

Article

Visual Servoing of a Moving Target by an Unmanned Aerial Vehicle

Ching-Wen Chen, Hsin-Ai Hung, Po-Hung Yang and Teng-Hu Cheng * 

Department of Mechanical Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan; amyking90511@gmail.com (C.-W.C.); louise.me08g@nctu.edu.tw (H.-A.H.); s933611@hotmail.com (P.-H.Y.)

* Correspondence: tenghu@g2.nctu.edu.tw

Abstract: To track moving targets undergoing unknown translational and rotational motions, a tracking controller is developed for unmanned aerial vehicles (UAVs). The main challenges are to control both the relative position and orientation between the target and the UAV to within desired values, and to guarantee that the generated control input to the UAV is feasible (i.e., below its motion capability). Moreover, the UAV is controlled to ensure that the target always remains within the field of view of the onboard camera. These control objectives were achieved by developing a nonlinear-model predictive controller, in which the future motion of the target is predicted by quadratic programming (QP). Since constraints of the feature vector and the control input are considered when solving the optimal control problem, the control inputs can be bounded and the target can remain inside the image. Three simulations were performed to compare the efficacy and performance of the developed controller with a traditional image-based visual servoing controller.

Keywords: unscented kalman filter; quadratic programming; estimation; tracking of moving targets; UAV



Citation: Chen, C.-W.; Hung, H.-A.; Yang, P.-H.; Cheng, T.-H. Visual Servoing of a Moving Target by an Unmanned Aerial Vehicle. *Sensors* **2021**, *21*, 5708. <https://doi.org/10.3390/s21175708>

Received: 26 July 2021

Accepted: 20 August 2021

Published: 25 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Unmanned aerial vehicles (UAVs) have attracted much attention since the agility made them capable of adapting to diverse terrains and executing various tasks such as monitoring, rescue, and target tracking [1,2]. Studies of UAVs have focused on localizing a target from sensing data of cameras [3], radars [4], and sensor networks [5–7]. Due to the advantageous availability and cost of cameras, many approaches can reconstruct the environment by 2D image features. Algorithms for estimating the position of objects based on image features captured by stationary cameras [8] and moving cameras [9] have been presented. Methods have been developed to reconstruct the 3D model of an object from a perspective camera [10,11]. Other features such as lines [12], cylinders and spheres [13], or planes [14,15] are also applied to a structure-from-motion (SfM) task. However, the aforementioned SfM methods rely on triangulation to recover the depth, and extension of these works to the motion estimation of moving targets is still challenging.

Stationary targets: Image-based visual servoing (IBVS) is a system control method that guarantees a series of visual features of a target will converge to the desired setpoints in the image [16]. However, IBVS approaches have potential problems such as causing larger tracking errors or losing the tracking altogether, especially when the target motion is time-varying or not predicted correctly [17]. Visual predictive control (VPC) [18,19] is a method that combines model predictive control constraints such as in the field of view (FOV), output limitations of the actuator, and workspace. A nonlinear model predictive controller was applied to an underwater vehicle to generate a desired velocity while satisfying the constraints of the visibility [20]. Similarly, MPC has been used to control a mobile robot while tracking a stationary feature point [21], to keep a visual feature of the target in the curtain position of the image [22,23], and to maximize the visibility of a target and minimize the velocity of its image feature when utilizing quadrotors. Artificial patterns

have been used to predict all of the missing feature points due to occlusion problems and to ensure the execution of IBVS for navigation [24]. The aforementioned MPC approaches all considered only stationary targets, and so the present work designed an MPC to track a moving target, and moreover incorporated its motion estimation into the control sequence as a feedforward term in order to generate smooth control inputs.

Dynamic targets: Knowledge of motion of targets is crucial in tracking control; however, the knowledge is unknown and difficult to predict, and so various forms of image features are used for feedback control as to keep the target in FOV [25,26]. Many image processing methods have been developed for feature extraction and matching, such as RGB-based methods [27,28], scale-invariant feature-transform (SIFT) [29,30], and speeded-up robust-features (SURF) [31,32]. However, there are some drawbacks for these methods in terms of object detection and estimation of the relative motion between the camera and the target. Visual-based optimization methods have been applied a quadrotor [33], and it aimed to track a moving target while avoiding obstacles and the target position is assumed to be known in advance. Model-based optimization was applied to achieve perception and action objectives for robust sensing of a UAV [22]; however, the image features were extracted for indoor localization, rather than target tracking. Bounding boxes and feature points have been utilized as image features for tracking human beings [34–37], but the targets were moving at speeds that were so low that their motions could be ignored during tracking. Moreover, the orientation of the target is unavailable [38]. MPC has been applied to track a target moving along a periodic path, with this motion constraint reducing the complexity of the controller design [39]. However, the aforementioned works did not take the interaction between the target and the UAV into account, and the relative angle between the camera and the target was not modeled or measured.

This work models the interaction between a UAV and target and includes it in the developed controller, so that the tracking performance can be improved given certain motion, control, and FOV constraints. A DNN (e.g., YOLO [40]) is designed and trained to generate a bounding box in the image that encloses the target and predicts the angle of the target as viewed by the UAV, since both types of information are required to estimate target motion in an unscented Kalman filter (UKF). Finally, a high-order polynomial that emulates the target motion is obtained using quadratic programming (QP), which is used in the controller to predict the future motion of the target. The contributions of this work can be summarized as follows:

1. An optimal tracking controller is developed to track a moving target undergoing unknown motion while meeting motion, control, and sensing constraints, where the relative motion between the target and the UAV is estimated.
2. In contrast to previous approaches [34–38], this work models the dynamics of the relative rotation between the target and the UAV, and the relative angle can be controlled to a predefined desired value, which can be applied in applications such as the automatic searching, detection, and recognition of car license plate.
3. The controller is designed to ensure that the target remains within the FOV.
4. Compared to the IBVS controller, the developed controller can ensure smooth control input, less energy, and smaller tracking error.
5. The developed control architecture can be applied to track other moving targets as long as it can be detected by the YOLO network.

The organization of this work is as follows: Section 2 formulates the interaction between the target and the UAV as well as the control objectives. Section 3 describes how the target velocity is estimated based on the UKF and the bounding box in the image. Section 4 designs a controller to meet the control objectives. Simulations are presented in Section 5 to verify the efficacy of the developed controllers.

2. Preliminaries

Control Objectives

Figure 1 depicts the kinematics of a dynamic target and a camera fixed on a UAV, and the superscript \mathcal{G} and the subscript \mathcal{C} denote the inertial frame and the camera frame, respectively. The vector between the target and the camera can be expressed as

$$r_{q/c} = r_q - r_c, \quad (1)$$

where $r_q = [x_q, y_q, z_q]^T$ is defined as the target position, $r_c = [x_c, y_c, z_c]^T$ is the camera position, $r_{q/c} = [X, Y, Z]^T$ is defined in the camera frame in order to facilitate the subsequent analysis, and Z defined in $r_{q/c}$ is the depth $d \in \mathbb{R}$. Based on (1), the relative velocity $\dot{r}_{q/c}$ can be expressed as

$$\dot{r}_{q/c} = V_q - V_c - \omega_c \times r_{q/c}, \quad (2)$$

where $V_c \triangleq [v_{cx}, v_{cy}, v_{cz}]^T$ and $\omega_c \triangleq [\omega_{cx}, \omega_{cy}, \omega_{cz}]^T$ are defined as the translational and angular velocities of the camera. In (2), $V_q = [v_{qx}, v_{qy}, v_{qz}]^T$ denoted the translational velocity of the target is unknown and will be estimated, and V_c and ω_c are the control input to be designed in the subsequent sections.

The orientation of the target denoted as $\psi \in \mathbb{R}$ shown in Figure 2 can be expressed as:

$$\psi = \arctan\left(\frac{y_c^g - y_q^g}{x_c^g - x_q^g}\right) - \arctan\left(\frac{V_{qy}^g}{V_{qx}^g}\right), \quad (3)$$

where the superscript denotes the value defined in the global frame. Taking the time derivative on both sides of (3) yields the relative angular velocity between the target and the UAV:

$$\begin{aligned} \dot{\psi} = & \frac{(v_{cy}^g - v_{qy}^g)(x_c^g - x_q^g) - (v_{cx}^g - v_{qx}^g)(y_c^g - y_q^g)}{(x_c^g - x_q^g)^2 + (y_c^g - y_q^g)^2} \\ & - \frac{a_{qy}^g v_{qx}^g - a_{qx}^g v_{qy}^g}{(v_{qx}^g)^2 + (v_{qy}^g)^2}. \end{aligned} \quad (4)$$

The control objective is to achieve

$$r_{q/c} \rightarrow [0, 0, d_{des}]^T, \quad \psi \rightarrow \psi_{des}, \quad (5)$$

where implies that the target remains at the center of the image, and d_{des} and $\psi_{des} \in \mathbb{R}$ are the desired depth and angle, with respect to the target. The limitation of existing results is relaxed by making Assumption 1.

Assumption 1. $r_q, \dot{r}_q, \ddot{r}_q \in \mathcal{L}_\infty$.

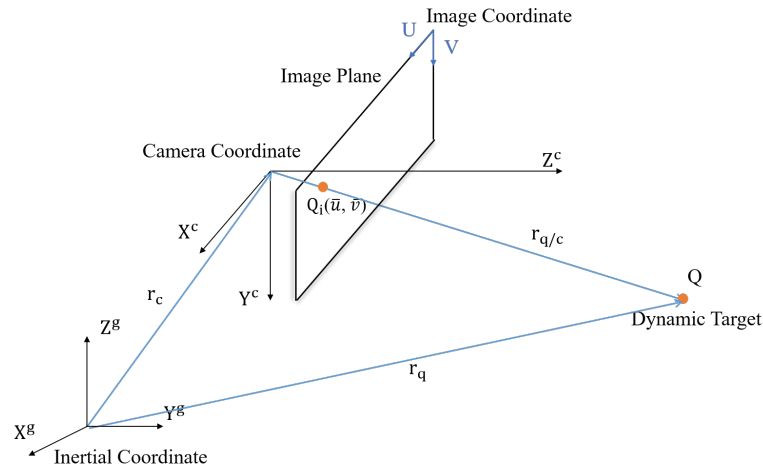


Figure 1. Kinematics model.

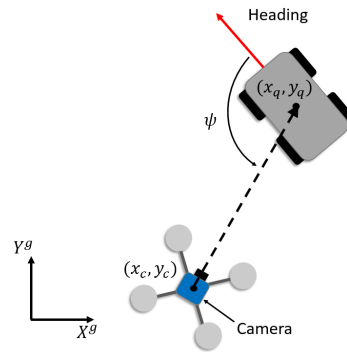


Figure 2. The definition of the orientation of target ψ observed by the UAV.

3. Estimating the Motion of the Moving Target

3.1. Kinematics Model

To model the tightly coupled motion between the camera and the target, the states of the system are defined as

$$\mathbf{x} = [x_1, x_2, x_3, \psi, r_q^T, V_q^T]^T, \quad (6)$$

where

$$[x_1, x_2, x_3] = \left[\frac{X}{Z}, \frac{Y}{Z}, \frac{1}{Z} \right] \quad (7)$$

denotes the state related to the image feature. By taking the time derivative of (6), the dynamics of the visual servoing system can be obtained as

$$\dot{\mathbf{x}} = \begin{bmatrix} v_{qx}x_3 - v_{qz}x_1x_3 + \zeta_1 + \eta_1 \\ v_{qy}x_3 - v_{qz}x_2x_3 + \zeta_2 + \eta_2 \\ -v_{qz}x_3^2 + v_{cx}x_3^2 - (\omega_{cy}x_1 - \omega_{cx}x_2)x_3 \\ \frac{(v_{cy}^s - v_{qy}^s)(x_c^s - x_q^s) - (v_{cx}^s - v_{qx}^s)(y_c^s - y_q^s)}{(x_c^s - x_q^s)^2 + (y_c^s - y_q^s)^2} - \frac{a_{qy}^s v_{qx}^s - a_{qx}^s v_{qy}^s}{(v_{qx}^s)^2 + (v_{qy}^s)^2} \\ V_q \\ 0_{3 \times 1} \end{bmatrix}, \quad (8)$$

where (2) is used and $\zeta_1, \zeta_2, \eta_1, \eta_2, x_{c/q}, y_{c/q} \in \mathbb{R}$ are defined as

$$\zeta_1 = \omega_{cz}x_2 - \omega_{cy} - \omega_{cy}x_1^2 + \omega_{cx}x_1x_2$$

$$\zeta_2 = -\omega_{cz}x_1 + \omega_{cx} + \omega_{cx}x_2^2 - \omega_{cy}x_1x_2$$

$$\eta_1 = (v_{cz}x_1 - v_{cx})x_3$$

$$\eta_2 = (v_{cz}x_2 - v_{cy})x_3$$

$$x_{c/q} = x_c - x_q$$

$$y_{c/q} = y_c - y_q.$$

3.2. Measurements: Bounding Box and Orientation

This section defines two measurements performed using a YOLO DNN as depicted in Figure 3. The bounding box and the orientation measurement provide information to correct the states in the UKF, and the detailed description of the DNN can be found in [41].

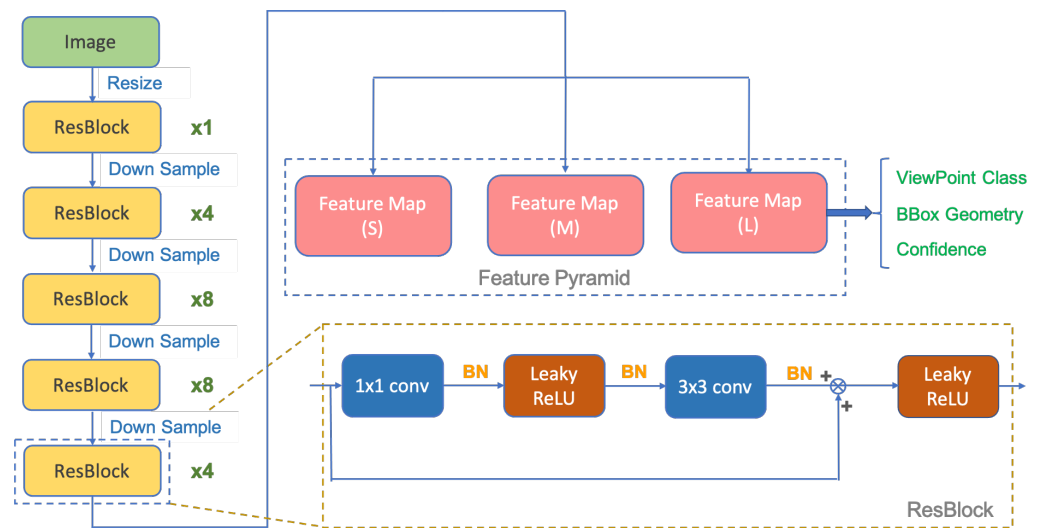


Figure 3. The network structure of a YOLO network, which take resized images as inputs and can generate three outputs, i.e., pose, bounding box, detection.

However, it is hard to defined the accuracy of the estimation since the output of the DNN network is classified into 24 classes. For example, as shown in the video (<https://www.youtube.com/watch?v=KMQD7KzsnPE&list=PLrxYXaxBXgRqAUZyX5TsTsPaBtfSYFpHL&index=15>) (accessed on 5 June 2021), the upper window demonstrates the estimation of the angle of the target, which is classified into 24 classes from 0 to 360 degree in order to reduce computational efforts. Therefore, when the camera moved from the left to the front, the angle only experienced 6 classes, and the accuracy calculated by comparing the differences between the 6 classes and the continuous variation of angles is not rigorously defined in the existing literature. Nevertheless, the estimation is affordable to be used for sensing the angle of the target in our application.

3.2.1. Bounding Box

Based on the pinhole model, states x_1 and x_2 can be obtained as

$$x_1 = \frac{\bar{u} - c_u}{f_x}$$

$$x_2 = \frac{\bar{v} - c_v}{f_y} \quad (9)$$

$$x_3 = \frac{1}{d'} \quad (10)$$

where f_x and f_y are the focal lengths of the camera, $[c_u, c_v]$ represents the center of the image frame, \bar{u} and \bar{v} represent the center of bounding box that encloses the detected moving target in the image frame as depicted in Figure 1, and d is the depth obtained from the pinhole model

$$d = \sqrt{\frac{A f_x f_y}{a}}, \quad (11)$$

where a is the area of the bounding box, and A is the ground truth side area of the target (i.e., typical size for a sedan is $4.6 \text{ m} \times 1.5 \text{ m}$).

Remark 1. The estimate of distance to the vehicle is obtained by (11) and is not directly generated from the DNN. That is, given the generated bounding box, its area A (i.e., unit in pixel) can be calculated and used to calculate the depth by (11). Therefore, as long as the generated bounding box can accurately enclose the target in the image and the focal lengths are accurately obtained by calibration, the estimated depth will be accurate.

Based on the camera measurement, pinhole model, orientation of the target, and the location of the UAV, the measurement model can be derived as

$$h(\mathbf{x}) = \begin{bmatrix} f_x x_1 + c_u \\ f_y x_2 + c_v \\ 1/x_3 \\ \psi \\ r_c \end{bmatrix}, \quad (12)$$

where (1), (7), and (9) are used, and ψ_m is ψ measured as described in Section 3.2.2, and d is the relative distance between the target and the UAV. Measurements d and ϕ are defined in detail in Section 3.2.

Remark 2. Changes in the target velocity can result in inaccurate state estimations, and therefore a UKF [42] is applied to address this issue. The model mismatch between the real and the model defined in (8) is considered as process noise, and the states can be estimated and updated to approach the ground truth in the UKF.

3.2.2. Orientation Measurement

As shown in Figure 2, ψ_m can be obtained from a YOLO network with the captured image from the onboard camera and is used to measure where the UAV is located observed by the camera, so that the controller developed in Section 4 can ensure the control objective $\psi \rightarrow \psi_{des}$ defined in (5) can be achieved.

3.3. Observability

Since the system dynamics defined in (8) is nonlinear, the approach to evaluate the observability of the system developed in (13) of [43] is applied to establish Theorem 1.

Theorem 1. The system defined in (8) is observable.

Proof. Given an observability matrix O defined as

$$O \triangleq \frac{\partial \phi}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathcal{L}_f^0(h(\mathbf{x}))}{\partial x_1} & \cdots & \frac{\partial \mathcal{L}_f^0(h(\mathbf{x}))}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{L}_f^{n-1}(h(\mathbf{x}))}{\partial x_1} & \cdots & \frac{\partial \mathcal{L}_f^{n-1}(h(\mathbf{x}))}{\partial x_n} \end{bmatrix}, \quad (13)$$

where x_1, \dots, x_n are the elements of state \mathbf{x} defined in (6), $h(\mathbf{x})$ is the measurement defined in (12), and $\mathcal{L}_f^i(h(\mathbf{x}))$ denoted the i th order Lie derivative of $h(\mathbf{x})$ is defined as

$$\begin{aligned}\mathcal{L}_f^0(h(\mathbf{x})) &\triangleq h(\mathbf{x}) \\ \mathcal{L}_f^1(h(\mathbf{x})) &\triangleq \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \cdot f \\ \mathcal{L}_f^2(h(\mathbf{x})) &\triangleq \frac{\partial}{\partial \mathbf{x}} [\mathcal{L}_f^1(h(\mathbf{x}))] \cdot f \\ &\vdots \\ \mathcal{L}_f^k(h(\mathbf{x})) &\triangleq \frac{\partial}{\partial \mathbf{x}} [\mathcal{L}_f^{k-1}(h(\mathbf{x}))] \cdot f,\end{aligned}$$

where $f = \dot{\mathbf{x}}$ is the vector field, and the submatrix of observability matrix O denoted as $O_s \in \mathbb{R}^{14 \times 10}$ is defined as

$$O_s \triangleq \begin{bmatrix} \frac{\partial \mathcal{L}_f^0(h(\mathbf{x}))}{\partial \mathbf{x}} \\ \frac{\partial \mathcal{L}_f^1(h(\mathbf{x}))}{\partial \mathbf{x}} \end{bmatrix},$$

and can be rewritten as

$$O_s = \begin{bmatrix} f_x & 0 & 0 & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 & 0 & 0 \\ 0 & 0 & -x_3^{-2} & 0 & 0 & 0 \\ & & & 1 & 0 & \\ 0 & 0 & 0 & 0 & I_3 & 0_{3 \times 3} \\ - & - & - & - & - & \\ & & \frac{\partial f_x \dot{x}_1}{\partial \mathbf{x}} & & & \\ & & \frac{\partial f_y \dot{x}_2}{\partial \mathbf{x}} & & & \\ & & \frac{\partial (-x_3^{-2} \dot{x}_3)}{\partial \mathbf{x}} & & & \\ & & \frac{\partial \psi}{\partial \mathbf{x}} & & & \\ 0 & 0 & 0 & 0 & 0 & I_3 \end{bmatrix}. \quad (14)$$

Based on (14), O_s is full-rank since $\text{rank}(O_s) = 10$ based on rows 1~7 and rows 12~14, which implies O is full-rank matrix and the system is observable. \square

4. Controller Architecture

Compared to the existing IBVS control methods [16], the controller developed in this work considers the constraints of visual features and control input in order to improve the tracking performance. The control input is calculated by minimizing the cost function designed in this section that considers not only the error of the feature vector but also the velocity of the moving target. Using the controller can lead to less-aggressive flying behavior and also decrease the energy consumption by choosing appropriate gain matrices in the cost function.

Controller Design

The dynamics model defined in (8) is derived for the process step in the UKF so that the target state can be estimated. The dynamics model of $s_m(t) = [x_1, x_2, x_3, \psi]^T \in \mathbb{R}^4$ used for prediction in the controller can be written as

$$\dot{s}_m = \begin{bmatrix} -x_3(v_{cx} - v_{qx}) + x_1x_3(v_{cz} - v_{qz}) - (1 + x_1^2)\omega_{cy} \\ -x_3(v_{cy} - v_{qy}) + x_2x_3(v_{cz} - v_{qz}) - \omega_{cy}x_1x_2 \\ x_3^2(v_{cz} - v_{qz}) - \omega_{cy}x_1x_3 \\ \frac{(v_{cy}^s - v_{qy}^s)(x_c^s - x_q^s) - (v_{cx}^s - v_{qx}^s)(y_c^s - y_q^s)}{(x_c^s - x_q^s)^2 + (y_c^s - y_q^s)^2} - \frac{a_{qy}^s v_{qx}^s - a_{qx}^s v_{qy}^s}{(v_{qx}^s)^2 + (v_{qy}^s)^2} \end{bmatrix}, \quad (15)$$

where the control input can be reduced to $u \triangleq [v_{qx}, v_{qy}, v_{qz}, \omega_{cy}]^T \in \mathbb{R}^4$. To deal with the modeling errors associated with the use of the dynamics model and the imperfect actuation of the UAV, an error signal ϵ_k is designed as:

$$\epsilon_k = s_k - s_{m,k} \quad (16)$$

where s_k is estimated using the UKF and $s_{m,k}$ is predicted using the model (15) at time k . The desired feature vector utilized in the controller can be defined as

$$s_{d,k} = s^* - \epsilon_k, \quad (17)$$

where $s^* \triangleq [x_1^*, x_2^*, x_3^*, \psi^*]^T \in \mathbb{R}^4$ is the reference feature vector prescribed by the user.

With (15) and (17), the optimal control problem (OCP) of the controller can be expressed as

$$\min_{s_m, u} \sum_{i=0}^{N_p-1} \left[\|s_{m,k+i/k} - s_{d,k+i/k}\|_{Q_s}^2 + \|u_{k+i/k} - u_{d,k+i/k}\|_{R_u}^2 \right] + \|s_{m,k+N_p/k} - s_{d,k+N_p/k}\|_{W_s}^2 \quad (18)$$

s.t.

$$\begin{aligned} s_{m,k+i+1/k} &= F_m(s_{m,k+i/k}, u_{k+i/k}), \quad i = 0, \dots, N_p - 1 \\ u_{d,k+i+1/k} &= u_{d,k+i/k} + \left[\ddot{\hat{r}}_i, \quad 0 \right]^T dt, \quad i = 0, \dots, N_p - 1 \\ s_{m,k+i/k} &\in \mathbb{S}, \quad i = 1, \dots, N_p \\ u_{k+i/k} &\in \mathbb{U}, \quad i = 0, \dots, N_p - 1 \\ s_{d,k+i/k} &= s^* - \epsilon_{k+i/k}, \quad i = 1, \dots, N_p \\ s_{m,k/k} &= s_k \\ s_{d,k/k} &= s_{d,k} \end{aligned} \quad (19)$$

where $u_d \in \mathbb{R}^4$ defined as the desired velocity that control input u needs to achieve and can be updated by $\ddot{\hat{r}}_i$ over the prediction horizon, where $\hat{r}_i(t) \in \mathbb{R}^3$ is an n th-order polynomial obtained by the QP and defined in the interval $[t_{i-1}, t_i]$ expressed as

$$\hat{r}_i(t) = \sum_{j=0}^n c_{ij}(t - t_{i-1})^j, \quad \text{for } t_{i-1} \leq t < t_i, \quad (20)$$

where $c_{ij} \in \mathbb{R}^3$ denotes the j th-order coefficient, $S_m \triangleq \{s_{m,k/k}, s_{m,k+1/k}, \dots, s_{m,k+N_p/k}\}$ denotes the predicted features at sampling time k based on the control sequence $U = \{u_{k/k}, u_{k+1/k}, \dots, u_{k+N_p-1/k}\}$ and the estimation s_k obtained using the UKF. Q_s , R_u , and W_s defined in (18) are positive weighting matrices, and $N_p \in \mathbb{N}$ is the prediction

horizon. Increasing N_p results in less-aggressive control inputs but also increases the computational effort. $F_m(\cdot)$ is the model defined in (15). Since only the linear velocity of the target can be estimated (i.e., \hat{r}_i), control input ω_{cy} only depends on the cost of feature errors $s_m - s_d$. \mathbb{S} and \mathbb{U} are the sets of constraints of the feature vector and the control input given by

$$\mathbb{S} \triangleq \left\{ s \in \mathbb{R}^4 \mid \begin{bmatrix} x_{1,min} \\ x_{2,min} \\ x_{3,min} \\ \psi_{min} \end{bmatrix} \leq s \leq \begin{bmatrix} x_{1,max} \\ x_{2,max} \\ x_{3,max} \\ \psi_{max} \end{bmatrix} \right\}$$

$$\mathbb{U} \triangleq \left\{ u \in \mathbb{R}^4 \mid \begin{bmatrix} v_{cx,min} \\ v_{cy,min} \\ v_{cz,min} \\ \omega_{cy,min} \end{bmatrix} \leq u \leq \begin{bmatrix} v_{cx,max} \\ v_{cy,max} \\ v_{cz,max} \\ \omega_{cy,max} \end{bmatrix} \right\}.$$

When solving the OCP at time k , ϵ_k would be calculated by (16) using the feature vector estimated using the UKF at time k and the feature vector predicted by the model (15) at time $k - 1$. This error signal is assumed to be constant over the prediction horizon; $s_{d,k+i/k}$, $i = 0, \dots, N_p$ would be constant over the prediction horizon.

The direct multiple shooting method is employed to solve the problem, since “lifting” the OCP to a higher dimension can usually speed up the rate of convergence. The OCP is repeated at every sampling time, with only the first vector $u_{k/k}$ in control set U being adopted in the system. Several software libraries can be used to solve this kind of nonlinear programming problem (NLP), and this study used CasADi for formulating the NLP and Ipopt for solving it.

Remark 3. The control architecture depicted in Figure 4 can be applied to general target tracking. That is, the feature vector and orientation generated from YOLO can be used to predict the motion of target v_q , which can facilitate tracking performance and, in turn, enhance the detection of bounding boxes.

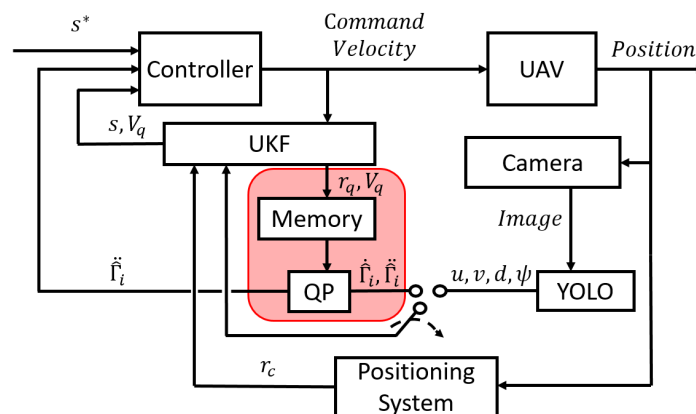


Figure 4. Block diagram of the controller, where u , v , d , and ψ are the measurements being updated using the UKF.

5. Simulations

5.1. Environment Setup

The Gazebo simulator under the ROS framework (version 18.04, Melodic) is used to conduct the simulations to verify the efficacy of the developed control algorithm, and the source code is posted on GitHub for public access (https://github.com/Networked-Control-Robotics-Lab/uav_nmpc_tracking_task.git) (accessed on 5 August 2021). A quadro-

tor implemented with the developed controller is deployed to track a moving vehicle. The initial locations of the target and the UAV are $[0, 0, 0]^T$ and $[2.18, 6.65, 1.75]^T$ and the initial orientations are $[0, 0, 90^\circ]^T$ and $[0, 0, -90^\circ]^T$, respectively, in the global frame as shown in Figure 5. The target moves on the XY plane following the non-holonomic dynamics, and its velocity is controlled by the user commands with a loop rate of 80 Hz. The resolution of the captured image is 640×480 , and the intrinsic parameters of the camera are

$$K = \begin{bmatrix} 381.36 & 0 & 320 \\ 0 & 381.36 & 240 \\ 0 & 0 & 1 \end{bmatrix}.$$

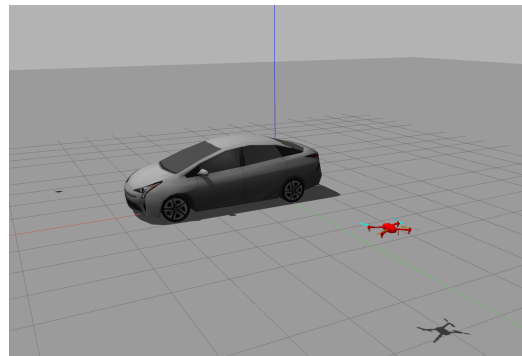


Figure 5. Setup of the car and UAV.

The control goal in the tracking task is to keep the moving target within the FOV of the camera at the desired depth and angle relative to the target. The reference feature vector was

$$s^* = \left[\frac{(320 - 320)}{381.36}, \frac{(312 - 240)}{381.36}, 1/7, 90^\circ \right]^T \in \mathbb{R}^4 \quad (21)$$

in (17) which defines the desired feature vector, as shown in Figure 6.

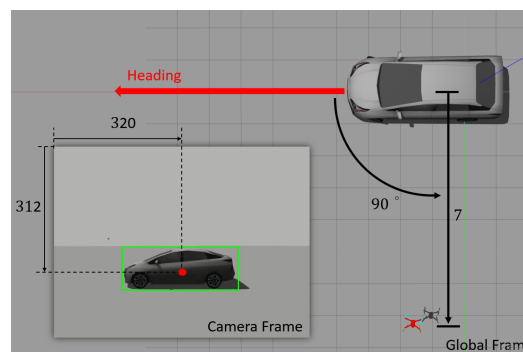


Figure 6. The red dot inside the bounding box represents the reference feature vector as defined in (21) and is used in the simulation as prescribed by the user.

In order to maintain the visibility and address the limitations of the actuator while tracking the moving target, the constraints of the states and constraints of the control inputs were considered in the numerical simulations, as listed in Tables 1 and 2. The parameters for tuning the sampling time, predicted horizons, and the weighting matrices in the cost function are presented in Table 3.

Table 1. State Constraints.

States	Min.	Max.
x_1	−0.84	0.84
x_2	−0.63	0.63
x_3	0.07	1
ψ	0°	180°

Table 2. Control Constraints.

Inputs	Min.	Max.
v_{cx}, v_{cy}, v_{cz} (m/s)	−10	10
w_{cy} (rad/s)	−0.59	0.59

Table 3. Control Parameter Settings.

Parameter	Value
Sampling Time (Δk)	0.025 s
Horizon Length (N_p)	50
Q_s	$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 100 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$
R_u	$\begin{bmatrix} 0.02 & 1 & 1 & 1 \\ 1 & 0.03 & 1 & 1 \\ 1 & 1 & 0.01 & 1 \\ 1 & 1 & 1 & 0.3 \end{bmatrix}$

5.2. Simulation Results

To demonstrate the practicability and efficiency of the developed controller, three simulations are described that compared the controller and the traditional IBVS controller. In the first simulation, the UAV tracked a target that was at a relative angle that changed over time in order to show the benefit of including the angle dynamics in the dynamics model. In the second simulation, the UAV tracked a target moving over a z-shaped path in order to compare the tracking performance of different designs of controllers in an aggressive-motion condition. In the third simulation, the IBVS controller and the controller were utilized to demonstrate the tracking efficiency in an aggressive-motion condition.

5.2.1. Simulation 1: Controller with Relative Rotational Dynamics

The scenario of Simulation 1 is depicted in Figure 7. The controller designed in Section 4 was implemented to track a target at a time-varying angle. In Case 1, the rotational dynamics was not considered in the dynamics model, and the relative orientation was measured directly using a YOLO network. In Case 2, the angle dynamics was taken into consideration in the UKF, which made the relative orientation more robust to noisy or intermittent measurements from the YOLO network.

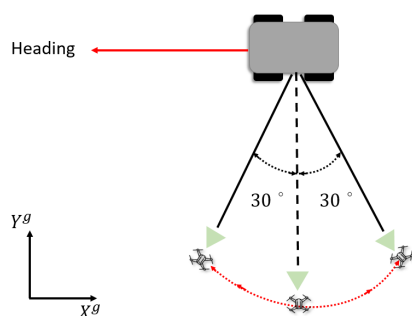


Figure 7. A UAV tracks a target at a desired relative angle that changes over time. The desired relative angle was $\psi^* = 30 \sin(t) + 90$.

Comparison of State Feature Vectors

Figures 8 and 9 show the trajectories of the features (i.e., the center of the bounding box) in the image frame over time.

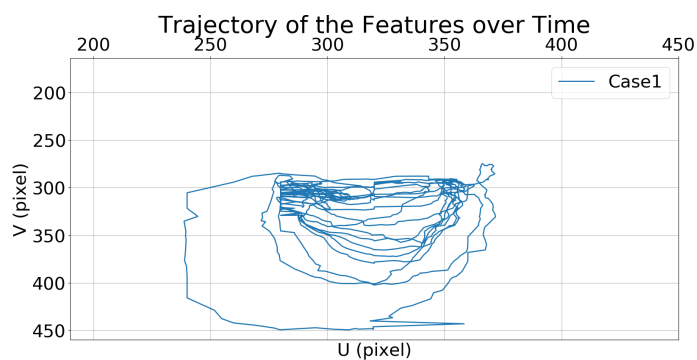


Figure 8. Simulation 1: trajectory of the moving target in the image frame for Case 1.

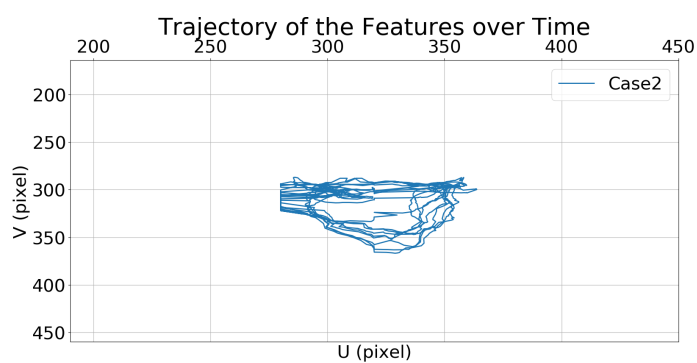


Figure 9. Simulation 1: trajectory of the moving target in the image frame for Case 2.

The performances are quantified as the root mean square (RMS) tracking errors defined in Table 4.

Table 4. Simulation 1: RMS errors.

Case	U-Axis	V-Axis
1	29.75	28.93
2	25.00	18.18

Figure 10 compares the state vector tracking errors in Cases 1 and 2. The trends in the state vectors are consistent in the two cases.



Figure 10. Simulation 1: Comparisons of the state vector tracking errors.

Comparison of Control Inputs

Figures 11 and 12 show the velocity control inputs for Case 1 and Case 2 in Simulation 1, respectively, and the upper and lower bounds on the control inputs defined in Table 2 are highlighted.

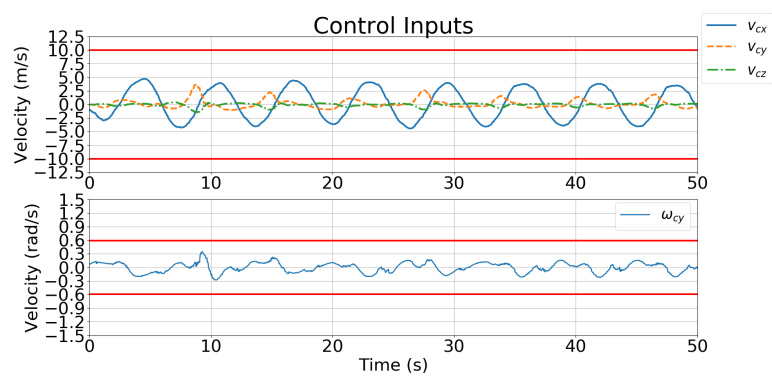


Figure 11. Simulation 1: velocity control input for Case 1.

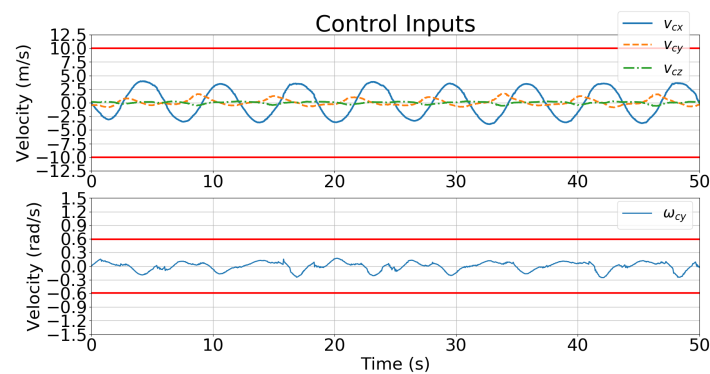


Figure 12. Simulation 1: velocity control input for Case 2.

Figure 13 compares the total control inputs, which shows that they were lower in Case 2 than in Case 1. “The total control inputs are defined as the summation of $u_{k+i/k}$ solved from the cost problem defined in (18) over the simulation time period (i.e., 50 s). The higher the value, the higher the energy consumption since the energy consumption of the UAV is proportional to the cube of its speed $u_{k+i/k}$ defined below (15) based on [44]”.

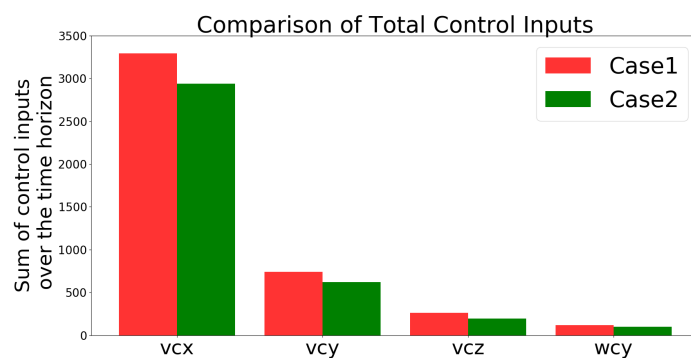


Figure 13. Simulation 1: comparison of the total amount of control inputs.

5.2.2. Simulation 2: Controller with Target Motion Pattern

The scenario of Simulation 2 is depicted in Figure 14, where the UAV is controlled to track a target moving over a z-shaped path. The controllers for Cases 1 and 2 are compared. In Case 1, the difference in the contiguous control inputs to the UAV (i.e., differential control cost) is considered in the controller. In Case 2, by leveraging the motion pattern estimated using the UKF, the difference between the velocities of the UAV and the target can be considered in the controller designed in Section 4.

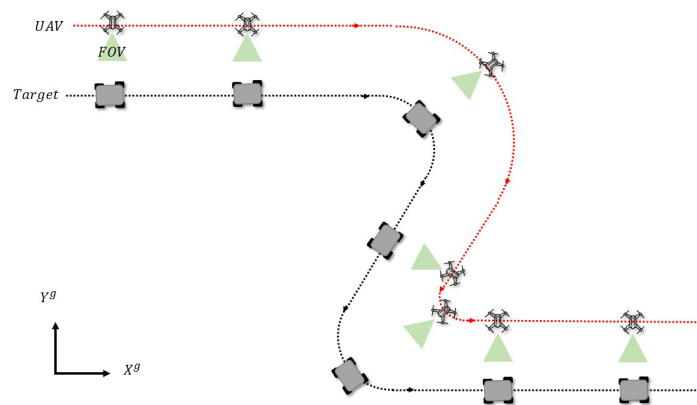


Figure 14. A UAV tracks a moving target moving over a z-shaped path. The trajectory of the target is unknown to the UAV. The target moves at 1 m/s in the heading direction from 3 s to 13 s, and first turns at -0.2 m/s^2 from 13 s to 23 s. The target keeps moving at 1 m/s from 23 s to 33 s, and performed a second turn at 0.2 m/s^2 from 33 s to 43 s, and then keeps moving at 1 m/s.

Comparison of State Feature Vectors

Figures 15 and 16 show the trajectories of the target features in the image frame over time, which indicate that the feature trajectory in Case 2 moves over a smaller range compared to that in Case 1.

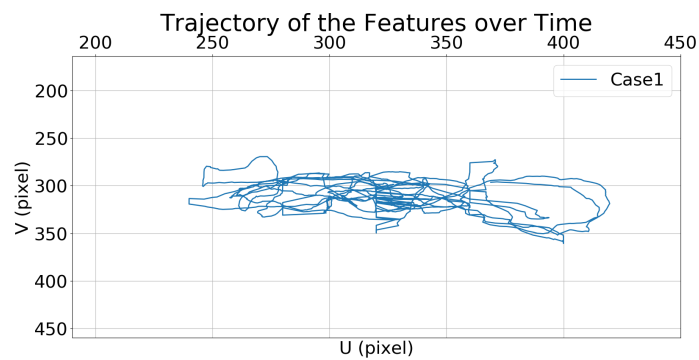


Figure 15. Simulation 2: trajectory of the moving target in the image frame for Case 1.

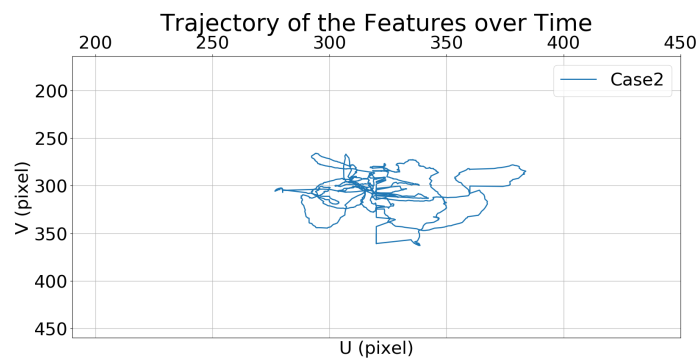


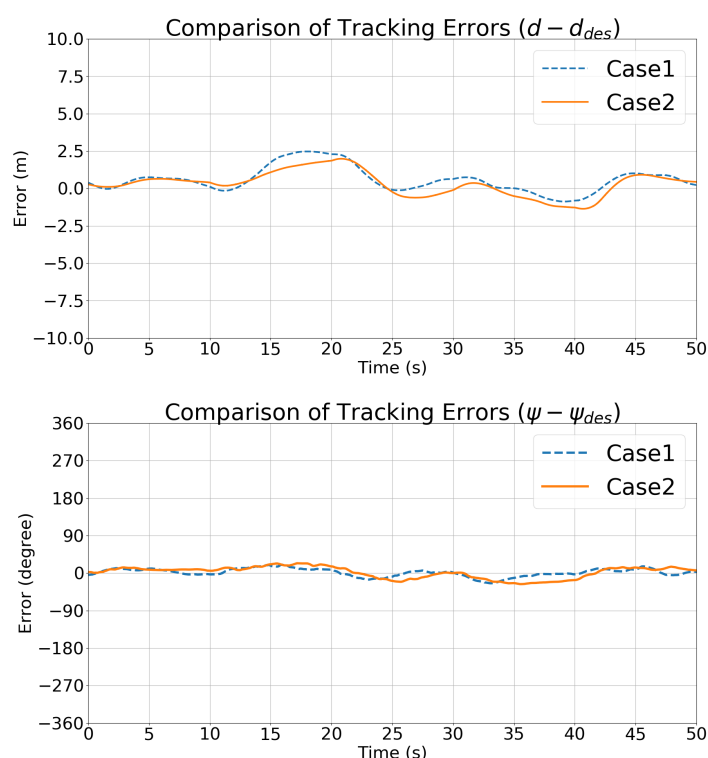
Figure 16. Simulation 2: trajectory of the moving target in the image frame for Case 2.

The performances are quantified based on the RMS tracking errors defined in Table 5.

Table 5. Simulation 2: RMS errors.

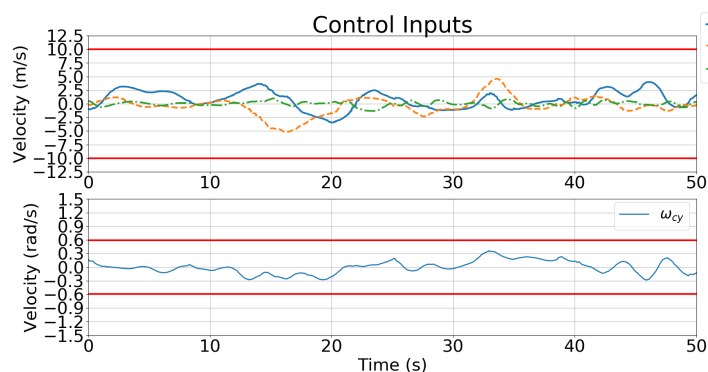
Case	U-Axis	V-Axis
1	40.92	16.60
2	18.80	20.42

Figure 17 compares the depth and relative angle tracking errors in Case 1 and Case 2, which also shows that Case 2 performed better than Case 1. Thanks to the motion pattern estimated using the UKF, the controller developed in Section 4 performed better than the traditional optimal controller with differential control cost.

**Figure 17.** Simulation 2: comparisons of the state vector tracking errors.

Comparison of Control Inputs

Figures 18 and 19 show the velocity control inputs for Cases 1 and 2, respectively, and the upper and lower bounds on the control inputs defined in Table 2 are highlighted. The control inputs were much more stable and smoother for Case 2 than for Case 1.

**Figure 18.** Simulation 2: velocity control input for Case 1.

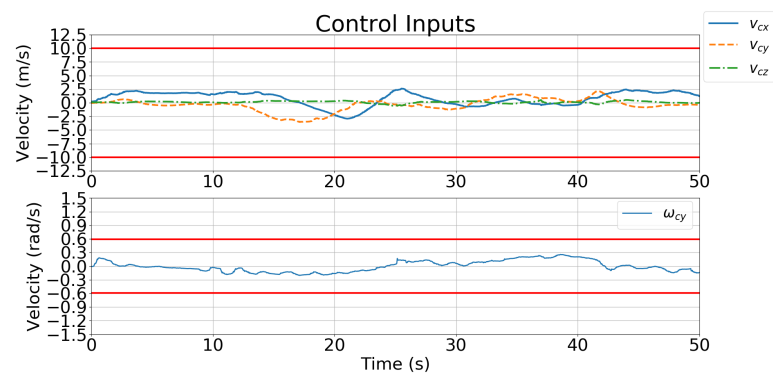


Figure 19. Simulation 2: velocity control input for Case 2.

Figure 20 compares the total control inputs in Cases 1 and 2, and shows that these inputs were lower for Case 2 than for Case 1.

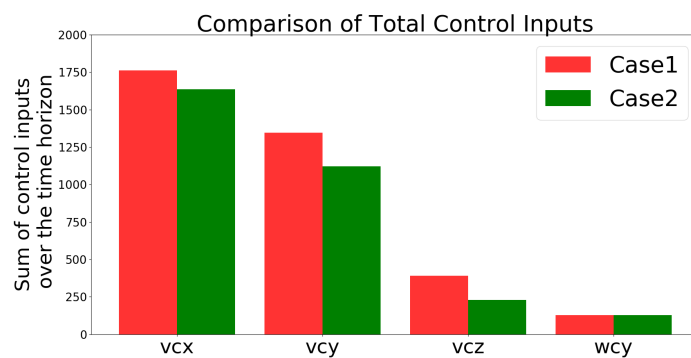


Figure 20. Simulation 2: comparison of the total amount of the control inputs.

5.2.3. Simulation 3: IBVS vs. the Developed Controller

The scenario in this simulation is the same as that depicted in Figure 14, but the performances of the IBVS controller and the controller designed in Section 4 are compared to track a target moving over a z-shaped path in Cases 1 and 2.

Comparison of State Feature Vectors

Figures 16 and 21 show the trajectories of the features in the image frame over time. The stability of the IBVS controller and the controller is also evident from Table 6, which lists the RMS tracking errors. Figure 22 compares the state vector tracking errors in Cases 1 and 2.

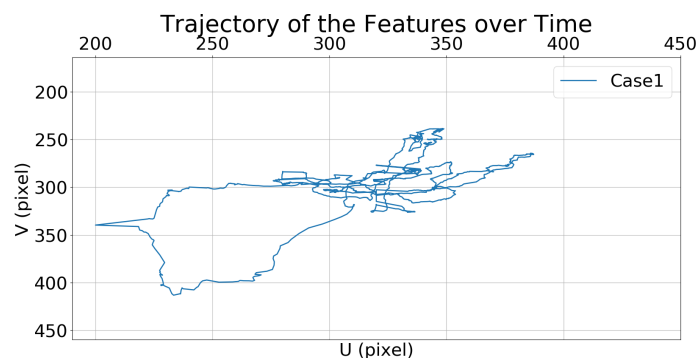
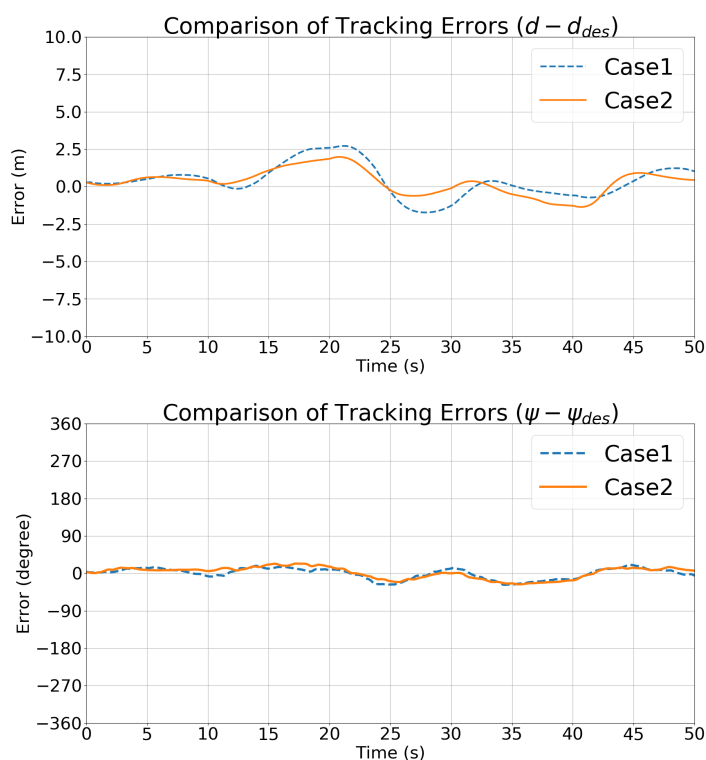


Figure 21. Case 1: trajectory of the moving target in the image frame using the IBVS controller.

The performances quantified based on the RMS tracking errors are presented in Table 6.

Table 6. Simulation 3: RMS errors.

Case	U-Axis	V-Axis
1	34.92	35.23
2	18.80	20.42

**Figure 22.** Simulation 3: comparisons of the state vector tracking errors.

Comparison of Control Inputs

Figures 19 and 23 show the velocity control inputs to the IBVS controller and the developed controller, respectively, and the upper and lower bounds on the control inputs defined in Table 2 are highlighted. Figure 23 shows that the linear and angular velocity control input generated for the IBVS controller keeps fluctuating while tracking. In contrast, Figure 19 shows that the linear and angular velocity control inputs of the controller are stable and smooth.

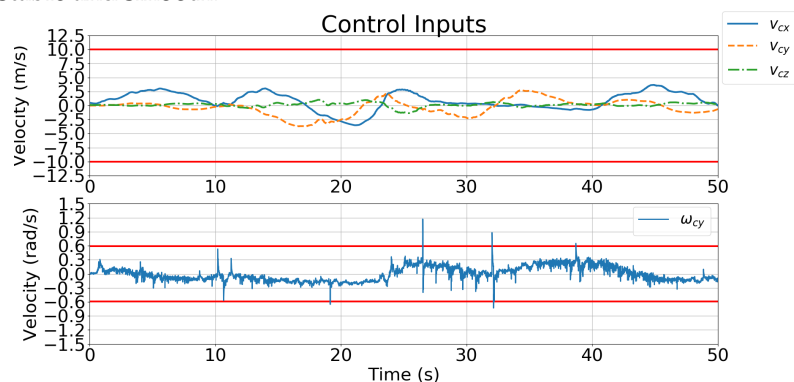
**Figure 23.** Case 1: velocity control input of the IBVS controller.

Figure 24 shows that compared with the IBVS controller, the controller needs much less energy to track the moving target. In other words, the controller is much more efficient than the IBVS controller since it saves energy by not generating unnecessary motion.

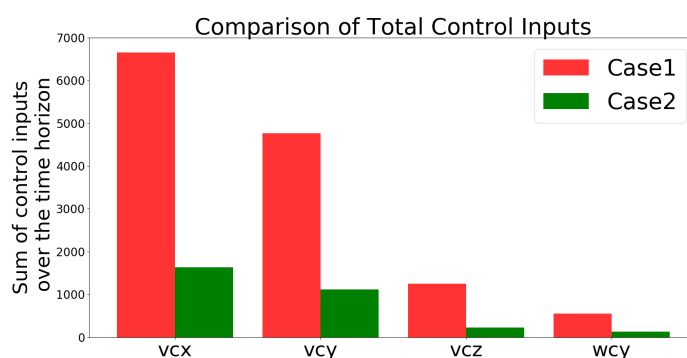


Figure 24. Comparison of the total amounts of the control inputs.

IBVS Controller vs. The Developed Controller

The IBVS controller relies on the image-feature error of the target for feedback, which results in the motion of the UAV becoming unstable and oscillating with the inclusion of a large error (i.e., large displacement and rotation) as shown in Figure 24. In contrast, the developed controller is designed to minimize the cost function that considers not only the current feature error but also predicts the future states of the image features in the predicted horizons according to the dynamics model. Additionally, the constraints of the control inputs are taken into consideration in order to avoid excessive and unreasonable motions. Figure 23 shows that the angular velocity exceeds the limitation of the control inputs defined in Table 2; however, the angular velocity generated by the controller in Figure 19 prevents unreasonable motion from occurring.

6. Conclusions

A tracking controller has been developed to track a moving target, and it relies on bounding box features generated by a YOLO network and a target motion pattern obtained by QP, in which the target states are estimated using a UKF. The main features of the developed controller ensure the following characteristics:

1. The control effort is less than for traditional controllers, while the tracking error is quantified by the RMS error and is less than that for a traditional controller.
2. The value of the control input to the controller always remains within the UAV motion capabilities.
3. The target can be tracked, and its image features always remain within the image, which is not guaranteed for a traditional IBVS controller.

All of the above results were verified in three simulations. The first simulation illustrated that the controller, which considers the relative rotational dynamics, can perform better in tracking a target at a specified angle. The second simulation indicated that using the predicted motion pattern of a target in the controller can improve the tracking performance compared to the traditional optimal controller, and the controller also requires less control effort for tracking. The third simulation compared the controller with a traditional IBVS controller, and showed that the controller requires much less control effort and achieves a smaller tracking error. This work has demonstrated both the control efficacy and high performance of the developed controller.

Author Contributions: Conceptualization, T.-H.C.; methodology, C.-W.C.; software, C.-W.C. and H.-A.H.; validation, C.-W.C. and P.-H.Y.; formal analysis, C.-W.C. and H.-A.H.; investigation, C.-W.C. and H.-A.H.; resources, T.-H.C.; data curation, C.-W.C. and H.-A.H.; writing—original draft preparation, C.-W.C.; writing—review and editing, T.-H.C.; visualization, C.-W.C. and H.-A.H.; supervision, T.-H.C.; project administration, T.-H.C.; funding acquisition, T.-H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Ministry of Science and Technology, Taiwan (Grant Number MOST 110-2222-E-A49-005-) and by Pervasive Artificial Intelligence Research (PAIR) Labs, Taiwan (Grant Number MOST 110-2634-F-009-018-).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Minaeian, S.; Liu, J.; Son, Y. Vision-based target detection and localization via a team of cooperative uav and ugvs. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 1005–1016. [\[CrossRef\]](#)
2. Cavaliere, D.; Loia, V.; Saggese, A.; Senatore, S.; Vento, M. Semantically enhanced uavs to increase the aerial scene understanding. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 555–567. [\[CrossRef\]](#)
3. Chitrakaran, V.; Dawson, D.M.; Dixon, W.E.; Chen, J. Identification of a moving object's velocity with a fixed camera. *Automatica* **2005**, *41*, 553–562. [\[CrossRef\]](#)
4. Schlichenmaier, J.; Selvaraj, N.; Stolz, M.; Waldschmidt, C. Template matching for radar-based orientation and position estimation in automotive scenarios. In Proceedings of the IEEE MTT-S International Conference Microwaves Intelligent Mobility (ICMIM), Nagoya, Japan, 19–21 March 2017; pp. 95–98.
5. Stroupe, A.; Martin, M.; Balch, T. Distributed sensor fusion for object position estimation by multi-robot systems. In Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea, 21–26 May 2001; pp. 1092–1098.
6. Kamthe, A.; Jiang, L.; Dudys, M.; Cerpa, A. Scopes: Smart cameras object position estimation system. In *EWSN '09*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 279–295.
7. Ahn, H.S.; Ko, K.H. Simple pedestrian localization algorithms based on distributed wireless sensor networks. *IEEE Trans. Ind. Electron.* **2009**, *56*, 4296–4302.
8. Chitrakaran, V.K.; Dawson, D.M. A lyapunov-based method for estimation of euclidean position of static features using a single camera. In Proceedings of the American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 1988–1993.
9. Braganza, D.; Dawson, D.M.; Hughes, T. Euclidean position estimation of static features using a moving camera with known velocities. In Proceedings of the IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 2695–2700.
10. Chhatkuli, A.; Pizarro, D.; Collins, T.; Bartoli, A. Inextensible non-rigid structure-from-motion by second-order cone programming. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2428–2441. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Agudo, A.; Moreno-Noguer, F. Simultaneous pose and non-rigid shape with particle dynamics. In Proceedings of the IEEE Conference Computer Vision Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
12. Mateus, A.; Tahri, O.; Miraldo, P. Active Structure-from-Motion for 3D Straight Lines. *arXiv* **2018**, arXiv:1807.00753.
13. Spica, R.; Giordano, P.R.; Chaumette, F. Active structure from motion: Application to point, sphere, and cylinder. *IEEE Trans. Robot.* **2014**, *30*, 1499–1513. [\[CrossRef\]](#)
14. Spica, R.; Giordano, P.R.; Chaumette, F. Plane estimation by active vision from point features and image moments. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 6003–6010. [\[CrossRef\]](#)
15. Giordano, P.R.; Spica, R.; Chaumette, F. Learning the shape of image moments for optimal 3d structure estimation. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 5990–5996. [\[CrossRef\]](#)
16. Chaumette, F.; Hutchinson, S. Visual servo control part I: Basic approaches. *IEEE Rob. Autom. Mag.* **2006**, *13*, 82–90. [\[CrossRef\]](#)
17. Keshmiri, M.; Xie, W. Image-based visual servoing using an optimized trajectory planning technique. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 359–370. [\[CrossRef\]](#)
18. Allibert, G.; Courtial, E.; Chaumette, F. Predictive control for constrained image-based visual servoing. *IEEE Trans. Robot.* **2010**, *26*, 933–939. [\[CrossRef\]](#)
19. Qiu, Z.; Hu, S.; Liang, X. Model predictive control for uncalibrated and constrained image-based visual servoing without joint velocity measurements. *IEEE Access* **2019**, *7*, 73540–73554. [\[CrossRef\]](#)
20. 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. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Ke, F.; Li, Z.; Xiao, H.; Zhang, X. Visual servoing of constrained mobile robots based on model predictive control. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1428–1438. [\[CrossRef\]](#)
22. Falanga, D.; Foehn, P.; Lu, P.; Scaramuzza, D. PAMPC: Perception-aware model predictive control for quadrotors. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018; pp. 1–8.

23. Razzanelli, M.; Innocenti, M.; Pannocchia, G.; Pollini, L. Vision-based Model Predictive Control for Unmanned Aerial Vehicles Automatic Trajectory Generation and Tracking. Available online: <https://arc.aiaa.org/doi/abs/10.2514/6.2019-1409> (accessed on 5 June 2021).
24. Shi, H.; Sun, G.; Wang, Y.; Hwang, K. Adaptive image-based visual servoing with temporary loss of the visual signal. *IEEE Trans. Ind. Inf.* **2019**, *15*, 1956–1965. [\[CrossRef\]](#)
25. Mondragon, I.; Campoy, P.; Olivares-Mendez, M.; Martinez, C. 3D object following based on visual information for unmanned aerial vehicles. In Proceedings of the IEEE Colombian Conference on Automatic Control, Bogota, Colombia, 1–4 October 2011; pp. 1–7.
26. Wang, H.; Yang, B.; Wang, J.; Liang, X.; Chen, W.; Liu, Y. Adaptive visual servoing of contour features. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 811–822. [\[CrossRef\]](#)
27. Siradjuddin, I.; Tundung, S.P.; Indah, A.S.; Adhisuwignjo, S. A real-time model based visual servoing application for a differential drive mobile robot using beaglebone black embedded system. In Proceedings of the 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), Langkawi, Malaysia, 18–20 October 2015; pp. 186–192.
28. Pan, W.; Lyu, M.; Hwang, K.-S.; Ju, M.-Y.; Shi, H. A neuro-fuzzy visual servoing controller for an articulated manipulator. *IEEE Access* **2018**, *6*, 3346–3357. [\[CrossRef\]](#)
29. Lang, H.; Wang, Y.; de Silva, C.W. Vision based object identification and tracking for mobile robot visual servo control. In Proceedings of the IEEE ICCA 2010, Xiamen, China, 9–11 June 2010; pp. 92–96.
30. Pence, W.G.; Farello, F.; Alqasemi, R.; Sun, Y.; Dubey, R. Visual servoing control of a 9-dof wmra to perform adl tasks. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 916–922.
31. Djelal, N.; Saadia, N.; Ramdane-Cherif, A. Target tracking based on surf and image based visual servoing. In Proceedings of the CCCA12, Marseilles, France, 6–8 December 2012; pp. 1–5.
32. Anh, T.L.; Song, J. Robotic grasping based on efficient tracking and visual servoing using local feature descriptors. *Int. J. Precis. Eng. Manuf.* **2012**, *13*, 387–393. [\[CrossRef\]](#)
33. Penin, B.; Giordano, P.R.; Chaumette, F. Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3725–3732. [\[CrossRef\]](#)
34. Hulens, D.; Goedemé, T. Autonomous flying cameraman with embedded person detection and tracking while applying cinematographic rules. In Proceedings of the Conference Comput. Robot Vision, Edmonton, AB, Canada, 17–19 May 2017; pp. 56–63.
35. Gupta, M.; Kumar, S.; Behera, L.; Subramanian, V.K. A novel vision-based tracking algorithm for a human-following mobile robot. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1415–1427. [\[CrossRef\]](#)
36. Liu, Y.; Wang, Q.; Hu, H.; He, Y. A novel real-time moving target tracking and path planning system for a quadrotor uav in unknown unstructured outdoor scenes. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2362–2372. [\[CrossRef\]](#)
37. Lee, J.H.; Millard, J.D.; Lusk, P.C.; Beard, R.W. Autonomous target following with monocular camera on uas using recursive-ransac tracker. In Proceedings of the International Conference on Unmanned Aircraft Systems, Dallas, TX, USA, 12–15 June 2018; pp. 1070–1074.
38. Cao, Z.; Chen, X.; Yu, Y.; Yu, J.; Liu, X.; 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. [\[CrossRef\]](#)
39. Limon, D.; Pereira, M.; Peña, D.; Alamo, T.; Jones, C.N.; Zeilinger, M.N. MPC for tracking periodic references. *IEEE Trans. Autom. Control* **2016**, *61*, 1123–1128. [\[CrossRef\]](#)
40. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
41. Li, J.-M.; Chen, C.-W.; Cheng, T.-H. Motion prediction and robust tracking of a dynamic and temporarily-occluded target by an unmanned aerial vehicle. *IEEE Trans. Control Syst. Technol.* **2020**, *29*, 1623–1635. [\[CrossRef\]](#)
42. Wan, E.A.; Menve, R.V. The unscented kalman filter for nonlinear estimation. In Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium, Lake Louise, AB, Canada, 1–4 October 2000; pp. 153–158.
43. Whalen, A.J.; Brennan, S.N.; Sauer, T.D.; Schiff, S.J. Observability and controllability of nonlinear networks: The role of symmetry. *Phys. Rev. X* **2015**, *5*, 011005. Available online: <https://link.aps.org/doi/10.1103/PhysRevX.5.011005> (accessed on 5 June 2021). [\[CrossRef\]](#)
44. Sun, Y.; Xu, D.; Ng, D.W.K.; Dai, L.; Schober, R. Optimal 3d-trajectory design and resource allocation for solar-powered UAV communication systems. *IEEE Trans. Commun.* **2019**, *67*, 4281–4298. [\[CrossRef\]](#)