Simulation Processes for Onboard State Estimation in a Small UAV Environment

Marsel Omeri*, Marko Radanovic, Ernesto Emmanuel Santana Cruz, Romualdo Moreno Ortiz

Department of Telecommunications and Systems Engineering, Autonomous University of Barcelona
Carrer dels Emprius, 2, 08202 Sabadell, Spain; *marsel.omeri@uab.cat

SNE 32(1), 2022, 9-14, DOI: 10.11128/sne.32.tn.10592
Received: 2020-11-10 (Selected EUROSIM 2019 Postconference Publication); Revised: 2021-10-18; Accepted: 2021-12-15
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper describes the processes for estimating and analyzing the position states onboard small unmanned aerial vehicles in the low-altitude simulation environment. Those processes are explained using two simulation platforms, Robot Operating System and Gazebo. They comprise different system functionalities from trajectory generation, linear-kinematic trajectory conversion, path controllers to the modeling and configuration of various aerial vehicles, onboard positioning and navigation sensors, and creation and visualization of the simulation environment. We model and simulate the vehicle trajectories to determine 3-D positions from the GPS sensor, along with estimated positions from the fusion of GPS/IMU and Altimeter. The simulated results have provided a dataset on lateral and vertical trajectory profile guidance and prediction in the low altitude airspaces for follow-up research on the common reference altitude determination, as well as the definition of the well clear and collision states for the detect and avoid functions of the small unmanned aerial vehicles.

Introduction

To accommodate the future demand for low-altitude small unmanned aerial vehicles (sUAVs) [1], the previous air traffic management experiences indicate that this demand must be organized to balance traffic efficiency and safety [2,3]. Additionally, sUAV operators demand their missions to operate beyond visual line of sight (BVLOS), requiring autonomous capabilities [4].
This paper presents a test study on real-time simulation processes for estimating and analyzing future sUAV position states.

This estimation is based on the vehicle's trajectory inputs from the modeled sensor systems [7]. We use two simulation frameworks: Robot Operating System (ROS) [8, 9] and Gazebo [10, 11], to generate trajectories and simple pairwise traffic scenarios, together with the implementation of navigation, guidance, and control modules. In particular, the study has applied the GPS/ IMU sensor system with configured tracking rates [12], compliant with the sUAV performance model. In this paper, we use different system functionalities in both frameworks, from the trajectory generation, controllers to the modeling and configuration of different sUAVs, onboard positioning and navigation, and visualization of the simulation environment. The generated sUAV trajectories are further processed for extraction and data analysis to determine the 3-D positions in discrete moments for the kinematic trajectory profile. The simulated results have provided a dataset on the trajectory guidance and prediction in the low altitude airspaces, valid for follow-up research on the common reference altitude determination, definition of the well clear and collision states for the detect and avoid sUAV functionalities. Figure 1 illustrates a process flow of the simulated sUAV trajectory data exchange.

The rest of the paper is organized as follows. Section 1 explains in detail the ROS-Gazebo simulation framework with the main functionalities and the integrated sUAV performance and onboard sensor models, while Section 2 elaborates the data extraction and analysis method developed using the MATLAB Robotics System Toolbox, inclusive of multi-sensor fusion for position estimates. Section 3 describes the simulation results obtained for a given scenario with two sUAV trajectories and the output data analysis as a potential for follow-up research. Finally, concluding remarks and future directions are provided in Section 4.

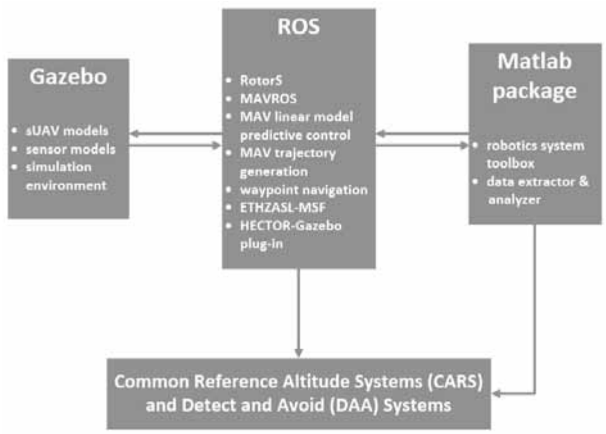

Figure 1: Process flow of the simulated sUAV trajectory data exchange.

1 Simulation Overview

Overall simulation is based on RotorS [14], a framework that enables simulation and tests different sUAVs, sensors, controllers, and state estimation algorithms. Also, it may be extended further to implement high-level operations like collision detection, avoidance, and so on. RotorS is developed based on ROS [15], a rapidly evolving middle-ware in robotics and automation, and Gazebo.

We extend RotorS with some other packages as mentioned above to design an encounter trajectory scenario. Each sUAV feeds odometry data from onboard sensors to the linear MPC and attitude PID controller. To generate and navigate the trajectory, we use the waypoint_navigation package [16], which also depends on the trajectory_generation_ros package. The former allows the user to input 3D ENU coordinates as trajectory input or GPS waypoints.

1.1 sUAV models

The Gazebo framework contains an extensive base of the sUAV performance types, from different MAV types, low-altitude-short-endurance (LASE), low-altitude-long -endurance (LALE), medium-altitude-long-endurance (MALE), to high-altitude-long-endurance (HALE) types. In this study, the simulation and testing are done on the LALE sUAV types. That sUAV is commonly referred to as a drone and is characterized by the local mission functions covering the flight ranges up to 10 km. The following three sUAV models are selected for the simulated trajectory guidance and testing scenarios: Firefly, Pelican, and Hummingbird (Figure 2). Table 1 lists their performance characteristics (AGL – Above Ground Level).

sUAV models from *the AscTec family*	Fire-fly	Peli-can	Hum-mingbird
maximum payload weight [kg]	0.6	0.65	0.2
maximum take-off weight [kg]	1.6	1.65	0.71
maximum flight time/endurance [min]	14	16	20
maximum airspeed [m/s]	15	16	15
maximum climb rate [m/s]	8	8	5
maximum range [m]	4500	4500	4500
maximum altitude AGL [m]	1000	1000	1000

Table 1: Performance characteristics of the selected sUAV models [17].

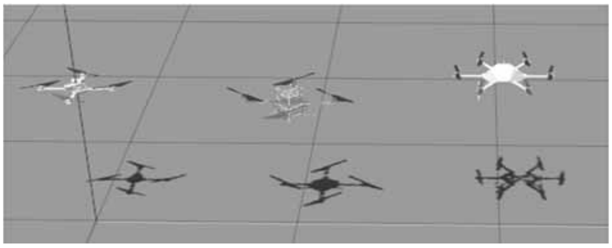

Figure 2: Three selected AscTec-family sUAV models: Firefly (on the left), Pelican (in the middle), and Hummingbird (on the right).

1.2 Sensor models

In this paper, the sUAV is equipped with the following onboard sensors:

- IMU-based on hector gazebo plugins
- GPS – based on hector gazebo plugins
- Altimeter- based on hector gazebo plugins

The simulation environment is designed as such; the GPS reference is the same as the launching point of sUAV. We implement a sensor fusion module for position estimation based on a ROS multi-sensor fusion (MSF) package [18]. This package uses EKF, where IMU is feeding data for the Prediction State, and GPS/Altimeter are the Update State sensors. Typically, GPS/IMU produces good results for outdoor environments, and it is the most common onboard sensors combination when it comes to commercial sUAVs. In addition, we include a Barometer to see how it affects altitude estimation. Table 2 illustrates the parameters for each sensor.

Sensor	Update Rate [Hz]	Simulated Noise (Gaussian) [m]
GPS	10	0.01, 0.01, 0.01
IMU	100	0.35, 0.35, 0.30
Altimeter	20	0.1

Table 2: Sensor parameters.

1.3 Simulation Environment

Gazebo allows a visual, 3-D simulation of a scenario consisting of cyber-physical systems, e.g., ground-rovers, UAVs, and other objects like simple obstacles, or surrounding environmental elements, together composing what it is called a Gazebo world. Thus, in Gazebo, realistic scenarios for cyber-physical systems, including the surrounding environment, can be created.

A sUAV model with its properties presents the main object in Gazebo. From the physical point of view, a single sUAV as point mass is linked to a reference coordinate system that can be set in the GPS frame or as the positive Euclidean space (i.e., ENU – East, Nord – Up). Figure 3 illustrates the launching phase of a single sUAV in the ENU coordinate system. The blue line presents the Up direction, i.e., the positive z-axis, the green one the Nord direction (i.e., the positive y-axis). In contrast, the red one denotes the East direction (i.e., the positive x-axis). Equivalently, in the GPS frame, the East and Nord directions are replaced by the geographic coordinates: longitude and latitude.

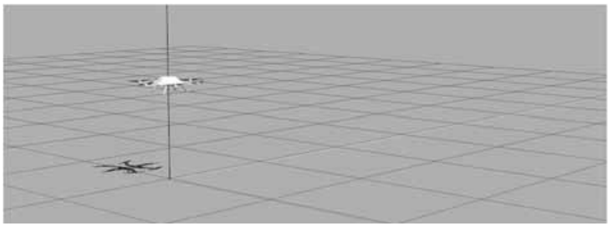

Figure 3: Single sUAV operate in ENU coordinate system.

By default, Gazebo allows a visual 3-D simulation of a single UAV trajectory only, focusing on the guidance and control aspects of trajectory profile configured by the inputs previously explained. Since the Gazebo framework is developed mostly for the sUAV types that operate over the local, short-range mission profiles (Figure 4), the environment is customized as per this type of trajectory. Instantaneous and significant heading changes characterize the local, short-range profiles.

The most crucial aspect of visualization is altitude control. Such controller onboard sUAV has one of the managing functions that replace the control of a remote human pilot on the ground. The trajectory profile, either created as the 4-D structure or 3-D structure with assigned airspeed, must be complied with the principle of creating:

1. starting/launching waypoint and following waypoint,
2. second last and last waypoint.

In both cases, the waypoints' geographic location (latitude and longitude) must be maintained constant while allowing an UAV to reach the required altitude AGL or descent to the ground, i.e., changing only the value of z-coordinate. When the altitude controller achieves the needed elevation, it activates to maintain the constant vertical profile or change it if required by the planned trajectory (Figure 4).

Gazebo framework allows the simulation of a single sUAV by default. However, it is possible to extend the simulation to two or more sUAV trajectories and create a potential traffic scenario. Regardless of the number of trajectories added in simulation (2, 3, etc.), each added sUAV is controlled and managed with respect to the reference sUAV, i.e., the one linked to the reference coordinate system.

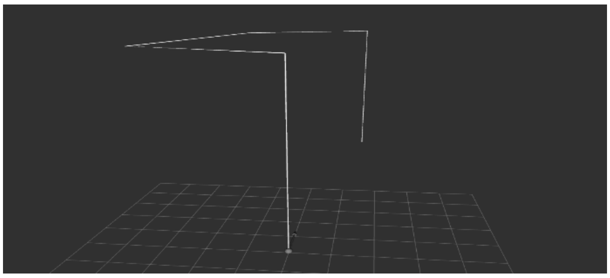

Figure 4: Local, short-range mission profile supported by the altitude controller onboard sUAV.

Nevertheless, the functional requirements of simulated sUAVs and the cooperative or non-cooperative task nature are out of this paper's scope. To conclude, in this study, only two sUAVs with their 3-D trajectory profiles have been further analyzed, creating a pairwise traffic scenario (Figure 5).

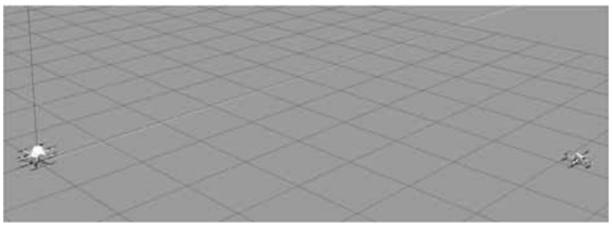

Figure 5: Generation of another sUAV in Gazebo with respect to the reference sUAV

2 Data Extraction and Analysis

This section explains in detail the data extraction procedure and analysis based on the GPS-IMU-Altimeter data capturing. It further elaborates on the multi-sensor fusion (MSF) 3D-position estimation method.

2.1 Data extraction and analysis

In this simulation study, we have recorded the published data from the sensors like GPS, IMU, Altimeter, and the data from the MSF method. These output data are saved as *rosbag* data files. For demonstration purposes, we saved the data from UAV1 and UAV2 in a file named *firefly_hummingbird_scenario1.bag*. To obtain the former, we have created a *ROS launch_file* containing the topics which we are interested in analyzing:

- firefly/fix and firefly/hummingbird – containing GPS position values in WGS84 coordinate frame,
- firefly/gps/points and hummingbird/gps/points – containing GPS position values in ENU coordinate frame,
- firefly/MSF/UpdatePose and hummingbird/MSF/UpdatePose - containing position values in ENU coordinate frame.

There have been used two different methods to extract the information from *firefly_scenario1.bag*:

- writing a *ros_python* node which converts bag file to a CSV file,
- using Robotics System Toolbox in Matlab.

By this procedure, we want to test the interoperability of ROS with other simulation frameworks, which we tend to use in our future research. Figure 6 illustrates the data extraction procedure. The functions have been generated based on the Robotics Toolbox to read messages from different ROS/Gazebo sensors. These messages are converted into Matlab files (.mat) for later processing. Also, we include a ROS *node* written in Python, which allows the output data to be saved in CSV format

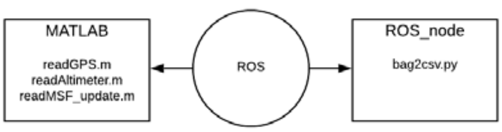

Figure 6: Data extraction procedure from ROS/Gazebo sensor.

2.2 Sensor fusion for 3-D position estimation

The main objective of this simulation work was to see how the altitude controller would preserve the altitude of the given trajectories. Therefore, we have also included onboard the sUAV an altimeter sensor to increase the accuracy of estimation. These data are quite sensitive for different applications related to inspection, coverage area, defining standard reference altitude for sUAVs, etc.

The estimation process has been fused a GPS/IMU sensors with an Altimeter using the MSF ROS package, illustrated in Figure 7. The MSF working procedure is explained in chapter 2.2, and, in this case, it gives 3D positions for both sUAVs.

Figure 7: Sensor fusion process.

3 Simulation Results

This section presents the data output as results obtained for the simulated trajectories of a pair of sUAVs. It analyses the sigma values, i.e., the lateral and vertical trajectory profile errors as differences between the planned and estimated 3-D positions, and discusses the future their potential use in the implementation in the operational domain. All simulations and data processing were run on a PC with Linux Ubuntu 18.04.2 (64bit) processor Intel(R) Core (TM) i7-8700 CUP @ 3.2GHz x 12, 16 GB RAM, and Nvidia Graphic Card Quadro P1000. The codes used for ROS/Gazebo were written in C++, Python, and XML, whereas MATLAB and Python functions were developed for data extraction.

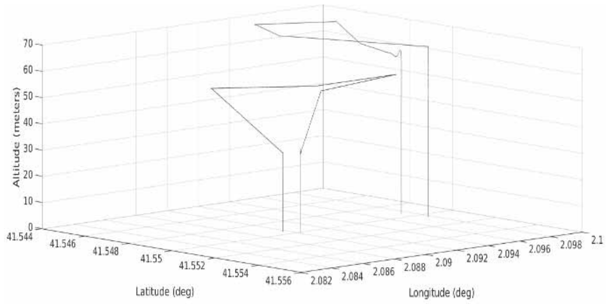

Figure 8: GPS 3-D tracking profile.

3.1 Simulated data output and estimation

For this simulation study, two LALE sUAV trajectories have been created in a 3-D configuration, governed by the initial airspeed value of 15 m/s and the initial acceleration of 3 m/s^2 (both for sUAV1 and sUAV2). The trajectory waypoints have been planned as per Table 3.

The simulation scenario output data are illustrated in Figure 8 and Figure 9. Input data files for two sUAVs are denoted as *drone1_scenario1.yaml* and *drone2_scenario1.yaml*, respectively.

sUAV_ID	Latitude [°]	Longitude [°]	Altitude AGL [m]
	41.549582	2.089708	0.0
	41.549582	2.089708	30.0
	41.553057	2.089743	60.0
sUAV1	41.553895	2.097228	70.0
	41.550498	2.097869	70.0
	41.550917	2.093766	60.0
	41.549370	2.090490	30.0
	41.549370	2.090490	0.0
	41.550493	2.097766	0.0
	41.550493	2.097766	65.0
	41.553465	2.092379	68.0
sUAV2	41.552516	2.089675	68.0
	41.550276	2.091821	68.0
	41.551607	2.094963	65.0
	41.550748	2.096422	60.0
	41.550748	2.096422	0.0

Table 3: Trajectory waypoints for sUAV1 and sUAV2.

After the simulation, the data were extracted from GPS/IMU and then plotted their trajectory profiles. It is shown that the scenario was designed to have changes in both the altitudes and heading directions in the case of both sUAVs. A subplot of the data extracted from the altimeter has been included to emphasize the fluctuations in altitude. Figure 10 reports the estimated 3-D positions of trajectories based on GPS/IMU and altimeter sensor fusion. These results are further analyzed to identify the deviations on the trajectory profiles.

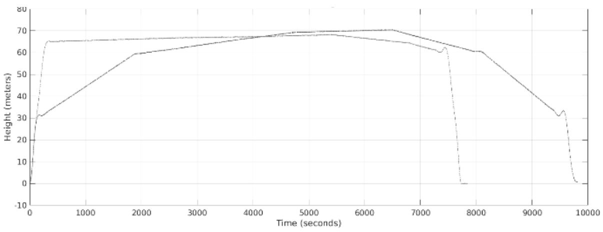

Figure 9: Altimeter tracking profile.

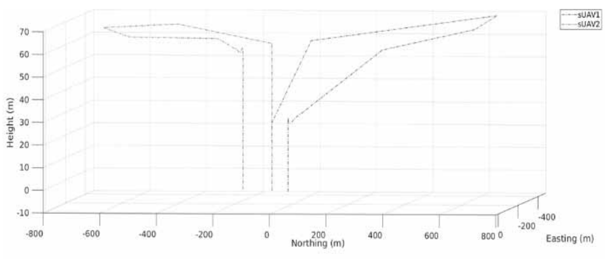

Figure 10: Estimated 3-D positions tracking profile.

3.2 Estimation analysis

This section describes and demonstrates the position estimation errors for vertical and horizontal profiles. The errors are defined as the difference between GPS_output and MSF_output data, where MSF is the module fusing GPS/IMU and Altimeter sensors.

We calculate and plot the lateral and 3-D errors by simply calculating deviations on trajectories. Moreover, a plot of error in altitude (z-axis) is provided to give a complete picture of the estimation (Figure 11, Figure 12, and Figure 13). Finally, the calculation is given as follows:

$$\sigma_{xy} = \sqrt{\sigma_x^2 + \sigma_y^2} \tag{1}$$

$$\sigma = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \tag{2}$$

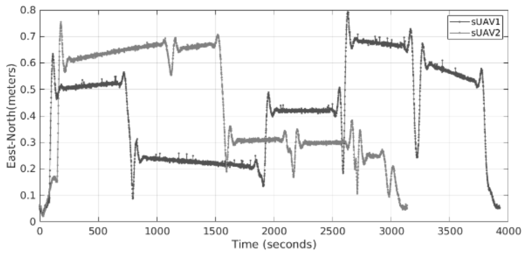

Figure 11: Horizontal profile errors over time for sUAV1 (blue) and sUAV2 (red).

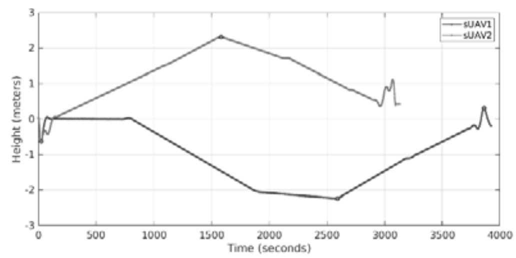

Figure 12: Vertical profile errors over time for sUAV1 (blue) and sUAV2 (red).

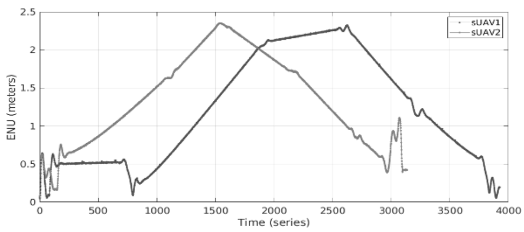

Figure 13: 3-D profile errors over time for sUAV1 (blue) and sUAV2 (red).

4 Conclusion and Future Work

This paper focuses on the test study on real-time estimation of the relative, pairwise sUAV 3-D trajectory positions in the extended ROS/Gazebo simulation framework.

The assessment is performed based on the planned trajectory inputs, using the modelled GPS-IMU-Altimeter sensor fusion to identify the lateral and vertical profile errors and resulting 3-D profile errors. Results indicate that the combined sensor fusion for the onboard tracking and guidance functions provides meaningful insight on the future investigation of the standard altitude reference for a multi-sUAV urban environment, along with a possible testing standard for the definition of the Well Clear and collision detection and avoidance thresholds in different scenario types, such as encounter, intersection or overtaking.

Acknowledgments

This work was supported by the Ministry of Economy, Industry, and Competitivity through the project "GESTION AUTOMATIZADA DE TRAFICO AEREO PARA RPAS" (TRA2017-88724-R).

References

[1] Ren L, Castillo-Effen M, Yu H, Johnson E, Yoon Y, Takuma N, Ippolito CA. Small unmanned aircraft system (sUAS) categorization framework for low altitude traffic services. In: Proceedngs of the 36th IEEE/AIAA Digital Avionics Systems Conference (DASC) 2017, pp. 1-10. St. Petersburg, USA, September 2017. doi: 10.1109/DASC.2017.8101996.

[2] Prevot T, Rios J, Kopardekar P, Robinson JE, Johnson M, Jung J. UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations. In: 16th AIAA Aviation Technology Integration and Operation Conference, AIAA Aviation 2016. Washington, USA, June 2016. doi: 10.2514/6.2016-3292.

[3] Jiang T, Geller J, Ni D, Collura J. Unmanned Aircraft System traffic management: Concept of operation and system architecture. International Journal of Transportation Science and Technology 5(3), 123–135 (2016).

[4] Brooker P. Introducing Unmanned Aircraft Systems into a High Reliability ATC System. Journal of Navigation 66(5), 719-735 (2013).

[5] Zhan W, Wang W, Chen N, Wang C. Efficient UAV Path Planning with Multiconstraints in a 3D Large Battlefield Environment. Mathematical Problems in Engineering 2014, 1-12 (2014). doi: 10.1155/2014/597092.

[6] Weiss S, Achtelik MA, Lynen S, Achtelik MC, Kneip L, Chli M, Siegwart R. Monocular vision for long-term micro aerial vehicle state estimation: A compendium. Journal Field Roboics 30(5), 803-831 (2013). doi: 10.1002/rob.21466.

[7] Giovanneschi F, et al. An adaptive sensing approach for the detection of small UAV: first investigation of static sensor network and moving sensor platform. In: Proceedings of the Signal Processing, Sensor/Information Fusion and Target Recognition XXVII, 106460S. Orlando, USA, April 2018. doi: 10.1117/12.2304758.

[8] Badger J, Gooding D, Ensley K, Hambuchen K, Thackston A. ROS in space: A case study on robonaut 2. Robot Operating System (ROS), 343-373 (2016). Springer, Cham.

[9] Chen H, Kakiuchi Y, Saito M, Okada K, Inaba M. View-based multi-touch gesture interface for furniture manipulation robots. Advanced Robotics and its Social Impacts, IEEE, 39-42 (2011).

[10] Koenig N, Howard A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2004, pp. 2149-2154. Sendai, Japan, September 2004. doi: 10.1109/IROS.2004.1389727

[11] Nagaty A, Saeedi S, Thibault C, Seto M, Li H. Control and navigation framework for quadrotor helicopters. Journal of intelligent and robotic systems 70(1-4), 1-12 (2013).

[12] Desa Hazry, Mohd Sofian, A. Zul Azfar. Study of Inertial Measurement Unit Sensor. In Proceedings of the International Conference on Man-Machine Systems (ICoMMS), Batu Ferringhi, Penang, Malaysia, October 2009.

[13] Caron F, Duflos E, Pomorski D, Vanheeghe P. GPS/IMU data fusion using multi-sensor Kalman filtering: introduction of contextual aspects. Information Fusion 7 (2), 221-230 (2006).

[14] Furrer F, Burri M, Achtelik M, Siegwart R. RotorS - A modular gazebo MAV simulator framework. Studies in Computational Intelligence 2016, 595-625 (2016).

[15] Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R, Ng AY. ROS: an open-source Robot Operating System. In ICRA workshop on open source software, 3(3.2), p. 5 (2009).

[16] GitHub Website, 3. https://github.com/ehz-asl/waypoint_navigator.

[17] ASCENDING TECHNOLOGIES Website, http://www.asctec.de.

[18] http://wiki.ros.org/ethzasl_sensor_fusion/Tutorials/Introductory%20Tutorial%20for%20Multi-Sensor%20Fusion%20Framework