Laser-Based Control of Rotary-Wing UAVs

Alexandre Luís Cordeiro Gomes
alexandre.c.gomes@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

December 2015

Abstract

This paper addresses the problem of using exclusively sensors on board Unmanned Aerial Vehicles (UAVs) to derive attitude determination tools and trajectory tracking strategies. Firstly, this work discusses the perception of the outside world by the vehicle and the formulation of a mathematical description containing information regarding its position and attitude relative to the structure. For this purpose, a geometry is set and the best fit to the data provided by a LiDAR sensor is selected, after a robust outlier filtering process. With this information, several methods for obtaining the attitude are proposed. These include a fast and comprehensive yaw estimator, based on continuity, and a closed-form solution for the Wahba's problem and a nonlinear filter for a full attitude estimation, both on the group of rotation matrices. Both simulated and experimental results are provided for the performance evaluation of the perception algorithms. Building on these results, a nonlinear control strategy is designed with the objective of providing an accurate trajectory tracking control relative to the structure, with guaranteed asymptotic stability.
Keywords: UAVs, LiDAR sensor, geometry fitting, attitude determination, Lyapunov stability, trajectory control.

1. Introduction

An Unmanned Aerial Vehicle (UAV), also known as a drone, is an aircraft without a human pilot aboard. They were initially developed with military purposes, namely for situations where manned flight was considered too risky or difficult [3, 13]. During these high precision tasks, they required a set of sensors that would allow them to maneuver in the world as accurately as possible. As with other technology advancements, the world soon realized that these small vehicles, equipped with a great variety of sensors, could be used in tasks other than warfare.

Regarding the sensors on board, they need to be able to create a digital awareness of the vehicle's movement in the world. The world as we know it is a three Dimensional (3D) space, where any movement of a rigid body can be decomposed into a combination of translations and rotations along three different axes. In an aircraft, the origin is usually at the vehicle's center of mass and a rotation about its longitudinal, lateral, and vertical axes is called roll, pitch, and yaw respectively. Usually, UAVs contain an Inertial Measurement Unit (IMU) and a Global Positioning System (GPS), which are used to estimate the information regarding the acceleration, velocity and position of the vehicle. A Light Detection And Ranging (LiDAR) system is a sensor that measures distance by illuminating a target with a laser and analyzing the reflected light. It can be seen as a combination of laser-focused imaging with the ability of the Radio Detection And Ranging (RADAR) system to calculate a distance by measuring the time-of-flight of a signal [4]. During navigation tasks, there is always the possibility of sensor failure due to unforeseen events. Having a GPS on board is one of the main reasons these vehicles are able to fly autonomously, a feature that can become compromised in the vicinity of large infrastructures or under bridges. This happens since the GPS signal can be easily occluded by these structures. The LiDAR can be an interesting alternative, from the redundancy point of view, that may be used to overcome signal propagation issues and even calculate relevant data in a more accurate way.

The technological evolution has led to an increase in the demand for more and larger wind turbines, cellphone towers, and power lines, to name a few. All these large buildings and facilities are critical infrastructures that require maintenance through structural inspections and health monitoring, which in general requires complex and expensive routine inspections and monitoring procedures. Those processes can become inefficient in situations where the access is difficult, time-consuming, and often dangerous [1]. Accurate health monitoring and diagnosis of these infrastructures will increase the efficiency of maintenance and repair plans, with inherent benefits in terms of cost reduction and damage minimization in case of disaster. Small vehicles such as UAVs constitute a tailor-made solution, with the objective of producing accurate data sets, in real time, with the required spatial and temporal resolutions and thereby providing quantitative information. To increase safety and efficiency in structure inspections, by using UAVs, is an important step forward.

UAVs are rapidly evolving to become highly capable

sensing platforms, able to navigate and track trajectories with great accuracy. While the motion control of aerial vehicles in free flight is reaching its maturity, new challenges that involve interaction with the environment are being embraced. Using local sensors, such as IMUs, some quantities required for control tasks can be obtained depending entirely on the vehicle. As controlling the full motion of an aerial vehicle is not usually viable with only an IMU, this paper aims to take that interaction with the environment one step further and obtain additional measurements, from what the vehicle is perceiving of the world. Moreover, the use of a LiDAR may present an interesting enhancement to structural inspections, since it provides a way to maintain a lock on the target. This relative positioning method has the potential to be more effective and subject to less drift and error accumulation over time than traditional absolute positioning systems, as it depends directly on LiDAR scans acquired at a high frequency.

The basis for this paper can be found in [6], where a LiDAR sensor was used to provide a relative positioning solution to a structure and then to build the appropriated controllers. The idea of the current proposal is to extend that relative description to the vehicle's attitude, to allow a structure dependent trajectory. Furthermore, the control techniques investigated in the aforementioned work rely on the local linearization of a nonlinear system, used afterwards within a gain scheduling strategy. The present work intends to compute a nonlinear controller, for the original system, that ensures asymptotic stability with a large region of attraction. Several contributions can be derived from this work. Regarding the landmark detection, the framework for processing data from a laser sensor is the first, which besides being invaluable for this application it can be simply adjusted to work with different goals. It should be noted that these tools will allow for a 3D attitude to be retrieved from a sensor measuring in a 2D space. When it comes to the attitude estimation, the most relevant contribution is a nonlinear filter, that computes the rotation matrix describing the motion of a vehicle based on the fusion of measurements from several local sensors, including a LiDAR and separate components from an IMU. Finally, the motion control design yields a trajectory tracking controller solely based on local sensory information, therefore providing a relative positioning solution.

Concerning the organization of the paper, firstly Section 2 discusses the detection part of the approach, more specifically how the vehicle perceives the world and translates it to an interpretation intuitive and accurate enough to be manipulated. For this purpose, the LiDAR sensor used for the development of this work was investigated, to acquire a sense about its features and draw some conclusions regarding its suitability for this application. After establishing the characteristics of the source information, the pier detection consists of building a strategy based on the Split & Merge algorithm for the extraction of landmark features. This procedure is paired with a robust outlier identification and removal algorithm that intends to improve the performance of the detection stage. Then, a least square estimation is compared with a reduced space Hough transform algorithm in the process of fitting the data to build a description of the landmarks.

The following step is to actually process the data and derive several methods to obtain an attitude description consistent with the movement of the vehicle in Section 3, so that it can be used for establishing control strategies later on. Among them, there is a history-based yaw tracker, a solution to the Wahba's problem on the group of rotation matrices, and a nonlinear observer with filtering capabilities and a proof of stability. The following logical step towards validation is a proper evaluation of the previously developed work, in Section 4, both in simulation and the real world.

Next, Section 5 discusses the control strategies developed and their implementation in both simulation and the real world. The focus is on obtaining closed-loop solutions for the problem of trajectory stabilization, with respect to a fixed target, by determining an error variable that adequately represents the vehicle's position and orientation error relative to it, in the inertial space. The concepts from Lyapunov control theory are applied, to ensure the asymptotic stability of the derived control system. In the end, the conclusions obtained from the results presented along the paper are drawn in Section 6.

2. Environment Perception

There are several important reference frames during the LiDAR data processing, the first being the Inertial Frame $\{I\}$, which for simplicity is considered to be the local tangent plane with the North East Down (NED) convention. There is also the Body Frame $\{B\}$, with the origin at the vehicle's center of mass, the X axis pointing forward along the longitudinal axis of the vehicle, the Z axis pointing downward along its vertical, and the Y axis set to make this frame orthonormal. Finally, there is the intermediate Horizontal Frame $\{H\}$, which can be interpreted as a projection of $\{B\}$ on the XY plane of $\{I\}$.

2.1. LiDAR Reliability

The sensor's precision, and consequently the quality of the measurements, is a very sensitive matter, since it is one of the most relevant limiting factors in what concerns the accuracy of the data processing and the following results.

In [12], the accuracy of the data points from a two Dimensional (2D) Hokuyo UTM-30LX LiDAR, a sensor quite popular in small vehicles, was analysed while detecting moving cylindrical targets. This sensor has an angle opening of 270°, with a blind cone of 90° around its 6 o'clock, an angular resolution of 0.25°, a scanning frequency of 40 Hz, and a range of 30 m, weighting roughly 200 g. The described approach was based on the detection of markers belonging to a circular section from the data points provided by the LiDAR, after undergoing an outlier avoidance method and a least-squares circular fitting.

2.2. Detection Strategies

Detecting a structure and obtaining the pertinent information, needed to determine the vehicle's attitude, greatly depends on the knowledge of its geometry. Taking into account the most common types of infrastructures, first a cylindrical and then a cuboid geometry were analyzed, where the former was discovered to be not as reliable as the latter due to the sensor accuracy. Considering piers with a rectangular section is a good alternative for LiDAR-based perception and control, given the wide range of landmarks in which the curvature does not play a role in the fitted shape, as with cylindrical structures.

Facing these structures, the LiDAR will hit one or two faces of the pier, depending on its relative pose, and the intersection of this sensor's plane with these faces will result in two straight lines, which from hereon will be called edges. The idea behind attitude determination is that a specific movement, in roll or pitch, has an impact on both the length of the edges and the angle between them. That effect can be interpreted as the contribution of these two components, but not with a clear separation. The first stage is related with identifying how many edges the vehicle is encountering at each moment. For that purpose, a strategy based on the Split & Merge algorithm, which can be found in [9] and was originally proposed in [10], was developed.

The principle is to determine if the LiDAR is detecting one or two edges, in order to divide a set of measurements in the XY plane, denoted by $\boldsymbol{M}$, accordingly and isolate each edge. Ideally, the boundary point at each end of the LiDAR scan corresponds to a corner of the section. In $\mathbb{R}^2$, the perpendicular offset from a point $\boldsymbol{p} = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$ to a line is given by

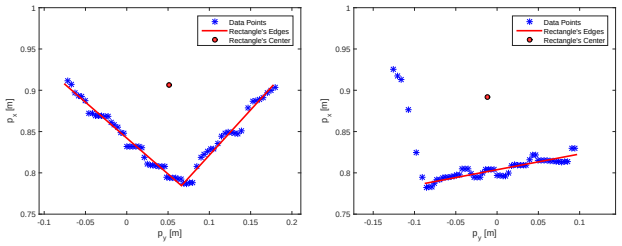
$$\boldsymbol{p}^T \boldsymbol{n} + c = e, \quad \text{with } \|\boldsymbol{n}\| = 1 \tag{1}$$

where $\boldsymbol{n} = \begin{bmatrix} n_x & n_y \end{bmatrix}^T$ is the unit vector normal to the line, c is the perpendicular offset from the line to the origin, and the error e is the actual offset, being null for points belonging to it. In the algorithm used in this work, the error e from every point on the data set to that line can then be calculated by direct substitution and the point furthest away will be the corner candidate, which upon confirmation can be used to split the data set into two smaller ones. This verification can be performed using the distance between a corner and the diagonal uniting the closest two other corners in a rectangle, defined as

$$e_{corner} = \frac{L_1 L_2}{\sqrt{L_1^2 + L_2^2}} \tag{2}$$

where L_i with $i = 1, 2$ are the dimensions of the pier, known beforehand. If the error of the potential corner stays below a threshold, there is most likely only one edge. In the remaining situations, the assumption is that there are two edges, but this can also happen due to the presence of boundary points resulting from reflections or measurement errors during the transition between one and two edges. Therefore, there is also a required minimum number of points to ensure a sufficiently accurate edge.

Figure 1 presents the output of the pier detection mechanism, in two different time instants of an experiment performed with the rotors off. The former is near the beginning, with both edges clearly visible, whereas the latter corresponds to the transition stage, where the algorithm helps deciding how many edges there are. For the sake of understanding what the algorithm does, after selecting a potential corner and confirming its legitimacy in both cases, the data points at the far left of Figure 1b were rejected for lacking in quantity. Continuing the movement around the pier, there would come a moment when the potential corner would not be accepted anymore, due to its proximity to the line uniting the two closest corners.



(a) Two edges. (b) One edge.

Figure 1: Cuboid pier detection during an experiment.

2.3. Outlier Identification and Removal

In the majority of locations around the pier, the detection strategy will consistently yield either one or two well defined edges. However, reflections of the beams and noise in the measurements generates outliers in the boundaries of the data set. With that in mind, an additional algorithm was developed and applied before the edge separation, improving the effectiveness of the previous task. This algorithm is meant to filter the outliers from the beginning and end of the data set $\boldsymbol{M}$, based on the relative distance between data points and two separate criteria. A way to account for the resolution of the LiDAR, depending on the distance from the target, is to find the average of the minimum distances between points, in the entire data set.

The outliers did not appear with a well defined pattern, in fact it was extremely random and at times very difficult to actually distinguish between a valid and an invalid data point. It is worth mentioning that these outliers can appear either as isolated points, which is more common, or small clusters of points, hence the existence of two different criteria. This leads to two different stages of trimming, for both the beginning and the end of the data set, the first for small clusters and the second for the remaining isolated points.

2.4. Geometry Fitting

After the data set separation, the identification implies defining the equation of the line for each existing edge. For this application, an alternative is to use a least squares estimation line fitting. The problem at hand is in the form

$$\begin{aligned} &\underset{c,n_x,n_y}{\text{Minimize}} \quad \|\boldsymbol{e}\|^2 = \sum_{i=1}^{N} e_i^2 \\ &\text{Subject to} \quad \begin{bmatrix} 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_N & y_N \end{bmatrix} \begin{bmatrix} c \\ n_x \\ n_y \end{bmatrix} = \begin{bmatrix} e_1 \\ \vdots \\ e_N \end{bmatrix}, \\ &\qquad\qquad n_x^2 + n_y^2 = 1 \end{aligned} \tag{3}$$

which can also be found in [5]. It consists of minimizing the sum of the squares of the perpendicular offset between the points and fitted line. With some mathematical manipulations, including a QR decomposition, the variables in this optimization can be determined by the Singular Value Decomposition (SVD) of a reduced problem.

Another option to obtain the fit to the data points is to use the Hough transform, a feature extraction technique used in image analysis, computer vision, and digital image processing [11]. Both these procedures were applied and produced coherent results, however there are some differences in their performance. On one hand, the least squares estimation is the fastest, whereas a reduced space version of the Hough transform takes 5 – 10 times longer to complete, due to its iterative nature. On the other hand, the former is highly affected by existing outliers as it tries to minimize the offset of all points to the estimated line, while the voting mechanism in the latter is very efficient in detecting the points that are most likely valid and adjusting the line only to them. In terms of effectiveness, despite definitely being a more accurate solution, the improvements of the reduced space Hough transform against the least squares estimation translate into an almost negligible change in the angle of the line, in the majority of cases. Therefore, the bottleneck in this situation comes down to time. Since this procedure has to be applied to every scan of the LiDAR, the time difference between these methods causes the Hough transform to become an inviable alternative, leading to the choice of the least squares estimation.

After obtaining the line parameters for an edge, out of the entire data subset the only necessary data points are the boundaries and the corner, if it exists. These relevant data points need to be corrected according to the obtained line fit through

$$\overline{\boldsymbol{p}} = \boldsymbol{p} - \boldsymbol{n}e = \boldsymbol{p} - \boldsymbol{n}(\boldsymbol{p}^T\boldsymbol{n} + c) \tag{4}$$

in order to determine where they should have been in the first place. It should be noted that if there are two edges and consequently a corner, this point is shared by both edges and therefore needs to be corrected using the information from the two fits to provide the best estimation of the intersection.

After correcting the data points, the estimation of the length of the edges can be made. Having two edges, the following requirement is that both have to be larger than a factor times the smaller real dimension, otherwise the smallest edge is discarded. This factor has to be tuned according to the circumstances. This verification occurs in order to exclude exceptional cases where the data set identification's conditions were passed but the result does not have true physical meaning.

2.5. Complete Representation

The corrected boundary points and corner, if it exists, can be identified as the start and end points of each edge, denoted here by $\boldsymbol{p}_{s_i}$ and $\boldsymbol{p}_{e_i}$ with $i = 1, 2$ respectively. The edges are represented by a matrix $\boldsymbol{Q}$, expressed in $\{B\}$, where its columns are the vectors $\boldsymbol{Q}_i = \begin{bmatrix} Q_{i_x} & Q_{i_y} \end{bmatrix}^T$ with $i = 1, 2$, such that $\boldsymbol{Q}_i = \boldsymbol{p}_{e_i} - \boldsymbol{p}_{s_i}$. Notice that all these variables are time-varying and depend on the motion of the vehicle. In particular, the correspondence between boundary points and corner and the points $\boldsymbol{p}_{s_i}$ and $\boldsymbol{p}_{e_i}$ can be obtained by knowing the initial condition, while encountering the pier, and observing it evolve over time.

Planning a trajectory and navigating around a pier is a task highly dependent on the location of its center, since it is the pier's most logical fixed point for maintaining a reference. This location is ideally invariant relative to the movement of the vehicle, and it can be determined at all times given the real geometry of the pier and the knowledge regarding the number and details of the edges being encountered. With that information, an estimation of the center location can be computed and later used for control tasks, concerning the navigation around the structure.

The representation of the edges in $\{I\}$ is also of great interest, so that they can be related through the rotation matrix from $\{I\}$ to $\{B\}$, denoted by ${}^B_I\boldsymbol{R}$, according to ${}^I\boldsymbol{Q} = {}^I_B\boldsymbol{R}{}^B\boldsymbol{Q}$. As illustrated in Figure 2, their projection in the XY plane of this reference frame corresponds to the section of the pier and can be defined as $\boldsymbol{l}_i$, where its norm is L_i with $i = 1, 2$. Regarding the Z coordinate, represented by h_i with $i = 1, 2$, in $\{B\}$ it is null but in $\{I\}$ it is the direct result of the rotation the vehicle is exhibiting momentarily. Therefore, the edges in $\{I\}$ become

$${}^I\boldsymbol{Q}_i = \boldsymbol{l}_i \pm h_i\boldsymbol{e}_3, \text{ for } i = 1, 2 \tag{5}$$

where $\boldsymbol{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ is the versor of the Z axis. Knowing the dimensions of the pier and the length of the edges, which is independent of the reference frame, the Z coordinate can be obtained through

$$h_i = \sqrt{\|{}^B\boldsymbol{Q}_i\|^2 - L_i^2}, \text{ for } i = 1, 2 \tag{6}$$

However, this calculation only considers the length of the edges. Applying (6), for each edge, together with the cross product of both edges, with information on the angle between them, leads to an updated problem. The objective of the optimization, formulated as

$$\begin{aligned} &\underset{h_1^2,h_2^2}{\text{Minimize}} \quad \|\boldsymbol{\varepsilon}\|^2 = \sum_{i=1}^{3} \varepsilon_i^2 \quad \text{s.t.} \\ &\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ L_2^2 & L_1^2 \end{bmatrix} \begin{bmatrix} h_1^2 \\ h_2^2 \end{bmatrix} - \begin{bmatrix} \|{}^B\boldsymbol{Q}_1\|^2 - L_1^2 \\ \|{}^B\boldsymbol{Q}_2\|^2 - L_2^2 \\ \|S({}^B\boldsymbol{Q}_1){}^B\boldsymbol{Q}_2\|^2 - L_1^2L_2^2 \end{bmatrix} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}, \\ &h_i^2 \geq 0 \text{ for } i = 1, 2 \end{aligned} \tag{7}$$

is to minimize an error variable $\boldsymbol{\varepsilon}$, relating properties of the edges in $\{I\}$ and $\{B\}$. In (7), the $S(.)$ func-

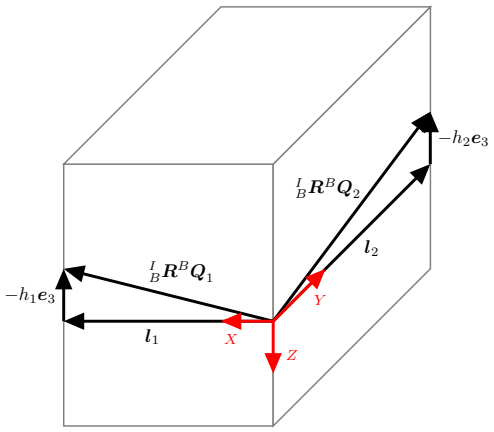


Figure 2: Decomposition of the edges in $\{I\}$.

tion produces a skew-symmetric matrix, defined such that, for any vector $\boldsymbol{\nu}$, the $S(\boldsymbol{\nu}_1)\boldsymbol{\nu}_2 = \boldsymbol{\nu}_1 \times \boldsymbol{\nu}_2$ equality verifies.

It should be noted that the variables being optimized are the squares of the Z coordinates, naturally being subject to non-negativity constraints, so in the end the desired result will be the square root of their values. This means there are always two options for the value, according to the sign. The approach to solve this ambiguity aims to maintain continuity, by choosing the closest value to the previous one, assuming there are no swift movements around the leveled flight.

An additional idea to make the optimization more robust and improve its result can be investigated by assuming that, between consecutive time steps, the change in the Z coordinates is very small. Considering that assumption, which is fairly reasonable for a short number of time steps, the n previous samples can be incorporated into the current formulation. Expanding the amount of information used in the optimization shows a slight improvement of the results for increasing n, thus the number of backward coherent samples was decided taking into account the best compromise between the optimization's stability and the vehicle's motion flexibility.

3. Attitude Determination Methods

The next step is to analyze the description created in Section 2 and propose several methods capable of accurately extracting a partial or the full attitude of the vehicle. The first approach is to consider a decoupling of the vehicle's movement into the XY plane and the Z axis, in Section 3.1. On the other hand, Section 3.2 goes one step further and elaborates on two additional options to compute the attitude of the vehicle, this time considering the 3D world altogether.

3.1. Partial Motion

The edges are detected with the LiDAR in $\{B\}$, but their dimension is known and corresponds to the norm of ${}^H\boldsymbol{Q}_i = \boldsymbol{l}_i$ with $i = 1, 2$. The rotation matrix from $\{B\}$ to $\{H\}$ can be used to obtain the projection of the edges

$$\|\boldsymbol{\Pi}_{\boldsymbol{e}_3}\,{}^H\boldsymbol{Q}_i\|^2 = \|\boldsymbol{\Pi}_{\boldsymbol{e}_3}\,{}^H_B\boldsymbol{R}^B\boldsymbol{Q}_i\|^2 = L_i^2, \text{ for } i = 1, 2 \quad (8)$$

where $\boldsymbol{\Pi}_{\boldsymbol{e}_3} = \boldsymbol{I} - \boldsymbol{e}_3\boldsymbol{e}_3^T$. Since ${}^H_B\boldsymbol{R}$ is a function of the roll and pitch angles, ϕ and θ respectively, they can be estimated with (8).

However, this method heavily relies on the length of the edges, which greatly depends on the boundary points, that in turn are subject to the highest measurement errors. Despite the lengths not being accurate enough, the angle between the edges is also affected by the motion of the vehicle but it was not used yet. This quantity does not depend solely on a pair of points, as the lengths, rather it is the result of fitting an entire set of points, thus carrying less uncertainty.

3.1.1. Yaw Estimator

Although retrieving the roll and pitch angles independently was concluded to be a process with not enough accuracy, they can be obtained through an IMU and then combined with data from the LiDAR, to provide a better estimate of the yaw angle ψ. This is useful since, without the LiDAR measurements, the yaw is being acquired as a combination of the measurements from gyroscopes and magnetometers, the latter being susceptible to electromagnetic disturbances and having an unknown drift, in general.

Taking on an approach based on the angle between the edges rather than their length, the idea behind the projection introduced in (8) is revisited. Starting with the roll and pitch angles from the IMU, the edges ${}^B\boldsymbol{Q}$ can be projected to the XY plane of $\{H\}$, leaving the yaw angle ψ as the only unknown. Given the fact that, no matter what roll and pitch the vehicle has in a given instant, the projection of the data points to $\{H\}$ will always result in two orthogonal lines, then the edge identification step can be reformulated. Following this perspective, the data points can be rotated to $\{H\}$ before the constrained least squares problem instead of after, through

$${}^H\boldsymbol{M} = \boldsymbol{\Pi}_{\boldsymbol{e}_3}\,{}^H_B\boldsymbol{R}^B\boldsymbol{M} \quad (9)$$

Having the data points matching two orthogonal lines, (3) can be rewritten as

$$\begin{aligned} &\underset{c_1,c_2,n_x,n_y}{\text{Minimize}} \quad \|\boldsymbol{e}\|^2 = \sum_{i=1}^{N_1+N_2} e_i^2 \\ &\text{s.t.} \begin{bmatrix} 1 & 0 & x_{1,1} & y_{1,1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & x_{1,N_1} & y_{1,N_1} \\ 0 & 1 & y_{2,1} & -x_{2,1} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & y_{2,N_2} & -x_{2,N_2} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ n_x \\ n_y \end{bmatrix} = \begin{bmatrix} e_1 \\ \vdots \\ e_{N_1+N_2} \end{bmatrix}, \\ &\quad n_x^2 + n_y^2 = 1 \end{aligned} \quad (10)$$

to consider the minimization of two straight orthogonal lines simultaneously. With this approach, the least squares estimation becomes independent of the vehicle's roll and pitch and avoids the uncertainty of the angle between the edges, which corresponds to 90°. This means the error in the estimated heading to the structure can be reduced, since the data points of both edges

now contribute to an unified objective, with a joint error, rather than two separated goals with unrelatable errors. However, the accuracy of the roll and pitch estimates, obtained from the IMU, will also have an impact on the accuracy the yaw estimate. Performing the rotation of an edge from $\{H\}$ to $\{I\}$ only depends on the yaw angle ψ, whose estimate can be obtained through

$$ {}^{I}\boldsymbol{Q}_i = {}^{I}_{H}\boldsymbol{R}\,{}^{H}\boldsymbol{Q}_i \Leftrightarrow \hat{\psi} = \tan^{-1}\left(-\frac{Q_{i_y}}{Q_{i_x}}\right), \text{for } i \in \{1,2\} \tag{11} $$

It should be noted that the yaw usually has its reference at the magnetic North, so applying this method implies setting an artificial reference. In this case, it was chosen that the yaw angle would be set relative to the edge of the structure initially at the left.

Furthermore, when navigating around the structure, the LiDAR captures a sequence of time steps corresponding to different views, forming a progression. Most of the times there are effectively two edges, but in the transition stage, the accuracy of the data points and the consequent identification can originate misleading situations. As mentioned in Section 2.2, there are several reasons why there could be an intermittent edge, depending on the existence of outliers and the fulfillment of the edge splitting criteria. This leads to degenerated situations, where there is only one edge and it has a tilt from the perfect 90° angle, with respect to the vehicle. In practice, the case at hand ends up being one of the eight different possibilities from Figure 3, therefore an algorithm was developed to detect these cases and ensure a continuous transition scheme.

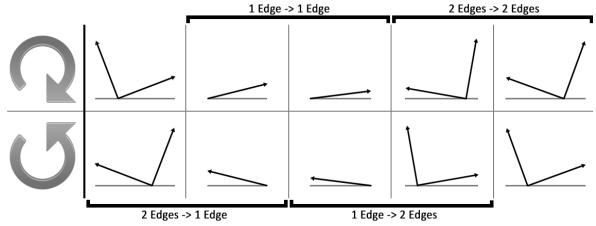


Figure 3: Progression of the view around the structure with all the transition cases.

3.2. Full Motion

Considering the complete motion capabilities of a vehicle in attitude involves procedures which can be more expensive, at a computational level. First, taking on a problem known in spacecraft attitude determination for many years, Section 3.2.1 describes what this robust solution consists of and how it can be used for this purpose with the information available from the LiDAR and specific components of an IMU. This formulation involves the computation of the optimal rotation matrix from $\{I\}$ to $\{B\}$, establishing the relationship between the observations in their main reference frames. The next is designed from scratch in Section 3.2.2, as an attempt to build a customized tool for this application in particular, where the stability principles and control theory behind it are mentioned as well. It is also presented the proof of convergence of this method, which is actually an observer fusing LiDAR data with raw information from the IMU, based on the boundedness of the bias it incurs throughout time.

The information provided by the IMU is a product of the combination of three types of sensors, namely accelerometers, gyroscopes, and magnetometers. The data coming from accelerometers has a high precision, but by depending on gravity, it cannot be used to describe the motion around the Z axis. Using the information from the gyroscopes complements that description, however it implies the integration of angular velocity over time, which is a process that accumulates errors and generates a bias with growing significance. The magnetometers are supposed to provide an additional correction, in the vertical axis, hereby completing the 3D attitude solution. Different combinations of these sensors can be made, to obtain alternative descriptions of the vehicle's attitude.

3.2.1. Wahba's Problem Application

The challenge commonly known as Wahba's problem refers to the determination of the three-axis attitude of an aircraft, by estimating the proper orthogonal matrix ${}^{B}_{I}\hat{\boldsymbol{R}}$ that minimizes the least squares loss function

$$ L({}^{B}_{I}\hat{\boldsymbol{R}}) = \frac{1}{2}\sum_{i=1}^{n_{obs}} \boldsymbol{w}_i \| {}^{B}\boldsymbol{O}_i - {}^{B}_{I}\hat{\boldsymbol{R}}\,{}^{I}\boldsymbol{O}_i \|^2 \tag{12} $$

where ${}^{I}\boldsymbol{O}$ is the set of measurements represented in $\{I\}$, ${}^{B}\boldsymbol{O}$ is the same set but represented in $\{B\}$, $\boldsymbol{w}$ is a vector with positive weights associated with each individual measurement and n_{obs} is the total number of observations [8]. The loss function is based on the idea that, if the measurements are free of errors and the true rotation matrix ${}^{B}_{I}\boldsymbol{R}$ is the same for all measurements, then ${}^{B}\boldsymbol{O}_i = {}^{B}_{I}\boldsymbol{R}\,{}^{I}\boldsymbol{O}_i, \forall i$.

Having the description of the edges in both $\{I\}$ and $\{B\}$, they can be directly introduced in the loss function. However, despite the fact that most of the times there are two edges that fully define the attitude of the vehicle, when only one edge is being seen, an ambiguity arises due to an additional DOF (Degree Of Freedom). The accelerometer within the IMU can be used to provide an extra piece of information and obtain an unequivocal attitude, assuming that the vehicle's acceleration is negligible relative to the gravitational acceleration. With that in mind, the input of the minimization will correspond to the observations matrix ${}^{*}\boldsymbol{O} = \begin{bmatrix} {}^{*}\boldsymbol{Q} & {}^{*}\boldsymbol{a} \end{bmatrix} = \begin{bmatrix} {}^{*}\boldsymbol{Q}_1 & {}^{*}\boldsymbol{Q}_2 & {}^{*}\boldsymbol{a} \end{bmatrix}$, where the additional observation is the normalized acceleration vector and $*$ represents each of the two reference frames. The closed-form solution of the rotation matrix estimate ${}^{B}_{I}\hat{\boldsymbol{R}}$ is found through

$$ {}^{B}_{I}\hat{\boldsymbol{R}} = \boldsymbol{U}\,\mathrm{diag}(1, 1, \det(\boldsymbol{U})\det(\boldsymbol{V}))\boldsymbol{V}^T \tag{13} $$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices, obtained from the SVD of a generic matrix $\boldsymbol{H} = \sum_{i=1}^{n_{obs}} \boldsymbol{w}_i\,{}^{B}\boldsymbol{O}_i\,{}^{I}\boldsymbol{O}_i^T$.

3.2.2. Rotation Matrix Observer

The magnetometers are very sensible to electromagnetic radiation and suffer from interferences that cause disruptions, so replacing them with the LiDAR can lead to a more accurate attitude determination tool. The LiDAR has its highest accuracy when it is used to estimate the movement about the Z axis, whereas the information it yields for the remaining axes is more susceptible to measurement errors. Thus, the accelerometer is the perfect addition to minimize those uncertainties. The kinematics of ${}^B_I\boldsymbol{R}$ are given by

$$ {}^B_I\dot{\boldsymbol{R}} = -S(\boldsymbol{\omega}){}^B_I\boldsymbol{R} \tag{14} $$

where $\boldsymbol{\omega}$ is the angular velocity, and the same system is replicated for the estimator such that

$$ {}^B_I\dot{\hat{\boldsymbol{R}}} = -S(\hat{\boldsymbol{\omega}}){}^B_I\hat{\boldsymbol{R}} \tag{15} $$

where $\hat{\boldsymbol{\omega}}$ is yet to be determined. It should be noted that these kinematics are built based solely on the integration of the angular velocity information from the IMU. Moreover, a variable $\tilde{\boldsymbol{R}} = {}^B_I\boldsymbol{R}{}^B_I\hat{\boldsymbol{R}}^T$ can be defined to represent the error between the true rotation matrix and its estimate.

Theorem 3.1. *Considering the system in* (14) *and its estimator in* (15)*, if* $\hat{\boldsymbol{\omega}}$ *is defined as*

$$ \hat{\boldsymbol{\omega}} = \boldsymbol{\omega} + K_{obs}\sum_{i=1}^{3} S({}^B\boldsymbol{O}_i)\tilde{\boldsymbol{R}}^T{}^B\boldsymbol{O}_i \tag{16} $$

where $K_{obs} > 0$ *is a tunable gain, then the equilibrium point* $\tilde{\boldsymbol{R}} = \boldsymbol{I}$*, from the error system, is almost globally asymptotically stable.*

Proof. The Lyapunov stability theory is based on the analysis of a function, generally associated with the energy of the system at hand. In this case, an expression related with the error in the rotation matrix such as

$$ V(\tilde{\boldsymbol{R}}) = \mathrm{tr}(\boldsymbol{I} - \tilde{\boldsymbol{R}}) \tag{17} $$

where $\boldsymbol{I}$ is the identity matrix, can be considered a reasonable candidate Lyapunov function to evaluate the stability of the system.

In order to guarantee the stability requirements for an equilibrium point $\tilde{\boldsymbol{R}} = \boldsymbol{I}$, the Lyapunov function needs to be positive definite, i.e. $V(\boldsymbol{I}) = 0$ and $V(\tilde{\boldsymbol{R}}) > 0 \forall \tilde{\boldsymbol{R}} \in \mathcal{SO}(3) \setminus \{\boldsymbol{I}\}$, and its derivative must be negative semi-definite, meaning that $\dot{V}(\boldsymbol{I}) = 0$ and $\dot{V}(\tilde{\boldsymbol{R}}) \leq 0 \forall \tilde{\boldsymbol{R}} \in \mathcal{SO}(3) \setminus \{\boldsymbol{I}\}$, and negative definite to reach asymptotic stability, verifying $\dot{V}(\boldsymbol{I}) = 0$ and $\dot{V}(\tilde{\boldsymbol{R}}) < 0 \forall \tilde{\boldsymbol{R}} \in \mathcal{SO}(3) \setminus \{\boldsymbol{I}\}$. Using (16), the derivative of the Lyapunov function becomes

$$ \dot{V}(\tilde{\boldsymbol{R}}) = -K_{obs}\sum_{i=1}^{3} {}^B\boldsymbol{O}_i^T\left(\boldsymbol{I} - \tilde{\boldsymbol{R}}^2\right){}^B\boldsymbol{O}_i \tag{18} $$

which fulfills the stability requirements in the domain $\tilde{\boldsymbol{R}} \in \mathcal{SO}(3) : V(\tilde{\boldsymbol{R}}) < 4 - \epsilon \forall 0 < \epsilon < 4$. In fact, it can be shown that the error system is almost globally asymptotically stable, i.e. it is stable and attractive except for a zero measure set of initial conditions [2]. □

The integration of the measurements from the gyroscopes suffers from drift over time. In the end, the estimate will converge to the rotation matrix obtained from a biased angular velocity $\overline{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{\delta\omega}$, which means the bias will still be present in ${}^B_I\hat{\boldsymbol{R}}$ and there will always be an error between the true and the estimated rotation matrix. This error will add up to the effect that the magnetic field's distortion causes on the data fusion with the magnetometers, within the IMU.

In order to determine what is the actual effect of the bias, in the behavior of the convergence process, the biased angular velocity is considered in the Lyapunov function. After the necessary mathematical developments, it can be shown that the Lyapunov function's derivative is negative definite while

$$ |\sin(\xi)| > \frac{1}{K_{obs}\lambda_{min}({}^B\overline{\boldsymbol{O}})}\|\boldsymbol{\delta\omega}\| \tag{19} $$

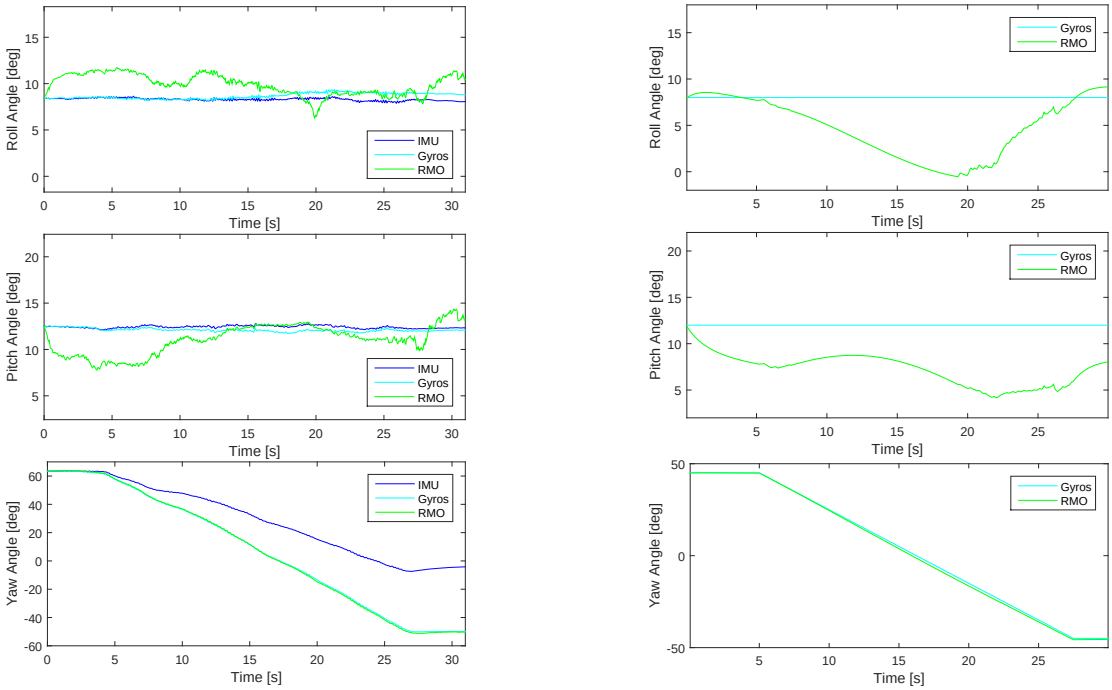
where ξ is the angle of rotation, when adopting the angle-axis representation for $\tilde{\boldsymbol{R}}$, and ${}^B\overline{\boldsymbol{O}} = {}^B\boldsymbol{O}{}^B\boldsymbol{O}^T - \mathrm{tr}({}^B\boldsymbol{O}{}^B\boldsymbol{O}^T)$ is a negative definite matrix. Thus, in the presence of a sufficiently small angular velocity bias, the estimation error has an ultimate bound [7]. It is worth noting that, even when the LiDAR is encountering only one edge, the accelerometer provides an additional measurement. This prevents the ${}^B\boldsymbol{O}$ matrix from having just one meaningful column, which would lead to $\lambda_{min}({}^B\overline{\boldsymbol{O}}) = 0$. Without bias and having at least two measurements, the derivative of the Lyapunov function will be negative definite, whereas when bias is added this can only happen when $|\sin(\xi)|$ is high enough to overpower the term with $\|\boldsymbol{\delta\omega}\|$, according to (19).

4. Performance Evaluation

Two scenarios were planned, with the intention of comparing the methods in equal circumstances. Because the attitude in the horizontal plane maintains a strong dependence with the accuracy of the edge's lengths, the first was conducted in null roll and pitch conditions, while the second considered reasonable values for these angles. This way, the impact of the sensor's noise and the effectiveness of the data treatment could be evaluated. The three methods presented in Section 3 were tested, both in a real and in a simulated environment.

The yaw estimator presented a good performance in both environments, reveling an adequate adjustment to real data and a proper simulation at the same time. However, despite being accurate and having a low computational load, this method can only describe the motion in the vertical axis, which led the creation of the full attitude estimation alternatives.

Considering the application of the Wahba's problem, the results of the experiments corresponded to the expectation, by achieving a close but noisy description of the attitude, when compared to the reference from the IMU. When it comes to the simulation, because only the kinematics associated with the rotation matrix were being modeled and the accelerometer was not available, the concerns initially raised with this method



(a) Experiment.

(b) Simulation.

Figure 4: Attitude of the quadcopter from the IMU, the gyroscopes and the rotation matrix observer.

were confirmed. The lack of sufficient measurements in the simulated environment, together with the artificial noise affecting the length of the edges, led to an unrealistic reproduction of the behavior observed during the experiments.

The rotation matrix observer obtained the best results, presented in Figure 4, where the attitude description obtained in the experiments can be seen as a filtered version of the previous method, maintaining a similar proximity to the reference. Figure 4a also highlights the offset mentioned in Section 3.2.2, regarding the attitude obtained from the IMU, with the effect of the magnetometers, and through the integration of the angular velocity from the gyroscopes. In the simulation, the absence of the accelerometer also degraded the response. However, it only limits the convergence in the horizontal plane's components of the attitude, since this method does not have the same restrictions regarding the minimum number of measurements as the Wahba's problem application. Therefore, the simulated environment presents a reasonable representation of the events carried out during the experiments.

It should be noted that the oscillation in the last two methods, in the roll and pitch angles, are directly related to the uncertainties while determining the length of the edges. Moreover, all methods presented a yaw motion description very similar among themselves and to what was planned in reality, unlike what was obtained with the IMU. For implementation purposes, the last method shows the most promising behavior.

5. Synthesis of Controllers

The final stage of this paper concerns the integration of the procedures in the vehicle and the design of controllers, set to achieve the initial goal of tracking a trajectory, defined relatively to a structure.

First, the code developed in MATLAB needed to be rewritten in C, to allow the vehicle's CPU to run it directly. This conversion process consisted of adding new code to what already worked as a whole before, so it needed to have a compatible execution time and blend seamlessly with the pre-existent framework.

After ensuring a successful code migration, a heading locking controller was developed, with the objective of maintaining a certain heading to a pier. This control system is shown in Figure 5 and it was built based on a first order model of the yaw kinematics, whose closed loop transfer function is given by

$$\frac{\omega_z(s)}{\omega_{z_d}(s)} = \frac{1}{T_{\dot{\psi}}s + 1} \tag{20}$$

where $T_{\dot{\psi}}$ is the time constant of the model. A simulated version of the controlled system was designed, obtaining a prompt response to a step input, with no overshoot or steady state error, using a P controller. For the real implementation, both an experimental setup with a single DOF in yaw and a real-time controller interface for tuning were conceived. Because the vehicle had a dead zone, the gain of the P controller needed to be high enough to compensate it and reach a satisfactory response, revealing a great match between the simulation and the real system.

The full position controller was then considered, where the simplification of the force balance that describes this system led to the error dynamics

$$\begin{cases} \dot{\tilde{\boldsymbol{p}}} = \dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_d = \boldsymbol{v} - \boldsymbol{v}_d \\ \dot{\tilde{\boldsymbol{v}}} = \dot{\boldsymbol{v}} - \dot{\boldsymbol{v}}_d = g\boldsymbol{e}_3 - \frac{T}{m}\boldsymbol{r}_3 - \dot{\boldsymbol{v}}_d \\ \dot{\tilde{\boldsymbol{r}}}_3 = \dot{\boldsymbol{r}}_3 - \dot{\boldsymbol{r}}_{3_d} = -S(\boldsymbol{r}_3)\boldsymbol{R}^T \begin{bmatrix} \omega_x & \omega_y & 0 \end{bmatrix}^T - \dot{\boldsymbol{r}}_{3_d} \end{cases} \tag{21}$$

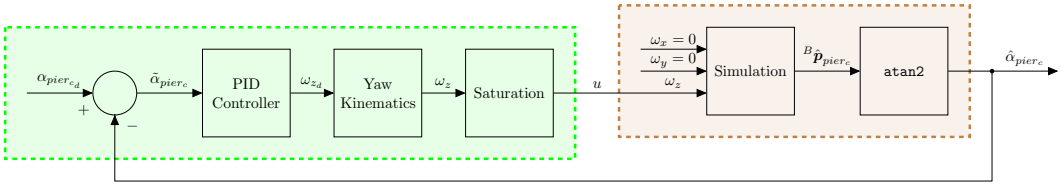


Figure 5: Block diagram of the heading locking controlled system distinguishing the controller (green), and the system plus sensor and processing (brown).

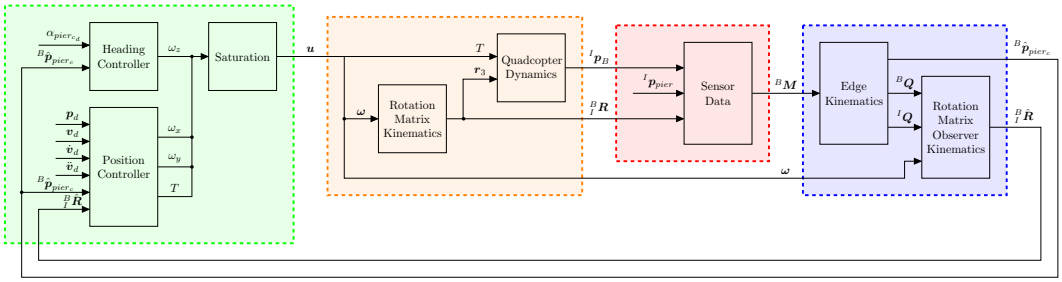


Figure 6: Block diagram of the trajectory tracking controlled system distinguishing the controller (green), the system (orange), the sensor (red), and the processing (blue).

where $\boldsymbol{p}$, $\boldsymbol{v}$, $\boldsymbol{r}_3$, and T are the true position, velocity, third column of the rotation matrix from $\{B\}$ to $\{I\}$, denoted here by $\boldsymbol{R}^T$ for simplicity of notation, and thrust input respectively, $\boldsymbol{p}_d$, $\boldsymbol{v}_d$, $\boldsymbol{r}_{3_d}$, and T_d are their desired values respectively, where the last two will be set shortly, m is the mass, and g is the gravitational acceleration. The approach to stabilize this nonlinear system, represented in $\{I\}$, consists of briefly dividing it into two simpler subsystems, introducing a new state $\boldsymbol{x} = \begin{bmatrix}\tilde{\boldsymbol{p}} & \tilde{\boldsymbol{v}}\end{bmatrix}^T$ for the first. For this task, it is assumed full access to the states and no saturation considerations are being made.

Theorem 5.1. *Considering the system in* (21)*, if* $T = T_d \boldsymbol{r}_3^T \boldsymbol{r}_{3_d}$*,* $\boldsymbol{r}_{3_d} = \frac{m}{T_d}\boldsymbol{F}$*,* $T_d = m\|\boldsymbol{F}\|$*, and an alternative input* $\overline{\boldsymbol{\omega}}$ *is defined as*

$$\overline{\boldsymbol{\omega}} = \frac{1}{\|\boldsymbol{F}\|} S(\boldsymbol{r}_{3_d})\dot{\boldsymbol{F}} - S(\boldsymbol{r}_3)\left[\frac{2T_d}{m}\left(\boldsymbol{P}_{12}\tilde{\boldsymbol{p}} + \boldsymbol{P}_{22}\tilde{\boldsymbol{v}}\right) + K_{\tilde{\boldsymbol{r}}_3}\tilde{\boldsymbol{r}}_3\right] \tag{22}$$

where $\boldsymbol{F} = g\boldsymbol{e}_3 - \dot{\boldsymbol{v}}_d + K_{\tilde{\boldsymbol{p}}}\tilde{\boldsymbol{p}} + K_{\tilde{\boldsymbol{v}}}\tilde{\boldsymbol{v}}$*,* $\dot{\boldsymbol{F}} = -\ddot{\boldsymbol{v}}_d + K_{\tilde{\boldsymbol{p}}}\dot{\tilde{\boldsymbol{p}}} + K_{\tilde{\boldsymbol{v}}}\dot{\tilde{\boldsymbol{v}}}$*,* $\boldsymbol{P}_{12}$ *and* $\boldsymbol{P}_{22}$ *are constant design matrices, and* $K_{\tilde{\boldsymbol{r}}_3} > 0$ *is the controller's third state gain, then the closed loop system is asymptotically stable.*

Proof. This strategy was combined with knowledge derived from Lyapunov's control theory, in order to obtain the function

$$V(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{r}}_3) = \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x} + \frac{1}{2}\tilde{\boldsymbol{r}}_3^T\tilde{\boldsymbol{r}}_3 \tag{23}$$

where $\boldsymbol{P}$ is a symmetric weighting matrix. Using (22), the derivative of this function corresponds to

$$\dot{V}(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{r}}_3) = -\boldsymbol{x}^T\boldsymbol{\Theta}\boldsymbol{x} - K_{\tilde{\boldsymbol{r}}_3}\|S(\boldsymbol{r}_3)\tilde{\boldsymbol{r}}_3\|^2 \tag{24}$$

where $\boldsymbol{\Theta}$ is a positive definite weighting matrix that establishes a relative importance between the errors in position and velocity and in $\boldsymbol{r}_3$. Therefore, (24) is composed of only negative definite terms and asymptotic stability is guaranteed, as originally intended. □

The constant design matrices $\boldsymbol{P}_{12}$ and $\boldsymbol{P}_{22}$ from Theorem 5.1 correspond to the blocks of $\boldsymbol{P}$ that are relevant to the control task. Using the Lyapunov Equation, they are given by

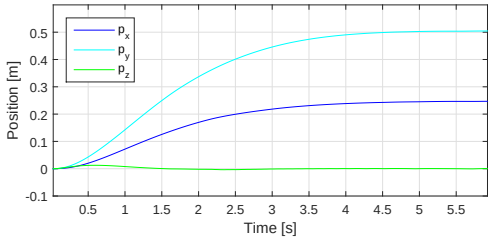
$$\boldsymbol{P}_{12} = \frac{f_\Theta}{2K_{\tilde{\boldsymbol{p}}}}\boldsymbol{I} \text{ and } \boldsymbol{P}_{22} = \frac{f_\Theta}{2K_{\tilde{\boldsymbol{v}}}}\left(1 + \frac{1}{K_{\tilde{\boldsymbol{p}}}}\right)\boldsymbol{I} \tag{25}$$

where its symmetry property was used, f_Θ is a scaling factor, and $K_{\tilde{\boldsymbol{p}}} > 0$ and $K_{\tilde{\boldsymbol{v}}} > 0$ are the controller's position and linear velocity gains respectively. The input to the system dynamics can now be defined as

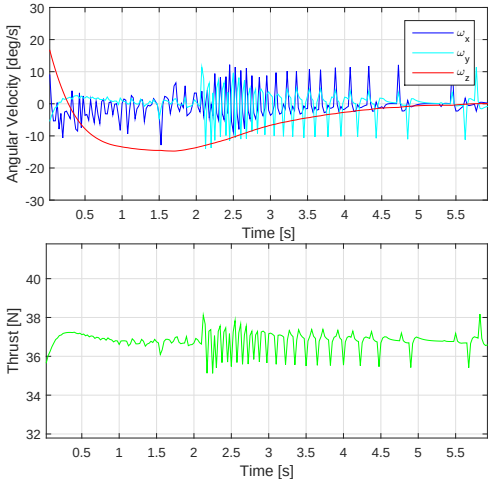
$$\boldsymbol{u} = \begin{bmatrix}\omega_x & \omega_y & \omega_z & T\end{bmatrix}^T \tag{26}$$

where $\begin{bmatrix}\omega_x & \omega_y & 0\end{bmatrix}^T = -\boldsymbol{R}S(\boldsymbol{r}_3)^2\overline{\boldsymbol{\omega}}$, given $\dot{\boldsymbol{r}}_3 = -S(\boldsymbol{r}_3)\overline{\boldsymbol{\omega}}$, and ω_z can be computed with the previous controller. The simulation of the new controlled system was implemented as presented in Figure 6, requiring a deep restructuring of the working simulated environment. The missing step was to tune parameters of the controller to obtain an optimal response. The attitude subsystem, concerning $\boldsymbol{r}_3$, constitutes the inner loop of the system, therefore it has to be faster than the position subsystem. With that in mind, the parameters need to be chosen with the goal of keeping the errors in position and velocity small, with respect to the attitude. Following that strategy accomplished that objective and yielded a swift response, with no overshoot or steady state error, as well as an acceptable control effort, shown in Figures 7a and 7b respectively.

After that, the experimental setup was upgraded to unlock the additional two DOFs, necessary for the proposed motion of the vehicle, and the interface was adjusted to accommodate the real-time tuning of the new parameters. For the implementation of this controller in the quadcopter, the trials are currently in preparation, for a complete testing, and experimental data can be found in future works.



(a) Position response.



(b) Control input.

Figure 7: Simulated output of the trajectory tracking controlled system to a step input $\boldsymbol{p}_d = [0.25, 0.5, 0]^T$ m, $\boldsymbol{v}_d = [0, 0, 0]^T$ m/s, and $\alpha_{pier_{c_d}} = 0°$.

6. Conclusions

This paper proposed a solution to the problem of laser-based control of rotary-wing UAVs, considering the entire process comprising the acquisition and treatment of the sensor's measurements, the development of methods to compute the relevant quantities to describe the motion of the vehicle, and the design and implementation of stable and effective controllers, based on that information.

Every stage yielded results that can be used on their own for a variety of other applications, given the modularity effort employed throughout this work. The most significant contributions from this work are the LiDAR data processing tools, the rotation matrix observer, and the trajectory tracking controller. Given their performance in both experiments and simulations, they could be brought to real world applications.

The directions for future work focus on two fronts, the first being the completion of the experimental testing of the trajectory tracking controller, currently in its final stages of planning. Afterwards, there are the considerations concerning the improvement of the developed algorithms or the approaches themselves. A significant amount of the effort behind this paper was put on treating the data from the LiDAR. The use of more accurate LiDAR sensors may lead to a more reliable processing of information. A proposal would be to also investigate what other available methodologies have to offer, in terms of an increase in performance. With the goal of making the transition of these controllers to real applications, further testing in even more adverse conditions may also be a point of interest. Furthermore, employing LiDAR sensors with a higher accuracy can also bring back the discussion on cylindrical structures and therefore open the solutions developed in this paper to a wider range of possibilities.

Acknowledgements

The author would like to deeply thank the guidance provided by Drs. Rita Cunha and Bruno Guerreiro, who were more than just supervisors during this project and helped push this work one step further. Additionally, my gratitude goes to Bruno Gomes and José Tojeira, the people that gave me all the technical support I could need to accomplish the intended goals.

References

[1] M. Atwater. Structural inspection reaches new heights while inspector stays grounded. Engineering.com, September 2013.

[2] S. Brás, R. Cunha, C. J. Silvestre, and P. J. Oliveira. Nonlinear attitude observer based on range and inertial measurements. *Control Systems Technology, IEEE Transactions on*, 21(5):1889–1897, 2013.

[3] U. E. Franke. Civilian drones: Fixing an image problem? ISN Blog, ETH Zurich, January 2015.

[4] G. Goyer and R. Watson. The laser and its application to meteorology. *Bulletin of the American Meteorological Society*, 44(9):564–575, 1963.

[5] C. A. Groenwall and M. C. Millnert. Vehicle size and orientation estimation using geometric fitting. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 412–423. International Society for Optics and Photonics, 2001.

[6] B. J. N. Guerreiro. *Sensor-Based Control and Localization of Autonomous Vehicles in Unknown Environments*. PhD thesis, Instituto Superior Técnico, Universidade de Lisboa, 2013.

[7] H. K. Khalil and J. Grizzle. *Nonlinear systems*, volume 3. Prentice hall New Jersey, 1996.

[8] F. L. Markley. Attitude determination using vector observations and the singular value decomposition. *The Journal of the Astronautical Sciences*, 36(3):245–258, 1988.

[9] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart. A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1929–1934. IEEE, 2005.

[10] T. Pavlidis and S. L. Horowitz. Segmentation of plane curves. *IEEE transactions on Computers*, 23(8):860–870, 1974.

[11] L. Shapiro and G. C. Stockman. Computer vision. 2001. *ed: Prentice Hall*, 2001.

[12] M. Teixidó, T. Pallejà, D. Font, M. Tresanchez, J. Moreno, and J. Palacín. Two-dimensional radial laser scanning for circular marker detection and external mobile robot tracking. *Sensors*, 12(12):16482–16497, 2012.

[13] B. P. Tice. Unmanned aerial vehicles - the force multiplier of the 1990s. *Airpower Journal*, 1991.