MAV local pose estimation with a 2D laser scanner: A case study for electric tower inspection

Carlos Viña, Pascal Morin

MAV local pose estimation with a 2D laser scanner: A case study for electric tower inspection

Carlos Viña and Pascal Morin
Institut des Systèmes Intelligents et de Robotique (ISIR)
Sorbonne Université, CNRS UMR 7222, France.

Abstract

Automation of inspection tasks is crucial for the development of the power industry, where MAVs have shown a great potential. Self-localization in this context remains a key issue, and is the main subject of this work. This article presents a methodology to obtain complete 3D local pose estimates in electric tower inspection tasks with MAVs, using an on-board sensor setup consisting of a 2D LiDAR, a barometer sensor and an IMU. First, we present a method to track the tower's cross-sections in the laser scans, and give insights on how this can be used to model electric towers. Then, we show how the popular ICP algorithm, that is typically limited to indoor navigation, can be adapted to this scenario, and propose two different implementations to retrieve pose information. This is complemented with attitude estimates from the IMU measurements, based on a gain-scheduled non-linear observer formulation. An altitude observer to compensate for barometer drift is also presented. Finally, we address velocity estimation with views to feedback position control. Validations based on simulations and experimental data are presented.

Keywords: MAV, airborne laser scanning, 2D LiDAR, barometer, inertial measurement unit, ICP, state estimation.

1 Introduction

Power utilities, such as transmission line 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. For this purpose, aerial surveys have been increasingly common as they allow covering vast areas in relatively short periods of time, by relying on remote sensing technologies such as thermal imaging, aerial imaging, optical satellites, among others [27, 21]. 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 [45, 21]. In ALS applications, powerful 3D light detection and ranging (LiDAR) sensors are mounted on manned aircraft, such as helicopters [27, 21, 28], 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 processed afterwards for a wide variety of classification or reconstruction tasks such as detecting power lines [28, 41], vegetation management [45] and making 3-D models of the electric towers [20]. Nonetheless, the high operational costs of piloted aircraft have constrained the proliferation of these applications. The automation of inspection tasks has thus become a key subject of research in the power industry, in which unmanned air vehicles (UAVs) have surfaced as an attractive solution, as they provide an affordable and flexible mean of gathering spatial data [30, 16, 22]. This has been mainly fuelled by developments in lithium polymer batteries that have led to larger flight durations and increased payload capabilities. However, these small platforms currently can't carry the heavy LiDARs required in most ALS applications, and research on inspection tasks with UAVs has mainly focused on vision-based approaches instead [27, 16, 15, 10]. Rapid advances in light-weight LiDARs have made them an appealing alternative for UAVs, and while performance and precision remain far from their 3D counterparts, they can be used for basic

and affordable ALS applications, which has already been demonstrated in previous works, for example, for power line monitoring [19].

In the context of power utility inspection, GPS sensors remain the predominant choice for achieving autonomous flight capabilities with UAVs [30]. Nonetheless, a GPS signal is not always accurate, can be perturbed by the strong electromagnetic fields in the proximity of the power lines [35] 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 [27, 30, 22]. Vision-based navigation systems have been proposed as a substitute in numerous works, relying mostly on tracking and following the power lines [15, 10]. On the other hand, light-weight LiDARs can also be employed for autonomous navigation purposes, and have been successfully used for indoor flights with micro air vehicles (MAVs) [1, 40, 11, 6, 37]. These sensors excel when navigating in cluttered environments, as they directly measure the distance to surrounding objects and naturally open the way for sense-and-avoid functionalities required for safe flights. As a consequence, they can allow achieving higher levels of autonomy and close-up inspections in power line corridors, which is hard to accomplish with other sensors. In this work, we focus on the inspection of transmission line towers, and we explore how 2D LiDARs coupled with commonly available sensors can be used for pose estimation purposes in these scenarios.

2 Problem statement

One of the first tasks that any autonomous platform must achieve is self-localization. Thus, our primary goal is to obtain real-time estimates of a MAV's 6 degree of freedom (DoF) pose with respect to an electric tower, using uniquely on-board sensors and processing capabilities. 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. 1. For this first case study, we focus 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 [20, 12], and are not considered in this work.

After treating the self-localization problem, the last part of this study focuses on obtaining velocity estimates and sensor fusion techniques are used for this purpose. Accurate velocity estimates are necessary in the control loop to successfully stabilize a MAV's position. Feedback position control, however, is not addressed in this study. The long term aim of this work is to achieve autonomous inspection capabilities of electric towers with MAVs.

3 Related works

While laser range finders have been largely popular among ground robots for autonomous navigation tasks, aerial robots present additional complications that make similar applications not so straightforward. First, flying robots' motion is three-dimensional. Then, payload limitations prevent the use of more powerful sensors such as 3D laser range finders. Finally, flying robots have fast dynamics that make them harder to stabilize and any state estimation has to be made with low delays and an adequate level of accuracy. This means that estimation and control algorithms must be preferably implemented on board, and must run at high speeds, which limits the complexity of the algorithms that can be used, so as to avoid significant processing delays. Nonetheless, fully autonomous capabilities for MAVs equipped with 2D LiDARs have been shown in numerous previous works [1, 40, 11, 6, 37, 18, 39, 5, 50], which have mainly focused on indoors scenarios. Most of these studies adopt a similar strategy: the first goal is to obtain fast and accurate 3D pose estimates from the embarked sensors' measurements, preferably using on-board processing capabilities; then, a second goal is to to derive estimates of the linear velocities using the pose estimates and sensor fusion techniques. We now present how several previous works have addressed these two tasks.

Figure 1: A common high voltage transmission line.

3.1 Laser-based local pose estimation on-board MAVs

Regarding the first goal, this is partly achieved by aligning pairs of laser scans to recover the MAV's relative displacement, a technique known as *scan matching* or *scan registration*. While these algorithms can pose a heavy computational burden, satisfying real-time results have been obtained from adaptations of well-known techniques, such as the iterative closest point (ICP) algorithm [2, 40, 6, 37], and the correlative scan matching (CSM) algorithm [31, 1, 11]. Following the satisfying results previously obtained on-board MAVs, and due to its simplicity and efficiency, the ICP algorithm was chosen for the scan registrations.

3.1.1 Typical approach with the ICP algorithm on-board MAVs.

A classic implementation of the ICP algorithm in navigation tasks consists in aligning the current laser scan to the preceding scan. This is known as *incremental scan matching*, and is known to lead to drift over time [40, 37, 6]. An alternative is to use a *keyframe* approach [6], with a reference scan instead fixed at some initial time. As long as the robot remains in the proximity of this keyframe, and as long as there is sufficient overlap, the estimation error remains bounded and the results are drift-free. The ICP implementations proposed in this article go along this line of work.

In general, on MAVs equipped with 2D LiDARs, the ICP algorithm is limited to aligning pairs of 2D laser scans to recover 2D pose estimates. The remaining states are estimated from separate sensing (e.g., IMU for attitude estimation [1, 40, 11, 6, 37] and laser altimeter for altitude estimation [1, 40]). However, to align pairs of 2D laser scans the measurements must be taken within the same plane. This poses a major drawback for aerial robots, and requires coping with the 3D motion. A simple solution is to project the laser points to a common horizontal plane using attitude estimations from IMUs [1, 40, 37, 6]. Then, the projected scans are aligned with the ICP algorithm. Nonetheless, this has the underlying assumption that surrounding objects are planar and height-invariant, which holds for common indoors scenarios, with mainly straight walls. In an inspection scene, this assumption doesn't hold as the electric towers have a geometry that varies greatly in 3D. Hence, in our scenario aligning pairs of 2D scans in similar way isn't possible. In this work we explore alternative ways in which pose information can be recovered from the laser scans, by exploiting basic knowledge of the tower's geometry. We also explore two different ways in which the ICP algorithm can be extended to electric tower case.

3.1.2 Limitations of the ICP algorithm for self-localization.

It is important to note that scan matching techniques, such as ICP, only guarantee local convergence and depend highly on a good initial guess[2, 36]. A bad initialization may lead ICP to converge to a local minimum far from the optimal solution. Furthermore, these techniques typically can't recover from large estimation errors. Globally optimal solutions for the ICP algorithm have been studied in the past [47], but are typically too slow for state estimation purposes. In literature, to overcome these issues, it is common for simultaneous localization and mapping (SLAM) techniques to be used in parallel [1, 40, 11, 6, 50]. These algorithms provide pose estimates with guaranteed global consistency, that are less sensitive to initialization errors, and that can allow detecting and correcting errors from scan matching. The faster local pose estimates are still required as an odometric input to SLAM, to initialize and speed up the mapping process [1, 6]. However, SLAM remains very computationally expensive and is commonly performed off-board [1, 11], with only a handful of studies achieving on-board capabilities [40, 6], at very low rates (2-10Hz). Thus, the global pose estimates are seldom included directly in the control loop, and are mainly limited to providing periodic corrections to the real-time pose estimates from scan matching [1], and to perform higher level tasks such as path planning [11, 6] and obstacle avoidance [11]. For the purposes of this article, we focus only on the local pose estimation problem, keeping in mind that mapping methods can be used in parallel.

Another complex issue is that scan matching performance has a strong dependence on the shape of the surrounding environment, as the laser scans must capture sufficient geometric detail in order to extract any useful pose information. The algorithm will thus fail under highly unstructured scenarios, often faced outdoors, or featureless scenarios, such as long hallways or circular rooms. This, in reality, corresponds to inherent

limitations of laser range sensing [1]. Previous works have addressed this issue incorporating multiple sensing modalities, such as GPS sensors, ultrasonic sensors and cameras [39, 5]. This, however, goes beyond the scope of this work.

3.2 Altitude estimation on-board MAVs

On MAVs equipped with 2D LiDARs, altitude is commonly estimated by placing mirrors to reflect multiple laser rays downwards and directly measuring the distance to the ground assuming that the ground elevation is piecewise constant for the most part [1, 11, 6]. However, to account for potential discontinuities and changing floor elevations several solutions have been proposed, such as creating multilevel grid maps of the ground [11], or creating histograms of the range measurements to detect edges and floor level changes [6]. While this has proven to be effective when navigating indoors, performance remains highly dependent on the floor's layout, which can be very irregular in typical outdoors inspection scenarios.

On the other hand, barometric sensors are also popular 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 [49], and IMUs [38, 42]. More recently, differential barometry has been gaining popularity [43, 13] . 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.

3.3 Attitude estimation on-board 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, 14, 26]. On one hand, magnetometers provide measurements of the surrounding magnetic field in the body attached frame, and allow deducing the MAV's heading [48, 25]. 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 [26, 25, 34], 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 [14]. This in exchange can introduce important latencies in the estimations. Thus, complementary attitude information is commonly obtained from gyroscopes, 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 [14].

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 [14] or extended Kalman filters [51, 23], 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 [24]. An alternative is to use non-linear observer design techniques, which present strong robustness properties and guaranteed exponential convergence [24, 25]. Numerous recent works have shown successful results in obtaining accurate attitude estimates from noisy and biased measurements using low-cost IMUs [24, 44]. In this work we adopt a non-linear observer formulation to obtain attitude estimates.

3.4 Velocity estimation on-board MAVs

Literature regarding MAV velocity estimation is very vast, and is linked to the type of sensing used on-board. We focus on the approaches applied on MAVs equipped with 2D LiDARs. On one side, directly

differentiating the position estimates is avoided as this provides noisy and inaccurate results [37, 6]. Instead, sensor fusion techniques are employed to achieve high quality results by combining laser estimates and inertial measurements. Stochastic filters, such as EKFs, are predominantly used for this purpose [1, 40, 18], while simpler complementary filters have also provided satisfying results [37]. Other works focus on using a cascade of filters for further noise reduction. Dryanovski et al. [6], 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. [40] propose a cascade of two separate EKFs to achieve accurate results and high rates.

4 Technical background

4.1 Sensor setup

One of the first design challenges with MAVs is choosing the right on-board sensor setup, which is tailored to the specific task at hand. In this section we present our choice for the sensor setup.

4.1.1 2D laser rangefinder.

Since odometric sensors to measure raw displacements aren't available for MAVs, alternative approaches have to be used. In this work we are interested in using laser range measurements from LiDARs for this purpose. However, due to payload limitations only 2D LiDARs can be used [1, 11, 6], and complete 3D pose estimates can't be obtained from the laser range measurements alone. Thus additional sensing has to be used together with sensor fusion techniques to provide reliable 3D pose estimates.

4.1.2 Inertial measurement unit (IMU).

At the heart of MAV platforms one commonly finds inertial measurement units (IMUs) comprised of a three-axis accelerometer, a three-axis rate gyroscope, and a magnetometer [25]. In this work magnetometers aren't used as they are highly sensitive to magnetic interference, and are very unreliable in the proximity of the power lines. We thus only rely on an accelerometer and a gyroscope for inertial measurements.

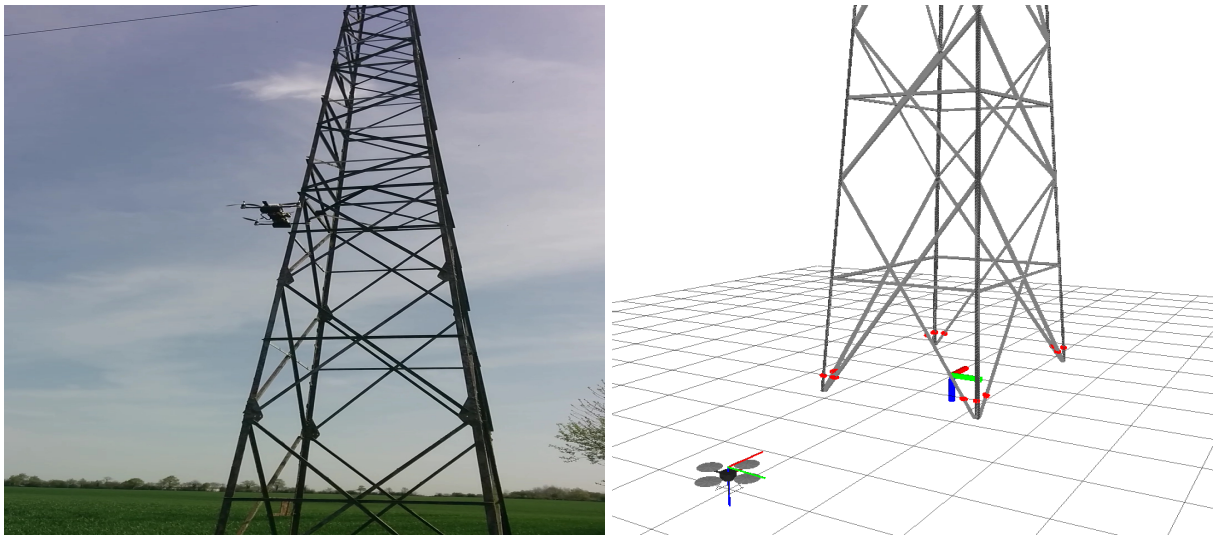
4.1.3 Altitude sensor.

With respect to laser altimeters, barometers allow measuring height without any influence of the ground's layout, and are thus more appropriate for outdoors navigation. We mainly focus on barometers as a source of altitude information. While recent works have obtained impressive results with differential barometry [43, 13], the focus of this work is using on-board sensing only, and differential barometry was not considered.

4.2 Experimental setup

Several experiments were carried out with a quadrotor platform developed at our lab, shown in Fig. 2. This MAV was equipped with a Hokuyo URG-30LX 2D laser scanner mounted horizontally on top and providing measurements at 40 Hz. This sensor was connected to an on-board Odroid-XU computer, where all the laser data acquisition was performed. A Quantec Quanton flight controller card based on an STM32 microcontroller was used to estimate the quadrotor's attitude from measurements obtained from an MPU6000 3 axis accelerometer/gyrometer unit. Lastly, at the time of the acquisitions, the MAV was equipped with an SF10/A laser altimeter from Lightware Optoelectronics, which provides readings at 20 Hz of the distance to the ground along the body-fixed vertical axis. This platform was used towards the beginning of this research to conduct several test flights in front of real electric towers (see Fig. 3). The acquired data was then analysed and served as a basis to the methodology developed in this work. While our final results are mostly based on simulations, and focus on using barometer sensors for altitude estimation, interesting experimental results from these initial test flights will be presented where altitude information was obtained from the laser altimeter.

Figure 2: 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.



(a) (b)

Figure 3: **(a)** Acquiring laser measurements on an electric tower from a 60 kV distribution line, with the quadrotor from Fig. 2. **(b)** The equivalent simulation setup.

4.3 Simulation setup

The approaches proposed in this work were validated in simulations using the Gazebo simulation environment [17] and ROS as an interfacing middleware [33], on a PC with an Intel 3.4 GHz Quad-Core processor and 8 GB of RAM. The Hector quadrotor stack from ROS [29] was used to simulate the quadrotor kinematics and dynamics. Regarding the sensors, the simulated IMU published gyrometer and accelerometer readings at 100 Hz, and the barometer sensor provided measurements at 20 Hz. The 2D laser scanner from Gazebo 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 $2.5m \times 3.5m$ at the ground level, and $1.5m \times 2m$ at a height of $10m$. These dimensions roughly correspond to those of the tower from Fig. 3a. The complete simulation setup is shown in Fig. 3b. All algorithm development was done using C++, and the registration and sample consensus modules from the open source Point Cloud Library (PCL) [4].

4.4 Notation

Let us denote by $\mathcal{I}$ an inertial NED (North-East-Down) frame located at the center of the tower at the ground level. Let $\mathcal{B}$ denote a body-attached frame in the MAV's center of mass. For simplicity, we consider that this frame coincides with the sensor frames. Then, let $\boldsymbol{\xi} = (x, y, z)^\intercal$ denote the position vector of $\mathcal{B}$ with respect to $\mathcal{I}$ (i.e., the position vector of the MAV's center of mass) expressed in $\mathcal{I}$. Next, R denotes the rotation matrix from $\mathcal{B}$ to $\mathcal{I}$. Using the Z-X-Y Euler angle convention with roll ϕ, pitch θ and yaw ψ angles, this rotation matrix is expressed as

$$R(\psi, \phi, \theta) = R_z(\psi)R_x(\phi)R_y(\theta) = \begin{pmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\theta c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{pmatrix}. \tag{1}$$

With this notation, $T_{\mathbf{X}}$ denotes a rigid body transformation parametrized by a vector $\mathbf{X}$, such that

$$T_{\mathbf{X}}(\mathbf{p}) = R(\psi, \phi, \theta)\mathbf{p} + \boldsymbol{\xi}\,, \quad \mathbf{p} \in \mathbb{R}^3, \tag{2}$$

for $\mathbf{X} = (x, y, z, \phi, \theta, \psi)$. This defines the 6 DoF rigid body transformation from $\mathcal{B}$ to $\mathcal{I}$, that transforms the coordinates of a point from $\mathcal{B}$ to $\mathcal{I}$.

We now recall the basic translational dynamics of multirotor aircraft with respect to the inertial frame [25]:

$$\left\{ \begin{array}{rcl} \dot{\boldsymbol{\xi}} & = & \boldsymbol{v} \\ \dot{\boldsymbol{v}} & = & g\mathbf{e_3} + \frac{\boldsymbol{F}}{m} \end{array} \right. \tag{3}$$

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 $\mathcal{I}$. At zero air-velocity, these forces are reduced to the thrust force generated by the propellers. Developing Eq. (3) one obtains

$$\left\{ \begin{array}{l} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = \frac{F_x}{m} \\ \dot{v}_y = \frac{F_y}{m} \\ \dot{v}_z = g + \frac{F_z}{m} \end{array} \right. \tag{4}$$

These equations will be used in our observer formulation to fuse the information from the multiple embarked sensors and to recover velocity estimates.

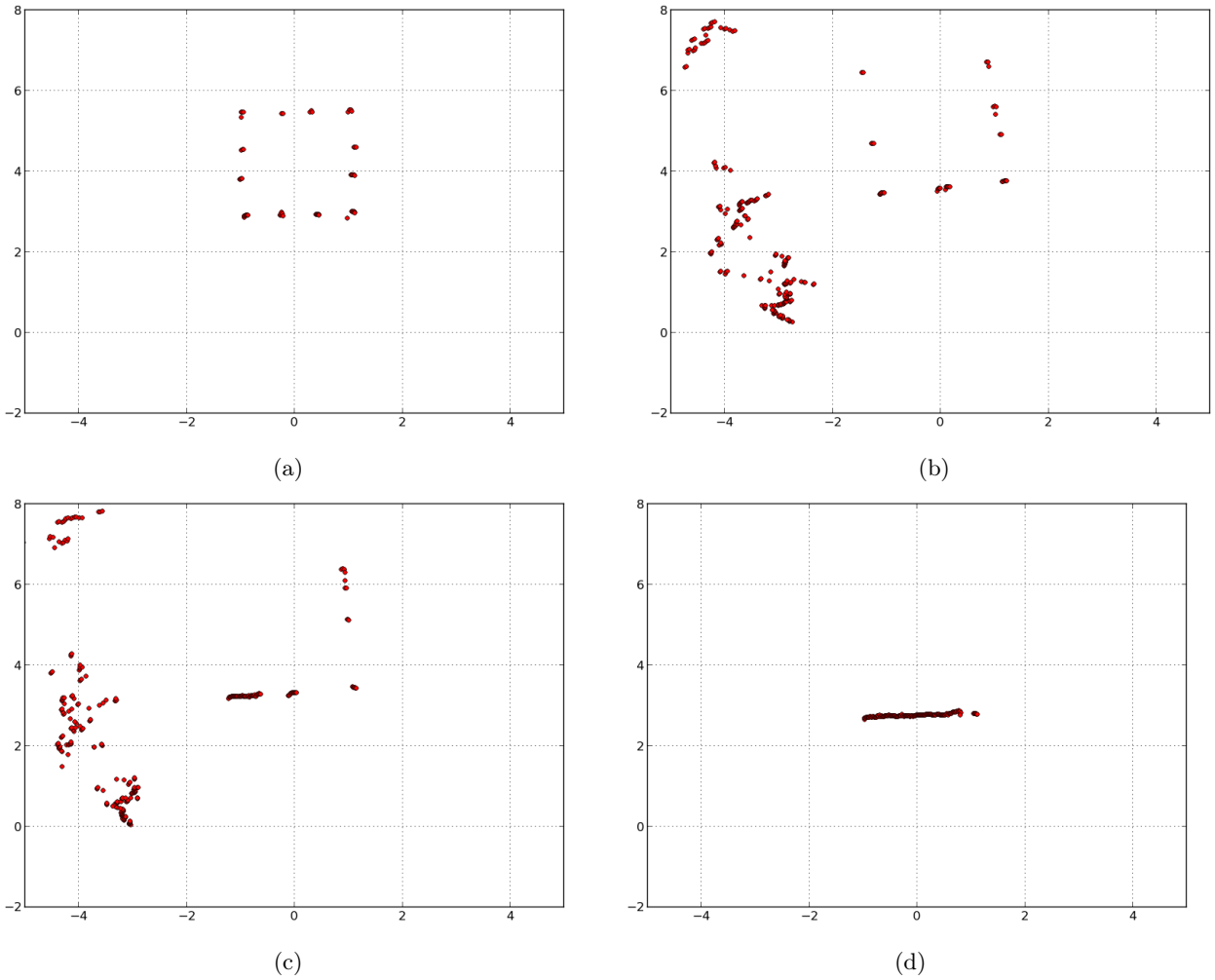


Figure 4: Laser range measurements acquired on the tower from Fig. 3a. In the best case, all sides are visible **(a)**. Occlusions sometimes block the lateral and backsides from view (**b-d**). In the worst case, only the front side is visible (**d**). This happens when horizontal bars on the tower block the lateral and back sides from view.

5 2D local pose estimation

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. Specifically, we explore how basic geometric knowledge of the scene can be exploited for this purpose, without the help of additional sensing. As already mentioned, we focus on the body of electric towers made up of rectangular cross-sections. Measurements taken with a 2D LiDAR on the electric tower from Fig. 3a are shown on Fig. 4, where the portion of the tower can be easily identified. The large open spaces on the surface of the tower allow capturing measurements on all of the tower's faces (Fig. 4a). However, due to occlusions, the entire cross-section isn't always visible (Fig. 4b-4d), and very different scanned structures can be observed. In the worst case scenario (Fig. 4d), 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.

Tracking the tower thus requires accounting for the different cases that can be faced. The idea is to gradually extract notable features from the laser scans, using basic geometric assumptions, to determine the position and

orientation of the tower. 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. The coordinate vectors of the left and right corners of this front line segment, expressed in $\mathcal{B}$, are denoted as $\mathbf{p}_{\text{left}}$ and $\mathbf{p}_{\text{right}}$ respectively. Since $\mathcal{L}_{\text{front}}$ remains visible even in the worst case scenario (Fig. 4d), tracking this line is at the heart of our proposed approach. Then, as the different sides become visible (Fig. 4a-4c), more features are available, such as the side lines $\mathcal{L}_{\text{left}}$ and $\mathcal{L}_{\text{right}}$, which 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 modelled, as it is seldom visible and provides unreliable information. Then, the shape of the contour captured by the laser beams allows establishing a connection between the different features. We consider that this contour is rectangular. However, for this assumption to hold, the scan plane must remain horizontal. This will be discussed in more detail at the end of the section.

We now present a parametrization of the laser scans based on our observations of Fig. 4. Since the goal is to track the cross-sections directly in the laser scans, 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, $\mathcal{C} = \{\mathcal{O}_{\mathcal{C}}, \overrightarrow{\imath}_{\mathcal{C}}, \overrightarrow{\jmath}_{\mathcal{C}}\}$ 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 $\mathcal{F} = \{\mathcal{O}_{\mathcal{F}}, \overrightarrow{\imath}_{\mathcal{F}}, \overrightarrow{\jmath}_{\mathcal{F}}\}$ 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 $\mathcal{C}$. Next, the dimensions of the cross-section are the width and the depth, which are denoted as d_{width} and d_{depth} respectively, which vary considerably with height due to the tower's structure. These dimensions are unknown beforehand, and will be estimated from the laser scans. The complete parametrization is shown in Fig. 5. Note that $\boldsymbol{\xi}_{\mathcal{C}}$ can be determined 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 is important, for example, for a 3D reconstruction of the tower, as will be discussed later. The following subsections describe the different steps implemented to track the cross-sections.

5.1 Scan segmentation

This 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.4b and Fig.4c), 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. Automatic initialization and adapting the outlier rejection radius to the estimated tower dimensions are subject of future work. Next, the remaining laser scan is divided into three subsets of points (expressed in $\mathcal{B}$)

$$\begin{aligned} \mathbf{S}_{\text{front}} &= \{\mathbf{p}_{F,i} = (x_{F,i}, y_{F,i})^{\intercal}, \quad i = 1, ..., N_F\} \\ \mathbf{S}_{\text{left}} &= \{\mathbf{p}_{L,j} = (x_{L,j}, y_{L,j})^{\intercal}, \quad j = 1, ..., N_L\} \\ \mathbf{S}_{\text{right}} &= \{\mathbf{p}_{R,k} = (x_{R,k}, y_{R,k})^{\intercal}, \quad k = 1, ..., N_R\} \end{aligned} \tag{5}$$

which correspond to the front, left and right sides respectively. In the worst case scenario only the front side is visible (Fig. 4d), 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.

5.1.1 Extracting the front side.

The random sample consensus (RANSAC) algorithm [7] was used for this purpose, which is a well-known technique for point cloud segmentation due to its robustness to outliers and noise. This algorithm allows finding instances of $\mathcal{L}_{\text{front}}$ in the laser scans, which was parametrized according to the line equation in its general form

$$\mathcal{L}_{\text{front}} : c_F + n_x x_F + n_y y_F = 0, \quad n_x^2 + n_y^2 = 1. \tag{6}$$

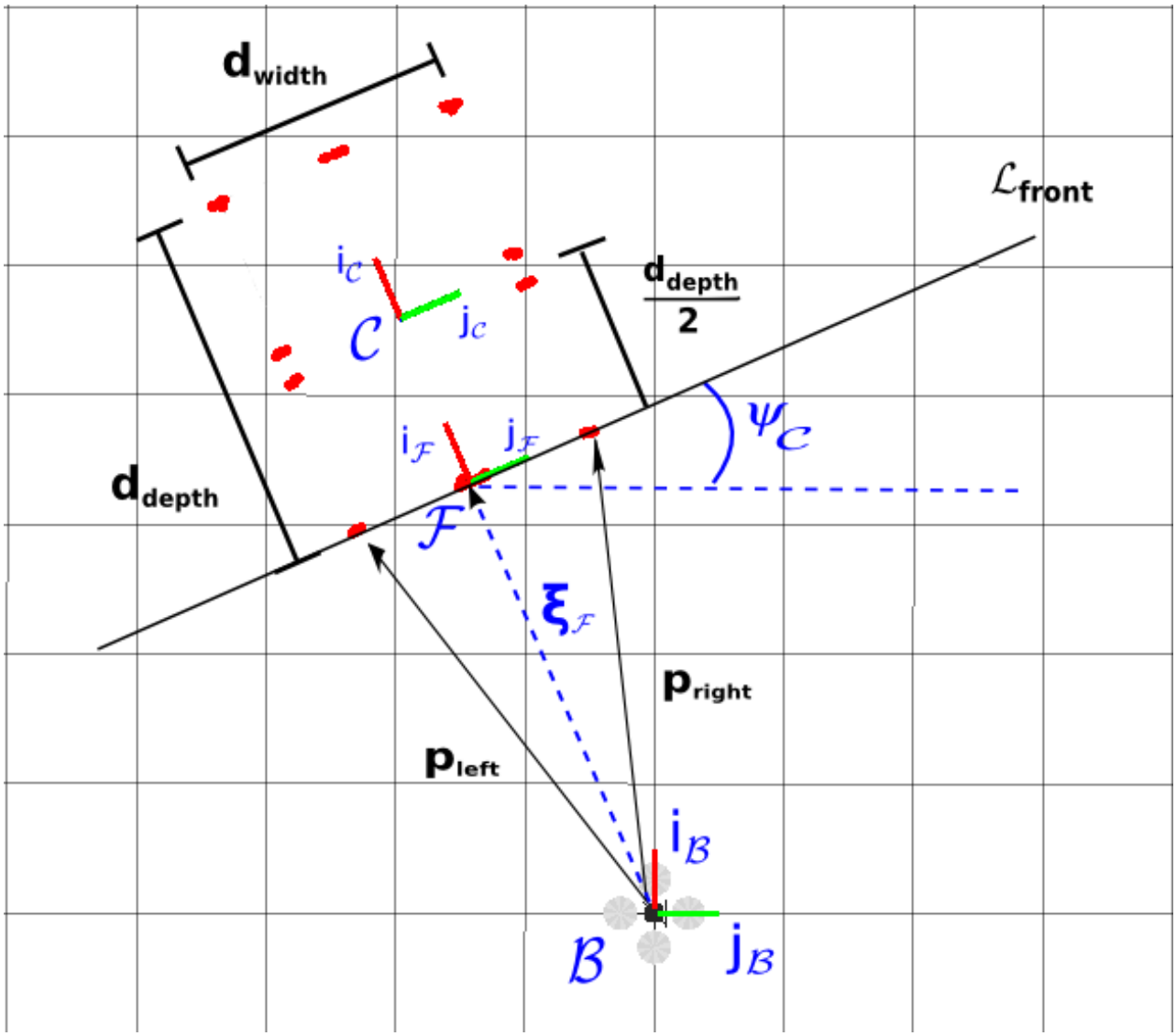


Figure 5: Parametrization of the electric tower's cross-section.

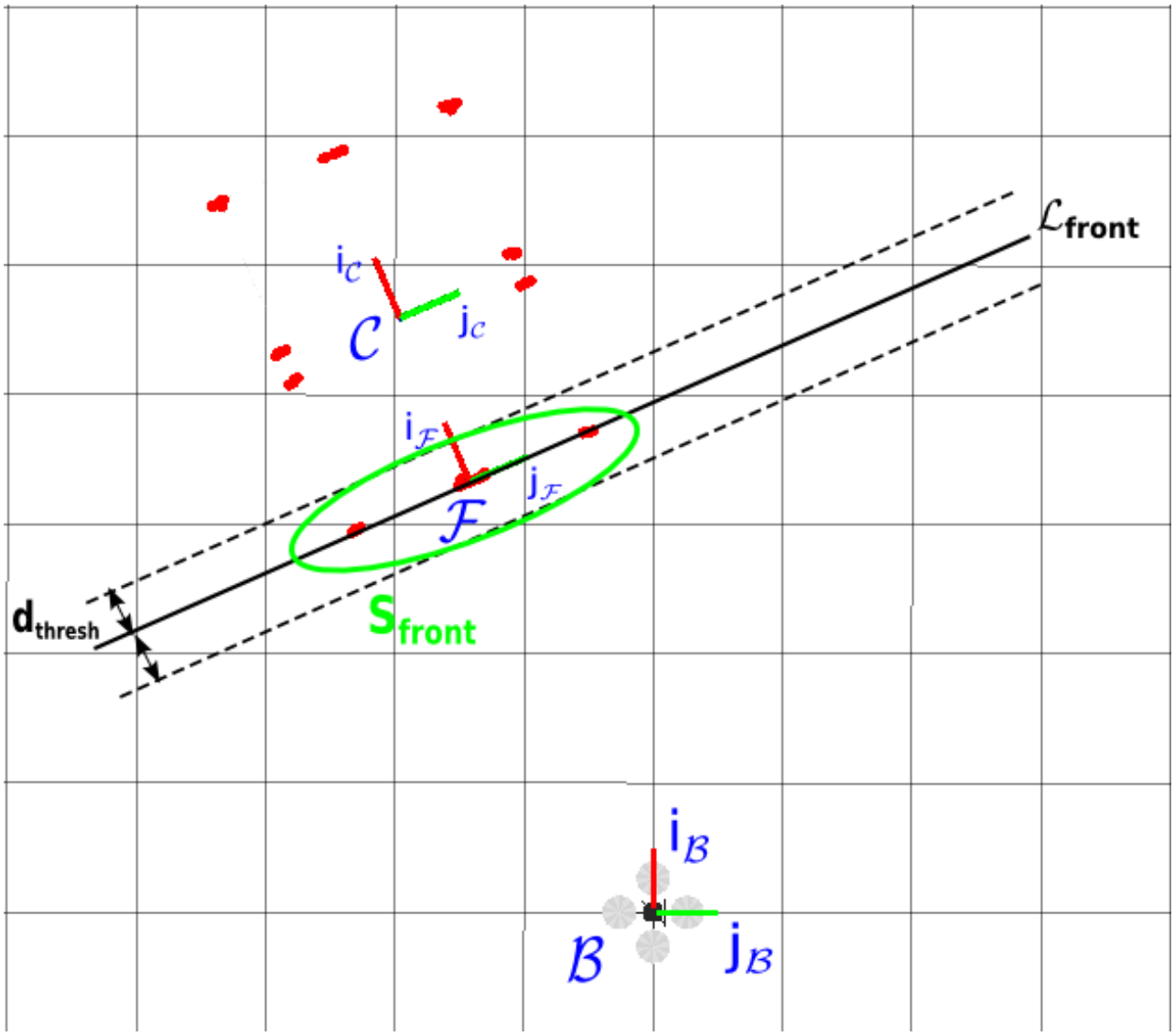


Figure 6: Detecting the front side.

where (n_x, n_y) are the coordinates of the normal vector, expressed in $\mathcal{B}$, and (x_F, y_F) are the coordinates of a point on the front line, also expressed in $\mathcal{B}$. To avoid mistakenly extracting the sidelines, a maximal inclination $\psi_{\max}$ was imposed to the line model, which was determined from the previously extracted front line. For the first scan, it was assumed that there was a rough knowledge of the MAV's orientation with respect to the tower. Then, the RANSAC algorithm considers as inliers all points that fall within a distance threshold d_{thresh} from the model (as shown in Fig. 6), and the subset $\mathbf{S}_{\text{front}}$ is obtained upon convergence. As a result, an initial estimate of the coefficients of $\mathcal{L}_{\text{front}}$ are also obtained.

5.1.2 Extracting the lateral sides.

Next, we determine if the lateral sides are visible in the laser scan. First, 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. Now, a search region can be determined for the left and right sides by tracing a line through each of the front corners, perpendicular to the front line. 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. 7. 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.

5.2 Geometric fitting

The goal is now to find the geometric model that best fits the extracted points. From the previous step, three different situations can arise. First, if no side was detected, the estimation process stops since no useful information is available. Second, if only the front side $\mathbf{S}_{\text{front}}$ was detected, 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. Lastly, if 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 to recalculate $\mathcal{L}_{\text{front}}$ which better fits the data, and to obtain $\mathcal{L}_{\text{left}}$ and $\mathcal{L}_{\text{right}}$. 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, recalling the definition from Eq. (6), their normal vector is $(-n_y, n_x)$ and the cross-section is defined by

$$\begin{cases} \mathcal{L}_{\text{front}}: & c_F + n_x x_F + n_y y_F = 0, \\ \mathcal{L}_{\text{left}}: & c_L - n_y x_L + n_x y_L = 0, \\ \mathcal{L}_{\text{right}}: & c_R - n_y x_R + n_x y_R = 0, \\ & n_x^2 + n_y^2 = 1. \end{cases} \tag{7}$$

Then, evaluating the extracted point sets $\mathbf{S}_{\text{front}}$, $\mathbf{S}_{\text{left}}$ and $\mathbf{S}_{\text{right}}$ from Eq. (5) with their respective line in Eq. (7), and expressing in matrix form, one obtains

$$\begin{pmatrix} 1 & 0 & 0 & x_{F,1} & y_{F,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & x_{F,N_F} & y_{F,N_F} \\ 0 & 1 & 0 & y_{L,1} & -x_{L,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & y_{L,N_L} & -x_{L,N_L} \\ 0 & 0 & 1 & y_{R,1} & -x_{R,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & y_{R,N_R} & -x_{R,N_R} \end{pmatrix} \begin{pmatrix} c_F \\ c_L \\ c_R \\ n_x \\ n_y \end{pmatrix} = \mathbf{r}. \tag{8}$$

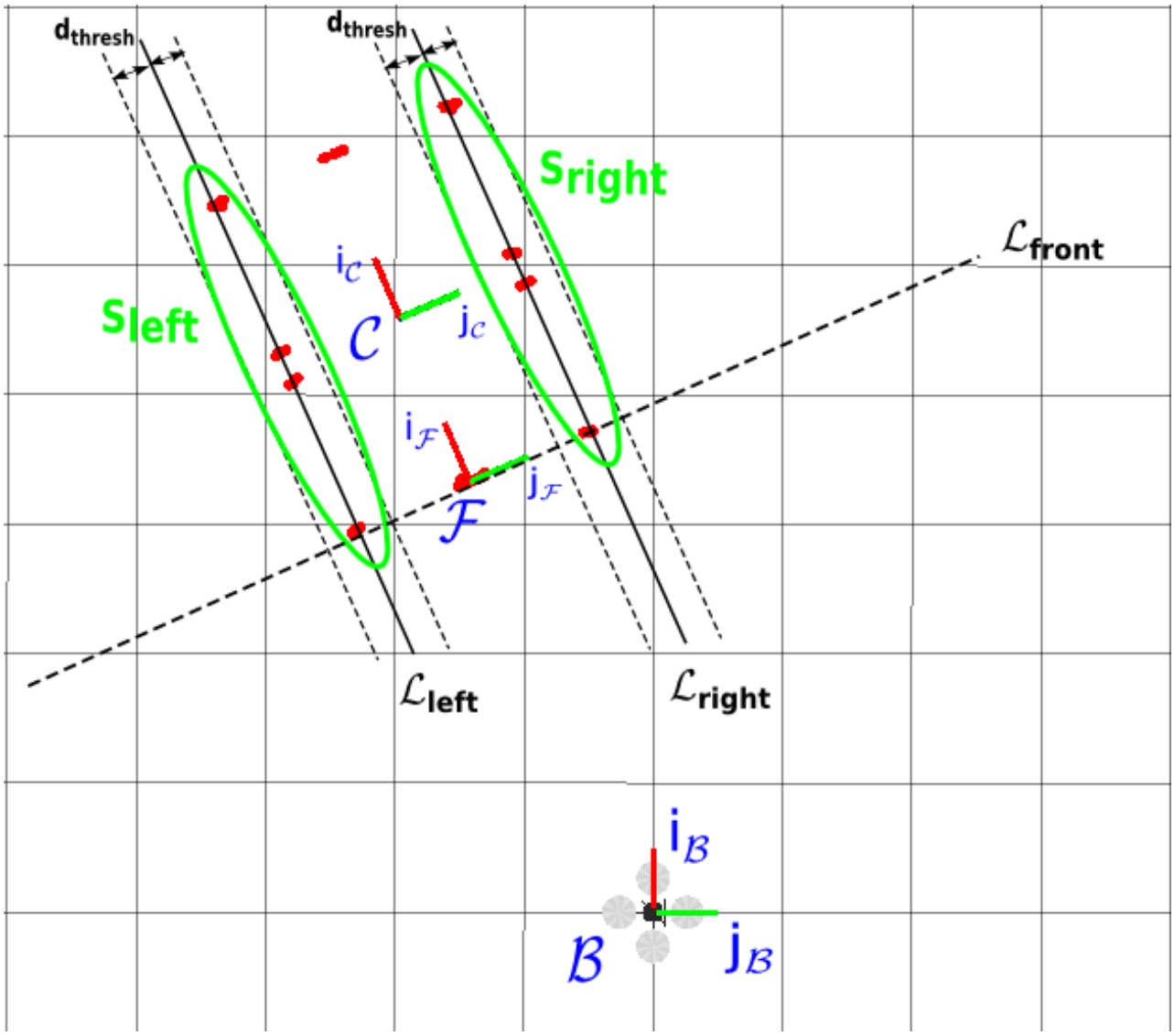


Figure 7: Detecting the left and right sides.

where $\mathbf{r} = (r_1, \cdots, r_N)^\intercal$, with $N = N_F + N_L + N_R$, are the residuals, and $|r_i|$ corresponds to the distance from a point to the line. The geometric fitting problem is formulated as finding the coefficients of Eq. (7) for which the sum of squared distances is minimal. That is

$$\min \|\boldsymbol{r}\|^2 = \min \sum_{i=1}^{N} r_i^2, \quad \text{subject to Eq. (8) and } n_x^2 + n_y^2 = 1, \tag{9}$$

which is a constrained least squares problem, with non-linear constraint $n_x^2 + n_y^2 = 1$ to guarantee solution uniqueness. This is solved numerically following the procedure proposed in [9]. The end result is an estimate of the parameters of Eq. (7). 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.

5.3 Calculating the position and orientation

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

$$\psi_{\mathcal{C}} = \text{arctan2}(n_y, n_x). \tag{10}$$

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

$$\boldsymbol{\xi}_{\mathcal{F}} = \frac{\mathbf{p}_{\text{right}} + \mathbf{p}_{\text{left}}}{2}. \tag{11}$$

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

$$\boldsymbol{\xi}_{\mathcal{C}} = \boldsymbol{\xi}_{\mathcal{F}} + \frac{d_{\text{depth}}}{2} \begin{pmatrix} \cos\psi_{\mathcal{C}} \\ \sin\psi_{\mathcal{C}} \end{pmatrix}. \tag{12}$$

It is important to highlight that the visible cross-section can change drastically from one scan to the other, as was shown in Fig. 4. This in return can produce 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.

5.4 Limitations

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. However, external disturbances, such as strong winds, can produce large inclinations and bring the MAV to a configuration where the geometric model from Eq. (7) is no longer valid. Under such circumstances, tracking the tower with this approach will result inaccurate.

Another underlying constraint is that the MAV must always fly on the same side of the tower. This occurs because 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. This will cause shifts in the position and orientation estimates, since they are defined with respect to $\mathcal{L}_{\text{front}}$ (Eq. (10) and Eq. (11)).

5.5 Simulation results

Simulations were carried out using the setup from Fig. 3b to evaluate the performance of the proposed tracking algorithm. The initial position of the tower's center with respect to the MAV was given, and the outlier rejection

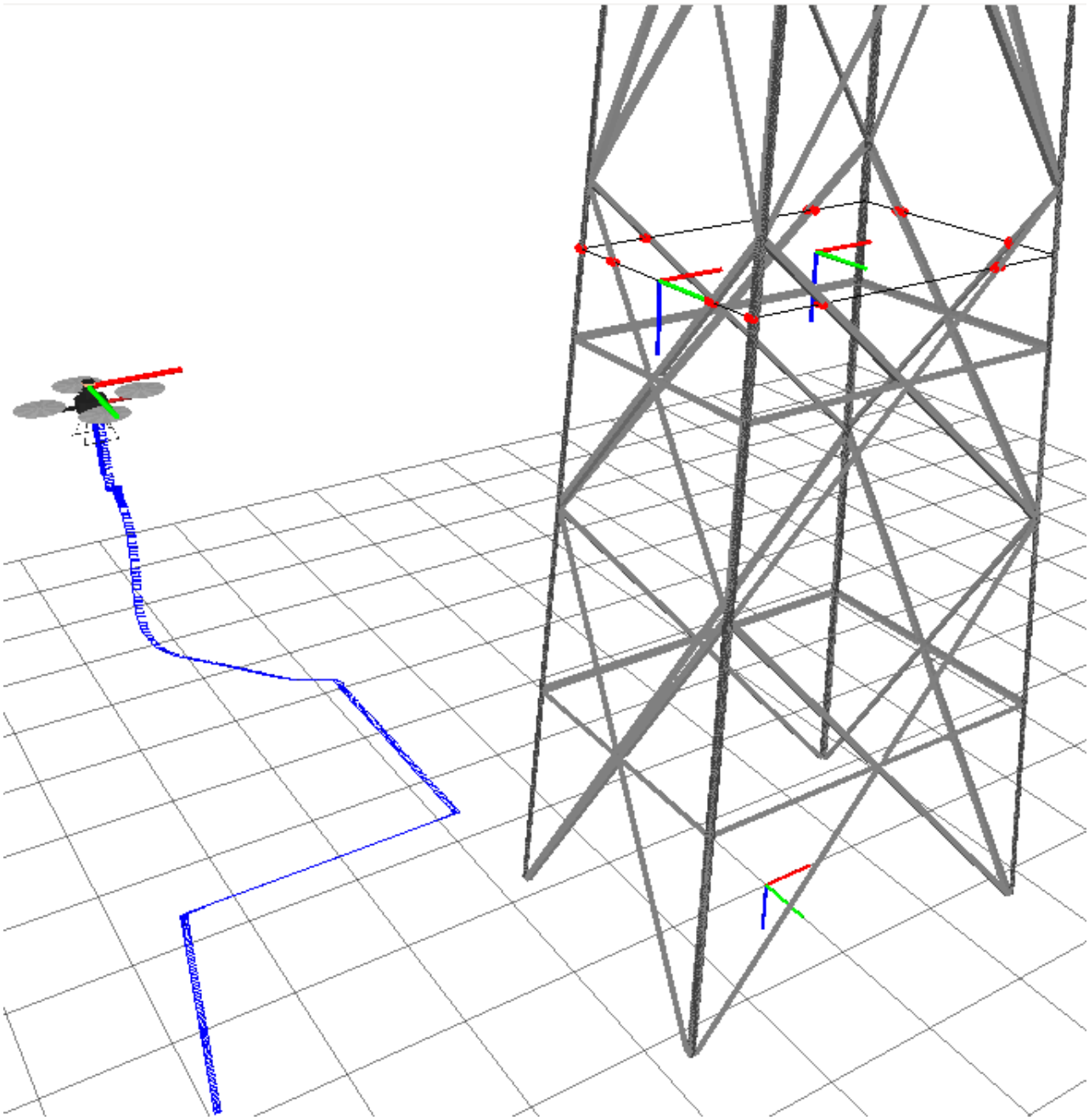
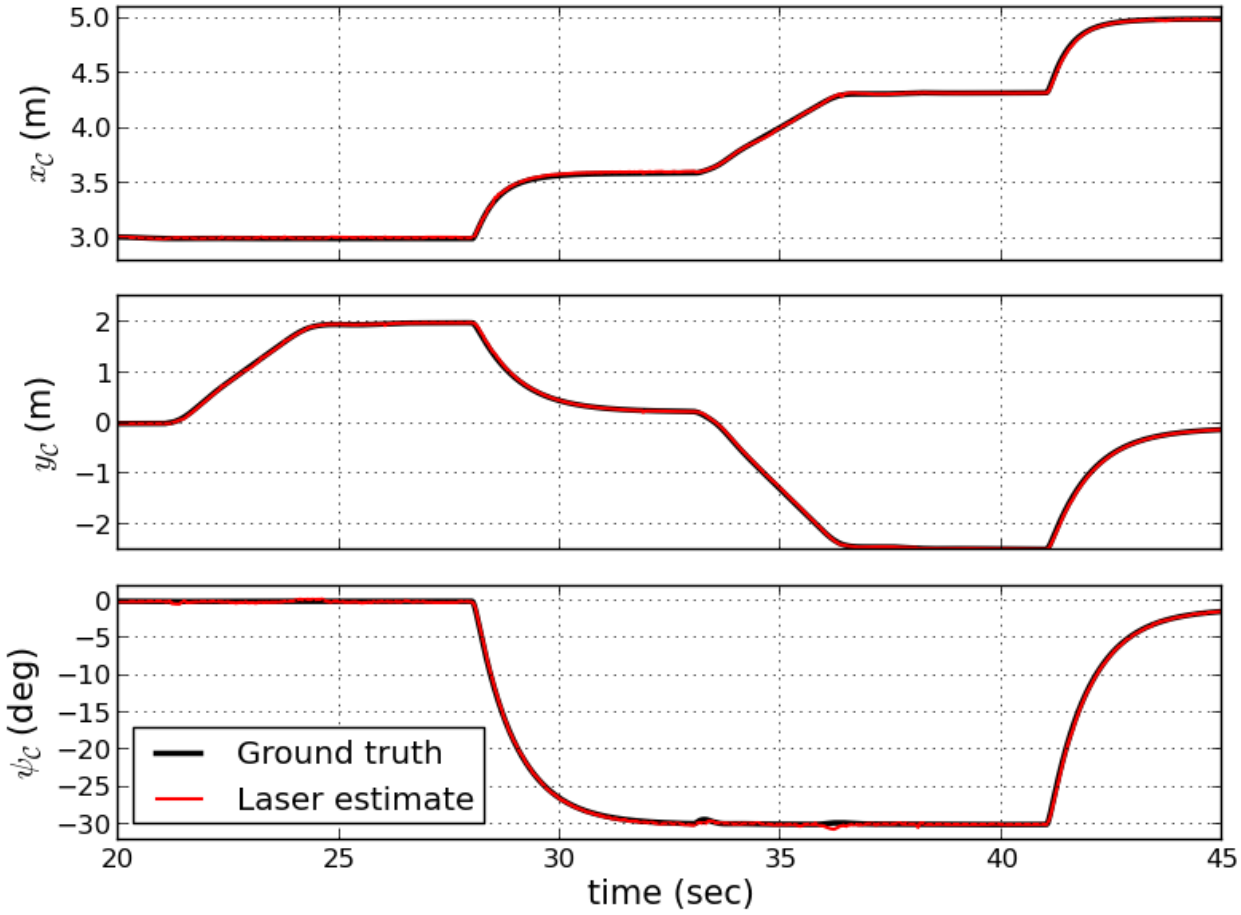
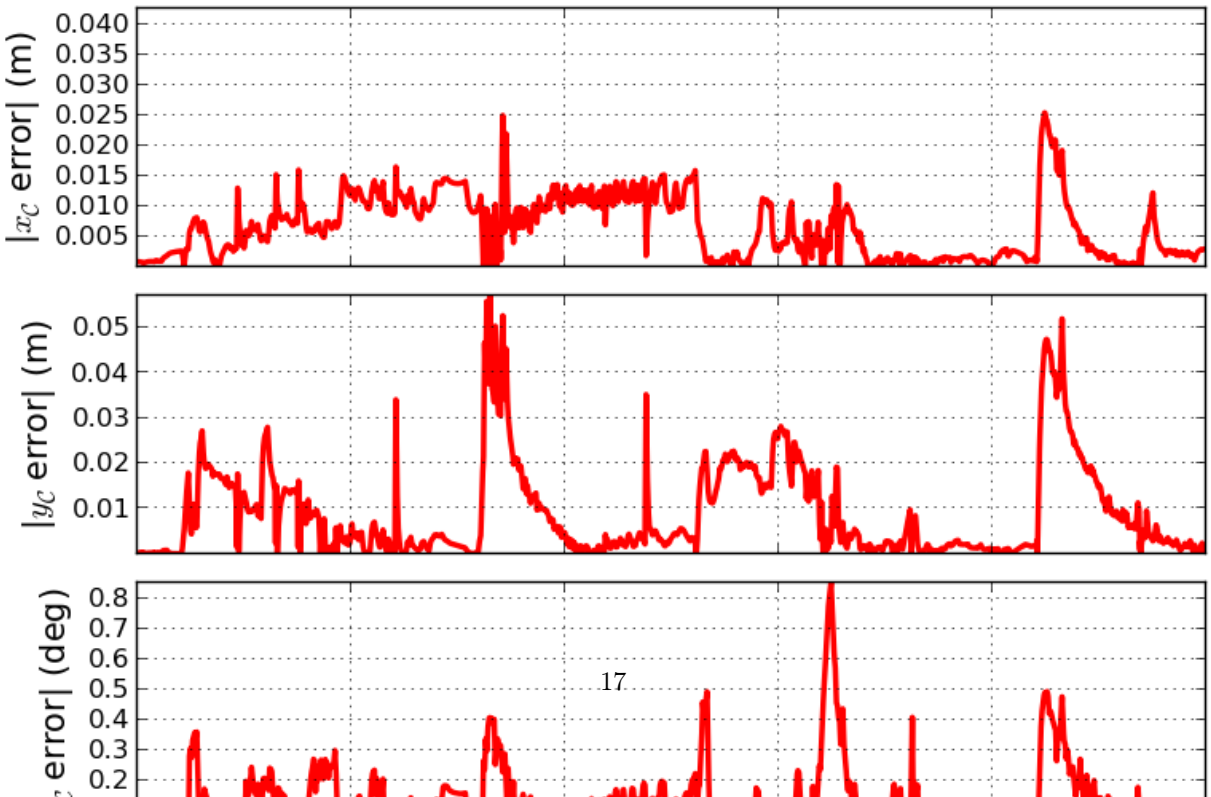


Figure 8: The simulated flight in front of the electric tower. The blue line indicates the trajectory followed by the quadrotor. An example of a tracked cross-section is visible on the right.


$x_\mathcal{C}$ (m)
$y_\mathcal{C}$ (m)
$\psi_\mathcal{C}$ (deg)
Ground truth
Laser estimate
time (sec)

(a)


$|x_\mathcal{C}$ error$|$ (m)
$|y_\mathcal{C}$ error$|$ (m)
error$|$ (deg)

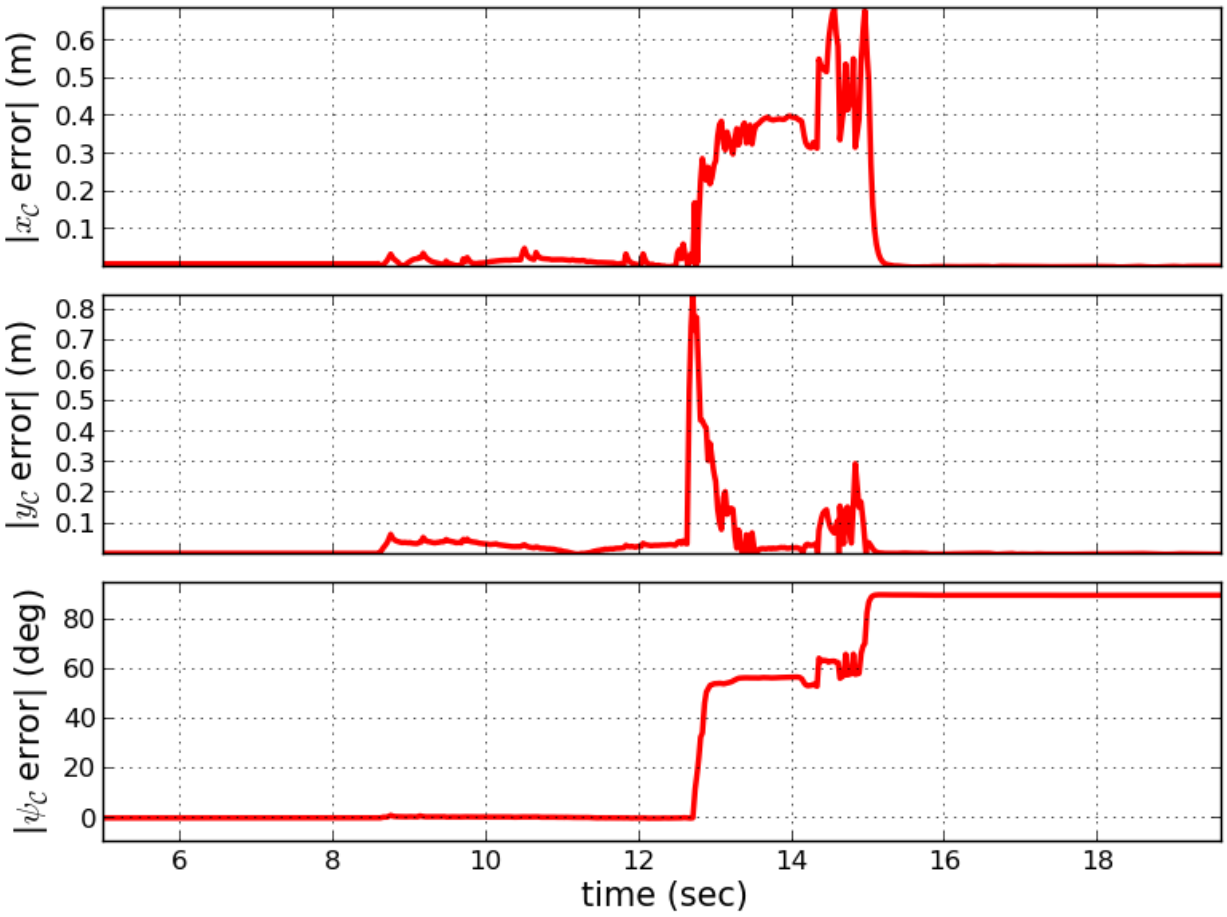


Figure 10: An example of the tracking method failing for a flight around the electric tower. Failure occurs at $t = 13s$.

radius (as discussed in the segmentation section) was set to 4 meters. The parameters for the RANSAC scan segmentation were chosen as $d_{\text{thresh}} = 5\text{cm}$ and $\psi_{\max} = 10°$. In the first test, the MAV was flown in front of one side of the tower for different heights and distances from the tower as shown in Fig. 8. This figure also illustrates an example of a tracked cross-section with its corresponding front and center frames. The resulting position and orientation estimates are compared to the simulation ground truth in Fig. 9a, for a portion of the flight. As can be seen, throughout this flight the proposed approach is capable of effectively tracking the tower's center. This is further verified from the absolute estimation errors, shown in Fig. 9b, which remain below 5cm for the translation components, and below 1° for the yaw angle.

In a second test, the MAV was flown around the tower, and the results are shown in Fig. 10. In this case, the algorithm clearly fails at $t = 13s$. This happens when the MAV transitions from one side of the tower to the other and the algorithm then starts tracking a different front line. This causes the 90° error in the orientation as seen in Fig. 10. While the algorithm can track the center of the tower again ($t = 15s$ as the position errors drop), the orientation error isn't corrected. This illustrates one of the main limitations of the proposed approach.

5.6 Experimental results

The proposed tracking algorithm was also tested on data previously acquired from several manual test flights, where the MAV from Fig. 2 was flown vertically in front of an electric tower, as shown in Fig. 3a. An initial rough guess of the tower's center with respect to the MAV was given, and the outlier rejection radius was set to 4 meters. As already mentioned, besides the 2D LiDAR, the MAV was additionally equipped with a laser altimeter and an IMU. Unfortunately, at the time of the acquisitions no GPS sensor was 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. 11 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. 11.

5.6.1 Modelling the electric tower.

A by-product of tracking the cross-section's center 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. 11, and the final result is shown in Fig. 12. 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 tower's body from the estimated dimensions presented in Fig. 11. A simple approach is to approximate each face as a planar segment [20], and the edges of the tower as the intersection of adjacent planes m_j ($j = 1, ..., 4$), expressed as

$$m_j : \quad a_j x + b_j y + c_j z + d_j = 0, \quad j = 1, \ldots, 4, \tag{13}$$

where each m_j is associated with a face of the tower. Obtaining the planes' coefficients results straight-forward from Fig. 11, as the slope of the fitted lines are directly related to the slopes of the planes. For example, for this particular case this resulted in

$$\begin{cases} m_1 : & -x - 0.062z - 1.643 = 0 \\ m_2 : & y - 0.046z - 1.265 = 0 \\ m_3 : & x - 0.062z - 1.643 = 0 \\ m_4 : & -y - 0.046z - 1.265 = 0 \end{cases} \tag{14}$$

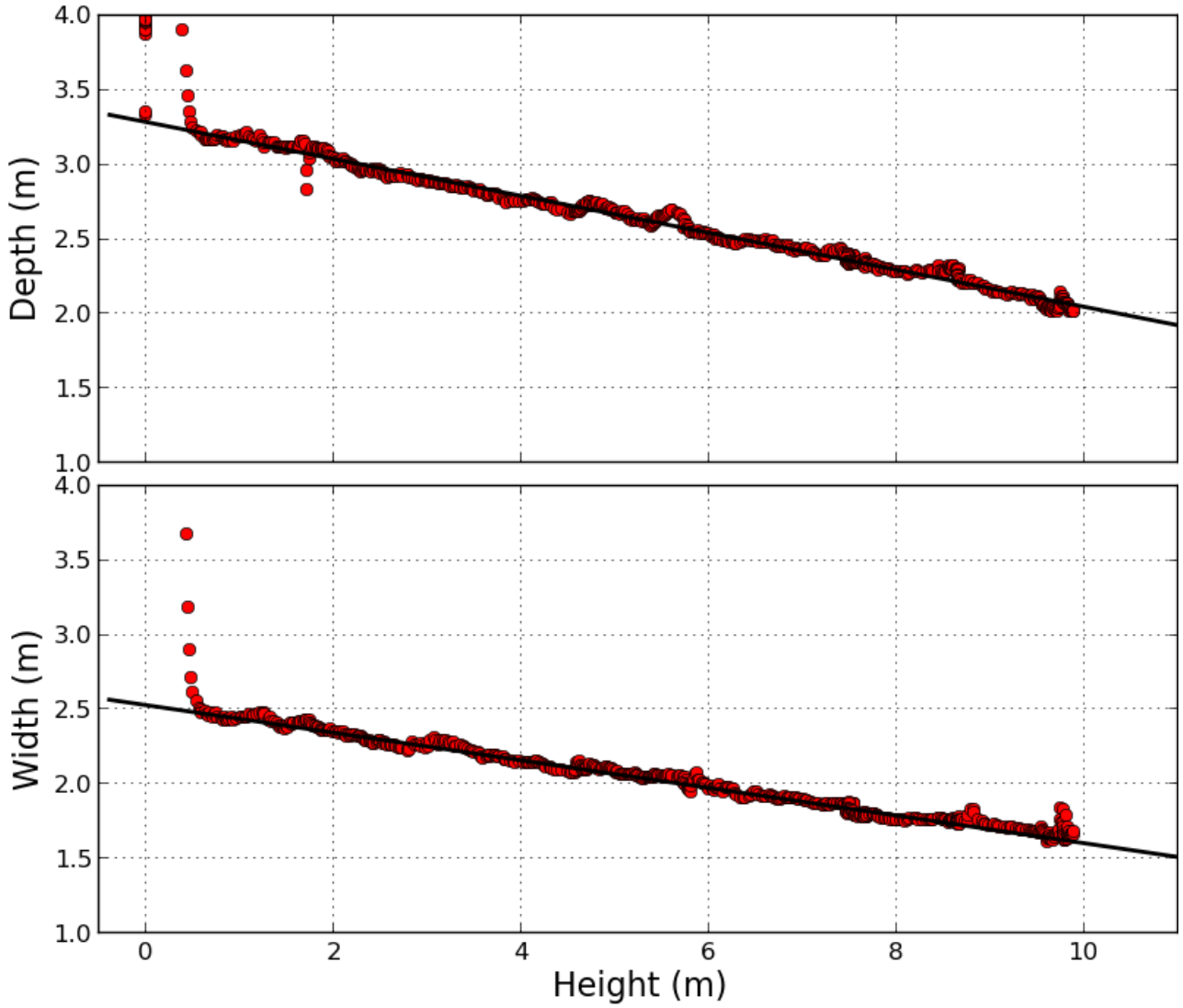


Figure 11: The estimated depth and width as a function of the height for the electric tower from Fig. 3a, fitted with straight lines.

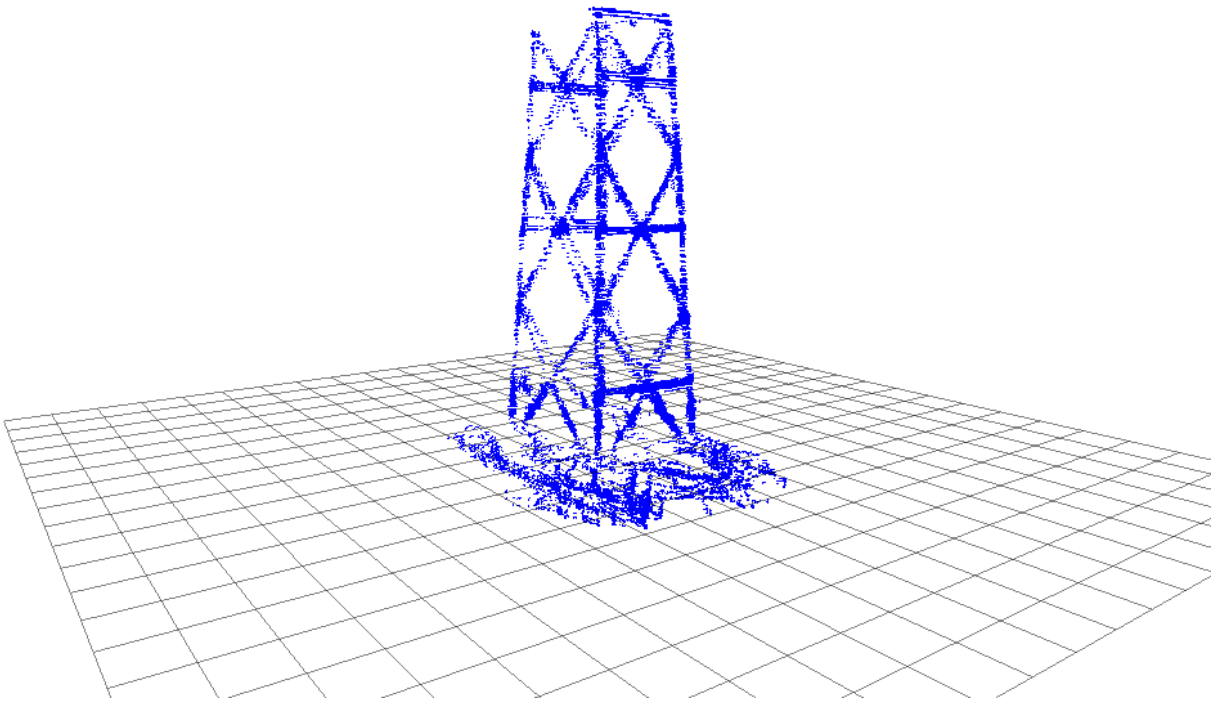


Figure 12: Partial 3D point cloud reconstruction of the electric tower from Fig. 3a, for a vertical flight in front of the tower. The laser scans are aligned using the tracked cross-section's center, the quadrotor's altitude (from the laser altimeter) and attitude (from the IMU measurements).

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

5.7 Discussion

Since the final goal is to achieve autonomous navigation capabilities, all of the MAV's 6 DoF must be determined. For this purpose, this proposed tracking approach could be complemented with additional sensing to recover complete 3D pose estimates, for example, using inertial measurements to estimate the roll and pitch angles, and an altitude sensor, such as a laser altimeter or a barometer. However, 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. An alternative strategy is thus 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 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. 11 and Fig. 12). A second step would then focus on 3D pose estimation and navigation, using the estimated model to track the tower in general flight conditions. With such a model-based approach to recover pose estimates, the scan plane no longer needs to remain horizontal and less restrictions are imposed on the MAV's movement. For the following sections, we 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.

6 3D local pose estimation

In this section, we present how to obtain complete 3D pose estimates with our sensor setup. As is typically done with MAVs, the estimation process is broken down into several components [6, 40]. Recalling that the complete 6 DoF pose from $\mathcal{B}$ to the inertial frame $\mathcal{I}$ is described by $\{x, y, z, \phi, \theta, \psi\}$, the 3D pose is reconstructed as follows: $\{x, y, \psi\}$ are estimated from the laser range measurements; then, as will be discussed, $\{z\}$ is estimated from the laser range measurements fused with the barometer measurements; finally, $\{\phi, \theta\}$ are obtained by fusing accelerometer and gyrometer measurements from the IMU. The following subsections explain each component of the estimation process.

We first explore how the classic ICP algorithm that has been successful indoors can be extended to the case of an electric tower inspection. This technique requires the surrounding environment to have sufficient geometric detail and is not suitable for highly unstructured scenarios often faced outdoors [6]. However, in an outdoors 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 the previous section to retrieve 2D pose estimates, and will now be used to adapt the ICP algorithm. While common implementations focus on aligning pairs of scans to retrieve pose information in 2D, we instead treat the problem in 3D by introducing previous knowledge of the tower's geometry in the registration process. We now present two possible implementations of the ICP algorithm.

6.1 Adapting the ICP algorithm: First proposed approach

In this first approach we follow a line of work typically adopted with the ICP algorithm in navigation tasks, consisting in aligning point clouds. The idea is to maintain the approach as general as possible, as no specific parametrization of the scene is required and pose information is recovered directly from the point correspondences. Let the *current* scan be represented by a set of 2D points, denoted $\mathbf{S}_p = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{N_p}\}$. For simplicity, consider that $\mathbf{S}_p$ is expressed in the body attached frame $\mathcal{B}$. Then, let $\mathbf{S}_q = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{N_q}\}$

denote the 3D *reference* set, expressed in the inertial frame $\mathcal{I}$, which corresponds to a 3D point cloud reconstruction of the inspection scene, assumed to be acquired beforehand, e.g., from our tracking approach as discussed in the previous section. The goal is to find the rigid body transformation that best aligns $\mathbf{S}_p$ to $\mathbf{S}_q$. The baseline ICP [2] 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 2D points $\mathbf{p}_i \in \mathbf{S}_p$ into 3D coordinates in the inertial frame $\mathcal{I}$, obtaining $\mathbf{S}_{p'}$. For the first iteration, the parameter vector is $\mathbf{X}_0 = (x_{\text{laser}}, y_{\text{laser}}, z_{\text{laser}}, \phi_{\text{imu}}, \theta_{\text{imu}}, \psi_{\text{laser}})$, such that $\{x_{\text{laser}}, y_{\text{laser}}, z_{\text{laser}}, \psi_{\text{laser}}\}$ are obtained from the scan registration for the previous laser scan and $\{\phi_{\text{imu}}, \theta_{\text{imu}}\}$ from the IMU attitude estimation (as will be explained briefly).

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 [2]. To speed up the matching process we make use of K-D trees, as is commonly done with ICP [2, 36].

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 [36], which in this case are typically due to surrounding vegetation.

4. **Minimization:** The goal is to find the transformation $T_{\mathbf{X}_{\text{min}}}$ that minimizes the sum of squared errors, using the Euclidean distance as the distance metric [2]. For the N remaining point pairs $(\mathbf{p}'_i, \mathbf{q}_i)$, this leads to the following optimization problem:

$$\mathbf{X}_{\text{min}} = \arg\min_{\mathbf{X}} \sum_{i=1}^{N} \|T_{\mathbf{x}}(\mathbf{p}'_{\mathbf{i}}) - \mathbf{q}_{\mathbf{i}}\|^2, \quad \text{such that } (\phi, \theta) = (0, 0), \tag{15}$$

   which is solved with the Levenberg-Marquardt algorithm, since it allows to obtain accurate results and deal with initialization errors without significant speed losses [8]. The components ϕ and θ of $\mathbf{X}$ are neglected during the minimization, since ϕ_{imu} and θ_{imu} used at the initialization are precise and reliable. This reduces the optimization problem from a 6-dimensional space to a 4-dimensional space, which further limits the risk of divergence due to local minima, and provides a more reliable solution. This is the main modification of the algorithm.

5. Finally, the current estimate is updated as

$$T_{\mathbf{X}_{k+1}} = T_{\mathbf{X}_{\text{min}}} \cdot T_{\mathbf{X}_k} \tag{16}$$

   Due to the previous step, $T_{\mathbf{X}_{\text{min}}}$ only updates the $\{x, y, z, \psi\}$ components of $T_{\mathbf{X}_k}$ in each iteration.

The end result of the scan registration process is an estimation of the 3D translation vector $(x_{\text{laser}}, y_{\text{laser}}, z_{\text{laser}})$ and the yaw angle ψ_{laser}. The main novelty is thus that altitude estimates can now be recovered, which is a direct consequence of introducing a 3D point cloud reconstruction of the tower in the registration process.

6.1.1 Limitations.

Besides the drawbacks inherent to the ICP algorithm discussed at the beginning of this article, other limitations can be pointed out. Evidently, this approach is restricted to sections of the tower captured in the 3D point cloud reconstruction. Pose estimates can't be recovered in previously unexplored or occluded sections. 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 the previous section this requires exploring extensive portions of the tower. Other existing solutions rely on offline processing of data from powerful and expensive

3D LiDARs capable of capturing dense measurements from long distances [20, 12]. This, however, goes beyond the scope of this work.

Further complication arise regarding the altitude estimates. 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 instead determined from the point correspondences with the 3D point cloud uniquely. The altitude estimates are thus more unreliable and prone to errors, as will be seen in the simulation results. Furthermore, 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. These drawbacks justify the use of an additional barometer sensor. However, as will be seen, this proposed ICP implementation 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.

6.2 Adapting the ICP algorithm: Second proposed approach

The difficulties in obtaining an accurate 3D point cloud reconstruction of the inspection scene can render the previous approach impractical. Nonetheless, 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 [2]. Therefore, an alternative is to align the laser scans onto the simplified planar representation of the tower body from Eq. (13), which is simpler to obtain than a complete point cloud reconstruction, as previously discussed. To achieve this, we adopt a projection-based matching strategy [3, 32], where, after initialization, 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, as will be seen, allows obtaining significant speed gains [36].

Thus, in this approach, the matching step (step 2) 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. (13)), 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, the output is $(x_{\text{laser}}, y_{\text{laser}}, z_{\text{laser}}, \psi_{\text{laser}})$.

6.2.1 Limitations.

One of the main drawbacks of this formulation is that it applies specifically to the case of rectangular cross-sections. The projection strategy 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. As before, no altitude information can be recovered if the faces of the tower are completely vertical.

6.3 Altitude estimation

The altitude estimates obtained previously from the laser range measurements have a strong dependence on the shape of the tower and can result unreliable. In contrast, barometer measurements are independent from the shape of surrounding structures, but suffer from drift over time due to varying atmospheric conditions. Better results can be obtained by combining both sources of altitude information. Recalling the vertical dynamics in $\mathcal{I}$ from Eq. (4),

$$\begin{cases} \dot{z} &= v_z \\ \dot{v}_z &= g + \frac{F_z}{m} \end{cases} \tag{17}$$

Accurate vertical velocity estimates can be obtained by fusing the barometer and IMU measurements [13, 43], and are thus obtained separately, as will be addressed in a later discussion. Therefore, in this section we consider that v_z is a known input, and instead use the following system

$$\begin{cases} \dot{z} &= v_z \\ \dot{b}_z &= 0, \end{cases} \tag{18}$$

where b_z is the unknown barometer drift, which is modelled as a constant as it varies slowly in time, and is defined by the relationship $z_{\text{baro}} = z + b_z$, with z_{baro} denoting the barometer measurement. This leads to a simple second order feedback observer formulation

$$\begin{cases} \dot{\hat{z}} &= v_z - k_z(\hat{z} - \overline{z}_1) \\ \dot{\hat{b}}_z &= -k_{b_z}(\hat{z} - \overline{z}_2) \end{cases} \tag{19}$$

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

$$\overline{z}_n = \lambda_n(z_{\text{baro}} - \hat{b}_z) + (1 - \lambda_n)z_{\text{laser}} \quad \text{with} \quad 0 \leq \lambda_n \leq 1,\ n = 1, 2. \tag{20}$$

which is the weighted sum of the laser altitude estimates z_{laser} and barometer readings z_{baro} compensated for bias. 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. The stability analysis for this observer and details on how to tune the gains (k_z, k_{b_z}) are given in the Appendix.

6.4 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

$$\boldsymbol{\gamma} = R^\intercal \boldsymbol{e_3} \tag{21}$$

with $\boldsymbol{e_3} = (0, 0, 1)^\intercal$. From the rotation matrix definition of Eq. (1), it follows that $\boldsymbol{\gamma}$ contains implicitly the MAV's roll and pitch angles, since

$$\begin{aligned} \phi &= \arcsin(\gamma_2) \\ \theta &= \text{atan2}(-\gamma_1, \gamma_3). \end{aligned} \tag{22}$$

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

$$\dot{R} = RS(\boldsymbol{\omega}), \tag{23}$$

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. (21) and Eq. (23), and results in

$$\dot{\boldsymbol{\gamma}} = \boldsymbol{\gamma} \times \boldsymbol{\omega}. \tag{24}$$

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]

$$\mathbf{a}_m = R^\intercal(\dot{\boldsymbol{v}} - g\boldsymbol{e_3}) = R^\intercal \dot{\boldsymbol{v}} - g\boldsymbol{\gamma}. \tag{25}$$

Then, under the assumption of negligible linear acceleration, one has [26]

$$\mathbf{a}_m \approx -g\boldsymbol{\gamma}, \tag{26}$$

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

$$\dot{\hat{\boldsymbol{\gamma}}} = \hat{\boldsymbol{\gamma}} \times \left(\boldsymbol{\omega}_m - k_\gamma(\mathbf{a}_m \times \hat{\boldsymbol{\gamma}})\right), \quad k_\gamma > 0 \tag{27}$$

with $\boldsymbol{\omega}_m$ the angular velocities measured by the gyrometer in $\mathcal{B}$.

To analyse the stability of this estimator, consider the candidate Lyapunov function $L = 1 - \boldsymbol{\gamma}^T\hat{\boldsymbol{\gamma}}$. From Eq. (26) and Eq. (27) one has

$$\dot{\hat{\boldsymbol{\gamma}}} \approx \hat{\boldsymbol{\gamma}} \times (\boldsymbol{\omega}_m - k_\gamma g(\hat{\boldsymbol{\gamma}} \times \boldsymbol{\gamma})). \tag{28}$$

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 $\boldsymbol{\gamma}$.

6.4.1 Gain scheduling.

The approximation from Eq.(26) is commonly used in attitude estimation when dealing with accelerometers [26], but only holds when flying at constant velocity or near stationary flight conditions. 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.(26) 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 [46], and can provide short-term rotations accurately [51].

A basic strategy is thus to detect highly accelerated states by comparing the magnitude of the accelerometer readings to the gravity acceleration [34, 48, 46]. Let $\tilde{a}_m$ denote the absolute accelerometer measurement error with respect to gravity as

$$\tilde{a}_m = |\|\boldsymbol{a}_m\| - g|, \quad g = 9.81 \ \frac{m}{s^2}. \tag{29}$$

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. [48] 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. [46] set a nominal gain for hovering state, which is then decreased linearly during transitions to high acceleration states. We adopt a strategy similar to [46]. Let k_L and k_H denote the nominal gains during low and high acceleration states respectively, the idea is to transition smoothly between these gains. The following gain scheduling approach is proposed

$$k_\gamma(\tilde{a}_m) = k_L e^{-\alpha\tilde{a}_m} + k_H(1 - e^{-\alpha\tilde{a}_m}), \quad \alpha > 0, \tag{30}$$

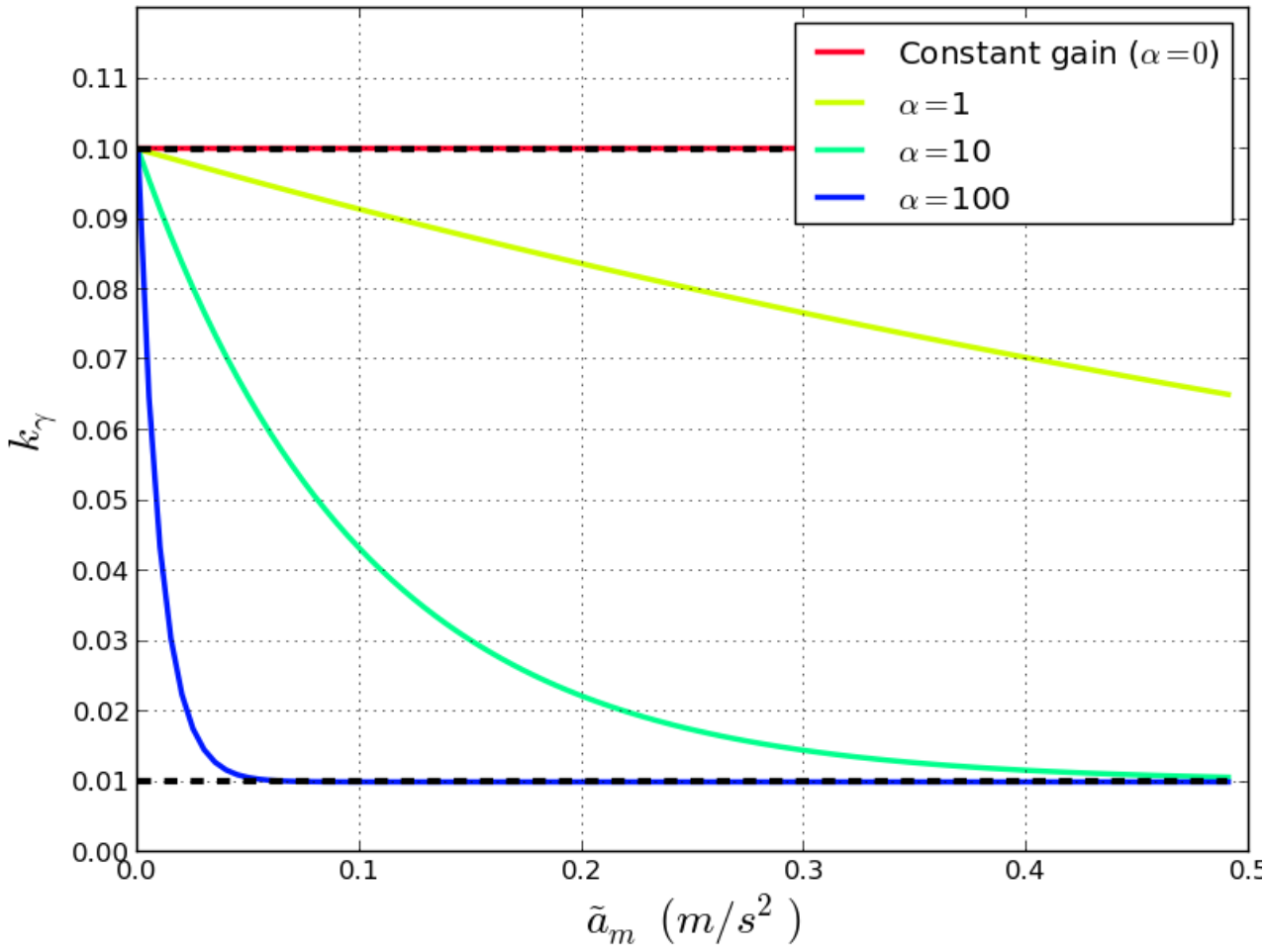


Figure 13: An example of the attitude observer gains according to Eq. (30), for different values of α.

where α is an arbitrary positive constant that determines the steepness of the transitions between k_L and k_H. It is simple to verify that 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. This is further illustrated on Fig. 13 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.

6.4.2 Complete rotation matrix reconstruction.

The estimated roll ϕ_{imu} and pitch θ_{imu} angles are recovered from $\hat{\boldsymbol{\gamma}}$ and Eq. (22) as

$$\begin{aligned} \phi_{\text{imu}} &= \arcsin\left(\hat{\gamma}_2\right) \\ \theta_{\text{imu}} &= \operatorname{atan2}\left(-\hat{\gamma}_1, \hat{\gamma}_3\right). \end{aligned} \tag{31}$$

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

$$\hat{R} = R_z(\psi_{\text{laser}}) R_x(\phi_{\text{imu}}) R_y(\theta_{\text{imu}}). \tag{32}$$

This matrix is subsequently used at each initialization of the laser scan registration, and for the velocity estimation described in the following section.

7 Velocity estimation

In the previous section, the complete 6 DoF pose of the MAV was determined from the sensor measurements. The goal is now to derive velocity estimates by combining the pose estimates with the inertial measurements from the IMU. For this purpose, we make use of the translational dynamics of the MAV with respect to the inertial frame $\mathcal{I}$ from Eq. (3) to formulate velocity observers. In the following analysis, the external aerodynamic forces $\boldsymbol{F} = (F_x, F_y, F_z)^\intercal$ from Eq. (4) are determined from the accelerometer readings $\boldsymbol{a}_m$ and the estimated attitude $\hat{R}$ as

$$\boldsymbol{F} = m\hat{R}\boldsymbol{a}_m. \tag{33}$$

Since different sensors are used for the different states, the horizontal and vertical velocity components are analysed separately.

7.1 Horizontal velocity estimation

From Eq. (4) it follows that the dynamics for $\{x, y\}$ in $\mathcal{I}$ are two independent second-order systems. Estimating the horizontal velocities results straightforward, and is achieved with simple feedback state observers defined as

$$\begin{cases} \dot{\hat{x}} = \hat{v}_x - k_x(\hat{x} - x_{\text{laser}}) \\ \dot{\hat{v}}_x = \dfrac{F_x}{m} - k_{v_x}(\hat{x} - x_{\text{laser}}), \quad k_x, k_{v_x} > 0 \end{cases}$$

$$\begin{cases} \dot{\hat{y}} = \hat{v}_y - k_y(\hat{y} - y_{\text{laser}}) \\ \dot{\hat{v}}_y = \dfrac{F_y}{m} - k_{v_y}(\hat{y} - y_{\text{laser}}), \quad k_y, k_{v_y} > 0 \end{cases} \tag{34}$$

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, and $(x_{\text{laser}}, y_{\text{laser}})$ are the estimates obtained from the laser scan registration described in the previous section.

7.2 Vertical velocity estimation

As previously mentioned, satisfying estimates of the vertical velocity can be recovered from barometer and accelerometer measurements [13, 43]. As will be seen, these estimates remain accurate even in the presence of barometer drift. Recalling the vertical dynamics from Eq. (17), we now formulate the following feedback state observer

$$\begin{cases} \dot{\hat{z}} = \hat{v}_z - k_z(\hat{z} - z_{\text{baro}}) \\ \dot{\hat{v}}_z = g + \dfrac{F_z}{m} - k_{v_z}(\hat{z} - z_{\text{baro}}), \quad k_z, k_{v_z} > 0 \end{cases} \tag{35}$$

where (k_z, k_{v_z}) are the observer gains, and z_{baro} are the barometer altitude measurements. The altitude estimates from the laser scan registration are not included as they only degrade the performance. The vertical velocity estimates $\hat{v}_z$ are subsequently used as an input to the altitude observer from Eq. (19).

8 Simulation results: 3D local pose estimation

The purpose of this section is to assess the performance of the different components of the pose estimation process. The results presented here were obtained from simulated flights carried out using the previously discussed simulation setup, illustrated in Fig. 3b. For the flights, a set of waypoints was given for the quadrotor to follow, accounting for a complete displacement around the tower. 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. 14.

8.1 Attitude estimation results

The attitude observer from Eq. (27) 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 this observer throughout the flight. Fig. 15a illustrates the deviations of the accelerometer readings from the acceleration of gravity ($\tilde{a}_m$ from Eq. (29)) for a portion of the flight. As can be seen, the MAV spends larger amounts of time in low acceleration states ($\tilde{a}_m \approx 0$). Then, the peaks correspond to instants when the MAV accelerates towards a different waypoint. The idea is to adapt the observer to these peaks by lowering the estimation gain k_γ. This was carried out with the gain scheduling approach from Eq. (30). Based on results observed in the simulations, the nominal gains were set to $k_L = 0.1$ and $k_H = 0.01$. Moreover, the estimation process was repeated for different values of the parameter α, from $\alpha = 0$, which corresponds to the constant gain case since $k_\gamma = k_L$ (from Eq. (30)), to $\alpha = 100$. As explained, this parameter determines the steepness of the transitions between the two nominal gains. The resulting scheduled gains for a portion of the flight are shown on Fig. 15b. When comparing the two figures, it can be noted the gain k_γ rapidly drops in the presence of acceleration peaks (e.g., $t = 26s$ and $t = 30s$), which is the desired behaviour. However, as α increases, the gains can result overly sensitive to small changes in $\tilde{a}_m$ ($t = 38s$ for $\alpha = 100$).

Next, the absolute estimation errors with respect to the simulation ground truth are shown on Fig. 16a and Fig. 16b for the roll and pitch angles respectively. When comparing Fig. 16a and Fig. 16b to Fig. 16c, it can be noted that the observer can accurately trace the roll and pitch angles in low acceleration states (when $\tilde{a}_m \approx 0$) for all cases, and the errors for the most part remain below 1°. On the other hand, the largest estimation errors correspond to peaks in $\tilde{a}_m$ (e.g., $t = 26s$ and $t = 40s$), reaching a maximum for the constant gain observer of 2.45° for the roll angle, and 2.62° for the pitch angle. In contrast, as the parameter α is increased, error peaks related to $\tilde{a}_m$ are now largely suppressed and the overall performance is improved with the simple gain scheduling strategy. Based on these results, a gain scheduled attitude estimation with $\alpha = 10$ was used for the following sections, as it offers a good trade-off between sensibility to changes in $\tilde{a}_m$ and estimation error reduction.

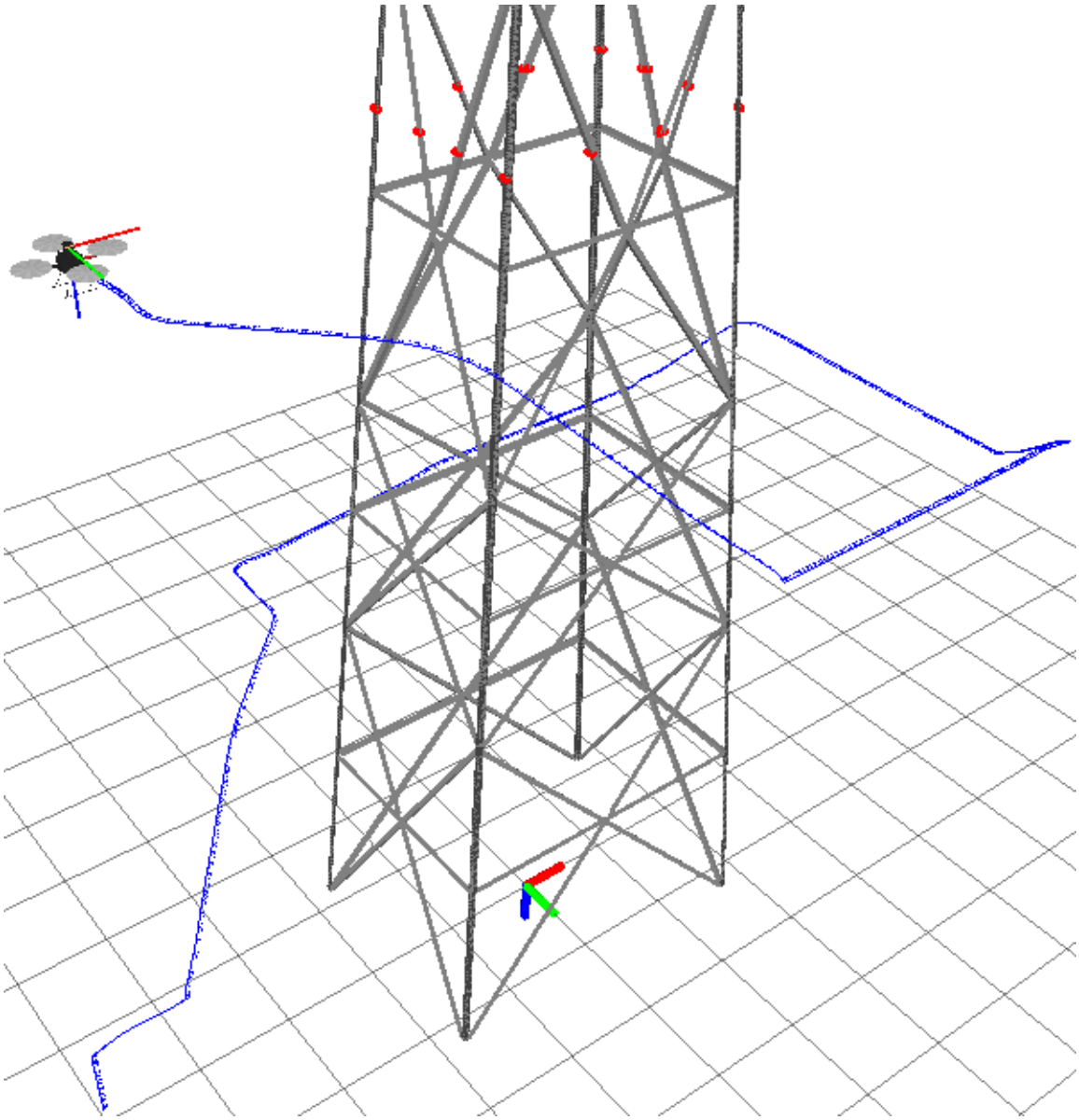
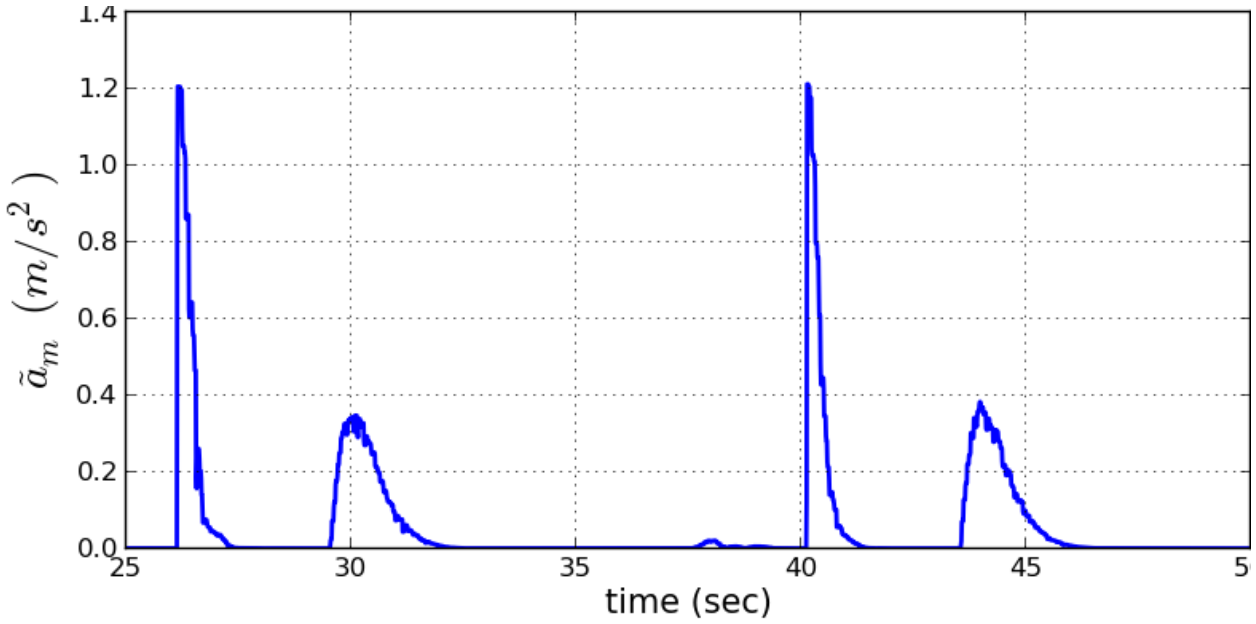
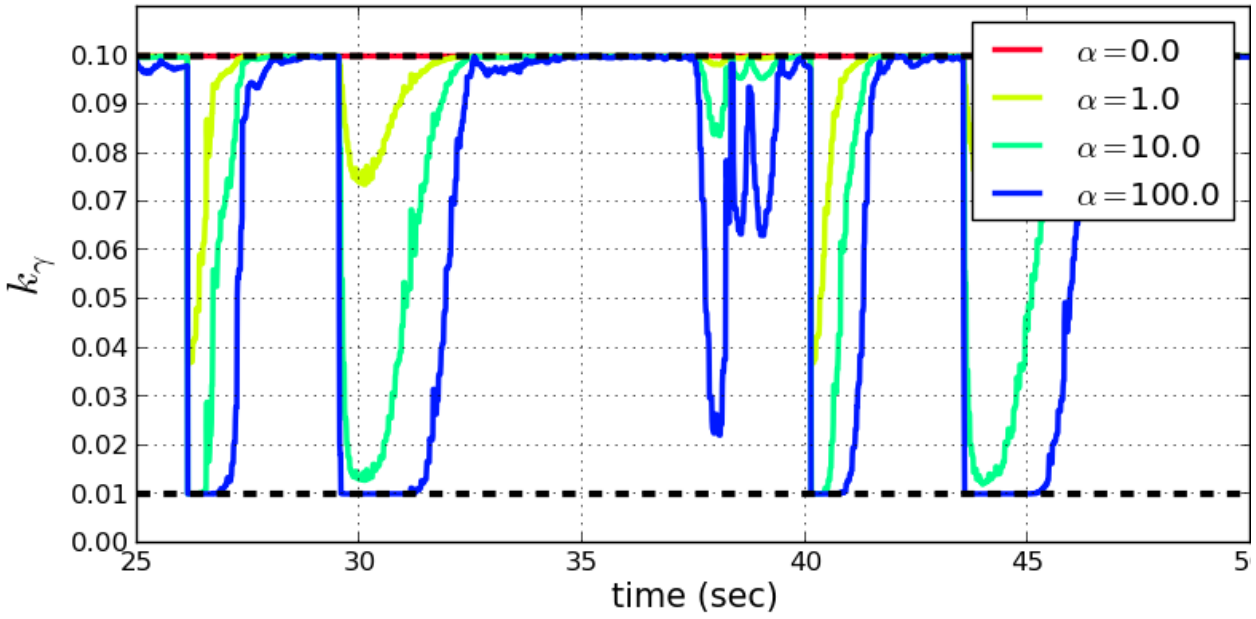


Figure 14: The simulated flight around the tower. The blue line indicates the trajectory. Throughout the flight the quadrotor was oriented towards the center of the tower.

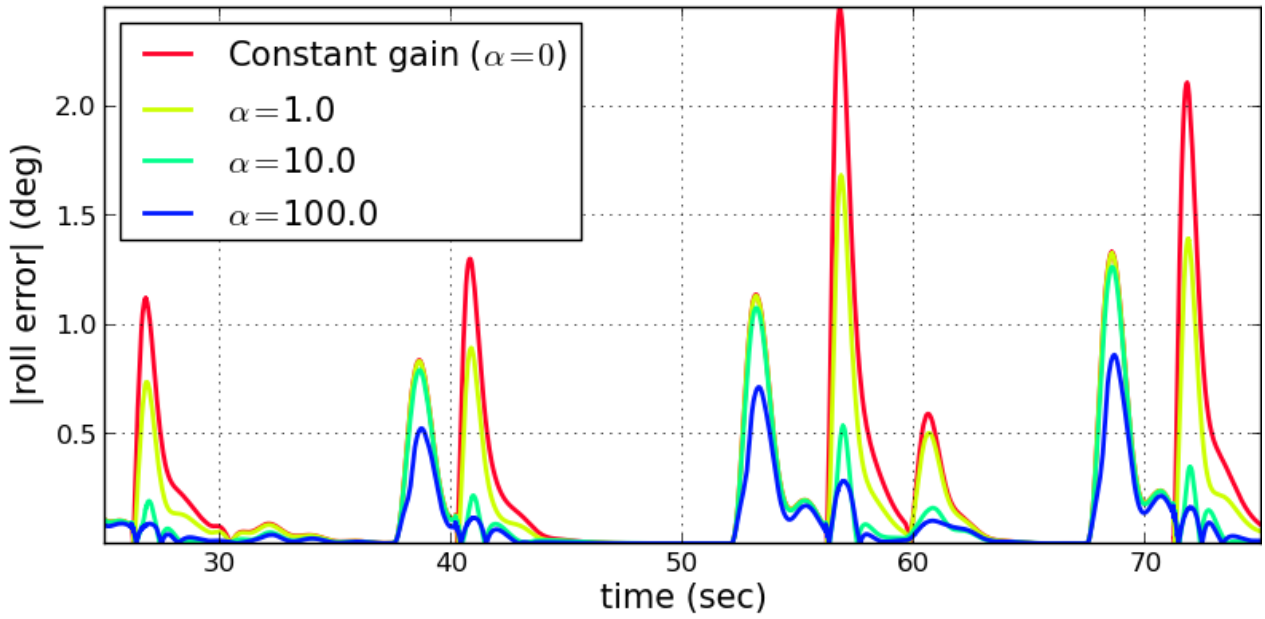


(a)

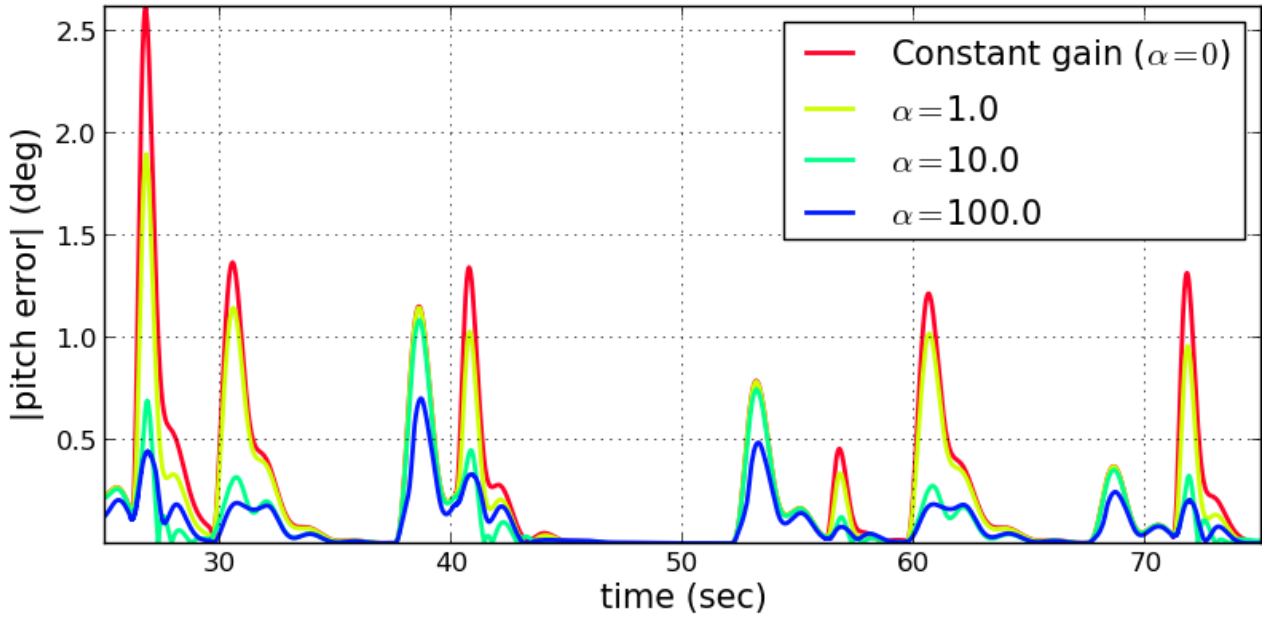


(b)

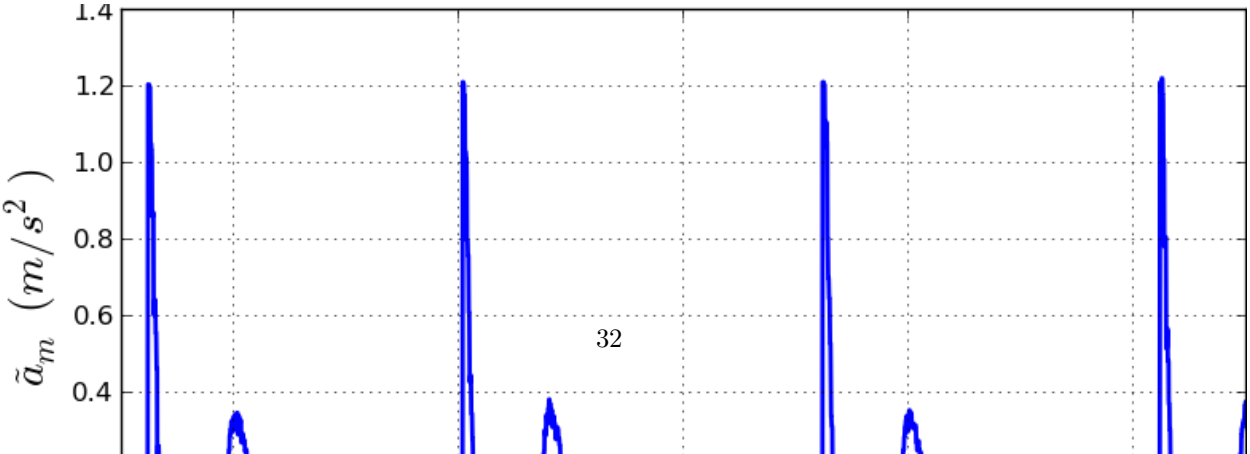
Figure 15: For a portion of the flight from Fig. 14: **(a)** The deviations of the accelerometer readings from gravity according to Eq. (29). **(b)** The resulting scheduled gains for different values of α. The gains become more reactive for larger values of α.

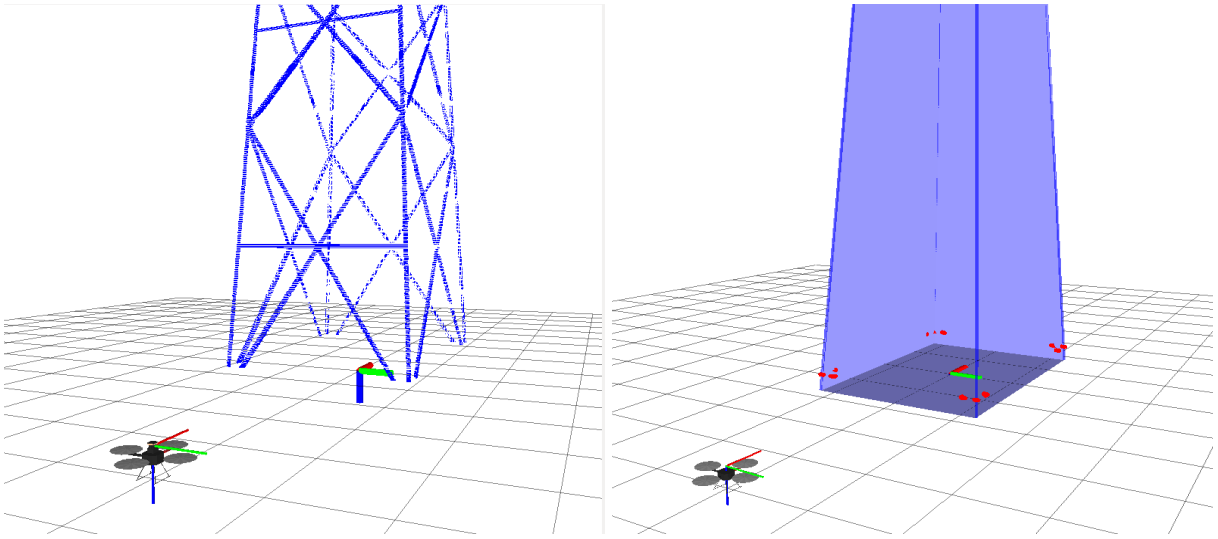

Constant gain ($\alpha=0$)
$\alpha=1.0$
$\alpha=10.0$
$\alpha=100.0$
|roll error| (deg)
time (sec)

(a)


Constant gain ($\alpha=0$)
$\alpha=1.0$
$\alpha=10.0$
$\alpha=100.0$
|pitch error| (deg)
time (sec)

(b)


$\tilde{a}_m \ (m/s^2)$



(a) (b)

Figure 17: The two models used as reference in the ICP algorithm to align the laser scans: **(a)** Point cloud reconstruction. **(b)** Planar model.

8.2 Laser-based local pose estimation 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 shown in Fig. 17a, which was obtained beforehand from the simulated tower shown in Fig. 3b, following the same procedure for Fig. 12 based on our tracking approach. In the second case, the laser scans were instead aligned to the planar representation of the tower illustrated in Fig. 17b, which was also obtained beforehand following the procedure used for Fig. 11 and Eq. (14), relying on our tracking approach. Here, the estimated coefficients from Eq. (7) resulted in

$$\begin{cases} m_1: & -x - 0.076z - 1.749 = 0 \\ m_2: & y - 0.046z - 1.219 = 0 \\ m_3: & x - 0.076z - 1.749 = 0 \\ m_4: & -y - 0.046z - 1.219 = 0 \end{cases} \tag{36}$$

which correspond to the front, right, back and left sides respectively. In both ICP implementations, an initial rough knowledge of the MAV's position with respect to the tower was given for the first scan registration. For each subsequent laser scan, the estimation process was initialized with the results from the previous scan registration and attitude estimates from the IMU measurements.

We now analyse the performance of the two approaches. First, the computation time required for the scan registration in both cases is shown in Fig. 18. As expected, using the planar model results in significantly faster estimates with an average of 1.4ms, compared to the point cloud case average of 16ms. This shows the effectiveness of the projection-based matching strategy used to establish point correspondences, which avoids the computationally extensive correspondence search required for the point cloud registration.

Next, Fig. 19 compares the MAV's ground truth position with the estimates from both approaches. 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. 20. For the planar model case, the $\{x, y\}$ errors remain below 5cm (Fig. 20a and Fig. 20b). Furthermore, the yaw estimates are also very precise, with a maximum error of 0.8° (Fig. 20d). In these figures it can be noted that the point cloud approach achieves

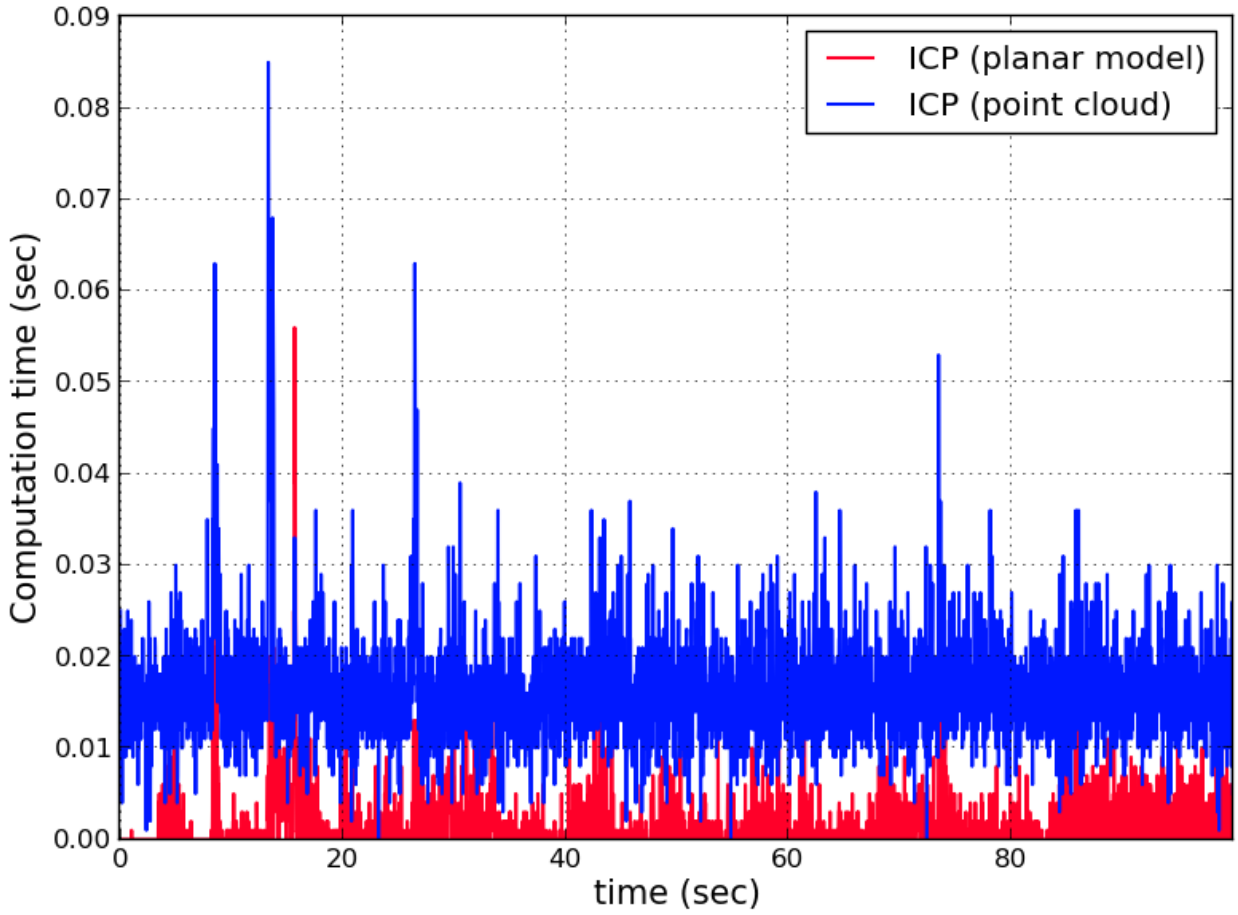


Figure 18: Comparing the computation time for the laser scan registrations. The planar model approach is approximately ten times faster.

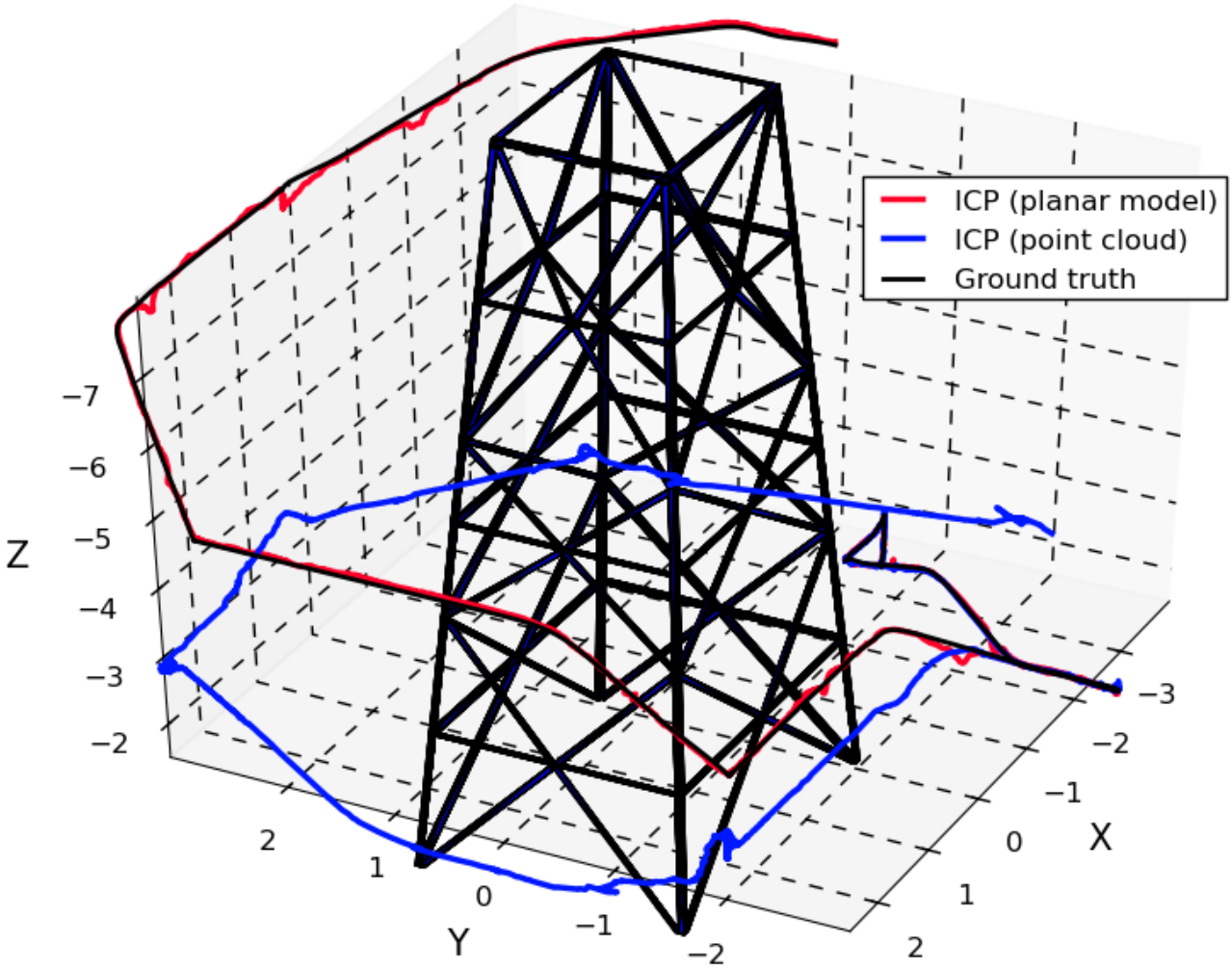
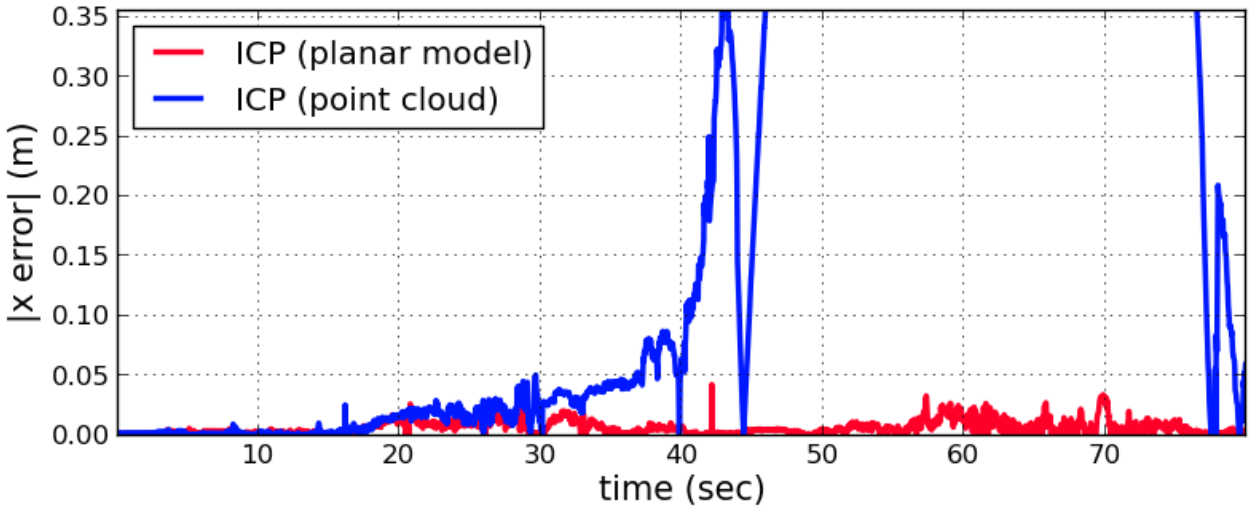
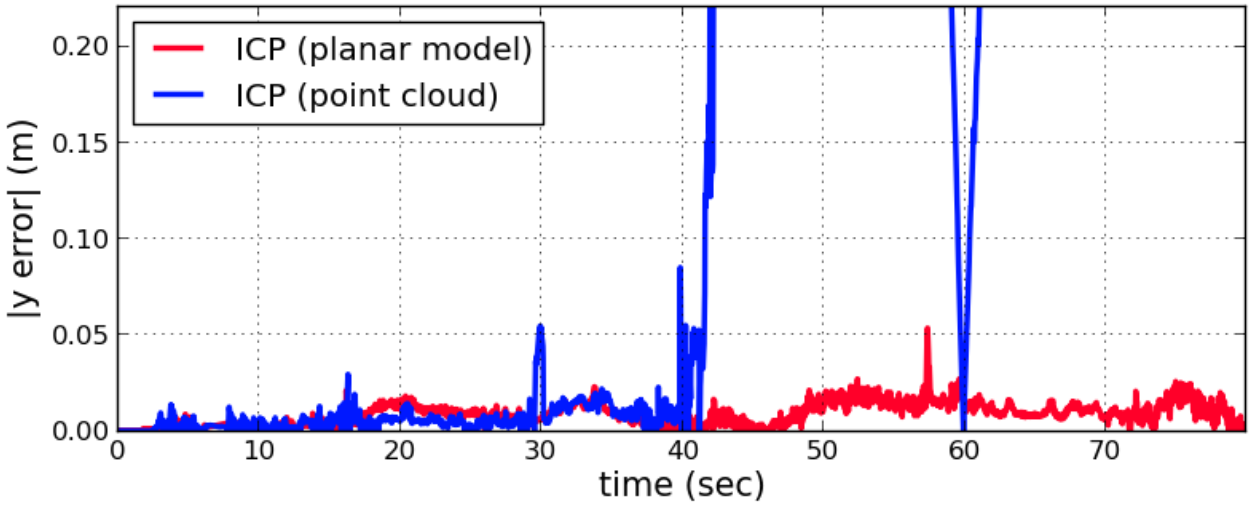


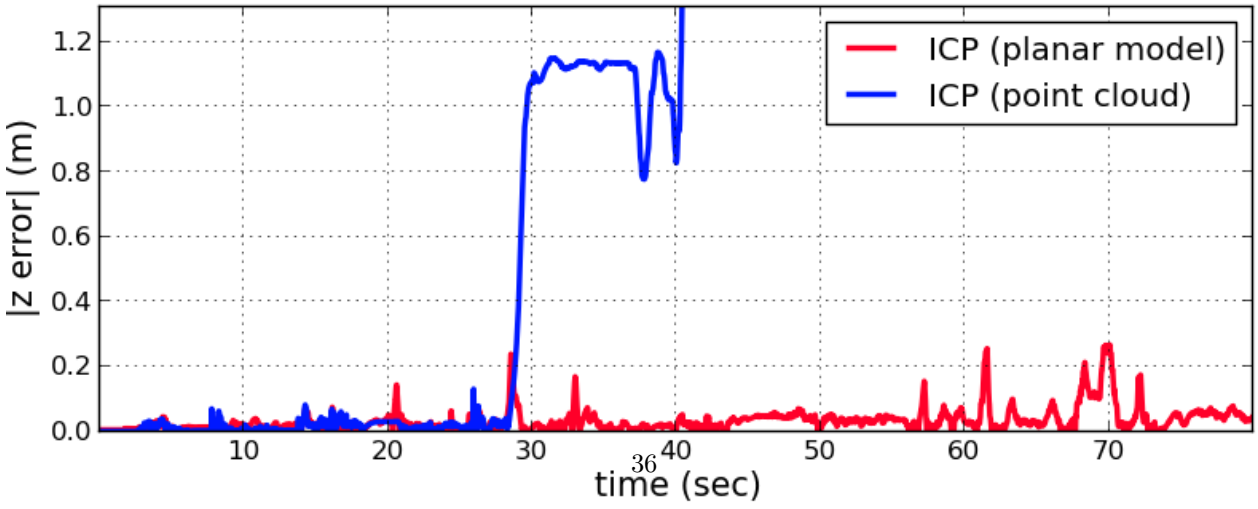
Figure 19: Comparing the ICP position estimates with the ground truth, for the simulated flight from Fig. 14. The point cloud approach fails before finishing the flight.


ICP (planar model)
ICP (point cloud)
|x error| (m)
time (sec)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
10
20
30
40
50
60
70

(a)


ICP (planar model)
ICP (point cloud)
|y error| (m)
time (sec)
0.00
0.05
0.10
0.15
0.20
0
10
20
30
40
50
60
70

(b)


ICP (planar model)
ICP (point cloud)
|z error| (m)
time (sec)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
10
20
30
40
50
60
70

(c)

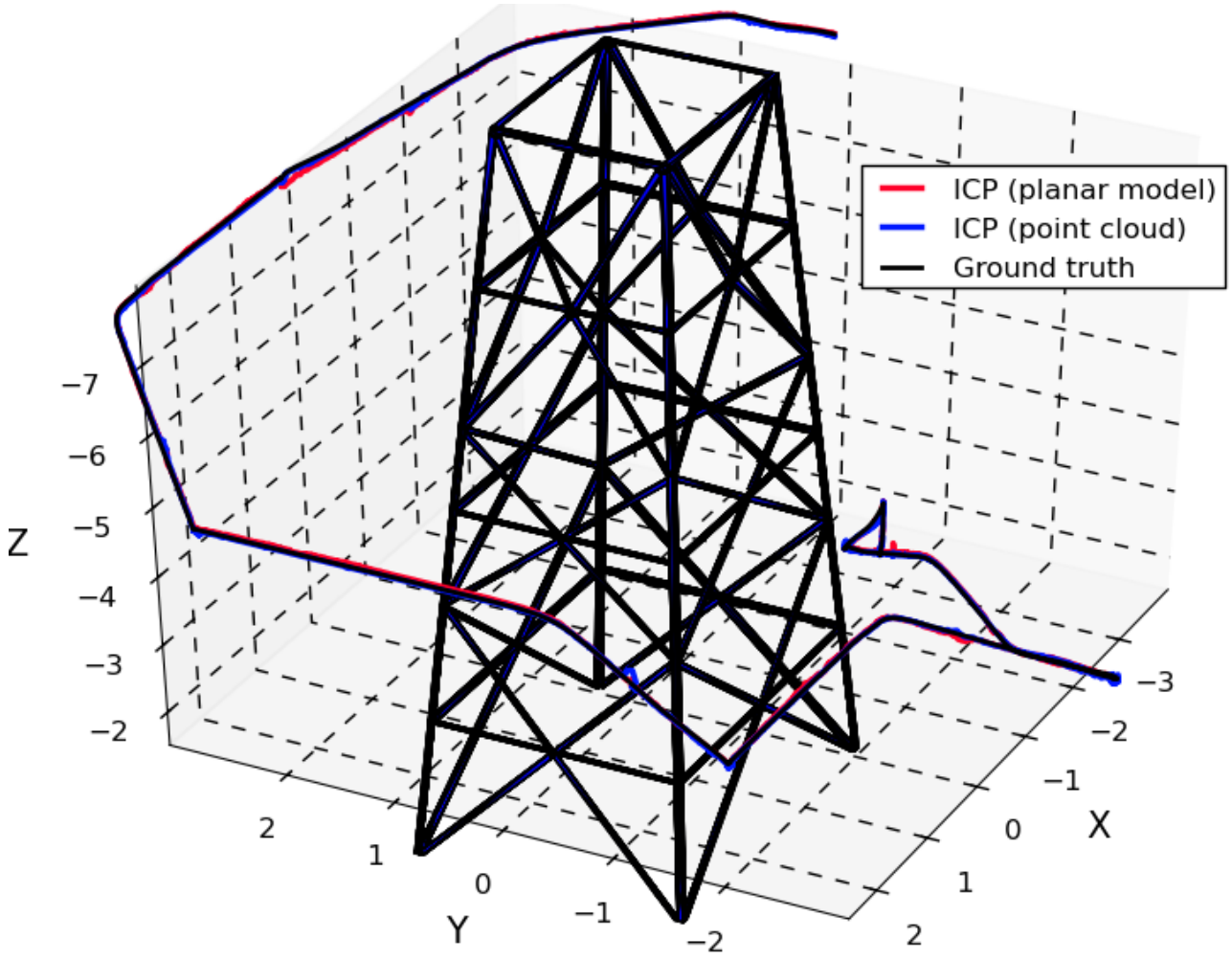
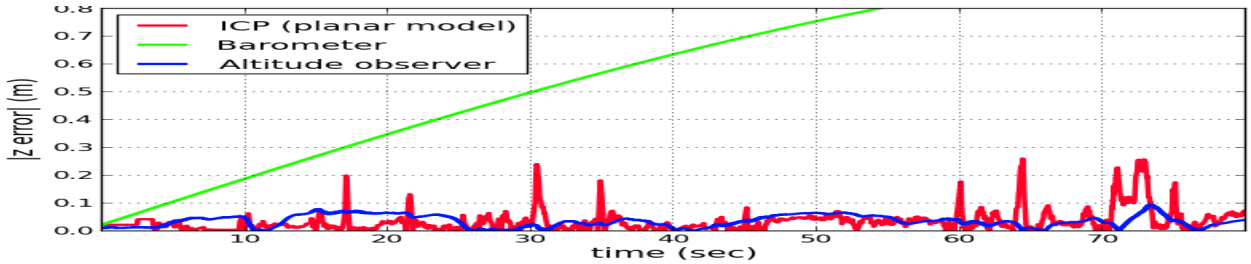
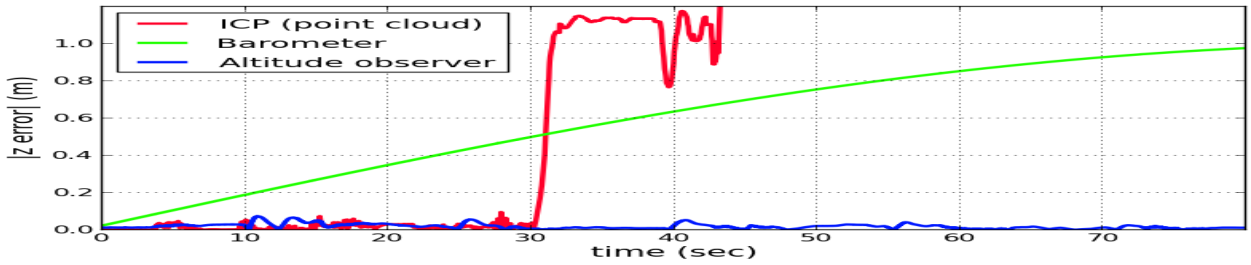


Figure 21: The ICP position estimates after introducing the altitude observer. The large altitude errors are corrected and the point cloud approach no longer fails.

similar performance before failing (near $t = 40s$). Then, particular attention must be given to the altitude estimation errors from Fig. 20c. As previously pointed out, the horizontally placed 2D laser scanner captures very limited altitude information. As a result, it was observed throughout the simulations that the altitude estimates were easily deteriorated in complicated situations, for example, when the horizontal bars on the tower block most of the sides from the sensor's view (as in Fig. 4d). For the planar model case, this typically caused spurious estimates, with the absolute error jumping above 20cm (e.g., $t = 20s$ and $t = 30s$ in Fig. 20c). For the point cloud case, this eventually caused the approach to completely fail (around $t = 30s$ in Fig. 20c). Despite these complications, in Fig. 20c it can be observed that the altitude errors remain at acceptable levels below 10cm throughout most of the flight for the planar case, and similarly for the point cloud approach before failure. Properly exploiting the limited altitude information requires special attention and is addressed in the following section.



(a)



(b)

Figure 22: 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.

8.3 Altitude estimation results

We now present the results for the altitude observer from Eq. (19), which fuses the laser altitude estimates from both implementations of the ICP algorithm with the barometer measurements. Barometer readings are sensitive to changes in atmospheric conditions (strong winds, temperature changes), which 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 meter per minute. As previously mentioned, the weights from Eq. (20) were chosen as $\lambda_1 = 1$, to rely mainly on the barometer measurements to estimate the 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. In these simulations, the observer's output was used at each scan registration initialization, instead of the laser altitude estimates. The impact of this can be observed in Fig. 21, where the complete position estimates of the ICP implementations are once again compared to the simulation ground truth. In contrast to the results from Fig. 19, it can be noted that the introduction of the altitude observer allows correcting the large altitude errors that previously caused the point cloud case to fail, which now offers a similar performance to the planar model case. Furthermore, Fig. 22 presents a comparison of the absolute errors of the barometer measurements, the altitude estimates of the ICP algorithm (without the aid of the observer), and the altitude observer's output. In both cases it can be noted that, 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. For the planar model case in Fig. 22a, the spurious error peaks are largely filtered, and the maximum error is lowered to 10cm. More importantly, for the point cloud case in Fig. 22b, the observer manages to avoid failing at $t = 30s$ and provides continuous estimates throughout the entire flight. The effectiveness of this formulation is further verified in Fig. 23, as the observer manages to estimate the previously unknown barometer drift, with less than 10cm of error.

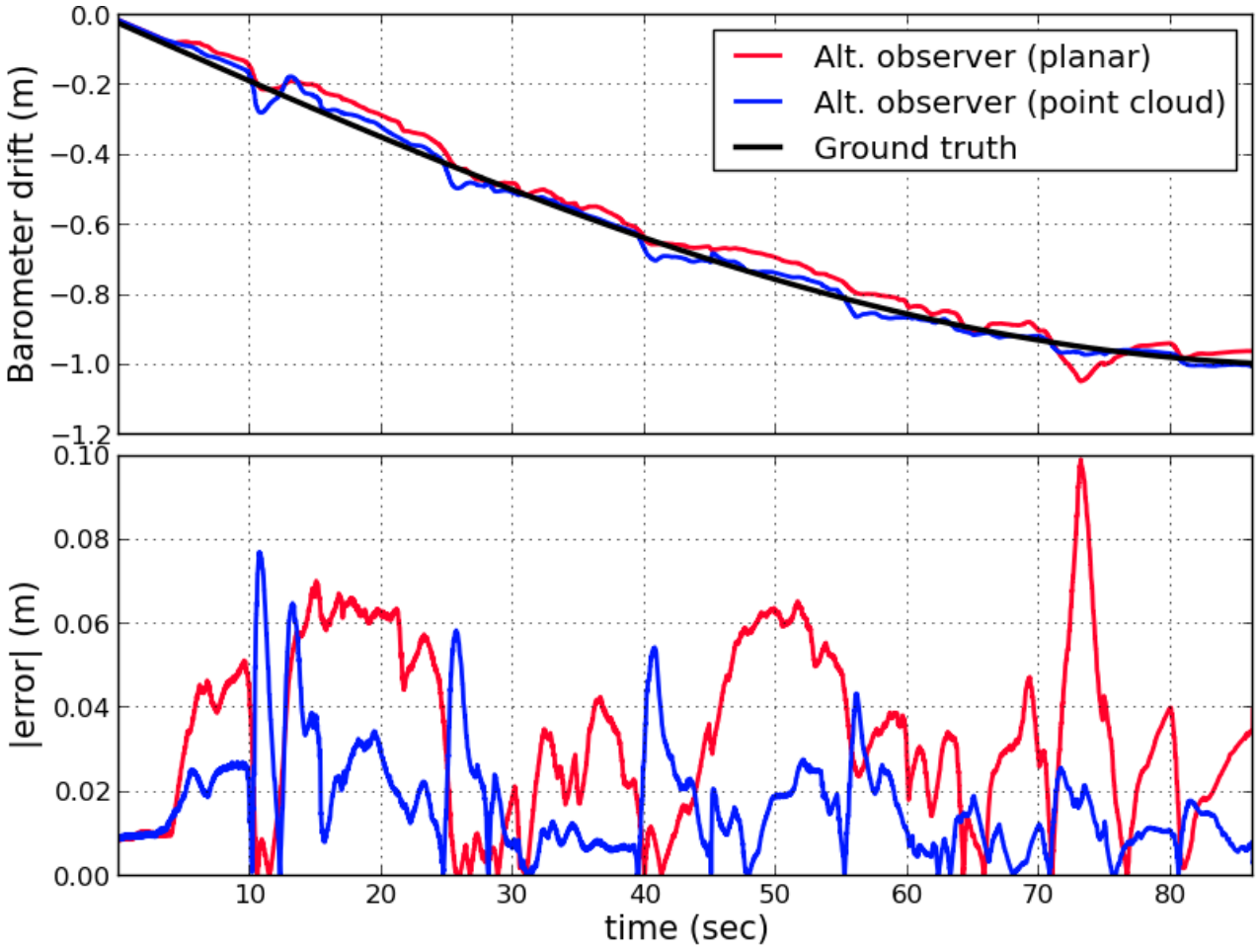


Figure 23: For the altitude observer and both ICP implementations: (Top) Comparing the barometer drift estimates with the ground truth. (Bottom) Absolute estimation errors. The observer succeeds in both cases.

9 Simulation results: Velocity estimation

9.1 Horizontal velocity estimation results

The $\{x, y\}$ estimates from the laser scan registration were used as an input to the velocity observers from Eq. (34), where they were fused with the accelerometer readings to recover the horizontal velocity estimates. The estimation gains were chosen identical for both axis as $(k_x, k_{v_x}) = (6.4, 16)$ and $(k_y, k_{v_y}) = (6.4, 16)$. The estimated horizontal velocities for a portion of the flight and both implementations of the ICP algorithm are shown in Fig. 24. For both axis, the high estimation gains allow the speed estimates to converge fast towards the ground truth. Furthermore, the good quality of the position estimates from the scan registration allows the velocity errors to remain below $10\frac{cm}{s}$.

9.2 Vertical velocity estimation results

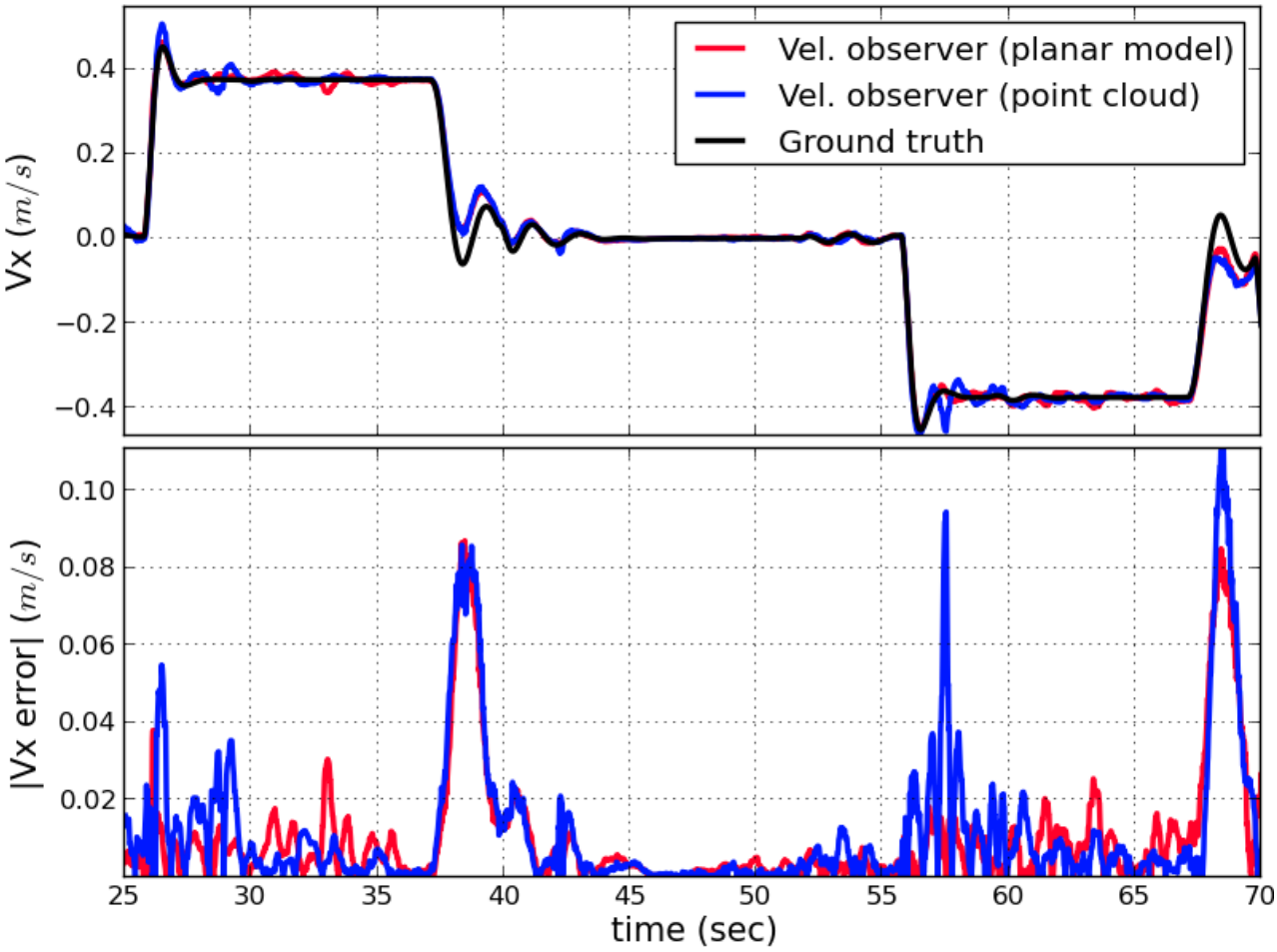
Finally, the observer from Eq. (35) was used to recover vertical velocity estimates from the barometer and accelerometer readings. As before, the observer gains were chosen as $(k_z, k_{v_z}) = (6.4, 16)$. Fig. 25a shows the estimation results without barometer drift. As can be seen, the vertical velocity estimates are sufficiently accurate without the need of the laser estimates, as they remain below $1.5\frac{cm}{s}$. Then, the velocity estimates in the presence of barometer drift are shown in Fig. 25b. With respect to the previous case, the estimation error slightly increases, but remains within acceptable levels.

10 Conclusions and future work

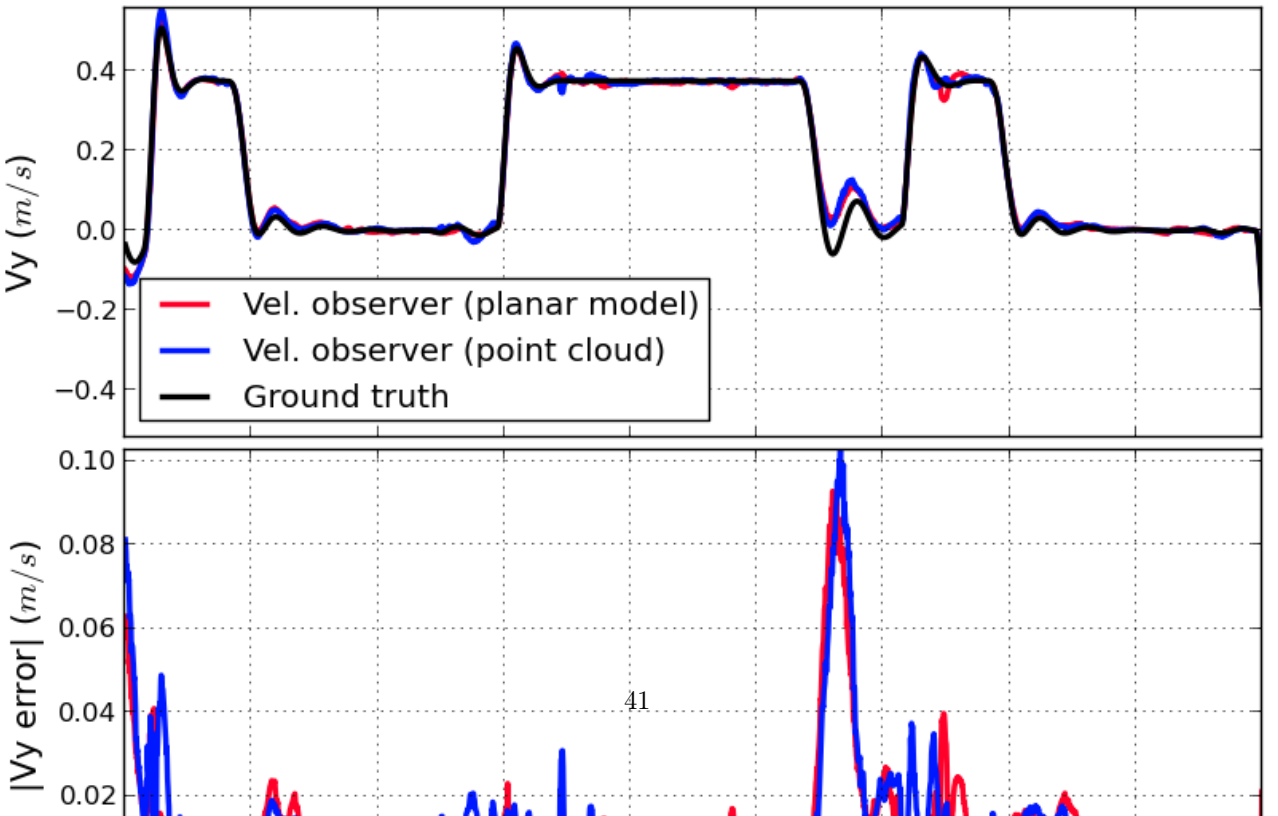
In this article we have presented a methodology to recover complete 3D pose estimates in electric tower inspection tasks with MAVs, using a sensor setup consisting of a 2D LiDAR, a barometer sensor and an IMU. First, we addressed 2D local pose estimation using uniquely the laser range measurements. Basic geometric knowledge of the tower was used to extract the notable features captured in the individual laser scans, which were then used to track the cross-sections and to estimate their previously unknown dimensions. Simulations yielded satisfying results under simple flight conditions, but the assumptions used by this approach proved too restrictive for general inspection tasks. It was shown that this tracking method could instead be used with additional sensing to model the tower. This was tested on data acquired from real flights, and results were presented for a partial point cloud reconstruction of the tower's body and a simplified planar representation derived from the dimensions estimated on-flight. The inspection task was thus divided into two steps, tower modelling and pose estimation.

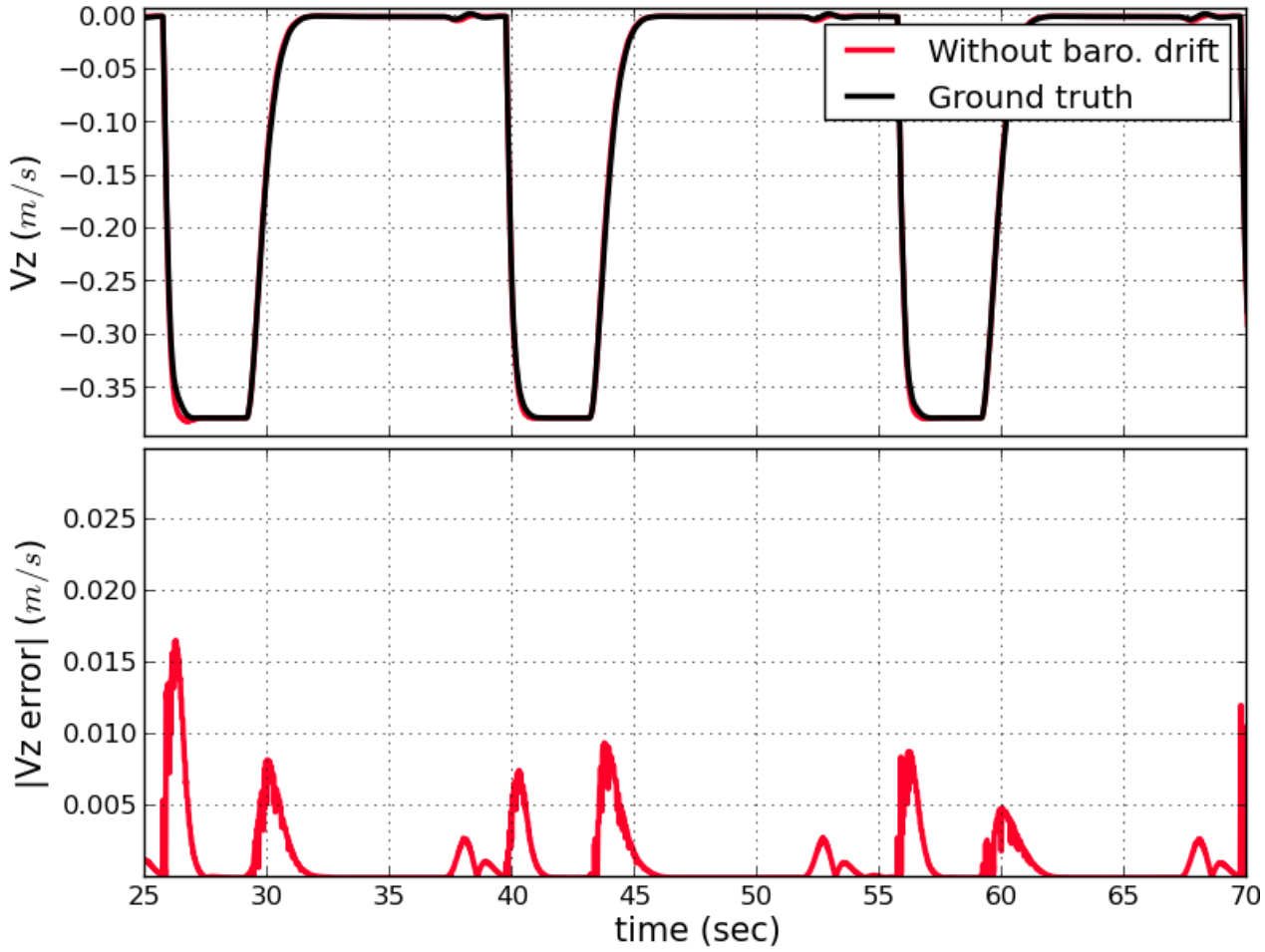
Then, we focused on 3D local pose estimation using the complete sensor setup, which was divided into three components. At the lowest level, a non-linear observer formulation to estimate the roll and pitch angles from the accelerometer and gyrometer measurements was presented. A gain scheduling approach to adapt the observer to changing flight dynamics was also introduced. 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 and a projection-based matching strategy. In both cases, the registration process was carried out in 3D and aided by the attitude estimates from IMU measurements, which allowed recovering altitude information. Lastly, a third component fused the barometer measurements and the altitude estimates from the scan registration. This simple formulation allowed estimating the unknown barometer drift in the process. Each of these components were validated in simulations. When combined, they showed satisfying results in terms of accuracy and computation time.

Finally, velocity estimation was achieved with simple feedback observers to exploit the MAV's dynamics. On one hand, the pose estimates were fused with inertial measurements to recover horizontal velocity estimates. On the other hand, the barometer measurements were fused with accelerometer measurements to recover the vertical velocity component. Simulations were also used to validate the efficiency of these estimations.

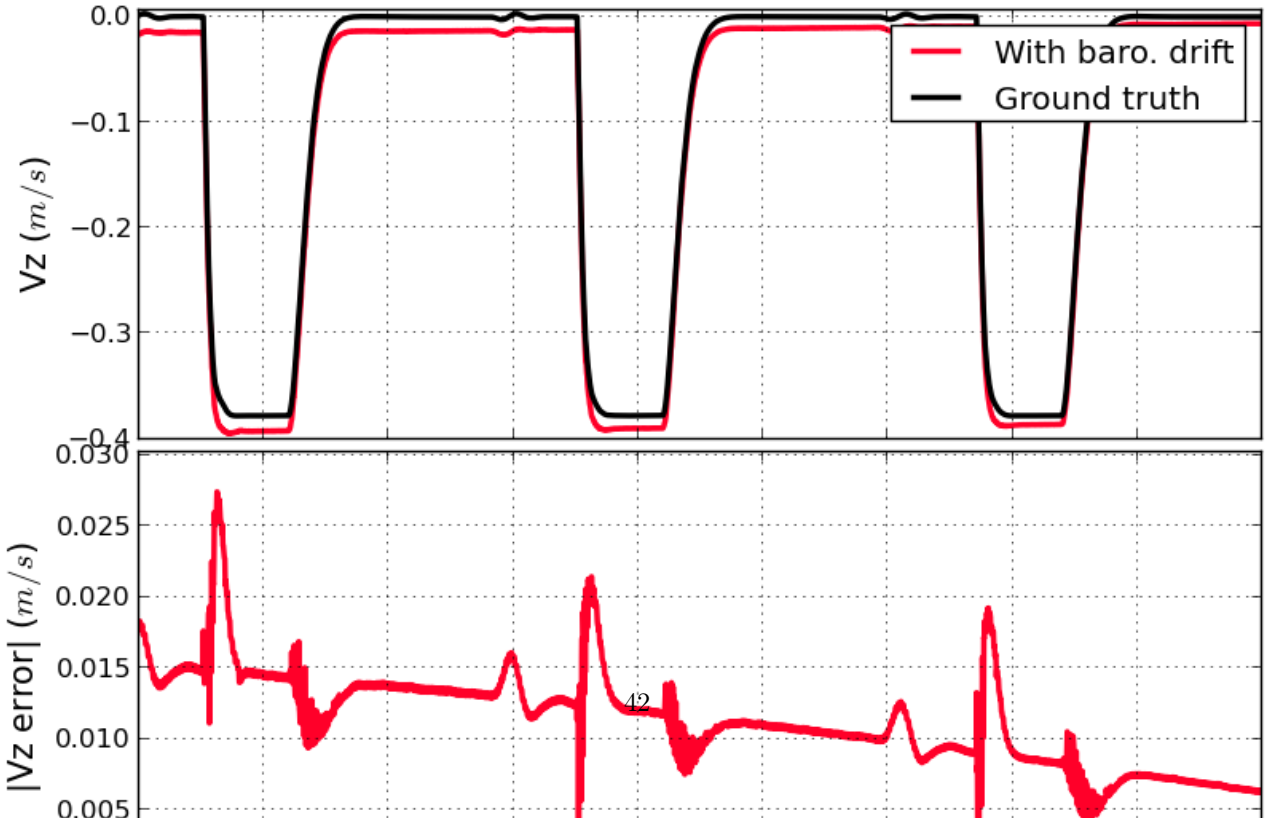

Vel. observer (planar model)
Vel. observer (point cloud)
Ground truth
Vx (m/s)
|Vx error| (m/s)
time (sec)

(a)


Vel. observer (planar model)
Vel. observer (point cloud)
Ground truth
Vy (m/s)
|Vy error| (m/s)


Without baro. drift
Ground truth
Vz (m/s)
|Vz error| (m/s)
time (sec)

(a)


With baro. drift
Ground truth
Vz (m/s)
|Vz error| (m/s)

An immediate continuation of this work includes introducing the pose and velocity estimates in the feedback control loop to stabilize the MAV's position. We are also interested in conducting further experimental validations of the methods proposed in this work. Since this study was limited to electric tower bodies with rectangular cross-sections, it would also result interesting to extend the methodology to the complete structure, including the head of the tower, and to more complex tower geometries.

11 Appendix

11.1 Stability analysis and gain tuning of the altitude observer

To analyse the stability of our proposed altitude observer formulation from Eq. (19), we first deduce error dynamics of the system. Modelling the barometer measurements as $z_{\text{baro}} = z + b_z$ and the laser estimates as $z_{\text{laser}} = z$, and substituting in Eq. (20), one obtains

$$\overline{z}_n = z - \lambda_n \tilde{b}_z, \quad 0 \leq \lambda_n \leq 1, \quad n = 1, 2, \tag{37}$$

where $\tilde{b}_z = \hat{b}_z - b_z$ is the bias estimation error. Substituting this in Eq. (19), and subtracting the vertical dynamics from Eq. (17), one obtains the error dynamics of the system as

$$\begin{cases} \dot{\tilde{z}} &= -k_z(\tilde{z} + \lambda_1 \tilde{b}_z) \\ \dot{\tilde{b}}_z &= -k_{b_z}(\tilde{z} + \lambda_2 \tilde{b}_z) \end{cases} \tag{38}$$

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

$$\begin{bmatrix} \dot{\tilde{z}} \\ \dot{\tilde{b}}_z \end{bmatrix} = \begin{bmatrix} -k_z & -\lambda_1 k_z \\ -k_{b_z} & -\lambda_2 k_{b_z} \end{bmatrix} \begin{bmatrix} \tilde{z} \\ \tilde{b}_z \end{bmatrix} = \tilde{\mathbf{A}} \begin{bmatrix} \tilde{z} \\ \tilde{b}_z \end{bmatrix} \tag{39}$$

Stability analysis follows, by analysing the roots of the characteristic polynomial of Eq. (39), obtained from solving $\det(\mathrm{s}\mathbf{I} - \tilde{\mathbf{A}}) = 0$. This results in

$$s^2 + (\lambda_2 k_{b_z} + k_z)s + k_{b_z} k_z (\lambda_2 - \lambda_1) = 0, \quad \lambda_1 \neq \lambda_2, \tag{40}$$

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

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \quad \zeta, \omega_n > 0, \tag{41}$$

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 one hand, if $\lambda_2 = 0$, then solving by substitution one obtains

$$\begin{cases} k_z = 2\zeta\omega_n \\ k_{b_z} = -\dfrac{\omega_n^2}{\lambda_1 k_z}, \quad \lambda_2 = 0,\ 0 < \lambda_1 \leq 1. \end{cases} \tag{42}$$

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. This was the case considered in the simulations, where $\zeta = 1.1$ (overdamped response), $\omega_n = 3.0$ and $\lambda_1 = 1$ lead to $(k_z, k_{b_z}) = (6.6, -1.36)$.

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

$$\begin{cases} k_z = \dfrac{2\zeta\omega_n \mp \sqrt{(2\zeta\omega_n)^2 - \frac{4\lambda_2\omega_n^2}{\lambda_2-\lambda_1}}}{2} \\ k_{b_z} = \dfrac{2\zeta\omega_n \pm \sqrt{(2\zeta\omega_n)^2 - \frac{4\lambda_2\omega_n^2}{\lambda_2-\lambda_1}}}{2\lambda_2}, \quad \lambda_2 > 0,\ \lambda_1 \neq \lambda_2. \end{cases} \tag{43}$$

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

$$\Delta = (2\zeta\omega_n)^2 - \frac{4\lambda_2\omega_n^2}{\lambda_2 - \lambda_1} \geq 0, \tag{44}$$

leading to the following inequality

$$\zeta^2 \geq \frac{\lambda_2}{\lambda_2 - \lambda_1}, \tag{45}$$

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. 45 holds.

References

[1] 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.

[2] 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.

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

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

[5] D. Droeschel, J. Stückler, and S. Behnke. Local multi-resolution surfel grids for mav motion estimation and 3d mapping. In *Intelligent Autonomous Systems 13*, pages 429–442. Springer, 2016.

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

[7] 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.

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

[9] W. Gander and J. Hrebicek, editors. *Solving problems in scientific computing using Maple and Matlab®*. Springer-Verlag Berlin Heidelberg, 4 edition, 2004.

[10] 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), 2005.*, pages 288–295, 2005.

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

[12] Bo Guo, Xianfeng Huang, Qingquan Li, Fan Zhang, Jiasong Zhu, and Chisheng Wang. A stochastic geometry method for pylon reconstruction from airborne lidar data. *Remote Sensing*, 8(3):243, 2016.

[13] 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), 2016*, pages 714–719, 2016.

[14] 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.

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

[16] 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.

[17] 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), 2004*, volume 3, pages 2149–2154 vol.3, Sept 2004.

[18] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf. A flexible and scalable SLAM system with full 3d motion estimation. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2011.*, pages 155–160.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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), 2013*, pages 3923–3929, November 2013.

[24] 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.

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

[27] 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.

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

[29] 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, 2012*, pages 400–411. Springer, 2012.

[30] 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.

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

[32] 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), 2003*, pages 276–283, 2003.

[33] 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.

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

[35] 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), 2014*, pages 1734–1740, July 2014.

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

[37] 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), 2012*, pages 2202–2209, May 2012.

[38] 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.

[39] 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.

[40] 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), 2011*, pages 20–25, May 2011.

[41] 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.

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

[43] 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, 2008*, pages 191–196, March 2008.

[44] 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, 2007*, pages 6424–6429, December 2007.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

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

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

[51] 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.