates. Then, the slower SLAM process is used to provide periodic corrections of the state estimates, without directly introducing the SLAM module in the fusion filter. After each correction, the subsequent relative state measurements are simply computed with respect to the corrected states, resulting in globally consistent real-time estimates. This simple and intuitive approach was tested on a MAV equipped with a 2D LiDAR, an IMU and a laser altimeter [19, 75], obtaining a stable position control and achieving complete autonomy in several indoors and outdoors flights (up to 745 m in an urban canyon).

On the other hand, Shen et al. [20] instead fuse absolute and relative state information in a filtering framework consisting in a cascade of two EKFs. The first filter is designed to smooth the SLAM estimates and provide initial compensation for large delays, by combining the pose tracking and SLAM outputs. This first stage provides a 20 Hz pose estimate, which is then fused by the second EKF with inertial measurements, to recover linear velocity and filtered 6 DoF pose estimates at 100 Hz that are directly fed to the control loop. While their platform manages to navigate autonomously, this work is restricted to indoor scenes due to the underlying assumptions of their ICP-based 2D laser odometry (described in Sec. 1.2.1.1).

1.4.3.1 Recent developments on MAV multi-sensor fusion

The main drawback of the previously mentioned works is that they do not leverage the possibility of using multiple sources of absolute state measurements for more robustness and versatility. Certain scenarios may cause individual sensors to be unreliable. On the one hand, GPS sensors fail in indoor environments and suffer constant outages in urban scenes. On the other hand, 2D laser scanners provide unreliable information in highly complex 3D environments. For example, in [19] their platform was unable to operate in areas with dense vegetation and no additional structure, or featureless environments, such as wide open spaces and long corridors.

Thus, navigating freely in unconstrained large-scale environments remains a complex issue [19]. Recent developments on multi-sensor fusion seek to overcome this problem by combining multiple remote sensing modalities [58, 63, 81, 82].

Filtering techniques. In Sec. 1.3.1, the Stochastic Cloning Kalman filter (SC-KF) was introduced as an optimal way of incorporating relative state measurements in a Kalman filter framework. While this efficiently reduces drift in the state estimates [61], this formulation doesn't incorporate the absolute state measurements necessary for eliminating drift in large-scale flights. Several recent works propose extensions of the SC-KF that address this, such as the Multi-sensor Fusion EKF (MSF-EKF) [58] and an Unscented Kalman Filter (UKF) formulation [63]. These frameworks also allow handling an arbitrary number of inputs, in a loosely coupled and computationally efficient way. On the one hand, Lynen et al. test their MSF-EKF formulation on a MAV platform equipped with a GPS sensor, visual SLAM, an IMU and a pressure sensor, in a large-scale outdoors scene [58]. On the other hand, Shen et al. test their UKF formulation on a MAV platform relying on vision and laser odometry, a GPS sensor, an IMU and a barometer in a large-scale industrial complex with multiple indoors and outdoors scenarios, including wide open spaces and densely vegetated areas [63]. In both cases, the MAVs demonstrated large-scale flight capabilities in complete autonomy and all computation was performed on-board.

Filtering frameworks owe their popularity in sensor fusion to their capability of processing data with minimal latency. Standard approaches achieve this by limiting operations to the most recent states. This, however, comes at a loss of information and at the cost of estimation quality. Moreover, this proves a major set-back when handling multi-sensor systems, as measurements typically arrive at different rates and can experience significant delays. This issue is formally known as *out-of-sequence measurements* [81], and requires a special treatment in filtering techniques. These are often approximate solutions, such as extrapolations [83] or maintaining a buffer of past states and measurements [58, 63], which do not incorporate delayed measurements optimally.

Smoothing techniques. An alternative approach to sensor fusion relies instead on *smoothing* techniques, which maintain a complete history of states and measurements and treat sensor fusion as a non-linear optimization problem [81,82]. By formulating this problem in terms of a graph representation, such as factor graphs [82], new sensors can be incorporated in a simple and intuitive way, and no special considerations are required for out-of-sequence measurements or relative state measurements, which were problematic for recursive filters (Sec. 1.3.1). Moreover, this representation allows extending graph optimization techniques from graph-based SLAM algorithms (described in Sec. 1.4.2.2) to sensor fusion [81, 82], allowing to handle non-linear systems efficiently.

Maintaining a complete history of states and measurements leads to an increasingly

large optimization problem not suitable for online applications. However, recent developments in *incremental smoothing* techniques for SLAM [77], which only optimize the part of the graph affected by new measurements, have lead to similar applications in sensor fusion problems, with real-time capabilities [82]. Further validations have demonstrated their applicability on MAV platforms on simulations [82, 84] and experimental tests indoors [84]. Despite showing similar levels of accuracy when compared against a standard EKF in several scenarios [84], incremental smoothing was better capable of handling non-linear systems than the EKF [82]. Another example includes [85], where smoothing is used to fuse measurements from a GPS sensor, a stereo camera and an IMU, for autonomous large-scale river mapping with a MAV platform [85]. While current applications to MAVs have been limited, the benefits of smoothing techniques and the graph formulation of sensor fusion are of great practical interest and a promising tool for future research on multi-sensor fusion in MAVs.

1.4.4 Achieving high-level navigation tasks

To summarise, the previous sections discuss different tools required for localization in real-time (Sec. 1.2) and in large-scale environments (Sec. 1.4.1 and Sec. 1.4.2). Together with control, this is sufficient for basic autonomous waypoint navigation. However, a robot can approach a desired location in many different ways, and obstacles can be present in the environment that must be detected and avoided. A completely autonomous platform must be capable of making high-level decisions about how to reach a desired location in an efficient and safe way. This requires performing tasks such as real-time obstacle avoidance and trajectory planning. This section presents a non-exhaustive list of high-level navigation tasks that have been achieved on-board MAVs equipped with 2D laser scanners. For a more complete survey, including other remote sensing capabilities, the reader is referred to [1].

Obstacle avoidance. Detecting and avoiding nearby obstacles is one of the basic functionalities that any autonomous MAV must have to navigate safely. Laser scanners naturally provide the capability of achieving this by directly measuring the distance to surrounding objects. However, proper obstacle avoidance requires a broad perception of the surroundings, and 2D laser scanners have a limited field of view. Several recent works have aimed at using actuated light-weight 2D LiDARs, which are constantly rotated about a fixed axis to assemble 3D point clouds from the 2D measurements [85–88]. Other works use multiple sensing modalities to complement laser range sensing, such as Kinect

sensors [89], stereo cameras and ultrasound sensors [85, 88]. The latter allow detecting transparent surfaces, that can't be detected by laser beams. While obstacle avoidance is often embedded in path planning techniques, fast reactive approaches, which act directly from the available sensor inputs, are also commonly used. On MAVs, these are often variants of artificial potential fields [21, 86–88].

Path planning. This step consists in determining a collision-free trajectory between an initial and a goal location (waypoint), given a map of the scene (e.g., from SLAM). To reduce planning complexity, multi-layered planning approaches are often adopted [86–88]. On a higher layer, a global planner computes a globally consistent path from the current location to the next waypoint. This path is cost-optimal with respect to the current map and takes into account known obstacles. However, previously unknown obstacles (dynamic and static) not reflected in an outdated map can be present in the scene. To react to such situations, a lower layer (at faster rates) contains a local planner that refines the global path based on the current local perception of the environment. On the lowest layer, a fast reactive obstacle avoidance module is often used for additional safety. A survey of well-known motion planning techniques for MAVs is provided in [90]. Several techniques used on MAVs equipped with LiDARs include D* lite [21], A* [86–88], Rapidly-exploring Random Trees-Star (RRT*) [89] and, more recently, Sparse Tangential Networks (SPARTAN) [91].

Exploration. Exploration characterizes a completely autonomous platform and relies on all the previously discussed tasks. This task concerns applications in which a prior map of the environment is not available and the robot must determine, based on its on-board sensing, effective ways of navigating in complete autonomy. The idea of an exploration algorithm is to generate a set of goal points from the local perception of the environment, that will enable a robot to travel an unknown region. These goals are then used as an input to a local motion planner which will generate the trajectories for the robot. On MAVs, exploration algorithms are often inspired on the well-known frontier-based approach, where the goals are placed on boundaries between known and unknown areas [19, 89, 92–94]. Different strategies seek to find a balance between generating goals that will guide the MAV towards unexplored areas, and goals that will provide sufficient sensor information to keep the platform well-localized.

Previous works on MAV exploration equipped with LiDARs include [19, 89, 93, 94]. In [19], Bachrach et al. present an autonomous MAV platform for indoor scenes and urban

canyons. Shen et al. demonstrated completely autonomous capabilities for multi-floor indoor environments [89]. More recently, Nuske et al. [93] developed a platform capable of exploring and mapping rivers in complete autonomy, without using a GPS sensor. Yoder et al. [94] present a platform capable of building 3D models of infrastructure autonomously, where the only user input is a 3D bounding box around the structure, and present results on a train bridge. In these works, no human interaction or previous knowledge of the scene were required. Most of these studies couple LiDARs with additional sensing modalities to aid perception and navigation, such as GPS sensors, optic flow, stereo cameras and ultrasound sensors [89, 92–94]. Moreover, [89, 92–94] perform all computation on-board.

1.5 Conclusions

This chapters has addressed numerous topics related to MAV autonomous navigation. Pose tracking, as discussed in Sec. 1.2, is the main topic of Chapters 2-4. In particular, most of the discussions in these chapters are dedicated to developing efficient laser odometry techniques, the main focus of this thesis, for tackling difficulties that arise from a MAV's 3D motion and the structure of the surrounding environment. For this purpose, different scenarios have been studied, which include a typical indoor scene (Chapter 2) and a more complex electric tower inspection scene (Chapters 3-4). Then, sensor fusion and feedback control, which was not reviewed in this chapter, are briefly discussed in Chapter 5. Due to time constraints, complex sensor fusion strategies for small-scale (Sec. 1.3.1) and large-scale environments (Sec. 1.4.3) could not be considered throughout this work. These topics, together with high-level tasks such as global localization and SLAM, discussed in Sec. 1.4, remain subjects of future continuations of this work.

CHAPTER 2

Laser-based pose estimation for planar environments

Contents

2.1 Overview **26**

2.2 The planar environment assumption **27**

2.3 The Iterative Closest Point (ICP) algorithm **28**

2.3.1 The Metric-based Iterative Closest Point (MbICP) algorithm . 30

2.3.2 Limitations of ICP-based approaches for state estimation . . . 32

2.4 Proposed pose tracking approach **32**

2.4.1 2D laser odometry 32

2.4.2 Attitude estimation 34

2.4.3 Height estimation 36

2.5 Experimental results **36**

2.6 Conclusions **42**

2.1 Overview

In this chapter, we are interested in studying laser-based pose estimation in environments composed of planar, vertically flat objects. This situation is often faced in numerous tasks, such as surveillance, search and rescue, inspection, among others. Typical scenarios include urban and indoor scenes, which are the most commonly studied case in

literature [19–22, 95]. However, these are typically GPS-denied environments and navigating autonomously can be challenging for MAVs due to the lack of a direct source of location information, which must instead be inferred from range sensing. As discussed in Sec. 1.2, we are interested in 2D laser scanners and scan registration techniques for this purpose (i.e., 2D laser odometry). Moreover, our goal is achieving real-time capabilities relying uniquely in on-board processing power.

Scan registration methods that have been successful for achieving on-board real-time capabilities on MAVs were discussed in Sec. 1.2.1. This included variants of the Iterative Closest Point (ICP) algorithm [34,37] and the Correlative Scan Matcher (CSM) [43]. Probabilistic methods, such as CSM, have clear advantages with respect to deterministic approaches in terms of robustness to noise, outliers and initialization errors. However, this comes in exchange for a higher computational cost. This is evident when comparing the experimental results from [19] to [20, 22], where, despite using the same on-board processing capabilities, the platforms that relied on ICP-based methods were capable of achieving more high level tasks on-board, such as mapping and path planning, without requiring the use of a ground station for additional computations. Based on these observations, and on their simplicity and flexibility, the ICP-based approaches were chosen for the 2D scan registrations.

This chapter is organized as follows: In Sec. 2.2, we discuss the implications of the MAV's 3D motion in the 2D laser scan registrations and how this is handled in planar environments. Then, Sec. 2.3 gives a brief overview of the ICP algorithm and common considerations to improve convergence rates and accuracy. The preferred variant, the Metric-based ICP (MbICP) algorithm, is discussed in Sec. 2.3.1 and additional considerations for a practical implementation of this technique on-board a MAV are presented in Sec. 2.4.1. While the main focus of this chapter is 2D laser odometry, we present a complete 6 DoF pose tracking approach in real-time and on-board with our sensor setup. Thus, attitude and height estimation are also briefly discussed in Sec 2.4.2 and Sec. 2.4.3, respectively. Finally, experimental validations for indoor flights on a quadrotor platform are presented in Sec. 2.5.

2.2 The planar environment assumption

In pair-wise scan registrations, a basic criteria to achieve adequate performance is that the scans must have sufficient overlap. Range measurements obtained from a 2D laser scanner can only overlap if they are taken within the same plane. This doesn't pose

a major problem for ground robots, as they typically move on flat grounds and have a planar motion. However, this poses constraints on aerial platforms, which navigate in 3D environments. Here, laser scans can measure different surfaces at different heights. Moreover, changes in the attitude, notably the roll and pitch angles, directly affect the range measurements. In this case, directly aligning pairs of scans will lead to false correspondences and poor performances. Hence, additional steps must be taken to account for the MAV's 3D motion.

As previously mentioned, laser range sensors are mostly used in cluttered, GPS-denied environments. This includes indoors and urban scenes that are mostly composed of straight walls. In these situations, it is possible to assume that surrounding surfaces are planar and invariant to height. In a first instance, this allows ignoring in the scan registrations the altitude component of the aerial robot's motion, which is instead estimated from separate sensing, e.g., laser altimeter (Sec. 1.2.2.1) or pressure sensors (Sec. 1.2.2.2). Then, horizontal displacements on MAVs require to constantly tilt about the roll and pitch angles, which causes the scan plane to change accordingly if the sensor is rigidly attached to the platform. A solution, commonly adopted in literature, is to exploit the roll and pitch angles estimated from IMU measurements. These angles are used to project orthogonally the 2D laser endpoints to a common horizontal plane. The projected scans can then be aligned with any of the methods previously described in Sec. 1.2.1 to recover the 2D pose. This simple solution has proved effective in practical experiences in numerous works [20–23], and will be adopted in this chapter for the laser scan registrations.

2.3 The Iterative Closest Point (ICP) algorithm

The Iterative Closest Point (ICP) algorithm [34] is the most extensively used registration technique for aligning three-dimensional shapes relying on geometry. Its popularity comes from its simple and generalized formulation that can be extended to solve problems from different fields such as pattern recognition, medical imagery and photogrammetry. The idea of an ICP-based solution for robotics problems was pioneered by Zhang et al. [96] for object recognition and visual navigation, and later popularized by Lu and Milios [30] for pose tracking from 2D range scans. This has opened the way to numerous successful applications in mobile robot navigation, where ICP has been used mostly for aligning laser range data. With the continued progress of LiDAR technologies and the development of reliable open source libraries such as Libpointmatcher [97] and the registration module from the Point Cloud Library (PCL) [98], both in C++, ICP remains a popular

registration method in robotics applications.

When used for pair-wise laser scan registrations, ICP starts with a *reference* scan, the *current* scan and a rough guess of their relative rigid body transformation (initial alignment). Then, the algorithm seeks to align the current scan with the reference scan. The procedure can be summarized in two steps:

1. **Matching:** Establishing point correspondences between the scans with a given *association criterion*.
2. **Minimization:** Computing the rigid body transformation that minimizes the alignment error (sum of squared distances) for a given *distance metric*.

In the baseline ICP [34], the matching step establishes associations based on the *closest-point rule* [30], i.e., for each point in the current scan, the corresponding point in the reference scan is the one within the shortest Euclidean distance. This nearest-neighbour search is generally the most time-consuming step of the algorithm. A common approach is to store the reference scan in a K-D tree, a space partitioning structure for organizing points that greatly reduces search time [37]. Then, in the minimization step, the baseline ICP considers a *point-to-point metric* (Euclidean distance) [34]. Several closed-form solutions to this optimization problem are listed in [37], including orthonormal matrices, dual quaternions, and SVD-based approaches, which provide similar levels of accuracy and stability.

The key concept of ICP is that even with imperfect initial associations, minimizing the alignment error results in better estimates that consequently allows for better associations. By repeating these two steps iteratively, Besl et al. [34] demonstrate that the algorithm converges monotonically to a local minimum. Whether this solution is close or not to the global minimum depends on several factors, such as a good initial guess, sensor noise and the input geometry.

Efficient ICP variants

The previous description characterizes the basic formulation of the ICP algorithm. Common issues with this approach include slow convergence rates and a tendency to get trapped in local minima. Different strategies can be adopted to improve the convergence behaviour and accuracy of the registrations. Here we briefly highlight some common

considerations between variants of the ICP algorithm. A complete overview is given in [36, 37].

Point selection (sampling). The computational cost of the correspondence search scales rapidly with the size of the data sets. To cope with this, a strategy consists in reducing the size of one or both scans, by *selecting* a subset of points for the registration process. Simple approaches consist in random or uniform sampling [37], but risk degrading accuracy as small features that provide relevant information can be left out. More complex approaches, such as [99] and normal space sampling [37], take into account the underlying geometry and select points that contain useful information to constrain the transformations, but consider the input data to be in the form of a mesh. In robotics applications with 2D lidars, the data is commonly sparse point clouds. In such cases, sampling is often ignored, using the raw point clouds instead [100].

Association criterion. Point correspondences drawn from closest-point rule with the Euclidean distance capture limited rotation information [30]. Alternative association criteria, such as the matching-range rule [30] and the metric from the Metric-based ICP (MbICP) variant [101, 102], seek to account for this considering heuristics or distance metrics that better reflect rotational motion. Other popular association criteria include normal shooting and the closest compatible point [37].

Handling outliers. In certain cases, it is possible for the matching step to establish incorrect correspondences (outliers). Since laser scans captured from a mobile robot only overlap partially, this is often the result of drawing point pairs from non-overlapping (occluded) areas. Other common sources of outliers include sensor noise or moving objects. Introducing these outliers in the minimization step can have a large impact in the stability and accuracy of the algorithm. Therefore, it is essential to perform a *rejection* step after the correspondence search to eliminate possible outliers. In general, point pairs that are far from each other, especially in later iterations of the algorithm, are likely to be incorrect correspondences. Common strategies include setting maximum distance thresholds or keeping a percentage of the best alignments [37]. More complex strategies based on robust statistics are presented in [36]. An alternative approach of dealing with outliers relies on *weighting* corresponding pairs. These weights are designed to reduce influence of erroneous point pairs using different criteria such as the distance between points or the orientation of the normals [37]. In either case, the effectiveness of the approach is highly data-dependent.

Distance metric. In the baseline ICP, the point-to-point metric is another cause of slow converges rates [38]. This occurs in part since the only source of information are the point correspondences and the constraints introduced by different pairs can be incompatible [103]. While this metric remains useful in unstructured scenarios [36], better results can be in obtained structured scenes by considering higher level information about the local structure around the points, such as normals or the curvature [38]. A popular alternative is the point-to-plane metric [103], which minimizes the distance between a point and the tangent plane at its corresponding point. This is in fact a generalization of the point-to-line metric for 2D registrations, for which a closed form solution was proposed in the well-known Point-to-line ICP (PlICP) variant [33]. Furthermore, Mitra et al. [38] propose a point-to-surface distance approximation which generalizes the point-to-point and point-to-plane metrics. All these metrics exhibit faster quadratic convergence rates and are less prone to getting trapped in wrong local minima [33, 38, 104]. On the other hand, the previously mentioned MbICP metric, which explicitly takes into account a rotational component in its formulation, can also be used in the minimization step [101, 102]. Experimental validations show that this metric leads to more robust and precise results than the point-to-point metric, specially to large rotation errors [101].

Optimization solver. The closed-form solutions to the minimization step mentioned in Sec. 2.3 are specific to the Euclidean distance metric. More recently, there has been increasing interest in using non-linear solvers, such as Gauss-Newton [104] and Levenberg-Marquardt [104, 105], which allow minimizing more generic error metrics, such as the point-to-plane metric [37], and have a wider basin of convergence that allows reducing the dependence on the initial guess [105].

Preferred ICP variant

It has been generally noted that the MbICP algorithm is one of the more robust ICP variants, in particular to large rotation errors [33, 101, 102]. The MbICP has a good trade-off between simplicity and accuracy that isn't possible with most approaches. The main drawback is that typical methods for speeding the correspondence search, such as K-D trees, can't be applied due to the distance metric used [106]. However, optimizations to the correspondence search could lead to promising performance results. We now present a summary of the MbICP algorithm [101].

2.3.1 The Metric-based Iterative Closest Point (MbICP) algorithm

As previously explained, the baseline ICP algorithm matches the closest points based on the Euclidean distance. However, separations between pairs of points can also be caused by rotational motion, an effect that is not clearly explained by the Euclidean distance. Therefore, point correspondences obtained in the baseline ICP algorithm capture limited rotation information. In [101], Minguez et al. present the Metric-based ICP (MbICP) algorithm, which relies on a distance metric that simultaneously accounts for translational and rotational displacements. The distance d_p between two points $\mathbf{p_1}$ and $\mathbf{p_2}$ in $\mathbb{R}^2$ is defined as follows

$$d_p(\mathbf{p_1}, \mathbf{p_2}) = \min_{\mathbf{X}} \sqrt{x^2 + y^2 + L^2\psi^2}, \quad \text{such that } T_{\mathbf{X}}(\mathbf{p_1}) = \mathbf{p_2}, \tag{2.1}$$

where L is a positive real number that acts as a weighing factor between rotation and translation, ψ denotes the yaw angle and $\mathbf{X} = (x, y, \psi)$ such that $T_{\mathbf{X}}$ is a 2D rigid body transformation as

$$T_{\mathbf{X}}(\mathbf{p}) = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \mathbf{p} + \begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathbf{p} \in \mathbb{R}^2, \tag{2.2}$$

A closed-form expression for Eq. (2.1) can be derived by considering small rotations. Thus, an approximate distance d_p^{ap} is obtained by linearising about $\psi = 0$, resulting in [101]

$$d_p^{ap}(\mathbf{p_1}, \mathbf{p_2}) = \sqrt{\|\mathbf{p_1} - \mathbf{p_2}\|^2 - \frac{\|\mathbf{p_1} \times (\mathbf{p_1} - \mathbf{p_2})\|^2}{\|\mathbf{p_1}\|^2 + L^2}}. \tag{2.3}$$

In MbICP, this distance metric is used to establish point correspondences and in the minimization step.

Let us denote by $\mathcal{I}$ an inertial NED (North-East-Down) frame and by $\mathcal{B}$ the body frame attached to the MAV's center of mass. For simplicity, in the following discussions we consider that the sensor frames (IMU, 2D laser scanner and laser altimeter) coincide with $\mathcal{B}$. Let $\mathbf{S}_p = \{\mathbf{p}_i \mid i = 1, \cdots, N_p\}$ denote the current 2D scan, expressed in $\mathcal{B}$. Then, $\mathbf{S}_q = \{\mathbf{q}_j \mid j = 1, \cdots, N_q\}$ denotes the 2D reference scan expressed in $\mathcal{I}$. Given $T_{\mathbf{X_0}}$, an initial rough estimate of the 2D rigid body transformation from $\mathcal{B}$ to $\mathcal{I}$, such that $\mathbf{X_0} = (x_0, y_0, \psi_0)$, the goal is to refine this guess by aligning $\mathbf{S_p}$ to $\mathbf{S_q}$. Hence, each iteration k of the MbICP algorithm is performed as follows

1. **Initialization:** The current estimate $T_{\mathbf{X_k}}$ is used to transform all points $\mathbf{p_i} \in \mathbf{S_p}$ into the reference frame of $\mathbf{S_q}$, obtaining the transformed points $\mathbf{p'_i} \in \mathbf{S_{p'}}$.

2. **Matching:** In [101], to account for the discrete nature of the laser measurements, for each $\mathbf{q_j} \in \mathbf{S_q}$, consecutive points $[\mathbf{q_j}\ \mathbf{q_{j+1}}]$ are assumed to be joined by a line segment. Then, following Eq. (2.1), let d_{ps} denote the distance between a point $\mathbf{p'_i}$ to the line segment delimited by $[\mathbf{q_j}\ \mathbf{q_{j+1}}]$ as [101]

$$d_{ps}(\mathbf{p'_i}, [\mathbf{q_j}\ \mathbf{q_{j+1}}]) = \min_{\lambda \in [0,1]} d_p(\mathbf{p'_i}, \mathbf{q_j} + \lambda(\mathbf{q_{j+1}} - \mathbf{q_j})). \tag{2.4}$$

   where λ is a factor that allows interpolating between $\mathbf{q_j}$ and $\mathbf{q_{j+1}}$. Next, solving this equation for λ allows finding the closest point $\mathbf{q}_*$ from $[\mathbf{q_j}\ \mathbf{q_{j+1}}]$ to $\mathbf{p'_i}$, consistent with Eq. (2.1), as [101]

$$\mathbf{q}_* = \begin{cases} \mathbf{q_j} & \text{if } \lambda < 0 \\ \mathbf{q_j} + \lambda(\mathbf{q_{j+1}} - \mathbf{q_j}) & \text{if } 0 \leq \lambda \leq 1 \\ \mathbf{q_{j+1}} & \text{if } \lambda > 1 \end{cases} \tag{2.5}$$

   For each $\mathbf{p'_i} \in \mathbf{S_{p'}}$, this is repeated for every consecutive point pair in $\mathbf{S_q}$, and the point $\mathbf{q}_*$ which yields the smallest distance is chosen as the corresponding point.

3. **Minimization:** The goal is to find the transformation $T_{\mathbf{X}_{\min}}$ that minimizes the sum of squared errors, using d_p^{ap} (Eq. (2.3)) as the distance metric. For the N point pairs $(\mathbf{p'_i}, \mathbf{q_i})$, this leads to the following least squares problem

$$\mathbf{X}_{\min} = \arg\min_{\mathbf{X}} \sum_{i=1}^{N} d_p^{ap}(T_{\mathbf{X}}(\mathbf{p'_i}), \mathbf{q_i})^2. \tag{2.6}$$

   A closed-form solution is presented in [101].

4. Finally, the current estimate is updated as

$$T_{\mathbf{X}_{k+1}} = T_{\mathbf{X}_{\min}} \cdot T_{\mathbf{X}_k} \tag{2.7}$$

Upon convergence, the result is an updated $\{x, y, \psi\}$. Note that this formulation is for point cloud registrations in $\mathbb{R}^2$. More recently, a generalization of this algorithm to $\mathbb{R}^3$ was presented in [102].

2.3.2 Limitations of ICP-based approaches for state estimation

Scan registration techniques as a source of odometry were discussed in Sec. 1.2.1. It was noted that deterministic approaches, such as ICP, require a precise initial guess and typically cannot recover from large estimation errors. These limitations are a direct consequence of the convergence behavior of ICP, which only guarantees convergence to a local minimum. While globally optimal solutions for the ICP algorithm have been studied in the past [107], they are typically too slow for state estimation purposes. Instead, as discussed in Sec. 1.4.2, simultaneous localization and mapping (SLAM) techniques provide pose estimates with guaranteed global consistency, that are less sensitive to initialization errors [19–22,95]. Thus, in navigation tasks, dealing with these convergence-related issues requires coupling laser odometry and SLAM, which is done effectively through sensor fusion techniques (Sec. 1.4.3).

Other limitations are related to the type of remote sensing used. First, using a 2D laser scanner required the planar environment assumption in order to account for the MAV's 3D motion and properly register the laser scans. As was seen from previous works in Sec. 1.2.1, this assumption restricts registration methods like ICP to structured scenarios, such as indoors scenes. Second, in certain scenarios the laser scans fail to capture sufficient geometric detail in order to extract any useful pose information. The ICP algorithm will thus fail under highly unstructured scenarios, often faced outdoors, or featureless scenarios, such as long hallways or circular rooms. As was discussed in Sec. 1.4.3, handling this issue requires incorporating multiple sensing modalities, such as GPS sensors, ultrasonic sensors and cameras [85,88,89,92–94]

2.4 Proposed pose tracking approach

In this part of our work, the main focus was recovering the 2D pose from the laser scan registrations. Nonetheless, in this section we present an approach to track the complete 6 DoF pose of a MAV in planar environments using our sensor setup. Recalling that the complete 6 DoF pose from the body frame $\mathcal{B}$ to the inertial frame $\mathcal{I}$ is described by $\{x, y, z, \phi, \theta, \psi\}$, and following the modular scheme presented in Sec. 1.2.1, the estimation process is broken down into three components: the 2D pose $\{x, y, \psi\}$ is obtained from a 2D laser odometry based on the MbICP algorithm (Sec. 2.4.1); then, the roll and pitch angles $\{\phi, \theta\}$ are recovered from IMU (accelerometer and gyrometer) measurements (Sec. 2.4.2); lastly, the height $\{z\}$ is estimated from the laser altimeter (Sec. 2.4.3). The following

subsections describe each component of our proposed pose tracking approach.

2.4.1 2D laser odometry

The MbICP algorithm was chosen for the 2D laser odometry. In this section, we present how the standard algorithm described in Sec. 2.3.1 was adapted to account for the MAV's 3D motion and to achieve real-time capabilities on-board. The output of this procedure is $\{x_{\text{laser}}, y_{\text{laser}}, \psi_{\text{laser}}\}$, an estimate of the MAV's 2D pose.

Keyframe-based registrations. A classic implementation of the ICP algorithm in navigation tasks consists in aligning the current laser scan to the immediate preceding scan. This is known as *incremental scan matching*, and rapidly leads to drift over time as the estimation errors accumulate without bound [19, 20, 22]. An alternative is to use a *keyframe* approach, similar to [22]. Here, a reference scan is instead fixed at some initial time, and is only updated when the robot travels beyond a fixed radius or if there isn't sufficient overlap with the incoming laser scans. While the keyframe remains unchanged, the estimation errors remain bounded and the registrations are drift-free [22]. In this work, we use a single keyframe for the scan registrations.

Horizontal projection. As explained in Sec. 2.2, the MAV's 3D motion is accounted for by assuming that surrounding objects are planar and invariant with height. This is done by projecting orthogonally the laser endpoints to a common horizontal plane as in [20–23]. Let $\mathbf{S}_{\mathbf{p}^*} = \{\mathbf{p}_\mathbf{i}^* = (p_x^*, p_y^*, 0)^\intercal \quad | \quad i = 1, \cdots, N_p\}$ denote the current *raw* 2D range measurements, expressed in $\mathcal{B}$. Then, considering the roll and pitch angles estimated from the IMU measurements ($\{\phi_{\text{imu}}, \theta_{\text{imu}}\}$ as will be explained in Sec. 2.4.2), the point set $\mathbf{S}_\mathbf{p}$ from Sec. 2.3.1 is in reality the projected laser scan, and each point $\mathbf{p}_\mathbf{i} \in \mathbf{S}_\mathbf{p}$ is obtained as

$$\mathbf{p}_\mathbf{i} = R_x(\phi_{\text{imu}}) R_y(\theta_{\text{imu}}) \mathbf{p}_\mathbf{i}^*, \quad i = 1, \cdots, N_p, \tag{2.8}$$

and setting the third coordinate of each $\mathbf{p}_\mathbf{i}$ to zero.

Adaptive breakpoint detector. In the correspondence search of the MbICP algorithm, consecutive points in the reference scan $\mathbf{S}_\mathbf{q}$ are considered to be joined by line segments. Therefore, point pairs with large discontinuities, or *breakpoints*, in $\mathbf{S}_\mathbf{q}$ have to be identified in order to avoid pairing points that measure different objects. As in [102], we rely on the adaptive breakpoint detector from [108] for this purpose. This algorithm

determines distance thresholds that adapt to the range scan distances to identify large gaps between points and to segment the scan into sets of continuous points. Identifying breakpoints also allows detecting and removing false measurements that occur when a laser beam falls on the edge of an object (also known as the *mixed pixels problem* [29]), which can degrade de quality of the estimates.

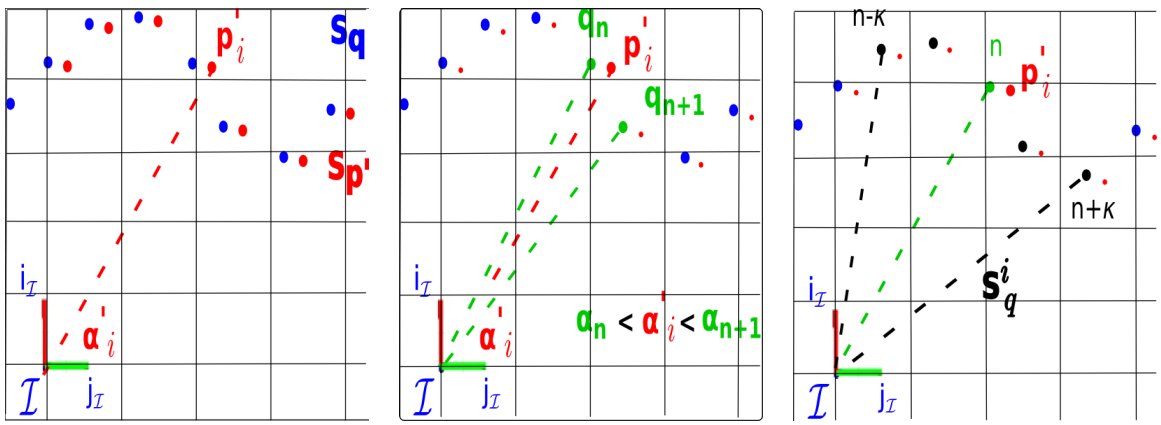
Improving the correspondence search. The correspondence search is the most time-consuming step and a careful implementation can lead to significant registration speed gains. Assuming that the initial guess $T_{\mathbf{X_0}}$ is precise, corresponding points between the transformed set $\mathbf{S_{p'}}$ and the reference scan $\mathbf{S_q}$ will have similar bearing angles. This is not a strong assumption, since the relative displacement between scans is small for high measurement frequencies. Then, a simple procedure based on limiting the search region considering the bearing angle is illustrated in Fig. 2.1. Here, after initialization (step 1 from Sec. 2.3.1), a subset of points $\mathbf{S^i_q}$ is extracted for each $\mathbf{p'_i} \in \mathbf{S_{p'}}$ as follows

1. The bearing angle α'_i of $\mathbf{p'_i}$ is calculated (Fig. 2.1a).

2. The segment $[\mathbf{q_{n_i}}\ \mathbf{q_{n_i+1}}]$ in $\mathbf{S_q}$ that contains $\mathbf{p'_i}$ is found, such that $\alpha_{n_i} < \alpha'_i < \alpha_{n_i+1}$ (Fig. 2.1b). This can be achieved efficiently if the reference scan is stored is an ascending angular order. Then, instead of searching through $\mathbf{S_q}$, the index n_i can be directly calculated as
$$n_i = \lfloor \frac{\alpha'_i - \alpha_0}{\Delta\alpha_{laser}} \rfloor, \tag{2.9}$$
where $\lfloor \cdot \rfloor$ is the floor function, $\Delta\alpha_{laser}$ is the known angular precision of the laser scanner, and α_0 is the bearing angle of the first point $\mathbf{q_0} \in \mathbf{S_q}$.

3. The subset of points $\mathbf{S^i_q}$ is recovered as
$$\mathbf{S^i_q} = \{\mathbf{q_j} \in \mathbf{S_q} \quad | \quad n_i - \kappa < j < n_i + \kappa\}, \tag{2.10}$$
where the parameter κ determines the bounds of the angular search window (Fig. 2.1c). This parameter depends on the sensor's angular precision and the angular speed between consecutive scans.

The extracted subset $\mathbf{S^i_q}$ is then used in the correspondence search for $\mathbf{p'_i}$ (step 2 of the MbICP algorithm) instead of $\mathbf{S_q}$.



(a) (b) (c)

Figure 2.1: Determining the search region for the correspondence search.

Outlier rejection. As discussed in Sec. 2.3, an outlier rejection strategy is required after the correspondence search to reduce the impact of incorrect matches in the minimization step. A simple rejection strategy was considered, where corresponding pairs separated by more than a fixed distance threshold $d_{\min}$ were removed.

2.4.2 Attitude estimation

We now present our proposed non-linear observer formulation using the accelerometer and gyroscope measurements. As yaw estimates are already obtained from the laser scan registration, the main goal is to recover estimates of the roll ϕ and pitch θ angles. First, let $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3)^\intercal$ denote the vertical axis of $\mathcal{I}$ expressed in $\mathcal{B}$ as

$$\boldsymbol{\gamma} = R^\intercal \boldsymbol{e_3} \tag{2.11}$$

with $\boldsymbol{e_3} = (0, 0, 1)^\intercal$. Using the Z-X-Y Euler angle convention, the rotation matrix R is expressed as

$$R(\psi, \phi, \theta) = R_z(\psi) R_x(\phi) R_y(\theta) = \begin{pmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\theta c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{pmatrix}. \tag{2.12}$$

From this rotation matrix definition, it follows that $\boldsymbol{\gamma}$ contains implicitly the MAV's roll and pitch angles, since

$$\begin{aligned} \phi &= \arcsin\left(\gamma_2\right) \\ \theta &= \operatorname{atan2}\left(-\gamma_1, \gamma_3\right). \end{aligned} \tag{2.13}$$

Recalling that a MAV's rotational kinematics is given by [25]

$$\dot{R} = RS(\boldsymbol{\omega}), \tag{2.14}$$

with $S(.)$ the skew-symmetric matrix associated with the cross-product (i.e., $S(\mathbf{x})\mathbf{y} = \mathbf{x} \times \mathbf{y}$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^3$), and $\boldsymbol{\omega}$ the angular velocity vector from $\mathcal{B}$ to $\mathcal{I}$, expressed in $\mathcal{B}$. Then, the kinematics of $\boldsymbol{\gamma}$ can be deduced from Eq. (2.11) and Eq. (2.14), and results in

$$\dot{\boldsymbol{\gamma}} = \boldsymbol{\gamma} \times \boldsymbol{\omega}. \tag{2.15}$$

This is the basis of our observer formulation. As previously mentioned, the goal is to recover roll and pitch estimates from the gyrometer and accelerometer readings. Let $\mathbf{a}_m$ denote the accelerometer measurements expressed in $\mathcal{B}$, which measure the specific acceleration acting on the MAV's airframe [25]

$$\mathbf{a}_m = R^{\intercal}(\dot{\boldsymbol{v}} - g\boldsymbol{e_3}) = R^{\intercal}\dot{\boldsymbol{v}} - g\boldsymbol{\gamma}. \tag{2.16}$$

Then, under the assumption of negligible linear acceleration, one has [54]

$$\mathbf{a}_m \approx -g\boldsymbol{\gamma}, \tag{2.17}$$

which shows that accelerometers provide direct observations of the roll and pitch angles (and of $\boldsymbol{\gamma}$). Thus, the following non-linear observer for $\boldsymbol{\gamma}$ is proposed

$$\dot{\hat{\boldsymbol{\gamma}}} = \hat{\boldsymbol{\gamma}} \times \left(\boldsymbol{\omega}_m - k_\gamma(\mathbf{a}_m \times \hat{\boldsymbol{\gamma}})\right), \quad k_\gamma > 0 \tag{2.18}$$

with $\boldsymbol{\omega}_m$ the angular velocities measured by the gyrometer in $\mathcal{B}$ and k_γ the positive scalar observer gain.

To analyse the stability of this estimator, consider the candidate Lyapunov function $L = 1 - \boldsymbol{\gamma}^T\hat{\boldsymbol{\gamma}}$. From Eq. (2.17) and Eq. (2.18) one has

$$\dot{\hat{\boldsymbol{\gamma}}} \approx \hat{\boldsymbol{\gamma}} \times (\boldsymbol{\omega}_m - k_\gamma g(\hat{\boldsymbol{\gamma}} \times \boldsymbol{\gamma})). \tag{2.19}$$

Then, assuming that this approximation of $\dot{\hat{\boldsymbol{\gamma}}}$ is perfect, and that $\boldsymbol{\omega}_m = \boldsymbol{\omega}$, it can be proven that $\dot{L} = -k_\gamma g \|\hat{\boldsymbol{\gamma}} \times \boldsymbol{\gamma}\|^2$, which is decreasing along the solutions of the system if, initially, $\hat{\boldsymbol{\gamma}}$ and $\boldsymbol{\gamma}$ are not opposite to each other, and $k_\gamma > 0$. This implies in particular the convergence of $\hat{\boldsymbol{\gamma}}$ to $\boldsymbol{\gamma}$.

Complete rotation matrix reconstruction.

The estimated roll ϕ_{imu} and pitch θ_{imu} angles are recovered from $\hat{\boldsymbol{\gamma}}$ and Eq. (2.13) as

$$\begin{aligned} \phi_{\text{imu}} &= \arcsin\left(\hat{\gamma}_2\right) \\ \theta_{\text{imu}} &= \operatorname{atan2}\left(-\hat{\gamma}_1, \hat{\gamma}_3\right). \end{aligned} \tag{2.20}$$

Finally, the complete estimated rotation matrix $\hat{R}$ is recovered by combining the estimated angles as

$$\hat{R} = R_z(\psi_{\text{laser}}) R_x(\phi_{\text{imu}}) R_y(\theta_{\text{imu}}), \tag{2.21}$$

where ψ_{laser} is obtained from the scan registrations, as described in Sec. 2.4.1.

2.4.3 Height estimation

A laser altimeter was considered for the height estimation. This sensor provides a single measurement h_m of the distance to the ground along the body-fixed vertical axis. This measurement is corrected with the estimated attitude in order to take into account the MAV's inclination. This is done by projecting the altimeter measurement on the estimated $\hat{\boldsymbol{\gamma}}$ described in the previous section, which yields the following estimate of the altitude

$$\overline{z} = h_m(\mathbf{e_3}^\intercal \cdot \hat{\boldsymbol{\gamma}}) = h_m \hat{\gamma}_3. \tag{2.22}$$

In addition, $\overline{z}$ is filtered through a second-order low-pass filter to reduce the effect of noise and ground irregularities. Considering a constant vertical velocity model, this yields

$$\begin{cases} \dot{\hat{z}} = & \hat{v}_z - k_z(\hat{z} - \overline{z}) \\ \dot{\hat{v}}_z = & -k_{v_z}(\hat{z} - \overline{z}), \quad k_z, k_{v_z} > 0, \end{cases} \tag{2.23}$$

where (k_z, k_{v_z}) are the positive scalar observer gains and $(\hat{z}, \hat{v}_z)$ are the estimated height and vertical velocity respectively.

(a)

(b)

Figure 2.2: The indoor experimental setup. **(a)** Quadrotor developed at ISIR, equipped with a Hokuyo URG-30LX 2D LiDAR, an MPU6000 3 axis accelerometer/gyrometer unit and an SF10/A laser altimeter from Lightware Optoelectronics. **(b)** The quadrotor follows a set of waypoints relying exclusively on the on-board estimates and position controller.

2.5 Experimental results

The three modules designed for tracking the MAV's 6DoF pose indoors, discussed in Sec. 2.4, were implemented and tested on the quadrotor platform from Fig. 2.2a. On-board computation was distributed between a "low-level" board, which was a Quantec Quanton flight controller card with an STM32 microcontroller, and a "high-level" board, an Odroid XU computer. The low-level board received data from an MPU6000 3 axis accelerometer/gyrometer IMU and computed the MAV's roll and pitch angles (see Sec. 2.4.2). An SF10/A laser altimeter from Lightware Optoelectronics, providing readings at 20 Hz, was also connected to this low-level card to estimate the MAV's height, as proposed in Sec. 2.4.3. Lastly, the high-level board received range readings from a Hokuyo URG-30LX

2D laser scanner at 40Hz, and performed all laser processing, as described in Sec. 2.4.1. The complete experimental setup is shown in Fig. 2.2b.

The test flights consisted in having the quadrotor follow a set of waypoints, which were chosen to account for different cases such as large translations (up to $2m$), large rotations (up to $30°$), and simultaneous translation and rotation, at a fixed height of $1m$ from the ground level. During the test flights, a motion capture (MoCap) system was used to keep track of the MAV's pose, providing a ground truth that was used for comparison purposes and to perform automatic take-off and landing. After take-off, a reference scan was set for the MbICP algorithm and the corresponding ground truth pose was used as an initial guess for the scan registrations. All subsequent scan registrations were initialized form the previous scan registration output. We now present the results for flights performed relying on the on-board state estimates and position controller. The control design and architecture will be discussed in detail in Chapter. 5. In this section we focus on validating the proposed 6 DoF pose tracking approach (Sec. 2.4).

2D laser odometry results

The 2D laser odometry was obtained from the MbICP algorithm, as described in Sec. 2.4.1. The parameter L from the MbICP distance metric (Eq. (2.1)) was set to $L = 3$, as proposed in [101, 102]. Then, the parameter κ from the proposed correspondence search (Eq. (2.10)) was chosen as $\kappa = 4$, and the outlier rejection threshold for the point correspondences was set to $d_{\min} = 10cm$. These last two parameters where determined experimentally to provide reasonable results. An example of a registered scan is shown in Fig. 2.3.

The laser odometry estimates are compared to the ground truth pose in Fig. 2.4, for the duration of the flight. The smallest absolute estimation errors for all states occur near the starting position (at $t = 0$ in Fig. 2.4d-2.4f), since at this position the measured laser scans have the most overlap with the reference scan. Then, after the initial displacement ($t = 11s$ in Fig. 2.4a), the scans now captured by the sensor no longer completely overlap with the reference scan. This imperfect alignment is reflected in the estimation errors, which increase slightly until converging to a local minima once the MAV is stabilized at the desired location. This is more noticeable for the position errors, which converge up to $4cm$ from the ground truth position ($t \in [30s, 55s]$ in Fig. 2.4d, and $t \in [15s, 30s]$ in Fig. 2.4e). The same effect is difficult to observe for the yaw estimates in Fig. 2.4f due to the small magnitude of the absolute estimation errors, and the combined effect of the

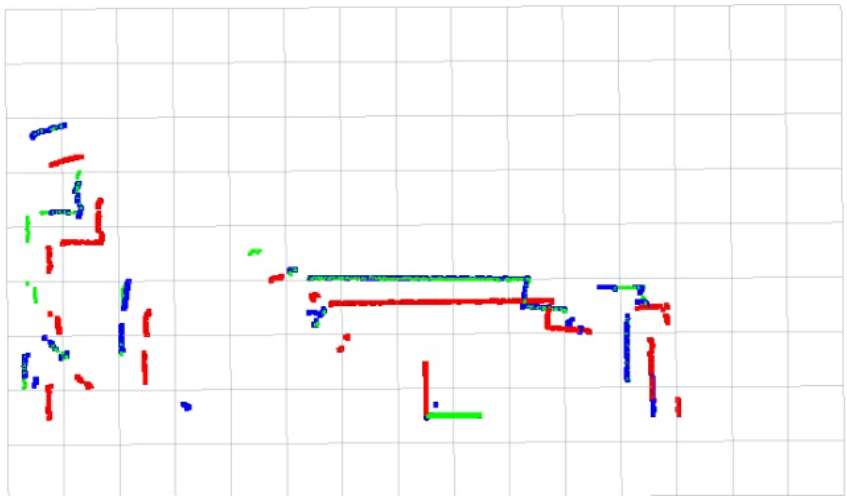


Figure 2.3: A sample of the observed laser scans obtained from the indoor flight shown in Fig. 2.2b. The current laser scan (in red) is aligned to the reference scan (in blue), recovering transformed laser scan (in green).

MoCap tracking noise and laser odometry noise.

Next, several peaks are noticeable in the estimation errors, e.g., $t = 11s$ and $t = 30s$ in Fig. 2.4d-2.4e, $t = 20s$ and $t = 40s$ in Fig. 2.4f. When compared to Fig. 2.4a-2.4c, it can be noted that these peaks correspond to transitions between waypoints. In these situations, the main source of error are the MAV inclinations that occur as the platform navigates towards a new desired location. This introduces further scan misalignments, which are partially handled by the horizontal projections (discussed in Sec. 2.4.1). Another source of error are the MAV velocities, both linear and angular, which can cause distortions in the laser scans. This occurs since the individual points within a scan are now captured at different sensor positions, instead of a unique fixed frame. However, as long as these velocities remain small in comparison to the scan measurement frequencies, this effect remains negligible.

In all cases, it can be noted that the laser odometry estimates follow closely the MoCap measurements. From the absolute estimation errors, the largest position error for both axes reaches $6cm$ ($t = 30s$ in Fig. 2.4e), while the largest absolute yaw error reaches $1.2°$ ($t = 40s$ in Fig. 2.4f).

Attitude estimation results

The non-linear attitude observer proposed in Eq. (2.18) was used to estimate the MAV's roll and pitch angles from the accelerometer and gyrometer readings. The observer gain

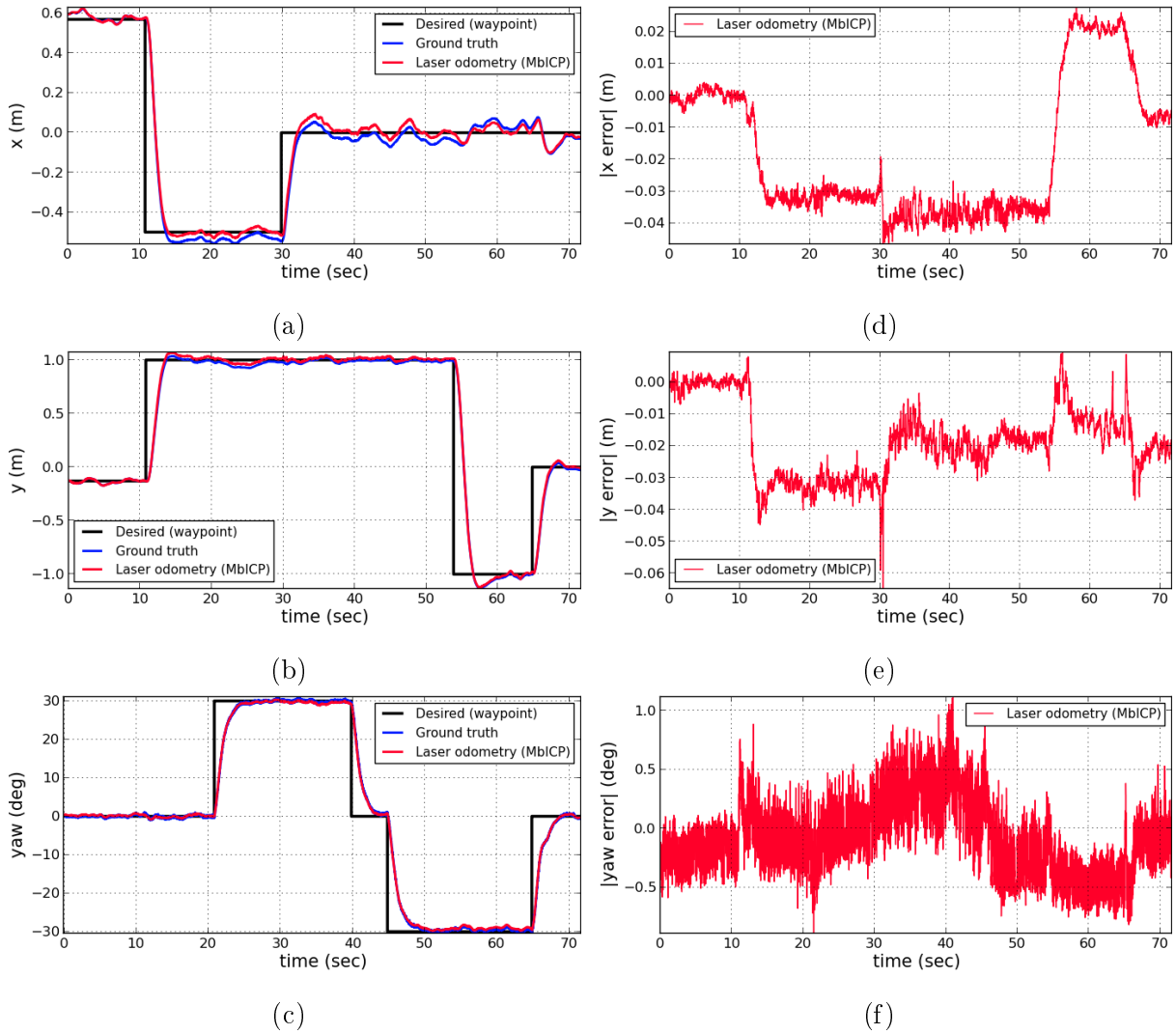


Figure 2.4: For an indoor test flight, following a set of waypoints (as in Fig. 2.2b): **(a-c)** Comparing the 2D laser odometry estimates with the ground truth. **(d-f)** Absolute estimation with respect to the ground truth.

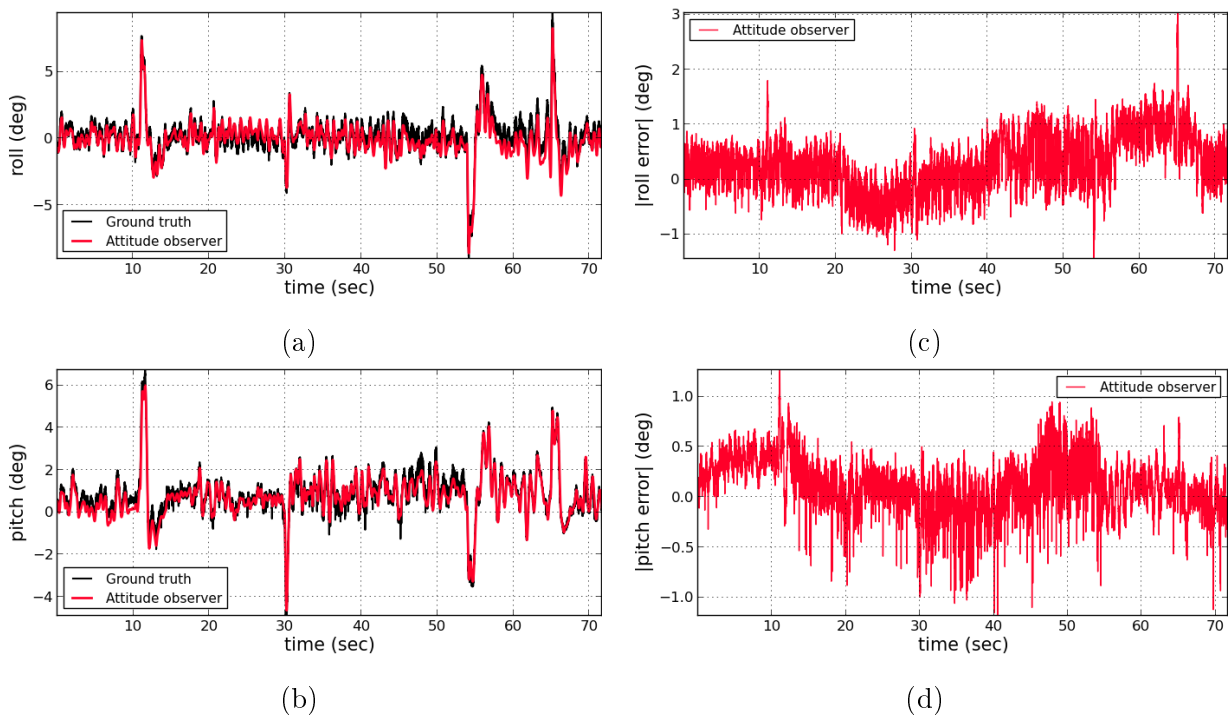
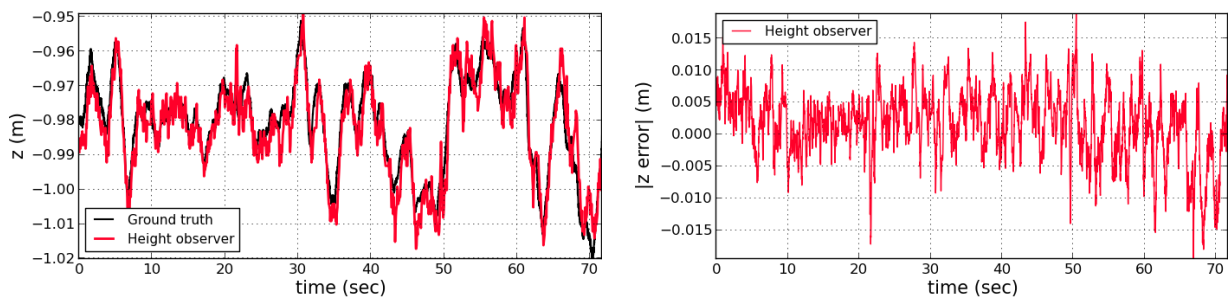


Figure 2.5: Roll and pitch angle estimation results, fusing accelerometer and gyrometer measurements with the attitude observer from Eq. (2.18). **(a-b)** Comparing the attitude estimates against the MoCap ground truth. **(c-d)** The absolute estimation errors with respect to the ground truth.



(a) (b)

Figure 2.6: Height estimation results using the laser altimeter measurements and a second-order low-pass filter, as described in Sec. 2.4.3, for an indoors flight. **(a)** Comparing the altitude estimates against the MoCap ground truth. **(b)** Absolute estimation errors with respect to the ground truth.

was set to $k_\gamma = 0.04$, which was determined experimentally. The estimation results are shown in Fig. 2.5. When compared against the MoCap ground truth, both angles are accurately estimated by the observer. The maximum errors were 3° for the roll angle ($t = 65s$ in Fig. 2.5c) and 1.2° for the pitch angle ($t = 11s$ in Fig. 2.5d). Note that the larger MAV inclinations (e.g., $t = 11s$, $t = 30s$ and $t = 55s$ in Fig. 2.5a-2.5b) coincide with displacements in the horizontal plane (Fig. 2.4a and Fig. 2.4b). As will be discussed in Chapter 5, this is actually part of the control strategy, since horizontal displacements are produced by tilting the MAV, which generates linear accelerations about the horizontal axis from the thrust force produced by the on-board propellers.

Height estimation results

Lastly, we present the height estimation results obtained from the laser altimeter measurements for one of the test flights. The second-order low-pass filter from Eq. (2.23) was used to smooth the laser measurements. The filter gains were chosen as $(k_z, k_{v_z}) = (10, 36)$ and the results are shown in Fig. 2.6. Recall, that throughout the flight, the MAV was stabilized at $1m$ from the ground level. When compared against the MoCap ground truth (Fig. 2.6a) it can be noted that the height was estimated with high precision, as expected since the flights were performed over a flat ground. The maximum estimation error was $2cm$ ($t = 68$ in Fig. 2.6b).

2.6 Conclusions

In this chapter we have presented a complete 6 DoF pose tracking approach for MAVs navigating in planar, heigh-invariant environments, such as indoor scenes. The on-board sensor setup consisted in a 2D laser scanner, an IMU (accelerometer and gyrometer) and a laser altimeter. The MbICP algorithm was chosen for the 2D laser odometry and the corresponding modifications required for an effective application on-board a MAV were described. Attitude estimation was performed from IMU measurements relying on a non-linear observer formulation. Lastly, height estimation relied on filtering laser altimeter measurements. We have presented results for autonomous waypoint following in an indoor scenario, which have shown the efficiency of the pose tracking approach in this first simple case study.

Indoor scenes are the most studied case in literature, as the structure allows capturing rich geometric detail in the laser scans, facilitating the use of traditional scan registration techniques, such as ICP, for pose tracking. No further work was dedicated to this case, since we instead focused on the more challenging scenario of electric tower inspection scenes. The purpose of this part of our work was to serve as an initial experience for developing the methodology that will be presented in the following chapters.

CHAPTER 3

Laser-based pose estimation for electric tower inspection: Feature-based approach

Contents

3.1 Overview **44**

3.2 Characteristics of the laser range measurements on an electric tower **45**

3.2.1 Notable features 47

3.3 Tracking the cross-sections of the electric tower **48**

3.3.1 Scan segmentation 49

3.3.2 Geometric fitting 50

3.3.3 Calculating the position and orientation 52

3.4 Results **53**

3.4.1 Simulation results 53

3.4.2 Experimental results 58

3.4.3 Limitations 59

3.5 Conclusions **61**

3.1 Overview

Automation of inspection tasks is crucial for the development of the power industry, where MAVs have shown a great potential. In this context, GPS sensors remain the preferred

Figure 3.1: The quadrotor from Fig. 2.2a was manually flown in front of an electric tower from a 60 kV distribution line, to register the laser range measurements.

choice for localization purposes [8]. However, these sensors provide no perception of the surrounding environment, and applications relying on GPS measurements uniquely are limited to waypoint navigation at large distances from the inspected objects [2, 8, 10]. Self-localization in inspection tasks remains a key issue, and is the main subject of this chapter. In particular, we focus on light-weight 2D LiDARs and explore how they can be used for pose estimation purposes in these scenarios.

The approach used to estimate a robot's pose from laser range measurements depends greatly on the structure of the surrounding environment. In the previous chapter, we studied the ordinary case of indoor scenarios, where the solid and planar surface of surrounding objects allowed the individual 2D laser scans to capture rich geometric detail. Relative motion could be estimated from traditional scan matching algorithms, such as the ICP algorithm, in real-time and using on-board processing power [19, 20, 22]. In this chapter, we seek to achieve similar capabilities when localizing a MAV with respect to an electric tower. More specifically, we explore how basic geometric knowledge of the scene can be exploited for this purpose. Our main interest are steel lattice towers made up of rectangular cross-sections commonly used to support high-voltage transmission lines, such as the one shown in Fig. 3.1. For this first case study, we concentrate on the tower's body, which makes up the largest portion of the structure. The tower heads have a more complex structure that require an extensive parametrisation [7, 26], and were not considered in this work.

This chapter is structured as follows. First, in Sec. 3.2 we discuss important character-

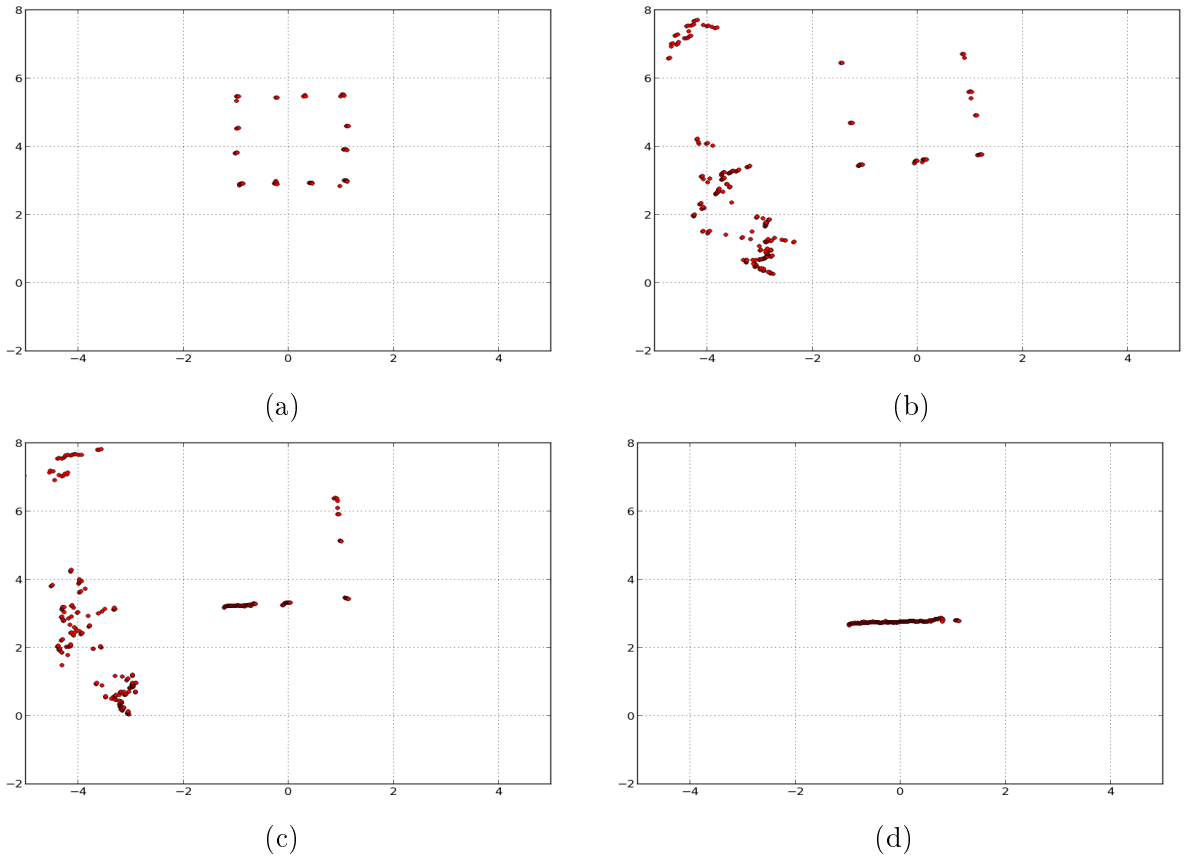


Figure 3.2: Laser range measurements acquired on the tower from Fig. 3.1: **(a)** All sides are visible (the best case scenario). **(b)-(c)** Occlusions sometimes block the lateral and backsides from view. **(d)** Only the front side is visible (the worst case scenario). This happens when horizontal bars on the tower block the lateral and back sides from view.

istics observed in laser range measurements taken on a real electric tower, and highlight differences with typical indoor scenes. Then, based on these observations, in Sec. 3.3 we focus on the electric tower's geometry, and explore how basic knowledge can be exploited to recover 2D pose information directly from the individual laser scans. Experimental and simulation results are then presented in Sec. 3.4.1 and Sec. 3.4.2 respectively. This chapter concludes with a brief discussion on the main limitations and extensions of the proposed tracking approach in Sec. 3.4.3.

3.2 Characteristics of the laser range measurements on an electric tower

In order to determine an appropriate approach to recover pose information from the laser scans under this scenario, we first analyse the characteristics of 2D laser range measurements taken on an electric tower. In several test flights, the quadrotor platform from Fig. 2.2a, equipped with a Hokuyo UTM-30LX 2D laser scanner, was manually flown in front of a real electric tower, as was shown in Fig. 3.1. Some key cases observed from the 2D laser scans are presented in Fig. 3.2. Several inconvenients can be highlighted in these figures:

- The 2D laser scans only capture a cross-section of the tower, whose dimensions vary greatly with height.
- The visible contour is very discontinuous and few laser scans fall on the surface of the tower. Moreover, the visible cross-section can change drastically between consecutive scans. Thus, overlap between pairs of 2D laser scans is very limited.
- Very different structures can be observed due to the large open spaces on the surface of the tower. In the best case, all of the tower's faces are captured (Fig. 3.2a). In certain cases, there is only a partial view of the cross-section (Fig. 3.2b-3.2d). In the worst case scenario (Fig. 3.2d), horizontal bars that are part of the tower's structure block the lateral and back sides from view and only the front side of the tower is captured in the scans.
- The laser scans can capture surrounding vegetation (Fig. 3.2b-3.2d).

These observations show that estimating the MAV's pose from the laser scans under this scenario isn't a trivial task. In particular, the 3D geometry of the tower implies that the planar and height-invariance assumption discussed in Sec. 2.2 is not valid in this scenario. Moreover, due to the insufficient overlap between 2D laser scans, aligning pairs of 2D scans with scan matching techniques to recover pose information, as was done indoors in Chap. 2, is not an appropriate approach.

The contour captured by the laser scans is the intersection of the scan plane with the tower's surface, and its shape and size contains implicit pose information. Furthermore, this contour can be easily distinguished from surrounding unstructured vegetation. Thus, instead of focusing on the individual points, we focus on the geometry of the cross-section

captured by the individual laser scans. The idea is to gradually extract the features visible in the laser scans, to account for the different cases observed in Fig. 3.2, then to use the extracted features and basic knowledge of the tower's geometry to determine the position and orientation of the tower.

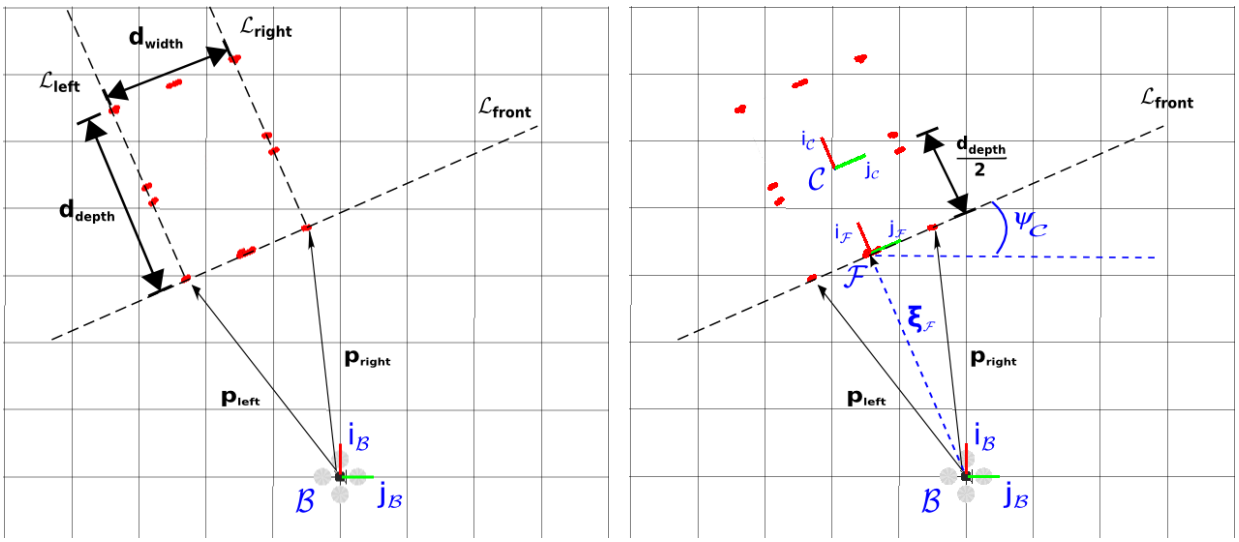
3.2.1 Notable features

Based on the observations from Fig. 3.2, we break down the cross-section into its main features as follows:

- The largest concentration of laser beams fall on the side closest to the MAV, and the line segment formed by these points is the most notable feature in the laser scans. This front line, denoted as $\mathcal{L}_{\text{front}}$, allows recovering essential position and orientation information. Since $\mathcal{L}_{\text{front}}$ remains visible even in the worst case scenario (Fig. 3.2d), tracking this line is at the heart of our proposed approach.
- The coordinate vectors, expressed in $\mathcal{B}$, of the left and right corners of $\mathcal{L}_{\text{front}}$ are denoted as $\mathbf{p}_{\text{left}}$ and $\mathbf{p}_{\text{right}}$ respectively.
- The lateral sides $\mathcal{L}_{\text{left}}$ and $\mathcal{L}_{\text{right}}$ aren't always visible (Fig. 3.2a-3.2c), but provide complementary orientation information and allow determining the depth (and hence the center) of the cross-section. The back side of the tower is not explicitly taken into account, as it is seldom visible and provides unreliable information.

These features are illustrated in Fig. 3.3a. Lastly, the dimensions of the cross-section are the depth and the width, denoted d_{depth} and d_{width} respectively, which are initially unknown but will be estimated on-flight.

As already mentioned, we focus on the body of electric towers made up of rectangular cross-sections. Hence, we consider that the tower contour captured in the scans is rectangular due to the tower's shape, which can be clearly identified in Fig. 3.2. However, for this assumption to hold, the scan plane must remain horizontal. The limitations of this assumptions will be discussed in more detail in Sec. 3.4.3.



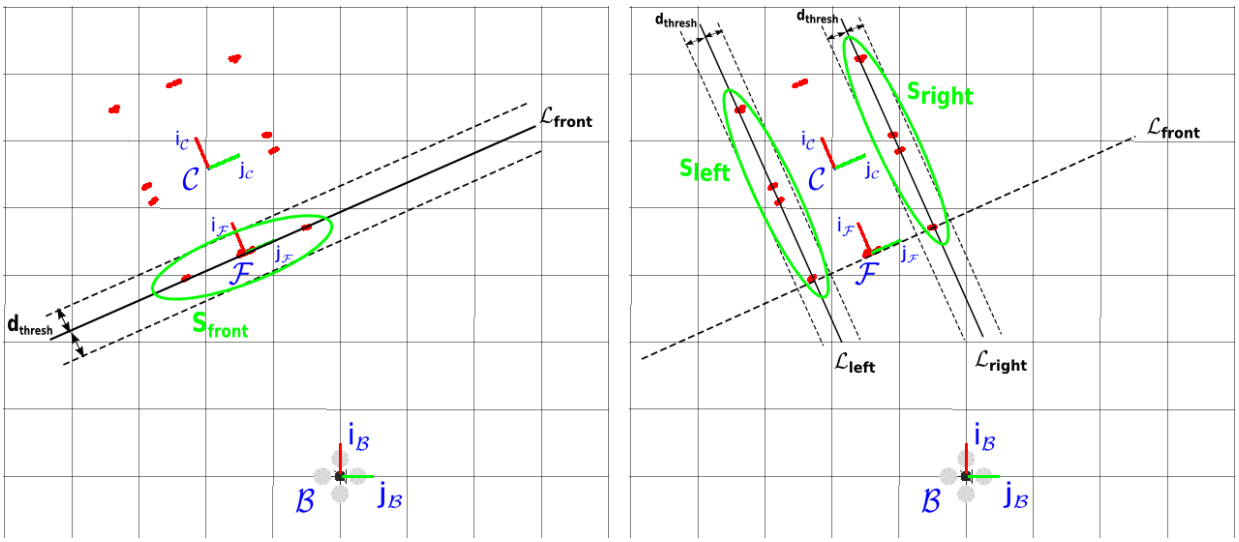
(a) (b)

Figure 3.3: Based on observations from Fig. 3.2: **(a)** Notable cross-section features captured in the 2D laser scans. **(b)** Proposed parametrization of the electric tower's cross-section.

3.3 Tracking the cross-sections of the electric tower

In this section we focus on tracking the cross-sections captured by the individual 2D laser scans, which is analogous to determining the 2D pose of the MAV with respect to the electric tower. Moreover, we seek to achieve this directly from the laser scans without the aid of external sensing. Let $\{x_{\mathcal{C}}, y_{\mathcal{C}}, \psi_{\mathcal{C}}\}$ denote the 2D pose of the cross-section's center with respect to the body frame $\mathcal{B}$. Then, $\mathcal{C} = \{\mathcal{O}_{\mathcal{C}}, \vec{\imath}_{\mathcal{C}}, \vec{\jmath}_{\mathcal{C}}\}$ denotes the center-attached frame, $\boldsymbol{\xi}_{\mathcal{C}} = (x_{\mathcal{C}}, y_{\mathcal{C}})^{\intercal}$ denotes the position vector of $\mathcal{C}$ with respect to $\mathcal{B}$, expressed in $\mathcal{B}$, and $\psi_{\mathcal{C}}$ denotes the orientation of $\mathcal{C}$ with respect to $\mathcal{B}$. For a completely horizontal scan plane, this frame is aligned with the inertial frame $\mathcal{I}$. A second frame $\mathcal{F} = \{\mathcal{O}_{\mathcal{F}}, \vec{\imath}_{\mathcal{F}}, \vec{\jmath}_{\mathcal{F}}\}$ is attached to the front side's center, with corresponding position vector $\boldsymbol{\xi}_{\mathcal{F}}$ with respect to $\mathcal{B}$, expressed in $\mathcal{B}$, and similar orientation to $\mathcal{C}$.

The complete parametrization is shown in Fig. 3.3b. Note that $\boldsymbol{\xi}_{\mathcal{F}}$ is determined from the two front corners, and $\boldsymbol{\xi}_{\mathcal{C}}$ is calculated from $\boldsymbol{\xi}_{\mathcal{F}}$ and d_{depth}. If the goal is to stabilize the MAV in front of the tower, then tracking $\mathcal{F}$ is sufficient and the task is greatly simplified. The center-attached frame $\mathcal{C}$ is important, for example, for a 3D reconstruction of the tower, as will be discussed later. The following subsections describe the three main steps implemented to track the cross-sections directly from the 2D laser scans: scan segmentation, geometric fitting and calculating the 2D pose.



(a) (b)

Figure 3.4: Illustrating the proposed scan segmentation process: **(a)** Detecting the front side. **(b)** Detecting the left and right sides.

3.3.1 Scan segmentation

This first step consists in detecting and classifying the laser beams that fall on the surface of the tower. First, measurements that fall outside of the tower, such as nearby vegetation (Fig.3.2b and Fig.3.2c), can perturb the tracking process and must be extracted from the laser scans. We handle this by setting a fixed outlier rejection radius from the tracked tower center, and removing points outside this radius. For the first laser scan, we provide an initial rough guess of the tower's position. The laser scan is thus divided into three subsets of points (expressed in $\mathcal{B}$)

$$\begin{aligned}
\mathbf{S}_{\text{front}} &= \{\mathbf{p}_{F,i} = (x_{F,i}, y_{F,i})^{\intercal}, \quad i = 1, ..., N_F\} \\
\mathbf{S}_{\text{left}} &= \{\mathbf{p}_{L,j} = (x_{L,j}, y_{L,j})^{\intercal}, \quad j = 1, ..., N_L\} \\
\mathbf{S}_{\text{right}} &= \{\mathbf{p}_{R,k} = (x_{R,k}, y_{R,k})^{\intercal}, \quad k = 1, ..., N_R\}
\end{aligned} \tag{3.1}$$

which correspond to the front, left and right sides respectively. In the worst case scenario only the front side is visible (Fig. 3.2d), so $\mathbf{S}_{\text{front}}$ is extracted first. Then, it can be determined if the lateral sides $\mathbf{S}_{\text{left}}$ and $\mathbf{S}_{\text{right}}$ are visible in the scan.

3.3.1.1 Extracting the front line segment

The Random Sample Consensus (RANSAC) algorithm [109] is used for this purpose, which is a well-known technique for point cloud segmentation due to its robustness to outliers and noise. This is an iterative method in which an instance of a geometric model is found in a data set by randomly sampling minimal subsets of points to generate a hypothesis of the model, testing the quality of the guess by finding all nearby points (i.e., the *inliers*) and selecting the instance with the highest number of inliers. In our case, it is used to find instances $\mathcal{L}_{\text{front}}$ in the laser scans, where points that fall within a distance threshold d_{thresh} are considered as inliers (as shown in Fig. 3.4a). Here, $\mathcal{L}_{\text{front}}$ is parametrized according to the line equation in its general form

$$\mathcal{L}_{\text{front}} : c_F + n_x x_F + n_y y_F = 0, \quad n_x^2 + n_y^2 = 1. \tag{3.2}$$

where (n_x, n_y) are the coordinates of the normal vector, expressed in $\mathcal{B}$, and (x_F, y_F) represent the coordinates of an arbitrary point on the front line, also expressed in $\mathcal{B}$. Furthermore, a maximal inclination ψ_{max}, with respect to the previously extracted front line, is imposed to the line model to avoid mistakenly extracting the sidelines.

For the first scan, it is assumed that there is a rough knowledge of the MAV's orientation with respect to the tower. Upon convergence, the subset $\mathbf{S}_{\text{front}}$ and an initial estimate of the coefficients of $\mathcal{L}_{\text{front}}$ are obtained. Next, the front side's corners are identified from the extracted points. Since the lateral sides of the tower are perpendicular to the front line, projecting their points onto the estimated $\mathcal{L}_{\text{front}}$ results in a high concentration of points around the location of the front corners. Thus, $\mathbf{p}_{\text{right}}$ and $\mathbf{p}_{\text{left}}$ are obtained as the two endpoints of the projected points on the front line.

3.3.1.2 Extracting the lateral sides

Next, we determine if the lateral sides are visible in the laser scan. A search region is determined for the left and right sides by tracing a perpendicular line to $\mathcal{L}_{\text{front}}$ through each of the front corners determined in the previous step. The candidate points for $\mathbf{S}_{\text{left}}$ and $\mathbf{S}_{\text{right}}$ are extracted by selecting points within the distance threshold d_{thresh} as shown on Fig. 3.4b. The candidate point sets are accepted only if they contain at least N_{min} points, and if the maximum separation between the points is at least d_{min}. This is done to determine if the sides are sufficiently visible to provide reliable information.

3.3.2 Geometric fitting

At this point, the laser beams that fall on the surface of the tower have been identified and classified according to which side they measure. The goal is now to find the geometric model that best fits the extracted points. In the previous step, the RANSAC algorithm provided an initial estimate of the coefficients of $\mathcal{L}_{\text{front}}$. However, a better solution can be obtained by taking into account the geometric constraints that make up the rectangular shape of the cross-section, while recovering the remaining coefficients of $\mathcal{L}_{\text{left}}$ and $\mathcal{L}_{\text{right}}$. Keeping in mind the cases illustrated in Fig. 3.2, three different situations can arise from the scan segmentation step

- **Case 1:** No side was detected. Then, the estimation process stops since no useful information is available.
- **Case 2:** Only the front side was detected. Then, the coefficients for $\mathcal{L}_{\text{front}}$ are directly provided by the RANSAC algorithm and the orientation can be estimated, but no depth information is available and the center of the cross-section can't be determined.
- **Case 3:** The front side and at least one of the lateral sides was detected. Then, the rectangular shape of the cross-section can be taken into account. This allows obtaining $\mathcal{L}_{\text{left}}$ and $\mathcal{L}_{\text{right}}$, and a more precise estimate of $\mathcal{L}_{\text{front}}$.

We thus focus on the third case. The following formulation applies to the case when both $\mathbf{S}_{\text{left}}$ and $\mathbf{S}_{\text{right}}$ are detected, but the same procedure is valid when only one of the lateral sides is found. Since the lateral sides are perpendicular to $\mathcal{L}_{\text{front}}$, then, from Eq. (3.2) their normal vector is $(-n_y, n_x)$. Here, the cross-section is defined by

$$\begin{cases} \mathcal{L}_{\text{front}}: & c_F + n_x x_F + n_y y_F = 0, \\ \mathcal{L}_{\text{left}}: & c_L - n_y x_L + n_x y_L = 0, \\ \mathcal{L}_{\text{right}}: & c_R - n_y x_R + n_x y_R = 0, \\ & n_x^2 + n_y^2 = 1. \end{cases} \tag{3.3}$$

Then, evaluating the extracted point sets $\mathbf{S}_{\text{front}}$, $\mathbf{S}_{\text{left}}$ and $\mathbf{S}_{\text{right}}$ from Eq. (3.1) with their respective line from Eq. (3.3), and expressing in matrix form, one obtains

$$\begin{pmatrix} 1 & 0 & 0 & x_{F,1} & y_{F,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & x_{F,N_F} & y_{F,N_F} \\ 0 & 1 & 0 & y_{L,1} & -x_{L,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & y_{L,N_L} & -x_{L,N_L} \\ 0 & 0 & 1 & y_{R,1} & -x_{R,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & y_{R,N_R} & -x_{R,N_R} \end{pmatrix} \begin{pmatrix} c_F \\ c_L \\ c_R \\ n_x \\ n_y \end{pmatrix} = \boldsymbol{\rho}. \tag{3.4}$$

where $\boldsymbol{\rho} = (\rho_1, \cdots, \rho_N)^\intercal$, with $N = N_F + N_L + N_R$, are the residuals. From the line equation in its general form (Eq. (3.2)), it follows that the absolute value of these residuals $|\rho_i|$ corresponds to the perpendicular distance from the point to the line. Hence, the geometric fitting problem is formulated as finding the coefficients of Eq. (3.3) for which the sum of squared distances (residuals) is minimal. That is

$$\begin{aligned} \min \|\boldsymbol{\rho}\|^2 = \min \sum_{i=1}^{N} \rho_i^2, \quad & \text{subject to Eq. (3.4)}, \\ & \text{and } n_x^2 + n_y^2 = 1, \end{aligned} \tag{3.5}$$

which is a constrained least squares problem. From Eq. (3.4) it can be seen that there are more equations than unknowns ($N = N_F + N_L + N_R$ equations for 5 unknown variables) and this is in reality an overdetermined system without an exact solution. An approximate solution can be obtained numerically following the procedure described in [110]. Denoting Eq. (3.4) in the form $A\mathbf{x} = \boldsymbol{\rho}$, the first step is to reduce the size of this linear system. This is done by applying a QR decomposition to the A matrix, which allows finding an orthogonal matrix Q and an upper triangular matrix R such that $A = QR$. Then,

multiplying both sides of Eq. (3.4) by $Q^\intercal$, this leads to

$$Q^\intercal QR\mathbf{x} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} & r_{15} \\ 0 & r_{22} & r_{23} & r_{24} & r_{25} \\ 0 & 0 & r_{33} & r_{34} & r_{35} \\ 0 & 0 & 0 & r_{44} & r_{45} \\ 0 & 0 & 0 & 0 & r_{55} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} c_F \\ c_L \\ c_R \\ n_x \\ n_y \end{pmatrix} = Q^\intercal \boldsymbol{\rho}, \quad n_x^2 + n_y^2 = 1. \tag{3.6}$$

where $Q^\intercal Q = I$ and the norm of the right hand side remains unchanged since Q is orthogonal. Noting that the non-linear constraint is only applied to the last two unknowns (n_x, n_y), the minimisation problem from Eq. (3.5) is reduced to

$$\min_{n_x, n_y} \|B \begin{pmatrix} n_x \\ n_y \end{pmatrix} \|^2, \quad \text{subject to } n_x^2 + n_y^2 = 1, \quad B = \begin{pmatrix} r_{44} & r_{45} \\ 0 & r_{55} \end{pmatrix} \tag{3.7}$$

which is a simple least squares problem solved with Singular Value Decomposition (SVD). Here, the minimum norm is obtained from the smallest singular value of B, and the corresponding right singular vector gives the estimated parameters (n_x, n_y) . The remaining coefficients (c_F, c_L, c_R) are calculated by back-substitution of (n_x, n_y) in Eq. (3.6).

To summarize, the end result is an estimate of the parameters of Eq. (3.3). At this point, $\mathbf{p}_{\text{left}}$ and $\mathbf{p}_{\text{right}}$ are recalculated from the line intersections, as they will be required in the following step.

3.3.3 Calculating the position and orientation

We first determine the position and orientation of the front frame $\mathcal{F}$. Recovering the orientation of the tower (illustrated in Fig. 3.3b) results straightforward from the coefficients of $\mathcal{L}_{\text{front}}$, as

$$\psi_{\mathcal{C}} = \text{arctan2}(n_y, n_x). \tag{3.8}$$

Then, $\boldsymbol{\xi}_{\mathcal{F}}$ is calculated as the midpoint between $\mathbf{p}_{\text{right}}$ and $\mathbf{p}_{\text{left}}$ as

$$\boldsymbol{\xi}_{\mathcal{F}} = \frac{\mathbf{p}_{\text{right}} + \mathbf{p}_{\text{left}}}{2}. \tag{3.9}$$

Next, the dimensions of the cross-section are determined. The width d_{width} corresponds to the distance between the two front corners and the depth d_{depth} is chosen as the distance of the point in $\mathbf{S}_{\text{left}}$ or $\mathbf{S}_{\text{right}}$ furthest from $\mathcal{L}_{\text{front}}$. Finally, the coordinates of $\boldsymbol{\xi}_{\mathcal{C}}$ are calculated as

$$\boldsymbol{\xi}_{\mathcal{C}} = \boldsymbol{\xi}_{\mathcal{F}} + \frac{d_{\text{depth}}}{2} \begin{pmatrix} \cos\psi_{\mathcal{C}} \\ \sin\psi_{\mathcal{C}} \end{pmatrix}. \tag{3.10}$$

It is important to highlight that the visible cross-section can change drastically from one scan to the other, as was shown in Fig. 3.2. This in return can induce large jumps in the estimates, since they are obtained from each individual laser scan. To reduce this effect, and to obtain smoother results, $\boldsymbol{\xi}_{\mathcal{F}}$, $\psi_{\mathcal{C}}$ and d_{depth} are filtered using first-order low-pass filters.

3.4 Results

As previously mentioned, tracking the tower's cross-sections is analogous to determining the MAV's 2D pose with respect to the electric tower, thus acting as a 2D laser odometry. The remaining states can be obtained as described in the previous chapter: attitude estimation from IMU measurements (Sec. 2.4.2) and height estimation from laser altimeter measurements (Sec. 2.4.3). In the following results, we focus on validating the tracking procedure as described in this chapter. Experimental and simulation tests were performed for this purpose, as illustrated in Fig. 3.1 and Fig. 3.5. We now describe in more detail the setups observed in these figures, the tests performed in each case and the final results.

3.4.1 Simulation results

The simulated flights were performed using the Gazebo simulation environment [111] and ROS as an interfacing middleware [112], on a PC with an Intel 3.4 GHz Quad-Core processor and 8 GB of RAM. The Hector quadrotor stack from ROS [113] was used to simulate the quadrotor kinematics and dynamics. The 2D tracking approach described in Sec. 3.3 works directly on the individual laser scans and does not rely on external sensing. Hence, in the following simulated flights the 2D laser scanner was the only relevant sensor, which was set to match the characteristics of a Hokuyo URG-30LX sensor: 40 Hz scan frequency, 0.25° angular resolution and 270° field of view (thus 1080 measurements per scan). This sensor was mounted horizontally on top of the simulated quadrotor. A CAD model of an electric tower body was used, whose dimensions are 2.5 m × 3.5 m at the

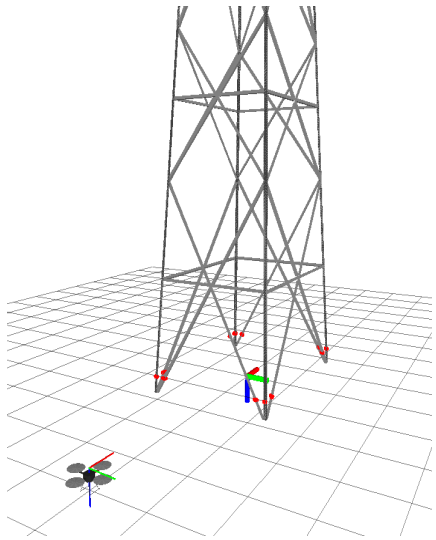


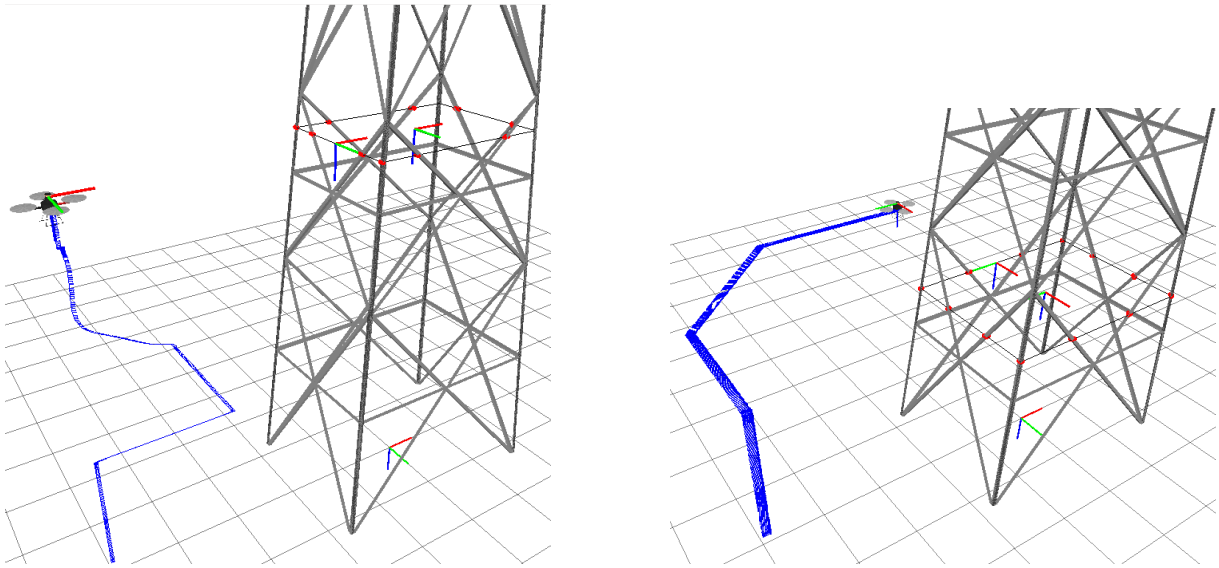
Figure 3.5: The simulation setup in the Gazebo simulation environment. A CAD model of an electric tower was used, whose dimensions roughly correspond to those of the tower from Fig. 3.1.

ground level, and 1.5 m $\times$ 2 m at a height of 10 m. These dimensions roughly correspond to those of the tower body from Fig. 3.1. The complete simulation setup is shown in Fig. 3.5. All algorithm development was done using C++, and the sample consensus module from the open source Point Cloud Library (PCL) [98].

Two different cases were analysed in these simulations, illustrated in Fig. 3.6. In a first test, the MAV was flown in front of only one side of the tower. During this flight, the MAV attained different heights and operated at different distances from the tower's center as shown in Fig. 3.6a. This figure also illustrates an example of a tracked cross-section with its corresponding front $\mathcal{F}$ and center $\mathcal{C}$ frames. In the second test, the MAV was flown around the tower at a fixed height (Fig. 3.6b). During these simulations, the pose information from the simulation ground truth was used to provide an initial position of the tower's center with respect to the MAV, and to stabilize the MAV's position. For the scan segmentations (Sec. 3.3.1), the outlier rejection radius was set to 4 m, and the parameters for the front line segment extraction were chosen as d_{thresh}= 5 cm and $\psi_{\max} = 10°$ (Sec. 3.3.1.1), which provided reasonable results in the simulations. We now present the results for each of the studied cases.

3.4.1.1 First case: flight in front of the tower.

In this first test, the tracked position $\boldsymbol{\xi}_{\mathcal{C}} = (x_{\mathcal{C}}, y_{\mathcal{C}})^{\intercal}$ and orientation $\psi_{\mathcal{C}}$ of the tower's center, for the duration of this flight (Fig. 3.6a), are compared to the simulation ground truth



(a) (b)

Figure 3.6: The two simulated flights. The blue line indicates the trajectory followed by the quadrotor. An example of a tracked cross-section is visible on the laser scan (red points), with corresponding estimated front $\mathcal{F}$ and center frames $\mathcal{C}$. **(a)** First case study for a flight in front of the tower, at different heights. **(b)** Second case study for a flight around the tower, at a fixed height.

in Fig. 3.7a-3.7c. The corresponding absolute estimation errors are shown in Fig. 3.7d-3.7f. Then, Fig. 3.7g illustrates the MAV's elevation with respect to the ground level, obtained from the ground truth.

Before the first horizontal motion (t =15 s in Fig. 3.7a), the MAV undergoes a vertical displacement of 2 m ($t \in [0s, 15s]$ in Fig. 3.7g). During this time-lapse, it can be noted in Fig. 3.7d-3.7c that the tracking errors for all states are minimal. Then, at t=15 s, the MAV advances 2 m in the x-axis towards the electric tower (Fig. 3.7a). Here, the MAV must tilt, in order to advance towards the desired location, and the horizontal scan plane assumption (used throughout Sec. 3.3) no longer holds. This resulted in an error peak of 2.5 cm (t=15 s in Fig. 3.7d). Throughout the flight, similar error peaks can be observed, which can be traced back to instances where the MAV translates about the horizontal plane and undergoes tilting motion (e.g., t=24 s in Fig. 3.7d, t=46 s in Fig. 3.7e and t=51 s in Fig. 3.7f). In these simulations, MAV inclinations were below 6° (for the roll and pitch angles) and the associated error peaks remained within acceptable levels. Throughout the rest of this flight the proposed approach is capable of effectively tracking the tower's center.

Lastly, as our feature-based tracking method works on a data-set of reduced size

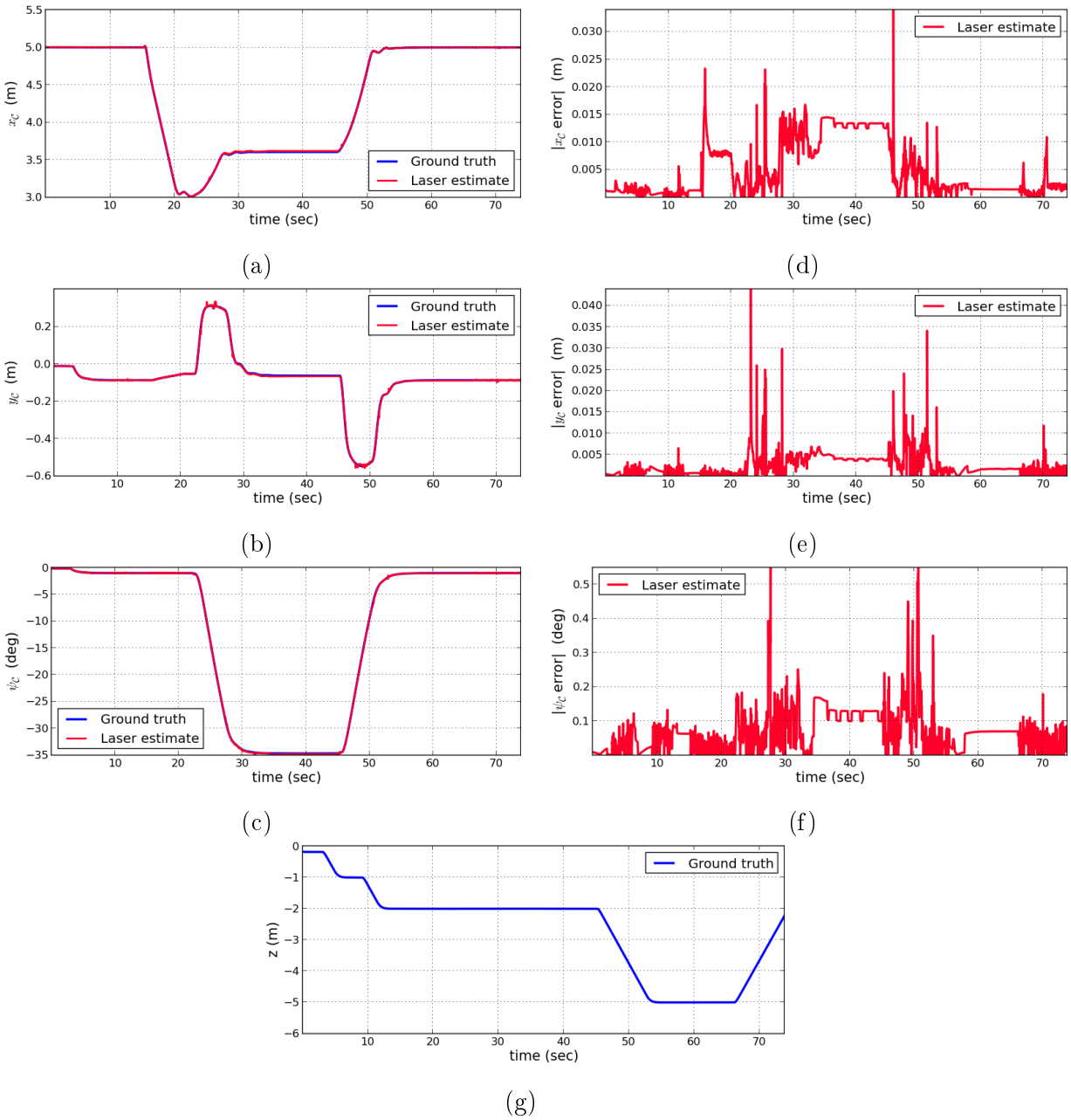


Figure 3.7: For a flight in front of the simulated electric tower (from Fig. 3.6a): **(a-c)** Comparing the tracked position $\boldsymbol{\xi_C} = (x_C, y_C)^\intercal$ and orientation ψ_C of the tower's center with the simulation ground truth. **(d-f)** The absolute tracking errors, with respect to the simulation ground truth. **(g)** The MAV's elevation with respect to the ground level, obtained from the ground truth.

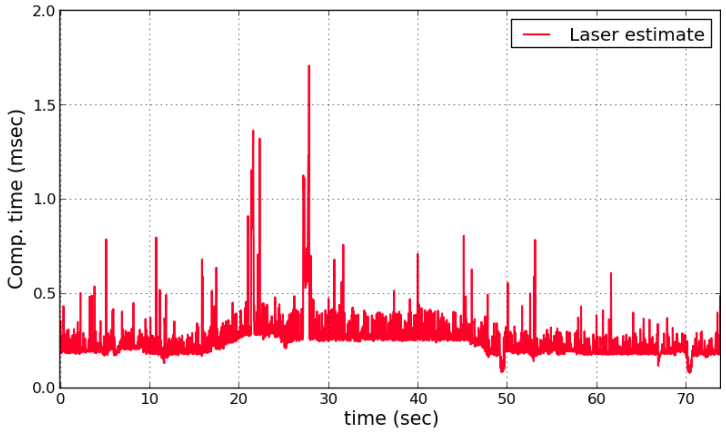


Figure 3.8: Processing time (in ms) of the tracking method for the results shown in Fig. 3.7. The proposed tracking method processes laser scans at an average time of 0.3 ms.

(due to the scan segmentations) and avoids time-consuming correspondence searches of iterative registration techniques, such as ICP, pose information can be recovered at high rates. This is illustrated in Fig. 3.8, where it can be noted that the tracking method was able to process laser scans at an average of 0.3 ms, with maximum processing time of 2 ms. In this case, the bottleneck of the estimation frequency is the sensor scan rate, which only provides laser scans every 25 ms.

3.4.1.2 Second case: flight around the tower

The tracking results for this second test are shown in Fig. 3.9. Recalling from Sec. 3.3, the proposed tracking method relies heavily on finding the line closest to the MAV (i.e., the front line). Thus, in this case, as the MAV transitions from one side of the tower to another (t=16 s), the algorithm attempts to track a different front line. Initially, this produces the large shift of 45° in the orientation estimates (t=16 s in Fig. 3.7f), and an increase in the position errors ($t \in$ (16s, 20s) in Fig. 3.9d-3.9e). Eventually, as the MAV completes the turn, the position of cross-section's center is effectively estimated again, as the errors decrease towards zero (t=20 s in Fig. 3.9d-3.9e). However, the orientation error is not corrected, and instead remains at 90° in Fig. 3.7f as a new front line is tracked. This illustrates one of the main limitations of the proposed approach, as it requires the MAV to remain on the same side of the tower throughout the flight.

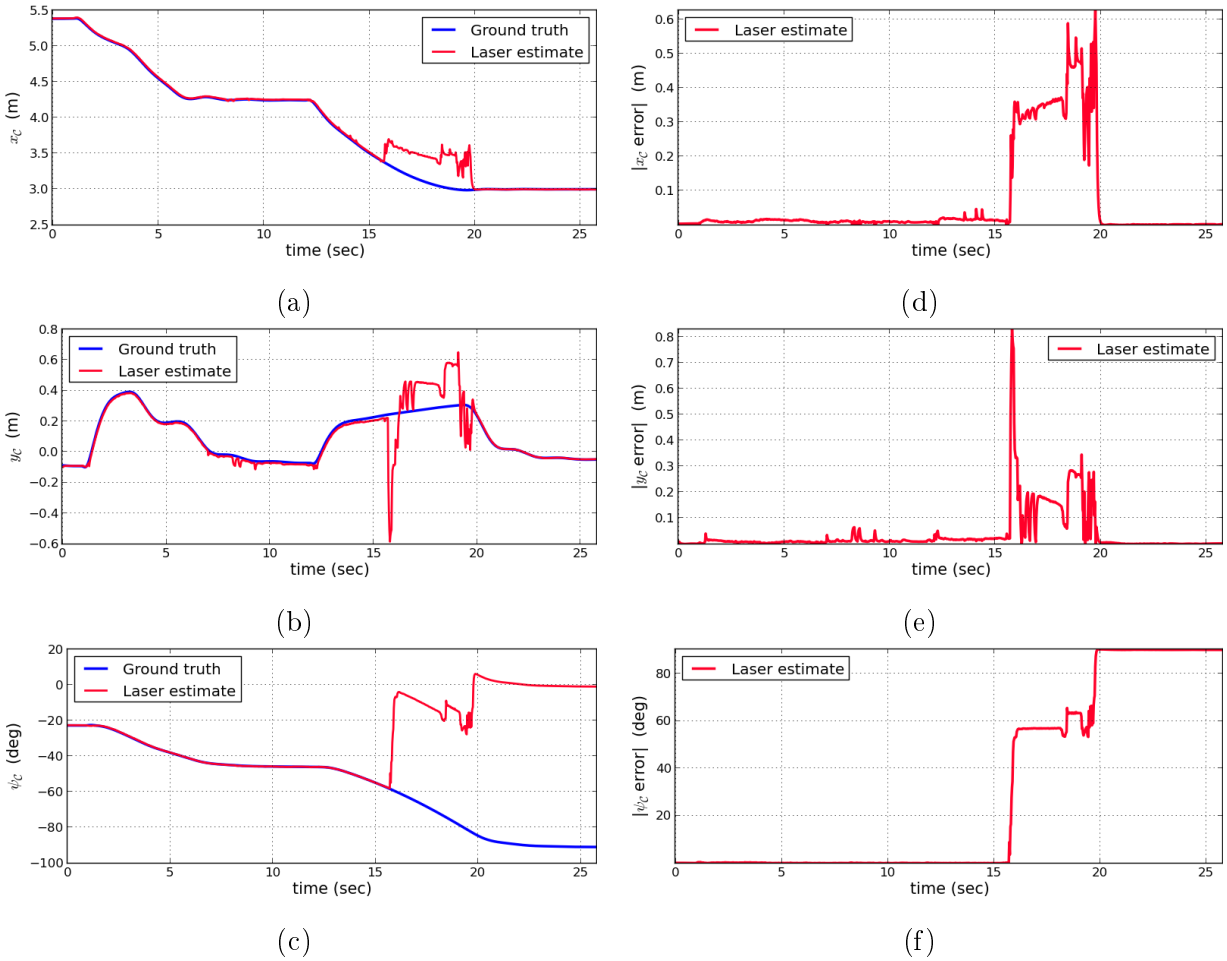


Figure 3.9: For a flight around the simulated electric tower (from Fig. 3.6b). **(a-c)** Comparing the tracked position $\boldsymbol{\xi_\mathcal{C}} = (x_\mathcal{C}, y_\mathcal{C})^\intercal$ and orientation $\psi_\mathcal{C}$ of the tower's center with the simulation ground truth. **(d-f)** The absolute tracking errors, with respect to the simulation ground truth. The tracking method fails (starting at t=16 s) when the MAV transitions from one side of the tower to another.

3.4.2 Experimental results

Several manual test flights were performed with the MAV platform from Fig. 2.2a on a real electric tower (as previously shown in Fig. 3.1), using the same sensor setup as the indoor flights (Chapter 2): an MPU6000 3 axis accelerometer/gyrometer IMU, a SF10 Lightware Optoelectronics laser altimeter and a Hokuyo URG-30LX 2D laser scanner. On-board, the same approaches discussed in Sec. 2.4.3 and Sec. 2.4.2 were used to recover height and attitude estimates. In these test flights the MAV was flown vertically in front of the electric tower, and all information captured on-board was recorded. Then, the registered laser scans were tested with our tracking method, using the same scan segmentation parameters as in the simulations. An initial rough guess of the pose of the tower's center with respect to the MAV was also given.

Unfortunately, at the time of the acquisitions a GPS sensor was not used and a ground truth is not available to determine the estimation errors. However, recalling that our tracking algorithm estimates the previously unknown depth and width of the tower's cross-sections, an alternative way of validating the approach is to determine if these dimensions are coherent with the 3D geometry of the real tower. Thus, Fig. 3.10 illustrates the estimated dimensions combined with their corresponding estimated height from the laser altimeter readings, for one of the test flights. The efficiency of the 2D tracking algorithm is evident, since electric towers with rectangular cross-sections have a depth and width that vary linearly with height, a behaviour that is clearly reflected in Fig. 3.10.

3.4.2.1 Modelling the electric tower.

A by-product of tracking the cross-section's center, and of estimating the previously unknown tower dimensions, is the possibility of deriving a 3D representation of the electric tower from the observed data, such as a 3D point cloud reconstruction from the laser scans. A simple procedure consists in transforming each 2D scan into the estimated center frame $\mathcal{C}$, and projecting into 3D coordinates using the height measurements and the attitude estimates from the IMU measurements. This was tested on the same vertical flight data used to obtain Fig. 3.10, and the final result is shown in Fig. 3.11. Here, the efficiency of the tracking method is also evident, as the point cloud is capable of capturing a great amount of detail, and presents minimal deformations despite being made from data acquired on-flight.

A second possibility is to instead derive an abstract 3D geometric representation of the

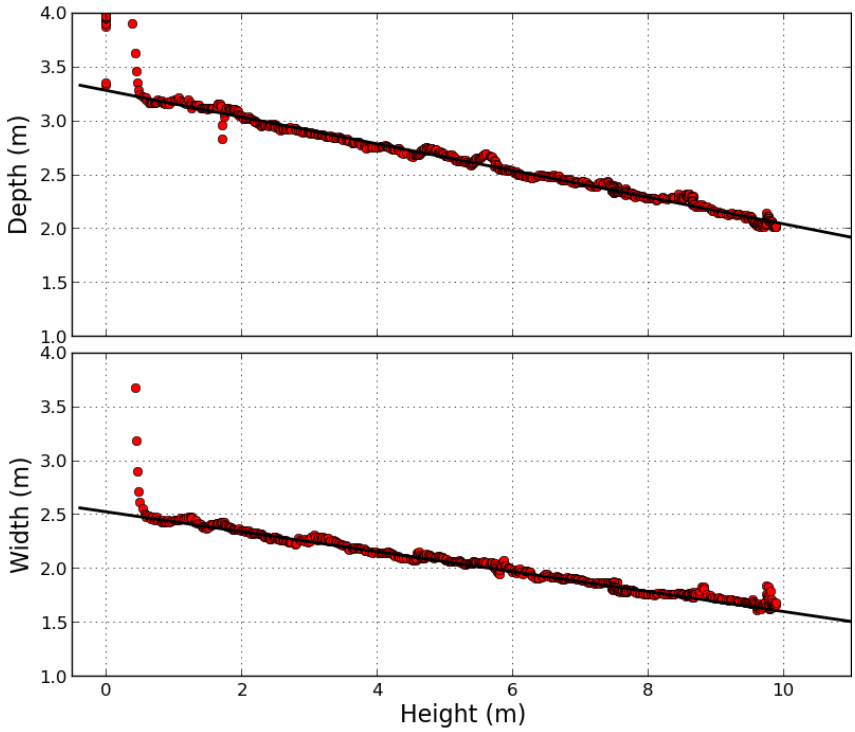


Figure 3.10: The estimated depth and width as a function of the height for the electric tower from Fig. 3.1, fitted with straight lines.

tower's body from the estimated dimensions presented in Fig. 3.10. A simple approach is to approximate each face as a planar segment [7], and the edges of the tower as the intersection of adjacent planes m_j $(j = 1, ..., 4)$, expressed as

$$m_j : \quad a_j x + b_j y + c_j z + d_j = 0, \quad j = 1, \ldots, 4, \tag{3.11}$$

where each m_j is associated with a face of the tower. Obtaining the planes' coefficients results straight-forward, as this is simply a 3D representation of the graphs illustrated in Fig. 3.10 (the slope of the fitted lines are directly related to the slopes of the planes). For example, in this particular case this resulted in

$$\begin{cases} m_1 : & -x - 0.062z - 1.643 = 0 \\ m_2 : & y - 0.046z - 1.265 = 0 \\ m_3 : & x - 0.062z - 1.643 = 0 \\ m_4 : & -y - 0.046z - 1.265 = 0 \end{cases} \tag{3.12}$$

which correspond to the front, right, back and left sides respectively. With respect to an accurate point cloud reconstruction, which would require exploring extensive sections of the electric tower, this simplified planar representation can be obtained with more ease,

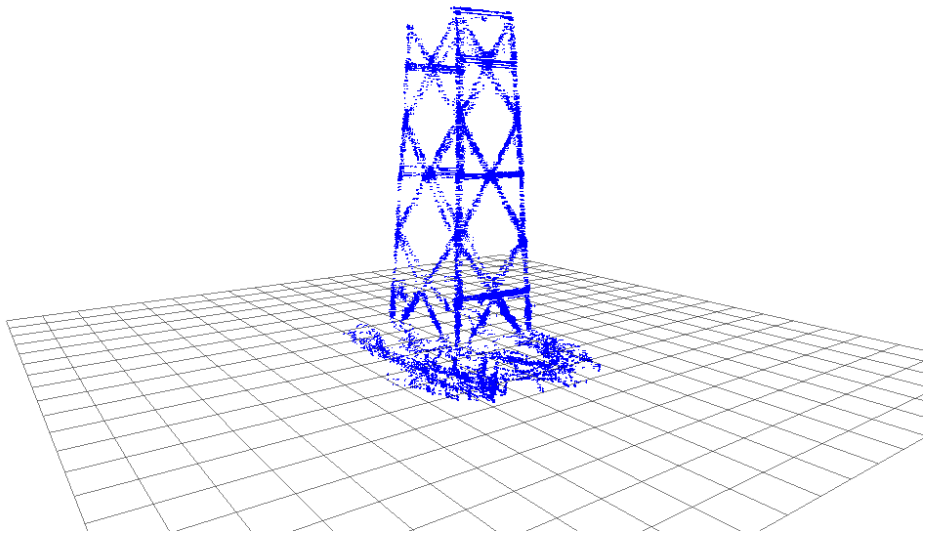


Figure 3.11: Partial 3D point cloud reconstruction of the electric tower from Fig. 3.1, for a vertical flight in front of the tower. The laser scans are aligned using the tracked cross-section's center, the quadrotor's altitude (from the laser altimeter) and attitude (from the IMU measurements).

as it only requires exploring a portion of the tower. As will be seen in Chapter 4, the main importance of these results is that both 3D representations of the tower can be exploited for pose estimation purposes.

3.4.3 Limitations

We now summarize the main limitations observed in the experimental and simulation results. First, throughout the formulation of the tracking approach it was assumed that the cross-sections captured in the scans were rectangular. For this assumption to hold, the scan plane must remain horizontal. This is reasonable for most inspection tasks, where careful inspections require the MAV to operate at low speeds and inclinations remain small. While errors due to such small inclinations can remain within tolerable levels, as shown in the simulation results, external disturbances, such as strong winds, can produce large inclinations and bring the MAV to a configuration where the geometric model from Eq. (3.3) is no longer valid. Under such circumstances, tracking the tower with this approach will result inaccurate.

Another important constraint, demonstrated in Sec. 3.4.1, is that the MAV must always fly on the same side of the tower. This occurs as the entire approach is based on

tracking $\mathcal{L}_{\text{front}}$. Since this line corresponds to the side of the tower closest to the MAV, if the MAV navigates around the tower eventually a different line will be tracked. As illustrated in Fig. 3.9, this causes shifts in the position and orientation estimates, since they are defined with respect to $\mathcal{L}_{\text{front}}$ (Eq. (3.8) and Eq. (3.9)).

Proposed strategy

The constraints imposed on the MAV's motion by this tracking approach are too restrictive for general inspection tasks that may require navigating continuously on all sides of the tower. However, two advantages can be attributed to this method. First, the previously unknown dimensions of the tower can be estimated on-flight (Fig. 3.10 and Fig. 3.11). Second, the laser scans can be processed at fast rates (Fig. 3.8). Thus, to exploit these advantages, an alternative strategy is to divide the inspection task into two steps. A first step consists in modelling the electric tower, which would allow to compensate for the limited information captured by the individual 2D laser scans. The idea is to perform an initial vertical flight in front of the tower, in which our tracking algorithm is capable of providing a quantitative model of the tower (Fig. 3.10 and Fig. 3.11).

Then, a second step would focus on 3D pose estimation and navigation, using the estimated model to track the tower in general flight conditions, and external sensing to estimate the remaining degrees of freedom of the MAV (as in Chapter 2). With such a model-based approach to retrieve pose information from the laser range measurements, the scan plane no longer needs to remain horizontal and less restrictions are imposed on the MAV's movement. This is the main topic of the following chapter, where we will consider that the first modelling step has already been performed based on our tracking approach, and instead focus the discussion on how to recover the complete 3D pose estimates.

3.5 Conclusions

In this chapter, we have presented a method to track the position and orientation of an electric tower from 2D laser range measurements, captured on a flying MAV platform. This method was conceived for electric towers with rectangular cross-sections, typical in high-voltage transmission lines. It consists in gradually extracting notable tower features from the laser scans and using the known geometry of the cross-sections to locate the tower in 2D. The approach requires the scan plane to remain horizontal and the MAV is limited to flying in one side of the tower at a time, which are constraints too restrictive for

general inspection tasks. However, in results from simulations and experimental flights, it was shown that, for a vertical flight in front of the tower, the feature-based approach can process scans at fast rates (with an average of 0.3 ms) and effectively estimate on-flight the previously unknown dimensions of the tower. When combined with additional sensing (height sensor and attitude estimates), it can be used to create 3D geometric representations of the tower, such as a point cloud reconstruction.

CHAPTER 4

Laser-based pose estimation for electric tower inspection: Model-based approach

Contents

4.1 Overview **62**

4.2 Adapting the ICP algorithm to the electric tower scene . . . **63**

4.2.1 First proposed approach: using the tower point cloud reconstruction 64

4.2.2 Second proposed approach: using the tower planar representation 65

4.2.3 Limitations 66

4.3 Proposed pose tracking approach **66**

4.3.1 Laser odometry 67

4.3.2 Height estimation 67

4.3.3 Attitude estimation 70

4.4 Simulation results **71**

4.4.1 Attitude estimation results 72

4.4.2 Laser odometry results 73

4.4.3 Height estimation results 80

4.5 Conclusions **82**

4.1 Overview

The previous chapter presented an initial attempt on addressing the self-localization problem in an electric tower inspection scene, for MAVs relying on 2D LiDARs for range sensing. A feature-based approach was proposed to track the tower cross-sections in the individual laser scans. This method was tailor-fitted to towers with rectangular cross-sections, requiring numerous parametrizations and assumptions that proved too restrictive for localization purposes in general inspection tasks (Sec. 3.4.3).

In this chapter, we explore how traditional scan matching methods, particularly the ICP algorithm [34], can be extended to the electric tower scene to obtain a more general and reliable localization approach. These techniques require the surrounding environment to have sufficient geometric detail and are not suitable for highly unstructured scenarios, often faced outdoors [22]. However, in an outdoor inspection scene, the rigid and well-defined structure of the electric towers have sufficient geometric detail to easily contrast from surrounding unstructured objects. This was exploited in Chapter 3 to track the tower cross-sections in 2D, and will now be used to adapt the ICP algorithm.

As was noted in Sec. 3.2, the conventional approach of aligning pairs of 2D scans through 2D scan matching, which has proven effective in indoors environments (Chapter 2), was not adapted to the electric tower case. Due to the tower's 3D geometry, the planar assumption required to cope with the MAV's 3D motion (Sec. 2.2) does not hold and the overlap between pairs of 2D scans is too limited to allow for efficient pair-wise registrations. Instead, we seek to tackle these limitations by introducing information from external sensing into the registration process and aligning the individual laser scans to a 3D model of the scene acquired beforehand. This builds on the results presented in Sec. 3.4.2.1, where the previously unknown dimensions of the tower could be determined on-flight, which, when combined with external sensing (altitude sensor and IMU), allowed to derive 3D representations of the electric tower (Fig. 3.10 and Fig. 3.11). As before, our main interest are steel lattice towers made up of rectangular cross-sections.

This chapter is organized as follows. Two different adaptations of the ICP algorithm to the electric tower scene are presented in Sec. 4.2. Then, in Sec. 4.3 we discuss our final 3D pose tracking approach for the electric tower scenario, starting with the laser odometry based on the two ICP adaptations in Sec. 4.3.1. Next, regarding height estima-

tion, the previous chapters relied on laser altimeters. In this chapter, we adopt barometer sensors, more appropriate for outdoor scenes, and present a two-stage height estimation scheme in Sec. 4.3.2. Lastly, in Sec. 4.3.3.1 we present gain-scheduling approach for our attitude observer from Sec 2.4.2, which improves estimation performance for changing MAV dynamic conditions. This chapter ends with validations based on simulations in Sec. 4.4.

4.2 Adapting the ICP algorithm to the electric tower scene

Common implementations of the ICP algorithm with 2D laser scans limit the registration process to 2D space, as was discussed in Chapter 2. As will be discussed in Sec. 4.2.1 and Sec. 4.2.2, we instead solve the registrations in 3D space by introducing MAV states estimated separately from external sensing and previous knowledge of the tower's geometry. Since the ICP algorithm can be applied to a wide variety of representations of geometric data such as line sets, triangle sets, parametric surfaces, among others [34], we explore two possible implementations of the ICP algorithm adapted to the electric tower scene.

In the first case (Sec. 4.2.1) we align the individual laser scans to a 3D point cloud reconstruction of the electric tower. This requires very few modifications of the baseline ICP algorithm described in Chapter 2, and profits from the key advantages of ICP, such as simplicity and generality, since no specific parametrization of the tower is used. On the downside, for this approach to be effective, the 3D point cloud must accurately capture the complete electric tower, which is a complex task. With our tracking algorithm from Sec. 3.3 this required exploring extensive portions of the tower, as was illustrated in Fig. 3.11. Existing solutions rely on offline processing of data from powerful and expensive 3D LiDARs capable of capturing dense measurements from long distances [7, 26]. This, however, goes beyond the scope of this work. In the rest of this chapter, we assume that a point cloud reconstruction of the tower was obtained beforehand with our tracking approach from Sec. 3.3.

The difficulties in obtaining an accurate 3D point cloud reconstruction of the inspection scene can render the previous approach impractical. In Sec. 3.4.3 we presented a geometric abstraction of the tower, where each face is represented by a plane, that is simpler to obtain than a complete and accurate point cloud reconstruction. Thus, in Sec. 4.2.2 we present a second approach which aligns the laser scans onto the simplified planar representation

of the tower body from Eq. (3.11). The reasoning behind this approach lies in focusing on the tower cross-sections captured by the laser scans (Fig. 3.2), instead of the individual laser points. The size and shape of the observed cross-section depends directly on the pose of the laser scanner with respect to the tower. We can determine where the scan plane intersects the surface of the tower, and recover the pose of the sensor, if we know the 3D geometry of the tower's contour. The planar representation of the tower provides this information. On the downside, the strategy adopted, which will be explained briefly, is specific to the considered towers with rectangular cross-sections and would have to be changed for a different tower geometry. In contrast, the point cloud approach is more general in this matter, and wouldn't require any modifications.

4.2.1 First proposed approach: using the tower point cloud reconstruction

Let us denote by $\mathcal{I}$ an inertial NED (North-East-Down) frame located at the center of the tower at the ground level, and $\mathcal{B}$ a body-attached frame in the MAV's center of mass, assumed to coincide with the sensor frames for simplicity. Next, consider a vector $\mathbf{X} = (x, y, z, \phi, \theta, \psi)$, such that

$$T_{\mathbf{X}}(\mathbf{p}) = R(\psi, \phi, \theta)\mathbf{p} + \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad \mathbf{p} \in \mathbb{R}^3, \tag{4.1}$$

denotes a 6 DoF rigid body transformation, where the rotation matrix R is defined with the Z-X-Y Euler angle convention (see Eq. (2.12)).

Then, let the current scan be represented by a set of 2D points, denoted $\mathbf{S}_p = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{N_p}\}$, expressed in the body attached frame $\mathcal{B}$. Moreover, let $\mathbf{S}_q = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{N_q}\}$ denote the 3D reference model, expressed in the inertial frame $\mathcal{I}$, which corresponds to a 3D point cloud reconstruction of the inspection scene acquired beforehand, e.g., from our tracking approach as discussed in the Chapter 3. Given $T_{\mathbf{X_0}}$, an initial rough estimate of the 6 DoF rigid body transformation from $\mathcal{B}$ to $\mathcal{I}$, such that $\mathbf{X_0} = (x_0, y_0, z_0, \phi_0, \theta_0, \psi_0)$, the goal is to refine this guess by aligning $\mathbf{S_p}$ to $\mathbf{S_q}$. The baseline ICP [34] was used, with several modifications, notably in the minimization step. Each iteration k (starting from $k = 0$) is carried out as follows:

1. **Initialization:** The current estimate $T_{\mathbf{X}_k}$ is used to transform all points $\mathbf{p}_i \in \mathbf{S}_p$

into 3D coordinates in the inertial frame $\mathcal{I}$, obtaining $\mathbf{p}'_i \in \mathbf{S}_{p'}$.

2. **Matching:** Corresponding pairs $(\mathbf{p}'_i, \mathbf{q}_i)$ are established by associating each point in $\mathbf{S}_{p'}$ to the closest point in $\mathbf{S}_q$. This correspondence search is the most time consuming step of the algorithm [34]. Thus, K-D trees are used to speed up the matching process, as is commonly done with ICP [34,37]. This is a space-partitioning data structure that organizes points in a binary tree structure for efficient closest point searches.

3. **Rejection:** Point pairs separated by more than a fixed distance threshold d_{min} are removed. This is mainly helpful with accuracy and stability in the presence of outliers [37], which in this case are typically due to surrounding vegetation.

4. **Minimization:** The goal is to find the transformation $T_{\mathbf{X}_{\min}}$ that minimizes the sum of squared errors, using the Euclidean distance as the distance metric [34]. For the N remaining point pairs $(\mathbf{p}'_i, \mathbf{q}_i)$, this leads to the following optimization problem:

$$\begin{aligned} \mathbf{X}_{\min} = \arg\min_{\mathbf{X}} \sum_{i=1}^{N} \|T_{\mathbf{x}}(\mathbf{p}'_i) - \mathbf{q}_i\|^2, \\ \text{such that } (\phi, \theta) = (0, 0), \end{aligned} \tag{4.2}$$

   which is solved with the Levenberg-Marquardt algorithm, since it allows obtaining accurate results and deals with initialization errors without significant speed losses [105]. Note that the components $\{\phi, \theta\}$ of $\mathbf{X}$ are neglected during the minimization. The underlying assumption is that $\{\phi_0, \theta_0\}$ from the initial guess $T_{\mathbf{X_0}}$ are precise and reliable. The reasoning behind this assumption will be clarified in Sec. 4.3.1. Thus, by reducing the optimization problem from a 6-dimensional space to a 4-dimensional space, this limits the risk of divergence due to local minima, and provides a more stable and reliable solution. This is the main modification of the algorithm.

5. Finally, the current estimate is updated as

$$T_{\mathbf{X}_{k+1}} = T_{\mathbf{X}_{\min}} \cdot T_{\mathbf{X}_k} \tag{4.3}$$

Upon convergence, only the $\{x, y, z, \psi\}$ components of the initial guess $T_{\mathbf{X_0}}$ are updated (due to Eq. (4.2)). The main novelty is that altitude information can now be recovered, despite using 2D laser scans, which is a direct consequence of introducing a 3D point cloud reconstruction of the tower as a reference model in the registration process. As will be

seen in Sec. 4.3.1 and in the simulation results from Sec. 4.4.2, the key of this formulation is what source of information to use for each state of $\mathbf{X_0}$.

4.2.2 Second proposed approach: using the tower planar representation

Here, we seek to align the laser scans onto the simplified planar representation of the tower body from Eq. (3.11). To achieve this, we adopt a projection-based matching strategy [114, 115], where, after initialization (step 1 of of Sec. 4.2.1), the corresponding points $\mathbf{q}_i$ are calculated as the orthogonal projection of every point $\mathbf{p}'_i \in \mathbf{S}_{p'}$ onto the closest planar segment from m_j. This substitutes the time-consuming correspondence search previously used and allows obtaining significant speed gains [37], as will be shown in the simulation results (Sec. 4.4). Thus, in this approach, the matching step (step 2 of Sec. 4.2.1) for each point $\mathbf{p}'_i$ is now carried out as follows

- For the tower face m_j (starting with $j = 1$), calculate the two edge lines $\mathcal{L}_A$ and $\mathcal{L}_B$ as the intersection with the two adjacent planes.
- Project $\mathbf{p}'_i$ orthogonally to the plane equation of m_j (Eq. (3.11)), obtaining $\mathbf{q}$. We have to determine if $\mathbf{q}$ falls within the planar segment delimited by $\mathcal{L}_A$ and $\mathcal{L}_B$. This is done as follows:
  - Project $\mathbf{p}'_i$ to the edge lines $\mathcal{L}_A$ and $\mathcal{L}_B$, obtaining $\mathbf{q}_A$ and $\mathbf{q}_B$ respectively.
  - Let $\overline{AB} = \mathbf{q}_B - \mathbf{q}_A$.
  - Calculate the normalized projection $\rho = \frac{(\mathbf{q}-\mathbf{q}_A)\cdot(\overline{AB})}{\|\overline{AB}\|^2}$.
  - If $0 < \rho < 1$, then $\mathbf{q}$ falls within the planar segment, and the projection is $\mathbf{q}$.
  - If $\rho \leq 0$, then $\mathbf{q}$ falls outside of the planar segment and the projection is $\mathbf{q}_A$.
  - If $\rho \geq 1$, then $\mathbf{q}$ falls outside of the planar segment and the projection is $\mathbf{q}_B$.

These steps are repeated for the four faces of the tower, and the projected point which yields the minimum distance to $\mathbf{p}'_i$ is chosen as the corresponding point $\mathbf{q}_i$. Then, the remaining steps from the previous implementation are left unchanged. As before, only the $\{x, y, z, \psi\}$ components of the initial guess are updated.

4.2.3 Limitations

The inherent limitations of the ICP algorithm discussed in Sec. 2.3.2 also apply in this case. Moreover, while the main novelty of both approaches is the capability of recovering altitude information in the scan registrations, these altitude estimates suffer from several drawbacks. For a 2D LiDAR, measurements from the individual scans fall within the same plane and don't directly capture the MAV's altitude, which is determined uniquely from the point correspondences with the 3D point cloud. Convergence issues characteristic of the ICP algorithm, such as convergence to wrong local minima and sensitivity to input geometry, are thus amplified for the altitude estimates. In particular, altitude estimation is highly dependant on the inclination of the faces of the tower. In the worst case scenario, no altitude information can be recovered for completely vertical faces, which is a situation rarely faced with high voltage electric towers considered in this work. Nonetheless, these drawbacks justify the use of an additional barometer sensor. However, as will be seen in the simulation results (Sec. 4.4), both ICP implementations will overall perform well if the electric tower remains within the sensor's field of view, and particularly stable results can be achieved for near-hovering conditions. This quality holds for altitude estimates, and will be exploited to track the barometer drift in Sec. 4.3.2.2

4.3 Proposed pose tracking approach

In this section, we present how to track the MAV's 6 DoF pose with our sensor setup: a 2D laser scanner, an IMU (gyrometer and accelerometer) and a barometer sensor. As is typically done with MAVs, the estimation process is broken down into several components [20, 22]. Recalling that the complete 6 DoF pose from $\mathcal{B}$ to the inertial frame $\mathcal{I}$ is described by $\{x, y, z, \phi, \theta, \psi\}$. First, Sec. 4.3.1 explains how $\{x, y, z, \psi\}$ are estimated from the laser range measurements. Then, Sec. 4.3.2 presents a height estimation scheme that fuses barometer measurements with altitude estimates recovered from the scan registrations. Finally, in Sec. 4.3.3, $\{\phi, \theta\}$ are obtained by fusing accelerometer and gyrometer measurements from the IMU. We now explain each component of the pose tracking process.

4.3.1 Laser odometry

Both ICP-based approaches proposed in Sec. 4.2 were considered for the laser odometry. As previously explained, they provide estimates of the $\{x, y, z, \psi\}$ components of the MAV's pose. In both approaches, the roll and pith angles $\{\phi, \theta\}$ were left out of the optimization process (Eq. (4.2)). We consider that these states are estimated separately from IMU measurements, as will be discussed in Sec. 4.3.3. In practice, such estimates are of high quality, and are exploited to provide an accurate and reliable initial guess $\{\phi_0, \theta_0\}$ for the scan registrations, as was assumed in Sec. 4.2.1. This is the key to reducing the dimension of the optimization problem (Eq. (4.2)) and obtaining a more stable solution.

Next, the main novelty from the ICP formulations presented in Sec. 4.2.1 and Sec. 4.2.2 was the capability of recovering altitude information from the scan registrations However, as was discussed in Sec. 4.2.3, these estimates can result unreliable due to the planar nature of the 2D laser scans and the high dependence on the tower's shape. We thus study two different cases to assess the quality of these altitude estimates, and to determine a proper way of exploiting them:

- **Using laser odometry and an IMU:** For each laser scan, the registration is initialized with $\{x_0, y_0, z_0, \psi_0\}$ from the previous scan registration and $\{\phi_0, \theta_0\}$ from the attitude estimator (Sec. 4.3.3). The simulation results for this case will be presented in Sec. 4.4.2.1.
- **Using laser odometry, an IMU and an additional altitude sensor:** For each laser scan, the registration is initialized with $\{x_0, y_0, \psi_0\}$ from the previous scan registration, $\{\phi_0, \theta_0\}$ from the attitude estimator (Sec. 4.3.3) and $\{z_0\}$ from the external altitude source. As will be seen in Sec. 4.3.2.2, this altitude source is an observer that fuses barometer measurements with laser altitude estimates. The simulation results for this case will be presented in Sec. 4.4.2.2.

Both cases will be tested for both proposed ICP-based approaches in Sec. 4.4.2. In any case, the output of the laser odometry is $\{x_{\text{laser}}, y_{\text{laser}}, z_{\text{laser}}, \psi_{\text{laser}}\}$.

4.3.2 Height estimation

In Sec. 4.2.3 it was noted that the altitude estimates recovered from both scan registration approaches can result unreliable due to the strong dependence on the shape of the

tower and the planar nature of the 2D laser scans. Here, we seek an effective way of exploiting this limited altitude information by coupling with external height sensing. In Chapter 2 and Chapter 3, a laser altimeter was used to directly measure and estimate the MAV's height. However, the performance of laser altimeters depends highly on the floor's layout, which can be very irregular in typical outdoor environments. Instead, we pursue using pressure height sensors, a common alternative for outdoor scenes. As discussed in Sec. 1.2.2.2, barometer measurements are independent from the shape of surrounding structures, but suffer from drift over time due to varying atmospheric conditions.

Thus, the goal is to combine both sources of altitude information, i.e., barometer readings and altitude estimates from the scan registrations, to tackle their respective drawbacks. Recalling the vertical dynamics in the inertial frame $\mathcal{I}$ from Eq. (5.9),

$$\begin{cases} \dot{z} &= v_z \\ \dot{v}_z &= g + \frac{F_z}{m} \end{cases} \tag{4.4}$$

where v_z is the vertical velocity of the MAV's body frame $\mathcal{B}$ with respect to $\mathcal{I}$, expressed in $\mathcal{I}$. Then, m is the MAV's mass and F_z is the vertical component of the aerodynamic forces acting on the MAV, expressed in $\mathcal{I}$. Satisfying estimates of v_z can be recovered from barometer and accelerometer measurements [51, 52]. As will be shown in the simulation results from Sec. 4.4.3, these estimates remain accurate even in the presence of barometer drift. The altitude estimates from the laser scan registrations are not included in the vertical velocity estimation as it was noted that they only degrade the accuracy.

We propose a two-stage system of observers to recover accurate and robust estimates of the MAV's altitude. In the first stage, an observer determines the vertical velocity by fusing barometer and accelerometer readings. Then, the estimated velocity is fed into the second stage, which combines the barometer readings and the laser odometry altitude estimates, to recover a unique, filtered altitude estimate. In the process, an estimate of the barometer's drift is also recovered. We now explain each of the stages of the height estimation process.

4.3.2.1 First stage: Vertical velocity observer

From the MAV's vertical dynamics (Eq. (4.4)), we now formulate the following second-order state observer

$$\begin{cases} \dot{\hat{z}} = \hat{v}_z - k_z(\hat{z} - z_{\text{baro}}) \\ \dot{\hat{v}}_z = g + \dfrac{F_z}{m} - k_{v_z}(\hat{z} - z_{\text{baro}}), \quad k_z, k_{v_z} > 0 \end{cases} \tag{4.5}$$

where (k_z, k_{v_z}) are the observer gains, z_{baro} are the barometer altitude measurements. Recalling from Eq. (5.9), the complete external aerodynamic forces acting on the MAV are $\mathbf{F} = (F_x, F_y, F_z)^\intercal$, then F_z is determined from the accelerometer readings $\boldsymbol{a}_m$ and the estimated attitude $\hat{R}$ (from Eq. (2.21) applied to the gain-scheduled observer from Sec. 4.3.3) as

$$\mathbf{F} = m\hat{R}\boldsymbol{a}_m. \tag{4.6}$$

4.3.2.2 Second stage: Fusing barometer and laser odometry measurements

In this second stage, v_z from Eq. (4.4) is considered as a known input, which is provided by the observer from the first stage (Eq. (4.5)). We instead use the following system

$$\begin{cases} \dot{z} &= v_z \\ \dot{b}_z &= 0, \end{cases} \tag{4.7}$$

where b_z is the unknown barometer drift, which, in practice, varies slowly with time and is modelled as a constant.

From Eq. (4.7), a simple second-order observer can be formulated as

$$\begin{cases} \dot{\hat{z}} &= v_z - k_z(\hat{z} - \overline{z}_1) \\ \dot{\hat{b}}_z &= -k_{b_z}(\hat{z} - \overline{z}_2) \end{cases} \tag{4.8}$$

where (k_z, k_{b_z}) are the estimation gains, and $\overline{z}_n$ is an auxiliary variable defined as

$$\overline{z}_n = \lambda_n(z_{\text{baro}} - \hat{b}_z) + (1 - \lambda_n)z_{\text{laser}} \quad \text{with} \quad \begin{aligned} &0 \leq \lambda_n \leq 1, \\ &n = 1, 2. \end{aligned} \tag{4.9}$$

which is the weighted sum of the laser altitude estimates z_{laser} and barometer readings z_{baro} compensated with the estimated bias $\hat{b}_z$. The weights λ_n allow one to determine how each sensor contributes to the estimation of each state. In particular, as λ_n increases, higher

priority is given to the barometer readings. The reasoning behind this parametrization is to use the laser estimates mainly to keep track of slowly varying barometer bias $\hat{b}_z$, and to maintain the more reliable barometer measurements to estimate $\hat{z}$. Choosing the weights $\lambda_1 = 1$ and $\lambda_2 = 0$ achieves this purpose.

To analyse the stability of the altitude observer formulation from Eq. (4.8), we first deduce error dynamics of the system. Modelling the system inputs as

$$\begin{aligned} z_{\text{baro}} &= z + b_z \\ z_{\text{laser}} &= z. \end{aligned} \tag{4.10}$$

Then, by substituting Eq. (4.10) and Eq. (4.9) in Eq. (4.8), and subtracting the vertical dynamics from Eq. (4.4), the error dynamics of the system are

$$\begin{cases} \dot{\tilde{z}} &= -k_z(\tilde{z} + \lambda_1 \tilde{b}_z) \\ \dot{\tilde{b}}_z &= -k_{b_z}(\tilde{z} + \lambda_2 \tilde{b}_z) \end{cases} \tag{4.11}$$

where $\tilde{z} = \hat{z} - z$ and $\tilde{b}_z = \hat{b}_z - b_z$. In matrix form, this is expressed as

$$\begin{bmatrix} \dot{\tilde{z}} \\ \dot{\tilde{b}}_z \end{bmatrix} = \begin{bmatrix} -k_z & -\lambda_1 k_z \\ -k_{b_z} & -\lambda_2 k_{b_z} \end{bmatrix} \begin{bmatrix} \tilde{z} \\ \tilde{b}_z \end{bmatrix} = \tilde{\mathbf{A}} \begin{bmatrix} \tilde{z} \\ \tilde{b}_z \end{bmatrix}. \tag{4.12}$$

Stability analysis follows, by verifying that the conditions $\det(\tilde{\mathbf{A}}) > 0$ and $tr(\tilde{\mathbf{A}}) < 0$ hold. For the choice of weights $\lambda_1 = 1$ and $\lambda_2 = 0$, this leads to

$$\begin{cases} k_z = 2\zeta\omega_n \\ k_{b_z} = -\dfrac{\omega_n}{2\zeta}, \end{cases} \tag{4.13}$$

where the damping ratio ζ and the natural frequency ω_n define the closed-loop poles, which have negative real part if $\zeta, \omega_n > 0$. More details on how to tune the gains (k_z, k_{b_z}) are given in the Appendix.

4.3.3 Attitude estimation

For attitude estimation, we use the same non-linear observer formulation from Eq. (2.18) (described in Sec. 2.4.2). However, the estimated roll and pitch angles from the IMU measurements were introduced in the scan registration process (Sec. 4.3.1). Now, angular

errors have a direct influence in the quality of the scan registrations and more attention has to be paid to possible sources of attitude estimation errors.

The attitude observer from Eq. (2.18) was based on the assumption that accelerometer readings provide direct observations of the roll and pitch angles by measuring the gravity vector (Eq.(2.17)). This approximation is commonly used in attitude estimation when dealing with accelerometers [54], but only holds when flying at constant velocity or near stationary flight conditions. A typical source of error are high acceleration states where the assumption from Eq.(2.17) does not hold. A simple improvement to the observer's performance can be obtained by adapting the estimation gains to such changing MAV dynamic states, i.e., *gain scheduling* [55, 116].

4.3.3.1 Gain scheduling

An added benefit of non-linear observer formulations is that the estimation gains can be tuned in real-time during flight [25]. This can be exploited to adapt the observer to changing dynamic conditions, in particular, to high acceleration states where the assumption from Eq.(2.17) is no longer valid and estimation performance is deteriorated. In such situations, which typically last for short periods of time, it is better to lower the estimation gains and to rely on the gyrometer measurements since they are scarcely affected by the linear accelerations [116] and can provide short-term rotations accurately [57].

A basic strategy consists in detecting highly accelerated states by comparing the magnitude of the accelerometer readings to the gravity acceleration [55,56,116]. Let $\tilde{a}_m$ denote the absolute accelerometer measurement error with respect to gravity as

$$\tilde{a}_m = |\|\boldsymbol{a}_m\| - g|, \quad g = 9.81\ \frac{m}{s^2}. \tag{4.14}$$

This magnitude provides a simple criteria to determine the dynamic state of the MAV, as $\tilde{a}_m \approx 0$ for near-hovering conditions, and large values of $\tilde{a}_m$ correspond to highly dynamic motion. The estimation gains can then be adapted accordingly. Yoo et al. [55] adopt a simple switching strategy to choose the gain between a set of nominal values corresponding to no-acceleration, low-acceleration or high-acceleration states. Instead, Valenti et al. [116] set a nominal gain for hovering state, which is then decreased linearly during transitions to high acceleration states. We adopt a strategy similar to [116]. Recalling the non-linear observer from Eq. (2.18)

$$\dot{\hat{\boldsymbol{\gamma}}} = \hat{\boldsymbol{\gamma}} \times (\boldsymbol{\omega}_m - k_\gamma(\mathbf{a}_m \times \hat{\boldsymbol{\gamma}})), \quad k_\gamma > 0.$$

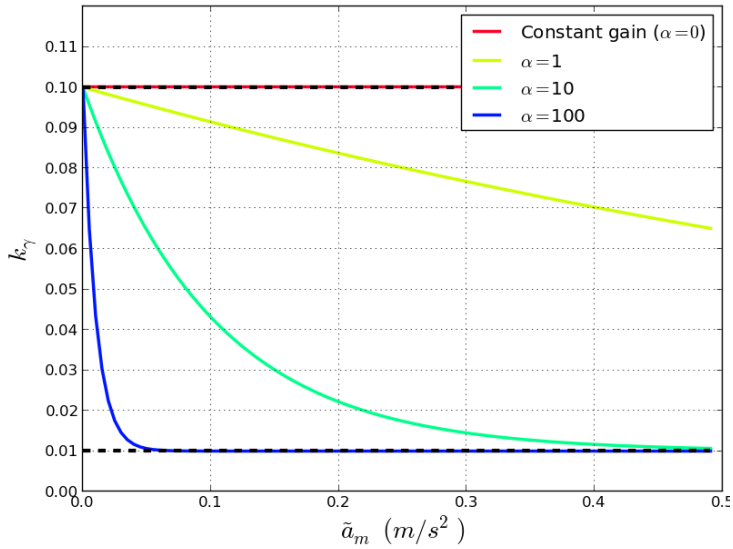


Figure 4.1: An example of the proposed gain scheduling approach for the attitude observer from Eq. (2.18), according to Eq. (4.15), for different values of α. This parameter determines the steepness of the transitions between the nominal gains k_L and k_H. As $\tilde{a}_m$ increases, the gains k_γ decrease from k_L towards k_H. For higher values of α the transitions are faster.

For the observer gain k_γ, the following gain scheduling approach is proposed

$$k_\gamma(\tilde{a}_m) = k_L e^{-\alpha \tilde{a}_m} + k_H(1 - e^{-\alpha \tilde{a}_m}), \quad \alpha > 0, \tag{4.15}$$

where k_L and k_H denote the nominal gains during low and high acceleration states respectively, and α is an arbitrary positive constant that determines the steepness of the transitions between k_L and k_H. The idea is to transition smoothly between these nominal gains. This is illustrated on Fig. 4.1 for $k_L = 0.1$, $k_H = 0.01$ and different values of α. It can be noted that $\alpha = 0$ corresponds to the constant gain case, and as α increases, the gains decrease faster towards k_H. Furthermore, as $\tilde{a}_m \approx 0$, then k_γ remains near k_L, and as $\tilde{a}_m$ increases, then k_γ decreases towards k_H, which is the desired behaviour.

4.4 Simulation results

As in the previous chapter, simulations were carried out using the Gazebo simulation environment [111] and ROS as an interfacing middleware [112]. The quadrotor kinematics and dynamics were simulated from the Hector quadrotor stack from ROS [113]. Regarding the sensors, the simulated IMU published gyrometer and accelerometer readings at 100 Hz, and the barometer sensor provided measurements at 20 Hz. A 2D laser scanner was

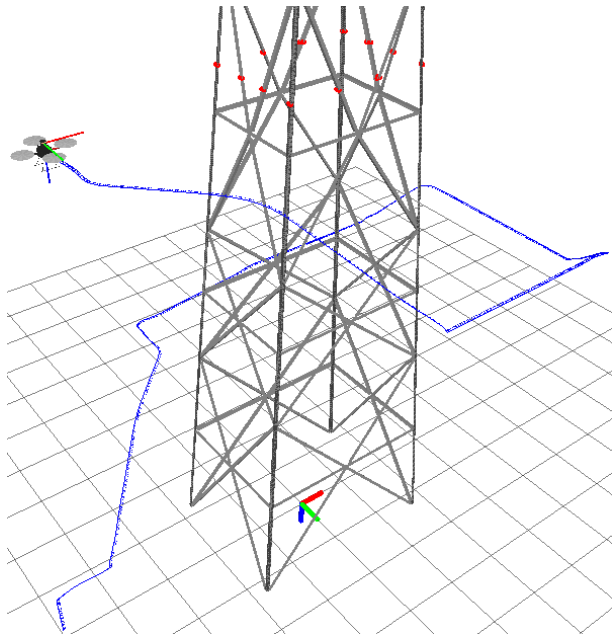


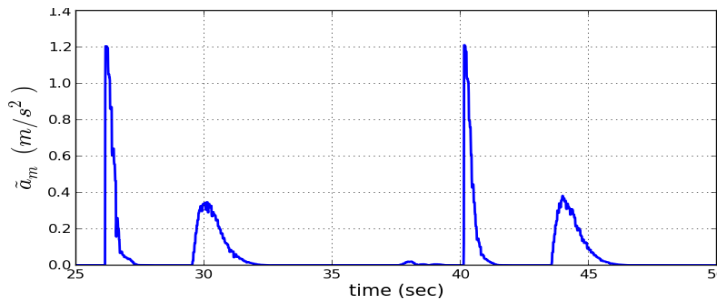
Figure 4.2: The simulated flight around the tower. The blue line indicates the trajectory. Throughout the flight the quadrotor was oriented towards the center of the tower.

mounted horizontally on top of the simulated quadrotor, providing readings at 40 Hz, with 0.25° angular resolution and 270° field of view (thus 1080 measurements per scan). All algorithm development was done using C++, and the registration module from the open source Point Cloud Library (PCL) [98].

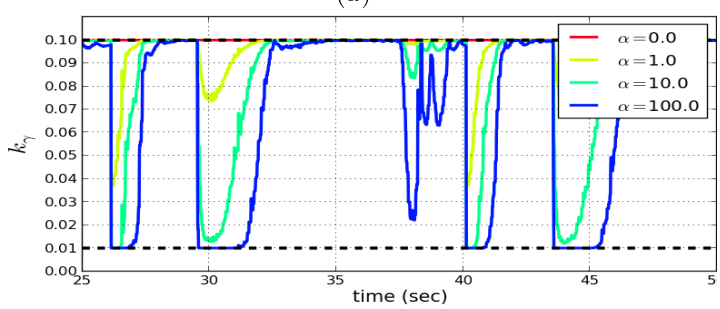
For these flights, a set of waypoints was given for the quadrotor to follow, accounting for a complete displacement around a CAD model of an electric tower (see Fig. 3.5). Meanwhile, the MAV's yaw angle was oriented towards the center of the tower, so that the latter remains in the LiDAR's field of view. Since the focus of this section is to assess the quality of the pose estimates, the simulation ground truth is directly used to stabilize the MAV's position and attitude. The complete flight is shown in Fig. 4.2.

4.4.1 Attitude estimation results

The attitude observer from Eq. (2.18) was used to fuse the accelerometer and gyrometer measurements and recover estimates of the roll and pitch angles $\{\phi, \theta\}$. We now analyse the performance of the proposed gain scheduling approach from Sec. 4.3.3.1. Based on results observed in the simulations, the nominal gains from Eq. (4.15) were set to $k_L = 0.1$ and $k_H = 0.01$. The estimation process was repeated for different values of the parameter α, starting from $\alpha = 0$, which corresponds to the constant gain case (since $k_\gamma = k_L$ from



(a)



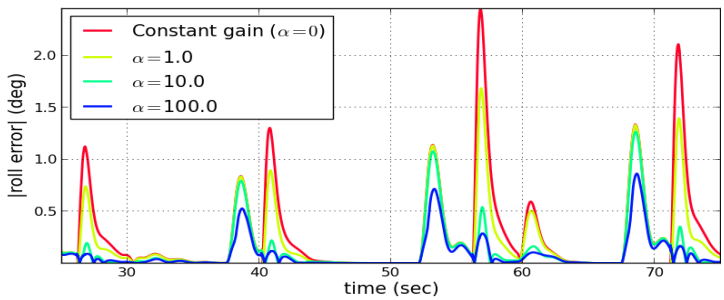
(b)

Figure 4.3: For a portion of the flight from Fig. 4.2: **(a)** The deviations of the accelerometer readings from gravity according to Eq. (4.14). **(b)** The resulting scheduled gains for different values of α. The gains become more reactive for larger values of α.

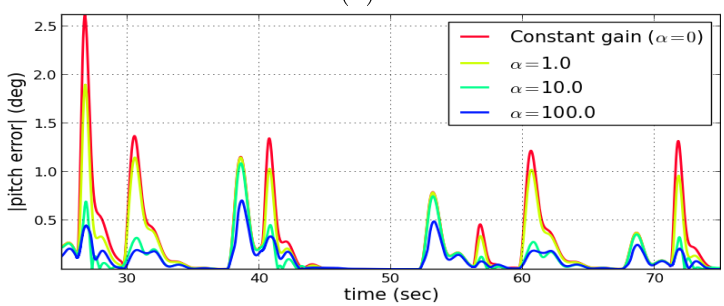
Eq. (4.15)), to $\alpha = 100$. As explained, this parameter determines the steepness of the transitions between the two nominal gains.

Then, Fig. 4.3a illustrates the deviations of the accelerometer readings from the acceleration of gravity ($\tilde{a}_m$ from Eq. (4.14)) for a portion of the flight. As can be seen, the MAV spends larger amounts of time in low acceleration states ($\tilde{a}_m \approx 0$). The peaks in $\tilde{a}_m$ correspond to instants when the MAV accelerates towards a different waypoint. The gain scheduling approach from Eq. (4.15) adapts the observer to these peaks by lowering the estimation gain k_γ. The resulting scheduled gains for the same portion of the flight are shown on Fig. 4.3b. When comparing the two figures, it can be noted the gain k_γ rapidly drops in the presence of acceleration peaks (e.g., $t = 26$ s and $t = 40$ s), which is the desired behaviour. However, as α increases, the gains can result overly sensitive to changes in $\tilde{a}_m$. This can lead to excessive transitions for small changes in $\tilde{a}_m$ (e.g., $t = 38$ s for $\alpha = 100$ in Fig. 4.3b) and unnecessary prolonged low gain states .

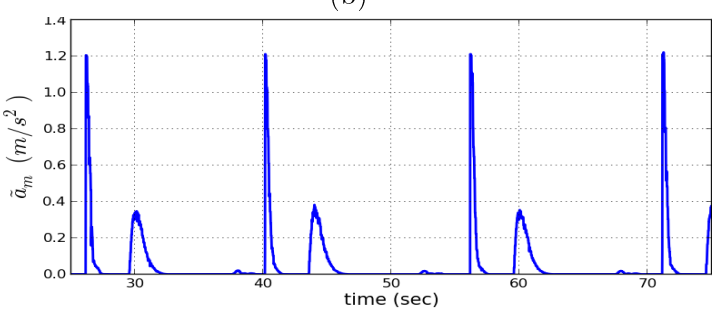
Next, the absolute attitude estimation errors with respect to the simulation ground truth are shown on Fig. 4.4. When comparing Fig. 4.4a and Fig. 4.4b to Fig. 4.4c, it can be noted the largest estimation errors correspond to peaks in $\tilde{a}_m$. The maximum errors are obtained in the constant gain case, reaching 2.45° for the roll angle ($t = 55$ s in



(a)

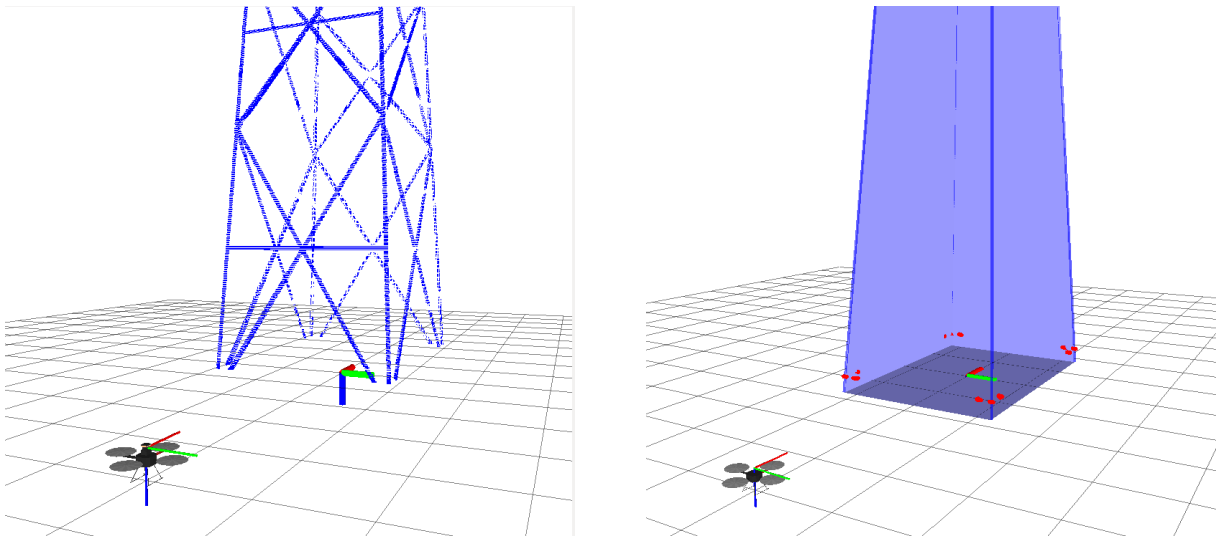


(b)



(c)

Figure 4.4: For the attitude observer from Eq. (2.18) and different values of α: **(a)** Absolute roll estimation error. **(b)** Absolute pitch estimation error. **(c)** The corresponding $\tilde{a}_m$. As α increases, the errors caused by peaks in $\tilde{a}_m$ are reduced.



(a) (b)

Figure 4.5: The two models used in the simulations as references for the ICP algorithm to align the laser scans: **(a)** Point cloud reconstruction. **(b)** Planar model.

Fig. 4.4a) and 2.62° for the pitch angle ($t = 26$ s in Fig. 4.4b). In contrast, in these same time instances the errors are greatly reduced as the parameter α is increased (by up to 2° for $\alpha = 100$). In general, error peaks related to $\tilde{a}_m$ are largely suppressed and the overall performance is improved with the simple gain scheduling strategy.

Based on these observations, a gain scheduled attitude estimation with $\alpha = 10$ was used for the following experiences. This choice offers a good trade-off between sensibility to changes in $\tilde{a}_m$, as seen in Fig. 4.4, and estimation error reduction, as shown in Fig. 4.3.

4.4.2 Laser odometry results

The two proposed implementations of the ICP algorithm were tested in the simulations to recover estimates of the remaining states $\{x, y, z, \psi\}$. In the first case, the laser scans were aligned to the 3D point cloud reconstruction of the tower shown in Fig. 4.5a, which was obtained beforehand with our tracking approach (see Fig. 3.11). In the second case, the laser scans were instead aligned to the planar representation of the tower illustrated in Fig. 4.5b, which was also obtained beforehand with our tracking approach (see Fig. 3.10

and Eq. (3.3)). Here, the plane coefficients from Eq. (3.3) resulted in

$$\begin{cases} m_1: & -x - 0.076z - 1.749 = 0 \\ m_2: & y - 0.046z - 1.219 = 0 \\ m_3: & x - 0.076z - 1.749 = 0 \\ m_4: & -y - 0.046z - 1.219 = 0 \end{cases} \tag{4.16}$$

which correspond to the front, right, back and left faces of the tower, respectively.

As explained in Sec. 4.3.1, two different experiments were performed. In the first case, the scan registrations were aided with attitude estimates recovered from IMU measurements, as discussed in Sec. 4.3.3. Then, in the second experience, the scan registrations are further aided with altitude estimates from the laser-barometer altitude fusion presented in Sec. 4.3.2. In both cases, an initial rough knowledge of the MAV's position with respect to the tower was given for the first laser scan. We now present the results for both experiments.

4.4.2.1 Laser odometry without the altitude observer

The results for this case are shown in Fig. 4.6. First, Fig. 4.6a presents the trajectory followed by the MAV, and compares the MAV's ground truth position with the estimates from both ICP implementations. As can be seen, the results obtained with the planar model approach effectively follow the ground truth for the duration of the flight. However, the point cloud approach ultimately fails before completing the flight. This can be further observed from the absolute errors shown in Fig. 4.6. For the planar model case, the $\{x, y\}$ errors remain below 5 cm (Fig. 4.6c and Fig. 4.6d). Furthermore, the yaw estimates are also very precise, with a maximum error of 0.8° (Fig. 4.6b). In these figures it can be noted that the point cloud approach initially achieves similar performance before completely failing (near $t = 40$ s).

Particular attention must be given to the altitude estimation errors from Fig. 4.6e. As pointed out in Sec. 4.2.3, the horizontally placed 2D laser scanner captures very limited altitude information. It was observed throughout the simulations that the altitude estimates were easily deteriorated in complicated situations, for example, when the horizontal bars block most of the tower's cross-section from the sensor's view (as in Fig. 3.2d). For the planar model case, this typically caused spurious estimates, with the absolute altitude error jumping above 20 cm (e.g., $t = 20$ s and $t = 30$ s in Fig. 4.6e). For the point cloud case, this caused even larger altitude errors (around $t = 30$ s in Fig. 4.6e) which

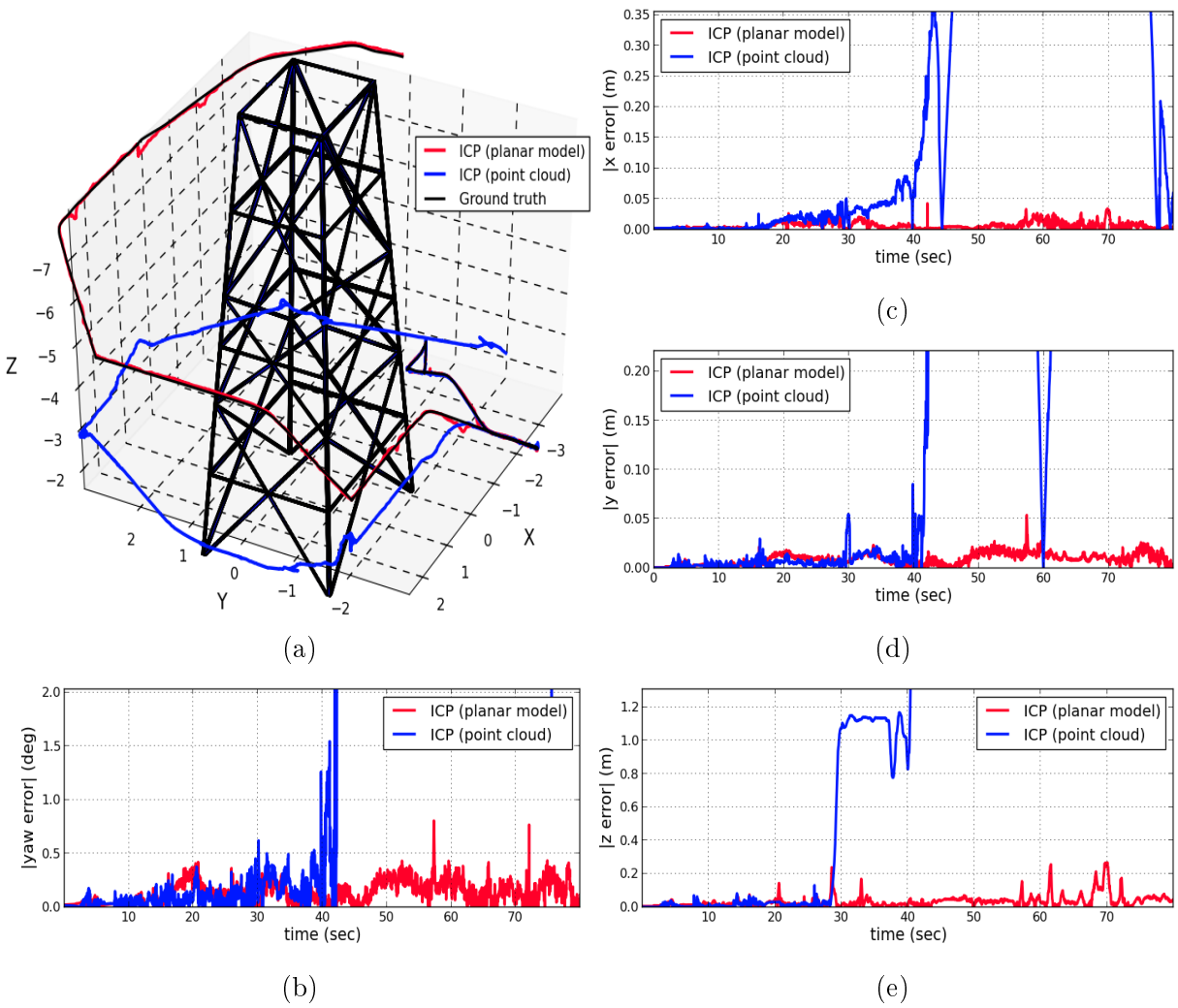


Figure 4.6: Laser odometry results for the simulated flight from Fig. 4.2. For both ICP implementations: **(a)** Comparing the laser odometry outputs with the ground truth. The point cloud approach fails before finishing the flight. **(b)-(e)** Absolute estimation errors with respect to the simulation ground truth.

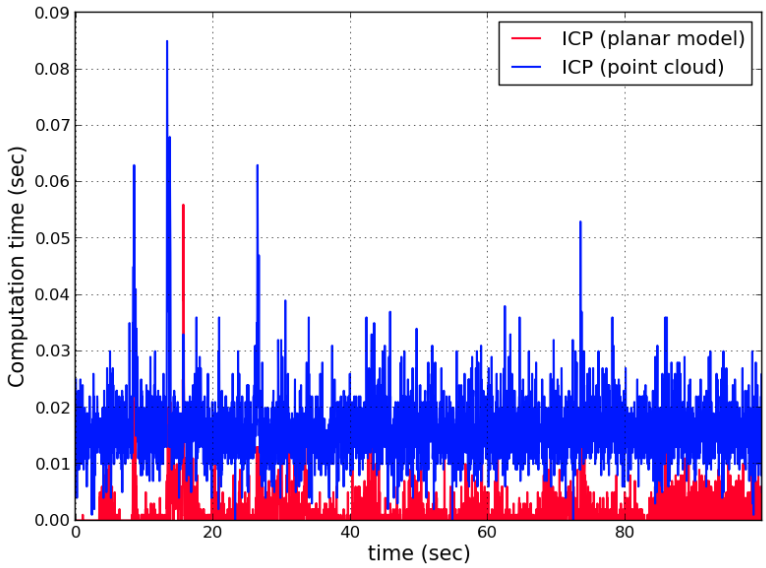


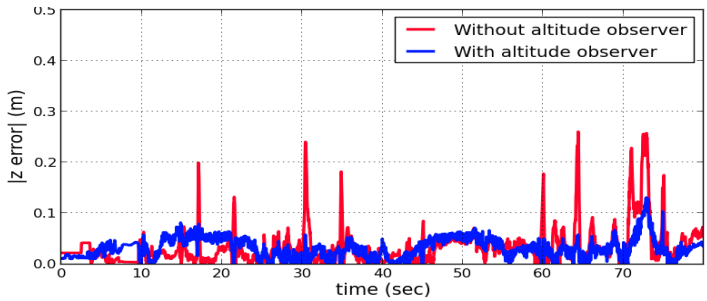
Figure 4.7: Comparing the computation time for the laser scan registrations. The planar model approach (in red) is approximately ten times faster. The significant speed gains are the result of the projection-based matching strategy adopted in Sec. 4.2.2.

eventually lead the complete approach to fail ($t = 40$ s in Fig. 4.6b-Fig. 4.6d) before finishing the flight. Thus, these results validate the observations from Sec. 4.2.3 regarding the limitations of the scan registrations. It can be concluded that using this altitude information directly, e.g., for control purposes, is inadequate. More reliable results can be obtained by fusing these altitude estimates with barometer readings, as will be presented in Sec. 4.4.2.2.

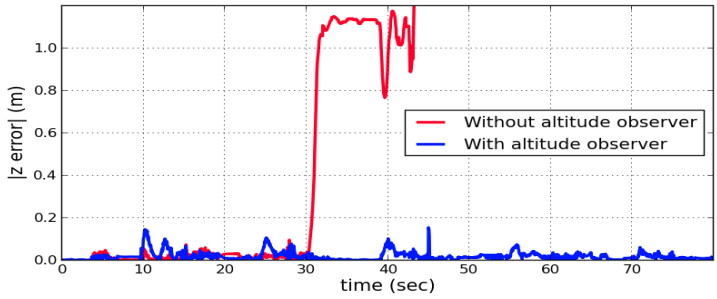
Lastly, the computation time required for the scan registration in both cases is shown in Fig. 4.7. As expected, using the planar model results in significantly faster estimates with an average of 1.4 ms, compared to the point cloud case average of 16 ms. This shows the effectiveness of the projection-based matching strategy adopted in Sec. 4.2.2 to establish point correspondences, which avoids the high computational cost of the extensive correspondence search required for the point cloud registration.

4.4.2.2 Laser odometry with the aid of the altitude observer

We now analyse the performance of the laser odometry from the two ICP-based approaches with the aid of the altitude observer from Eq. (4.8), which fuses the altitude estimates from the laser odometry with barometer readings. In the following results, the observer's filtered output was used at each ICP initialization, instead of the laser altitude estimates



(a) ICP with planar model.



(b) ICP with point cloud model.

Figure 4.8: Comparing the laser odometry altitude estimates without the aid of the altitude observer (in red) from Fig. 4.6e and with the aid of the altitude observer (in blue). **(a)** In the planar model approach spurious peaks are reduced. **(b)** In the point cloud approach, large altitude errors are avoided (at $t = 30$ s) and the approach no longer fails.

(unlike the previous section). The barometer readings were simulated with drift. The details on the choice of estimation gains and the simulated barometer drift will be clarified in Sec. 4.4.3. The interest here is to compare the impact of the altitude observer in the scan registrations.

First, Fig. 4.8 illustrates the absolute altitude estimation errors of the laser odometry before and after the introduction of the altitude observer, for the planar model approach (Fig. 4.8a) and the point cloud approach (Fig. 4.8b). In the planar model approach the spurious peaks that were observed in the previous section are largely suppressed, and the maximum error is now lowered to 12 cm ($t = 75$ s in Fig. 4.8a). In the point cloud approach, the large altitude errors that caused the approach to fail are now avoided (at $t = 30$ s in Fig. 4.8b). With the introduction of the altitude observer estimates, both approaches achieve similar accuracy.

The complete laser odometry estimates with the aid of the altitude observer are shown in Fig 4.9. It can be observed in Fig. 4.9a that both approaches now manage to effectively follow the simulation ground truth throughout the duration of the flight. When comparing the absolute estimation errors (Fig. 4.9b-Fig. 4.9e), similar levels of accuracy is obtained for all the estimated states. With regards to the computation time, however, no significant changes occur with respect to the previous results from Fig. 4.7. The planar model approach remains significantly faster than the point cloud approach.

It can be concluded that notable improvements were obtained in both cases, with regards to the altitude estimates. It is important to highlight, that this was achieved despite the presence of drift in the barometer readings. As will be seen in Sec. 4.4.3.2, the laser altitude estimates allow tracking this barometer drift. Then, the combination of different sources of altitude information in the observer provides a better initialization for the scan registrations, which in return allows tackling some of the problems observed in the previous section.

4.4.3 Height estimation results

We now present more detailed results for the two-stage height estimation approach described in Sec. 4.3.2. As previously mentioned, the barometer measurements are sensitive to changes in atmospheric conditions (strong winds, temperature changes). In practice, this generally translates into a slowly varying drift. In order to study the observer's behavior under large barometer drift, this was simulated as a sinusoid with a maximum speed of 1 m/min. The following results were obtained for the simulated flight from Fig. 4.2.

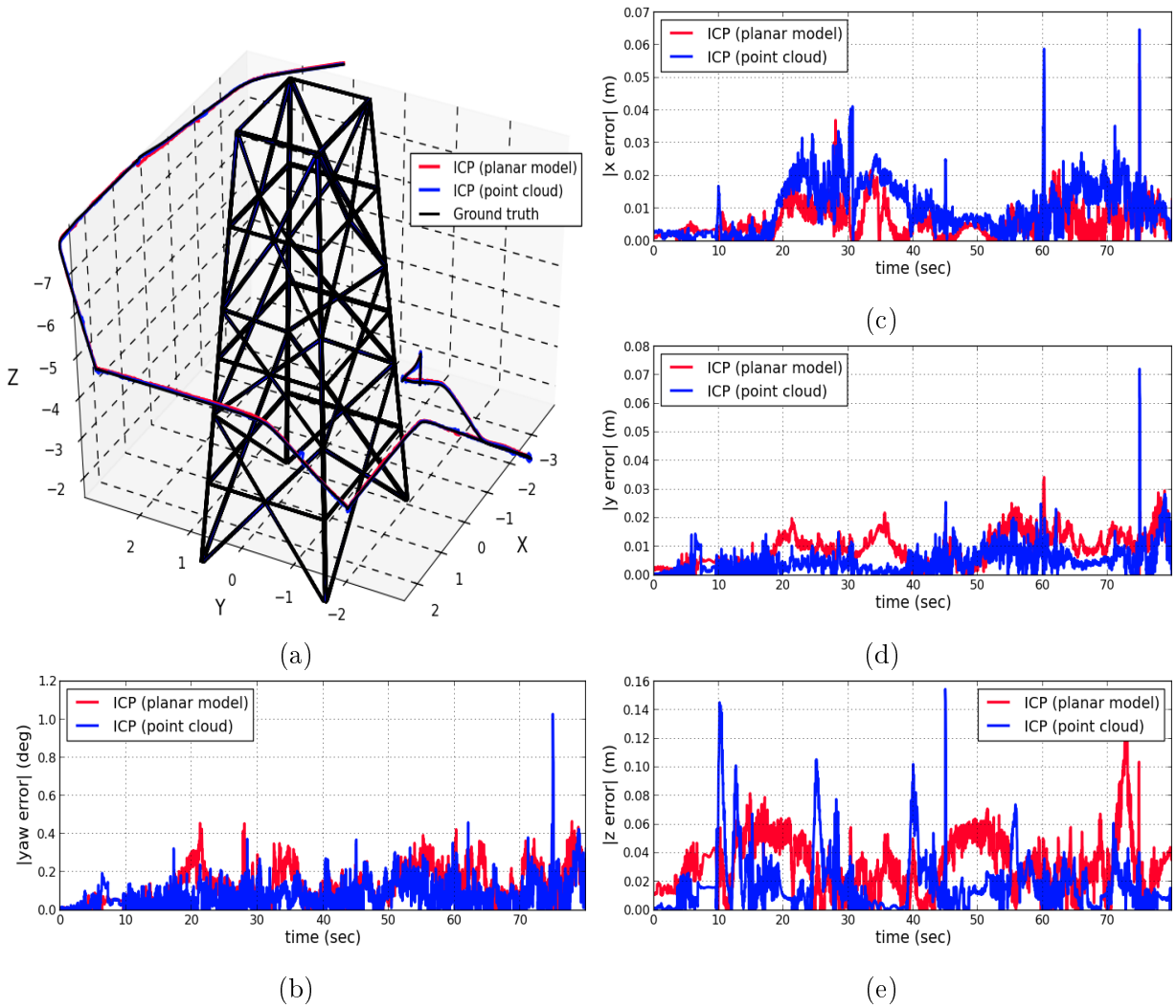


Figure 4.9: Laser odometry results after introducing the altitude observer estimates in the scan registration initializations. For both ICP implementations: **(a)** Comparing the laser odometry outputs with the ground truth. In contrast to Fig. 4.6, the point cloud approach now finishes the flight successfully. **(b)-(e)** Absolute estimation errors with respect to the simulation ground truth. Both approaches now achieve similar accuracy.

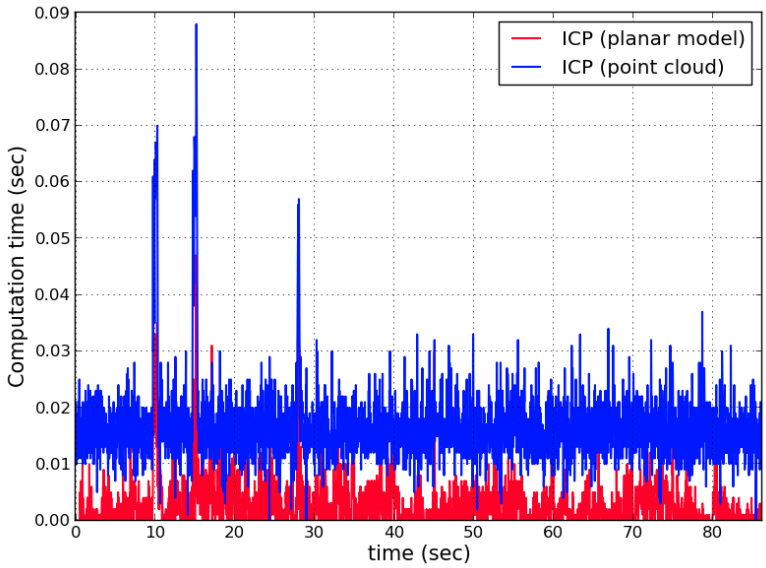


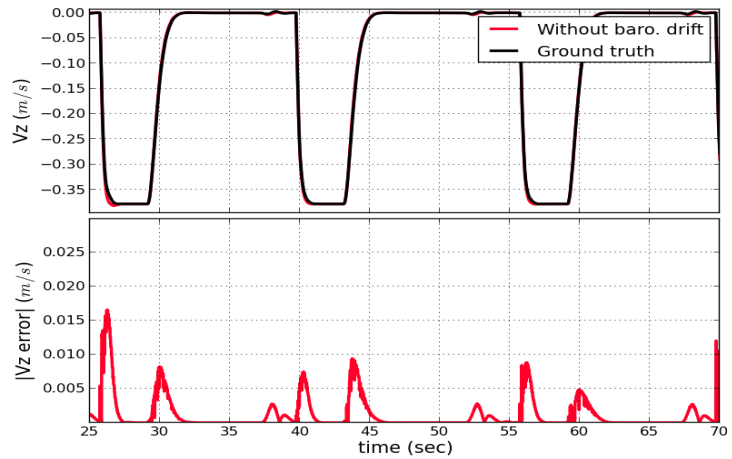
Figure 4.10: Comparing the computation time for the scan registrations when using the altitude observer estimates in the initializations. There are no significant changes with respect to Fig. 4.7. The planar model approach (in red) remains significantly faster than the point cloud approach (in blue).

4.4.3.1 Vertical velocity estimation

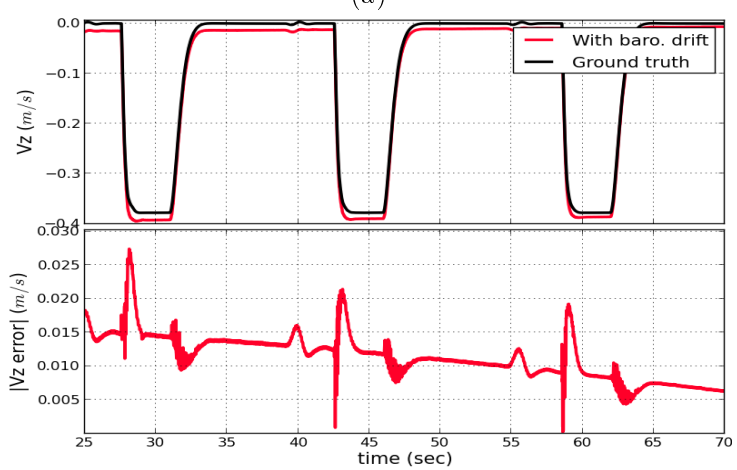
Regarding the first stage, the observer from Eq. (4.5) was used to recover vertical velocity estimates from the barometer and accelerometer readings. As this observer is a simple second-order system, the estimation gains were chosen as $(k_z, k_{v_z}) = (6.4, 16)$, which provide an adequate underdamped response. Fig. 4.11a shows the estimation results without barometer drift. As can be seen, the vertical velocity estimates are sufficiently accurate without the need of the laser estimates, as they remain below 1.5 cm/s. Then, the velocity estimates in the presence of barometer drift are shown in Fig. 4.11b. With respect to the previous case, the estimation error slightly increases, but remains within acceptable levels.

4.4.3.2 Fusing barometer and laser odometry measurements

Regarding the second stage of the height estimation process, the proposed altitude observer from Eq. (4.8) was tested for each implementation of the ICP algorithm. This observer uses the previously obtained velocity estimates, and fuses the ICP altitude estimates with barometer readings. As previously mentioned, the weights from Eq. (4.9) were chosen as $\lambda_1 = 1$, to rely mainly on the barometer measurements to estimate the

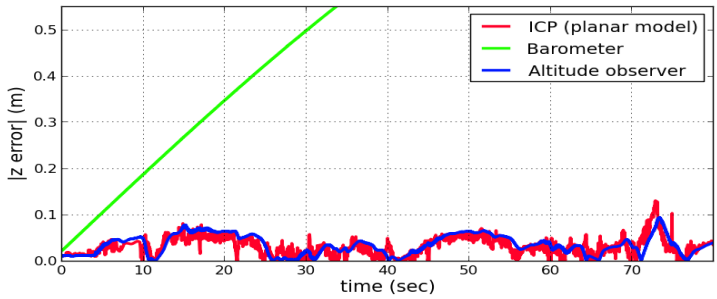


(a)

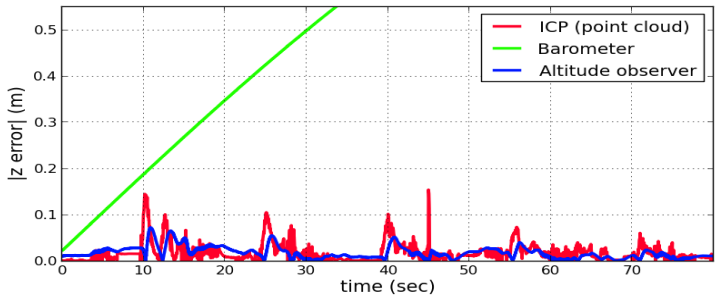


(b)

Figure 4.11: Vertical velocity estimates (Vz) obtained by fusing barometer and IMU measurements. **(a)** Without barometer drift. **(b)** With barometer drift.



(a)



(b)

Figure 4.12: The absolute altitude errors for ICP without the aid of the altitude observer, the barometer measurements with drift, and the altitude observer for: **(a)** ICP with planar model. **(b)** ICP with point cloud reconstruction.

altitude, and $\lambda_2 = 0$, to rely on the laser estimates to estimate the barometer drift. Then, the estimation gains were set to $(k_z, k_{b_z}) = (6.6, -1.36)$, which achieved a good performance in the simulations. An explanation on how to determine these gains is given in the Appendix.

Similar to the results previously shown in Sec. 4.4.2.2, the altitude observer's output was used at each scan registration initialization. Fig. 4.12 compares the absolute errors, with respect to the simulation ground truth, of the observer inputs (i.e., the barometer measurements and the altitude estimates from the ICP-based laser odometry) and the corresponding filtered output. While the barometer readings accumulate a large error over time, the presence of this drift doesn't significantly degrade the quality observer's altitude estimates, which instead provides notable improvements with respect to both inputs. It can be noted that the altitude observer provides smoother altitude estimates than the laser odometry (for both ICP approaches), while rejecting the large errors from the barometer drift. The effectiveness of this formulation is further verified in Fig. 4.13, as the observer manages to estimate the previously unknown barometer drift, with less than 10 cm of error.

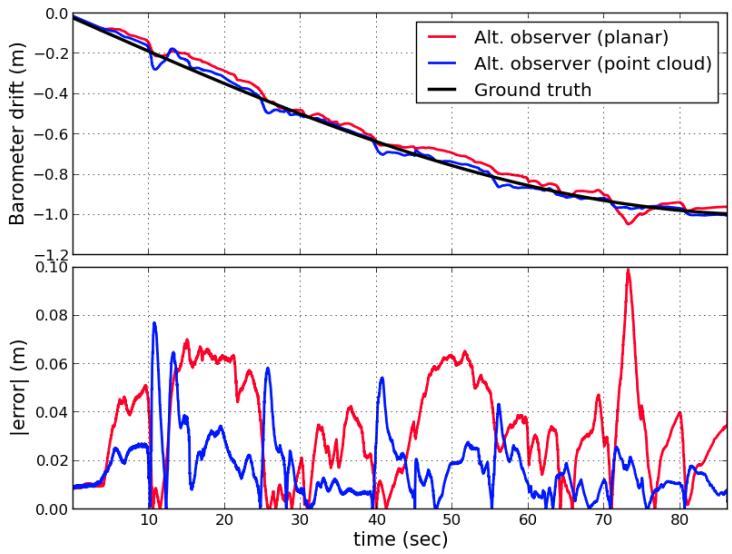


Figure 4.13: For the altitude observer and both ICP implementations: (Top) Comparing the barometer drift estimates with the ground truth. (Bottom) Absolute estimation errors. The observer succeeds in both cases.

4.5 Conclusions

In this chapter, we have presented a 6 DoF pose tracking approach for a MAV navigating in an electric tower inspection scene, with an on-board sensor setup consisting in a 2D laser scanner, an IMU (accelerometer and gyrometer) and a barometer sensor. The estimation process was divided into three modules: laser odometry, attitude estimation and height estimation. Attitude estimation relied on a non-linear observer formulation, used in previous chapters, to estimate the roll and pitch angles from the accelerometer and gyrometer measurements. A gain scheduling approach was proposed to adapt the observer to changing flight dynamics. Then, the four remaining states were determined from the laser scans with two proposed implementations of the ICP algorithm. The first approach consisted in aligning the 2D laser scans to a 3D point cloud reconstruction of the tower, and the second approach relied instead on a simplified planar representation of the tower's contour and a projection-based matching strategy that allowed obtaining significant speed gains. In both cases, the registration process was carried out in 3D, aided by the attitude estimates from IMU measurements, allowing to recover altitude information in the process. In order to exploit this, and to obtain more robust altitude estimates, a two-stage height estimation approach was proposed to fuse the laser odometry estimates with barometer and accelerometer readings. This simple formulation allowed estimating the unknown barometer drift in the process. Results based on simulations were presented to validate each of the proposed modules. It was shown that by combining all three mod-

ules (laser odometry, attitude observer and height estimation) satisfying results could be obtained for the complete 6 DoF pose of the MAV, in terms of accuracy and computation time.

CHAPTER 5

MAV sensor fusion and feedback control

Contents

5.1 Overview **84**

5.2 MAV rigid body dynamics **85**

5.3 Sensor fusion **86**

5.3.1 Rotation subsystem 86

5.3.2 Translation subsystem 87

5.4 Control **88**

5.4.1 Attitude control 88

5.4.2 Position control 90

5.5 Results **90**

5.5.1 Indoor scene experimental results 91

5.5.2 Electric tower scene simulation results 94

5.6 Conclusions **101**

5.1 Overview

In previous chapters we have addressed the problem of pose tracking, obtaining the complete 6 degree-of-freedom (DoF) pose of a MAV platform following a similar modular approach: attitude estimation, height estimation and laser odometry. The latter, relying on a 2D laser scanner, has been the main focus of this thesis. The methodology adopted in previous chapters has been largely aimed at tackling the difficulties encountered from

the 2D nature of the range scans, notably, the impact of the MAV's 3D motion and of the surrounding geometry in the quality of the state estimates. Two different scenarios have been studied, corresponding to a typical indoor scene (Chapter 2) and an electric tower inspection scene (Chapter 3 and Chapter 4). In both cases, the proposed frameworks provided the desired real-time pose estimates from on-board sensing. Building on the previously obtained results, in this chapter we present a brief overview of the sensor fusion and feedback control algorithms, and present initial results on waypoint following.

This chapter is structured as follows. A brief review of the rigid body dynamics of the quadrotor is given in Sec. 5.2. Next, in Sec. 5.3 we introduce the sensor fusion algorithms which provide the necessary information for the closed-loop control. The complete feedback control design is then discussed in Sec. 5.4, consisting in attitude (Sec. 5.4.1) and position control (Sec. 5.4.2). The complete on-board control and estimation architecture is described in Sec. 5.5.1, together with experimental results on waypoint following for the indoor scene. This chapter ends with simulation results for the electric tower scene in Sec. 5.5.2.

5.2 MAV rigid body dynamics

Before discussing our sensor fusion and control algorithms, we first describe the dynamic model used for the control design. Recalling, $\mathcal{I}$ denotes an inertial NED (North-East-Down) frame and $\mathcal{B}$ the body frame, attached to the MAV's center of mass, the orientation of which coincides with that of $\mathcal{I}$ when the MAV is in hover. The dynamic equations can then be divided in two subsystems [25, 117]: translational and rotational.

On the one hand, let $\boldsymbol{\xi} = (x, y, z)^\intercal$ denote the position vector of $\mathcal{B}$ with respect to $\mathcal{I}$, expressed in $\mathcal{I}$. Then, the basic translational dynamics of multirotor aircraft with respect to an inertial frame $\mathcal{I}$ are given by the following equations [25]

$$\dot{\boldsymbol{\xi}} = \boldsymbol{v} \tag{5.1a}$$

$$\dot{\boldsymbol{v}} = g\mathbf{e_3} + \tfrac{\boldsymbol{F}}{m}, \tag{5.1b}$$

where $\boldsymbol{v} = (v_x, v_y, v_z)^\intercal$ denotes the linear velocity of $\mathcal{B}$ with respect to $\mathcal{I}$, expressed in $\mathcal{I}$, g is the gravity constant, $\mathbf{e_3} = (0, 0, 1)^\intercal$, m is the MAV's mass, and $\boldsymbol{F} = (F_x, F_y, F_z)^\intercal$ is the coordinate vector of the aerodynamic forces acting on the MAV, expressed in $\mathcal{I}$.

On the other hand, let R denote the rotation matrix from $\mathcal{B}$ to $\mathcal{I}$. Then, the basic

MAV rotational dynamics with respect to $\mathcal{I}$ are [25]

$$\dot{R} = RS(\boldsymbol{\omega}) \tag{5.2a}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\Gamma} + \boldsymbol{\Gamma}_{\boldsymbol{p}}, \tag{5.2b}$$

where $\boldsymbol{\omega}$ is the angular velocity vector from $\mathcal{B}$ to $\mathcal{I}$, expressed in $\mathcal{B}$. $S(.)$ the skew-symmetric matrix associated with the cross product (i.e., $S(\mathbf{x})\mathbf{y} = \mathbf{x} \times \mathbf{y}$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$), $\mathbf{J}$ the inertia matrix, $\boldsymbol{\Gamma}$ the control torque, and $\boldsymbol{\Gamma}_{\boldsymbol{p}}$ a possible perturbation torque (aerodynamic perturbations, mass offsets, etc).

A link between the two subsystems is established through the aerodynamic forces $\boldsymbol{F}$ from Eq. (5.1b), which can be expressed as

$$\boldsymbol{F} = -TR\mathbf{e_3} + \boldsymbol{F}_{\boldsymbol{p}}, \tag{5.3}$$

where T is the thrust force generated by the propellers and $\boldsymbol{F}_{\boldsymbol{p}}$ are perturbation forces comprised mainly of secondary aerodynamic effects, such as rotor drag and blade flapping effects [25]. It follows that translations depend on the MAV's roll, pitch and yaw angles (but not on their time derivatives) through the rotation matrix. In contrast, the rotation system is independent of the MAV's translations. MAV position control strategies exploit this fact by rotating the platform in order to produce from the thrust force the linear accelerations necessary to stabilize the MAV's position [118].

5.3 Sensor fusion

In the previous chapters, the complete 6 DoF pose of the MAV $\{x, y, z, \phi, \theta, \psi\}$ was determined from the sensor measurements. As discussed in Sec. 1.3, the sensor fusion problem acts as an intermediary step between pose tracking and control. Here, the goal is to combine the tracked 6 DoF pose with inertial measurements using the MAV's dynamics (Eq. (5.1) and Eq. (5.2)). This allows recovering filtered state estimates $\{\hat{x}, \hat{y}, \hat{z}, \hat{\phi}, \hat{\theta}, \hat{\psi}\}$ and linear velocity estimates $\hat{\boldsymbol{v}} = (\hat{v}_x, \hat{v}_y, \hat{v}_z)^\intercal$, required for a stable closed-loop position control. These outputs are directly fed into the control module, which will be described in Sec. 5.4. We now present how the MAV rotation and translation dynamics are employed for sensor fusion purposes.

5.3.1 Rotation subsystem

We now recall the non-linear attitude observer formulation considered in the previous chapters (presented in Eq. (2.18)). This observer is derived from the MAV's kinematics (Eq. (5.2a)) and fuses accelerometer measurements $\mathbf{a_m}$ and the angular velocities measured by the gyrometer $\boldsymbol{\omega}_m = (\omega_1, \omega_2, \omega_3)^\intercal$, both expressed in $\mathcal{B}$, as

$$\dot{\hat{\boldsymbol{\gamma}}} = \hat{\boldsymbol{\gamma}} \times (\boldsymbol{\omega}_m - k_{obs,\gamma}(\mathbf{a}_m \times \hat{\boldsymbol{\gamma}})), \quad k_{obs,\gamma} > 0 \tag{5.4}$$

where $k_{obs,\gamma}$ is the positive scalar estimation gain, and $\hat{\boldsymbol{\gamma}} = (\hat{\gamma}_1, \hat{\gamma}_2, \hat{\gamma}_3)^\intercal$ is an estimate of $\boldsymbol{\gamma}$, the vertical vector of $\mathcal{I}$ expressed in $\mathcal{B}$. This vector contains implicitly the MAV's roll and pitch angles (see Eq. (2.20)) as

$$\begin{aligned} \hat{\phi} &= \arcsin(\hat{\gamma}_2) \\ \hat{\theta} &= \operatorname{atan2}(-\hat{\gamma}_1, \hat{\gamma}_3). \end{aligned} \tag{5.5}$$

Then, in Chapter 3 and Chapter 4, the yaw angle was estimated separately from the laser odometry. A simple improvement, which further exploits the high IMU measurement rates, is to fuse ψ_{laser}, the laser odometry estimates, with ω_3, the angular velocity about the vertical axis of $\mathcal{B}$ measured by the gyrometer, as

$$\dot{\hat{\psi}} = \omega_3 - k_{obs,\psi}(\hat{\psi} - \psi_{\text{laser}}), \quad k_{obs,\psi} > 0. \tag{5.6}$$

where $k_{obs,\psi}$ is a positive scalar estimation gain and $\hat{\psi}$ is the filtered yaw estimate.

Finally, the complete estimated rotation matrix $\hat{R}$ is recovered by combining the estimated angles as

$$\hat{R} = R_z(\hat{\psi})R_x(\hat{\phi})R_y(\hat{\theta}), \tag{5.7}$$

using the Z-X-Y Euler angle convention. This complete estimation of the MAV's attitude will be considered in the following discussions regarding sensor fusion and the translation dynamics, and in the control design (Sec 5.4).

5.3.2 Translation subsystem

Relying on the translational dynamics from Eq. (5.1), it is possible to formulate linear velocity observers that fuse inertial measurements and estimates of the MAV's position $\boldsymbol{\xi} = (x, y, z)^\intercal$. The goal is to recover estimates of the linear velocities $\hat{\boldsymbol{v}} = (\hat{v}_x, \hat{v}_y, \hat{v}_z)^\intercal$

and filtered position estimates $\hat{\boldsymbol{\xi}} = (\hat{x}, \hat{y}, \hat{z})^\intercal$. In the following discussions, the external aerodynamic forces $\boldsymbol{F}$ from Eq. (5.1) are determined from the accelerometer readings $\boldsymbol{a}_m$ and the estimated attitude $\hat{R}$ from Eq. (5.7) as

$$\boldsymbol{F} = m\hat{R}\boldsymbol{a}_m, \tag{5.8}$$

where $\boldsymbol{F} = (F_x, F_y, F_z)^\intercal$. Then, developing Eq. (5.1), one obtains

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = \frac{F_x}{m} \\ \dot{v}_y = \frac{F_y}{m} \\ \dot{v}_z = g + \frac{F_z}{m} \end{cases} \tag{5.9}$$

From this equation it can be noted that the horizontal and vertical translation components are decoupled. We now discuss them separately.

Horizontal component

In previous chapters, the horizontal translation was estimated from the laser odometry (see Sec. 2.4.1 for the indoor scene and Sec. 4.3.1 for the electric tower scene). The goal is now to recover horizontal velocity estimates by fusing laser odometry estimates, denoted $(x_{\text{laser}}, y_{\text{laser}})$, with accelerometer measurements. From Eq. (5.9), the dynamics for $\{x, y\}$ in $\mathcal{I}$ are obtained as two independent second-order systems, which leads to simple feedback state observers defined as

$$\begin{cases} \dot{\hat{x}} = \hat{v}_x - k_x(\hat{x} - x_{\text{laser}}) \\ \dot{\hat{v}}_x = \dfrac{F_x}{m} - k_{v_x}(\hat{x} - x_{\text{laser}}), \quad k_x, k_{v_x} > 0 \end{cases}$$
$$\begin{cases} \dot{\hat{y}} = \hat{v}_y - k_y(\hat{y} - y_{\text{laser}}) \\ \dot{\hat{v}}_y = \dfrac{F_y}{m} - k_{v_y}(\hat{y} - y_{\text{laser}}), \quad k_y, k_{v_y} > 0 \end{cases} \tag{5.10}$$

where (k_x, k_{v_x}) and (k_y, k_{v_y}) are the scalar observer gains, which guarantee exponential convergence if they are positive, then $\{\hat{v}_x, \hat{v}_y\}$ are estimates of the horizontal components of the translational velocity and $\{\hat{x}, \hat{y}\}$ are the filtered horizontal position estimates.

Vertical component

The goal is to recover vertical velocity estimates, which was already addressed in previous chapters. For the indoor scene (Chapter 2), a second-order low-pass filter based on a constant velocity model was applied directly to laser altimeter measurements in Eq. (2.23). For the electric tower scene (Chapter 4), a two-stage observer scheme was proposed in Sec. 4.3.2 relying on the vertical dynamics contained in Eq. (5.9) to fuse barometer measurements with accelerometer readings (Eq. (4.5)) and altitude estimates from the ICP-based laser odometry (Eq. (4.8)). In both cases, the outputs are $\hat{z}$, filtered altitude estimates, and $\hat{v}_z$, an estimate of the MAV's vertical velocity.

5.4 Control

Given the the sensor fusion module described in the previous section, we now have all the necessary information to introduce our control design. Acting on the MAV's airframe are the thrust force T (Eq. (5.3)) and the torque vector $\boldsymbol{\Gamma} = (\Gamma_x, \Gamma_y, \Gamma_z)^\intercal$ (Eq. (5.2b)). On a first instance, let us consider these variables as the four control inputs of the system. We seek to determine a *desired* torque vector $\boldsymbol{\Gamma_d}$ and thrust force T_d that drive a MAV towards a desired location $\boldsymbol{\xi_d}$ and heading ψ_d. Since the rotational dynamics from Eq. 5.2 are completely independent of the translation dynamics from Eq. 5.1, a common control strategy is to first stabilize the MAV's attitude in an inner loop [118]. This attitude control loop computes $\boldsymbol{\Gamma_d}$, as will be discussed in Sec. 5.4.1. Then, the position is stabilized in an outer loop at slower rates [118]. This translation control loop determines T_d, as will be presented in Sec. 5.4.2.

In reality, the thrust force and torque vector described previously are a product of the combined effort of the four embarked propellers (motors), and their corresponding speeds constitute the actual control inputs of the system. Desired torques and thrusts are translated to the required motor speeds via the allocation matrix [25], which consists in intrinsic parameters of the MAV platform such as the rotor disk area, rotor radius, thrust coefficient, among others. The reference motor speeds are then fed to a motor speed control loop, the lowest level of the control hierarchy. This part, however, is not addressed in this section.

We now give a brief overview of the control design.

5.4.1 Attitude control

The attitude control proceeds to compute the desired torque vector $\boldsymbol{\Gamma_d}$ in two steps:

1. Computation of the reference angular velocities $\boldsymbol{\omega_d} = (\omega_{d,1}, \omega_{d,2}, \omega_{d,3})^\intercal$, such that $\omega_{d,1}, \omega_{d,2}$ are obtained from $\hat{\boldsymbol{\gamma}}$ and $\boldsymbol{\gamma_d}$, and $\omega_{d,3}$ from $\hat{\psi}$ and ψ_d.
2. Computation of $\boldsymbol{\Gamma_d}$ from $\boldsymbol{\omega}$ and $\boldsymbol{\omega_d}$.

Here, the reference value $\boldsymbol{\gamma_d}$ is provided by the position controller, as will be presented in the following section. Then, since the yaw angle is independent from the remaining MAV states, ψ_d is provided separately and often serves secondary goals, such as orienting the MAV to maintain an object within the laser scanner's field of view (e.g., the electric tower in Chapter 4).

The aim of Step 1 is to derive angular velocity controls that ensure convergence of $\boldsymbol{\gamma}$ to $\boldsymbol{\gamma_d}$ and ψ to ψ_d. First, the desired angular velocities $\omega_{d,1}, \omega_{d,2}$ are simply defined as

$$\begin{cases} \omega_{d,1} &= -k_\gamma(\hat{\gamma}_2 - \gamma_{d,2}) \\ \omega_{d,2} &= k_\gamma(\hat{\gamma}_1 - \gamma_{d,1}), \quad k_\gamma > 0 \end{cases} \tag{5.11}$$

Using the candidate Lyapunov function $L = \|\boldsymbol{\gamma} - \boldsymbol{\gamma_d}\|^2$, one verifies from Eq. (5.2a) that this expression ensures local asymptotic stability of $\boldsymbol{\gamma} = \boldsymbol{\gamma_d}$ if $\boldsymbol{\omega} = \boldsymbol{\omega_d}$, $\hat{\boldsymbol{\gamma}} = \boldsymbol{\gamma}$, $\boldsymbol{\gamma_d}$ is constant, and $\omega_3 = 0$. A deeper stability analysis of this simple controller, showing semi-global stability properties, is given in [119].

Next, the independent yaw variable is controlled by defining the yaw angular velocity $\omega_{d,3}$, as a simple PI controller:

$$\omega_{d,3} = -k_\psi(\hat{\psi} - \psi_d) + k_{i,\psi} \int (\hat{\psi} - \psi_d), \tag{5.12}$$

with $\hat{\psi}$ recovered from Eq. 5.6.

Finally, the objective of Step 2 is to derive a torque control that ensures convergence of $\boldsymbol{\omega}$ to $\boldsymbol{\omega_d}$. A classical way to define such a desired control torque $\boldsymbol{\Gamma_d}$ is to use a simple high-gain linear controller: $\boldsymbol{\Gamma_d} = -k_\omega \mathbf{J}(\boldsymbol{\omega_m} - \boldsymbol{\omega_d})$. In order to account for the perturbation torque $\boldsymbol{\Gamma_p}$ in Eq. (5.2b), another classical solution is to add an integral correction term (i.e., a term proportional to the integral of $\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma_d}$). We define the desired torque control as

$$\boldsymbol{\Gamma_d} = -k_\omega \mathbf{J}(\boldsymbol{\omega_m} - \boldsymbol{\omega_d}) - \hat{\boldsymbol{\Gamma}}_{\boldsymbol{p}}, \quad k_\omega > 0 \tag{5.13}$$

where $\hat{\boldsymbol{\Gamma}}_{\boldsymbol{p}}$ is an estimate of $\boldsymbol{\Gamma}_{\boldsymbol{p}}$, considering a constant perturbation model for Eq. (5.2b) as

$$\left\{ \begin{array}{rcl} \mathbf{J}\dot{\boldsymbol{\omega}} & = & -\boldsymbol{\omega}_{\boldsymbol{m}} \times \mathbf{J}\boldsymbol{\omega}_{\boldsymbol{m}} + \boldsymbol{\Gamma} + \boldsymbol{\Gamma}_{\boldsymbol{p}} \\ \dot{\boldsymbol{\Gamma}}_{\boldsymbol{p}} & = & 0, \end{array} \right. \tag{5.14}$$

which leads to the following observer for $\boldsymbol{\Gamma}_{\boldsymbol{p}}$

$$\left\{ \begin{array}{rcl} \mathbf{J}\dot{\hat{\boldsymbol{\omega}}} & = & -\boldsymbol{\omega}_{\boldsymbol{m}} \times \mathbf{J}\boldsymbol{\omega}_{\boldsymbol{m}} + \boldsymbol{\Gamma} + \hat{\boldsymbol{\Gamma}}_{\boldsymbol{p}} - k_{obs,\omega}(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_{\boldsymbol{m}}) \\ \dot{\hat{\boldsymbol{\Gamma}}}_{\boldsymbol{p}} & = & -k_{obs,\Gamma}(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_{\boldsymbol{m}}), \quad k_{obs,\omega}, k_{obs,\Gamma} > 0 \end{array} \right. \tag{5.15}$$

By considering the dynamics of estimation errors $\tilde{\boldsymbol{\omega}} = \hat{\boldsymbol{\omega}} - \boldsymbol{\omega}, \tilde{\boldsymbol{\Gamma}}_{\boldsymbol{p}} = \boldsymbol{\Gamma}_{\boldsymbol{p}} - \hat{\boldsymbol{\Gamma}}_{\boldsymbol{p}}$, one verifies from (5.15) and Eq. (5.2b) that $\tilde{\boldsymbol{\omega}}$ and $\tilde{\boldsymbol{\Gamma}}_{\boldsymbol{p}}$ converge to zero asymptotically if $\boldsymbol{\omega}_{\boldsymbol{m}} = \boldsymbol{\omega}$ and $\boldsymbol{\Gamma}_{\boldsymbol{p}}$ is constant. Using large estimation gains $k_{obs,\omega}, k_{obs,\Gamma}$ then ensures small ultimate estimation errors if $\dot{\boldsymbol{\Gamma}}_{\boldsymbol{p}}$ is not too large. The main advantage of this solution for the compensation of $\boldsymbol{\Gamma}_{\boldsymbol{p}}$, compared to a classical integral-like solution, is to take full advantage of gyrometer measurements $\boldsymbol{\omega}_{\boldsymbol{m}}$ that come at high frequency and provide accurate estimates of the angular velocity. This allows for a fast estimation of $\boldsymbol{\Gamma}_{\boldsymbol{p}}$. By contrast, integral correction terms do not take benefit of these measurements and they are known to slow down the time-response.

5.4.2 Position control

The position control of quadrotors with position measurements in the inertial frame has been addressed in many publications (see, e.g., [120] for a survey). We briefly show how it is achieved with the attitude control described previously. Let $\boldsymbol{\xi}_{\boldsymbol{d}}$ denote a desired position trajectory, in the inertial frame $\mathcal{I}$, and $\boldsymbol{v}_{\boldsymbol{d}} := \dot{\boldsymbol{\xi}}_{\boldsymbol{d}}$. The hierarchical approach (see [118, 120, 121] for more details) consists in rewritting Eq. (5.1) as:

$$\left\{ \begin{array}{rcl} \dot{\boldsymbol{\xi}} & = & \boldsymbol{v} \\ \dot{\boldsymbol{v}} & = & -\beta(\boldsymbol{\xi}, \boldsymbol{v}, t) + g\mathbf{e_3} - \frac{T}{m}R\mathbf{e_3} + \boldsymbol{F}_{\boldsymbol{p}} + \beta(\boldsymbol{\xi}, \boldsymbol{v}, t) \end{array} \right. \tag{5.16}$$

where $\beta(\boldsymbol{\xi}, \boldsymbol{v}, t)$ is a feedback controller designed so as to make the desired trajectory $(\boldsymbol{\xi}_{\boldsymbol{d}}, \boldsymbol{v}_{\boldsymbol{d}})$ asymptotically stable for the system

$$\left\{ \begin{array}{rcl} \dot{\boldsymbol{\xi}} & = & \boldsymbol{v} \\ \dot{\boldsymbol{v}} & = & -\beta(\boldsymbol{\xi}, \boldsymbol{v}, t) + \boldsymbol{F}_{\boldsymbol{p}}. \end{array} \right. \tag{5.17}$$

A possible choice for $\beta(\boldsymbol{\xi}, \boldsymbol{v}, t)$ is

$$\beta(\boldsymbol{\xi}, \boldsymbol{v}, t) = -k_p(\boldsymbol{\xi} - \boldsymbol{\xi_d}) - k_v(\boldsymbol{v} - \boldsymbol{v_d}) + k_i \int (\boldsymbol{\xi} - \boldsymbol{\xi_d}) \tag{5.18}$$

where the integral correction term guarantees asymptotic stability if both $\boldsymbol{v_d}$ and $\boldsymbol{F_p}$ are constant. From (5.16), the objective is then to make $g\mathbf{e_3} - \frac{T}{m}R\mathbf{e_3} + \beta(\boldsymbol{\xi}, \boldsymbol{v}, t)$ converge to zero. One has

$$g\mathbf{e_3} - \tfrac{T}{m}R\mathbf{e_3} + \beta(\boldsymbol{\xi}, \boldsymbol{v}, t) = 0$$
$$\Longleftrightarrow$$
$$g\boldsymbol{\gamma} - \tfrac{T}{m}\mathbf{e_3} = -R^T\beta(\boldsymbol{\xi}, \boldsymbol{v}, t)$$

This leads to the following expression for $\boldsymbol{\gamma_d} = (\gamma_{d,1}, \gamma_{d,2}, \gamma_{d,3})^\intercal$ and T_d

$$\begin{cases} \gamma_{d,1} &= -\frac{1}{g}\mathbf{e_1}^T R^T \beta(\boldsymbol{\xi}, \boldsymbol{v}, t) \\ \gamma_{d,2} &= -\frac{1}{g}\mathbf{e_2}^T R^T \beta(\boldsymbol{\xi}, \boldsymbol{v}, t) \\ \gamma_{d,3} &= \sqrt{1 - \gamma_{d,1}^2 - \gamma_{d,2}^2} \\ T_d &= m\mathbf{e_3}^T R^T \beta(\boldsymbol{\xi}, \boldsymbol{v}, t) \end{cases} \tag{5.19}$$

A stability analysis of the closed-loop system, with $\boldsymbol{\gamma_d}, T_d$ defined as above and $\boldsymbol{\omega}$ given by the desired values (5.11) is provided in [119]. The computation of $\boldsymbol{\gamma_d}$, T_d, defined above require setting $R = \hat{R}$ (from Eq. (5.7)) and $\boldsymbol{\xi} = \hat{\boldsymbol{\xi}}$, $\boldsymbol{v} = \hat{\boldsymbol{v}}$ from the linear velocity observers previously described (Sec. 5.3.2). The reference $\boldsymbol{\gamma_d}$ is then passed on to the attitude control discussed in the previous section.

5.5 Results

We now present the results of tests performed with our proposed sensor fusion and control designs, under the two scenarios considered in previous chapters: indoors (Chapter 2) and the electric tower inspection scene (Chapter 3 and Chapter 4).

5.5.1 Indoor scene experimental results

We first recall our experimental setup for an indoor scenario, which was previously presented in Sec. 2.5 (see Fig. 2.2b). On-board, the control and state estimation architecture is divided into three levels as follows. At the lowest level, the brushless motor speeds are controlled with AutoQuad ESC 32 electronic speed controllers (ESCs). At the next

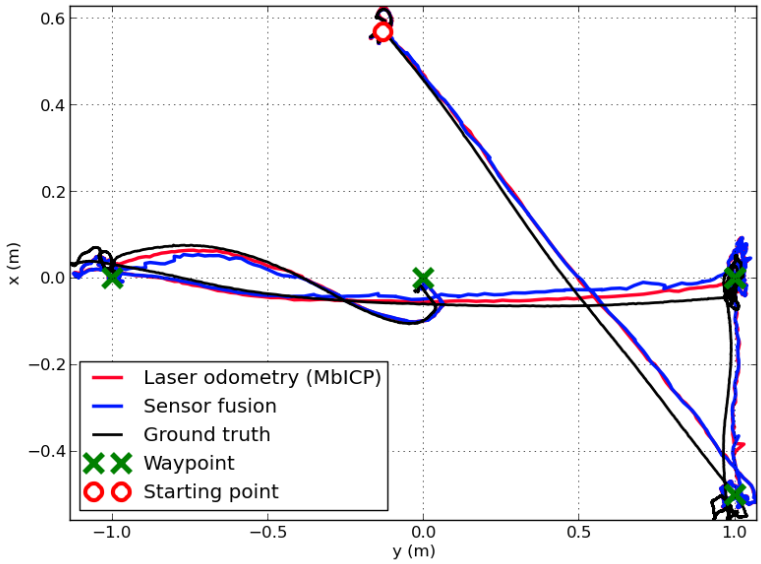


Figure 5.1: The 2D laser odometry and resulting sensor fusion outputs for an indoor flight from (see Fig. 2.2b). The MAV follows a set of pre-defined waypoints, at a fixed height of 1 m, using on-board processing and sensing. The ground truth shows the actual path followed by the MAV.

level, an SF10/A laser altimeter from Lightware Optielectronics and an MPU6000 3 axis accelerometer/gyrometer IMU are connected to a Quantec Quanton flight controller card with an STM32 microcontroller, which is in charge of estimation and control of the roll and pitch angles (see Sec. 5.3.1 and Sec. 5.4.1). Altitude estimation from the altimeter measurements are also performed in this "low-level" card (see Sec. 2.4.3). These estimates are then sent to an Odroid XU computer at the highest level of the hierarchy, which, among other tasks, is in charge of altitude control. A Hokuyo URG-30LX 2D laser scanner is connected to this on-board computer, where all laser processing is performed, together with sensor fusion and control algorithms for the remaining states (the 2D pose $\{x, y, \psi\}$). During the flights, the high-level card sends to the low-level Quanton card reference values for the thrust, roll/pitch angles, and yaw angular velocity. This low-level card then computes reference velocities for the brushless motors that are sent to the ESCs.

For the test flights, a set of waypoints were chosen to account for different cases such as large horizontal translations, large rotations, and simultaneous translation and rotation. The goal was for the quadrotor to follow these waypoints relying exclusively on the on-board state estimates and control algorithms. In the following discussions, we focus on the high-level component of the on-board architecture. Altitude results are not shown since the MAV was simply kept at a constant height of 1 m from the ground level, and altitude estimation and control were performed separetely from the laser altimeter measurements.

We thus present results based on the laser odometry, which was carried out with the MbICP algorithm (see Sec. 2.4.1), providing estimates of the MAV's 2D pose $\{x, y, \psi\}$. A motion capture (MoCap) system was used to keep track of the MAV's position. This provided a ground truth that was used for comparison purposes and to perform automatic take-off and landing. After take-off, the laser odometry was initialized with the known pose (from the MoCap) and by fixing a reference scan for the MbICP algorithm. All subsequent scan registrations were initialized from the previous scan registration result. The complete trajectory for one of these test flights is shown in Fig. 5.1, together with the laser odometry estimates and corresponding sensor fusion outputs. In the following, we discuss in detail the results illustrated in this figure.

5.5.1.1 Sensor fusion

The high-level sensor fusion for the indoor scene consists in the horizontal velocity observers from Eq. (5.10), which fuse 2D laser odometry and accelerometer measurements, and the yaw observer from Eq. (5.6), which fuses laser odometry yaw estimates and gyrometer measurements. The corresponding estimation gains were set to $(k_x, k_{v_x}) = (9.6, 36)$ and $(k_y, k_{v_y}) = (9.6, 36)$ for Eq. (5.10), and $k_{obs,\psi} = 6.0$ for Eq. (5.6).

The sensor fusion pose estimates are shown in Fig. 5.2, and compared against the laser odometry and the ground truth. It can be noted that the fused estimates closely replicate the MbICP output. The precision with respect to the ground truth remains similar to the laser odometry in the hovering state phases (once the MAV reaches the desired location), which remain within 4 cm of error for the position (Fig. 5.2a-5.2b) and $1°$ for the orientation (Fig. 5.2c). However, during transitions between waypoints ($t = 10$ s, $t = 30$ s, $t = 55$ s and $t = 65$ s in Fig. 5.2a-5.2c), the observers often present larger error peaks. This is mainly due to the relatively high observer gains, which lead to a reactive observer response to odometry noise and errors. These gains were required to recover adequate linear velocity estimates.

Finally, the linear velocity estimates are compared to the ground truth in Fig. 5.3. The larger error peaks observed in Fig. 5.3b (e.g., $t = 10$ s, $t = 55$ s and $t = 65$ s) correspond directly to transitions between waypoints (when compared to Fig. 5.2a and Fig. 5.2b), and are a product of laser odometry errors. Nonetheless, the maximum errors observed in Fig. 5.3b, of 20 cm/s for V_x ($t = 30$ s) and 30 cm/s for V_y ($t = 55$ s), remain within acceptable levels. For the duration of the flight, these velocity estimates converge quickly to the ground truth thanks to the choice of estimation gains and the high estimation

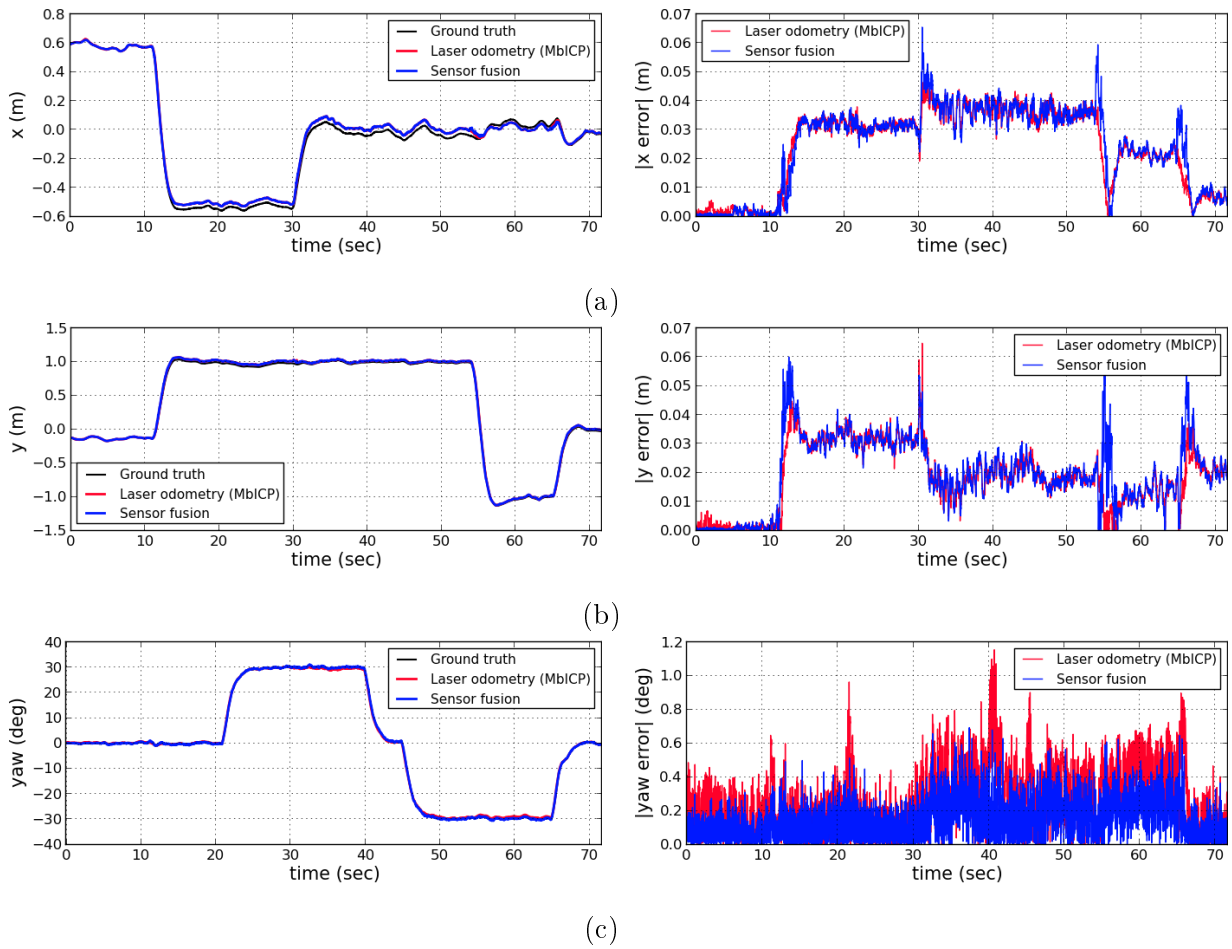
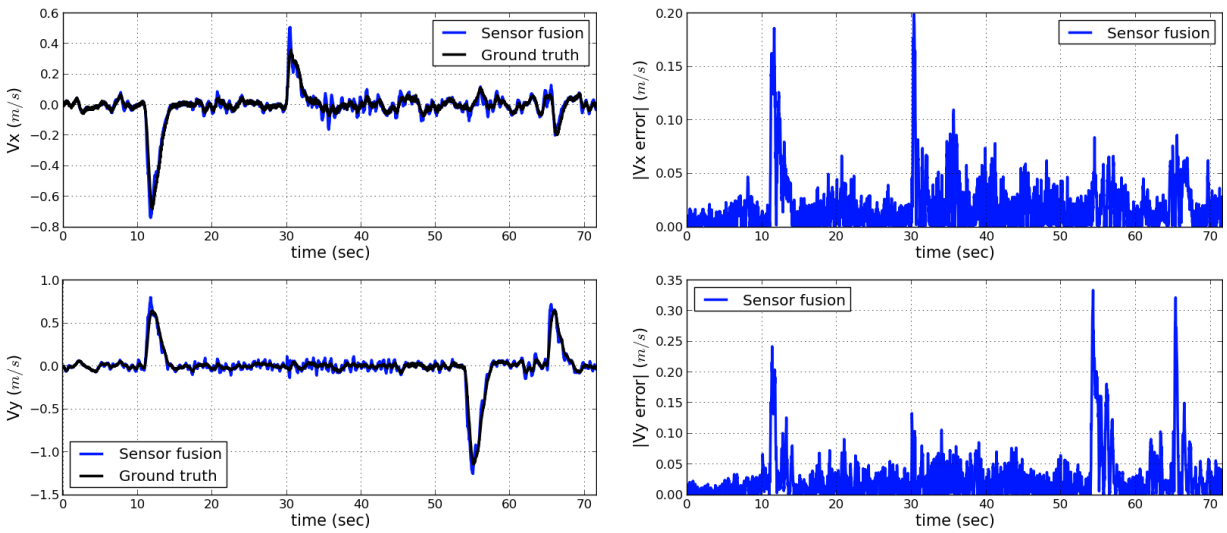


Figure 5.2: The results from fusing the 2D laser odometry with accelerometer readings (Eq. (5.10)) and gyrometer readings (Eq. (5.6)) for the flight from Fig. 5.1. **(left)** Comparing the laser odometry and sensor fusion outputs to the MoCap ground truth for the **(a)** x, **(b)** y and **(c)** ψ components. **(right)** The corresponding absolute estimation errors. The sensor fusion closely follows the laser odometry for all states.



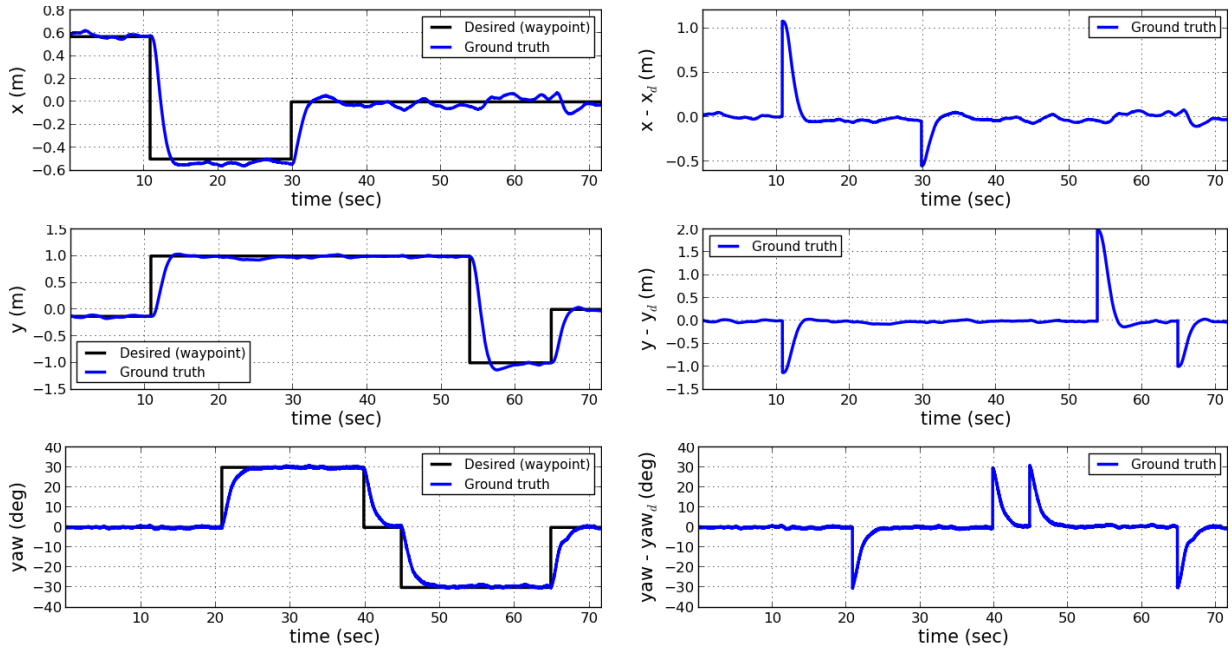
(a) (b)

Figure 5.3: The horizontal velocity estimation results from fusing laser odometry and accelerometer readings with the observer proposed in Eq. (5.10). **(a)** The estimated horizontal velocities follow the ground truth closely thanks to the choice of observer gains and high laser odometry rates (near 40 Hz). **(b)** The absolute estimation errors, where the larger peaks correspond directly to transitions between waypoints (when compared to Fig. 5.2a and Fig. 5.2b).

frequency of the laser odometry (near 40 Hz).

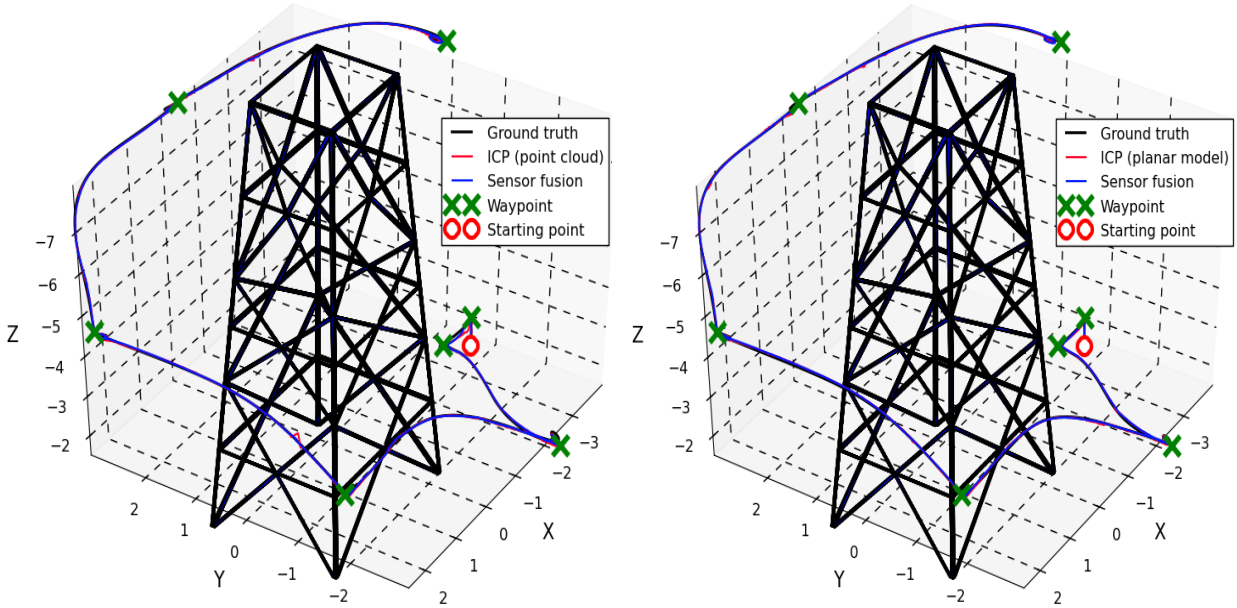
5.5.1.2 Feedback control

In the experimental flight from Fig. 5.1, the MAV's complete 6 DoF pose was stabilized using on-board sensing and processing capabilities. We now present the results for the high-level closed-loop control, focusing on the 2D pose $\{x, y, \psi\}$ estimated from the laser odometry. The pose and velocity estimates recovered in the preceding sensor fusion step, presented in Fig. 5.2 and Fig. 5.3, were fed back to the control loop at a fixed rate of 50Hz. The control gains for $\{x, y\}$ were set to $(k_p, k_v) = (1.44, 1.92)$ (Eq. (5.18)) and $k_\psi = 1.0$ for the yaw angle control (Eq. (5.12)). Both controllers included an integral term with a gain of $k_i = 0.5$ to handle steady state errors. In Fig. 5.4, we illustrate the response of the control system in the time domain, for the duration of the flight. Overall, the MAV's states are effectively driven towards the desired location, as observed from the measured ground truth in Fig. 5.4a. This is further verified from the tracking errors shown in Fig. 5.4b, which converge asymptotically towards zero. In steady state, the integral terms allow achieving near-zero errors, which remain within ± 5 cm for $\{x, y\}$, and $\pm 0.8°$, for the yaw angle.



(a) (b)

Figure 5.4: The resulting closed-loop control of $\{x, y, \psi\}$ with the MbICP laser odometry, for the duration of the flight from Fig. 5.1. **(a)** Time response of the control system for different desired locations. The ground truth pose from the MoCap system shows that all states converge to the desired location. **(b)** In all cases the tracking errors converge asymptotically towards zero.



(a) (b)

Figure 5.5: Testing the sensor fusion (Sec. 5.3) and feedback control (Sec. 5.4) designs in the electric tower scene. In separate simulations, the laser odometry relied on the ICP adaptations from Sec. 4.2, using **(a)** the tower point cloud reconstruction (see Fig. 4.5a) and **(b)** the planar tower representation (see Fig. 4.5b) for the scan registrations. In both simulations, the MAV follows a set of waypoints using state estimates from on-board sensing: 2D laser scanner, gyrometer, accelerometer and barometer.

5.5.2 Electric tower scene simulation results

A set of experiments similar to Sec. 4.4.2 were performed for the electric tower scene. This consisted in simulated flights in the Gazebo simulation environment [111], around a CAD model of an electric tower body with rectangular cross-sections (see Fig. 3.5). A similar control architecture to the indoor scene was implemented in these simulations. The main difference, is that the laser altimeter used indoors was replaced with a barometer sensor, and altitude estimation is now performed at the high-level stage. The barometer readings were simulated with a slowly-varying drift (a sinusoid with a maximum speed of 1 m/min), as in the previous tower scene simulations (Sec. 4.4.2.2). As for the indoor scene, we seek to validate the high-level component of the architecture, which corresponds to $\{x, y, z, \psi\}$.

In the simulations, several waypoints were given for the quadrotor to follow, accounting for a complete flight around the tower, at varying heights. The platform was stabilized using uniquely state estimates from on-board sensing: 2D laser scanner, accelerometer, gyrometer and barometer. The laser odometry in this case consisted in the two adaptations of the ICP algorithm proposed in Sec. 4.2, which rely on a point cloud or a planar

representation of the electric tower to register the laser scans. Here, we assume that these models were obtained beforehand (see Fig. 4.5a and Fig. 4.5b). In order to perform the closed-loop control for both ICP adaptations, the simulations had to be performed separately. In both cases, a rough guess of the MAV's location was provided for the first laser scan. All subsequent scan registrations were initialized with the most recent sensor fusion estimates. An example of the two simulated flights is shown in Fig. 5.5, where similar results were obtained in both experiments. In the following, we will discuss in detail the results illustrated in this figure.

5.5.2.1 Sensor fusion

The high-level sensor fusion design used for the indoor scene was also considered in these simulations. This included the horizontal velocity observers from Eq. (5.10), with gains $(k_x, k_{v_x}) = (9.6, 36)$ and $(k_y, k_{v_y}) = (9.6, 36)$, and the yaw observer from Eq. (5.6), with gain $k_{obs,\psi} = 6.0$. In addition, the two-stage altitude estimation approach proposed in Sec. 4.3.2 was included. The first stage (Eq. (4.5)) fuses barometer and accelerometer readings to recover vertical velocity estimates. The gains for this stage were set to $(k_z, k_{v_z}) = (9.6, 36)$. The second stage (Eq. (4.8)) then fuses barometer measurements and laser odometry estimates. The final output are filtered estimates of the altitude and barometer drift. Here, the gains were set to $(k_z, k_{b_z}) = (13.2, -2.727)$ and parameters $\lambda_1 = 1.0$ and $\lambda_2 = 0.0$. Recalling from Sec. 4.3.2.2, with this choice of parameters, altitude estimates rely mainly on barometer readings, while the barometer drift is tracked from the laser odometry.

The absolute sensor fusion estimation errors (with respect to the simulation ground truth) for the two simulated flights are shown in Fig. 5.6, and compared against the corresponding laser odometry. For the $\{x, y\}$ components (Fig. 5.6a-5.6b), it can be observed that the velocity observers filter several spurious laser odometry estimates, and the fused estimates remain below 6 cm of error for both ICP formulations. Then, for the altitude $\{z\}$ estimation (Fig. 5.6c), in both cases the altitude observer simultaneously filters the laser odometry estimates and compensates for the barometer drift, remaining below 10 cm of error. Lastly, for $\{\psi\}$ (Fig. 5.6d), the fused yaw estimates present several peaks (e.g., $t = 20$ s, $t = 40$ s) which correspond mainly to transient observer responses to large orientation differences between waypoints (up to 70° as will be seen in Fig. 5.8b and Fig. 5.9b). Nonetheless, the fused estimates remain below 1.2° of error.

It can be noted that, overall, the results observed in Fig. 5.6 are similar for both

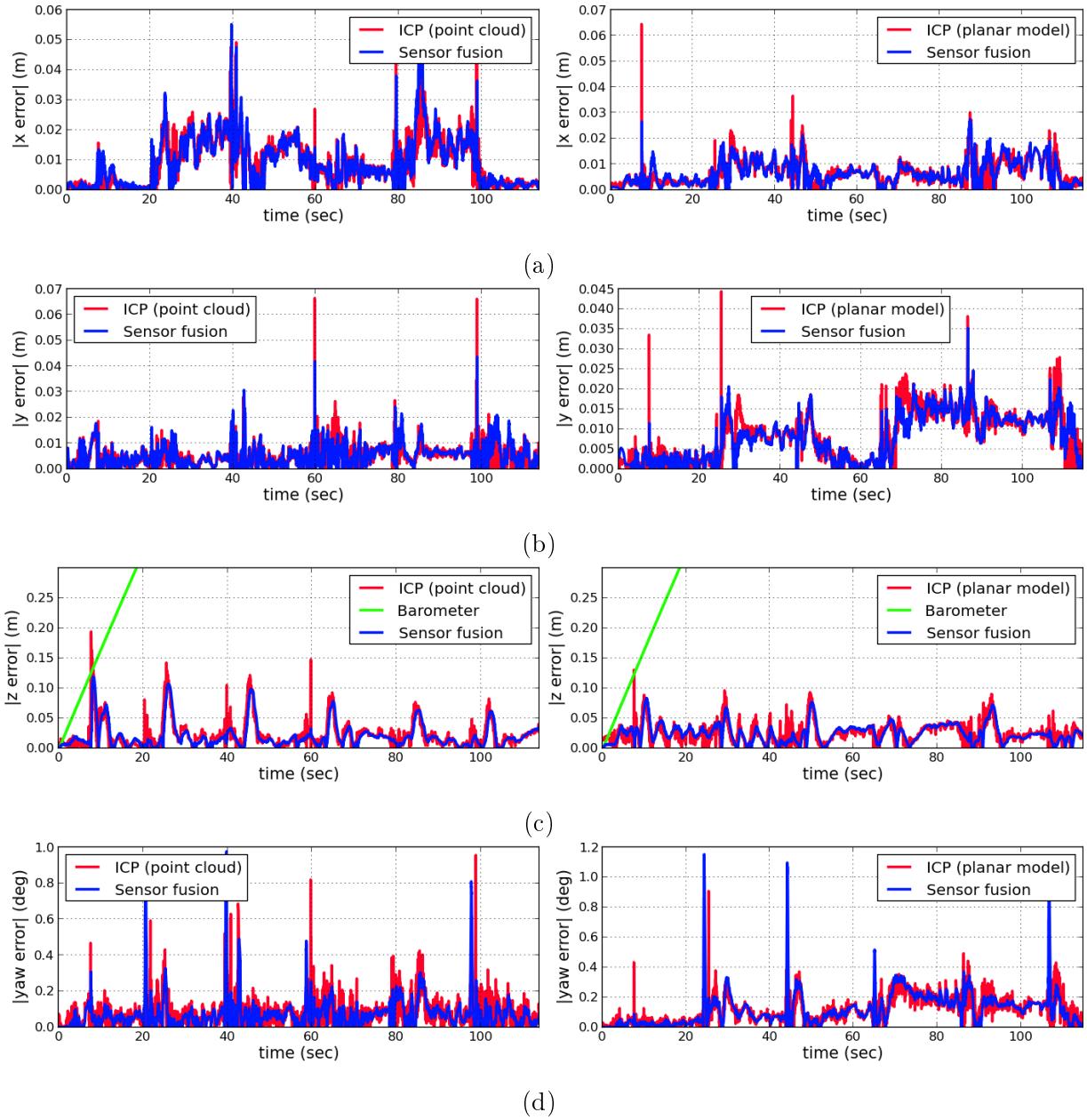
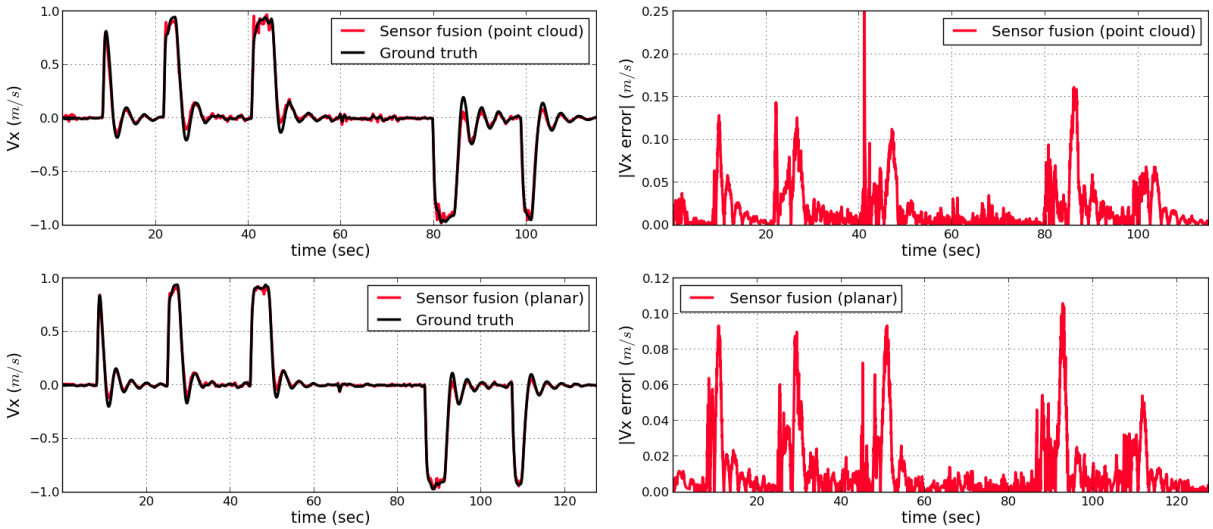
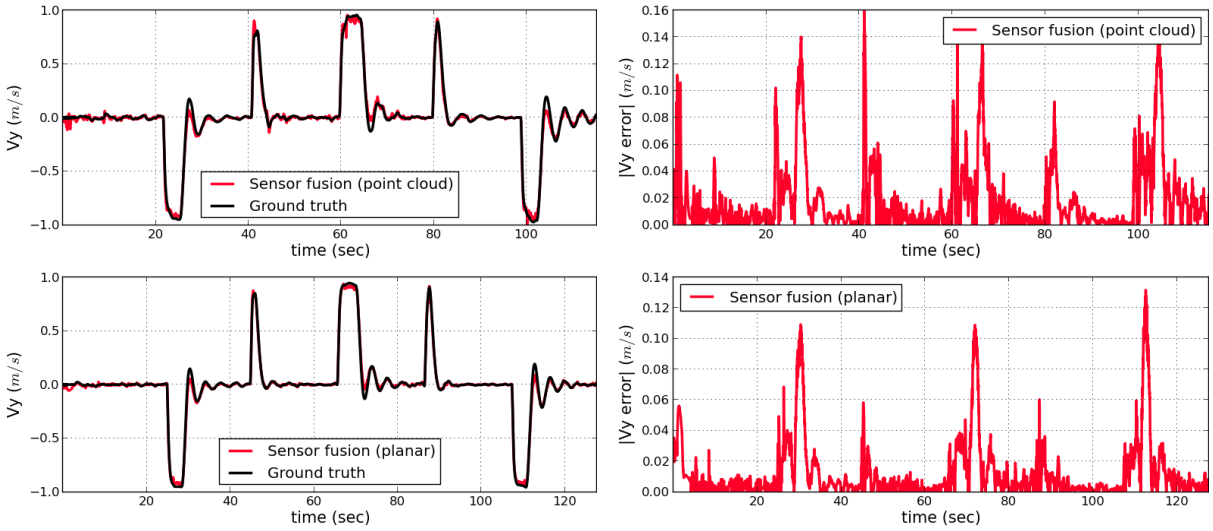


Figure 5.6: The absolute estimation errors of **(a)** x, **(b)** y, **(c)** z and **(d)** ψ, for the simulated flights from Fig. 5.5. **(left)** Comparing the absolute errors of the fused state estimates to the ICP-based laser odometry with the point cloud model (from Fig. 5.5a). **(right)** Comparing the absolute errors of the fused state estimates to the ICP-based laser odometry with the planar tower model (from Fig. 5.5b).



(a)



(b)

Figure 5.7: The horizontal velocity estimates recovered from fusing the laser odometry and accelerometer readings (see Eq. (5.10)). **(a)** Comparing the V_x estimates for the point cloud approach (top), and the planar model approach (bottom). **(b)** Making the same comparison for V_y. In general, the velocity estimation errors for the planar case are smaller. However, both approaches converge rapidly towards the ground truth.

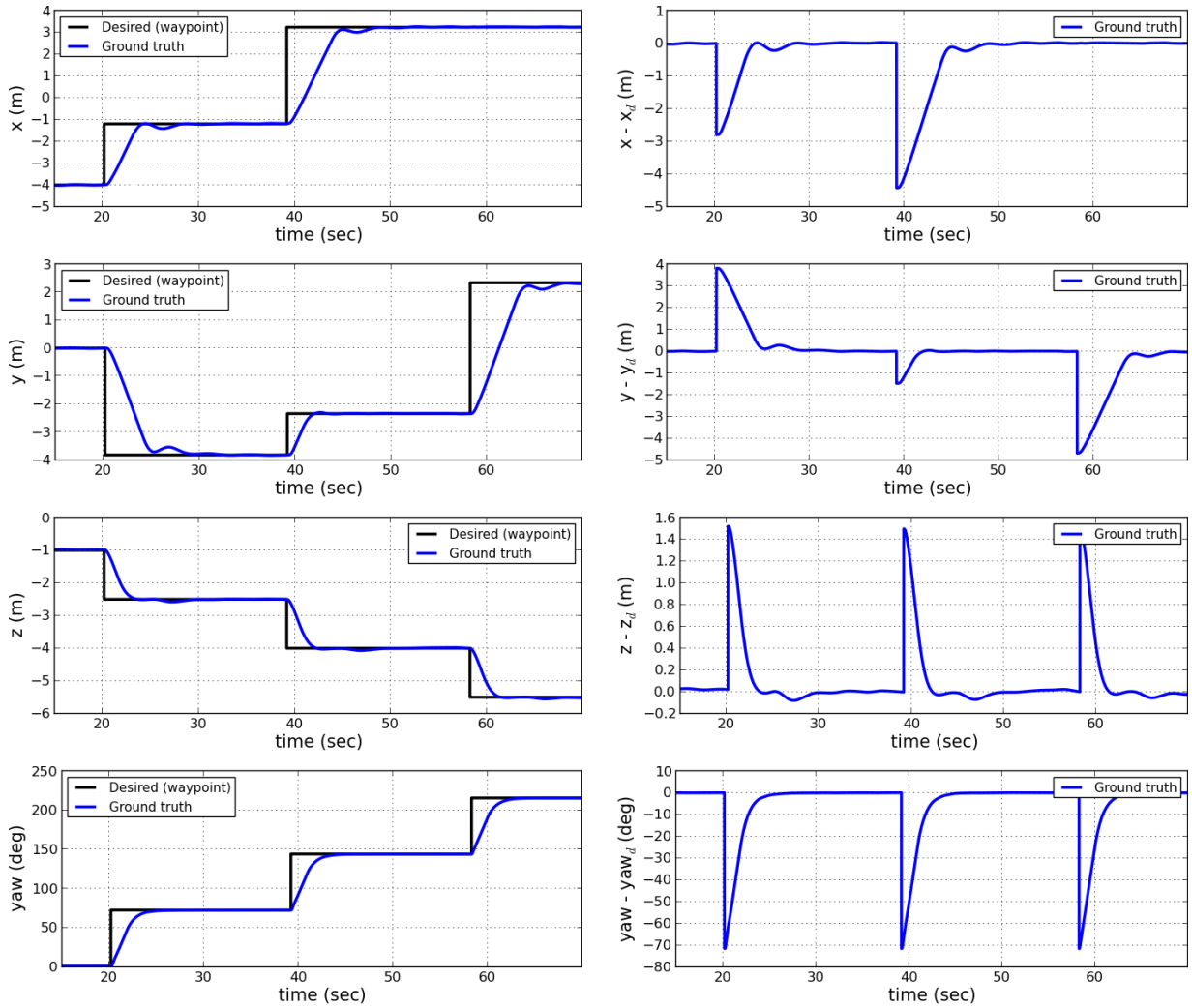
ICP approaches. The main difference, not illustrated in these figures, are the registration rates achieved in each case. As was discussed in Sec. 4.4.2, the scan registrations for the planar model are approximately an order of magnitude faster than the point cloud model registrations (see Fig. 4.7 and Fig. 4.10). This is reflected in the horizontal velocity estimates shown in Fig. 5.7, where it can be observed that, in general, velocity estimation errors for the planar model case are smaller. For V_x (Fig. 5.7a), the maximum error reaches 10 cm/s, for the planar case, 25 cm/s, for the point cloud case. For V_y (Fig. 5.7b), the maximum errors are instead 12 cm/s, for the planar case, and 16 cm/s, for the point cloud case. Nonetheless, in both cases the estimates effectively follow the ground truth velocities for both axis, and errors remain within acceptable levels. Note that vertical velocity V_z is not shown in this figure, since it is deduced from barometer and accelerometer readings, and is the same for both cases (see Fig. 4.11).

5.5.2.2 Feedback control

In the simulations, all of the MAV's 6 DoF were controlled using state estimates from on-board sensing. This included the pose and velocity estimates shown in Fig. 5.6 and Fig. 5.7, fed at a fixed rate of 50 Hz to the MAV control loop, and the $\{\phi, \theta\}$ estimates from accelerometer and gyrometer readings (Eq. (5.4)) not shown in the previous section, used for the low-level attitude control (see Sec. 5.3.1). The time-response for the high-level closed-loop control corresponding to $\{x, y, z, \psi\}$ are shown in Fig. 5.8, for scan registrations with the point cloud model, and in Fig. 5.9, for scan registrations with the planar tower model. In both cases, the control gains for $\{x, y, z\}$ were set to $(k_p, k_v) = (1.96, 2.24)$ (Eq. (5.10)) and $k_\psi = 1.0$ for the yaw control (Eq. (5.6)). No integral terms were used in these simulations. In both cases, the tracking errors converge asymptotically towards zero, as the proposed control architecture manages to effectively stabilize the MAV's pose at the desired location.

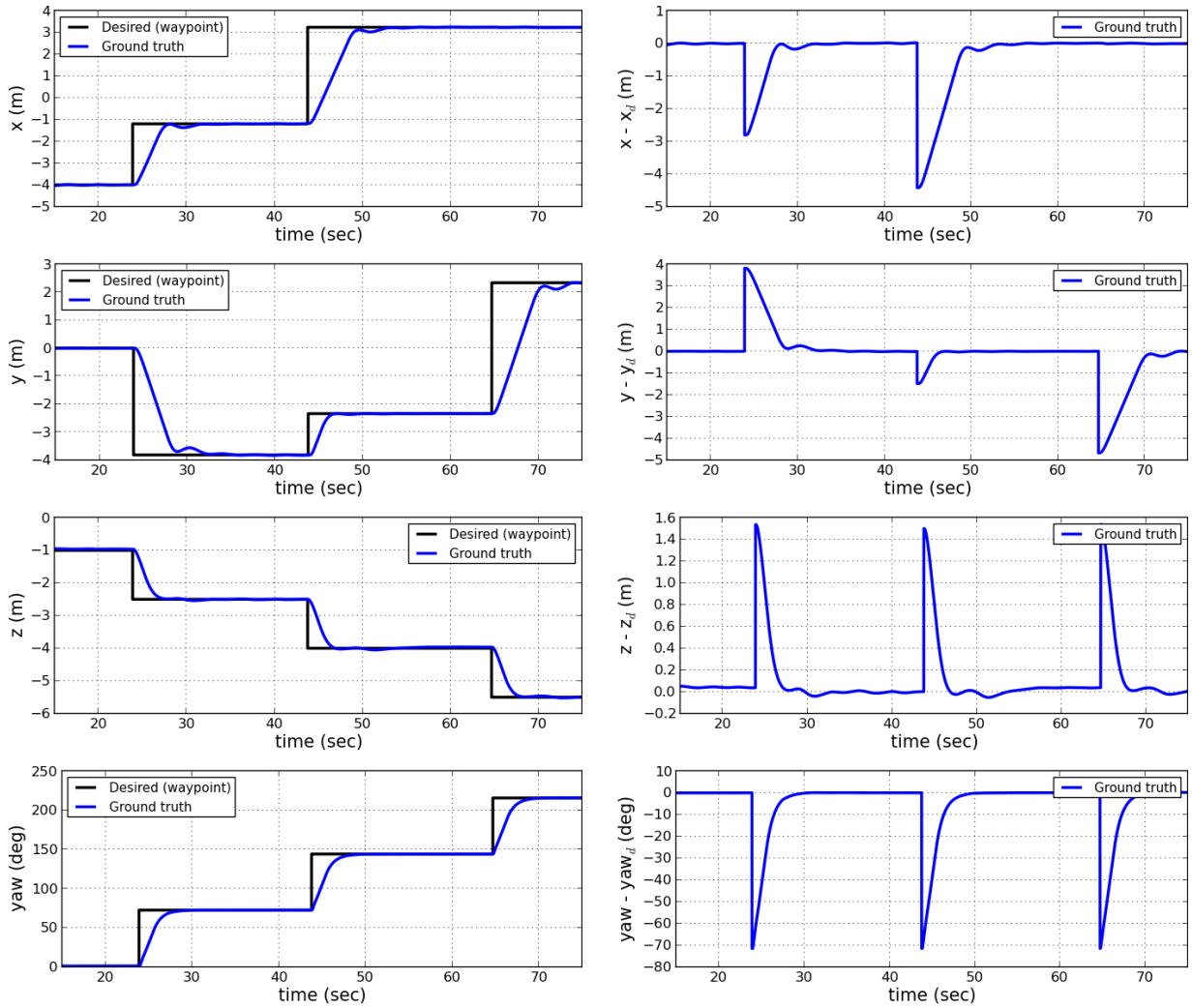
5.6 Conclusions

In this chapter we have provided a brief insight into sensor fusion and control for a MAV platform with our sensor setup. On the one hand, the sensor fusion design built on the real-time pose tracking results, obtained in previous chapters, and the MAV's rigid body dynamics to obtain the required state estimates for a stable control loop. On the other hand, the control design sought to determine the necessary torques and thrust forces that



(a) (b)

Figure 5.8: The resulting closed-loop control of $\{x, y, z, \psi\}$ for the laser odometry with the point cloud approach. **(a)** Time response of the control system for a 70 s portion of the simulated flight and different desired locations. The ground truth pose of the MAV shows that all states converge to the desired location. **(b)** The tracking errors converge asymptotically towards zero for all states.



(a) (b)

Figure 5.9: The resulting closed-loop control of $\{x, y, z, \psi\}$ for the laser odometry with the planar representation approach. **(a)** Time response of the control system for a 70 s portion of the simulated flight and different desired locations. The ground truth pose of the MAV shows that all states converge to the desired location. **(b)** The tracking errors converge asymptotically towards zero for all states.

stabilize a MAV's states to a desired location, relying on the sensor fusion estimates. For practical implementation, the on-board control and estimation architecture was divided into a low-level component, which primarily handled roll/pitch angle estimation and control, and a high-level component, which handled estimation and control of the remaining states. Results were presented for the high-level component, which mainly relied on the on-board laser odometry. These results were obtained from experimental flights in an indoor scene and simulated flights in an electric tower scene. During the test flights, the MAV platform successfully followed sets of pre-defined waypoints with the proposed sensor fusion and control designs.

Experimental validations on the electric towers were not possible as no platform was available for these tests: the platforms used at the beginning of this thesis are no longer maintained and, in view of the recent CNRS regulations for MAVs, they would not be authorized to fly outdoors. Time did not allow us to assemble a new platform for such outdoor tests.

Conclusions and future work

In this thesis, we have presented different methodologies for recovering complete 6 DoF pose estimates on-board MAVs equipped with 2D laser scanners, coupled with an IMU (accelerometer and gyrometer) and a height sensor (laser altimeter o barometer). The main challenges were achieving real-time capabilities, while accounting for the MAV's 3D motion and the influence of the surrounding environment in the laser scans. With respect to this last point, two different scenarios were studied throughout this work.

The first case consisted in a common indoor scene, composed of planar and height-invariant objects, widely studied in literature. A 2D laser odometry based on an adaptation of the Metric-based Iterative Closest Point (MbICP) algorithm was proposed, which uses a fast search in polar coordinates to align pairs of 2D scans projected onto a common horizontal plane. For the remaining states, a non-linear observer formulation was proposed to estimate the MAV's roll and pitch angles from gyrometer and accelerometer measurements. Height estimates were recovered from laser altimeter readings, compensanted for the MAV's inclinations and filtered through a second-order low-pass filter. Experimental flights validated the accuracy of the complete 3D pose tracking approach.

The second scenario was an electric tower inspection scene, which was the main focus of this work. Compared to the indoor case, this scene presented numerous additional challenges. These included the 3D geometry of the towers, the poor information captured by the scans due to the large gaps on the tower's surface and the limited overlap between pairs of 2D laser scans. It was noted that the traditional method of aligning pairs of 2D laser scans through scan matching was not appropriate for this case. Thus, a first approach was proposed which used basic knowledge of the tower's geometry, in this case rectangular cross-sections, to gradually extract notable features captured in the individual 2D laser scans and track the tower's cross-section. While simulation results showed satisfying tracking results under simple flight conditions (vertical flight in front of the tower) the assumptions underlying the method proved too restrictive for general inspection tasks.

It was shown that the proposed tracking method could instead be used with additional sensing to create 3D geometric representations of the tower, as it estimates on-flight the previously unknown dimensions of the cross-sections. This was tested and verified on data obtained from real flights to obtain a 3D point cloud reconstruction and an abstract planar representation of the tower.

Thus, building on these results, a second model-based approach to pose estimation in the electric tower scene was proposed. This consisted in two adaptations of the ICP algorithm, where the 2D laser scans were aligned to a 3D point cloud reconstruction of the tower, on the one hand, and the simplified planar representation of the tower, on the other. In the latter, a projection-based matching strategy was used which allowed obtaining significant speed gains. In both cases, the registration process was performed in 3D and aided by attitude estimates recovered from a proposed gain-scheduled non-linear attitude observer that fused gyrometer and accelerometer measurements. Altitude information was recovered in the process, which was fed into a two-stage altitude observer that fused the laser odometry estimates with barometer readings. This simple formulation allowed estimating the unknown barometer drift in the process. The complete pose tracking approach was validated based on simulated flights, where it was demonstrated that satisfying results could be obtained for the complete 6 DoF pose of the MAV, in terms of accuracy and computation time.

In the final chapter of this thesis, our designs for sensor fusion and feedback control were briefly presented, for the two previously mentioned scenarios. First, sensor fusion relied on simple observers to improve the pose estimates, as decribed in previous chapters, and recover linear velocity estimates. Then, the control loop relied on the sensor fusion outputs to determine the necessary torques and thrust forces that stabilize a MAV's states to a desired location. Results were presented for waypoint following, from experimental flights performed indoors, and simulated flights in the electric tower scene. In both cases, the MAV platform successfully stabilized at the desired locations with the proposed sensor fusion and control designs.

Continuations of this work must address two aspects. The first aspect relates to the methods proposed and developed in this thesis. In this case, immediate future work includes the complete experimental validation of the pose tracking and control designs proposed for the electric tower scene in Chapter 4 and Chapter 5. Moreover, since this study was limited to electric tower bodies with rectangular cross-sections, it would result interesting to develop extensions of the methodology proposed in Chapter 4 to the complete tower structure, including the head of the tower, and to more complex tower

geometries. On the one hand, for the point cloud approach, such extensions would be straight-forward, as it does not rely on a specific parametrization of the structure. On the other hand, the planar approach would require revisiting and generalizing the projection-based matching strategy, which was specific to the planar tower representation. This would further allow testing the planar approach on scenes with predominantly planar structure, different from the electric tower, such as urban scenarios, where the computational efficiency of a projection-based matching strategy for scan registrations could lead to interesting results.

The second aspect relates instead to the long-term goal of achieving completely autonomous inspection capabilities. As described in Chapter 1, pose tracking is only the initial step in a large chain of interconnected tasks. Building on the pose tracking methods proposed in this thesis, future work should seek to develop the set of high-level tools described in Chapter 1. Starting with obstacle avoidance techniques, adapted to the electric tower scene, and proceeding towards complete mapping strategies, for example, relying on state-of-the-art graph-based SLAM methods capable of handling large maps efficiently. Path planning and exploration techniques could then be considered, ultimately opening the way to performing inspection tasks with MAVs in an unsupervised way or with minimal human intervention. For additional robustness, multiple remote sensing modalities should also be considered, together with multi-sensor fusion techniques for large-scale environments, such as smoothing techniques, that have seen promising developments for MAVs in recent years.

Appendices

APPENDIX A

Stability analysis and gain tuning of the altitude observer

In Chapter 4, a method was presented for fusing barometer readings with altitude estimates from a laser odometry. Considering a constant barometer drift b_z and known vertical velocity v_z, which was estimated separately from barometer and accelerometer measurements, a simple second-order observer (Eq. (4.8)) was formulated as

$$\begin{cases} \dot{\hat{z}} &= v_z - k_z(\hat{z} - \overline{z}_1) \\ \dot{\hat{b}}_z &= -k_{b_z}(\hat{z} - \overline{z}_2) \end{cases} \tag{A.1}$$

where (k_z, k_{b_z}) are the estimation gains, and $\overline{z}_n$ is an auxiliary variable defined as

$$\overline{z}_n = \lambda_n(z_{\text{baro}} - \hat{b}_z) + (1 - \lambda_n)z_{\text{laser}} \quad \text{with} \quad \begin{array}{l} 0 \leq \lambda_n \leq 1, \\ n = 1, 2. \end{array} \tag{A.2}$$

which is the weighted sum of the laser altitude estimates z_{laser} and barometer readings z_{baro} compensated with the estimated bias $\hat{b}_z$.

To analyse the stability of the altitude observer formulation from Eq. (A.1), we first deduce error dynamics of the system. Modelling the system inputs as $z_{\text{baro}} = z + b_z$ and $z_{\text{laser}} = z$, and substituting in Eq. (A.2), one obtains

$$\overline{z}_n = z - \lambda_n \tilde{b}_z, \quad 0 \leq \lambda_n \leq 1, \quad n = 1, 2, \tag{A.3}$$

where $\tilde{b}_z = \hat{b}_z - b_z$ is the bias estimation error. Substituting this in Eq. (A.1), and subtracting the vertical dynamics from Eq. (4.4), one obtains the error dynamics of the

system as

$$\begin{cases} \dot{\tilde{z}} &= -k_z(\tilde{z} + \lambda_1 \tilde{b}_z) \\ \dot{\tilde{b}}_z &= -k_{b_z}(\tilde{z} + \lambda_2 \tilde{b}_z) \end{cases} \tag{A.4}$$

where $\tilde{z} = \hat{z} - z$. In matrix form, this is expressed as

$$\begin{bmatrix} \dot{\tilde{z}} \\ \dot{\tilde{b}}_z \end{bmatrix} = \begin{bmatrix} -k_z & -\lambda_1 k_z \\ -k_{b_z} & -\lambda_2 k_{b_z} \end{bmatrix} \begin{bmatrix} \tilde{z} \\ \tilde{b}_z \end{bmatrix} = \tilde{\mathbf{A}} \begin{bmatrix} \tilde{z} \\ \tilde{b}_z \end{bmatrix} \tag{A.5}$$

Stability analysis follows, by analysing the roots of the characteristic polynomial of Eq. (A.5), obtained from solving $\det(\mathrm{s}\mathbf{I} - \tilde{\mathbf{A}}) = 0$. This results in

$$s^2 + (\lambda_2 k_{b_z} + k_z)s + k_{b_z} k_z (\lambda_2 - \lambda_1) = 0, \quad \lambda_1 \neq \lambda_2, \tag{A.6}$$

where the $\lambda_1 \neq \lambda_2$ condition avoids a null constant term in the polynomial. Then, exponential convergence is guaranteed if the two roots of the characteristic polynomial have negative real parts. This can be achieved with a simple pole placement approach. Recalling the characteristic polynomial for a second order system

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \quad \zeta, \omega_n > 0, \tag{A.7}$$

where the damping ratio ζ and the natural frequency ω_n define the closed-loop poles, which have negative real part if $\zeta, \omega_n > 0$. The observer gains are then determined by comparing the coefficients of both polynomials. This results in two cases depending on the value of λ_2.

On the one hand, if $\lambda_2 = 0$, then solving by substitution the system of equations resulting from Eq. (A.6) and Eq. (A.7), one obtains

$$\begin{cases} k_z = 2\zeta\omega_n \\ k_{b_z} = -\dfrac{\omega_n^2}{\lambda_1 k_z}, \quad \lambda_2 = 0,\ 0 < \lambda_1 \leq 1. \end{cases} \tag{A.8}$$

In this simple case, to determine (k_z, k_{b_z}), one must first choose the closed-loop poles for the desired system response, which defines the value of ζ and ω_n, and then set λ_1 to the desired value. As explained in Sec. 4.3.2, the weights λ_n allow one to determine how each sensor contributes to the estimation of each state b_z and z. In particular, as λ_n increases, higher priority is given to the barometer readings. By setting the weights as $\lambda_2 = 0$ and $\lambda_1 = 1$, the laser estimates are mainly used to keep track of the barometer bias $\hat{b}_z$,

while the more reliable barometer measurements determine $\hat{z}$. Then, considering $\zeta = 1.1$ (overdamped response) and $\omega_n = 3.0$ in Eq. (A.8), one obtains $(k_z, k_{b_z}) = (6.6, -1.36)$, which were the gains used in the simulations from Chapter 4.

On the other hand, if $\lambda_2 \neq 0$, this leads to a quadratic expression for k_z and k_{b_z}, obtaining

$$\begin{cases} k_z = \dfrac{2\zeta\omega_n \mp \sqrt{(2\zeta\omega_n)^2 - \frac{4\lambda_2\omega_n^2}{\lambda_2-\lambda_1}}}{2} \\ k_{b_z} = \dfrac{2\zeta\omega_n \pm \sqrt{(2\zeta\omega_n)^2 - \frac{4\lambda_2\omega_n^2}{\lambda_2-\lambda_1}}}{2\lambda_2}, \quad 0 < \lambda_2 \leq 1,\ \lambda_1 \neq \lambda_2. \end{cases} \tag{A.9}$$

Then, to avoid complex gains, the discriminant Δ must be nonnegative. That is,

$$\Delta = (2\zeta\omega_n)^2 - \frac{4\lambda_2\omega_n^2}{\lambda_2 - \lambda_1} \geq 0, \tag{A.10}$$

leading to the following inequality

$$\zeta^2 \geq \frac{\lambda_2}{\lambda_2 - \lambda_1}, \tag{A.11}$$

which conditions the values of ζ and λ_n. In this case, a simple way of tuning the gains is to first choose the closed-loop poles, obtaining ζ and ω_n, then set λ_2 to the desired value and finally set λ_1 ensuring that Eq. A.11 holds.

Bibliography

[1] F. Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378, 2012.

[2] L. Matikainen, M. Lehtomäki, E. Ahokas, J. Hyyppä, M. Karjalainen, A. Jaakkola, A. Kukko, and T. Heinonen. Remote sensing methods for power line corridor surveys. *ISPRS Journal of Photogrammetry and Remote Sensing*, 119:10–31, 2016.

[3] R.V. Ussyshkin, L. Theriault, M. Sitar, and T. Kou. Advantages of airborne lidar technology in power line asset management. In *International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM)*, pages 1–5, January 2011.

[4] Z. Li, T. Bruggemann, J. Ford, L. Mejias, and Y. Liu. Toward automated power line corridor monitoring using advanced aircraft control and multisource feature fusion. *Journal of Field Robotics*, 29(1):4–24, 2012.

[5] R.A. McLaughlin. Extracting transmission lines from airborne LIDAR data. *IEEE Geoscience and Remote Sensing Letters*, 3(2):222–226, April 2006.

[6] G. Sohn, Y. Jwa, and H. B. Kim. Automatic powerline scene classification and reconstruction using airborne lidar data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, 13(167172):28, 2012.

[7] Q. Li, Z. Chen, and Q. Hu. A model-driven approach for 3D modeling of pylon from airborne LiDAR data. *Remote Sensing*, 7(9):11501–11524, 2015.

[8] S. Montambault, J. Beaudry, K. Toussaint, and N. Pouliot. On the application of VTOL UAVs to the inspection of power utility assets. In *1st International Conference on Applied Robotics for the Power Industry (CARPI)*, pages 1–7, Oct 2010.

[9] J. Katrašnik, F. Pernuš, and B. Likar. A survey of mobile robots for distribution power line inspection. *IEEE Transactions on Power Delivery*, 25(1):485–493, 2010.

[10] L. Luque-Vega, B. Castillo-Toledo, A. Loukianov, and L. Gonzalez-Jimenez. Power line inspection via an unmanned aerial system based on the quadrotor helicopter. In *IEEE Mediterranean Electrotechnical Conference (MELECON)*, pages 393–397, 2014.

[11] P.-L Richard, N. Pouliot, and S. Montambault. Introduction of a lidar-based obstacle detection system on the linescout power line robot. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1734–1740, July 2014.

[12] S. Hrabar. 3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 807–814, 2008.

[13] D. Jones. Power line inspection - A UAV concept. In *The IEE Forum on Autonomous Systems*, pages 8–pp. IET, 2005.

[14] I. Golightly and D. Jones. Visual control of an unmanned aerial vehicle for power line inspection. In *Proceedings of IEEE International Conference On Advanced Robotics (ICAR)*, pages 288–295, 2005.

[15] P. Campoy, J. Correa, I. Mondragón, C. Martínez, M. Olivares, L. Mejías, and J. Artieda. Computer vision onboard UAVs for civilian tasks. *Journal of Intelligent & Robotic Systems*, 54(1):105–135, 2009.

[16] M. Williams, D. Jones, and G. Earp. Obstacle avoidance during aerial inspection of power lines. *Aircraft Engineering and Aerospace Technology*, 73(5):472–479, 2001.

[17] P. Campoy, P. Garcia, A. Barrientos, J. del Cerro, I. Aguirre, A. Roa, R. Garcia, and J. Muñoz. An stereoscopic vision system guiding an autonomous helicopter for overhead power cable inspection. *Robot Vision*, pages 115–124, 2001.

[18] C. Sun, R. Jones, H. Talbot, X. Wu, K. Cheong, R. Beare, M. Buckley, and M. Berman. Measuring the distance of vegetation from powerlines using stereo vision. *ISPRS journal of photogrammetry and remote sensing*, 60(4):269–283, 2006.

[19] A. Bachrach, S. Prentice, R. He, and N. Roy. RANGE – Robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics*, 28(5):644–666, 2011.

[20] S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 20–25, May 2011.

[21] S. Grzonka, G. Grisetti, and W. Burgard. A fully autonomous indoor quadrotor. *IEEE Transactions on Robotics*, 28(1):90–100, February 2012.

[22] I. Dryanovski, R. Valenti, and J. Xiao. An open-source navigation system for micro aerial vehicles. *Autonomous Robots*, 34(3):177–188, 2013.

[23] I. Sa and P. Corke. System identification, estimation and control for a cost effective open-source quadcopter. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2202–2209, May 2012.

[24] K. Kuhnert and L. Kuhnert. Light-weight sensor package for precision 3D measurement with micro UAVs eg power-line monitoring. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(2):235–240, 2013.

[25] R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics & and Automation Magazine*, (19):20–32, 2012.

[26] B. Guo, X. Huang, Q. Li, F. Zhang, J. Zhu, and C. Wang. A stochastic geometry method for pylon reconstruction from airborne lidar data. *Remote Sensing*, 8(3):243, 2016.

[27] K. Lingemann, H. Surmann, A. Nuchter, and J. Hertzberg. Indoor and outdoor localization for fast mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2185–2190, 2004.

[28] A. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2D and 3D face recognition: A survey. *Pattern recognition letters*, 28(14):1885–1906, 2007.

[29] Y. Okubo, C. Ye, and J. Borenstein. Characterization of the Hokuyo URG-04LX laser rangefinder for mobile robot obstacle negotiation. *International Society for Optics and Photonics in SPIE Defense, Security, and Sensing*, pages 733212–733212, 2009.

[30] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic systems*, 18(3):249–275, 1997.

[31] A. Censi, L. Iocchi, and G. Grisetti. Scan matching in the Hough domain. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2739–2744, 2005.

[32] Q. Li and J. Griffiths. Iterative closest geometric objects registration. *Computers & mathematics with applications*, 40(10-11):1171–1188, 2000.

[33] A. Censi. An ICP variant using a point-to-line metric. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 19–25, 2008.

[34] P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[35] A. Diosi and L. Kleeman. Laser scan matching in polar coordinates with application to slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3317–3322, Aug 2005.

[36] F. Pomerleau, F. Colas, R. Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, 2015.

[37] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[38] N. J Mitra, N. Gelfand, H. Pottmann, and L. Guibas. Registration of point cloud data from a geometric optimization perspective. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 22–31, 2004.

[39] A. Gressin, C. Mallet, J. Demantké, and N. David. Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS journal of photogrammetry and remote sensing*, 79:240–251, 2013.

[40] R. Fisher and K. Konolige. *Range Sensors*, pages 521–542. Springer Berlin Heidelberg, 2008.

[41] D. Hähnel and W. Burgard. Probabilistic matching for 3D scan registration. In *Proc. of the VDI-Conference Robotik*, 2002.

[42] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

[43] E. Olson. Real-time correlative scan matching. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4387–4393, 2009.

[44] L. Montesano, J. Minguez, and L. Montano. Probabilistic scan matching for motion estimation in unstructured environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3499–3504, 2005.

[45] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. volume 2, page 435, 2009.

[46] A. Censi. Scan matching in a probabilistic framework. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2291–2296, 2006.

[47] R. Sandhu, S. Dambreville, and A. Tannenbaum. Point set registration via particle filtering and stochastic dynamics. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1459–1473, 2010.

[48] V. Zaliva and F. Franchetti. Barometric and GPS altitude sensor fusion. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7525–7529, 2014.

[49] A. Sabatini and V. Genovese. A sensor fusion method for tracking vertical velocity and height based on inertial and barometric altimeter measurements. *MDPI Sensors*, 14(8):13324–13347, July 2014.

[50] Y. Son and S. Oh. A barometer-IMU fusion method for vertical velocity and height estimation. In *IEEE Sensors*, pages 1–4, 2015.

[51] M. Tanigawa, H. Luinge, L. Schipper, and P. Slycke. Drift-free dynamic height sensor using MEMS IMU aided by MEMS pressure sensor. In *Workshop on Positioning Navigation and Communication*, pages 191–196, March 2008.

[52] P. Gąsior, A. Bondyra, S. Gardecki, and W. Giernacki. Robust estimation algorithm of altitude and vertical velocity for multirotor UAVs. In *IEEE International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 714–719, 2016.

[53] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proc. of the AIAA Guidance, Navigation, and Control Conference*, volume 2, 2007.

[54] P. Martin and E. Salaün. The true role of accelerometer feedback in quadrotor control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1623–1629, May 2010.

[55] T. Yoo, S. Hong, H. Yoon, and S. Park. Gain-scheduled complementary filter design for a mems based attitude and heading reference system. *Sensors*, 11(4):3816–3830, 2011.

[56] H. Rehbinder and X. Hu. Drift-free attitude estimation for accelerated rigid bodies. *Automatica*, 40(4):653–659, April 2004.

[57] H. Zhao and Z. Wang. Motion Measurement Using Inertial Sensors, Ultrasonic Sensors, and Magnetometers With Extended Kalman Filter for Data Fusion. *IEEE Sensors Journal*, 12(5):943–953, May 2012.

[58] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart. A robust and modular multi-sensor fusion approach applied to MAV navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3923–3929, November 2013.

[59] R. Mahony, T. Hamel, and J. M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, June 2008.

[60] A. Tayebi, S. McGilvray, A. Roberts, and M. Moallem. Attitude estimation and stabilization of a rigid body using low-cost sensors. In *IEEE Conference on Decision and Control*, pages 6424–6429, December 2007.

[61] S. Roumeliotis and J. Burdick. Stochastic cloning: A generalized framework for processing relative state measurements. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1788–1795, 2002.

[62] A. Mourikis, S. Roumeliotis, and J. Burdick. SC-KF mobile robot localization: A stochastic cloning Kalman filter for processing relative-state measurements. *IEEE Transactions on Robotics*, 23(4):717–730, 2007.

[63] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar. Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft mav. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4974–4981, 2014.

[64] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999(343-349):2–2, 1999.

[65] S. Thrun. Particle filters in robotics. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 511–518. Morgan Kaufmann Publishers Inc., 2002.

[66] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1-2):99–141, 2001.

[67] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics Automation Magazine*, 13(2):99–110, June 2006.

[68] S. Thrun and J. Leonard. Simultaneous localization and mapping. In *Springer handbook of robotics*, pages 871–889. Springer, 2008.

[69] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[70] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[71] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Aaai*, pages 593–598, 2002.

[72] K. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems*, pages 1015–1021, 2000.

[73] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based SLAM with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2432–2437, 2005.

[74] J. Santos, D. Portugal, and R. Rocha. An evaluation of 2D SLAM techniques available in robot operating system. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6, 2013.

[75] A. Bachrach. *Autonomous flight in unstructured and unknown indoor environments*. PhD thesis, Massachusetts Institute of Technology, 2009.

[76] S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429, 2006.

[77] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[78] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 273–278, 2010.

[79] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, 2011.

[80] G. Grisetti, C. Stachniss, and W. Burgard. Nonlinear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):428–439, 2009.

[81] A. Ranganathan, M. Kaess, and F. Dellaert. Fast 3d pose estimation with out-of-sequence measurements. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2486–2493, 2007.

[82] V. Indelman, S. Williams, M. Kaess, and F. Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *IEEE International Conference on Information Fusion (FUSION)*, pages 2154–2161, 2012.

[83] T. Larsen, N. Andersen, O. Ravn, and N. Poulsen. Incorporation of time delayed measurements in a discrete-time kalman filter. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 4, pages 3972–3977, 1998.

[84] S. Lange, N. Sünderhauf, and P. Protzel. Incremental smoothing vs. filtering for sensor fusion on an indoor UAV. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1773–1778, 2013.

[85] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske, and S. Singh. River mapping from a flying robot: state estimation, river detection, and obstacle mapping. *Autonomous Robots*, 33(1-2):189–214, 2012.

[86] M. Nieuwenhuisen, D. Droeschel, M. Beul, and S. Behnke. Obstacle detection and navigation planning for autonomous micro aerial vehicles. In *IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1040–1047, 2014.

[87] M. Nieuwenhuisen, D. Droeschel, M. Beul, and S. Behnke. Autonomous navigation for micro aerial vehicles in complex GNSS-denied environments. October 2015.

[88] D. Droeschel, M. Nieuwenhuisen, M. Beul, D. Holz, J. Stückler, and S. Behnke. Multilayered mapping and navigation for autonomous micro aerial vehicles. *Journal of Field Robotics*, 33(4):451–475, 2015.

[89] S Shen, N. Michael, and V. Kumar. Autonomous indoor 3D exploration with a micro-aerial vehicle. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9–15, 2012.

[90] C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *Journal of Intelligent and Robotic Systems*, 57(1-4):65, 2010.

[91] H. Cover, S. Choudhury, S. Scherer, and S. Singh. Sparse tangential network (SPARTAN): Motion planning for micro aerial vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2820–2825, 2013.

[92] S. Jain, S. Nuske, A. Chambers, L. Yoder, H. Cover, L. Chamberlain, S. Scherer, and S. Singh. Autonomous river exploration. In *Field and Service Robotics*, pages 93–106. Springer, 2015.

[93] S. Nuske, S. Choudhury, S. Jain, A. Chambers, L. Yoder, S. Scherer, L. Chamberlain, H. Cover, and S. Singh. Autonomous exploration and motion planning for an unmanned aerial vehicle navigating rivers. *Journal of Field Robotics*, 32(8):1141–1162, 2015.

[94] L. Yoder and S. Scherer. Autonomous exploration for infrastructure modeling with a micro aerial vehicle. In *Field and service robotics*, pages 427–440. Springer, 2016.

[95] J. Zhang and S. Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, 2014.

[96] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[97] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013.

[98] R. Bogdan Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[99] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy. Geometrically stable sampling for the ICP algorithm. In *Proceedings of the IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 260–267, 2003.

[100] A. Nuchter, K. Lingemann, and J. Hertzberg. Cached KD tree search for ICP algorithms. In *IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 419–426, 2007.

[101] J. Minguez, F. Lamiraux, and L. Montesano. Metric-based scan matching algorithms for mobile robot displacement estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, page 3557, 2005.

[102] L. Armesto, J. Minguez, and L. Montesano. A generalization of the metric-based iterative closest point technique for 3D scan matching. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1367–1372, 2010.

[103] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2724–2729, 1991.

[104] H. Pottmann, Q. Huang, Y. Yang, and S. Hu. Geometry and convergence analysis of algorithms for registration of 3D shapes. *International Journal of Computer Vision*, 67(3):277–296, 2006.

[105] A. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21(13):1145–1153, 2003.

[106] D. Campbell, M. Whitty, and S. Lim. Mobile 3D indoor mapping using the continuous normal distributions transform. In *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–9, 2012.

[107] J. Yang, H. Li, D. Campbell, and Y. Jia. Go-ICP: a globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2241–2254, 2016.

[108] G. Borges and M. Aldon. Line extraction in 2D range images for mobile robotics. *Journal of Intelligent & Robotic Systems*, 40(3):267–297, 2004.

[109] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[110] W. Gander and J. Hrebicek, editors. *Some Least Squares Problems*. Springer-Verlag Berlin Heidelberg, 4 edition, 2004.

[111] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2149–2154 vol.3, Sept 2004.

[112] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng. ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[113] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk. Comprehensive simulation of quadrotor UAVs using ROS and Gazebo. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pages 400–411. Springer, 2012.

[114] G. Blais and M. D. Levine. Registering multiview range data to create 3D computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, Aug 1995.

[115] S-Y. Park and M. Subbarao. A fast point-to-tangent plane technique for multi-view registration. In *IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 276–283, 2003.

[116] R. Valenti, I. Dryanovski, and J. Xiao. Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs. *Sensors*, 15(8):19302–19330, 2015.

[117] S. Bouabdallah and R Siegwart. Full control of a quadrotor. In *IEEE/RSJ international conference on Intelligent robots and systems (IROS)*, pages 153–158, 2007.

[118] F. Kendoul, I. Fantoni, R. Lozano, et al. Asymptotic stability of hierarchical inner-outer loop-based flight controllers. In *Proceedings of the 17th IFAC world congress*, pages 1741–1746, 2008.

[119] H. De Plinval, P. Morin, and P. Mouyon. Stabilization of a class of underactuated vehicles with uncertain position measurements and application to visual servoing. *Automatica*, 77:155–169, 2017.

[120] M.-D. Hua, T. Hamel, P. Morin, and C. Samson. Introduction to feedback control of underactuated vtol vehicles. *IEEE Control Systems Magazine*, pages 61–75, February 2013.

[121] J.-M. Pflimlin, T. Hamel, P. Soueres, and R. Mahony. A hierarchical control strategy for the autonomous navigation of a ducted fan flying robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2491–2496, 2006.