drones

MDPI

Article

Approximate Optimal Curve Path Tracking Control for Nonlinear Systems with Asymmetric Input Constraints

Yajing Wang, **Xiangke Wang** * and **Lincheng Shen**

College of Intelligence Science and Technology, National University of Defense Technology,
Changsha 410073, China
* Correspondence: xkwang@nudt.edu.cn

Abstract: This paper proposes an approximate optimal curve-path-tracking control algorithm for partially unknown nonlinear systems subject to asymmetric control input constraints. Firstly, the problem is simplified by introducing a feedforward control law, and a dedicated design for optimal control with asymmetric input constraints is provided by redesigning the control cost function in a non-quadratic form. Then, the optimality and stability of the derived optimal control policy is demonstrated. To solve the underlying tracking Hamilton–Jacobi–Bellman (HJB) equation in consideration of partially unknown systems, an integral reinforcement learning (IRL) algorithm is utilized using the neural network (NN)-based value function approximation. Finally, the effectiveness and generalization of the proposed method is verified by experiments carried out on a high-fidelity hardware-in-the-loop (HIL) simulation system for fixed-wing unmanned aerial vehicles (UAVs) in comparison with three other typical path-tracking control algorithms.

Keywords: optimal tracking control; asymmetric input constraints; integral reinforcement learning; fixed-wing UAVs

check for updates

Citation: Wang, Y.; Wang, X.; Shen, L. Approximate Optimal Curve Path Tracking Control for Nonlinear Systems with Asymmetric Input Constraints. *Drones* **2022**, *6*, 319. https://doi.org/10.3390/drones6110319

Academic Editor: Francesco Nex

Received: 5 September 2022
Accepted: 21 October 2022
Published: 26 October 2022

1. Introduction

The optimal tracking control problem (OTCP) is of major importance in a variety of applications for robotic systems such as wheeled vehicles, unmanned ground vehicles (UGVs), unmanned aerial vehicles (UAVs), etc. The aim is to find a control policy to drive the specified system, given a particular reference path to follow in an optimal manner [1–6]. The reference paths are generally generated by a separate mission planner according to specific tasks, and optimization is usually achieved by minimizing an objective function regarding energy cost, tracking error cost, and/or the traveling time cost.

With the rapid development of unmanned systems, algorithms to solve OTCPs have been widely studied in the literature. Addressing the OTCPs involves solving the underlying Hamilton–Jacobi–Bellman (HJB) equation. For linear systems, the HJB equation is replaced by the Riccati equation, and the numerical solution is generally available. However, for nonlinear robotic systems subject to asymmetric input constraints, such as fixed-wing UAVs and autonomous underwater vehicles (AUVs) [7–9], it is still a challenging issue. To deal with this difficulty while guaranteeing tracking performance for nonlinear systems, various methods have been developed to find approximate optimal control efficacy. One idea is to simplify or transform the objective function to be optimized to obtain a solution to an approximate or equivalent optimal control problem. For instance, nonlinear model predictive control (MPC) is used to obtain a near optimal path-following control law for UAVs by truncating the time horizon and minimizing a finite-horizon tracking objective function in [7,8]. Another idea aims to solve the approximate solution directly. An offline policy iteration (PI) strategy is utilized to obtain the near optimal solution by solving a sequence of Bellman equation iteratively [10]. However, in the abovmentioned methods, the complete dynamics of the system are generally required and the curse of dimensionality might occur. To deal with this issue, an approximate dynamic programming (ADP) scheme was developed and has received increasing interest in the optimal control area [11–13].

ADP, which combines the concept of reinforcement learning (RL) and Bellman's principle of optimality, was first introduced in [11] to handle the curse of dimensionality that might occur in the classical dynamic programming (DP) scheme for solving optimal control problems. The main idea is to approximate the solution to the HJB equation using some parametric function approximation techniques, for which aneural network (NN) is the most commonly used scheme, such as a single-NN based value function approximation and the actor–critic dual-NN structure [14]. For continuous-time nonlinear systems, Ref. [15] proposed a data-based ADP algorithm to relax the dependence on the internal dynamics of the control system, which is also called integral reinforcement learning (IRL), to learn the solution to the HJB equation using only partial knowledge about the system dynamics. After that, the IRL scheme became widely used in various nonlinear optimization control problems, including optimal tracking control, control with input constraints, control with unknown or partially unknown systems, etc. [7,14–16].

The IRL-based methods are powerful tools used to solve nonlinear optimal control problems. However, the OTCP for nonlinear systems with partially unknown dynamics and asymmetric input constraints, especially for curve path tracking is still open to study. Firstly, the stability of the IRL-based methods for nonlinear constrained systems are generally hard to prove. Moreover, the changing curvature in the curve-path-tracking control problem makes it more difficult to stabilize the tracking error compared to the widely studied regulation control or circular path-tracking control problems. Moreover, the asymmetric input constraints are more difficult to deal with than commonly discussed symmetric constraints.

Motivated by the desire to solve the OTCP with the curve path for partially unknown nonlinear systems with asymmetric input constraints, this paper introduces a feedforward control law to simplify the problem and redesigns the non-quadratic form control input cost function and utilizes an NN-based IRL scheme to solve an approximate optimal control policy. The three main contributions are:

1. An approximate optimal curve-path-tracking control policy is developed for nonlinear systems with a feedforward control law, which handles the time-varying dynamics of the reference states caused by the curvature variation, and a data-driven IRL algorithm is developed to solve the approximate optimal control policy, in which a single-NN structure for value function approximation is utilized, reducing the computation burden and simplifying the algorithm structure.
2. The non-quadratic control cost function is redesigned via a constraint transformation with the introduced feedforward control law, which solves the challenge of asymmetric control input constraints that traditional methods cannot handle directly, and satisfactory input constraints are guaranteed with proof.
3. The proposed approximate optimal path-tracking control algorithm is validated via hardware-in-the-loop (HIL) simulations for fixed-wing UAVs in comparison with three other typical path-tracking algorithms. The result shows that the proposed algorithm not only has much less fluctuation and smaller root mean squared error (RMSE) of the tracking error but also naturally meets the control input constraints.

2. Problem Formulation

This section briefly formulates the OTCP of nonlinear systems subject to asymmetric control input constraints.

Consider the following affine nonlinear kinematic systems:

$$\dot{\mathbf{x}}_k(t) = f_k(\mathbf{x}_k(t)) + g_k(\mathbf{x}_k(t))\mathbf{u}(t), \tag{1}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_1}$ is the vector of system motion states that we focus on, $f_k(\cdot) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_1}$ is the internal kinematic dynamics, $g_k(\cdot) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_1 \times m}$ is the control input dynamics of system, and $\mathbf{u} \in \mathbb{R}^m$ is the control input, which is constrained by

$$\lambda_j^{min} \leq \mathbf{u}^j \leq \lambda_j^{max}, \ \ j = 1, \dots, m \tag{2}$$

where λ_j^{min} and λ_j^{max} are the minimum and the maximum thresholds of control input $\mathbf{u}^j$, which are decided by characteristics of the actuator, and not always satisfying $\lambda_j^{min} = -\lambda_j^{max}$.

Remark 1. *The asymmetric control input constraint (2) is widespread in practical systems, such as fixed-wing UAVs and autonomous underwater vehicles (AUVs) [1,7–9,17]. For these systems, existing control algorithms that consider only symmetric input constraints cannot be utilized directly.*

This paper studies the OTCP with curve paths for system (1) with input constraint (2). Thus, we focus on the tracking performance of the above motion states $\mathbf{x}_k$ with reference to the reference motion states $\mathbf{x}_{k_d}$ specified by the corresponding virtual target point (VTP) $\mathbf{p}_d$ on the reference path. Then, the considered tracking control system is described as

$$\begin{cases} \dot{\mathbf{x}}_e = f_e(\mathbf{x}_e, \mathbf{x}_d) + g_e(\mathbf{x}_e, \mathbf{x}_d)\mathbf{u} \\ \dot{\mathbf{x}}_d = f_d(\mathbf{x}_d) \end{cases}, \tag{3}$$

where $\mathbf{x}_e = \mathbf{x}_k - \mathbf{x}_{k_d}$ describes the tracking error state, $\mathbf{x}_d = [\mathbf{x}_{k_d}^\top, \mathbf{x}_{c_d}^\top]^\top \in \mathbb{R}^{n_2}$ represents the bounded state vector related to the reference motion states, not subject to human control, $\mathbf{x}_{k_d} \in \mathbb{R}^{n_1}$ is the reference motion states, and $\mathbf{x}_{c_d} \in \mathbb{R}^{n_2 - n_1}$ describes some other related system variables, $n_2 - n_1 \geq 0$. The continuous-time functions, $f_e(\cdot)$ and $g_e(\cdot)$, are internal dynamics and control input dynamics of the tracking error system, $f_d(\mathbf{x}_d)$ is the dynamics of the reference states and is decided by the task setting. Obviously, the specific form of $f_e(\cdot)$ and $g_e(\cdot)$ is closely related to the specific $f_d(\cdot)$. For the tracking control problem of system (3), the complete system state is denoted as $\mathbf{x} = [\mathbf{x}_e^\top, \mathbf{x}_d^\top]^\top$. Then there is $\mathbf{x} \in \mathbb{R}^n, n = n_1 + n_2$.

Remark 2. *Suppose that the reference path is generated by a separate mission planner, and $\mathbf{x}_{c_d}$ describes system dynamic parameters determined by the task setting, such as the moving speed of the VTP along the reference path. Then, it is reasonable to suppose that $f_d(\cdot)$ is known, which describes the shape of the reference path as well as the motion dynamics of the reference point along the path.*

Then, in the problem of curve-path-tracking control, given the reference motion state $\mathbf{x}_{k_d}$ corresponding to $\mathbf{p}_d$, denote the curvature of the reference path at this point as κ_d, and the speed of the point moving along the path as v_d . The dynamics of the reference states can be more specifically described as

$$\dot{\mathbf{x}}_d = f_d(\mathbf{x}_d) = \begin{bmatrix} \dot{\mathbf{x}}_{k_d} \\ \dot{v}_d \\ \dot{\kappa}_d \end{bmatrix} = \begin{bmatrix} f_{k_d}(\mathbf{x}_{k_d}, \kappa_d, v_d) \\ f_{v_d}(\mathbf{x}_{k_d}) \\ f_{\kappa_d}(\mathbf{x}_{k_d}) \end{bmatrix}. \tag{4}$$

Then the control objective is to find an optimal control policy $\mathbf{u}^*$ that consumes at the least cost to drive the tracking error $\mathbf{x}_e$ to converge to $\mathbf{0}$. To this end, take the objective function as

$$J(\mathbf{x}(0), \mathbf{u}) = \int_0^\infty [E(\mathbf{x}) + U(\mathbf{u})] \mathrm{d}\tau, \quad \mathbf{x}(0) \in \mathbb{X}, \tag{5}$$

where $\mathbb{X} \subseteq \mathbb{R}^n$ is a compact set containing the origin of the tracking error, $E(\mathbf{x}) = \mathbf{x}_e^T(t)\mathbf{Q}\mathbf{x}_e(t)$ is the quadratic tracking error cost with the positive definite diagonal matrix $\mathbf{Q}$, and $U(\mathbf{u})$ is the positive semi-definite control cost to be designed.

Now, referring to the concept in optimal control theory in [18], we define the admissible control for OTCP as follows.

Definition 1. *A control policy $\mathbf{u}(t) = \boldsymbol{\mu}(\mathbf{x}(t))$ is said to be admissible, denoted as $\mathbf{u}(t) \in \mathbb{U}$, with respect to objective function (5) for the tracking control system (3), if $\boldsymbol{\mu}(\mathbf{x}(t))$ is continuous on $\mathbb{X}$ and satisfies constraints (2), and the corresponding state trajectory $\mathbf{x}(t)$ makes $J(\mathbf{x}(0)) < \infty, \forall \mathbf{x}(0) \in \mathbb{X}$.*

Then, the main objective of this paper is to find the optimal control policy $\mathbf{u}^* \in \mathbb{U}$ that minimizes the objective function (5), and before we illustrate the design of solving $\mathbf{u}^*$, the following assumption is made in this paper.

Assumption 1. *For any initial state $\mathbf{x}(0) \in \mathbb{X}$, given the dynamic function $f_d(\cdot)$ of the reference state, there exists an admissible control $\mathbf{u}^{(0)} \in \mathbb{U}$, i.e., $\mathbf{u}^{(0)}$, which satisfies constraints (2), is continuous to $\mathbf{x}$ on set $\mathbb{X}$, and stabilizes the tracking error in (3).*

3. Optimal Control Design for Curve Path Tracking with Asymmetric Control Input Constraints

To find the optimal curve-path-tracking control policy $\mathbf{u}^*$ for system (3), this section first introduces a feedforward control law which helps to deal with the variation of the reference state dynamics. Then a dedicated design for a control cost function, which enables natural satisfactory of the asymmetric input constraint, is proposed (2).

Note that the main difficulty of curve-path-tracking control compared with that of regulation or straight/circular path tracking, is that the dynamics of the reference motion states $\mathbf{x}_{k_d}$ is time-varying because of the varying curvature of the reference path. To drive the tracking error to converge to $\mathbf{0}$, when $\mathbf{x}_e = \mathbf{0}$, it needs $\dot{\mathbf{x}}_e = \mathbf{0}$. The point is, different from regulation control problem, there needs to be a non-zero steady-state control law (denoted as $\bar{\mathbf{u}}$) because of the varying dynamics of $\mathbf{x}_{k_d}$, such that

$$\dot{\mathbf{x}}_e(t)|_{\mathbf{x}_e=\mathbf{0}} = f_e(\mathbf{0}, \mathbf{x}_d(t)) + g_e(\mathbf{0}, \mathbf{x}_d(t))\bar{\mathbf{u}}(\mathbf{0}, \mathbf{x}_d(t)) = \mathbf{0}. \tag{6}$$

It is easy to know that this non-zero steady-state control input $\bar{\mathbf{u}}$ mainly depends on the dynamics of reference states. Therefore, we rewrite the dynamic function of the reference motion state in (4) in the following form:

$$\dot{\mathbf{x}}_{k_d} = f_{k_d}(\mathbf{x}_{k_d}, \kappa_d, v_d) = f_k(\mathbf{x}_{k_d}) + g_k(\mathbf{x}_{k_d})\mathbf{u}_d(\mathbf{x}_{k_d}, \kappa_d, v_d). \tag{7}$$

Importing (7) and (1) into (6), we can obtain

$$\bar{\mathbf{u}}(\mathbf{0}, \mathbf{x}_d) = \mathbf{u}_d(\mathbf{x}_{k_d}, \kappa_d, v_d).$$

Then for $\mathbf{x}_e \neq \mathbf{0}$, we extend the above result to define the feedforward control $\bar{\mathbf{u}}(\mathbf{x})$ as

$$\bar{\mathbf{u}}(\mathbf{x}|\mathbf{x}_d) = \mathbf{u}_d(\mathbf{x}_{k_d}, \kappa_d, v_d). \tag{8}$$

Remark 3.

1. *The rewriting of (7) is reasonable for practical robotic systems since the reference state as well as the associated constraint conditions are well-concerned by the separate mission planner and will be illustrated by examples in later experiments.*
2. *The feedforward control law $\bar{\mathbf{u}}$ here is not an admissible control policy, which cannot drive a non-zero tracking error to $\mathbf{0}$, but is to be taken as a part of the control policy for the tracking control system.*

Now, this paper explains how to solve the desired optimal tracking control strategy $\mathbf{u}^*$ that satisfies the asymmetric control input constraint (2) in a simplified way by using $\bar{\mathbf{u}}$.

Given the dynamic function $f_d(\cdot)$ of reference states, $\bar{\mathbf{u}}$ can be obtained in real time according to (8). Then, the complete tracking control policy can be described as

$$\mathbf{u} \triangleq \bar{\mathbf{u}} + \tilde{\mathbf{u}},$$

where $\tilde{\mathbf{u}}$ is the feedback control to be solved. Importing $\tilde{\mathbf{u}}$ into the tracking error state equation in (3) generates

$$\begin{aligned}\dot{\mathbf{x}}_e &= f_e(\mathbf{x}_e, \mathbf{x}_d) + g_e(\mathbf{x}_e, \mathbf{x}_d)(\bar{\mathbf{u}} + \tilde{\mathbf{u}}) \\ &= \bar{f}_e(\mathbf{x}_e, \mathbf{x}_d) + g_e(\mathbf{x}_e, \mathbf{x}_d)\tilde{\mathbf{u}},\end{aligned} \tag{9}$$

where

$$\bar{f}_e(\mathbf{x}_e, \mathbf{x}_d) = f_e(\mathbf{x}_e, \mathbf{x}_d) + g_e(\mathbf{x}_e, \mathbf{x}_d)\bar{\mathbf{u}}.$$

Thus it holds that $\bar{f}_e(\mathbf{0}, \mathbf{x}_d) = \mathbf{0}$. Then, to solve the optimal control policy $\mathbf{u}^*$ is actually equivalent to solve the optimal feedback control $\tilde{\mathbf{u}}^* \triangleq \mathbf{u}^* - \bar{\mathbf{u}}$.

Therefore, in the consideration of control input constraint (2), referring to [10,16,19], the control cost in (5) is designed as

$$U(\mathbf{u}) = \tilde{U}(\tilde{\mathbf{u}}) = 2\sum_{j=1}^{m} \int_0^{\tilde{\mathbf{u}}^j} \tilde{\lambda}_j r_j \tanh^{-1}(s/\tilde{\lambda}_j)\mathrm{d}s, \tag{10}$$

which is a semi-positive definite function, and the greater the absolute value of the control input component $\tilde{\mathbf{u}}^j$, the greater the function value. So, being a part of the objective function, it can help to find an energy-optimal solution, and $r_j > 0$ is the weight coefficient with reference to component j. The main difference of (10) compared with that in [10,16,19] is that the threshold parameter in the integrand, i.e., $\tilde{\lambda}_j \geq 0$, is not a constant directly obtained from a symmetric control constraint but redefined for the asymmetric constraint (2) with the introduced feedforward control law as

$$\tilde{\lambda}_j = \begin{cases} -(\lambda_j^{min} - \bar{\mathbf{u}}^j), & \text{if} \quad \tilde{\mathbf{u}}^j < 0. \\ \lambda_j^{max} - \bar{\mathbf{u}}^j, & \text{if} \quad \tilde{\mathbf{u}}^j \geq 0. \end{cases} \tag{11}$$

This design allows for the natural satisfaction of the asymmetric control input constraint 2, which will be illustrated later in Lemma 1.

Then for tracking control system (3) subject to asymmetric control input constraint (2), given an initial state $\mathbf{x}(0) \in \mathbb{X}$ and the objective fuction (5) with (10), we define the optimal value function $V^*(\mathbf{x}) \in \mathcal{C}^1$ as

$$V^*(\mathbf{x}(t)) = \min_{\mathbf{u}\in\mathbb{U}} J(\mathbf{x}(t), \mathbf{u}) = \min_{\tilde{\mathbf{u}}|\bar{\mathbf{u}}+\tilde{\mathbf{u}}\in\mathbb{U}} J(\mathbf{x}(t), \tilde{\mathbf{u}}). \tag{12}$$

Correspondingly, the Hamiltonian is constructed as

$$\begin{aligned}H(\mathbf{x}, \tilde{\mathbf{u}}, V^*) &= E(\mathbf{x}) + \tilde{U}(\tilde{\mathbf{u}}) + (\nabla V^*)^\top \dot{\mathbf{x}} \\ &= E(\mathbf{x}) + \tilde{U}(\tilde{\mathbf{u}}) + (\nabla_{\mathbf{x}_e} V^*)^\top \left[\bar{f}_e(\mathbf{x}_e, \mathbf{x}_d) + g_e(\mathbf{x}_e, \mathbf{x}_d)\tilde{\mathbf{u}}\right],\end{aligned}$$

where $\nabla V^* = \frac{\partial V^*}{\partial \mathbf{x}}, \nabla_{\mathbf{x}_e} V^* = \frac{\partial V^*}{\partial \mathbf{x}_e}$. Then, according to the principle of optimality, $\tilde{\mathbf{u}}^*$ satisfies

$$\begin{aligned}H(\mathbf{x}, \tilde{\mathbf{u}}^*, V^*) &= E(\mathbf{x}) + \tilde{U}(\tilde{\mathbf{u}}^*) + (\nabla_{\mathbf{x}_e} V^*)^\top \left[\bar{f}_e(\mathbf{x}_e, \mathbf{x}_d) + g_e(\mathbf{x}_e, \mathbf{x}_d)\tilde{\mathbf{u}}^*\right] \\ &= 0.\end{aligned} \tag{13}$$

Then using the stationary condition, the optimal feedback control $\tilde{\mathbf{u}}^*$ can be obtained as

$$\tilde{\mathbf{u}}^* = -\mathbf{\Lambda} \tanh\left[\frac{1}{2}(\mathbf{\Lambda}\mathbf{R})^{-1} g_e^\top(\mathbf{x})\nabla_{\mathbf{x}_e} V^*\right], \tag{14}$$

where $\mathbf{\Lambda}, \mathbf{R} \in \mathbb{R}^{m\times m}$ are diagonal matrices constructed by $\tilde{\lambda}_j$ and $r_j, (j \in \{1, \cdots, m\})$, respectively, i.e.,

$$\mathbf{\Lambda} = \begin{bmatrix} \tilde{\lambda}_1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \tilde{\lambda}_m \end{bmatrix},$$

$$\boldsymbol{R} = \begin{bmatrix} r_1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & r_m \end{bmatrix}.$$

Then the optimal tracking control policy $\mathbf{u}^* \triangleq \bar{\mathbf{u}} + \tilde{\mathbf{u}}^*$ is

$$\mathbf{u}^* = \bar{\mathbf{u}} - \mathbf{\Lambda} \tanh\left[\frac{1}{2}(\mathbf{\Lambda}\mathbf{R})^{-1} g_e^\top(\mathbf{x}) \nabla_{\mathbf{x}_e} V^*\right]. \tag{15}$$

Importing $\tilde{\mathbf{u}}^*$ into (10), we can obtain the optimal control cost

$$\begin{aligned} \tilde{U}(\tilde{\mathbf{u}}^*) &= 2\sum_{j=1}^{m} \int_0^{\tilde{\mathbf{u}}^{j^*}} \tilde{\lambda}_j r_j (\tanh^{-1}(s/\tilde{\lambda}_j))^T \mathrm{d}s \\ &= (\nabla_{\mathbf{x}_e} V^*)^\top g_e(\mathbf{x}) \mathbf{\Lambda} \tanh(D^*) + \mathrm{diag}^\top(\mathbf{\Lambda}\mathbf{R}\mathbf{\Lambda}) \ln(\mathbf{1}_2 - \tanh^2(D^*)), \end{aligned} \tag{16}$$

where $D^* = 1/2(\mathbf{\Lambda}\mathbf{R})^{-1} g_e^\top(\mathbf{x}) \nabla_{\mathbf{x}_e} V^*$, $\mathrm{diag}^\top(\cdot)$ represents the vector constructed by the matrix main diagonal elements, $\mathbf{1}_2 = (1, 1)^\top$.

Further, importing (16) into (13), the tracking HJB equation turns into

$$E(\mathbf{x}) + (\nabla_{\mathbf{x}_e} V^*)^\top \bar{f}_e(\mathbf{x}_e, \mathbf{x}_d) + \mathrm{diag}^\top(\mathbf{\Lambda}\mathbf{R}\mathbf{\Lambda}) \ln(\mathbf{1}_2 - \tanh^2(D^*)) = 0. \tag{17}$$

Then if one can obtain the solution V^* by solving (17), (15) would provide the desired optimal tracking control policy.

Now we propose the following lemma.

Lemma 1. *With the non-quadratic control cost function (10), the optimal control policy* $\mathbf{u}^*$ *in (15) satisfies the asymmetric constraint (2) naturally.*

Proof. Under Assumption 1, there exists an admissible control $\mathbf{u}^{(0)} \in \mathbb{U}$ such that

$$\lambda_j^{min} \leq \mathbf{u}^{j^{(0)}} \leq \lambda_j^{max}. \tag{18}$$

Denote $\mathbf{u}^{(0)}$ as

$$\mathbf{u}^{(0)} = \tilde{\mathbf{u}}^{(0)} + \bar{\mathbf{u}}.$$

Since $\mathbf{u}^{(0)}$ is an admissible control law, according to Definition 1, there must be

$$\begin{aligned} \dot{\mathbf{x}}_e(t)|_{\mathbf{x}_e=\mathbf{0}} &= f_e(\mathbf{0}, \mathbf{x}_d(t)) + g_e(\mathbf{0}, \mathbf{x}_d(t))\mathbf{u}^{(0)}(t) \\ &= f_e(\mathbf{0}, \mathbf{x}_d(t)) + g_e(\mathbf{0}, \mathbf{x}_d(t))\left[\tilde{\mathbf{u}}^{(0)} + \bar{\mathbf{u}}\right] \\ &= \mathbf{0} + g_e(\mathbf{0}, \mathbf{x}_d(t))\tilde{\mathbf{u}}^{(0)} \\ &= \mathbf{0}. \end{aligned} \tag{19}$$

Thus we have

$$\tilde{\mathbf{u}}^{(0)}(\mathbf{0}, \mathbf{x}_d) = \mathbf{0}.$$

Putting $\tilde{\mathbf{u}}^{(0)}$ into (18) generates

$$\lambda_j^{min} \leq \bar{\mathbf{u}}^j(\mathbf{0}, \mathbf{x}_d) \leq \lambda_j^{max}.$$

Then according to definition of $\tilde{\lambda}_j$ in (11) and the extended feedforward control defined in (8), we have

$$\tilde{\lambda}_j \geq 0.$$

Since $-1 \leq \tanh(\cdot) \leq 1$, according to (14) and (11), the feedback control $\tilde{\mathbf{u}}^*$ satisfies

$$\begin{cases} \tilde{\mathbf{u}}^{j^*} \geq -\tilde{\lambda}_j = \lambda_j^{min} - \bar{\mathbf{u}}^j, & \text{when} \quad \tilde{\mathbf{u}}^{j^*} < 0. \\ \tilde{\mathbf{u}}^{j^*} \leq \tilde{\lambda}_j = \lambda_j^{max} - \bar{\mathbf{u}}^j, & \text{when} \quad \tilde{\mathbf{u}}^{j^*} \geq 0. \end{cases} \tag{20}$$

Then combining (20) with (15), we have

$$\lambda_j^{min} \leq \mathbf{u}^{j^*} \leq \lambda_j^{max}.$$

This completes the proof. □

Next, the following theorem provides the optimality and stability analysis of $\mathbf{u}^*$.

Theorem 1. *For tracking control system (3), given the dynamics function $f_d(\cdot)$ of the reference state, initial state $\mathbf{x}(0) \in \mathbb{X}$ and the objective function (5) with (10), assume V^* is a smooth positive definite solution to (17), then the optimal control policy given by (15) has the following properties:*

- *$\forall \mathbf{u} \in \mathbb{U}$, $\mathbf{u}^*$ minimizes objective function $J(\mathbf{x}(0), \mathbf{u})$;*
- *$\mathbf{u}^*$ stabilizes the tracking error $\mathbf{x}_e$ gradually.*

Proof. First, we prove that $\mathbf{u}^*$ minimizes the objective function J.

Given the initial state $\mathbf{x}(0)$ and the solution of HJB equation (17) as V^*, it holds that

$$\int_0^\infty \dot{V}^*(\mathbf{x}(t))\mathrm{d}t = -V^*(\mathbf{x}(0)). \tag{21}$$

Thus for any admissible control $\mathbf{u} = \bar{\mathbf{u}} + \tilde{\mathbf{u}}$, the corresponding objective function (5) can be represented as

$$J(\mathbf{x}(0), \mathbf{u}) = \int_0^\infty \left[E(\mathbf{x}) + \tilde{U}(\tilde{\mathbf{u}})\right]\mathrm{d}t + \int_0^\infty \dot{V}^*(\mathbf{x}(t))\mathrm{d}t + V^*(\mathbf{x}(0)). \tag{22}$$

Deriving V^* alone the state trajectory corresponding to $\mathbf{u}_i$, we have

$$\dot{V}^*(\mathbf{x}(t)) = \frac{\partial V^*}{\partial \mathbf{x}}\dot{\mathbf{x}} = (\nabla_{\mathbf{x}_e} V^*)^\top \left[\bar{f}_e(\mathbf{x}_e, \mathbf{x}_d) + g_e(\mathbf{x}_e, \mathbf{x}_d)\tilde{\mathbf{u}}\right],$$

and

$$\begin{aligned} J(\mathbf{x}(0), \mathbf{u}) = &\int_0^\infty \left[E(\mathbf{x}) + \tilde{U}(\tilde{\mathbf{u}})\right]\mathrm{d}\tau \\ &+ \int_0^\infty (\nabla_{\mathbf{x}_e} V^*)^\top \left[\bar{f}_e(\mathbf{x}_e, \mathbf{x}_d) + g_e(\mathbf{x}_e, \mathbf{x}_d)\tilde{\mathbf{u}}\right]\mathrm{d}\tau + V^*(\mathbf{x}(0)). \end{aligned} \tag{23}$$

By adding and subtracting $\int_0^\infty (\nabla_{\mathbf{x}_e} V^*)^\top g_e(\mathbf{x})\tilde{\mathbf{u}}^* \mathrm{d}\tau$ and $\int_0^\infty \tilde{U}(\tilde{\mathbf{u}}^*)\mathrm{d}\tau$ to the right side of the equation, it generates

$$\begin{aligned} &J(\mathbf{x}(0), \mathbf{u}) \\ = &\int_0^\infty \left[E(\mathbf{x}) + \tilde{U}(\tilde{\mathbf{u}}^*)\right]\mathrm{d}\tau + \int_0^\infty (\nabla_{\mathbf{x}_e} V^*)^\top \left[\bar{f}_e(\mathbf{x}_e, \mathbf{x}_d) + g_e(\mathbf{x}_e, \mathbf{x}_d)\tilde{\mathbf{u}}^*\right]\mathrm{d}\tau + V^*(\mathbf{x}(0)) \\ &+ \int_0^\infty (\nabla_{\mathbf{x}_e} V^*)^\top g_e(\mathbf{x}_e, \mathbf{x}_d)(\tilde{\mathbf{u}} - \tilde{\mathbf{u}}^*)\mathrm{d}\tau + \int_0^\infty \tilde{U}(\tilde{\mathbf{u}})\mathrm{d}\tau - \int_0^\infty \tilde{U}(\tilde{\mathbf{u}}^*)\mathrm{d}\tau. \end{aligned} \tag{24}$$

Then combine (24) with HJB equation (13), we further obtain

$$\begin{aligned} &J(\mathbf{x}(0),\mathbf{u}) \\ =&\int_0^\infty H(\mathbf{x},\tilde{\mathbf{u}}^*,V^*)\mathrm{d}\tau + V^*(\mathbf{x}(0)) + \int_0^\infty (\nabla_{\mathbf{x}_e}V^*)^\top g_e(\mathbf{x}_e,\mathbf{x}_d)(\tilde{\mathbf{u}}-\tilde{\mathbf{u}}^*)\mathrm{d}\tau \\ &+\int_0^\infty \tilde{U}(\tilde{\mathbf{u}})\mathrm{d}\tau - \int_0^\infty \tilde{U}(\tilde{\mathbf{u}}^*)\mathrm{d}\tau \\ =&V^*(\mathbf{x}(0)) + \int_0^\infty \left[(\nabla_{\mathbf{x}_e}V^*)^\top g_e(\mathbf{x})(\tilde{\mathbf{u}}-\tilde{\mathbf{u}}^*) + 2\sum_{j=1}^m \int_{\tilde{\mathbf{u}}^{j*}}^{\tilde{\mathbf{u}}^j} \tilde{\lambda}_j r_j \tanh^{-1}(s/\tilde{\lambda}_j)\mathrm{d}s\right]\mathrm{d}\tau. \end{aligned} \tag{25}$$

Denote that

$$M = (\nabla_{\mathbf{x}_e}V^*)^\top g_e(\mathbf{x})(\tilde{\mathbf{u}}-\tilde{\mathbf{u}}^*) + 2\sum_{j=1}^m \int_{\tilde{\mathbf{u}}^{j*}}^{\tilde{\mathbf{u}}^j} \tilde{\lambda}_j r_j \tanh^{-1}(s/\tilde{\lambda}_j)\mathrm{d}s. \tag{26}$$

Then to prove that $\mathbf{u}^*$ minimizes J, one needs to prove that $M > 0$ for all admissible control $\mathbf{u} \neq \mathbf{u}^*$, and that $M = 0$ if and only if $\mathbf{u} = \mathbf{u}^*$.

Based on (14), there is

$$(\nabla_{\mathbf{x}_e}V^*)^\top g_e(\mathbf{x}) = -2\mathbf{\Lambda}\mathbf{R}\tanh^{-1}\Big((\mathbf{\Lambda}^{-1}\tilde{\mathbf{u}}^*)^\top\Big). \tag{27}$$

Then importing (27) to M, we obtain

$$\begin{aligned} M &= 2\Big[\mathbf{\Lambda}\boldsymbol{R}\tanh^{-1}(\mathbf{\Lambda}^{-1}\tilde{\mathbf{u}}^*)\Big]^\top(\tilde{\mathbf{u}}^*-\tilde{\mathbf{u}}) + 2\sum_{j=1}^m \int_{\tilde{\mathbf{u}}^{j*}}^{\tilde{\mathbf{u}}^j} \tilde{\lambda}_j r_j \tanh^{-1}(s/\tilde{\lambda}_j)\mathrm{d}s \\ &= 2\sum_{j=1}^m r_j\left[\tilde{\lambda}_j \tanh^{-1}(\tilde{\mathbf{u}}^{j*}/\tilde{\lambda}_j)(\tilde{\mathbf{u}}^{j*}-\tilde{\mathbf{u}}^j) + \int_{\tilde{\mathbf{u}}^{j*}}^{\tilde{\mathbf{u}}^j} \tilde{\lambda}_j \tanh^{-1}(s/\tilde{\lambda}_j)\mathrm{d}s\right]. \end{aligned} \tag{28}$$

To help to analyze, define a function $\varsigma_a(x_1,x_2)$ as

$$\varsigma_a(x_1,x_2) = a\tanh^{-1}(x_1/a)(x_1-x_2) + \int_{x_1}^{x_2} a\tanh^{-1}(s/a)\mathrm{d}s,$$

where $a > 0$, $-a \le x_1, x_2 \le a$. Since $\tanh^{-1}(\cdot)$ increases monotonically, when $x_1 < x_2$, there must be a $\hat{x} \in (x_1,x_2)$, such that

$$a\tanh^{-1}(\hat{x}/a)(x_2-x_1) = \int_{x_1}^{x_2} a\tanh^{-1}(s/a)\mathrm{d}s,$$

and that $\tanh^{-1}(\hat{x}/a) > \tanh^{-1}(x_1/a)$. Then importing $\tanh^{-1}(\hat{x}/a)$ into $\varsigma_a(x_1,x_2)$ generates

$$\begin{aligned} \varsigma_a(x_1,x_2) &= a\tanh^{-1}(x_1/a)(x_1-x_2) + a\tanh^{-1}(\hat{x}/a)(x_2-x_1) \\ &= a\Big(\tanh^{-1}(\hat{x}/a) - \tanh^{-1}(x_1/a)\Big)(x_2-x_1) \\ &> 0. \end{aligned} \tag{29}$$

Likewise, when $x_1 > x_2$, there must also be a $\hat{x} \in (x_2,x_1)$, such that

$$a\tanh^{-1}(\hat{x}/a)(x_1-x_2) = -\int_{x_1}^{x_2} a\tanh^{-1}(s/a)\mathrm{d}s,$$

and that $\tanh^{-1}(x_1/a) > \tanh^{-1}(\hat{x}/a)$. Then importing $\tanh^{-1}(\hat{x}/a)$ into $\varsigma_a(x_1,x_2)$ we have

$$\begin{aligned}\varsigma_a(x_1,x_2) &= a\tanh^{-1}(x_1/a)(x_1-x_2) - a\tanh^{-1}(\hat{x}/a)(x_1-x_2)\\ &= a\Big(\tanh^{-1}(x_1/a) - \tanh^{-1}(\hat{x}/a)\Big)(x_1-x_2)\\ &> 0.\end{aligned} \tag{30}$$

Further, when $x_1 = x_2$, it holds that $\varsigma_a(x_1,x_2) = 0$. That is, $\varsigma_a(x_1,x_2) = 0$ only when $x_1 = x_2$, and $\varsigma_a(x_1,x_2) > 0$, when $x_1 \neq x_2$.

Combining the above conclusion with (28), M can be represented as

$$M = 2\sum_{j=1}^{m} r_j \varsigma_{\bar{\lambda}_j}(\tilde{\mathbf{u}}^{j^*}, \tilde{\mathbf{u}}^{j}). \tag{31}$$

Then, there is

$$\begin{cases} M > 0 & \text{if} \quad \exists j \in \{1,\cdots,m\}, \quad \text{s.t.} \quad \tilde{\mathbf{u}}^{j} \neq \tilde{\mathbf{u}}^{j^*}.\\ M = 0 & \text{if} \quad \forall j \in \{1,\cdots,m\}, \quad \text{s.t.} \quad \tilde{\mathbf{u}}^{j} = \tilde{\mathbf{u}}^{j^*}.\end{cases} \tag{32}$$

Therefore, $J(\mathbf{x}(0),\mathbf{u}) \geq V^*(\mathbf{x}(0))$ holds for all $\mathbf{u} \in \mathbb{U}$, in which the $=$ holds only when $\mathbf{u} = \mathbf{u}^* \triangleq \bar{\mathbf{u}} + \tilde{\mathbf{u}}^*$.

Next, we prove that the tracking error $\mathbf{x}_e$ is gradually stabilized with $\mathbf{u}^*$.

Note that $V^*(\mathbf{x})$ is a positive semi-definite function. Take $V^*(\mathbf{x})$ as the Lyapunov function of the tracking control system (3), then there is

$$\dot{V}^*(\mathbf{x}(t)) = -\mathbf{x}_e^\top \mathbf{Q}\mathbf{x}_e - \tilde{U}(\tilde{\mathbf{u}}^*) \leq 0. \tag{33}$$

It is known from the proof of Lemma 1 that $\tilde{\mathbf{u}}^*(\mathbf{0},\mathbf{x}_d) = \mathbf{0}$, then the "=" in (33) holds only if $\mathbf{x}_e = \mathbf{0}$. Thus, $\mathbf{u}^*$ gradually stabilizes $\mathbf{x}_e$.

This completes the proof. □

4. IRL-Based Approximate Optimal Solution

The last section provides the design of the optimal tracking control policy $\mathbf{u}^*$. However, to solve $\mathbf{u}^*$ involves solving the HJB Equation (17), which is highly nonlinear to V^*. In consideration of the difficulty in solving (17), this section provides an NN-based IRL algorithm to obtain an approximate optimal solution.

With the optimal value function denoted as V^*, the following integral form of the value function is taken according to the idea of IRL:

$$V^*(\mathbf{x}(t)) = \int_t^{t+T} \Big[\mathbf{x}_e^\top \mathbf{Q}\mathbf{x}_e + \tilde{U}(\tilde{\mathbf{u}}^*)\Big]\mathrm{d}\tau + V^*(\mathbf{x}(t+T)), \tag{34}$$

where the integral reinforcement interval $T > 0$.

Then the IRL-based PI Algorithm 1 is presented as follows.

Algorithm 1 IRL-based optimal path-tracking algorithm

```
1: Policy evaluation: NN weights update
     $V^{(k)}(\mathbf{x}(t)) = \int_t^{t+T} \Big[\mathbf{x}_e^\top \mathbf{Q}\mathbf{x}_e + \tilde{U}\Big(\tilde{\mathbf{u}}^{(k)}\Big)\Big]\mathrm{d}\tau + V^{(k)}(\mathbf{x}(t+T)).$   (35)
2: Policy improvement:
     $\mathbf{u}^{(k+1)} = -\mathbf{\Lambda}\tanh\Big(D^{(k)}\Big) + \bar{\mathbf{u}},$   (36)
   where $D^{(k)} = 1/2(\mathbf{\Lambda}\mathbf{R})^{-1} g_e^\top(\mathbf{x})\nabla_{\mathbf{x}_e} V^{(k)}$.
```

Remark 4. *Equation (34) is equivalent to the HJB equation (17) in the way that (34) and (17) have the same positive definite solution V^*, and according to the result of traditional PI algorithm, given an initial admissible control $\mathbf{u}^{(0)}$, then for all $k \geq 0$, iteratively solving (35) for $V^{(k)}$, there always exists an admissible control $\mathbf{u}^{(k+1)}$ with (36), and when $k \to \infty$, $\mathbf{u}^{(k)}$ and $V^{(k)}$ uniformly converge to $\mathbf{u}^*$ and V^*[10,20].*

To implement Algorithm 1, this paper introduces a single-layer NN with p neurons to approximate the value function:

$$V^{(k)}(\mathbf{x}) = \left(\boldsymbol{W}_c^{(k)}\right)^\top \boldsymbol{\sigma}(\mathbf{x}) + \varepsilon(\mathbf{x}), \tag{37}$$

and

$$\nabla V^{(k)}(\mathbf{x}) = (\nabla\boldsymbol{\sigma}(\mathbf{x}))^\top \boldsymbol{W}_c^{(k)} + \nabla\varepsilon(\mathbf{x}), \tag{38}$$

where $\boldsymbol{W}_c^{(k)} \in \mathbb{R}^p$ is the optimal weight vector to approximate $V^{(k)}$, $\boldsymbol{\sigma}(\cdot) : \mathbb{R}^n \to \mathbb{R}^p$ is the vector of continuously differentiable bounded basis functions, and ε is the approximation error. Then, according to work in [10], when the number of neurons $p \to \infty$, the fitting error ε would be close to 0, and [21] points out that, even when the number of neurons is limited, the fitting error is still bounded. Therefore, ε and $\nabla\varepsilon$ are bounded over the compact set $\mathbb{X}$, i.e., there exist constants $b_\varepsilon > 0$ and $b_\varepsilon^{'} > 0$ such that $|\varepsilon(\mathbf{x})| \leq b_\varepsilon$, $\|\nabla\varepsilon(\mathbf{x})\| \leq b_\varepsilon^{'}$.

Putting (37) into (35), we obtain the tracking Bellman error as

$$\varepsilon_c^{(k)}(t) = \int_t^{t+T} \left[\mathbf{x}_e^\top \mathbf{Q}\mathbf{x}_e + \tilde{U}\left(\tilde{\mathbf{u}}^{(k)}\right)\right] \mathrm{d}\tau + \left(\boldsymbol{W}_c^{(k)}\right)^\top \Delta\boldsymbol{\sigma}(\mathbf{x}(t)), \tag{39}$$

where $\Delta\boldsymbol{\sigma}(\mathbf{x}(t)) = \boldsymbol{\sigma}(\mathbf{x}(t+T)) - \boldsymbol{\sigma}(\mathbf{x}(t))$. Then, there exists a positive constant ε_{max} such that $|\varepsilon_c{}^{(k)}(t)| \leq \varepsilon_{max}, \forall t \geq 0$.

Since the optimal weight vector $\boldsymbol{W}_c^{(k)}$ in (37) is unknown, the value function is approximated in the iteration as

$$\hat{V}^{(k)}(\mathbf{x}) = \left(\hat{\boldsymbol{W}}_c^{(k)}\right)^\top \sigma(\mathbf{x}), \tag{40}$$

where $\hat{\boldsymbol{W}}_c^{(k)}$ is the estimation of $\boldsymbol{W}_c^{(k)}$. Then, the estimation of $\varepsilon_c^{(k)}(t)$ is

$$\hat{e}_c^{(k)}(t) = \int_t^{t+T} \left[\mathbf{x}_e^\top \mathbf{Q}\mathbf{x}_e + \tilde{U}\left(\tilde{\mathbf{u}}^{(k)}\right)\right] \mathrm{d}\tau + \left(\hat{\boldsymbol{W}}_c^{(k)}\right)^\top \Delta\sigma(\mathbf{x}(t)). \tag{41}$$

To find the best weight vector $\boldsymbol{W}_c^{(k)}$ of $V^{(k)}$, the tuning law of the weight estimation $\hat{\boldsymbol{W}}_c^{(k)}$ should minimize the estimated Bellman error $\hat{e}_c^{(k)}$. Utilizing the gradient decent scheme and considering the objective function $E_c = \frac{1}{2}\left(\hat{e}_c^{(k)}\right)^2$, we take the tuning law for the weight vector as

$$\dot{\hat{\boldsymbol{W}}}_c^{(k)} = -\alpha_c\delta\frac{\partial E_c}{\partial\hat{\boldsymbol{W}}_c^{(k)}} = -\alpha_c\frac{\Delta\boldsymbol{\sigma}(\mathbf{x})}{(\Delta\boldsymbol{\sigma}^T(\mathbf{x})\Delta\boldsymbol{\sigma}(\mathbf{x})+1)^2}\hat{e}_c^{(k)}, \tag{42}$$

where $\alpha_c > 0$ is the learning rate, and $\delta = \frac{1}{(\Delta\boldsymbol{\sigma}^T(\mathbf{x})\Delta\boldsymbol{\sigma}(\mathbf{x})+1)^2}$ is used for normalization [16]. Then, taking the sampling period as equal to the integral reinforcement interval T, after each N sampling period, the NN weights of online IRL-based PI for the approximate tracking control policy after k^{th} iterations is updated by

$$\hat{\boldsymbol{W}}_c^{(k+1)} = \hat{\boldsymbol{W}}_c^{(k)} - \alpha_c\frac{1}{N}\sum_{j=0}^{N-1}\frac{\Delta\boldsymbol{\sigma}_j(\mathbf{x})}{\left(\Delta\boldsymbol{\sigma}_j^\top(\mathbf{x})\Delta\boldsymbol{\sigma}_j(\mathbf{x})+1\right)^2}\hat{e}_{c_j}^{(k)}.$$

Importing $\hat{\boldsymbol{W}}_c^{(k+1)}$ into (36), we obtain the improved control policy

$$\hat{\mathbf{u}}^{(k+1)} = -\boldsymbol{\Lambda} \tanh\left(\hat{D}^{(k+1)}\right) + \bar{\mathbf{u}}, \tag{43}$$

where $\hat{D}^{(k+1)} = 1/2(\boldsymbol{\Lambda}\mathbf{R})^{-1} g_e^\top(\mathbf{x})(\nabla\boldsymbol{\sigma}(\mathbf{x}))^\top \hat{\boldsymbol{W}}_c^{(k+1)}$. Then, given an initial approximated weight $\hat{\boldsymbol{W}}_c^{(0)}$ corresponding to an admissible initial control $\mathbf{u}^{(0)}$, the online IRL-based PI can be performed as in Figure 1.

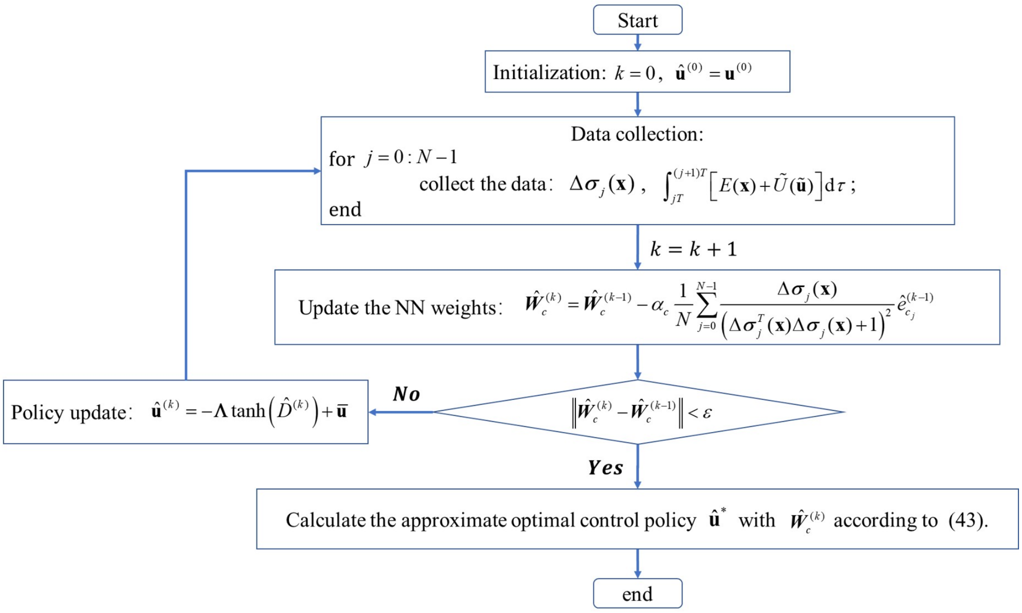

Start
Initialization: $k=0$, $\hat{\mathbf{u}}^{(0)}=\mathbf{u}^{(0)}$
Data collection:
for $j=0:N-1$
collect the data: $\Delta\boldsymbol{\sigma}_j(\mathbf{x})$, $\int_{jT}^{(j+1)T}\left[E(\mathbf{x})+\tilde{U}(\tilde{\mathbf{u}})\right]\mathrm{d}\tau$;
end
$k = k + 1$
Update the NN weights: $\hat{\boldsymbol{W}}_c^{(k)}=\hat{\boldsymbol{W}}_c^{(k-1)}-\alpha_c\frac{1}{N}\sum_{j=0}^{N-1}\frac{\Delta\boldsymbol{\sigma}_j(\mathbf{x})}{\left(\Delta\boldsymbol{\sigma}_j^T(\mathbf{x})\Delta\boldsymbol{\sigma}_j(\mathbf{x})+1\right)^2}\hat{e}_{c_j}^{(k-1)}$
$\left\|\hat{\boldsymbol{W}}_c^{(k)}-\hat{\boldsymbol{W}}_c^{(k-1)}\right\|<\varepsilon$
No
Policy update: $\hat{\mathbf{u}}^{(k)}=-\Lambda\tanh\left(\hat{D}^{(k)}\right)+\bar{\mathbf{u}}$
Yes
Calculate the approximate optimal control policy $\hat{\mathbf{u}}^*$ with $\hat{\boldsymbol{W}}_c^{(k)}$ according to (43).
end

Figure 1. The flowchart of the online integral reinforcement learning (IRL)-based policy iteration algorithm for approximate optimal tracking control policy.

Remark 5. *Let $\mathbf{u}^{(0)}$ be any admissible bounded control policy in the algorithm in Figure 1, and take (42) as the tuning law of the critic NN weights. If $\Delta\bar{\sigma} = \Delta\sigma(\mathbf{x})/(\Delta\sigma^T(\mathbf{x})\Delta\sigma(\mathbf{x})+1)$ is persistently exciting (PE), i.e., if there exist $\gamma_1 > 0$ and $\gamma_2 > 0$ such that $\forall t > 0$*

$$\gamma_1 \boldsymbol{I} \le \int_t^{t+T} \Delta\bar{\sigma}\Delta\bar{\sigma}^T \mathrm{d}\tau \le \gamma_2 \boldsymbol{I}, \tag{44}$$

where $\boldsymbol{I}$ is the unit matrix, then for the bounded reconstruction error $\varepsilon_c{}^{(k)}$ in (41), the critic weight estimation error $\tilde{\boldsymbol{W}}_c^{(k)} = \boldsymbol{W}_c^{(k)} - \hat{\boldsymbol{W}}_c^{(k)}$ converges exponentially fast to a residual set [13–15].

5. Application to Fixed-Wing UAVs

This section verifies the proposed method on the OTCP for fixed-wing UAVs curve path tracking in HIL simulations in comparison with three other typical path tracking algorithms.

5.1. Problem Formulation

The system state of fixed-wing UAVs denoted by $\mathbf{x}_k = (x, y, \psi)^\top$ includes the position of the UAV in the inertial system $\mathbf{p} = (x, y)^\top$ and the heading angle ψ. The control input $\mathbf{u}$ comprises the airspeed u_v and heading rate u_ω, which are constrained by

$$\begin{aligned} v_{stall} \le u_v \le v_{max}, \\ -\omega_{max} \le u_\omega \le \omega_{max}, \end{aligned} \tag{45}$$

where $v_{stall} > 0$ is the minimum stall speed, v_{max}, and ω_{max} are the maximum speed and heading rate, respectively, determined by executor features.

Given the VTP $\mathbf{p}_d(t)$ at time t, the corresponding reference motion state $\mathbf{x}_{k_d}(t) = (x_d, y_d, \psi_d)^\top$ is then designated. Let the VTP move at a constant speed v_d along the reference path. Then, denote the curve length of one point with reference to the start point along the path as l. Given the parameterized function $\mathcal{Q}(l)$ of the reference path, the curvature κ_d at $\mathbf{p}_d(t)$ can be calculated. Then, the reference state dynamics are obtained:

$$\dot{\mathbf{x}}_d = \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\psi}_d \\ \dot{v}_d \\ \dot{\kappa}_d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ f_{\kappa_d}(\mathbf{x}_{k_d}|\mathcal{Q}(l)) \end{bmatrix} + \begin{bmatrix} \cos\psi_d & 0 \\ \sin\psi_d & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_d \\ v_d\kappa_d \end{bmatrix}. \tag{46}$$

Then, the feedforward control law is

$$\bar{\mathbf{u}} = \begin{bmatrix} v_d \\ v_d\kappa_d \end{bmatrix}.$$

Define the tracking error in a local Frenet–Serret coordinate system $\{F\}$ as $\mathbf{x}_e = (x_e, y_e, \psi_e)^\top$ [22,23]. Then, let $\mathbf{Q} = \mathbf{I}_{3\times3}$ and $\mathbf{R} = \mathbf{I}_{2\times2}$. The goal is to solve the optimal control $\mathbf{u}^*$ that minimizes the objective function (5) with (10).

5.2. Approximate Optimal Control Policy Learning

This subsection utilizes the proposed method to find an approximate optimal policy for OTCP of fixed-wing UAVs formulated in the last subsection.

The learning process is carried out on Matlab 2018. Table 1 presents the parameter settings, and the nonlinear kinematics of fixed-wing UAVs is modeled by

$$\dot{\mathbf{x}}_k = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_v \\ u_\omega \end{bmatrix}. \tag{47}$$

Given coordinates of five waypoints, the reference curve path is generated using the third order B-spline curve algorithm (See Figure 2a). Given the reference state of the start point on the reference path, the initial state of the UAV is randomly chosen within $x_e(0), y_e(0) \in [-50, 50], \psi_e(0) \in [-\pi, \pi]$. The basis for the value function approximation is selected as

$$\boldsymbol{\sigma} = \begin{bmatrix} x_e & y_e & \psi_e & x_e y_e & x_e\psi_e & y_e\psi_e & x_e^2 & y_e^2 & \psi_e^2 \end{bmatrix}^\top. \tag{48}$$

The value function NN weights are initialized as

$$\hat{W}_c^{(0)} = \begin{bmatrix} 0.1 & 0.1 & -0.5 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.5 \end{bmatrix}^\top, \tag{49}$$

which corresponds to an admissible but non-optimal control policy $\mathbf{u}^{(0)}$. Given the initial NN weights and the corresponding admissible initial control policy, the tracking data are collected online, and the NN weights are updated once a batch of a specific amount of data is collected according to the flow in Figure 1.

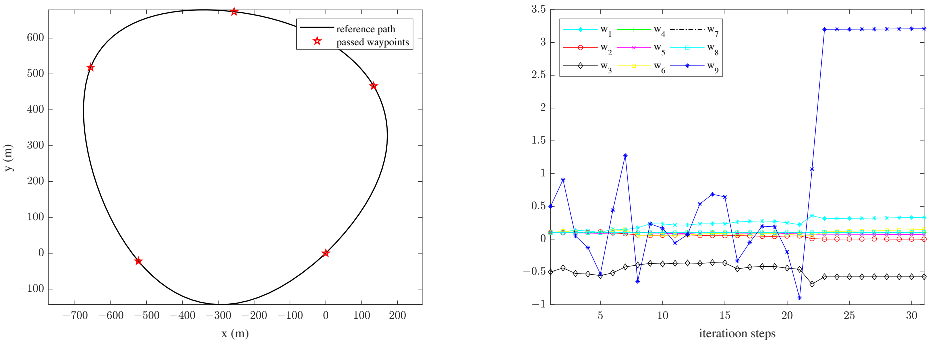
Table 1. Parameter settings for optimal policy learning.

Symbol	Value	Meaning
v_d (m/s)	19	the cruising speed of the UAV
v_{stall} (m/s)	14	the minimum stall speed of the UAV
v_{max} (m/s)	24	the maximum speed of the UAV
ω_{max} (rad/s)	0.6	the maximum heading rate

The iterative process of critic NN weight estimates are provided in Figure 2b, which converges to a steady value in 23 steps, and the final NN weights are

$$\hat{\boldsymbol{W}}_c^{(23)} = \begin{bmatrix} 0.329 & 0.002 & 0.574 & 0.102 & 0.074 & 0.120 & 0.100 & 0.103 & 3.204 \end{bmatrix}^\top, \tag{50}$$

which provides an approximate optimal path-tracking control policy for fixed-wing UAVs. In the process of policy training, we found that w_3 and w_9 demonstrate stronger oscillation compared with other NN weights, which is also presented in Figure 2b. This is because both of the corresponding activation functions are one-variable functions of the heading angle error ψ_e, which is set to be within $[-\pi, \pi]$ during the training, and the value ranges of x_e and y_e are set to be $[-300, 300]$. Thus, there is no unified metric for the three components. As a result, the weight of the activation function would be much more sensitive to variation of the approximated function value.



(**a**) Reference path

(**b**) NN weights iteration

Figure 2. The reference path for policy learning and the neural network (NN) weights iteration.

5.3. HIL Simulation Test and Result Analysis

To fully validate the effectiveness of the proposed method on OTCP of fixed-wing UAVs, the learned control policy was tested on a high-fidelity HIL simulation system in comparison with three other typical path-tracking algorithms [5]: the pure pursuit and line of sight algorithm (PLOS), the nonlinear Lyapunov guidance method (NLGL), and the backstepping control method (BS). The HIL simulation system consists of a swarm control station, the host computer, a Pixhawk autopilot, a QGround Control, and an X-Plane aircraft simulator. Specifically, the swarm control station, which is used to give task instructions and displays the current status of the system, was developed by the authors' team. The host computer was used to simulate the onboard computer of the physical aircraft, which receives and processes task instructions from the control station and state information from onboard sensors and generates and sends control commands to Pixhawk. The Pixhawk autopilot is a widely-used open-source autopilot, and it processes and generates control commands for the underlying actuators and collects and sends back the sensor data. The X-Plane aircraft simulator, which is a high-fidelity aircraft simulator, provides the physical engine and dynamics simulation of the UAV, and the QGround Control performs as an information transfer station between X-Plane and Pixhawk (see Figure 3 for the flow of control commands and sate information).

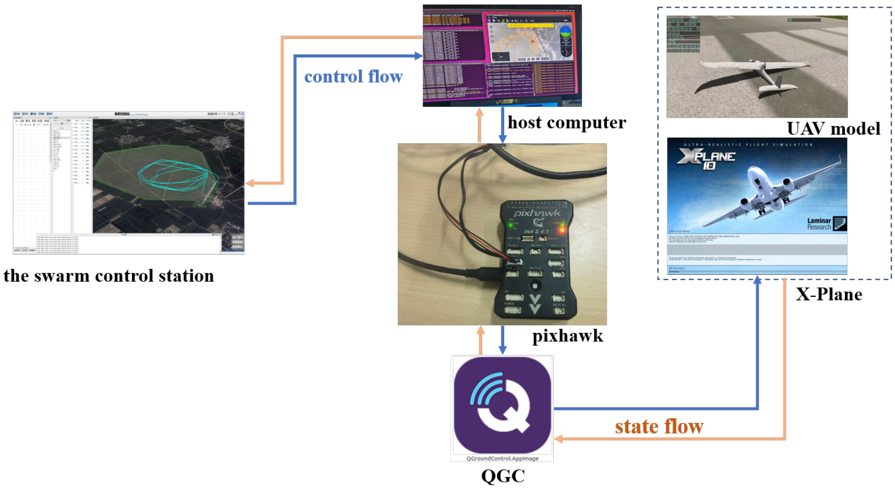


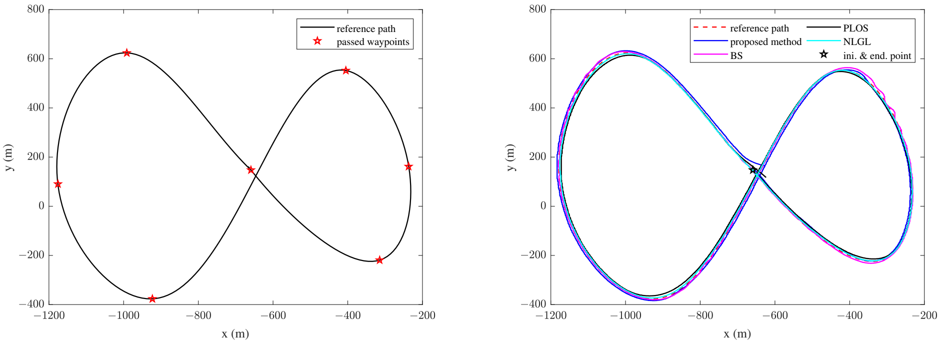
Figure 3. The high-fidelity hardware-in-the-loop (HIL) simulation system.

Note that:

1. The reference path in HIL simulations shown in Figure 4a is generated by QGround Control with eight waypoints (provided in Table 2) on an experimental airport, which is different from that used for policy learning and has larger curvature changes.
2. Speed constraints in the aircraft simulator during the test were $10 \leq u_v \leq 18$, different from settings in policy learning (which is the same as a practical UAV platform).

Table 2. Waypoints of the reference path in HIL simulation tests.

	WP 1	**WP 2**	**WP 3**	**WP 4**
Latitude Longitude	34.0245 113.7068	34.0288 113.7032	34.0240 113.7012	34.0198 113.7040
	WP 5	**WP 6**	**WP 7**	**WP 8**
Latitude Longitude	34.0282 113.7096	34.0247 113.7114	34.0212 113.7106	34.0245 113.7068



(**a**) (**b**)

Figure 4. The reference path and tracking trajectories in HIL simulation tests: (**a**) reference path; (**b**) tracking trajectory.

In spite of the abovementioned differences between settings of policy learning process and HIL simulation, the learned control policy provided satisfying tracking performance in the comparison HIL simulation. The path-tracking trajectories are presented in Figure 4b, which shows that all of the four algorithms can stably track the reference curve path. Figures 5 and 6 further show the heading and the cross-tracking errors of the four algorithms. From these two figures, we can see that the learned control policy using the proposed method leads to a smooth curve-path-tracking trajectory with a small lateral steady-state tracking error and near zero heading and forward steady-state tracking errors. Moreover, heading tracking errors of BS, PLOS, and NLGL, the forward tracking error of BS, and the lateral tracking error of PLOS and NLGL, show significant fluctuations compared with the proposed method, especially when the UAV moves up to the corners of the reference path. This is because the heading tracking error and the information of curvature variation of the reference path are not considered in the three algorithms. Therefore, the three algorithms cannot achieve a satisfactory curve-path-tracking control performance as in the straight-line and circular path-tracking control problems, and the proposed method can provide more stable and smooth tracking performance. Figure 6 also shows that both PLOS and NLGL algorithms have a significant steady-state forward error. The main reason is that, the tracking performance of the two algorithms is very dependent on the update rule of the VTP, which is required to be updated ahead of a distance before the UAV's arrival, and the algorithms would fail to track the path if this distance is not large enough (such as, smaller than about 20 m). Finally, Figure 7 provides the control input using the provided method, which verifies that the input constraints are naturally satisfied instead of being forcibly cut down during the whole path-tracking period.

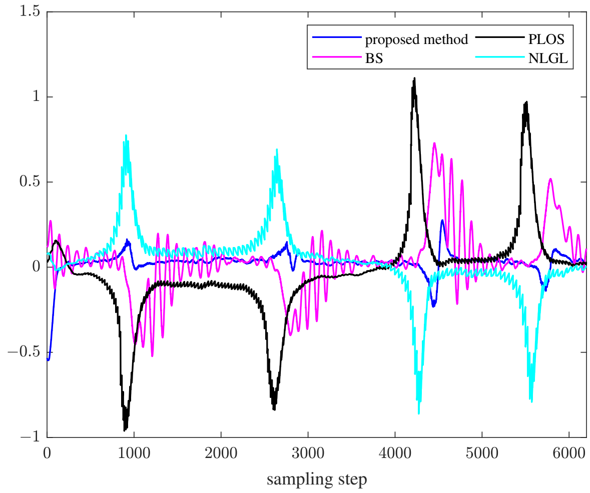


Figure 5. The heading error comparison.

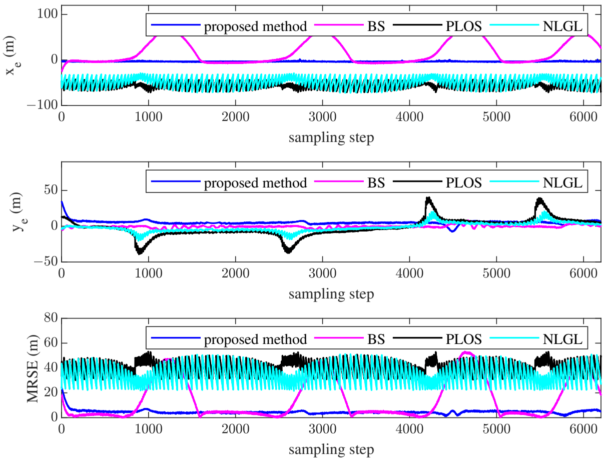


Figure 6. The cross-tracking error and the root mean squared error comparison.

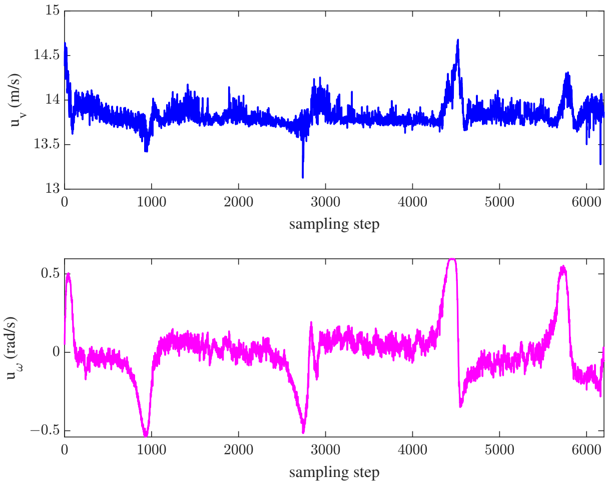


Figure 7. The control input using the proposed method.

6. Conclusions

This paper developed an approximate optimal control scheme for OTCP of nonlinear systems with asymmetric input constraints. Especially, the difficulty brought by the varying curvature of the curve reference path is handled by introducing a feedforward control law. The effectiveness was verified in a high-fidelity HIL system for fixed-wing UAVs. The result confirmed the effectiveness and generalization of the learned control policy and indicates the capability of ADP theory in complicated nonlinear systems. Future work will study the robust control of such control systems under external disturbance.

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

Funding: This research was funded by National Natural Science Foundation of China grant number 61973309; Natural Science Foundation of Hunan Province grant number 2021JJ10053 and Hunan Provincial Innovation Foundation for Postgraduate grant number CX20210009.

Data Availability Statement: Not applicable.

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

References

1. Yang, J.; Liu, C.; Coombes, M.; Yan, Y.; Chen, W.H. Optimal Path Following for Small Fixed-Wing UAVs under Wind Disturbances. *IEEE Trans. Control Syst. Technol.* **2021**, *29*, 996–1008. [CrossRef]
2. Kang, J.G.; Kim, T.; Kwon, L.; Kim, H.D.; Park, J.S. Design and Implementation of a UUV Tracking Algorithm for a USV. *Drones* **2022**, *6*, 66. [CrossRef]
3. Ratnoo, A.; Sujit, P.B.; Kothari, M. Adaptive Optimal Path Following for High Wind Flights. *IFAC Proc. Vol.* **2011**, *44*, 12985–12990. [CrossRef]
4. Lin, F.; Chen, Y.; Zhao, Y.; Wang, S. Path Tracking of Autonomous Vehicle Based on Adaptive Model Predictive Control. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1–12. [CrossRef]
5. Sujit, P.B.; Saripalli, S.; Sousa, J.B. Unmanned Aerial Vehicle Path Following: A Survey and Analysis of Algorithms for Fixed-Wing Unmanned Aerial Vehicles. *IEEE Control Syst. Mag.* **2014**, *34*, 42–59.
6. Chen, S.; Chen, H.; Negrut, D. Implementation of MPC-Based Path Tracking for Autonomous Vehicles Considering Three Vehicle Dynamics Models with Different Fidelities. *Automot. Innov.* **2020**, *3*, 386–399. [CrossRef]
7. Rucco, A.; Aguiar, A.P.; Pereira, F.L.; de Sousa, J.B. A Predictive Path-Following Approach for Fixed-Wing Unmanned Aerial Vehicles in Presence of Wind Disturbances. *Adv. Intell. Syst. Comput.* **2016**, *417*, 623–634. [CrossRef]

8. Alessandretti, A.; Aguiar, A.P. A Planar Path-Following Model Predictive Controller for Fixed-Wing Unmanned Aerial Vehicles. In Proceedings of the 11th International Workshop on Robot Motion and Control (RoMoCo), Wasowo, Poland, 3–5 July 2017; pp. 59–64. [CrossRef]
9. Chen, H.; Cong, Y.; Wang, X.; Xu, X.; Shen, L. Coordinated Path-Following Control of Fixed-Wing Unmanned Aerial Vehicles. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 2540–2554. [CrossRef]
10. Abu-Khalaf, M.; Lewis, F.L. Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach. *Automatica* **2005**, *41*, 779–791. [CrossRef]
11. Powell, W.B. *Approximate Dynamic Programming*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2007. 10.1002/9780470182963. [CrossRef]
12. Yang, X.; He, H.; Liu, D.; Zhu, Y. Adaptive Dynamic Programming for Robust Neural Control of Unknown Continuous-Time Non-Linear Systems. *IET Control Theory Appl.* **2017**, *11*, 2307–2316. [CrossRef]
13. Jiang, H.; Zhang, H.; Luo, Y.; Han, J. Neural-Network-Based Robust Control Schemes for Nonlinear Multiplayer Systems with Uncertainties via Adaptive Dynamic Programming. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 579–588. [CrossRef]
14. Vamvoudakis, K.G.; Lewis, F.L. Online Actor-Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem. *Automatica* **2010**, *46*, 878–888. [CrossRef]
15. Vrabie, D.; Lewis, F. Neural Network Approach to Continuous-Time Direct Adaptive Optimal Control for Partially Unknown Nonlinear Systems. *Neural Netw.* **2009**, *22*, 237–246. [CrossRef] [PubMed]
16. Modares, H.; Lewis, F.L. Optimal Tracking Control of Nonlinear Partially-Unknown Constrained-Input Systems Using Integral Reinforcement Learning. *Automatica* **2014**, *50*, 1780–1792. [CrossRef]
17. Yan, J.; Yu, Y.; Wang, X. Distance-Based Formation Control for Fixed-Wing UAVs with Input Constraints: A Low Gain Method. *Drones* **2022**, *6*, 159. [CrossRef]
18. Lewis, F.L.; Vrabie, D.L.; Syrmos, V.L. *Optimal Control*, 3rd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2012. [CrossRef]
19. Adhyaru, D.M.; Kar, I.N.; Gopal, M. Bounded Robust Control of Nonlinear Systems Using Neural Network–Based HJB Solution. *Neural Comput. Appl.* **2010**, *20*, 91–103. [CrossRef]
20. Liu, D.; Yang, X.; Li, H. Adaptive Optimal Control for a Class of Continuous-Time Affine Nonlinear Systems with Unknown Internal Dynamics. *Neural Comput. Appl.* **2012**, *237*, 2012, 23, 1843–1850. [CrossRef]
21. Hornik, K.; Stinchcombe, M.; White, H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Netw.* **1990**, *3*, 551–560. [CrossRef]
22. Aguiar, A.P.; Hespanha, J.P.; Kokotović, P.V. Performance Limitations in Reference Tracking and Path Following for Nonlinear Systems. *Automatica* **2008**, *44*, 598–610. [CrossRef]
23. Wang, Y.; Wang, X.; Zhao, S.; Shen, L. Vector Field Based Sliding Mode Control of Curved Path Following for Miniature Unmanned Aerial Vehicles in Winds. *J. Syst. Sci. Complex.* **2018**, *31*, 302–324. [CrossRef]