

Trajectory and Guidance Mode for autonomously landing an UAV on a naval platform using a vision approach

Filipe Morais, Tiago Ramalho & Pedro Sinogas
Tekever Autonomous Systems
Tekever
1990-113, Lisboa, Portugal

Mario Monteiro Marques, Nuno Pessanha Santos & Victor Lobo
Centro de Investigação Naval (CINAV)
Marinha de Guerra Portuguesa (MGP)
2810-001, Almada, Portugal

Abstract—It is proposed a navigation system based on the unmanned aerial vehicle (UAV) onboard camera that can land on a moving naval platform. It is considered that, besides the usual sensors, the aircraft is equipped with a RGB digital camera capable of detecting markers located on the ship's deck. The proposed method uses a framework based on a Kalman filter (KF) and an efficient perspective-n-point (EPnP) approach to estimate the position, attitude and velocity of the target (landing system). This information is combined and used to generate a landing trajectory. The algorithm is suitable for real time implementation in standard workstations because of the low computational time required.

Keywords—Computer Vision; Autonomous Vehicles; Kalman Filters; Military Systems; Pose estimation; Perspective-n-Point

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are changing the way military and civil operations are carried out. This technology is bringing higher levels of efficiency to tasks such as data and image acquisition, localization and tracking of specific targets, map building, communication relays, pipeline surveying, border patrolling, military operations, policing duties, persistent wide area surveillance, search and rescue, and traffic surveillance.

Nowadays, fast patrol boats (FPB) have an important role in this kind of missions but their efficiency can be significantly improved by the support of UAVs. The available landing site in this kind of ship is usually a small and irregular area (around 5x6m area in the stern section). The typical short autonomy of the small UAVs used in this kind of operations makes on-board take-off and landing a requirement for sea operations.

The vision-based methods to autonomously land an UAV have been subject of some attention in related literature, with some good results proven in the case of rotary-wing aircrafts such as in [1]. Methods developed for fixed-wing aircrafts usually use restraint systems with distinctive colors for an easy detection in camera images and direct visual servoing to correct the UAV's trajectory, as in [2], [3] and [4].

All the methods mentioned bellow, are for direct visual servoing; they do not explicitly estimate the target's 3D

position or orientation. This makes them purely reactive and can difficult the landing operation on a naval platform affected by wave and wind, for example. This is aggravated when the vessel is moving in a direction not coincident with the restraint system's orientation. A naval platform will thus have one single valid entry point, leading to the need to specifically estimate the target's 3D position, its velocity and reaction. This allows an adequate trajectory for the landing phase.



Fig. 1. Aircraft and naval platform with landmarks and restraint system.

Another limitation of the already existing methods is that the guiding mechanism (e.g. distinctive color or shape) must necessarily belong to the restraint system. An approach, in which the visual cues perceived by the UAV (markers) do not necessarily lie in the landing area, was therefore found more suitable due to the limited space available in a ship's deck.

We adopted a combination of a KF and an EPnP framework for the target state estimation. This approach requires the presence of (at least) four non-collinear reference points. The points' relative position to the target is constant and previously known. This setup can easily be achieved by setting signaling lights to the ends of the restraint system or other fixed points on the platform. The reference points must be visible and identifiable to at least one hundred meters. The correct markers' identification is essential, since they can be confused with each other and with other ship lights (e.g. Navigation lights). To overcome this, we are chose to use directional lights to create additional physical constraints. These can then be used to eliminate existing ambiguities. Some assumptions regarding the UAV behavior were also made: it does not fly in an inverted position, for instance.

This system uses the UAV onboard camera to obtain the position (in pixels) of the reference points. This information is

then used in the navigation phase, which generates commands for both the aircraft and camera (direction and zoom).

In Section II we describe the used notation in this paper. In Section III the overall UAV system architecture is illustrated. In Section IV the trajectory and guidance architecture is described, which in turn is divided in the following components: (1) Image Processing, (2) Absolute pose measure, (3) Filtering, (4) Trajectory generation and (5) Control. In Section V some experiments concerning simulation results are presented. The experimental results available at the time of publication are presented in Section VI. Finally, in Section VII we present the conclusions of the paper.

II. NOTATION

Throughout this paper the following notation will be used:

- Vectors will be denoted by a lower-case letter whereas matrices will be represented by an upper-case letter;
- Vectors may have a leading super-script stating the reference frame in which they are represented (e.g. ${}^B v$ would represent the vector v in the $\{B\}$ frame);
- Rotation matrices will always be represented by the letter R . The matrix will have both a (leading) super-script and a subscript, indicating the reference frames between which the mapping refers (e.g. ${}^W_B R$ would be the rotation matrix from $\{B\}$ to $\{W\}$, such that ${}^W v = {}^W_B R {}^B v$).

III. SYSTEM ARCHITECTURE

The used UAV's (Fig. 2) characteristics are:

- **Wingspan:** 1800 mm;
- **Length:** 1200 mm;
- **Weight:** 3 Kg + 2 Kg payload (max);
- **Cruise Airspeed:** 55Km/h;
- **Autonomy:** 2h;
- Global Positioning System (GPS);
- Attitude and Heading Reference System (AHRS);
- Auto-pilot.



Fig. 2. Test UAV used.

The aircraft communicates via data link, obtaining in real time telemetry and video data. An operator typically receives this information in a Ground Control Station (GCS), which also enables it to send commands to the aircraft and update its mission in real time. A Nano GCS can also be used as an emergency system or in cases where extreme mobility is required. All ground devices are ruggedized in order to allow its use in rough conditions. The overall system architecture can be seen in Fig. 3.

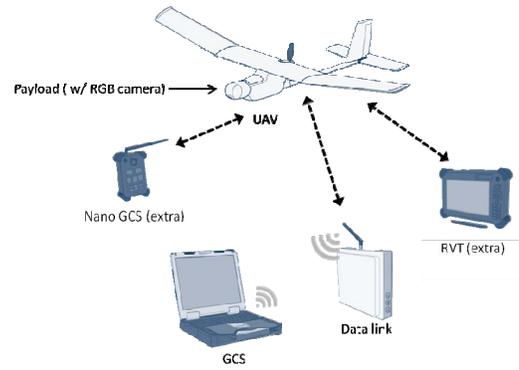


Fig. 3. System Architecture.

IV. TRAJECTORY AND GUIDANCE ARCHITECTURE

This section introduces the proposed trajectory and guidance architecture. The proposed method is modular and can be divided in five parts:

- **Image processing** – The landmarks are identified in the video stream;
- **Absolute pose measure** – An absolute pose measure is obtained by solving the perspective-n-point problem and using data from the UAV sensors;
- **Filtering** – In this stage, we estimate the position, behavior and absolute velocity of the target applying a filtering step to the measurements obtained in the previous stage;
- **Trajectory generation** – In this stage, we generate a trajectory suitable for the UAV landing;
- **Control** – In this stage, we identify the control commands given to the UAV and to the camera.

A. Image Processing

Since the used markers are light beacons, their identification can be done by analyzing its luminance (threshold level), followed by morphological operations. We thus extract the blobs and remove the outliers both by individual criteria (e.g. minimum and maximum area) and relative criteria (e.g. geometric consistence between them, given a pre-defined disposition pattern). Since the use of a simple threshold leads to poor results, the proposed detection mechanism goes as follows:

- To convert the image to binary using a very restrictive threshold (i.e. some beacons can be considered as background);
- To extract blobs and reject outliers with individual criteria;
- To test subsets (with size equal to the known number of markers) of the detected blobs for relative consistency. If no consistent subset is found, use a lower threshold level and repeat;
- The process ends when a geometrically consistent set of markers is detected or the binary threshold gets too low

(i.e. most of the image is already classified as potential beacons) and in this case the frame is discarded.

B. Absolute pose measure

Let us assume we have set of n reference points (markers) whose 3D coordinates are known in a local coordinate system $\{T\}$ and whose 2D image projections are also known. The position of the landmarks in any given video frame in homogeneous coordinates (\bar{P}_1) is given by:

$${}^c_z \bar{P}_1 = K [{}^c_{\bar{T}}R \mid {}^c\Delta] \bar{T}P_{ref} \quad (1)$$

where ${}^c_{\bar{T}}R$ is the rotational mapping between the reference systems of the target, $\{T\}$ and of the camera, $\{C\}$, ${}^c\Delta$ is the displacement between the camera and the center of the target (expressed in the camera referential), K is the camera intrinsic parameters matrix and $\bar{T}P_{ref}$ are the homogeneous coordinates of the reference points in the reference frame of the target. It is noted that by choosing the reference $\{T\}$ appropriately, $\bar{T}P_{ref}$ is constant and known. The matrix K is a function of the zoom level and can be obtained through simple calibration procedures [5] [6].

There are innumerable approaches in the literature to solve the perspective- n -point problem and obtain ${}^c_{\bar{T}}R$ and ${}^c\Delta$ from \bar{P}_1 and $\bar{T}P_{ref}$ [7] [8]. The EPnP approach was chosen since it can be used for any number of points (larger than three) as well as for its speed (it offers an approximate solution to the problem) and accuracy. Although other solutions to generate 3D pose measurements from RGB images exist (such as using LED arrays and line-based methods), the n landmarks system was chosen due to its simplicity, making it easy for practical implementations [3].

A measurement of the target center position in the inertial frame $\{W\}$ is obtained given the UAV and camera position in the same referential, which can be known using GPS and ARHS. The center of the target corresponds to the origin of the referential $\{T\}$ as in:

$${}^wP_{target} = {}^w_C R {}^c\Delta + {}^wP_{camera} \quad (2)$$

where ${}^wP_{target}$ and ${}^wP_{camera}$ are the target 3D coordinates and camera respectively in the inertial frame $\{W\}$. The target rotation matrix concerning the orientation is obtained as in:

$${}^w_{\bar{T}}R = {}^w_C R {}^c_{\bar{T}}R \quad (3)$$

The rotation matrix can then be decomposed into the three standard Euler angles representation (yaw, pitch and roll).

C. Filtering

The pose measures obtained from the method described in the last section are affected by error from various sources such as: sensor noise (e.g. ARHS, GPS, etc.), camera calibration errors (intrinsic matrix), image reference points' detection algorithm error, radial and tangential distortion in the image and modeling approximations. Even if the pose measures were infinitely accurate, it would still be difficult to land successfully on a moving naval platform, due to the lack of predictive components. Thus, it is extremely important to filter the pose measures in order to reduce the errors and to estimate

the speed of the vessel. For the target motion model we assumed that:

- The naval platform is co-operative and does not perform evasive maneuvers (constant orientation and velocity);
- The target roll angle average is zero (i.e. exclusively induced by zero-mean wave motion);
- The target yaw and pitch oscillate around a constant value;
- The average vertical target velocity is zero.

To estimate the variables dynamically we used a Kalman Filter [10] considering constant velocity and orientation between frames. The measurements, z , are the inertial position and Euler angles obtained in the previous stage.

$$z = \begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \\ \bar{\theta} \\ \bar{\psi} \end{bmatrix} \quad (4)$$

The state vector X_t is then defined by the target 3D position in the world coordinate frame, the target linear planar velocity and two attitude angles.

$$z = \begin{bmatrix} X \\ Y \\ Z \\ v_x \\ v_y \\ \theta \\ \psi \end{bmatrix} \quad (5)$$

The process covariance matrix, Q , can be considered as a tuning parameter. Still, its values should be quite low, due to the co-operation assumption. The measurement covariance matrix, R , is a function of \hat{X}_t , since the noise of the EPnP output varies drastically with the distance to the target, as well as with the camera perspective. The camera's level of zoom is also taken into account.

The filtering stage must have low computational time, since the landing maneuver is delicate and is desirable to have security mechanisms that allow early detection of problems in order to avoid collision. Dynamic constraints (physical and statistical – Mahalanobis distance) were also applied to remove measure outliers.

D. Trajectory generation

The UAV trajectory is chosen according to the state machine represented in Fig. 4. For each state the behavior of the UAV and the transition conditions are:

- **Search** – This is the only stage where the output of the KF is not used. In fact, in this stage the aircraft is simply driven to the last reported position of the GCS; the altitude is predefined taking into account both safety and visibility constraints. The state transition is made when the target is sighted and the KF's covariance matrix is below a predefined threshold;

- **Loiter** – In this state, the aircraft makes loiters behind the target to stay in line-of-sight with the target and obtain pose measurements. The aircraft advances to the approach stage when the state covariance is below a predefined threshold;
- **Approach** – In this stage the UAV is guided to the landing cone (two-hundred meters behind the target). The reference trajectory is a straight line to that point and the UAV will also decrease altitude if necessary. The stage transition to the land stage is made when the UAV is in the correct position, altitude and attitude to start the landing maneuver. If the UAV is outside the predefined trajectory it returns to the loiter stage;
- **Land** – In this stage the UAV travels in a straight line towards the center of the target and also decreases altitude to achieve the target altitude. If the target is moving the position is estimated at the (also estimated) time of impact (landing). The stage transition to the final no return landing mode is made based on the time/distance to the target. If the UAV is approaching the target outside the predefined landing trajectory or accuracy envelopes a go-around is performed;
- **No return** – In this stage the UAV will move straight to the target, staying in this mode until it reaches the target or a predefined interval of time is exceeded if this occurs a go around is performed;
- **Go around** – The aircraft performs a go-around to avoid collision or a dangerous landing (the predefined maneuver is to increase altitude and turn right). After a safe altitude is achieved the UAV returns to the Search stage.

In any of the mentioned stages above, except for the Land and No Return, the UAV may return directly to the search mode if the target is not sighted for a long time or the state covariance is too high. The camera is kept pointed to the estimated position of the target to ensure that vision-based measurements are available. The exception is the No Return stage; in this final stage the camera is kept pointing forward since the proximity to the target would make its control too unstable, obtaining images affected by motion blur.

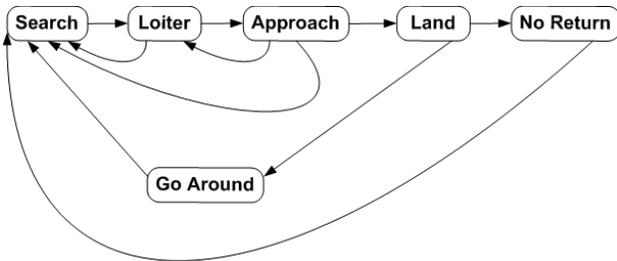


Fig. 4. Trajectory State Machine.

E. Control

We assumed that the UAV is already equipped with previously developed and validated low-level controllers, including an auto-pilot system capable of following the generated trajectories. For simulation purposes only, a line-of-

sight navigation scheme (based on the ones presented in [11] and [12]) was implemented. While this will certainly not mimic the aircraft’s behavior exactly, it was considered to be accurate enough to validate the proposed methodology in this work stage.

V. SIMULATION RESULTS

A. Simulation Conditions

A simulation scheme implemented in MatLab Simulink was built for testing purposes, as seen in Fig. 5. The blue blocks represents the systems that, in practice, will be replaced by their physical counterparts. The green ones correspond to the implementation of the several algorithms that constitute the proposed methodology.

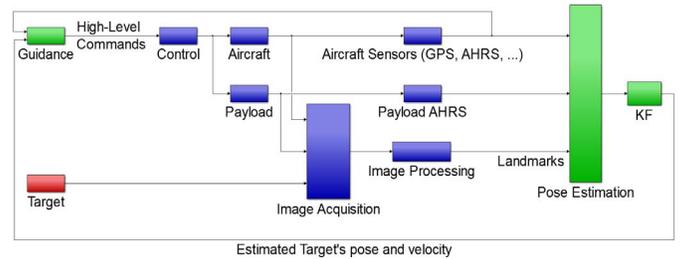


Fig. 5. Simulator Architecture.

Simple models of the aircraft and the camera gimbal were implemented, including saturations representing physical limits (e.g. turning speed, descending rate, etc.). All aircraft sensors mentioned in the previous sections (including the RGB camera) were modeled with appropriate noise and signal output frequencies. Noise was also added to the measured reference points position (in the image), to simulate both model and in the image processing errors a module was responsible for the detection of the reference points. A simple constant wind model was also considered. Biases were considered in the barometer (leading to an altitude bias, which is commonly found in practice) and camera/gimbal disturbances are also considered.

Furthermore, some line of sight considerations were also taken into account. This means that the target is not visible to the aircraft from the rear. The target is also considered to be out of sight when it is too far from the aircraft/camera, as well as when the projection of the markers land on pixels too close to each other (e.g. due to a bad aiming angle). Finally, the aircraft simulated is unable to aim backwards since it is considered to be carrying a frontal camera.

Simulations with both a stopped and a moving target were performed. In the moving target case, the target’s yaw was considered to be fairly aligned with its bearing. This is because we assume that the retention mechanism will be aligned with the naval platform’s rear. The misalignment considered is to take into account sideslip induced by wave motion or other external factor.

In all simulations the target’s attitude was kept oscillatory around constant values of yaw and pitch (unknown to the UAV), and zero roll. The oscillations were intended to represent a simple model of wave or wind induced motion.

B. Simulation Results

The figures presented in this section show the results of a typical simulation instance, where the aircraft chases and lands on a moving platform. Fig. 6 shows the guidance state along the landing manoeuvre. The 3D target position throughout time can be seen in Fig. 7.

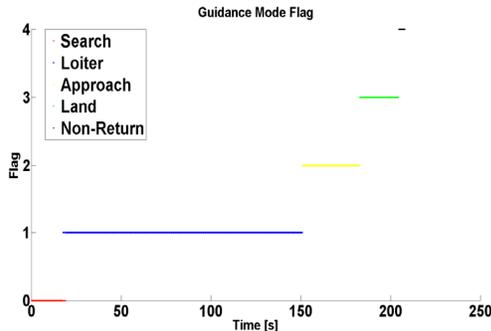


Fig. 6. Guidance mode along maneuver.

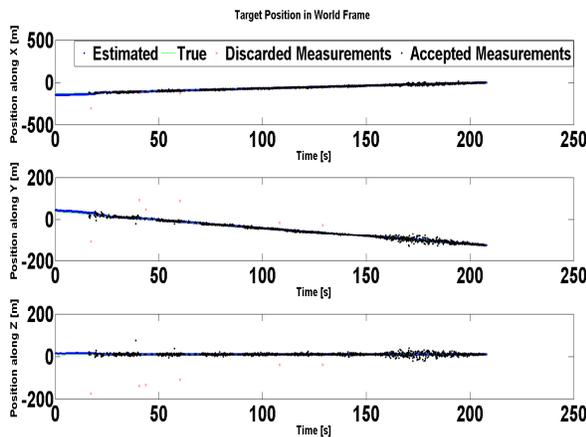


Fig. 7. Target Position (3D).

It can be noticed that until around $t = 20$ s, the position estimative is biased. This is due to the fact that the aircraft has not yet sighted the target and thus no pose measurements are available to the filter (notice in Fig. 6 that until then the aircraft is in Search mode). This bias is swiftly corrected after the aircraft spots the target. It is also important to notice the performance of the outliers' rejection block; many highly erroneous measurements were automatically and correctly discarded by the filter. In Fig. 8 both the true and the estimated target velocity can be seen.

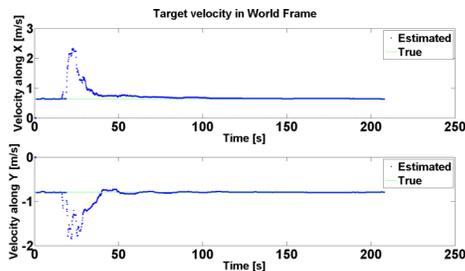


Fig. 8. Target Velocity.

It can be noticed that, even before the aircraft has a line of sight towards the target, the estimator already possesses a fairly good estimative of the target's speed. This is due to the GCS messages providing it with GPS speed, which should be fairly accurate. The large transient observed at around $t = 20$ s (when the aircraft gets its first images of the target) could certainly be improved by filter parameter tuning; however, it was decided to keep it this way in simulation to ensure then that the filter is capable of recovering from grossly wrong initial velocity conditions and quickly converge to the correct values, using the image-based measures alone. The target attitude throughout the landing maneuver is represented in Fig.9. The green line (marked as true) is the nominal target attitude, that is, the target orientation before wave induced motion.

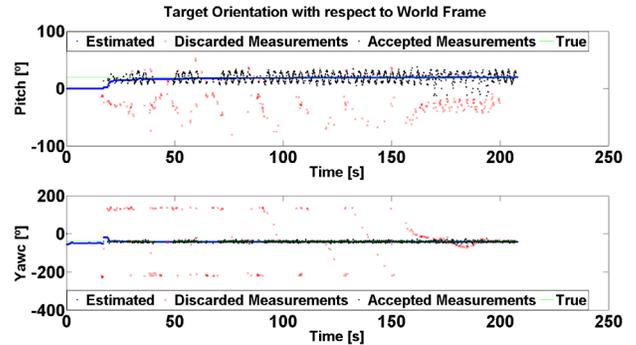


Fig. 9. Target orientation.

Similarly to the behavior of the position estimative, it can be seen that the yaw estimative is biased until the aircraft gains line-of-sight with the target. This occurs until image-based pose measurements are available and the filter assumes that the target's orientation is aligned with its planar velocity ignoring side-slip. The pitch angle was initialized to zero and only converges to the correct value after the aircraft starts collecting pose measurements on the target. As before, it is also important to notice the performance (and necessity) of the outliers' rejection mechanisms. The wave-induced oscillations are also quite visible in the pitch measurements. Fig. 10 is particularly enlightening when it comes to understanding the behavior of the aircraft throughout the maneuver.

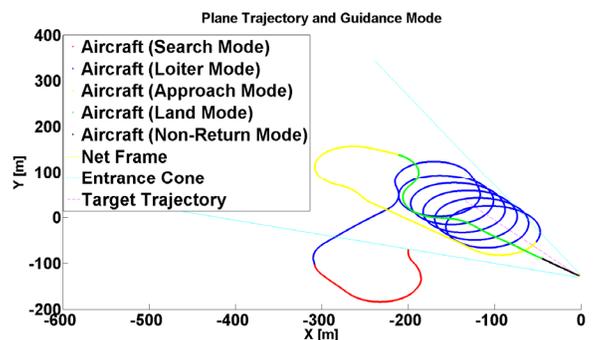


Fig. 10. Plane and target trajectory.

It is quite visible that, during the Land stage, the aircraft is not trailing the target. This occurs because in this particular simulation, the target's direction of movement is not coincident with its yaw angle (i.e. the naval platform is side slipping). In

fact, it can be noticed that the aircraft proceeded fairly along the normal to the landing cone (drawn at the moment of impact), as opposed to the target's trajectory. This desirable behavior is explained by the predictive capacity referred to in section IV.D.

Finally, the zoom level is automatically defined by the payload using an embedded controller (Fig. 11). During the landing maneuver (aircraft gets closer to the target) the preferred zoom level diminishes. The largest number of outliers occur at lower zoom levels; this reinforces the importance of using a precise aiming system, so as to employ the maximum zoom possible.

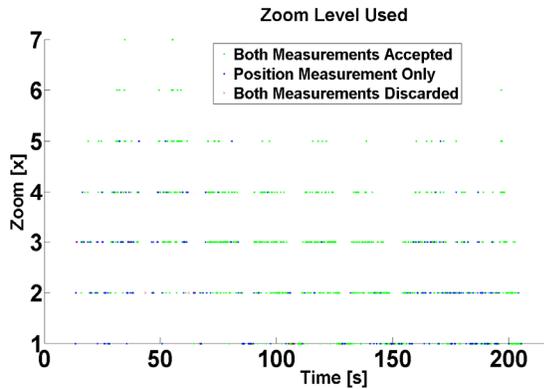


Fig. 11. Zoom levels.

VI. EXPERIMENTAL RESULTS

Some preliminary real tests were conducted to validate three main assumptions required by the proposed system: the visibility and automatic detection of light beacons in aerial images; the robustness of a net-based retention system, in terms of its capability of both holding the aircraft and not damaging it; the capability (in terms of maneuverability) of the aircraft to trail and align itself with the naval platform rear section.

A. Beacon Visibility and Detection

This tests were made guiding the UAV above light beacons and post-processing the videos with the techniques described in IV.A. Fig. 12 shows an aerial image taken by the aircraft's camera, with the three light beacons present in the scenario highlighted with green circles. The red circle aims to draw attention to an outlier, though it looks like another beacon it is actually light reflections from the GCS and other equipment.



Fig. 12. Aerial image with light beacons.

Fig. 13 show the result of the binary conversion process. This may seem to be too permissive, since large areas of terrain were identified as possible beacons. However, it can be easily explained by the light coloring of the terrain.

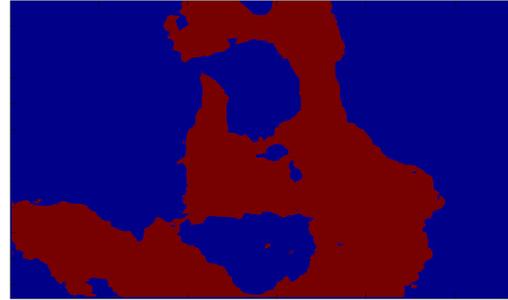


Fig. 13. Binarised image.

Finally, the labeled beacons can be seen in Fig. 14. Notice that the whole background was discarded in the outlier's rejection procedure. The right-most beacon was not correctly identified and in fact it is not easily detected by human inspection. The left-most identified beacon corresponds to the outlier mentioned previously. Since the beacons used for this test were not deployed under any specific geometric configuration (e.g. four in a rectangular formation) it was not possible to apply geometry-based outliers' rejection at the blob level, nor the adaptive threshold method proposed in IV.A.

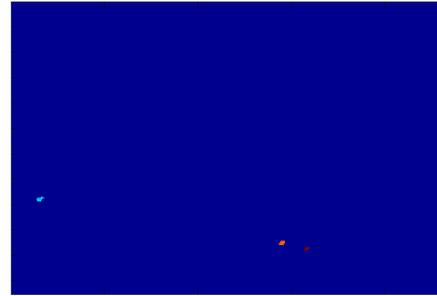


Fig. 14. Labeled image.

B. Net-based retention system

A prototype net-based retention system was tested attached to a football goal frame, as can be seen in Fig. 15. The aircraft was repeatedly sent against the retention system, both manually and in flight (with small joystick corrections). The retention system proved itself able to hold the aircraft, as in Fig. 16. The only worrying issue detected was that, when the aircraft landed close to the base of the net, it would sometimes hit the ground, with consequent possible mechanical injury. This issue should not occur in practice, since in the naval platform the retention system will be extended above an opening.



Fig. 15. Net-based retention system prototype.



Fig. 16. Aircraft trapped in the retention system.

C. Aircraft maneuverability

To assess that the aircraft was physically capable of trailing and aligning itself with a moving naval platform some sea trials were conducted. The aircraft was flying autonomously and commanded to follow a trajectory that would make it fly by above the naval platform. Small (manual) joystick commands were then sent to the aircraft to correct its position and alignment when it approached the vessel, mimicking the intended behavior of the image-based closed loop guidance system. Fig. 17 shows one of the aerial images taken during a fly by. It was seen that, with few corrections, the aircraft was capable of trailing and flying by the vessel aligned with its rear.



Fig. 17. Aerial image taken during sea trials.

VII. CONCLUSIONS

The light beacons detection in aerial images was tested and validated. Even though the experiments reported in VI.A lead both to a false negative and a false positive, the reasons behind both are fully understood and easy to overcome in practice. It should also be taken into account that some external factors can affect the system reliability such as the naval platform's wake or water light reflections. Colored beacons can be used to overcome this issue. The markers do not necessarily need to be attached to the retention system. This is relevant due to the space restrictions inherent to naval platforms. In simulation environment it was shown that the proposed navigation system is capable of leading an airborne UAV towards a moving target in the presence of GPS and barometer biases. In the simulations performed the aircraft was equipped with a three-axis (roll-yaw-pitch) payload, which makes it capable to keep the camera aimed at the target during the loiter stage. Since many fixed-wing UAVs employ two-axis (roll-pitch) nose payloads instead, it should be noted that in that case the behavior of the aircraft during that stage needs to be adapted, to a sigmoid trajectory for instance. The simulations performed have confirmed that, if the co-operations requirements stated in subsection IV.C are met, results improve when the target is

moving compared to the case when it is stopped. This occurs because the aircraft approaches the naval platform from behind and when the latter is moving the relative velocity between the two is reduced giving the aircraft more time to maneuver. Moreover, the time spent at a small distance to the target is increased. This is important since the output of the EPnP algorithm tends to be more precise when the camera is closer to the target, and thus it is desirable to have large quantities of images taken at low distance. One of the steps which may benefit greatly from the gathering of experimental data is the outliers' rejection block used in the KF. Measurement outliers differ quite significantly from the typical measurement's behavior, and are thus extremely difficult to model. Therefore, it is quite possible that physical experiments will lead to the identification of types of outliers not considered in simulation.

ACKNOWLEDGMENT

This work was funded by the EU Commission within the National Strategic Reference Framework (QREN) under grant agreement 23260 (AUTOLAND).

REFERENCES

- [1] S. Saripalli, J. F. Montgomery and G. S. Sukhatme, "Visually Guided Landing of an Unmanned Aerial Vehicle," *IEEE Transactions on Robotics and Automation*, vol. 19, 2003.
- [2] K. Edwards, J. Archibald, W. Fife and D. J. Lee, "A Vision System for Precision MAV Targeted Landing," *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 125-130, 2007.
- [3] H. J. Kim, M. K. H. Lim, C. Park, S. Yoon, D. Lee, H. Choi, G. Oh and Y. Kim, "Fully Autonomous Vision-Based Net-Recovery Landing System for a Fixed-Wing UAV," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1320-1333, 2013.
- [4] S. Huh and D. H. Shim, "A vision-based landing system for small unmanned aerial using an airbag," *Control Engineering Practice*, pp. 812-823, 2010.
- [5] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation*, pp. 323-344, 1987.
- [6] R. Y. Tsai, "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 364-374, 1986.
- [7] V. Lepetit, F. Moreno-Noguer and P. Fua, "Epnp: An accurate O(n) solution to the pnp problem," *International journal of computer vision*, pp. 155-166, 2009.
- [8] E. Montijano and C. Sages, "Fast Pose Estimation for Visual Navigation Using Homographies," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2704-2709, 2009.
- [9] A. Ansac and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 578-589, 2003.
- [10] R. E. Kalman, "A New Approach To Linear Filtering and Prediction Problems," *Transaction of the ASME -- Journal of Basic Engineering*, vol. 82, pp. 35-45, 1960.
- [11] S. Park, J. Deyst and J. How, "A New Nonlinear Guidance Logic for Trajectory Tracking," *AIAA Guidance, Navigation and Control Conference*, 2004.
- [12] R. Curry, M. Lizarraga, B. Mairs and G. Elkaim, "L2+, an Improved Line of Sight Guidance Law for UAVs," *American Control Conference*, 2013.