Landing a UAV in Harsh Winds and Turbulent Open Waters

Parakh M. Gupta, Éric Pairet, *Member, IEEE*, Tiago Nascimento, *Senior Member, IEEE*, and Martin Saska, *Member, IEEE*

***Abstract*—Landing an unmanned aerial vehicle unmanned aerial vehicle (UAV) on top of an unmanned surface vehicle (USV) in harsh open waters is a challenging problem, owing to forces that can damage the UAV due to a severe roll and/or pitch angle of the USV during touchdown. To tackle this, we propose a novel model predictive control (MPC) approach enabling a UAV to land autonomously on a USV in these harsh conditions. The MPC employs a novel objective function and an online decomposition of the oscillatory motion of the vessel to predict, attempt, and accomplish the landing during near-zero tilt of the landing platform. The nonlinear prediction of the motion of the vessel is performed using visual data from an onboard camera. Therefore, the system does not require any communication with the USV or a control station. The proposed method was analyzed in numerous robotics simulations in harsh and extreme conditions and further validated in various real-world scenarios.**

***Index Terms*—Aerial systems: mechanics and control, UAV, MPC, optimization and optimal control, multi-robot systems, dynamics.**

I. Introduction

HETEROGENEOUS robot teams that are composed of UAVs and USVs are aimed to provide higher efficiency and decrease the high risk posed to human life in marine applications. An example of such an application is the process of cleaning oceans to rid them of oil spills and non-biodegradable waste [1]. While the UAVs can act as the eyes in the sky for surveying, identifying, and localizing the clean-up targets, the USVs are much better suited to the actual clean-up as this task requires heavy equipment and lifting capabilities close to the water surface.

These clean-up missions can be performed autonomously by UAVs and can be conducted several dozen kilometers away from a harbor or shore. Although UAVs have short battery lives to be able to fly long distances, their strength lies in their agility and their ability to perform short-duration hover missions [2]. We can compensate for this short battery life by making a UAV and USV behave as a team, where-in the UAV can charge quickly during the mission for rapid redeployment. However, the precipitous and violent nature of the sea poses daunting challenges for landing on the USV deck, especially due to the precision required for recharging operations.

When landing on a USV, the first challenge is estimating and predicting the movement of the deck of the USV before landing. A fast-moving deck can damage the UAV during landing through high impulse impacts, while a tilted deck can result in the UAV rolling or falling off the deck before the landing is complete. Additionally, a tilted deck can cause an erroneous response from the controller of the UAV during landing, which would jeopardize the landing position since multi-rotors are under-actuated vehicles with coupled angular and linear acceleration vectors. The second challenge that we focus on is attempting a landing without active communication between the UAV and the USV. Relying on a required communication channel with a high frame rate and low latency would introduce a significant source of failure in real open-water applications. To increase reliability and applicability, we attempt to build a decentralized solution that does not rely on communication between the agents. Thus, we aim to study various aspects of the dynamics of UAVs and USVs to develop a framework for predicting and landing on the USV with high precision and reliability in demanding conditions including wind and waves, often seen in harsh environments. Finally, in this work, we can define harsh environments as those that contain open water with waves with a height of up to 4 meters, and a wind velocity of up to 12 m/s, which corresponds to a Beaufort scale of 6. For intended applications, this would produce a tilt in the range of $[-0.5,0.5]$ radians for the USV.

II. Related Works

Riola et al. [3] show that the behavior of a ship can be predicted based on its past motion up to short prediction horizons if corrected by measured ship motion. Unsurprisingly, the topic of wave predictions is highly relevant to the shipping industry as it

Manuscript received 16 August 2022; accepted 5 December 2022. Date of publication 23 December 2022; date of current version 30 December 2022. This letter was recommended for publication by Associate Editor M. Garratt and Editor P. Pounds upon evaluation of the reviewers' comments. This work was supported in part by the Technology Innovation Institute - Sole Proprietorship LLC, UAE, through Research Project under Grant TII/ARRC/2055/2021, in part by CTU under Grant SGS20/174/OHK3/3T/13, in part by the Czech Science Foundation (GAČR) through Research under Grant 20-10280S, in part by the European Union's Horizon 2020 Research and Innovation Program AERIAL-CORE under Grant 871479, and in part by the Ministry of Education of the Czech Republic through the OP VVV funded under Grant CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics". *(Corresponding author: Parakh M. Gupta.)*

Parakh M. Gupta and Martin Saska are with the Department of Cybernetics, Czech Technical University in Prague, 12000 Prague, Czech Republic (e-mail: guptapar@fel.cvut.cz; saska@labe.felk.cvut.cz).

Tiago Nascimento is with the Department of Cybernetics, Czech Technical University in Prague 58055-000 Prague, Czech Republic, and also with the Department of Computer Systems, Universidade Federal da Paraíba, Brazil (e-mail: tiagopn@ci.ufpb.br).

Éric Pairet is with the Technology Innovation Institute Abu Dhabi 9639, UAE (e-mail: eric.pairet@tii.ae).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3231831, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3231831

is needed to prevent cables from slacking while trying to offload cargo from ships using port-side cranes. Both Küchler et al. [4] and Neupert et al. [5] describe an active heave compensation for port-side cranes using a periodic oscillation model that proves to be effective.

Building on a similar model, Marconi et al. [6] and Lee et al. [7] present sophisticated approaches for fixed-wing UAVs landing on vessels. Both works adapt the model using a Kalman filter and use this heave motion of the ship to predict the altitude of the landing pad. However, these works do not focus on a rolling and pitching deck. Meng et al. [8] take a different approach and use an auto-regressive-model on the fixed-wing UAV to observe and predict the ship motion by breaking it into sinusoidal components. In addition, Ngo and Sultan [9] predict the quiescent periods for landing a helicopter on a ship based on the model of the vessel, but they do not tackle the problem of predictions of the motion for landing on an untilted deck. This leads to short opportunistic windows that have to be adhered to, even if the conditions change rapidly. All of the above-mentioned works present results only in simulation environments that are not harsh or extreme.

The research on solutions for multi-rotor aerial vehicles landing on marine vessels is recent. One of the first works by Polvara et al. [10] uses a fiducial marker located on the platform and an extended Kalman filter (EKF) that estimates the position of the USV. In contrast, the approach presented by Abujoub et al. [11] relies on a LiDAR onboard the UAV to find the pose of the landing pad to learn the behavior of the platform by hovering above it. However, they classify the window of landing into go or no-go intervals. Both preliminary works were validated in lenient simulation conditions.

More recently, researchers have begun testing their approaches through real-world experiments. For example, Xu et al. [12] use a fiducial marker for a decentralized approach, so as to follow the USV and use a PD controller for landing once the USV is discovered. For the second challenge of achieving decentralization, Lee et al. [13] present an interesting solution to finding a ship and its pose using classical vision algorithms. Zhang et al. [14] take a different approach and present a learning-based linear controller that receives inputs from a fiducial marker to land the UAV on a USV that is subject to the waves of a lake. Furthermore, some works also present the application of an MPC controller that enables a flexible-blade helicopter to land on a marine vessel [15], [16]. These works use a non-linear MPC to achieve near-perfect performance but do so using a numerical benchmark that doesn't run in real-time or in a real-world experiment. Our work differs from these by using simplifications and a new approach that fills these gaps of real-time computing and applicability without a significant drop in landing performance. We use these for comparison in our experimental section to demonstrate the same. The most advanced research presented with real-world flight data is the work by Persson et al. [17], which presents an MPC for a UAV autonomous landing on a moving boat.

For the purpose of our work, we assume that the USV can be found by the UAV by ascending to a given altitude during the mission without the need for conducting a planned search which is beyond the scope of this letter. We also assume that the motion of the USV perpendicular to the water surface is minimal (the USV is waiting for the UAV to land while controlling its global positioning on the water in order to remain stationary, rather than drifting with the waves). Furthermore, we assume that the USV is under the influence of waves, which results in periodic oscillations of the USV deck in each axis of a coordinated system with an origin at the USV center of mass. For hardware, we assume that the UAV is equipped with a 2MP downward facing camera, an onboard computer for image processing and computing the MPC, and that the USV is equipped with a landing pattern to recognize relative pose.

Fig. 1. UAV landing on USV in real-world experiments.

The main difference between our proposed approach and [17] is that our controller uses the non-linear model of the USV for landing on a rapidly tilting deck and does not employ any communication between the UAV and the USV, as motivated by real-world applicability. To the best of the authors' knowledge, it is the first approach using USV motion prediction in control feedback of a decentralized controller. In summary, our contributions are: (1) we present a novel objective function for finding an optimum landing trajectory that utilizes an MPC algorithm to predict the future of the UAV and USV, without communication; (2) we propose a decentralized vision-based method for observing and predicting the motion of a USV through the use of an online observer that adapts the USV motion model using observations from a downward-facing camera; (3) our proposed approach enables landing on a highly undulating platform with no prior knowledge of the dimensions of the USV; and (4) we propose a prediction algorithm that is designed to prevent a velocity overshoot at the set point for landing with minimal impulse transfer from the surface upon touchdown. The relevant media from this work has been made available as supplementary material on http://mrs.felk.cvut.cz/ral-landing-on-usv.

III. Proposed Non-Linear Estimator-Based MPC

In this section, we present our proposed approach which consists of a UAV prediction model and a simplified USV prediction model. Our proposed controller must satisfy two hard constraints imposed by real-world conditions, which are: 1) The controller must perform its computation under a time constraint of 50 ms (20 Hz); and 2) There is no communication between the UAV and the USV and so, the only method for estimating the state of the USV motion is by visual pose estimation enabled by the AprilTag on the landing platform. Thus, for the sake of clarity, we will call our approach **MPC-NE** (Model Predictive Controler - Non-linear Estimator). Fig. 2 presents the control pipeline used in this work; the contribution is encapsulated in Fig. 2. For the UAV prediction model, a discrete linear time-invariant system is used, while the USV model uses a more complex linearised model to be described subsequently. The 6-degrees of freedom (DOF) USV pose $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{bmatrix}^T$ is estimated in the world frame through the detection of the

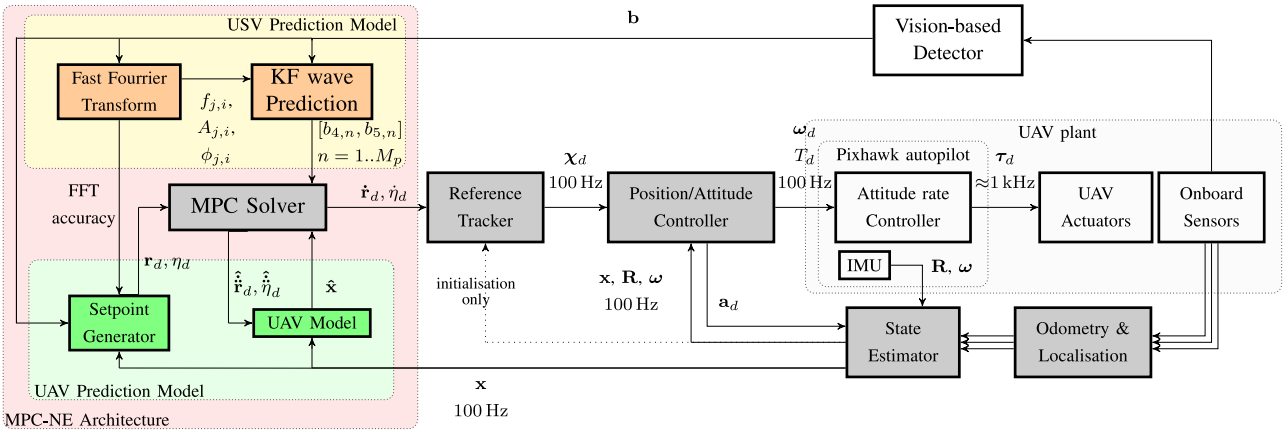


Fig. 2. The *MPC* landing controller (yellow block) is integrated into the MRS system (grey blocks) and supplies the desired reference (velocity $\dot{\mathbf{r}}_d = [\dot{x} \quad \dot{y} \quad \dot{z}]^T$ and heading rate $\dot{\eta}_d$). In the MRS system, the first layer containing a *Reference tracker* processes the desired reference and gives a full-state reference $\boldsymbol{\chi}$ to the attitude controller. The feedback *Position/Attitude controller* produces the desired thrust and angular velocities (T_d, $\boldsymbol{\omega}_d$) for the Pixhawk flight controller (Attitude rate controller). The *State estimator* fuses data from *Odometry & localization* methods to create an estimate of the UAV translation and rotation ($\mathbf{x}$, $\mathbf{R}$). Finally, the *Vision-based Detector* obtains the visual data from the camera and sends the pose information $\mathbf{b}$ of the USV to the MPC.

fiducial tag in the center of the landing pad from the on-board camera of the UAV. The pose of the UAV is fused and accounted for to estimate the correct world frame pose of the USV. This pose information is fed to a fast Fourier transform (FFT) node (based on [18]) which identifies the frequencies, amplitudes, and phases of the N periodic oscillations that make up the USV motion in pitch and roll axes. These identified modes are used to initialize a linear Kalman observer node that corrects the observed state and predicts future motion. These predictions are sent to the MPC controller, which uses them to estimate the feasibility of landing in the near future, i.e., if a sufficiently low tilt of the USV can be found inside the predefined prediction horizon. In turn, it generates the desired linear velocities for x, y, and z axes, as well as the desired angular velocity in heading η. The MPC also receives the estimated UAV state vector $\mathbf{x} = [\ x \ \ \dot{x} \ \ \ddot{x} \ \ y \ \ \dot{y} \ \ \ddot{y} \ \ z \ \ \dot{z} \ \ \ddot{z} \ \ \eta \ \ \dot{\eta} \ \ \ddot{\eta} \ ]^T$ using onboard state estimation proposed by our team in [19].

A finite-state automaton-based approach is used to direct our mission. Based on this, a setpoint generator node commands the aircraft to increase its altitude until the vision marker can be found. Once it is found, the reference of the MPC is changed by the setpoint generator, such that it can hover at a preset altitude above the identified marker. Subsequently, the UAV waits for enough data to be gathered so that the FFT accuracy threshold requirement can be met. Once it is met, the setpoint generator sets the global reference for landing. Then, the MPC begins to use the future motion predictions of the wave to determine a suitable time for landing.

A. USV Prediction Model

USV models can be classified into two different types: Maneuvering Theory and Seakeeping Theory [20].

Owing to the assumptions made in Section II, we choose to focus on the Seakeeping theory since it concerns near-stationary vessels. In addition, the use of a decentralized approach brings challenges in estimating the true odometry of the USV, as converging to reliable estimates of linear and angular velocities of the USV is infeasible. Therefore, we leverage the pose estimate from the camera efficiently by focusing only on the kinematics of the USV.

Our USV prediction model is composed of three parts: a fast Fourier transform, a Kalman observer, and a wave prediction model. First, the FFT performs a decomposition on the pose data obtained from the vision pipeline. The identified modes of these oscillations are used to initialize a Kalman observer that adapts the amplitude and the phase of the wave using the observed values online. Finally, the amplitudes and phases from the Kalman observer are sent to the wave prediction model to enable future wave predictions.

1) FFT-Based Modelling: We assume that the motion of the USV is composed of N_j periodic waves and a non-periodic term that accounts for random noise in tracking the various components for each jth axis. Thus, let the state vector $\mathbf{b}$ be represented by the linear pose b_j for $j \in \{1, 2, 3\}$ for x, y, z axes, respectively, and the angular pose be represented by $j \in \{4, 5, 6\}$ about these axes in the same order.

Note here that, a sufficiently large ship/boat (intended application) would exhibit sufficiently low amplitude oscillations in Z-axis such that they can be handled by changing the reference at every camera frame (as shown here). Thus, the periodic motion of the USV in an axis can be represented as a function of time such that:

$$b_j(t) = b_{j,off} + \sum_{i=1}^{N_j} A_{j,i} \sin \underbrace{(2\pi f_{j,i} t + \phi_{j,i})}_{\Phi_{j,i}}, \tag{1}$$

with $f_{j,i}$ denoting the frequency, $A_{j,i}$ the amplitude, and $\phi_{j,i}$ the phase. Additionally, $b_{j,off}$ is the non-periodic term accounting for random noise. For the initial condition, $\Phi_{j,i}(t)$ is equal to $\Phi_{j,i}(t_{FFT})$, which is the phase obtained as the output of FFT at the time of identification t_{FFT}. In sea conditions, these frequency components can change frequently due to changing winds. Therefore, the pose is sampled continuously and an FFT is run every ΔT_{FFT} seconds. For each axis, we discard the modes that are below a certain threshold amplitude $A_{j,threshold}$, where

$$A_{j,threshold} = A_{gate} \cdot \max\{A_{j,0}, A_{j,1}, \ldots, A_{j,N_j}\}. \tag{2}$$

For reliable performance, and upon tuning on real-world data, we assume $A_{gate}(= 0.02)$ to be a suitable cutoff. This prevents us from identifying noise components as low-amplitude periodic oscillations without losing more than 2% of the accuracy. These erroneous components cause a loss of performance in the Kalman observer, which is explained in the next section.

2) Kalman Observer: The Kalman observer uses a linear model to refine the estimate of identified amplitude and phase of each mode. The observer is necessary because, while the FFT accurately identifies the frequency components, the amplitude and phase outputs are averages for the entire ΔT_{FFT} sampling interval. Therefore, the observer receives new parameters for all identified modes every ΔT_{FFT} seconds. In order to allow sufficient time for the observer parameters to converge to true values, we do not reinitialize the pre-identified modes with the new parameters. Instead, only the newly identified modes are initialized, while discarding the old modes that no longer exist.

To assemble the model, we first write the ordinary differential equation (ODE) for *each mode* for a given axis at any time t. We use $\mathbf{v}_{j,i}$ to denote the ith mode of the USV state vector component b_j in the jth axis. Thus, the derivative of the mode $(\forall j, j \in 1 \dots 6)$ is

$$\dot{\mathbf{v}}_{j,i} = \underbrace{\begin{bmatrix} 0 & 1 \\ -(2\pi f_{j,i})^2 & 0 \end{bmatrix}}_{\mathbf{B}(t_{FFT})} \mathbf{v}_{j,i}, \tag{3}$$

and the mode at time t is

$$\mathbf{v}_{j,i} = \begin{bmatrix} A_{j,i} \sin\left(\Phi_{j,i}(t)\right) \\ 2\pi A_{j,i} f_{j,i} \cos\left(\Phi_{j,i}(t)\right) \end{bmatrix}. \tag{4}$$

Next, we derive the observer model by adding the ODEs of each mode. Thus,

$$\dot{\mathbf{v}}_j(t) = \underbrace{\begin{bmatrix} \mathbf{B}_{j,1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{j,2} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}_{j,N} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\overline{\mathbf{B}}_j} \underbrace{\begin{bmatrix} \mathbf{v}_{j,1} \\ \mathbf{v}_{j,2} \\ \vdots \\ \mathbf{v}_{j,N_j} \\ \mathbf{v}_{j,off} \end{bmatrix}}_{\mathbf{v}_j(t)}. \tag{5}$$

Hence, the output for each axis is

$$b_j(t) = \underbrace{\begin{bmatrix} \mathbf{C}_{j,1} & \mathbf{C}_{j,2} & \cdots & \mathbf{C}_{j,N} & \mathbf{C}_{j,off} \end{bmatrix}}_{\overline{\mathbf{C}}_j} \mathbf{v}_j(t). \tag{6}$$

Note, that each component of the output vector of the mode can be found as

$$b_{j,i} = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{C}_{j,i}} \mathbf{v}_{j,i}. \tag{7}$$

Now, for the brevity of explanation and the readability of the equations, we write the following relation *for only one axial DOF* of the USV in discrete-time. In addition, we clarify that it can be applied to all 6 of the DOF. Furthermore, notice that a time instance $t = k\Delta T + t_{FFT}$, wherein ΔT is the discrete sampling time for new pose observations. Thus, we have a straightforward change in notation such that, for example, $\mathbf{v}_j(t) \equiv \mathbf{v}_j^{(k)}$. Thus, by using the integral approximation method, we have that

$$\mathbf{v}_j^{(k+1)} = \underbrace{\exp\left(\overline{\mathbf{B}}_j \Delta T\right)}_{\mathbf{\Psi}_j} \mathbf{v}_j^{(k)}, \text{ and } \quad b_j^{(k)} = \overline{\mathbf{C}}_j \, \mathbf{v}_j^{(k)}. \tag{8}$$

Then, we continuously estimate the amplitude $A_{j,i}$ and phase $\phi_{j,i}$ of each mode every ΔT using the Kalman Filter. First, $\mathbf{Q}$ is initialised using a diagonal matrix $\mathbf{Q_I} = \lambda \mathbf{I}$, such that

$$\mathbf{Q} = \frac{1}{2}\left(\mathbf{\Psi} \mathbf{Q_I} \mathbf{\Psi}^T + \mathbf{Q_I}\right) \Delta T, \tag{9}$$

where λ is the gain parameter for the process noise observed in the model.

Meanwhile, the observation noise matrix $\mathbf{R}$ is set to the mean amplitude of the observed noise in the system. Thereafter, we apply the filter equations as follows:

$$\begin{aligned} \hat{\mathbf{v}}_j^{(k)} &= \mathbf{\Psi}_j \mathbf{v}_j^{(k-1)}, \\ \hat{\mathbf{P}}^{(k)} &= \mathbf{\Psi}_j \mathbf{P}^{(k-1)} \mathbf{\Psi}_j^T + \mathbf{Q}, \\ \hat{b}_j^{(k)} &= \overline{\mathbf{C}}_j \hat{\mathbf{v}}_j^{(k)}, \\ \mathbf{L}^{(k)} &= \hat{\mathbf{P}}^{(k)} \overline{\mathbf{C}}_j^T \left(\overline{\mathbf{C}}_j \hat{\mathbf{P}}^{(k)} \overline{\mathbf{C}}_j^T + \mathbf{R}\right)^{-1}, \\ \mathbf{v}_j^{(k)} &= \hat{\mathbf{v}}_j^{(k)} + \mathbf{L}^{(k)} \left(b_{j,m} - \hat{b}_j^{(k)}\right), \\ \mathbf{P}^{(k)} &= \left(\mathbf{I} - \mathbf{L}^{(k)} \overline{\mathbf{C}}_j\right) \hat{\mathbf{P}}^{(k)}, \end{aligned} \tag{10}$$

where $\hat{}$ shows the predicted value of the vector/matrix, $\mathbf{I}$ is an identity matrix, $b_{j,m}$ is the measured value of b_j, $\mathbf{L} \in \mathbb{R}^{2(N+1)}$ is the Kalman gain matrix of the system, $\mathbf{P}$ and $\mathbf{Q} \in \mathbb{R}^{2(N+1)\times 2(N+1)}$ are the process co-variance and system noise matrices, respectively, and $\mathbf{R} \in \mathbb{R}$ is observation noise.

At every identification, the relevant elements corresponding to both of the modes that are no longer present, as well as the newly identified modes of the $\mathbf{\Psi}$ matrix, are re-initialized. The corresponding co-variance terms for these modes are reset to maintain a consistent prediction without large deviations.

3) Wave Prediction: Let us now define t_{obs} as the time instant where the last observation was performed, since the prediction algorithm is not run when there are no new observations. Thus, by running the Kalman observer at t_{obs} we find the new amplitude $A_{j,i}(t_{obs})$ and phase $\Phi_{j,i}(t_{obs})$. At the same instant in time t_{obs}, we can extract the corresponding $\mathbf{v}_{j,i}$ and use (4) to acquire:

$$\Phi_{j,i}(t_{obs}) = \arctan\left(\frac{2\pi f_{j,i} [\mathbf{v}_{j,i}]^{1,1}}{[\mathbf{v}_{j,i}]^{2,1}}\right),$$

and

$$A_{j,i}(t_{obs}) = \frac{[\mathbf{v}_{j,i}]^{1,1}}{\sin\left(\Phi_{j,i}(t_{obs})\right)}, \tag{11}$$

where $[\mathbf{v}_{j,i}]^{m,n}$ represents the element corresponding to the mth row and nth column of the vector.

This enables us to predict the wave behavior at a future time $t > t_{obs}$ as

$$\begin{aligned} b_j(t) = \sum_{i=1}^{N_j} A_{j,i}(t_{obs}) \sin\left[2\pi f_{j,i}(t - t_{obs}) + \Phi_{j,i}(t_{obs})\right] \\ + [\mathbf{v}_{j,off}]^{1,1}. \end{aligned} \tag{12}$$

B. UAV Prediction Model

The UAV prediction model used in the proposed MPC is based on the Euler approximation of a set of single particle kinematics equations. Here, we employ the following discrete linear time-invariant system:

$$\mathbf{x}^{(k+1)} = \mathbf{D}\mathbf{x}^{(k)} + \mathbf{E}\mathbf{u}^{(k)}, \text{ with } \mathbf{u}^{(k)} = \begin{bmatrix} \dddot{x} & \dddot{y} & \dddot{z} & \dddot{\eta} \end{bmatrix}^T. \tag{13}$$

In the model represented by (13), the state matrix $\mathbf{D}$ and the input matrix $\mathbf{E}$ can be found through the Kronecker product ($\otimes$), such that:

$$\underset{12\times12}{\mathbf{D}} = \underset{4\times4}{\mathbf{I}} \otimes \underset{3\times3}{\mathbf{D}'}, \text{ with } \mathbf{D}' = \begin{bmatrix} 1 & \Delta t_{pred} & \frac{\Delta t_{pred}^2}{2} \\ 0 & 1 & \Delta t_{pred} \\ 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

$$\underset{12\times4}{\mathbf{E}} = \underset{4\times4}{\mathbf{I}} \otimes \underset{3\times1}{\mathbf{E}'}, \text{ with } \mathbf{E}' = \begin{bmatrix} \frac{\Delta t_{pred}^3}{6} \\ \frac{\Delta t_{pred}^2}{2} \\ \Delta t_{pred} \end{bmatrix}, \tag{15}$$

where $\mathbf{I}$ is an identity matrix, with a prediction made every $\Delta t_{pred} = 0.01$ seconds.

Hence, the state vector represents the states of the system and their derivatives up to acceleration in each axis, and the control input is the jerk experienced in those axes.

C. MPC Objective Function

Once we have defined a prediction model of the UAV and the USV, we can formulate an objective function to enable both way-point navigation and landing. For the sake of simplification, we will omit the superscript $(.)^{(k)}$, which represents a discrete instant in time. Therefore, we can define the objective function J as:

$$\min_{\mathbf{u}_1,\ldots,\mathbf{u}_{M_c}} J(\mathbf{x},\mathbf{u}) = \underbrace{\sum_{m=1}^{M_p} \tilde{\mathbf{x}}_m^T \mathbf{S}\tilde{\mathbf{x}}_m + \mathbf{h}_m^T \mathbf{T}\mathbf{h}_m}_{J_1} + \underbrace{\sum_{m=1}^{M_p} \alpha_L \times g(\tilde{z}_m, b_{4,m}, b_{5,m})}_{J_2},$$

subject to:

$$\begin{aligned} \tilde{\mathbf{x}}_m &= \mathbf{x}_m - \overset{*}{\mathbf{x}}_m, \\ \tilde{z}_m &= z_m - \overset{*}{z}_m, \\ \mathbf{h}_m &= \mathbf{u}_m - \mathbf{u}_{m-1}, \\ \mathbf{x}_{m+1} &= \mathbf{D}\mathbf{x}_m + \mathbf{E}\mathbf{u}_m \ \forall\, m \le M_c, \\ \mathbf{x}_{m+1} &= \mathbf{D}\mathbf{x}_m + \mathbf{E}\mathbf{u}_{M_c} \ \forall\, m > M_c, \\ \mathbf{u}_{\min} &\le \mathbf{u}_m \le \mathbf{u}_{\max}, \\ \mathbf{x}_0 &= \mathbf{x}_{initial}, \\ \mathbf{u}_0 &= \mathbf{u}_{initial}, \\ &\forall\, \{m : m \in \mathbb{N}, 1 \le m \le M_p\}, \end{aligned} \tag{16}$$

where $\overset{*}{\mathbf{x}}_m$ is the desired state, $\tilde{\mathbf{x}}_m$ is the error vector, $\tilde{z}_m$ is the error in z_m position, $\mathbf{h}_m$ is the rate of control input change to ensure smooth input to the UAV, $M_p(=100)$ is the prediction horizon, and $M_c(=40)$ is the control horizon. $\mathbf{S}$ and $\mathbf{T}$ are the corresponding penalty matrices with configurable weights for performance tuning, while $\alpha_L(=1200)$ is a weight chosen for the tuning of the objective function $g(.)$. Additionally, $b_{4,m}, b_{5,m}$ are the pitch and roll angles in discrete time of the USV about its x and y axes, according to (12).

We emphasize that $\overset{*}{\mathbf{x}}_m$ (including $\overset{*}{z}$) can either be a series of points (trajectory) or a single point (step input). This would enable the UAV to keep up with a drifting USV if the XY-position state of the usv is estimated independently. However, a slowly drifting USV is within the dynamic limits of the UAV so as to be compensated by the single-point reference that can be updated after every observation (depending on the camera frame rate). We demonstrate and test this in this linked video. While we do not constrain the output of the MPC, we apply soft constraints to the velocity and acceleration states of the model, such that $v \ge v_{max}$ and $a \ge a_{max}$ incur a high penalty in the objective function.

Herein, we have introduced a novel objective function J_2 (described in the next section) which can account for the predicted motion of the USV, producing a smooth control input to change the altitude of the UAV without any abrupt maneuvers. Using this function, we are able to incorporate the finite state automaton approach using a sigmoid activation function without explicitly describing the possible landing condition. The UAV is able to follow the descend trajectory generated by the MPC by autonomously adjusting its hover distance above the USV. Additionally, it enables us to tune the parameters to control the variance of the resulting landing angles about the mean value of zero-tilt. It is important to mention that the term J_1 in our cost function, is a classical quadratic objective function largely used in robotics and well documented for its feasibility and stability. On the other hand, the J_2 term is different from usual works in the literature because it tackles the terminal cost of the optimization step as a potential barrier.

We employ a non-linear optimization library (NLOPT [21] [22]) which provides the near-optimum solution for the objective function. In order to exercise velocity-based control, the first input from the series of optimum control inputs calculated by the solver is then used to calculate the next state using (13). The velocities for this predicted next state are then passed to the system as the velocity references for the UAV to track (as seen in Fig. 2).

Since the term J_1 primarily contributes to the position control and J_2 contributes to the landing approach, the term J_2 remains disabled until the conditions for the landing approach are satisfied.

D. Landing Approach

We define the function for landing cost as a combination of sigmoids, such that:

$$g(\tilde{z}_m, b_{4,m}, b_{5,m}) = f(\tilde{z}_m) \cdot ((b_{4,m})^2 + (b_{5,m})^2), \tag{17}$$

where $f(\tilde{z}_m)$ is such that

$$f(\tilde{z}_m) = \begin{cases} \left(1.0 + \exp\left(-\dfrac{\tilde{z}_m - h_d}{-0.15}\right)\right)^{-1}, & \text{if } \tilde{z}_m \ge 0.16 \\ \left(1.0 + \exp\left(\dfrac{\tilde{z}_m - 0.1}{-0.01}\right)\right)^{-1}, & \text{otherwise,} \end{cases} \tag{18}$$

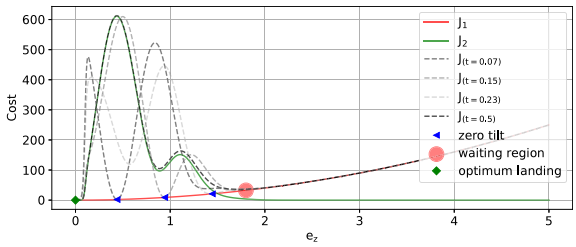


Fig. 3. An example illustration of the effective cost function values obtained during the landing approach.

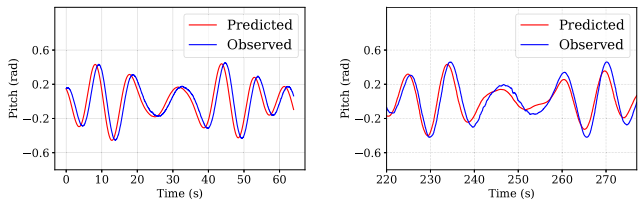


Fig. 4. Comparison between the predictions made by the system using the onboard inertial measurement unit (IMU) data (left) and using the vision data (right).

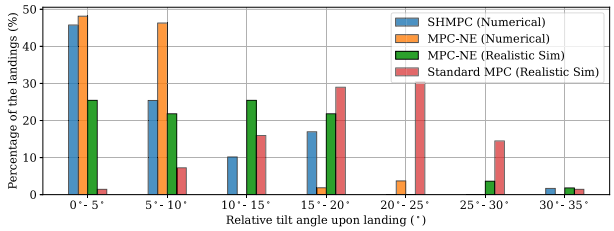


Fig. 5. Histogram comparison between the proposed approach and the standard approach during the touchdown of the UAV on the USV deck.

where h_d controls the waiting region (see Fig. 3) during a landing attempt. Empirically, $h_d = 1.1$ was chosen for our experiments. For the scope of this letter, we assume that the USV has relatively negligible motion in its x and y axes, which is a fair assumption for the problem of landing. The propulsion of the USV may easily compensate for the drift generated by the water currents in order to facilitate landing. It is also safe to assume that $\tilde{z} \geq 0$, as the UAV cannot approach from beneath the USV. In order to activate J_2 to start the landing phase, two conditions must be met. First, **FFT accuracy is higher than a given threshold to detect slow oscillations.** Second, **The position errors in x and y are below a predefined threshold (i.e., $\tilde{x}, \tilde{y} \approx 0$) and horizontal velocities v_x, v_y are also minimal.**

To demonstrate the interaction of J_2 with J_1 during the landing approach, we present a *highly-simplified* plot of the objective function (see Fig. 3) using one mode each for pitch and roll axes. When J_2 is activated, we acquire a combined plot governed by both the (17) and the residual error $\tilde{z}$ in J_1. In Fig. 3, the value of the objective function encounters a peak that continuously evolves as a function of time. This peak acts as a potential barrier. The higher cost associated with the peak holds the aircraft in the waiting region (as marked in the plot). Meanwhile the USV model generates predictions for the future of the USV motion during every iteration of the MPC. The USV sometimes gets close enough to a zero tilt wherein a feasible solution appears, as shown by the *zero-tilt* points in the plot. The UAV is then able to *insert* itself into the time-varying trajectory of these special feasible points by reducing its altitude and approaching in such a way that the cost continues to decrease along the locus of these points. Therefore, the UAV is able to follow the zero-tilt points and finish at the optimum landing point, where touchdown is confirmed by the system based on thrust and other information from onboard sensors.

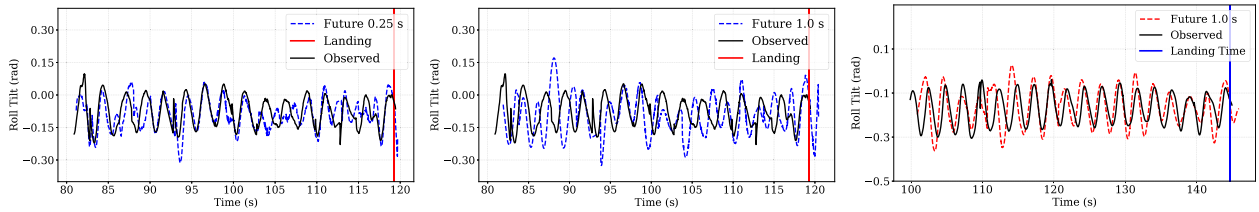
IV. Simulation Experiments

We demonstrate our simulation results in two scenarios: with a numerical simulation, and with a realistic ROS-based Gazebo simulator [19]. The **SHMPC** presented in [15] is shown to work numerically. Thus, we use a similar numerical implementation of our work (**MPC-NE**) to allow us to perform a fair comparison with the state-of-the-art. In this comparison, the non-linear optimization problem is solved by [22] for a landing maneuver of 3 meters and assumes true knowledge of the future motion of the USV. The second comparison is performed using real-time flight with our proposed **MPC-NE** inside the Gazebo simulator. For this comparison, we use a **standard MPC** [19] designed for waypoint navigation. For this standard approach, the UAV attempts to locate the target, and lands after a programmed, uniformly randomly distributed delay between 0 and 100 seconds. We select this duration owing to the periodicity of the tilt angle of the USV. We use a T650 quadrotor frame weighing 3.6 kg carrying a Garmin LiDAR for laser-ranging of altitude and an Intel Realsense D435 camera for live in-simulation video. The video output of the Realsense camera is sent to our system to enable processing on the vision node. (Section III). The 3D model of the USV is similar to our real-world experiments and is affixed with an AprilTag [23] marker for pose estimation. We note that in *both* the comparisons, we push the boundary of performance and test our work in rough sea states, and drive our USV model using a wave generator with 4-5 components of oscillation in both pitch and roll axes, and tilt angles up to 30° (0.5 radians). The frequency components are set such that brief windows for feasible landing exist. Wind disturbances are not considered in this environment since it is tackled by body disturbances estimated by the low-level control feedback pipeline (see Fig. 2).

A. Prediction Results

First, we demonstrate the ability of our system to predict the wave motion up to 1 s into the future based on the observed frequency components and our model of the system. The performance of the system is tested in two scenarios: a 100 Hz odometry output from the simulated USV IMU (used as ground truth), and a 30 Hz stream from the AprilTag.

As we see in Fig. 4, the high-frequency IMU-based predictions are able to match the observed wave reliably without introducing noise. The observer is able to adapt the observed frequency, amplitude, and phase of the modes of the oscillations and converge reliably. As opposed to that, we see slight deviations in the vision-based predictions compared to the IMU results. This deviation in performance can be explained by two factors. First, the linearisation of the model in the time-domain causes inaccuracies that grow as the sampling time increases. Due to this, the three-fold sampling rate of the IMU leads to



(a) Vision—0.25 s into the future. (b) Vision—1.0 s into the future. (c) IMU—1.0 s into the future.

Fig. 6. Comparison between the predictions made using vision (a-b) and using the onboard IMU of the USV (c).

faster and more accurate convergence. Second, the output rate of the AprilTag identification node fluctuates around 30 Hz, depending on the computational load of the onboard computer of the UAV (or simulation computer, in this case). This leads to the misidentification of modes, as the FFT algorithm requires a fixed sample rate for observations. However, this sufficiently proves the ability of the proposed system to reliably predict wave behavior, which will be used for the USV landing further down the pipeline.

B. Landing Results

To continue, we present the ability of our system to land on a platform while tilt angles are sufficiently close to zero. Thus, we present Fig. 5 that shows the results of the numerical comparison between our MPC-NE and the state-of-the-art SHMPC. Note here that the MPC-NE lands $\approx 94\%$ within 10° of tilt, while the SHMPC lands $\approx 71\%$ of the same tilt interval. In this same comparison, the solution time per iteration of our MPC-NE was 9 times lower at 102 ms compared to 917 ms for SH-MPC.

Furthermore, the same figure presents the results of the realistic simulations using Gazebo. It is important to highlight the difference between a numerical simulation result and a realistic simulation result. This is explained by the existing constraints of processing time that demand the algorithms to be processed in real-time. Note here that the MPC-NE is able to conduct 72% of its landings within $15^\circ(0.26$ rad) of tilt compared to 23% of landings using the standard MPC approach. In addition, the proposed approach reduces the 80th percentile result by $9^\circ(0.16$ rad) in comparison to the standard approach. For this comparison, we classify a landing conducted at a tilt angle of less than $20^\circ(0.35$ rad) as successful. Therefore, even in challenging tilts of up to 0.5 radians, the proposed approach had only three failures, while the standard approach fails approximately 50% of the landings. Finally, we also highlight that, even in an unrealistic and challenging scenario, our system is able to conduct 70% of the landings within 50 seconds of reaching its FFT accuracy threshold.

V. Real-World Experiments

To test the contributions and proposed algorithms in the real world, we performed landings on an oscillating target at an open water reservoir.

For the purpose of this experiment, we employed a 4.5 kg T650 quadrotor equipped with vertical pontoons [24] for safety over water (see Fig. 1). In addition, the sensor stack included a Garmin LiDAR for laser-ranging of altitude, a Basler camera for the live video feed, and an Intel NUC for onboard real-time processing of the algorithms, data, and video. The target is a special custom-made USV [25] equipped with a 2 m $\times$ 2 m landing zone, affixed with an AprilTag [23] for 6-DOF pose-estimation. The experimental conditions subjected the UAV to a wind of 7 m/s and a USV oscillating with an amplitude of 0.3 radians.

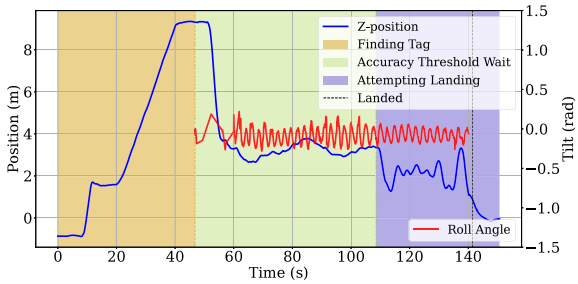


Fig. 7. Plot of a selected real-world open-water experiment (video).

A. Prediction Results

Here we demonstrate our prediction pipeline in two scenarios: a 30 Hz stream from AprilTag, and a 100 Hz stream from the IMU. The prediction results for the real-world experiments are presented in Fig. 6 and discussed below.

For predictions based on vision-based pose estimation, as seen in Fig. 6(a), the near-term future correlates well with the observed motion. However, Fig. 6(b) indicates that the predictions for the long-term future can suffer in accuracy. This correlates well with the simulation results as shown in Fig. 4 and can be attributed to the higher sampling time-step and its higher variability. Occasionally, it also exhibits convergence and consecutive divergence as more data is fed into the pipeline. For ground truth, we use Fig. 6(c) to demonstrate the effectiveness of the pipeline in robustly predicting the future of the USV. However, since MPC exhibits a higher reliance on the predictions that are temporally proximal, the predictions for 0.25 and 0.50 seconds into the future offer robust support for preventing a landing during an infeasible window. The chosen angle for landing is also sufficiently low in order to demonstrate the prediction capabilities and the selection of a feasible landing window.

B. Landing Results

We demonstrate the real-world landing process through Fig. 7. In these experiments, the UAV was able to land within 50 seconds of acquiring the required FFT accuracy. This coincides with our findings in simulation experiments. Additionally, the tilt angles upon touchdown were less than $5^\circ(0.09$ rad).

VI. Conclusion

In this letter, we proposed an MPC that enables a UAV to land autonomously on a tilting USV. The MPC employs a novel objective function and an online decomposition of the motion of the vessel in order to attempt and complete the landing during a near-zero tilt of the landing platform. We successfully demonstrated that we are able to model and predict the behaviour of the UAV and USV without active communication between them. Further, we establish a novel approach for landing on the USV using these predictions, which autonomously adjusts the relative altitude for the UAV to ensure that the landing occurs as close to the zero-tilt state of the landing deck as possible, increasing safeness of the landing phase and reducing impact forces on the landing UAV. In comparison to state-of-the-art approaches, we achieved significant improvement in the case of landing in demanding conditions with high waves and high winds without knowing the dimensions of the USV.

References

[1] L. C. Lebreton et al., "River plastic emissions to the world's oceans," *Nature Commun.*, vol. 8, no. 1, pp. 1–10, 2017.

[2] T. Nascimento and M. Saska, "Position and attitude control of multi-rotor aerial vehicles: A survey," *Annu. Rev. Control*, vol. 48, pp. 129–146, 2019.

[3] J. M. Riola, J. J. Diaz, and J. M. Giron-Sierra, "The prediction of calm opportunities for landing on a ship: Aspects of the problem," in *Proc. IEEE OCEANS*, 2011, pp. 1–8.

[4] S. Küchler, T. Mahl, J. Neupert, K. Schneider, and O. Sawodny, "Active control for an offshore crane using prediction of the vessels motion," *IEEE/ASME Trans. Mechatron.*, vol. 16, no. 2, pp. 297–309, Apr. 2011.

[5] J. Neupert, T. Mahl, B. Haessig, O. Sawodny, and K. Schneider, "A heave compensation approach for offshore cranes," in *Proc. IEEE Amer. Control Conf.*, pp. 538–543, 2008.

[6] L. Marconi, A. Isidori, and A. Serrani, "Autonomous vertical landing on an oscillating platform: An internal-model based approach," *Automatica*, vol. 38, pp. 21–32, 2002.

[7] S. Lee et al., "Sliding mode guidance and control for UAV carrier landing," *Trans. Aerosp. Electron. Syst.*, vol. 55, pp. 951–966, Apr. 2019.

[8] Y. Meng, W. Wang, H. Han, and J. Ban, "A visual/inertial integrated landing guidance method for UAV landing on the ship," *Aerosp. Sci. Technol.*, vol. 85, pp. 474–480, 2019.

[9] T. D. Ngo and C. Sultan, "Nonlinear helicopter and ship models for predictive control of ship landing operations," in *Proc. AIAA Guid., Navigation, Control Conf.*, 2014, pp. 1–19.

[10] R. Polvara, S. Sharma, J. Wan, A. Manning, and R. Sutton, "Vision-based autonomous landing of a quadrotor on the perturbed deck of an unmanned surface vehicle," *Drones*, vol. 2, 2018, Art. no. 15.

[11] S. Abujoub, J. McPhee, C. Westin, and R. A. Irani, "Unmanned aerial vehicle landing on maritime vessels using signal prediction of the ship motion," in *Proc. OCEANS MTS/IEEE Charleston*, 2018, pp. 1–9.

[12] Z.-C. Xu, B.-B. Hu, B. Liu, X. Wang, and H.-T. Zhang, "Vision-based autonomous landing of unmanned aerial vehicle on a motional unmanned surface vessel," in *Proc. IEEE 39th Chin. Control Conf.*, 2020, pp. 6845–6850.

[13] B. Lee, V. Saj, and M. Benedict, "Machine learning vision and nonlinear control approach for autonomous ship landing of vertical flight aircraft," in *Proc. 77th Annu. Vertical Flight Soc. Forum Technol. Display*, 2021.

[14] H.-T. Zhang et al., "Visual navigation and landing control of an unmanned aerial vehicle on a moving autonomous surface vehicle via adaptive learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5345–5355, Dec. 2021.

[15] W. B. Greer and C. Sultan, "Shrinking horizon model predictive control method for helicopter–Ship touchdown," *J. Guid., Control, Dyn.*, vol. 43, no. 5, pp. 884–900, 2020.

[16] T. D. Ngo and C. Sultan, "Variable horizon model predictive control for helicopter landing on moving decks," *J. Guid., Control, Dyn.*, vol. 45, no. 4, pp. 774–780, 2022.

[17] L. Persson and B. Wahlberg, "Model predictive control for autonomous ship landing in a search and rescue scenario," in *Proc. AIAA Scitech Forum*, 2019, Art. no. 1169.

[18] M. Frigo and S.G. Johnson, "The design and implementation of FFTW3," *Proc. IEEE*, vol. 93, no. 2, pp. 216–231, Feb. 2005.

[19] T. Baca et al., "The MRS UAV system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles," *J. Intell. Robot. Syst.*, vol. 102, no. 26, pp. 1–28, May 2021.

[20] T. Fossen, "Marine Control Systems: Guidance, navigation and control of ships, rigs and underwater vehicles," *Marine Cybern.*, 2002.

[21] J. M. Gablonsky and C. T. Kelley, "A locally-biased form of the direct algorithm," *J. Glob. Optim.*, vol. 21, pp. 27–37, 2001.

[22] S. G. Johnson, "The nlopt nonlinear-optimization package," [Online]. Available: http://github.com/stevengj/nlopt

[23] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible layouts for fiducial tags," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1898–1903.

[24] D. Hert et al., "Mrs modular UAV hardware platforms for supporting research in real-world outdoor and indoor environments," in *Proc. Int. Conf. Unmanned Aircraft Syst.*, 2022, pp. 1264–1273.

[25] E. Pairet et al., "Nukhada USV: A robot for autonomous surveying and support to underwater operations," in *Proc. IEEE OCEANS Chennai*, 2022, pp. 1–6.