A Visual-Inertial Navigation System Using AprilTag for Real-Time MAV Applications

Joao Paulo de Almeida Barbosa
ADAS Systems Engineering - AVL UK
Aeronautical Institute of Technology (ITA)
Coventry, UK
barbosajoaopa@gmail.com

Stiven Schwanz Dias
Technology Development
Embraer S.A.
Sao Jose dos Campos, Brazil
stivendias@gmail.com

Davi Antônio dos Santos
Mechanical Engineering
Aeronautical Institute of Technology (ITA)
Sao Jose dos Campos, Brazil
davists@ita.br

***Abstract*—Typical industrial applications, which can benefit from using Micro Aerial Vehicles (MAV) take place at well-controlled environments. In this case, the MAVs can rely on a fiducial system to improve their navigation based on a reliable map of the environment. In this work, we assess the use of artificial landmarks for accurate and robust visual-inertial navigation within a controlled environment. The use of artificial landmarks presents relevant advantages over techniques that assume an unknown environment. They require less processing effort and provide more robustness for dynamic and large environments. As a result, one can employ simpler, affordable platforms enabling commercially attractive applications due to cost reduction, energy efficiency, simplicity and determinism during operation. Therefore, this work presents the design and development of an accurate and efficient visual-inertial navigation system to provide feedback information allowing low and high-level control functionalities for MAV applications.**

***Index Terms*—MAV, fiducial system, visual-inertial navigation, real-time**

I. Introduction

In some applications, the use of GPS may not represent a robust solution for navigation of MAVs [2]. The GPS depends on a complex satellite mesh infrastructure that provides the information to the platform receiver, it means that its signal is susceptible to interferences, can be jammed or masked, and might not be available in dense tree forest, in some urban areas, underground or in indoor environments [3].

Camera vision system appeared as a suitable solution for replacing the GPS in MAV's navigation systems [3]. Cameras are low cost, small and light, and the image is able to provide direct measures of the position [4]. The system can be all embedded in the platform reducing the possibilities of external interference and the influence of environment in the navigation system [5].

Navigation of MAVs using visual-inertial techniques is largely explored and the approaches to solve this problem may vary from different filtering methods to different techniques to extract navigation cues from sensors. In this wide context, it is possible to divide recent related works into two general groups: one that considers totally unknown environment [6]–[8] and other that deals with apriori mapped features [9]–[13].

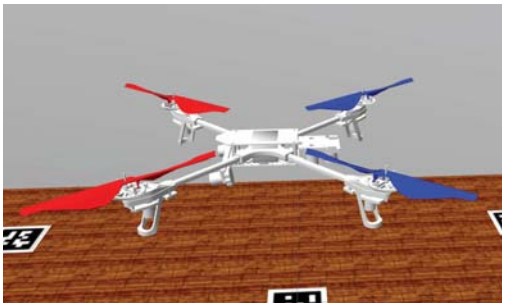

Fig. 1: ARDRONE model from RotorS with Camera and IMU sensors coupled.

Although projects that use the premise of an unknown world are more recent and demonstrate to be more flexible, they also add more complexity in the embedded algorithm to guarantee expanded flexibility and high robustness. On the other hand, several applications may count on a relatively controlled environment, such as warehouse, manufacturing cells, construction plants, indoor and urban safe areas [14]–[16]. For these cases, flexibility for variations in the environments can be neglected, making the platform more specific for controlled environment applications. Therefore, this focused approach presents advantages in computer efficiency and robustness maintaining the solution effectiveness for environments that allow preparation with artificial landmarks.

Considering that some applications, mainly those encountered in industry scenario, are in a well-controlled environment, the use of artificial landmarks presents relevant advantages over techniques that use an unknown map as an assumption. Tags that allow estimation of position and attitude with optimized algorithms present advantages such as less processing effort and more robustness for dynamic and large environments. The result of these advantages is reflected in a simpler platform more attractive commercially. The attractiveness is due to cost

This work was supported by FCMF Brazil and AVL Powertrain UK.

978-1-5386-7544-1/18/$31.00 ©2018 IEEE

reduction, simplicity, determinism during operation, energy efficiency, and even the possibility to implement other complex functions, such as obstacle avoidance, automatic landing or swarm flight operation.

Therefore, the motivation of this project is to explore the use of artificial landmarks in an accurate and robust visual-inertial navigation system exposing gains of this approach when applied in a controlled environment. The objective is the development of an accurate and efficient visual-inertial navigation algorithm using AprilTag 2 [1] in an Extended Kalman Filter scheme for an apriori known map. The algorithm shall be able to provide navigation parameters for low and high-level control functionalities besides being simple and robust for industrial applications, optimizing the algorithm through environment capabilities maintaining the quality of information expected from a navigation estimator for these environments.

II. Problem Description

This work is concerned with an inertial-vision navigation problem of a sensor platform containing a triad of rate-gyros, a triad of accelerometers, and a camera. The problem is defined as a nonlinear estimation of position, velocity, and attitude. Information of the platform pose and attitude are extracted from the camera by means of fiducial system identification. A landmark map is assumed to be available.

Consider the navigation environment, the sensor platform, and the CCSs illustrated in Figure 2. The ground CCS, $S_{\mathrm{G}} \triangleq \{X_{\mathrm{G}}, Y_{\mathrm{G}}, Z_{\mathrm{G}}\}$, is attached to the ground at point G, with the Z_{G} axis aligned with the local vertical. The platform CCS, $S_{\mathrm{P}} \triangleq \{X_{\mathrm{P}}, Y_{\mathrm{P}}, Z_{\mathrm{P}}\}$, is fixed on the platform at a known point P. The position vector $\mathbf{r}^{\mathrm{P/G}}$ and velocity vector $\mathbf{v}^{\mathrm{P/G}}$ represent, respectively, the position and velocity of S_{P} w.r.t. S_{G}. Thus, the S_{G} representations of $\mathbf{r}^{\,\mathrm{P/G}}$ and $\mathbf{v}^{\,\mathrm{P/G}}$ are given by $\mathbf{r}_{\mathrm{G}}^{\mathrm{P/G}} \in \mathbb{R}^3$ and $\mathbf{v}_{\mathrm{G}}^{\mathrm{P/G}} \in \mathbb{R}^3$, respectively.

Assume that the inertial system is concentric and aligned with S_{P} and $\boldsymbol{\alpha} \in \mathbb{R}^3$ defines the platform attitude w.r.t. S_{G} by stacking the three Euler angles in the sequence 1-2-3. Thus, the kinematics model is defined by the following state equation:

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) + \mathbf{G}\left(\mathbf{x(t)}\right)\mathbf{w}(t), \tag{1}$$

where $\mathbf{x}(t)$ is the state vector composed by position, linear velocity, attitude and biases of the sensors

$$\mathbf{x}(t) \triangleq \left[\left(\mathbf{r}_{\mathrm{G}}^{\mathrm{G/P}}\right)^{\mathrm{T}} \quad \left(\mathbf{v}_{\mathrm{G}}^{\mathrm{G/P}}\right)^{\mathrm{T}} \quad \boldsymbol{\alpha}^{\mathrm{T}} \quad {}^{a}\boldsymbol{\beta}^{\mathrm{T}} \quad {}^{g}\boldsymbol{\beta}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{15}, \tag{2}$$

note that ${}^{a}\boldsymbol{\beta} \in \mathbb{R}^3$ is the accelerometer bias and ${}^{a}\boldsymbol{\beta} \in \mathbb{R}^3$ is the gyroscope bias. Both are modeled as a Winer process.

Figure 2 also presents a pinhole camera model [20]. The camera CCS, $S_{\mathrm{C}} \triangleq \{X_{\mathrm{C}}, Y_{\mathrm{C}}, Z_{\mathrm{C}}\}$, has its origin located at C. Each landmark (AprilTag) in the scenario has its own CCS represented by $S_{\mathrm{Ti}} \triangleq \{X_{\mathrm{Ti}}, Y_{\mathrm{Ti}}, Z_{\mathrm{Ti}}\}$, where the index $i = 1, ..., q$ represents each one of the q visible known landmarks in the environment.

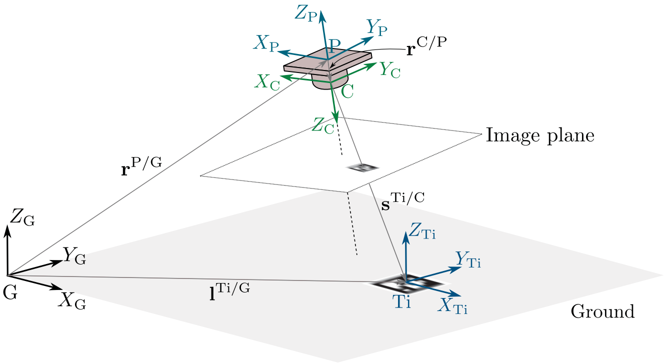

Fig. 2: Navigation platform and a pinhole camera model identifying an AprilTag.

The aerial platform position ($\mathbf{r}^{\mathrm{P/G}}$) and orientation ($\boldsymbol{\alpha}$) w.r.t. S_{G} are predicted using the kinematics model. The camera position ($\mathbf{r}^{\mathrm{C/P}}$) and rotation ($\mathbf{D}^{\mathrm{C/P}}$) w.r.t. S_{P} are considered known (parameters). Finally, the i-th landmark position ($\mathbf{l}^{\mathrm{Ti/G}}$) and rotation ($\mathbf{D}^{\mathrm{Ti/G}}$) w.r.t. S_{G} are part of the known map information.

From geometry, one can determine the i-th landmark position, $\mathbf{s}^{\mathrm{Ti/C}}$, and rotation matrix, $\mathbf{D}^{\mathrm{Ti/C}}$, w.r.t. S_{C}. Therefore, the following measurement equations are defined

$$\mathbf{y}^{i} = \mathbf{h}^{i}(\mathbf{x}) + \mathbf{n}^{i}, i = 1, ..., q, \tag{3}$$

where $\mathbf{n}^{i} \in \mathbb{R}^3$ is a noise term, assumed to be zero-mean white Gaussian process, with known covariance $\mathbf{R}^{i} \in \mathbb{R}^{3\times 3}$ and

$$\mathbf{h}^{i}(x) \triangleq \begin{bmatrix} \mathbf{s}_{\mathrm{C}}^{\mathrm{Ti/C}} \\ \boldsymbol{\gamma}^{i} \end{bmatrix} \tag{4}$$

III. Proposed Solution

A. Position and Attitude Measurement from the Tags

When compared to other fiducial systems, such as ARToolkit variations [21] and ARTag [22], AprilTag is proved to deliver better results in terms of robustness [18]. In addition, its computational performance and flexibility were considerably improved in the second version [1], making it possible to embed and adapt the identification algorithm for specific applications. These technical features, in parallel with the high use in robotics, drove the decision of using AprilTag fiducial system as landmarks in the present work.

The AprilTag algorithm provides useful information about the identified tag, such as corners and center position. Thus, by knowing the size of the tag, its position in the scenario and the camera extrinsic parameters, one can use a PnP solver algorithm to define the position and orientation of the landmark. In this project, the size and position of the tags are part of the map, the camera extrisinc parameters are considered known and the solver used is the *solvePnP* from *OpenCV* [23].

B. General Filter Design

The present work adopts the continuous-discrete formulation of the extended Kalman filter [24]. Figure 3 demonstrates the general overview of the proposed EKF architecture to estimate the navigation parameters of the problem described in the Section II.

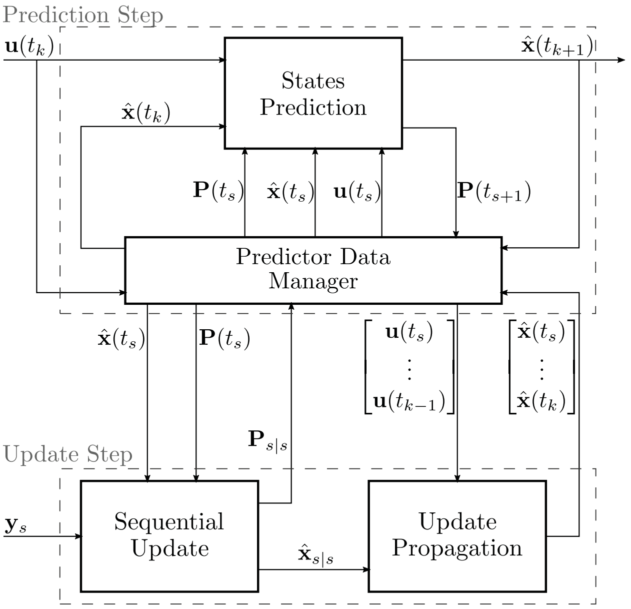

Fig. 3: Filter Architecture Design.

The presented filter architecture was defined to deal with real-time and implementation issues that are mainly related to the time consumption for processing the images on an embedded system. Note from Figure 3 that the proposed filter is always in predictor mode. Next subsections explain deeper each the prediction and update steps.

C. Prediction

The states prediction step is based on the kinematics model. The input vector $\mathbf{u}(t)$ for this step is acquired from the IMU. Thus, as Inertial Navigation Systems have a high acquisition rate, it allows the prediction step to run at a higher frequency. It must ensure that navigations parameters are estimated and available in a rate that allows low-level control loop for platform stabilization even for periods that states update step is not executed.

This process is represented by the "State Prediction" block in the Figure 3. As prediction and update steps are at different rates, their interface is possible due to the presence of the "Predictor Data Manager" block shown in the Figure 3. It stores and retrieves buffer data of input, state and covariance estimates to sync the filter steps. Thus, its main function is to ensure the filter cycles are performed correctly in real-time even with prediction and update steps in different execution rates.

D. Update

The update step compares the predicted measure values from the measurement model with the real measures from AprilTag detections. Since image acquisition and processing for tag detection and feature extraction demands a considerable computational effort, the measurement of each image has limited update rate. Therefore, the update step is conditioned to execute at a lower rate when compared to prediction step.

Note from Figure 3 that this process is represented in the block "Sequential Update". It is executed sequentially to process all identified tags efficiently in terms of computational processing.

Image acquisition and processing require high computational effort, a fact that limits the frequency to acquire, process and extract features in a conventional embedded system. The time consumption to extract useful information also impacts the filter operation. The update step receives measures related to images that were acquired in a past instant with reference to the filter estimation cycle [26]. Figure 4 demonstrates an image acquired at instant t_s of filter cycle execution where the image processing extended for n estimation cycles until obtaining the measure needed to execute an update step.

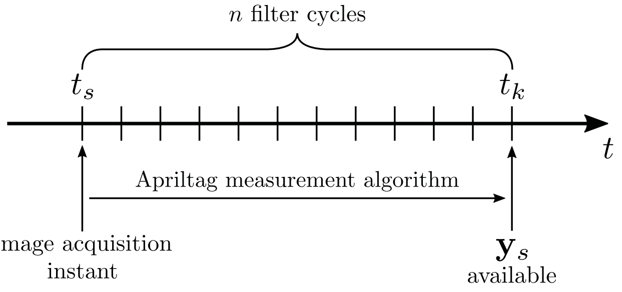

Fig. 4: Image acquisition instant consideration.

For MAVs, with fast dynamics, this timing issue can not be neglected. Therefore, as the updated state is in the past (t_s), it is necessary to propagate the corrected state until current filter cycle instant (t_k). This process is represented by the "Update Propagation" block in the Figure 3.

IV. Development and Simulation

A. Development - Robotic Operating System

Architecture design, shown in Figure 3, shows that the proposed filter depends on two inputs, IMU readings $\mathbf{u}(t_k)$ and AprilTag measures $\mathbf{y}_s$, and provides one output, the states prediction $\hat{\mathbf{x}}(t_{k+1})$. Thus, thinking in a real implementation, an IMU driver would be the responsible to provide inertial measures while a camera driver should be able to provide the image to an AprilTag measurement process. On the other hand, the navigation parameters provided by the filter shall be available to other applications, such as low and high-level control processes.

Therefore, as IMU readings, camera image and navigation parameters may be used by several different operations, a modular software architecture represents a coherent choice to develop the complete solution. Figure 5 presents a schematic of the software architecture.

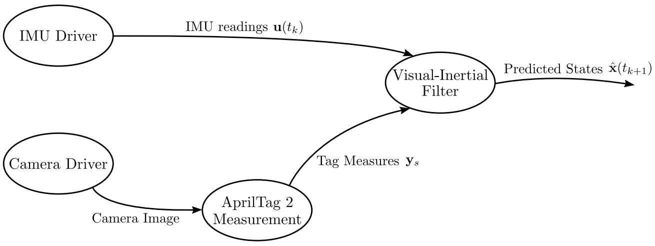
IMU Driver
IMU readings $\mathbf{u}(t_k)$
Visual-Inertial Filter
Predicted States $\hat{\mathbf{x}}(t_{k+1})$
Camera Driver
Tag Measures $\mathbf{y}_s$
Camera Image
AprilTag 2 Measurement

Fig. 5: Software Architecture Scheme.

Once the architecture is meant to be very modular with specific communication between the processes, Robotic Operating System (ROS) is an adequate framework to be used as the development environment of the proposed solution [27].

B. Simulation Model - Gazebo

A simulation environment with high physical and dynamical fidelity that enables tests with ROS packages is a suitable choice to be used in this type of development. That is the reason that droves the choice of Gazebo [28] as the simulation tool of this project.

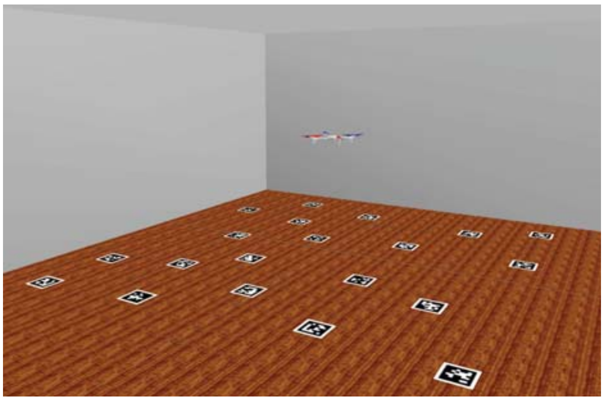

Fig. 6: Simulation model created from Gazebo world tools and ARDRONE model of RotorS.

As the proposed application is a visual-inertial navigation system to MAVs, it is necessary to have a quad-rotor model with trustworthy aerodynamics, inertial and camera sensors, and a control system to enable generating trajectories and testing navigation parameters feedback. For this purpose, RotorS [29] is the Gazebo MAV framework used.

Figure 6 is the simulation model created based on the real scenario. Twenty-one AprilTags of family 36h11, are placed on the arena floor, in specific positions and orientations with respect to the world coordinate system S_{G}. For the MAV object, the ARDRONE model with default parameters from RotorS is used. IMU and Camera are sensor models composed of Gazebo models and plugins.

C. Simulation Architecture

Sensors plugins also interface the simulation model with ROS environment. In this case, they are responsible to publish their reading values. This is the abstraction level of the simulation for applications that use these data. It means that in a real environment, information from sensors would come from a ROS driver through the same topic, while in the simulation it comes from a script (plugins) that emulates this driver. The abstraction layer in the simulation allows the same application tested in the simulator to be loaded in a real MAV platform.

Figure 7 presents a schematic of the simulation architecture demonstrating how the simulator model interfaces with the ROS nodes created. In summary, trajectory planning node, position controller node, motors plugin and ground truth plugin are used to move the MAV model in the simulation scenario. With the movement, IMU data and camera images are updated and published by their respective plugins. Images topic is subscribed by AprilTag 2 measurement node to generate tags measures. Finally, visual-inertial filter node subscribes IMU data and tags measures to estimate navigation states.

V. Simulations and Results

A. Computational Simulation Parameters

The configuration required by the proposed application is the state and measurement filter noise parameters, that are configured according to the noise characteristics of sensors used. Thus, Table I describes these parameters set up in the implemented filter. Note that they are calibrated to achieve a desirable filter output according to the sensors used in the simulation model.

TABLE I: State and measurement noise parameters.

State noise	$\mathbf{Q} = \mathrm{diag}\{{}^{a}\mathbf{Q}, {}^{g}\mathbf{Q}, {}^{\beta_a}\mathbf{Q}, {}^{\beta_g}\mathbf{Q}\}$
covariance matrix	${}^{a}\mathbf{Q} = 5 \times 10^{-3}\mathbf{I}_3\ (\mathrm{m/s^2})^2$
	${}^{g}\mathbf{Q} = 1 \times 10^{-6}\ \mathrm{rad}^2$
	${}^{\beta_a}\mathbf{Q} = 1.0 \times 10^{-4}\ (\mathrm{rad/s})^2$
	${}^{\beta_g}\mathbf{Q} = 1.0 \times 10^{-6}\mathbf{I}_3\ (\mathrm{m/s^2})^2$
Measurement noise	$\mathbf{R} = \mathrm{diag}\{{}^{s}\mathbf{R}, {}^{\gamma}\mathbf{R}^{i}\}$
covariance matrix	${}^{s}\mathbf{R} = 1.0 \times 10^{-4}\mathbf{I}_3\ \mathrm{m}^2$
	${}^{\gamma}\mathbf{R} = 1.0 \times 10^{-2}\mathbf{I}_3\ \mathrm{m}^2$

B. Hovering Convergence

For hovering convergence analyses, Monte Carlo (MC) simulation with 100 runs is performed. Quad-rotor is stabilized and hovering in a predefined position and filter initial state condition varies around the true state for each MC realization.

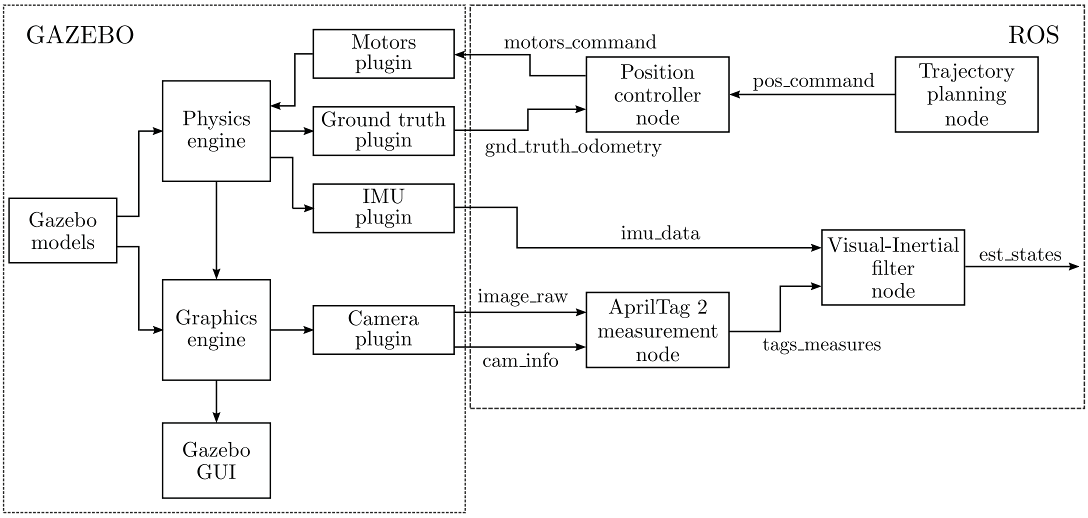

Fig. 7: Scheme of the complete simulation architecture.

Figures 8 and 9 show position and velocity estimation errors for the MC realizations.

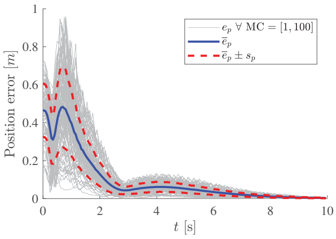

Fig. 8: Position error.

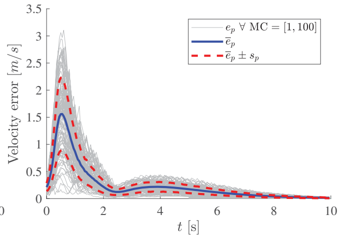

Fig. 9: Velocity error.

Note that position and velocity errors converge in about two seconds of hovering. The peak in velocity is due to the position error convergence. Considering that mean initial position error is about $0.5m$, and other states also have errors, two seconds is considered acceptable for an initial convergence of the states. Another relevant aspect is the fact that error decreases to very small levels. It is related to two facts. Firstly, the platform is very stabilized by a control application using the ground truth as feedback. Secondly, the AprilTag 2 measurement process is very accurate and consistent over time for frames from similar positions.

Figures 10, 11 and 12 show row, pitch and yaw estimation errors, respectively, for the MC realizations.

Attitude parameters present a slower convergence when compared to position and velocity. It happens because attitude is influenced by gyroscope readings in the states' equation and by tag orientation measures in the measurement equations. Thus, as ${}^{\gamma}\mathbf{R}$ is bigger than ${}^{g}\mathbf{Q}$, the update step acts slower in the MAV attitude error convergence. However, as the quad-rotor orientation is used in low-level control, usually to maintain the stability, it is important to have an estimator capable to perceive rapid changes in attitude. It can not be guaranteed measurements in the update step once it executes at a small rate and tags measures are delayed. Thus, a smaller ${}^{g}\mathbf{Q}$ allows the prediction step to reflect the rapid changes perceived by the gyroscope.

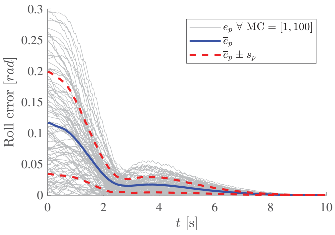

Fig. 10: Roll error.

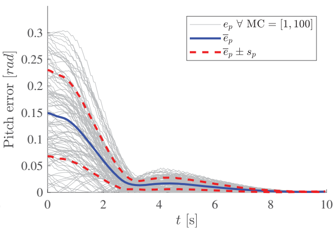

Fig. 11: Pitch error.

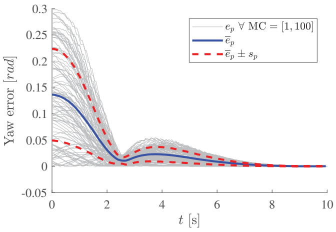

Fig. 12: Yaw error.

C. Dynamic Tracking

In order to evaluate filter convergence and accuracy during trajectories containing fast turning maneuvers, it is necessary to stimulate position changing in the quad-rotor model.

Quad-rotor true initial state is the simulation initial state of the ARDRONE model $\mathbf{x}_0 = [0\ 0\ 0.08\ \mathbf{0}_{1\times 12}]^{\mathrm{T}}$ and the initial estimate state of the filter is defined as $\hat{\mathbf{x}}_{0|0} = [\mathbf{0}_{1\times 15}]^{\mathrm{T}}$.

Figures 13 and 14 preset a top view of the trajectory comparing true path with position estimates from the filter.

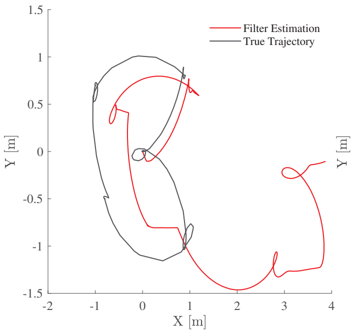

Fig. 13: With no tags.

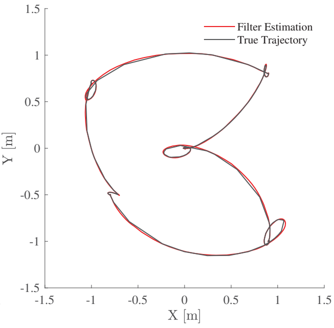

Fig. 14: With tags.

Figure 13 is the filter estimation without tag measures available to update states, it means just prediction step at 100 Hz was responsible to determine the navigation parameters. Note that it diverges in position over time. This divergence is the drift caused mainly by IMU errors propagated in states equations. The numerical method used has a small impact on this drift. However, Figure 14 shows the filter output when tag measures are available and the update step is executed at 10 Hz. Note that tags measures have a considerable impact on the estimation outcome. The position does not drift over time for the entire trajectory and it maintains small errors throughout the time. It ensures the filter is performing its function for position estimation. Figure 15 shows the same behavior for the Z axis.

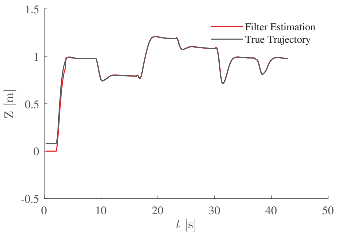

Fig. 15: Z position over time.

Figures 16 and 17 exhibit the error of state estimation in position and velocity, respectively, throughout the simulation time.

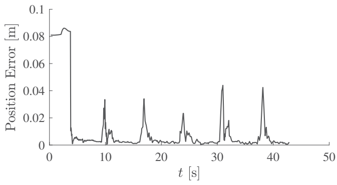

Fig. 16: Position error.

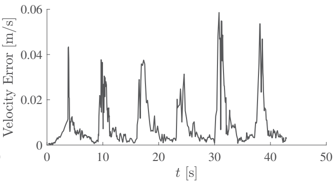

Fig. 17: Velocity error.

Note from Figure 16 that initial position error was around 0.08 m due to the difference between Z position initialization and MAV initial position. In 3 seconds, the MAV camera starts to get tags in the field of view and provide tag measures to the filter, then error decreases quickly. It may also be observed that error peaks in both, position and velocity, happens when MAV is performing fast turning maneuvers. However, these peaks are limited to 0.05 m in position and 0.06 m/s in velocity for the dynamics of this MAV model. These error values are considered acceptable for high-level control applications.

Figures 18 and 19 exhibit the error of state estimation in roll and pitch, respectively, throughout the simulation time.

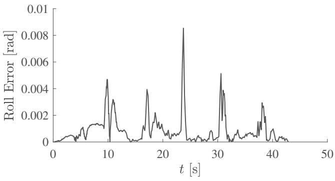

Fig. 18: Roll error.

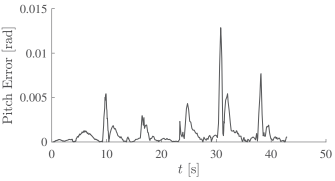

Fig. 19: Pitch error.

Note that Roll and Pitch errors are noisier and converge slower than position. It is due to the less influence of the update step on the attitude estimation as explained in Section V-B. On the other hand, it may be noticed in Figures 18 and 19 that error is limited to 0.009 rad in Roll and 0.014 rad in Pitch, and most of the time both are lower than 0.005 rad.

VI. Conclusions and Future Work

Filter results for hovering convergence show that proposed solution is capable of providing accurate states estimation in approximately two seconds even for highly erroneous starting conditions. On the other hand, dynamic tracking experiment demonstrates that the filter is able to follow the MAV movements with no drift over time. Therefore, it is capable to provide reliable navigation parameters over the entire trajectory. It means that higher level applications can use the filter output.

For future work, implementation in an embedded system would provide a better sense of the filter computational performance. Moreover, development of low and high-level applications using the filter output would give more confidence on the filter use. Finally, a real application for a controlled environment, such as industrial manufacturing cell, with several moving objects and large spaces, using this filter, is the complete proof of its effectiveness for the problem approached.

References

[1] J. Wang and E. Olson, “AprilTag 2: Efficient and robust fiducial detection,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[2] C. N. Taylor, M. J. Veth, J. F. Raquet, and M. M. Miller, “Comparison of two image and inertial sensor fusion techniques for navigation in unmapped environments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 946–958, 2011.

[3] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," 2007.

[4] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE robotics & automation magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[5] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Visual–inertial navigation systems for aerial robotics: Sensor fusion and technology," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 260–275, 2017.

[6] R. Brockers, M. Hummenberger, S. Weiss, and L. Matthies, "Towards autonomous navigation of miniature uav," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 631–637.

[7] G. Loianno and V. Kumar, "Vision-based fast navigation of micro aerial vehicles," in *SPIE Defense+ Security*. International Society for Optics and Photonics, 2016, pp. 98 361T–98 361T.

[8] D. Abeywardena, S. Huang, B. Barnes, G. Dissanayake, and S. Kodagoda, "Fast, on-board, model-aided visual-inertial odometry system for quadrotor micro aerial vehicles," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1530–1537.

[9] S. Kyristsis, A. Antonopoulos, T. Chanialakis, E. Stefanakis, C. Linardos, A. Tripolitsiotis, and P. Partsinevelos, "Towards autonomous modular uav missions: The detection, geo-location and landing paradigm," *Sensors*, vol. 16, no. 11, p. 1844, 2016.

[10] S. M. Chaves, R. W. Wolcott, and R. M. Eustice, "Neec research: Toward gps-denied landing of unmanned aerial vehicles on ships at sea," *Naval Engineers Journal*, vol. 127, no. 1, pp. 23–35, 2015.

[11] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1, pp. 21–39, 2012.

[12] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart, "Vision based position control for mavs using one single circular landmark," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 495–512, 2011.

[13] C. Kemp, "Visual control of a miniature quad-rotor helicopter," Ph.D. dissertation, University of Cambridge, 2006.

[14] M. Asadpour, B. Van den Bergh, D. Giustiniano, K. Hummel, S. Pollin, and B. Plattner, "Micro aerial vehicle networks: An experimental analysis of challenges and opportunities," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 141–149, 2014.

[15] Q. Li, D.-C. Li, Q.-f. Wu, L.-w. Tang, Y. Huo, Y.-x. Zhang, and N. Cheng, "Autonomous navigation and environment modeling for mavs in 3-d enclosed industrial environments," *Computers in Industry*, vol. 64, no. 9, pp. 1161–1177, 2013.

[16] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "The pixhawk open-source computer vision framework for mavs," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, no. 1, p. C22, 2011.

[17] D. Titterton and J. L. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.

[18] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[19] C.-H. Chu, D.-N. Yang, and M.-S. Chen, "Image stablization for 2d barcode in handheld devices," in *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 2007, pp. 697–706.

[20] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2004, vol. 1.

[21] D. Wagner and D. Schmalstieg, "Artoolkitplus for pose tracking on mobile devices," in *Proc. Computer Vision Winter Workshop, 2007*, 2007, pp. 139–146.

[22] M. Fiala, "Artag revision 1, a fiducial marker system using digital techniques," *National Research Council Publication*, vol. 47419, pp. 1–47, 2004.

[23] G. Bradski and A. Kaehler, "Opencv," *Dr. DobbâĂŹs journal of software tools*, vol. 3, 2000.

[24] A. Gelb, *Applied optimal estimation*. MIT press, 1974.

[25] T. B. Schön, "Estimation of nonlinear dynamic systems: Theory and applications," Ph.D. dissertation, Institutionen för systemteknik, 2006.

[26] S. M. Weiss, "Vision based navigation for micro helicopters," Ph.D. dissertation, 2012.

[27] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.

[28] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 2149–2154.

[29] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors - a modular gazebo mav simulator framework," in *Robot Operating System (ROS)*. Springer, 2016, pp. 595–625.