Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 16003–16008

Visual Marker based Multi-Sensor Fusion State Estimation ⋆

Jose Luis Sanchez-Lopez * **Victor Arellano-Quintana** **,*** **Marco Tognon** *** **Pascual Campoy** * **Antonio Franchi** ***

* *Centre for Automation and Robotics (CAR), CSIC-UPM (Technical University of Madrid), Madrid, Spain (e-mail: {jl.sanchez, pascual.campoy}@upm.es)*

** *ESIME-UA, National Polytechnic Institute, Mexico City, Mexico (e-mail: varellanoq1500@alumno.ipn.mx)*

*** *LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France (e-mail: {victor.arellano, marco.tognon, antonio.franchi}@laas.fr)*

Abstract:
This paper presents the description and experimental results of a versatile Visual Marker based Multi-Sensor Fusion State Estimation that allows to combine a variable optional number of sensors and positioning algorithms in a loosely-coupling fashion, incorporating visual markers to increase its performances. This technique allows an aerial robot to navigate in different environments and carrying out different missions with the same state estimation architecture, exploiting the best from every sensor. The state estimation algorithm has been successfully tested controlling a quadrotor equipped with an extra IMU and a RGB camera used only to detect visual markers. The entire framework runs on an onboard computer, including the controllers and the proposed state estimator. The whole software is made publicly available to the scientific community through an open source implementation.

Keywords: State Estimation; Sensor Fusion; Autonomous Mobile Robots; Inertial Sensors; Image Sensors; Position Estimation

1. INTRODUCTION

1.1 Motivation

Full state estimation of a multirotor aerial robot, including its pose and derivatives, is a complex task that is highly dependent on the environment and on the application.

The state estimation is performed using the information provided by the sensors. Nevertheless, there are no limitation-free sensors, e.g., IMUs (accelerometers and gyroscopes) suffer from high noises and drifts; GPS receivers are only working outdoors with small precisions and at a slow rate; air-pressure based sensors do not work properly in indoors areas due to the disturbances generated by the propellers; magnetometers are unreliable due to the electromagnetic disturbances created by the aerial platform, and specially by some elements in the environment such as steel girders; optical-flow based sensors that measure the ground speed, like the px4flow, require a textured, static and planar floor, and also require a small angle between the normal of the floor and the sensor axis; motion capture systems are only suitable in indoor areas and their installation is very expensive; ranging sensors, like LIDAR or RGB-D cameras, require the presence of objects to observe; RGB cameras, both monocular and stereo require textured environments and proper visibility conditions without fog or smoke. Therefore, a combination of sensors is required to produce a versatile and reliable state estimation.

Using vision-based algorithms for state estimation has become a trend in the last years because of the low-cost, and the light weight of cameras. Visual markers based approaches require the environment to have visually salient objects, and have the advantage of being able to handle untextured environment (e.g., a big white wall with a small window) or non-static scenes (e.g., an aerial robot pushing a box in a storehouse).

1.2 Related Works

Multiple Sensor Fusion (MSF) for state estimation is a common feature of all the works related to navigation of aerial robots. Loianno et al. (2015) propose a tightly-coupled Visual Inertial Odometry (VIO) algorithm used on a smartphone powered aerial robot that fuses visual keypoints with IMU measurements. Despite of showing poorer results in terms of accuracy, loosely-coupled approaches are preferred due to their versatility. Lynen et al. (2013) and Shen et al. (2014) show a loosely-coupling modular MSF algorithm for aerial robots. The former uses an EKF approach while the latter uses an UKF one. Both works are able to combine relative pose measurements from any number of sensors. A loosely-coupled approach is used in (Stegagno et al., 2014), showing the first example of a quadrotor estimating its state onboard using IMU and a RGB-D sensor while being bilaterally tele-operated, with force feedback computed from the onboard sensor data. Liu et al. (2016) developed a loosely-coupled algorithm to fuse monocular SLAM and IMU measurements

⋆ During this work Jose Luis Sanchez-Lopez has been funded by the Eiffel Excellence Scholarship Program of the French Ministry of Foreign Affairs and International Development and Victor Arellano-Quintana has been funded by a scholarship from CONACyT for studies abroad.
This work has been partially funded by the European Unions Horizon 2020 research and innovation programme under grant agreement No 644271 AEROARMS.

10.1016/j.ifacol.2017.08.1911

decoupling angular and linear states. All these loosely-coupled algorithms use an IMU sensor in the prediction stage, which limits to one the number of usable IMUs. Burri et al. (2015) propose to use a dynamic model of the vehicle in the prediction model instead, which increases the complexity of the model and requires more parameters to be identified.

The two main algorithms used in loosely-coupling state estimation are the well-known EKF and UKF. LaViola (2003) recommend to use EKF when attitude estimation is involved due to its simplicity. Additionally, Markley (2004b) and Markley (2004a) analyze how to handle quaternions in EKF state estimation, comparing Multiplicative and Additive approaches, finally recommending the multiplicative one for being theoretically consistent. Other important aspect needed when working in state estimation is the estimation of the state (*direct*) or the estimation of the error state (*indirect*). Panich (2010) justifies the usage of indirect approaches, what can be seen in most of the previously cited works. Shojaie et al. (2007) suggest to iterate the EKF to improve its accuracy and robustness against linear error propagation. Finally, to incorporate time-delayed measurements, a buffer that keeps track of the measurements and states is the most common option, seen in some of the previously cited works.

While natural visual markers detection is improved, reliable fiducial visual markers are widely used in the literature. Zhang et al. (2002) present a comparison on different fiducial visual markers, while Garrido-Jurado et al. (2014) present the preferred by us, the ArUco visual markers.

Some works, like Lim and Lee (2009) use fiducial visual markers for the navigation of a robot, but they do not include any other sensor. Neunert et al. (2016) fuse IMU measurements and visual markers in a state estimation algorithm that maps the visual markers in the image plane, making the algorithm very dependent on the marker type.

1.3 Contributions and outline

In this work, we present the first results of a versatile and robust state estimator, capable of being used in a wide range of environments and applications.

This paper continues our research on state estimation presented in Sanchez-Lopez et al. (2016) and Pestana et al. (2016). The proposed state estimator combines the information given by different sensors and by other state estimators, by means of an algorithm that incorporates different state of the art techniques (such as an extended Kalman filter). Moreover, the proposed state estimator includes the possibility of using visual markers to increase the robustness and precision of the state estimation at the only cost of augmenting the environment with these visual markers.

We have tested the state estimator with the minimal sensor setup of an IMU and a camera detecting visual markers, working at frequencies around 250 [Hz] (the IMU rate) with average errors less than 4 [cm] and 1 [°] for the estimated position and attitude, respectively.

The remainder of the paper is organized as follows: Sec. 2 describes the proposed algorithm for visual marker based multi-sensor fusion state estimation. Section 3 shows the estimator experimental results. Finally, Sec. 4 concludes the paper and indicates some future works.

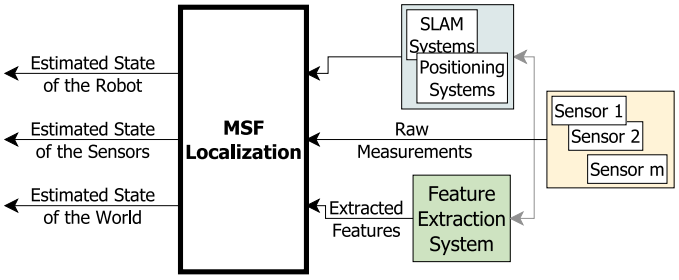


Fig. 1. General architecture of the multi-sensor fusion state estimation.

1.4 Notation and basic operations

Attitudes are represented by unit quaternions $\boldsymbol{q}$ following the Hamilton notation. $\boldsymbol{q}_A \otimes \boldsymbol{q}_B$ represents the composition operation of the quaternions $\boldsymbol{q}_A$ and $\boldsymbol{q}_B$. $(\boldsymbol{q}_A)^*$ represents the conjugate of the quaternion $\boldsymbol{q}_A$. $\boldsymbol{q}\{\boldsymbol{\nu}\}$ is the operation rotation vector to quaternion of $\boldsymbol{\nu}$. The notation $\boldsymbol{\nu}_A \times \boldsymbol{\nu}_B$ represents the cross product of the vectors $\boldsymbol{\nu}_A$ and $\boldsymbol{\nu}_B$.

The pose of frame A in coordinates of frame B, p_A^B, is represented by means of the position $\boldsymbol{t}_A^B$; and attitude $\boldsymbol{q}_A^B$. $\boldsymbol{\nu}_{A|B}^C$ represents the vectorial quantity $\boldsymbol{\nu}$ (e.g., velocity or acceleration) of the frame A with respect to the frame B in coordinates of frame C.

The equation $\boldsymbol{x}(k) = \boldsymbol{f}(\boldsymbol{x}(k-1), \boldsymbol{\mu}, \boldsymbol{u}(k-1), \boldsymbol{n}_f)$ represents a process model, and $\boldsymbol{z}(k) = \boldsymbol{h}(\boldsymbol{x}(k), \boldsymbol{\mu}, \boldsymbol{n}_z)$ represents a measurement model. $\boldsymbol{n}_\star$ represents a Gaussian white noise vector.

2. VISUAL MARKER BASED MULTI-SENSOR FUSION STATE ESTIMATION

The proposed state estimator has been designed to be versatile with a variable number of inputs, depending on the sensor setup and configuration defined by a particular mission or environment. The state estimator is therefore defined as a set of (optional) components (see Fig. 1) that are gathered in the following groups: 1) world (Sec. 2.1), 2) robot (Sec. 2.2), and 3) sensors (Sec. 2.3).

We propose an EKF-based state estimator with the following advanced features:

- Inclusion of stochastic parameters to model uncertainty of fixed coefficients.
- Error-state (i.e., indirect) formulation, accumulating the noise over the error state.
- Quaternions for attitude representation, to avoid singularities.
- Multiplicative solution, to deal with quaternions accurately.
- Time-delayed measurement compensation, by means of a circular buffer.
- Iterative nature, to quickly converge to the real state when the estimated state is far from the real state.

The following subsections provide further details of the previously mentioned features of the proposed estimator.

State and parameters Our algorithm considers two types of variables: On the one hand, the state is a random variable whose value changes over the time. On the other hand, parameters are random variables whose values do not change over the time. A normal distribution is assumed to represent both of them. In the case in which the

covariance of the normal distribution that represents a parameter is zero, then the parameter is assumed to be deterministic. The use of both stochastic and deterministic parameters increases the accuracy of the state estimation without inflating the state.

Observability The observability of the state must be satisfied for a particular state estimation setup. For the sake of generality, and due to page limitations, we do not analyze it in the present paper. Martinelli (2012) does an equivalent observability analysis for a tightly-coupled visual inertial state estimation.

Error-state formulation Our error-state formulation can distinguish between: 1) true-state ($\boldsymbol{x}$), 2) nominal-state ($\check{\boldsymbol{x}}$), and 3) error-state ($\delta\boldsymbol{x}$). The true-state is expressed as a composition of the nominal-state and the error-state (local perturbation: $\boldsymbol{x} = \check{\boldsymbol{x}} \oplus \delta\boldsymbol{x}$). The idea is to consider the nominal-state as large-signal (integrable in non-linear fashion) and the error-state as small signal (linearly integrable and suitable for linear-Gaussian filtering).

Mapping of world elements The mapping stage is carried out in a traditional EKF-SLAM fashion by augmenting the state and the covariance matrix with the new mapped element. The performance of the estimator decreases with the number of mapped elements. Since it is assumed to be limited, this is not a big inconvenience.

Buffer for time-delayed measurements To be able to accurately incorporate time-delayed measurements, a buffer approach is used. In the buffer, all the measurements and states are stored and organized by their timestamp. Olson (2010) shows a solution for accurate time-stamped synchronization.

Three different actions can take place in the buffer:

1) A new measurement arrives: The measurement is added to the buffer by its timestamp. After this, an estimation step is done in the new buffer element. Finally, the estimated states of the buffer are updated from the new buffer element to the newest (by timestamp) element.
2) New prediction step: The prediction of the state is done synchronously at a constant rate. This allows to accurately integrate the prediction model, keeping the estimated state on the buffer.
3) Request to get the current state estimate: When an estimate of the state is requested (e.g., by the controller), a prediction is done based on the newest (by timestamp) buffer element.

Iterative algorithm The proposed algorithm uses an iterated approach to improve the convergence to the real state when the estimated state is far from the real state. The update stage is iterated until a maximum number of iterations is reached or until the estimate of the state has not changed more than a threshold.

To better understand the following modules, Fig. 2 describes the reference frames and the main transformations involved in the proposed state estimator.

2.1 World

These modules include the following elements:

Gravity This component includes the acceleration of gravity in world coordinates, $\boldsymbol{g}^W_{|W}$. In order to increase the accuracy in the state estimation, it might be considered as a state with the following process model:

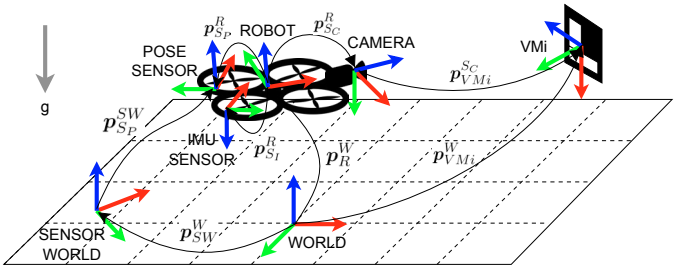


Fig. 2. Reference Frames involved on the Visual-Marker based Multi-Sensor Fusion State Estimator

$$\boldsymbol{g}^W_{|W}(k) = \boldsymbol{g}^W_{|W}(k-1). \tag{1}$$

Reference frames These components include the elements of the environment described with a reference frame, $RF\star$, by means of its pose (position and attitude) in World coordinates: $\boldsymbol{t}^W_{RF\star}$ and $\boldsymbol{q}^W_{RF\star}$. They are assumed to be static, but a more complex model could be added, being their process model:

$$\boldsymbol{t}^W_{RF\star}(k) = \boldsymbol{t}^W_{RF\star}(k-1) \tag{2}$$
$$\boldsymbol{q}^W_{RF\star}(k) = \boldsymbol{q}^W_{RF\star}(k-1). \tag{3}$$

2.2 Robot

This module includes the robot related information. The used model does not include any input command, estimating the full state up to the acceleration. It neither includes any parameter. This model is more versatile and faster than those that use an IMU as input command or the aerial robot model. Nevertheless, its prediction is only useful in the very short term, quickly diverging, being therefore highly dependent on the sensor measurements. In addition, it does not represent any extra robot state that might be useful (e.g., battery level).

The state is given by:

- Pose (position and attitude) of the robot in world coordinates: $\boldsymbol{t}^W_R$ and $\boldsymbol{q}^W_R$.
- Velocity (linear and angular) of the robot w.r.t. world in world coordinates: $\boldsymbol{v}^W_{R|W}$ and $\boldsymbol{\omega}^W_{R|W}$.
- Acceleration (linear and angular) of the robot w.r.t. world in world coordinates: $\boldsymbol{a}^W_{R|W}$ and $\boldsymbol{\alpha}^W_{R|W}$.

Its process model (with noises added as local perturbations) is:

$$\begin{aligned}\boldsymbol{t}^W_R(k) =& \boldsymbol{t}^W_R(k-1) + \boldsymbol{v}^W_{R|W}(k-1)\cdot\Delta t \\ &+ \boldsymbol{a}^W_{R|W}(k-1)\cdot\frac{\Delta t^2}{2}\end{aligned} \tag{4}$$
$$\boldsymbol{v}^W_{R|W}(k) = \boldsymbol{v}^W_{R|W}(k-1) + \boldsymbol{a}^W_{R|W}(k-1)\cdot\Delta t \tag{5}$$
$$\boldsymbol{a}^W_{R|W}(k) = \boldsymbol{a}^W_{R|W}(k-1) + \boldsymbol{n}_{a^W_{R|W}} \tag{6}$$
$$\boldsymbol{q}^W_R(k) = \delta\boldsymbol{q} \otimes \boldsymbol{q}^W_R(k-1) \tag{7}$$
$$\boldsymbol{\omega}^W_{R|W}(k) = \boldsymbol{\omega}^W_{R|W}(k-1) + \boldsymbol{\alpha}^W_{R|W}(k-1)\cdot\Delta t \tag{8}$$
$$\boldsymbol{\alpha}^W_{R|W}(k) = \boldsymbol{\alpha}^W_{R|W}(k-1) + \boldsymbol{n}_{\alpha^W_{R|W}}. \tag{9}$$

where:

$$\delta\boldsymbol{q} = \boldsymbol{q}\left\{\bar{\boldsymbol{\omega}}\cdot\Delta t\right\} + \frac{\Delta t^2}{24}\cdot\delta\boldsymbol{q}\left[\boldsymbol{\alpha}\right] \tag{10}$$

$$\bar{\boldsymbol{\omega}}\cdot\Delta t = \left(\boldsymbol{\omega}^W_{R|W}(k-1) + \frac{\Delta t}{2}\cdot\boldsymbol{\alpha}^W_{R|W}(k-1)\right)\Delta t \tag{11}$$

$$\delta\boldsymbol{q}\left[\boldsymbol{\alpha}\right] = \begin{bmatrix} 0 \\ \boldsymbol{\alpha}^W_{R|W}(k-1)\times\left(\boldsymbol{\omega}^W_{R|W}(k)\right) \end{bmatrix}. \tag{12}$$

Note that, since $\delta\boldsymbol{q}$ is not unitary, a re-normalization is required after getting $\boldsymbol{q}^W_R(k)$.

2.3 Sensors

All the sensors components, $S\star$, share a similar common part that represent their reference frame. They have the pose (position and attitude) of the sensor in robot coordinates as state or parameters: $\boldsymbol{t}^R_{S\star}$ and $\boldsymbol{q}^R_{S\star}$. They are described by the following noise-free process model:

$$\boldsymbol{t}^R_{S\star}(k) = \boldsymbol{t}^R_{S\star}(k-1) \tag{13}$$

$$\boldsymbol{q}^R_{S\star}(k) = \boldsymbol{q}^R_{S\star}(k-1). \tag{14}$$

The following different kinds of sensors have been taken into account in our work:

IMU sensor The IMU sensor component, S_I, adds to common state or parameters, the biases on the measurement of the accelerometer, $\boldsymbol{b}_{accel}$, and gyro $\boldsymbol{b}_{gyro}$. It also includes as zero covariance parameters, the sensitivity matrices on the measurement of the accelerometer, $\boldsymbol{S}_{accel}$, and gyro, $\boldsymbol{S}_{gyro}$.

The dynamics of the non-static biases are modeled as a random process, because they randomly change their value with the time, so, its noisy process model is given by:

$$\boldsymbol{b}_{accel}(k) = \boldsymbol{b}_{accel}(k-1) + \boldsymbol{n}_{b_{accel}} \tag{15}$$

$$\boldsymbol{b}_{gyro}(k) = \boldsymbol{b}_{gyro}(k-1) + \boldsymbol{n}_{b_{gyro}}. \tag{16}$$

The measurement model of the accelerometer is:

$$\boldsymbol{z}_{accel} = \boldsymbol{S}_{accel}\cdot\left(\boldsymbol{a}^{S_I}_{S_I|W} - \boldsymbol{g}^{S_I}_{|W}\right) + \boldsymbol{b}_{accel} + \boldsymbol{n}_{accel}, \tag{17}$$

being:

$$\boldsymbol{g}^{S_I}_{|W} = \left(\boldsymbol{q}^W_{S_I}\right)^*\otimes\boldsymbol{g}^W_{|W}\otimes\boldsymbol{q}^W_{S_I} \tag{18}$$

$$\boldsymbol{a}^{S_I}_{S_I|W} = \left(\boldsymbol{q}^W_{S_I}\right)^*\otimes\boldsymbol{a}^W_{R|W}\otimes\boldsymbol{q}^W_{S_I} + \left(\boldsymbol{q}^R_{S_I}\right)^*\otimes\boldsymbol{a}^R_{Fict}\otimes\boldsymbol{q}^R_{S_I} \tag{19}$$

$$\boldsymbol{a}^R_{Fict} = \boldsymbol{\omega}^R_{R|W}\times\boldsymbol{\omega}^R_{R|W}\times\boldsymbol{t}^R_{S_I} + \boldsymbol{\alpha}^R_{R|W}\times\boldsymbol{t}^R_{S_I}. \tag{20}$$

Similarly, the measurement model of the gyro is:

$$\boldsymbol{z}_{gyro} = \boldsymbol{S}_{gyro}\cdot\boldsymbol{\omega}^{S_I}_{S_I|W} + \boldsymbol{b}_{gyro} + \boldsymbol{n}_{gyro}, \tag{21}$$

being:

$$\boldsymbol{\omega}^{S_I}_{S_I|W} = \left(\boldsymbol{q}^W_{S_I}\right)^*\otimes\boldsymbol{\omega}^W_{R|W}\otimes\boldsymbol{q}^W_{S_I} \tag{22}$$

Coded visual marker detector The visual marker detector is run on the images acquired by a camera, S_C. The coded visual markers are uniquely labeled, and represented by a reference frame, VMi.

Its measurement model (with local perturbation) is:

$$\boldsymbol{z}_t = \left(\boldsymbol{q}^W_{S_C}\right)^*\otimes\left(\boldsymbol{t}^W_{VMi} - \boldsymbol{t}^W_{S_C}\right)\otimes\boldsymbol{q}^W_{S_C} + \boldsymbol{n}_{z_t} \tag{23}$$

$$\boldsymbol{z}_q = \left(\left(\boldsymbol{q}^W_{S_C}\right)^*\otimes\boldsymbol{q}^W_{VMi}\right)\otimes\boldsymbol{q}\left\{\boldsymbol{n}_{z_q}\right\}, \tag{24}$$

Pose sensor This generic sensor component, S_P, represents any SLAM and positioning systems such as GPS or Motion Capture Systems. They provide measurements in coordinates of their particular world reference frames SW.

Its measurement model (with local perturbation) is:

$$\boldsymbol{z}_t = \left(\boldsymbol{q}^W_{SW}\right)^*\otimes\left(\boldsymbol{t}^W_{S_P} - \boldsymbol{t}^W_{SW}\right)\otimes\boldsymbol{q}^W_{SW} + \boldsymbol{n}_{z_t} \tag{25}$$

$$\boldsymbol{z}_q = \left(\left(\boldsymbol{q}^W_{SW}\right)^*\otimes\boldsymbol{q}^W_R\otimes\boldsymbol{q}^R_{S_P}\right)\otimes\boldsymbol{q}\left\{\boldsymbol{n}_{z_q}\right\}, \tag{26}$$

3. RESULTS

In this section, the proposed algorithm for state estimation is tested through real experiments controlling a quadrotor.

3.1 System Setup

A Mikrokopter quadrotor is used as the aerial platform, as seen in Fig. 3a. Its Flight Controller board is only used to acquire its embedded IMU measurements at 1 kHz and as interface with the motor controllers. The quadrotor is equipped with an extra IMU Phidgets 1044 (250 Hz) and a UI-3241-LE-C-HQ camera with a wide angle lens (set to 640 × 480 @ 30 Hz). An Intel NUC5i7RYH is mounted onboard in which all the software runs. Finally, a desktop computer is used as the Ground Control Station (GCS) for user visualization and mission commanding purposes.

The system architecture is represented in Fig. 3b and has the following components:

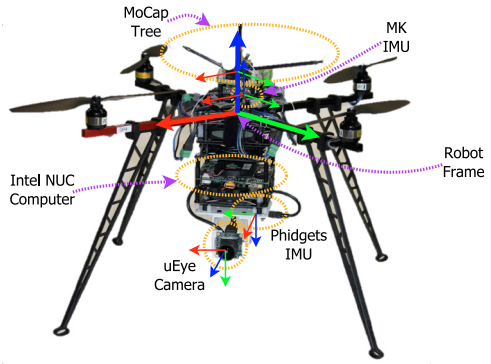
- *Aruco Eye* [1]: Visual marker detector based on Garrido-Jurado et al. (2014).
- *POM* [2]: Based on Crassidis and Markley (2003), it fuses the MK-IMU measurements with the proposed state estimator output. This cascaded approach confers robustness to the system when closing the hard-real time control loop at 1 kHz.
- *Controller*: A standard SE(3) controller (attitude, velocity and position) for the quadrotor, which greatly benefits from the in-house built direct motor speed controller by Franchi and Mallet (2017).
- *Trajectory Commander*: Generates a smooth trajectory from the commanded waypoints.
- *Visual Marker Obstacle Detector* [3]: Generates usable geometric primitives based on the visual markers of the environment.

3.2 Experimental Results

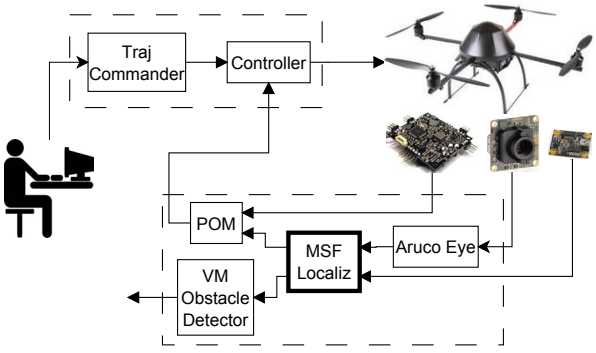
The goal of this experiment is to perform a simple trajectory using only onboard sensors showing the performances of the proposed estimator. After the take-off the robot is required to follow a circular trajectory on an horizontal plane at height 1.3 [m], oscillating along the x-axis of the inertial frame between −0.5 and 0.6 [m], and along the y-axis between −0.5 and 0.25 [m], with a frequency of 0.8 [Hz]. The trajectory is performed while controlling the yaw angle to be equal to 45[°] in order to always head toward the wall on which 5 markers are placed. This wall is developed along the x-axis of the inertial frame and placed at $y = 1.32$[m]. A video of the experiment can be found at `https://youtu.be/8viNirMet_4`.

This simple, but rich enough trajectory, is used to show the capabilities of the proposed method on estimating the

[1] `https://github.com/joselusl/aruco_eye`
[2] `https://git.openrobots.org/projects/pom-genom3`
[3] `https://bitbucket.org/joselusl/vm_obstacle_detector`



(a) Hardware used for the experiments. The markers for the MOCAP are used only for the ground truth.



(b) System architecture used for the experiments.

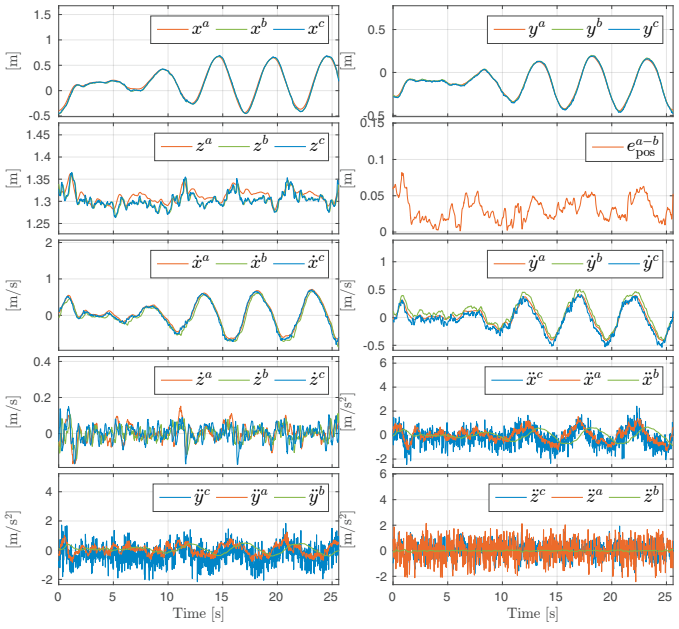
Fig. 3. System setup used for the experiments.

pose of the robot and its first and second derivatives, as it is shown in Fig. 4. In particular this figure shows the estimated variables for three different approaches:

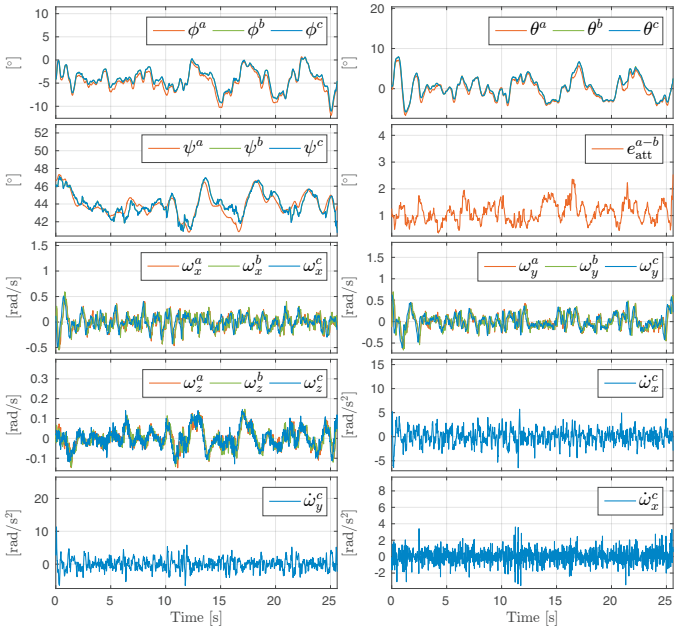
a) *POM+MOCAP*: POM is used to fuse the MK-IMU measurement with the one coming from a motion capture system (MOCAP). This estimated state is used as a ground truth to evaluate the performance of our method;
b) *POM+MSF*: as said before, the use of POM on top of our methods gives robustness to the final estimation;
c) *MSF*: the estimated state provided by the proposed method without any additional filtering.

Furthermore, in the plots we show the mean estimation error on the position and on the orientation of the robot with respect to the methods a) and b), i.e., e_{pos}^{a-b} and e_{att}^{a-b}, respectively. Looking at these quantities we can see how the estimation errors remain always bounded. In particular the error stays between a minimum value less then 1 [cm] and a maximum value about 7 [cm] for the estimation of the position; and between a minimum value less then 1 [°] and a maximum value about 2 [°] for the estimation of the attitude. However, notice that the peaks on the errors coincide with the moments in which the aerial vehicle is at the farthest position from the markers. Indeed the more the distance between the camera and a marker, the higher the noise on the visual marker detector.

From Fig. 4, comparing methods b) and c) with method a), can be seen that the linear and angular velocity and acceleration are well estimated, except for some small biases due to calibration errors. We highlight the fact that the additional filtered method b), as expected, reduces the noise, in particular on the estimated linear acceleration, but, on the other hand, introduces a small delay. The last origin of



(a) Estimated variables related to the translational dynamics



(b) Estimated variables related to the rotational dynamics. Roll-pitch-yaw are used only for visualization purposes, all the estimator computation are done using unit quaternions.

Fig. 4. Experimental results following a circular trajectory at a constant altitude.

estimation errors, and in particular of biases, consists on the calibration errors that have to be minimized in order to obtain the best performances of the estimator.

4. CONCLUSIONS AND FUTURE WORK

This paper presented the first results of a versatile Multi-Sensor Fusion State Estimation for aerial robots, that allows combining, with the same state estimation architecture, a variable optional number of sensors and positioning algorithms in a loosely-coupling fashion. This lets exploit the best from every sensor, providing flexibility in the working environment and mission executed. It also includes

the use of visual markers in a SLAM loop to increase the performance of the state estimation.

We have combined in an EKF-based state estimator the following features: 1) Inclusion of stochastic parameters to model uncertainty of fixed coefficients; 2) Error-state (indirect) method, accumulating the noise over the error state; 3) Quaternions for attitude representation, to avoid singularities; 4) Multiplicative update, to deal with quaternions accurately; 5) Time-delayed measurement compensation, by means of a circular buffer; 6) Iterative method, to quickly converge to the real state when the estimated state is far from the real state.

Our state estimator have been successfully tested with a minimal sensor setup in real experiments, controlling a quadrotor equipped with an extra IMU and a RGB camera used only to detect visual markers, running on an onboard computer. The state estimation output is comparable to the ground truth (given by a motion capture system) with average errors less than 4 [cm] and 1 [°].

Our final contribution is the release of the algorithm as an open-source software [4], allowing the scientific community to use it in their experiments.

As future work, we are planning to add more sensors, like unscaled pose sensors (e.g. mono-vSLAM algorithms) and drifting pose sensors (for odometry algorithms like mono-VO algorithms). Integration of force sensors or force observers, as e.g., the ones presented in (Yüksel et al., 2014b) and in (Ryll et al., 2017) will be extremely helpful in order to estimate the state of aerial vehicles while in contact with the environment, a scenario that is becoming very popular in the literature, see, e.g., (Yüksel et al., 2014a) and (Gioioso et al., 2014) and references therein.

Other planned future work is to explore other more efficient approaches for the mapping, like Sparse Extended Information Filters, that better scales up when the number of mapped elements grow significantly.

Finally, we plan to investigate the extension to multiple robots performing mutual localization in critical conditions, as, e.g., the anonymous measurement case introduced in Franchi et al. (2013) and Stegagno et al. (2016).

REFERENCES

Burri, M., Dtwiler, M., Achtelik, M.W., and Siegwart, R. (2015). Robust state estimation for micro aerial vehicles based on system dynamics. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 5278–5283.

Crassidis, J.L. and Markley, F.L. (2003). Unscented filtering for spacecraft attitude estimation. *Journal of guidance, control, and dynamics*, 26(4), 536–542.

Franchi, A. and Mallet, A. (2017). Adaptive closed-loop speed control of BLDC motors with applications to multi-rotor aerial vehicles. In *2017 IEEE Int. Conf. on Robotics and Automation*. Singapore.

Franchi, A., Oriolo, G., and Stegagno, P. (2013). Mutual localization in multi-robot systems using anonymous relative measurements. *The International Journal of Robotics Research*, 32(11), 1302–1322.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., and Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6), 2280 – 2292.

Gioioso, G., Ryll, M., Prattichizzo, D., Bülthoff, H.H., and Franchi, A. (2014). Turning a near-hovering controlled quadrotor into a 3D force effector. In *2014 IEEE Int. Conf. on Robotics and Automation*, 6278–6284. Hong Kong, China.

LaViola, J.J. (2003). A comparison of unscented and extended kalman filtering for estimating quaternion motion. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, 2435–2440 vol.3.

Lim, H. and Lee, Y.S. (2009). Real-time single camera slam using fiducial markers. In *ICCAS-SICE, 2009*, 177–182.

Liu, C., Prior, S.D., Teacy, W.L., and Warner, M. (2016). Computationally efficient visual–inertial sensor fusion for global positioning system–denied navigation on a small quadrotor. *Advances in Mechanical Engineering*, 8(3).

Loianno, G., Mulgaonkar, Y., Brunner, C., Ahuja, D., Ramanandan, A., Chari, M., Diaz, S., and Kumar, V. (2015). Smartphones power flying robots. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 1256–1263.

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

Markley, F.L. (2004a). Attitude estimation or quaternion estimation? *Journal of Astronautical Sciences*, 52(1), 221–238.

Markley, F.L. (2004b). Multiplicative vs. additive filtering for spacecraft attitude determination. In *Proceedings of the 6th Conference on Dynamics and Control of Systems and Structures in Space (DCSSS)*, volume 22.

Martinelli, A. (2012). Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, 28(1), 44–60.

Neunert, M., Bloesch, M., and Buchli, J. (2016). An open source, fiducial based, visual-inertial motion capture system. In *2016 19th International Conference on Information Fusion (FUSION)*, 1523–1530.

Olson, E. (2010). A passive solution to the sensor synchronization problem. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 1059–1064.

Panich, S. (2010). Indirect kalman filter in mobile robot application. *Journal of Mathematics and Statistics*, 6(3).

Pestana, J., Sanchez-Lopez, J.L., de la Puente, P., Carrio, A., and Campoy, P. (2016). A vision-based quadrotor multi-robot solution for the indoor autonomy challenge of the 2013 international micro air vehicle competition. *Journal of Intelligent & Robotic Systems*, 84(1), 601–620.

Ryll, M., Muscio, G., Pierri, F., Cataldi, E., Antonelli, G., Caccavale, F., and Franchi, A. (2017). 6D physical interaction with a fully actuated aerial robot. In *2017 IEEE Int. Conf. on Robotics and Automation*. Singapore.

Sanchez-Lopez, J.L., Pestana, J., de la Puente, P., and Campoy, P. (2016). 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*, 84(1), 779–797.

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

Shojaie, K., Ahmadi, K., and Shahri, A.M. (2007). Effects of iteration in kalman filters family for improvement of estimation accuracy in simultaneous localization and mapping. In *2007 IEEE/ASME international conf. on advanced intelligent mechatronics*, 1–6.

Stegagno, P., Basile, M., Bülthoff, H.H., and Franchi, A. (2014). A semi-autonomous UAV platform for indoor remote operation with visual and haptic feedback. In *2014 IEEE Int. Conf. on Robotics and Automation*, 3862–3869. Hong Kong, China.

Stegagno, P., Cognetti, M., Oriolo, G., Bülthoff, H.H., and Franchi, A. (2016). Ground and aerial mutual localization using anonymous relative-bearing measurements. *IEEE Trans. on Robotics*, 32(5), 1133–1151.

Yüksel, B., Secchi, C., Bülthoff, H.H., and Franchi, A. (2014a). Aerial physical interaction via reshaping of the physical properties: Passivity-based control methods for nonlinear force observers. In *ICRA 2014 Workshop: Aerial robots physically interacting with the environment*. Hong Kong, China.

Yüksel, B., Secchi, C., Bülthoff, H.H., and Franchi, A. (2014b). A nonlinear force observer for quadrotors and application to physical interactive tasks. In *2014 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 433–440. Besançon, France.

Zhang, X., Fronz, S., and Navab, N. (2002). Visual marker detection and decoding in ar systems: A comparative study. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02.

[4] `https://bitbucket.org/joselusl/msf_localization`