Unmanned Aerial Vehicle Landing on Maritime Vessels using Signal Prediction of the Ship Motion

Shadi Abujoub, Johanna McPhee, Cassidy Westin, Rishad A. Irani
Department of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Canada
shadi.abujoub@carleton.ca; johanna.mcphee3@carleton.ca; cassidy.westin@carleton.ca; rishad.irani@carleton.ca

***Abstract*—Unmanned aerial vehicles (UAVs) are becoming more prevalent in maritime operations. For safe operation, one of the key challenges of using UAVs at sea is the relative motion that exists between the UAV and ship. For perpetual maritime operations, UAV systems need to be able to land safely on ocean vessels. Determining a 'quiescent period', where the roll and pitch angles of the ship are below a danger threshold, is a challenging problem for UAV systems. In general, current strategies rely on reactive systems and often use sensors on board the maritime vessel. The scope of the current paper is a proof-of-concept methodology which uses a signal prediction algorithm to facilitate safer autonomous UAV-ship landings. This study uses laser ranging and detecting devices (LIDAR) in conjunction with a signal prediction algorithm (SPA) to forecast when the ship motion is within safe landing limits. ShipMo3D was used to generate twelve trial cases for UAV-ship landings on a 33 m ship. The results show that with the use of the SPA, the number of UAV landing attempts was decreased by an average of 2 attempts, per test case, when compared to a system that did not use an SPA. Moreover, the results indicate that with revised tuning of the SPA, the likelihood of a safe landing can be further improved.**

I. Introduction

Unmanned aerial vehicles (UAV) are used for widespread civilian and military purposes such as surveillance [1], oceanographic data collection [2], wildlife monitoring [3] and search and rescue [4]. For any vertical take-off and landing (VTOL) vehicle, ship deck landings present challenges that encompass both self-induced dynamics and the stochastic nature of ship motion [5], [6]. Determining safe times to land on a ship typically relies on the skill of the pilot to account for and to predict changes in ship motion and pose.

Current UAV-naval systems implement combinations of GPS, visual systems, and mounted radar and radio frequency beacons to orient the UAV in relation to the ship deck [7]. Researchers have examined methods that allow for autonomous flight control, and landings on both stationary and moving decks. Garratt and Anavatti [8] used a neural network controller to produce heave trajectories for an unmanned helicopter. Hervas et al. [9] developed a landing control algorithm for unmanned vehicles on moving platforms that controlled the landing based solely on the relative heave motion between UAV and ship deck. While their simulations showed the algorithm was effective, the algorithm did not discriminate between safe landing times nor provide a method for tracking the ship trajectory. Ngo and Sultan [10] presented a model predictive control (MPC) design for helicopter shipboard operations in the presence of ship airwakes and rough seas. While the MPC method proved to be feasible in simulations, the researchers did not address the issue of the compatibility of their control method to other helicopter-ship combinations other than that which was simulated. A need therefore arises for a method that is more general for various ships and autonomous crafts.

Determining safe landing times is an important consideration for safe autonomous landings. Scherer et al. [11] were successful in mapping safe landing zones (LZ) on a ship deck using three-dimensional scanning Laser Detection and Ranging (LIDAR) and GPS/inertial navigation systems. Ferrier et al. [12] used the energy index (EI) to predict quiescent periods in the ship motion. Determining the EI requires lateral and vertical velocities and accelerations as well as roll and pitch displacements and velocities and thus is dependent on communication between the autonomous craft and ship. Although prediction of safe landing times is useful, integrating the prediction into a single autonomous landing system will require the prediction method to be independent of vessel type and without the need for ship to UAV communication.

Signal prediction independent of vessel type is currently used for active heave compensation on ships utilizing towing winches, allowing the system to actively pay-in or out line to the towed object to reduce transmitted motion of submerged equipment [13]. It is hypothesized that a similar motion prediction algorithm used by Woodacre et al. [13] and advanced by McPhee and Irani [14] could prove beneficial to UAV-ship landings, thus allowing UAVs to land within specified motion thresholds.

The authors of this paper provide a proof-of-concept (PoC) for increasing the success of UAV-ship landings that uses LIDAR technology for ship tracking and signal prediction to forecast safe landing times. The proposed system relies on information to guide the UAV to the LZ and land without any assistive deck-mounted hardware. The current study does not focus on the method to guide an UAV to the LZ, rather, the study focuses on the descent while over the LZ. The landing descent is achieved without assistive deck-mounted hardware, however, there are three LIDARs mounted on the UAV which provide the input for the motion prediction algorithm to forecast quiescent periods. Quiescent periods are determined based on the roll and pitch of the ship deck. The ship heave

978-1-5386-4814-8/18/$31.00 ©2018 IEEE

motion is monitored for UAV altitude control; however for the scope of this paper, heave is not considered for predicting safe landing opportunities.

The PoC uses a model-based approach in the MATLAB modeling environment, SimScape, and for the ship motion, time series outputs from ShipMo3D are used. The following section introduces the formulation and tuning of the signal prediction algorithm (SPA). The overview of entire PoC system model is presented in Section III where the UAV test platform, flight sequence and control scheme are described. Section IV describes how the LIDARs are modeled in the simulation environment for the purpose of detecting the pose of the ship. Test conditions are presented in Section V and the generation of ship motion is discussed. Simulation results are analyzed in Section VI and in Section VII the paper ends with concluding remarks and future work.

II. Signal Prediction Algorithm – SPA

In the current application, a signal prediction algorithm is used to predict future safe landing times in the roll and pitch motion of the ship. To predict the motion, it is assumed that both the roll and pitch motion are periodic and can be decomposed into *N* sine waves. Using the method advanced by McPhee and Irani [14], the signal prediction algorithm is composed of three distinct parts: mode detection, state estimation and prediction. During mode detection, the mean values of the different modes, i.e. amplitude, frequency and phase, are identified via a Fast Fourier Transform (FFT). An observer is initialized by these mean values and adapts the signal parameters on-line. The adapted parameters are then used to predict the motion.

As a function of time, *t*, the periodic motion *s* can be expressed as

$$s(t) = \sum_{i=1}^{N} A_i \sin\left(2\pi f_i t + \phi_i\right) + v(t) \tag{1}$$

where *N* is the number of identified modes, A_i, f_i and ϕ_i are the amplitude, frequency and phase of the identified mode *i* respectively. The offset term *v(t)* accounts for any deviation from the zero mean. For on-line estimation, a discrete Kalman filter estimates the motion and adapts the parameter vectors by comparing the measured motion *s(t)* with the modeled motion. With the adapted observer parameters A_{obs} and ϕ_{obs} in place of A_i and ϕ_i, equation (1) is used to forecast the motion sequence over a prediction horizon PH. Summing over each mode *i* at time step *k* the predicted motion sequence $s_{Pred}(t_k + PH)$ is

$$s_{Pred}(t_k + PH) = \sum_{i=1}^{N} A_{obs,i,k} \sin\left(2\pi f_i (t_k + PH) + \phi_{obs,i,k}\right) + v(t) \tag{2}$$

The prediction s_{Pred} from equation (2) was shown to be accurate in forecasting ship motion over short-term prediction horizons up to 3 seconds [14]. For longer prediction horizons, the error of the predicted signal increases significantly and should not be used in applications requiring an exact motion profile. However, when looking at trends in the signal, such as future amplitudes in the signal for a Go-NoGo state, the SPA can give useful information over long-term predictions.

A Go-NoGo state is determined given a set of criteria that defines a desired outcome. When conditions are favorable, a Go state is output and when conditions are unfavorable, a NoGo state is output. To tune the SPA for the purposes of determining a Go-NoGo state based on long-term predictions, the criteria for outputting a Go state must be described such that the increasing error associated with longer prediction horizons is taken into account. Specifically, a Go criterion that requires the forecast signal to fall within 1 standard deviation of the mean of the prediction was shown to produce a Go-NoGo state in agreement to the desired outcome for prediction horizons up to 10 seconds [14]. In the work described in the following sections, the same Go criterion is used.

A. SPA Tuning for a Parallel Configuration

For the PoC, both the roll and pitch motion of the ship are monitored for determining safe landing times i.e. the roll and pitch angles must fall within threshold values in order for a Go state to be output and land attempt to be made. Two SPAs in parallel aid in monitoring the roll and pitch motion by forecasting each motion independently over the prediction horizon PH. Considering both signals are expected to differ in amplitude and frequency, it is important that each SPA is initialized separately.

Tuning was performed in accordance to McPhee and Irani [14], however a recursive method for determining the peak sensitivity parameter for the SPA mode detection was explored such that the sensitivity parameter can be amended on-line to better suit each signal with time. The latter is achieved during the SPA operation where the amplitudes of the incoming motion signal are monitored and the sensitivity parameter is assigned and updated based on the maximum amplitude observed in the signal. The peak sensitivity PS describes the minimum peak height; a threshold value used by the peak detection algorithm to determine the dominant peaks in the FFT spectrum. For the work presented in this paper, PS is re-evaluated every 5 seconds and is described as

$$PS = \begin{cases} 0.05 & , \quad 0^\circ \leq |s| \leq 4^\circ \\ 0.1 & , \quad 4^\circ < |s| \leq 10^\circ \\ 1 & , \quad |s| > 10^\circ \end{cases} \tag{3}$$

where $|s|$ is the roll or pitch input to the SPA from equation (1). The PS parameters in equation (3) were manually input and future work will be required to refine the PS parameters based on SPA performance.

III. Methodology

For the PoC, a quadrotor was chosen which would use the SPA onboard a UAV to facilitate with landing descent. Quadrotors are an established flight platform with well documented

dynamics and relatively simple actuation and control when compared to other rotor-craft. The mass moment of inertia, mass, and flight dynamic characteristics may be manipulated in SimScape to emulate a specific test vehicle. This study aims to result in a system independent of the flight vehicle and transferable to other platforms. The following sections discuss the UAV, model system control and simulation progression.

A. UAV Platform

The UAV system has a mass of 4 kg and four arms spanning 0.6 m in length. Each arm extends from the center body and carries a motor which actuates a 0.25 m blade. The motors are generic DC brushless motors that require a 12 V power supply. Physical testing of the motors produced an rpm to thrust relation that is used in the model.

Fig. 1 highlights the control scheme for the UAV. The four motors are actuated via a motor mixer which delegates motor power based on the incoming altitude, pitch, roll, and yaw commands. Altitude is selected via flight sequence logic, and is maintained through proportional-integral-derivative control (PID). Yaw is controlled using PID control, although the current model does not actively alter yaw from its initial position as quad rotors have the ability to traverse freely without re-orienting their heading. Pitch and roll have standard PID controllers with a corrective angular acceleration feed-forward loop to minimize the oscillatory motion associated with underactuated systems. Nominal ship tracking is achieved using two outer loop proportional and integral (PI) controllers which feed into the roll and pitch controllers. Within the simulation, the ship tracking controller relies on the X and Y coordinates from the ship along with the UAV's X and Y positions in the world frame. These control schemes allow the UAV to track the position of the landing zone when the controller is provided the position as a function of time. Aside from the ship LZ coordinates, the UAV model is not provided with any other information from the ship. Future users of the current work can use their own methods to control the UAV's flight dynamics and their methods to track the location of the LZ.

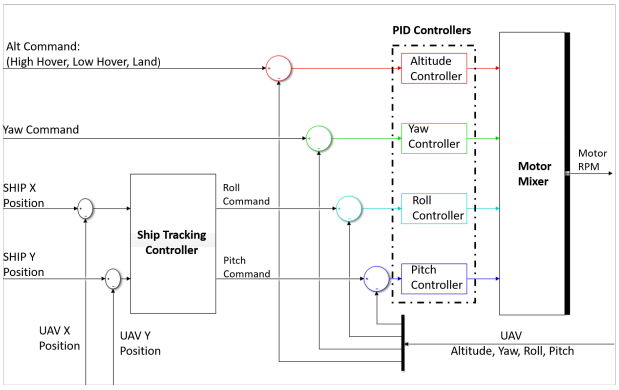

Fig. 1: Model Control

B. Model Construction and Logic Overview

Fig. 2 is a high-level diagram of the information flow within the simulator model. The ship motion data is imported into the simulation work space and applied to the model of the ship body as a function of time. The UAV is within the same model and launches a set distance behind the ship. The UAV actively tracks the position of the ship using the control methods described. While above the ship, each LIDAR measures the distance to the deck and relays that information to an algorithm for ship pose detection. The ship pose detection algorithm accounts for the position of the UAV in space, the known orientation of the LIDARs and the magnitude of each LIDAR reading. Ship pitch and roll angles are then transferred to the SPA to determine when it is safe to land.

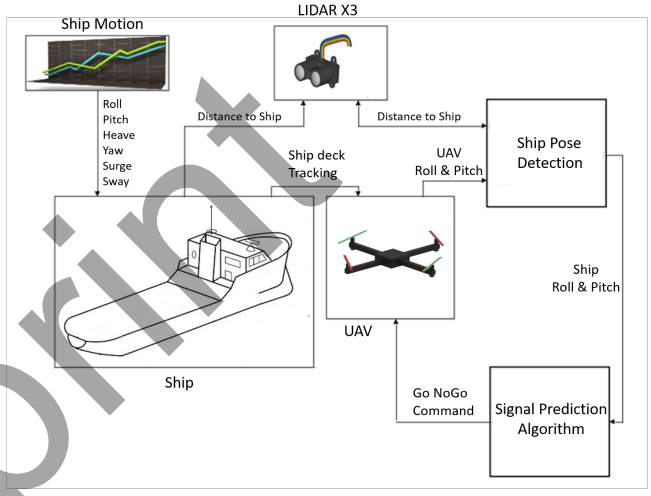

Fig. 2: Model Information Flow

C. UAV Flight Sequence

Fig. 3 highlights the three flight stages:

1) Launch
2) Ship Tracking
3) Landing

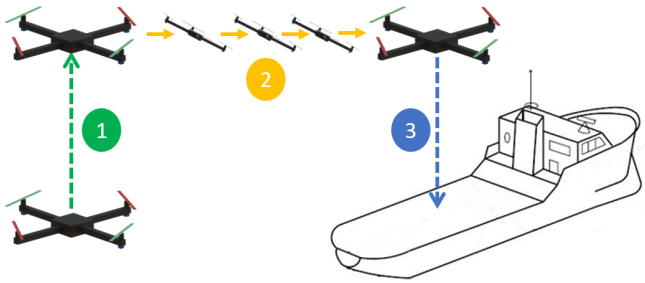

Fig. 3: Simulation Flight Sequence

In phase 1, the UAV ascends vertically to its cruising altitude. The cruising altitude is typically dependent on the mission of the UAV. The simulations in this work use a

cruising altitude of 10 m to expedite the landing phase. The vertical takeoff velocity was conservatively selected as 1 m/s, but it is a parameter which may be adjusted based on UAV performance. During phase 2, to emulate the LZ homing, the ship position is continuously relayed to the UAV tracking algorithm. Operating within predefined pitch and roll thresholds, the UAV proceeds toward the landing zone while maintaining its 10 m altitude.

The landing operation, final descent, is highlighted in phase 3 and the logic progression is outlined in Fig. 4. Phase 3 begins when the UAV is within proximity to the ship. At this point, the LIDARs power on and the ship pose detection algorithm sends data to initialize the SPA. During initialization, the UAV descends to its low hover position of 3 m relative to the deck. The SPA continually sends a Go or NoGo command based off of predicted future ship roll and pitch amplitudes. As the distance between the UAV and the deck decreases, the SPA prediction horizon PH decreases by the following relationship

$$PH = \begin{cases} \frac{L_d}{V_L} & , \quad 5 \leq \frac{L_d}{V_L} \\ 5 & , \quad 5 > \frac{L_d}{V_L} \end{cases} \tag{4}$$

where L_D is the UAV's distance to the ship, and V_L is the landing velocity. A maximum prediction horizon of 5 s was selected due to the diminishing accuracy of longer SPA predictions. With a rate of decent of 0.5 m/s the UAV would take 6 seconds to land if there was no heave motion. However, in these tests the heave varied by up to 1.5 m resulting in landing times spanning 3 s to 9 s. Adequate results were achieved using a maximum prediction horizon of 5 s. To compensate for undesirable heave motion, future work could incorporate an SPA to predict heave motion coupled with a model predictive controller for UAV altitude during landing. Uncompensated heave motion may lead to dangerous landing scenarios such as high impact or failed landings. Currently, when the SPA predicts a Go condition, the UAV will continue to lower its altitude regardless of heave motion. If a NoGo condition is predicted, the UAV will return to its low hover position until a Go command is received.

D. Go-NoGo Command Latching

To remove unwanted fluctuation in the Go-NoGo command signal, McPhee and Irani [14] proposed a latching algorithm that latches to a Go or NoGo command. The proposed latching algorithm evaluates the incoming Go-NoGo commands, determined by the SPA, and latches to a command if no change is observed in the Go-NoGo commands over an evaluation interval. Before re-evaluating and re-latching, a latched state will run for the selected time period, regardless of the incoming Go-NoGo states.

It is proposed here that the evaluation interval should be selected based in part on the response of the physical UAV components, and that the latching period should be selected based on the UAV's rate-of-descent. For the purposes of the PoC, an evaluation interval of 0.4 s and a latching period of 1 s were selected. In practice, physical tests should be used to evaluate both intervals which may change based on UAV response times and rates-of-descent.

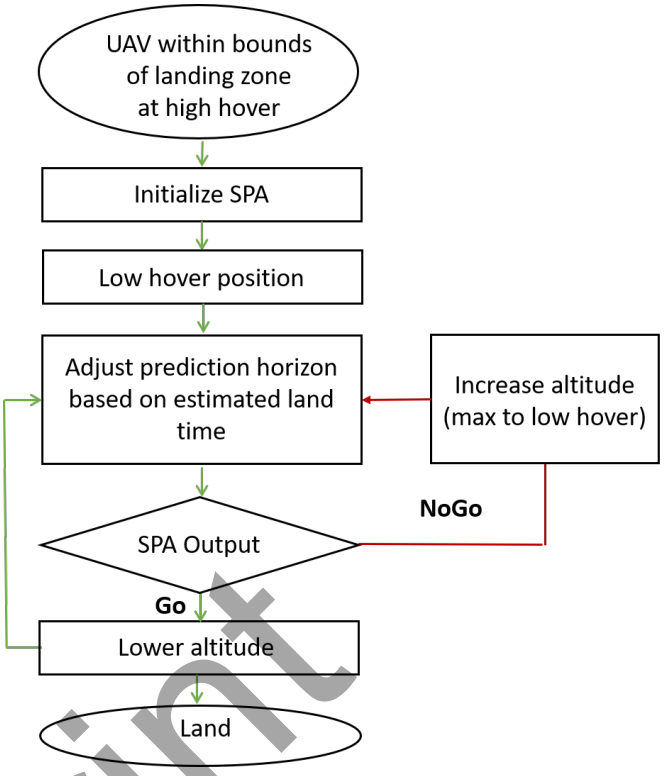

Fig. 4: Flowchart explaining the logic of the landing sequence and SPA use

IV. LIDAR Simulation

The LIDAR sensors in the model are idealized, where process and measurement noise are not addressed. However, this section will later address noise and the concepts to minimize signal disruption. Fig. 5 illustrates the notation which will be used in the following section to derive the output of each LIDAR. In order to simulate the LIDAR, the location of each LIDAR as well as the plane of the ship deck must be found. Three points $l_{i:1,2,3}$ were placed under the UAV as a reference location for each LIDAR. Points l_i are represented using position vectors $\vec{L}_i$ with respect to the global coordinate frame, where i corresponds to either the 1^{st} 2^{nd} or 3^{rd} LIDAR. Using the cross product between vectors $\vec{L}_i$ the normal vector is found by:

$$\vec{n}_L = (\vec{L}_1 - \vec{L}_2) \times (\vec{L}_1 - \vec{L}_3) \tag{5}$$

Vector $\vec{n}_L$ is the normalized to form the UAV's unit normal vector $\hat{n}_L$. Similarly within the simulation, three points $s_{i:1,2,3}$, and corresponding position vectors $\vec{S}_{i:1,2,3}$, were placed on the deck of the ship. Using the cross product between position vectors $\vec{S}_i$ the normal vector of the ship is found and then normalized $\hat{n}_s$.

Using the UAV and ship orientation, the simulated LIDAR outputs must now be found. Through vector addition, an

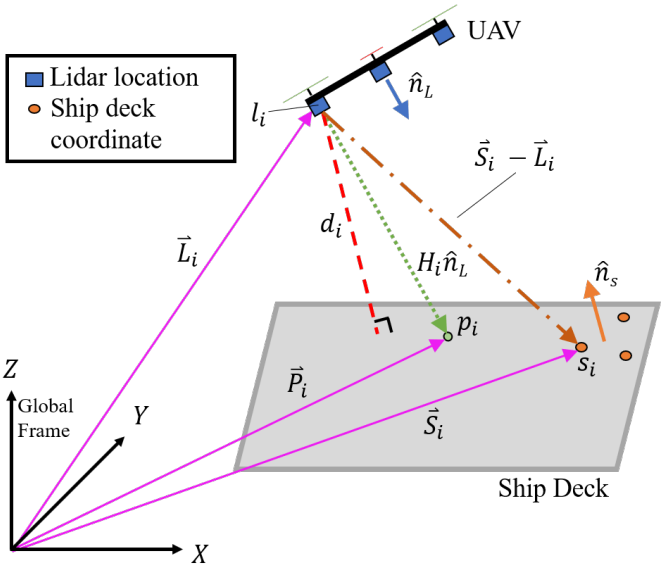

Fig. 5: Schematic of UAV and the notation used to derive the simulated LIDAR outputs

equation which relates the magnitude and direction of the LIDAR output to its final and initial vectors is

$$H_i\hat{n}_L = \vec{P}_i - \vec{L}_i \tag{6}$$

where H_i is the magnitude of the LIDAR output and $\vec{P}_i$ is the position vector of point p_i where the LIDAR intersects the plane of the ship deck. The shortest distance between a LIDAR and the plane of the deck d_i can be found by taking the dot product between the ship's normalized unit vector with any vector originating at the LIDAR and intersecting a known point on the ship's plane, thus:

$$d_i = \hat{n}_S \cdot (\vec{S}_i - \vec{L}_i) \tag{7}$$

d_i will always be perpendicular to the plane of the ship regardless of the pose of the UAV. The same offset may be expressed in terms of the unknown position vector $\vec{P}_i$ as

$$d_i = \hat{n}_S \cdot (\vec{P}_i - \vec{L}_i) \tag{8}$$

The magnitude of each LIDAR H_i is then found by substituting equation 6 into equation 8,

$$H_i = \frac{d_i}{\hat{n}_s \cdot \hat{n}_L} \tag{9}$$

In practice, the hardware LIDARS would output H_i. Within the simulation, the magnitude of the LIDARS has been found through the above process, the pose of the ship can be now be determined.

A. Ship Pose Detection

The LIDARs pass their magnitude information H_i into a separate algorithm which calculates the ship pose based on variations in the LIDAR outputs and angular pose of the UAV. This ship pose algorithm would be implemented with a physical UAV to determine the ship's deck motion. Fig. 6 illustrates the deck motion in relation to the UAV and notation. The algorithm works by identifying the three points, p_{UAV_i}, where the lasers intersects the deck with respect to the UAV. Without any noise, the points of intersection will be equivalent to p_i from the previous section, but will be found in the UAV's coordinate frame. Rotation of the UAV during flight alters the LIDARs directional vector which initially faces down $[0, 0, -1]$. To properly orient the LIDAR vectors during UAV manoeuvres, the vectors are multiplied by a corresponding rotation matrix. Following the right hand rule, the UAV pitch and roll angles are taken from the UAV's inertial measurement unit (IMU), where rotations about the x-axis are roll Φ and and rotations about the y-axis are pitch Θ. The corresponding rotation matrices are:

$$R_x(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\Phi & -sin\Phi \\ 0 & sin\Phi & cos\Phi \end{bmatrix} \tag{10}$$

$$R_y(\Theta) = \begin{bmatrix} cos\Theta & 0 & sin\Theta \\ 0 & 1 & 0 \\ -sin\Theta & 0 & cos\Theta \end{bmatrix} \tag{11}$$

Using the above rotation matrices all LIDAR vectors are properly oriented in the UAV's coordinate frame. It is then possible to solve for the position vectors for each LIDAR P_{UAV_i} in the UAV's frame

$$\vec{P}_{UAV_i} = R_x(\Phi)R_y(\Theta)(\vec{c}H_i + \vec{k}_i) \tag{12}$$

where $H_i\hat{c}$ is the vector formed between the LIDARs magnitude and the original directional vector for each LIDAR, and $\vec{k}_i$ is the LIDAR's positional offset from the UAV's center of rotation.

Similar to equation 5, the unit normal vector of the ship $\hat{n}_S$ is found by normalizing the cross product between $\vec{P}_{UAV_i}$. The first index of $\hat{n}_S$ is the ship's roll angle and the third index is the pitch angle. The ship's roll and pitch angles are computed in the UAV's non-rotating body frame and are therefore equivalent to the ship's pitch and roll in the global frame. These roll and pitch angles are subsequently fed to their respective SPA. Without LIDAR noise, the normalized unit vector will be equivalent to $\hat{n}_s$ found in the previous section. The addition of noise complicates the system and the potential of error increases.

B. LIDAR Selection and Noise

For the PoC, the LIDARs were selected and modeled as Garmin LIDARlite single pulse LIDARs that have a specified accuracy of ± 2.5 cm at 10 m. The LIDARs were simulated in idealized circumstances; however as model fidelity is increased, conditions will no longer be ideal with the addition of noise. With added noise, discrepancies of ± 2.5 cm on each LIDAR will result in large fluctuations in the calculated normal vector of the ship. Moreover, the smaller the grouping of points

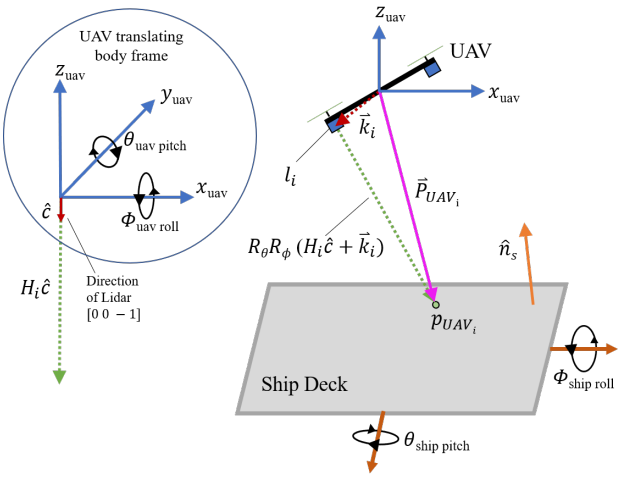

Fig. 6: Ship pose detection

p_{UAV_i} yields greater associated error in the calculated values. Errors may be minimized by either changing the viewing angle of the LIDARs or spacing the LIDARs further apart. Inconsistencies in deck texture will also contribute adversely to ship pose estimation. Furthermore, the SPA will attempt to predict a noisy signal unless the input signal is filtered. Since noise cannot be assumed to be periodic, prediction error accumulates immediately; hence the noisier the signal, the more likely it is that the forecast signal will output false Go-NoGo states. It is therefore important that most noise in the LIDAR signals be filtered out before sending the signals to the SPA.

Kalman filters and other filtering techniques are currently being explored to minimize process noise. If process noise is not sufficiently reduced in future models and physical testing, flash LIDARs may be introduced. A flash LIDAR incorporates a 2D array of photo diode sensors capable of detecting a point cloud at every time step. A large point cloud allows for a multitude of normal vectors to be found for the ship. Using neighbor polling, a technique which compares normal vectors in a point cloud to each other, and with the increased number of points, the more likely normal vector for the ship can be found.

V. Simulation Test Conditions

Using ShipMo3D [15], a 33 m craft was constructed which is comparable to a medium sized patrol or training vessel. Ship motion was simulated at three speeds of 6, 8, and 10 kn and at headings of 15°, 30°, 60°, and 90° in a seaway. The seaway was modelled with a unidirectional Bretschneider spectrum where the significant wave height was 3.25 m and a peak wave period of 9.7 seconds. Table I overviews the testing parameters of the twelve simulation trials. The root mean square (RMS) value of pitch, roll, and heave was calculated for each trial and included in Table I.

Two simulations were conducted for each trial, one with the SPA on and one with the SPA off. The SPA thresholds

TABLE I: TEST PARAMETERS USED IN SIMULATIONS

Generated Ship Motion					
Trial	Velocity [kn]	Heading	RMS Pitch	RMS Roll	RMS Heave [m]
1	6	15°	2.3°	1.8°	0.5
2	8	15°	2.4°	1.9°	0.5
3	10	15°	2.5°	2.0°	0.5
4	6	30°	2.1°	3.5°	0.5
5	8	30°	2.2°	3.9°	0.5
6	10	30°	2.3°	4.2°	0.5
7	6	60°	1.4°	6.5°	0.5
8	8	60°	1.5°	6.9°	0.5
9	10	60°	1.5°	7.4°	0.5
10	6	90°	0.1°	7.7°	0.5
11	8	90°	0.1°	7.7°	0.5
12	10	90°	0.2°	7.7°	0.5

were selected from the U.S Coast Guard *Ship-Helicopter Operational Procedures Manual* [17]. The *Go* threshold was set as 2° for pitch and 5° for roll. In every simulation the UAV follows the flight procedure outlined in Fig. 3 having a high hover altitude of 10 m, and a low hover altitude of 3 m above the ship deck. For all test cases, the UAV will commit to landing if the LIDAR output is less than 1 m, regardless of the ship pose.

VI. Results and Discussion

Fig. 7 highlights results from Trial 3 with the conditions outlined in Table I. The first subplot of Fig. 7 shows, as functions of time, the Go-NoGo state and the distance of the UAV to the deck of the ship for the cases when the SPA is active and inactive. The dotted light blue line provides the UAV distance from the ship deck when the SPA is active, while the dashed pink line provides the UAV distance from the ship deck when the SPA is not used. A zero value for the UAV distance from the ship deck indicates that the UAV has touched the surface of the ship deck and is considered a landing, the landings are indicated by a circle in the first subplot of Fig. 7. In the same subplot, the SPA Go-NoGo output is displayed as a blue dash-dot line and the instantaneous Go-NoGo, not based on a prediction, is displaced as a solid green line. A non-zero value of the Go-NoGo signals indicate a Go state. The next subplot of Fig. 7 shows, as a function of time, the ship pitch as a solid line and corresponding NoGo limits, dashed lines. Similarly, the third subplot of the figure shows the ship roll. In the global frame and as a function of time, the last subplot of Fig. 7 plots the ships heave motion with a solid black line, and the UAVs vertical position for when the SPA is used, dotted blue line, and when the SPA is not used, dashed pink line.

During the simulation, at $t = 33.5$ s the UAV reaches the ship, initiating the recovery phase and activating the LIDARs. The SPA begins receiving pitch and roll information as soon as the LIDARs power on. Once centered above the landing zone, $t = 38$ s, the UAV proceeds to its low hover position

of 3 m relative to the ship deck — indicated by the gradual decreasing value of the UAV altitude in the forth subplot. In the simulation, the transition from a high hover to a low hover is not affected by wave action. The number of landing attempts for the UAV is also recorded for each test case: SPA-on and SPA-off.

Without the SPA, the UAV landed at $t \approx 58$ s, which happened to be during unsafe conditions due to pitch exceeding the 2° threshold. This landing was preceded by 6 aborted landing attempts. Although the ship pose detection algorithm indicated a *NoGo* condition, the UAV was forced to land as it was not able to ascend faster than the heave action of the ship. The 'forced' landing could result in damage to the UAV or the ship. The results of this simulated test case highlight the potential benefit of the SPA system.

Under the same testing conditions, but using the SPA, the landing was delayed 13 s, but occurred during a *Go* condition. The UAV successfully landed on the 1st attempt, whereas without the SPA the UAV had 6 failed landing attempts before ultimately landing in an unsafe condition.

Table II highlights the results of the trials conducted with the twelve sets of ship motion data. Without the SPA, the UAV landed within safe ship motion thresholds 83% of the time compared to 92% with the SPA.

In Trial 8, with the SPA on, following an aborted landing descent, the UAV had a forced landing due to rising ship heave coupled with the UAV's land safe mechanism which attempts to land the UAV when the distance to the ship is less than 1 m. Although the overall ship motion was within the threshold limits, a *NoGo* condition was the output from the SPA, resulting in a marginally successful landing. However, the simulations did not account for impact forces during forced landings which could adversely effect the UAV. The SPA in the current implementation has trouble with very low frequency signals which adversely affected the results of Trial 8. Using the same axes as Fig. 7, Trial 8 is displayed in Fig. 8, and the low frequency wave actions are indicated. Although the ship motion was safe, the SPA predicted the motion would exceed the *Go* criteria. It is hypothesized that a recursive method could be used to tune the SPA for low frequency motion.

The final 10 s of Trial 9 is highlighted in Fig. 9. Trial 9 resulted in the UAV landing during an unsafe condition when the SPA was used. In Trial 9, the SPA latched onto a Go command for an extended period of time, as well as having a delayed Go command. The results in Fig. 9 indicate the responsiveness of the SPA should be increased and that the latching time be decreased. A drawback of increasing the responsiveness is that the SPA is more likely to output false Go commands which may not fit the prediction horizon requirements. Furthermore, decreasing the latching time may lead to fluctuations in the Go-NoGo commands leading to an increased number in landing attempts. Future work will examine how to make the SPA more robust for the various sea states which the system could encounter.

For all trials, using the SPA increased the landing time by an average of 42 s, but with the added benefit of decreasing the number of landing attempts by an average of 2 attempts. Observing only cases where a safe landing occurred, excluding trials 3, 6 and 9, using the SPA increased the landing time by an average of 17 s, and decreased the average number of landing attempts by 4.

TABLE II: PRELIMINARY RESULTS

Trial	SPA	Land Attempts	Safe Landing	Land Time (s)
1	On	2	✓	66.7
	Off	3	✓	54.7
2	On	4	✓	98.8
	Off	5	✓	57.7
3	On	1	✓	65.4
	Off	7	✗	57.9
4	On	2	✓	68.6
	Off	3	✓	52.4
5	On	7	✓	137.7
	Off	15	✓	85.0
6	On	12	✓	200.7
	Off	3	✗	55.9
7	On	1	✓	56.7
	Off	4	✓	56.6
8	On	4	✓*	69.5
	Off	6	✓	63.1
9	On	8	✗	259.7
	Off	1	✓	57.8
10	On	2	✓	71.6
	Off	8	✓	63.1
11	On	1	✓	71.6
	Off	8	✓	63.1
12	On	1	✓	71.6
	Off	7	✓	63.6

VII. Conclusion and Future Work

The results of the proof-of-concept study show promise of incorporating the SPA Go-NoGo system on a UAV. When the SPA only considered the ship's pitch and roll, the number of landing attempts decreased, however, the time which was spent over the landing zone increased. Future work will examine noise sensitivity on the single point LIDARs and filtering methods. Although initial research focused on using the SPA for roll and pitch motion prediction, heave motion has created a challenging dynamic between the UAV and ship which must be addressed to ensure safe landings. Future work will incorporate an additional SPA to predict the ship's heave motion and the control of the UAV's altitude. Within the simulations, the 'forced' landing impact forces between the UAV and the ship could also be analyzed adding additional criteria for a safe landing. Further refining of the sensitivity evaluation function in equation (3) could be performed with the goal of continual mode detection for the SPA.

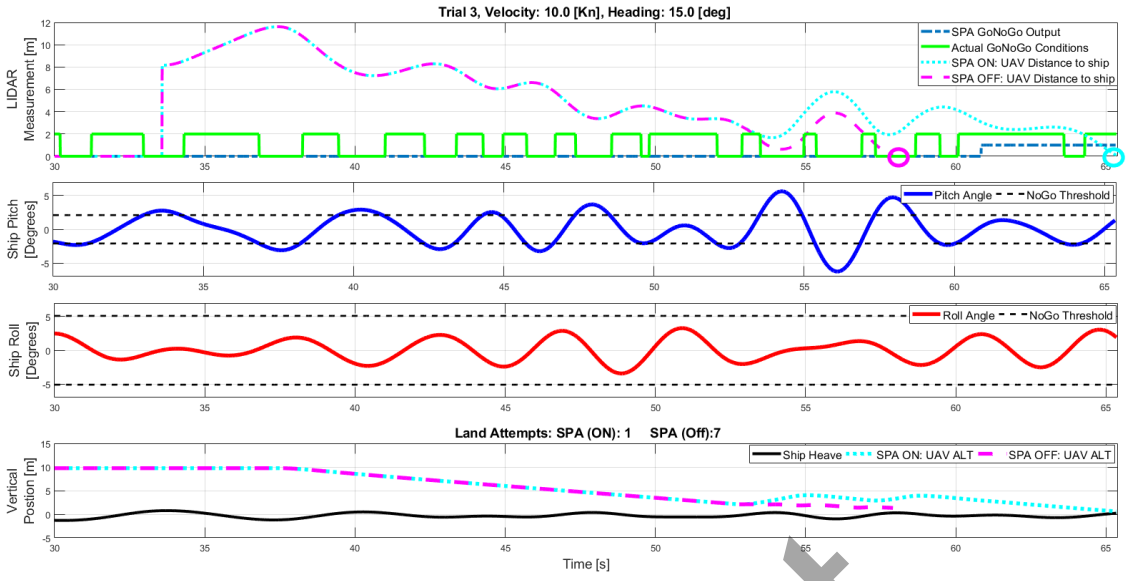

Fig. 7: Simulation Results of Trial 3

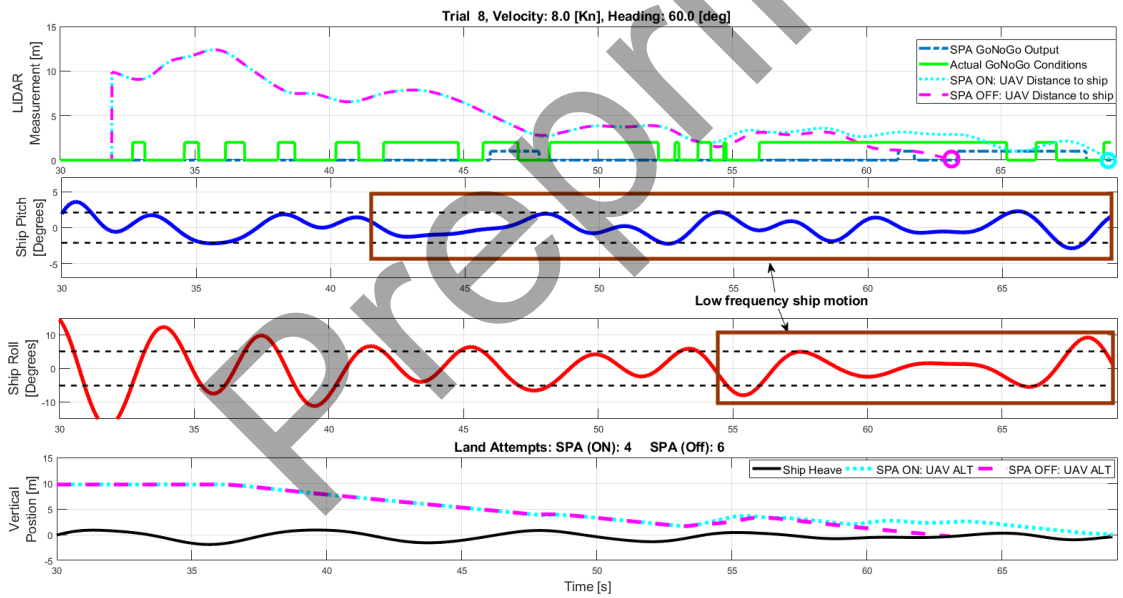

Fig. 8: Simulation Results of Trial 8

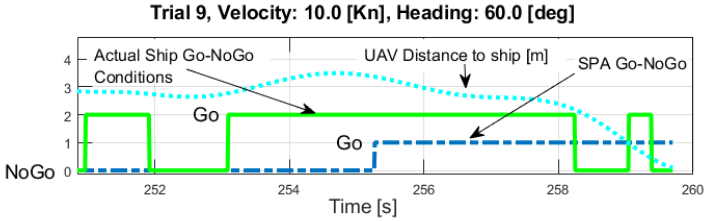

Fig. 9: Unsuccessful landing attempt using SPA of Trial 9

ACKNOWLEDGMENT

For their financial support, the authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would also like to thank DSA LTD (Dynamic Systems Analysis Ltd) for the in-kind donation of ShipMo3D. Finally, we thank Carleton University for their general support of the work.

REFERENCES

[1] S. O'Young et al., *RAVEN: A maritime surveillance project using small UAV*, 2007 IEEE Conference on Emerging Technologies and Factory Automation, Patras, 2007.

[2] M. Faria, et al., *Coordinating UAVs and AUVs for oceanographic field experiments: Challenges and lessons learned*, In IEEE Robotics and Automation (ICRA).

[3] J. Fortuna, et al., *Using low cost open source UAVs for marine wild life monitoring - Field Report*, IFAC Proceedings Volumes, 2013.

[4] P. Westall, et al.,*Vision-based UAV maritime search and rescue using point target detection*, In Proceedings AIAC12 - Twelfth Australian International Aerospace Congress, Melbourne, Australia 2007.

[5] D.W. Yeo, et al., *Downwash Detection and Avoidance with Small Quadrotor Helicopters*, Journal of Guidance, Control, and Dynamics, 2016.

[6] Ø.F. Auestad, et al., *Heave Motion Estimation on a Craft Using a Strapdown Inertial Measurement Unit*, IFAC Proceedings Volumes, 2013.

[7] Sankalp Arora, Sezal Jain, Sebastian Scherer, Stephen Nuske, Lyle Chamberlain and Sanjiv Singh, *Infrastructure-free Shipdeck Tracking for Autonomous Landing*, The Robotics Institute, Carnegie Mellon University,2013

[8] M. Garratt et al., *Non-linear control of heave for an unmanned helicopter using a neural network*, Journal of Intelligent and Robotic Systems, 2012.

[9] J. Hervas, et al., *Automatic Landing Control of Unmanned Aerial Vehicles on Moving Platforms*, in IEEE 23rd International Symposium on Industrial Electronics, Istambul, 2014.

[10] T. Ngo et al., *Nonlinear Helicopter and Ship Models for Predictive Control of Ship Landing Operations*, in AIAA Guidance, Navigation, and Control, National Harbor, 2014.

[11] S. Scherer, et al., *Autonomous landing at unprepared sites by a full-scale helicopter*, In Robotics and Autonomous Systems, 2012

[12] B. Ferrier, et al., *Fire Scout Launch and Recovery Considerations in Parametric- Like Roll Condition*, In the proceedings of American Society of Naval Engineers: Launch and Recovery Symposium, Linthicum, Maryland, 2016.

[13] J.K. Woodacre, R.J. Bauer, R.A. Irani, *Hydraulic valve-based active-heave compensation using a model-predictive controller with non-linear valve compensations*,Ocean Engineering, Volume 152, 2018, Pages 47-56.

[14] J. McPhee et al., *On-line Determination of a Go-NoGo State Using a Continous Estimation of the System Response*, Proceedings of The Canadian Society for Mechanical Engineering International Congress, Toronto, Canada, 2018.

[15] Kevin McTaggart, *ShipMo3D version 3.0 user manual for creating ship models*, Tech. Rep. TM 2011-308, Defence Research and Development Canada, 2012.

[16] National Defence, and Royal Canadian Navy. Patrol Craft Training Vessels (Orca-Class). Canada.ca, 14 Sept. 2015,

[17] United States Coast Guard, *Shipboard-Helicopter Operational Procedures Manual*, COMDTINST M3710.2D, Pages 464-490 (2001)

[18] R. Lozano, et al., *2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems*. Volume 46, Issue 30, Pages 1-359 (2013)