

Visual Servoing MPC framework for Shipboard Autonomous Landing

Luís G. L. de Paula*, Vijay S. Dwivedi[†], Hyo-Sang Shin[‡], and Antonios Tsourdos[§]
Cranfield University, Faculty of Engineering and Applied Sciences, MK43 0AL, Cranfield, UK

Autonomous shipboard landing under harsh sea state conditions is a challenging task, with platform oscillations being one of the critical challenges to expand the flight envelope. Conventional GNC systems usually rely on deck-mounted equipment to supply relative pose data, which limits interoperability within a naval fleet. Image Based Vision Servoing (IBVS) controllers provide a complementary solution that only relies on feature detections and image plane processing. However, current approaches are limited to mild sea state conditions and mainly consider current ship states within the landing criteria. This can increase the risk of unsafe conditions under harsh sea states, such as rollover. This paper introduces a framework that integrates visual servoing with a nonlinear model predictive control (NMPC) to address the attitude oscillations of the landing pad by introducing a custom cost barrier function. Vessel states are forecasted in real-time using a Fast Fourier Transform (FFT) combined with a Discrete Kalman Filter (DKF), enabling short-term prediction without requiring prior knowledge of vessel dynamics. Simulation results and flight test validation demonstrate that the proposed approach achieves a higher success rate under high sea state condition compared to both conventional IBVS based on current or predicted states.

I. Nomenclature

<i>AR</i>	=	Auto regressive model	<i>DKF</i>	=	Discrete Kalman Filter
<i>DoF</i>	=	Degrees of Freedom	<i>FF-IBVS</i>	=	Feed Forward IBVS
<i>FFT</i>	=	Fast Fourier Transform	<i>FoV</i>	=	Field of View
<i>FTDVS</i>	=	Finite-Time Dynamic Visual Servoing	<i>IBVS</i>	=	Image-Based Visual Servo
<i>INDI</i>	=	Incremental Nonlinear Dynamic Inversion	<i>ISG</i>	=	Input Scaling Gain
<i>MoCap</i>	=	Motion Capture	<i>MPC</i>	=	Model Predictive Control
<i>MPC-NE</i>	=	MPC - Nonlinear Estimator	<i>NLP</i>	=	Nonlinear Programming
<i>NMPC</i>	=	Nonlinear MPC	<i>PH</i>	=	Prediction Horizon
<i>QP</i>	=	Quadratic Programming	<i>RAO</i>	=	Response Amplitude Operators
<i>SHOL</i>	=	Ship-Helicopter Operating Limits	<i>SS</i>	=	Sea State
<i>SyID</i>	=	System Identification	<i>VIP</i>	=	Virtual Image Plane
<i>VS-NMPC</i>	=	Visual Servoing NMPC	<i>VTOL</i>	=	Vertical Take-off and Landing

II. Introduction

SMALL-scale Unmanned Aerial Vehicles (UAVs) have increasingly shifted from fixed-wing configurations to multirotors, a transition enabled by advances in on-board processing and computer vision [1]. Autonomous Vertical Take-Off and Landing (VTOL) aircraft are widely used in military and civilian operations [2]. In offshore environments, UAVs undertake critical missions such as industrial inspections [3], search and rescue (SAR) operations [4], and maritime surveillance or reconnaissance tasks [5]. Among the most challenging scenarios for Guidance, Navigation, and Control (GNC) are shipboard landings, whether involving UAVs or human-piloted rotorcraft. Naval platforms impose severe

*PhD Student, Cranfield University, Faculty of Engineering and Applied Sciences, MK43 0AL, Cranfield, UK. luis.leandrodepaula@cranfield.ac.uk

[†]Research Fellow, Faculty of Engineering and Applied Sciences, MK43 0AL, Cranfield, UK. VijayShankar.Dwivedi@cranfield.ac.uk

[‡]Professor, Cranfield University, Faculty of Engineering and Applied Sciences, MK43 0AL, Cranfield, UK and Cho Chun Shik Graduate School of Mobility, KAIST, Daejeon, 34141, South Korea. hyosangshin@kaist.ac.kr

[§]Professor, Faculty of Engineering and Applied Sciences, MK43 0AL, Cranfield, UK. a.tsourdos@cranfield.ac.uk

constraints such as deck motion, turbulent air wakes, sea spray, and low visibility. Extending operations to harsh weather environments can increase operational capability, especially in time-critical missions such as SAR. However, depending on mission-specific constraints, heavy-class vessels may not always be able to provide optimal conditions for the UAV landing procedure [6]. Furthermore, achieving this requires the GNC system to account for the fast dynamics of the landing deck. Another major challenge lies in the restrictive nature of shipboard landing envelopes, as exemplified by the civilian SHOL (Ship Helicopter Operating Limit) [7]. These limitations are primarily designed to prevent unsafe scenarios, such as rollover after touchdown or exceeding the structural limits of the landing gear. Such strict constraints pose significant challenges for enabling safe autonomous landings. One approach relies on forecasting the states of the vessel to identify suitable landing windows. However, achieving accurate long-term forecast often requires cooperative communication and access to detailed ship models. Furthermore, when vessel-specific models are considered, the systems must be updated to ensure seamless compatibility across diverse naval platforms, a key operational requirement [5]. In addition, integration of the forecast algorithm with the GNC is also a key challenge. Usually operational GNC solutions for UAV landings at sea are based on relative navigation systems [8]. These approaches require onboard deck hardware that can again limit interoperability between different naval vessels. Computer vision-based GNC systems offer a viable complementary approach, as the pose can be estimated even under challenging conditions [9]. Moreover, Image-Based Visual Servoing (IBVS) enables the use of visual features, such as landing marks, to guide control strategies based on image-plane tracking.

The main contribution of this paper is the development of a UAV speed controller based on a Nonlinear Model Predictive Control (NMPC) framework that integrates Image-Based Visual Servoing (IBVS) kinematics with a short-term forecast algorithm for the target states. The proposed forecasting approach does not rely on prior knowledge of the vessel model. Short-term prediction is achieved using an enhanced Fast Fourier Transform (FFT) combined with a Discrete Kalman Filter (DKF). Improvements include a discrete sliding window and a sigmoid transition function designed to take advantage of the DKF convergence from previous windows. Within the NMPC formulation, the predicted target states are incorporated through a cost barrier function, allowing the controller to continuously generate control inputs that prevent landings under conditions exceeding safe roll limits. The proposed method is evaluated under simulated roll motions representative of harsh sea conditions, that is Sea State 6 (SS 6), and compared against both conventional and improved IBVS controllers.

III. Literature Review

An in-depth survey of flight-tested approaches that take into account moving targets with dynamics similar to those of ships is presented in [8]. When considering heavy class vessels, the most important DoF are roll and heave motions [10]. Although autonomous landing on vessels is a well-established capability, significant challenges remain when extending operations to rough sea conditions. One key operational requirement is to avoid relying on hardware installed on the vessel, which would improve interoperability across different naval platforms [5]. In this context, vision-based controllers provide a complementary solution that requires minimal changes on the vessel [8].

Visual servoing has been investigated as a solution to autonomously track targets, with reinforcement learning achieving better results than traditional partitioned IBVS [11]. However, the results showed that the design of the reward function was already challenging for ground-based targets, and scenarios that involve targets with a tilt and heaving motion during landing were not addressed. Although limited to intermediate sea states, successful results have been reported using controllers based on conventional IBVS with a feedforward term (FF-IBVS) [8, 12, 13]. In [13], autonomous landings were demonstrated in conditions up to Sea State 4. The approach incorporates a virtual image plane strategy to compensate for the UAV attitudes. Quadrilateral interpolation of the visual features is used to reduce tracking oscillations caused by landing pad motion. To prevent feature loss, a vertical control strategy was introduced based on the proximity of features to the border of the image plane. Since the landing criteria are based on the current state of the vessel [12, 13], safety concerns such as rollover could still arise in high sea state due to the faster dynamics of the landing platform. To address this issue, [14] proposed a novel method called Finite-Time Dynamic Visual Servoing (FTDVS). In this approach, the landing pad features are first identified in the image plane. Optical flow data, estimated from the motion of the projected sides of the landing pad, is then used in a finite-time tracking controller. Additionally, a peak detection module based on the oscillations of the projected side lengths estimates the period of the target motion, providing timing cues for safe landing. Although successful landings were demonstrated through flight tests, the investigated scenarios were limited to simplified sinusoidal motion profiles.

Other vision-based approaches using Model Predictive Control (MPC) have been successfully flight tested against adverse sea state [15, 16]. In [16], a novel Quadratic Programming (QP) guidance algorithm with a variable prediction

horizon (PH) scheme was presented to address direct landing procedures. The MPC framework incorporates forecasted target states obtained from an Auto-Regressive (AR) model trained on Motion Capture data. Flight tests were performed under a wide range of high-sea-state conditions, with a scaled-down control bandwidth using an Explicit Model Following Controller. However, a limited number of attempts still resulted in unsafe terminal conditions, particularly in vertical speed and attitude. These failures were attributed to limited robustness against poor forecasts. The AR model showed acceptable forecast accuracy for horizons up to 0.5 s, but exhibited significant deviations at 1.5 s.

Focusing on the conventional vertical landing procedure, the authors in [15] successfully validated a real-time Nonlinear Estimator MPC (MPC-NE) path planning using forecasts with a horizon of up to 0.5 s. The cost function included barrier terms computed from ship states and the relative height between the UAV and the landing pad. The forecasting algorithm combined a Fast Fourier Transform (FFT) with a Discrete Kalman Filter (DKF) update scheme similar to [17]. Simulation results showed a success rate of 94% for a 10° landing pad tilt limit. Outdoor tests demonstrated the feasibility of the approach, but no statistical assessment was provided regarding landing accuracy. This forecast method was further explored in [10] while tracking a high hover position with a visual servo-controller, however, no landing attempts were made.

In [18] authors propose a NMPC scheme that includes both UAV and visual servoing dynamics to handle visibility and control input constraints. The dynamics of the UAV was identified using an online data-driven approach based on Dynamic Mode Decomposition with control (DMDc) [19]. Although limited to simulations and static targets, the controller was able to maintain features within the image plane. Nevertheless, FoV constraints were defined within the virtual image plane, which could still result in feature loss due to UAV attitude. A more recent approach that integrates NMPC with IBVS is presented in [20], where both bounded disturbances and long-term field-of-view (FoV) constraints are taken into account. A two-stage nested landing tag design enables reliable detection from high-altitude positions. By incorporating the UAV attitude controller into the NMPC framework, the authors directly addressed FoV constraints. The feasibility of the proposed method was validated through simulations and real flight experiments, demonstrating its effectiveness in maintaining features within the FoV during the landing procedure. However, the MPC formulation does not consider landing pad oscillations and therefore does not actively mitigate rollover risks associated with dynamic platform motion.

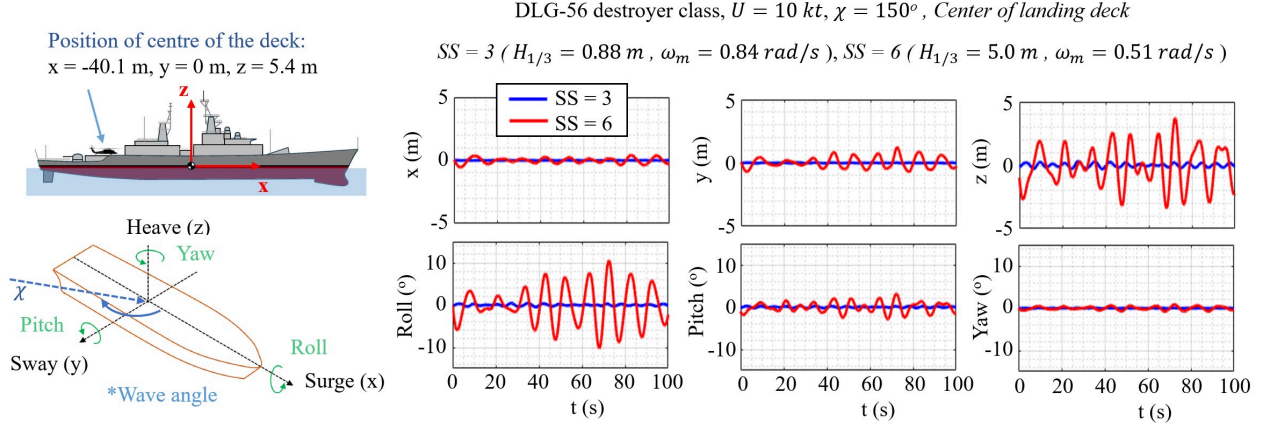
This paper presents a novel approach that integrates visual servoing kinematics into an NMPC framework, using custom cost functions to account for the motion of the landing pad during the autonomous landing. The proposed method is benchmarked against IBVS, with landing decisions based on current or predicted target states. The feasibility of the controller is demonstrated through flight tests using a scaled-down experimental setup that replicates the dynamic behavior of the landing platform.

IV. Methodology

This chapter presents an overview of the methodology for both simulations and flight tests. The vessel dynamics, the baseline IBVS, and the forecast algorithm are described based on the findings presented in previous work by the authors [10]. Then the novel VS-NMPC framework is presented.

A. Vessel Motion

The ship simulation module is based on Response Amplitude Operators (RAO) transfer functions from a destroyer class vessel (USS DLG-26) [21, 22] with a collection of random inputs from a Bretschneider wave spectra [23], similar to [24]. Further details on ship motion module can be found in [10]. Fig. 1 shows simulations comparing mild sea state (SS 3) with harsh conditions (SS 6), where U is the vessel forward speed, χ the wave encounter angle, ω_m the wave modal frequency, and $H_{1/3}$ the significant wave height. As a result, focus should be given to both heave and roll DoF since oscillations are higher than the other ones in harsh high sea state [8]. However, the controllers presented in this paper will initially be tested against landing pad roll motion.



B. IBVS controller

Assuming a pinhole camera, the visual servoing dynamics is given by Eq. 1 following the formulation presented in [10].

$$\dot{\xi}_i = \begin{pmatrix} \dot{\bar{u}}_i \\ \dot{\bar{v}}_i \end{pmatrix} = \underbrace{\begin{pmatrix} -\frac{f_u}{z_i} & 0 & \frac{\bar{u}_i}{z_i} & \frac{\bar{u}_i \bar{v}_i}{f_v} & -f_u - \frac{\bar{u}_i^2}{f_u} & \bar{v}_i \frac{f_u}{f_v} \\ 0 & -\frac{f_v}{z_i} & \frac{\bar{v}_i}{z_i} & f_v + \frac{\bar{v}_i^2}{f_v} & -\frac{\bar{u}_i \bar{v}_i}{f_u} & -\bar{u}_i \frac{f_v}{f_u} \end{pmatrix}}_{J_p} \begin{pmatrix} v_{c,x} \\ v_{c,y} \\ v_{c,z} \\ \omega_{c,x} \\ \omega_{c,y} \\ \omega_{c,z} \end{pmatrix} \quad (1)$$

where J_p is the image jacobian matrix, $(v_{x,c}, v_{y,c}, v_{z,c}, \omega_{x,c}, \omega_{y,c}, \omega_{z,c})^T$ is the camera velocity vector, $f' = (f_u, f_v)^T$ the camera focal length in pixels, $p_i = (x_i, y_i, z_i)^T$ the i -th feature point coordinates in world frame, and $\bar{s}_i = (\bar{u}_i, \bar{v}_i)^T$ the projected coordinates in the image plane. Traditional IBVS then explores the image plane errors by computing the tracking speed as in Eq. 2.

$$v_c = -\lambda_{6 \times 6} \begin{pmatrix} \hat{J}_p(\bar{p}_1, \hat{z}_1, f') \\ \hat{J}_p(\bar{p}_2, \hat{z}_2, f') \\ \vdots \\ \hat{J}_p(\bar{p}_n, \hat{z}_n, f') \end{pmatrix}^\dagger \begin{pmatrix} \bar{s}_{1,d} - \bar{s}_1 \\ \bar{s}_{2,d} - \bar{s}_2 \\ \vdots \\ \bar{s}_{n,d} - \bar{s}_n \end{pmatrix} = -\lambda \mathcal{J}^\dagger(p_i, \hat{z}_i, f') \mathcal{S}(\bar{s}_{i,d}, \bar{s}_i) \quad (2)$$

where n is the total number of detected features, $v_c \in \mathbb{R}^6$ is the desired camera velocity, $\hat{J}_p$ is the estimated jacobian matrix based on the measured feature height $\hat{z}_n$, $\mathcal{J}^\dagger$ is the pseudo-inverse of $\hat{J}_p$, and λ is the controller gain matrix given by $\lambda = \text{diag}(\lambda_{v_x}, \lambda_{v_y}, \lambda_{v_z}, \lambda_{\omega_x}, \lambda_{\omega_y}, \lambda_{\omega_z})$. Since the UAV inner controller concatenates the attitude controller and speed controller, not all v_c are reachable. Therefore, a Virtual Image Plane (VIP) is introduced by compensating the attitudes of the UAV. Following [10], this transformation can be defined as in Eq. 3. This approach also prevents conditions that result in velocity commands opposite to the required one [13].

$$\bar{s}_{i,v} = \begin{pmatrix} \bar{u}_{i,v} \\ \bar{v}_{i,v} \end{pmatrix} = \begin{pmatrix} f_u \frac{x_{i,v}}{z_{i,v}} \\ f_v \frac{y_{i,v}}{z_{i,v}} \end{pmatrix} = \begin{pmatrix} \frac{(\bar{u}_{i,c} \cos \phi - f_u \sin \phi) f_u f_v}{\bar{u}_{i,c} f_v \cos \theta \sin \phi + \bar{v}_{i,c} f_u \sin \theta + f_u f_v \cos \phi \cos \theta} \\ \frac{(-\bar{u}_{i,c} f_v \sin \phi \sin \theta + \bar{v}_{i,c} f_u \cos \theta - f_u f_v \sin \theta \cos \phi) f_v}{\bar{u}_{i,c} f_v \cos \theta \sin \phi + \bar{v}_{i,c} f_u \sin \theta + f_u f_v \cos \phi \cos \theta} \end{pmatrix} \quad (3)$$

where $\bar{s}_{i,v} = (\bar{u}_{i,v}, \bar{v}_{i,v})^T$ are the virtual image plane coordinates of the i -th feature, ϕ and θ are the roll and pitch angles of the UAV, respectively. It is important to note that the VIP transformation does not handle the FoV constraints, which are still given in the original image plane. To overcome that issue, a vertical controller is introduced as described in [10].

Since the forecasted states of the target will be used to find a suitable landing window, it is not necessary to track the attitude of the platform while in the descent phase. by using a quadrilateral interpolation [10, 13], it is possible to mitigate unnecessary tracking oscillations introduced by roll and pitch oscillations of the landing pad. Following the procedure in [10], the larger side is used as the interpolation reference to provide additional height clearance above the landing deck.

As a result of the VIP transformation and quadrilateral interpolation, only $\mathbf{v}_c = (v_{x,c}, v_{y,c}, v_{z,c}, \omega_{z,c})^T \in \mathbb{R}^4$ within the IBVS framework, which requires the gains given by $\lambda_s = \text{diag}(\lambda_{v_x}, \lambda_{v_y}, \lambda_{v_z}, \lambda_{\omega_z})$. Therefore, the tracking speeds can be computed as in Eq. 4 assuming $(l_{x,c}, l_{y,c}, l_{z,c})^T$ as the camera position in the UAV body frame and $l_{z,c} = 0$.

$$\mathbf{v}_b = \begin{bmatrix} v_{x,b} \\ v_{y,b} \\ v_{z,b} \\ \omega_{z,b} \end{bmatrix} = \begin{bmatrix} v_{x,c} \\ v_{y,c} \\ v_{z,c} \\ \omega_{z,c} \end{bmatrix} + \begin{bmatrix} -\omega_{z,c} l_{y,c} \\ \omega_{z,c} l_{x,c} \\ 0 \\ 0 \end{bmatrix}. \quad (4)$$

C. Vessel motion prediction

The forecast algorithm is briefly described here as further details can be found in [10, 15]. In this paper, the landing deck has an Aruco board, thus the relative target states can be estimated using the OpenCV library [25]. If the attitudes and position of the UAV are known, then the targets states relative to the world frame can be computed. For each past ΔT_w time window, considering a sliding window of $\delta t_{sliding}$, the vessel motion can be estimated based on an FFT given by Eq. 5. In order to filter noise, a dynamic cutoff amplitude is used as described in [10].

$$\eta(t) = \eta_0 + \sum_{i=1}^n \eta_i \cos(2\pi f_i t + \gamma_i), t \in [t_k, t_k + \Delta T_w] \quad (5)$$

where η is one of the DoF of the landing pad motion ($X_s, Y_s, Z_s, \psi_s, \theta_s, \phi_s$). Therefore, the overall dynamics can be represented by the linear state space given by Eqs. 6 and 7.

$$\dot{\xi}_\eta(t) = \begin{pmatrix} 0_{2,2} & 0_{2,2} & \dots & 0_{2,2} & 0_{2,2} \\ \mathbf{A}_1 & 0_{2,2} & \dots & 0_{2,2} & 0_{2,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_{2,2} & 0_{2,2} & \dots & \mathbf{A}_n & 0_{2,2} \end{pmatrix} \xi_\eta(t) = \mathcal{A} \xi_\eta(t), \quad (6)$$

$$\eta(t) = \begin{pmatrix} \mathbf{C}_0 & \mathbf{C}_1 & \dots & \mathbf{C}_n \end{pmatrix} \xi_\eta(t) = \mathbf{C} \xi_\eta(t)$$

$$\mathbf{A}_i = \begin{pmatrix} 0 & 1 \\ -(2\pi f_i)^2 & 0 \end{pmatrix}, \quad \mathbf{C}_i = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \xi_\eta(t) = \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{pmatrix}, \quad \mathbf{v}_i = \begin{pmatrix} \eta_i \cos(2\pi f_i t + \gamma_i) \\ -2\pi f_i \eta_i \sin(2\pi f_i t + \gamma_i) \end{pmatrix} \quad (7)$$

noting that $f_0 = 0$ and $\phi_0 = 0$ for the FFT offset, $\xi_\eta \in \mathbb{R}^{(2n+1) \times 1}$, $\mathbf{A}_i \in \mathbb{R}^{2 \times 2}$, $\mathcal{A} \in \mathbb{R}^{2(n+1) \times 2(n+1)}$, $\mathbf{C}_i \in \mathbb{R}^{2 \times 1}$, $\mathbf{C} \in \mathbb{R}^{2(n+1) \times 1}$, and $\mathbf{v}_i \in \mathbb{R}^{2 \times 1}$.

Assuming that the modes do not change between the last window and the current incomplete one, it is possible to update the amplitude and phase of each mode by introducing a DKF as the vision pipeline sampling rate is higher than ΔT_w . Modes that are still being detected will use the DKF covariances from the previous window. As presented in [15], the DKF equations are given by 9.

$$\begin{aligned} \dot{\xi}_\eta^{(k+1)} &= e^{\mathcal{A} \delta t} \xi_\eta^{(k)} = \mathcal{A}_d \xi_\eta^{(k)} \\ \eta^{(k)} &= \mathbf{C} \xi_\eta^{(k)} \end{aligned} \quad (8)$$

$$\begin{aligned}
\hat{\xi}^{(k)} &= \mathcal{A}_d \xi^{(k-1)} & \mathbf{L}^{(k)} &= \hat{\mathbf{P}}^{(k)} \mathbf{C}^T \left(\mathbf{C} \hat{\mathbf{P}}^{(k)} \mathbf{C}^T + \mathbf{R} \right)^{-1} \\
\hat{\xi}^{(k)} &= \hat{\xi}^{(k)} + \mathbf{L}^{(k)} (\eta_m - \hat{\eta}^{(k)}) & \mathbf{P}^{(k)} &= \left(\mathbf{I} - \mathbf{L}^{(k)} \mathbf{C} \right) \hat{\mathbf{P}}^{(k)} \\
\hat{\mathbf{P}}^{(k)} &= \mathcal{A}_d \mathbf{P}^{(k-1)} \mathcal{A}_d^T + \mathbf{Q} & \hat{\eta}^{(k)} &= \mathbf{C} \hat{\xi}^{(k)}
\end{aligned} \tag{9}$$

where η_m is the measurement provided by the vision pipeline, $\mathbf{L} \in \mathbb{R}^{2(n+1) \times 1}$ is the Kalman gain, $\mathbf{P} \in \mathbb{R}^{2(n+1) \times 2(n+1)}$ is the state covariance matrix, $\mathbf{Q} \in \mathbb{R}^{2(n+1) \times 2(n+1)}$ is the process noise covariance matrix, and $\mathbf{R} \in \mathbb{R}$ is the measurement covariance. The matrix $\mathbb{R}$ is computed from the mean amplitude of the measurement noise, while the covariance initialization is $\mathbf{Q} = \frac{\lambda}{2} (\mathcal{A} \mathcal{A}^T + \mathbf{I}) \delta t$ with the gain parameter λ . As a result, Eqs. 10, 11 and 12 provide the forecast of target states. Furthermore, a data quality verification, as described in [10], is implemented to address intermittent data acquisition, such as brief occlusions.

$$\Phi_i(t_m) = \gamma_i(t_m) + 2\pi f_i t_m = \arctan \left(\frac{2\pi f_i v_{i,(1,1)}}{v_{i,(1,1)}} \right) \tag{10}$$

$$\eta_i(t_m) = \frac{v_{i,(1,1)}}{\sin(\Phi_i(t_m))} \tag{11}$$

$$\begin{aligned}
\eta(t) &= \eta_0 + \sum_{i=1}^n \eta_i(t_m) \cos(2\pi f_i(t - t_m) + \Phi_i(t_m)), t > t_m \\
\dot{\eta}(t) &= - \sum_{i=1}^n 2\pi f_i \eta_i(t_m) \sin(2\pi f_i(t - t_m) + \Phi_i(t_m)), t > t_m
\end{aligned} \tag{12}$$

where t_m is the current time. A sigmoid function defined by Eq. 13 is introduced to smoothen the discontinuities between the discrete windows. By doing so, it is possible to take advantage of the DKF convergence from the last window. As a result, large deviations are filtered [10].

$$\mathcal{K}(\delta t_e) = \frac{1}{(1 + e^{-c_1(\delta t_e - c_2)})}, \quad c_1 = \frac{10}{\Delta t_t}, \quad c_2 = \frac{\Delta t_t}{2} \tag{13}$$

D. Visual Servoing Nonlinear Model Predictive Control

First, it is assumed that the UAV built-in controller is able to track the input speed $\boldsymbol{\mu} = (v_{x,n}, v_{y,n}, v_{z,n}, \omega_{z,b})^T \in \mathbb{R}^4$ given by Eq. 14. This can be achieved by using GPS data or even other relative navigation systems such as optical flow.

$$\begin{bmatrix} v_{x,n} \\ v_{y,n} \\ v_{z,n} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \phi \sin \theta & \cos \phi \sin \theta \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} v_{x,b} \\ v_{y,b} \\ v_{z,b} \end{bmatrix}. \tag{14}$$

The proposed VS-NMPC is based on the visual servoing kinematics of the image features presented in Eq. 1. Considering i -th features, this equation can be converted to the discrete time state space as presented in Eq. 15.

$$\xi_{i,n+1} = \mathbf{f}_{d,\Delta t}(\xi_{i,n}, z_n, \boldsymbol{\mu}_n) \tag{15}$$

where n represents the time steps, $f_{d,\Delta t}$ is the discrete servoing dynamics at a sampling rate Δt , and $\mathbf{z}_n$ is the depth between the UAV and the landing platform. In this paper a 4th order Runge-Kutta (RK4) is used for numerical integration. Note that for each feature, the depth is assumed to be the same. This assumption is based on the robustness of IBVS to depth errors [26]. The position error of each feature in the image plane are defined in Eq. 16 considering i -th image features.

$$\tilde{\xi}_n = \xi_n - \xi_{n,\text{ref}} = \begin{bmatrix} \xi_{1,n} - \xi_{r,1,n} \\ \xi_{2,n} - \xi_{r,2,n} \\ \vdots \\ \xi_{i,n} - \xi_{r,i,n} \end{bmatrix}. \tag{16}$$

For the non-linear optimization framework, the cost function will include three terms: quadratic terms for both image feature error and input cost, and an additional term which resembles the barriers functions proposed by [15]. The cost equation is given as follows

$$\min_{\mu_1, \dots, \mu_N} V(\xi, \mu) = \underbrace{\sum_{n=1}^N \tilde{\xi}_n^T \mathbf{Q}_\xi \tilde{\xi}_n + \mu_n^T \mathbf{Q}_\mu \mu_n}_{J_1} + \underbrace{\sum_{n=1}^N \alpha_g \cdot g(z_n, \eta_n)}_{J_2}, \quad (17)$$

subjected to:

$$\tilde{\xi}_n = \xi_n - \xi_{n,\text{ref}}, \quad (18)$$

$$\xi_{n+1} = \mathbf{f}_{d,\Delta t}(\xi_n, z_n, \mu_n), \quad (19)$$

$$\xi_n \in \mathcal{S} = \{\xi \in \mathbb{R}^{2i \times 1} : \mathbf{s}_{\min} \leq \xi \leq \mathbf{s}_{\max}\}, \quad (20)$$

$$\mu_n \in \mathcal{U} = \{\mu \in \mathbb{R}^{4 \times 1} : \mu_{\min} \leq \mu \leq \mu_{\max}\}. \quad (21)$$

where N is the prediction horizon, $\mathcal{U}$ and $\mathcal{S}$ represents the image plane and control input constraint sets, $\mathbf{Q}_\xi$ and $\mathbf{Q}_\mu$ are diagonal weighting matrices, J_1 corresponds to the traditional quadratic cost, and J_2 is the additional cost that takes into account the target states. For this paper only the target roll angle is considered, therefore α_g is a scalar weight, η_n corresponds to the roll angle, and $g(z_n, \phi_n)$ is the custom barrier function, which will be described later. Note that, as the focus is solely on short-term forecasts, the control horizon is set equal to the prediction horizon. Moreover, for each time step the optimization runs and provides only the first control input to the UAV.

Based on the work developed in [15] the barrier function $g(z_n, \phi_n)$ is defined in Eqs. 22 and 23.

$$g(z_n, \phi_n) = f(z_n) \cdot (\phi_n)^2 \quad (22)$$

$$f(z_n) = \begin{cases} f_1(z_n) = \left(1.0 + \exp\left(\frac{z_n - z_{d,1}}{\alpha_1}\right)\right)^{-1}, & \text{if } z_n \geq z_{ref}, \\ f_2(z_n) = \left(1.0 + \exp\left(\frac{z_n - 0.1}{\alpha_2}\right)\right)^{-1}, & \text{if } z_n < z_{ref}. \end{cases} \quad (23)$$

Fundamentally, this function represents two sigmoid functions with a transition height of z_{ref} , where $z_{d,1}$ and $z_{d,2}$ are parameters that define the offset, while α_1 and α_2 control its steepness. Note that $\alpha_1 > 0$ and $\alpha_2 < 0$. As presented in Eq. 17, the combined sigmoid functions in Eq. 23 are weighted by both the forecast roll angle and the scalar weight α_g . This approach ensures that the cost function increases when high roll angles are detected, effectively preventing the UAV from descending below the threshold height z_{ref} . Therefore, the steepness controlled by α_2 will actually define the extension of the waiting region, while α_1 the aggressiveness of the landing. By properly tuning the relationship between α_g and the other weight matrices, $\mathbf{Q}_\xi$ and $\mathbf{Q}_\mu$, it is possible to restrict the landing envelope. An example of a $f(z_n)$ function is given in Fig. 2.

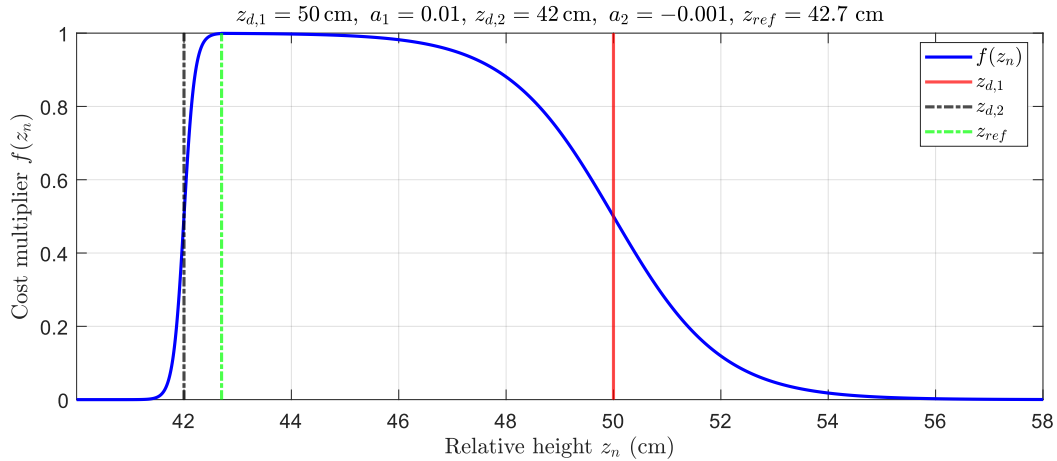


Fig. 2 Example of $f(z_n)$.

The CasADi optimization interface [27] which includes the IPOPT NLP solver [28] is used in this paper with a direct multiple shooting. In contrast to [18], the UAV dynamics are excluded from the MPC framework due to performance and feasibility constraints in real-time applications. One reason is the number of states considered by the VS-NMPC, for instance, four features, which is the minimum required by the IBVS, will result in a total of eight states. However, a notable drawback of this approach is that the visual servoing dynamics fails to account for the UAV's control bandwidth limitations. To address this issue, an Input Scaling Gain (ISG) technique inspired by [29] is introduced. Originally, this method was proposed to mitigate oscillations in the Incremental Nonlinear Dynamics Inversion (INDI) controller caused by delays from the angular acceleration estimator. The ISG achieves this by incorporating a discrete low-pass filter into the control outputs, effectively enhancing stability but limiting tracking performance and resulting in a suboptimal control input as well. Therefore, this can be achieved by multiplying the output from the VS-NMPC by a diagonal gain matrix Λ .

V. Simulations

Simulations were performed to compare results provided by three approaches: (i) baseline IBVS with landing criteria based on current target states, (ii) IBVS with landing criteria based on FFT+DKF forecasted target states (IBVS + FFT/DKF), and (iii) VS-NMPC using FFT+DKF forecasts. Since the UAV considered for the flight tests has a minimum operation height of 30 cm due to limitations of the optical flow and LiDAR systems, it was considered a simulated landing height. That means that whenever the UAV reaches values below the simulated landing height (42 cm), the attitude of the landing would represent the conditions faced by the UAV during touchdown. Therefore, for the simulations, the landing procedure definitions are outlined in Fig. 3 and compared with a real scale, for reference.

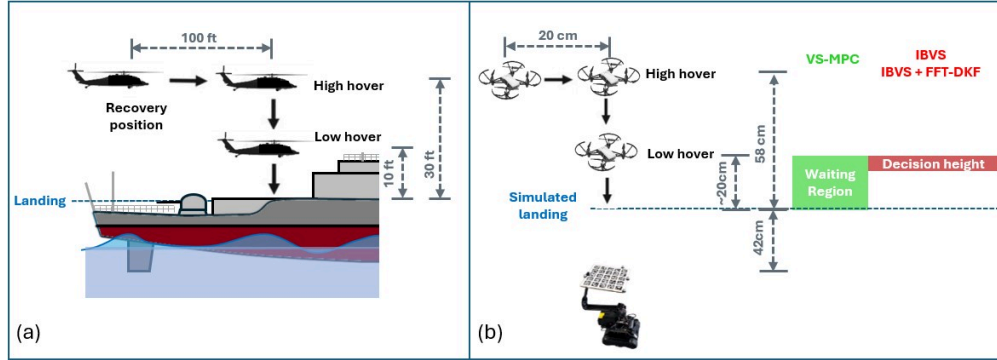


Fig. 3 Landing procedure: (a) example of real scale for MQ-8C rotorcraft [5], (b) definitions for the low scale simulations.

All controllers are based on the detections of four features, which are the Aruco tags positioned in the corner of the board. An example of the vision pipeline is presented in Fig. 4.

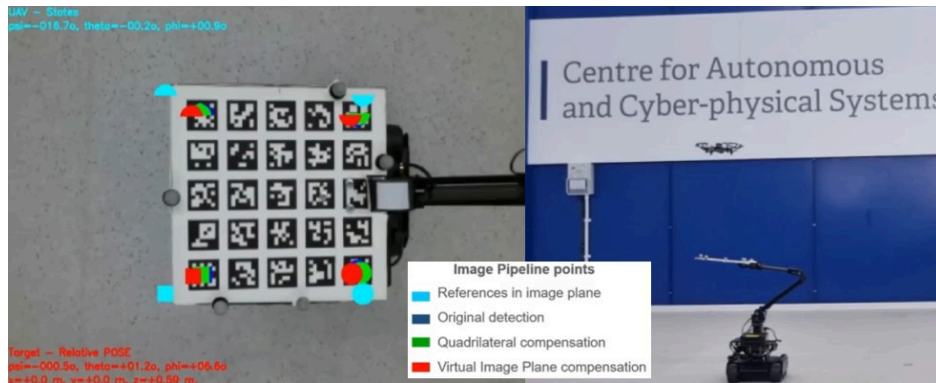


Fig. 4 Vision pipeline with detection of four corners.

For both IBVS controller (i) and (ii), the landing procedure consists of a first stage which is the high hover tracking, and then a second one would be the lower hover tracking. Next, the landing was triggered based on the landing pad states and the tracking errors on the image plane. A hierarchical controller was designed to handle the transition between stages. The transition from the first stage to the second one is triggered based on the following condition: the four features provided by the VIP (red points in Fig. 4) should stay inside a circle of 40 pixels radius from their respective tracking references for at least 3 s. During the second stage, the criteria to trigger includes the same image plane tolerance. Moreover, for (i) the absolute value of the current roll angle of the landing pad should be less than 5° , while (ii) requires that all the angles forecasted up to 1 s ahead are within the same range. This also means that if the UAV reaches the low hover position, it will wait for the forecast algorithm to finish at least the first ΔT_w before attempting the landing. In both IBVS cases, during the landing phase, a descent rate of 20 cm/s during the landing phase. For the VS-NMPC controller the barrier function is the same presented in Fig. 2. Thus, this results in a waiting region of 20 cm above the simulated landing height as detailed in Fig. 3(b). It is also tuned to prioritize landings within the roll range of 5° . These criteria are summarized in Table 1.

Table 1 Comparison of Visual Servoing Methods

Stage	Baseline IBVS	IBVS FFT/DKF	VS-NMPC
From high to low hover	Features within 40 pixel radius of tracking values for at least 3 s		
From low hover to landing	Features within 40 pixel radius for 3s and $ \phi_0 < 5^\circ$	Features within 40 pixel radius for 3s and $ \phi_n < 5^\circ$, for $i \in \{0, 1, \dots, N\}$	Based on barrier function and landing pad states

For the simulations, a simplified approach was used in order to provide a baseline for the UAV dynamics. The outer loop of the UAV was identified through a black box approach without delays. The database considered was the control inputs provided to the UAV and the output were the translational speeds and yaw rate (v_b). The UAV position and attitude were captured using a VICON motion capture system [30], while a Kalman filter was implemented to estimate translational speeds and yaw rate. The system identification assumed decoupled dynamics as the study does not focus on aggressive maneuvers. This simplified model also provides a reference for comparing the visual servoing controllers. An example of training and validation data for v_x, b is illustrated in Fig. 5.

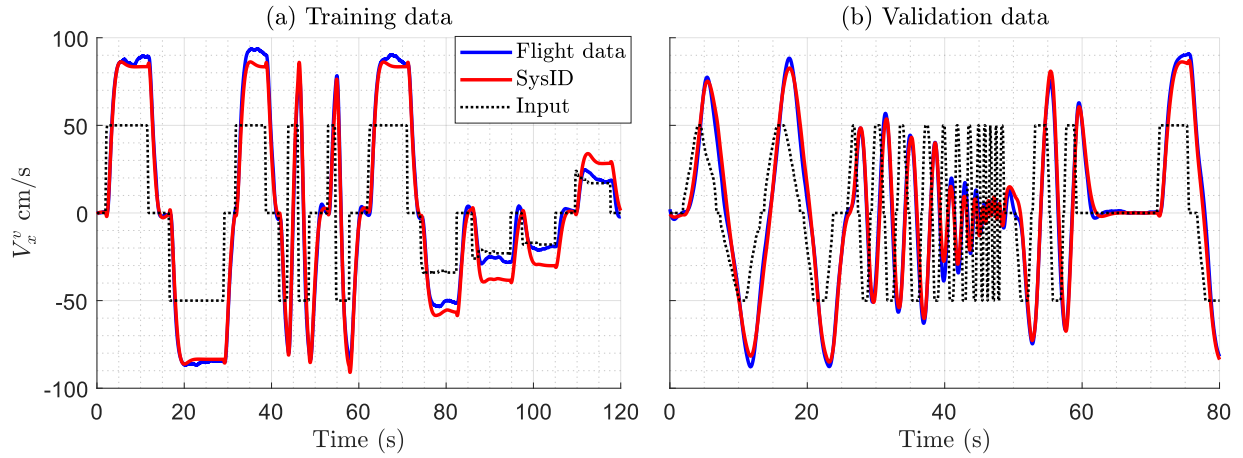


Fig. 5 v_x, b system identification.

As a result, the UAV dynamics considered for the simulations are given by the transfer functions given by Eqs. 24, 25, 26, and 27, where $\mathbf{u}_{cr} = (u_x, u_y, u_z, u_\omega)^T \in \mathbb{R}^4$ is the control input. Moreover, it was assumed no ground effect within the simulations.

$$\frac{v_{x,n}}{u_{x,b}} = \frac{-0.6919s + 3.4}{s^2 + 2.11s + 2.036}. \quad (24)$$

$$\frac{v_{y,n}}{u_{y,b}} = \frac{-0.935s + 4.046}{s^2 + 2.438s + 2.43}. \quad (25)$$

$$\frac{v_{z,n}}{u_{z,b}} = \frac{-0.8683s + 4.351}{s^2 + 3.394s + 7.077}. \quad (26)$$

$$\frac{\omega_{z,b}}{u_{\omega_{z,b}}} = \frac{-1.543s + 15.25}{s^2 + 10.24s + 23.16}. \quad (27)$$

It is important to note that the simulations do not account for the influence of the attitude of the UAV on the features positioned on the image plane. Essentially, the camera is assumed to be mounted on gimbals that maintain a downward orientation, regardless of the UAV attitude. However, the results still provide valuable insight into the performance of each control strategy. Furthermore, the UAV camera is modeled using a pinhole camera, incorporating the intrinsic parameters derived from the calibration process described in [10].

For the first stage, the tracking feature was placed in the image plane to achieve a relative height of $z_n = 100$ cm, while for the second stage, the relative height was set to $z_n = 42$ cm. Both IBVS solutions are based on the following gains $\lambda_s = \text{diag}(0.15, 0.15, 0.15, 1)$. For VS-NMPC strategy, the following parameters were defined: $\Lambda = \text{diag}(0.05, 0.05, 0.2, 0.2)$, $\mathbf{Q}_\xi = \mathbf{I}$, $\mathbf{Q}_\mu = 1.0 \times 10^3$, $\alpha_g = 1.0 \times 10^8$, $N = 20$, $\Delta t = 0.05s$. Whenever the UAV reaches a height below $z_n = 42$ cm the UAV was repositioned in the first stage.

The landing pad roll and the relative height are shown in Fig. 6. This figure presents the results of the VS-NMPC controller for the first 5 min of simulation. Successful landings were defined as simulated touchdowns with target attitude of $|\phi_{target}| < 5^\circ$. Fig. 7 illustrates the results comparing each controller for the same time period.

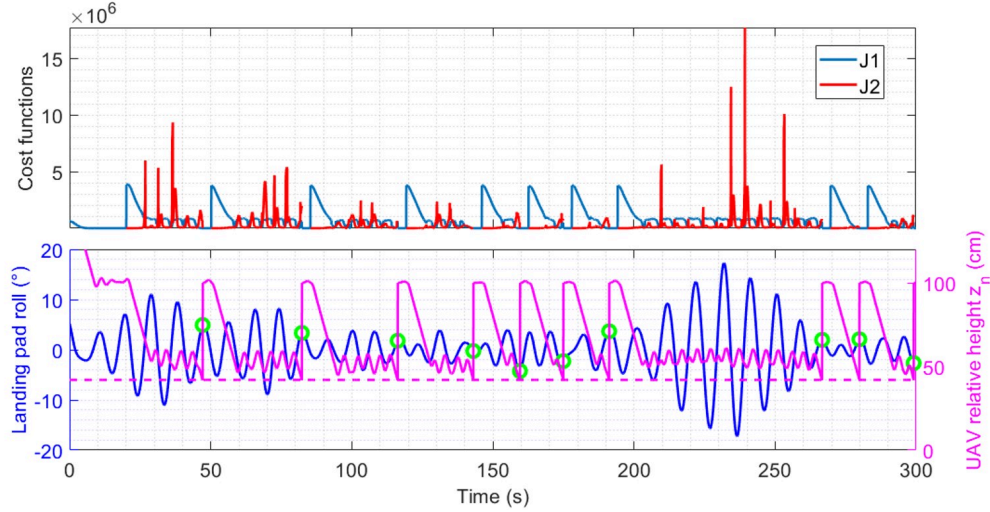


Fig. 6 VS-NMPC: J_1 and J_2 cost functions.

In Fig. 7, the dynamics of the vessel exhibits a notable change between the time frames of 200 s and 270 s, transitioning to a motion of higher amplitude. Under these adverse conditions, both the baseline and the improved IBVS results in unsafe landing conditions, whereas the VS-NMPC successfully avoids them. Although IBVS + FFT/DKF still take into account the forecasted states, the fast dynamics of the target still triggers unsafe landings. However, the VS-NMPC attempts fewer landings when compared with the other IBVS, even when the conditions are within the safe threshold (for instance, the time interval from 100 to 200 s).

The statistical analysis for the entire simulation is summarized in Table 2. Although the VS-NMPC method performed fewer landing attempts, it achieved a higher success rate. An improvement 14% was achieved compared to the baseline IBVS approach, which is similar to [12, 13].

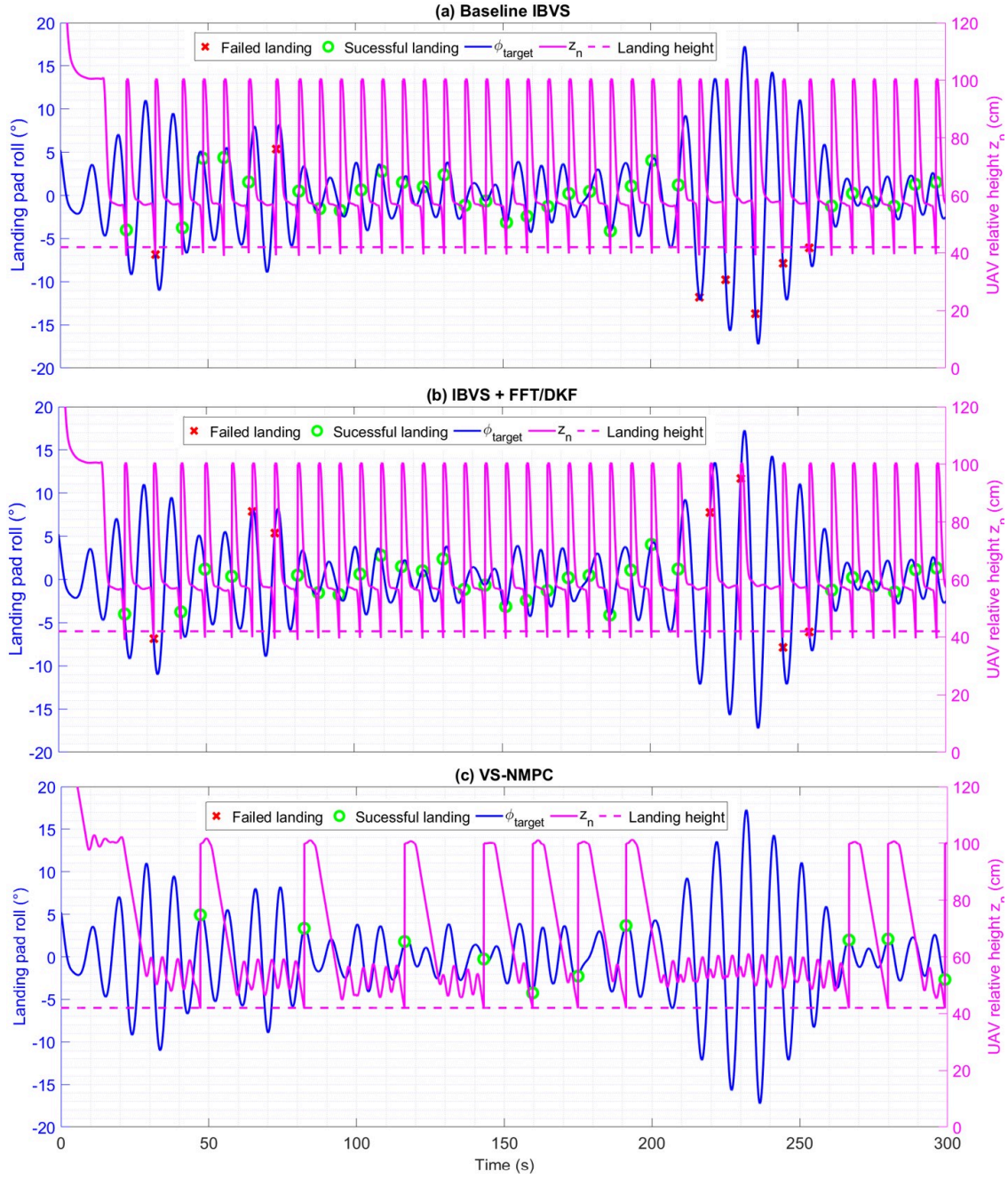


Fig. 7 First 5 min of simulation for (a) Baseline IBVS, (b) IBVS + FFT/DKF, and (c) VS-NMPC.

Table 2 Performance comparison of different methods.

Method	Total Attempts	Successful Landing Rate (%)	Attempts per Minute
Baseline IBVS	157	76%	8
IBVS+FFT/DKF	150	81%	8
VS-NMPC	39	90%	2

VI. Flight tests

The flight tests were performed using the same hardware as [10]. Fig 8 presents details of both UGV and UAV, while Fig. 9 outlines the overview architecture of the experimental setup. Initial flight tests were performed with the robotic arm simulating the roll motion under SS 6 using the dynamics described in Chapter IV.A.

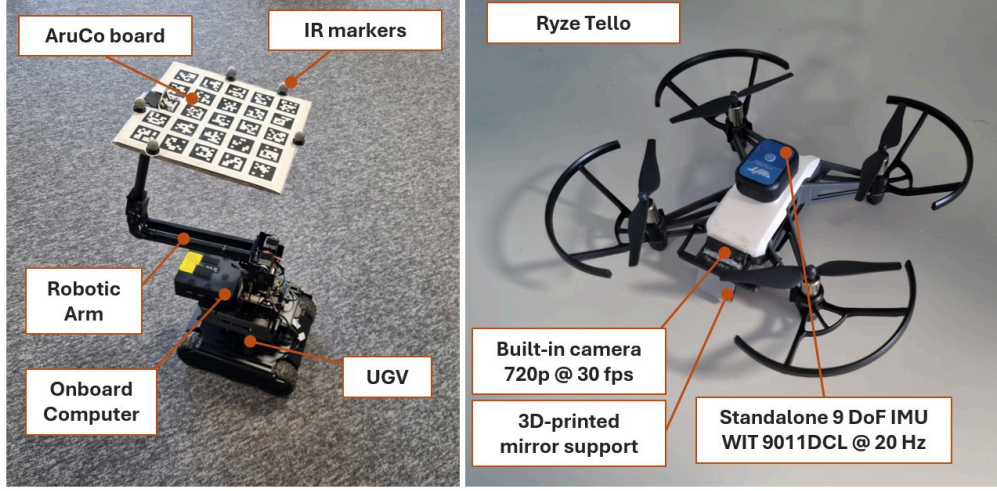


Fig. 8 UAV and UGV detailed description.

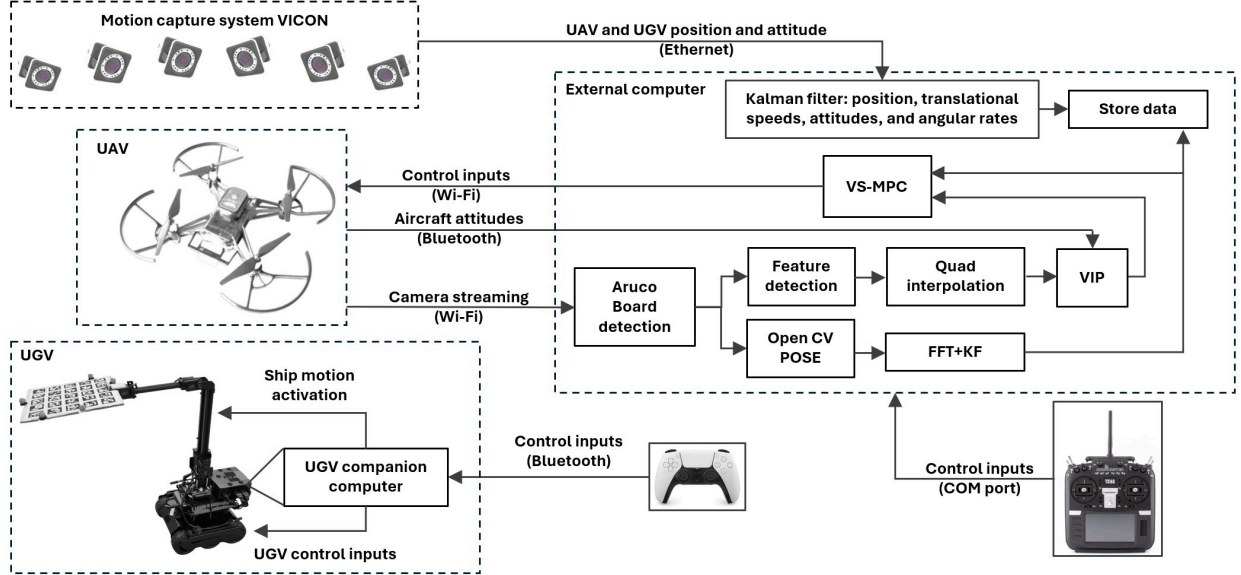


Fig. 9 Overall experimental setup.

For the flight, the simulated landing height was set 47 cm. An additional sigmoid barrier was introduced below that height to avoid possible collisions with the landing pad. Note that this function will only be activated if the simulated touchdown is not triggered. Therefore, the equations are presented in Eq. 28 along with definitions $z_{d,1}$, $z_{d,2}$, α_1 , α_2 , and z_{ref} . As a result, the final barrier function is described in Fig. 10.

$$f(z_n) = \begin{cases} f_1(z_n) = \left(1.0 + \exp\left(\frac{z_n - 56}{0.01}\right)\right)^{-1}, & \text{if } z_n \geq 49 \text{ cm}, \\ f_2(z_n) = \left(1.0 + \exp\left(\frac{z_n - 48}{-0.001}\right)\right)^{-1} + \left(1.0 + \exp\left(\frac{z_n - 44}{0.005}\right)\right)^{-1}, & \text{if } z_n < 49 \text{ cm}. \end{cases} \quad (28)$$

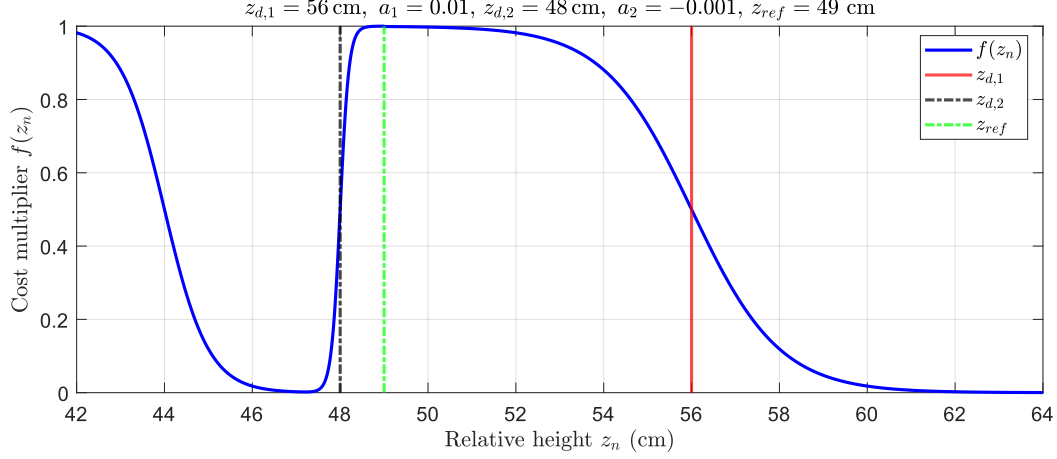


Fig. 10 Barrier function definition for flight tests.

The following parameters were used in VS-NMPC flight tests: $\lambda_s = \text{diag}(0.10, 0.10, 0.20, 0.30)$, and, for the second stage, $\alpha_g = 5.0 \times 10^7$. The remaining parameters are the same as the simulations. Two flight tests were performed and the results are provided in Fig. 11, and Fig. 12. Both figures show the evolution of J_1 and J_2 during the autonomous landing. Fig. 12 also presents the roll angle of the landing pad based on the motion capture data.

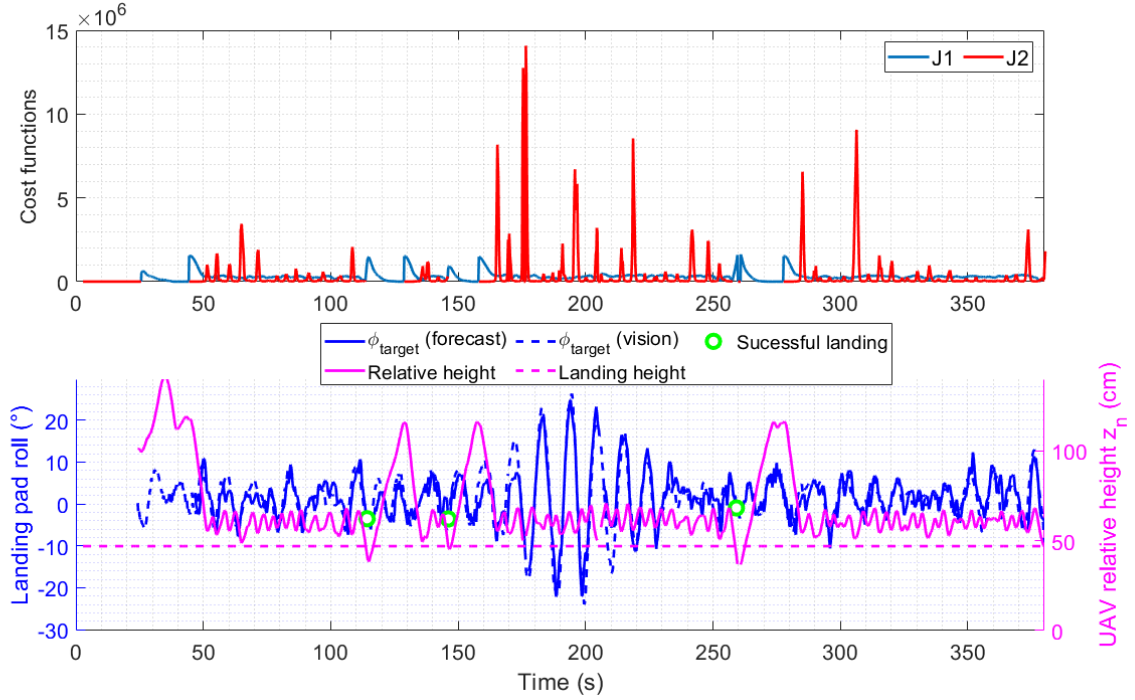


Fig. 11 Results for the first flight test.

First, it can be seen that the vision pipeline matches the true UAV attitude in Fig. 12. The roll forecast is shown in each plot considering 1 s ahead; however, this is done to indirectly provide a performance of the FFT+DKF algorithm as it is the worst-case scenario within the whole forecast window. Although the overall forecast accuracy matches the vision pipeline, it can be seen that poor results are achieved in some specific time stamps even resulting in considerable phase delays. For example, this can be seen in Fig. 11 between 150 and 180 s, and in Fig. 12 between 60 and 80 s.

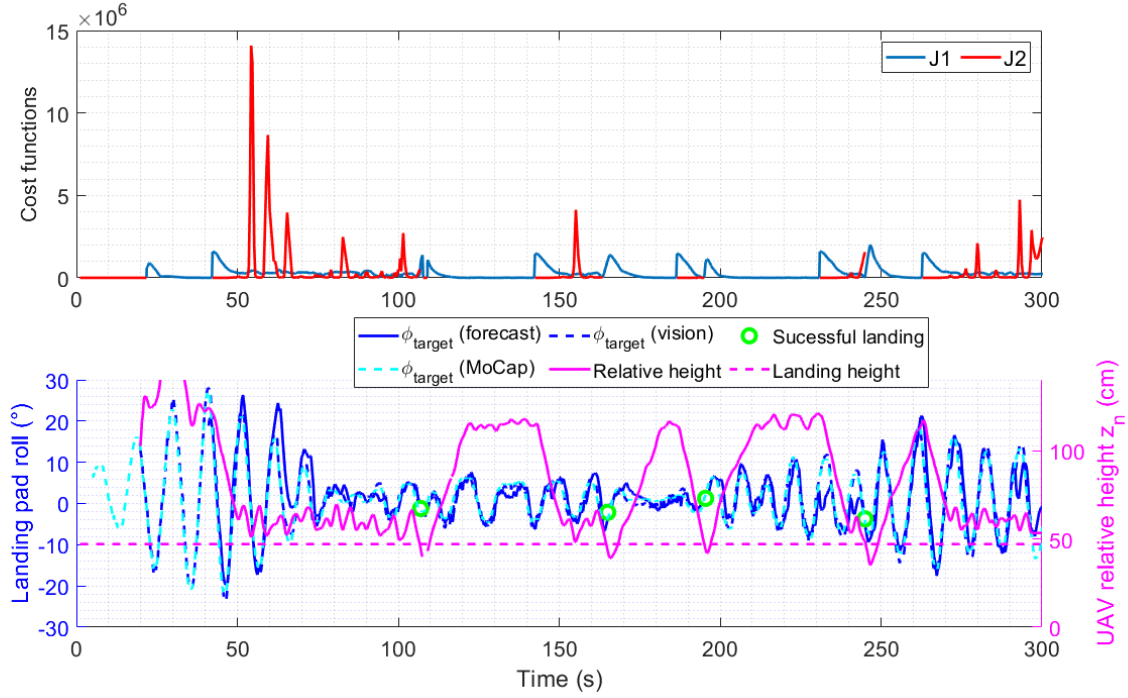


Fig. 12 Results for the second flight test.

In total, 7 landings were performed, all of them being successful. In Fig. 11 between 150 and 230 s the VS-NMPC controller does not allow the UAV to proceed with the landing. As mentioned, the ISG intentionally decreases the controller bandwidth to take into account the dynamics of the UAV. However, one drawback is that the total number of landing attempts is significantly reduced.

Although the success rate of the flight tests demonstrated the feasibility of the approach, only two flight tests were considered. Additional flight tests in different sea state conditions could provide more insight into the controller performance. Further improvements of the VS-NMPC could consider other states beyond the roll angle. Currently, the proposed controller does not track the heave motion of the platform, which is an important DoF for heavy-class vessels. In addition, the number of landing attempts was still considered low, which could be considered a drawback in scenarios with time constraints, for instance, in scenarios where the UAV presents a low remaining power supply.

VII. Conclusion and future works

This paper presented an evaluation of visual servoing controllers for autonomous landing of UAVs on dynamic landing pad with roll motion. In this study, the platform motion was simulated using a scaled-down setup that includes the motion of a heavy-class destroyer under sea state condition 6. As a result, autonomous landings were performed using a standard IBVS, an enhanced IBVS that included target forecast states in to the landing criteria, and the novel VS-NMPC approach. The VS-NMPC integrates visual servoing kinematics within the MPC framework, incorporating target dynamics through barrier cost functions. Although the VS-NMPC method resulted in fewer landing attempts, simulations demonstrated its superior performance by achieving a higher success rate under challenging sea-state conditions. Two flight tests showed the feasibility of the VS-NMPC controller, although further tests are required to assess its performance.

Future work should incorporate heave motion, as it is an important DoF for the scope of autonomous landing on vessels, and explore new barrier functions that include roll rate and heave motion for better performance. Additionally, while FoV constraints were considered, the current approach relies on the features within the virtual image plane. A more accurate method should account for the original image plane to avoid losing features.

Acknowledgments

This work was supported and sponsored by MBDA UK and the Brazilian Air Force through sponsorship P20227.

References

- [1] Chopra, I., “Small UAS and Delivery Drones: Challenges and Opportunities the 38th Alexander A. Nikolsky Honorary Lecture,” *Journal of the American Helicopter Society*, Vol. 66, No. 4, 2021, pp. 1–41. <https://doi.org/10.4050/JAHS.66.042001>.
- [2] Mohsan, S. A. H., Othman, N. Q. H., Li, Y., Alsharif, M. H., and Khan, M. A., “Unmanned Aerial Vehicles (UAVs): Practical Aspects, Applications, Open Challenges, Security Issues, and Future Trends,” *Intelligent Service Robotics*, Vol. 16, No. 1, 2023, pp. 109–137. <https://doi.org/10.1007/s11370-022-00452-4>.
- [3] Shafiee, M., Zhou, Z., Mei, L., Dinmohammadi, F., Karama, J., and Flynn, D., “Unmanned Aerial Drones for Inspection of Offshore Wind Turbines: A Mission-Critical Failure Analysis,” *Robotics*, Vol. 10, No. 1, 2021, pp. 1–27. <https://doi.org/10.3390/robotics10010026>.
- [4] Li, J., Zhang, G., Jiang, C., and Zhang, W., “A Survey of Maritime Unmanned Search System: Theory, Applications and Future Directions,” *Ocean Engineering*, Vol. 285, October, 2023, p. 115359. <https://doi.org/10.1016/j.oceaneng.2023.115359>.
- [5] Duehring, D., Gandhe, A., and Egan, B., “Test and Evaluation of Image-Based Navigation for Shipboard Landing on the MQ-8C Fire Scout Unmanned Aircraft System,” *Proceedings of the Vertical Flight Society 79th Annual Forum*, The Vertical Flight Society, West Palm Beach, FL, USA, 2023, pp. 1–10. <https://doi.org/10.4050/f-0079-2023-18188>.
- [6] Hoencamp, A., “Helicopter-Ship Qualification Testing,” Ph.D. thesis, Mechanical, Maritime and Materials Engineering, TU Delft, Delft, Netherlands, 2015. URL <https://repository.tudelft.nl/record/uuid:711fe747-a30b-4e15-81d5-e93e5ce06ac5>.
- [7] Helideck Certification Agency, “Helideck Limitations List Part C,” <https://www.helidecks.org/wp-content/uploads/2023/07/HLL-Part-C-P-R-H-Tables-Feb-2023.pdf>, 2014. Last accessed: November 25, 2024.
- [8] Leandro de Paula, L. G., Dwivedi, V. S., Shin, H.-S., and Tsourdos, A., “A Review on VTOL Autonomous Landing Strategies on Naval Dynamic Surfaces,” *Proceedings of the 2024 CEAS EuroGNC conference*, Bristol, UK, 2024, pp. 1–30. URL <https://eurognc.ceas.org/archive/EuroGNC2024/html/CEAS-GNC-2024-048.html>, cEAS-GNC-2024-048.
- [9] Britcher, V., Datta, A., and Chopra, I., “Refinement and Characterization of Feature-Based Vision Algorithm for Ship-Deck Landing under Degraded Visibility,” *Journal of the American Helicopter Society*, Vol. 70, No. 1, 2025, pp. 1–17. <https://doi.org/10.4050/JAHS.70.012009>.
- [10] de Paula, L. G. L., Dwivedi, V. S., Shin, H.-S., and Tsourdos, A., “Towards Autonomous Landing on Dynamic Naval Surfaces Under Harsh Sea State Motion,” *AIAA SCITECH 2025 Forum*, Orlando, Florida, USA, 2025, pp. 1–23. <https://doi.org/10.2514/6.2025-2607>.
- [11] Sampedro, C., Rodriguez-Ramos, A., Gil, I., Mejias, L., and Campoy, P., “Image-Based Visual Servoing Controller for Multirotor Aerial Robots Using Deep Reinforcement Learning,” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 979–986. <https://doi.org/10.1109/IROS.2018.8594249>.
- [12] Wynn, J. S., and McLain, T. W., “Visual Servoing with Feed-Forward for Precision Shipboard Landing of an Autonomous Multirotor,” *2019 American Control Conference (ACC)*, Philadelphia, PA, USA, 2019, pp. 3928–3935. <https://doi.org/10.23919/ACC.2019.8814694>.
- [13] Cho, G., Choi, J., Bae, G., and Oh, H., “Autonomous Ship Deck Landing of a Quadrotor UAV using Feed-forward Image-based Visual Servoing,” *Aerospace Science and Technology*, Vol. 130, November, 2022, p. 107869. <https://doi.org/10.1016/j.ast.2022.107869>.
- [14] Arif, A., Wang, H., Castaneda, H., and Wang, Y., “Finite-Time Tracking of Moving Platform with Single Camera for Quadrotor Autonomous Landing,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 70, No. 6, 2023, pp. 2573–2586. <https://doi.org/10.1109/TCSI.2023.3256063>.
- [15] Gupta, P. M., Pairet, E., Nascimento, T., and Saska, M., “Landing a UAV in Harsh Winds and Turbulent Open Waters,” *IEEE Robotics and Automation Letters*, Vol. 8, No. 2, 2022, pp. 744–751. <https://doi.org/10.1109/LRA.2022.3231831>.
- [16] Hendrick, C. M., Jaques, E. R., Horn, J. F., Langelaan, J. W., and Sydney, A. J., “Model-Scale Evaluation of Autonomous Ship Landing Guidance and Control Modes,” *Journal of the American Helicopter Society*, Vol. 69, No. 2, 2024, pp. 1–20. <https://doi.org/10.4050/JAHS.69.022008>.

- [17] Abujoub, S., McPhee, J., and Irani, R. A., "Methodologies for landing autonomous aerial vehicles on maritime vessels," *Aerospace Science and Technology*, Vol. 106, November, 2020, p. 106169. <https://doi.org/10.1016/j.ast.2020.106169>.
- [18] Recalde, L. F., Varela, J., Guevara, B. S., Andaluz, V., Gimenez, J., and Gandolfo, D., "Constrained Visual Servoing of Quadrotors Based on Model Predictive Control," *IFAC-PapersOnLine*, Vol. 55, No. 37, 2022, pp. 353–360. <https://doi.org/10.1016/j.ifacol.2022.11.209>, 2nd Modeling, Estimation and Control Conference MECC 2022.
- [19] Proctor, J. L., Brunton, S. L., and Kutz, J. N., "Dynamic Mode Decomposition with Control," *SIAM Journal on Applied Dynamical Systems*, Vol. 15, No. 1, 2016, pp. 142–161. <https://doi.org/10.1137/15M1013857>.
- [20] Yang, L., Wang, X., Liu, Z., and Shen, L., "Robust Online Predictive Visual Servoing for Autonomous Landing of a Rotor UAV," *IEEE Transactions on Intelligent Vehicles*, Vol. 9, No. 2, 2024, pp. 1–17. <https://doi.org/10.1109/TIV.2024.3385283>.
- [21] Bales, S., Meyers, W., and Rossignol, G., "Response Predictions of Helicopter Landing Platform for The USS Belknap (DLG-26) And USS Garcia (DE-1040)-Class Destroyers," Technical Report AD0766450, US Navy Ship Research and Development Center, Bethesda, Maryland, 1973.
- [22] Applebee, T., and Baitis, A., "Response Amplitude Operator Predictions for the USS Belknap (DLG-26) And USS Joseph Hewes (DE-1052) Class Destroyers," Technical Report ADA002233, US Naval Ship Research and Development Center, Bethesda, Maryland, 1974.
- [23] BALES, S. L., "Designing Ships to the Natural Environment," *Naval Engineers Journal*, Vol. 95, No. 2, 1983, pp. 31–40. <https://doi.org/10.1111/j.1559-3584.1983.tb00574.x>.
- [24] Kang, Y., Park, B. J., Cho, A., Yoo, C. S., Kim, Y., Choi, S., Koo, S. O., and Oh, S., "A Precision Landing Test on Motion Platform and Shipboard of a Tilt-Rotor UAV Based on RTK-GNSS," *International Journal of Aeronautical and Space Sciences*, Vol. 19, No. 4, 2018, pp. 994–1005. <https://doi.org/10.1007/s42405-018-0081-8>.
- [25] OpenCV, "Detection of ArUco Boards," https://gregorkovalcik.github.io/opencv_contrib/tutorial_aruco_board_detection.html, n.d. Accessed: 10 Nov 2024.
- [26] Corke, P., Jachimczyk, W., and Pillat, R., *Image Formation*, Springer International Publishing, Cham, 2023, Chaps. 15 Vision-Based Control, pp. 543–591. https://doi.org/10.1007/978-3-031-07262-8_13.
- [27] Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M., "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, Vol. 11, No. 1, 2019, pp. 1–36. <https://doi.org/10.1007/s12532-018-0139-4>.
- [28] Wächter, A., and Biegler, L. T., "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57. <https://doi.org/10.1007/s10107-004-0559-y>.
- [29] Cordeiro, R. A., Marton, A. S., Azinheira, J. R., Carvalho, J. R. H., and Moutinho, A., "Increased Robustness to Delay in Incremental Controllers Using Input Scaling Gain," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 58, No. 2, 2022, pp. 1199–1210. <https://doi.org/10.1109/TAES.2021.3123215>.
- [30] Quintero, J. M. C., Garcia-Nieto, S., Simarro, R., and Ignatyev, D. I., "Improving Racing Drones Flight Analysis: A Data-Driven Approach Using Motion Capture Systems," *Drones*, Vol. 8, No. 12, 2024, p. 742. <https://doi.org/10.3390/drones8120742>.



DSPACE

<https://dspace.org/>

Visual servoing MPC framework for shipboard autonomous landing

de Paula, Luís G. L.; Dwivedi, Vijay S.; Shin, Hyo-Sang; Tsourdos, Antonios

2025-07-21

American Institute of Aeronautics and Astronautics (AIAA)

<https://doi.org/10.2514/6.2025-3547>

de Paula LGL, Dwivedi VS, Shin H-S, Tsourdos A. (2025) Visual servoing MPC framework for shipboard autonomous landing. In: AIAA Aviation Forum and Ascend 2025, 21-25 July 2025, Las Vegas, USA, Paper number AIAA 2025-3547