Autonomous Landing of an Unmanned Aerial Vehicle on an Autonomous Marine Vehicle

T.K. Venugopalan
Nanyang Technological University,
Singapore.
venu0008@e.ntu.edu.sg

Tawfiq Taher
Singapore-MIT Alliance for Research and Technology,
Singapore.
tawfiq@smart.mit.edu

Prof. George Barbastathis
Massachusetts Institute of Technology
Cambridge MA, U.S.A.
gbarb@mit.edu

***Abstract*— In the recent past, there has been a lot of interest in developing UAVs (Unmanned Aerial Vehicles) to perform a variety of challenging tasks ranging from military defense, surveillance, environmental sensing, etc. This research is one that focuses on quadrotor UAVs deployed for environmental sensing in the oceans. We have developed an algorithm to autonomously control a quadrotor to track and land on the landing pad on a marine vehicle, an autonomous kayak. The algorithm takes up the challenge of tough landing conditions prevalent in the oceans due to winds and currents (causing the target to rock and drift). It has currently been developed for the commercially available AR Drone quadrotor. Landing pad sensing was achieved through image processing techniques using MATLAB. Testing has been carried out both indoors, and outdoors over open water, with a success rate of over 75%. This autonomous control algorithm for the quadrotor would enhance its operating region, preventing the need for it to fly back to the base station and thereby saving valuable flight time in far-off ocean deployments.**

***Index Terms* - UAVs (Unmanned Aerial Vehicles), quadrotor, AR Drone, Autonomous Surface Craft, Kayak.** *(Key words)*

I. Introduction

UAVs have undoubtedly shown their immense utility in numerous domains. As research in developing this adolescent field continues today, one of the recent interests is in autonomous landing of UAVs in various terrains and environments.

The objective of this research was to develop an algorithm to autonomously control a quadrotor to track and land on a marine vehicle such as an autonomous kayak [1]. This finds applicability in quadrotors deployed for environmental sensing in the ocean.

The purpose of developing such an algorithm is manifold. Typically, the flight time of a quadrotor ranges from 15-25 min. The autonomous landing feature eliminates the necessity for flying between the base station and the destination, which could have easily taken 5-10 min of the total time. Alternatively, the quadrotor can be transported using autonomous kayaks. Further, since the control is completely autonomous, it eliminates the need for manual landing of the quadrotor at the base station. The tracking algorithm with its robust state machine design can further be extended for applications such as surveillance of a marine vehicle to safeguard against oncoming ships, speeding boats, etc.

The commercially available AR Drone quadrotor by Parrot USA was chosen to test the algorithm. The AR Drone quadrotor is capable of autonomous take-off, landing, hover and performing pitching, rolling and yawing motions. All this can be controlled remotely by a smart phone /tablet (Android/iOS) through Wi-Fi. Necessary mobile-apps have been developed by Parrot USA. Further, it consists of two cameras, one vertical and one horizontal. This served as a great starting platform for the current research.

To accomplish the task of detecting the landing pad (through image processing) and for autonomously controlling the quadrotor, a GCS (Ground Control Station) was developed which communicates with the AR Drone through Wi-Fi. The GCS processes the video feed from the AR Drone and sends suitable control commands for proper positioning and landing. The design of the control and image processing algorithm along with the state machine design have been elaborated in the sections to follow.

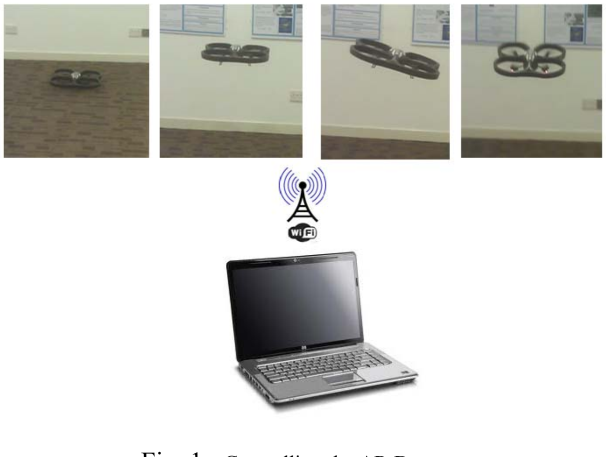


Fig. 1. Controlling the AR.Drone

978-1-4673-0831-1/12/$31.00 ©2012 IEEE

II. LITERATURE REVIEW

Extensive research has been done in autonomous landing of UAVs under various situations, environments and terrains. There are different approaches for autonomous flight controls [2]. In the research done by Theodore and Tischler, an autonomous landing control for a helicopter was developed using camera-image processing [3]. Voos implemented a non-linear control algorithm, specifically on quadrotors, to perform autonomous landing [4]. Additionally, the algorithm was tested with moving platforms as was the requirement for the current research.

In some research work, an autonomous landing control for a helicopter using a pan-tilt camera was developed [5] [6]. Further the system was tested under outdoor weather conditions and significant results were obtained. These researches were a great starting point for the current project.

Further, various methods of sensing the landing pad have been employed in the past, such as – Camera Image processing, IR-LED sensors, LASER+Retroreflectors , Tethering using a cord, Kinect Radar, GPS, etc. IR based sensing or Laser based sensing is more applicable for indoor flights due to ground reflections of sunlight that are prevalent outdoors. Among the other alternatives, camera based detection came out as one of the most convenient and cost-effective method with a reasonable accuracy and hence was adopted in the current research.

There are lots of research works on simulation of autonomous landing [7] [8]. Most of the autonomous landing algorithms are developed for indoor flights [9]. However, not much research has been done in for marine environments where strong wind exists and the landing pad drifts continuously. In the few researches that do exist in this field like [10], tethering is employed for landing which uses complicated and expensive sensors.

The current research explores into this challenging arena of autonomous landing of a UAV in a marine environment. Monocular vision based image processing has been incorporated in the sensing algorithm which is computationally light weight, cost-effective and accurate.

III. ALGORITHM DESIGN

The algorithm adopted consists of two parts namely – the Image processing for sensing the landing pad and the Control design for navigation.

A. Image Processing

The foremost and most challenging aspect of the research was to develop the image processing algorithm to detect the landing pad. Using this information, the control commands were developed to land the quadrotor accurately. The image processing code was written in MATLAB using the "Image Processing Toolbox" and integrated into the C-code for controlling the quadrotor. For integrating MATLAB into C, the MATLAB Engine Library was used.

1) *The Principle:* Image based detection of landing pad most often consists of one or a combination of the following-

- Shape Detection (Squares/Rectangles/ Circles)
- Color Detection
- Pattern Recognition (A composite shape)
- Image Recognition (By matching with a pre-registered photograph)

Pattern or Image recognition is more reliable for detecting the landing pad over simple shape or color detection. Especially in the context of deploying the quadrotor in the oceans, there is a need to prevent stray floating objects to be mistaken as landing pads. However, since the image processing is not performed onboard, pattern or image recognition can become computationally very intensive. Hence as a compromise between these approaches, landing pad detection was achieved through color recognition, along with pattern recognition (for a second level of authentication). This authentication happens only at the stage of initial detection of the landing pad since it needs the visibility of the entire pad (see figure 8). Once the descent routine is invoked, the authentication will no longer take place which also saves computation time.

2) *The Landing Pad:* The landing pad design incorporates both a pattern as well as distinctive colors which help its detection through image processing. The figure below shows the landing pad design used.

It consists of two concentric rectangles. The smaller rectangle is of fluorescent green color and the larger rectangle is of fluorescent red. The reason for such a choice is-

- The ocean prominently consists of the blue color. Hence Green and Red, the other primary colors, were chosen, since they do not contain any component of blue in them.
- Concentric rectangular pattern was chosen since they do not appear of a different shape when viewed at an oblique angle by the camera (Unlike circles which appear as an ellipse and squares which appear as a rectangle when viewed obliquely).

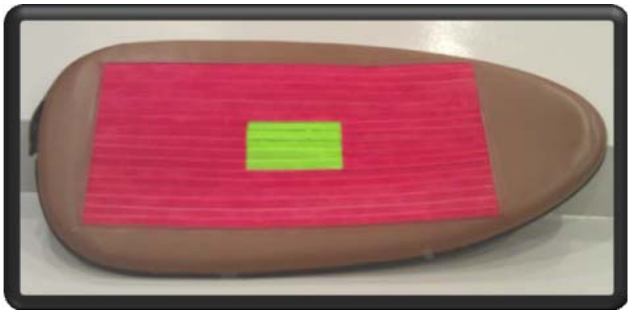
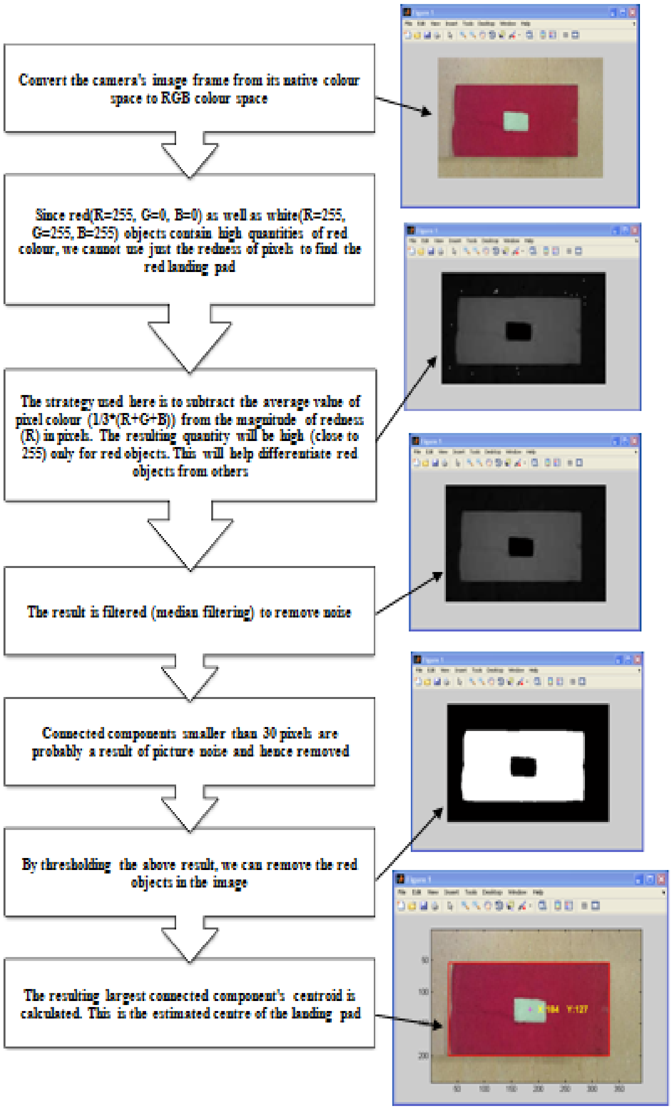


Fig. 2. Landing Pad (Dimensions - 120cm by 40cm)

- The larger rectangle is easily visible even at altitudes of 5 m, and would help guide the quadrotor at high altitudes, while the smaller one would help guide it precisely to the centre of the pad, just before landing.

3) Color Based Detection: The strategy for detecting the fluorescent red color is as follows. The strategy for fluorescent green is similar to illustration below.


Convert the camera's image frame from its native colour space to RGB colour space

Since red(R=255, G=0, B=0) as well as white(R=255, G=255, B=255) objects contain high quantities of red colour, we cannot use just the redness of pixels to find the red landing pad

The strategy used here is to subtract the average value of pixel colour (1/3*(R+G+B)) from the magnitude of redness (R) in pixels. The resulting quantity will be high (close to 255) only for red objects. This will help differentiate red objects from others

The result is filtered (median filtering) to remove noise

Connected components smaller than 30 pixels are probably a result of picture noise and hence removed

By thresholding the above result, we can remove the red objects in the image

The resulting largest connected component's centroid is calculated. This is the estimated centre of the landing pad

Fig. 3. Color detection process flow in MATLAB

From the above process flow, we can estimate the centre of the landing pad. The distance from the centre of the pad from the centre of the image frame will form the error feedback for the controller is then to minimize. The task of the controller then becomes to minimize this error and keep the landing pad always at the centre of the frame.

4) *Pattern Detection:* The pattern to be recognized is the concentric rectangles in the landing pad. This is to authenticate the landing pad and ensure the image processing has not picked up some stray floating objects in the ocean.

Once the color detection algorithm picks out the red and the green parts of the landing pad as separate objects, we need to ensure the object's shape is a rectangle. This can be verified through a set of criteria like the perimeter (in pixel^2) must be close to its length (px)*breadth (px), a rectangle drawn around the extremities of the object would be occupied almost entirely of the object's pixels, etc. Further, the centroids of the two concentric rectangles must match. However, there are a couple of challenging issues with this approach as illustrated in figure 4. Details of how these problems were countered are provided below.

a) Skew Correction: The skew in the image due to oblique viewing was eliminated by finding the orientation of the landing pad. Since the landing pad is a rectangle, it has a major axis and a minor axis. The angle between the major axis and the horizontal can be used to know by what angle the image needs to be rotated to align the rectangle. Once this was done, deciphering whether the object is a rectangle or not, was simplified.

b) Brightness Correction: Due to shadows/non-uniform brightness, the shape of the landing pad gets distorted and hence the pattern recognition fails. To prevent this, the strategy followed is represented in figure 5. This transformation applied to a sample is shown in figure 6.

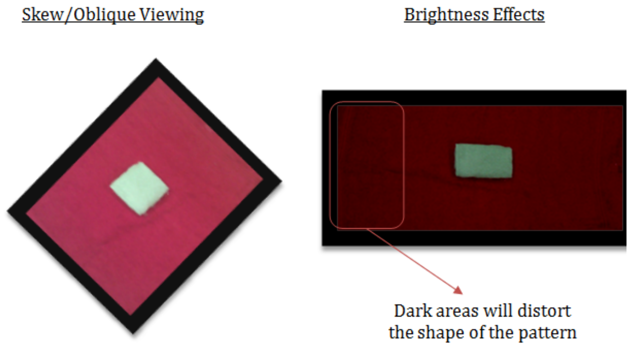


Fig. 4. Problems in recognizing the pattern

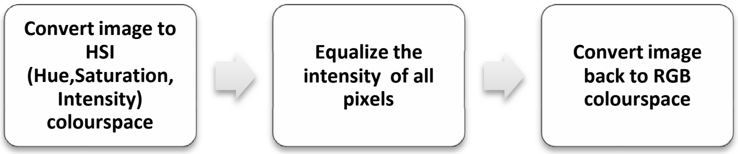
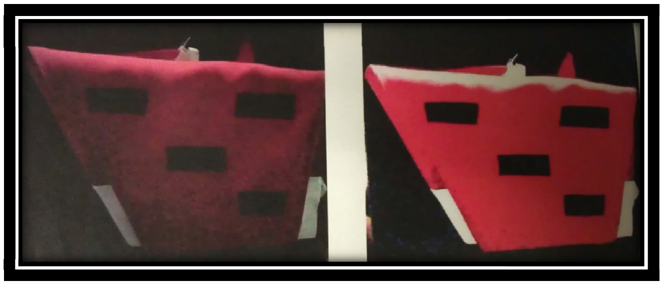

Convert image to HSI (Hue,Saturation, Intensity) colourspace
Equalize the intensity of all pixels
Convert image back to RGB colourspace

Fig. 5. Brightness Correction Strategy

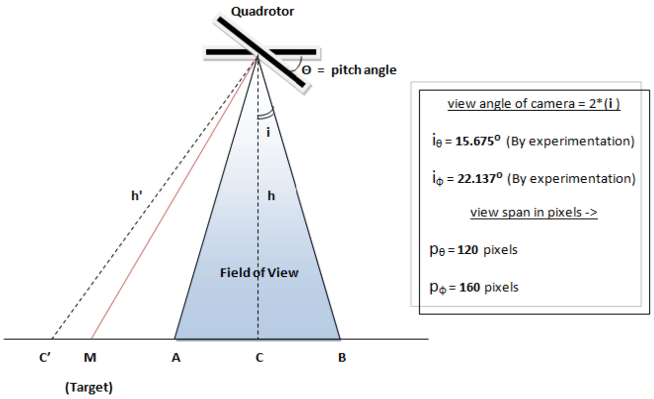


Before Correction After Correction

Fig. 6. Brightness Correction applied to a sample

5) *Camera Correction:* The final but critical aspect of the image processing is the camera correction. Taking a scenario when the quadrotor is tilted but the landing pad appears to be at the centre of the camera's image, the image processing will assume the quadrotor is directly above the landing pad. However, since this is not true, it is required to eliminate the wrongly calculated position of the landing pad when the image is captured obliquely by the camera.

This calculation for pitch correction is explained in figure 7. The final calculated value of 'MC' is fed back by the image processing as the y position error to the controller. The roll correction can be assumed to be similar, yielding the corrected value of x position error.



$MC = ?$

$MC = h*\tan(\theta - x)$

$= CC' - MC'$

$$= \boxed{h' * sin(\theta) - \frac{C'M(\text{in pixels})}{120} * h' * cos(\theta)[tan(\theta + i) - tan(\theta - i)]}$$

(a) $MC' = \frac{C'M(\text{in pixels})}{120} * h[tan(\theta + i) - tan(\theta - i)]$

(b) $h = h' * cos(\theta)$

(c) $CC' = h' * sin(\theta)$

Fig. 7. Camera Correction for pitch induced error

B. Control Design

1) *State Machine Approach:* The state machine design developed to achieve the objective of landing the quadrotor is shown in figure 8.

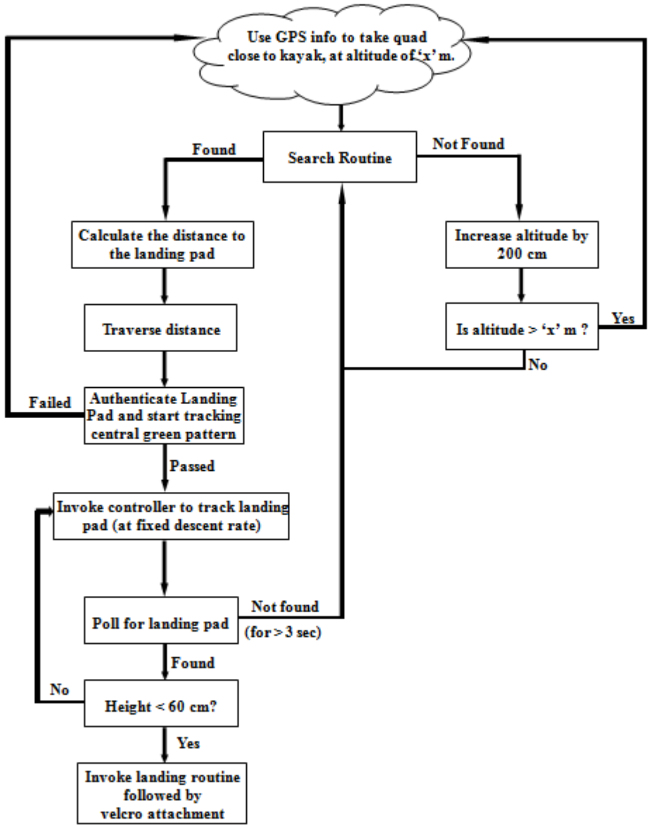

Use GPS info to take quad close to kayak, at altitude of 'x' m.
Search Routine
Found
Not Found
Calculate the distance to the landing pad
Increase altitude by 200 cm
Traverse distance
Is altitude > 'x' m ?
Yes
No
Failed
Authenticate Landing Pad and start tracking central green pattern
Passed
Invoke controller to track landing pad (at fixed descent rate)
Poll for landing pad
Not found
(for > 3 sec)
Found
No
Height < 60 cm?
Yes
Invoke landing routine followed by velcro attachment

Fig. 8. State Machine Design

This state machine design gives the overall picture of the control design, for the autonomous landing. The sections of the state machine such as the – Search routine, Control and Landing Routine have been elaborated in the sections to follow.

2) *Software Architecture:* The software architecture and the process flow loop used for communicating with the quadrotor are depicted in figures 9 and 10:

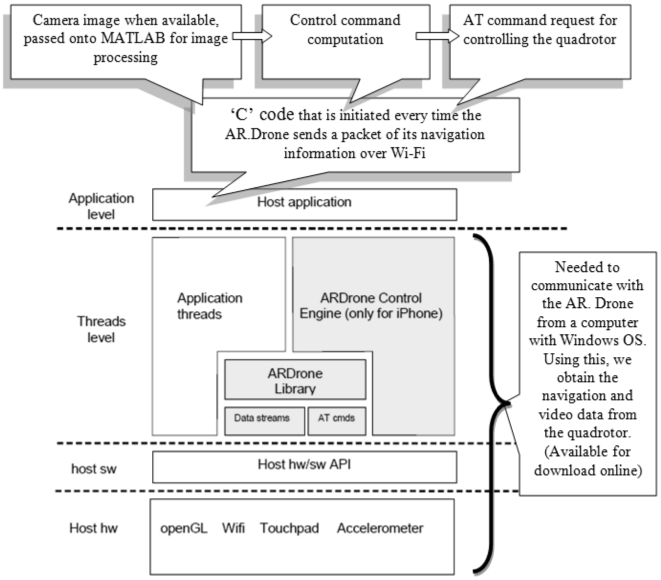

Camera image when available, passed onto MATLAB for image processing
Control command computation
AT command request for controlling the quadrotor
'C' code that is initiated every time the AR.Drone sends a packet of its navigation information over Wi-Fi
Application level
Host application
Threads level
Application threads
ARDrone Control Engine (only for iPhone)
ARDrone Library
Data streams
AT cmds
host sw
Host hw/sw API
Host hw
openGL
Wifi
Touchpad
Accelerometer
Needed to communicate with the AR. Drone from a computer with Windows OS. Using this, we obtain the navigation and video data from the quadrotor. (Available for download online)

Fig. 9. Software Architecture

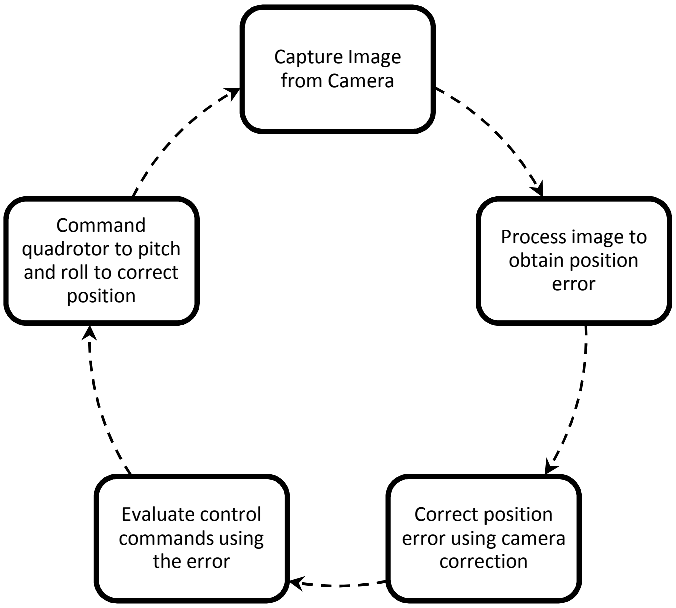

Capture Image from Camera
Process image to obtain position error
Correct position error using camera correction
Evaluate control commands using the error
Command quadrotor to pitch and roll to correct position

Fig. 10. Process Flow Loop

3) *Control Loop:* The control loop is a modified version of the traditional PID (Proportional-Integral-Derivative) control. That is, the two control inputs namely- pitch and roll, which help change the (x,y) position of the quadrotor, is generated using the formula:

$$roll = K_{p1} * x_{error} + K_{d1} * \dot{x}_{error} + K_{i1} * \int_0^t x_{error} \qquad (1)$$

$$pitch = K_{p2} * y_{error} + K_{d2} * \dot{y}_{error} + K_{i2} * \int_0^t y_{error} \qquad (2)$$

$^*x_{error}$, y_{error} are in the body axis

These PID gains were estimated using the Ziegler-Nichols tuning method and further fine-tuning was done based on trial-and-error. Though this PID control worked very well in controlling the quadrotor indoors, in the presence of the wind, it led to highly unstable performance. Further, it exhibited "hunting" around the centre of the landing pad due to the fact that we are trying to minimize two state variables – x position and y position, simultaneously. Hence the direction of motion of the quadrotor at a given point in time will not necessarily be towards the centre of the landing pad but rather towards that direction that best minimizes both the control inputs – roll and pitch.

Due to this, the inbuilt stabilizing algorithm of AR Drone has been put to good use. The "hover" mode available in the AR Drone uses the gyros as well as internal vision-based optical flow estimation to accurately keep the quadrotor stationary at a point. We made use of this mode to serve as the inner loop of the control to stabilize against wind and keep the quadrotor always in vicinity of the target. In the outer loop, we periodically use the computed PID control commands to "nudge" the quadrotor towards the centre of the landing pad. This led to a far better performance outdoors and prevented hunting for landing pad. Figure 11 summarizes this control algorithm.

However, with the same control input, the displacement of the quadrotor due to nudging can vary depending on the wind direction. That is, if the nudging is in the direction of the wind, the displacement is larger and vice-versa. Hence, for a smoother performance, the PID gains were made to vary automatically around the values found earlier, to offset the effect of wind. So in a sense, the gains were made adaptive in nature.

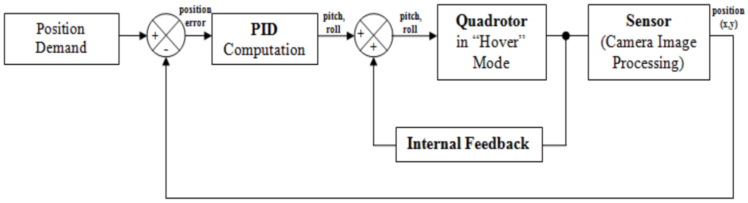


Fig. 11. Control Loop

4) *Search Routine:* A novel aspect of the control design is the Search Routine. The bottom camera of the AR Drone has only a 63□ diagonal view. Since it does not have a pan-tilt facility, the field of view is restricted.

To enlarge this field of view, we perform a circular tilting of the camera around the vertical. This was done through a sinusoidal variation of pitch and a 90□ phase-shifted sinusoidal variation of roll. The increase in the field of view is equal to the amplitude (in °) of roll-pitch sinusoid. Further, due to symmetric tilting, the quadrotor does not get displaced from the initial position. Figure 12 describes this search routine.

As a result of this, the field of view increased by 160-200%. A MATLAB graph of the pitch and roll commands during a test flight is shown in Figure 13.

5) *Landing Routine:* Once the quadrotor is positioned near the landing pad within a height of 60 cm, a pure PID based control takes over and positions the quadrotor while it is about to land. This ensures fast response to position errors. Some of the challenges in achieving this landing routine were:

- The closer the landing pad, the more precise the positioning needs to be. Else, the landing pad would easily be lost from the camera frame.
- Wind and motion of the landing pad during landing will further enhance the difficulty.

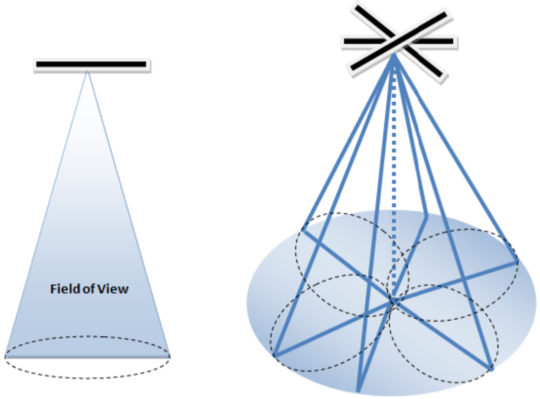


Fig. 12. Search Routine to increase the field of view

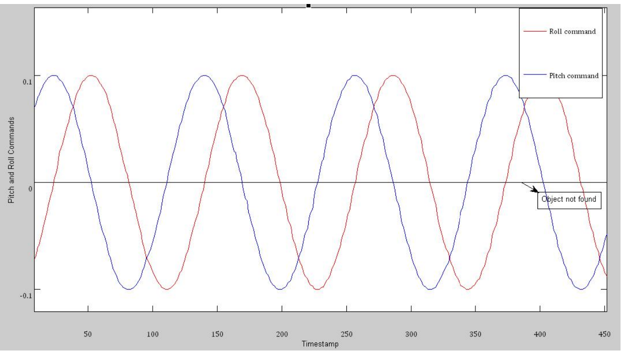


Fig. 13. Search Routine Commands when the object is not found

Most often, the landing pad's centre is lost from the camera frame due to the wind or the drifting of the landing pad. This disables the feedback for estimating the position error. At any moment during the final landing phase, if the landing pad is lost from the camera frame, the controller uses the previous history of the landing pad position to guide the quadrotor.

The PID gains had to be accurately chosen for the landing routine to prevent overshoot as well as to have a fast response. Once the quadrotor lands, the Velcro on the quadrotor's legs gets attached to the landing pad and holds it in place.

IV. Testing & Results

A. Indoor Flight Tests

To test whether the algorithm developed was capable of performing well in marine conditions, it was first tested indoors under 3 scenarios separately –

- No Disturbance
- Wind Disturbance (from a portable fan)
- Moving Landing Pad with Oscillatory motions (to simulate drifting due to ocean tides and oscillatory motions due to rocking)

1) *No Disturbance:* The landing algorithm was first tested indoors without any disturbance.

Some noteworthy observations from the tests were –

- The Image Processing was able to detect the landing pad accurately with a frame rate of about 20 fps.
- Artificial lights, if too bright, resulted in the landing pad's colors to appear nearly white. Similarly, if too dim, the colors appeared too dull to be picked up by the image processing. Hence the brightness of the lights needed to be moderated. However, no such problem occurred under natural sunlight.

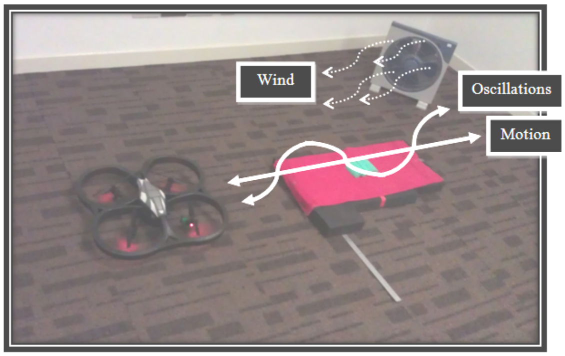


Fig. 14. Indoor Flight Testing

- The control algorithm proved to be rugged and ensured the quadrotor was always in the vicinity of the target.
- Overshoots in positioning were minimal and total time to land was low (< 0.5 min).
- The algorithm ensured the landing pad was always visible and close to the centre of the camera frame. The accuracy of landing was within ±12 cm from the centre of the pad.

The result of the repeated tests is summarized in Table I.

2) Wind Disturbance: Wind disturbance from a portable fan was used to simulate windy conditions indoors. In the vicinity of the landing pad, a mild windy condition was set up to assess performance.

Some noteworthy observations from the tests were –

- The control algorithm proved to be rugged in spite of the wind and ensured that the quadrotor is always in the vicinity of the target.
- Overshoots in positioning the quadrotor increased significantly. Hence the time required for landing was high.
- When the quadrotor was in the process of touchdown (<50 cm from ground), it often lost the landing pad from the frame and depended more on the history of the landing pad position to guide the quadrotor. Nevertheless, the accuracy of landing was within ±15 cm from the pad.

The result of the repeated tests is summarized in Table II.

TABLE I. FLIGHT TEST RESULTS WITH NO DISTURBANCE

With No Disturbance		
Average Landing time **(min)**	**Average Accuracy** **(cm)**	**% Success**
0.5 min	± 12 cm from centre of pad	85 – 90 %

TABLE II. FLIGHT TEST RESULTS WITH WIND DISTURBANCE

Wind Disturbance		
Average Landing time **(min)**	**Average Accuracy** **(cm)**	**% Success**
1.0 min	± 15 cm from centre of pad	80 – 85 %

3) Moving Landing Pad: While trying to land the quadrotor on a kayak deployed in the ocean, the kayak would drift due to ocean tides, making it difficult to land. Also, it would exhibit rolling and pitching motions in water. To simulate this, the landing pad was moved around manually at < 1 m/s velocity and the motion was made jerky similar to rocking in water.

Some noteworthy observations from the tests were –

- The quadrotor was able to follow the landing pad smoothly as long as the motion was within 1 m/s.
- Whenever the target was missed, the search routine was quick to recapture the position of the landing pad.
- Oscillations in landing pad position did not affect the control and ensured stability at all times.
- However, when the landing pad changed its direction of motion abruptly, the quadrotor seemed to take a long time overcoming the inertia of its previous motion. Hence, it often missed the target and needed to search for it again.
- The landing routine proved good in following the target whilst it was moving, and landed the quadrotor accurately. However, sudden large displacements of the landing pad when the quadrotor is below 30 cm from the ground used to take the inner green pattern of landing pad outside the camera's field of view and hence resulted in bad landings.

The result of the repeated tests is summarized in Table III.

4) *Flight Test Results in MATLAB:* To illustrate the control algorithm's commands during flight, the MATLAB generated graphs in figure 15 and figure 16 show the control commands during various stages of a flight test.

TABLE III. FLIGHT TEST RESULTS WITH A MOVING LANDING PAD

Moving Landing Pad		
Average Landing time **(min)**	**Average Accuracy** **(cm)**	**% Success**
1.5 min	± 20 cm from centre of pad	75 – 80 %

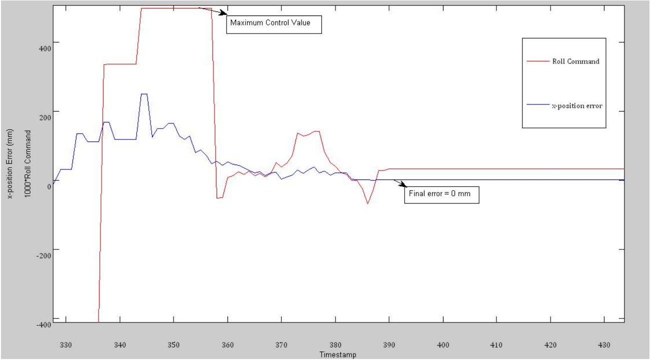


Fig. 15. X-position error correction just before landing

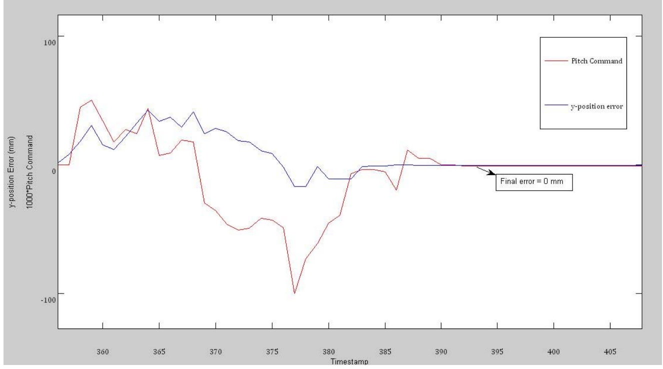


Fig. 16. Y-position error correction just before landing

B. Outdoor Flight Tests

Following the indoor flight tests, the control algorithm was tested outdoors over open water, at a reservoir. The kayak was positioned 20 m from the shore, from where the quadrotor was launched. Moderate wind and mild choppy waves prevailed during testing. The quadrotor successfully landed on the autonomous kayak during the trial conducted, with an accuracy of ±15 cm, confirming the ability of the algorithm to take the challenges of the marine environment. This was achieved with an average landing time of close to 1 min. The trial was repeated to verify the performance and resulted in an almost identical tracking and landing accuracy. Videos of the successful landing were also taken to augment this research. Figures 18, 19 and 20 are photos taken during the flight testing.

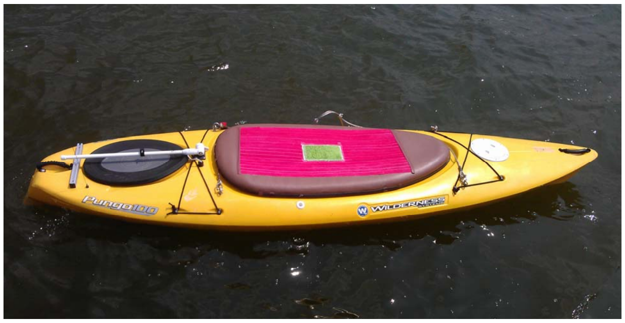


Fig. 17. Autonomous Kayak with the landing pad

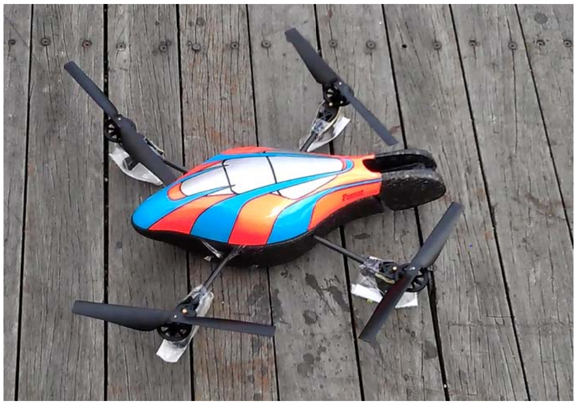
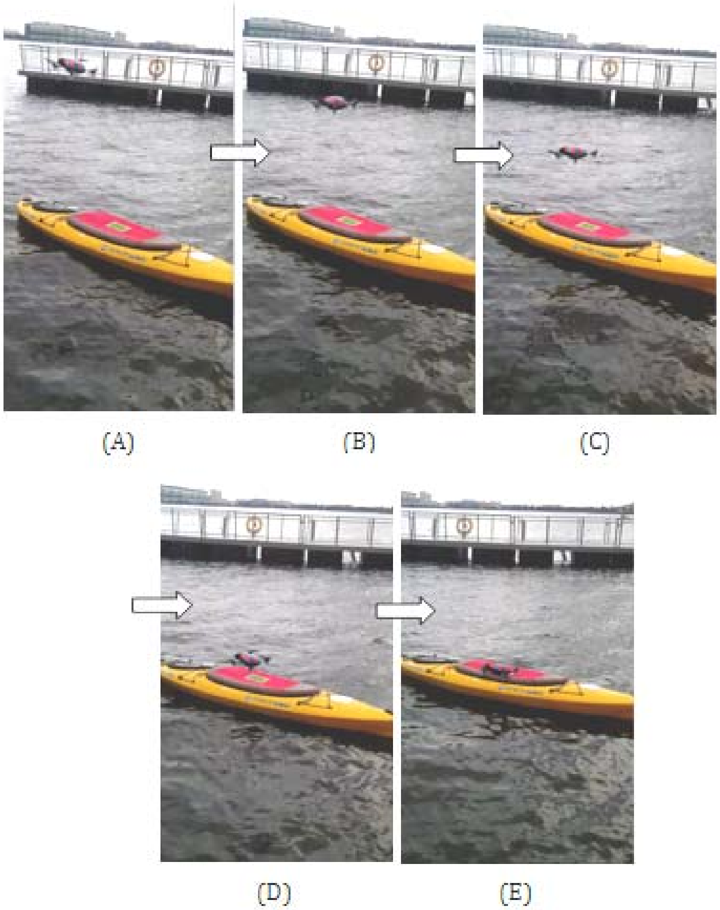


Fig. 18. Quadrotor used for the testing



V. Conclusion

The objective of this research was to develop an algorithm, to autonomously land a quadrotor on an autonomous marine vehicle. The AR Drone quadrotor was opted as the test vehicle. A control algorithm for this quadrotor was developed for performing the autonomous landing. To complement this, a Search Routine, Landing Routine and a State Machine approach were developed. Using MATLAB, the image processing module required to detect the landing pad was created and incorporated into the control algorithm. Further, camera correction, skew correction and brightness correction codes were designed to solve the implementation issues faced.

The preliminary stages of testing were conducted indoors. A simulation of the marine environment was achieved by replicating wind disturbance by means of a portable fan and effects of sea currents by a moving landing pad with oscillatory motions. Results of the testing were satisfactory with a worst case landing accuracy of ± 20 cm from centre of landing pad and a % success of 75-80 %.

Final testing was done outdoors over open water with moderate wind and water ripple conditions. The flight tests resulted in successful landings with an accuracy of ±15 cm, videos of which have been captured.

The current system utilizes Wi-Fi for communicating with the quadrotor. Hence, Wi-Fi induced noise due to interference and Wi-Fi lags degrade the performance. Hence it is believed that an on-board version of the GCS may improve performance. The quadrotor used for this work is a very basic vehicle and not suitable for high end processing and research. Research class quadrotors are capable of hovering and navigating more precisely even in presence of strong wind. So, if this algorithm is implemented in such quadrotors, it may improve the performance further more.

Acknowledgment

The research described in this WAVES (Water and Air Vehicles for Environmental Sensing) Lab project was funded in whole by the Singapore National Research foundation (NRF) through the Center for Environmental Sensing and Modeling (CENSAM) at Singapore-MIT Alliance for Research and Technology (SMART) Centre.

Reference

[1] J. Curcio, J. Leonard, A. Patrikalakis, “SCOUT – A Low Cost Autonomous Surface Craft for Research in Cooperative Autonomy”, in Proceedings of MTS/IEEE Oceans 2005, pages 725-729 vol. 1, September 2005

[2] J.-B. Brachet, R. Cleaz, A. Denis, A. Diedrich, D. King, P. Mitchell, D. Morales, J. Onnée, T. Robinson, O. Toupet, B. Wong., “VII . Autonomous Flight,” pp. 72-196. http://ocw.mit.edu/

[3] C. T. Theodore and M. B. Tischler, “Precision Autonomous Landing Adaptive Control Experiment (PALACE)”, 25th Army Science Conference, Orlando, FL, November 2006

[4] H. Voos, “Nonlinear tracking and landing controller for quadrotor aerial robots.”, 2010 IEEE International Conference on Control Applications (CCA), pp. 2136- 2141, September 2010

[5] S. P. Soundararaj, A. K. Sujeeth, and A. Saxena, “Autonomous indoor helicopter flight using a single onboard camera,” 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5307-5314, October 2009.

[6] T. Merz, S. Duranti, and G. Conte, “Autonomous Landing of an Unmanned Helicopter based on Vision and Inertial Sensing,” In Proc. of the 9th International Symposium on Experimental Robotics, Singapore, June 2004

[7] I. Zohar, A. Ailon, and H. Guterman, “An automatic stabilization system for quadrotors with applications to vertical take-off and landing.” AUVSI-IEEE Joint Conference, Israel, October 2010

[8] H. Voos and B. Nourghassemi, “Nonlinear Control of Stabilized Flight and Landing for Quadrotor UAVs,” 7th Workshop on Advanced Control and Diagnosis, Zielona Góra, Poland, November 2009.

[9] K. E. Wenzel, A. Masselli, and A. Zell, “Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle,” Journal of Intelligent & Robotic Systems, vol. 61, no. 1–4, pp. 221-238, October 2010.

[10] Oh, S.-R. O. S.-R., Pathak, K., Agrawal, S. K., Pota, H. R., & Garratt, M., “Approaches for a tether-guided landing of an autonomous helicopter.”, IEEE Transactions on Robotics (Vol. 22, pp. 536-544). Volume: 22 , Issue: 3, June 2006