MDPI

Article

Vision-based Autonomous Landing of a Quadrotor on the Perturbed Deck of an Unmanned Surface Vehicle

Riccardo Polvara [1]* 0000-0001-8318-7269, Sanjay Sharma [1], Jian Wan [1], Andrew Manning [1] and Robert Sutton [1]

[1] Autonomous Marine Systems Research Group, School of Engineering, University of Plymouth, UK; name.surname@plymouth.ac.uk

* Correspondence: riccardo.polvara@plymouth.ac.uk; Tel.: +44-07492-050558

Version April 16, 2018 submitted to Drones

Abstract: Autonomous landing on the deck of an unmanned surface vehicle (USV) is still a major challenge for unmanned aerial vehicles(UAVs). In this paper, a fiducial marker is located on the platform so as to facilitate the task since it is possible to retrieve its six-degrees of freedom relative-pose in an easy way. To compensate interruption in the marker's observations, an extended Kalman filter (EKF) estimates the current USV's position with reference to the last known position. Validation experiments have been performed in a simulated environment under various marine conditions. The results confirmed the EKF provides estimates accurate enough to direct the UAV in proximity of the autonomous vessel such that the marker becomes visible again. Using only the odometry and the inertial measurements for the estimation, this method is found to be applicable even under adverse weather conditions in the absence of global positioning system.

Keywords: Unmanned Aerial Vehicle; Position Control; Computer Vision; Image Processing

0. Introduction

In the last few years, significant interest has grown towards Unmanned Aerial vehicles (UAVs), as described in [1]. The applications involving UAVs range from scientific exploration and data collection [2–4], to commercial services, military reconnaissance and law enforcement [5,6], search and rescue [7,8], patrolling [9] and even entertainment [10].

Among different UAVs topologies, helicopter flight capabilities such as hovering or vertical take-off and landing (VTOL) represent a valuable advantage over fixed-wing aircraft. The ability of autonomously landing is very important for unmanned aerial vehicles, and landing on the deck of a un-/manned ship is still an open research area. Landing an UAV on an unmanned surface vehicle (USV) is a complex multi-agent problem [11] and solutions to this can be used fro numerous applications such as disaster monitoring [12], coastal surveillance [13,14] and wildlife monitoring [15,16]. In addition, a flying vehicle can also represents an additional sensor data source when planning a safe collision-free path for USVs [17].

Flying an UAV in the marine environment encounters rough and unpredictable operating conditions due to the influence of wind or wave in the manoeuvre compare to land. Apart from above, there are various other challenges associated with the operation of UAVs. For example, the inaccuracy of low-cost GPS units mounted on most UAV and the influence of the electrical noise generated by the motors and on-board computers on magnetometers. In addition to this, the estimation of the USV's movements is a difficult task due to natural disturbances (e.g. winds, sea currents etc.). This poses difficulty for an UAV to land on a moving marine vehicle with a low quality pose information. To overcome these issues, the camera mounted on the UAV and commonly used during surveillance mission [18], can also be used to increase the accuracy of the relative-pose estimates between the

aerial vehicle and the landing platform [19]. The adoption of fiducial markers on the vessel's deck is proposed as solution to further improve the estimate results. To increase the robustness of the approach, a state estimation filter is adopted for predicting the 6 degrees-of-freedom (DOF) pose of the landing deck which is not perceived by the UAV's cameras. This work can be considered as the natural consequence of [20], in which the developed algorithm has been tested against a mobile ground robot, without any pitch and roll movements of the landing platform.

In terms of the paper organisation, Section 1 presents the method existing in literature about autonomous landing for UAVs, while Section 2 introduces the quad-copter model, the image processing library used for the deck identification, the UAV controller and the pose estimation filter. In Section 3 three experiments, each with a different kind of perturbation acting on the landing platform, are presented and discussed. Finally, conclusions and future works are shown in Section 4.

1. State of the Art

Autonomous landing is until now one of the most dangerous challenges for UAV. Inertial Navigation Systems (INS) and Global Navigation Satellite System (GNSS) are the traditional sensors of the navigation system. On the other hand, INS accumulates error while integrating position and velocity of the vehicle and the GNSS sometimes fails when satellites are occluded by buildings. At this stage, vision-based landing became attractive because it is passive and does not require any special equipment other than a camera (generally already mounted on the vehicle) and a processing unit. The problem of accurately landing using vision-based control has been well studied. For a detailed survey about autonomously landing, please refer to [21–23]. Here, only a small amount of works are presented.

In [24] and [25] an IR-LED helipad is adopted for robust tracking and landing, while a more traditional T-shaped and H-shaped helipad are used respectively in [26–29]. The landing site is searched for an area whose pixels have a contrast value below a given threshold in [30]. In [31] a Light Imaging, Detection, And Ranging (LIDAR) sensor is combined with a camera and the approach has been tested with a full-scale helicopter. Bio-inspired by the honeybees that use optic flow to guide landing, [32] follow the approach for fixed-wing UAV. The same has been done in [33,34] showing that by maintaining constant optic flow during the manoeuvre, the vehicle can be easily controlled. Hovering and landing control of a UAV on a large textured moving platform enabled by measuring optical flow is achieved in [35]. In [36], a vision algorithm based on multiple view geometry detects a known target and computes the relative position and orientation. The controller is able to control only the x and y positions to hover on the platform. In a similar work [37], the authors were also able to regulate the UAV's orientation to a set point hover. In [38] an omnidirectional camera has been used to extend the field of view of the observations.

Four light sources have been located on a ground robot and homography is used to perform autonomous take-off, tracking, and landing on a UGV [39]. In order to land on a ground robot, [40] introduces a switching control approach based on optical flow to react when the landing pad is out of the UAV's camera field of view. In [41], the authors propose the use of an IR camera to track a ship from long distances using its shape, when the ship-deck and rotocraft are close in. Similarly, [42] address the problem of landing on a ship moving only on a 2D plane without its motion known in advance.

The work presented in this paper must be collocated among vision-based methods. Differently from most of them, given the platform used it relies on a pair of low resolution fixed RGB cameras, without requiring the vehicle to be provided with other sensors. Furthermore, instead of estimating the current pose of the UAV, in order to land on a moving platform we employ an extended Kalman filter for predicting the current position of the vessel on whose deck the landing pad is located. The estimate is forwarded in input to our control algorithm that update the last observed USV's pose and send a new command to the UAV. In this way, even if the landing pad is not within the camera's field of view any more, the UAV can start a recovery manoeuvre that, differently from other works, is

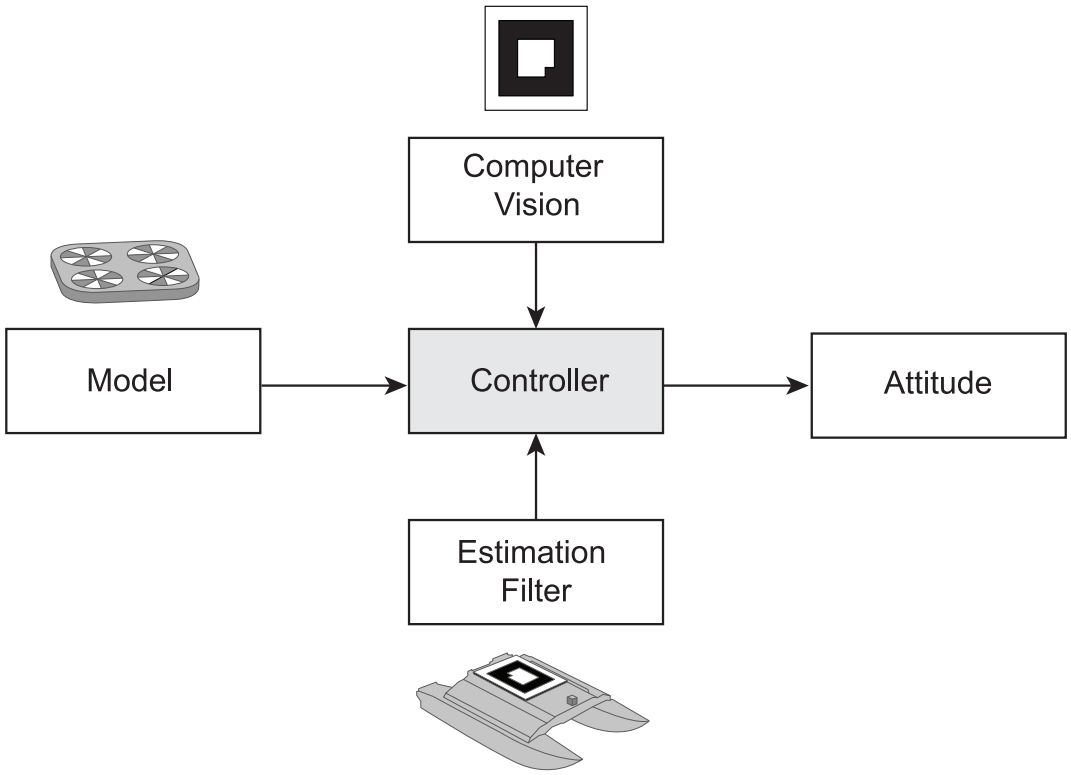

Figure 1. Different components are integrated for achieving autonomous landing on the deck of an unmanned surface vehicle.

taking the drone in proximity of its final destination. In this way it can compensate interruptions in the tracking due to changes in attitude of the USV's deck on which the pad is located.

2. Methods

In this section all the components used for accomplishing the autonomous landing on an USV are introduced. Initially, the aerial vehicle, together with its mathematical formulation, is described. Successively, the *ar_pose* computer vision library is presented. In the end, the controller and the pose estimation filter are discussed. A graphical representation of these components is depicted in Fig. 1 and a video showing the overall working principle is available online [1].

2.1. Quad-copter model

The quad-copter in this study is an affordable ($250 USD in 2017) AR Drone 2.0 built by the French company Parrot and it comprises multiple sensors such as two cameras, a processing unit, gyroscope, accelerometers, magnetometer, altimeter and pressure sensor. It is equipped with an external hull for indoor navigation and it is mainly piloted using smart-phones and tablets through the application released by the producer over a WiFi network. Despite the availability of an official software development kit (SDK), the Robot Operating System (ROS) [43] framework is used to communicate with it, using in particular the *ardrone-autonomy* package developed by the Autonomy Laboratory of Simon Fraser University, and the the *tum-ardrone* package [44–46] developed within the TUM Computer Vision Group in Munich. These package run within ROS Indigo on a GNU/Linux Ubuntu 14.04 LTS machine. The specification of the UAV are as follow:

[1] Video showing the working principle of the algorithm: https://youtu.be/J1ib9PIsr-8

- Dimensions: 53 cm x 52 cm (hull included);
- Weight: 420 g;
- Inertial Measurements Units (IMU) including gyroscope, accelerometer, magnetometer, altimeter and pressure sensor;
- Front-camera with a High-definition (HD) resolution (1280x720), a field of view (FOV) of 73.5° × 58.5° and video streamed at 30 frame per second (fps);
- Bottom-camera with a Quarted Video graphics Array (QVGA) resolution (320x240), a FOV of 47.5° × 36.5° and video streamed at 60 fps;
- Central processing unit running an embedded version of Linux operating system;

The downward-looking camera is mainly used to estimate the horizontal velocity and the accuracy of the estimation highly depends on the ground texture and the quad-copter's altitude. Only one of the two video streams can be streamed at the same time. Sensors data are generated at 200Hz. The on-board controller (closed-source) is used to act on the roll Φ and pitch Θ, the yaw Ψ and the altitude of the platform z. Control commands $u = (\Phi, \Theta, \Psi, z) \in$ [-1,1] are sent to the quad-copter at a frequency of 100Hz.

While defining the UAV dynamics model, the vehicle must be considered as a rigid body with 6-DOF able to generate the necessary forces and moments for moving [47]. The equations of motion are expressed in the body-fixed reference frame $\mathcal{B}$ [48]:

$$\begin{cases} m\dot{V} + \Omega \times mV = F \\ J\dot{\Omega} + \Omega \times J\Omega = \Gamma^b \end{cases} \tag{1}$$

where $V = [u, v, w]^T$ and $\Omega = [p, q, r]^T$ represent, respectively, the linear and angular velocities of the UAV in $\mathcal{B}$. F is the translational force combining gravity, thrust and other components, while $J \in R^{3\times3}$ is the inertial matrix subject to F and torque vector Γ^b.

The orientation of the UAV in air is given by a rotation matrix R from $\mathcal{B}$ to the inertial reference frame $\mathcal{I}$:

$$\begin{aligned} R &= R_\psi R_\theta R_\phi \\ &= \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \end{aligned} \tag{2}$$

where $\eta = [\phi, \theta, \psi]^T$ is the Euler angles vector and $s.$ and $c.$ are abbreviations for $sin(.)$ and $cos(.)$.

Given the transformation from the body frame $\mathcal{B}$ to the inertial frame $\mathcal{I}$, the gravitational force and the translational dynamics in $\mathcal{I}$ are obtained in the following way:

$$\begin{cases} \dot{\xi} = v \\ m\dot{v} = RF^b - mge_3^i \end{cases} \tag{3}$$

where g is the gravitational acceleration and F^b is the resulting force in $\mathcal{B}$, $\xi = [x, y, z]^T$ and $v = [\dot{x}, \dot{y}, \dot{z}]^T$ are the UAV's position and velocity in $\mathcal{I}$.

2.2. Augmented Reality

The UAV's body frame follows right-handed z-up convention such that the positive x-axis is oriented along the UAV's forward direction of travel. Both camera frames are fixed with respect to the UAV's body one, but translated and rotated in such a way that the positive z-axis points out of the camera lens, the x-axis points to the right from the image centre and the y-axis points down. The USV's frame also follows the same convention and is positioned at the centre of the landing platform.

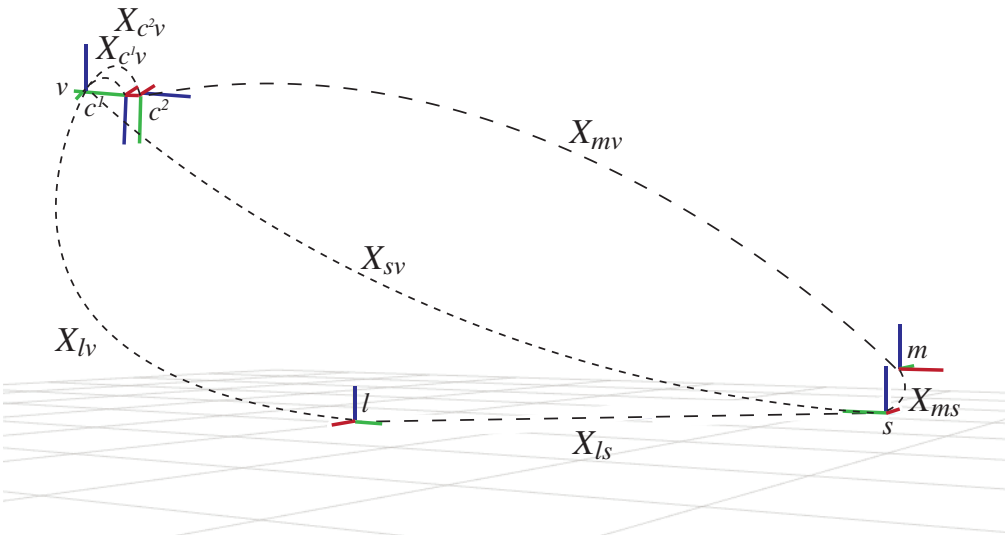

Figure 2. Coordinate frames for the landing systems. X_{lv} represents the UAV's pose with reference to the local frame and, in the same way, X_{ls} for the USV. X_{c^1v} and X_{c^2v} are the transformation between the down-looking camera and frontal cameras, respectively, and the vehicle's body frame. X_{mv} and X_{ms} are the pose from the visual marker to the UAV and to the USV, respectively. Finally, X_{sv} is the pose from the USV to the UAV.

Finally, it has been defined a local frame fixed with respect to the world and initialized by the system at an arbitrary location. In Fig. 2 the coordinate systems previously described are depicted.

The pose of frame j with respect to frame i is now defined as the 6-DOF vector:

$$x_{i,j} = [{}^i t_{i,j}^T, \Theta_{i,j}^T]^T = [x_{i,j}, y_{i,j}, z_{i,j}, \phi_{i,j}, \theta_{i,j}, \psi_{i,j}]^T \tag{4}$$

composed of the translation vector from frame i to frame j and the the Euler angles ϕ, θ, ψ.

Then, the homogeneous coordinate transformation from frame j to frame i can be written as:

$${}^i_j H = \begin{bmatrix} {}^i_j R & {}^i t_i^T j \\ 0 & 1 \end{bmatrix} \tag{5}$$

where ${}^i_j R$ is the orthonormal rotation matrix that rotates frame j into frame i and is defined as:

$${}^i_j R = rotxyz(\Theta_i j) = rotz(\psi_i j)^T roty(\theta_i j)^T rotx(\phi_i j)^T \tag{6}$$

Fig. 3 offers a graphical representation of the problem studied: retrieving the homogeneous matrix H offers the possibility to calculate the UAV's pose with reference to the USV expressed as translation and rotation along and around three axis respectively.

In this work, augmented reality (AR) visual markers are adopted for identifying the landing platform. As described in [49], "in a AR virtual objects super-imposed upon or composited with the real world. Therefore, AR supplements reality".

The *ar_pose* ROS package [50], a wrapper for the *ARToolkit* library widely used in human computer interaction (HCI) [51,52], is used for achieving this task. The *ar_pose* markers are high-contrast 2D tags designed to be robust to low image resolution, occlusions, rotations and lighting variation. For this reason it is considered suitable for a possible application in a marine scenario, where the landing platform can be subject to adverse conditions that can affect its direct observation.

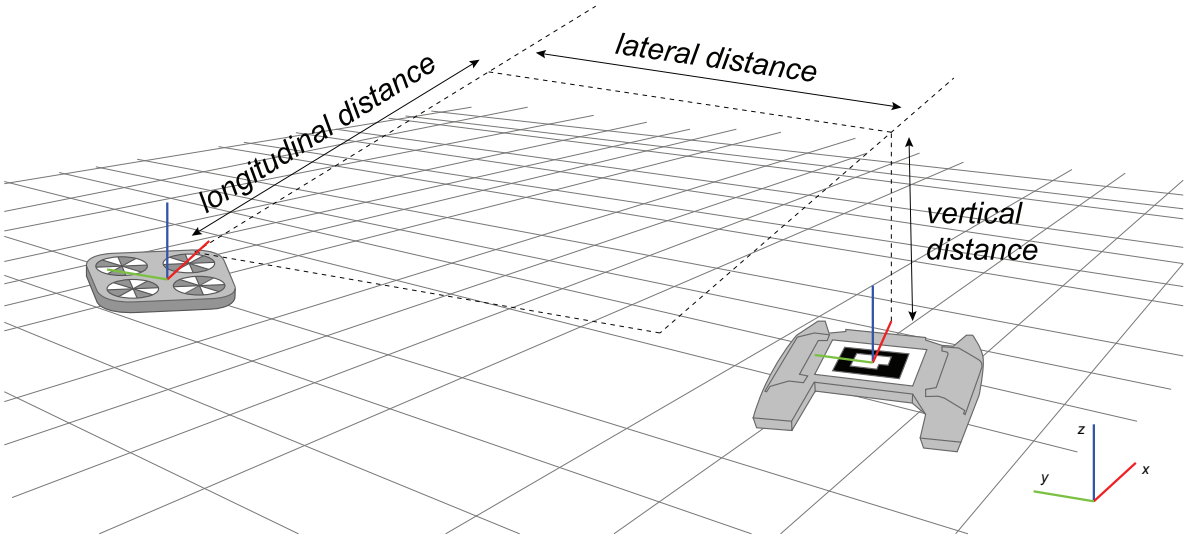

Figure 3. The image processing algorithm estimates the distances between the UAV and the visual marker.

In order to use this library, the camera calibration file, the marker's dimension and the proper topic's name must be defined inside a configuration file. The package subscribes to one of the two cameras. Pixels in the current frame are clustered based on similar gradient and candidate markers are identified. The Direct Linear Transform (DLT) algorithm [53] maps the tag's coordinate frame to the camera's one, and the candidate marker is searched for within a database containing pre-trained markers. The points in the marker's frame and camera's frame are respectively denoted as ${}^{M}\mathrm{P}$ and ${}^{C}\mathrm{P}$. So, the transformation from one frame to the other is defined as follow:

$$ {}^{M}P = {}^{M}_{C}H\,{}^{C}P = {}^{M}_{C}H^{-1}\,{}^{C}P = {}^{C}_{M}H\,{}^{C}P \tag{7} $$

where ${}^{M}_{C}H$ and ${}^{C}_{M}H$ represent the transforms from the marker to the camera frame and vice versa, respectively.

Using the camera's calibration file and the actual size of the marker of interest, the 6-DOF relative-pose of the marker's frame with respect to the UAV camera is estimated at a frequency of 1 Hz. For the current and the last marker's observation, the time stamp and the transformation are recorded. These informations are then used to detect if the marker has been lost and to actuate a compensatory behaviour.

2.3. Controller

In order to control the drone in a less complex way, the PID controller offered by the tum_ardrone package has been replaced with a (critically) damped spring one.

In the original work of [46], for each of the four degrees of freedom (roll Φ, pitch $\bar{\Theta}$, the yaw Ψ and the altitude $\bar{z}$), a separate PID controller is employed. Each of them is used to steer the quad-copter toward a desired goal position $p = (\hat{x}, \hat{y}, \hat{z}, \hat{\psi}) \in \mathbb{R}^4$ in a global coordinate system. The generated controls are then transformed into a robotic-centric coordinate frame and sent to the UAV at 100Hz.

In this paper, in order to simplify the process of tuning the controller's parameters, a damped spring controller has been adopted. In the implementation, only two parameters, K_direct and K_rp, were used to modify the spring strength of the directly controlled dimensions (yaw and z) and the leaning ones (x and y). An additional one, $xy\_damping\_factor$, is responsible to approximate a damped spring and to account external disturbances such as air resistance and wind.

Table 1. The controller parameters used in the simulation performed.

Parameter	Value	Parameter	Value
K_direct	5.0	K_rp	0.3
droneMass [kg]	0.525	max_yaw [rad/s]	1.0
xy_damping_factor st19	0.65	max_gaz_rise [m/s]	1.0
max_gaz_drop [m/s]	-0.1	max_rp	1.0

The controller inputs are variations in the angles of roll, pitch, yaw, and altitude, respectively denoted as u_Φ,u_Θ, u_Ψ and u_z, defined as follows:

$$u_\Phi = -K\_rp(\hat{x} - x) + c\_rp(\hat{\dot{x}} - \dot{x}) \tag{8}$$

$$u_\Theta = -K\_rp(\hat{y} - y) + c\_rp(\hat{\dot{y}} - \dot{y}) \tag{9}$$

$$u_\Psi = -K\_direct(\hat{\psi} - \psi) + c\_direct(\hat{\dot{\psi}} - \dot{\psi}) \tag{10}$$

$$u_z = -K\_direct(\hat{z} - z) + c\_direct(\hat{\dot{z}} - \dot{z}) \tag{11}$$

where c_rp and c_direct are the damping coefficients calcuolated in the following way:

$$c\_rp = xy\_damping\_factor \cdot 2\sqrt{K\_rp \cdot droneMass} \tag{12}$$

$$c\_direct = 2\sqrt{K\_direct \cdot droneMass} \tag{13}$$

Therefore, instead of controlling nine independent parameters (three for the yaw, three for the vertical speed and three for roll and pitch paired together) the control problem is reduced to the three described above (namely *K_direct*, *K_rp* and *xy_damping_factor*).

The remaining controller parameters are platform dependent variables and they are kept always constant during all the trials. Ignoring *droneMass* which does not require an additional description more than its name, *max_yaw*, *max_gaz_rise* and *max_gaz_drop* limit the rotation and linear speed on the yaw and z-axis, respectively. In the end, *max_rp* limits the maximum leaning command sent.

The controller's parameters are the same across all the experiments performed and they are shown in Table 1. The *K_rp* parameter, responsible to control the roll and pitch behaviour, is kept small in order to guarantee smooth movements along the leaning dimensions. In the same way, *max_gaz_drop* has been reduced to a value of 0.1 for decreasing the descending velocity. On the other hand, the *max_yaw* parameter, used to control the yaw speed, has been set to its maximum value because the drone must align with the base in the minimum amount of time possible. The others have been left to their default values.

2.4. Pose estimation

To increase the robustness and efficiency of the approach, an extended Kalman filter (EKF) has been adopted here for estimating the pose of the landing platform [54]. In fact, it may happen the UAV lose the track of the fiducial marker while approaching and descending on it. In order to redirect the flying vehicle in the right direction, the EKF estimates the USV current pose that is then processed and forwarded to the controller. For estimation purposes, the odometry and inertial data are fused together to increase the accuracy [55,56]. The state vector is defined as $\mathbf{x} = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]$, with x, y, z and $\dot{x}, \dot{y}, \dot{z}$ representing respectively the global positions and velocity, and ϕ, θ, ψ and $\dot{\phi}, \dot{\theta}, \dot{\psi}$ the

attitude of the vessel. Considering the sensor readings, the estimation process satisfies the following equations:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \tag{14}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} {\mathbf{F}_k}^T + \mathbf{Q}_k \tag{15}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} {\mathbf{H}_k}^T (\mathbf{H}_k \mathbf{P}_{k|k-1} {\mathbf{H}_k}^T + \mathbf{R}_k)^{-1} \tag{16}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \tag{17}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \tag{18}$$

where k represents a discrete time instant, $\mathbf{F}_k$ is a kinematic constant velocity model, $\mathbf{H}_k$ is the observation model, $\mathbf{z}_k$ is the measurements vector, $\mathbf{I}$ is an identity matrix, $\mathbf{Q}_k$ is the process covariance matrix and $\mathbf{R}_k$ is the measurement covariance matrix.

The working principle of the EKF in this case is detailed below:

- the filter estimates the USV's pose at 50Hz and its encoding is saved in an hash table using the time stamp as key;
- when the UAV loses the track, the hash table is accessed and the last record inserted (the most recent estimate produced by the filter) together with the one having as key the time stamp of the last recorded observation are retrieved;
- the deck's current position with reference to the old one is calculated using geometric relationship;
- the controller command are updated including the new relative position;

The procedure described above is iterated until the UAV is redirected above the visual marker and can perceive it through its bottom camera.

2.5. Methodology

Algorithm 1 Landing Algorithm

```
1:  while not landed do
2:      last_known_pose =NULL
3:      if marker_visible then
4:          last_known_pose ← detect_landing_marker()
5:          if last_known_pose < user_defined_threshold then
6:              controller.send_commands(land)
7:              landed ← true
8:          end if
9:      else
10:         usv_pose ← ekf.estimates_pose()
11:         last_known_pose ← last_known_pose + usv_pose
12:     end if
13:     trajectory ← calculate_trj(last_known_pose)
14:     attitude_cmd ← controller.calculate_cmd(trajectory)
15:     controller.send_commands(attutude_cmd)
16: end while
```

The following section explains how the algorithm 1 works. The code is publicly available on our repository [2].

The quad-copter flies using its fixed non-tilting frontal camera, approaching the landing site on the USV's deck identified only by a fiducial marker. This, which scope is to outline the landing area, has to be perceived during all the landing manoeuvre. This is a requirement for precise landing despite the state estimator can compensate interruption in observation. When a visual marker is detected, the image processing library computes the 6-DOF relative-pose between the marker itself and the UAV. The result is used to make the quad-copter approaching the marker with the right orientation. To obtain this result, a damped spring controller reduces the error on the $x-$, $y-$ and $z-$axis and on the quad-copter's *yaw*. On attaining close proximity to the marker, the marker leaves the field of view of

[2] Github repository: https://github.com/pulver22/ardrone_tf_controller

the frontal camera. This is due to hardware limitation of fixed non-tilting cameras. To overcome this problem, the video stream from the frontal camera is interrupted and acquired from the one located under the UAV and downward-looking. The quad-copter continues the landing manoeuvre keeping the marker at the centre of the second camera's FOV. Otherwise, a compensatory behaviour is adopted: the EKF estimates the actual position of the USV and the drone is redirected close to it while increasing its altitude. Increasing the altitude allows to enlarge the field of view of the bottom camera, that is quite limited. In this way, it is guaranteed that the marker will be soon perceived and centred by the aerial vehicle. When an experimentally defined distance from the marker is reached, the drone lands safely. This distance depends on the side length of the marker used. In fact, with a smaller marker it would be possible to decrease this value but it would become impossible to perceive the marker at longer distance. We found that a marker side length of approximately 0.30 meters represents a good trade-off for making the marker visible at long and close distance at the same time. As a consequence, we decide to use 0.75 meters as distance for starting the touchdown phase of the descending manoeuvre, during which the power of the motors is progressively reduced until complete shut-down. The use of visual markers allows the estimation of the full 6-DOF pose information of the aerial and surface vehicles. In this way, landing operations in rough sea condition with a significant pitching and rolling deck can still be addressed.

3. Results and Discussion

All the experiments has been conducted inside a simulated environment built on Gazebo 2.2.X and offering a 3D model of the AR Drone 2.0. To the scope of this work, the existing simulator has been partially rewritten and extended to support multiple different robots at the same time. The Kingfisher USV, produced by Clearpath Robotics, has been used as floating base. It is a small catamaran with a dimension of are 135 x 98 cm, that can be deployed in a autonomous or tele-operated way. It is equipped with a flat plane representing a versatile deck for UAVs of small dimension. On this surface a square visual marker is placed. Previous research demonstrated a linear relationship is existing between the side length of the marker and its observability. Therefore, we opted for a side length of 0.3 meters that represents a good compromise, making the marker visible in the range [0.5, 6.5] meters.

The algorithm has been tested under multiple conditions, namely three. In the first scenario, the USV is subjected only to a rolling movement while floating in the same position for all the length of the experiment; in the second scenario, the USV is subjected only to a pitching movement; while in the last scenario the USV is subject to both rolling and pitching disturbances at the same time. Fig. 4 illustrates the rotation angles around their corresponding axis. In all the simulations, the disturbances are modelled as a signal having a maximum amplitude of 5 degrees and a frequency of 0.2 Hz. Rolling and pitching of a vessel generate upward and downward acceleration forces directed tangentially to the direction of rotation, which cause linear motion knowns as swaying and surging along the transverse or longitudinal axis respectively [57].

3.1. Rolling Platform

In this subsection, the results of a landing manoeuvre on a rolling floating base are reported. In particular, Fig. 5 illustrates the UAV and the USV's trajectory, respectively in blue and red, in the UAV's reference frame; while Fig. 6 and Fig. 7 show the controller commands and the salient moments of the manoeuvre respectively.

The marker has been successfully recognised at a distance of 3.74 meters in front of the UAV, and at 0.09 meter on its left. The displacement on the $z-$axis, used as reference for the altitude, was of 0.84 meter instead. The UAV, with the parameters reported in the previous table 1, has been able to complete the landing in 25 seconds.

The quad-copter approaches the landing base trying to keep it at the centre (in a range of $\pm$ 10 degrees) of its camera's FOV. In the case the marker leaves this interval of tolerance, the UAV would rotate around its $z-$axis in order to centre it again. The approach continues until the UAV's low

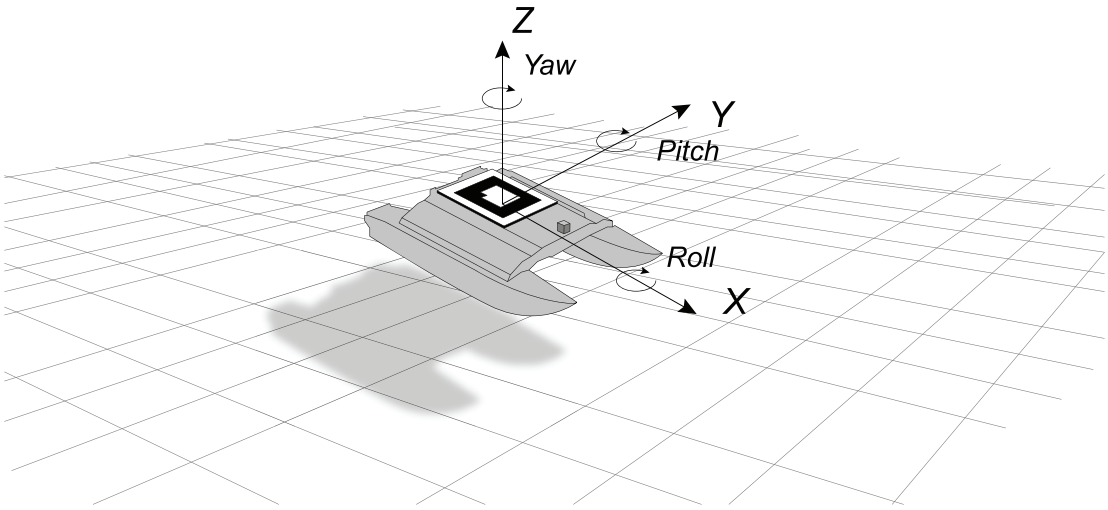

Figure 4. The movements around the vertical, longitudinal and lateral axis of the USV are called yaw, roll and pitch respectively.

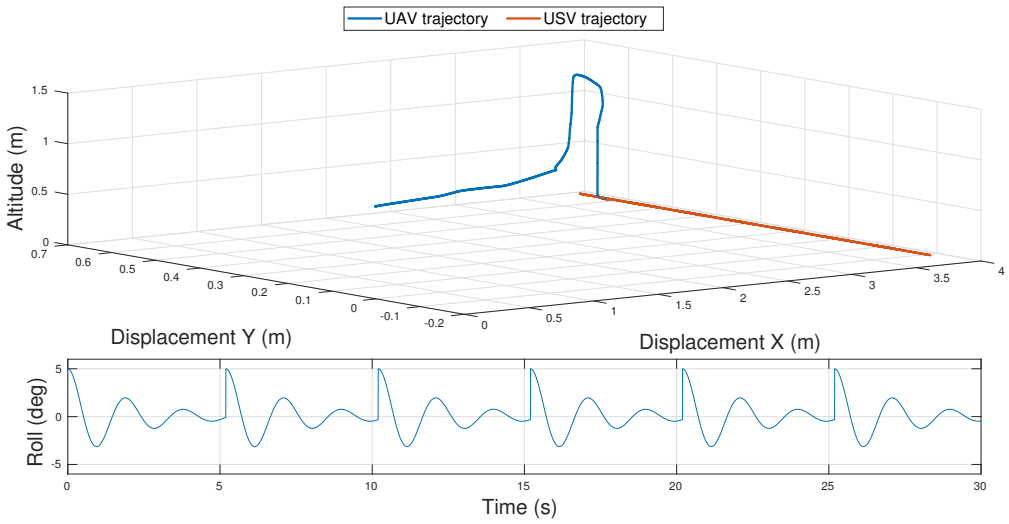

Figure 5. Above: The UAV and USV 3D trajectories, in blue and red respectively, in the UAV's reference frame. Bottom: The roll disturbances the USV is subject to.

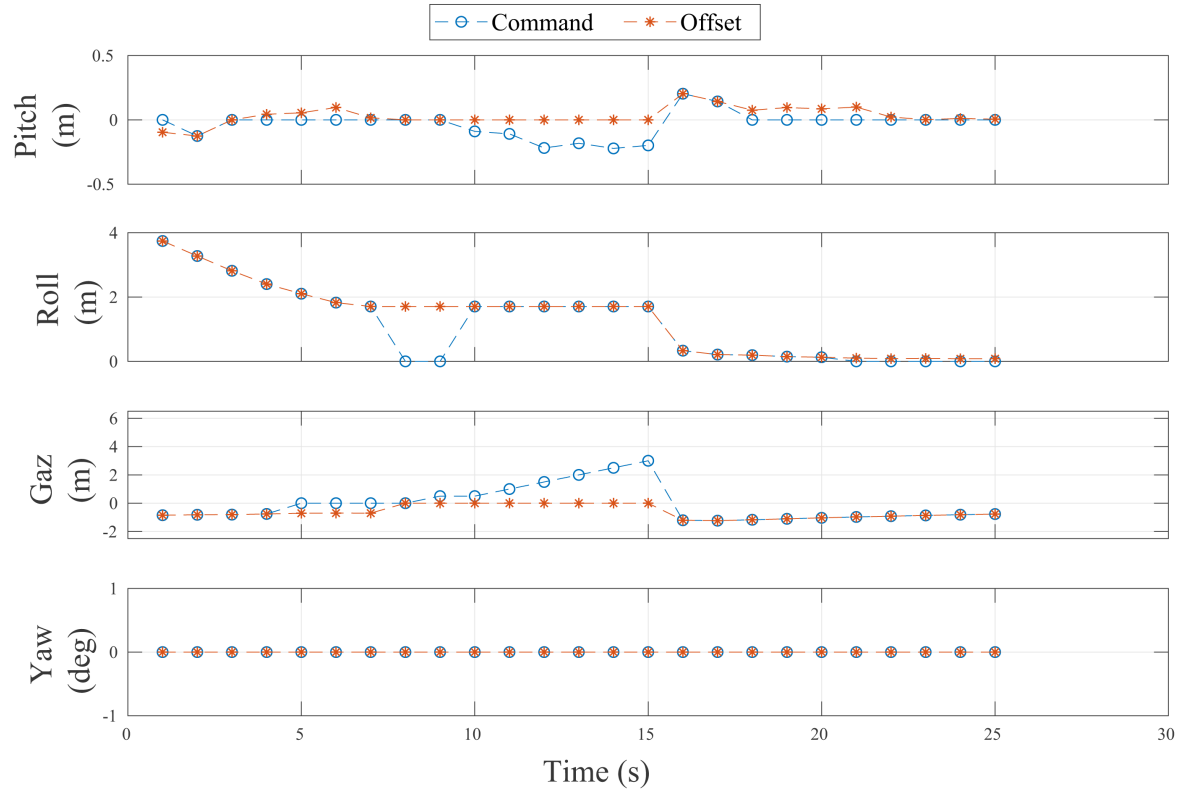

Figure 6. Controller commands and visual offsets in the experiment with a rolling landing platform.

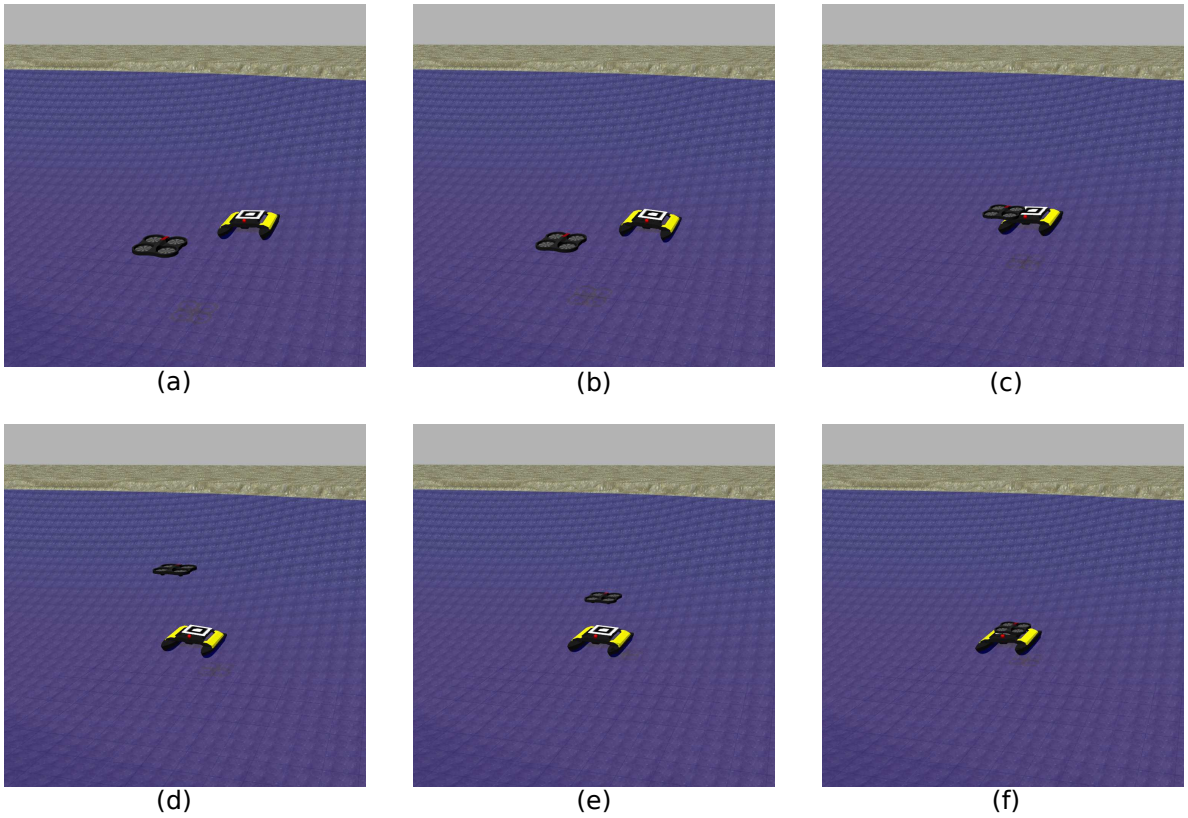

Figure 7. Landing manoeuvre of a VTOL UAV on a USV subject only to rolling disturbances.

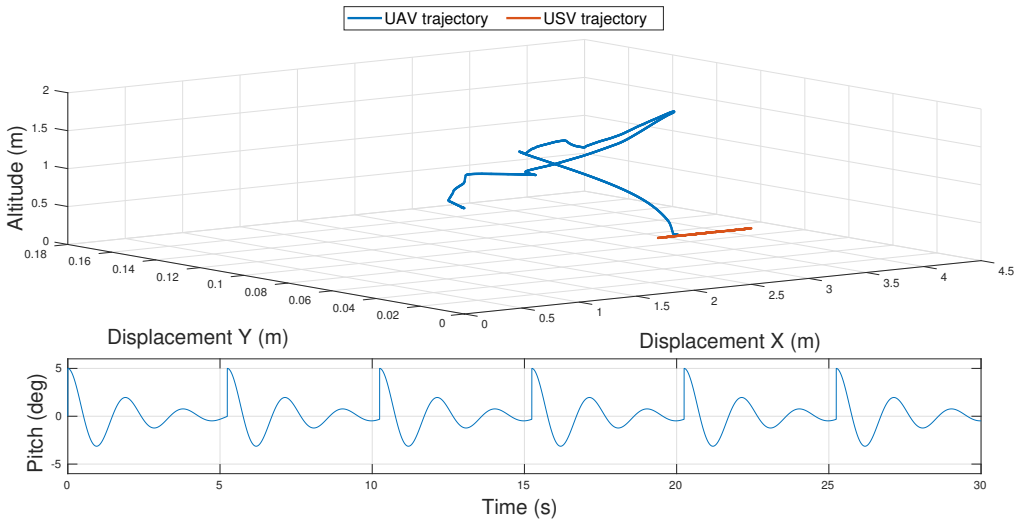

Figure 8. Above: The UAV and USV 3D trajectories, in blue and red respectively, in the UAV's reference frame. Bottom: The pitch disturbances the USV is subject to.

altitude prevents the marker to be seen from the frontal camera, as shown in Fig. 7-a ($t = 10s$). At this point, the video stream is switched from the frontal camera to the one located at the bottom of the quad-copter and looking down, and new commands are generated and sent. The UAV is instructed to move towards the last known position of the landing platform but increasing its altitude in order to enlarge the area covered by its bottom camera. At $t = 15s$, as represented in Fig. 7-b, the UAV is located exactly above the marker and it can now complete the landing phase: it descends while trying to keep the marker at the centre of its FOV, as shown in Fig. 7-c. Small velocity commands are sent on the leaning direction (x and y, respectively) in order to approach the final position with high accuracy. Finally, at $t = 25s$ the UAV reaches the minimum altitude required to shut-down its motors and land on the platform (Fig. 7-f).

The commands generated from the relative-pose between the UAV and the landing platform's frame are illustrated in Fig. 6. Here, the controller's commands are plotted against the perception from the camera. As it is possible to see in the figure, for most of the travel the two curves of the commands and of the observations overlap perfectly. When they do not, the marker is lost and the UAV actuates the compensatory behaviour: the estimation filter's output, namely the USV's predicted pose, is combined with the latest vision observation in order to generate new commands for the UAV. In this way it is possible to explain changing in roll, pitch and altitude in the graph. Since the UAV has the same yaw of the floating base, namely they have the same orientation along the z-axis, no rotation commands are issued for this degree of freedom.

Few words are reserved for the pitch's data between $t = 18s$ and $t = 22s$, and the gaz's ones between $t = 5s$ and $t = 8s$. In this case, the offsets are below a user-defined threshold and a null command is sent instead. The use of a threshold has been introduced for speeding up the landing phase: while testing the controller, it was noticed the UAV spent a lot of time while trying to align perfectly on the three axis with the centre of the landing plane, sometimes without any success. This has been identified as a limitation of controllers with fixed values parameters and a new more versatile solution is already planned as future work.

3.2. Pitching Platform

In this subsection an experiment with a pitching floating platform is reported. As before, the time for completing the landing manoeuvre is not considered as key-factor but the attention is on the ability of the UAV to approach and land on the USV with high precision.

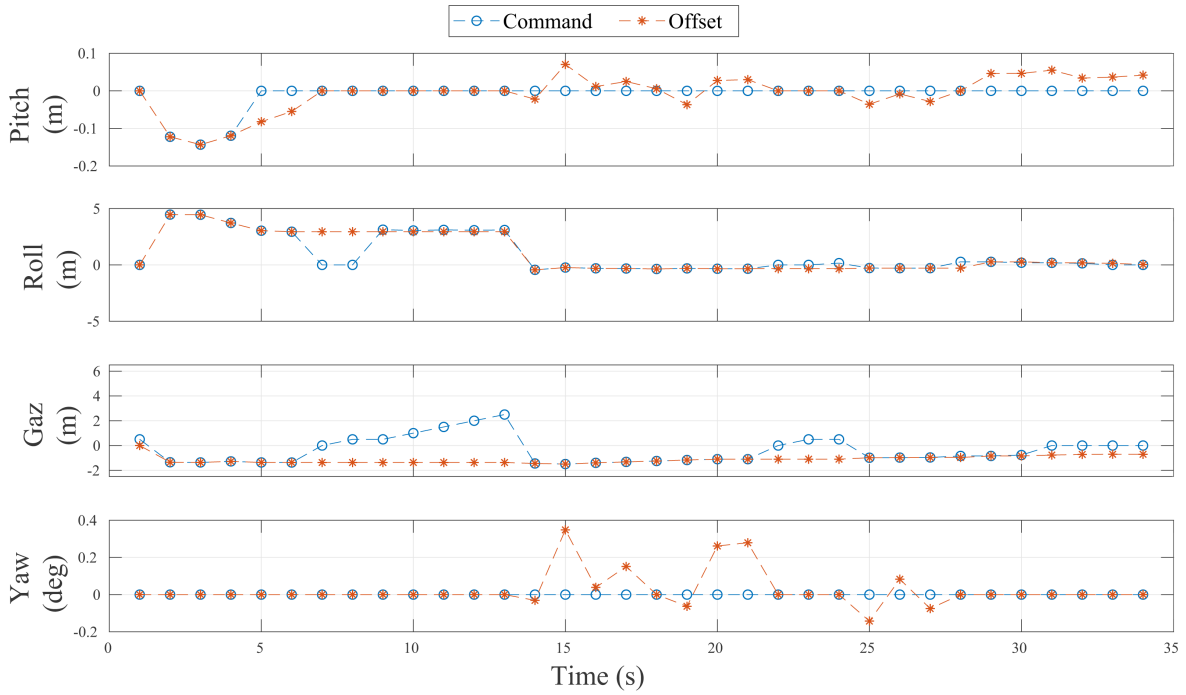

Figure 9. Controller commands and visual offsets in the experiment with a pitching landing platform.

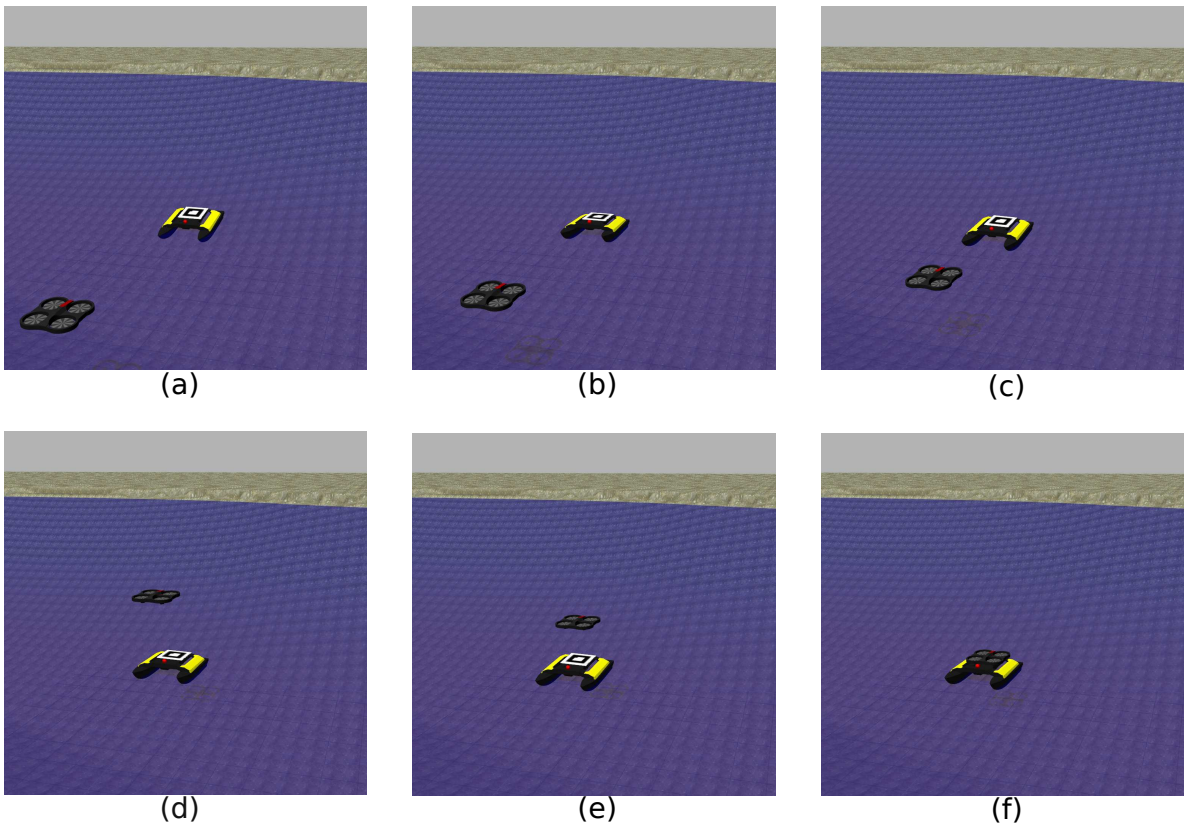

Figure 10. Landing manoeuvre of a VTOL UAV on a USV subject only to pitching disturbances.

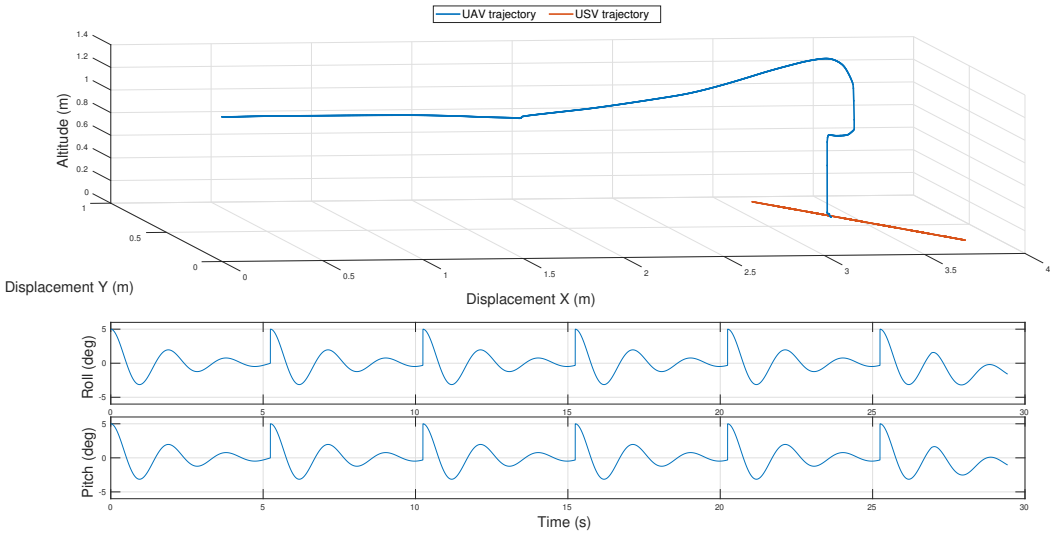

Figure 11. Above: The UAV and USV 3D trajectories, in blue and red respectively, in the UAV's reference frame. Bottom: Both the roll and pitch disturbances the USV is subject to.

As in the previous experiments, the two vehicles 3D trajectory are reported in Fig. 8 in the UAV's reference frame, the controller commands in Fig. 9 and example frames in Fig. 10. The quad-copter, with the same controller parameters of before, was able to follow and land on the visual marker in almost 34 seconds after identifying it 4.46 meters ahead and 0.12 meter on its left.

As in the case of a rolling base, Fig. 10-a shows the UAV starts moving in order to keep the visual marker at the centre of its frontal camera's field of view. This is what happens at time $t = 26s$ and shown in Fig. 10-b. At $t = 6s$ the UAV reaches its minimum altitude and it is now impossible for it to see the visual marker, as illustrated in Fig. 10-c. At this point, the video stream starts to be acquired from the bottom camera and the USV's estimated position is sent to the controller. At the same time, instructing the UAV to increase its altitude to augment the total area covered with its downward-looking camera. Doing this, at $t = 13s$ the UAV is located exactly above the USV. The landing base is at centre of the camera's FOV, therefore a null velocity command is sent to stop the USV. Fig. 10-e and 10-f show the UAV can then descend slowly to centre the marker properly and, in the end, land on it.

Further analysis can be done with the results reported in Fig. 9. In the same way of the experiment with a rolling deck, the curve of the controller's commands and the one related to the offsets overlap for most of the time. All the considerations made before still hold: while the marker is lost, the EKF is able to estimate the landing platform's current pose with reference to the instant of time when the marker has been lost. This relative-pose is added to the last observation in order to produce a new command.

This is what is possible to see in the plot between $t = 21s$ and $t = 25s$. Here, the two curves differ: while all the offsets remain constant because no new marker observations have been done by the UAV, the commands (gaz and roll) slightly change. The plot is now discussed in more details. While the yaw and the pitch commands remain identical to 0 because the UAV is already aligned with the landing base (within the predefined bounds), the UAV's roll command is changed including at every instant the new relative-pose (changing on the longitudinal direction) of the USV.

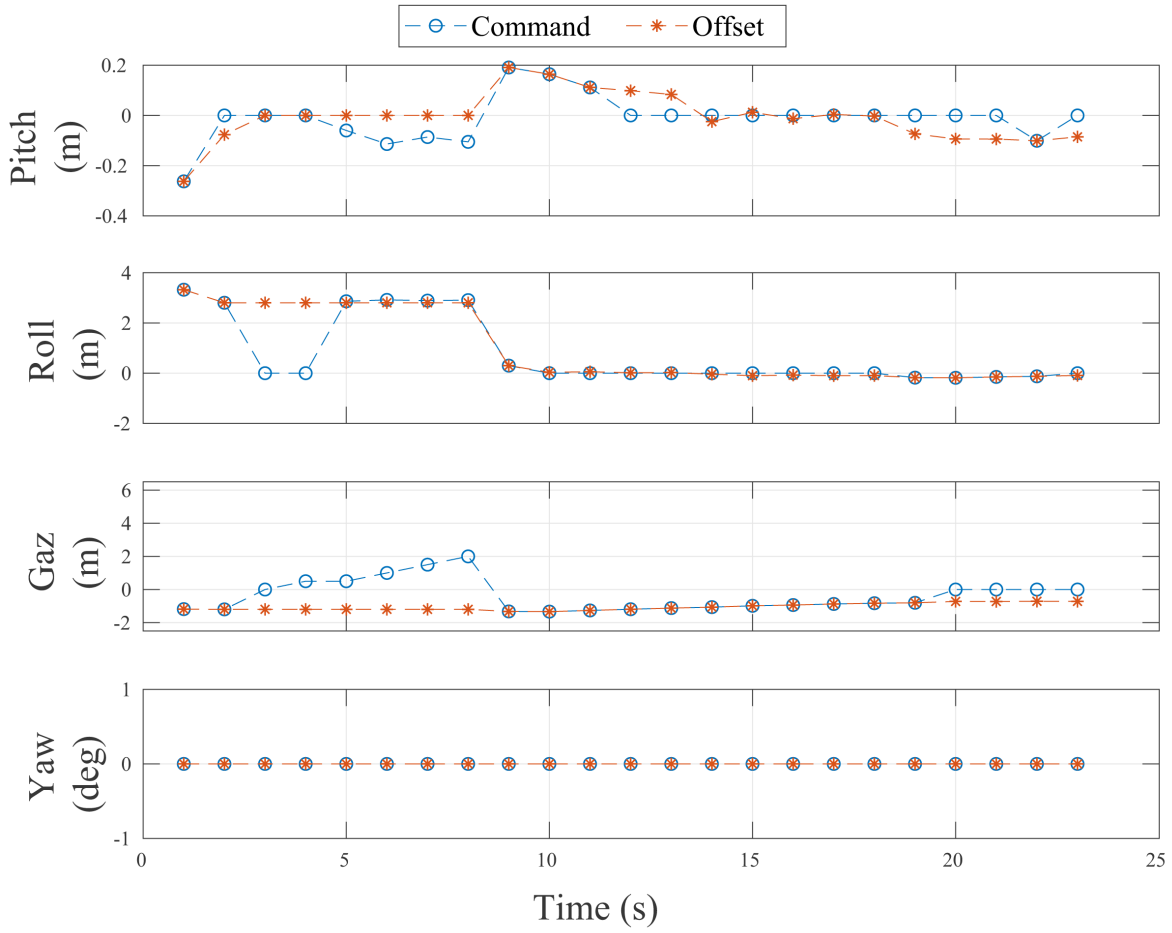

Figure 12. Controller commands and visual offsets in the experiment with a pitching and rolling landing platform, in order to simulate complex marine scenarios.

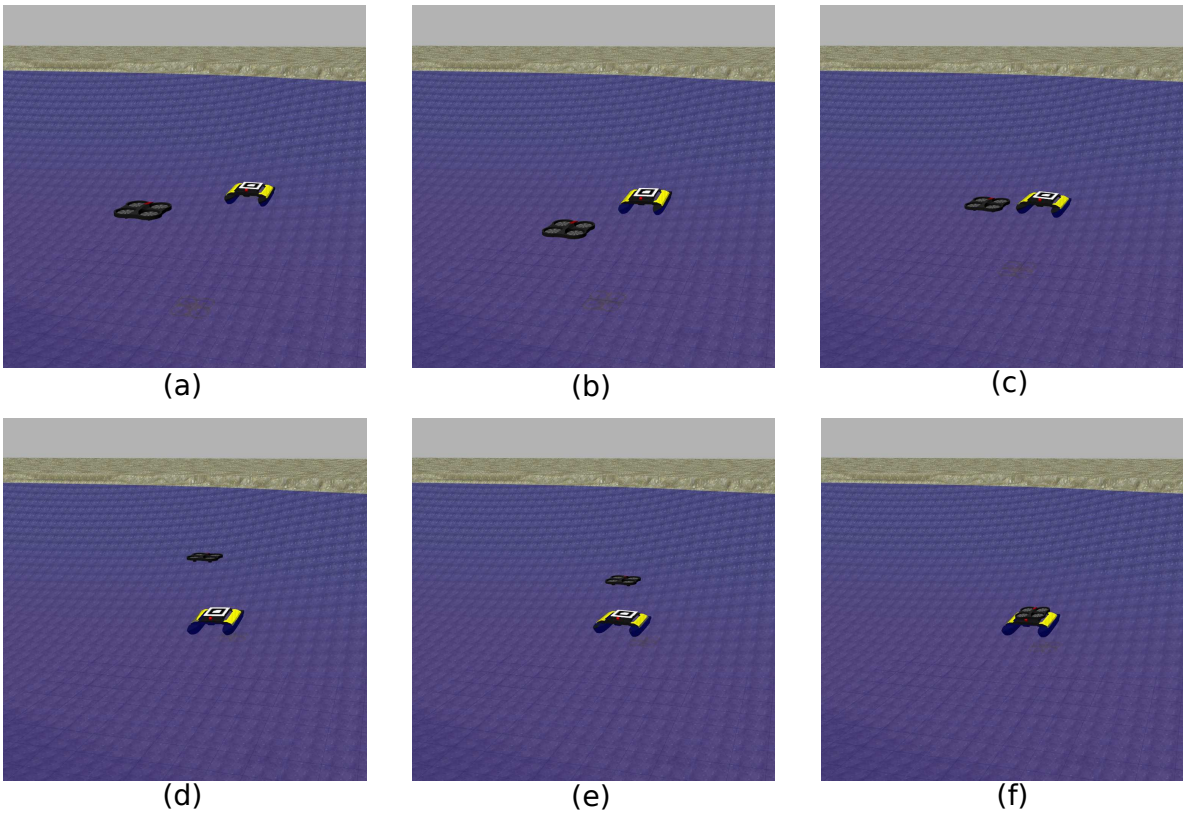

Figure 13. Landing manoeuvre of a VTOL UAV on a USV subject to both rolling and pitching disturbances, in order to simulate complex marine scenarios.

3.3. Rolling and Pitching Platform

A last simulation has been done with a floating platform that is subject to both rolling and pitching stresses. The goal of this experiment is to test the developed landing algorithm against simulated harsh marine conditions.
The results are reported in Fig. 11, showing the both vehicles trajectories along a 23 seconds operation. The UAV successfully accomplish the landing maneovre starting from an initial marker's identification 3.71 meters in front of it and 0.30 meters on its left. Fig. 12 shows the comparison between the offsets obtained through the vision algorithm and the commands sent to the controller. It is possible to see that, as in the previous experiments, the curve of the offsets and the one related to the commands mainly overlap. All the analysis made before are still valid, but it is interesting to notice how the framework proposed is able to react properly also when the landing platform is subject to complex disturbances. The salient moments of the flight are illustrated in Fig. 13

4. Conclusion and Future Directions

In this paper, a solution to make an unmanned aerial vehicle to autonomously land on the deck of a USV is presented. It resides only on the UAV's on-board sensors and on the adoption of visual marker on the landing platform. In this way, the UAV can estimate the 6-DOF landing area position through an image processing algorithm. The adoption of a pose estimation filter - in this case an extended Kalman filter - allows to overcome issues with fixed non-tilting cameras and the image processing algorithm. Not involving GPS signals in the pose estimation and in the generation of flight commands, allows the UAV to land also in situations where this signal is not available (indoor scenario or adverse weather conditions).

The validation of the approach has been done in simulation with a quad-rotor and an unmanned surface vehicle as platform on which to land. Three different experiments were performed, each of them with a different type of disturbance acting on the landing base. In all scenarios successful results were obtained.

The future research is twofold. From a practical point of view, the proposed approach needs to be tested in a real environment with an unmanned surface vehicle in order to test its robustness against real wind and sea currents. The second aspect is more related to the identified limitation of the algorithm itself. Therefore it is suggested to develop an adaptive controller, possibly based on intelligent solution such as artificial neural networks or fuzzy logic, where the gain of the controller change depending on the distance to the landing base.

Author Contributions: Riccardo Polvara wrote the paper, developed the software, and conducted all the experiments. Sanjay Sharma and Jian Wan provided valuable feedback and revisions. Andrew Manning and Robert Sutton contributed in writing and proofreading the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

1. Kumar, V.; Michael, N. Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research* **2012**, *31*, 1279–1291.
2. Shim, D.; Chung, H.; Kim, H.J.; Sastry, S. Autonomous exploration in unknown urban environments for unmanned aerial vehicles. Proc. AIAA GN&C Conference, 2005, pp. 1–8.
3. Bourgault, F.; Göktogan, A.; Furukawa, T.; Durrant-Whyte, H.F. Coordinated search for a lost target in a Bayesian world. *Advanced Robotics* **2004**, *18*, 979–1000, [http://dx.doi.org/10.1163/1568553042674707].
4. Neumann, P.P.; Bennetts, V.H.; Lilienthal, A.J.; Bartholmai, M.; Schiller, J.H. Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms. *Advanced Robotics* **2013**, *27*, 725–738, [http://dx.doi.org/10.1080/01691864.2013.779052].
5. Murphy, D.W.; Cycon, J. Applications for mini VTOL UAV for law enforcement. Enabling technologies for law enforcement and security. International Society for Optics and Photonics, 1999, pp. 35–43.
6. Colorado, J.; Perez, M.; Mondragon, I.; Mendez, D.; Parra, C.; Devia, C.; Martinez-Moritz, J.; Neira, L. An integrated aerial system for landmine detection: SDR-based Ground Penetrating Radar onboard an autonomous drone. *Advanced Robotics* **2017**, *31*, 791–808, [http://dx.doi.org/10.1080/01691864.2017.1351393].
7. Tomic, T.; Schmid, K.; Lutz, P.; Domel, A.; Kassecker, M.; Mair, E.; Grixa, I.L.; Ruess, F.; Suppa, M.; Burschka, D. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *IEEE robotics & automation magazine* **2012**, *19*, 46–56.
8. Kruijff, G.; Kruijff-Korbayová, I.; Keshavdas, S.; Larochelle, B.; Janíček, M.; Colas, F.; Liu, M.; Pomerleau, F.; Siegwart, R.; Neerincx, M.; Looije, R.; Smets, N.; Mioch, T.; van Diggelen, J.; Pirri, F.; Gianni, M.; Ferri, F.; Menna, M.; Worst, R.; Linder, T.; Tretyakov, V.; Surmann, H.; Svoboda, T.; Reinštein, M.; Zimmermann, K.; Petříček, T.; Hlaváč, V. Designing, developing, and deploying systems to support human–robot teams in disaster response. *Advanced Robotics* **2014**, *28*, 1547–1570, [http://dx.doi.org/10.1080/01691864.2014.985335].
9. Minaeian, S.; Liu, J.; Son, Y.J. Vision-based target detection and localization via a team of cooperative UAV and UGVs. *IEEE Transactions on systems, man, and cybernetics: systems* **2016**, *46*, 1005–1016.
10. Kim, S.J.; Jeong, Y.; Park, S.; Ryu, K.; Oh, G. A Survey of Drone use for Entertainment and AVR (Augmented and Virtual Reality). In *Augmented Reality and Virtual Reality*; Springer, 2018; pp. 339–352.
11. Polvara, R.; Sharma, S.; Sutton, R.; Wan, J.; Manning, A. Toward a Multi-agent System for Marine Observation. In *Advances in Cooperative Robotics*; World Scientific, 2017; pp. 225–232.
12. Murphy, R.R.; Steimle, E.; Griffin, C.; Cullins, C.; Hall, M.; Pratt, K. Cooperative use of unmanned sea surface and micro aerial vehicles at Hurricane Wilma. *Journal of Field Robotics* **2008**, *25*, 164–180.
13. Pereira, E.; Bencatel, R.; Correia, J.; Félix, L.; Gonçalves, G.; Morgado, J.; Sousa, J. Unmanned air vehicles for coastal and environmental research. *Journal of Coastal Research* **2009**, pp. 1557–1561.

14. Pinto, E.; Santana, P.; Barata, J. On collaborative aerial and surface robots for environmental monitoring of water bodies. Doctoral Conference on Computing, Electrical and Industrial Systems. Springer, 2013, pp. 183–191.
15. Linchant, J.; Lisein, J.; Semeki, J.; Lejeune, P.; Vermeulen, C. Are unmanned aircraft systems (UASs) the future of wildlife monitoring? A review of accomplishments and challenges. *Mammal Review* **2015**, *45*, 239–252.
16. Watts, A.C.; Ambrosia, V.G.; Hinkley, E.A. Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use. *Remote Sensing* **2012**, *4*, 1671–1692.
17. Polvara, R.; Sharma, S.; Wan, J.; Manning, A.; Sutton, R. Obstacle Avoidance Approaches for Autonomous Navigation of Unmanned Surface Vehicles. *Journal of Navigation* **2017**, p. 1–16.
18. Stacy, N.; Craig, D.; Staromlynska, J.; Smith, R. The Global Hawk UAV Australian deployment: imaging radar sensor modifications and employment for maritime surveillance. Geoscience and Remote Sensing Symposium, 2002. IGARSS'02. 2002 IEEE International. IEEE, 2002, Vol. 2, pp. 699–701.
19. Ettinger, S.M.; Nechyba, M.C.; Ifju, P.G.; Waszak, M. Vision-guided flight stability and control for micro air vehicles. *Advanced Robotics* **2003**, *17*, 617–640, [http://dx.doi.org/10.1163/156855303769156983].
20. Polvara, R.; Sharma, S.; Wan, J.; Manning, A.; Sutton, R. Towards Autonomous Landing on a Moving Vessel through Fiducial Markers. IEEE European Conference on Mobile Robotics (ECMR). IEEE, 2017, pp. 1–6.
21. Kong, W.; Zhou, D.; Zhang, D.; Zhang, J. Vision-based autonomous landing system for unmanned aerial vehicle: A survey. Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on. IEEE, 2014, pp. 1–8.
22. Kendoul, F. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics* **2012**, *29*, 315–378.
23. Sanz, D.; Valente, J.; del Cerro, J.; Colorado, J.; Barrientos, A. Safe operation of mini UAVs: a review of regulation and best practices. *Advanced Robotics* **2015**, *29*, 1221–1233, [http://dx.doi.org/10.1080/01691864.2015.1051111].
24. Masselli, A.; Yang, S.; Wenzel, K.E.; Zell, A. A cross-platform comparison of visual marker based approaches for autonomous flight of quadrocopters. 2013 International Conference on Unmanned Aircraft Systems (ICUAS), 2013, pp. 685–693.
25. Wenzel, K.E.; Rosset, P.; Zell, A. Low-cost visual tracking of a landing place and hovering flight control with a microcontroller. Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009. Springer, 2009, pp. 297–311.
26. Cesetti, A.; Frontoni, E.; Mancini, A.; Zingaretti, P.; Longhi, S. A vision-based guidance system for UAV navigation and safe landing using natural landmarks. Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009. Springer, 2009, pp. 233–257.
27. Yang, S.; Scherer, S.A.; Zell, A. An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *Journal of Intelligent & Robotic Systems* **2013**, pp. 1–17.
28. Saripalli, S.; Montgomery, J.F.; Sukhatme, G.S. Visually guided landing of an unmanned aerial vehicle. *IEEE transactions on robotics and automation* **2003**, *19*, 371–380.
29. Hrabar, S.; Sukhatme, G.S. Omnidirectional vision for an autonomous helicopter. Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on. IEEE, 2003, Vol. 1, pp. 558–563.
30. Garcia-Pardo, P.J.; Sukhatme, G.S.; Montgomery, J.F. Towards vision-based safe landing for an autonomous helicopter. *Robotics and Autonomous Systems* **2002**, *38*, 19–29.
31. Scherer, S.; Chamberlain, L.; Singh, S. Autonomous landing at unprepared sites by a full-scale helicopter. *Robotics and Autonomous Systems* **2012**, *60*, 1545–1562.
32. Barber, B.; McLain, T.; Edwards, B. Vision-based landing of fixed-wing miniature air vehicles. *Journal of Aerospace computing, Information, and Communication* **2009**, *6*, 207–226.
33. Chahl, J.S.; Srinivasan, M.V.; Zhang, S.W. Landing strategies in honeybees and applications to uninhabited airborne vehicles. *The International Journal of Robotics Research* **2004**, *23*, 101–110.
34. Ruffier, F.; Franceschini, N. Optic flow regulation in unsteady environments: a tethered mav achieves terrain following and targeted landing over a moving platform. *Journal of Intelligent & Robotic Systems* **2015**, *79*, 275–293.

35. Herissé, B.; Hamel, T.; Mahony, R.; Russotto, F.X. Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow. *IEEE Transactions on Robotics* **2012**, *28*, 77–89.
36. Shakernia, O.; Vidal, R.; Sharp, C.S.; Ma, Y.; Sastry, S. Multiple view motion estimation and control for landing an unmanned aerial vehicle. Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on. IEEE, 2002, Vol. 3, pp. 2793–2798.
37. Saripalli, S.; Montgomery, J.F.; Sukhatme, G.S. Vision-based autonomous landing of an unmanned aerial vehicle. Robotics and automation, 2002. Proceedings. ICRA'02. IEEE international conference on. IEEE, 2002, Vol. 3, pp. 2799–2804.
38. Kim, J.; Jung, Y.; Lee, D.; Shim, D.H. Outdoor autonomous landing on a moving platform for quadrotors using an omnidirectional camera. Unmanned Aircraft Systems (ICUAS), 2014 International Conference on. IEEE, 2014, pp. 1243–1252.
39. Wenzel, K.E.; Masselli, A.; Zell, A. Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle. *Journal of intelligent & robotic systems* **2011**, *61*, 221–238.
40. Gomez-Balderas, J.E.; Flores, G.; Carrillo, L.G.; Lozano, R. Tracking a ground moving target with a quadrotor using switching control. *Journal of Intelligent & Robotic Systems* **2013**, *70*, 65–78.
41. Yakimenko, O.A.; Kaminer, I.I.; Lentz, W.J.; Ghyzel, P. Unmanned aircraft navigation for shipboard landing using infrared vision. *IEEE Transactions on Aerospace and Electronic Systems* **2002**, *38*, 1181–1200.
42. Belkhouche, F. Autonomous Navigation of an Unmanned Air Vehicle Towards a Moving Ship. *Advanced Robotics* **2008**, *22*, 361–379, [http://dx.doi.org/10.1163/156855308X292628].
43. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: an open-source Robot Operating System. ICRA workshop on open source software. Kobe, Japan, 2009, Vol. 3, p. 5.
44. Engel, J.; Sturm, J.; Cremers, D. Accurate figure flying with a quadrocopter using onboard visual and inertial sensing. *Imu* **2012**, *320*, 240.
45. Engel, J.; Sturm, J.; Cremers, D. Camera-based navigation of a low-cost quadrocopter. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 2815–2821.
46. Engel, J.; Sturm, J.; Cremers, D. Scale-Aware Navigation of a Low-Cost Quadrocopter with a Monocular Camera. *Robotics and Autonomous Systems (RAS)* **2014**, *62*, 1646—1656.
47. Nonami, K.; Kendoul, F.; Suzuki, S.; Wang, W.; Nakazawa, D. *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*, 1st ed.; Springer Publishing Company, Incorporated, 2010.
48. Goldstein, H. *Classical mechanics*; World student series, Addison-Wesley: Reading (Mass.), Menlo Park (Calif.), Amsterdam, 1980.
49. Azuma, R.T. A survey of augmented reality. *Presence: Teleoperators and virtual environments* **1997**, *6*, 355–385.
50. Dryanovsk, I.; Morris, B.; Duonteil, G. ar_pose. http://wiki.ros.org/ar_pose, 2010.
51. Kato, H.; Billinghurst, M. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on. IEEE, 1999, pp. 85–94.
52. Abawi, D.F.; Bienwald, J.; Dorner, R. Accuracy in optical tracking with fiducial markers: An accuracy function for ARToolKit. Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2004, pp. 260–261.
53. Hartley, R.I.; Zisserman, A. *Multiple View Geometry in Computer Vision*, second ed.; Cambridge University Press, ISBN: 0521540518, 2004.
54. Moore, T.; Stouch, D. A generalized extended kalman filter implementation for the robot operating system. In *Intelligent Autonomous Systems 13*; Springer, 2016; pp. 335–348.
55. Jetto, L.; Longhi, S.; Venturini, G. Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots. *IEEE Transactions on Robotics and Automation* **1999**, *15*, 219–229.
56. Chenavier, F.; Crowley, J.L. Position estimation for a mobile robot using vision and odometry. Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on. IEEE, 1992, pp. 2588–2593.
57. Handbook, C. Cargo loss prevention information from German marine insurers. *GDV, Berlin* **2003**.