Linear Velocity-free Visual Servoing Control for Unmanned Helicopter Landing on a Ship with Visibility Constraint

Yanting Huang, Ming Zhu, Zewei Zheng, and Kin Huat Low

***Abstract*—In this paper, a constrained image-based visual servoing control method for the shipboard landing problem of unmanned helicopters is proposed. First, the pitch and roll motion of ship are predicted by an auto-regressive (AR) model to determine an appropriate period for landing. Subsequently, a novel robust sliding mode controller without linear velocity measurements is developed on the basis of the perspective image feature in a virtual image plane. Meanwhile, a modified Chebyshev Neural Network (CNN) is proposed to estimate the uncertainties including the linear acceleration of ship and translational perturbation, while an adaptive law is employed to compensate the influence of rotational disturbances. The whole controller only requires the measurements feedback of a vision sensor and an inertial measurement unit (IMU). Ulteriorly, to prevent the visual target on the ship from going beyond the field of view of camera, the constrained controller is developed by a control barrier function and a quadratic programming, where the unknown relative velocity is estimated by a velocity observer. Finally, simulations are implemented to substantiate the capability of the presented shipboard landing control method.**

***Index Terms*—unmanned helicopter, shipboard landing, visual servoing, output-feedback control, visibility constraint.**

I. Introduction

RECENTLY, the unmanned helicopter has received an increasing interest in the area of maritime operations on account of its superiority of vertical taking-off and landing, low-speed flight, low-height hovering and high maneuverability [1]. These advantages motivate the offshore application of the unmanned helicopter such as surveillance, inspections, search and rescue missions [2]. Therefore, it is crucial but still challenging for the unmanned helicopter to achieve autonomous shipboard landing [3].

This work was supported by the Beijing Natural Science Foundation (no. 4202038), China Postdoctoral Science Foundation (no. 2020TQ0028), the National Natural Science Foundation of China (no. 61827901), the National Key R&D Program of China (no. 2018YFC1506401), China Scholarship Council, and NTU ATMRI Project Agreement Research Grant (no. 2016-01). (*Corresponding author: Zewei Zheng*)

Yanting Huang is with the School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China, and Air Traffic Management Research Institute (ATMRI), Nanyang Technological University 639798, Singapore. (e-mail: huangyanting@buaa.edu.cn; N1906773C@e.ntu.edu.sg)

Ming Zhu is with the Institute of Unmanned System, Beihang University, Beijing 100191, China. (e-mail: zhuming@buaa.edu.cn)

Zewei Zheng is with the Seventh Research Division, School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, China. (e-mail: zeweizheng@buaa.edu.cn)

Kin Huat Low is with the School of Mechanical and Aerospace Engineering, Nanyang Technological University 639798, Singapore. (e-mail:mkhlow@ntu.edu.sg)

The first challenge for autonomous shipboard landing of the helicopter is caused by 6 degree-of-freedom (DOF) motion characteristics of ship in the complex sea environment. Unfortunately, it is impossible to control the relative position and the relative attitude concurrently because of the under-actuated property of helicopter. In order to avoid collision, an effective solution is to utilize a ship motion predictor. With the predicted information, the landing phase will be switched on only when there would be a future period in which the ship's pitch and roll remain small values [4]. Thus, the necessary controlled states are reduced to 4 DOF including relative position and relative yaw. The ship motion prediction information can be acquired by many methods, like Kalman filter [5], particle filter [6], and AR model [7]. This paper focuses on AR model which can provide the prediction information several seconds ahead in a short calculation time.

The second challenge for autonomous shipboard landing of the helicopter is to develop an accurate navigation system. Some existing technologies including tracking radar [8], Global Positioning System (GPS) [9], and Inertial Navigation System [10] all require extra communication subsystems to send the measured ship pose to the helicopter, which increases the helicopter's payload. Alternatively, vision sensors with the properties of light weight and low cost can directly provide the relative pose between two vehicles. Visual servoing is generally divided into two classifications: Position-based Visual Servoing (PBVS) and Image-based Visual Servoing (IBVS) [11]. The former reconstructs the relative pose according to the 3D image information and control the vehicle with the estimated pose directly [12], while the latter control the vehicle indirectly by eliminating the image feature errors [11]. In this case, IBVS can avoid complicated reconstruction and provide more robustness against camera calibration errors [13]. The image feature, which can reflect the relative pose between vehicle and target, plays a significant role in IBVS. An effective image feature is defined on a virtual image plane which remains aligned with the visual target plane. Compared with other image features such as homography matrix [14] and spherical projection [15], the virtual image feature is unrelated with pitch and roll channels so as to simplify the image kinematics.

The third challenge for autonomous shipboard landing of the helicopter is to design a robust and reliable IBVS controller. The relative linear velocity between helicopter and ship are unavailable when only a vision sensor and an inertial measurement unit are equipped on the helicopter. Considering this,

the optical flow was exploited for the relative velocity measurement in [16], [17], [18]. However, the optical flow method has limitations to measure the relative velocity between two moving vehicles that real-time and practicality may not be guaranteed because of the complicated calculation. Cao et al. [19] proposed a trajectory observer to estimate the target's velocity based on nonlinear tracking-differntiator, but an accelerometer was required to obtain the quadrotor's velocity. A translational velocity observer was presented in [20] and [21] to estimate the relative velocity between quadrotor and target using actual image features. The aforementioned work did not address external disturbances, whereas they are necessary to be considered to improve the robustness of controller in the shipboard landing. There were a few methods like adaptive control [22], neural network [23], local filters [24], extended state observer [25] and adaptive disturbance observer [26] to handle uncertain external disturbances. Two output-feedback IBVS algorithms without the linear velocity measurement but with the assumption of constant disturbances were designed in [27] and [28], respectively. Another robust visual servoing control method for the quadrotor tracking a planar moving target was presented in [29], where an auxiliary system was introduced to avoid the measurement of linear velocity and the disturbance term is assumed to have a known constant boundary. Nevertheless, the IBVS researches which do not require the linear velocity measurement and consider time-varying disturbances with unknown bounds are still very limited.

Furthermore, it is indispensable to guarantee the visual target on the ship to be confined within the camera's field of view (FoV) for all time in the shipboard landing, which is called the visibility constraint. Losing sight of the target will lead to landing failure and even crash. Many efforts have been done to overcome the visibility problem. The saturated control input was designed in [30] to decrease the chance of visual features' loss. The work [31] proposed a constrained control method for landing UAVs on the basis of a subtle transformation from the output constraint into the input constraint. Confining the vehicle's attitude within small values is also an optional approach [19], [32]. The control law in [18] restricted thrust, desired pitch and roll to their predefined compact sets, which helps the visual target remaining within the FoV of camera. The work [21] proposed a solution to ensure that the projection coordinates of the static visual target on the image plane are within a specified circle by means of control barrier function (CBF). The underlying idea of CBFs is to restrict the control input in a linear constraint to ensure the states stay in the given admissible set by formulating a quadratic programming. CBFs have been widely investigated in the constrained control problem [33], [34].

This paper proposes a vision-based output-feedback control approach for autonomously landing the helicopter on a maneuvering ship, satisfying the visibility constraint. The major contributions in this paper are summarized as: (i) An IBVS control scheme for helicopter landing on a 6 DOF ship is presented based on an output-feedback controller and a ship motion predictor. (ii) A novel robust sliding mode control algorithm with two auxiliary systems is presented to track the desired image feature and conquer the lack of relative linear velocity measurement. (iii) A modified Chebyshev Neural Network without the measurement of relative linear velocity is integrated into the controller to estimate the uncertainties including time-varying external disturbances and the linear acceleration of ship. (iv) A constrained controller is developed via the control barrier function and the quadratic programming to keep the image target on the ship always in the FoV of camera, where a velocity estimator is introduced to reckon the relative translational velocity.

The remaining part of this paper proceeds as follows: Mathematical preliminaries are listed in Section II; Section III gives the problem formulation; Section IV is concerned with the detailed design of IBVS shipboard landing including ship motion prediction, output-feedback controller and further constrained controller; In Section V, simulations are executed to validate the presented method, followed by conclusion in Section VI.

II. Preliminaries

A. Notations

Through the whole of this paper, Notations are employed as follows. $|\cdot|$ refers to the absolute value of a scalar. $\langle\cdot\rangle$ describe the absolute value of each component in a vector. $\|\cdot\|$ means the Euclidean norm of a vector. $(\cdot)\circ(\cdot)$ refers to the Hadamard product of two matrices. $\mathrm{tr}(\cdot)$ represents the trace. $(\cdot)\odot(\cdot)$ is defined as the multiplication operation between two unit quaternions $\boldsymbol{Q}_1=[\eta_{01},\boldsymbol{\eta}_1^T]^T$ and $\boldsymbol{Q}_2=[\eta_{02},\boldsymbol{\eta}_2^T]^T$,

$$\boldsymbol{Q}_1\odot\boldsymbol{Q}_2=\begin{bmatrix}\eta_{01}\eta_{02}-\boldsymbol{\eta}_1^T\boldsymbol{\eta}_2\\ \eta_{01}\boldsymbol{\eta}_2+\eta_{02}\boldsymbol{\eta}_1+S(\boldsymbol{\eta}_1)\boldsymbol{\eta}_2\end{bmatrix}.$$

For a vector $\boldsymbol{a}\in\mathbb{R}^n$, $\mathbf{Tanh}(\boldsymbol{a})=[\tanh(a_1),\tanh(a_2),\cdots,\tanh(a_n)]^T$ is defined as a hyperbolic tangent function vector; $\mathrm{Sat}_{\bar{\boldsymbol{a}}}(\boldsymbol{a})=[\mathrm{sat}(a_1),\mathrm{sat}(a_2),\cdots,\mathrm{sat}(a_n)]^T$ is defined as a saturation function vector, where $\bar{\boldsymbol{a}}=[\bar{a}_1,\bar{a}_2,\cdots,\bar{a}_n]^T$, and

$$\mathrm{sat}(a_i)=\begin{cases}\bar{a}_i & a_i>\bar{a}_i\\ a_i & -\bar{a}_i\le a_i\le\bar{a}_i\\ -\bar{a}_i & a_i<-\bar{a}_i\end{cases}$$

For a vector $\boldsymbol{b}=[b_1,b_2,b_3]^T$, $\boldsymbol{S}(\boldsymbol{b})$ is the skew-symmetric matrix defined as

$$\boldsymbol{S}(\boldsymbol{b})=\begin{bmatrix}0&-b_3&b_2\\ b_3&0&-b_1\\ -b_2&b_1&0\end{bmatrix}$$

B. Chebyshev Neural Network

It has been proven that the CNN enable to commonly approximate any smooth function with any degree of accuracy on a compact set [23]. Suppose a smooth nonlinear function vector $\boldsymbol{F}(\boldsymbol{x}):\mathbb{R}^m\to\mathbb{R}^n$, it can be estimated by the following l-order CNN:

$$\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{W}^*\boldsymbol{\Phi}(\boldsymbol{x})+\boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{\epsilon}$ is the estimation error, $\boldsymbol{\Phi}(\boldsymbol{x})$ is the basis function vector obtained by

$$\begin{aligned}\boldsymbol{\Phi}(\boldsymbol{x}) = [1, T_1(x_1), \cdots, T_l(x_1), T_1(x_2), \cdots, T_l(x_2) \\ \cdots, T_1(x_m), \cdots, T_l(x_m)]^T\end{aligned} \tag{2}$$

with the Chebyshev polynomials $T_i(x_j)$ generated by the recursive formula:

$$T_{i+1}(x_j) = 2x_j T_i(x_j) - T_{i-1}(x_j), \quad T_0(x_j) = 1 \tag{3}$$

with $i = 1, \cdots, l-1; j = 1, \cdots, m$. $\boldsymbol{W}^* \in \mathbb{R}^{n \times (lm+1)}$ denotes an optimal weight matrix satisfying

$$\boldsymbol{W}^* = \underset{\hat{\boldsymbol{W}} \in \Omega_w}{\arg\min} \left\{ \sup_{\boldsymbol{x} \in \Omega_x} |\boldsymbol{F}(\boldsymbol{x}) - \hat{\boldsymbol{W}} \boldsymbol{\Phi}(\boldsymbol{x})| \right\} \tag{4}$$

with $\hat{\boldsymbol{W}}$ as the approximation of $\boldsymbol{W}^*$, Ω_x as the set of appropriate bounds of $\boldsymbol{x}$, and Ω_w denoted as

$$\Omega_w = \left\{ \hat{\boldsymbol{W}} | \text{tr}\{\hat{\boldsymbol{W}}^T \hat{\boldsymbol{W}}\} \leq W_M \right\} \tag{5}$$

with W_M as a positive constant.

C. Lemmas

Lemma 1: [35] Consider the following system

$$\ddot{\boldsymbol{\xi}} = -a_1 \textbf{Tanh}(b_1 \boldsymbol{\xi} + b_2 \dot{\boldsymbol{\xi}}) - a_2 \textbf{Tanh}(b_2 \dot{\boldsymbol{\xi}}) + \boldsymbol{\Delta} \tag{6}$$

where $\boldsymbol{\xi} \in \mathbb{R}^n$, $\boldsymbol{\Delta} \in \mathbb{R}^n$, and $a_i (i = 1, 2)$ and $b_j (j = 1, 2, 3)$ are positive constants. If there exists $\bar{\Delta} > 0$ and $\bar{t} > 0$ satisfying $\|\boldsymbol{\Delta}\| \leq \bar{\Delta} (t > \bar{t})$, and the parameters are selected based on the rules $b_1/b_2^2 > 2\bar{\Delta}$, $a_2 > (2\bar{\Delta} b_1/b_2^2)/(b_1/b_2^2 - 2\bar{\Delta})$, $\sqrt{a_2 b_1/b_2^2} \leq a_1 \leq \frac{1}{2}(a_2 + b_1/b_2^2)$, then $\boldsymbol{\xi}$ and $\dot{\boldsymbol{\xi}}$ ultimately converge to the attractive set

$$\boldsymbol{\Upsilon} = \left\{ [\boldsymbol{\xi}^T, \dot{\boldsymbol{\xi}}^T]^T \in \mathbb{R}^{2n} | \| [b_1 \boldsymbol{\xi}^T + b_2 \dot{\boldsymbol{\xi}}^T, b_2 \dot{\boldsymbol{\xi}}^T]^T \| < \bar{\xi} \right\} \tag{7}$$

where $\bar{\xi}$ satisfies $\frac{a_2 + (b_1/b_2^2)}{a_2 b_1/b_2^2 \bar{\Delta}} < \frac{\tanh^2(\bar{\xi})}{\bar{\xi}} < \frac{1}{2}$.

Lemma 2: [36] For any $x \in \mathbb{R}$ and $\varpi > 0$, the following inequality holds

$$0 \leq |x| - x \tanh(\frac{x}{\varpi}) \leq \kappa \varpi \tag{8}$$

where $\kappa = 0.2785$ satisfying $\kappa = e^{-(\kappa+1)}$.

Lemma 3: [37] Consider a system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ and a defined set $\mathcal{L} = \{\boldsymbol{x} \in \mathbb{R}^n | h(\boldsymbol{x}) \geq 0\}$, if the following inequality holds,

$$\dot{h}(\boldsymbol{x}) \geq -\kappa(h(\boldsymbol{x})), \quad \forall \boldsymbol{x} \in \mathcal{N} \tag{9}$$

then the set $\mathcal{L}$ is forward invariant, where the function $h : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and named as a zeroing (control) barrier function, $\kappa(\cdot)$ is a locally Lipschitz extended class κ function, and $\mathcal{L} \subseteq \mathcal{N} \subset \mathbb{R}^n$.

Proposition 1: Consider a system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ and a defined set $\mathcal{L} = \{\boldsymbol{x} \in \mathbb{R}^n | h(\boldsymbol{x}) \geq 0\}$, if the following inequality holds,

$$\dot{H}(\boldsymbol{x}) \geq -\kappa_2(H(\boldsymbol{x})), \quad \forall \boldsymbol{x} \in \mathcal{N} \tag{10}$$

then the set $\mathcal{L}$ is forward invariant, where $H(\boldsymbol{x}) = \dot{h}(\boldsymbol{x}) + \kappa_1(h(\boldsymbol{x}))$ is named as an extended zeroing barrier function, and $\kappa_1(\cdot)$ and $\kappa_2(\cdot)$ are locally Lipschitz extended class κ functions.

proof: According to Lemma 3, it can be proved with ease that the set $\mathcal{L}^* = \{\boldsymbol{x} \in \mathbb{R}^n | H(\boldsymbol{x}) \geq 0\}$ is forward invariant, that is, $\dot{h}(\boldsymbol{x}) \geq -\kappa_1(h(\boldsymbol{x}))$ always holds. Thus, the set $\mathcal{L}$ is also forward invariant.

III. Problem Formulation

A. Reference frames

The established reference frames are shown in Fig.1. The frame $\mathcal{F}_e = \{O_e, \boldsymbol{x_e}, \boldsymbol{y_e}, \boldsymbol{z_e}\}$ is regarded as the inertial frame. The frame $\mathcal{F}_q = \{O_q, \boldsymbol{x_q}, \boldsymbol{y_q}, \boldsymbol{z_q}\}$ denotes the body frame for the helicopter, where O_q is the geometric center point of the helicopter. We assume the camera frame $\mathcal{F}_c = \{O_c, \boldsymbol{x_c}, \boldsymbol{y_c}, \boldsymbol{z_c}\}$ (adhered to the projection center of camera) is accordant with the body frame $\mathcal{F}_q$ to facilitate the controller design. To release this assumption, two frames can be related by a constant translation transformation. The virtual image frame $\mathcal{F}_v = \{O_v, \boldsymbol{x_v}, \boldsymbol{y_v}, \boldsymbol{z_v}\}$ only inherits the translational motion of the helicopter and has no rotational motion, whose corresponding plane is the virtual image plane.

B. Helicopter Model

Using 'Newton-Euler' formulae, the helicopter model is established as [38]

$$\begin{cases} \dot{\boldsymbol{\xi}} = \boldsymbol{V} \\ \dot{\boldsymbol{V}} = -\dfrac{T_m}{m} \boldsymbol{R} \boldsymbol{e}_3 + g \boldsymbol{e}_3 + \boldsymbol{d}_t \\ \dot{\boldsymbol{Q}} = \boldsymbol{G} \boldsymbol{\Omega} \\ \boldsymbol{J} \dot{\boldsymbol{\Omega}} = -S(\boldsymbol{\Omega}) \boldsymbol{J} \boldsymbol{\Omega} + \boldsymbol{\tau} + \boldsymbol{d}_\tau \end{cases} \tag{11}$$

where $\boldsymbol{\xi}$ and $\boldsymbol{V}$ denote the position and the linear velocity of helicopter in $\mathcal{F}_e$, respectively; $\boldsymbol{Q} = [\eta_0, \boldsymbol{\eta}^T]^T$ is the unit quaternion satisfying $\eta_0^2 + \boldsymbol{\eta}^T \boldsymbol{\eta} = 1$; $\boldsymbol{\Omega}$ represents the angular velocity vector in $\mathcal{F}_q$; m is the total mass of helicopter; g is the gravity constant; $\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{Q}) = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\eta}^T \\ \eta_0 \boldsymbol{I}_3 + S(\boldsymbol{\eta}) \end{bmatrix}$; $\boldsymbol{J} = \text{diag}\{J_x, J_y, J_z\}$ is the inertial moment matrix; $\boldsymbol{e}_3 = [0; 0; 1]$. The matrix $\boldsymbol{R}$ is a coordinate transition matrix from the body frame $\mathcal{F}_q$ to the inertial frame $\mathcal{F}_e$, which is given by

$$\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{Q}) = (\eta_0^2 - \boldsymbol{\eta}^T \boldsymbol{\eta}) \boldsymbol{I}_3 + 2 \boldsymbol{\eta} \boldsymbol{\eta}^T + 2 \eta_0 S(\boldsymbol{\eta})$$

T_m and $\boldsymbol{\tau}$ are respectively the main rotor thrust and the torque of helicopter. $\boldsymbol{d}_t$ and $\boldsymbol{d}_\tau$ include unmodeled dynamics and external disturbances.

C. Image Feature and Visual Dynamics

Consider a visual point P on the deck of ship, which can be seen in Fig.1, its coordinate defined in the virtual camera frame $\mathcal{F}_v$ and the camera frame $\mathcal{F}_c$ are ${}^v\boldsymbol{p} = [{}^vx, {}^vy, {}^vz]$ and ${}^c\boldsymbol{p} = [{}^cx, {}^cy, {}^cz]$, respectively. Then, its perspective projections on the virtual image plane and the actual image plane are denoted as $({}^vu, {}^vn)$ and $({}^cu, {}^cn)$, where

$$\begin{cases} {}^cu = \lambda \frac{{}^cx}{{}^cz} \\ {}^cn = \lambda \frac{{}^cy}{{}^cz} \end{cases} \qquad \begin{cases} {}^vu = \lambda \frac{{}^vx}{{}^vz} \\ {}^vn = \lambda \frac{{}^vy}{{}^vz} \end{cases}$$

where λ is the focal length of camera. Then, their derivatives are given by

$$\begin{bmatrix} {}^v\dot{u} \\ {}^v\dot{n} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{{}^vz} & 0 & \frac{{}^vu}{{}^vz} \\ 0 & -\frac{\lambda}{{}^vz} & \frac{{}^vn}{{}^vz} \end{bmatrix} ({}^v\boldsymbol{V} - {}^v\boldsymbol{V}_t) \tag{12}$$

$$\begin{aligned}\begin{bmatrix} {}^c\dot{u} \\ {}^c\dot{n} \end{bmatrix} &= \begin{bmatrix} -\frac{\lambda}{{}^cz} & 0 & \frac{{}^cu}{{}^cz} \\ 0 & -\frac{\lambda}{{}^cz} & \frac{{}^cn}{{}^cz} \end{bmatrix} \boldsymbol{R}^T({}^v\boldsymbol{V} - {}^v\boldsymbol{V}_t) \\ &+ \begin{bmatrix} \frac{{}^cu{}^cn}{\lambda} & -\lambda - \frac{{}^cu^2}{\lambda} & {}^cn \\ \lambda + \frac{{}^cn^2}{\lambda} & -\frac{{}^cu{}^cn}{\lambda} & -{}^cu \end{bmatrix} \boldsymbol{\Omega} \end{aligned} \tag{13}$$

where ${}^v\boldsymbol{V}$ and ${}^v\boldsymbol{V}_t$ are the translational velocities of helicopter and ship in $\mathcal{F}_v$, respectively.

In term of the visual target's perspective projection on the virtual image plane, the following image features are extracted to describe the translational motion of helicopter.

$$q_x = q_z\frac{{}^vu_g}{\lambda}, \qquad q_y = q_z\frac{{}^vn_g}{\lambda}, \qquad q_z = z^*\sqrt{\frac{a^*}{a}} \tag{14}$$

where ${}^vu_g = \frac{1}{N}\sum_{k=1}^N {}^vu_k$ and ${}^vn_g = \frac{1}{N}\sum_{k=1}^N {}^vn_k$ stand for the barycenter of visual target in the virtual image plane, $({}^vu_k, {}^vn_k)$ is the coordinate of the kth visual point, $a = {}^v\mu_{20} + {}^v\mu_{02}$, ${}^v\mu_{ij} = \sum_{k=1}^N({}^vu_k - {}^vu_g)^i({}^vn_k - {}^vn_g)^j$ is the centered moment, a^* is the desired value of a with the property of $z^*\sqrt{a^*} = z\sqrt{a}$, and z^* is the desired vertical position.

Define an image feature vector as $\boldsymbol{q} = [q_x, q_y, q_z]^T$, then the error dynamic model of the image feature is considered as

$$\begin{cases} \dot{\bar{\boldsymbol{q}}} = -\boldsymbol{V}_a \\ \dot{\boldsymbol{V}}_a = -\frac{T_m}{m}\boldsymbol{R}_c\boldsymbol{e}_3 + g\boldsymbol{e}_3 - \boldsymbol{D} \end{cases} \tag{15}$$

perturbed by $\bar{\boldsymbol{R}} = \frac{T_m}{m}(\boldsymbol{R}_c - \boldsymbol{R})\boldsymbol{e}_3$, where $\bar{\boldsymbol{q}} = \boldsymbol{q} - \boldsymbol{q}_d$, $\boldsymbol{V}_a = {}^v\boldsymbol{V} - {}^v\boldsymbol{V}_t$, $\boldsymbol{D} = {}^v\dot{\boldsymbol{V}}_t - \boldsymbol{d}_t$, $\boldsymbol{R}_c = \boldsymbol{R}(\boldsymbol{Q}_c)$, $\boldsymbol{Q}_c$ is the desired attitude, and $\boldsymbol{q}_d$ is the desired image feature.

***Assumption** 1:* The linear acceleration of ship ${}^v\dot{\boldsymbol{V}}_t$, the translational disturbances $\boldsymbol{d}_t$, and the rotational disturbances $\boldsymbol{d}_\tau$ are all bounded, and $\langle \boldsymbol{d}_\tau \rangle \leq \bar{\boldsymbol{d}}_\tau$, where $\bar{\boldsymbol{d}}_\tau \in \mathbb{R}^3$ are unknown positive constant vectors.

D. Control Objective

The control objective is to devise a control thrust T_m and a control torque $\boldsymbol{\tau}$ without linear velocity measurements, which can converge the image feature vector $\boldsymbol{q}$ to the desired value $\boldsymbol{q}_d = [0, 0, z^*]^T$ satisfying the visibility constraint, so that the ship landing of the helicopter can be accomplished safely and reliably.

***Remark** 1:* It's worthwhile to noting that z^* is selected as the vertical distance between the camera and the bottom bracket of the helicopter.

IV. IBVS Shipboard Landing Design

The IBVS shipboard landing procedure is summarized as follows and its block diagram is displayed in Fig.2.

1. The pitch and roll motion of ship are forecast online. When it is predicted that they will remain small values during a future period, the landing phase will be switched on.

2. Based on the hierarchical control strategy, an output-feedback controller is devised with a novel sliding mode control technique. In the translational motion loop, a control command $\boldsymbol{U} = \frac{T_m}{m}\boldsymbol{R}_c\boldsymbol{e}_3$ satisfying the non-singular condition is designed to track the desired image feature (equivalently track the desired relative position). In the rotational motion loop, a control torque $\boldsymbol{\tau}$ is devised to track the desired attitude, which is extracted from $\boldsymbol{U}$.

3. Consider the visibility constraint, the command control $\boldsymbol{U}$ is adjusted by a control barrier function and a quadratic programming.

***Assumption** 2:* There exists a period during which the pitch and roll of ship are both small. In this period, the target plane can be considered as parallel with the virtual camera plane in the controller design.

A. Ship Motion Prediction

Generally, the AR model is established to estimate the current motion information by a linear combination of the known past motion information [39], shown as follows:

$$y(t) = \sum_{i=1}^{k} a_i y(t-i) + y_e(t) \tag{16}$$

where k is the order of model, a_i is the coefficient of model, and $y_e(t)$ is the estimation error.

Define a vector $\boldsymbol{y} = [y(k+1), y(k+2), \cdots, y(N)]^T$ with N the sampled amount of the past motion information, then a vector-form AR prediction model can be expressed as

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{a}_c + \boldsymbol{y}_e \tag{17}$$

where $\boldsymbol{a}_c = [a_1, a_2, \cdots, a_k]^T$, $\boldsymbol{y}_e = [y_e(k+1), y_e(k+2), \cdots, y_e(N)]^T$, $\boldsymbol{B} = \begin{bmatrix} y(k) & y(k-1) & \cdots & y(1) \\ y(k+1) & y(k) & \cdots & y(2) \\ \cdots & \cdots & \cdots & \cdots \\ y(N-1) & y(N-2) & \cdots & y(N-k) \end{bmatrix}$.

In the sequel, the coefficient vector $\boldsymbol{a}_c$ can be estimated as $\hat{\boldsymbol{a}}_c = \left[\boldsymbol{B}^T\boldsymbol{B}\right]^{-1}\boldsymbol{B}^T\boldsymbol{y}$ by virtue of minimizing $J = (\boldsymbol{y} - \boldsymbol{B}\hat{\boldsymbol{a}}_c)^T(\boldsymbol{y} - \boldsymbol{B}\hat{\boldsymbol{a}}_c)$. The optimal order k is specified according to the following Akaike Information Criterion (AIC) method:

$$AIC(k) = \ln\left(\frac{(\boldsymbol{y} - \boldsymbol{B}\hat{\boldsymbol{a}}_c)^T(\boldsymbol{y} - \boldsymbol{B}\hat{\boldsymbol{a}}_c)}{N} + \frac{2k}{N}\right) \tag{18}$$

$$k^* = \arg\min_k AIC(k) \tag{19}$$

where $k = 1, 2, 3...M$. To avoid distortion of the AR model, the max order is normally set as $N/3$.

Therefore, the subsequent motion is predicted by

$$y(N+c) = \begin{cases} \sum_{i=1}^{k^*} \hat{a}_i y(N+c-i) & c = 1 \\ \sum_{i=1}^{c} \hat{a}_i y(N+c-i) + \sum_{i=c+1}^{k^*} \hat{a}_i \hat{y}(N+c-i) & 1 < c \leq k^* \\ \sum_{i=1}^{k^*} \hat{a}_i \hat{y}(N+c-i) & c > k^* \end{cases} \tag{20}$$

where c is the prediction step.

The roll motion and the pitch motion of ship can be regarded as the combination of sinusoidal functions [40]. Therefore, the AR model can be adopted to forecast the ship motion by simply replacing y in (17)-(20) with ϕ_s and θ_s. The specific prediction process is illustrated in Fig.3.

***Remark** 2:* The roll and the pitch of ship are available with the measurements of camera and IMU. The relative attitude between the helicopter and the ship can be retrieved by decomposing the homography matrix [41], while the attitude of helicopter can be directly measured by the IMU. Therefore, it is simple to obtain the attitude of ship.

***Remark** 3:* If there exists an time interval $[t_1, t_1 + \Delta t]$ satisfying the following condition

$$(\phi_s, \theta_s) \in \left\{ \frac{\sum_{t_1}^{t_1+\Delta t} |\phi_s|}{\Delta t} \leq \bar{\phi}, \frac{\sum_{t_1}^{t_1+\Delta t} |\theta_s|}{\Delta t} \leq \bar{\theta} \right\} \quad (21)$$

where $\bar{\phi}$ and $\bar{\theta}$ are small constants, the landing phase should be switched on at t_1 and completed in the interval $[t_1, t_1+\Delta t]$.

B. Output-feedback Controller

1) Translational motion Control: Define an error variable $\boldsymbol{\delta}_1 = \boldsymbol{q} - \boldsymbol{q}_d - \boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_2$ with two auxiliary states $\boldsymbol{\zeta}_1$ and $\boldsymbol{\zeta}_2$, and a novel sliding mode surface $\boldsymbol{s}_1 = \dot{\boldsymbol{\delta}}_1 + \alpha \boldsymbol{\delta}_1$ with a positive constant $\alpha > 0$. Then the derivative of $\boldsymbol{s}_1$ is

$$\dot{\boldsymbol{s}}_1 = \frac{T_m}{m} \boldsymbol{R}_c \boldsymbol{e}_3 - g\boldsymbol{e}_3 + \boldsymbol{D} - \ddot{\boldsymbol{\zeta}}_1 - \ddot{\boldsymbol{\zeta}}_2 + \alpha(\boldsymbol{s}_1 - \alpha \boldsymbol{\delta}_1) \quad (22)$$

Design an auxiliary system

$$\ddot{\boldsymbol{\zeta}}_1 = -k_\alpha \tanh(k_m \boldsymbol{\zeta}_1 + k_n \dot{\boldsymbol{\zeta}}_1) - k_\beta \tanh(k_n \dot{\boldsymbol{\zeta}}_1) + l_1 \boldsymbol{\zeta}_2 + l_2 \dot{\boldsymbol{\zeta}}_2 \quad (23)$$

where k_α,k_β, l_1, and l_2 are positive constants; $\boldsymbol{\zeta}_2$ and $\dot{\boldsymbol{\zeta}}_2$ are outputs of the following another auxiliary system.

$$\begin{cases} \dot{\boldsymbol{\zeta}}_2 = \boldsymbol{\rho} + k_\gamma \boldsymbol{\delta}_1 \\ \dot{\boldsymbol{\rho}} = k_\gamma \alpha \boldsymbol{\delta}_1 - l_1 \boldsymbol{\zeta}_2 - l_2(\boldsymbol{\rho} + k_\gamma \boldsymbol{\delta}_1) - \alpha^2 \boldsymbol{\delta}_1 \end{cases} \quad (24)$$

where $\boldsymbol{\rho}(0) = -k_\gamma \boldsymbol{\delta}_1(0)$, $k_\gamma > 0$. Then, we have

$$\ddot{\boldsymbol{\zeta}}_2 = k_\gamma \boldsymbol{s}_1 - l_1 \boldsymbol{\zeta}_2 - l_2 \dot{\boldsymbol{\zeta}}_2 - \alpha^2 \boldsymbol{\delta}_1 \quad (25)$$

Therefore, (22) can be rewritten as

$$\begin{aligned} \dot{\boldsymbol{s}}_1 =& \frac{T_m}{m} \boldsymbol{R}_c \boldsymbol{e}_3 - g\boldsymbol{e}_3 + \boldsymbol{D} + k_\alpha \tanh(k_m \boldsymbol{\zeta}_1 + k_n \dot{\boldsymbol{\zeta}}_1) \\ &+ k_\beta \tanh(k_n \dot{\boldsymbol{\zeta}}_1) - k_\gamma \boldsymbol{s}_1 + \alpha \boldsymbol{s}_1 \end{aligned} \quad (26)$$

As claimed by subsection B in Section II, the uncertain term $\boldsymbol{D}$ can be approximated by

$$\boldsymbol{D} = \boldsymbol{W} \boldsymbol{\Phi}(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2) + \boldsymbol{\epsilon} \quad (27)$$

where $\boldsymbol{W}$ is an adjustable weight matrix with $\text{tr}\{\boldsymbol{W}^T \boldsymbol{W}\} \leq W_M$.

Design the control input

$$\begin{aligned} \boldsymbol{U} =& \frac{T_m}{m} \boldsymbol{R}_c \boldsymbol{e}_3 = g\boldsymbol{e}_3 - k_\alpha \tanh(k_m \boldsymbol{\zeta}_1 + k_n \dot{\boldsymbol{\zeta}}_1) \\ &- k_\beta \tanh(k_n \dot{\boldsymbol{\zeta}}_1) - \hat{\boldsymbol{D}} \end{aligned} \quad (28)$$

where $\hat{\boldsymbol{D}}$ is the estimation of $\boldsymbol{D}$ defined by

$$\hat{\boldsymbol{D}} = \boldsymbol{Sat}_{\boldsymbol{\Theta}_D}(\hat{\boldsymbol{W}} \boldsymbol{\Phi}) \quad (29)$$

where $\hat{\boldsymbol{W}}$ is the estimation of NN weight matrix, whose update law is given by

$$\begin{cases} \dot{\hat{\boldsymbol{W}}} = \frac{\chi}{k_\gamma} \int_0^t \left[(l_1 \boldsymbol{\zeta}_2 + l_2 \dot{\boldsymbol{\zeta}}_2 + \alpha^2 \boldsymbol{\delta}_1) \boldsymbol{\Phi}^T(\nu) - \dot{\boldsymbol{\zeta}}_2 \dot{\boldsymbol{\Phi}}^T(\nu) \right] d\nu \\ \qquad + \frac{\chi}{k_\gamma} \dot{\boldsymbol{\zeta}}_2 \boldsymbol{\Phi}^T - \sigma \chi \int_0^t \hat{\boldsymbol{W}}(\nu) d\nu \quad if \langle \hat{\boldsymbol{D}} \rangle < \Theta_D \\ \dot{\hat{\boldsymbol{W}}} = 0 \quad if \langle \hat{\boldsymbol{D}} \rangle \geq \Theta_D \end{cases} \quad (30)$$

where χ and σ are positive constants.

Due to $\|\boldsymbol{R}_c \boldsymbol{e}_3\| = 1$, the rotor thrust T_m is derived as

$$T_m = m \|\boldsymbol{U}\| \quad (31)$$

***Remark** 4:* The order of Chebyshev polynomial basis function is set as 3 in this paper, that is, $\boldsymbol{\Phi}(\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2)$ is obtained by

$$\begin{aligned} \boldsymbol{\Phi} = \Big[& 1, \boldsymbol{\zeta}_1^T, (2\boldsymbol{\zeta}_1^2 - 1)^T, (4\boldsymbol{\zeta}_1^3 - 3\boldsymbol{\zeta}_1)^T, \boldsymbol{\zeta}_2{}^T, (2\boldsymbol{\zeta}_2^2 - 1)^T, \\ & (4\boldsymbol{\zeta}_2^3 - 3\boldsymbol{\zeta}_2)^T \Big]^T \end{aligned} \quad (32)$$

$\dot{\boldsymbol{\Phi}}$ in (30) can be computed as follows:

$$\begin{aligned} \dot{\boldsymbol{\Phi}} = \Big[& 0, \dot{\boldsymbol{\zeta}}_1^T, (4\boldsymbol{\zeta}_1 \circ \dot{\boldsymbol{\zeta}}_1)^T, (12\boldsymbol{\zeta}_1^2 \circ \dot{\boldsymbol{\zeta}}_1 - 3\dot{\boldsymbol{\zeta}}_1)^T, \dot{\boldsymbol{\zeta}}_2^T, \\ & (4\boldsymbol{\zeta}_2 \circ \dot{\boldsymbol{\zeta}}_2)^T, (12\boldsymbol{\zeta}_2^2 \circ \dot{\boldsymbol{\zeta}}_2 - 3\dot{\boldsymbol{\zeta}}_2)^T \Big]^T \end{aligned} \quad (33)$$

where $\dot{\boldsymbol{\zeta}}_1$ and $\dot{\boldsymbol{\zeta}}_2$ are from the auxiliary systems (23) and (24).

***Theorem** 1:* Consider the translational subsystem (15) under Assumption 1, if there exists $\bar{d}_\zeta > 0$, and the controller parameters are chosen by

$$\begin{aligned} & \frac{k_m}{k_n^2} > 2\bar{d}_\zeta \\ & k_\beta > (2\bar{d}_\zeta \frac{k_m}{k_n^2}) / (\frac{k_m}{k_n^2} - 2\bar{d}_\zeta) \\ & \sqrt{k_\beta \frac{k_m}{k_n^2}} \leq k_\alpha \leq \frac{1}{2}(k_\beta + \frac{k_m}{k_n^2}) \\ & k_\gamma - \alpha > \frac{1}{2} \\ & k_\alpha + k_\beta + \Theta_{D3} < g \end{aligned} \quad (34)$$

then the designed control input (28) with the adaptive law (30) can guarantee that $\boldsymbol{U}$ satisfies $U_z > 0$ and the tracking error $\bar{\boldsymbol{q}}$ is uniformly ultimately bounded.

proof: Consider a Lyapunov function candidate as

$$V_1 = \frac{1}{2} \boldsymbol{s}_1^T \boldsymbol{s}_1 + \frac{1}{2\chi} \text{tr}\{\tilde{\boldsymbol{W}}^T \tilde{\boldsymbol{W}}\} \quad (35)$$

where $\tilde{\boldsymbol{W}} = \boldsymbol{W} - \hat{\boldsymbol{W}}$ is the estimation error of CNN weight matrix. By substituting (28) into (26), the derivation of (35) can be obtained as

$$\dot{V}_1 = -(k_\gamma - \alpha) \boldsymbol{s}_1^T \boldsymbol{s}_1 + \boldsymbol{s}_1^T (\boldsymbol{D} - \hat{\boldsymbol{D}}) - \frac{1}{\chi} \text{tr}\{\tilde{\boldsymbol{W}}^T \dot{\hat{\boldsymbol{W}}}\} \quad (36)$$

According to (30), if $\langle \hat{\boldsymbol{D}} \rangle < \Theta_D$, then

$$\begin{aligned}\dot{V}_1 =& -(k_\gamma - \alpha)\boldsymbol{s}_1^T\boldsymbol{s}_1 + \boldsymbol{s}_1^T(\boldsymbol{D} - \hat{\boldsymbol{D}}) - \text{tr}\left\{\tilde{\boldsymbol{W}}^T\left(\frac{1}{k_r}(l_1\boldsymbol{\zeta}_2 \right.\right.\\ &\left.\left. + l_2\dot{\boldsymbol{\zeta}}_2 + \alpha^2\boldsymbol{\delta}_1\right)\boldsymbol{\Phi}^T - \frac{1}{k_r}\dot{\boldsymbol{\zeta}}_2\dot{\boldsymbol{\Phi}}^T + \frac{1}{k_r}(\dot{\boldsymbol{\zeta}}_2\dot{\boldsymbol{\Phi}}^T + \ddot{\boldsymbol{\zeta}}_2\boldsymbol{\Phi}^T)\right.\\ &\left. - \sigma\hat{\boldsymbol{W}}\right)\Big\}\\ =& -(k_\gamma - \alpha)\boldsymbol{s}_1^T\boldsymbol{s}_1 + \boldsymbol{s}_1^T\epsilon + \sigma\text{tr}\{\tilde{\boldsymbol{W}}^T(\boldsymbol{W} - \tilde{\boldsymbol{W}})\}\\ \leq& -(k_\gamma - \alpha - \frac{1}{2})\boldsymbol{s}_1^T\boldsymbol{s}_1 + \sigma\text{tr}\left\{\frac{1}{2}\tilde{\boldsymbol{W}}^T\tilde{\boldsymbol{W}} + \frac{1}{2}\boldsymbol{W}^T\boldsymbol{W}\right\}\\ &- \sigma\text{tr}\{\boldsymbol{W}^T\boldsymbol{W}\} + \frac{1}{2}\epsilon^2\\ \leq& -(k_\gamma - \alpha - \frac{1}{2})\boldsymbol{s}_1^T\boldsymbol{s}_1 - \frac{1}{2}\sigma\text{tr}\left\{\tilde{\boldsymbol{W}}^T\tilde{\boldsymbol{W}}\right\} + \frac{1}{2}\sigma W_M + \frac{1}{2}\epsilon^2\\ \leq& -b_pV_1 + c_{p1}\end{aligned} \tag{37}$$

where $b_p = \min\{2(k_\gamma - \alpha) - 1, \sigma\chi\}$, and $c_{p1} = \frac{1}{2}\sigma W_M + \frac{1}{2}\epsilon^2$. Thus, $\boldsymbol{s}$ is bounded and ultimately converges to the set

$$\mathcal{Z}_{s1} = \left\{\boldsymbol{s}_1 \in \mathbb{R}^3 | \|\boldsymbol{s}_1\| < \sqrt{\frac{c_{p1}}{b_p}}\right\} \tag{38}$$

If $\langle \hat{\boldsymbol{D}} \rangle \geq \Theta_D$, then $\hat{\boldsymbol{D}} = \boldsymbol{\Theta}_D$, and (36) is simplified as

$$\dot{V}_1 = -(k_r - \alpha)\boldsymbol{s}_1^T\boldsymbol{s}_1 + \boldsymbol{s}_1^T(\boldsymbol{D} - \boldsymbol{\Theta}_D) \tag{39}$$

With the Assumption 1, $\|\boldsymbol{D} - \boldsymbol{\Theta}_D\|$ is also bounded by an constant $\bar{D}_e$. Then, we have

$$\dot{V}_1 = -(k_\gamma - \alpha - \frac{1}{2})\boldsymbol{s}_1^T\boldsymbol{s}_1 + c_{p2} \tag{40}$$

where $c_{p2} = \frac{1}{2}\bar{D_e}^2$. Thus, $\boldsymbol{s}_1$ is bounded and ultimately converges to the set

$$\mathcal{Z}_{s2} = \left\{\boldsymbol{s}_1 \in \mathbb{R}^3 | \|\boldsymbol{s}_1\| < \sqrt{\frac{c_{p2}}{b_p}}\right\} \tag{41}$$

Overall, $\boldsymbol{s}_1$ is ultimately converges to the set $\mathcal{Z}_s = \mathcal{Z}_{s1} \vee \mathcal{Z}_{s2}$.

Define $\boldsymbol{\zeta}_2^* = \left[\boldsymbol{\zeta}_2^T, \dot{\boldsymbol{\zeta}}_2^T\right]^T$. From (25), we can obtain

$$\begin{aligned}\dot{\boldsymbol{\zeta}}_2^* =& -\begin{bmatrix}\boldsymbol{0} & -\boldsymbol{I}_3\\ l_1\boldsymbol{I}_3 & l_2\boldsymbol{I}_3\end{bmatrix}\boldsymbol{\zeta}_2^* + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{d}_s\end{bmatrix}\\ =& -\boldsymbol{A}\boldsymbol{\zeta}_2^* + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{d}_s\end{bmatrix}\end{aligned} \tag{42}$$

where $\boldsymbol{d}_s = k_\gamma\boldsymbol{s}_1 - \alpha^2\boldsymbol{\delta}_1$, satisfying $\|\boldsymbol{d}_s\| \leq \bar{d}_s$, when $t > \bar{t}(\bar{t} > 0)$. It is simple to certify that $\boldsymbol{\zeta}_2^*$ is bounded and converges to a small set

$$\mathcal{Z}_{\zeta 2} = \left\{\left[\boldsymbol{\zeta}_2^T, \dot{\boldsymbol{\zeta}}_2^T\right]^T \in \mathbb{R}^6 | \|[\boldsymbol{\zeta}_2^T, \dot{\boldsymbol{\zeta}}_2^T\| < \sqrt{\frac{\bar{d}_s^2}{\lambda_{\min}(\boldsymbol{A})}}\right\} \tag{43}$$

.

Then, the convergence set of $l_1\boldsymbol{\zeta}_2 + l_2\dot{\boldsymbol{\zeta}}_2$ is

$$\begin{aligned}\|l_1\boldsymbol{\zeta}_2 + l_2\dot{\boldsymbol{\zeta}}_2\| &\leq \max\{l_1, l_2\}\|\boldsymbol{\zeta}_2 + \dot{\boldsymbol{\zeta}}_2\|\\ &\leq \sqrt{2}\max\{l_1, l_2\}\|[\boldsymbol{\zeta}_2^T, \dot{\boldsymbol{\zeta}}_2^T\|\\ &\leq \max\{l_1, l_2\}\sqrt{\frac{2\bar{d}_s^2}{\lambda_{\min}(\boldsymbol{A})}}\\ &= \bar{d}_\zeta\end{aligned} \tag{44}$$

According to Lemma 1 and (23), $\boldsymbol{\zeta}_1$ and $\dot{\boldsymbol{\zeta}}_1$ ultimately converge to the set

$$\mathcal{Z}_{\zeta 1} = \left\{[\boldsymbol{\zeta}_1^T, \dot{\boldsymbol{\zeta}}_1^T]^T \in \mathbb{R}^6 | \|[k_m\boldsymbol{\zeta}_1^T + k_n\dot{\boldsymbol{\zeta}}_1^T, k_n\dot{\boldsymbol{\zeta}}_1^T]^T\| < \bar{\zeta}_1\right\} \tag{45}$$

where $\bar{\zeta}$ satisfies $\frac{k_\beta + (k_m/k_n^2)}{k_\beta k_m/{k_n}^2 d_\zeta} < \frac{\tanh^2(\bar{\zeta})}{\bar{\zeta}} < \frac{1}{2}$.

Therefore, the tracking error $\bar{\boldsymbol{q}}$ is ultimately bounded.

2) Command attitude extractions: Obviously, the designed controller satisfies $U_z \neq 0$, then the desired attitude of the inner loop $\boldsymbol{Q}_c = [\eta_{0c}, \boldsymbol{\eta}_c^T]^T$ can be calculated as

$$\begin{cases}\eta_{0c} = \sqrt{\frac{1}{2} + \frac{U_z}{2\|\boldsymbol{U}\|}}\\ \boldsymbol{\eta}_c = \frac{1}{2\|\boldsymbol{U}\|\eta_{0c}}[-U_y, U_x, 0]^T\end{cases} \tag{46}$$

According to (11), the desired angular velocity $\boldsymbol{\Omega}_c$ and its derivative $\dot{\boldsymbol{\Omega}}_c$ are obtained as

$$\boldsymbol{\Omega}_c = 4\boldsymbol{G}_c^T\dot{\boldsymbol{Q}}_c \tag{47}$$

$$\dot{\boldsymbol{\Omega}}_c = 4\boldsymbol{G}_c^T\left(\ddot{\boldsymbol{Q}}_c - \dot{\boldsymbol{G}}_c\boldsymbol{\Omega}_c\right) \tag{48}$$

where $\boldsymbol{G}_c = \boldsymbol{G}(\boldsymbol{Q}_c)$ with $\boldsymbol{G}_c^T\boldsymbol{G}_c = \frac{1}{4}\boldsymbol{I}_3$, and $\dot{\boldsymbol{G}}_c = \boldsymbol{G}(\dot{\boldsymbol{Q}}_c)$. For the sake of avoiding the complicated computation, a second-order command filter is employed to acquire $\dot{\boldsymbol{Q}}_c$ and $\ddot{\boldsymbol{Q}}_c$.

$$\begin{cases}\dot{\boldsymbol{X}}_1 = \boldsymbol{X}_2\\ \dot{\boldsymbol{X}}_2 = -2\lambda_n\omega_n\boldsymbol{X}_2 - \omega_n^2(\boldsymbol{X}_1 - \boldsymbol{Q}_c)\end{cases} \tag{49}$$

where λ_n and ω_n are the damping ratio and the damping frequency, respectively.

3) Attitude control: Define the attitude tracking error variable $\boldsymbol{Q}_e = [\eta_{0e}, \boldsymbol{\eta}_e^T]^T = \boldsymbol{Q}_c^{-1} \odot \boldsymbol{Q}$, with $\boldsymbol{Q}_c^{-1} = [\eta_{0c}, -\boldsymbol{\eta}_c^T]^T$. The kinematic and dynamic model of the attitude error can be obtained as

$$\begin{cases}\dot{\boldsymbol{Q}}_e = \boldsymbol{G}_e\boldsymbol{\Omega}_e\\ \boldsymbol{J}\dot{\boldsymbol{\Omega}}_e = -S(\boldsymbol{\Omega})\boldsymbol{J}\boldsymbol{\Omega} + \boldsymbol{\tau} + \boldsymbol{d}_\tau + \boldsymbol{J}S(\boldsymbol{\Omega_e})\boldsymbol{R}_e\boldsymbol{\Omega}_c - \boldsymbol{J}\boldsymbol{R}_e\dot{\boldsymbol{\Omega}}_c\end{cases} \tag{50}$$

where $\boldsymbol{G}_e = \frac{1}{2}\begin{bmatrix}-\boldsymbol{\eta}_e^T\\ \eta_{0e}\boldsymbol{I}_3 + S(\boldsymbol{\eta}_e)\end{bmatrix}$, $\boldsymbol{\Omega}_e = \boldsymbol{\Omega} - \boldsymbol{R}_e\boldsymbol{\Omega}_c$, and $\boldsymbol{R}_e = \boldsymbol{R}^T\boldsymbol{R}_c$ with the property $\dot{\boldsymbol{R}}_e = -\boldsymbol{S}(\boldsymbol{\Omega}_e)\boldsymbol{R}_e$.

Define a sliding mode surface $\boldsymbol{s}_2 = \boldsymbol{\eta}_e + \beta\boldsymbol{\Omega}_e$ with a positive constant β. The derivative of $\boldsymbol{s}_2$ is

$$\begin{aligned}\boldsymbol{J}\dot{\boldsymbol{s}}_2 =& \frac{1}{2}\left(\eta_{0e}\boldsymbol{I}_3 + S(\boldsymbol{\eta}_e)\right)\boldsymbol{\Omega}_e + \beta\left(-S(\boldsymbol{\Omega})\boldsymbol{J}\boldsymbol{\Omega} + \boldsymbol{\tau}\right.\\ &\left. + \boldsymbol{d}_\tau + \boldsymbol{J}S(\boldsymbol{\Omega_e})\boldsymbol{R}_e\boldsymbol{\Omega}_c - \boldsymbol{J}\boldsymbol{R}_e\dot{\boldsymbol{\Omega}}_c\right)\end{aligned} \tag{51}$$

Design the control input as

$$\begin{aligned}\boldsymbol{\tau} =& -\frac{1}{\beta}k_s\boldsymbol{s}_2 + S(\boldsymbol{\Omega})\boldsymbol{J}\boldsymbol{\Omega} - \boldsymbol{J}S(\boldsymbol{\Omega_e})\boldsymbol{R}_e\boldsymbol{\Omega}_c + \boldsymbol{J}\boldsymbol{R}_e\dot{\boldsymbol{\Omega}}_c\\ &- \frac{1}{2\beta}\boldsymbol{J}\left(\eta_{0e}\boldsymbol{I}_3 + S(\boldsymbol{\eta}_e)\right)\boldsymbol{\Omega}_e - \hat{\boldsymbol{d}}_\tau \circ \tanh\left(\frac{\boldsymbol{s}_2}{\varpi}\right)\end{aligned} \tag{52}$$

with the adaptive law

$$\dot{\hat{\boldsymbol{d}}}_\tau = \gamma\beta\left(\boldsymbol{s}_2 \circ \tanh\left(\frac{\boldsymbol{s}_2}{\varpi}\right) - k_d\hat{\boldsymbol{d}}_\tau\right) \tag{53}$$

where k_s, γ, and k_d are positive constants.

***Theorem** 2:* Consider the rotational subsystem in (11) under Assumption 1, if the controller parameter is chosen by $k_s > \frac{1}{4\beta}$, the designed control input (52) with adaptive law (53) can ensure that the tracking error $\boldsymbol{\eta}_e$ is uniformly ultimately bounded.

proof: Consider a Lyapunov function candidate as

$$\begin{aligned} V_2 &= \frac{1}{2}\boldsymbol{s}_2^T\boldsymbol{J}\boldsymbol{s}_2 + \frac{1}{2}\boldsymbol{\eta}_e^T\boldsymbol{\eta}_e + \frac{1}{2}(1-\eta_{0e})^2 + \frac{1}{2\gamma}\tilde{\boldsymbol{d}_\tau}^T\tilde{\boldsymbol{d}_\tau} \\ &= \frac{1}{2}\boldsymbol{s}_2^T\boldsymbol{J}\boldsymbol{s}_2 + (1-\eta_{0e}) + \frac{1}{2\gamma}\tilde{\boldsymbol{d}_\tau}^T\tilde{\boldsymbol{d}_\tau} \end{aligned} \tag{54}$$

where $\tilde{\boldsymbol{d}_\tau} = \bar{\boldsymbol{d}_\tau} - \hat{\boldsymbol{d}_\tau}$ is the approximation error. Differentiate V_2, we obtain

$$\begin{aligned} \dot{V}_2 =& \boldsymbol{s}_2^T\boldsymbol{J}\dot{\boldsymbol{s}}_2 + \frac{1}{2\beta}\boldsymbol{\eta}_e^T(\boldsymbol{s}_2 - \boldsymbol{\eta}_e) - \frac{1}{\gamma}\tilde{\boldsymbol{d}_\tau}^T\dot{\tilde{\boldsymbol{d}}}_\tau \\ \leq& \boldsymbol{s}_2^T\boldsymbol{J}\dot{\boldsymbol{s}}_2 - \frac{1}{4\beta}\boldsymbol{\eta}_e^T\boldsymbol{\eta}_e + \frac{1}{4\beta}\boldsymbol{s}_2^T\boldsymbol{s}_2 - \frac{1}{\gamma}\tilde{\boldsymbol{d}_\tau}^T\dot{\tilde{\boldsymbol{d}}}_\tau \end{aligned} \tag{55}$$

Substituting (51) and (52) into (55) yields

$$\begin{aligned} \dot{V}_2 \leq& -(k_s - \frac{1}{4\beta})\boldsymbol{s}_2^T\boldsymbol{s}_2 + \beta\boldsymbol{s}_2^T\left(\boldsymbol{d}_\tau - \hat{\boldsymbol{d}}_\tau \circ \tanh\left(\frac{\boldsymbol{s}_2}{\epsilon}\right)\right) \\ & - \frac{1}{4\beta}\boldsymbol{\eta}_e^T\boldsymbol{\eta}_e - \beta\tilde{\boldsymbol{d}_\tau}^T\left(\boldsymbol{s}_2 \circ \tanh\left(\frac{\boldsymbol{s}_2}{\epsilon}\right) - k_d\hat{\boldsymbol{d}_\tau}\right) \end{aligned} \tag{56}$$

As stated by Lemma 2, the following inequality holds:

$$\boldsymbol{s}_2^T\boldsymbol{d}_\tau \leq \boldsymbol{s}_2^T\bar{\boldsymbol{d}_\tau} \leq \boldsymbol{s}_2^T\bar{\boldsymbol{d}_\tau} \circ \tanh(\frac{\boldsymbol{s}_2}{\epsilon}) + \sigma_d$$

where $\sigma_d = \sum_{i=1}^{3}\epsilon\kappa\bar{\boldsymbol{d}_{\tau i}}$. Then, with the adaptive law (53), (56) becomes

$$\begin{aligned} \dot{V}_2 \leq& -(k_s - \frac{1}{4\beta})\boldsymbol{s}_2^T\boldsymbol{s}_2 - \frac{1}{4\beta}\boldsymbol{\eta}_e^T\boldsymbol{\eta}_e - \frac{1}{2}\beta k_d\tilde{\boldsymbol{d}_\tau}^T\tilde{\boldsymbol{d}_\tau} \\ & + \frac{1}{2}\beta k_d\bar{\boldsymbol{d}_\tau}^T\bar{\boldsymbol{d}_\tau} \\ \leq& -b_a\|\boldsymbol{\Xi}\|^2 + c_a \end{aligned} \tag{57}$$

where $\boldsymbol{\Xi} = [\boldsymbol{s}_2^T, \boldsymbol{\eta}_e^T, \tilde{\boldsymbol{d}_\tau}^T]^T$, $b_a = \min\left\{\left(k_s - \frac{1}{4\beta}\right), \frac{1}{4\beta}, \frac{1}{2}\beta k_d\right\}$, and $c_a = \frac{1}{2}\beta k_d\bar{\boldsymbol{d}_\tau}^T\bar{\boldsymbol{d}_\tau}$. Under these conditions, $\dot{V}_2$ is strictly negative outside the compact set

$$\mathcal{Z}_\Xi = \left\{\boldsymbol{\Xi} \in \mathbb{R}^9 | \|\boldsymbol{\Xi}\| < \sqrt{\frac{c_a}{b_a}}\right\} \tag{58}$$

Thus, $\boldsymbol{\eta}_e$ is bounded and ultimately converges to a small neighbour set of zero.

4) Stability of the whole system:

***Theorem** 3:* For the cascaded error system composed of (15) and (50), if the rotor trust T_m and the torque $\boldsymbol{\tau}$ are designed by (31) and (52) with update laws (30) and (53), then the whole closed loop system is uniformly ultimately bounded.

proof: In the opinion of [42], if the following conditions are achieved simultaneously, then the closed-loop cascaded system is uniformly ultimately bounded.

(a) The translational motion error subsystem (15) is uniformly ultimately bounded;
(b) The rotational motion error subsystem (50) is uniformly ultimately bounded;
(c) The error term $\bar{\boldsymbol{R}} = \frac{T_m}{m}(\boldsymbol{R}_c - \boldsymbol{R})\boldsymbol{e}_3$ satisfies $\|\bar{\boldsymbol{R}}\| \leq w\|\boldsymbol{\eta}_e\|$, with w as a positive constant.

The items (a) and (b) have been certified by Theorem 1 and Theorem 2. As for (c), the following formula holds:

$$\begin{aligned} \|\bar{\boldsymbol{R}}\| &= \|\boldsymbol{R}_c(\boldsymbol{I}_3 - \boldsymbol{R}_e^T)\boldsymbol{e}_3\frac{T_m}{m}\| \\ &= \left((\boldsymbol{R}_c(\boldsymbol{I}_3 - \boldsymbol{R}_e^T)\boldsymbol{e}_3)^T(\boldsymbol{R}_c(\boldsymbol{I}_3 - \boldsymbol{R}_e^T)\boldsymbol{e}_3)\right)^{\frac{1}{2}}\frac{T_m}{m} \\ &= \left(\boldsymbol{e}_3^T\left(\boldsymbol{R}_e + \boldsymbol{R}_e^T - 2\boldsymbol{I}_3\right)\boldsymbol{e}_3\right)^{\frac{1}{2}}\frac{T_m}{m} \\ &= 2\frac{T_m}{m}\sqrt{\eta_{e1}^2 + \eta_{e2}^2} \\ &\leq w\|\boldsymbol{\eta}_e\| \end{aligned} \tag{59}$$

where $w = \max(T_m)/m$, η_{e1} and η_{e2} are elements of $\boldsymbol{\eta}_e$. Therefore, we can deduce that the whole closed loop system is uniformly ultimately bounded.

C. Controller with Visibility Constraint

In order to guarantee a visual point always in the FoV of camera, its projected coordinate on the virtual image plane should be imposed in the following forward invariant set.

$$\mathcal{L} = \left\{({}^v u, {}^v n) | {}^v u^2 + {}^v n^2 \leq \iota^2\right\} \tag{60}$$

where ι is the visible radius of virtual image plane.

***Remark** 5:* To ensure all visual points of visual target in the FoV of camera, ${}^v u$ and ${}^v n$ in (60) should be selected as the coordinate of a point with the max value of ${}^v u^2 + {}^v n^2$ in every control step, namely

$$({}^v u, {}^v n) = \left\{({}^v u_i, {}^v n_i) | \max({}^v u_i^2 + {}^v n_i^2)\right\}$$

where $i = 1, 2, \cdots, r$, r is the number of pixels of the visual target.

Define a zeroing barrier function as

$$H = \dot{h} + k_1 h \tag{61}$$

where $h = \iota^2 - {}^v u^2 - {}^v n^2$. According to Proposition 1, if the constraint $\dot{H} \geq -k_2 H$ is satisfied, then the set (60) is forward invariant. In other words,

$$\ddot{h} + (k_1 + k_2)\dot{h} + k_1 k_2 h \geq 0 \tag{62}$$

should be guaranteed, where k_1 and k_2 are positive constants. For convenience, the superscripts v in ${}^v u$, ${}^v n$ and ${}^v z$ are omitted in the following derivation. According to (12), the derivative and the second-order derivative of h are given by

$$\dot{h} = 2\begin{bmatrix} \frac{\lambda u}{z} & \frac{\lambda n}{z} & -\frac{u^2+n^2}{z} \end{bmatrix}\boldsymbol{V}_a \tag{63}$$

$$\begin{aligned} \ddot{h} =& 2\begin{bmatrix} \lambda\frac{\dot{u}z - u\dot{z}}{z^2} & \lambda\frac{\dot{n}z - n\dot{z}}{z^2} & -\frac{2(u\dot{u}+n\dot{n})z + (u^2+n^2)\dot{z}}{z^2} \end{bmatrix}\boldsymbol{V}_a \\ & + 2\begin{bmatrix} \frac{\lambda u}{z} & \frac{\lambda n}{z} & -\frac{u^2+n^2}{z} \end{bmatrix}\dot{\boldsymbol{V}}_a \end{aligned} \tag{64}$$

where $\boldsymbol{V}_a$ can be estimated via the following velocity observer

$$\begin{cases} \hat{\boldsymbol{V}}_a = \boldsymbol{v}_o - \Lambda_1\tilde{\boldsymbol{q}} \\ \dot{\boldsymbol{v}}_o = -\boldsymbol{U} + g\boldsymbol{e}_3 - \hat{\boldsymbol{D}} - \Lambda_1\Lambda_2\tilde{\boldsymbol{q}} - \tilde{\boldsymbol{q}} \\ \dot{\tilde{\boldsymbol{q}}} = -\hat{\boldsymbol{V}}_a + \Lambda_2\tilde{\boldsymbol{q}} \end{cases} \tag{65}$$

with $\tilde{\boldsymbol{q}} = \boldsymbol{q} - \hat{\boldsymbol{q}}$, $\Lambda_1 > 0$, $\Lambda_2 > 0$, and $\hat{\boldsymbol{D}}$ obtained from (29).

A quadratic programming is employed to minimally adjust the desired control input $\boldsymbol{U}$ in (28), such that $\boldsymbol{U}$ is replaced by the constrained control input

$$\begin{aligned} \boldsymbol{U}_{con}^{*} = \arg\min_{\boldsymbol{U}_{con}} \|\boldsymbol{U}_{con} - \boldsymbol{U}\|^2 \\ \text{s.t.} \qquad \digamma \boldsymbol{U}_{con} \leq \boldsymbol{E} \end{aligned} \tag{66}$$

where

$$\digamma = 2\begin{bmatrix} \frac{\lambda u}{z} & \frac{\lambda n}{z} & -\frac{u^2+n^2}{z} \end{bmatrix}$$

$$\begin{aligned} \boldsymbol{E} =& 2\begin{bmatrix} \lambda\frac{\dot{u}z-u\dot{z}}{z^2} & \lambda\frac{\dot{n}z-n\dot{z}}{z^2} & -\frac{2(u\dot{u}+n\dot{n})z+(u^2+n^2)\dot{z}}{z^2} \end{bmatrix}\hat{\boldsymbol{V}}_a \\ &+2(k_1+k_2)\begin{bmatrix} \frac{\lambda u}{z} & \frac{\lambda n}{z} & -\frac{u^2+n^2}{z} \end{bmatrix}\hat{\boldsymbol{V}}_a + k_1k_2(\iota^2 - u^2 - n^2) \\ &+2\begin{bmatrix} \frac{\lambda u}{z} & \frac{\lambda n}{z} & -\frac{u^2+n^2}{z} \end{bmatrix}(g\boldsymbol{e}_3 - \hat{\boldsymbol{D}}) \end{aligned}$$

***Remark** 6:* It should be mentioned that the projected coordinate of visual point in (60) is on the virtual image plane, because it is assumed that the helicopter is equipped with a pan-tilt-zoom camera, which is popularly employed in practice. In this case, the image plane is consistent with the virtual image plane, thus the problem of visibility constraint is simplified. In another case, if the camera is strictly attached on the helicopter without rotation, the visible requirement is modified as $h_c = \iota^2 - {}^c u^2 - {}^c n^2 \geq 0$. With (13), its derivatives are

$$\begin{aligned} \dot{h}_c =& -2\begin{bmatrix} -\frac{\lambda u}{z} & -\frac{\lambda n}{z} & \frac{u^2+n^2}{z} \end{bmatrix}\boldsymbol{R}^T\boldsymbol{V}_a \\ &-2\begin{bmatrix} \frac{u^2n+n^3}{\lambda}+n\lambda & -\frac{un^2+u^3}{\lambda}-u\lambda & 0 \end{bmatrix}\boldsymbol{\Omega} \end{aligned} \tag{67}$$

$$\ddot{h}_c = -2\begin{bmatrix} u & n \end{bmatrix}\begin{bmatrix} \ddot{u} \\ \ddot{n} \end{bmatrix} - 2\begin{bmatrix} \dot{u} & \dot{n} \end{bmatrix}\begin{bmatrix} \dot{u} \\ \dot{n} \end{bmatrix} \tag{68}$$

$$\begin{aligned} \begin{bmatrix} \ddot{u} \\ \ddot{n} \end{bmatrix} =& \begin{bmatrix} -\frac{\lambda}{z} & 0 & \frac{u}{z} \\ 0 & -\frac{\lambda}{z} & \frac{n}{z} \end{bmatrix}\left(\boldsymbol{R}^T\dot{\boldsymbol{V}}_a - \boldsymbol{S}(\boldsymbol{\Omega})\boldsymbol{R}^T\boldsymbol{V}_a\right) \\ &+\begin{bmatrix} \frac{\lambda\dot{z}}{z^2} & 0 & \frac{\dot{u}z-\dot{z}u}{z^2} \\ 0 & \frac{\lambda\dot{z}}{z^2} & \frac{\dot{n}z-\dot{z}n}{z^2} \end{bmatrix}\boldsymbol{R}^T\boldsymbol{V}_a \\ &+\begin{bmatrix} \frac{un}{\lambda} & -\lambda-\frac{u^2}{\lambda} & n \\ \lambda+\frac{n^2}{\lambda} & -\frac{un}{\lambda} & -u \end{bmatrix}\dot{\boldsymbol{\Omega}} \\ &+\begin{bmatrix} \frac{\dot{u}n+u\dot{n}}{\lambda} & -1-\frac{2u\dot{u}}{\lambda} & \dot{n} \\ 1+\frac{2n\dot{n}}{\lambda} & -\frac{\dot{u}n+u\dot{n}}{\lambda} & \dot{u} \end{bmatrix}\boldsymbol{\Omega} \end{aligned} \tag{69}$$

where the superscript c in ${}^c u$, ${}^c n$ and ${}^c z$ is also omitted. In this case, the measurement of angular acceleration of the helicopter $\dot{\boldsymbol{\Omega}}$ is required as well.

V. Simulations

In this section, we perform several numerical simulations on MATLAB to evaluate the presented IBVS shipboard landing method.

The model parameters of helicopter are selected as $m = 7.4(\text{kg})$, $\boldsymbol{J} = \text{diag}\{0.16, 0.3, 0.31\}(\text{kgm}^2)$ [43]. The focal length of camera is settled as 3.2(mm). The visual targets attached on the ship are assumed as four vertexes of a rectangle with a side length of 0.2(m) (P_1,P_2,P_3,P_4). The center of this rectangle is the desired landing point. The desired image feature is settled as $\boldsymbol{q}_d = [0, 0, 0.1](\text{m})$, where $0.1(\text{m})$ is the vertical distance between the camera and the bottom bracket of the helicopter. a^* is calculated as 8.192×10^{-5} when the camera is 0.1m above the target. To be close to the practical landing system, the measurement noises and actuator saturation are considered in the simulation. The attitude measurement error of ship and helicopter is given by $0.01\text{rand}(0, 1)(\text{rad})$. The measurement error of extracted image feature is assumed as $0.01\text{rand}(1, 2)$. The control inputs are limited by $T_m \in [0, 85](\text{N})$ and $\tau_i \in [-20, 20](\text{Nm})(i = 1, 2, 3)$.

To exemplify the reliability of the proposed IBVS controller in different conditions, we specify two different landing scenarios.

1) Scenario A: In this scenario, the autonomous shipboard landing is carried out in a relatively peaceful environment, which means that the ship motion and the external disturbances are relatively small. The ship motion is assumed as

$$\begin{cases} x_s(t) = 0.5(t - t_0)(\text{m}) \\ y_s(t) = 0.2\sin(0.6(t - t_0)) + 0.6\sin(0.4(t - t_0))(\text{m}) \\ z_s(t) = 0.12\sin(0.8(t - t_0)) + 0.2\sin(0.7(t - t_0))(\text{m}) \\ \phi_s(t) = 10^{-2} \times (2.1\sin(0.46t) + 4.3\sin(0.54t) + 2.9\sin(0.62t) \\ \qquad +2.2\sin(0.7t))\,(\text{rad}) \\ \theta_s(t) = 10^{-3} \times (0.5\sin(0.46t) + 9.64\sin(0.54t) + 7.25\sin(0.7t) \\ \qquad +8.45\sin(0.82t))\,(\text{rad}) \\ \psi_s(t) = \arctan(y_s(t), x_s(t))(\text{rad}) \end{cases}$$

where t_0 is the switching time of the landing phase. The external disturbances are described as $\boldsymbol{d}_t = [-0.44\sin(0.5t) + 0.45\cos(0.5t), 0.45\sin(0.5t) - 0.44\cos(0.5t), -0.23\sin(0.5t) + 0.23\cos(0.5t)]^T(\text{N})$ and $\boldsymbol{d}_\tau = [0.3 - 0.04\sin(0.5t) + 0.05\cos(0.5t), 0.3 + 0.05\sin(0.5t) - 0.04\cos(0.5t), 0.3 - 0.03\sin(0.5t) + 0.03\cos(0.5t)]^T(\text{Nm})$. The initial conditions of the helicopter are set as $\boldsymbol{\xi}(0) = [-3, 3, -4]^T(\text{m})$, $\boldsymbol{Q}(0) = \text{angle2quat}(10\pi/180, -10\pi/180, 0(\text{rad}))^T$, $\boldsymbol{V}(0) = [0, 0, 0]^T(\text{m/s})$, $\boldsymbol{\Omega}(0) = [0, 0, 0]^T(\text{rad/s})$, where angle2quat is a transformation function from Euler angle to quaternion. Besides, The switching parameters of landing phase in Remark 3 is $\bar{\phi} = \bar{\theta} = \pi/180(\text{rad})$ and $\Delta t = 10(\text{s})$.

2) Scenario B: This scenario is to simulate the autonomous shipboard landing in a relatively harsh environment. The ship motion is modeled as

$$\begin{cases} x_s(t) = 0.3(t - t_0)(\text{m}) \\ y_s(t) = 0.1(t - t_0)(\text{m}) \\ z_s(t) = 0.2\sin(0.8(t - t_0)) + 0.32\sin(0.7(t - t_0))(\text{m}) \\ \phi_s(t) = 10^{-1} \times (1.02\sin(0.4t) + 1.06\sin(0.52t) + 0.58\sin(0.6t) \\ \qquad +0.24\sin(0.45t))\,(\text{rad}) \\ \theta_s(t) = 10^{-2} \times (0.18\sin(0.44t) + 3.37\sin(0.58t) + 2.19\sin(0.7t) \\ \qquad +2.96\sin(0.82t))\,(\text{rad}) \\ \psi_s(t) = \arctan(y_s(t), x_s(t))(\text{rad}) \end{cases}$$

The external disturbances in this scenario are 1.5 times bigger than that in scenario A. The initial position of the helicopter is $\boldsymbol{\xi}(0) = [-2, 4, -5]^T(\text{m})$. Other initial conditions are the

same with that in scenario A. The switching parameters of landing phase in Remark 3 is $\bar{\phi} = \bar{\theta} = 1.2\pi/180(\text{rad})$ and $\Delta t = 10(\text{s})$.

A. Simulations of IBVS controller without visibility constraint

First, the IBVS controller without the consideration of visibility constraint is validated in two different landing scenarios. The control parameters are listed as $\alpha = 4.5$, $k_\alpha = 1.5$, $k_\beta = 2$, $k_m = k_n = 1$, $k_\gamma = 95$, $l_1 = 10$, $l_2 = 22$, $\chi = 100$, $\sigma = 0.028$, $\boldsymbol{\Theta}_D = [1.2, 1.2, 1.2]^T$, $\beta = 0.3$, $\gamma = 1.5$, $k_s = 25$, $\delta = 0.01$, $k_d = 0.85$. The parameters of command filter are $\lambda_n = 25$, $\omega_n = 1.5$. The initial weight matrix is set as $\hat{\boldsymbol{W}}(0) = \boldsymbol{0}^{3\times 19}$.

The simulation results of two scenarios are presented in Figs.4-11. The ship motion prediction results are displayed in Fig. 4, where the prediction starts from 80s. From the comparison between the predicted motion and the actual motion, it is obvious that the AR model method has satisfactory prediction accuracy. On the basis of Remark 2 in Section IV, Fig. 5 is shown to decide the switching time of landing phase explicitly. It should be mentioned that the prediction range of each step is the ship motion in the next 10s (Δt), and the interval between two predictions step is 5s Intuitively, the landing phase should be switched on at 95s in Scenario A and 130s in Scenario B. Fig.6 illustrates 3D and 2D XOY-plane trajectories of the helicopter and the ship in the landing phase. The time histories of image feature and two auxiliary variables in Fig.7 indicates that the helicopter can successfully complete the shipboard landing within 10s with good performance in spite of different initial error condition, ship motion and external disturbances in two scenarios. Fig.8 describes the actual attitude and desired attitude in quaternion form. It can be seen that a good attitude tracking performance is also ensured. Fig.9 depicts the comparison between the estimated value and the actual value of the uncertain term $\boldsymbol{D}$, which indicates that the modified Chebyshev neural network method provides high approximation precision after 3s. Fig.10 shows estimated upper bounds of torque disturbances. The control input signals are collected in Fig.11. Therefore, it indicates that the proposed IBVS controller can complete the autonomous shipboard landing in different scenarios.

In addition, Fig.12 shows projections of visual points on the virtual image plane, where square points represent the initial locations, quinquangular points represent the final locations, and the green circle is presumed as the camera's FoV. It is apparent that the visibility constraint is violated in this situation (shown as the arrows in Fig.12). This problem will be addressed in the next simulation.

B. Simulations of IBVS controller with visibility constraint

In this subsection, the visibility constraint is taken into consideration and the control command is modified by (66). The radius of the visible image plane is specified as $\iota = 7 * 10^{-3}(\text{m})$. The parameters of velocity estimator are selected as $\Lambda_1 = \Lambda_2 = 15$. The gains of zero barrier function are $k_1 = k_2 = 10$. The simulation results can be viewed in Figs.13-18. The 3D trajectories of helicopter and ship are summarized in Fig.13. The control command signals are shown in Fig.14. Specially, Fig.15 demonstrates that all visual points can be restricted inside the camera's FoV. Meanwhile, the visibility constraint function h is always bigger than zero (see Fig.16). Fig.17 shows that the relative linear velocity can be estimated accurately with the velocity observer.

Furthermore, a host of stochastic simulations are performed under different external disturbances within a certain range. In these simulations, external disturbances in Scenario A are modified as $\boldsymbol{d}_{ti} = \text{rand}(-0.5, 0.5)\sin(\text{rand}(0,1)t) + \text{rand}(-0.5, 0.5)\cos(\text{rand}(0,1)t)(\text{N})$ and $\boldsymbol{d}_{\tau i} = 0.3 + \text{rand}(-0.05, 0.05)\sin(\text{rand}(0,1)t) + \text{rand}(-0.05, 0.05)\cos(\text{rand}(0,1)t)(\text{Nm})$, $(i = 1, 2, 3)$, while disturbances in Scenario B are still set as 1.5 times bigger than Scenario A. We can observe from Fig.18 that the statistical landing points are all within the desired zone, which demonstrates the reliability and robustness of the proposed control method.

It can be well illustrated from the simulation results that the proposed constrained IBVS controller can not only achieve the autonomous shipboard landing with high robustness and high accuracy, but also guarantee the visibility of the visual target, despite the 6-DOF moving ship, unknown relative linear velocity, disturbances and measurement noises.

VI. Conclusion

A shipboard landing control method of the helicopter consisting of a novel constrained image-based visual servoing output-feedback control algorithm and a ship motion predictor is proposed. Based on the perspective projection of visual target on the virtual image plane, the image feature instead of relative position between helicopter and ship is extracted as the feedback of controller. In the translational motion control loop, a robust sliding mode control approach with two auxiliary systems is designed to track the desired image feature in the premise of the lack of relative linear velocity. Likewise, the Chebyshev Neural Network is modified to estimate uncertain term resulting from unknown ship motion and perturbations without the requirement of linear velocity. In the attitude control loop, a sliding mode control approach with an adaptive law is employed to track the desired attitude and compensate for the influence of perturbations. After the nominal control design, the control barrier function with a quadratic programming is added to confirm the visual target always inside the visible range of camera, where the unknown relative velocity is estimated by a velocity observer. Additionally, the pitch and roll motion of ship is forecast by the auto-regressive model to determine the start time of landing phase. The stability and effectiveness of the shipboard landing control system has been proven by theoretical analysis and simulations.

References

[1] M. Chen, P. Shi, and C. Lim, "Adaptive neural fault-tolerant control of a 3-dof model helicopter system," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 2, pp. 260–270, 2016.

[2] A. Abdessameud and A. Tayebi, "Global trajectory tracking control of vtol-uavs without linear velocity measurements," *Automatica*, vol. 46, no. 6, pp. 1053–1059, 2010.

[3] S. Lin, M. A. Garratt, and A. J. Lambert, "Monocular vision-based real-time target recognition and tracking for autonomously landing an uav in a cluttered shipboard environment," *Autonomous Robots*, vol. 41, no. 4, pp. 881–901, 2016.

[4] K. Xia, S. Lee, and H. Son, "Adaptive control for multi-rotor uavs autonomous ship landing with mission planning," *Aerospace Science and Technology*, vol. 96, p. 105549, 2020.

[5] J. Boskovic and J. Redding, "An autonomous carrier landing system for unmannned aerial vehicles," in *AIAA Guidance, Navigation, and Control Conference*, 2009.

[6] Z. Zhen, S. Jiang, and J. Jiang, "Preview control and particle filtering for automatic carrier landing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 6, pp. 2662–2674, 2018.

[7] Y. Meng, W. Wang, H. Han, and J. Ban, "A visual/inertial integrated landing guidance method for uav landing on the ship," *Aerospace Science and Technology*, vol. 85, pp. 474–480, 2019.

[8] J. M. Urnes and R. K. Hess, "Development of the f/a-18a automatic carrier landing system," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 3, pp. 289–295, 1985.

[9] J. Rife, S. Khanafseh, S. Pullen, D. De Lorenzo, K. Ung-Suok, M. Koenig, C. Tsung-Yu, B. Kempny, and B. Pervan, "Navigation, interference suppression, and fault monitoring in the sea-based joint precision approach and landing system," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 1958–1975, 2008.

[10] O. So-Ryeok, K. Pathak, S. K. Agrawal, H. R. Pota, and M. Garratt, "Approaches for a tether-guided landing of an autonomous helicopter," *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 536–544, 2006.

[11] D. Xu, J. Lu, P. Wang, Z. Zhang, and Z. Liang, "Partially decoupled image-based visual servoing using different sensitive features," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2233–2243, 2017.

[12] L. R. García Carrillo, E. Rondon, A. Sanchez, A. Dzul, and R. Lozano, "Stabilization and trajectory tracking of a quad-rotor using vision," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 103–118, 2010.

[13] Z. Liu and F. Wang and Y. Zhang, "Adaptive visual tracking control for manipulator with actuator fuzzy dead-zone constraint and unmodeled dynamic," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 10, pp. 1301–1312, 2015.

[14] H. De Plinval, P. Morin, P. Mouyon, and T. Hamel, "Visual servoing for underactuated vtol uavs: a linear, homography-based framework," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 16, pp. 2285–2308, 2014.

[15] F. Le Bras, T. Hamel, R. Mahony, and A. Treil, "Output feedback observation and control for visual servoing of vtol uavs," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 9, pp. 1008–1030, 2011.

[16] P. Serra, R. Cunha, T. Hamel, D. Cabecinhas, and C. Silvestre, "Landing of a quadrotor on a moving target using dynamic image-based visual servo control," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1524–1535, 2016.

[17] B. Herissé, T. Hamel, R. Mahony, and F. X. Russotto, "Landing a vtol unmanned aerial vehicle on a moving platform using optical flow," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 77–89, 2012.

[18] H. Xie and A. F. Lynch, "Input saturated visual servoing for unmanned aerial vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 952–960, 2017.

[19] Z. Cao, X. Chen, Y. Yu, J. Yu, X. Liu, C. Zhou, and M. Tan, "Image dynamics-based visual servoing for quadrotors tracking a target with a nonlinear trajectory observer," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 376–384, 2020.

[20] D. Zheng, H. Wang, J. Wang, S. Chen, W. Chen, and X. Liang, "Image-based visual servoing of a quadrotor using virtual camera approach," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 972–982, 2017.

[21] D. Zheng, H. Wang, W. Chen, and Y. Wang, "Planning and tracking in image space for image-based visual servoing of a quadrotor," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3376–3385, 2018.

[22] J. Huang, W. Wang, C. Wen, and G. Li, "Adaptive event-triggered control of nonlinear systems with controller and parameter estimator triggering," *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 318–324, 2020.

[23] A.-M. Zou and K. D. Kumar, "Adaptive attitude control of spacecraft without velocity measurements using chebyshev neural network," *Acta Astronautica*, vol. 66, no. 5-6, pp. 769–779, 2010.

[24] W. Wang, C. Wen, J. Huang, and Z. Li, "Hierarchical decomposition based consensus tracking for uncertain interconnected systems via distributed adaptive output feedback control," *IEEE Transactions on Automatic Control*, vol. 61, no. 7, pp. 1938–1945, 2016.

[25] Z. Peng and J. Wang, "Output-feedback path-following control of autonomous underwater vehicles based on an extended state observer and projection neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 535–544, 2018.

[26] X. Hu, X. Wei, J. Han, and X. Zhu, "Adaptive disturbance estimation and cancelation for ships under thruster saturation," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 13, pp. 5004–5020, 2020.

[27] A. Abdessameud and F. Janabi-Sharifi, "Image-based tracking control of vtol unmanned aerial vehicles," *Automatica*, vol. 53, pp. 111–119, 2015.

[28] H. Xie, A. F. Lynch, K. H. Low, and S. Mao, "Adaptive output-feedback image-based visual servoing for quadrotor unmanned aerial vehicles," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 1034–1041, 2020.

[29] H. Jabbari Asl, "Robust vision-based tracking control of vtol unmanned aerial vehicles," *Automatica*, vol. 107, pp. 425–432, 2019.

[30] H. Jabbari Asl and J. Yoon, "Robust image-based control of the quadrotor unmanned aerial vehicle," *Nonlinear Dynamics*, vol. 85, no. 3, pp. 2035–2048, 2016.

[31] L. Burlion and H. d. Plinval, "Keeping a ground point in the camera field of view of a landing uav," in *IEEE International Conference on Robotics and Automation*, pp. 5763–5768, 2013.

[32] N. Metni and T. Hamel, "A uav for bridge inspection: Visual servoing control law with orientation limits," *Automation in Construction*, vol. 17, no. 1, pp. 3–10, 2007.

[33] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[34] M. Z. Romdlony and B. Jayawardhana, "Stabilization with guaranteed safety using control lyapunov–barrier function," *Automatica*, vol. 66, pp. 39–47, 2016.

[35] Y. Zou and W. Huo, "Singularity-free backstepping controller for model helicopters," *ISA Transactions*, vol. 65, pp. 133–142, 2016.

[36] X. Jin, W. M. Haddad, Z. P. Jiang, A. Kanellopoulos, and K. G. Vamvoudakis, "An adaptive learning and control architecture for mitigating sensor and actuator attacks in connected autonomous vehicle platoons," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 3, pp. 1788–1802, 2019.

[37] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," pp. 54–61, 2015.

[38] Y. Zou and Z. Zheng, "A robust adaptive rbfnn augmenting backstepping control approach for a model-scaled helicopter," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2344–2352, 2015.

[39] Z. Baharudin, M. A. Zakariya, M. HarisMdKhir, P. Nallagownden, and M. Q. Raza, "Ar-based algorithms for short term load forecast," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, no. 6, pp. 1223–1229, 2014.

[40] R. Hess, "Simplified technique for modeling piloted rotorcraft operations near ships," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1339–1349, 2006.

[41] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, "An invitation to 3-d vision: From images to geometric models," *New York: Springer Verlag*, 2004.

[42] F. Lamnabhi-Lagarrigue, *Advanced Topics in Control Systems Theory (vol. # 311) Lecture Notes from FAP 2004*. Lecture Notes in Control and Information Sciences, 311, London: Springer-Verlag London Ltd, 2005.

[43] V. Gavrilets, *Autonomous Aerobatic Maneuvering of Miniature Helicopters*. Ph.D. thesis, Massachusetts Institute of Technology, 2003.

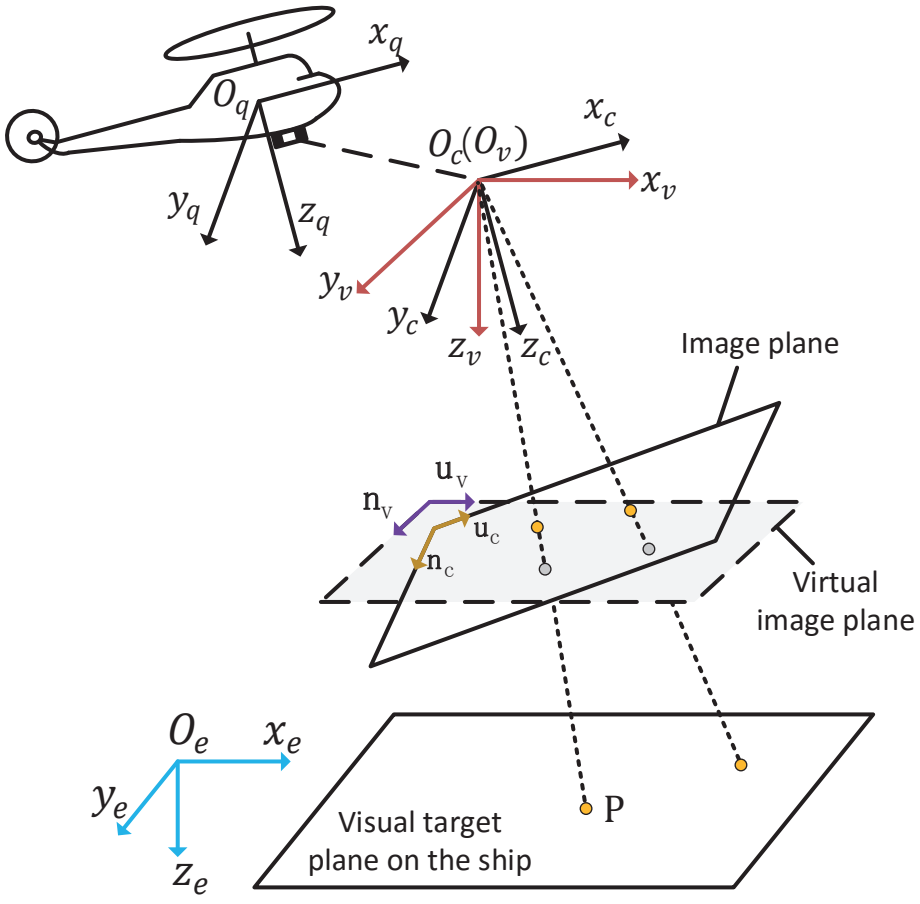

Fig. 1. Reference frames

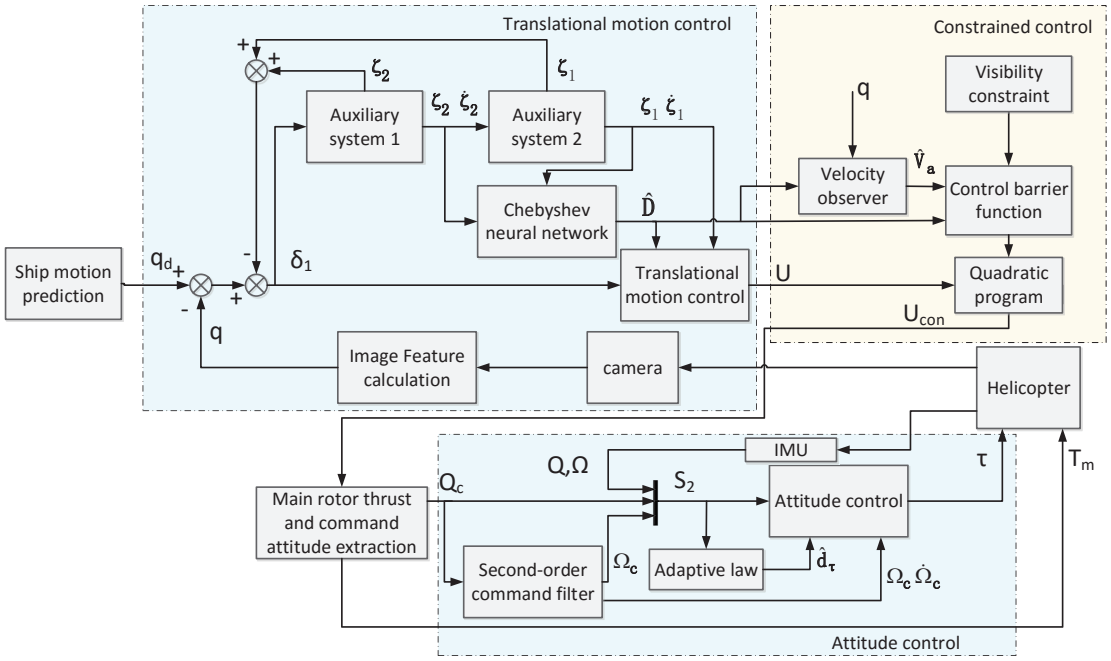

Fig. 2. Block diagram of IBVS shipboard landing design

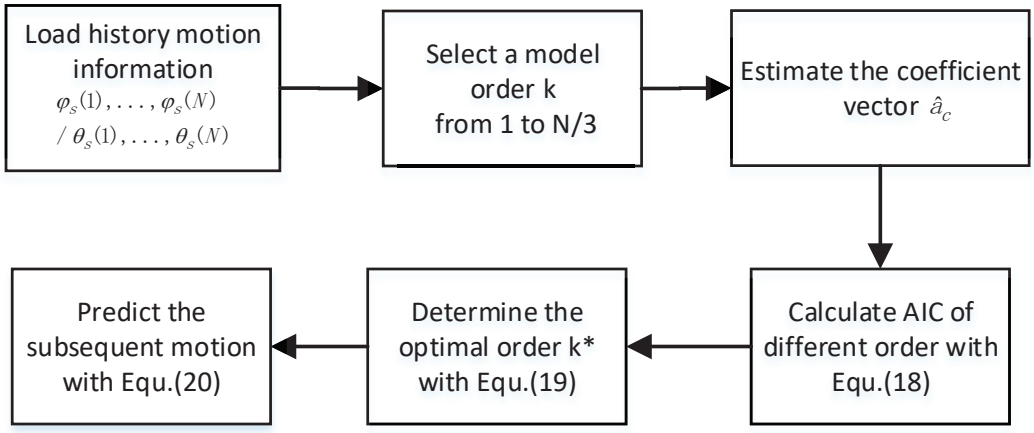

Fig. 3. Diagram of ship motion prediction process

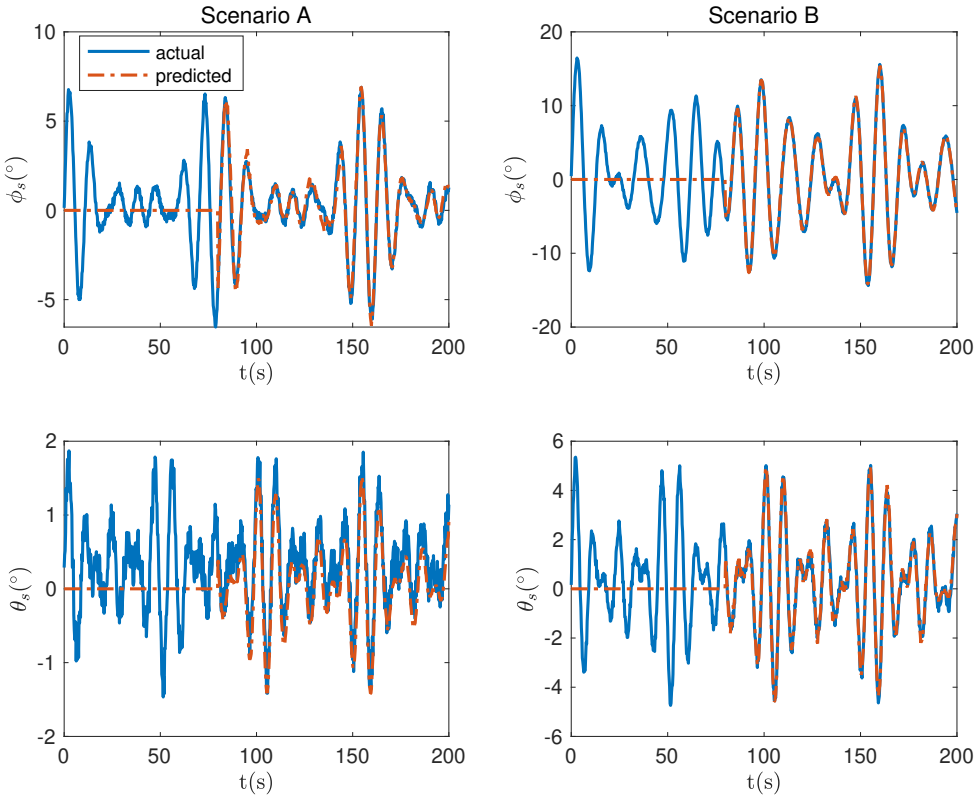

Fig. 4. The prediction result of ship's roll and pitch

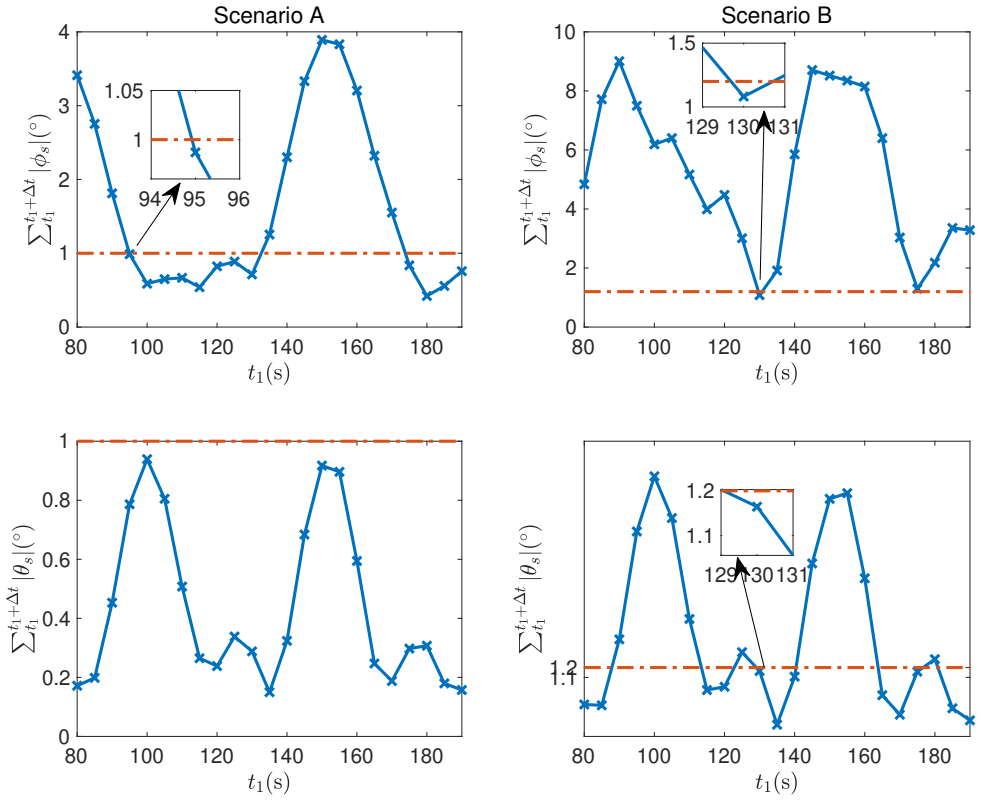

Fig. 5. Switching condition of the landing phase.

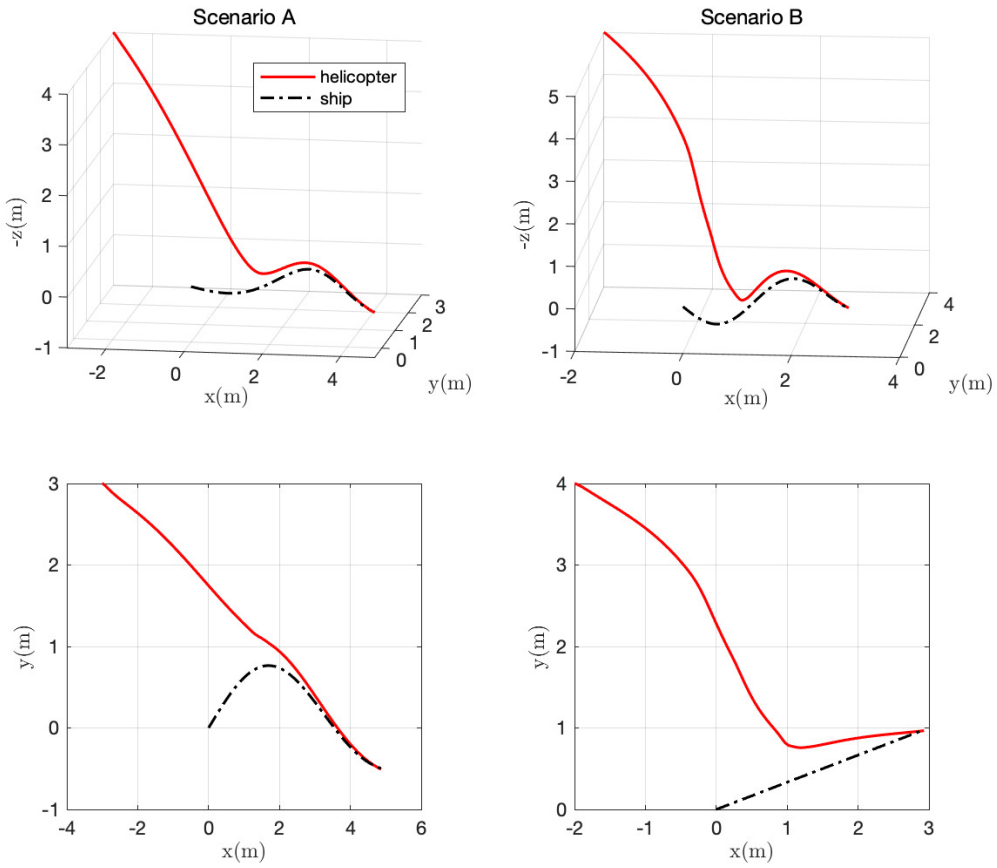

Fig. 6. Simulation A: 3D and 2D XOY-plane trajectories.

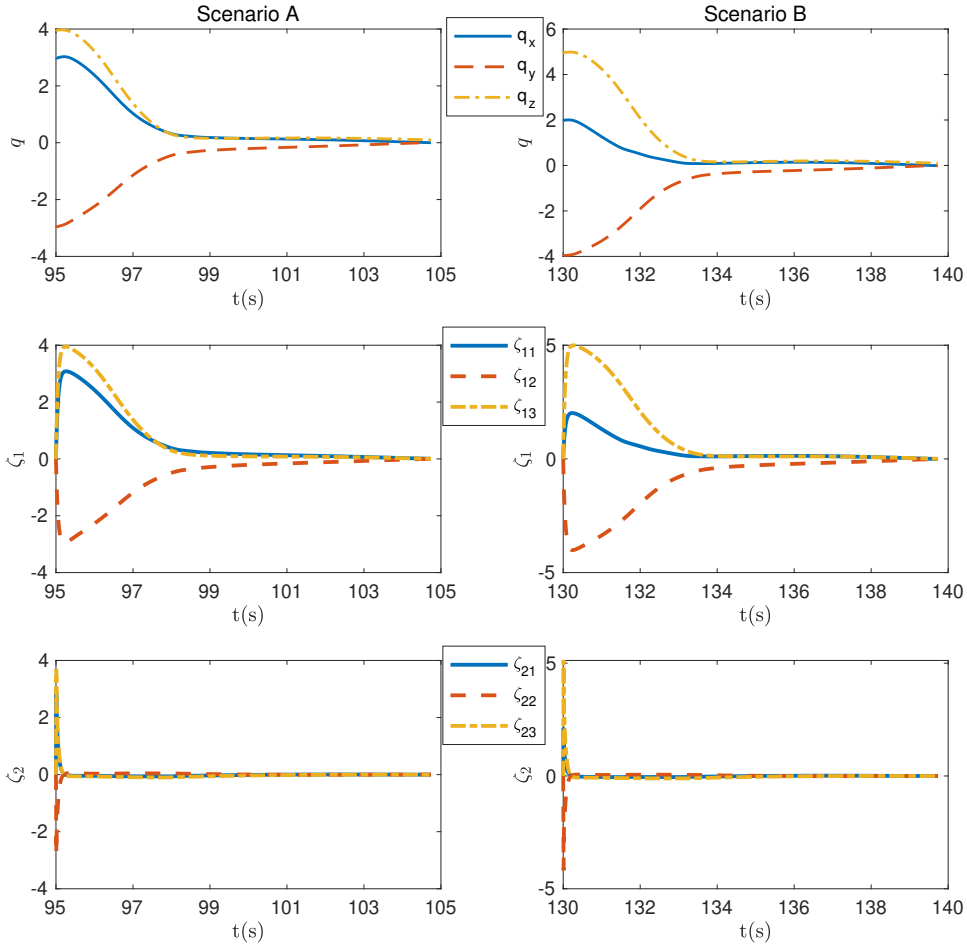

Fig. 7. Simulation A: Image feature, auxiliary variables $\boldsymbol{\zeta}_1$ and $\boldsymbol{\zeta}_2$

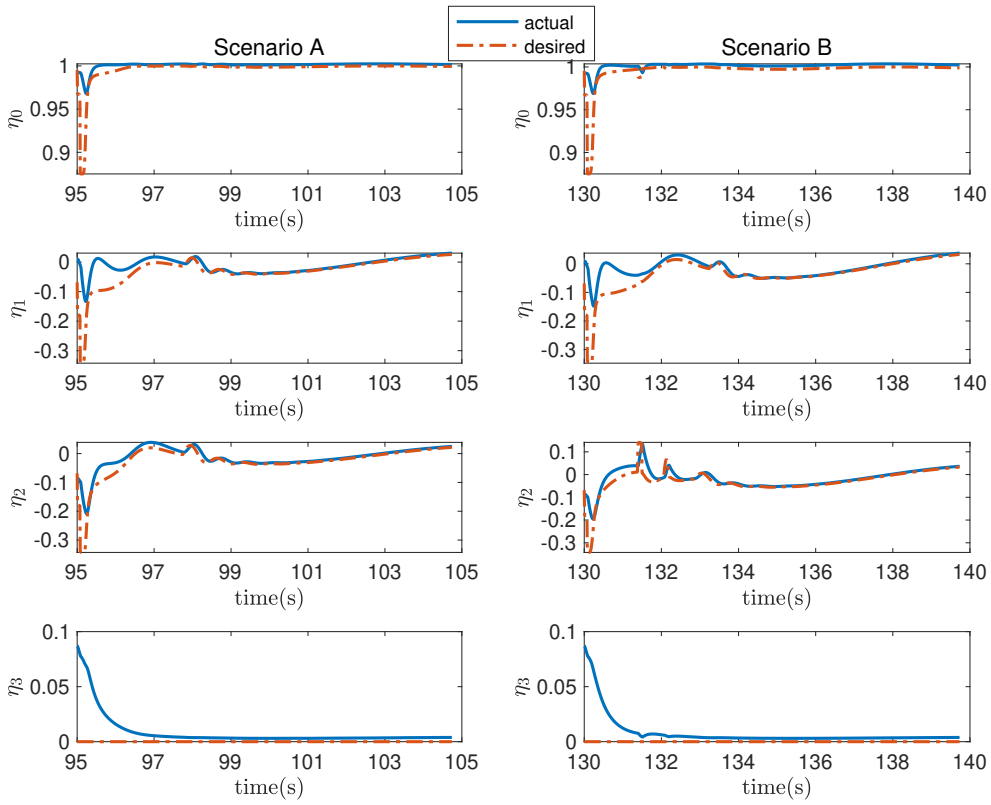

Fig. 8. Simulation A: Attitude.

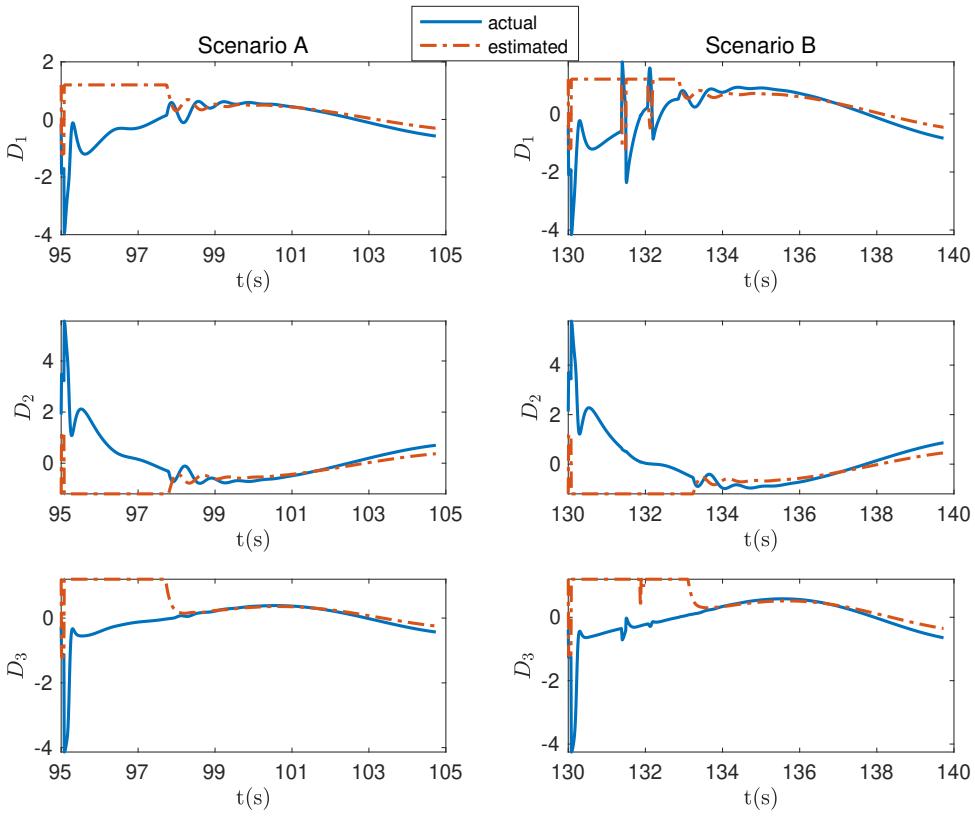

Fig. 9. Simulation A: Estimated translational uncertainties.

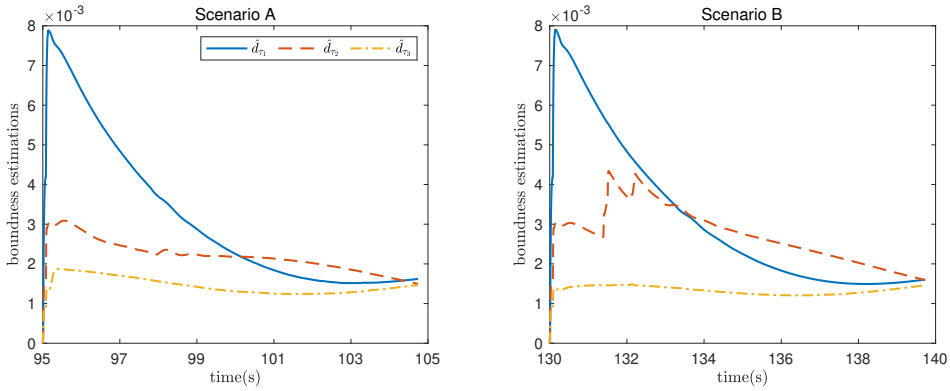

Fig. 10. Simulation A: Estimated upper bounds of torque disturbances.

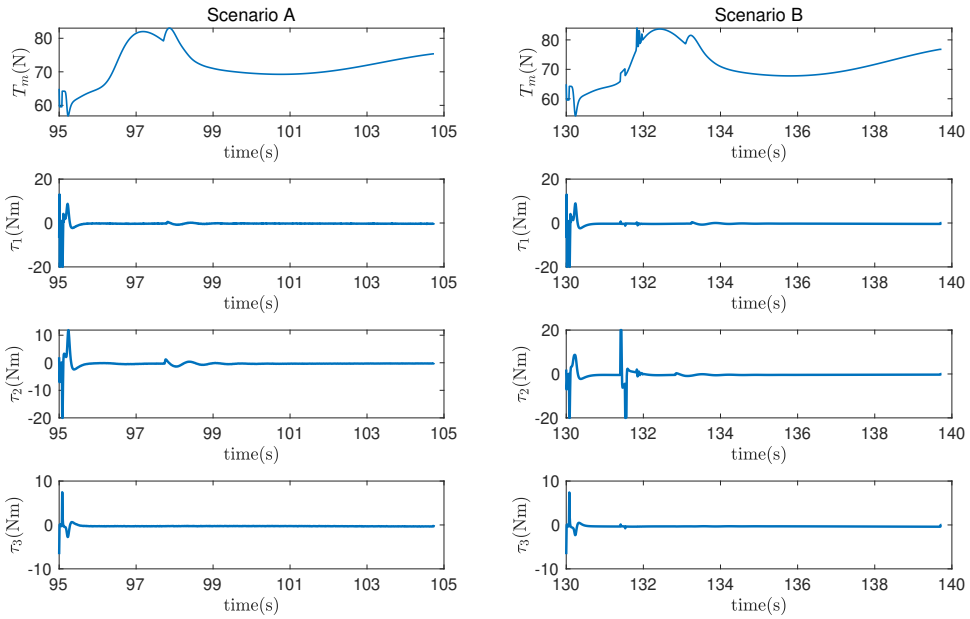

Fig. 11. Simulation A: Control inputs.

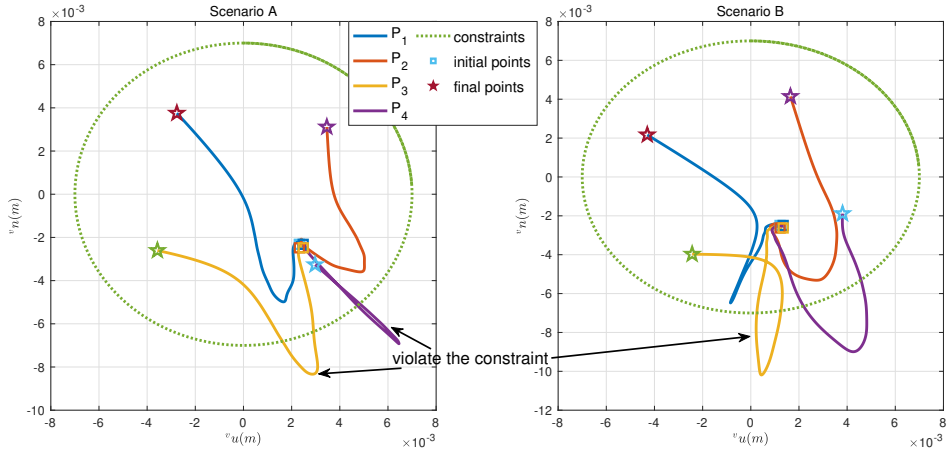

Fig. 12. Simulation A: Projections of visual points on the virtual image plane.

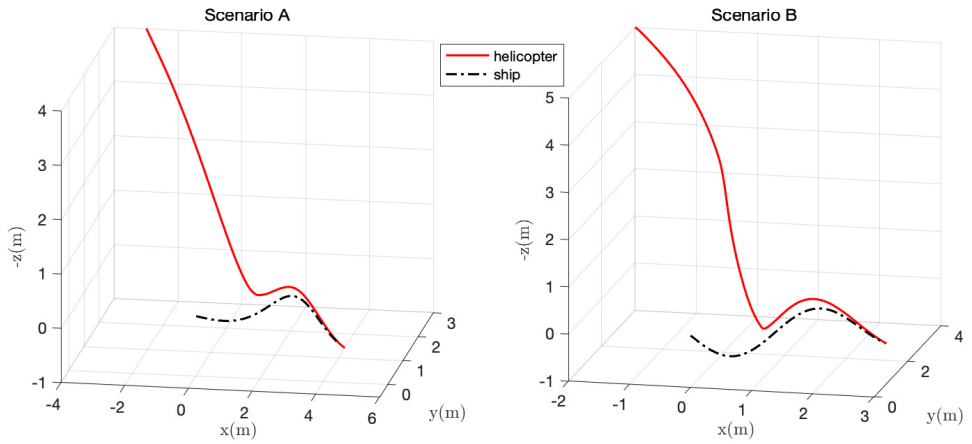

Fig. 13. Simulation B: 3D and 2D XOY-plane trajectories.

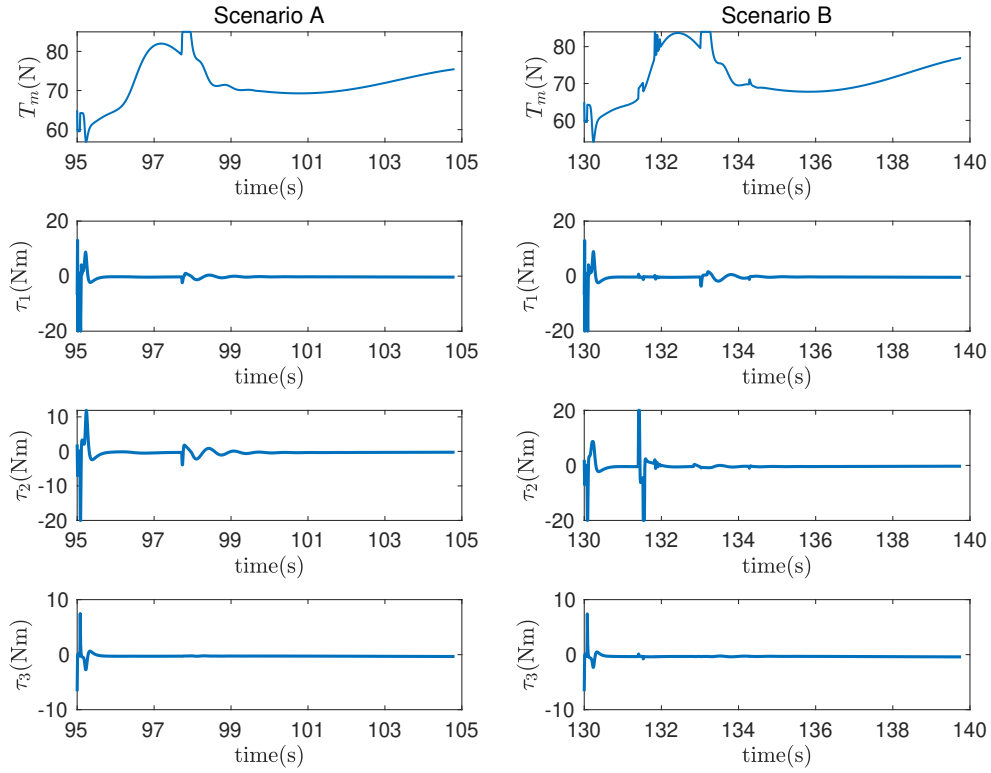

Fig. 14. Simulation B: Control inputs.

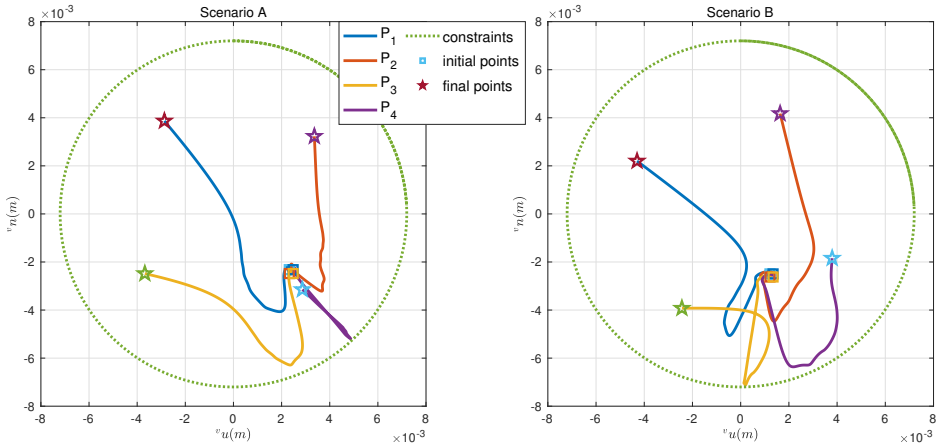

Fig. 15. Simulation B: Projections of visual points on the virtual image plane.

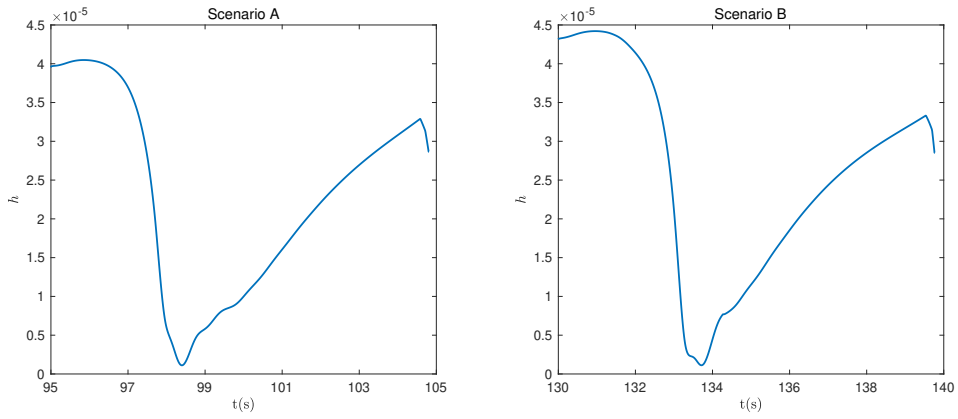

Fig. 16. Simulation B: Visibility constraint function.

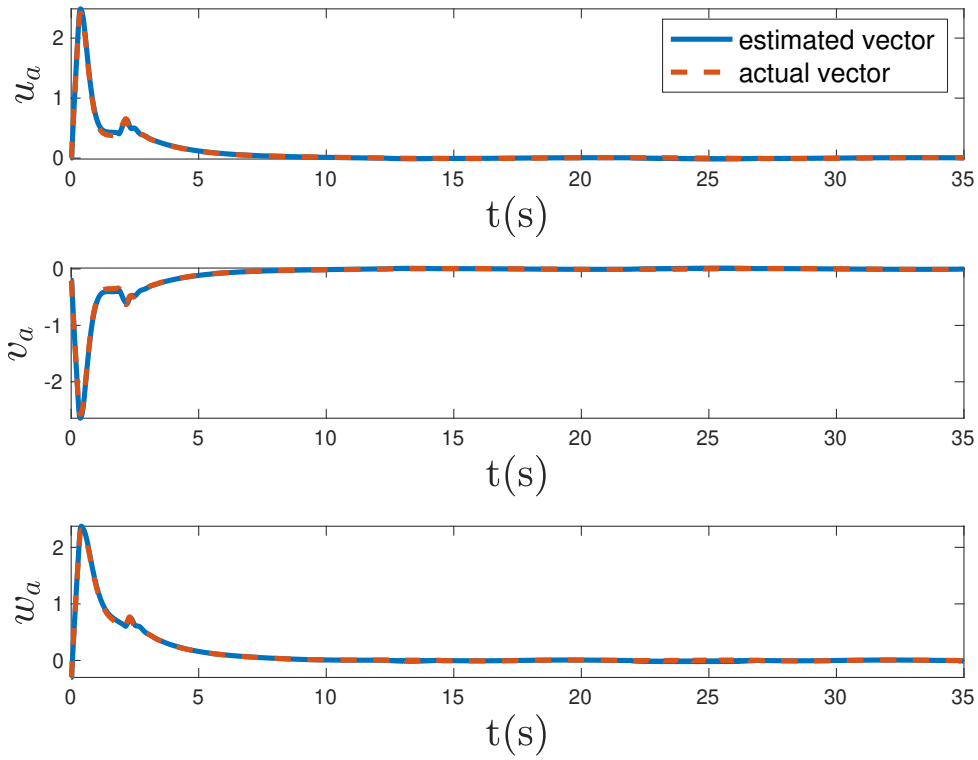

Fig. 17. Simulation B: Comparison of estimated and actual relative linear velocity.

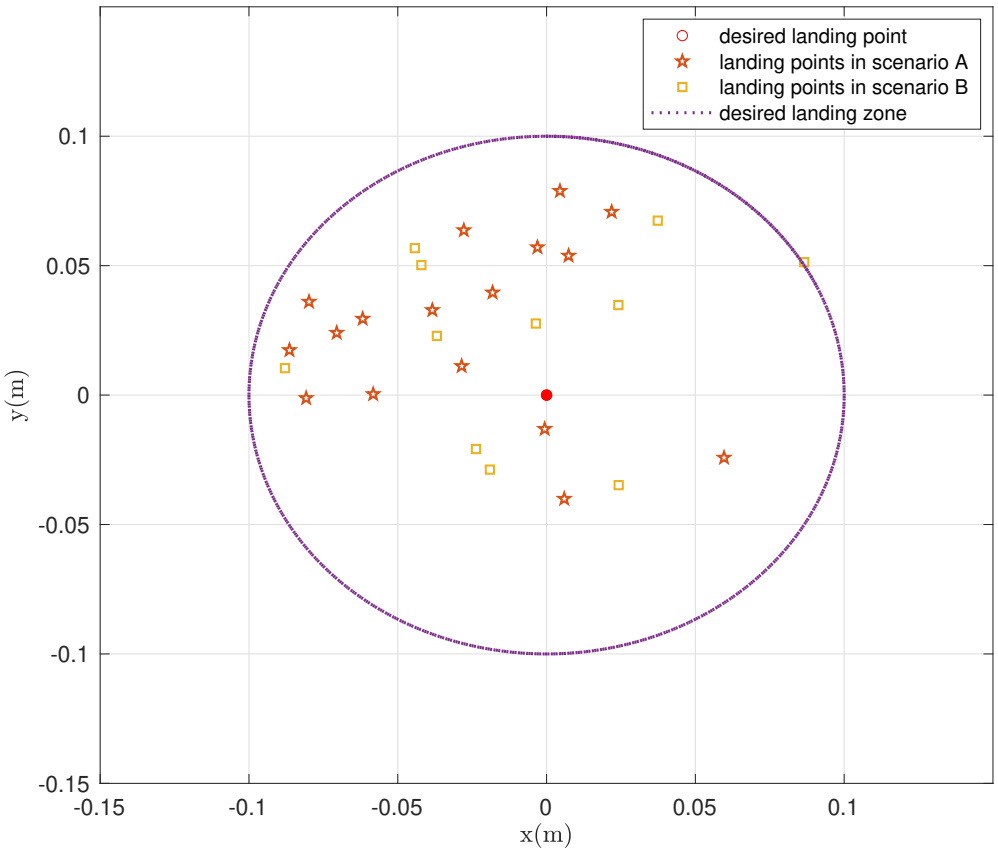

Fig. 18. Simulation B: Landing points.