

An Approach Toward Visual Autonomous Ship Board Landing of a VTOL UAV

Jose Luis Sanchez-Lopez · Jesus Pestana · Srikanth Saripalli · Pascual Campoy

Received: date / Accepted: date

Abstract We present the design and implementation of a vision based autonomous landing algorithm using a downward looking camera. To demonstrate the efficacy of our algorithms we emulate the dynamics of the ship-deck, for various sea states and different ships using a six degrees of freedom motion platform. We then present the design and implementation of our robust computer vision system to measure the pose of the shipdeck with respect to the vehicle. A Kalman filter is used in conjunction with our vision system to ensure the robustness of the estimates. We demonstrate the accuracy and robustness of our system to occlusions, variation in intensity, etc. using our testbed.

Keywords UAV · VTOL · Autonomous Landing · Computer Vision · Ship Deck Simulation

1 Introduction

In recent years, considerable resources have been devoted to the design, development and operation of Unmanned Aerial Vehicles (UAVs). The applications of such UAVs are diverse, ranging from scientific exploration and data collection, to provision of commercial services, military reconnaissance, and intelligence gathering. Other areas include law enforcement, search and rescue, and even entertainment. UAVs, particularly ones with vertical take-off and landing capabilities (VTOL), enable difficult tasks without endangering the life of human pilots. This potentially results in cost and size savings as well as increased operational

Jose Luis Sanchez-Lopez · Jesus Pestana · Pascual Campoy
Computer Vision Group
C.A.R. CSIC-Universidad Politecnica de Madrid
Calle Jose Gutierrez Abascal, 2
28006 - Madrid (Spain)
E-mail: jl.sanchez@upm.es

Srikanth Saripalli
ASTRIL
SESE - Arizona State University
ISTB4, 781 E Terrace Road
85287 - Tempe, AZ (USA)
E-mail: srikanth.saripalli@asu.edu

capabilities and performance. Currently the capabilities of such UAVs are limited. A helicopter is a compact VTOL capable platform extremely manoeuvrable.

The autonomous landing of VTOL UAVs is a very important capability for autonomous systems, and is useful for various tasks as search and rescue, law enforcement, and military scenarios. Our challenge is to provide the UAVs with the capability of autonomously land on ship deck platforms in extreme weather conditions.

Autonomous landing on a fixed platform has been studied since the emergence of Unmanned Aerial Vehicles (UAVs) and some existing solutions are provided [1], [2].

Various approaches to the problem of landing on moving targets have been studied. Some of them are on two degree of freedom platforms [3] or a simple moving platform [4], [5].

However, autonomously landing on a ship deck platform continues to be studied, and has only recently been solved for very favourable weather conditions [6], [7], [8], [9].

Computer Vision for landing has also been extensively studied. [1], [2], [3], [4], [5], [10] give a good overview and various applications of vision but none of them focus on the problem of landing an UAV on a 6 degree of freedom (DoF) moving platform. Various authors have focussed on the landing problem using special markers or helipad¹ structures [11], [12], [13].

The movement of a ship at Sea is due to the effect of the wave motion. The typical environmental conditions attributed to waves are grouped into several Sea States [8] (see table 1).

Sea State	Description	Significant Wave Height (m)
0	Calm (glassy)	0.0
1	Calm (ripples)	0.0 - 0.1
2	Smooth (wavelets)	0.1 - 0.5
3	Slight	0.5 - 1.25
4	Moderate	1.25 - 2.5
5	Rough	2.4 - 4.0
6	Very Rough	4.0 - 6.0
7	High	6.0 - 9.0
8	Very High	9.0 - 14.0
9	Phenomenal	Over 14

Table 1 Sea State Parameters. The Significant Wave Height is defined as the average value of the height (vertical distance between trough and crest) of the largest 1/3 of the waves present. The waves are modelled as sinusoidal function.

The ship can be modelled as a rigid body moving in the sea with six degrees of freedom, figure 1 (see [14]). Its movement in the sea depends on the Sea State, the physical parameters of the ship, and the wave direction.

Some authors use sinusoidal functions with a fixed amplitude and frequency only for heave movement [15]. Others use sinusoidal functions for every degree of freedom [9]. Finally, others define a different heave movement function [16]. None of these authors take into account the ship model or Sea State.

¹ In the paper, the word "heliport" is used to refer to the surface to land, and the word "helipad" to refer to the drawings painted on the heliport.

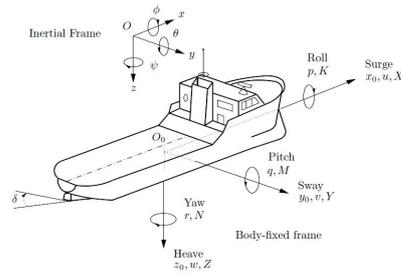


Fig. 1 Standard notation and sign conventions for ship motion description.

If we want to simulate the ship model, we can use physical models as the one used in games [17]; or control applications [14] or [7]. Unfortunately, these models are too complex, not realistic enough, and need additional ship parameters.

To consider the ship model in the simplest form, a register of sailing data [18] could be used to calculate the model [19]. With this approach, the Sea State and wave direction are ignored.

A better approach is the use of a simple physical model that consists of a sinusoidal function for each degree of freedom, whose parameters depend on the Sea state, wave direction, and of course the ship [8]. However, this approach is not random enough for our problem because each degree of freedom moves periodically.

The paper is organized as follows: in section "System Description" a system and equipment description is developed; the ship deck simulation is explained in section "Ship Deck Simulation"; section "Computer Vision System" describes how the computer vision system works; the state estimator is explained in section "State Estimation: Kalman Filter"; results of the whole system are described in section "Results"; finally, section "Conclusion and future work" concludes the paper.

2 System Description

The VTOL UAV proposed to be used in our study is a Rotomotion Inc, SR200 (figure 2). This helicopter is equipped with an autopilot, an inertial measure unit (IMU), a GPS sensor,



Fig. 2 Rotomotion SR 200 Gas powered Helicopter. Length: 2790 mm; Width: 760 mm; Height: 860 mm; Main Rotor Diameter: 3000 mm; Endurance: Up to 5 hours; and Maximum Payload: 22.7 kg

and a small onboard computer that simplifies the control task and is ideal for the development of autonomous capabilities for UAVs.

A white H surrounded by a white circle (figure 3) is painted on the heliport surface. These marks are the most extended marks to indicate the presence of a heliport surface.

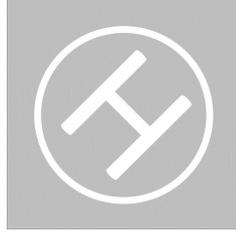


Fig. 3 Helipad Marks used in our application. They are typical marks.

To detect the heliport, and to measure its pose with respect to the helicopter's pose, we propose to use a single downward looking colour camera computer vision system (a single camera with three channels: RGB). We selected the single camera system instead of a stereo pair because we assume that the size of the square landing platform is known, and because stereo requires a very large baseline. As such, the 3-D reconstruction could be calculated using the platform model and the camera calibration parameters (see section "Computer Vision System"). The helicopter is also equipped with SONAR sensors that return the measurements of the distance to the floor when the helicopter is really close to it. These sensors allow us to detect the heliport pose in the very last stage of the landing when the helicopter is so near to the heliport that the computer vision system is not able to detect the marks.

3 Ship Deck Simulation

We simulate the movement of the ship deck on the Sea, using a Servos and Simulation Inc, Generic Motion System (model 710-6-500-220) with a $2.44 \times 2.44m^2$ gray surface as heliport (figure 4).



Fig. 4 Servos and Simulation Inc, 710-6-500-220 Generic Motion System. Number of axis: 6; Height: 48.6 cm; Floor Platform: $66 \times 68.6cm^2$; Power: 220 VAC @ 20 A; Payload: 226.8 kg; Max. Roll (x): $\pm 13^\circ$; Max. Pitch (y): $\pm 15^\circ$; Max. Yaw (z): $\pm 16^\circ$; Max. Surge (x): $\pm 10.2cm$; Max. Sway (y): $\pm 10.2cm$; and Max. Heave (z): $\pm 6.4cm$

Our approach for the ship deck motion simulation is an improvement of [8]. We propose a uniform random generation for the amplitude of each sinusoidal movement based on how the amplitude data corresponds to the top 1/10 waves. To obtain a continuous and derivable movement, we interpolate between two different sinusoidal function with a 5 degree polynomial.

Using MATLAB to achieve the ship simulation, we obtain, for a Sea State of 6, a Wave Direction of 60° , and a Oliver Hazard Perry Class FFG Frigate, the following plots (figures

5, 6 and 7). The shape of these plots looks similar to the available plots of ship movements in [18].

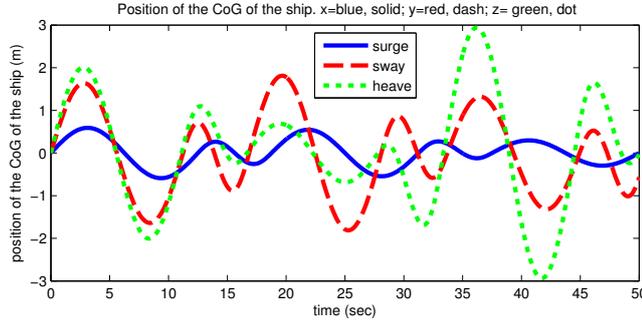


Fig. 5 Position of surge (x, blue), sway (y, red) and heave (z, green) of the simulated ship's Center of Gravity.

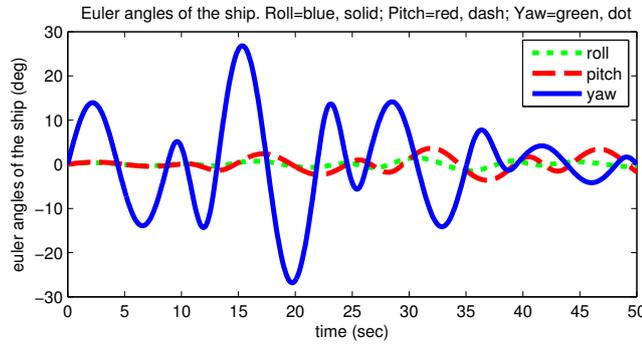


Fig. 6 Euler angles of the simulated ship: Roll (blue), Pitch (red) and Yaw (green).

Once the ship simulation is calculated, because of our platform motion is smaller than the real ship's (figure 4), and our helicopter is smaller than its manned counterpart (figure 2), the entire system has to be scaled down. The approach that we choose for the scaling consist on scale only the amplitudes of the movement of each DoF. We scale down the position DoF (x, y, and z) multiplying the simulated amplitude by the coefficient 1/90, and the angles (Yaw, Pitch, and Roll) by 1/3. While the scaling function is not realistic, it ensures that position and attitude are not being distorted and the platform is being used to the maximum extent possible.

The following step in the ship deck simulation is the calculation of the motor inputs of our platform through the Inverse Kinematics (figure 10) using the scaled ship simulation movement as the desired movement of our motion platform (figure 9). According to [20], we use the equation 1, defining a fixed reference system (attached to the bottom of the motion platform) and a mobile reference system (attached to the mobile part of the platform), to calculate the inverse kinematics:

$$L_i = \|O_r + O_{Rp} \cdot P_{bi} - O_{ai}\| \quad (1)$$

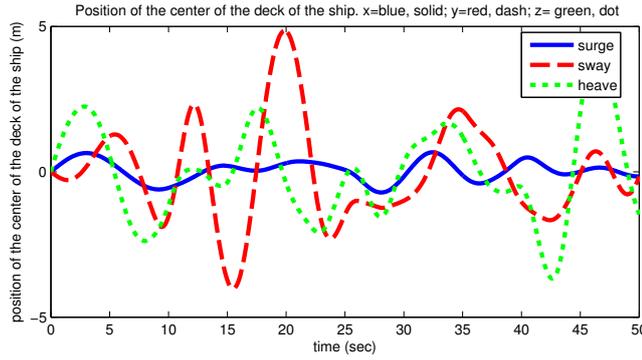


Fig. 7 Position of surge (x, blue), sway (y, red) and Heave (z, green) of the Center of the landing Deck of the simulated ship.

Where L_i is the longitude of the bar i of the motion platform; $O_r = [x, y, z]^t$ is the desired position of the mobile reference system respect to the fixed one; O_{Rp} is the 3-by-3 rotation matrix of the desired attitude of the mobile reference system respect to the fixed one; P_{bi} is the 3-by-1 vector of the position of the side of the bar i fixed to the mobile part of the platform, in coordinates of the mobile reference system; O_{ai} is the 3-by-1 vector of the position of the side of the bar i moved by the motor i , respect to the fixed reference system. O_{ai} depends on the motor input q_i that is the unknown of the equation; and $i = 1..6$ indicates the number of the bar of the motion platform (see figure 8).

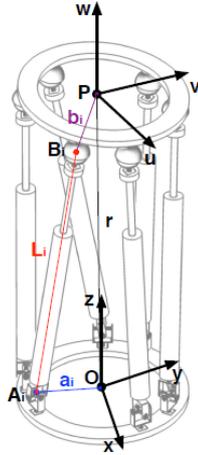


Fig. 8 Generic 6 DoF parallel robot.

When equation 1 has no solution inside the compatible values of q_i , a singular configuration is achieved. If that happens, we calculate the value that minimizes equation 1 that is the nearest achievable pose by the platform respect to the desired one.

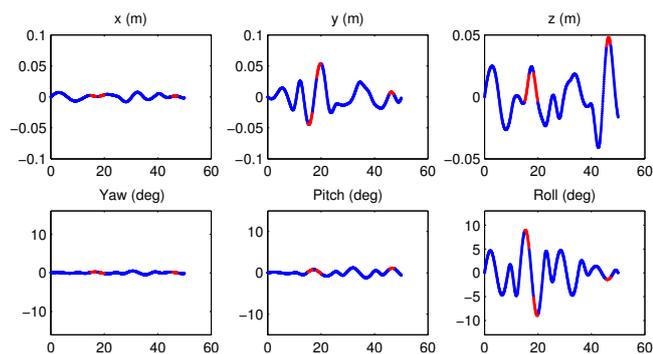


Fig. 9 Desired motion of our motion platform, for the ship simulation described in figures 5, 6 and 7. In red, points of singular configuration that are not achievable by our platform; in blue achievable points.

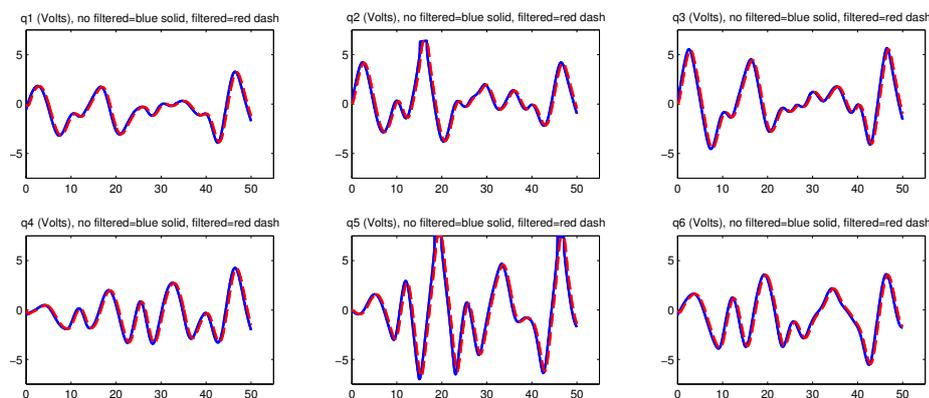


Fig. 10 Motor inputs (Volts). Blue solid line, motor inputs before filtering for the desired movement described in figure 9; red dash line, motor inputs after filtering.

The last step is the filtering of the calculated inputs in order to limit the speeds and accelerations because the inverse kinematics calculation does not take them into account (figure 10).

4 Computer Vision System

In order to measure the pose of the Landing Platform, we use a single downward looking camera computer vision system on board the helicopter as described in section "System Description".

As the helipad has no image descriptors (like SURF features), the detection and the tracking cannot be based on matching them with a previously known template. We have to use other features of the helipad, like the color or the marks (an H surrounded by a circle).

The computer vision algorithm has the following steps described below:

1. Image Acquisition and Preprocessing.
2. Helipad Zone Extraction.
3. Helipad Marks Extraction.

4. Heliport 3D Reconstruction.

The computer vision algorithm has been developed maximizing its accuracy, performance and robustness. We tried to avoid false positives using a very long decision tree. A false negative (no measure when it has to be measured) is better than a false positive (a wrong measurement), because the state estimator (section "State Estimation: Kalman Filter") can manage it more easily.

4.1 Image Acquisition and Preprocessing

To start with the computer vision algorithm, the colour image is acquired (figure 11). The camera gives the image in the RGB (Red-Green-Blue) colour space. Then, the image is converted to an intensity image and to the HSV (Hue-Saturation-Value) colour space. Both images are preprocessed with a mean filter and then, with an opening morphological transformation [21]. With this image preprocessing we are preparing the image to the following steps. If the preprocessing would not be done, the computer vision algorithm would work slower and with less accuracy, performance and robustness.



Fig. 11 Example of an Acquired Image

4.2 Heliport Zone Extraction

In this step, we work with the preprocessed HSV colour image. A colour thresholding is done to get a binary image with the gray pixels of the heliport. This binary image also requires a preprocessing step, that consist on a median filter followed by an opening morphological transformation. Then, the blobs are extracted and the small ones are deleted. With all these preprocessing, we clear all the noise and small regions. Finally, the blobs are filled in and an OR logical transformation is done to get a whole binary image that represent the candidate pixels to belong to the heliport. Note that the heliport zone extraction give us not only the gray pixels, but also the white ones of the H and circle marks because of the blob's filled in that we did. Note also, that this step gives also other gray regions that can be visible in the image, giving us some false positives that we will filter in the next stages.

4.3 Helipad Marks Extraction

To start with this stage, we element-by-element multiply the intensity image obtained in the first step described and the binary output image of the previous section, getting figure 12.

This resulting image is thresholded looking for the white pixels of the heliport marks (the H and the circle). In order to remove noise and prepare this binary image, it is preprocessed with a median filter and an opening morphological transformation. Then, the resulting blobs are calculated, filtering those with small area (figure 13). The segmentation steps ends, and the classification step starts.



Fig. 12 Example of an Intensity Image after Heliport Extraction

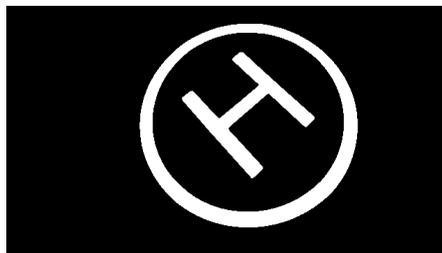


Fig. 13 Example of a Threshold Image, ready for look for Heliport Marks

The classification is done with a decision tree, where, each level gives some false positives, however, at the end of the tree, we will have one single solution. With this methodology, the speed and specially the accuracy of the computer vision algorithm is improved. In the first stages, an individual classification for Hs and circles is done. Then, we use both candidates (if found) to classify and verify them.

The first level is a fast classification using the Euler number (Euler number = connected components – number of holes) of each blob. H blobs have an Euler number equal to zero, and circle blobs' Euler number is one. Blobs with different Euler numbers are discarded. The Euler number is a scale, translation, rotation and homography invariant feature. The second level is a classification with a multi-layer perceptron (MLP) artificial neural network (ANN) with ten neurons in the hidden layer, tree outputs (H candidate, O candidate and other) and five inputs (see figure 14). The inputs are obtained with a principal component analyse (PCA) applied to the first seven invariant Hu Moments of each blob. Hu Moments are invariant to scaling, translation and rotation and are frequently used in Optical Character Recognition (OCR), [22], [23], [24]. Homography modifies a little these features, but they can be used in our decision tree. As the H has more information than the circle (because it is less symmetrical that the circle), we can use it in the third classification level. This third level uses the signature of the H candidates. The signature is invariant to rotation and translation; it preserves its shape to scaling; and some features of the shape (relative maximum and minimum) are preserved to homography. The H signature has four relative maximum and four relative minimum. The four maximum are the external corners of the H; and the four minimum are the bisectrix of the horizontal segment (above and below the centroid), and the bisectrix of the vertical segments (the external points). Because of its symmetry, if we connect the four maximum, we have a quadrilateral polygon whose center should be near to the centroid of the H blob. The same phenomenon appears with the minimum. The fourth level in our tree checks is the fact that all these three centers (center of maximum, center of minimum and centroid) have to be near in the image (distances in pixels have to be small). The fifth level checks the distance between the vertical straight lines of the maximum (vertical segment of the H) and the points of the minimum that should be in the vertical segment of the H. This distance has to be small (ideally zero).

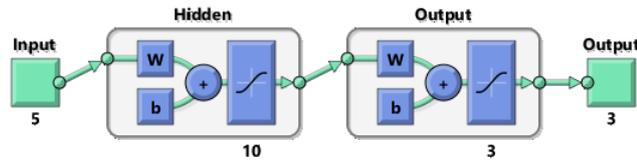


Fig. 14 Multi-layer perceptron (MLP) artificial neural network (ANN) with ten neurons in the hidden layer, tree outputs (H candidate, O candidate and other) and five inputs, used in the second level of the classification tree.

Hitherto, our classification tree uses only individual features to achieve its task. Now, we have to select only one H blob and one circle blob among all the resulting candidates that have to be compatible both together. The sixth level is based on the knowledge that the H has to be inside the circle. The seventh and last level calculates the coefficient between the area of the H and the area of the circle, which should be, more or less, a constant value. In these two last levels all H and circle blob candidates are tested, discarding those that do not satisfy the conditions checked in these last levels.

At the end of this step, we have the helipad marks (H and circle) extracted of the image (figures 15 and 16).

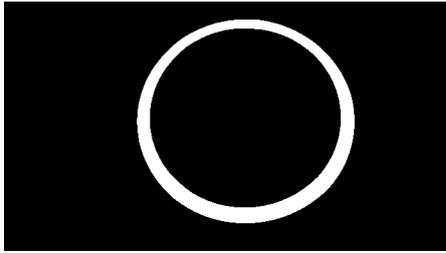


Fig. 15 Example of Circle selected blob

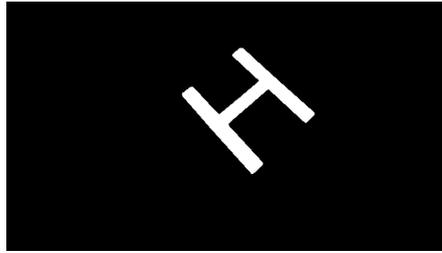


Fig. 16 Example of a H selected blob

4.4 Heliport 3D Reconstruction

The last step in the computer vision algorithm has to give us the 3D pose of the heliport with respect to the camera (on-board the helicopter). With the corners of the H of the helipad in the image (obtained thanks to its signature), we can calculate the homography matrix between these points of the image and the same points in a predefined target image. Then, using the homography matrix, we calculate the corners of the heliport knowing where are the corners in the target image (figure 17).

Once we have the corners of the heliport in the image, the 3D reconstruction has to be performed (figure 18). The reconstruction is based on the pin-hole camera model (equations 2 and 3, been $i = 1..4$), the square and known platform model (equation 4, been $i, j = 1..4$ and $i \neq j$; and equation 5, been $i, j, k = 1..4$ and $i \neq j \neq k$). The camera has to be previously calibrated (focal distance f , scale factors K_x and K_y and principal point C_x and

C_y , no distortion is assumed).

$$x_i \cdot f \cdot K_x - (x_{fi} - C_x) \cdot z_i = 0 \quad (2)$$

$$y_i \cdot f \cdot K_y - (y_{fi} - C_y) \cdot z_i = 0 \quad (3)$$

$$\|\mathbf{x}_i - \mathbf{x}_j\| = L_{ij} \quad (4)$$

$$(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{x}_j - \mathbf{x}_k) = 0 \quad (5)$$

Where $\mathbf{x}_i = [x_i, y_i, z_i]^t$ are the 3D coordinates of the point i in the central coordinate system, and x_{fi} and y_{fi} is the 2D coordinate of the point i in the camera lateral coordinate system.

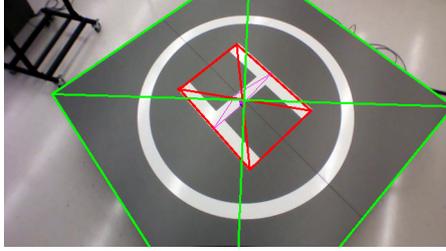


Fig. 17 Example of Output Image after the computer vision algorithm. In green, the heliport. In red, the H corners (maximum of signature). In purple, the minimum of the H signature

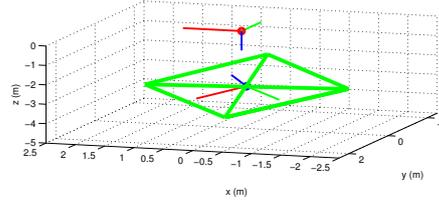


Fig. 18 Example of 3D reconstruction after the computer vision algorithm. The camera is fixed in the point $(0, 0, 0)$, looking downwards

5 State Estimation: Kalman Filter

In order to manage the measurements of the computer vision system, filtering the noise and calculating the pose of the heliport even when measurements are not available, a state estimator is needed.

In our problem we can see three coordinate systems: the first one, the World frame, fixed to the ground; the helicopter frame, fixed to the camera on-board the helicopter; and the last one, the heliport frame, fixed to the landing platform. The movement of the helicopter with respect to the World is modelled thanks to the helicopter model, and can be measured thanks to the IMU and GPS. The movement of the landing platform with respect to the World is unpredictable and we have no measure of it, but we have the measure of the movement of the helicopter with respect to the landing platform (the output of the computer vision system). If we assume that we have a good estimation of the pose of the helicopter frame, we can transform the computer vision measure into a measure between the World frame and the landing platform. With this transformation, we decouple the models (but not the measures), and it is easier to define them.

As the movement of the ship deck platform with respect to the World frame is unpredictable, we cannot assume a model to estimate its pose. We define the following model:

$$\frac{d^5 x_i}{dt^5} = 0 \quad (6)$$

Where x_i is the position of each DoF $(x, y, z, \theta, \psi, \phi)$ of the landing platform.

With this model, an Extended Kalman Filter is implemented to obtain the pose of the heliport, using the computer vision measurements and knowing the state of the helicopter.

6 Results

In this section, some examples of the results of previous sections are shown. More results and videos are available at <http://www.vision4uav.com/?q=jlsanchez/research>.

6.1 Computer Vision System

The computer vision system performance depends on different parameters:

- The camera selected: the resolution is a key factor in the detection step, and the parameters C_x , C_y , K_x , and K_y affect on the 3D reconstruction algorithm.
- The lens selected: with the resolution of the camera, is the other key factor in the detection step. Also, the f factor affects on the 3D reconstruction.
- The computer used: the framerate of the computer vision algorithm depends on the computer used.

Because of the ship deck platform is inside the lab, to test the computer vision, we additionally use a 19 cm x 19 cm scaled version of the heliport that allows us to test the computer vision system in an easy way. Of course, the lens is scaled too. The selected camera is a Point Gray Inc, Chameleon USB color camera (model CMLN-13S2C-CS), that can work with a resolution of 640 x 480 pixels at a rate of 24 frames per second (fps). The lens used is a FUJINON (model YV5x2.7R4B-2) with a focal length between 2.7 mm and 13.5 mm, which means an angle of view between $99^\circ \times 74^\circ$ and $20^\circ \times 15^\circ$. With this lens we can test our algorithm in different situations. We set the lens in a way that we were able to see our scaled heliport in a range distance between around 0.4 m and 1.5 m, that means between 2 and 8 times the size of the test heliport. For smaller distances, we assume that we are in the last step of the landing, and the SONARs would work. For bigger distances, we assume that we are far enough, and we can approximate the helicopter to the ship safely before starting the landing manoeuvre. Of course, these distances can be modified by changing the lens focal distance. Also someone could consider the use of two or more cameras to see the heliport at different distances, running the same computer vision algorithm.

Once the camera and the lens is set for our tests, we calibrated it, getting: $f \cdot K_x = 1307.549726$, $f \cdot K_y = 1307.549726$, $C_x = 319.5$ and $C_y = 239.5$. The distortion of the lens does not affect. If the distortion affects, a previous step in the computer vision algorithm could be done to correct it.

The used PC to run the algorithm is a 2010 laptop with a Dual-core Intel Core i3 CPU @ 2.26GHz. The framerate achieved is around 6 fps, but with a more powerful PC, this framerate could be improved easily up to 20 fps or more. By the way, this low framerate is enough for our application, because the helicopter is equipped with an IMU (and also a GPS) that allow us to know its pose faster. That means that with the computer vision system we only need to measure the pose of the ship with respect to the helicopter and the frequencies of the movement of the ship are much slower than 6 fps (as we saw in a previous section, they are between 0.07 and 0.13 Hz, so a framerate of 6 fps is more than 45 times faster).

In figures 19 to 21, the performance of the detection steps of the computer vision algorithm is tested. We can affirm that for different distances and angles of the heliport with respect to the camera, the algorithm is able to detect the heliport. We also can affirm that even with a lot of contamination or partial occlusions of the heliport or even with illumination changes, it can still be detected.

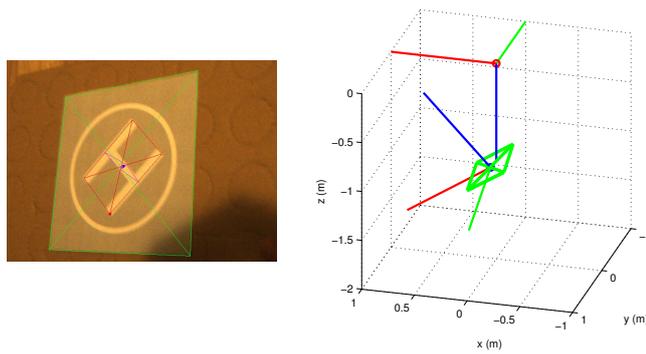


Fig. 19 Example of Heliport very tilted. The 3D reconstruction reflects the tilting, and the computer vision algorithm still works despite the huge tilting

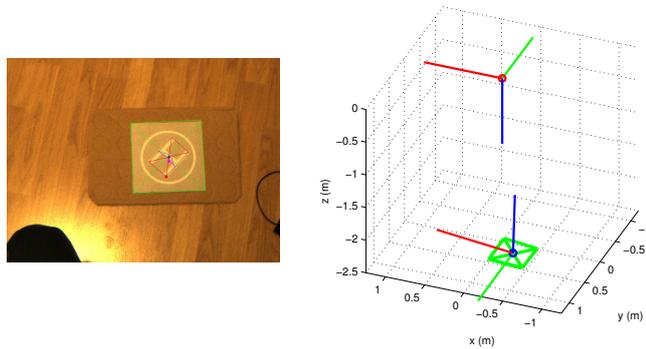


Fig. 20 Example of Heliport far away. The 3D reconstruction reflects the bigger distance. The computer vision algorithm is able to work in different ranges of heliport distances.

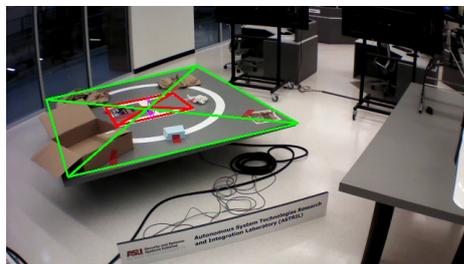


Fig. 21 Example of Contamination on the heliport. The computer vision algorithm works appropriately even with a really big and probably "unrealistic" contamination.

Other important issue to evaluate is not only the performance of the detection of the helipad, but also the final 3D reconstruction of the heliport. The accuracy of the computer vision system (for our camera-lens-heliport selection) depends on the pose of the heliport with respect to the camera. Tests showed us that, as we expected, the position of the center of the heliport (x , y and z coordinates) with respect to the camera frame is more accurate and less noisy than the Euler angles (θ , ψ and ϕ coordinates). Also, we can see that the noise of the euler angles is reduced if the heliport is tilted with respect to the camera, that mean, the

most dangerous poses. So, the less parallel the camera plane is with respect to the heliport plane, the system better works. Figures 22 to 24 show a test with the heliport parallel to the camera plane. Figures 25 to 27 show other test with the heliport tilted.

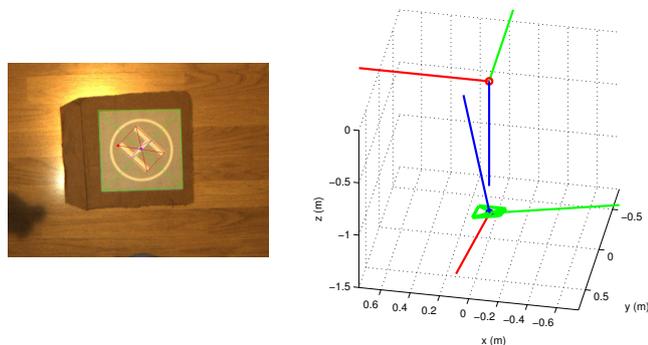


Fig. 22 Heliport at a distance of around 6 times the size of the helipad. The heliport plane is parallel to the camera plane. This is one of the cases in which the 3D reconstruction works worst (very noisy).

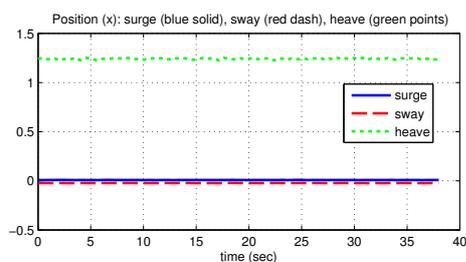


Fig. 23 Position x , y and z of the centre of the heliport with respect to the camera when is stationary like in figure 22. Mean values: 0.008 m, -0.0229 m, and 1.2415 m. Standard deviations: 0.000460 m, 0.000425 m, and 0.0069 m.

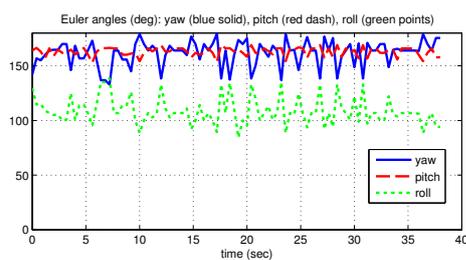


Fig. 24 Euler Angles Yaw, Pitch and Roll of the heliport with respect to the camera when is stationary like in figure 25. Mean values: 161.6°, 162.9°, and 108.6°. Standard deviations: 11.60°, 3.90°, and 12.47°.

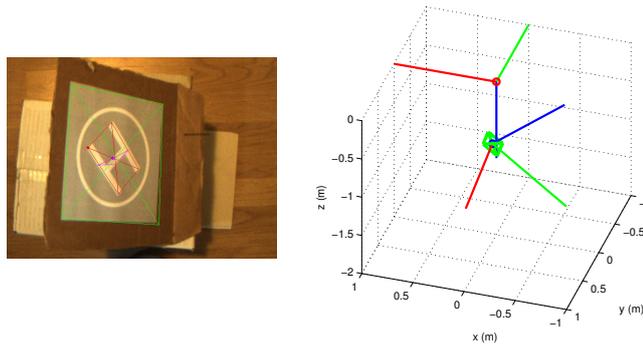


Fig. 25 Heliport at a distance of around 3 times the size of the heliport. The heliport plane is tilted with respect to the camera plane. In this case the 3D reconstruction works better (less noisy).

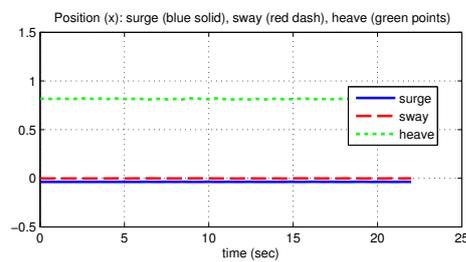


Fig. 26 Position x , y and z of the centre of the heliport with respect to the camera when is stationary like in figure 22. Mean values: -0.0433 m, -0.0142 m, and 0.6857 m. Standard deviations: 0.000425 m, 0.000526 m, and 0.0016 m.

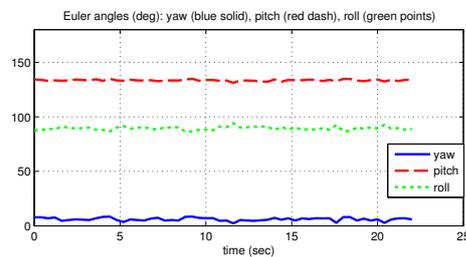


Fig. 27 Euler Angles Yaw, Pitch and Roll of the heliport with respect to the camera when is stationary like in figure 25. Mean values: 7.9° , 133.0° , and 72.8° . Standard deviations: 0.98° , 0.38° , and 1.22° .

6.2 State Estimation

In figure 28, one of the DoF of the motion platform (the z movement) is estimated using the Kalman Filter with the model proposed in section "State Estimation: Kalman Filter" and typical measures and noises of the computer vision system described, after the transformation to World frame's coordinates.

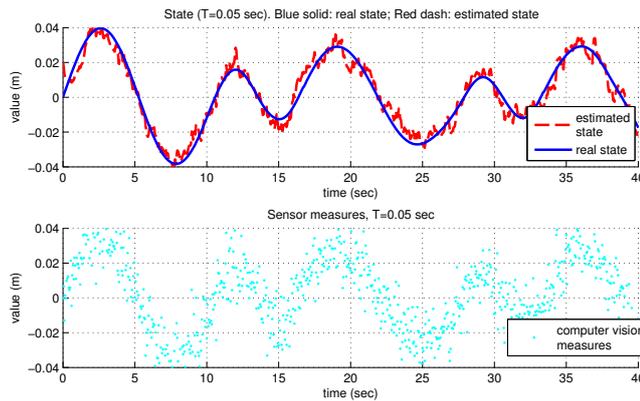


Fig. 28 Example of State Estimation of the pose of the motion platform. It is shown the z movement. In the upper plot, in blue, the real state, and in red, the estimated state. In the bottom plot, in cyan, the measurements of the computer vision system, with its typical noise, after the transformation to refer them to the World frame.

7 Conclusion and Future Work

In this paper a new and complete ship deck simulation for the autonomous landing of VTOL UAVs on ships is proposed using a real Motion Platform. This simulation fulfils the requirements of being accurate, realistic, random and simple enough, therefore we can use it easily without losing realism. The pose of this landing platform is measured using a single downward looking camera computer vision system on board the helicopter for standard grey helipads with an H surrounded by a circle. The computer vision requires the knowledge of the deck size for the 3D reconstruction. This algorithm was developed having in mind robustness, avoiding any false positive. Also, it works appropriately even with contamination on the helipad or light changes. A state estimator that uses the computer vision measures, calculates the state of the landing platform, removing its noise and avoiding the problems when the helipad is not detected. These are the first steps required to achieve a solution to the challenge of autonomously landing on a ship.

To complete this challenge, as future work, a state estimator which incorporates the helicopter model, and IMU measurements is required. Additionally, a controller will need to be designed and tested in order to close the control loop.

8 Acknowledgement

The authors would like to thank the Consejo Superior de Investigaciones Científicas (CSIC) of Spain for the JAE-Predoctoral first author's scholarship and the UECMUAVS Project (PIRSES-GA-2010) included in the Marie Curie Program and the Spanish Ministry of Science MICYT DPI2010- 20751-C02-01 for project funding.

References

1. S. Saripalli, J. F. Montgomery, G. S. Sukhatme, *Visually Guided Landing of an Unmanned Aerial Vehicle* IEEE Transactions on Robotics and Automation, Vol. 19, No. 3, Pp. 371-380, June 2003

2. D. Lee, T. Ryan, H. J. Kim, *Autonomous Landing of a VTOL UAV on a Moving Platform Using Image-based Visual Servoing* 2012 IEEE International Conference on Robotics and Automation, May 2012
3. S. Saripalli, *Vision-based Autonomous Landing of an Helicopter on a Moving Target* AIAA Guidance, Navigation and Control Conference, 2009
4. A. Palomino, P. Castillo, I. Fantoni, R. Lozano, C. Pegardy, *Control strategy using vision for the stabilization of an experimental PVTOL aircraft setup* Proceedings of the 42nd IEEE Conference on Decision and Control; Maui, Hawaii USA, December 2003
5. R. Lozano, P. Castillo, A. Dzul, *Global Stabilization of the PVTOL: Real-Time Application to A Mini-Aircraft* International Journal of Control, Vol. 7, No. 8, Pp. 735-740 May 2004
6. M. I. Lizarraga, *Autonomous Landing System for UAV* PhD. Thesis, March 2004
7. M. Bodson, M. Athans, *Multivariable Control of VTOL Aircraft for Shipboard Landing*, 1985 AIAA Guidance and Control Conference, January 1985
8. S. Khantsis, *Control System Design Using Evolutionary Algorithms for Autonomous Shipboard Recovery of Unmanned Aerial Vehicles*, Ph.D. Thesis, Royal Melbourne Institute of Technology, August 2006
9. S.-R. Oh, K. Pathak, S.K. Agrawal, H.R. Pota, M. Garratt, *Approaches for a Tether-Guided Landing of an Autonomous Helicopter* IEEE Transactions on Robotics, Vol. 22, No. 3, Pp 536-544, June 2006
10. C. Fu, J. Pestana, I. Mellado, J. L. Sanchez-Lopez, P. Campoy, *Visual Identification and Tracking for Vertical and Horizontal Targets in GPS-denied Indoor Environments* IMAV 2012, July 2012
11. C. S. Sharp, O. Shakernia, S. S. Sastry, *A Vision System for Landing an Unmanned Aerial Vehicle* Proceedings of the 2001 IEEE International Conference on Robotics and Automation, May 2001
12. Y. Fan, S. Haiqing, W. Hong, *A Vision-Based Algorithm for Landing Unmanned Aerial Vehicles* 2008 International Conference on Computer Science and Software Engineering, 2008
13. C. Xu, L. Qiu, M. Liu, B. Kong, Y. Ge, *Stereo Vision based Relative Pose and Motion Estimation for Unmanned Helicopter Landing* Proceedings of the 2006 IEEE International Conference on Information Acquisition, August 2006
14. T. Perez, M. Blanke, *Mathematical Ship Modelling for Control Applications*, Technical Report, 2002
15. L. Marconi, A. Isidori, A. Serrani, *Autonomous Vertical Landing on an Oscillating Platform: an internal-model based approach* Automatica, No. 38, Pp. 21-32, 2002
16. B. Hris, T. Hamel, R. Mahony, 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, February 2012
17. S.-K. Ueng, D. Lin, C.H. Liu, *A ship Motion Simulation System*, Springer Virtual Reality, Pp. 65-76, February 2008
18. A. Lawter, M. J. Griffin, *Motion Sickness and motion characteristics of vessels at sea* IEEE Transactions on Ergonomics, Vol. 31, No. 10, Pp. 1373-1394 1988
19. C. Reboulet, P. Mouyon, Dr. B. de Ferrier, B. Langlois, *Piloting of a VTOL-UAV to Shipboard Recovery* RTO MP-15, October 1998
20. R. J. Saltaren, L. J. Puglisi, *Robotica Aplicada Utilizando Matlab: Analisis y diseño de robots* Book, Seccion de Publicaciones de la ETSII-UPM, April 2012
21. R. C. Gonzalez, R. E. Woods, *Digital image processing* Prentice Hall, Pearson Education International, New Jersey, USA, 2002
22. Q. Chen, E. M. Petriu, *Optical Character Recognition for Model-based Object Recognition Applications* IEEE, 2003
23. M. Kmiec, *New Optical Character Recognition Method Based on Hu Invariant Moments and Weighted Voting* JOURNAL OF APPLIED COMPUTER SCIENCE, Vol. 19, No. 1, Pp. 33-50, 2011
24. R. S. Kunte, R. D. S. Samuel, *A simple and efficient optical character recognition system for basic symbols in printed Kannada text* Sadhana, Vol. 32, Part 5, Pp. 521-533, October 2007