Journal of
Marine Science and Engineering

MDPI

Article

Hovering Control of ROV Based on Event-Triggered Image Moments Nonlinear Model Predictive Control

Ziyang Shen [1,2,3], Yunxiu Zhang [2,3,*], Shengguo Cui [2,3,*], Peng Zhang [1,2,3] and Siyue Wang [1,2,3]

1 University of Chinese Academy of Sciences, Beijing 100049, China; shenziyang@sia.cn (Z.S.); zhangpeng251@mails.ucas.ac.cn (P.Z.); wangsiyue@sia.cn (S.W.)
2 State Key Laboratory of Robotics and Intelligent Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
3 Key Laboratory of Marine Robotics, Shenyang 110169, China
* Correspondence: zhangyunxiu@sia.cn (Y.Z.); csg@sia.cn (S.C.)

Abstract

Hovering control of remotely operated vehicles (ROVs) is a key fundamental capability for underwater operation tasks, and vision-based ROV hovering control has become a current research focus because of its significant advantages. To address the limited constraint-handling capability and high online optimization frequency in vision-based hovering control, this paper proposes an event-triggered image-moment nonlinear model predictive control method. The proposed method uses image moments as visual feedback features, constructs a prediction model in a reduced control-relevant subspace, and updates the optimal control sequence only when the event-triggering condition is satisfied. Joint simulation experiments were conducted under Ideal, Disturbed, Dynamic-target, and Complex scenarios. The results show that the proposed ETIM-NMPC method achieved effective hovering performance in the joint simulation environment, with translational RMSEs remaining below the simulation-level hovering accuracy reference of 0.15 m in all evaluated directions. Compared with IM-NMPC, ETIM-NMPC maintained comparable control performance while reducing the number of online optimization updates and the cumulative computation time. The trigger rates of ETIM-NMPC ranged from 45.42% to 78.33% across the four scenarios, indicating its on-demand update characteristic under different task conditions.

Keywords: remotely operated vehicles; vision-based hovering control; perspective image moments; nonlinear model predictive control; event-triggered mechanism

1. Introduction

As human attention to marine science and the marine economy continues to increase, remotely operated vehicles (ROVs) have been widely used in various underwater tasks such as marine surveys and marine resource development [1]. For underwater operations, reliable hovering capability is a fundamental requirement that enables ROVs to function as stable working platforms, especially for close-range tasks such as target observation, structural inspection, and intervention. Accordingly, hovering control has received sustained attention since the 1990s [2]. In this paper, hovering refers to the ability of an ROV to maintain a desired position and attitude relative to an artificial marker or a natural target while suppressing drift and oscillation during operation.

Hovering tasks usually rely on sensor assistance, and common sensors include acoustic devices and vision devices. Doppler velocity logs (DVLs) and inertial navigation systems

Academic Editor: Zhiguang Feng

Received: 28 March 2026
Revised: 26 April 2026
Accepted: 29 April 2026
Published: 9 May 2026

(INSs) are not often the first choice for hovering tasks because of their relatively large errors, limited local positioning accuracy, and high cost. In contrast, vision devices have the advantages of lower cost, easy integration, high resolution, and fast update rate, thus providing a feasible way to improve the performance of ROV hovering systems.

Visual servoing, also called vision-based control, refers to the use of image information acquired by vision sensors as feedback to achieve accurate control [3]. In terms of the type of visual information used, visual servo systems fall into three main branches: (1) position-based visual servoing (PBVS); (2) image-based visual servoing (IBVS); and (3) hybrid visual servoing (HVS), also called 2.5D visual servoing. In the PBVS and HVS methods, reconstructing the 3D pose from 2D image data usually depends heavily on prior target models and accurate camera calibration parameters. However, in underwater environments, these conditions are often difficult to satisfy adequately because of visual degradation and model uncertainty [4]. In contrast, IBVS directly constructs the control law from feature coordinates in the image space, yielding a simpler control structure as well as good robustness, computational friendliness, and adaptability to unknown environments. Therefore, this paper focuses on the IBVS hovering problem of ROVs.

For IBVS controller design, the accurate extraction of target features in the image is an important prerequisite for stable motion. Gao et al. [5] designed a visual hovering controller for an ROV using 2D image points as visual features, but assumed that depth information was provided by laser or sonar ranging sensors. Nourmohammadi et al. [6] designed a point-feature-based hovering controller for an ROV and designed a depth estimation method to address the lack of depth information. Therefore, existing visual features such as points, lines, and ellipses still have certain limitations in addressing the interaction-matrix singularity and missing depth information.

To overcome these problems, Jabbari et al. [7] proposed perspective image moments for visual servo control. By selecting properly defined image moments on a rotated image plane, namely the virtual image plane, the coupling in image dynamics can be effectively weakened and the controller design process can be simplified. In the field of UAV hovering and tracking control, Cao et al. [8] proposed an IBVS algorithm that combines perspective image moments with backstepping control for tracking a moving platform. Extending this type of feature to underwater vehicle control, Hong et al. [9] designed an $H\infty$-based HVS system, in which perspective image moments and the Euler angles of an AUV were jointly used as feedback to realize AUV hovering control. Notably, the above control methods are still inadequate for directly handling mechanical constraints and visibility constraints.

In practical applications, if these constraints are not explicitly considered, the control performance of the hovering system may deteriorate significantly. Therefore, it is necessary to introduce a constraint-handling mechanism into the controller design.

Model predictive control (MPC) is an advanced control method that uses a system model to predict future states and obtains control inputs by solving an optimization problem. Its prominent advantage is that it can naturally handle multiple system constraints [10,11]. Nonlinear model predictive control (NMPC) further extends MPC to nonlinear systems, and has therefore attracted widespread attention because of its strong ability to handle nonlinearity and multiple constraints.

In the field of visual servoing, Zhang et al. [12] proposed an NMPC-based IBVS control scheme to keep the target within the camera field of view. In AUV autonomous docking control, Ni et al. [13] proposed a nonlinear model predictive control method for docking trajectory tracking that handles thruster saturation and system-state constraints while ensuring the feasibility and stability of the control system.

For ROV systems, Dai et al. [14] designed a nonlinear model predictive controller combined with Gaussian-function sliding-mode observer compensation to achieve highprecision trajectory tracking. However, the above studies generally did not consider the online computational burden or unnecessary control updates of NMPC, which may cause additional delays in the controller solution and execution [15].

To address this issue, the idea of event-triggered (ET) NMPC was proposed [16–19]. For example, Aspragkathos et al. [20] proposed a new method that combines IBVS with event-triggered nonlinear model predictive control for a UAV hovering over deformable contour targets. Wu et al. [21] constructed an NMPC framework based on an adaptive integral event-triggered mechanism for AUV tracking of a moving docking station. In addition, Luan et al. [22] designed an event-triggered model predictive control method for ship dynamic positioning systems to reduce the communication load. Although event-triggered nonlinear model predictive control has shown good potential in several motion-control scenarios, its application in vision-based hovering control of ROVs remains insufficiently studied. In particular, reducing the computational burden of online optimization while satisfying visibility constraints and mechanical constraints remains an open problem.

To address the above problems, this paper proposes an event-triggered image-moment nonlinear model predictive control method for vision-based ROV hovering control. The main contributions of this paper are summarized as follows.

1. A vision-based hovering modeling framework is established, using perspective image moments on a virtual image plane. By defining perspective image-moment features on the virtual image plane and deriving the corresponding image-dynamics model, a compact visual feedback state is constructed for ROV hovering control.
2. An ETIM-NMPC method is designed by considering both input constraints and visual-feature constraints. A joint predictive state is constructed from the perspective image-moment state and the velocity states related to ROV hovering, and an event-triggered update mechanism is introduced to reduce unnecessary online optimization updates.
3. A joint simulation platform based on WEBOTS 2025a, Python 3.9.23, and MATLAB 2025a is established to verify the proposed method under different scenarios. The simulation results show that ETIM-NMPC maintains effective hovering performance while reducing online optimization updates and cumulative computation time compared with IM-NMPC.

The remainder of this paper is organized as follows. Section 2 presents the ROV dynamic model, image-feature construction, and image-kinematics model. Section 3 describes the proposed ETIM-NMPC method for the automatic ROV hovering task. Section 4 provides the simulation results and comparative analyses. Finally, Section 5 concludes this paper.

2. System Model

2.1. ROV Dynamic Model

To describe the spatial motion of the ROV, an earth-fixed frame $\{E - \xi\eta\zeta\}$ and a body-fixed frame $\{O_B - X_B Y_B Z_B\}$ are established, as shown in Figure 1. The earth-fixed frame is used to represent the pose of the ROV, while the body-fixed frame is fixed to the center of mass of the ROV and translates and rotates synchronously with it. The real camera frame is denoted by $\{O_C - X_C Y_C Z_C\}$, and the virtual camera frame is denoted by $\{O_V - X_V Y_V Z_V\}$.

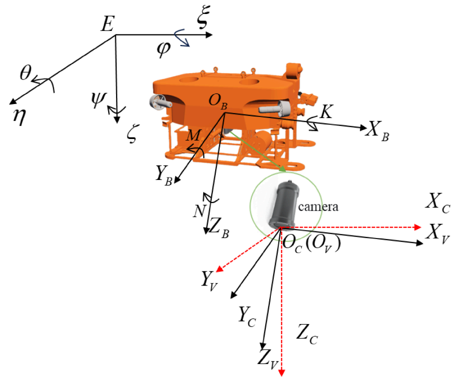

Figure 1. Coordinate frames of the vision-based hovering system.

The kinematic relationship of the ROV is expressed as follows [23]:

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{\nu} \tag{1}$$

where $\boldsymbol{\eta} = [x, y, z, \varphi, \theta, \psi]^{\mathrm{T}}$ is the pose vector of the ROV in the earth-fixed frame, $\boldsymbol{\nu} = [u, v, w, p, q, r]^{\mathrm{T}}$ is the velocity vector of the ROV in the body-fixed frame, and $J(\boldsymbol{\eta})$ is the Jacobian matrix.

The dynamic model of the ROV is established using the Newton–Euler method [24], and is expressed as follows:

$$\mathbf{M}\dot{\boldsymbol{\nu}}+\mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu}+\mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu}+\mathbf{g}(\boldsymbol{\eta})=\boldsymbol{\tau}+\boldsymbol{\tau}_{\mathbf{e}} \tag{2}$$

where $\mathbf{M}$ is the inertia matrix, $\mathbf{C}(\boldsymbol{\nu})$ is the Coriolis and centripetal matrix, $\mathbf{D}(\boldsymbol{\nu})$ is the damping matrix, $\mathbf{g}(\boldsymbol{\eta})$ is the restoring-force and restoring-moment term, $\boldsymbol{\tau} = [F_u, F_v, F_w, T_p, T_q, T_r]^{\mathrm{T}}$ is the resultant force and moment applied by the thrusters in six degrees of freedom, and $\boldsymbol{\tau}_{\mathbf{e}}$ is the external disturbance.

The platform used in this paper is the R-ROV [25], whose control input is generated jointly by eight thrusters. Let the thruster force vector be

$$\mathbf{T} = [T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8]^{\mathrm{T}} \tag{3}$$

Then, the relationship between the thruster forces and the generalized control input satisfies

$$\boldsymbol{\tau} = \overset{\frown}{\mathbf{B}}\mathbf{T} \tag{4}$$

where $\overset{\frown}{\mathbf{B}}$ is the thrust allocation matrix, whose structure is determined by the thruster installation angles and lever-arm parameters.

2.2. Image Features and IBVS Dynamics

To construct a visual feedback state that is suitable for hovering, a virtual camera frame $\{O_V\}$ is introduced on the basis of the real camera frame $\{O_C\}$, as shown in Figure 2. The virtual camera is assumed to be concentric with and move synchronously with the real camera while retaining the yaw angle of the real camera and setting the roll and pitch to zero. Under this assumption, the virtual image plane always remains parallel to the target plane so that the perspective image moments can be constructed on the virtual image plane, ultimately yielding good decoupling properties for image dynamics [26].

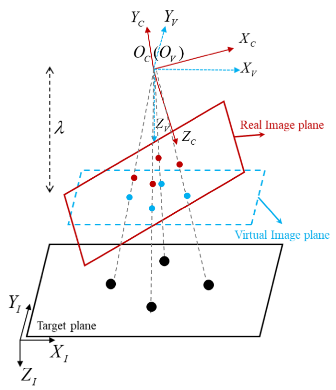

Figure 2. Geometric relationships among the virtual image plane, the real image plane, and the target plane.

Let the pixel coordinates p of a target point on the real image plane be

$$p = (x_n, y_n, 1)^{\mathrm{T}} \tag{5}$$

Its corresponding normalized coordinates $\overline{p}$ are expressed as follows:

$$\overline{p} = \mathbf{K}^{-1} p = (x_n, y_n, 1)^{\mathrm{T}} \tag{6}$$

where $\mathbf{K}$ is the camera intrinsic matrix. After reprojection from the real image plane to the virtual image plane, the normalized coordinates ${}^{V}\overline{p}$ on the virtual image plane can be obtained as follows:

$${}^{V}\overline{p} = ({}^{V}x_n, {}^{V}y_n, 1)^{\mathrm{T}} = \beta R_{\varphi\theta}\overline{p} \tag{7}$$

In Equation (7), $R_{\varphi\theta}$ denotes the roll–pitch compensation rotation matrix defined by the attitude of the real camera. Because the real and virtual camera frames are assumed to be concentric, the reprojection from the real normalized image plane to the virtual normalized image plane only requires rotational compensation, without any additional translational term. The normalization factor is given by $\beta = ([0\ \ 0\ \ 1]R_{\varphi\theta}\overline{p})^{-1}$. Since the real camera is rigidly mounted on the ROV, its attitude is determined by the ROV attitude and the fixed camera-to-body transformation. Therefore, the reprojection in Equation (7) is an attitude-dependent geometric transformation, which is consistent with the velocity transformation introduced in Equations (17)–(19).

Accordingly, the image moments ${}^{V}m_{ij}$ of the target region on the virtual image plane can be defined as follows [27]:

$${}^{V}m_{ij} = \iint_{\Omega} ({}^{V}x_n)^{i} ({}^{V}y_n)^{j} d\Omega \tag{8}$$

where Ω denotes the target region. The target centroid coordinates $({}^{V}x_g, {}^{V}y_g)$ can thus be obtained as follows:

$${}^{V}x_g = \frac{{}^{V}m_{10}}{{}^{V}m_{00}}, {}^{V}y_g = \frac{{}^{V}m_{01}}{{}^{V}m_{00}} \tag{9}$$

and the second-order central moments ${}^{V}\mu_{ij}$ are

$${}^{V}\mu_{ij} = \iint_{\Omega} ({}^{V}x_n - {}^{V}x_g)^{i} ({}^{V}y_n - {}^{V}y_g)^{j} d\Omega \tag{10}$$

The area parameter a is further defined as follows:

$$a = {}^{V}\mu_{20} + {}^{V}\mu_{02} \tag{11}$$

On this basis, the following four-dimensional perspective image moments $\mathbf{s}$ are selected as the visual feedback features:

$$\mathbf{s} = [s_1, s_2, s_3, s_4]^{\mathrm{T}} \tag{12}$$

$$s_1 = s_3{}^V x_g, s_2 = s_3{}^V y_g, s_3 = Z_d\sqrt{\frac{a_d}{a}}, s_4 = \frac{1}{2}\arctan\left(\frac{2^V\mu_{11}}{{}^V\mu_{20} - {}^V\mu_{02}}\right) \tag{13}$$

where a_d is the desired area parameter and Z_d denotes the desired normal distance from the camera to the target plane. In this paper, Z_d is specified by the desired ROV-to-target configuration and the fixed camera-to-body geometry, rather than being estimated online from image moments. Thus, s_1 and s_2 describe the translational deviation of the target on the virtual image plane, s_3 describes the scale variation, and s_4 describes the orientation variation [28].

For a stationary hovering target, the derivative of the image features and the virtual camera velocity satisfy

$$\dot{\mathbf{s}} = \mathbf{L}_s \boldsymbol{\nu}_V \tag{14}$$

where $\boldsymbol{\nu}_V = [{}^V v_x, {}^V v_y, {}^V v_z, {}^V \omega_z]^{\mathrm{T}}$ denotes the virtual camera velocity, including the translational velocity $[{}^V v_x, {}^V v_y, {}^V v_z]^{\mathrm{T}}$ of the virtual camera and the angular velocity ${}^V\omega_z$ about the Z-axis, and the image Jacobian matrix $\mathbf{L}_s$ is expressed as follows:

$$\mathbf{L}_s = \begin{bmatrix} -1 & 0 & 0 & s_2 \\ 0 & -1 & 0 & -s_1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{15}$$

Therefore, the matrix form of the image feature dynamics can be written as

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & s_2 \\ 0 & -1 & 0 & -s_1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} {}^V v_x \\ {}^V v_y \\ {}^V v_z \\ {}^V \omega_z \end{bmatrix} \tag{16}$$

This relationship shows that, on the virtual image plane, the target translation, scale, and orientation variation can all be expressed uniformly as functions of the virtual camera velocity.

Assumption 1. *Roll and pitch motions can be neglected.*

In the close-range visual hovering task considered here, the ROV operates at a low speed and mainly regulates its horizontal motion, vertical motion, and yaw motion relative to the target. In addition, passive restoring effects, such as metacentric stability, help us to keep roll and pitch motions within a small range during hovering. Therefore, their influence on the selected image-feature dynamics is neglected in the controller-side prediction model, and only the surge, sway, heave, and yaw-rate channels are retained for controller design.

The real camera is rigidly mounted on the ROV; thus, a fixed velocity transformation ${}^C\mathbf{T}_B$ exists between the real camera and the ROV. Moreover, the transformation ${}^V\mathbf{T}_C$ between the virtual camera and the real camera can be determined by the above geometric assumption. On the basis of Assumption 1, let the ROV velocity subvector be $\boldsymbol{\nu}_r = [u, v, w, r]^{\mathrm{T}}$ and the real camera velocity be $\boldsymbol{\nu}_C$; then, we have

$$\boldsymbol{\nu}_V = {}^V\mathbf{T}_C \boldsymbol{\nu}_C = {}^V\mathbf{T}_C{}^C\mathbf{T}_B \boldsymbol{\nu}_r \tag{17}$$

By substituting Equation (17) into Equation (14), we can obtain

$$\dot{\mathbf{s}} = \mathbf{L}_s {}^{V}\mathbf{T}_C {}^{C}\mathbf{T}_B \boldsymbol{\nu}_r \tag{18}$$

Thus, the above relation can be written in the four-DOF control subspace, as follows:

$$\dot{\mathbf{s}} = f(\mathbf{s}, \boldsymbol{\nu}_r) \tag{19}$$

Thus far, the image feature dynamics on the virtual image plane have been uniformly linked with the rigid-body motion of the ROV in Section 2.1.

2.3. Problem Statement

The overall structure of the vision-based hovering system is shown in Figure 3. Based on the image data acquired by the camera, the perspective image-moment features on the virtual image plane are calculated. Other sensors provide the attitude and velocity information of the ROV, which, together with the image features, forms the controller input. The ETIM-NMPC controller performs event-triggered optimization according to the current state and outputs the generalized control input. The actuator module maps the control input into thruster commands, thereby achieving vision-based hovering control of the ROV.

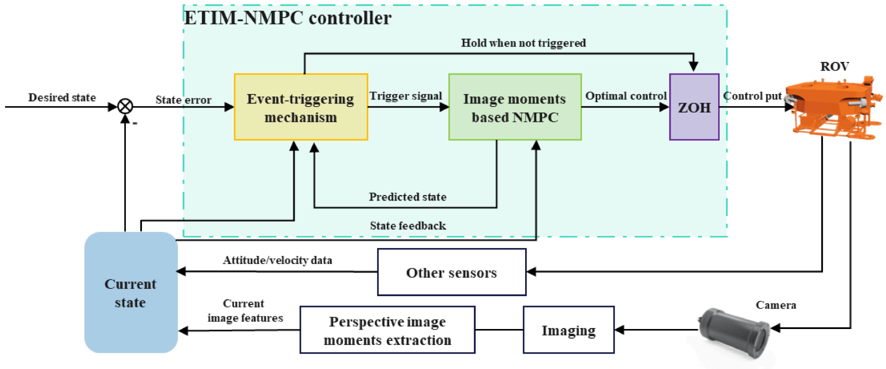

Figure 3. Overall structure of the vision-based hovering system.

3. ETIM-NMPC Controller Design

3.1. State Construction and Prediction Model

On the basis of the ROV dynamic model and image feature dynamics established in Section 2, an ETIM-NMPC controller for the vision-based hovering task is constructed in this paper. The control objective is to make the target image features converge to the desired state and to keep the ROV velocity states bounded under thruster input constraints and visual visibility constraints.

Assumption 2. *The effects of the Coriolis and centripetal terms can be neglected or absorbed into the modeling error.*

Under low-speed ROV motion, the influence of these terms is relatively weak compared with that of the main control-force terms, and they can therefore be incorporated uniformly into the model uncertainty treatment [29].

On the basis of Assumptions 1 and 2, the four-DOF velocity subsystem can be obtained as

$$\mathbf{M}_r \dot{\boldsymbol{\nu}}_r = -\mathbf{D}_r \boldsymbol{\nu}_r + \boldsymbol{\tau}_r + \boldsymbol{\tau}_{\mathbf{e}} \tag{20}$$

where $\boldsymbol{\tau}_r = [F_u, F_v, F_w, F_r]^{\mathrm{T}}$ is the control input, $\boldsymbol{\tau}_{\mathbf{e}}$ is the external disturbance, and the equivalent inertia matrix $\mathbf{M}_r$ and damping matrix $\mathbf{D}_r$ in the four-DOF space are defined as follows:

$$\begin{cases} \mathbf{M}_r = diag\{m, m, m, I_{zz}\} \\ \mathbf{D}_r = diag\left\{X_u + X_{u|u|}, Y_v + Y_{v|v|}, Z_w + Z_{w|w|}, N_r + N_{r|r|}\right\} \end{cases} \tag{21}$$

where m denotes the ROV body mass, I_{zz} denotes the rotational inertia, and the hydrodynamic coefficients $(X_u + X_{u|u|}, Y_v + Y_{v|v|}, Z_w + Z_{w|w|}, N_r + N_{r|r|})$ are used in the model.

From Equation (12), using the current visual feedback feature $\mathbf{s}_k = [s_{1,k}, s_{2,k}, s_{3,k}, s_{4,k}]^{\mathrm{T}}$ and further considering the ROV subvelocity state $\boldsymbol{\nu}_{r,k} = [u_k, v_k, w_k, r_k]^{\mathrm{T}}$, the joint state vector can be constructed as follows:

$$\boldsymbol{x}_k = \begin{bmatrix} \mathbf{s}_k \\ \boldsymbol{\nu}_{r,k} \end{bmatrix} = [s_{1,k}, s_{2,k}, s_{3,k}, s_{4,k}, u_k, v_k, w_k, r_k]^{\mathrm{T}} \tag{22}$$

By substituting Equation (19) into Equation (20), the continuous-time prediction model of the joint state can be uniformly expressed as follows:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\tau}_r) = \begin{bmatrix} f(\mathbf{s}, \boldsymbol{\nu}_r) \\ \mathbf{M}_r^{-1}(-\mathbf{D}_r\boldsymbol{\nu}_r + \boldsymbol{\tau}_r + \boldsymbol{\tau}_{\mathbf{e}}) \end{bmatrix} \tag{23}$$

Discretizing the above continuous model with the sampling period Δt yields

$$\mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \boldsymbol{\tau}_{r,k}) \tag{24}$$

Within the prediction horizon N_p, the state recursion $\mathbf{x}_{k+i+1|k}$ is expressed as follows:

$$\mathbf{x}_{k+i+1|k} = f_d(\mathbf{x}_{k+i|k}, \boldsymbol{\tau}_{r,k+i|k}), i = 0, 1, \ldots, N_p - 1 \tag{25}$$

3.2. Cost Function and Constraints

At time k, the controller takes the current joint state $\mathbf{x}_k$ as the initial value and solves the control sequence $\mathcal{T}_k$ over the prediction horizon N_p:

$$\mathcal{T}_k = \{\boldsymbol{\tau}_{r,k|k}, \boldsymbol{\tau}_{r,k+1|k}, \ldots, \boldsymbol{\tau}_{r,k+N_p-1|k}\} \tag{26}$$

Define the desired joint state as $\mathbf{x}^* = [s_1^*, s_2^*, s_3^*, s_4^*, u^*, v^*, w^*, r^*]^{\mathrm{T}}$. To balance image feature error convergence, velocity-state regulation, and control-input magnitude, the ETIM-NMPC cost function is constructed as follows:

$$\begin{gathered} \min J_k = \sum_{i=0}^{N_{p-1}} (||\mathbf{x}_{k+i|k} - \mathbf{x}_k^*||_{\mathbf{Q}}^2 + ||\boldsymbol{\tau}_{k+i|k}||_{\mathbf{R}}^2) + ||\mathbf{x}_{k+i|k} - \mathbf{x}_k^*||_{\mathbf{P}}^2 \\ \text{s.t. } \mathbf{x}_{k+i+1|k} = f_d(\mathbf{x}_{k+i|k}, \boldsymbol{\tau}_{r,k+i|k}), i = 0, 1, \ldots, N_p - 1 \\ \boldsymbol{\tau}_{r,k+i|k} \in \mathcal{U}, \mathbf{x}_{k+i|k} \in \mathcal{X} \end{gathered} \tag{27}$$

where $\mathbf{Q} \succcurlyeq 0$ is the state weighting matrix, $\mathbf{R} \succcurlyeq 0$ is the input weighting matrix, and $\mathbf{P} \succcurlyeq 0$ is the terminal weighting matrix.

The constraint sets on the input $\mathcal{U}$ and input increment $\Delta\mathcal{U}$ are defined as follows:

$$\begin{aligned} \mathcal{U} &= \left\{\boldsymbol{\tau}_r \in \mathbb{R}^4 \middle| \boldsymbol{\tau}_{r,\min} \le \boldsymbol{\tau}_r \le \boldsymbol{\tau}_{r,\max}\right\} \\ \Delta\mathcal{U} &= \left\{\Delta\boldsymbol{\tau}_r \in \mathbb{R}^4 \middle| \Delta\boldsymbol{\tau}_{r,\min} \le \Delta\boldsymbol{\tau}_r \le \Delta\boldsymbol{\tau}_{r,\max}\right\} \end{aligned} \tag{28}$$

The state constraint set $\mathcal{X}$ is defined as follows:

$$\mathcal{X} = \{\mathbf{x}|s_i \in (s_{i,\min}, s_{i,\max}), i = 1,2,3,4\} \tag{29}$$

where $(s_{i,\min}, s_{i,\max})$ denotes the admissible interval of image features constrained by the field of view.

Therefore, the hovering control problem is formulated as a finite-horizon nonlinear optimization problem with input constraints and visual-feature constraints.

3.3. Event-Triggered Mechanism and Control Procedure

Time-triggered NMPC resolves the optimization problem again at every sampling instant [30]. For the vision-based hovering task, when the system state changes slowly and the image feature error has already entered a small range, continuous online optimization increases the computational burden. To reduce the unnecessary online solution frequency, this paper introduces an event-triggered update mechanism: the optimization problem is resolved again only when the deviation of the actual state relative to the nominal predicted trajectory from the previous update reaches a preset threshold; otherwise, the remaining control inputs in the existing control plan continue to be applied.

Let the most recent triggering instant be t_k. After the optimization problem is solved at this instant, the nominal predicted state trajectory $\widehat{\mathbf{x}}(t_k + m|t_k)$ and the corresponding optimal control sequence $\widehat{\boldsymbol{\tau}}(t_k + m|t_k)$ are obtained as follows:

$$\begin{aligned} &\widehat{\mathbf{x}}(t_k + m|t_k), m = 0,1,\ldots,N_p \\ &\widehat{\boldsymbol{\tau}}(t_k + m|t_k), m = 0,1,\ldots,N_p - 1 \end{aligned} \tag{30}$$

For subsequent instances without retriggering $t_k + m$, the deviation between the current actual state and the nominal predicted state $\mathbf{e}(t_k + m)$ is defined as follows:

$$\mathbf{e}(t_k + m) = \mathbf{x}(t_k + m) - \widehat{\mathbf{x}}(t_k + m|t_k) \tag{31}$$

A weighted deviation index $\delta(t_k + m)$ is further introduced as follows

$$\delta(t_k + m) = \sqrt{\mathbf{e}^{\mathrm{T}}(t_k + m)\mathbf{W}\mathbf{e}(t_k + m)} \tag{32}$$

where $\mathbf{W} \succ 0$ denotes the deviation weighting matrix.

The remaining nominal cost function is defined as follows:

$$S_{\mathrm{nom}}(t_k + m) = \sum_{j=m}^{N_{p-1}} (||\widehat{\mathbf{x}}(t_k + j|t_k) - \mathbf{x}^*||^2_{\mathbf{Q}} + ||\widehat{\boldsymbol{\tau}}_r(t_k + j|t_k)||^2_{\mathbf{R}}) \tag{33}$$

Accordingly, the triggering threshold $\gamma(t_k + m)$ is constructed as follows:

$$\gamma(t_k + m) = \rho S_{\mathrm{nom}}(t_k + m) + \varepsilon \tag{34}$$

where $\rho > 0$ is the threshold scaling coefficient and $\varepsilon > 0$ is the minimum tolerance term. The main triggering criterion is then given by

$$c\delta(t_k + m) > \gamma(t_k + m) \tag{35}$$

where $c > 0$ is the proportional coefficient.

To avoid overly frequent triggering and enhance the closed-loop operating stability, the following auxiliary conditions are further imposed:

(1) Minimum intertrigger interval constraint. The time interval between any two adjacent triggering instants is not smaller than a preset number of steps.

(2) Forced triggering condition. When the image features approach the preset triggering boundary, when the orientation deviation becomes too large, or when the remaining length of the current control plan is insufficient, a new optimization is triggered immediately. The triggering boundary is set inside the optimization constraint boundary so that optimization starts before the state approaches the constraint boundary.

(3) Vision validity check. Before trigger evaluation, the current visual measurement is checked for validity. In this paper, a visual measurement is considered valid only if the vision module provides an image-moment measurement and the controller receives it successfully. Otherwise, the measurement is treated as invalid and is excluded from deviation calculation and trigger evaluation. During short-term visual loss, the controller continues to execute the remaining historical control plan; if the visual loss exceeds the allowable duration, the controller switches to the safe-holding mode.

(4) Solver anomaly handling. When the optimizer fails to return a feasible solution while the historical control plan is still valid, the remaining control sequence continues to be applied. If the solver fails continuously or if the historical plan becomes invalid, the update is stopped and the system enters protection mode.

The execution procedure of ETIM-NMPC at each discrete instant is summarized in Algorithm 1. According to the current visual and state information, the controller determines whether to update the NMPC solution or reuse the historical control plan, and switches to the safe-holding mode when abnormal conditions occur.

Algorithm 1. Execution Procedure of ETIM-NMPC.

Input: image features $\mathbf{s}$, velocity states $\boldsymbol{\nu}_r$, weighting matrices $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{P}$, historical control plan, threshold parameters ρ, ε, and c, minimum inter-trigger interval, and forced triggering conditions.

1: Acquire $\mathbf{s}$ and $\boldsymbol{\nu}_r$.
2: Construct the joint state vector $\boldsymbol{x}$.
3: **If** visual information is invalid **then**
4: **If** the invalid duration exceeds the allowable limit or the historical plan is invalid **then**
5: switch to the safe-holding mode;
6: **else**
7: reuse the historical control plan;
8: **end if**
9: **else**
10: Compute the state deviation and nominal cost;
11: Construct the triggering threshold γ according to (34);
12: Check the minimum inter-trigger interval and forced triggering conditions;
13: **If** the triggering condition in (35) is satisfied **then**
14: solve the NMPC problem and update the predicted trajectory and control sequence;
15: **else**
16: reuse the historical control plan;
17: **end if**
18: **if** continuous solver failure occurs or the historical plan becomes invalid **then**
19: switch to the safe-holding mode;
20: **end if**
21: **end if**

Output: updated control sequence $\mathcal{T}_k$, reused historical control plan, or safe-holding command.

Remark 1. *The event-triggered mechanism changes only the NMPC update instants, while the constrained prediction model and the optimal control sequence generation remain unchanged. When the triggering condition is not satisfied, the remaining elements of the previously optimized control sequence are executed over a limited interval. This design follows the general event-triggered NMPC framework, where recursive feasibility, terminal constraints, Lyapunov functions, or ISS arguments are commonly used for formal stability analysis [20,21]. In the present implementation, invalid visual measurements, persistent visual loss, solver anomalies, and an exhausted control plan are handled by the protective reuse and safe-holding procedures in Algorithm 1. These safeguards mitigate the implementation risks but do not constitute a formal recursive-feasibility or ISS-type stability proof. The rigorous stability analysis under non-uniform update intervals will be investigated in future work.*

4. Simulation Experiments

To verify the effectiveness of the proposed method, joint simulation experiments were carried out using WEBOTS 2025a (Cyberbotics) and MATLAB 2025a (MathWorks). The vision program was implemented using Python 3.9.23 and OpenCV 4.8.1. WEBOTS was used to build the ROV dynamic model, underwater scene, and visual environment; Python was used for visual recognition and image-moment extraction; and MATLAB was used for controller implementation and data processing. MATLAB communicated with WEBOTS through the Remote API, while the Python-based vision program communicated with MATLAB through a local TCP socket. The simulation environment is shown in Figure 4.

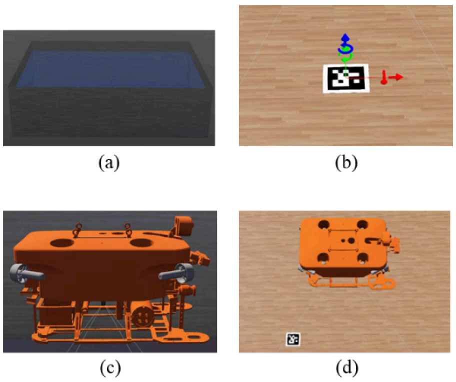

Figure 4. WEBOTS simulation environment. (**a**) Pool. (**b**) Target. (**c**) ROV. (**d**) Hovering.

In this paper, three image-moment-based controllers were compared in the simulations: IM-PID, IM-NMPC, and ETIM-NMPC. IM-PID was used as a conventional baseline, IM-NMPC was used as the time-triggered NMPC comparison method, and ETIM-NMPC was the proposed event-triggered NMPC method. The prediction model of the NMPC-based controllers was constructed in the reduced control-relevant subspace, whereas the WEBOTS simulation plant retained the six-degree-of-freedom motion capability and thruster-driven dynamics of the ROV.

Four simulation scenarios were designed: the Ideal scenario, the Disturbed scenario, the Dynamic-target scenario, and the Complex scenario. The Ideal scenario corresponds to a stationary target without current disturbance. The Disturbed scenario introduces current disturbance for a stationary target. The Dynamic-target scenario considers a moving target without current disturbance. The Complex scenario combines current disturbance and target motion. The key environmental parameters of the WEBOTS simulation platform and the main dynamic parameters of the ROV are listed in Tables 1 and 2, respectively. The main dynamic parameters and hydrodynamic coefficients of the ROV in Table 2 were adopted from the corresponding ROV parameter settings reported in [14], since the same ROV platform is considered in this paper.

Table 1. Key environmental parameters of WEBOTS.

Parameter	Value
Pool size	100 m × 50 m × 20 m
Simulation step	32 ms
Drag Force Coefficients	(0.46, 0.6, 2)
Drag Torque Coefficients	(0.001, 0.005, 0.001)
Viscous Resistance Torque Coefficient	0.005

Table 2. Main dynamic parameters of the R-ROV.

Parameter	Value	Parameter	Value
m	170 kg	$Y_{v\|v\|}$	158.2853 kg/m
I_{zz}	$10.07\ \mathrm{kg}\cdot\mathrm{m}^2$	Z_w	20 kg/s
X_u	2.39 kg/s	$Z_{w\|w\|}$	150 kg/m
$X_{u\|u\|}$	15.83 kg/m	N_r	$0.1\ \mathrm{kg}\cdot\mathrm{m}^2/(\mathrm{s}\times\mathrm{rad})$
Y_v	23.90 kg/s	$N_{r\|r\|}$	$0.01\ \mathrm{kg}\cdot\mathrm{m}^2/\mathrm{rad}^2$

The finite-horizon optimal control problem was formulated in CasADi and solved using the IPOPT solver. In the simulations, the WEBOTS basic time step was T = 32 ms, the NMPC discrete time step was dt = 0.1 s, and the prediction horizon was N_p = 10. The prediction model was discretized using the fourth-order Runge–Kutta method. The other NMPC controller parameters are set as follows: state weighting matrix **Q** = diag(240, 240, 260, 60, 1, 1, 2, 1), input weighting matrix **R** = diag(1, 1, 1.8, 1), and terminal weighting matrix **P** = diag(480, 480, 650, 120, 1, 1, 3, 1). The initial state of the ROV (x_0, y_0, z_0, ψ_0) = (0.7 m, 0.5 m, 1.6 m, 0.2618 rad), the desired state (x^*, y^*, z^*, ψ^*) = (0.0 m, 0.0 m, 1.0 m, 0.0000 rad), the corresponding desired image features (s_1*, s_2*, s_3*, s_4*) = (0, 0, 0.9, 0) and a unified simulation duration of 120 s were kept the same in all four scenarios, thus providing a unified reference for the comparison of the experimental results.

The relative-position root mean square error (RMSE) was used as the main quantitative index for evaluating the hovering accuracy. In this simulation study, hovering was considered effective when the translational RMSEs of ETIM-NMPC remained below 0.15 m in the evaluated directions. This criterion was used as a simulation-level accuracy reference, rather than as a physical ROV experimental validation threshold. In addition, image-feature errors, relative-position and yaw responses, control inputs, Cartesian-space trajectories, and computation-performance indices were used to compare the control behavior and online computational burden of the three controllers.

4.1. Ideal Scenario

In the Ideal scenario, the target was fixed and no current disturbance was applied. This scenario was used as the basic hovering-control case to evaluate the convergence behavior and control performance of the three image-moment-based controllers under nominal simulation conditions.

The image-feature errors, relative-position and yaw responses, control inputs, and Cartesian-space trajectories are shown in Figures 5–8. All three controllers drive the ROV toward the desired hovering state. Compared with IM-PID, the two NMPC-based controllers show smaller image-feature fluctuations, more coordinated relative-position responses, and more concentrated Cartesian trajectories near the final hovering region. The responses of ETIM-NMPC remain close to those of IM-NMPC, indicating that the event-triggered update strategy does not significantly deteriorate the nominal hovering response.

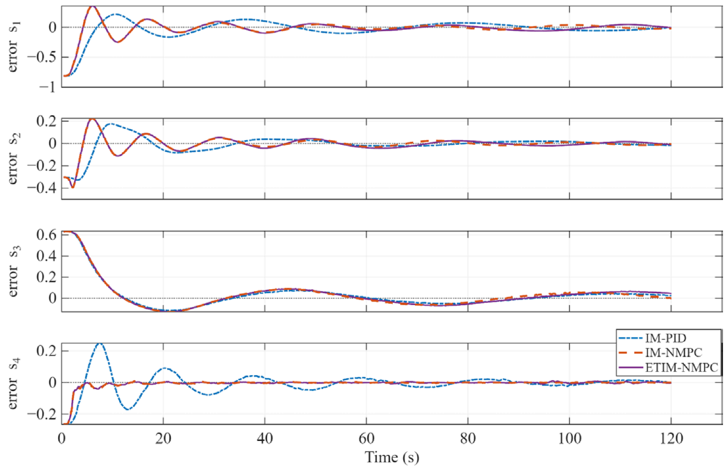

Figure 5. Image feature error—Ideal scenario.

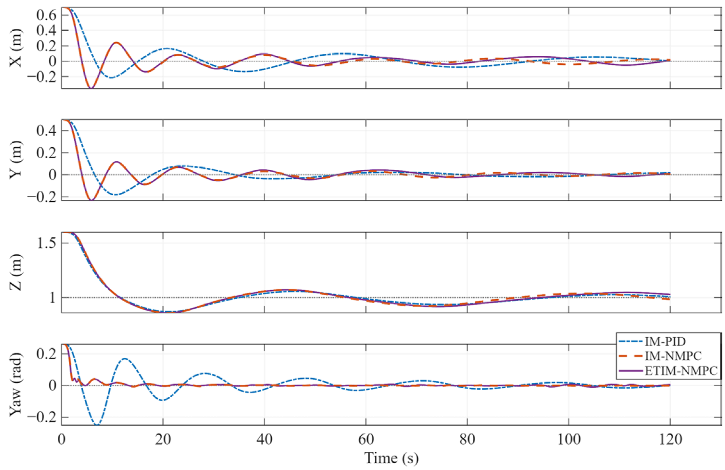

Figure 6. Relative position and yaw angle–Ideal scenario.

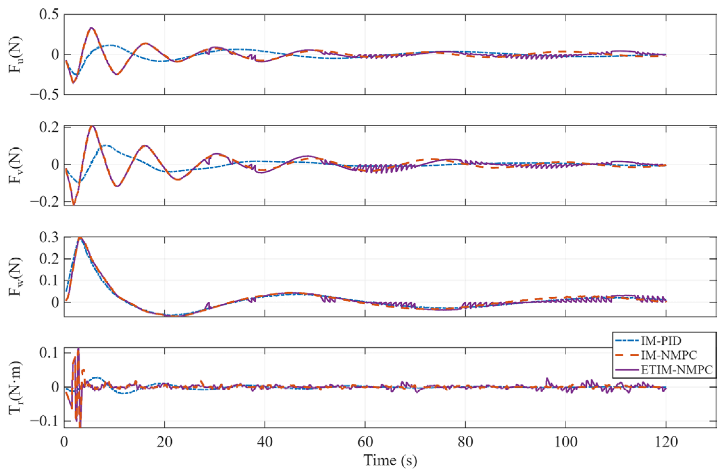

Figure 7. Variation in ROV control inputs—Ideal scenario.

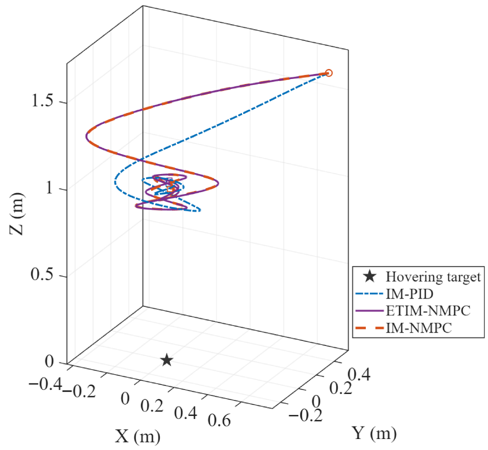

Figure 8. Cartesian-space trajectory—Ideal scenario.

The position RMSEs of the three controllers in the Ideal scenario are summarized in Table 3. For the proposed ETIM-NMPC method, the RMSEs in the x, y, and z directions are 0.1200 m, 0.0801 m, and 0.1299 m, respectively, all of which remain below the simulation-level hovering accuracy reference of 0.15 m. Moreover, its overall control performance remains close to that of IM-NMPC. IM-NMPC achieves the smallest RMSEs in the x and y directions, while ETIM-NMPC yields slightly larger but comparable values. A local metric crossing is observed in the z direction, where IM-PID produces the smallest RMSE. This is related to the relatively weak coupling of the heave-related channel and the direct loop tuning of the PID baseline. In contrast, the NMPC-based controllers optimize a weighted multivariable constrained objective and do not independently minimize the z-direction RMSE. Therefore, this local crossing should not be interpreted as the overall superiority of IM-PID.

Table 3. Position RMSEs of different algorithms—Ideal scenario.

Method	Rel-x (m)	Rel-y (m)	Rel-z (m)
IM-PID	0.1382	0.0922	0.1228
IM-NMPC	0.1170	0.0786	0.1287
ETIM-NMPC	0.1200	0.0801	0.1299

To examine the influence of the event-triggering parameters, a brief sensitivity analysis was conducted under the Ideal scenario. According to the event-triggered mechanism defined in Section 3.3, the sensitivity test mainly adjusts ρ, ε, and $\mathbf{W}$, while the proportional coefficient c in Equation (35) was kept unchanged as $c = 1$. The results are listed in Table 4.

Table 4. Sensitivity analysis of event-triggering parameters under the Ideal scenario.

Configuration	Parameter Variation	Trigger Rate (%)	Cumulative Computation Time (s)	RMSE-x (m)	RMSE-y (m)	RMSE-z (m)	X Settling Time (s)	Y Settling Time (s)
Configuration 1	$\rho = 0.015$; $\varepsilon = 0.0020$; $\mathbf{W} = \mathrm{diag}([20, 20, 25, 0, 2, 2, 2, 0])$	56.08	13.602850	0.1205	0.0803	0.1299	112.704	31.808
Configuration 2	$\rho = 0.012$; $\varepsilon = 0.0015$; $\mathbf{W} = \mathrm{diag}([20, 20, 25, 0, 2, 2, 2, 0])$	57.42	12.298212	0.1197	0.0797	0.1296	111.680	24.864
Configuration 3	$\rho = 0.012$; $\varepsilon = 0.0015$; $\mathbf{W} = \mathrm{diag}([28, 22, 25, 0, 2, 2, 2, 0])$	58.58	12.169543	0.1190	0.0798	0.1299	51.360	32.320

The three configurations produce RMSEs in a similar position, while the trigger rate, cumulative computation time, and settling behavior vary with ρ, ε, and $\mathbf{W}$. Configuration 3 gives the smallest RMSE-x and clearly reduces the x-direction settling time while maintaining a comparable cumulative computation time. Therefore, Configuration 3 was selected as the unified event-triggering parameter setting for the four main scenarios. This analysis provides practical tuning guidance, rather than proof of global optimality.

The computation performance under the Ideal scenario is summarized in Table 5. IM-NMPC solves the finite-horizon optimization problem at every control step, whereas ETIM-NMPC updates the optimal control sequence only when the triggering condition is satisfied. As a result, the number of optimization calls decreases from 1200 to 703, and the cumulative computation time decreases from 21.510327 s to 12.891672 s.

Table 5. Computation performance under the Ideal scenario.

Method	Triggered Times	Trigger Rate (%)	Average Computation Time (s)	Peak Computation Time (s)	Cumulative Computation Time (s)
IM-NMPC	1200	100.00	0.017925	0.184945	21.510327
ETIM-NMPC	703	58.58	0.010743	0.156174	12.891672

These results indicate that the computational benefit of ETIM-NMPC mainly comes from reducing online optimization updates and the cumulative computation time, rather than accelerating each individual optimization solve.

4.2. Disturbed Scenario

In the Disturbed scenario, a current disturbance was introduced while the target remained fixed. The current velocity was set to $[v_x, v_y, v_z] = [0.15, 0.10, 0]$ m/s, and the controller parameters were kept the same as those used in the Ideal scenario. This scenario was used to evaluate the hovering performance of the three controllers under external current disturbance.

The image-feature errors, relative-position and yaw responses, control inputs, and Cartesian-space trajectories are shown in Figures 9–12. Under current disturbance, all three controllers can still drive the ROV toward the desired hovering region. Compared with IM-PID, the two NMPC-based controllers show smaller horizontal-position deviations and more coordinated responses. The response of ETIM-NMPC remains close to that of IM-NMPC, while its control inputs remain bounded without sustained high-frequency oscillations.

The position RMSEs of the three controllers in the Disturbed scenario are summarized in Table 6. For the proposed ETIM-NMPC method, the RMSEs in the x, y, and z directions are 0.1171 m, 0.0995 m, and 0.1297 m, respectively, all of which remain below the simulation-level hovering accuracy reference of 0.15 m. Compared with IM-PID, ETIM-NMPC reduces the x- and y-direction RMSEs. Compared with IM-NMPC, ETIM-NMPC yields slightly larger but still comparable RMSEs.

Similar to the Ideal scenario, a local metric crossing is observed in the z direction, where IM-PID produces the smallest RMSE. This local result does not indicate the overall superiority of IM-PID, because the NMPC-based controllers optimize a weighted multivariable constrained objective, rather than independently minimizing the z-direction RMSE. In addition, the slightly lower x-direction RMSEs of IM-NMPC and ETIM-NMPC under current disturbance should be interpreted as a weak local metric crossing, rather than as evidence that current disturbance improves the control task.

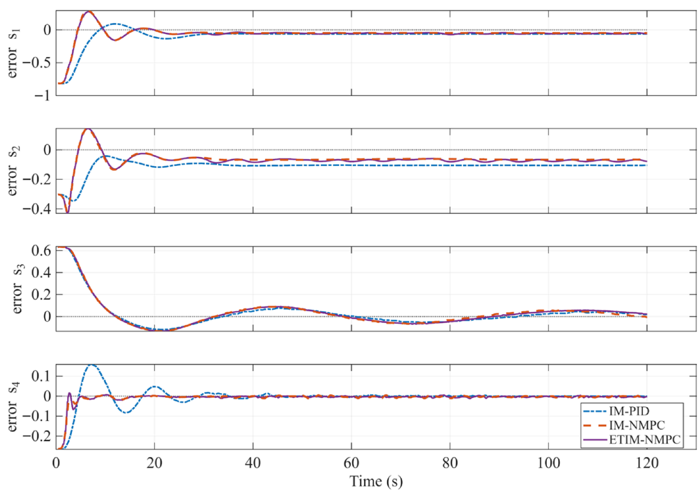

Figure 9. Image feature error—Disturbed scenario.

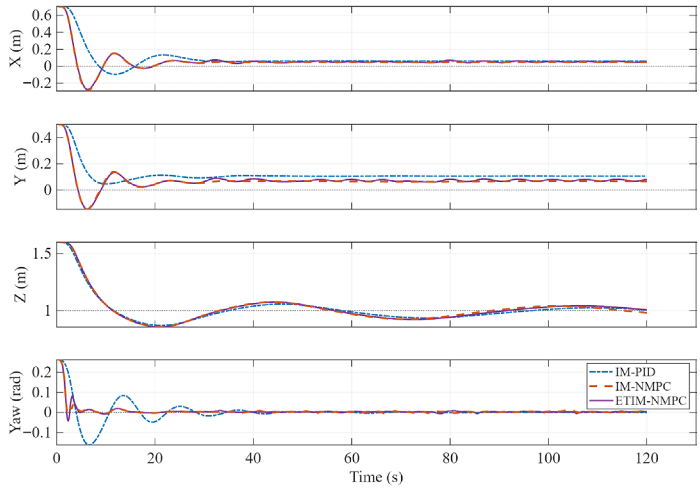

Figure 10. Relative position and yaw angle—Disturbed scenario.

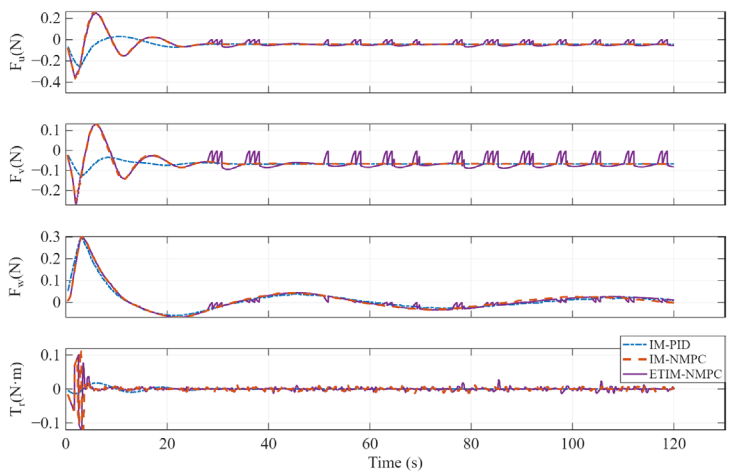

Figure 11. Variation in ROV control inputs—Disturbed scenario.

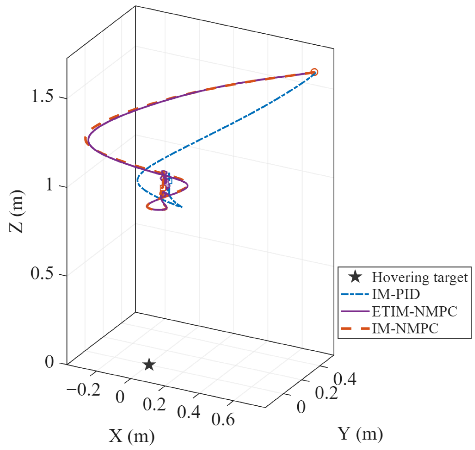

Figure 12. Cartesian-space trajectory—Disturbed scenario.

Table 6. Position RMSEs of different algorithms—Disturbed scenario.

Method	Rel-x (m)	Rel-y (m)	Rel-z (m)
IM-PID	0.1427	0.1317	0.1223
IM-NMPC	0.1160	0.0954	0.1292
ETIM-NMPC	0.1171	0.0995	0.1297

The computation performance under the Disturbed scenario is summarized in Table 7. The number of optimization calls of ETIM-NMPC decreases from 1200 to 940, and the cumulative computation time decreases from 20.515142 s to 16.525616 s. Compared with the Ideal scenario, the trigger rate increases because the current disturbance causes more frequent deviations from the nominal predicted trajectory.

Table 7. Computation performance under the Disturbed scenario.

Method	Triggered Times	Trigger Rate (%)	Average Computation Time (s)	Peak Computation Time (s)	Cumulative Computation Time (s)
IM-NMPC	1200	100.00	0.017096	0.158365	20.515142
ETIM-NMPC	940	78.33	0.013771	0.161474	16.525616

These results show that ETIM-NMPC still reduces the number of online optimization updates and the cumulative computation time under current disturbance. Since the peak computation time of ETIM-NMPC is slightly higher than that of IM-NMPC in this scenario, the computational benefit should still be interpreted as a reduced update frequency and accumulated computation time, rather than the faster execution of each individual optimization solve.

To further evaluate the influence of degraded visual feedback, an additional simulation-level visual degradation test was conducted under a representative combined-degradation condition. The combined degradation includes degraded camera appearance and intermittent visual measurement loss. The comparison between the nominal visual condition and the combined-degradation condition is shown in Figure 13.

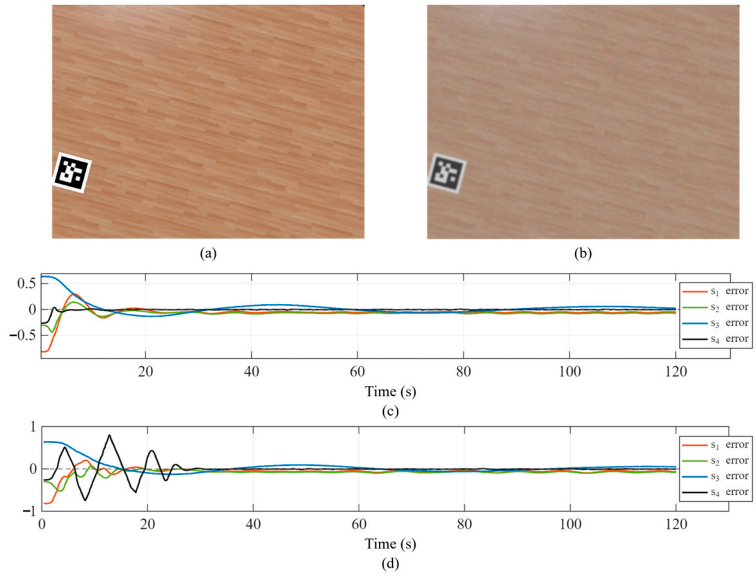

Figure 13. Visual degradation robustness comparison. (**a**) Initial camera view under the nominal visual condition. (**b**) Initial camera view under the combined-degradation condition. (**c**) Image-feature error response under the nominal visual condition. (**d**) Image-feature error response under the combined-degradation condition.

To quantify the availability of visual measurements, the valid visual measurement ratio η_v is defined as

$$\eta_v = \frac{N_{\text{valid}}}{N_{\text{total}}} \tag{36}$$

where N_{valid} denotes the number of control instants at which the image-moment measurement is successfully generated by the vision module and received by the controller, and N_{total} denotes the total number of control instants. The performance comparison under the nominal and combined-degradation visual conditions is summarized in Table 8.

Table 8. Performance comparison under nominal and combined-degradation visual conditions.

Condition	η_v	Triggered Times	Trigger Rate (%)	Average Computation Time (s)	Peak Computation Time (s)	Cumulative Computation Time (s)	RMSE-x (m)	RMSE-y (m)	RMSE-z (m)	Yaw RMSE (rad)
Nominal visual condition	0.999733	939	78.25	0.016961	0.157206	15.926149	0.1186	0.0997	0.1302	0.0263
Combined degradation	0.705945	1009	84.08	0.016332	0.156077	16.479129	0.1197	0.0978	0.1313	0.3011

The valid visual measurement ratio decreases from 0.999733 to 0.705945 under combined degradation. Meanwhile, the trigger rate increases from 78.25% to 84.08%, and the cumulative computation time increases from 15.926149 s to 16.479129 s. The position RMSEs remain close under the two visual conditions, while the yaw RMSE increases from 0.0263 rad to 0.3011 rad. These results indicate that the visual degradation mainly affects the orientation-related response and induces more frequent optimization updates. This test provides representative simulation-level evidence of visual-degradation robustness, rather than a complete validation under all real underwater visual conditions.

4.3. Dynamic-Target Scenario

In the Dynamic-target scenario, the target moved along a circular trajectory without current disturbance. This scenario was used to evaluate the hovering performance of the three controllers under a time-varying visual reference. The target trajectory was defined as follows:

$$\begin{aligned} x_t &= x_{t0} + 0.15\cos(2\pi \times 0.02t) \\ y_t &= y_{t0} + 0.15\sin(2\pi \times 0.02t), t = [T, 2T, \ldots, NT] \\ z_t &= z_{t0} \end{aligned} \tag{37}$$

Here, (x_t, y_t, z_t) denotes the target position, (x_{t0}, y_{t0}, z_{t0}) denotes the center of the circular trajectory, 0.15 m is the motion radius, and 0.02 Hz is the motion frequency. Therefore, this scenario represents a mild circular moving-reference case around the hovering region, rather than an aggressive target-tracking benchmark.

The image-feature errors, relative-position and yaw responses, control inputs, and Cartesian-space trajectories are shown in Figures 14–17. Under the moving visual reference, all three controllers keep the main feature errors bounded. Compared with IM-PID, the two NMPC-based controllers show smaller horizontal relative-position deviations and trajectories closer to the target trajectory. The response of ETIM-NMPC remains generally close to that of IM-NMPC, although slightly larger tracking errors appear in some intervals because the optimal control sequence is not updated at every control step.

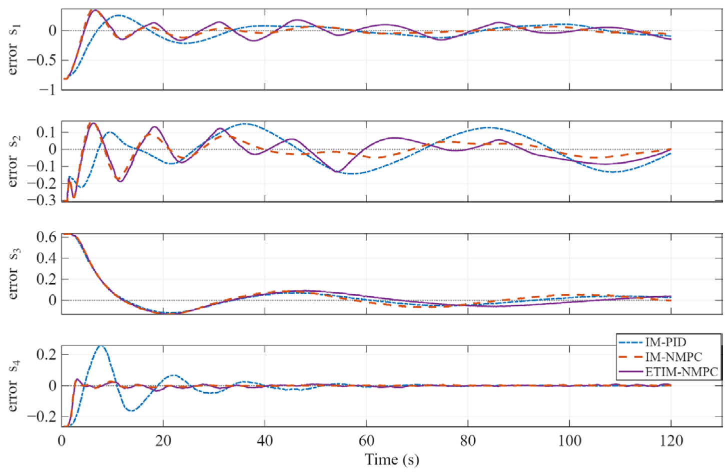

Figure 14. Image feature error—Dynamic-target scenario.

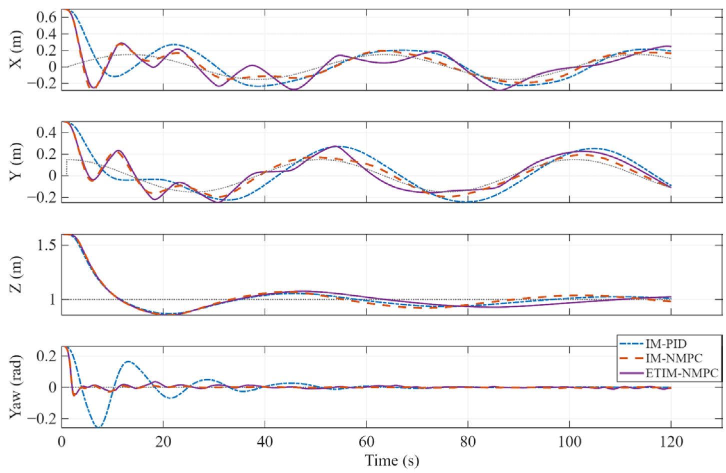

Figure 15. Relative position and yaw angle—Dynamic-target scenario.

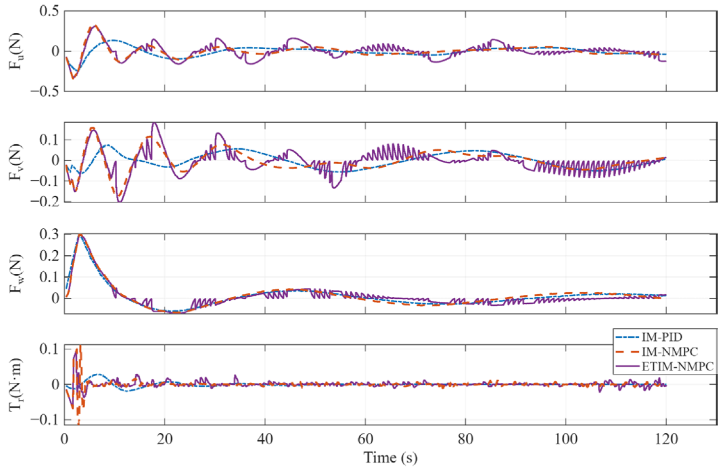

Figure 16. Variation in ROV control inputs—Dynamic-target scenario.

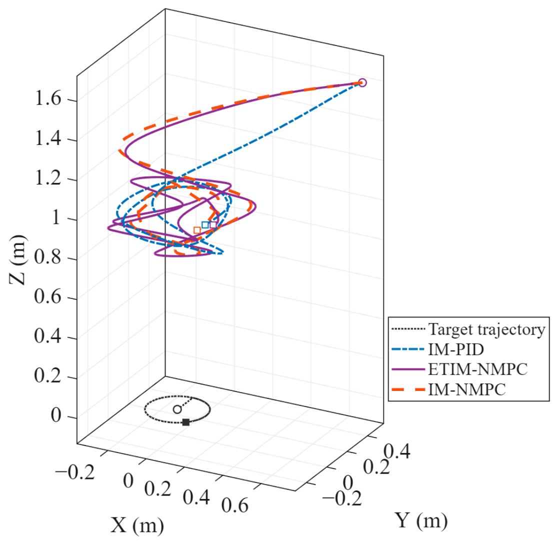

Figure 17. Cartesian-space trajectory with target trajectory—Dynamic-target scenario.

The position RMSEs of the three controllers in the Dynamic-target scenario are summarized in Table 9. For the proposed ETIM-NMPC method, the RMSEs in the x, y, and z directions are 0.1363 m, 0.0846 m, and 0.1290 m, respectively, all of which remain below the simulation-level hovering accuracy reference of 0.15 m. Compared with IM-PID, ETIM-NMPC reduces the x- and y-direction RMSEs. Compared with IM-NMPC, ETIM-NMPC yields larger but still bounded RMSEs.

Table 9. Position RMSEs of different algorithms—Dynamic-target scenario.

Method	Rel-x (m)	Rel-y (m)	Rel-z (m)
IM-PID	0.1496	0.1079	0.1232
IM-NMPC	0.1166	0.0715	0.1282
ETIM-NMPC	0.1363	0.0846	0.1290

The computation performance under the Dynamic-target scenario is summarized in Table 10. The number of optimization calls of ETIM-NMPC decreases from 1200 to 545, and the cumulative computation time decreases from 24.574179 s to 9.623686 s.

Table 10. Computation performance under the Dynamic-target scenario.

Method	Triggered Times	Trigger Rate (%)	Average Computation Time (s)	Peak Computation Time (s)	Cumulative Computation Time (s)
IM-NMPC	1200	100.00	0.020478	0.153533	24.574179
ETIM-NMPC	545	45.42	0.008020	0.154517	9.623686

These results show that ETIM-NMPC reduces the online optimization updates and cumulative computation time in the Dynamic-target scenario. Since the peak computation time of ETIM-NMPC is slightly higher than that of IM-NMPC, this improvement should still be interpreted as a reduced update frequency and accumulated computation time. Overall, ETIM-NMPC maintains acceptable hovering performance under a mild moving-reference condition while reducing the online computational burden. More aggressive target motions are beyond the scope of this scenario and will be considered in future work.

4.4. Complex Scenario

In the Complex scenario, the current disturbance and target motion were introduced simultaneously. The current velocity was set to $[v_x, v_y, v_z]$ = [0.15, 0.10, 0] m/s, and the target followed the same circular trajectory as that used in the Dynamic-target scenario. This scenario was used to evaluate the hovering performance of the three controllers under combined external disturbance and time-varying visual reference.

The image-feature errors, relative-position and yaw responses, control inputs, and Cartesian-space trajectories are shown in Figures 18–21. Compared with the previous scenarios, the Complex scenario imposes a higher regulation demand because the ROV must compensate for current disturbance while responding to the moving visual target. Compared with IM-PID, the two NMPC-based controllers show smaller horizontal-position deviations and trajectories that are closer to the target trajectory and the desired hovering region. The response of ETIM-NMPC remains generally close to that of IM-NMPC, while its control inputs remain bounded without sustained high-frequency oscillations.

The position RMSEs of the three controllers in the Complex scenario are summarized in Table 11. For the proposed ETIM-NMPC method, the RMSEs in the x, y, and z directions are 0.1374 m, 0.1025 m, and 0.1311 m, respectively, all of which remain below the simulation-level hovering accuracy reference of 0.15 m. Compared with IM-PID, ETIM-NMPC reduces the x- and y-direction RMSEs. Compared with IM-NMPC, ETIM-NMPC yields slightly larger but still bounded RMSEs.

A local metric crossing is again observed in the z direction, where IM-PID achieves the smallest RMSE. This result is limited to one local component of the multivariable hovering performance and should not be interpreted as the overall superiority of IM-PID.

The computation performance under the Complex scenario is summarized in Table 12. The number of optimization calls of ETIM-NMPC decreases from 1200 to 580, and the cumulative computation time decreases from 24.782852 s to 12.358048 s.

These results show that ETIM-NMPC reduces the online optimization updates and cumulative computation time under the combined disturbance and moving-target condition. Since the peak computation time of ETIM-NMPC is higher than that of IM-NMPC, this improvement should still be interpreted as a reduced update frequency and accumulated online computation, rather than faster execution of each individual optimization solve.

Overall, the four simulation scenarios show that ETIM-NMPC maintains a hovering performance that is close to that of IM-NMPC while reducing the number of online optimization updates. The proposed event-triggered mechanism therefore provides a bounded performance–efficiency trade-off: the control performance remains within the simulation-level hovering accuracy reference, while the cumulative online computational burden is reduced. This conclusion is based on joint simulation results and should not be regarded as a substitute for physical ROV validation under real underwater conditions.

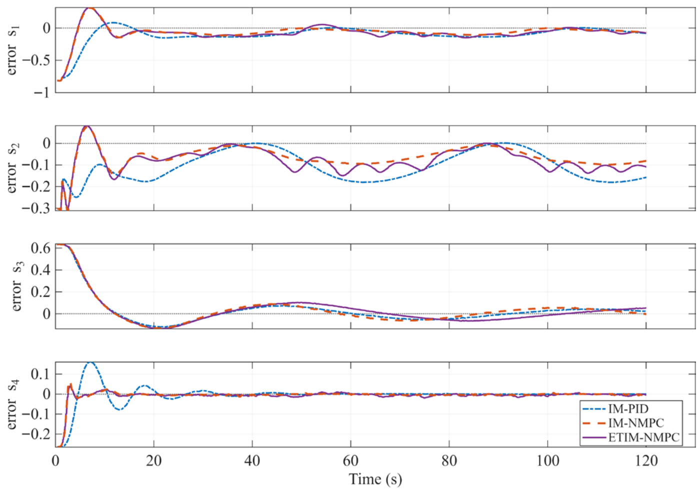

Figure 18. Image feature error—Complex scenario.

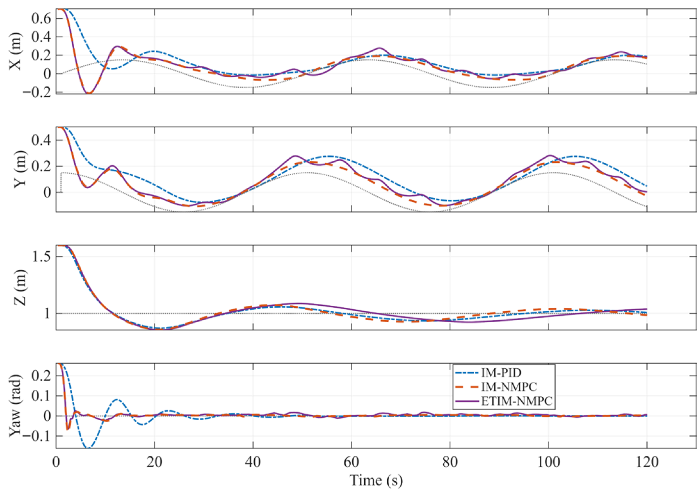

Figure 19. Relative position and yaw angle—Complex scenario.

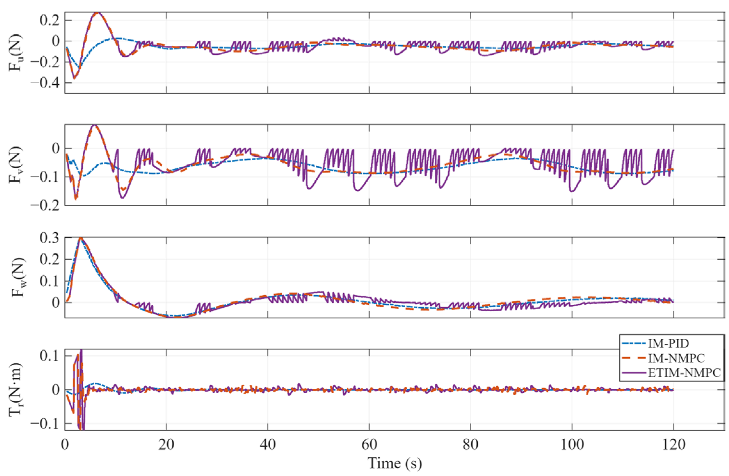

Figure 20. Variation in ROV control inputs—Complex scenario.

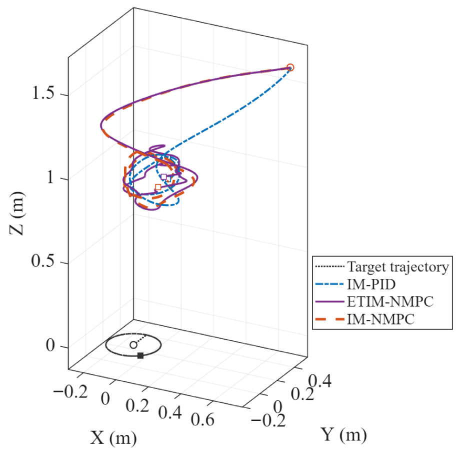

Figure 21. Cartesian-space trajectory with target trajectory—Complex scenario.

Table 11. Position RMSEs of different algorithms—Complex scenario.

Method	Rel-x (m)	Rel-y (m)	Rel-z (m)
IM-PID	0.1567	0.1347	0.1231
IM-NMPC	0.1268	0.0878	0.1287
ETIM-NMPC	0.1374	0.1025	0.1311

Table 12. Computation performance under the Complex scenario.

Method	Triggered Times	Trigger Rate (%)	Average Computation Time (s)	Peak Computation Time (s)	Cumulative Computation Time (s)
IM-NMPC	1200	100.00	0.020652	0.164515	24.782852
ETIM-NMPC	580	48.33	0.010298	0.182237	12.358048

5. Conclusions

To reduce the online optimization burden of vision-based ROV hovering control while preserving the constraint-handling capability, this paper proposed an event-triggered image-moment nonlinear model predictive control method. The proposed ETIM-NMPC method uses image moments as visual feedback features, constructs the prediction model in a reduced control-relevant subspace, and updates the optimal control sequence only when the event-triggering condition is satisfied.

Joint simulation experiments were conducted under the Ideal, Disturbed, Dynamic-target, and Complex scenarios. The results show that ETIM-NMPC achieved effective hovering performance in the joint simulation environment, with the translational RMSEs remaining below the simulation-level hovering accuracy reference of 0.15 m in all evaluated directions. Compared with IM-PID, ETIM-NMPC generally reduced the horizontal relative-position errors. Compared with IM-NMPC, it produced slightly larger errors in some local directions, but maintained comparable overall control performance.

Furthermore, the computation-performance results confirm the main advantage of the event-triggered design. The trigger rates of ETIM-NMPC were 58.58%, 78.33%, 45.42%, and 48.33% in the Ideal, Disturbed, Dynamic-target, and Complex scenarios, respectively. In all four scenarios, ETIM-NMPC reduced the number of online optimization updates and the cumulative computation time compared with IM-NMPC. Therefore, its computational benefit mainly comes from reducing the update frequency and accumulated online computation, rather than accelerating every individual optimization solve. The event-triggering sensitivity analysis further provides a practical tuning reference, and the combined visual-degradation test gives representative simulation-level evidence for evaluating degraded visual feedback.

Nevertheless, the present validation is still limited to joint simulation. Future work will focus on physical pool and sea experiments under realistic sensing, communication, hydrodynamic, and actuator uncertainties. In addition, rigorous recursive-feasibility and ISS-type stability analysis under non-uniform update intervals, as well as the practical upper bound of trackable target dynamics under faster target motion, will be further investigated. The proposed event-triggered predictive-control framework may also be extended to broader autonomous marine vehicle tasks that require visual feedback, real-time constraint handling, disturbance rejection, and limited onboard computational resources.

Author Contributions: Conceptualization, Z.S., Y.Z. and S.C.; methodology, Z.S., Y.Z. and S.C.; writing—original draft preparation, Z.S.; writing—review and editing, Y.Z., S.C., P.Z. and S.W.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Project of the Shenyang Institute of Automation, Chinese Academy of Sciences, grant number 2023JC3K05.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data can be obtained from the corresponding author.

Acknowledgments: The authors would like to thank the reviewers for their valuable comments and suggestions, which helped to improve the quality of this manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Capocci, R.; Dooly, G.; Omerdic, E.; Coleman, J.; Newe, T.; Toal, D. Inspection-Class Remotely Operated Vehicles—A Review. *J. Mar. Sci. Eng.* **2017**, *5*, 13. [CrossRef]
2. Sorensen, A.J. A survey of dynamic positioning control systems. *Annu. Rev. Control* **2011**, *35*, 123–136. [CrossRef]
3. Hutchinson, S.; Hager, G.D.; Corke, P.I. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.* **1996**, *12*, 651–670. [CrossRef]
4. Huang, H.; Bian, X.Y.; Cai, F.C.; Li, J.Y.; Jiang, T.; Zhang, Z.K.; Sun, C.Y. A review on visual servoing for underwater vehicle manipulation systems automatic control and case study. *Ocean Eng.* **2022**, *260*, 112065. [CrossRef]
5. Gao, J.; Proctor, A.A.; Shi, Y.; Bradley, C. Hierarchical Model Predictive Image-Based Visual Servoing of Underwater Vehicles with Adaptive Neural Network Dynamic Control. *IEEE Trans. Cybern.* **2016**, *46*, 2323–2334. [CrossRef]
6. Nourmohammadi, A.; Loueipour, M.; Hosseinnajad, A. Image-based Visual Servoing for Dynamic Positioning of a Work-Class ROV with Depth Computation. In Proceedings of the 9th International Conference on Control, Instrumentation and Automation, ICCIA 2023, Tehran, Iran, 20–21 December 2023.
7. Jabbari, H.; Oriolo, G.; Bolandi, H. Dynamic IBVS Control of an Underactuated UAV. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China, 11–14 December 2012.
8. Cao, Z.Q.; Chen, X.C.; Yu, Y.Y.; Yu, J.Z.; Liu, X.L.; Zhou, C.; Tan, M. Image Dynamics-Based Visual Servoing for Quadrotors Tracking a Target with a Nonlinear Trajectory Observer. *IEEE Trans. Syst. Man Cybern.-Syst.* **2020**, *50*, 376–384. [CrossRef]
9. Hong, L.; Wang, X.; Zhang, D. Robust hybrid visual servoing for hovering control of autonomous underwater vehicles in unstructured environments. *Ocean Eng.* **2025**, *339*, 122103. [CrossRef]
10. Li, J.H.; Sun, J.F.; Liu, L.Q.; Xu, J.S. Model predictive control for the tracking of autonomous mobile robot combined with a local path planning. *Meas. Control* **2021**, *54*, 1319–1325. [CrossRef]
11. Mayne, D.Q. Model predictive control: Recent developments and future promise. *Automatica* **2014**, *50*, 2967–2986. [CrossRef]
12. Zhang, K.W.; Shi, Y.; Sheng, H.Y. Robust Nonlinear Model Predictive Control Based Visual Servoing of Quadrotor UAVs. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 700–708. [CrossRef]
13. Ni, T.; Sima, C.; Li, S.B.; Zhang, L.D.; Wu, H.B.; Guo, J. Optimization of Trajectory Generation and Tracking Control Method for Autonomous Underwater Docking. *J. Mar. Sci. Eng.* **2024**, *12*, 1349. [CrossRef]
14. Dai, H.X.; Zhang, Y.X.; Cui, S.G.; Zheng, X.; Zhang, Q. Research on the Control Method for Remotely Operated Vehicle Active Docking with Autonomous Underwater Vehicles Based on GFSMO-NMPC. *J. Mar. Sci. Eng.* **2025**, *13*, 601. [CrossRef]

15. Fu, M.Y.; Wang, L.L. Finite-time coordinated path following control of underactuated surface vehicles based on event-triggered mechanism. *Ocean Eng.* **2022**, *246*, 110530. [CrossRef]
16. Wang, B.; Peng, J.; Zhou, J.; Zhao, L. Adaptive Event-Triggered Predictive Control for Agile Motion of Underwater Vehicles. *J. Mar. Sci. Eng.* **2025**, *13*, 1072. [CrossRef]
17. Zhang, Y.; Yao, J.; Qian, C. Event-Triggered Nonlinear Visual Predictive Control Strategy for Robots. *J. Intell. Robot. Syst.* **2025**, *111*, 80. [CrossRef]
18. Heshmati-Alamdari, S.; Karras, G.C.; Eqtami, A.; Kyriakopoulos, K.J. A Robust Self Triggered Image Based Visual Servoing Model Predictive Control Scheme for Small Autonomous Robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 5492–5497.
19. Heshmati-Alamdari, S.; Eqtami, A.; Karras, G.C.; Dimarogonas, D.V.; Kyriakopoulos, K.J. A Self-triggered Visual Servoing Model Predictive Control Scheme for Under-actuated Underwater Robotic Vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 3826–3831.
20. Aspragkathos, S.N.; Karras, G.C.; Kyriakopoulos, K.J. Event-Triggered Image Moments Predictive Control for Tracking Evolving Features Using UAVs. *IEEE Robot. Autom. Lett.* **2024**, *9*, 1019–1026. [CrossRef]
21. Wu, W.; Zhang, W.; Du, X.; Li, Z.; Wang, Q. Homing tracking control of autonomous underwater vehicle based on adaptive integral event-triggered nonlinear model predictive control. *Ocean Eng.* **2023**, *277*, 114243. [CrossRef]
22. Luan, T.; Tan, Z.; Sun, M.; Wang, H.; Li, M.; Yao, H. Double-mode robust model predictive control of ship dynamic positioning system based on event-triggered mechanism. *Ocean Eng.* **2023**, *286*, 115536. [CrossRef]
23. Gao, J.; Proctor, A.; Bradley, C. Adaptive neural network visual servo control for dynamic positioning of underwater vehicles. *Neurocomputing* **2015**, *167*, 604–613. [CrossRef]
24. Källström, C.G. *Guidance and Control of Ocean Vehicles*; Fossen, T.I., Ed.; Wiley: Chichester, UK, 1996; ISBN 0-471-94113-1.
25. Zhang, Y.X.; Zhang, Q.F.; Zhang, A.Q.; Chen, J.; Li, X.G.; He, Z. Acoustics-Based Autonomous Docking for a Deep-Sea Resident ROV. *China Ocean Eng.* **2022**, *36*, 100–111. [CrossRef]
26. Liu, J.; Gao, J.; Yan, W. Lyapunov-based Model Predictive Visual Servo Control of an Underwater Vehicle-Manipulator System. *IEEE Trans. Intell. Veh.* **2024**, *10*, 2414–2426. [CrossRef]
27. Asl, H.J. Robust vision-based tracking control of VTOL unmanned aerial vehicles. *Automatica* **2019**, *107*, 425–432. [CrossRef]
28. Lai, N.; Chen, Y.; Liang, J.; He, B.; Zhong, H.; Zhang, H.; Wang, Y. Image Dynamics-Based Visual Servo Control for Unmanned Aerial Manipulator with a Virtual Camera. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 5264–5274. [CrossRef]
29. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley and Sons: Hoboken, NJ, USA, 2011.
30. Liu, J.; Gao, J.; Yan, W.S.; Chen, Y.M.; Yang, B. Image-based visual servoing of underwater vehicles for tracking a moving target using model predictive control with motion estimation. *Int. J. Veh. Des.* **2023**, *91*, 46–66. [CrossRef]