Project 12: Autonomous Quadrotor Landing on a Moving Platform

Phase 1: Image Processing and Computer Vision
Detection of objects based on both color and shape in an image.
Real time object tracking based on both color and shape.

Mid Term Presentation
Topic
Rodriguez-Ramos, Alejandro, et al. "Towards fully autonomous landing on moving platforms for rotary unmanned aerial vehicles." *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2017.

Phase 2: Control for Robotic platforms
Task 1
Flying AR Parrot Drone using AR drone autonomy ROS.

Final Report
Title: Drone Simulation and Control
Helping material is available in the following link:
https://www.mathworks.com/videos/series/drone-simulation-and-control.html

2017 International Conference on
Unmanned Aircraft Systems (ICUAS)
June 13-16, 2017, Miami, FL, USA

Towards Fully Autonomous Landing on Moving Platforms for Rotary Unmanned Aerial Vehicles

Alejandro Rodriguez-Ramos, Carlos Sampedro,
Hriday Bavle, Zorana Milosevic, Alejandro Garcia-Vaquero and Pascual Campoy

Abstract— **Fully autonomous landing on moving platforms poses a problem of importance for Unmanned Aerial Vehicles (UAVs). Current approaches are usually based on tracking and following the moving platform by means of several techniques, which frequently lack performance in real applications.**

The aim of this paper is to prove a simple landing strategy is able to provide practical results. The presented approach is based on three stages: estimation, prediction and fast landing. As a preliminary phase, the problem is solved for a particular case of the IMAV 2016 competition. Subsequently, it is extended to a more generic and versatile approach. A thorough evaluation has been conducted with simulated and real flight experiments. Simulations have been performed utilizing Gazebo 6 and PX4 Software-In-The-Loop (SITL) and real flight experiments have been conducted with a custom quadrotor and a moving platform in an indoor environment.

I. INTRODUCTION

In recent years, considerable research has been conducted regarding the design, development, and operation of autonomous Unmanned Aerial Vehicles (UAV). In particular, multi-rotor UAVs can perform low-speed tracking and operate under situations where runaways are not available for taking-off and landing. A problem of importance for autonomous UAVs is the design of autopilots for landing on moving platforms which follow an arbitrary trajectory, such as ground transportation vehicles or ships.

The control problem for landing of autonomous UAVs is challenging since the vehicle dynamics are highly nonlinear and coupled with the unknown motion of the landing platform. Furthermore, multi-rotor UAVs aggressive control for autonomous landing requires high-frequency and reliable sensors, which are not always available except in certain experimental conditions *e.g.* motion capture systems.

The control of multi-rotor UAVs for autonomous landing on a moving platform has been addressed in several lines of research: Position-Based Visual Servoing (PBVS) [1][2], Image-Based Visual Servoing (IBVS) [3][4], tethered solutions [5][6][7][8], optic flow methods [9][10], high-speed landing with communication between moving platform and the UAV [11], bio-inspired [12] or machine learning based techniques [13][14]. (i) Position-Based Visual Servoing extracts the position of the target, relative to the UAV and tracks the target in movement. Active sensors have been used, such as Infrared-LEDs (IR-LEDs) on the target and an IR camera, combined with 4-Proportional-Integral-Derivative (PID) controllers [1]. Also, the target can be tracked utilizing computer vision algorithms and the position can be subsequently estimated with Perspective-n-Point (PnP) algorithms, assuming a known shape of the target [2]. (ii) Image-Based Visual Servoing has been tested, with motion capture systems [3], as well as more sophisticated approaches such as Dynamic IBVS with disturbance estimation, using ground truth data in simulation [4].

(iii) Other physically invasive approaches have been investigated, such as tethering a helicopter in order to deal with rough environments and estimating wind disturbances [5] or a tethered Micro Aerial Vehicle (MAV) with simpler 2DOF control [6]. (iv) Optical flow has also been applied, assuming that the normal of the platform is known and constant over time [9]. (v) A more practical technique uses a low cost approach to provide the moving platform with sensors in order to precisely estimate its state [11].

(vi) Other innovative approaches have also been designed and tested, such as trajectory calculation and optimal trajectory control, simulating the target position estimation and with a rectilinear target trajectory [15]. In addition, bio-inspired landing and decking approaches have proven to perform well in the real world, based on the time-to-contact (TTC) indicator but tested in static platforms and with some improvements in perception as future work [12]. Non-classical approaches have also been tested, such as the use of artificial neural networks to detect a suitable landing zone [13]. Other learning techniques, such as reinforcement learning, have proven to have a high potential in the autonomous landing field, but were only tested in simulation and with a static platform [14].

Prior studies have noted the importance of robust approaches against different conditions and the still considerable gap between performed experiments and real world applications. The main challenges to address in autonomous landing are low-frequency and non-reliable sensors and algorithms for perception, as well as on-research control techniques for nonlinear and slow-dynamics outdoor UAVs. Vision algorithms are useful, but closing the control loop with the camera can lead to poor performance in real fast-moving platforms. Furhermore, camera sensors which have lightweight and low price, aid the estimation of the relative pose between a UAV and a landing platform [16], [4] and have become a widespread sensing modality. How-

The authors would like to thank the Spanish Ministry of Science MICYT DPI2014-60139-R for project funding. The LAL UPM and the MONCLOA Campus of International Excellence are also acknowledged for the funding of the predoctoral contract of one of the authors.

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

978-1-5090-4494-8/17/$31.00 ©2017 IEEE

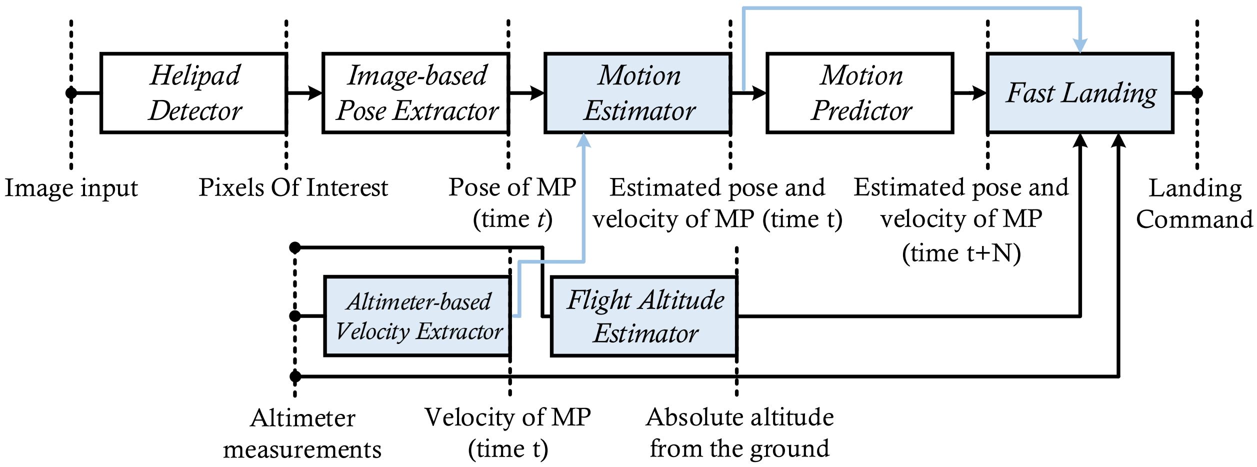

Helipad Detector
Image-based Pose Extractor
Motion Estimator
Motion Predictor
Fast Landing
Image input
Pixels Of Interest
Pose of MP (time *t*)
Estimated pose and velocity of MP (time t)
Estimated pose and velocity of MP (time t+N)
Landing Command
Altimeter-based Velocity Extractor
Flight Altitude Estimator
Altimeter measurements
Velocity of MP (time t)
Absolute altitude from the ground

Fig. 1: All the components involved in our approach: estimation (Helipad Detector, Image-based Pose Extractor, Altimeter-based Velocity Extractor, Flight Altitude Estimator and Motion Estimator), prediction (Motion Predictor) and fast landing (Fast Landing). The preliminary components developed for the IMAV 2016 competition are highlighted in light blue.

ever, developing an on-board, robust, high-precision moving platform detector using low-resolution cameras with limited computation resources is a highly challenging problem.

In addition to stated limitations, landing maneuvers are generally performed at the end of the mission (*e.g.* charging batteries [17] or swapping batteries [18][19]), with the only exception of UAVs intermediate ground operations such as logistical delivery [20]. The requirement to perform landing maneuvers at the end of a mission invokes hard power constrains and time limitation. As a consequence, the landing task has to be solved fast and accurately, avoiding possible repeated attempts which could lead to the complete exhaustion of the battery. One popular approach, which is included in the previous discussion, is following the platform (for instance, using PBVS or IBVS [1][2][3]). It requires additional computational power, which is a scarce resource in limited-sized UAVs, and can lead to poor performance in fast moving platforms. In general, the control of the highly nonlinear dynamics of a UAV, as well as the time and power consumption constraints, pose a challenging task with no straightforward solution.

For the purpose of dealing with these issues, the problem can be approached differently. Instead of following the moving platform as fast as possible, our approach points to predict its trajectory, in order to anticipate a future position, and land without the necessity of aggressive or complex maneuvers. In this paper, we propose the initial stage of a strategy based on estimation, prediction and fast landing. The prediction stage has been simplified but without losing generality, and providing a basic framework to start from. All estimation, prediction and fast landing techniques have been designed, tested and validated in simulation. Furthermore, a preliminary version of this technique has been validated in real experiments and in the international MAV competition (IMAV 2016[1]), demonstrating the capabilities of the approach to solve autonomous landing in a variety of scenarios.

The remainder of this paper is organized as follows: Section II presents the proposed approach. Section III shows the conducted experiments and their results. Finally, Section IV summarizes conclusions and future work.

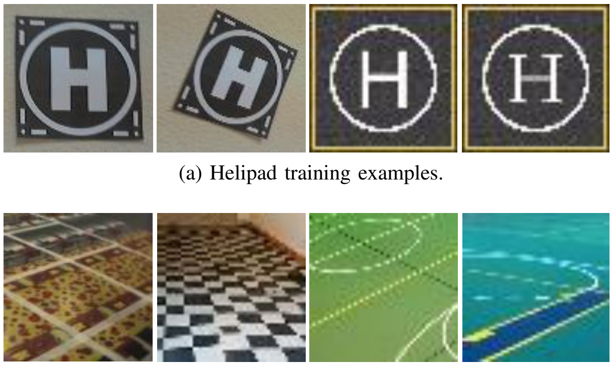
II. PROPOSED APPROACH

This study aims to provide practical results for autonomous landing on a Moving Platform (MP). The proposed solution consists of three stages: estimation, prediction and fast landing. In the first estimation stage, the multi-rotor is awaiting at a high altitude (hovering) and forcing the MP to be within its field of view, in order to estimate the motion of the MP (pose and velocity). In the prediction stage, the UAV is able to predict the motion of the MP in a predefined range of space. The prediction range will depend on the prediction algorithm capabilities, the maximum velocity of the multi-rotor and the randomness of the current trajectory. The aim is to have a high confidence about a certain position and velocity of the MP in a future instant of time.

After the estimation and prediction stages have converged, the multi-rotor proceeds to the fast landing stage. The multi-rotor moves to the predicted position of the MP, at a safe altitude to land. In that stage, utilizing the information of velocity provided (among others), a fast autonomous landing can be carried out.

Since the complete problem is complex, we have split the solution into several components in order to tackle it separately. There are seven components: the Helipad Detector, the Image-based Pose Extractor, the Altimeter-based Velocity Extractor, the Motion Estimator, the Flight Altitude Estimator, the Motion Predictor and the Fast Landing. In Fig.

[1] http://www.imavs.org/imav2016



(a) Helipad training examples.

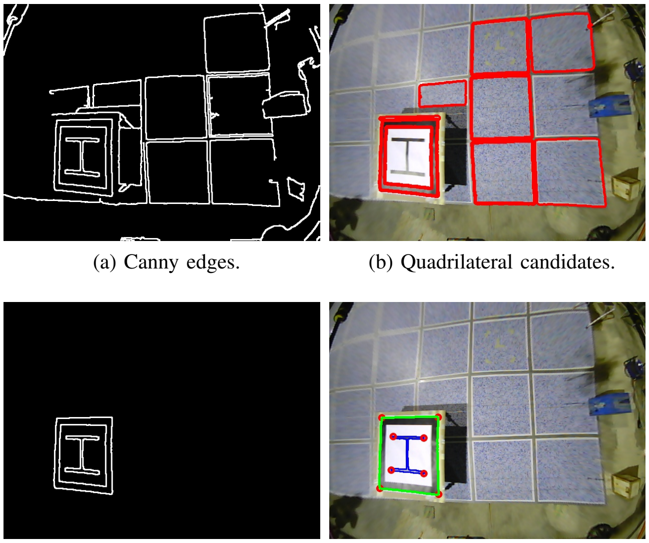
(b) Background training examples.

Fig. 2: Examples of cropped images of: (a) Four helipad images. The first two images are captured during a real flight, while the last two images are artificially generated. (b) Four background images. These and similar images are used for training and evaluating the supervised classifier.

1, all the stated components and their interconnections are depicted. It has to be highlighted that this work is focused on the MP detection, UAV and MP motion estimation, MP motion prediction and fast landing. Remaining components, such as control techniques, are based on a previous work, which is properly described in [21]. Furthermore, in IMAV 2016 competition, the MP was designed to follow a periodic trajectory. In this context, the Helipad Detector, the Image-based Pose Extractor and the Motion Predictor were not required. In this study, IMAV 2016 components have been extended in order to provide generality to the approach, making it suitable for landing on a variety of MP trajectories and under some constraints which are further described in the following subsections. Our approach realizes some limitations, but it is capable of providing practical results in real experiments and its modularity makes it extendible to more complex strategies. In the following subsections, all the components are explained in detail.

A. Helipad Detection

For a given trajectory of a MP, as long as the platform is within the camera sensor field-of-view, its known marker (in this case, the helipad) can be detected. In this paper, the proposed strategy for helipad detection is based on the combination of four main modules: a Region Proposal algorithm, based on the detection of quadrilateral objects within the image, a Feature Extraction module based on Histogram of Oriented Gradients (HOG) [22], a Supervised Learning Classifier for Helipad/Background classification, and a Post-processing module for detection refinement. In the next paragraphs a detailed explanation of the four modules is provided:



(a) Canny edges.

(b) Quadrilateral candidates.

(c) Positive detection contours.

(d) Final Detection result after post-processing.

Fig. 3: Helipad Detection results. (a) Canny edges extracted. (b) Quadrilateral candidates generated by the Region Proposal module (red color). (c) Contours of the positive detections after applying the supervised learning classifier. (d) Final Detection result after post-processing stage. In green color it is depicted the outer contour of the detected helipad, while in blue color it is shown the contour of the “H”.

- Region Proposal: This module is in charge of computing the most probable candidates within the image for being Helipad. The first step of the proposed candidate generator consists of an algorithm based on Canny edge detector [23]. Once the main edges in the image have been extracted (see Fig. 3a), the next step consists of applying a set of heuristics for quadrilateral object extraction (*i.e.* quadrilateral contours should have four vertices after contour approximation to their perimeter, cosine of the angles between joint edges are small, etc). All these possible quadrilateral objects (see Fig. 3b) conform the set of Regions Of Interest (ROIs) to be input to the classifier.
- Feature Extraction: In each of the proposed ROIs obtained from previous module, a descriptor vector based on HOG features is extracted. The configuration adopted in this paper for the HOG feature descriptor is described in detail:
  - Window size: 48×48 pixels.
  - Block Size: 16×16 pixels.
  - Block Stride: 8 pixels.
  - Cell Size: 8×8 pixels.
  - Histogram bins: 9 bins of 20 degrees each.
    Based on the previous configuration, the resulting HOG descriptor vector consists of a 900×1 vector, which is used as input to the supervised classifier.
- Supervised Learning Classifier: For the purpose of classifying each of the proposed ROI of the image as belonging to Helipad or Background class, a supervised learning classifier based on L2 Regularized Logistic Regression is applied, which is modeled by (1).

$$J(\omega) = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l} log(1 + e^{-y_i\omega^T x_i}) \qquad (1)$$

where ω are the parameters to be learned by the classifier. C is the regularization parameter, and (x_i, y_i) is the instance-label pair of the i_{th} training sample.

A total of 18475 images (9233 from Helipad class and 9242 from Background class) have been utilized for training and evaluating the proposed supervised classifier. Some image examples for training the classifier are depicted in Fig. 2a and 2b.

For selecting the optimal parameter for the L2 Regularized Logistic Regression, a 5-fold cross-validation over a range of selected values of the Regularization parameter (C, see (1)) has been conducted, obtaining an optimal value of $C_{opt} = 8$.

After all the hiperparameters of the supervised learning classifier have been selected, a training and evaluation stages have been performed in order to select the definitive model of the classifier. As shown in Tables I and II, the generalization error of the classifier is very low, pointing to good generalization features.

TABLE I: Confusion matrix obtained for the helipad detection in the training set

Actual class Predicted class	Helipad	Background
Helipad	7373	4
Background	14	7390

TABLE II: Confusion matrix obtained for the helipad detection in the test set

Actual class Predicted class	Helipad	Background
Helipad	1834	12
Background	12	1836

- Post-processing: This module is the responsible of the detection refinement applied in the ROIs considered as Helipad by the supervised classifier. The purpose of the post-processing algorithm is twofold. First, filter any false positive that can appear in the image, and second to extract corners in the object of interest (i.e. Helipad) for a posterior stage in which the relative pose of the object is required to be computed (i.e. Perspective-n-Point algorithms).

  For the purpose of filtering false positives and extracting the corners of the interior "H" contour of the helipad, a filter based on Hu moments [24] has been utilized. The proposed filter takes as input an array of contours in the image, and computes as the definitive candidate the one that has the minimum ratio between first and third and first and fourth hu moments. In Fig. 3d, the precise detection of the outer and inner corners of the Helipad can be appreciated, what will lead to a more accurate and robust pose estimation. In addition, in Fig. 3d it can be noticed the successful filtering result obtained after the post-processing stage for retrieving the contour of the "H".

B. Image-based Pose Extractor

The helipad detector, which was explained in Section II-A, uses each image frame to extract features of the helipad in pixel coordinates. With the 2D feature points of the helipad and their correspondences in the 3D space, the pose of the helipad with respect to the camera can be computed (utilizing a calibrated *pinhole* camera model). The 2D correspondences are located on the surface of the helipad (see Fig. 3d). In previous studies [25], we have benchmarked a set of state-of-art Perspective-n-Point (PnP) algorithms in order to determine the most suitable method for the case of 3D points gathered in the same plane (planar case). Based on this work and for the sake of implementation simplicity, we have selected the non-iterative *Robust* PnP (RPnP) [26] to compute the pose of the helipad with respect to the camera (C) (refer to Fig. 5). With the transformation between every frame of reference (see 2), this pose is latterly transformed from camera coordinates (C) to world coordinates (W), resulting in the pose of the MP in world coordinates $[x^W_{MP}, y^W_{MP}, z^W_{MP}]$.

$$T^W_{MP} = T^W_U \cdot T^U_C \cdot T^C_{MP} \qquad (2)$$

with T^W_{MP}, T^W_U, T^U_C, T^C_{MP} the transformation from a certain frame of reference (lower index) to another specified (upper index). In Fig. 5, all the frames of reference involved are best depicted.

C. Altimeter-based Velocity Extractor

In our proposed solution, the detection of the MP is also addressed by the usage of a laser altimeter. As depicted in Fig. 4, with a 1D-altimeter measurements buffer, the noise can be filtered and the period of time the MP is below the UAV, can be registered. The velocity of the MP v^W_{MP}, for periodic trajectories, is simply obtained with the platform size and the registered time. It can be better estimated when the platform completes more periods of its trajectory. It is assumed that the UAV velocity in x and y axes is negligible in the period of extraction (in world coordinates).

D. Motion Estimator

With the aim of estimating the motion of the MP (pose and velocity), we have integrated a Linear Kalman Filter (KF). With a linear process model, the velocity of the MP can be accurately estimated. The pose and velocity of the MP (in world coordinates) are described in the linear state space $x(k) = \left[x^W_{MP}, y^W_{MP}, z^W_{MP}, \dot{x}^W_{MP}, \dot{y}^W_{MP}, \dot{z}^W_{MP}\right]$ and state transition matrix F in (3).

$$F = \begin{bmatrix} 1 & 0 & 0 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

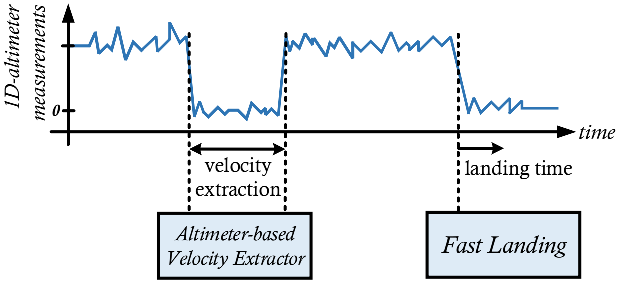


Fig. 4: Graphical representation of the 1D-altimeter measurements. The Altimeter-based Velocity Estimator is able to extract the velocity of the moving platform in each cycle of a periodic trajectory. The Fast Landing component uses the 1D-altimeter falling edge to trigger the fast landing stage.

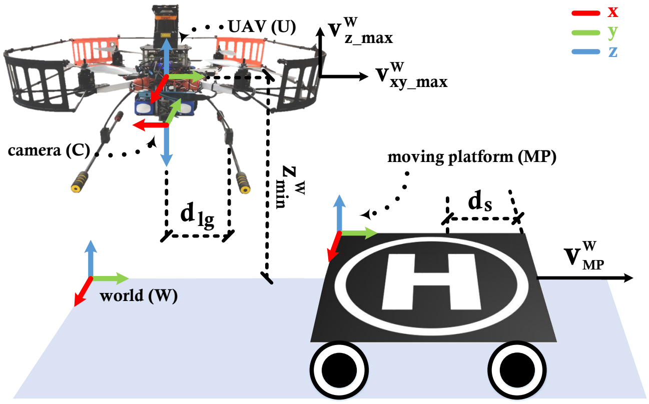


Fig. 5: Graphical representation of the UAV and the MP. The frames of reference are shown. All the variables involved in the Motion Predictor and the Fast Landing stages are included. v_{xy_max} is the maximum velocity of the UAV in the direction of the trajectory of the MP (in world coordinates). v_{z_max} is the maximum velocity in the $z - axis$ (in world coordinates). z^W_{min} is the minimum altitude for estimation (in world coordinates). d_{lg} is the horizontal distance from the 1D-altimeter sensor to one extreme of the landing gear. d_s is the minimum horizontal distance to consider the UAV inside the surface of the MP. $v^W_{MP} = ||[\dot{x}^W_{MP}, \dot{y}^W_{MP}]||$ is the velocity of the MP (in world coordinates).

With δt as the time between predictions. The observation or measurement model H can take the pose and/or the velocity of the MP into consideration. In the more generic case, only the pose from the Image-based Pose Extractor is included, since the Altimeter-based Velocity Extractor is meant only for periodic trajectories (see (4)).

$$H = \begin{bmatrix}1 & 1 & 1 & 0 & 0 & 0\end{bmatrix} \qquad (4)$$

It has to be noted that the velocity of the UAV is not considered in the measurement model. This is due to the fact that the UAV is approximated to have a zero velocity while estimating the motion of the MP, simplifying the conversion from the UAV (U) to the world (W) frame of coordinates. This is an approximation, but still completely valid in this application (as shown in Section III).

E. Flight Altitude Estimator

The 3D localization approach for the UAV is achieved by fusing the information of three components. The first component consists of a 2D SLAM module providing the 2D estimated pose of the UAV and the yaw [27]. The second component provides orientation (roll and pitch) [27] and the third component consists of a flight altitude estimator which is able to provide the estimated altitude of the UAV in presence of several objects on the ground below it [28].

For every stage in our approach (estimation, prediction and fast landing) the UAV has to maintain the desired flight altitude in presence of the MP. Conventional techniques using a single 1D-altimeter sensor for estimating the altitude of the UAV would fail to maintain the desired flight altitude, since the MP can alter the measurements provided by the altimeter sensor. A multi-sensor fusion technique was developed to overcome these issues. An Extended Kalman Filter (EKF) based state estimator was used for estimating the flight altitude of the UAV as well as the elevation of the ground below it.

The EKF-based state estimator consists of a kinematic process model without any robot inputs and a process noise. The state estimator fuses the measurements from an 1D-altimeter sensor (which in this case is a 1D-laser altimeter), an Inertial Measurement Unit (IMU, consisting of accelerometer and gyroscope) and a barometer. Each sensor contributes differently to the estimation (as inferred from the covariance matrix) and the bias of each sensor is included in the state. The whole description of the Flight Altitude Estimator can be found in [28].

The outputs of the Flight Altitude State Estimator component are, vertical linear position, velocity and acceleration, and the attitude and angular velocity of the UAV with respect to the world frame of reference. The elevation of the ground (ground object height) in also specified in world frame of reference. The state estimator incorporates the feature of auto calibrating the accelerometer and barometer sensors, in order to estimate their bias and to provide a more accurate estimation of the altitude.

F. Motion Predictor

The prediction stage can be a functionality of importance for complex trajectories of the MP. Nevertheless, for a wide variety of scenarios, where the trajectory can be approximated to linear in a short period of time, a constant-velocity model for the MP can be sufficient. This is only valid for short periods of time, where the trajectory can be considered as linear and the velocity can be approximated to constant. A more complex trajectory with sudden changes in the velocity and/or the direction is not within the scope of this technique. In any manner, the Motion Predictor can be improved in order to include a more complex trajectory model in the estimator *e.g.* adjusting the process model of a EKF. In this work, the Motion Predictor has been simplified, without losing generality, approximating the model of the MP as a constant-velocity model.

The prediction range will depend on the prediction algorithm capabilities, the maximum velocity of the multi-rotor in every axis of space and the randomness of the current trajectory. For a constant-velocity model, the minimum predicted distance d^U_{min} (in the direction of the MP trajectory and in UAV coordinates) to safely execute the Fast Landing stage, follows (5) and (6).

$$t_{safe} = \frac{z^W_{min} - z^{MP}_{safe}}{v^W_{z\_max}} \tag{5}$$

$$d^U_{min} = \frac{v^W_{xy\_max} - v^W_{MP}}{t_{safe}} \tag{6}$$

where t_{safe} is the minimum time required to reach a safe altitude to land on the MP. The minimum altitude to correctly estimate the motion of the MP, z^W_{min}, depends on the camera field-of-view, the Motion Estimator convergence time and the real velocity of the MP. The maximum safe altitude to land z^{MP}_{safe} depends on the altitude of the landing surface of the MP. The maximum velocity in z-axis (in world coordinates) is represented by $v^W_{z\_max}$ and the maximum velocity on the direction of the trajectory of the MP is represented by $v^W_{xy\_max}$. v^W_{MP} represents the estimated velocity of the MP in world coordinates (refer to Fig. 5). $v^W_{xy\_max}$ and v^W_{MP} are assumed to have the same direction in the period of prediction.

G. Fast Landing

In the Fast Landing stage, the landing time t_{land} (see Fig. 4), which is the minimum time required to safely land inside the surface of the MP, can be calculated based on the velocity of the MP v^W_{MP} and the falling time t_{fall} of the UAV, among others. The landing time is being accounted from the instant the platform is first detected, as depicted in Fig. 4, and can be computed following (7). In these calculations, it is assumed that the landing gear of the UAV is oriented in parallel to one edge of the MP at the moment to land.

$$t_{land} = \frac{d_s - d_{lg}}{v^W_{MP}} - t_{fall} \tag{7}$$

where d_s and d_{lg} are the safety distance and the distance from the sensor with respect to one extreme of the landing gear of the UAV, respectively. The safety distance d_s is the minimum distance from which the landing gear is considered to be able to land completely on the MP. d_{lg} represents the distance (in a plane, parallel to the surface of the MP) from the 1D-altimeter to one extreme of the landing gear (refer to Fig. 5). The falling time t_{fall} can be easily approximated from a simple experiment where the UAV lands on a static platform at a certain altitude. The platform velocity v^W_{MP} can be estimated by means of several types of sensors. In our architecture, as previously stated, both the Image-based Pose Extractor or the Altimeter-based Velocity Extractor can lead to a velocity estimation of the MP (see Fig. 1)

A key strength of the present implementation is its modularity and versatility, being able to switch each stage of the process to a different approach, as well as fusing some of them in less but more complex techniques, as explained in Section IV.

III. EXPERIMENTS AND RESULTS

In this section, the experiments that have been conducted are described in detail. The experimental setup is provided for both simulated and real flight experiments. Preliminary results of the IMAV 2016 competition are detailed. Simulation results for the complete approach are also provided. All the experiments can be visualized in `https://vimeo.com/205757662`, as well as our participation in the IMAV 2016 competition.

A. Experimental Setup

In this section, the experimental setup that has been utilized for simulated and real flight experiments is best described. All the software components have been developed using C++, under the standard C++ 11, using ROS (Robot Operating System) [29] as the communication framework between the whole set of components in the architecture. The operative system utilized for running the different processes in both simulated and real flights is Ubuntu 14.04 LTS.

1) Simulated Experiments: A simulated indoor environment has been generated using Gazebo 6[2]. The full simulation has been carried out through a Software-In-The-Loop (SITL) simulation running PX4 autopilot and Aerostack [21]. Aerostack is an open-source framework for autonomous UAV development, and its explanation is out of the scope of this work. More information is detailed in [21].

A model of a MP has been designed in Gazebo, capable of following any predefined trajectory. These simulations make simpler the transition to real-flight testing.

2) Real Experiments: A real indoor environment with a MP has been built. The MP can follow any trajectory and is manually commanded. Its maximum velocity is 0.5 m/s. A complete UAV platform has been designed for the IMAV 2016 competition, incorporating all the required sensors (270-degrees LIDAR at 40 Hz, 1D-laser altimeter at 32 Hz, bottom and front cameras at 30 Hz and PixHawk autopilot). An Intel NUC 6i5SYK has been chosen as the on-board computer.

B. Preliminary Results

With the aim of solving the IMAV 2016 competition, a basic set of components were designed, tested and validated, as seen in Fig. 1. The trajectory of the MP was periodic, as a predefined specification of the IMAV 2016 competition.

In the case of periodic trajectories, two experiments have been conducted, in both simulated and real environments. The experiment consists of a MP describing a linear-periodic trajectory, for both simulated and real flight scenarios. The

[2] http://gazebosim.org

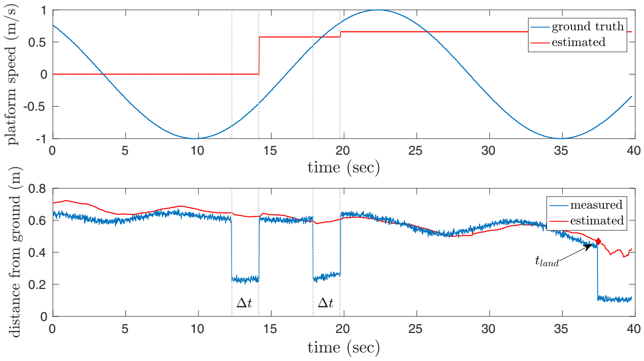


Fig. 6: Experiment conducted during the execution of a simulated flight with periodic trajectory of the MP. (Top) Ground truth and estimated velocity of the MP. The trajectory of the MP is linear and periodic in one axis. The velocity is approximated to a constant value in each Δt range and averaged between several estimations. The final value is $v_{MP}^{W} = 0.66$ m/s. (Bottom) Estimated absolute distance from the ground and laser altimeter measurements. The altitude estimation is carried out by an EKF (see Section II-E) in order to hold a constant altitude while flying. Two laps were required to land because the UAV altitude was exceeding the maximum safe landing altitude from the landing surface. $t_{land} = 58.37$ ms is the instant of time when landing is triggered.

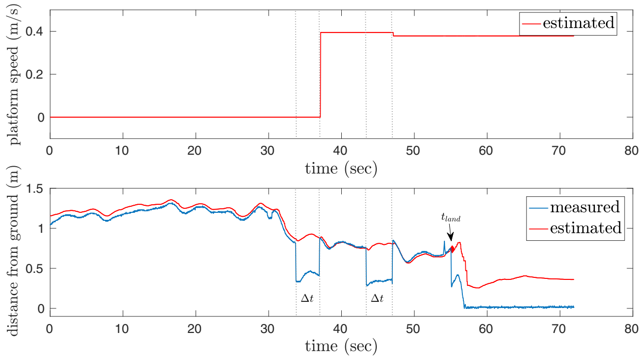


Fig. 7: Experiment conducted during the execution of a real flight with periodic trajectory of the MP. (Top) Estimated velocity of the MP. The trajectory of the MP is linear and periodic in one axis. The velocity is approximated to a constant value in each Δt range and averaged between several estimations. The final estimation is $v_{MP}^{W} = 0.3788$ m/s. (Bottom) Estimated absolute distance from the ground and 1D-altimeter measurements. The altitude estimation is carried out by an EKF (see Section II-E) in order to hold a constant altitude while flying. Two laps were required to land because the UAV altitude was exceeding the maximum safe landing altitude from the landing surface. $t_{land} = 153.68$ ms is the instant of time when landing is triggered.

UAV is assumed to know one position of the MP in the whole trajectory (since the UAV takes off and lands in the same position).

Fig. 6 illustrates the variation in velocity of the MP and its estimation by the UAV. The velocity is averaged in the period of estimation, which is the time the moving platform is underneath the UAV. The period of estimation is highlighted with two dotted lines and is represented by Δt. Both the 1D-altimeter measurements and the absolute estimated distance from the ground are also represented. As shown in Fig. 6, two laps of the MP are enough to estimate the velocity, which enables the Fast Landing stage. The third lap triggers the landing stage with the precomputed t_{land}, which is the minimum time to safely land, accounted from the first instant of detection of the platform.

The previous experiment was replicated in a real environment, as depicted in Fig. 7. In this scenario, the ground-truth velocity of the platform was not available. Again, two laps were required for a successful landing stage.

C. Results and Discussion

For our complete solution (which is composed of all the components presented in Fig. 1), the experiment has only been conducted in simulation. For the sake of simplicity, this experiment consists of a MP describing a non-periodic linear trajectory at a constant velocity. In a preliminary moment, the UAV estimates the velocity vector of the MP and predicts its future position based on a constant-velocity model. After estimation, the UAV hovers on top of the predicted position in order to trigger the fast landing stage. The performed experiment is enough to validate the stages of estimation, prediction and fast landing for non-periodic trajectories. As a remainder, for more complex or variable non-periodic trajectories, the only component to be enhanced is the prediction stage, as considered in Section IV.

Fig. 8 illustrates all estimation, prediction and fast landing stages. In the estimation stage, the UAV hovers at a higher altitude in order to force the MP to be within its field of view. This period of time is highlighted with two dotted lines in Fig. 8. The minimum altitude depends on the horizontal field of view of the bottom camera (set to 120.321° for this experiment), the speed of convergence of the estimation algorithm and the velocity of the MP. As shown in the results, the approximation of static UAV in the period of estimation is completely valid. After the estimation process has converged, the UAV predicts a future position in 3D space and a set point to the position controller is commanded. For more information about the position controller, please refer to [30]. As a final step, with the pre-estimated velocity, the fast landing stage is triggered when the target has reached the predicted position (down-pointing arrow in Fig. 8) and the landing is successfully carried out.

IV. CONCLUSIONS AND FUTURE WORK

This paper proposed a strategy for autonomous landing on a moving platform. Our approach was presented as a practical solution where landing is required, for an average linearly

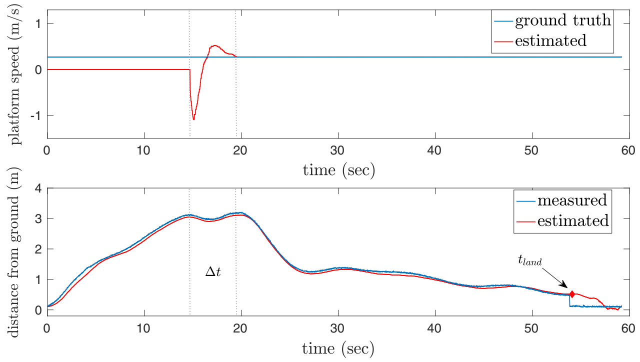


Fig. 8: Experiment conducted during the execution of a simulated flight with non-periodic trajectory of the MP. (Top) Ground truth and estimated velocity of the MP. The trajectory of the MP is linear in one axis, non-periodic and at a constant velocity of 0.27 m/s. The velocity is estimated at a rate of 100 Hz for $\Delta t = 4.76$ s seconds at $z^{W}_{min} \simeq$ 3 m height (until the algorithm converges). The converged value ($v^{W}_{MP} = 0.278$ m/s) is taken to enable the fast landing stage. (Bottom) Estimated absolute distance from the ground and 1D-altimeter measurements. The altitude estimation is carried out by an EKF (see Section II-E) in order to fly on top of irregular shapes and/or to hold a constant altitude while flying. After the estimation phase Δt, the UAV predicts a position at $d^{U}_{min} \simeq 8$ m in the future and flies to it ($v^{W}_{z\_max} \simeq$ 0.35 m/s, $v^{W}_{xy\_max} \simeq 1.2$ m/s and $z^{MP}_{min} \simeq 0.5$ m). Then, a safe altitude to land is achieved. $t_{land} = 269.75$ ms is the instant of time when landing is triggered.

controlled UAV. The solution was focused on a preliminary version meant for solving the IMAV 2016 competition. It was subsequently extended in order to make it more generic and versatile. We have proven our technique to work in simulation, with periodic and non-periodic trajectories, by means of a simulated environment in Gazebo 6 and PX4 Software-In-The-Loop (SITL). A real flight experiment with a custom quadrotor and a moving platform was also carried out in order to prove its practical feasibility with periodic trajectories. We have pointed out that both estimation and prediction stages can be crucial when interacting with objects in motion.

In the helipad detection part, immediate future work is lined towards the development of more powerful supervised learning modules, such as Convolutional Neural Networks (CNNs), which can greatly benefit not only the object recognition functionalities, but also the pose extraction stage. In the latter case, a thorough evaluation using ground truth training data will be conducted for assessing the feasibility of extracting the relative pose of the moving platform with respect to the UAV. Another key improvement in the prediction stage is the usage of a machine learning technique for prediction, in order to be able to predict more complicated trajectories of the moving platform (utilizing the historical information of the Motion Estimator as an input).

References

[1] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic take off, tracking and landing of a miniature uav on a moving carrier vehicle," *Journal of intelligent & robotic systems*, vol. 61, no. 1, pp. 221–238, 2011.

[2] Y. Bi and H. Duan, "Implementation of autonomous visual tracking and landing for a low-cost quadrotor," *Optik-International Journal for Light and Electron Optics*, vol. 124, no. 18, pp. 3296–3300, 2013.

[3] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a vtol uav on a moving platform using image-based visual servoing," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 971–976.

[4] P. Serra, R. Cunha, T. Hamel, D. Cabecinhas, and C. Silvestre, "Landing of a quadrotor on a moving target using dynamic image-based visual servo control," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1524–1535, 2016.

[5] F. Ruffier and N. Franceschini, "Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction," in *Robotics and Automation (ICRA), 2004 IEEE International Conference on*, vol. 3. IEEE, 2004, pp. 2339–2346.

[6] F. Ruffier and N. Franceschin, "Optic flow regulation in unsteady environments: a tethered mav achieves terrain following and targeted landing over a moving platform," *Journal of Intelligent & Robotic Systems*, vol. 79, no. 2, pp. 275–293, 2015.

[7] S.-R. Oh, K. Pathak, S. K. Agrawal, H. R. Pota, and M. Garrett, "Autonomous helicopter landing on a moving platform using a tether," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2006, pp. 3960–3965.

[8] S.-R. Oh, K. Pathak, S. K. Agrawal, H. R. Pota, and M. Garratt, "Approaches for a tether-guided landing of an autonomous helicopter," *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 536–544, 2006.

[9] B. Herissé, T. Hamel, R. Mahony, and F.-X. Russotto, "Landing a vtol unmanned aerial vehicle on a moving platform using optical flow," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 77–89, 2012.

[10] B. Herisse, T. Hamel, R. Mahony, and F.-X. Russotto, "The landing problem of a vtol unmanned aerial vehicle on a moving platform using optical flow," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1600–1605.

[11] A. Borowczyk, D.-T. Nguyen, A. P.-V. Nguyen, D. Q. Nguyen, D. Saussié, and J. L. Ny, "Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle," *arXiv preprint arXiv:1611.07329*, 2016.

[12] F. Kendoul and B. Ahmed, "Bio-inspired taupilot for automated aerial 4d docking and landing of unmanned aircraft systems," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 480–487.

[13] D. Maturana and S. Scherer, "3d convolutional neural networks for landing zone detection from lidar," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3471–3478.

[14] M. Shaker, M. N. Smith, S. Yue, and T. Duckett, "Vision-based landing of a simulated unmanned aerial vehicle with fast reinforcement learning," in *Emerging Security Technologies (EST), 2010 International Conference on*. IEEE, 2010, pp. 183–188.

[15] M. Saska, V. Spurnỳ, and V. Vonásek, "Predictive control and stabilization of nonholonomic formations with integrated spline-path planning," *Robotics and Autonomous Systems*, vol. 75, pp. 379–397, 2016.

[16] F. L. L. Medeiros, V. C. F. Gomes, M. R. C. d. Aquino, D. Geraldo, M. E. L. Honorato, and L. H. M. Dias, "A computer vision system for guidance of vtol uavs autonomous landing," in *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, Nov 2015, pp. 333–338.

[17] M. Valenti, D. Dale, J. P. How, D. P. D. Farias, and J. Vian, "Mission health management for 24/7 persistent surveillance operations," 2007.

[18] K. A. Swieringa, C. B. Hanson, J. R. Richardson, J. D. White, Z. Hasan, E. Qian, and A. Girard, "Autonomous battery swapping system for small-scale helicopters," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 3335–3340.

[19] K. A. O. Suzuki, P. Kemper Filho, and J. R. Morrison, "Automatic battery replacement system for uavs: Analysis and design," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 563–586, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10846-011-9616-y

[20] D. W. Byers, G. A. Hall, G. D. Hunter, C. G. Kennell, A. B. Macander, J. H. Milgram, and J. D. Strickland, "Unmanned aerial vehicle for logistical delivery," June 13 2006, uS Patent 7,059,566.

[21] J. L. Sanchez-Lopez, R. A. S. Fernández, H. Bavle, C. Sampedro, M. Molina, J. Pestana, and P. Campoy, "Aerostack: An architecture and open-source software framework for aerial robotics," in *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*. IEEE, 2016, pp. 332–341.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[23] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[24] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.

[25] A. Rodríguez-Ramos, C. Sampedro, A. Carrio, H. Bavle, R. A. Suarez-Fernandez, Z. Milosevic, and P. Campoy, "A monocular pose estimation strategy for uav autonomous navigation in gnss-denied environments," in *International Micro Air Vechicle Competition and Conference 2016*, P. Z. PENG and D. F. LIN, Eds., Beijing, PR of China, okt 2016, pp. 22–27.

[26] S. Li, C. Xu, and M. Xie, "A robust o (n) solution to the perspective-n-point problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1444–1450, 2012.

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

[28] H. Bavle, J. L. Sanchez-Lopez, A. Rodriguez-Ramos, C. Sampedro, and P. Campoy, "A flight altitude estimator for multirotor uavs in dynamic and unstructured indoor environments," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017.

[29] M. Quigley, J. Faust, T. Foote, and J. Leibs, "Ros: an open-source robot operating system."

[30] J. Pestana, I. Mellado-Bataller, J. L. Sanchez-Lopez, C. Fu, I. F. Mondragón, and P. Campoy, "A general purpose configurable controller for indoors and outdoors gps-denied navigation for multirotor unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, p. 387, 2014.