

Towards Autonomous Landing on a Moving Vessel through Fiducial Markers

Riccardo Polvara¹Sanjay Sharma¹Jian Wan¹Andrew Manning¹Robert Sutton¹

Abstract—This paper propose an autonomous landing method for unmanned aerial vehicles (UAVs), aiming to address those situations in which the landing pad is the deck of a ship. Fiducial marker are used to obtain the six-degrees of freedom (DOF) relative-pose of the UAV to the landing pad. In order to compensate interruptions of the video stream, an extended Kalman filter (EKF) is used to estimate the ship's current position with reference to its last known one, just using the odometry and the inertial data. Due to the difficulty of testing the proposed algorithm in the real world, synthetic simulations have been performed on a robotic test-bed comprising the AR Drone 2.0 and the Husky A200. The results show the EKF performs well enough in providing accurate information to direct the UAV in proximity of the other vehicle such that the marker becomes visible again. Due to the use of inertial measurements only in the data fusion process, this solution can be adopted in indoor navigation scenarios, when a global positioning system is not available.

I. INTRODUCTION

In the last few years, unmanned aerial vehicles (UAVs) attracted a lot of interest from the research and the military community, in particular those able of vertical take-off and landing (VTOL) [1].

The capability of autonomously landing, especially on the deck of a ship, is still an open research area. Given the marine conditions, characterised by adverse wind and sea currents, the estimation of the ship movements can be affected in such a way that landing is not always possible. Cameras have been identified as a solution in order to increase the estimation's accuracy; this can be further improved by the adoption of fiducial marker on the ship's deck (as depicted in Fig. 1). In the situation in which the marker tracking is interrupted, the 6-DOF ship's pose can be calculated with a state estimation filter. The choice of using an extended Kalman filter (EKF) to fuse only the odometry and inertial measurement unit (IMU) data of the moving vessel allows to successfully accomplish the task without relying on the global positioning system (GPS) signal. In this way, the algorithm can be used also in cluttered or GPS-denied environments.

In terms of the paper organisation, Section II presents the existing literature about autonomous landing for UAVs, while Section III formalise the solution proposed in this paper. In Section IV two kind of experiments, with a static landing platform and with a moving one, respectively, are presented and discussed. Finally, conclusions and future works are given in Section V.

¹Riccardo Polvara and the other authors are with the Autonomous Marine System Research Group, School of Engineering, Plymouth University, Drake Circus, Plymouth PL4 8AA, United Kingdom riccardo.polvara@plymouth.ac.uk

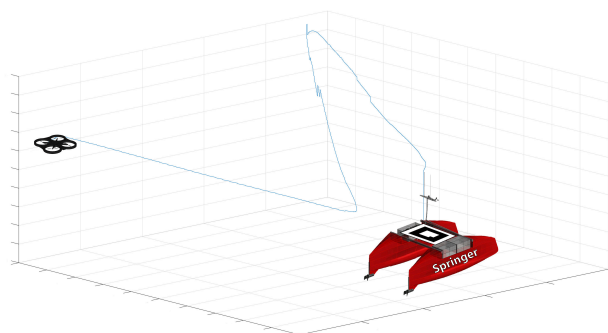


Fig. 1: An UAV landing on a fiducial marker located on the deck of an unmanned surface vehicle.

II. STATE OF THE ART

Autonomous landing is one of the most dangerous challenges for UAVs. Despite a historical large use of Inertial Navigation Systems (INS) and Global Navigation Satellite System (GNSS), vision-based solutions are becoming attractive because passive and not requiring any special equipment other than a camera and a processing unit [2], [3]. In [4] and [5] a IR-LED helipad is adopted for robust tracking and landing, while a T-shaped and an H-shaped helipad are respectively used in [6] and [7]. Here, the UAV's pose is calculated combining the projection of the pad with IMU measurements. In [8], the manoeuvre has been achieved after estimating the UAV's 3D relative position to a novel landing pad consisting of concentric circles, and assuming the vehicle is always parallel to the ground. Multiple circles of different sizes are also used in [9] with the scope of extending the detection range. In [12], multiple view geometry is used to hover after computing the relative position to a known target. In [13] authors, following the same approach, were also able to adjust the UAV's orientation. Other some bio-inspired works try to maintain a constant optic flow while descending [10], [11].

Fewer works address autonomous landing on a moving target. A first example is offered by [14], in which the authors landed on a slowly moving H-shaped marker (0.1 m/s). In [15] four light sources constitute a pattern identifiable up to 0.25m; unfortunately, this solution is not feasible when direct light interferes with the vision system. Differently, optical flow is used in [16] while an IR camera in [17].

III. METHODS

This section illustrates how autonomous landing has been achieved in this work.

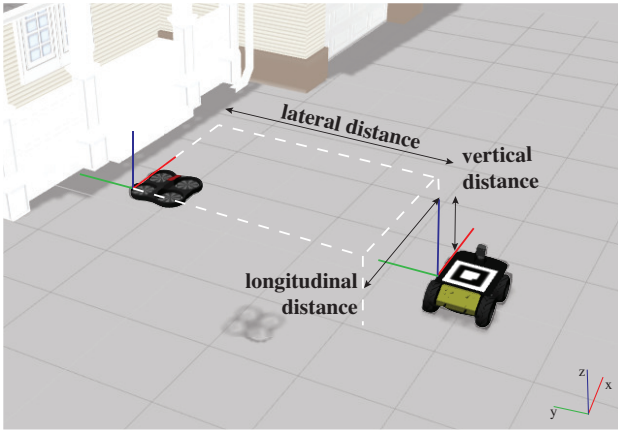


Fig. 2: The image processing algorithm estimates the 6-DOF UAV's relative-pose wrt the visual marker. Here, only the distances are reported for clarity.

A. Quad-copter model

To the scope of this paper, the Parrot AR Drone 2.0, widely used in research because affordable and integrating a complete sensor suite, has been chosen as testing platform. The Robot Operating System (ROS) [18] is used for interfacing the vehicle, using in particular the *ardrone-autonomy* and the *tum-ardrone* packages [19]. The specification of the UAV are as follow:

- Dimensions: 53 cm x 52 cm (hull included);
- Weight: 420 g;
- IMU including gyroscope, accelerometer, magnetometer, altimeter and pressure sensor and producing data at 200 Hz;
- Front-camera with a High-definition (HD) resolution (1280x720), a field of view (FOV) of 73.5 x 58.5 and video streamed at 30 frame per second (fps);
- Bottom-camera with a Quaterd Video Graphics Array (QVGA) resolution (320x240), a FOV of 47.5 x 36.5 and video streamed at 60 fps (mainly used for visual odometry);
- Central processing unit running an embedded version of Linux operating system;

The on-board controller (closed-source) is used to act on the roll Φ and pitch Θ , the yaw rotational speed Ψ and the vertical velocity of the platform \dot{z} . Control commands $u = (\Phi, \Theta, \Psi, \dot{z}) \in [-1, 1]$ are sent to the quad-copter at a frequency of 100Hz.

B. Augmented Reality

In this work, high-contrast 2D augmented reality (AR) markers are adopted for identifying the landing platform. The identification and tracking have been realised with the *ar_pose* ROS package, a wrapper for the *ARToolkit* human-computer interaction library[20].

The package subscribes to the camera's topic and the candidate marker is searched for within a database. Using the camera's calibration file and the actual size of the marker of

interest, the 6-DOF relative-pose of the marker's with respect to the UAV is estimated at a frequency of 1 Hz and shown in Fig. 2. The time stamp and the transformation for the current and the last marker's observation are stored to actuate a compensatory behaviour when the marker is lost.

C. Controller

Within the *tum-ardrone* package, a single PID controller is employed for three degrees of freedom (roll, pitch and yaw) and for the vertical velocity. Combined, they steer the quad-copter toward a desired goal pose $p = (\hat{x}, \hat{y}, \hat{z}, \hat{\psi}) \in \mathbb{R}^4$ in a global coordinate system sending commands at 100Hz.

In order to simplify the tuning process, the four PID controllers have been replaced by a single damped spring one. In the implementation used, two parameters, K_{direct} and K_{rp} , are responsible for modifying the spring strength of the directly controlled dimensions (*yaw* and *z*) and the leaning ones (*x* and *y*). An additional parameter, *xy_damping_factor*, takes account of external disturbances such as air resistance and the wind.

In this way, instead of controlling nine independent parameters (three for the yaw, three for the vertical speed and three for the pair roll-pitch) the control problem is reduced only to the three described above.

D. Pose estimation

An EKF has been adopted for estimating the position of the landing platform and offering more reliability when the marker is lost. For this purpose, data from the wheels and the IMU are fused in order to compensate the error affecting the odometry readings [21]. The filter predicts the actual position of the ship that is then forwarded to the controller. The EKF's estimate is used as follow:

- the platform's pose is estimated at 50Hz and saved in a hash table;
- when the tracking is interrupted, the table is accessed for retrieving the most recent estimation together with the last recorded observation;
- the deck's current relative-pose to the old known one is calculated;
- this information is forwarded to the controller and new commands are generated;

This procedure is iterated until the UAV is located above the marker and can newly perceive it.

E. Discussion

The quad-copter flies autonomously mainly using its fixed frontal camera, approaching the landing platform identified by a fiducial marker. Despite the EKF can compensate interruption in observation, it is required that the marker is perceived among all the landing manoeuvre. The *ar_pose* library computes the 6-DOF relative-pose between the UAV and the landing platform. This information is used by the controller to make the quad-copter approaching the marker with the right orientation.

Due to the hardware limitation of the UAV chosen, in particular the presence of two fixed non-tilting cameras, a

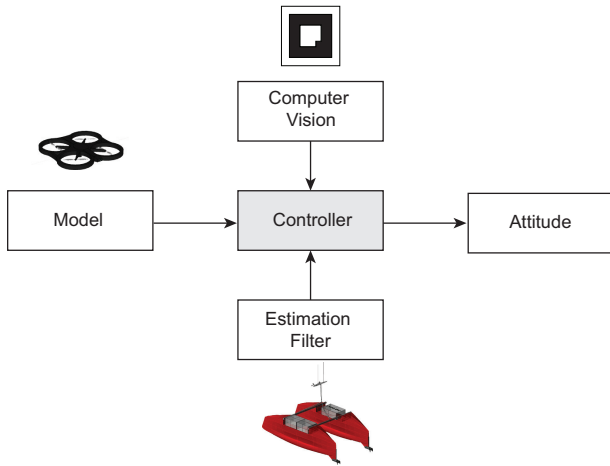


Fig. 3: Different components are integrated for achieving autonomous landing.

switching system has been adopted to guarantee maximum continuity while observing and tracking the landing pad. In fact, it happens that the UAV lose the tracking while descending and using the frontal camera. To overcome this problem, the video stream starts to be acquired from the downward-looking camera located at the bottom of the UAV. The quad-rotor can then try to land while centring the marker in the camera’s FOV. Otherwise, the EKF estimates the actual position of the landing base and the drone is there redirected increasing its altitude. In this way, it is guaranteed to perceive the marker soon. When an user-specified distance from the marker is met, the drone shuts down its motors and land on it. A graphical representation of the overall system is depicted in Fig. 3.

It is important to remark that visual marker permits the estimation of the full 6-DOF relative-pose, meaning that pitch and roll of the deck are considered. Nevertheless, the algorithm does not directly take care of these movements since they can be easily addressed mounting a pan-tilt unit under the landing platform in order to stabilise it whatever weather conditions. For this reason, it is possible to claim that the method developed is theoretically applicable to every landing scenario, especially when involving UGVs not subject to significant rolling and pitching behaviours.

IV. RESULTS

The proposed algorithm has been tested in simulation within a modified version of the *tum_simulator* package, a 3D environment built on Gazebo 2.2.X and offering a 3D model of the AR Drone 2.0. Given the absence of maritime robot models and marine scenario, the Husky A200 has been chosen as landing base. Here, a square fiducial marker having a side length of 0.31 meter was placed. Due to the paper’s length limitation, three experiments are now reported. In the first, the Husky remains in the same position for all the length of the flight. In the second experiment, the platform is moving in a straight line at constant speed. This is a common scenario when deploying manned/unmanned vessel

TABLE I: Controller parameters for the static landing platform experiment.

Parameter	Value	Parameter	Value
K_{direct}	5.0	K_{rp}	0.3
$xy_damping_factor$	0.65	max_gaz_drop (m/s)	0.1
max_yaw (rad/s)	1.0	–	–

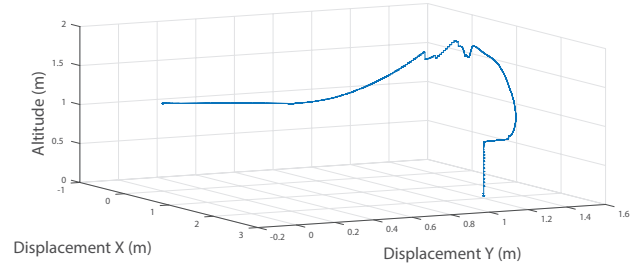


Fig. 4: Quad-copter trajectory in three-dimensional space during the experiment with a static platform.

that spends most of their time traversing keeping a fixed heading angle. As last, the UGV is also rotating.

A. Static Platform

The aim of this experiment is to test the alignment of the UAV with a visual marker. In Table I the controller setting is reported. The K_{rp} parameter, responsible for roll and pitch, is set to a small value to guarantee smooth movements while translating. In the same way, max_gaz_drop is reduced to a value of 0.1 for slowing down the descending manoeuvre. On the other hand, the max_yaw parameter, controlling the yaw rotational speed, is maximised because the alignment must be realised in the shortest amount of time possible.

Fig. 4 illustrates the UAV’s trajectory for this experiment. The fiducial marker has been successfully recognised at around 2 meters in front of the UAV, and at 1.2 meter on its left. The displacement on the z -axis, used as the reference

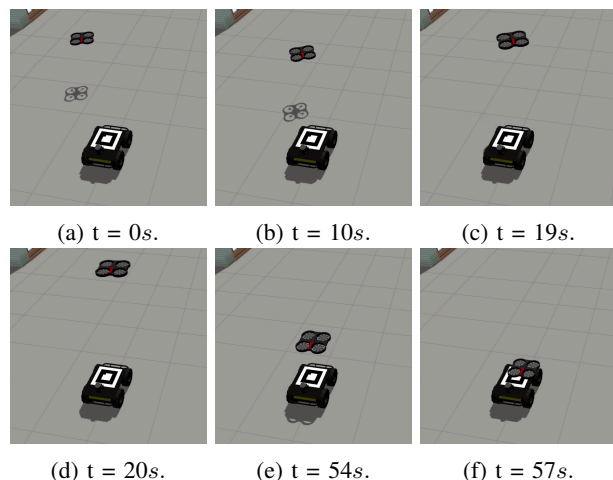


Fig. 5: Landing manoeuvre of a VTOL UAV on a static platform.

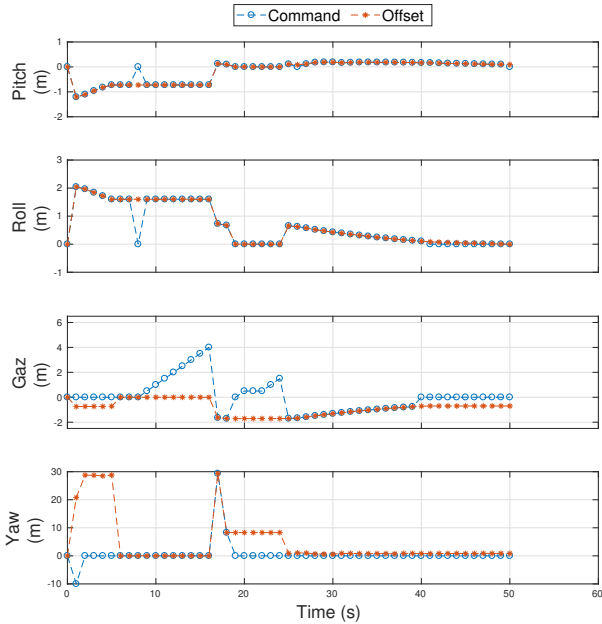


Fig. 6: Controller commands and visual offsets in the experiment with static landing platform.

for the altitude, was around 0.7 meter instead. The initial situation is illustrated in Fig. 5a. The UAV has been able to complete the landing in 50 seconds.

During the flight, the quad-copter approaches the pad keeping the marker at the centre of its camera’s FOV. An interval of confidence of 10 degrees is defined and the UAV rotates accordingly when the marker is out of it. While descending, the UAV’s low altitude prevents the frontal camera from perceiving the marker. This is basically what happens at $t = 10s$ and depicted in Fig. 5b. At this point, the video stream is switched to the downward-looking camera. The UAV is instructed to move towards the marker’s last known position while increasing its altitude for increasing the area covered by the camera. When the UAV is located exactly above the marker (more precisely at $t = 19s$ as in Fig. 5c), it can complete the landing finding first the right orientation ($t = 20s$) and then descending keeping the marker centred, as shown in Fig. 5d. At $t = 54s$ the quad-copter reaches a user-defined altitude (set to 0.75m): it can now shut down its motors and land on the platform (Fig. 5f).

In Fig. 6 the controller commands are plotted against the marker’s observation. For most of the time, the two curves overlap, meaning the marker’s observations are directly forwarded to the controller. The only portion in which they do not is between $t = 8s$ and $t = 25s$, when the marker is lost twice. Therefore, the compensatory behaviour is adopted, sending the UAV to the marker’s last known position while keeping its orientation constant. In this way, the pitch, roll and yaw commands do not change meanwhile the UAV’s altitude increases. A dedicated analysis is reserved for the

TABLE II: Controller parameters for the moving landing platform experiment.

Parameter	Value	Parameter	Value
K_direct	5.0	K_rp	1.0
xy_damping_factor	0.65	max_gaz_drop (m/s)	-0.15
max_yaw (rad/s)	1.0	–	–

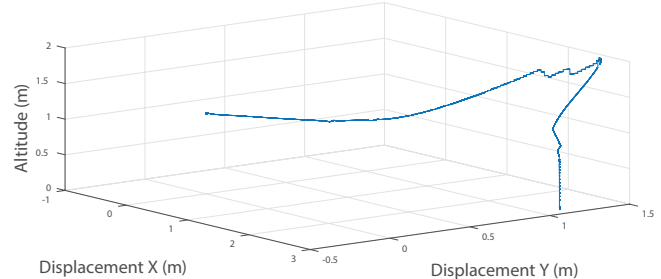


Fig. 7: Quad-copter trajectory in three-dimensional space and top view during the experiment with a moving platform proceeding in straight line.

altitude in the interval $t = [1, 5]s$ and $t = [40, 50]s$, and the yaw in $t = [1, 5]s$ and after $t = 18s$. Here, the offsets are below a user-defined threshold and a respective command equal to 0 is sent instead. The threshold is introduced because it was noticed that fixed-parameters controllers, like the PID or the one used, offer limited performances in a scenario like the one studied, in which high accuracy is required. An adaptive solution is planned as future work.

B. Moving Platform

In this subsection, the UAV is tested against landing on a moving platform. Due to the small size of the deck and the limitations posed by fixed non-tilting cameras, a stopping command is sent to the UGV once the quad-copter is located exactly above it. This decision was also forced by the trade-off posed by the dimension of the visual marker: a smaller one could be perceived from shortest distances because occupying a smaller portion in the camera’s FOV; on the other hand, it would be more difficult to perceive it from far.

The controller setting is reported in Table II. Differently, from the previous experiment, the K_{rp} parameter assumes now a bigger value to make the UAV to translate faster. To allow the UAV descending at high speed while maintaining the tracking, the value of the max_gaz_drop parameter is decreased. Lastly, max_yaw is kept at its maximum value to minimise the alignment time with the landing base. Regarding the UGV, a constant velocity command is sent to make it move in a straight line.

The flight trajectory is reported in Fig. 7. The quad-copter performed an autonomous landing in around 40 seconds.

The initial scenario is reported in Fig. 9a. At $t = 16s$, as seen in Fig. 9b, the UAV reaches an altitude such that it is impossible to continue to perceive the marker. The video stream is therefore acquired from the bottom camera and the UGV’s estimated position is sent to the controller while

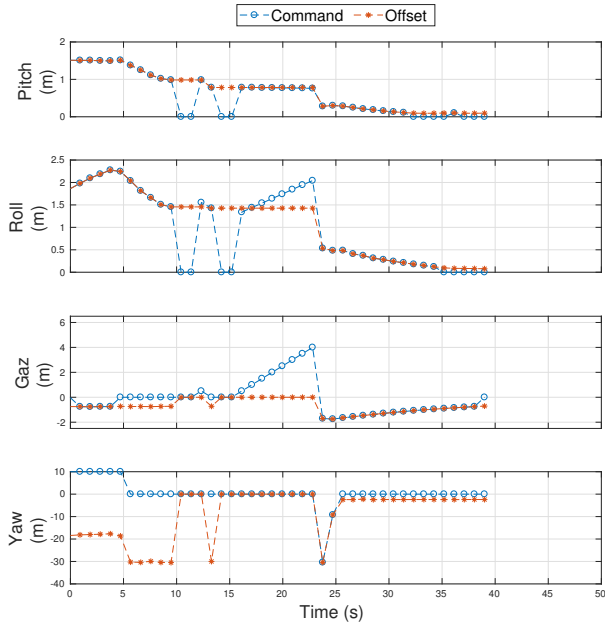


Fig. 8: Controller commands and visual offsets in the experiment with a mobile platform moving in straight line.

increasing the drone’s altitude. Doing so, at $t = 23s$ the UAV is located exactly above the UGV and it rotates accordingly until the proper orientation is reached (Fig. 9d). Because the landing base is at the centre of the camera’s FOV, a null velocity base command is sent to stop the UGV. Fig. 9e and 9f show the UAV descending slowly on the marker and landing on it.

Further analysis can be done with the results reported in Fig. 8. As in the first experiment, the curve of the controller’s commands and the offsets one overlap for most of the time. The difference with the previous case is given by the EKF’s contribution. It is possible to have an example looking at the plot in $t = [10, 23]s$. Here, the marker has been lost twice and the curves differ: if, on one hand, the offsets are constant because no new observations have been done, on the other hand, the commands slightly change. In $t = [16, 23]s$, while the yaw commands assume a constant value of 0 because the UAV is properly oriented, the pitch does not change because the UGV is moving only in a straight line, not deviating from the lateral direction. For the same reason, the roll command is updated to account every instant of the new relative-pose (changing the longitudinal direction) of the UGV. In the last part of the graph, the difference between the two curves for the roll (in $t = [35, 40]s$), the altitude (in $t = [5, 10]s$) and the yaw (from $t = 25s$ till the end) is justified by the adoption of a threshold to speed up the completion of the landing manoeuvre. To conclude, the two yaw’s curves deserve a deeper analysis. As already discussed, while using the frontal camera, the UAV keeps the marker at the centre of the FOV. If the UAV rotated to align, it would most probably lose the

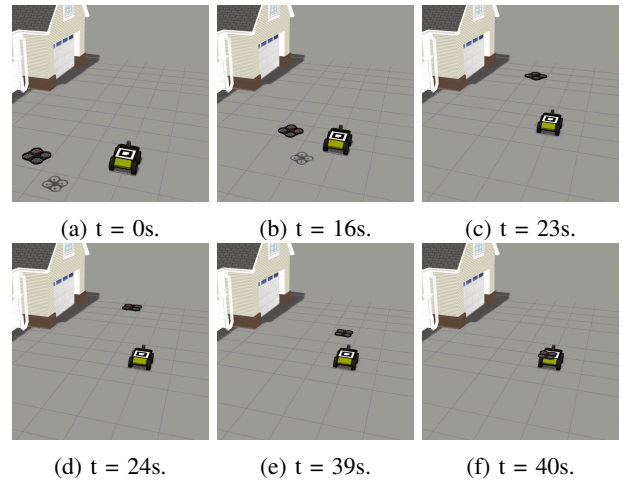


Fig. 9: Landing manoeuvre of a VTOL UAV on a mobile platform moving in straight line.

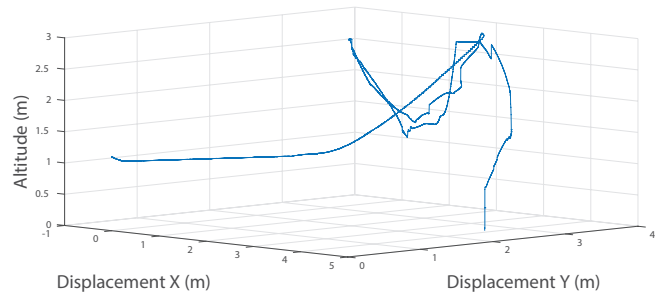


Fig. 10: Quad-copter trajectory in three-dimensional space and top view during the experiment with a moving platform that also rotate.

tracking. To avoid this situation, the rotation is performed only ten degrees per time and only if the marker is located at the edge of the camera’s frame. Doing so, the alignment is slowed down but a continuous tracking is guaranteed.

A last experiment has been done with a moving platform that proceeds not only in straight line but rotates changing its heading angle. Note that the UGV is able only to move in straight line or rotate on the spot at the same time. The goal of this simulation is to show how the EKF can compensate the lack of the vision technique in predicting where the UAV has to move to perceive the visual marker again. The controller parameters used are the same of the previous experiment and they are shown in Table II. Results are reported in Fig. 10, where the UAV’s trajectory is drawn. In Fig. 11 is shown the comparison between the offsets obtained through the vision algorithm and the commands sent to the controller. Here, it is possible to see that, as in the previous scenarios, the plot of the offsets and the one related to the commands overlap for most of the time. All the analysis made before still hold, but it is interesting to notice how the algorithm is able to react properly when the visual marker is lost, redirecting the UAV above it despite changes in position.

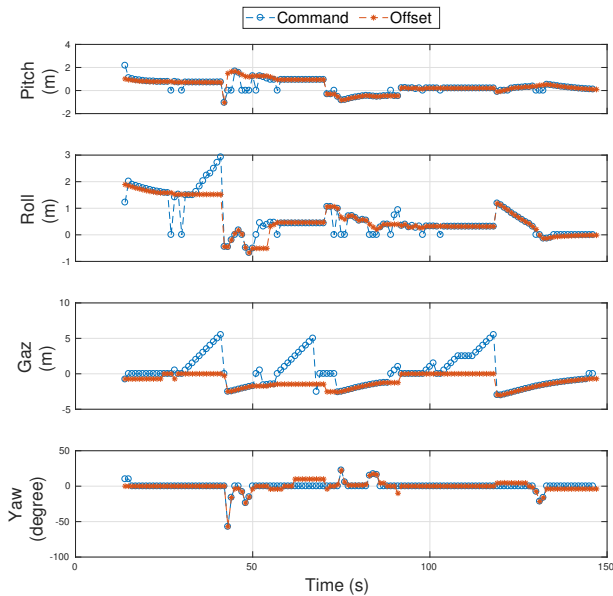


Fig. 11: Controller commands and visual offsets in the experiment with a mobile platform that also rotate.

V. CONCLUSION AND FUTURE DIRECTIONS

In this paper, autonomous landing on a ship's deck is studied and tested in a 3D simulator. Due to unavailability of marine robot and scenarios model, simulations have been performed to a ground unmanned vehicle. The solution proposed resides only on the UAV's on-board sensors and on the adoption of a visual marker on the landing platform to easily calculate its 6-DOF pose. In this way, despite not tested yet, the algorithm is supposed to be robust also when adverse marine conditions affect the rolling and pitching dimension of the vehicle on which the UAV must land on. The adoption of EKF allows overcoming issues with fixed non-tilting cameras. Not involving GPS signals makes this solution feasible also in indoor scenarios or adverse weather conditions.

Three experiments, against a static and a moving base, were performed to validate the approach. In all the cases, successful results have been obtained. An adaptive controller, based on an intelligent solution such as artificial neural networks or fuzzy logic, is identified as future research to compensate the limits of a fixed-parameters controller such as the one used in this work.

REFERENCES

[1] V. Kumar and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, 2012.
 [2] W. Kong, D. Zhou, D. Zhang, and J. Zhang, "Vision-based autonomous landing system for unmanned aerial vehicle: A survey," in *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*. IEEE, 2014, pp. 1–8.

[3] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, 2012.
 [4] A. Masselli, S. Yang, K. E. Wenzel, and A. Zell, "A cross-platform comparison of visual marker based approaches for autonomous flight of quadcopters," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, May 2013, pp. 685–693.
 [5] K. E. Wenzel, P. Rosset, and A. Zell, "Low-cost visual tracking of a landing place and hovering flight control with a microcontroller," in *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*. Springer, 2009, pp. 297–311.
 [6] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, and S. Longhi, "A vision-based guidance system for uav navigation and safe landing using natural landmarks," in *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*. Springer, 2009, pp. 233–257.
 [7] S. Yang, S. A. Scherer, and A. Zell, "An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle," *Journal of Intelligent & Robotic Systems*, pp. 1–17, 2013.
 [8] S. Lange, N. Sunderhauf, and P. Protzel, "A vision based onboard approach for landing and position control of an autonomous multirotor uav in gps-denied environments," in *2009 International Conference on Advanced Robotics*, June 2009, pp. 1–6.
 [9] T. Merz, S. Duranti, and G. Conte, *Autonomous Landing of an Unmanned Helicopter based on Vision and Inertial Sensing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 343–352. [Online]. Available: http://dx.doi.org/10.1007/11552246_33
 [10] B. Barber, T. McLain, and B. Edwards, "Vision-based landing of fixed-wing miniature air vehicles," *Journal of Aerospace computing, Information, and Communication*, vol. 6, no. 3, pp. 207–226, 2009.
 [11] F. Ruffier and N. Franceschini, "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.
 [12] O. Shakernia, R. Vidal, C. S. Sharp, Y. Ma, and S. Sastry, "Multiple view motion estimation and control for landing an unmanned aerial vehicle," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 3. IEEE, 2002, pp. 2793–2798.
 [13] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," in *Robotics and automation, 2002. Proceedings. ICRA'02. IEEE international conference on*, vol. 3. IEEE, 2002, pp. 2799–2804.
 [14] —, "Visually guided landing of an unmanned aerial vehicle," *IEEE transactions on robotics and automation*, vol. 19, no. 3, pp. 371–380, 2003.
 [15] 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.
 [16] 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.
 [17] O. A. Yakimenko, I. I. Kaminer, W. J. Lentz, and P. Ghyzel, "Unmanned aircraft navigation for shipboard landing using infrared vision," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 4, pp. 1181–1200, 2002.
 [18] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
 [19] J. Engel, J. Sturm, and D. Cremers, "Scale-aware navigation of a low-cost quadcopter with a monocular camera," *Robotics and Autonomous Systems (RAS)*, vol. 62, no. 11, pp. 1646–1656, 2014.
 [20] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*. IEEE, 1999, pp. 85–94.
 [21] L. Jetto, S. Longhi, and G. Venturini, "Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 219–229, 1999.