A Flight Altitude Estimator for Multirotor UAVs in Dynamic and Unstructured Indoor Environments

Hriday Bavle[1], Jose Luis Sanchez-Lopez[1], Alejandro Rodriguez-Ramos[1], Carlos Sampedro[1], and Pascual Campoy[1]

***Abstract*— A reliable estimation of the flight altitude in dynamic and unstructured indoor environments is an unsolved problem. Standalone available sensors, such as distance sensors, barometers and accelerometers, have multiple limitations in presence of non-flat ground surfaces, or in cluttered areas. To overcome these sensor limitations, maximizing their individual performance, this paper presents a modular EKF-based multi-sensor fusion approach for accurate vertical localization of multirotor UAVs in dynamic and unstructured indoor environments. The state estimator allows to combine the information provided by a variable number and type of sensors, including IMU, barometer and distance sensors, with the capabilities of sensor auto calibration and bias estimation, as well as a flexible configuration of the prediction and update stages. Several autonomous indoors real flights in unstructured environments have been conducted in order to validate our proposed state estimator, enabling the UAV to maintain the desired flight altitude when navigating over wide range of obstacles. Furthermore, it has been successfully used in IMAV 2016 competition. The presented work has been made publicly available to the scientific community as an open source software within the Aerostack[1] framework.**

I. Introduction

Nowadays, Unmanned Aerial Vehicles (UAVs) are highly demanded for their usage in civilian and military applications. Special interests are focused on multirotor UAVs due to their high maneuverability. Autonomous navigation of multirotor UAVs, both in outdoors and indoors scenarios is one of the main challenges that research community is still addressing. Several research efforts have been presented for outdoors autonomy [1], [2], [3] as well as for indoors autonomy [4], [5], [6], [7], [8], [9]. In outdoor scenarios, the localization and navigation problems are usually addressed fusing the information provided by the GNSS (Global Navigation Satellite System) with the measurements coming from the Inertial Measurement Unit (IMU). Localization and navigation problems indoors suffer a limitation of absence of information provided by the GNSS. Additionally, UAV navigation in small and unstructured indoor scenarios has to tackle with other kind of problems such as floor and ceiling effects.

For autonomous exploration of indoor scenarios, UAVs require accurate localization and mapping algorithms for navigation and control. The localization and mapping problem in such UAVs can be solved using a complete 3D SLAM using vision [10]. Jiaxin Li et al. [11] performed a survey of the several available SLAM algorithms, where they indicate that the 2D SLAM using range finder is a solved problem, whereas real time 3D SLAM using computer vision still remains an open problem

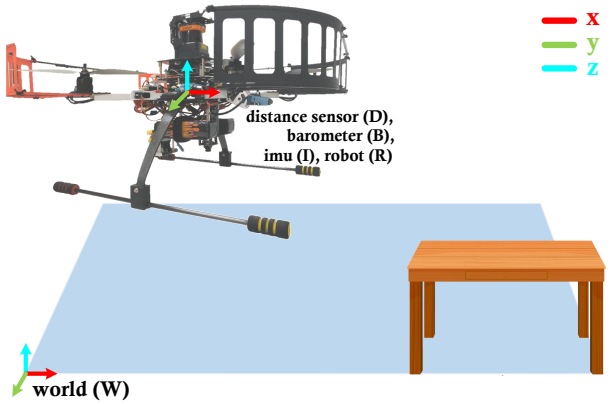

Fig. 1: The aerial platform used validating the proposed state estimator describing its frame of references.

As it can be assumed that the vertical and horizontal motion of multirotor UAVs can be decoupled, several research works have been focused on fusing the horizontal localization component (2D SLAM) with a vertical localization component. In order to maintain a constant desired flight altitude even when flying over non-flat surfaces (obstacles undetected by the horizontal localization component), it is essential to accurately estimate the flight altitude of the UAV.

The vertical localization in multirotor UAVs can be tackled using individual sensors like distance sensors (1D laser range finders, ultrasound sensors etc), barometer sensors or IMU (accelerometers). Directly using measurements from a single sensor like 1D laser range finder or ultrasound is possible for flat ground surfaces, as the flight altitude can be directly measured with respect to the ground, nevertheless in presence of non-flat surfaces, as in every indoor environment, the flight altitude cannot be directly measured with respect to the ground. Altitude measurements provided by a barometer can be very noisy as well as very biased in indoor environments due to the pressure changes that occur because of the ground, ceiling, and wall effects. Altitude calculated directly from accelerometers, integrating the accelerations in z-axis has a a huge drift over time due to the huge measurement noise

[1]All authors are with the Computer Vision and Aerial Robotics (CVAR) Group, Centre for Automation and Robotics (UPM-CSIC), Calle José Gutiérrez Abascal 2, Madrid (Spain). hriday.bavle@upm.es

[1]www.aerostack.org

resulting from the vibrations of the UAV.

In order to overcome the drawbacks of each sensor, being able to robustly and accurately estimate the flight altitude of a multirotor UAV in indoor environments, this paper presents a modular multi-sensor fusion state estimator, capable to fuse measurements from any number of different sensors such as IMUs, barometers and distance sensors. Additionally, the proposed state estimator is capable of estimating the elevation of the surface below in presence of highly unstructured obstacles, allowing the navigation over non-flat ground surfaces in indoor environments.

The remainder of the paper is structured as follows, Sect. II analyzes some related work. Sect. III presents the general features of our proposed modular state estimator, providing details of its components in Sect. IV. Several experiments that demonstrate the good performance of our proposed state estimator are shown in Sect. V. Finally, Sect. VI concludes the paper and points out some future work lines.

II. Related Work

In multiple works that demonstrate a fully autonomous approach in indoors environments, such as [4], [5], [6], [12], horizontal localization and mapping was achieved fusing visual information from ArUco visual markers [13] with orientations from the IMU and the vertical localization was achieved using an ultrasound sensor. These works provided no capabilities of estimating the elevation of the surface below the UAV and hence proved efficient only in presence of flat surfaces with zero elevation. In another fully autonomous work [14], used in International Aerial Robotics Competition (IARC) with several iRobots moving randomly below the UAV, the localization was performed with a 2D SLAM using several computer vision algorithms providing the horizontal localization and a px4flow sensor providing the vertical localization. This work did not estimate the elevation of the surface below the UAV containing all the iRobots for maintaining a stable flight altitude.

Opromolla et al. [15] performs 2D localization using laser sensor on a custom hardware platform which was never tested in real flights, hence never fusing the horizontal localization component with a vertical localization component. Grzonka et al. [8] proposes an approach of altitude filtering using fusion of IMU and laser data estimating the elevation of the surface below and thus the flight altitude of the UAV. This work lacks proper explanation of the approach as well does not show efficient results in presence of highly unstructured obstacles. Grzonka et al. [9] proposes an improved approach for filtering the altitude using two stage system of Kalman filters, the first stage is used to track the altitude z and the vertical velocity v_z combining inertial measurements, altitude measurements and already mapped levels under the UAV. And in the second stage the Kalman filter is used to estimate the elevation of the levels currently measured by the UAV. This work assumes that the surface under the UAV is piecewise constant and does not account for discontinuity in the surface elevation. Ivan Dryanovski et al. [16] proposes an approach for robust altitude estimation of the UAV by deflecting the rays from a horizontally mounted laser to the ground using a mirror. A histogram is computed of 20 altitude readings with a bin size of 2 cm. After detecting a peak of the histogram, an average of the readings belonging to that bin is computed thus estimating the flight altitude of the UAV as well the estimated floor level. This technique does not consider the noise in the measurements of the laser sensor when detecting an edge on the surface below, which can cause wrong estimation of the flight altitude and result in huge accumulation error over time.

III. Proposed Multi-Sensor Fusion State Estimator

In this work we present a modular state estimator based on an extended Kalman filter (EKF).

Our proposed state estimator, represented in Fig. 2, combines measurements from several sensors, including IMU, barometer and distance sensor, and estimates the robot states $\hat{\boldsymbol{x}}_{\boldsymbol{R}}$ and the ground object state $\hat{\boldsymbol{x}}_{\boldsymbol{G}}$ (elevation of the ground surface).

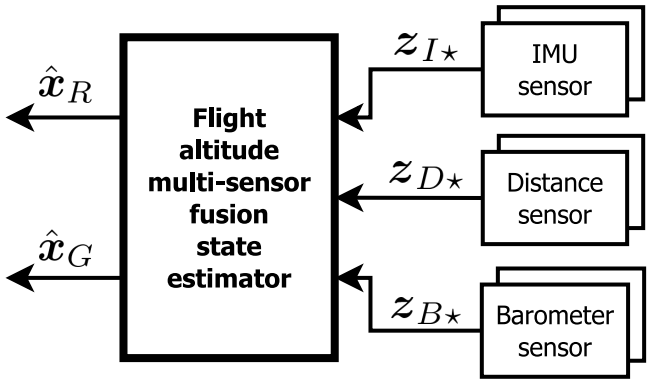

Fig. 2: The inputs and outputs of the proposed state estimator.

The proposed state estimator has the following main features:

- Modularity on the number of sensors used, what allows a usage of a variable number and kind of sensors for the state estimation.
- Auto calibration of sensors, allowing to estimate the internal state of some sensors (e.g. biases on the measurement), providing a more robust and accurate state estimation.
- Synchronous prediction of the estimated state: Whenever an estimation of the state is required, the prediction stage is executed.
- Asynchronous update of the estimated state based on the available measurements: Whenever a measurement arrives to the state estimator, both prediction and update stage are executed consecutively, allowing the accurate usage of sensors with different measurement rate.

As stated before, the proposed state estimator is modular, being its available components, deeply described in Sect. IV, the following: robot (R), ground object (G), IMU sensor ($I\star$), barometer sensor ($B\star$) and distance sensor ($D\star$).

In the following sections, the combination between the different components is described in detail. Sect. III-A, explains the state and process model composition taking into account the designed modularity; and Sect. III-C, indicates how the available measurements are gathered, forming the measurement model. Additionally, Sect. III-B and Sect. III-D present the widely known extended Kalman filter equations used for the prediction stage and update stage respectively. For the purpose of this paper, $\sin(x)$ and $\cos(x)$ are abbreviated as $s(x)$ and $c(x)$ respectively.

A. State and process model

The state vector, $\boldsymbol{x}$, is formed by the state vector of all the components, similar to the process model noise $\boldsymbol{n}$, being:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_R \\ \boldsymbol{x}_G \\ \boldsymbol{x}_{I\star} \\ \boldsymbol{x}_{B\star} \\ \boldsymbol{x}_{D\star} \end{bmatrix} \qquad \boldsymbol{n} = \begin{bmatrix} \boldsymbol{n}_R \\ \boldsymbol{n}_G \\ \boldsymbol{n}_{I\star} \\ \boldsymbol{n}_{B\star} \\ \boldsymbol{n}_{D\star} \end{bmatrix}$$

where the process model noise has an associated covariance matrix $\boldsymbol{T}$.

In the same way, the discrete time process model formed by all the individual discrete time process model of all the components, being:

$$\boldsymbol{x}(k) = \boldsymbol{f}\left(\boldsymbol{x}(k-1), \Delta t, \boldsymbol{n}\right)$$

where Δt is the change in time between k and $k+1$.

The Jacobian matrices of the discrete time process model are given by:

$$\boldsymbol{F}_x = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \qquad \boldsymbol{F}_n = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{n}}$$

B. Prediction Stage

During the prediction stage, the state $\boldsymbol{x}$ is estimated in the current instant time k, taking into account its previous estimate in $k-1$, and based on the process model, following the EKF equations:

$$\hat{\boldsymbol{x}}(k|k-1) = \boldsymbol{f}(\hat{\boldsymbol{x}}(k-1|k-1), \Delta t) \tag{1}$$

$$\boldsymbol{P}(k|k-1) = \boldsymbol{F}_x \cdot \boldsymbol{P}(k-1|k-1) \cdot {\boldsymbol{F}_x}^T + \boldsymbol{F}_n \cdot \boldsymbol{T} \cdot {\boldsymbol{F}_n}^T \tag{2}$$

being $\boldsymbol{P}$ the covariance matrix of the estimated state $\hat{\boldsymbol{x}}$.

C. Measurements and measurement model

The available measurements $\boldsymbol{z} = \{\boldsymbol{z}_{I\star},\ \boldsymbol{z}_{B\star},\ \boldsymbol{z}_{D\star}\}$ at a particular instant time k, determine the measurement model, that changes dynamically, being:

$$\boldsymbol{z}(k) = \boldsymbol{h}\left(\boldsymbol{x}(k), \boldsymbol{\epsilon}\right)$$

where $\boldsymbol{\epsilon}$ is the measurement model noise, with an associated covariance matrix $\boldsymbol{R}$.

The Jacobian matrices of the measurement model are given by:

$$\boldsymbol{H}_x = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \qquad \boldsymbol{H}_\epsilon = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\epsilon}}$$

D. Update Stage

During the update stage, the state is modified taking into account its prediction, the measurement model and the available measurements, following the EKF equations.

First of all, the measurement residual $\boldsymbol{v}$ is computed as:

$$\boldsymbol{v} = \tilde{\boldsymbol{z}}(k) - \boldsymbol{h}\left(\hat{\boldsymbol{x}}(k|k-1)\right) \tag{3}$$

with an associated covariance matrix:

$$\boldsymbol{S} = \boldsymbol{H}_x \cdot \boldsymbol{P}(k|k-1) \cdot {\boldsymbol{H}_x}^T + \boldsymbol{H}_\epsilon \cdot \boldsymbol{R} \cdot {\boldsymbol{H}_\epsilon}^T \tag{4}$$

The near optimal kalman gain K is then computed as

$$\boldsymbol{K} = \boldsymbol{P}(k|k-1) \cdot {\boldsymbol{H}_x}^T \cdot \boldsymbol{S}^{-1} \tag{5}$$

And the estimated state is updated as:

$$\boldsymbol{x}(k|k) = \boldsymbol{x}(k|k-1) + \boldsymbol{K} \cdot \boldsymbol{v} \tag{6}$$

$$\boldsymbol{P}(k|k) = (\boldsymbol{I}_n - \boldsymbol{K} \cdot \boldsymbol{H}_x) \cdot \boldsymbol{P}(k|k-1) \tag{7}$$

n being the dimension of the state $\boldsymbol{x}$, and $\boldsymbol{I}_n$ the identity matrix of dimension $n \times n$.

IV. Components of the proposed state estimator

This section describes the components of the proposed state estimator.

All the sensors are assumed to be located in the center of the robot, with their reference frame being aligned with the robot reference frame as shown in Fig. 1.

A. Aerial Robot

The aerial robot state $\boldsymbol{x}_R$ includes:

- Vertical coordinate of the robot reference frame in world coordinates, ${t_z}_R^W$.
- Vertical coordinate of the velocity of the robot with respect to world, in world coordinates, ${v_z}_{R|W}^W$.
- Vertical coordinate of the acceleration of the robot with respect to world, in world coordinates, ${a_z}_{R|W}^W$.
- Attitude of the robot reference frame in world coordinates, $\boldsymbol{\theta}_R^W = \begin{bmatrix} {\theta_x}_R^W,\ {\theta_y}_R^W,\ {\theta_z}_R^W \end{bmatrix}^T$, as Euler angles.
- Angular velocity of the robot with respect to robot in robot coordinates, $\boldsymbol{\omega}_{R|W}^R = \begin{bmatrix} {\omega_x}_{R|W}^R,\ {\omega_y}_{R|W}^R,\ {\omega_z}_{R|W}^R \end{bmatrix}^T$.

The aerial robot is assumed to change its vertical acceleration and its angular velocity slowly, permitting the usage of a constant vertical acceleration and constant angular velocity as process model. Thus this model is only valid when assuming the aerial robot is a multirotor. Other important consequence of this slow motion, is that their pitch and roll angles, can be approximated as zero when required: ${\theta_x}_R^W \approx 0$ and ${\theta_y}_R^W \approx 0$.

According to the kinematics of a moving body, relationship between Euler angles and angular velocities can be given as:

$$\begin{bmatrix} {\omega_x}_{R|W}^R \\ {\omega_y}_{R|W}^R \\ {\omega_z}_{R|W}^R \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\mathrm{s}({\theta_y}_R^W) \\ 0 & \mathrm{c}({\theta_x}_R^W) & \mathrm{s}({\theta_x}_R^W)\cdot\mathrm{c}({\theta_y}_R^W) \\ 0 & -\mathrm{s}({\theta_x}_R^W) & \mathrm{c}({\theta_x}_R^W)\cdot\mathrm{s}({\theta_y}_R^W) \end{bmatrix} \begin{bmatrix} \dot{{\theta_x}}_R^W \\ \dot{{\theta_y}}_R^W \\ \dot{{\theta_z}}_R^W \end{bmatrix} \tag{8}$$

As the ${\theta_x}_R^W \approx 0$ and ${\theta_y}_R^W \approx 0$ relationship between the Euler angles and angular velocities can be reduced as:

$$ {\omega_x}_{R|W}^R \approx \dot{{\theta_x}}_R^W \tag{9} $$

$$ {\omega_y}_{R|W}^R \approx \dot{{\theta_y}}_R^W \tag{10} $$

$$ {\omega_z}_{R|W}^R \approx \dot{{\theta_z}}_R^W \tag{11} $$

Therefore, the discrete time process model of the robot, $\boldsymbol{f}_R$, is simplified to:

$$ {t_z}_R^W(k) = {t_z}_R^W(k-1) + {v_z}_{R|W}^W(k-1) \cdot \Delta t \tag{12} $$

$$ + \frac{\Delta t^2}{2} \cdot {a_z}_{R|W}^W(k-1) \tag{13} $$

$$ {v_z}_{R|W}^W(k) = {v_z}_{R|W}^W(k-1) + {a_z}_{R|W}^W(k-1) \cdot \Delta t \tag{14} $$

$$ {a_z}_{R|W}^W(k) = {a_z}_{R|W}^W(k-1) + n_{a_z} \tag{15} $$

$$ \boldsymbol{\theta}_R^W(k) \approx \boldsymbol{\theta}_R^W(k-1) + \boldsymbol{\omega}_{R|W}^R(k-1) \cdot \Delta t \tag{16} $$

$$ \boldsymbol{\omega}_{R|W}^R(k) = \boldsymbol{\omega}_{R|W}^R(k-1) + \boldsymbol{n}_{\boldsymbol{\omega}} \tag{17} $$

where the noise of the process model of the robot, $\boldsymbol{n}_R$, with an associate covariance matrix $\boldsymbol{T}_R$, is given as:

$$ \boldsymbol{n}_R = \begin{bmatrix} n_{a_z} \\ \boldsymbol{n}_{\boldsymbol{\omega}} \end{bmatrix} \tag{18} $$

B. Ground object

The state of the ground object $\boldsymbol{x}_G$ includes:

- Altitude of the ground object in world coordinates, ${t_z}_G^W$.

The world reference frame is assumed to be in the lower location of the environment in the z-axis coordinate, that means, that the altitude of the ground object in world coordinates is always positive.

Being, the discrete time process model of the ground object, $\boldsymbol{f}_G$, given by:

$$ {t_z}_G^W(k) = \begin{cases} {t_z}_G^W(k-1) + n_{z_G}, & \text{if } {t_z}_G^W(k-1) > 0 \\ 0 + n_{z_G}, & \text{if } {t_z}_G^W(k-1) \leq 0 \end{cases} \tag{19} $$

where the noise of the process model of the ground object, $\boldsymbol{n}_G$, with an associate covariance matrix $\boldsymbol{T}_G$, is given as:

$$ \boldsymbol{n}_G = \begin{bmatrix} n_{z_G} \end{bmatrix} \tag{20} $$

C. IMU sensor: accelerometer, gyro and magnetometer

The IMU sensor considered in our state estimator is an intelligent component formed by an accelerometer, a gyro and a magnetometer, with an internal state estimator that does not provide the raw measurements, but a filtered estimation of them. This consideration is not too strong as every autopilot hardware used for multirotor platforms incorporates this kind of IMU sensor.

The acceleration measurement is assumed to have a bias modeled as a random process, as it randomly changes its value with the time.

The state of the IMU sensor, $\boldsymbol{x}_{I\star}$, includes:

- Bias on the vertical measurement of the acceleration, $b_{a_{z\star}}$.

Being the discrete time process model of the IMU sensor, $\boldsymbol{f}_{I\star}$, given by:

$$ b_{a_{z\star}}(k) = b_{a_{z\star}}(k-1) + n_{b_{a_{z\star}}} \tag{21} $$

where the noise of the process model of the IMU sensor, $\boldsymbol{n}_{I\star}$, with an associate covariance matrix $\boldsymbol{T}_{I\star}$, is given as:

$$ \boldsymbol{n}_{I\star} = \begin{bmatrix} n_{b_{a_{z\star}}} \end{bmatrix} \tag{22} $$

The IMU sensor provides the following measurements, $\boldsymbol{z}_{I\star}$:

- Acceleration measured, $\boldsymbol{z}_a = \begin{bmatrix} z_{ax},\ z_{ay},\ z_{az} \end{bmatrix}^T$.
- Attitude measured, $\boldsymbol{z}_\theta = \begin{bmatrix} z_{\theta x},\ z_{\theta y},\ z_{\theta z} \end{bmatrix}^T$.
- Angular velocity measured, $\boldsymbol{z}_\omega = \begin{bmatrix} z_{\omega x},\ z_{\omega y},\ z_{\omega z} \end{bmatrix}^T$.

Nevertheless, only the following measurements are taken into account: z_{az}, $z_{\theta x}$, $z_{\theta y}$, $z_{\omega x}$, and $z_{\omega y}$.

Being its measurement model, $\boldsymbol{h}_{I\star}$, given by:

- Acceleration measured:

$$ \boldsymbol{z}_a = \boldsymbol{a}_{R|W}^R - \boldsymbol{g}^R + \boldsymbol{b}_a + \boldsymbol{\epsilon}_a \tag{23} $$

$$ = \left(\boldsymbol{R}_R^W\right)^T \cdot \left(\boldsymbol{a}_{R|W}^W - \boldsymbol{g}^W\right) + \boldsymbol{b}_a + \boldsymbol{\epsilon}_a \tag{24} $$

where $\boldsymbol{R}_R^W$ is the 3D rotation matrix derived from the Euler angles.
Since we consider ${\theta_x}_R^W \approx 0$ and ${\theta_y}_R^W \approx 0$ the rotation matrix can be approximated as:

$$ \boldsymbol{R}_R^W \approx \begin{bmatrix} \mathrm{c}({\theta_z}_R^W) & -\mathrm{s}({\theta_z}_R^W) & 0 \\ \mathrm{s}({\theta_z}_R^W) & \mathrm{c}({\theta_z}_R^W) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{25} $$

As only the vertical component is taken into account:

$$ z_{az} \approx {a_z}_{R|W}^W - {g_z}^W + b_{a_{z\star}} + \epsilon_{az} \tag{26} $$

- Attitude measured:

$$ \boldsymbol{z}_\theta = \boldsymbol{\theta}_R^W + \boldsymbol{\epsilon}_\theta \tag{27} $$

And only two components are taken into account as:

$$ z_{\theta x} = {\theta_x}_R^W + \epsilon_{\theta x} \tag{28} $$

$$ z_{\theta y} = {\theta_y}_R^W + \epsilon_{\theta y} \tag{29} $$

- Angular velocity measured:

$$ \boldsymbol{z}_\omega = \boldsymbol{\omega}_{R|W}^R + \boldsymbol{\epsilon}_\omega \tag{30} $$

And only two components are taken into account as:

$$ z_{\omega x} = {\omega_x}_{R|W}^R + \epsilon_{\omega x} \tag{31} $$

$$ z_{\omega y} = {\omega_y}_{R|W}^R + \epsilon_{\omega y} \tag{32} $$

where the noise of the measurement model of the IMU sensor $\boldsymbol{\epsilon}_{I\star}$, with an associate covariance matrix $\boldsymbol{R}_{I\star}$, is given as:

$$\boldsymbol{\epsilon}_{I\star} = \begin{bmatrix} \epsilon_{a\,z} \\ \epsilon_{\theta x} \\ \epsilon_{\theta y} \\ \epsilon_{\omega x} \\ \epsilon_{\omega y} \end{bmatrix} \tag{33}$$

D. Barometer sensor

The Barometer sensor considered in our state estimator is an intelligent component that provides direct measurements of the flight altitude instead of raw pressure measurements thanks to an internal state estimator. As with the IMU sensor, this consideration is not too strong since every autopilot hardware used for multirotor platforms incorporates this kind of sensor.

Similar to the acceleration measurement, the barometer measurement is assumed to have a bias modeled as a random process, as it randomly changes its value with the time.

The state of the Barometer sensor, $\boldsymbol{x}_{B\star}$, includes:

- Bias on the vertical measurement of the barometer, $b_{bz\star}$.

Being the discrete time process model of the barometer sensor, $\boldsymbol{f}_{B\star}$, given by:

$$b_{bz\star}(k) = b_{bz\star}(k-1) + n_{b_{bz\star}} \tag{34}$$

where the noise of the process model of the barometer sensor, $\boldsymbol{n}_{B\star}$, with an associate covariance matrix $\boldsymbol{T}_{B\star}$, is given as:

$$\boldsymbol{n}_{B\star} = \begin{bmatrix} n_{b_{bz\star}} \end{bmatrix} \tag{35}$$

The barometer sensor provides the following measurements, $\boldsymbol{z}_{B\star}$:

- Vertical coordinate in world reference frame, z_{bz}

Being its measurement model, $\boldsymbol{h}_{B\star}$, given by:

$$z_{bz} = t_{z\,R}^{\;W} + b_{bz\star} + \epsilon_{bz} \tag{36}$$

where the noise of the measurement model of the barometer sensor $\boldsymbol{\epsilon}_{B\star}$, with an associate covariance matrix $\boldsymbol{R}_{B\star}$, is given as:

$$\boldsymbol{\epsilon}_{B\star} = \begin{bmatrix} \epsilon_{bz} \end{bmatrix} \tag{37}$$

E. Distance sensor

The distance sensor does not incorporate any state, and therefore, its does not add any process model.

The distance sensor provides the following measurement, $\boldsymbol{z}_{D\star}$:

- Distance measured to the ground object, z_d.

Being its measurement model, $\boldsymbol{h}_{D\star}$, given by:

$$z_d = \frac{t_{z\,R}^{\;W} - t_{z\,G}^{\;W}}{\cos(\theta_{x\,R}^{\;W}) \cdot \cos(\theta_{y\,R}^{\;W})} + \epsilon_d \tag{38}$$

where the noise of the measurement model of the distance sensor $\boldsymbol{\epsilon}_{D\star}$, with an associate covariance matrix $\boldsymbol{R}_{D\star}$, is given as:

$$\boldsymbol{\epsilon}_{D\star} = \begin{bmatrix} \epsilon_d \end{bmatrix} \tag{39}$$

V. Experiments and Results

In this section, the experiments that have been conducted in order to validate the proposed approach are presented. The aim of the validation is twofold. First, to demonstrate the capabilities of the designed state estimator in presence of several obstacles located on the ground and second, to evaluate the behavior of the proposed system in presence of unexpected sensor measurements.

This section is divided into two subsections, in Sect. V-A the entire experimental setup including the sensors and computational hardware as well as the software used is explained. In Sect. V-B the results obtained in real flights are shown and analyzed.

A. Experimental Setup

1) System Setup: Fig 1 represents the custom aerial platform used for validating the proposed state estimator. The autopilot utilized in the real flight experiments was Pixhawk. Three different onboard sensors were used in the presented state estimator: a 1D laser range finder *Lightware SF/10A* with a minimum and maximum range of 0 and 25 m respectively, a resolution of 1 cm and an accuracy of 0.05 m in the absence of sensor noise; a 9-axis IMU integrated within the Pixhawk autopilot; and a barometer also integrated with the Pixhawk autopilot with an altitude resolution of 10 cm.

The entire computation was performed on an on-board intel-NUC 6i5SYK computer. All the above mentioned sensors communicate with the on-board computer using USB communication.

The software modules were developed in C++, using C++ 11 standards. Regarding the software system, Aerostack software framework was utilized. A full description of Aerostack architecture and its components is out of scope of this paper, and can be reviewed at [4]. The complete description of the components used for the autonomous flights can be found at [17].

2) Real Flights configuration: In order to validate the approach, several fully autonomous real flights were performed in different challenging indoor scenarios.

- *Navigation in unknown and unstructured indoor environment:* The scenario, with a size of 5 m $\times$ 9 m, consisted of several obstacles with height smaller than the flight altitude of the UAV in order to be undetected by the 2D SLAM. The obstacles consisted of several tables with a uniform surface. Several chairs were placed with random discontinuity in their surface increasing the noise in the measurements provided by the 1D laser range finder.
- *Take-off and land from a platform:* The platform, located in an indoor scenario, with a size of 5 m $\times$ 9 m, was a square sized platform with an area of 140 cm^2 and height of 47 cm. In this experiment the taking-off and landing was performed over the platform.

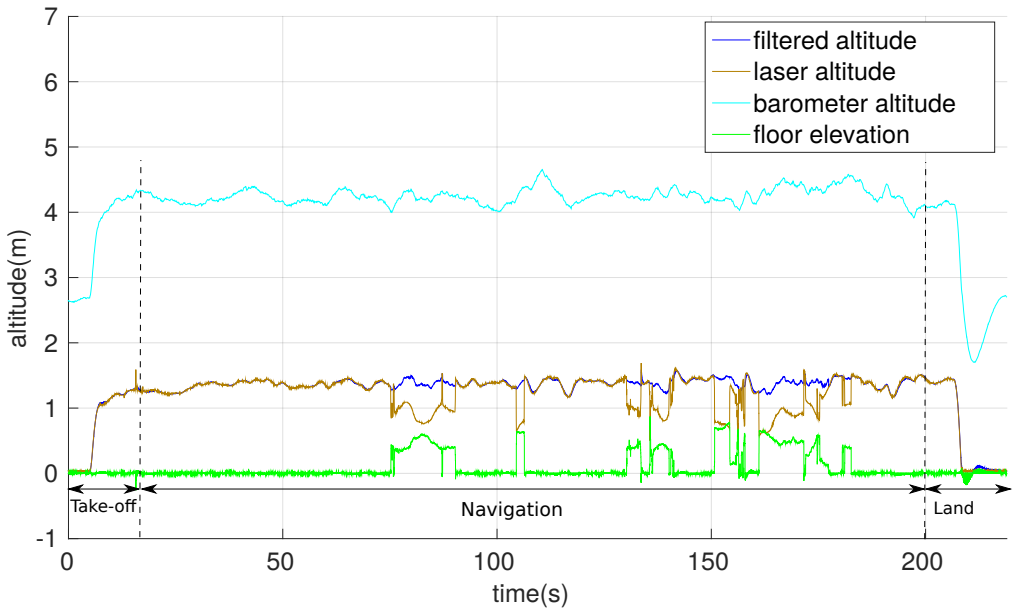

Fig. 3: Results obtained while performing take-off, navigation and land tasks during the execution of a real flight.

B. Results and Discussions

In this section, all the results obtained during the execution of the real flight experiments explained in Sect. V-A.2 are described in detail. In both the experiments the commanded flight altitude to the UAV is 140 cm.

- *Navigation in unknown and unstructured indoor environment:*
  Fig. 3 represents the results of the complete real flight performed during this experiment. The experiment is divided into three main phases: 1.Take-off 2.Navigation and 3.Land.
  - *Take-off* and *land phase.* As can be observed in Fig. 3 between time instances $t = 0$ s to $t = 10$ s the UAV performs the take-off task from a flat ground surface, whereas from instances $t = 210$ s to $t = 220$ s the land task is executed. It can be noticed that, the state estimator is able to accurately estimate the flight altitude without any delay during these phases (which is essential for fast take-off and land maneuvers) when compared to the altitude provided by the 1D laser range finder. As seen in Fig. 3, the barometer provides biased flight altitude measurements. Due to its auto calibration feature, the state estimator accurately estimates the barometer bias and estimates the accurate flight altitude.
  - *Navigation phase.* As highlighted in Fig. 4, the navigation is performed over a wide variety of obstacles in terms of size, height, 3D shape and surface. Despite the heterogeneity of the configuration, the state estimator is capable of providing a robust flight altitude estimate. Even in presence of obstacles composed of several uneven surfaces like chairs where the 1D laser range finder measurements have higher noise (see time instances between $t = 130$ s to $t = 140$ s and time instances $t = 150$ to $t = 160$ s in Fig. 4) the state estimator provides a good estimate of the flight altitude.

  Through this experiment, we demonstrate that despite of several noisy sensor measurements at several time instances, the state estimator is able to estimate the accurate flight altitude in order to maintain the desired flight altitude.
- *Take-off and land on a platform:*
  In this experiment, an important variation in terms of the configuration of the proposed scenario is presented. In this case, the UAV is required to take-off and land from an elevated surface (platform). The experiment can be divided as:
  - *Take-off from the platform.* As shown in Fig. 5 during time instances $t = 0$ s and $t = 20$ s the UAV which is landed on the platform, performs the take-off task. The state estimator is initialized with both, the initial flight altitude and the ground obstacle height equal to the height of the platform (47 cm). Although the measurements provided by the 1D laser altimeter are with respect to the platform when the UAV is on top of the platform (see take-off phase in Fig. 5), the state estimator is able to deal with the initial height offset introduced by the platform, providing an accurate flight altitude estimate on top of the platform.
  - *Navigation Phase.* When comparing the estimated altitude with the 1D laser range finder altitude, it can be observed in Fig. 5 that, after time instant $t = 40$ s the state estimator accurately estimates

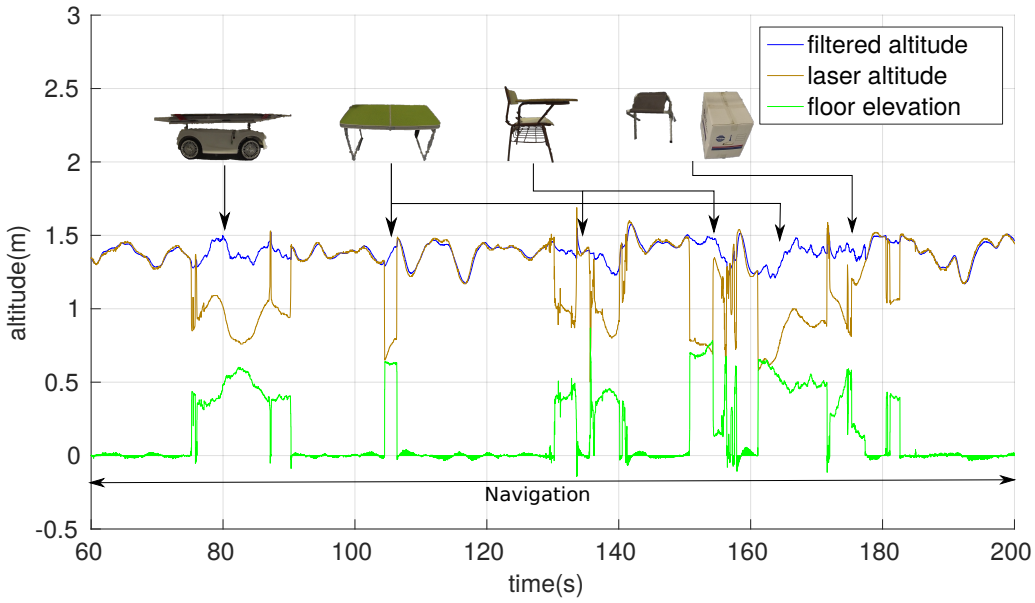

Fig. 4: Navigation phase of the same flight represented in Fig. 3, when the UAV is flying over several obstacles.

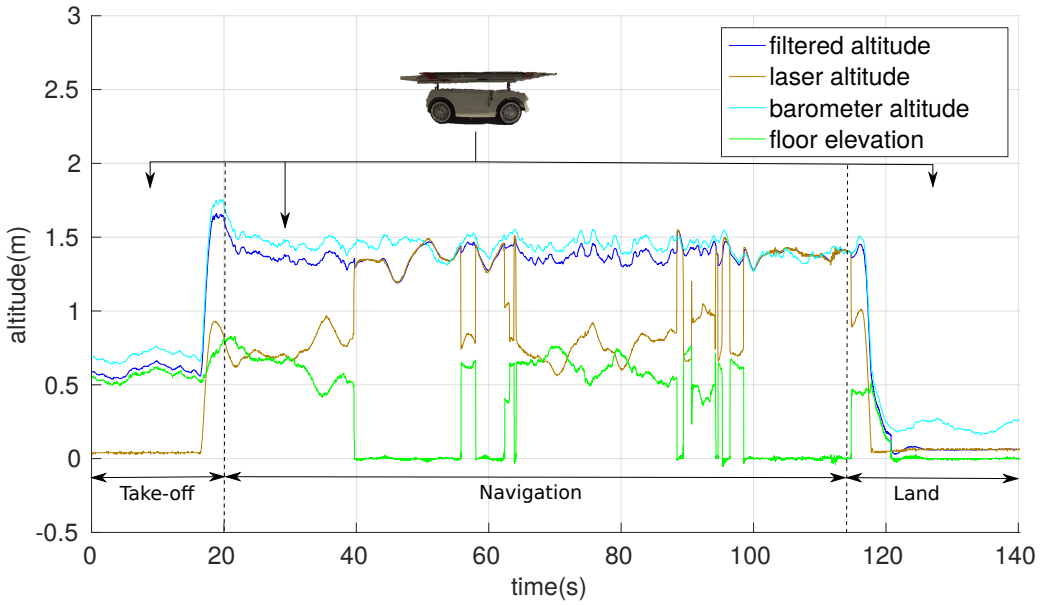

Fig. 5: Results obtained during the execution of a complete flight, where take-off and land tasks were performed over an elevated ground surface (platform).

the flight altitude as the UAV exits the platform below it and navigates over the obstacles without any divergence in its state.

– *Land on the platform.* Between time instances $t = 110$ s and $t = 120$ s the UAV performs the task of land on the platform (see Fig. 5). The land task consists of two stages the 1.Approach stage where the UAV starts descending on the platform and 2.The fast landing stage (a free fall of the UAV on the platform), is triggered if the current flight altitude is less than the threshold altitude to land. It can be highlighted through Fig. 5 that, during the approach stage the state estimator estimates accurately the height of the platform but due to high velocity and acceleration constraints during the fast landing stage, the estimated flight altitude drops to zero instead of the height of the platform. This error has no effect in any of the functionalities of the system, as land being the last task to be executed during a mission.

Through this experiment we demonstrate the good and acceptable performance of the state estimator to esti-

mate the flight altitude when taking-off and landing from elevated ground surfaces.

We compare the estimated flight altitude with the altitude measured by the 1D laser altimeter over flat ground surfaces to achieve an average error less than 10 cm. We refer the reader to a video[2] demonstrating the capabilities of the proposed state estimator. Additionally, our proposed state estimator was successfully used in IMAV 2016 indoors competition for solving the problem of vertical localization of the UAV. Further details can be found at [17].

VI. Conclusions and Future Work

In this paper we presented a robust multi-sensor fusion state estimator for accurately estimating the flight altitude of a multirotor UAV as well as the elevation of the floor below it in highly unstructured indoors environments.

Our proposed EKF-based state estimator has the following features: (1) modularity in the number of sensors used, in order to fuse the information provided by variable number and type of sensors, including IMU sensors, barometer sensors and distance sensors, thus improving robustness; (2) auto calibration of sensors, which improves the accuracy of the state estimation; (3) synchronous prediction of the estimated state whenever it is needed by the controller; and (4) asynchronous update of the estimated state whenever a measurement is received.

Several real indoor flight experiments have been conducted in order to validate the proposed approach, with a minimal sensor setup of an IMU, a barometer and a 1D laser range finder. We achieve an average error less than 10 cm when comparing the estimated altitude by the state estimator with the altitude measured by the 1D laser range finder over flat ground surfaces.

Finally, we release this algorithm as an open source software within Aerostack framework[3], allowing the scientific community to use it for their experiments.

The first line of future work is the improvement of the models used in the components of our proposed state estimator, including the possibility that the sensors are not coincident with the robot reference frame, and allowing the fast navigation of the aerial robot.

As a second line of future work, our focus is to add measurements from a downward facing light weight RGB-D sensor in the proposed state estimator. The RGB-D sensor provides a much richer information in the form of point cloud as compared to a 1D laser range finder which provides information only about a single point. We focus to use a plane segmentation method, for computing the planes from a point cloud data and calculating the relative change in altitude with respect to the computed planes.

References

[1] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[2] S. Huh, D. H. Shim, and J. Kim, "Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of uavs," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3158–3163.

[3] J. P. Carvalho, M. A. Jucá, A. Menezes, L. R. Olivi, A. L. M. Marcato, and A. B. dos Santos, "Autonomous uav outdoor flight controlled by an embedded system using odroid and ros," in *CONTROLO 2016*. Springer, 2017, pp. 423–437.

[4] J. L. Sanchez-Lopez, R. Suarez-Fernandez, H. Bavle, C. Sampedro, M. Molina, and P. Campoy, "Aerostack: An architecture and open-source software framework for aerial robotics," in *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, June 2016, p. 0.

[5] C. Sampedro, H. Bavle, J. L. Sanchez-Lopez, R. A. S. Fernández, A. Rodríguez-Ramos, M. Molina, and P. Campoy, "A flexible and dynamic mission planning architecture for uav swarm coordination," in *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*. IEEE, 2016, pp. 355–363.

[6] J. Pestana, J. L. Sanchez-Lopez, P. de la Puente, A. Carrio, and P. Campoy, "A vision-based quadrotor swarm for the participation in the 2013 international micro air vehicle competition," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*. IEEE, 2014, pp. 617–622.

[7] J. F. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano, "Quadrotor using minimal sensing for autonomous indoor flight," in *European Micro Air Vehicle Conference and Flight Competition (EMAV2007)*, no. LIS-CONF-2007-006, 2007.

[8] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 2878–2883.

[9] ——, "A fully autonomous indoor quadrotor," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, Feb 2012.

[10] R. Mur-Artal and J. D. Tardos, "Orb-slam2: an open-source slam system for monocular, stereo and rgb-d cameras," *arXiv preprint arXiv:1610.06475*, 2016.

[11] J. Li, Y. Bi, M. Lan, H. Qin, M. Shan, F. Lin, and B. M. Chen, "Real-time simultaneous localization and mapping for uav: A survey."

[12] J. L. Sanchez-Lopez, J. Pestana, P. de la Puente, and P. Campoy, "A reliable open-source system architecture for the fast designing and prototyping of autonomous multi-uav systems: Simulation and experimentation," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1-4, pp. 779–797, 2016.

[13] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[14] J. L. Sanchez-Lopez, J. Pestana, J.-F. Collumeau, R. Suarez-Fernandez, P. Campoy, and M. Molina, "A vision based aerial robot solution for the mission 7 of the international aerial robotics competition," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. IEEE, 2015, pp. 1391–1400.

[15] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, and A. Savvaris, "Lidar-inertial integration for uav localization and mapping in complex environments," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 649–656.

[16] I. Dryanovski, R. G. Valenti, and J. Xiao, "An open-source navigation system for micro aerial vehicles," *Autonomous Robots*, vol. 34, no. 3, pp. 177–188, 2013.

[17] C. Sampedro, H. Bavle, A. Rodriguez-Ramos, A. Carrio, R. A. Suárez Fernández, J. L. Sanchez-Lopez, and P. Campoy, "A fully-autonomous aerial robotic solution for the 2016 international micro air vehicle competition," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017.

[2] https://vimeo.com/205774733

[3] www.aerostack.org