Visual Servoed Autonomous Landing on a Surface Vessel

Nicolò Bastianelli Naticchi*, Marco Baglietto*†, Alessandro Sperindé*†, Enrico Simetti*†, Giuseppe Casalino*†
*DIBRIS, University of Genova, Via Opera Pia 13, 16145 Genova, Italy
†Interuniversity Research Center on Integrated Systems for the Marine Environment Via Opera Pia 13, 16145 Genova, Italy

***Abstract*—In this paper, we describe a quadrotor able to land autonomously on a moving platform by employing on-board sensors. The full pose of the platform is estimated using a vision system. Furthermore, an ultrasonic sensor is used to compute the relative vertical position and velocity of the platform w.r.t. the quadrotor. This redundant information is used to achieve greater robustness during the final landing phase. The landing algorithm is presented and an overview of the overall architecture is provided. Software-in-the-loop simulations have been performed to evaluate the performance and to analyse in a safe way the entire landing procedure. We have finally tested our landing system in a real world environment using a customized quadrotor. Results of the landing procedure performed with real quadrotor are presented.**

I. Introduction

The Monitoring And REscue Automation (MAREA) project aims to develop teams of advanced autonomous robotic systems. Regarding missions performed on sea surface, a possible structure for a marine area patrolling system consists of team made of a surface vehicle and a quadrotor. Such kind of structure can be used for protection and security of marine areas, as well as humanitarian search and rescue activities [12]. The performance and reliability of an autonomous mission involving quadrotors is mainly related with the capability of succeeding in landing [1]. Autonomous landing is one of the most challenging tasks and hence it is often performed by human operator. The scenario addressed is based on a customized quadrotor, which is free to fly on the sea surface. When the quadrotor completes its mission, it autonomously lands on the catamaran platform. The aim of this work, part of the MAREA project, is to improve the current architecture in the described scenario.

A. Related Work

The main objective of this work is to design a quadrotor capable of landing autonomously on a surface vehicle. More in detail, the surface vehicle is a catamaran, equipped with a landing platform and a Global Positioning System (GPS) receiver for the navigation system. To fulfill the task, a possible solution is to provide to the quadrotor the platform GPS coordinates. However, the classical GPS employment cannot guarantee an autonomous safe landing. Most of the time the information taken from the GPS is too inaccurate for complex manoeuvres [13]. In addition, the platform is subjected to an unpredictable oscillatory dynamics caused by the sea surface. A reasonable approach is to enhance the performance of the architecture using a vision system which can be adopted as primary or secondary localization source. GPS is employed to drive the quadrotor close to the point of interest and, then, vision system is capable of obtaining the relative pose of the target platform w.r.t. the quadrotor. In [9] an Extended Kalman Filter is developed to combine data coming from different sensors (inertial navigation, GPS, and visual sensor) and build a navigation system useful during landing procedure. Visual information are added to the navigation system only when the landing deck is visible; this helps to increase the accuracy of the landing phase. The use of computer vision for closing the loop is especially useful when the landing pad is in an unknown position or it is moving [7]. In this way, it is possible to directly detect a tracking error w.r.t. the target, instead of providing coordinates of target point w.r.t. an absolute frame. To make the platform recognizable during the platform approach, an artificial landmark is usually placed on it. For instance, in [8] a new helipad is designed which is composed of a cross inside a circle, which in turn is within a square. A vision-based autonomous landing is presented on a moving platform. For the validation phase, the platform was carried by a mobile robot moving on the ground. However, the drawback of computer vision from on-board camera is the requirement of having the platform always recognizable during the entire landing procedure. This can be solved via platform motion estimation [8] or having some features which always guarantee the platform tracking during the entire landing phase. In [10] a specific marker is adopted which is based on a series of concentric circles. These circles allow to detect the platform from closeby. Another possible solution is presented in [11], where a landing platform having tags with different dimensions is designed. The larger tags permit the detection from a higher altitudes, whereas the smaller ones from lower altitudes. *AprilTag system* was chosen for the design of the landing pad and it is used as visual fiducial system [4].

B. Contribution

The Interuniversity Center of Integrated Systems for the Marine Environment (ISME) has been focusing on marine

This publication was also made possible by NPRP grant #10-0213-170458 from the Qatar National Research Fund (a member of Qatar Foundation). The findings herein reflect the work, and are solely the responsibility of the authors.

robotic systems. An autonomous catamaran designed for real application was developed [2]. Our contribution has been the development of an autonomous guidance controller for quadrotors. It is capable of managing essential navigation actions typical of a quadrotor flight. As output, it generates desired commands for the autopilot. The platform approach to the catamaran has been tackled implementing a horizontal tracking and a vertical compensation of the landing site. Moreover, to handle various system behaviours, the whole procedure is organized with a finite state machine. The preliminary phase involved indoor testing, where the position measurements of the platform and the state estimation of the quadrotor are obtained using a motion capture system [3].
The employment of the motion capture system for the pose estimation of the platform, makes the system not versatile to any situation. In this paper, we present the integration of an on-board vision system and an ultrasonic sensor to the current quadrotor architecture. The system is now able to autonomously land on a moving platform using only on-board sensing. Furthermore, being a maritime application, we have also taken into account the unpredicted dynamics of the platform with the addition of possible roll and pitch variations. For the landing procedure, it is only necessary to know an approximate position of the landing site. Once the quadrotor reaches the landing site, it autonomously lands on the platform using monocular camera and an ultrasonic sensor.

II. System Overview

The overall architecture, shown in Fig. 6, is composed of different modules that are detailed in the following sections:

- *State estimation*
- *Vision system*
- *Ultrasonic Sensor*
- *Guidance controller*
- *Finite state machine*

A. State Estimation

In the indoor scenario, the quadrotor pose estimation is performed using a motion capture system, Optitrack[1]. It allows to compute position and orientation of the quadrotor in the space. The setup is composed of eight infrared cameras and passive reflective markers are situated on the drone's body. The position and the orientation of the drone is sent into the PX4[2] autopilot. These data are hence passed to the sensor fusion module. The sensor fusion part is a Kalman Filter which takes, as input, different sensors to estimate the quadrotor state. In our case, motion capture data and IMU measurements are considered.

B. Vision system

We employ an on-board vision system to estimate the pose of the moving platform w.r.t. the quadrotor. To simplify the platform detection, it is equipped with visually distinguishable tags. We chose to adopt an open-source vision system (*April-Tag system*) since it is widely used by the software community, it is well documented and robust [14] [15]. *AprilTag system* is able to compute the 3D position and orientation of specific tags w.r.t. the camera [4]. Regarding the landing platform, the employment of a single tag does not guarantee its identification during the whole landing procedure.

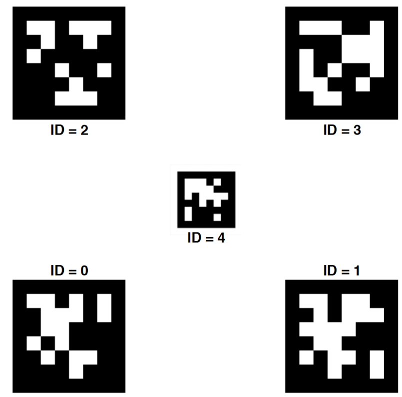


Fig. 1: Designed platform.

For our purposes, it is fundamental that the platform is recognizable as much as possible throughout the landing phase. Accordingly, we arrange four April tags which create a square shape and another one, the smallest, in the centre of the platform, as shown in Fig. 1. As the quadrotor approaches the landing pad, the biggest tags gradually leave the camera field of view while the smallest tag starts becoming detectable.

Platform Centre Estimation: The output of the vision system consists of the transformation matrices of each tag present in the scene w.r.t. the camera. In order to have the pose of the platform's centre, it is possible to compute offline the transformation matrices between the tag in the center, ${}^i_4\boldsymbol{T}$, and the i-th tags which compose a square shape, namely ${}^0_4\boldsymbol{T}$, ${}^1_4\boldsymbol{T}$, ${}^2_4\boldsymbol{T}$, ${}^3_4\boldsymbol{T}$. They are computed using motion capture system and their position w.r.t. the platform's centre can be precisely determined. The indexes in the transformation matrices represent the ID's of the tags. With this approach, just computing a post-multiplication, it is possible to have the pose of the platform's centre w.r.t. the camera:

$$ {}^c_4\boldsymbol{T} = {}^c_i\boldsymbol{T} \cdot {}^i_4\boldsymbol{T} \tag{1}$$

During the platform tracking it is possible to have more than one measures related to its centre; if the platform is clearly visible, five different measures are available. In principle all of these information are equally correct and a solution is to merge together these data. We use a weighted average considering how much are big the tags in the image. The larger is a tag in the image, the more reliable is considered its estimate. The weights are then represented by the area of each tag in the camera frame.

[1]Please refer to: https://www.optitrack.com
[2]Please refer to: https://px4.io

As output, we obtain the pose error of the platform centre w.r.t. the quadrotor.

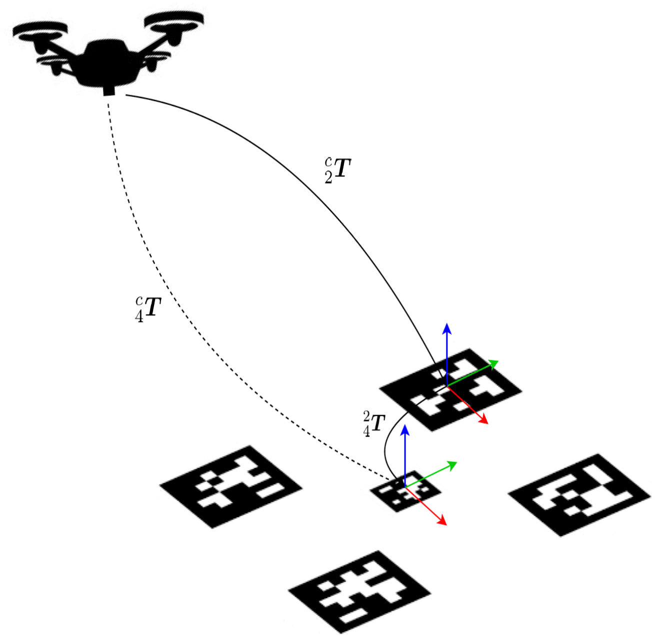


Fig. 2: Platform centre estimation.

The error is then injected into the guidance controller, which in turn generates desired commands for the autopilot. Figure 3 shows the adopted visual servoed architecture which is typically called *Dynamic position based look and move* [16]. The platform tracking can be achieved without the use of the motion capture system.

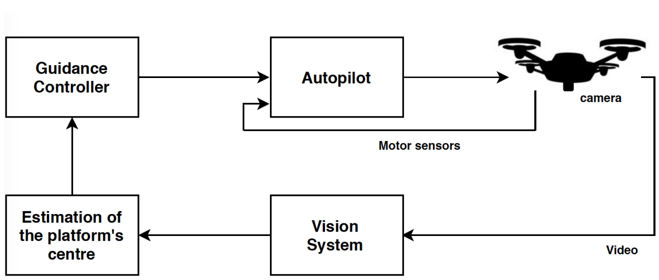


Fig. 3: Adopted visual servoed architecture.

C. Ultrasonic Sensor

Ultrasonic sensor is added to increase robustness during the landing. The quadrotor has been equipped with ultrasonic sensor pointing downward, which should be activated when the quadrotor is aligned vertically with the platform. Ultrasonic sensor is only used to measure the relative vertical distance between drone and platform. In addition, we estimate their relative vertical velocity by designing a basic Kalman Filter.

Relative Vertical Velocity Estimation: We design a basic Kalman Filter with relative vertical position and velocity as state variables.
Assuming the velocity as a constant during the acquisition time, the system model becomes:

$$\begin{cases} \boldsymbol{x}_k = \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{bmatrix} + \boldsymbol{w}_{k-1} \\ z_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + v_k \end{cases} \tag{2}$$

The output of the system is the distance provided by the sensor. The noise variance on the measurement vector, $\boldsymbol{R}$, is determined based on the variance of the raw data measurement distance doing repeated experiments. After having set $\boldsymbol{R}$, the input noise covariance matrix $\boldsymbol{Q}$ is tuned doing repeated tests. It is important to underline that the variables, $\boldsymbol{R}$ and $\boldsymbol{Q}$ are obtained using the motion capture system as reference. Following Figures, 4 and 5 compare the distance and the velocity obtained by motion capture system, by the ultrasonic sensor and by the Kalman Filter.

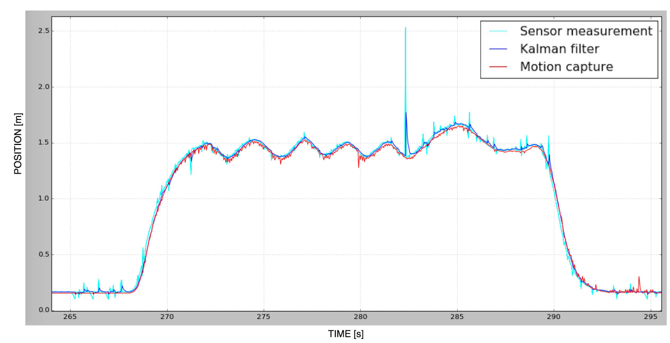


Fig. 4: Response time filtering performance by the setpoint of random changing position.

From Figure 4, it can be noticed that the raw sensor data have very high module noise peaks. If the velocity was estimated directly on the raw data, this would produce a very high instantaneous velocity for the quadrotor. The filtered signal attenuates the module of those peaks.

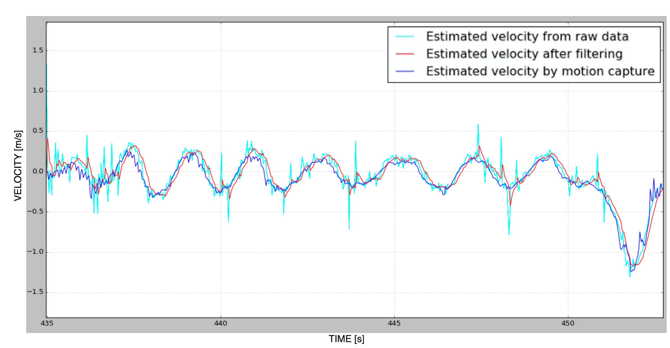


Fig. 5: Response time filtering performance by the setpoint of random changing velocity.

Figure 5 compares the estimated velocity coming from the Kalman filter, without filtering and the velocity obtained by the motion capture system used as reference. As conclusion, the relative altitude and the relative vertical velocity between the platform and the quadrotor are estimated at 30 Hz.

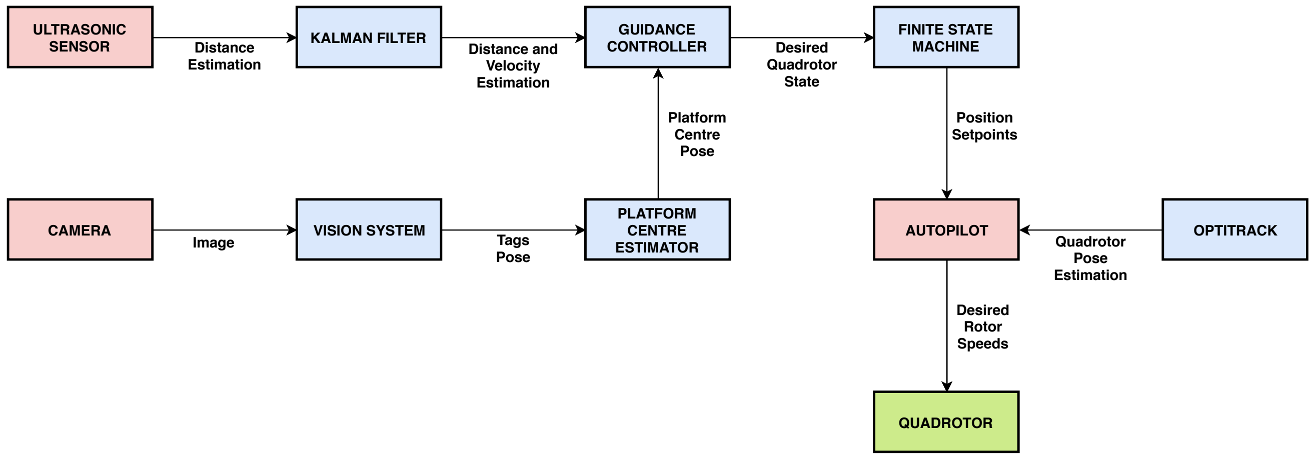


Fig. 6: Overall system description: hardware modules are represented by red boxes; software modules are represented by light blue boxes; quadrotor is represented with green box.

D. Guidance Controller

The guidance controller is strictly linked with the Navigation system and with the flight controller/autopilot [7]. In this work, the navigation system is composed of sensors inside the flight controller board (PixHawk [3]) and a motion capture system. The guidance controller takes, as input, data from the navigation system to know the actual pose of the quadrotor and, based on mission goals, it generates reference commands for the autopilot. The autopilot, based on the position setpoints, computes the desired rotor speeds for the drone whilst maintaining drone stability. The developed guidance controller is a software framework which deals with generating autonomous flight actions. It is a modular open-source architecture written in C++. It is composed of different modules which internally comunicate via Lightweight Communications and Marshalling (LCM) middleware [6].

Our architecture takes as input a finite set of tasks. Each of them represents a basic autonomous action that the quadrotor can perform. A sequence of tasks represents an entire mission. The finite set of tasks are encoded in a text file, loaded offine, which is parsed by the architecture obtaining, as output, the type of action and its desired references. The described architecture implements essential navigation actions which typically involve a generic flight of a quadrotor (*taking-off*, *climbing*, *cruising*, *descending*, *landing*). However, this paper mainly focuses on automonous landing action.

The landing procedure is managed with a finite state machine II-E which is called at each iteration. Each state represents a system behaviour; each arc represents a transition from one state to another and the transition is allowed only if the switching condition on the arc is verified. However, the algorithm is mainly composed of two phases explained in sections, II-D1 and II-D2. At state machine level, such continuous changing of states allows to manage the horizontal tracking and vertical compensation of the platform as separate tasks.

[3]Please refer to: http://pixhawk.org

1) Horizontal Platform Tracking: This phase is activated in the *Hold* state of the finite state machine. It allows the alignment of the drone's center of mass with the landing platform on the Π_{xy}. Defining the platform and the drone horizontal position respectively, $\boldsymbol{p}_{plat,xy}$ and $\boldsymbol{p}_{d,xy}$; the horizontal position error becomes:

$$\hat{\boldsymbol{e}}_{p,xy} = \boldsymbol{p}_{plat,xy} - \boldsymbol{p}_{d,xy} \tag{3}$$

The horizontal error is estimated by the on-board vision system and we design a PI regulation to produce position setpoints for the autopilot:

$$\boldsymbol{p}^*_{d,xy} = \boldsymbol{p}_{plat,xy} + K_P \cdot \hat{\boldsymbol{e}}_{p,xy} + K_I \cdot \int \hat{\boldsymbol{e}}_{p,xy} \tag{4}$$

Starting from the left, the first term is the starting condition of the integrator, the others two terms represent proportional and integral part of the controller.

2) Vertical Platform Compensation: This phase is triggered in the *Compensation* part of the finite state machine, shown in Fig. 7. It regards the compensation of possible platform oscillations while the quadrotor gets closer to it. It is achieved maintaining the relative vertical velocity between the drone and the platform to a desired value, $v^{desc}_{d,z}$, during the landing procedure. The desired descent rate is determined using a linear relation based on the vertical distance from the platform $\hat{e}_{p,z}$. Therefore, the desired vertical velocity with a proportional part on the velocity error, becomes:

$$v^*_{d,z} = v^{desc}_{d,z} + K_P \cdot \hat{e}_{v,z} \tag{5}$$

Our architecture produces position setpoints to inject into the autopilot. Hence, the desired position becomes:

$$p^*_{d,z} = p_{d,z} + v^*_{d,z} \tag{6}$$

In this phase, only ultrasonic sensor is employed for the estimation of the relative altitude between quadrotor and platform $\hat{e}_{p,z}$. Vision system data is not used at this stage since ultrasonic sensor gives the same information but with higher frequency. Subsequently, with the relative vertical distance obtained from the ultrasonic sensor, we estimate the relative vertical altitude velocity between the quadrotor and the platform $\hat{e}_{v,z}$ to fulfill the control law.

E. Finite State Machine

The finite state machine module manages the behavior of the quadrotor during the landing procedure. It has seven states: *initialization*, *inspection*, *holding*, *descending*, *ascending*, *compensation*, *landing* respectively. Figure 7 describes the finite state machine with its states and the respective transitions triggered by boolean algebra operations.

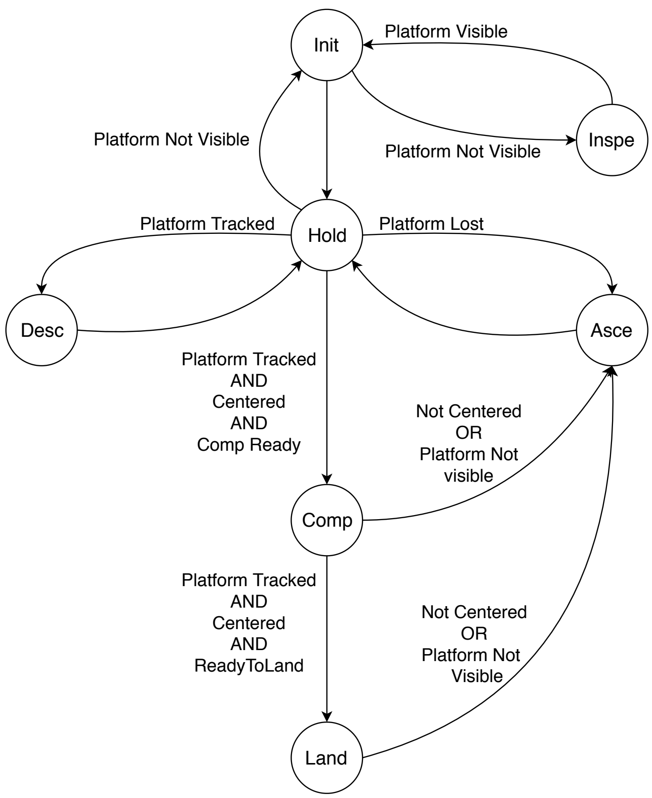


Fig. 7: State machine diagram.

1) Initialization: It is the entry point of the landing procedure. The quadrotor reaches its maximum tracking altitude. Once it is hovering, the guidance architecture starts to acquire vision data. If the platform is not visible, the state machine jumps into the inspection state.

2) Inspection: The quadrotor starts to explore the area at a fixed height, performing circles with an increasing size of the radius. This mode concludes when the quadrotor detects the landing platform. This process guarantees the success of the action even if we initially know an approximate position of the platform.

3) Holding: The quadrotor tracks horizontally the platform using visual feedback according to section II-D1. The transition to another state is possible if the switching conditions on the arcs are verified.

4) Descending: In this state, the quadrotor autonomously computes altitude waypoints to reach the minimum horizontal tracking altitude.

5) Ascending: In this state, the quadrotor autonomously computes altitude waypoints to reach the maximum horizontal tracking altitude. It mainly manages possible visual occlusion of the platform.

6) Compensation: The quadrotor is tracking horizontally the platform and at the same time it starts the vertical compensation according to the section II-D2. In this state, the relative vertical distance between quadrotor and platform is performed using ultrasonic sensor due to its high frequency update II-C. In this mode, the architecture checks also if roll and pitch platform variations are under a given threshold. If so, the system jumps into the final state.
The roll and pitch platform fluctuations are estimated using the vision system II-B.

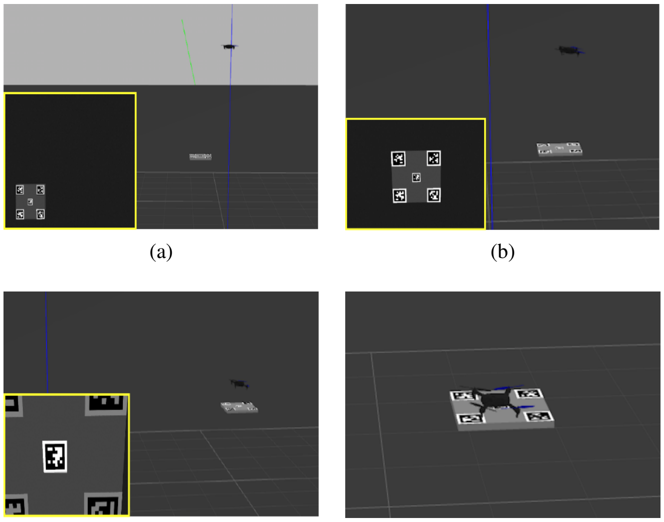
7) Landing: In this phase, the quadrotor is close enough to the platform. We employed on-board ultrasonic sensor to estimate the relative vertical distance between the quadrotor and the landing site. The motors are set to the minimum rate if the relative vertical distance is under a given threshold, 10 cm.

III. Experiments

We replicate the same experiments set to [3]. In the previous works, we assumed to know the exact position of the platform. Currently, the platform pose estimation is computed on-board using *AprilTag system* and ultrasonic sensor. Given that the platform subjected to ocean waves, its floating dynamics is simulated assuming the ocean surface modeled with the Pierson-Moskowitz spectrum [5]. We tested our landing procedure in simulated environment as well as in real world environment. It is important to emphasize that the whole system architecture is the same for both environments.

A. Simulation Environment and Results

We used Gazebo with PX4 software-in-the-loop to validate our architecture in simulation. We updated the simulation environment adding the landing platform with its tags. Moreover, a camera object is attached to the quadrotor model to emulate camera functionality and to run the vision system in simulation. Figure 8 summarises the main snapshots of the entire landing procedure with also the field of view of the on-board camera. Figure 8(a) shows the beginning of the landing procedure; the quadrotor is in the *Initialization* state and it looks at the platform recognizing at least one tag. Subsequently, the quadrotor jumps into the *Holding* state and it starts moving towards the platform using the vision feedback according to section II-D1. In the meanwhile, it approaches vertically the platform 8(b). When the switching condition becomes true, the quadrotor moves to the *Compensation* state. At this phase, the quadrotor performs the vertical platform compensation, section II-D2, using the data coming from a fake ultrasonic sensor whilst it tracks horizontally the platform 8(c).



(a) (b) (c) (d)

Fig. 8: Landing procedure: (a) tracking, (b) approaching, (c) wave compensation, (d) landing.

In the end, Figure 8(d) shows the quadrotor landed on the platform.

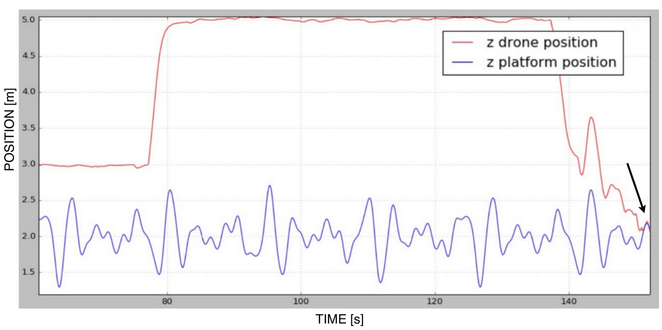


Fig. 9: Simulation results: z position of the quadrotor (red line) converging to the moving platform (blue line).

Therefore, Figure 9 shows the vertical approach of the quadrotor during the landing. The unpredicted vertical platform oscillation is represented by blue line while the height of the quadrotor is shown in red line. It is possible to see the quadrotor approaches vertically the platform compensating its vertical movement; all the while maintaining the horizontal tracking. In the end, it slowly lands. The landing point is underlined by the black arrow.

B. Real Environment

1) Landing platform: In the real environment, we introduce a mobile robot which is used to simulate ocean waves. It is a tracked ground vehicle with a 6 degree of freedom manipulator arm mounted on top of it, see Figure 10. On the second link, we place our designed platform. Figure 10 shows the ground vehicle and it is possible to see the designed platform and the quadrotor used for the experiments on top of it. In addition, Figure 11 emphasizes the manipulator arm employed for the waves simulation.

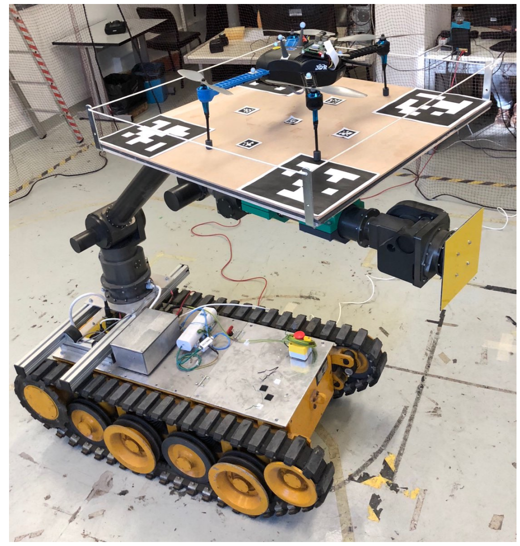


Fig. 10: Ground vehicle with its robotic arm.

Fig. 11: Front view of the mobile robot.

The manipulator arm moves autonomously according to the Pierson-Moskowitz spectrum. In real world experiments, the following parameters are considered:

- P-M spectrum frequencies are in the range from $[0,4$ rad/s - $1,3$ $rad/s]$.
- Amplitudes are calculated from the P-M spectrum assuming a 7 m/s wind.
- The wave motion has a maximum amplitude (calculated from the lowest point to the highest) of $0,5$ meters.
- The dimension of the landing pad is 70×70 cm.

It is important to underline that we add four more tags, w.r.t. the platform in simulation environment, to increase the robustness and the accuracy during the landing procedure

2) Quadrotor: For the testing phase in real environment, we employed IRIS quadrotor by 3DR [4]. We customized it for testing our architecture. It is equipped with a Pixhawk board as flight controller running the PX4 autopilot. Besides, an on-board computer, Raspberry Pi 3 [5] with Ubuntu MATE as operating system has been mounted on. The on-board computer allows to run software modules and it enables the communication with the autopilot board, see Figure 6. The employed quadrotor is shown in Figure 12.

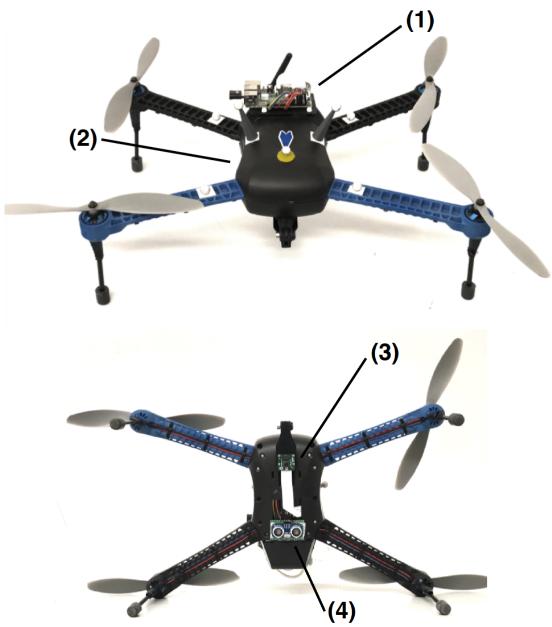


Fig. 12: Customization of IRIS quadrotor: (1) Raspberry Pi used as on-board computer. (2) PixHawk board located inside IRIS body. (3) Downward-looking Raspberry Pi camera used to detect the landing platform. (4) The HC-SR04 ultrasonic sensor used to compute relative vertical position and velocity of the quadrotor w.r.t. the platform.

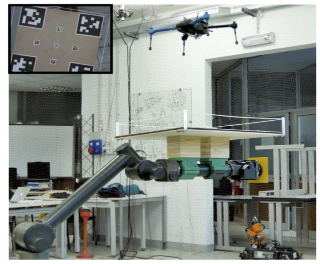
The quadrotor is also equipped with a Rasberry Pi camera module to acquire images with resolution of 640×480. The camera is looking towards the ground to detect the platform. Furthermore, we mounted HC-SR04 ultrasonic sensor, looking downward, to estimate the relative vertical distance and velocity between the platform and the quadrotor. Both the camera module and the ultrasonic sensor are connected to the Raspberry Pi board. The communication between the software modules runs using ROS.

3) Results: Figure 13 shows the camera view of the drone and the main snapshots of the landing procedure in indoor environment. It is important to say that we replicate the same scenario and the same tests performed in simulation environment. The vision system is used to perform the platform tracking. In the *Compensation* state, ultrasonic sensor is used to obtain relative vertical distance and to estimate relative velocity between the platform and quadrotor.

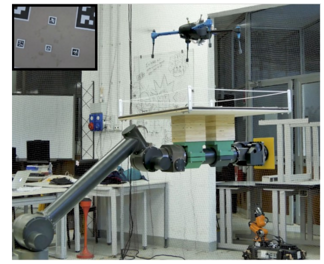
[4]Please refer to: https://3dr.com

[5]https://www.raspberrypi.org

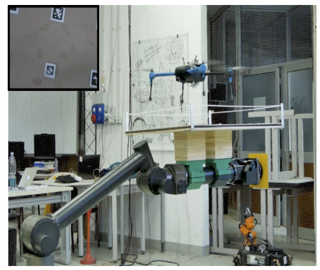
In Fig. 13(a), the drone is in the *Initialization* state,



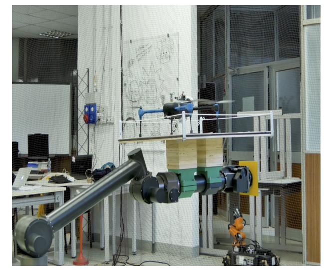
(a)



(b)



(c)

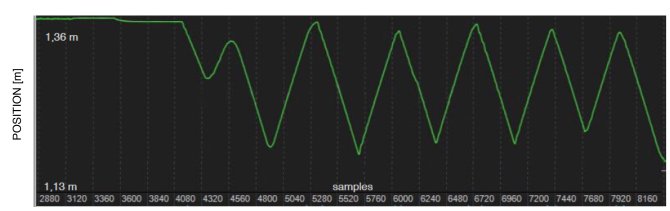


(d)

Fig. 13: Landing procedure: (a) tracking, (b) approaching, (c) wave compensation, (d) landing.

the starting point of the landing procedure. The platform is visible from the on-board vision system and hence the quadrotor moves into the *Holding* state; it performs the horizontal tracking of the platform, shown in Fig. 13(b). Subsequently, the Figure 13(c) shows the quadrotor in the *Compensation* state. At this stage, ultrasonic sensor is used to provide relative vertical distance and velocity between the platform and the quadrotor. The final snapshot, Figure 13(d), shows the quadrotor landed on the platform.

(a) Z trajectory of the quadrotor



(b) Z trajectory of the platform

Fig. 14: Indoor results: trajectory tracking of quadrotor and platform during landing procedure, yellow circle shows the landing point.

In addition, Figures 14 show the drone and platform trajectories respectively during platform approach with sampling rate at 120 Hz. It is possible to see the *Descending* state starting from the sample 3600 up to 5280. Then there is an instant in which the drone stayes in the *Holding* state cancelling out the horizontal error. Subsequently, it jumps in the *Compensation* state which ends with the landing, underlined by the yellow circle. It is important to underline that Figure 14(b) represents the vertical movement of the platform and Figure 14(a) represents the vertical movement of the quadrotor.

IV. Conclusion and Future Work

With this paper, we presented a quadrotor capable of landing autonomously on a moving platform by employing on-board computing and sensing. The sensors involved were monocular camera and ultrasonic sensor. The full pose of the platform was estimated using vision system. Ultrasonic sensor was used to compute the relative vertical position and velocity of the platform w.r.t. the quadrotor. Moreover, the landing procedure is presented and a complete description of the system architecture is provided. Section III underlined that our quadrotor is able to land also during harsh condition. The quadrotor compensates the unpredicted dynamics of the platform caused by the sea surface.

Fig. 15: Waterproof quadrotor.

We are currently working on performing the same tests outdoor using GPS for the quadrotor state estimation. In addition, we are transferring the hardware and software architecture on a waterproof quadrotor to test the landing procedure on a moving platform directly on the catamaran during maritime missions. The employed waterproof quadrotor is based on a shell designed by SwellPro [6] and it is shown in Fig. 15.

References

[1] W. Kong, D. Zhou, D. Zhang and J. Zhang, “Vision-based autonomous landing system for unmanned aerial vehicle: A survey” 2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI), Beijing, 2014, pp. 1-8.

[2] E. Simetti, S. Torelli and A. Sperindé, “Development of Modular USVs for Coastal Zone Monitoring” Sea Technology, May 2016.

[3] A. Nisticó, M. Baglietto, E. Simetti, G. Casalino and A. Sperindé, “Marea project: UAV landing procedure on a moving and floating platform” OCEANS 2017 - Anchorage, Anchorage, AK, 2017, pp. 1-10.

[4] E. Olson, “AprilTag: A robust and flexible visual fiducial system” 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 3400-3407.

[5] Z. Yinggu, F. Guoliang and Y. Jianqiang, “Controller design based on T-S fuzzy reasoning and ADRC for a flying boat” 2013 10th IEEE International Conference on Control and Automation (ICCA), Hangzhou, 2013, pp. 1578-1583.

[6] Please refer to: https://www.swellpro.com

[6] A.S. Huang, E. Olson and D. C. Moore, “Lcm: Lightweight communications and marshalling” 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, 2010, pp. 4057-4062.

[7] A. Gautam, P. B. Sujit and S. Saripalli, “A survey of autonomous landing techniques for UAVs” 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, 2014, pp. 1210-1218.

[8] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico and D. Scaramuzza, “Vision-based autonomous quadrotor landing on a moving platform” 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, 2017, pp. 200-207.

[9] X. Yang, L. Mejias and Matt Garratt, “Multi-sensor data fusion for UAV navigation during landing operations” 2011 Australian Conference on Robotics and Automation (ACRA), Melbourne, 2011, pp. 1-10.

[10] M. Verbandt, B. Theys and J. De Schutter, “Robust marker-tracking system for vision-based autonomous landing of VTOL UAVs” 2014 International Micro Air Vehicle Conference and Competition (IMAV), Delft, 2014.

[11] O. Araar, N. Aouf and I. Vitanov, “Vision Based Autonomous Landing of Multirotor UAV on Moving Platform” 2017 Journal of Intelligent and Robotic Systems, 85, pp. 369-384, 2017.

[12] G. Casalino B. Allotta, G. Antonelli, A. Caiti, G. Conte, G. Indiveri, C. Melchiorri, and E. Simetti, “ISME research trends: Marine robotics for emergencies at sea,” 2016 OCEANS, Shanghai, 2016, pp. 1-5.

[13] N. Abdelkrim, N. Aouf, A. Tsourdos and B. White, “Robust nonlinear filtering for INS/GPS UAV localization” 2008 16th Mediterranean Conference on Control and Automation, Ajaccio, 2008, pp. 695-702.

[14] E. Olson, J.H. Strom, R.D. Morton, A. Richardson, P. Ranganathan, R. Goeddel, M. Bulic, J. Crossman, and R. Marinier, “Progress toward multi-robot reconnaissance and the MAGIC 2010 competition” 2012 Journal of Field Robotics, 29, pp. 762-792, 2012.

[15] D. Mersch, A. Crespi, and L. Keller, “Tracking Individuals Shows Spatial Fidelity Is a Key Regulator of Ant Social Organization” 2013 Science New York, 340. 10.1126/science.

[16] A. C. Sanderson, and L. E. Weiss. “Image-based visual servo control using relational graph error signals. 1980 International Conference on Cybernetics and Society.