Machine Learning Vision and Nonlinear Control Approach for Autonomous Ship Landing of Vertical Flight Aircraft

Bochan Lee
Graduate Research Assistant
Texas A&M University
College Station, TX, USA

Vishnu Saj
Graduate Student
Texas A&M University
College Station, TX, USA

Moble Benedict
Associate Professor
Texas A&M University
College Station, TX, USA

ABSTRACT

The paper discusses a machine learning vision and nonlinear control approach for autonomous ship landing of vertical flight aircraft without utilizing GPS signal. The central idea involves automating the Navy helicopter ship landing procedure where the pilot utilizes the ship as the visual reference for long-range tracking, but refers to a standardized visual cue installed on most Navy ships called the ”horizon bar” for the final approach and landing phases. This idea is implemented using a uniquely designed nonlinear controller integrated with machine vision. The vision system utilizes machine learning based object detection for long-range ship tracking, and classical computer vision for object detection and the estimation of aircraft relative position and orientation during the final approach and landing phases. The nonlinear controller operates based on the information estimated by the vision system and has demonstrated robust tracking performance even in the presence of uncertainties. The developed autonomous ship landing system is implemented on a quad-rotor vertical take-off and landing (VTOL) capable unmanned aerial vehicle (UAV) equipped with an onboard camera and was demonstrated on a moving deck, which imitates realistic ship deck motions using a Stewart platform and a visual cue equivalent to the horizon bar. Extensive simulations and flight tests are conducted to demonstrate vertical landing safety, tracking capability, and landing accuracy while the deck is in motion.

NOTATION

A	Camera intrinsic matrix
a	Long-range controller constant
b	Close-range controller constant
c	Setpoint
CE	Current yaw estimate
CM	Current yaw measurement
c_u	Corner position-column
c_v	Corner position-row
d	Long-range controller constant
$de(t_k)$	Error difference between time t_k and t_{k-1}
dt_k	Time difference between time t_k and t_{k-1}
$e(t_k)$	Error at time t_k
$f(x)$	Probability density function
f_x	Focal length of camera in horizontal direction
f_y	Focal length of camera in vertical direction
$h(t_k)$	Height of bounding rectangle at time t_k
KG	Kalman gain
K_D	Derivative gain
K_I	Integral gain
K_P	Proportional gain
m	Long-range controller constant
O_{1x3}	1x3 zero matrix
PE	Previous yaw estimate
QN	Measurement noise covariance
R	Rotation matrix
$r(t_k)$	Relative position at time t_k
R_{basic}	Basic rotation matrix
$R_{modified}$	Modified rotation matrix
RN	Process noise covariance
s	Scaling factor
sw	Window size
t	Translation vector
u	Image pixel position-column
$u(t_k)$	Control law
u_0	Center of image-column
v	Image pixel position-row
v_0	Center of image-row
$w(t_k)$	Width of bounding rectangle at time t_k
X	Sideward relative distance, meter
Y	Vertical relative distance, meter
Z	Forward relative distance, meter
α	Relative yaw angle, rad
β	Relative pitch angle, rad
γ	Relative roll angle, rad
θ	Aircraft pitch angle, deg
μ	Mean value
ρ_u	Parameter to convert pixels to SI units-column
ρ_v	Parameter to convert pixels to SI units-row
σ	Standard deviation
ϕ	Aircraft roll angle, deg
ψ	Aircraft yaw angle, deg
$\nabla f(u,v)$	Image gradient

Presented at the Vertical Flight Society’s 77th Annual Forum & Technology Display, Virtual, May 10–14, 2021. Copyright © 2021 by the Vertical Flight Society. All rights reserved.

INTRODUCTION

Landing a helicopter on a small ship at rough sea states is an extremely challenging task even for human pilots due to the small landing space, six degrees of freedom ship deck motions, limited visual references for pilots, and lack of alternative landing spots. There have been many studies in the past that focused on automating helicopter ship landing by utilizing a wide array of sensors such as GPS, vision sensors, motion sensors, LIDAR, etc. This paper investigates a novel solution that falls under the category of a vision-based control system that does not use GPS signals and thus ensures its functionality in GPS-denied/-spoofed environments.

Previous efforts toward autonomous ship landing involve a common process that is to estimate or track ship deck motions first, and then control the aircraft attitude to match the ship motions for landing. In order to extract the ship deck motion information, various methods have been introduced such as tracking H landing marking (Ref. 1), T landing marking (Ref. 2), points dispersed on the deck (Ref. 3), lights (Ref. 4), and infrared cooperated targets on a ship (Refs. 5, 6). These vision-based methods have shown the limited capability for autonomous UAVs landing on ships undergoing significant motions representative of rough sea states. Fundamentally, a method that involves visual tracking of deck motion is not ideal for VTOL UAVs that approach a ship horizontally at low altitudes because its application range is limited to the vertical space where the deck can be captured. Also, actively controlling the UAV to match the complex deck motions could excite unstable UAV attitude dynamics. This is even more unsafe when the aircraft is in close proximity to the moving deck because even a small control error can cause a catastrophic accident due to an impact by the deck. Furthermore, none of the previous methods were based on the Navy helicopter ship landing procedure that was established and successfully executed by pilots for decades. In fact, visually tracking the moving landing deck is exactly the opposite of what Navy helicopter pilots are trained to do.

The present landing method is developed based on an in-depth understanding of the Navy helicopter ship landing procedure, which is discussed in Refs. (Refs. 7, 8). Contrary to intuition, Navy pilots are trained not to follow ship deck motions for two main reasons. First, spatial disorientation can occur when a pilot has no fixed, visible horizon to refer to, which is a critical element for maintaining a proper sense of helicopter attitude independent of ship motions. The key visual aid that helps pilots to land safely is a ”horizon reference bar” shown in Fig. 1, which is gyro-stabilized to indicate a perfect horizon regardless of ship motions and is widely used in most modern Navies (Refs. 9, 10). Thus, pilots can land a helicopter by referencing the horizon bar without responding to ship motions.

Second, constantly changing the helicopter attitude to match ship deck motions can trigger unstable helicopter dynamics introducing serious potential hazards. Hence, a pilot tries to control the helicopter in a stable manner independent of the ship's roll and pitch motions, and then lands vertically. It is also recommended to land quickly in order to prevent the ship deck from impacting one of the helicopter landing gear skids/wheels causing a rollover. This vertical landing manner is proven to be safe within the operating limits of the helicopter and currently being used in practice. The present technical approach towards automating such a landing procedure consists of machine vision to obtain the relative position and heading of the aircraft and a control system to execute the approach and landing maneuvers.

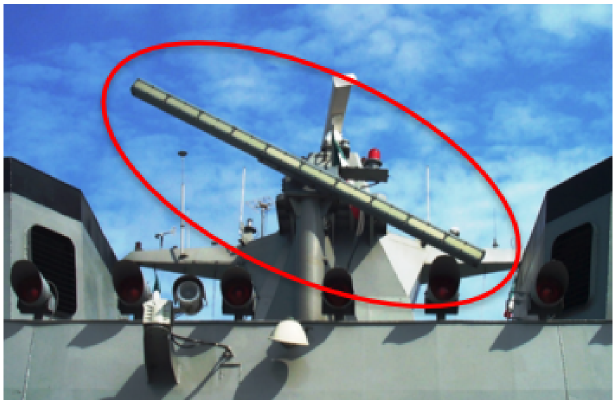

Figure 1. Horizon reference bar

The vision system is hybrid in nature with two different methods, a machine learning object detection and a classical computer vision method, each of which is designed to operate depending on the relative distance to the landing pad. In the long-distance, the machine learning object detection method is applied to identify the landing platform (the ship), and an image-based control is utilized in the autonomous flight control system. In recent years, there have been many studies to develop algorithms that guarantee fast detection as well as higher accuracy, which are essential to reliable UAV operations. Various algorithms and architectures of Convolutional Neural Network (CNN), a class of deep neural networks, have been proposed such as Region-based CNN (R-CNN) (Refs. 11–13), Single Shot Detector (SSD) (Ref. 14), and You Look Only Once (YOLO) (Refs. 15–17). R-CNN is classified as a two-stage detector that combines region-proposal algorithms with CNN to extract 2,000 regions via a selective search, then classifies the selected regions on the image. Later, its variant Faster R-CNN is introduced to improve the detection speed by replacing the slow selective region search process. Meanwhile, SSD and YOLO are classified as a one-stage detector that regards object detection as a regression problem by taking an input image and simultaneously learning the probability of an object class and bounding box coordinates. According to the studies that compared the state-of-the-art algorithms, YOLOv3 demonstrated faster detection performance than Faster R-CNN and SSD (Refs. 17, 18). Hence, the YOLOv3 algorithm is selected to train an object detector that is able to detect a ship and a horizon bar in real-time. It is not only visually tracking the object but that information is relayed in real-time to the autonomous flight control system.

Once the object is detected, it provides the object position and its bounding box in the image to the autonomous flight control system. Even though the actual relative distances are not estimated, the size of the object and its position in the image are sufficient information to control a UAV to approach the ship from a long distance. The verified maximum range is approximately 250 meters (820 feet) when the object occupies an area of 6 x 6 feet. Considering the range is proportional to the object's occupying area in the image, a typical small ship where the rear-side occupies 50 x 50 feet area can be detected from 17.3 kilometers (9.3 nautical miles) away.

On the other hand, obtaining accurate relative position and orientation from a captured image is crucial for the final approach and even more important for precise landing on the ship deck. Instead of detecting an object as a whole using machine learning, computer vision techniques are applied to extract particular points of interest. To name a few, edge and line detection (Refs. 19, 20), corner detection (Refs. 21, 22), and contour detection (Ref. 23) are previously applied to aircraft landing applications. In the present system, the visual cue to track is a horizon bar that has a rectangular shape and green color. To detect distinctive points on the bar, multiple processes such as image filtering, contour detection, corner points detection, and screening are conducted in this order. From the detected points in the 2D image, UAV relative positions and orientations are estimated using PnP (Perspective-n-Point) method. The accuracy of the present vision system to sub-centimeter and sub-degree levels have been previously demonstrated by the authors (Ref. 24).

By using the information provided by the vision system, the flight controller manipulates the UAV to approach and land. To this end, various control systems that uses vision sensors without GPS have been investigated in the literature, such as proportional-derivative (PD) control (Refs. 4, 25), a proportional-integral-derivative (PID) control (Refs. 3, 26), gain-scheduled PID control (Ref. 24), linear quadratic regulator (LQR) (Refs. 27–29), adaptive control (Refs. 30–32), discrete-time nonlinear model predictive control (Ref. 33), and reinforcement learning based control (Ref. 34). In the present system, a nonlinear control system with the Kalman filter is uniquely designed to operate accurately and robustly in the presence of time delays and sensor noise. The Kalman filter reduces the noise in estimation, however, small noise can be amplified when incorporated into the derivative controller due to the numerical differentiation process. A novel nonlinear controller is developed on top of the Kalman estimator to prevent the controller from responding to unrealistic estimations (or large fluctuations). It multiplies the estimation difference by the probability of its occurrence that follows a normal distribution. In this manner, the controller probabilistically perceives if the estimation is physically possible or not and then determines how to respond with a control input.

To demonstrate the safety of the vertical landing maneuver, which in this case is independent of the ship motions, realistic ship motions are implemented on a six degrees of freedom (DOF) motion platform. The first ship motion case is from the Oliver Hazard Perry Class FFG Frigate which is a small ship with a single landing deck. The ship motions at the sea state of 6 and the wave direction of 60° are scaled down for the 4ft x 4ft platform and are provided in (Ref. 1), which are also similar to measured ship motion data presented in (Ref. 35). The second ship motion case is the FFG 7 Class ship motion limits which are 3° of pitch and 8° of roll as defined in the Naval Air Training and Operating Procedures Standardization (NATOPS) (Ref. 36). The period of pitch and roll motions are selected as 10.1 secs and 6.5 seconds according to the study of typical small ship motions conducted by the Sandia National Lab (Ref. 37). Vertical landings are conducted at random instances of deck motions.

The developed autonomous system is systemically verified in every aspect through extensive simulations and flight tests. A platform is constructed to reproduce ship structure and deck motions using a 6 DOF Stewart platform. A quad-rotor-UAV Parrot ANAFI live streams a video to a base-station computer, which processes the image frames to transmit control inputs through WIFI. Indoor and outdoor flight tests demonstrated the robust autonomous tracking capability and the safety of vertical landing.

The following are the key contributions of this paper.

- Automate Navy helicopter ship landing procedure for UAVs.
- Develop and integrate a state-of-the-art machine learning based vision system with an autonomous flight controller for stable long-range tracking.
- Demonstrate a fast and reliable method to extract points of interest from an image by combining classical computer vision and screening algorithms.
- Develop a novel nonlinear controller along with a probabilistic algorithm to prevent large incorrect control inputs due to non-physical estimations to enable robust and smooth tracking.
- Demonstrate the accuracy and safety of vertical landing with scaled deck motion corresponding to a small ship at a sea state of 6.

The outline of the paper is as follows: In Section 2, details about the development of machine vision systems are described. In section 3, the nonlinear control systems are detailed and the performances are compared to other candidate controllers. The results of long-range tracking and vertical landing tests are presented in section 4 followed by the conclusions inferred from this study.

MACHINE VISION

The objective of the vision system is to provide the visual information required in a given situation. Hence, the development begins by understanding how a helicopter pilot perceives and acts during different situations while approaching and landing on a ship. First, the pilot visually confirms the ship's location from a long distance and then sets a course and speed for the approach. Second, once reaching close proximity of the ship, the horizon reference bar becomes visible, and from that point, the pilot stabilizes the helicopter by referring to the horizon bar, which remains horizontal independent of the ship motions, for a safe landing. In the present vision system, the same strategy is automated for VTOL UAVs by taking advantage of state-of-the-art machine learning object detection and classical computer vision methods.

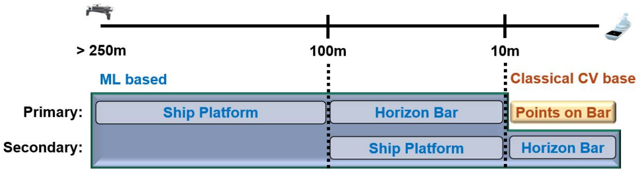

Figure 2. Tracking object depending on distance

As shown in Fig. 2, the ship, horizon bar, and corner points on the bar are used to perceive pose information depending on the distance. Note that it is only possible to detect the corner points at close proximity to the landing pad. The ship and horizon bar detections are used in the case when the corner points of the visual cue cannot be accurately detected.

Machine Learning Based Approach

The developed machine learning object detector is engaged at long distances, where detecting the object (or ship) as a whole is of interest. Classical computer vision may achieve the same task, however, it requires explicit algorithms for detection which leads to being complicated and also challenging to capture every aspect of the object. Instead, machine learning object detection learns the characteristics of the object thoroughly using neural networks. The speed and accuracy of this detection process are significantly improved by using Graphics Processing Units (GPUs).

Once it detects an object, it returns the position of the object and the size of its rectangle bounding box in the image. The position and size are used to determine approach course and speed, respectively. Since the detection range is proportional to the area occupied by the detected object in the image, during the early phases of approach the whole ship is detected to maximize the detection range. As the UAV gets closer to the ship, it also detects the horizon bar which provides more specific position and size information. Instead of having one object detector that identifies two objects (ship and horizon bar), two separate object detectors are developed for each object so that they do not have to distinguish one object from the other. The control system takes the information from the object detectors and takes the most appropriate control action in a given situation.

Developing the object detector involves three steps, which are, collecting images, labeling objects in the images, and training the object detector. First, 2000 images of the ship platform and 1000 images of the horizon bar are collected by UAV's onboard camera. To include various object figures in training sets, the perspective, lighting, weather condition, and distance were varied as much as possible. Second, the object in the collected images is labeled by drawing a bounding box and designating the object name. This data is stored as pixel positions and the object identification number. Third, the set of labeled images are used for training via a state-of-the-art machine learning technique. In order to implement the machine learning object detection in a real-time flight system, computational speed is the first factor to consider. According to the recent studies that compare the processing time of fast detecting algorithms (Refs. 14,17,18), YOLOv3 is faster than other algorithms such as SSD and Faster R-CNN while having similar accuracy in prediction. For this reason, the YOLOv3 algorithm is selected for the system. The object detection task consists of object classification and localization. Typically, an object detector is developed to detect multiple objects and classify them; however, in the present approach, two separate single object detectors are developed to detect the ship platform and horizon bar, respectively. By doing so, each detector only needs to find a particular object in the image, and the object class is automatically assigned without incurring the risk of false classification.

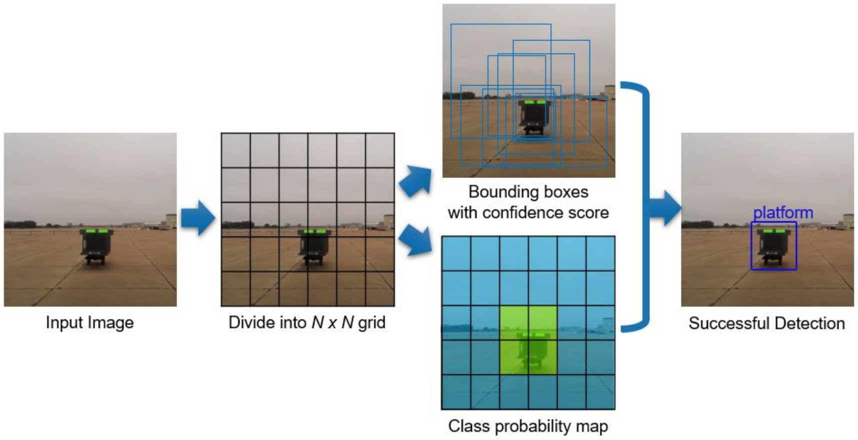

Figure 3. Ship platform detection process by YOLOv3 algorithm

The one-stage detector YOLOv3 regards the detection as a regression problem and uses a single neural network. It analyzes the entire image to predict the object-bounding box. As shown in Fig. 3, the input image is divided into an N x N grid and each grid cell predicts bounding boxes with a confidence score and class probability. To better detect the object in different sizes, it predicts bounding boxes at three different scales, which helps to detect the object from a far distance. The predictions of developed detector are encoded as an $N \times N \times [3 * (4+1+1)]$ tensor for $N \times N$ grid cells, 3 different scales, 4 bounding box offsets, 1 object confidence score, and 1 class prediction.

The observed maximum ranges for detecting the ship platform and horizon bar are approximately 250m and 100m, respectively. The real-time detection at different distances is shown as bounding boxes with object names in Fig. 4.

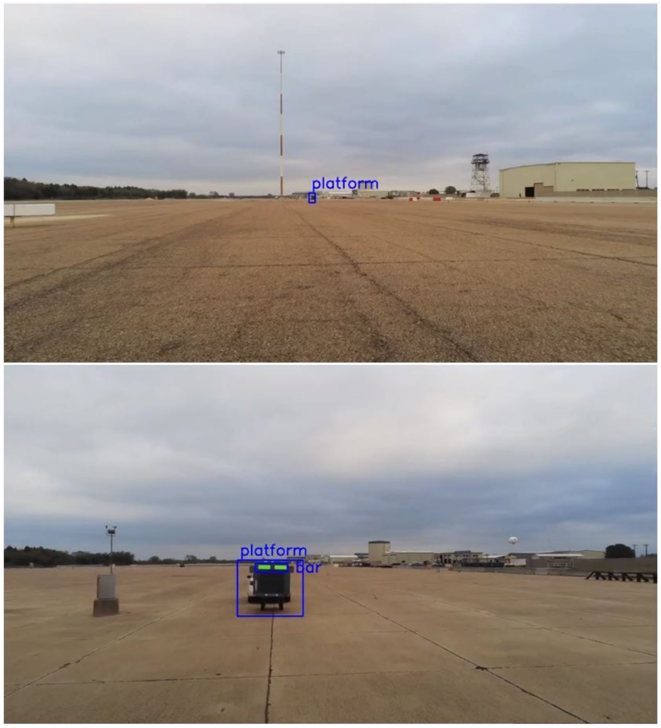

Figure 4. Long and mid-range real-time object detection result

Classical Computer Vision Based Approach

Once the UAV reaches close proximity of the ship platform, it is required to detect the visual cue and then estimate relative position and orientation for the final approach and landing. At close range, robust detection and estimation are achieved by combining computer vision techniques and developed screening algorithms. The vision system first detects the desired points of interest and then estimates the relative position and orientation. The visual cue does not need to have any particular form as long as the dimensions are known apriori. However, unique features that are distinguishable from the surroundings are favorable for detection. The current visual cue closely mimics the horizon bar on Navy ships and is installed normal to the aircraft approach course. It has two green-colored rectangles on a grey background with known size and separation as shown in Fig. 5.

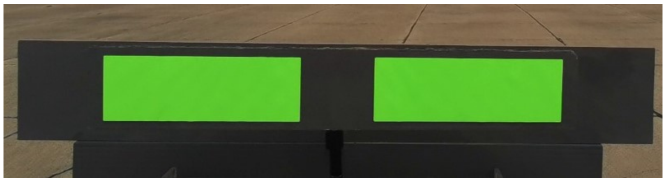

Figure 5. Installed horizon bar

Considering the characteristics of the installed horizon bar, the corner points of the green rectangles are determined as the targets to be detected. In order to ensure robust detection and estimation in scenarios involving large UAV movements and different light conditions, the established vision system sequentially conducts the image filtering, contour and corner detection, detected points screening, and the estimation of position and orientation.

Image Filtering: The initial approach involves applying a Hue-Saturation-Value (HSV) filter to sort out the green rectangles. The HSV filter is preferred over Red-Green-Blue (RGB) because the RGB color space is more sensitive to light conditions. To ensure the capture of the green rectangles in different light conditions, the greater range of HSV (H: 35 - 85, S: 70 - 255, V: 90 - 255) is assigned; however, this inevitably leads to the capture of undesired portions. In the HSV filtered image, there possibly exists small white patches outside the rectangles and black voids inside the rectangles. The morphological opening technique is used to remove the white patches in the image. It first erodes an image removing any small white patches and then dilates the eroded image to preserve the original shape and size. Morphological closing is used to fill up small voids in those rectangles by dilating an image first then eroding the dilated image. By performing these operations, the rectangles are depicted as white regions on the black background. The watershed algorithm (Ref. 38) is exploited to obtain clear boundaries of the rectangles. The two reference areas are obtained by eroding the processed image by 1% and dilating the processed image by 1%. The gap between the two areas is assumed to contain the boundaries of the rectangles. The algorithm simultaneously expands the area of the background and the rectangles toward each other until they meet at one pixel point. By connecting the points, clear boundaries of the rectangles are obtained. The images of each step are shown in Fig. 6.

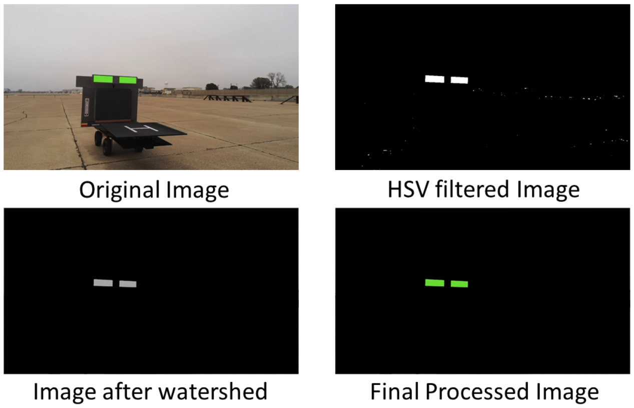
Original Image
HSV filtered Image
Image after watershed
Final Processed Image

Figure 6. Image filtering process

Contour and Corner Detection: Once the green rectangles are isolated by the image filtering process, the detection of contours and corner points is conducted. Even after the filtering, directly implementing any pre-existing corner detection algorithms is prone to detect some false corners. To detect the eight corner points precisely, the contours of the detected region are found and bounded in rectangles first. Thus, the size and shape of the detected areas are very close to the green rectangles and the corners of those bounding rectangles can be used as rough estimates of the actual corners. Second, the Förstner corner detection method is adopted to detect the corner points of the rectangles precisely based on the rough corners obtained by contour detection (Ref. 39). The Förstner corner detection increases the accuracy by sub-pixel refinement process. It is based on the fact that an ideal corner is a single point that tangent lines of the object cross perpendicular to each other as shown in Fig. 7. This way, false corner detection can be avoided.

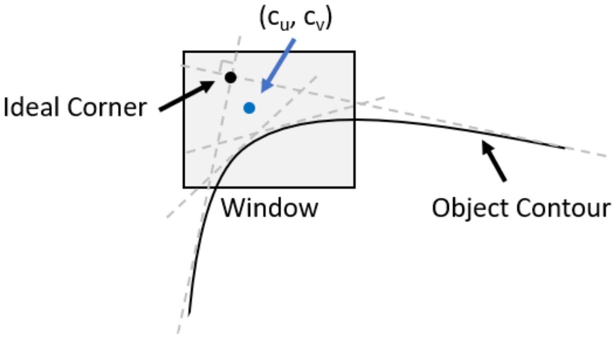

Figure 7. Förstner corner detection method

The pixel information around a corner is not perfectly clear. Therefore, an approximation process for defining the corner position (c_u, c_v) is required and is expressed in Eq. (1).

$$(\hat{c_u}, \hat{c_v}) = \underset{c_u, c_v}{\arg\min} \sum_{u,v \in N} ([\nabla f(u,v)]^T (u - c_u, v - c_v))^2 \tag{1}$$

The image gradient $\nabla f(u,v)$ at the image pixel position (u,v) is perpendicular to line from (u,v) to corner position (c_u, c_v). In order to obtain (c_u, c_v), a least square estimation in a small window is used. The projection of line from (c_u, c_v) to (u,v) on to the tangent line at (u,v) is required to be maximized in the given window. In other words, the projection of the image gradient onto the line segment connecting (c_u, c_v) and (u,v) has to be minimized for all (u,v) inside the window N. One crucial part here is the size of the window chosen for each image. It cannot be a fixed window size because the UAV is always moving towards the landing pad. The size of the detected region keeps increasing as it approaches closer to the visual cue. Hence a variable window size which is a function of the width and height is assigned as expressed in Eq. (2).

$$sw(w(t_k), h(t_k)) = \frac{1}{5}\{w(t_k) \times h(t_k)\} \tag{2}$$

$sw(w(t_k), h(t_k))$ is the window size at time t_k where $w(t_k)$ and $h(t_k)$ are the width and height of bounding rectangles in pixels. The results of the contours and corners detection are shown in Fig. 8.

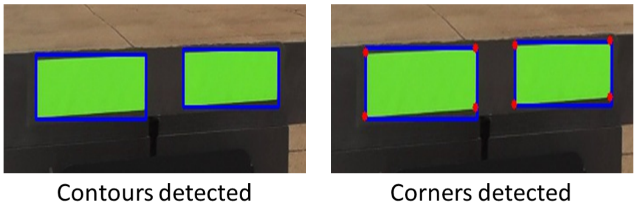
Contours detected
Corners detected

Figure 8. Detection of contours and corners

Detected Points Screening: The screening procedure is established to assure that no false corners are present. All the corners are sorted in a particular order as shown in Fig. 9, which helps in finding the length and slope of each side of the rectangles in the image. Even though they are not perfect rectangles in the image, width 1, 2, 3, and 4 have similar lengths and slopes. The height 1, 2, 3, and 4 also have similar lengths and slopes. A ±10% tolerance level is set for the lengths and a ±5% tolerance level is set for the slopes.

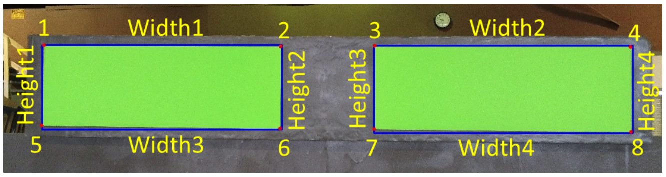

Figure 9. Detected corners in sorted order

Estimation of Relative Position and Orientation:

The estimation is based on a single camera calibration method using a planar object (Refs. 40, 41). The geometric relation of the image and real-world coordinates is shown in Fig. 10.

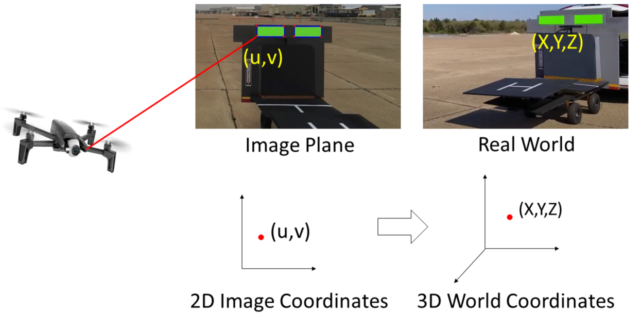

Figure 10. Geometric relation of image and real-world coordinates

A conventional pinhole camera model is used to derive the geometric relation. A 3D coordinate system can be defined with respect to the visual cue and there is a 2D coordinate system associated with the image frame. (X,Y,Z) is a point on the object described in the visual cue body-fixed frame, and (u,v) is a corresponding point on the image frame (pixel position in the image). The relationship between the image coordinates and the body-fixed coordinates in matrix form is described in Eq. (3).

$$s\begin{bmatrix}u\\v\\1\end{bmatrix}=A\begin{bmatrix}R & t\\0_{1x3} & 1\end{bmatrix}^{-1}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}\tag{3}$$

The equation is derived in a homogeneous coordinate system. In the homogeneous coordinate system, the 2D image point (u,v) becomes a 3D vector $(u,v,1)$ and the 3D world coordinates (X,Y,Z) become 4D vector $(X,Y,Z,1)$. The scaling factor s comes from transforming the homogeneous coordinates to cartesian coordinates. R is a 3 x 3 rotation matrix and t is a 3 x 1 translation vector. R matrix has the information about the camera orientation with respect to the visual cue and t provides the information on how far the camera is from the particular (X,Y,Z) point in the visual cue body-fixed frame. O_{1x3} is 1 x 3 zero matrix. The matrix which includes R, t, O_{1x3}, and 1 is called the camera extrinsic matrix, which varies from image to image. The matrix A is called the camera intrinsic matrix and its components are shown in Eq. (4).

$$A=\begin{bmatrix}1/\rho_u & 0 & u_0\\0 & 1/\rho_v & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}f_x & 0 & 0 & 0\\0 & f_y & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\tag{4}$$

$1/\rho_u$ and $1/\rho_v$ are parameters that scale the image coordinates, which are in pixels to values in the International System of Units (SI). u_0 and v_0 are the center position in the image plane. The first matrix shifts the center of the image plane to the top left corner point. The second matrix contains the focal length information f_x and f_y, which are the focal lengths of the camera in the x and y directions, respectively. The product of two matrices forms the camera intrinsic matrix, which depends on the particular camera. This unique camera intrinsic matrix can be found using a camera calibration (Ref. 42).

To determine the relative position and orientation of the camera, the Perspective-n-Point (PnP) algorithm is applied. Given a set of n 3-D coordinates of an object and its corresponding 2-D projections on the image, this algorithm solves Eq.(3) to obtain the rotation vectors R and the translation vectors t. There are 6 DOF for a camera, which are 3 DOF in rotation (roll, pitch, yaw) and 3 DOF in translation (X,Y,Z). A minimum of 3 points are required to find a solution, but the solution is not unique. There should be a minimum of 4 points to obtain a unique solution; however, it can be more reliable and redundant when there are more points. An iterative method is used for the PnP algorithm since it is robust for objects which consist of a planar surface and gave more accurate results. The iterative method is based on Levenberg-Marquardt optimization (Refs. 43, 44). In this method, the function minimizes re-projection error, which is the sum of squared distances between the observed image points (u,v) and projected object points (X,Y,Z). By default, the iterative algorithm sets the initial value of rotation and translation as zero and then updates during each iteration.

The RANSAC method (Refs. 45, 46) is used to find a rough initial guess for the extrinsic matrix in an iterative approach. The RANSAC method also identifies the outliers and removes them during the calculation. The initial guess and inliers are fed into the iterative algorithm to have a more accurate estimation. The solvePnP algorithm (Ref. 42) returns a rotation vector and the translation vector. The rotation vector can be converted to the rotation matrix using the Rodrigues function. Thus, once the rotation matrix (R) and translation vector (t) are obtained, $-R^{-1}t$ gives the relative distances in 3D from the camera to the origin of the visual cue body-fixed frame.

The present estimation method is modified from the existing algorithm to take advantage of the gimbal camera. Since the gimbal corrects for the roll and pitch motion of the UAV, the images only reflect the yaw motion (heading angle) with respect to the visual cue. Thus, roll and pitch angles computed by the image can be regarded as camera noise and gimbal correction errors. Therefore, the roll and pitch angles are excluded from the position estimation. The basic and modified rotation matrix are specified in Eqs. (5) and (6), respectively.

$$R_{basic}=\begin{bmatrix}c\alpha c\beta & c\alpha s\beta s\gamma-s\alpha c\gamma & c\alpha s\beta c\gamma+s\alpha s\gamma\\s\alpha c\beta & s\alpha s\beta s\gamma-c\alpha c\gamma & s\alpha s\beta c\gamma+c\alpha s\gamma\\-s\beta & c\beta s\gamma & c\beta c\gamma\end{bmatrix}\tag{5}$$

$$R_{modified}=\begin{bmatrix}\cos\alpha & -\sin\alpha & 0\\\sin\alpha & -\cos\alpha & 0\\0 & 0 & 1\end{bmatrix}\tag{6}$$

$$\begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} = -R_{modified}{}^{-1} t \tag{7}$$

α, β, and γ represent yaw, pitch, and roll angles, respectively. Each row of the R matrix contributes to their respective coordinates (X,Y,Z). By setting pitch and roll angle to zero, the modified rotation matrix is obtained from the basic rotation matrix. Only the yaw angle remains which takes care of the relative heading angle. The third row of the modified rotation matrix shows that the Z-coordinate is independent of the yaw angle, α. Hence, the yaw angle has no contribution in the estimation of forward relative distance (Z-coordinate) between the UAV and the visual cue. However, the sideward relative distance (X-coordinate) and vertical relative distance (Y-coordinate) are heavily dependent on the yaw angle. The camera position with respect to the visual cue is given in Eq. (7)

It is apparent from multiple experiments that the yaw angle estimation becomes noisy as the camera gets farther away from the visual cue. Since the estimation is based on the number of visual cue pixels in the image, the changes in pixels with distance result in the noise. Despite this sensitivity issue, yaw estimation still shows a reasonable trend within the range that the forward relative distance is accurately estimated. To utilize the yaw estimation trend, instead of directly taking the noisy yaw angle estimation, a simple low pass filter is configured to reduce the noise level. One of the commonly used real-time data filters is Kalman Filter. Kalman filter predicts the current estimate by taking into account the current measured value, the previous estimate, and the noise level in the data. A single state Kalman filter was opted over other real-time filters because of the lesser number of variables required while computation. Gaussian noise is assumed to be present in the measurement data with a specified mean and variance. The underlying equations involved in estimating the yaw are:

$$\begin{aligned} CE &= PE + KG \times CM \\ KG &= PrE \times \frac{1}{PrE + RN} \\ PrE &= PE + QN \end{aligned} \tag{8}$$

where CE is the current yaw estimate, PE is the previous yaw estimate, KG is the Kalman Gain, CM is the current yaw measurement, RN is the process noise covariance, and QN is the measurement noise covariance. RN and QN values are experimental values found by analyzing different sets of yaw angle measurements and they are set to 0.005 and 0.05, respectively.

The estimation method for the relative position and orientation is previously validated using a Vicon motion capture system, which demonstrated sub-centimeter and sub-degree accuracy as detailed in (Ref. 24).

NONLINEAR CONTROL SYSTEM

The sequential steps involved in the vision-based control system are live-streaming the video from UAV's onboard camera to an external base-station computer, image processing, feedback controller generating the control inputs, and transmitting the control inputs back to the UAV. The processing time is considerably affected by the type of detection method engaged in the vision system. The average time to complete one cycle is 0.5 seconds when the machine learning detection is engaged and 0.03 seconds when the detection of the rectangle corner points is engaged. In order to cope with the time delay as well as potential sensor and estimation noise, nonlinear controllers, which can adapt to different situations are developed. Particularly, five different flight modes are configured according to the situations perceived by the vision system as shown in Fig. 11.

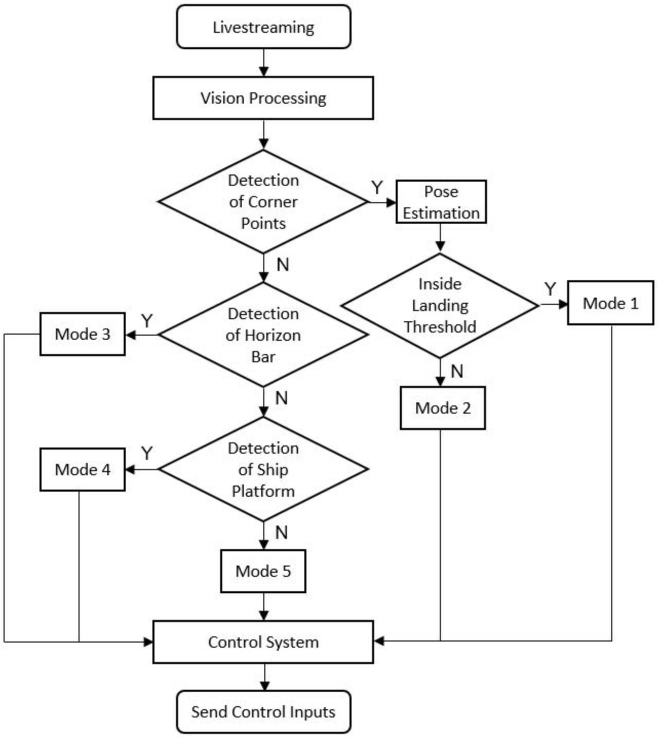

Figure 11. Flowchart showing vision-based control system

Long-range Tracking Controller

In the long-range where the entire landing platform is detected as a whole, the information provided by the vision system is the platform size and position in the camera view. The flight control system in this region deals with a relatively slow update rate that is 0.5 seconds on average. When the update rate is fast enough, the discrete system that receives sensor data at each update time can be a good approximation to the continuous system. Thus, integral and derivative controllers can be configured by using the summation and difference of error. However, the summation and difference are not good approximations for integral and derivative controllers when the update rate is slow. Hence, the nonlinear controller is designed to achieve the control task in the presence of the slow update rate by applying the nonlinear exponential gain $K_P\{e(t_k)\}$ as shown in Eq. (9).

$$
\begin{aligned}
e(t_k) &= r(t_k) - c \\
u(t_k) &= K_P\{e(t_k)\} \times e(t_k) \\
&= \begin{array}{ll} -m(e^{ae(t_k)} - d) \times e(t_k) & (e(t_k) < 0) \\ m(e^{ae(t_k)} - d) \times e(t_k) & (e(t_k) \geq 0) \end{array}
\end{aligned}
\tag{9}
$$

In Eq. (9), $e(t_k)$ denotes the error at time t_k, $r(t)$ is the relative position at time t_k, and c is the setpoint. The control law, $u(t_k)$, has exponential term in the nonlinear gain $K_P\{e(t_k)\}$ to decay control magnitude exponentially near zero error. The constants m, a, c, and d selected for the ship platform and bar tracking are provided in Table 1.

Table 1. Selected constants of long-range controllers

Flight Mode	Controller	m	a	c	d
Ship Platform Tracking	pitch	0.008	0	5000	0
	roll	1.2	0.0158	640	1
	heave	3.0	0.0108	360	1
Bar Tracking	pitch	0.004	0	3400	0
	roll	1.0	0.0158	640	1
	heave	5.0	0.0108	360	1

The desired object size, the image center position in the horizontal direction, and the image center position in the vertical direction are the setpoints for the pitch, roll, and heave controllers, respectively. Unlike the roll and heave controllers, the pitch controller with the assigned parameters returns to a linear proportional controller, which is sufficient to approach the ship in the long-range. There is also a yaw controller that can control the heading angle to set the approach course. It is designed as a nonlinear probabilistic control system, which is the same as the close-range tracking yaw controller and detailed in the following section. The only difference is that a magnetometer is used to read the current heading in the long-range and the vision system provides the estimated heading angle during close-range tracking.

The exponential nonlinear roll control input is shown in Fig. 12.

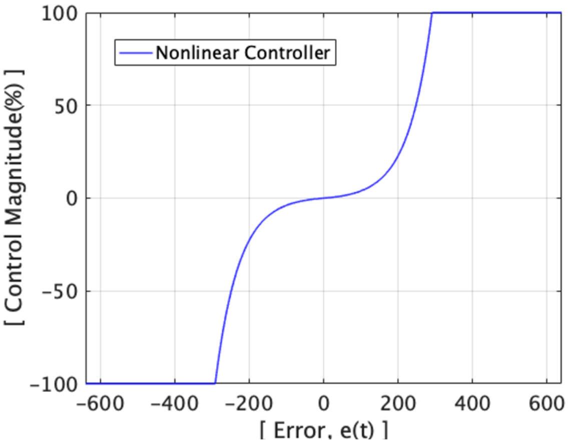

Figure 12. Exponential nonlinear roll control input

Depending on the constant value a, the rate of change of control magnitude varies. In the roll controller case, constant a is selected as 0.0158. The maximum and minimum control magnitudes are limited to 100 and -100, respectively. Even though it utilizes only the error at time t, it can minimize the overshoot around the setpoint where the error is zero by decaying quickly. On contrary, it yields a large control magnitude as the error becomes larger.

In the case of a slow update rate, the nonlinear controller is able to achieve stable setpoint tracking. The effect of the nonlinear roll controller is compared to the conventional linear proportional controller and proportional-integral-derivative (PID) controller as shown in Fig. 13.

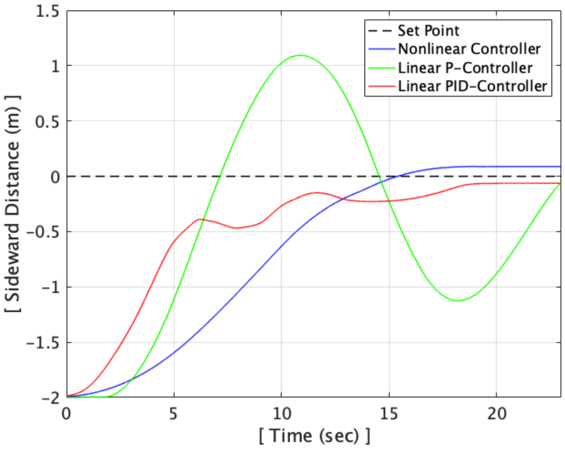

Figure 13. Effect of exponential nonlinear Controller

It demonstrates that the nonlinear controller can prevent the system from overshooting the setpoint. However, the linear proportional controller cannot stabilize the oscillations because it computes the control input by multiplying the error with a fixed gain value. Even the linear PID controller is not able to stabilize the oscillations in an effective fashion due to the slow update rate.

However, as seen from Fig. 13, the nonlinear controller has some steady-state error. Considering the long distances at which the nonlinear controller is engaged, the steady-state error is not an issue because this is not the final error of the entire approach and landing maneuver, but the initial error for the corner tracking system, which takes over the control at close distances. Therefore, it is more imperative to control the aircraft stably than to achieve the minimum steady-state error while the UAV is approaching from a long distance.

Close-range Tracking Controller

At close range, when the vision system can reliably detect the corners of the rectangles on the visual cue, the vision-based controller utilizes the estimated relative position and orientation data to yield control inputs. The average update rate is 0.03 seconds, which is significantly faster than the machine learning object detection that is applied at long distances.

Having integral and derivative controllers along with the proportional controller enables precise tracking. Even though the estimation provides accurate position and orientation data with sub-centimeter and sub-degree error, a small error difference between subsequent time steps can yield large and noisy control magnitude since it depends on the difference in error divided by the small update rate of 0.03 seconds. To take advantage of the derivative controller with minimum noise, the Kalman filter and nonlinear derivative controller are designed.

A single state Kalman filter is applied with the unity feedback, which means that it does not require the prediction from the model. This model-free estimator reduces the noise, however, it uses error difference values for estimation. Therefore, it will be affected by incorrect and unrealistic error difference values that may occur from time to time. To reject this intermittent unrealistic estimation effectively, a novel nonlinear derivative controller with linear proportional and integral controllers is designed by utilizing the normal distribution concept as described in Eq. (10).

$$\begin{aligned} de(t_k) &= e(t_k) - e(t_{k-1}) \\ u(t_k) &= K_P e(t_k) + K_I \sum_{k=1}^{k} e(t_k) dt_k + K_D\{de(t_k)\} \frac{de(t_k)}{dt_k} \end{aligned} \tag{10}$$

$de(t_k)$ is an error difference between time t_k and t_{k-1}, and $u(t_k)$ is a control law that has linear proportional and integral terms as well as the nonlinear derivative term. K_P and K_I are constant proportional and integral gains, and $K_D\{de(t_k)\}$ is the nonlinear derivative gain that is a function of error difference, $de(t_k)$, as specified in Eq. (11).

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-0.5\frac{(x-\mu)^2}{\sigma^2}} \\ K_D\{de(t_k)\} &= be^{-0.5\frac{(de(t_k)-\mu)^2}{\sigma^2}} \end{aligned} \tag{11}$$

$f(x)$ is the probability density function that forms a normal distribution. σ denotes the standard deviation and μ denotes the mean value. The area under the function indicates the probability that a certain range of deviation occurs. The probabilistic nonlinear derivative controller is constructed by taking the exponential term from the probability density function and multiplying constant b that determines the control magnitude. Based on the observation of aircraft movement, σ is determined as 0.04, which means the distance that the aircraft can move during 0.03 seconds has a 68.2 percent chance of being within 4 cm and a 95.4 percent chance of being within 8 cm. When b is 1 and μ is 0, the probabilistic nonlinear derivative controller computes derivative gains as shown in Fig. 14.

Depending on the error difference, $de(t_k)$, the corresponding nonlinear derivative gain, K_D, is selected and multiplied by the derivative term $de(t_k)/dt_k$. If the estimated error difference is too high (or unrealistic), then it takes a small K_D value to significantly minimize the control input. In this way, the controller does not respond to the large noise in the error data, which can trigger undesired and unstable maneuvers. Also, the magnitude of gain can vary according to the selection of constant b in Eq. (11). The effects of the Kalman filter and the probabilistic nonlinear controller are shown in Fig. 15.

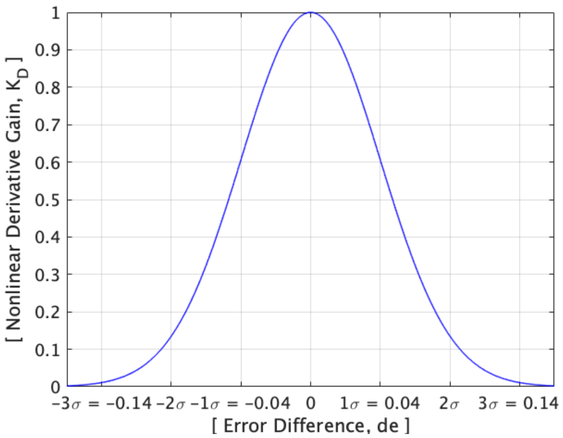

Figure 14. Probabilistic nonlinear derivative controller

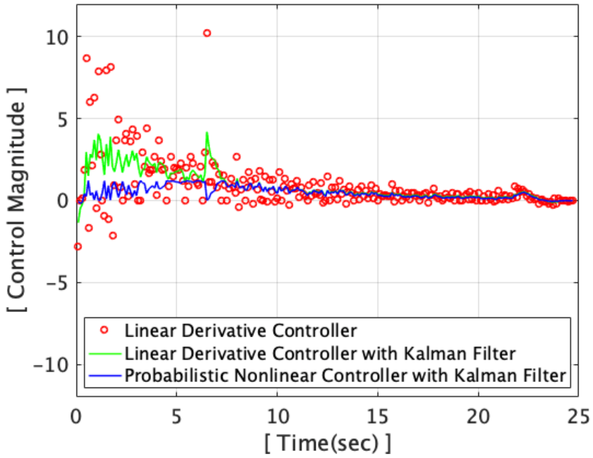

Figure 15. Effect of kalman filter and probabilistic nonlinear derivative controller

Red circles denote the control magnitude of the linear derivative controller without the Kalman filter. The high noise that occurs in the range of 0 to 10 seconds are due to the derivative term, $de(t_k)/dt_k$. This term is sensitive because the error is divided by a small dt_k, which has an average value of 0.03 seconds. Thus, even a small error in estimation can be magnified in the derivative controller. The green line is the result after applying the Kalman filter to the linear derivative controller. The noise is reduced, however, it still yields large control inputs in response to the incorrect estimation values. The blue line shows the control inputs generated by the probabilistic nonlinear controller with the Kalman filter and these inputs are relatively small and insensitive to the large unrealistic error differences. At the 6.5 second mark, the error difference de has a large value caused by incorrect estimation. In this case, the linear derivative controller with the Kalman filter reduced the magnitude to some extent but it is still affected by that particular spurious error value. However, the probabilistic nonlinear controller with the Kalman filter is able to screen out the wrong value, thereby minimizing the effect of incorrect error estimation. The finally selected gains through extensive flight tests and simulations are specified in Table 2.

Table 2. Selected gains of close-range controllers

Flight Mode	Controller	K_P	K_I	K_D		
				b	μ	σ
Corner Points Tracking	pitch	7.5	0.05	4.5	0.02	0.04
	roll	7.5	0.01	8.5	0	0.04
	heave	15	0.01	7	0	0.02
	yaw	5.5	0.05	1.75	0	5

FLIGHT TESTING

The objectives of flight testing are to verify the long-range tracking capability, landing accuracy, and safe vertical landing on a platform with 6 DOF motions. Extensive flight tests are conducted in realistic and challenging scenarios. First, the ship landing environment is mimicked by the landing pad via implementing the Oliver Hazard Perry Class FFG Frigate motions with helicopter ship landing limits. While the landing pad is in motion, the ship platform is also translating forward just like a real ship during a helicopter landing. During the approach and landing, the UAV makes transitions between the flight modes that engage with a particular vision and control method based on the perceived situation, and lands vertically without matching the UAV attitude to that of the pad. During outdoor flight testing the maximum wind speed experienced was around 9 m/s (17.5 knots) and the wind direction kept fluctuating over the course of these experiments. The GPS is only used to log positions during flight and not for control.

Experimental Setup

To simulate the 6 DOF motions of the ship deck, a Servos and Simulation Inc Generic Motion System (model 710-6-500-220) with a 4 ft × 4 ft (1.22 m x 1.22 m) landing deck as shown in Fig. 16. The ranges of roll, pitch, and yaw are ± 13°, ± 15°, and ± 16°, respectively. The ranges of surge, sway, and heave are ± 10.2 cm, ± 10.2 cm, and ± 6.4 cm, respectively.

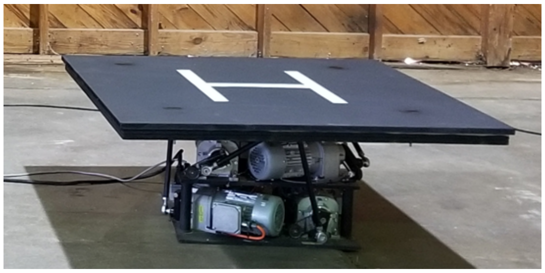

Figure 16. Servos and Simulation Inc. 6 DOF motion system with 4 ft x 4 ft Landing Deck

The ship platform is constructed including the horizon bar and motion deck as shown in Fig. 17. The width, height, and length of the ship platform are 5 ft, 5 ft, and 10 ft, respectively. The horizon bar always indicates a perfect horizon, and the motion deck has its own 6 DOF motions in addition to the forward translational motion, which is similar to what would be experienced on a real ship.

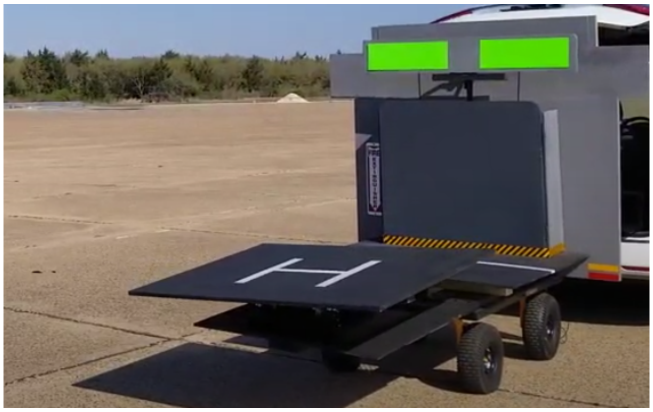

Figure 17. Constructed ship platform with horizon bar and motion deck

A Parrot ANAFI is selected as a representative VTOL UAV. It has a gimbal camera that can mechanically compensate for the roll and pitch motion of the UAV and live streams a video in 720p resolution. As shown in Fig. 18, it has the embedded inner-loop autopilot that controls each propeller's rpm to achieve the commanded inputs generated by the outer-loop vision-based control system developed in this study.

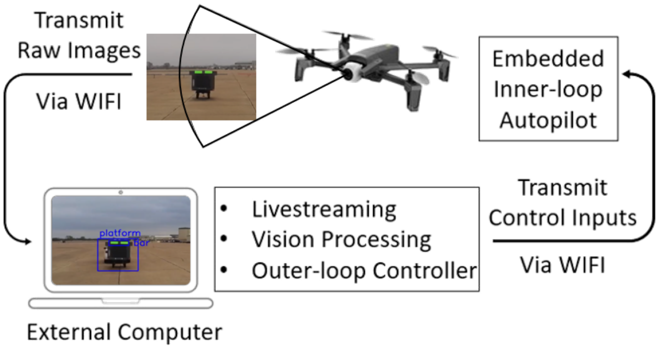

Figure 18. Process of autonomous flight system

The system used for processing is LENOVO Legion Y740-15IRH, which is composed of Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 6 Cores, and 12 Logical Processors. It features an integrated NVIDIA GeForce GTX 1660 Ti 6GB Graphics and 8GB of LPDDR4 memory with a 128-bit interface. The Ubuntu 18.04 with Nvidia driver version 440 and CUDA version 10.2 are used. The WIFI communication is established by an external TP-LINK TL-WN722N Wireless N150 High Gain USB Adapter.

Ship Motions and Vertical Landing Safety

In order to demonstrate that it is safe to land vertically without matching the UAV attitude dynamics to platform motion, landing tests are conducted while the 6 DOF platform is simulating two challenging ship motions. The first prescribed motion is the Oliver Hazard Perry Class frigate at the sea state of 6 and a wave direction of 60° introduced by Sanchez-Lopez, Jose Luis, et al. (Ref. 1). The frigate is 136 meters long and 14 meters wide and has a single flight deck. Sea state 6 is defined by the World Meteorological Organization (WMO) as a very rough condition that has a wave height of 4 to 6 meters. While the platform is undergoing this complicated motion, vertical landings are executed at random time instances as shown in Fig. 19.

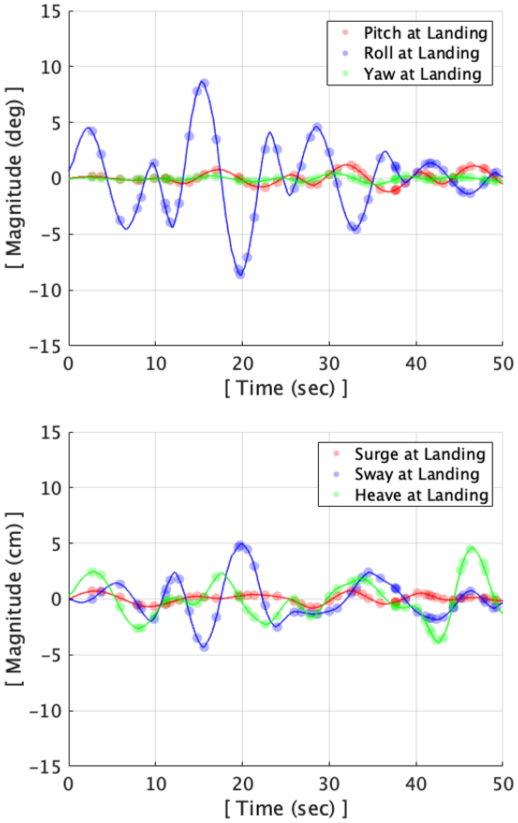

Figure 19. Oliver Hazard Perry class frigate motion and vertical landing moments

Solid red, blue, and green lines denote continuous angular (pitch, roll, and yaw) and linear (surge, sway and heave) motions of the 6 DOF platform. 50 landing tests are conducted in total and the motion of the platform at the time of each successful landing are depicted as dots. The randomly distributed landing timings demonstrate the vertical landing is safe at any moment of the Oliver Hazard Perry Class FFG frigate ship motions under the given conditions.

The second prescribed motion is the Navy helicopter ship landing limits defined by NATOPS. In the case of the FFG 7 Class Ships which the Oliver Hazard Perry frigate belongs to, the ship motion limits for landing are set as ± 8° of roll and ± 3° of pitch. Even though the limits are defined by the maximum roll and pitch magnitudes, the frequency of the motion is also a crucial factor for the motions. According to the report for a similar size ship (length: 152.4 m, width: 15.2 m) motions conducted by the Sandia National Laboratory (Ref. 37), the roll period is 10.1 seconds and the pitch period is 6.5 seconds as shown in Table 3.

Table 3. Typical ship characteristics

Ship Type	Length (m)	Width (m)	Roll Period (secs)	Pitch Period (secs)
Destroyer	152.4	15.2	10.1	6.5
Aircraft Carrier	304.8	38.1	15.8	8.8

The maximum pitch and roll magnitude with the reported periods are applied to the platform and 50 vertical landings are conducted at random moments as shown in Fig. 20.

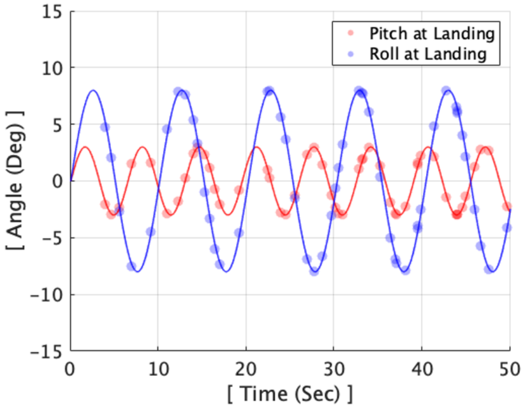

Figure 20. Motions of helicopter ship landing limits and vertical landing moments

Solid red and blue lines denote continuous pitch and roll motions of the 6 DOF platform. The dots are the motions at the time of successful landings. The randomly distributed landing timings demonstrate the vertical landings are safe independent of the motion as long as it is within the operational limits.

Tracking Capability and Landing Accuracy

The tracking capability and landing accuracy of the developed vision-based autonomous flight control system are verified in challenging situations such as random initial positions, maximum distance of 250 meters, realistic ship motions, communication latency, sensor noise, low visibility, and windy conditions. In this fully autonomous vision-based system, GPS and a magnetometer are used only to log positions and heading angles and not for control. During flight testing, the landing pad is mimicking the ship deck motions that are introduced in the previous section.

First, flight tests are conducted to verify the maximum range and smooth transition between flight modes. The trajectories from take-off to landing are logged as shown in Fig. 21.

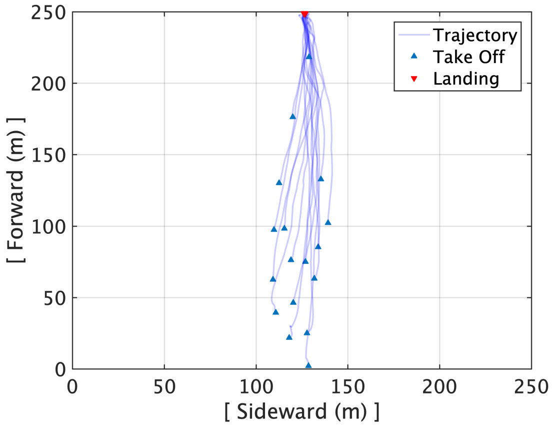

Figure 21. Trajectories of long-range tracking

The initial positions are widely distributed and the maximum distance between the UAV and the ship platform is approximately 250 meters. During flights, the flight modes are switched from the ship platform/horizon bar tracking to the corner points tracking depending on the distance. The results demonstrate stable long-range tracking capability in the presence of wind up to 9 m/s (17 knots). The time history of control inputs for a representative case is shown in Fig. 22.

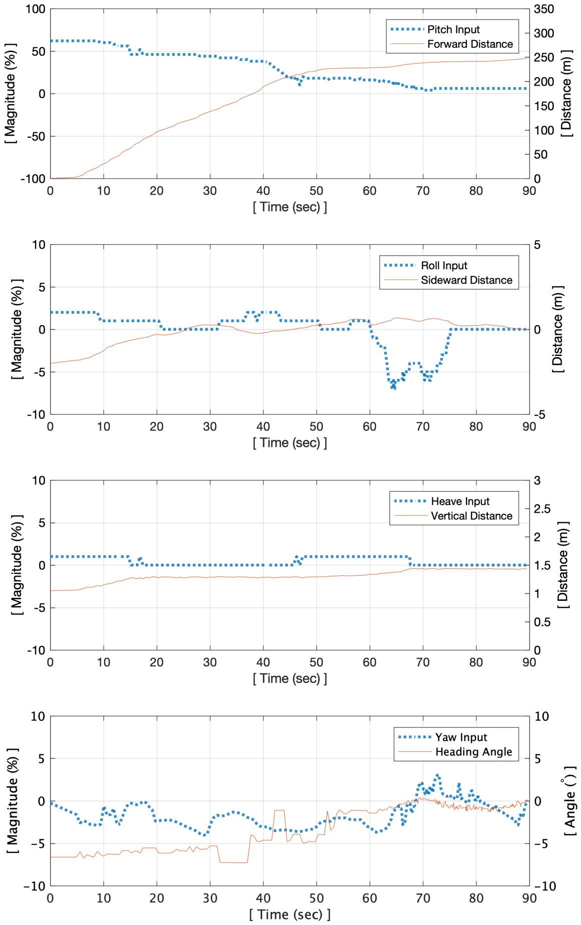

Figure 22. Control inputs, distances, and heading angle during long-range tracking in time

It shows the control inputs, distances, and heading angle after take-off until the execution of vertical landing. Each line denotes pitch, roll, heave, and yaw control inputs described as percentages that range from -100 to 100 and generated based on forward, sideward, vertical relative distance, and relative heading angle, respectively. Zero control input means neutral control input that maintains current UAV pose such as pitch, roll, altitude, and heading. While the UAV approaches the ship platform from 250 meters away, the vision-based flight control system effectively flies the UAV by smoothly switching between the flight modes from the machine learning object tracking for ship platform and horizon bar to the rectangle corner points tracking, depending on the relative distance.

Considering the UAV and landing pad size, the safe landing threshold is set by a 35 x 35 centimeter square from the pad center, therefore, once the UAV reaches the landing threshold the controller commands vertical landing. The final landing points from multiple flight tests are distributed within the set landing threshold as shown in Fig. 23.

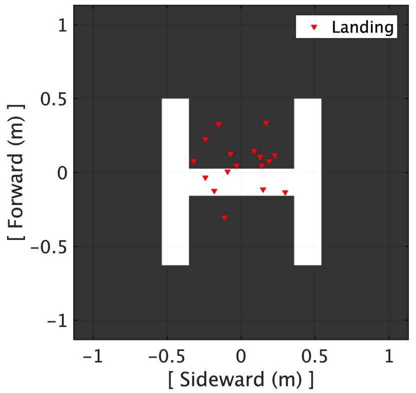

Figure 23. Landing points on pad of long-Range tracking

Second, flight tests are conducted to verify the robust tracking while the ship platform is moving in different courses with its own 6 DOF motions. The three representative trajectories of the ship platform and UAV are shown in Fig. 24.

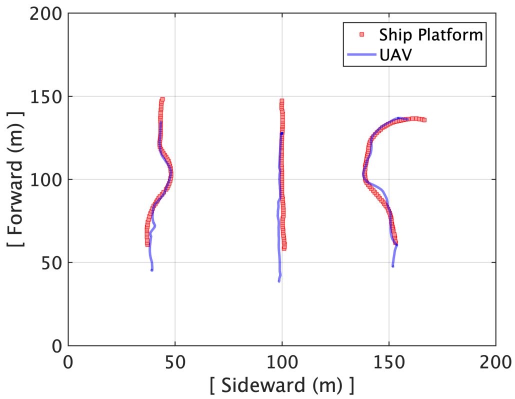

Figure 24. Trajectories of moving ship tracking

Red squares denote the trajectory of the ship platform and blue lines denote the trajectory of the UAV. It shows robust tracking while the ship platform is moving in straight, S-pattern, and 90° turn. The control inputs for S-pattern moving platform tracking are shown in Fig. 25.

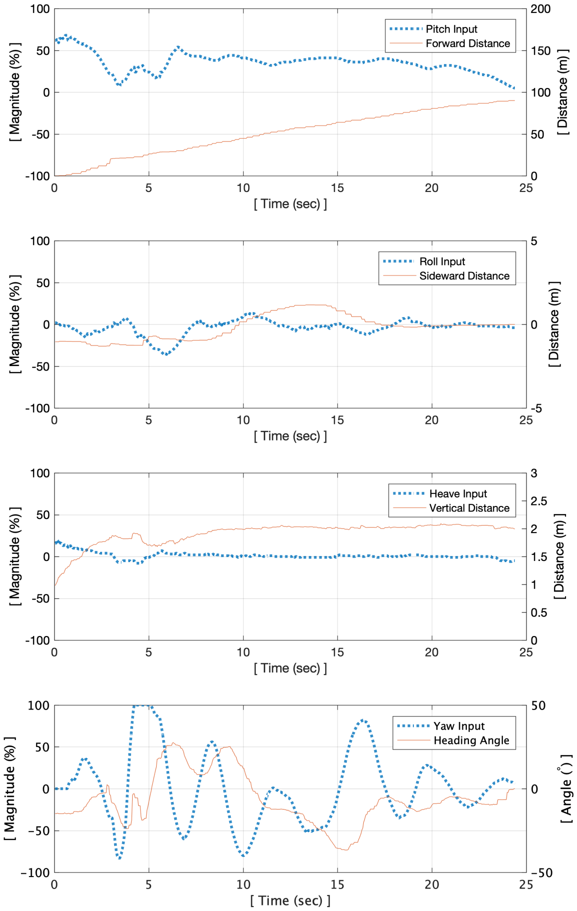

Figure 25. Control inputs, distances, and heading angle of moving ship tracking in time

It shows the control inputs, distances, and heading angle after take-off until execution of vertical landing while the UAV is tracking the ship platform moving in an S-pattern. The pitch control input is generated to approach the ship platform that varies its speed from 0 to 10 mph. Since the ship platform changes the course abruptly up to 130°, the relative sideward distance and heading also change to a great extent. Accordingly, the corresponding roll and yaw control inputs change aggressively to achieve zero relative sideward distance and heading. Vertical landings are conducted once the UAV flies into the 35 x 35 centimeters landing threshold while the ship platform is in motion. The final landing points from the multiple flight tests are distributed within the set landing threshold as shown in Fig. 26. The different flight test cases are shown in the video (Ref. 47).

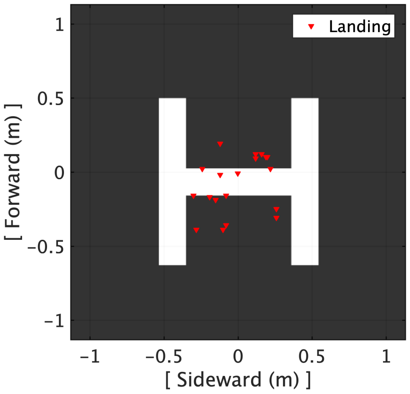

Figure 26. Landing points on pad of moving ship tracking

CONCLUSION

The goal of this study is to develop an autonomous ship landing solution for VTOL UAVs by closely following the Navy helicopter ship approach and landing procedure. This automation has been achieved by using a single onboard camera without using GPS. The developed ship landing system consists of a machine vision system and a nonlinear control system. The machine vision system is developed taking advantage of state-of-the-art machine learning and classical computer vision techniques. The control system is developed to generate situation-adaptive control inputs by introducing the idea of nonlinear gain variation and a probabilistic approach to limit the impact of incorrect estimations.

Multiple flight tests are systematically conducted to verify the safety of vertical landing maneuver, long-range detection and robust tracking capability, and landing accuracy. The UAV lands vertically independent of ship motions in the same manner as a Navy helicopter lands on a ship. 100 landing tests are successfully conducted on a moving deck, which mimicked realistic and challenging 6 DOF ship motions. The machine learning object detector trained by the YOLOv3 algorithm begins identifying the 6 x 6 ft ship platform from 250 meters away, which means the range in the case of a real ship landing can be 17.3 kilometers (9.3 nautical miles). This is estimated assuming that the rear-side of a typical small ship occupies an area of 50 x 50 ft. The unique nonlinear control system demonstrates robust tracking capability during a wide range of realistic scenarios such as random initial positions, complicated ship motions, communication latency, sensor noise, and in the presence of winds up to 20 mph. Since the landing command is triggered once the UAV flies into the pre-defined landing threshold area, the final landing points are randomly positioned within the area. The current landing threshold is 35 x 35 cm area which is sufficient to guarantee safe landing on a 122 x 122 cm (4 x 4 ft) landing deck.

Some of the key conclusions from this study are enumerated below.

1. The proposed vertical landing maneuver for VTOL UAVs, which involves not following the ship deck motions, was verified as a safe way of landing. This was

achieved through multiple landing tests on the deck mimicking challenging ship motions (Oliver Hazard Perry frigate at sea state 6) and NATOPS helicopter ship landing operational limits (roll: ±8, pitch: ±3).

2. The long-range vision system developed based on the state-of-the-art machine learning algorithm YOLOv3 demonstrated a 10 times greater detection range than the classical computer vision systems. The control system successfully utilized the detected object position and relative size as states for long-range tracking.

3. The biggest challenge to implement the machine learning based object detection on the real-time autonomous flight was the time delay. To cope with the time delay issue, the long-range controller was constructed that responded less sensitively to errors around the setpoint and aggressively to large errors, using the exponential gain. It enabled the UAV to stay in the appropriate flight course while approaching the ship platform.

4. The developed close-range vision system that combined the classical computer vision techniques and screening algorithms guaranteed fast and reliable detection of the visual cue and demonstrated precise relative position and orientation estimation. The update rate was approximately 15 times faster than the machine learning vision system and therefore, the control system was able to control the UAV more precisely using faster feedback.

5. Even after going through the configured Kalman estimator, the large/false estimation error can still occur at times. To prevent responding to such non-physical estimations, the probabilistic nonlinear controller was developed. It probabilistically perceives if the estimation is physically possible or not, based on the normal distribution curve and known UAV characteristics. Multiplying the estimation value by its probability can effectively reject responding to the false estimation. By this approach, the controller never generated abrupt large control input even when the vision system provided inaccurate estimations from time to time. This greatly improved the robustness of the tracking.

6. The landing accuracy depends on the pre-defined landing threshold. The accuracy can be increased by setting a smaller threshold; however, this will take more time and requires more control effort. Considering the UAV and landing pad size, the appropriate landing threshold of 35 x 35 cm was set and the UAV successfully landed within this area every time during the 100+ flight experiments.

The results demonstrate conclusively the feasibility of this novel autonomous approach and landing strategy for VTOL UAVs, which is inspired by the Navy helicopter ship landing. This is a significant accomplishment since there are no known efforts in the literature, which focused on automating the real helicopter ship landing procedure. The next goal of the study is to improve the wind disturbance rejection capability by combining the current control system with reinforcement learning techniques so that the aircraft could robustly fly through the ship wake during the approach and landing phases.

Author contact: Bochan Lee bochan.lee@tamu.edu
Vishnu Saj vishnu02saj@tamu.edu
Moble Benedict benedict@tamu.edu

ACKNOWLEDGMENTS

This work has been funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF award Numbers IIP-1161036 and CNS-1946890, along with significant contributions from C-UAS industry members.

REFERENCES

1. Sanchez-Lopez, J. L., Pestana, J., Saripalli, S., and Campoy, P., “An approach toward visual autonomous ship board landing of a VTOL UAV,” *Journal of Intelligent & Robotic Systems*, Vol. 74, (1), 2014, pp. 113–127.

2. Xu, G., Zhang, Y., Ji, S., Cheng, Y., and Tian, Y., “Research on computer vision-based for UAV autonomous landing on a ship,” *Pattern Recognition Letters*, Vol. 30, (6), 2009, pp. 600–605.

3. Truong, Q. H., Rakotomamonjy, T., Taghizad, A., and Biannic, J.-M., “Vision-based control for helicopter ship landing with handling qualities constraints,” *IFAC-PapersOnLine*, Vol. 49, (17), 2016, pp. 118–123.

4. Holmes, W. K., and Langelaan, J. W., “Autonomous ship-board landing using monocular vision,” Proc. 72nd Am. Helicopter Soc Forum, Vol. 2, 2016.

5. Meng, Y., Wang, W., Han, H., and Ban, J., “A visual/inertial integrated landing guidance method for UAV landing on the ship,” *Aerospace Science and Technology*, Vol. 85, 2019, pp. 474–480.

6. Yakimenko, O. A., Kaminer, I. I., Lentz, W. J., and Ghyzel, P., “Unmanned aircraft navigation for shipboard landing using infrared vision,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, (4), 2002, pp. 1181–1200.

7. Lumsden, B., Wilkinson, C., and Padfield, G., “Challenges at the helicopter-ship dynamic interface,” , 1998.

8. Colwell, J., “Maritime helicopter ship motion criteria-Challenges for operational guidance,” *Challenges for Operational Guidance-NATO RTO Systems Concepts and Integration Panel SCI-120. Berlin, Germany*, 2002.

9. Stingl, A. L., “Vtol aircraft flight system,” US Patent 3,487,553, January 6 1970.

10. "Helicopter operations from ships other than aircraft carriers(HOSTAC)," Vol. I, 2017.

11. Girshick, R., Donahue, J., Darrell, T., and Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition, 2014.

12. Girshick, R., "Fast r-cnn," Proceedings of the IEEE international conference on computer vision, 2015.

13. Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 39, (6), 2016, pp. 1137–1149.

14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., "Ssd: Single shot multibox detector," European conference on computer vision, 2016.

15. Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

16. Redmon, J., and Farhadi, A., "YOLO9000: better, faster, stronger," Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.

17. Redmon, J., and Farhadi, A., "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

18. Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., and Ouni, K., "Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3," 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), 2019.

19. Zishan, W. X. P. S. S., and Weiqun, S., "Computer vision scheme for autonomous landing of unmanned helicopter on ship deck [J]," *Journal of Beijing University of Aeronautics and Astronautics*, Vol. 6, 2007.

20. Wang, X., Pan, S., Song, Z., and Shen, W., "Vision based analytic 3D measurement algorithm for the autonomous landing of unmanned helicopter on ship deck," *Optical Technique*, Vol. 33, 2007, pp. 264–267.

21. Sharp, C. S., Shakernia, O., and Sastry, S. S., "A vision system for landing an unmanned aerial vehicle," Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), Vol. 2, 2001.

22. Jihong, L. S. H. C. Z., "A Method for Estimating Position and Orientation of an Unmanned Helicopter Based on Vanishing Line Information [J]," *Computer Engineering and Applications*, Vol. 9, 2004.

23. Lange, S., Sunderhauf, N., and Protzel, P., "A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments," 2009 International Conference on Advanced Robotics, 2009.

24. Lee, B., Saj, V., Benedict, M., and Kalathil, D., "A Vision-Based Control Method for Autonomous Landing of Vertical Flight Aircraft On a Moving Platform Without Using GPS," *arXiv preprint arXiv:2008.05699*, 2020.

25. Daly, J. M., Ma, Y., and Waslander, S. L., "Coordinated landing of a quadrotor on a skid-steered ground vehicle in the presence of time delays," *Autonomous Robots*, Vol. 38, (2), 2015, pp. 179–191.

26. Araar, O., Aouf, N., and Vitanov, I., "Vision based autonomous landing of multirotor UAV on moving platform," *Journal of Intelligent & Robotic Systems*, Vol. 85, (2), 2017, pp. 369–384.

27. Lee, B., *Helicopter Autonomous Ship Landing System*, Ph.D. thesis, 2018.

28. Lee, B., and Benedict, M., "Development and Validation of a Comprehensive Helicopter Flight Dynamics Code," AIAA Scitech 2020 Forum, 2020.

29. Ghamry, K. A., Dong, Y., Kamel, M. A., and Zhang, Y., "Real-time autonomous take-off, tracking and landing of UAV on a moving UGV platform," 2016 24th Mediterranean conference on control and automation (MED), 2016.

30. Hu, B., Lu, L., and Mishra, S., "Fast, safe and precise landing of a quadrotor on an oscillating platform," 2015 American Control Conference (ACC), 2015.

31. Kim, J., Jung, Y., Lee, D., and Shim, D. H., "Landing control on a mobile platform for multi-copters using an omnidirectional image sensor," *Journal of Intelligent & Robotic Systems*, Vol. 84, (1), 2016, pp. 529–541.

32. Xia, K., Lee, S., and Son, H., "Adaptive control for multi-rotor UAVs autonomous ship landing with mission planning," *Aerospace Science and Technology*, Vol. 96, 2020, pp. 105549.

33. Vlantis, P., Marantos, P., Bechlioulis, C. P., and Kyriakopoulos, K. J., "Quadrotor landing on an inclined platform of a moving ground vehicle," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015.

34. Rodriguez-Ramos, A., Sampedro, C., Bavle, H., De La Puente, P., and Campoy, P., "A deep reinforcement learning strategy for UAV autonomous landing on a moving platform," *Journal of Intelligent & Robotic Systems*, Vol. 93, (1-2), 2019, pp. 351–366.

35. Lawther, A., and Griffin, M. J., “Motion sickness and motion characteristics of vessels at sea,” *Ergonomics*, Vol. 31, (10), 1988, pp. 1373–1394.

36. “Helicopter operating procedures for air-capable ships NATOPS manual,” , 2003.

37. Doerry, A. W., “Ship dynamics for maritime ISAR imaging.” Technical report, Sandia National Laboratories, 2008.

38. Couprie, M., and Bertrand, G., “Topological gray-scale watershed transformation,” Vision Geometry VI, Vol. 3168, 1997.

39. Förstner, W., and Gülch, E., “A fast operator for detection and precise location of distinct points, corners and centres of circular features,” Proc. ISPRS intercommission conference on fast processing of photogrammetric data, 1987.

40. Zhang, Z., “Flexible camera calibration by viewing a plane from unknown orientations,” Proceedings of the seventh ieee international conference on computer vision, Vol. 1, 1999.

41. Araki, N., Sato, T., Konishi, Y., and Ishigaki, H., “Vehicle’s orientation measurement method by single-camera image using known-shaped planar object,” *Int. J. Innov. Comput. Inf. Control*, Vol. 7, (B), 2011, pp. 4477–4486.

42. OPenCV, “Camera calibration and 3D reconstruction,” https://docs.opencv.org/master/d9/d0c/group__calib3d.html, 2021 (accessed April 17, 2021).

43. Levenberg, K., “A method for the solution of certain non-linear problems in least squares,” *Quarterly of applied mathematics*, Vol. 2, (2), 1944, pp. 164–168.

44. Marquardt, D. W., “An algorithm for least-squares estimation of nonlinear parameters,” *Journal of the society for Industrial and Applied Mathematics*, Vol. 11, (2), 1963, pp. 431–441.

45. Fischler, M. A., and Bolles, R. C., “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, Vol. 24, (6), 1981, pp. 381–395.

46. Andrew, A. M., “Multiple view geometry in computer vision,” *Kybernetes*, 2001.

47. Lee, B., “Intelligent Ship Landing,” https://youtu.be/ExkyUOdgYaw, 2021 (accessed April 17, 2021).