


Article

NN-Based Parallel Model Predictive Control for a Quadrotor UAV

Jun Qi^{1,2,3}, Jiru Chu¹, Zhao Xu^{1,*}, Cong Huang³ and Minglei Zhu³ ¹ China Nuclear Power Engineering Co., Ltd., Beijing 100840, China² School of Automation, Chengdu University of Information Technology, Chengdu 610225, China³ School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

* Correspondence: xuzhaore01@163.com

Abstract: A novel neural network (NN)-based parallel model predictive control (PMPC) method is proposed to deal with the tracking problem of the quadrotor unmanned aerial vehicles (Q-UAVs) system in this article. It is well known that the dynamics of Q-UAVs are changeable while the system is operating in some specific environments. In this case, traditional NN-based MPC methods are not applicable because their model networks are pre-trained and kept constant throughout the process. To solve this problem, we propose the PMPC algorithm, which introduces parallel control structure and experience pool replay technology into the MPC method. In this algorithm, an NN-based artificial system runs in parallel with the UAV system to reconstruct its dynamics model. Furthermore, the experience replay technology is used to maintain the accuracy of the reconstructed model, so as to ensure the effectiveness of the model prediction algorithm. Furthermore, a convergence proof of the artificial system is also given in this paper. Finally, numerical results and analysis are given to demonstrate the effectiveness of the PMPC algorithm.

Keywords: MPC; neural networks; Q-UAVs; adaptive learning; parallel control



Citation: Qi, J.; Chu, J.; Xu, Z.; Huang, C.; Zhu, M. NN-Based Parallel Model Predictive Control for a Quadrotor UAV. *Processes* **2023**, *11*, 1706. <https://doi.org/10.3390/pr11061706>

Academic Editors: Qiuye Sun, Rui Wang and Zhengmao Li

Received: 14 April 2023

Revised: 19 May 2023

Accepted: 24 May 2023

Published: 2 June 2023



Copyright: © 2023 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

In the past decade, with the development of sensor and network technologies, Q-UAVs have attracted more and more attention from researchers. Due to the characteristics of efficient deployment and flexible mobility, Q-UAVs are widely used in the fields of nuclear power plant inspections [1,2], forest fire fighting [3,4] and land surveying [5,6]. However, it is challenging to design an effective control for the Q-UAVs system, because of the problems of under-actuating, strong non-linearity, and coupling. Recently, numerous algorithms have been developed by researchers under the framework of traditional control, such as feedback linearization control [7,8], fuzzy logic-based control [9,10], and sliding mode control (SMC) [11,12]. In [13], a non-linear PID controller is designed to deal with the tracking control problem of the Q-UAVs system, where the system energy is considered. A fuzzy-based backstepping SMC algorithm is developed to deal with the tracking problem of the UAVs with parameter uncertainties and external disturbances in [14]. In [15], a second-order SMC algorithm has been introduced to design the Q-UAV controller; however, the optimization characteristics, as an essential part of the control domain, are not considered in the above-mentioned traditional control algorithms.

It is well known, MPC is widely used in many fields as an effective optimal control method, such as the robot control [16,17], autonomous driving [18,19], and energy management [20,21], etc. Under the MPC algorithm scheme, the future system states and behaviors are predicted by the model of the plant, and the control law is optimized based on the analysis and evaluation of the prediction data. Due to obtaining the optimal control by minimizing the performance cost function which is subject to constraints, MPC also has a good performance in the robustness control. Many researchers have also introduced

the MPC algorithm into the field of Q-UAV control. In [22], a novel non-linear MPC is proposed to deal with the navigation problem where obstacle avoidance is considered. According to the MPC algorithm scheme, it can be seen that the implementation of the MPC algorithm relies on the accuracy of the prediction model. MPC algorithm is applied to find the optimal paths in the atmosphere with the maximum of the UAV's energy [23]. In [24], a novel adaptive MPC scheme is established for the angular rate and thrust control of a Q-UAV, where the thrust bound and the disturbances are considered; however, it is difficult to accurately model the dynamics of Q-UAVs due to the high complexity of the system.

Neural networks are widely used to approximate the unknown function for its universal approximation characteristics [25–28]. Under the MPC scheme, NN technology is introduced to approximate the dynamics model of the Q-UAVs. Many researchers have also made many contributions in this field. In [29], a novel reinforcement learning-based MPC algorithm is developed to deal with the tracking control problem with thrust vectoring capabilities. An offline learning approach is applied to learning the dynamics model of the hybrid Q-UAV in [30]. A Q-UAV controller is designed based on the learning-based MPC algorithm, which is used to provide levels of guarantees about safety, robustness, and convergence in [31]. Alessandro developed an active learning algorithm to deal with the control problem where the model of the Q-UAV is uncertainty-aware [32]. Several novel meta-learning approaches have been developed to approximate the distribution over different “tasks” [33,34]. In the research mentioned above, the NN model is pre-trained before the system runs, and the weight matrices are kept constant during the system operation. However, in some specific environments, such as nuclear radiation and forest fire environments, the dynamics model of the Q-UAVs changes as the system operates. The algorithms mentioned above are unreliable in this case.

Motivated by the above-mentioned problems, we propose a novel NN-based PMPC algorithm. The main contributions are summarized as follows:

- (1) Different from the exiting results [29], the model NN is not only used to approximate the position model of the Q-UAVs but also the whole dynamics model of the Q-UAVs. With our PMPC algorithm, the dynamics model of the Q-UAVs is fully unknown and which will make the optimal problem more difficult to be solved.
- (2) Different from the exiting results [30], the model NN is executed in parallel with the system and is continuously updated with the system runs in the PMPC algorithm. Compared with the unchangeable NN in [30,33,34], our algorithm can be applied in the condition where the dynamics model is changeable during the system running.
- (3) The experiment replay technology is introduced to the PMPC algorithm, which is used to maintain the accuracy of the reconstructed model to ensure the effectiveness of the model prediction algorithm.

The remainder of this article is organized as follows. The problem is formulated in Section 2. In Section 3, the parallel structure and the details of the PMPC algorithm are introduced. Numerical results and analysis are given to demonstrate the effectiveness of the PMPC algorithm in Section 4. Finally, the conclusion is presented in Section 5.

2. Problem Formulation

2.1. Dynamic Models of Q-UAVs

The dynamic models of Q-UAVs have been studied as a system with six degrees of freedom. Under the low speeds assumption, a simple, rigid-body model [35] of Q-UAVs is defined as

$$\begin{aligned}
 \ddot{x} &= (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{U_1}{m} \\
 \ddot{y} &= (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \frac{U_1}{m} \\
 \ddot{z} &= -g + (\cos \phi \cos \theta) \frac{U_1}{m} \\
 \ddot{\theta} &= \dot{\phi} \dot{\psi} \left(\frac{I_z - I_x}{I_y} \right) + \frac{J_R}{I_y} \dot{\phi} \Omega_R + \frac{L}{I_y} U_3 \\
 \ddot{\phi} &= \dot{\theta} \dot{\psi} \left(\frac{I_y - I_z}{I_x} \right) - \frac{J_R}{I_x} \dot{\theta} \Omega_R + \frac{L}{I_x} U_2 \\
 \ddot{\psi} &= \dot{\phi} \dot{\theta} \left(\frac{I_x - I_y}{I_z} \right) + \frac{1}{I_z} U_4
 \end{aligned} \tag{1}$$

where $\mathbf{S} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$, and $\mathbf{U} = [U_1, U_2, U_3, U_4]^T$ indicate the state vector and control input of the system, respectively. Furthermore, as shown in Figure 1, $\mathbf{P} = [x, y, z]$, and $\Theta = [\phi, \theta, \psi]$ denote the position and Euler angles in the inertial frame Γ_E , respectively; $\mathbf{V} = [\dot{x}, \dot{y}, \dot{z}]$ and $\mathbf{V}_a = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$ denote the velocity of the three axes and the angular velocity of three Euler angles, respectively; I_x, I_y , and I_z are the moments of inertia of the Q-UAV around three axes, respectively; and m is the mass of the Q-UAV; L is the length from the rotors to the center of mass; J_R and Ω_R are the moments of inertia and angular velocity of the propeller blades. Moreover, U_1, U_2, U_3 , and U_4 are the forces generated by the four propellers.

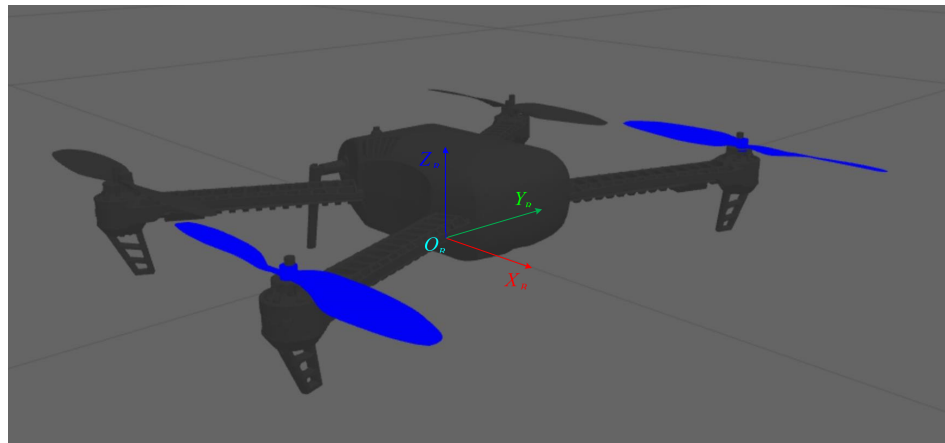


Figure 1. The sketch of Q-UAV with body frame \mathcal{O}_B .

2.2. Model Predictive Control

Consider a non-linear system defined as

$$\mathbf{r}_{k+1} = \mathfrak{F}(\mathbf{r}_k, \mathbf{u}_k) \tag{2}$$

where \mathbf{r}_k and \mathbf{u}_k are denoted the state and the control input of the system at the time step k , respectively.

Select the n_p as the finite prediction horizon steps, MPC is applied to minimize the performance cost over the n_p steps at each time step k . The optimized control sequence $\mathcal{U} = [\mathbf{u}_k, \mathbf{u}_{k+1}, \dots, \mathbf{u}_{k+n_p-1}]$ can be obtained by solving the optimal function. The first

term in the optimized control sequence is applied in the process and the rest will be applied to solve the optimal control function at the next time step $k + 1$, and so on.

2.3. Problem Statement

Under the traditional MPC scheme, the performance of the MPC algorithm is highly dependent on the accuracy of the model. However, the above model is constructed without consideration of the influences of some other sources, such as air drag, the gyroscope effect, and the hub force. To overcome the difficulty of the inaccurate model, a novel NN-based parallel control method is proposed in this paper.

3. Parallel MPC Method

In this section, the PMPC algorithm is introduced to obtain the optimal controller to deal with the tracking control problem for the Q-UAVs with a dynamic dynamics model.

3.1. Algorithm Structure

In the PMPC algorithm, an artificial system is introduced to expand the real problem, which runs in parallel with the real system. The artificial space, between the artificial and real system, is introduced to solve the expanded real problem. The structure of the parallel MPC method can be shown in Figure 2.

From Figure 2, the parallel MPC method can be divided into three steps. Firstly, an artificial system is introduced to rebuild the dynamic model of the real system by learning the data observed from the actual system. The artificial system can keep the accuracy of the model with periodic learning of the data in the experiment pool. This is used to ensure the accuracy of the predictive experiments throughout the actual system run. Secondly, based on the artificial system, predictive experiments are performed to analyze the behavior of the Q-UAV system and evaluate the performance of the control laws. Based on the evaluated results, the optimal control law is updated. Thirdly, the appropriate control is applied to the real system with the interactional execution between the artificial and real systems.

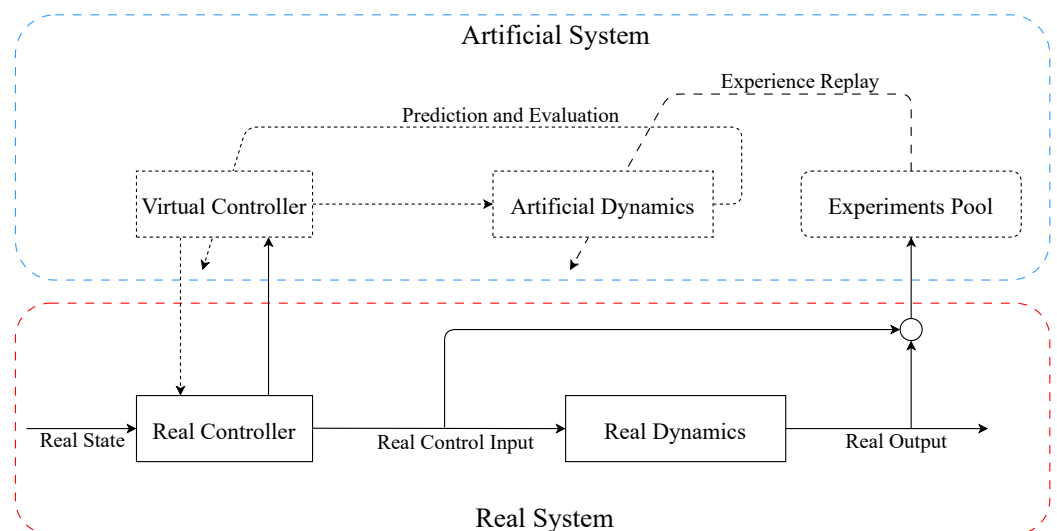


Figure 2. The structure of parallel MPC method.

3.2. Artificial System

Due to it being difficult to obtain accurate dynamic models of Q-UAVs, an artificial system, based on neural networks, is introduced to rebuild the dynamic model of the real system.

The discrete-time dynamic model of the Q-UAVs can be generally written as [36]

$$\mathbf{S}_{k+1} = F(\mathbf{S}_k, \mathbf{U}_k) \quad (3)$$

where \mathbf{X}_k and \mathbf{U}_k are the system state and control input at k instant.

According to the universal approximation property of the neural network, the ideal neural network representation of the system (3) can be written as

$$\mathbf{S}_{k+1} = \mathcal{W}_m^{*T} \Psi(\Phi(\mathbf{T}_k)) + \epsilon_{mk} \quad (4)$$

where $\mathbf{T}_k = [\mathbf{S}_k, \mathbf{U}_k]^T$, \mathcal{W}_m^* and ϵ_{mk} are the neural network input, the ideal weight matrix, and the reconstruction error, respectively. Furthermore, $\Psi(\cdot)$ and $\Phi(\cdot)$ are the activation functions.

The artificial system can be defined as:

$$\mathbf{S}_{k+1} = \mathcal{F}(\mathbf{S}_k, \mathbf{U}_k) \quad (5)$$

which also can be written in the form of a neural network

$$\hat{\mathbf{S}}_{k+1} = \hat{\mathcal{W}}_m^T \Psi(\Phi(\mathbf{T}_k)) \quad (6)$$

where $\hat{\mathcal{W}}_m$ is the approximation weight matrix of \mathcal{W}_m^* . The training process of the artificial system is expressed as Algorithm 1.

Algorithm 1 Neural Networks-Based Artificial System

Initialization:

- 1: Collect the data set as

$$\Omega_k = \{\mathbf{S}_{k-1}, \mathbf{S}_{k-2}, \dots, \mathbf{S}_{k-\mathcal{N}}\} \quad (7)$$

where \mathcal{N} is a large positive integer.

- 2: Create a neural network as (6).
- 3: Select the accuracy of modeling ζ .

Training:

- 4: Calculate the training error as

$$\tilde{\mathbf{S}}_k = \hat{\mathbf{S}}_k - \mathbf{S}_k \quad (8)$$

- 5: Adjust the weights to minimize the following error

$$E_m = \frac{1}{2\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \tilde{\mathbf{S}}_{k-i}^T \tilde{\mathbf{S}}_{k-i} \quad (9)$$

- 6: Update the weight matrix with the following function:

$$\hat{\mathcal{W}}_{m,k} = \hat{\mathcal{W}}_{m,k-1} - l_m \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \Psi(\Phi(\mathbf{S}_{k-i-1})) \tilde{\mathbf{S}}_{k-i}^T \quad (10)$$

with l_m denoting the learning rate.

- 7: Until the following function

$$\begin{aligned} & \|(\hat{\mathcal{W}}_{m,k-1} - \hat{\mathcal{W}}_{m,k})^T (\hat{\mathcal{W}}_{m,k-1} - \hat{\mathcal{W}}_{m,k}) \\ & + \tilde{\mathbf{S}}_k^T \tilde{\mathbf{S}}_k \| \leq \zeta \end{aligned} \quad (11)$$

is satisfied, return the weight matrix \mathcal{W}_m .

Next, we will prove the convergence of the artificial system. Before proceeding, the following assumptions are necessary.

Assumption 1. The reconstruction error ϵ_{mk} satisfied the condition as

$$\epsilon_{mk}^T \epsilon_{mk} \leq \gamma_m \tilde{\mathbf{S}}_k^T \tilde{\mathbf{S}}_k \quad (12)$$

Assumption 2. The activation function $\Psi(\cdot)$ and the reconstruction error ϵ_{mk} are bounded by the positive constant α_M and β_M , respectively.

Theorem 1. Let the artificial system be defined as (6), and let the weight matrix $\hat{\mathcal{W}}_m$ be updated according to (10). If Assumptions 1 and 2 hold, then the system identification error $\tilde{\mathbf{X}}_k$ is asymptotically stable and the error matrices $\tilde{\mathcal{W}}_m$ and both converge to zero, as $k \rightarrow \infty$.

Proof. Consider the artificial system (6), and select the Lyapunov function as

$$\mathcal{L}_m = \tilde{\mathbf{S}}_k^T \tilde{\mathbf{S}}_k + \frac{1}{l_m} \text{tr}\{\tilde{\mathcal{W}}_{m,k}^T \tilde{\mathcal{W}}_{m,k}\} \quad (13)$$

The difference of (13) can be written as

$$\begin{aligned} \Delta \mathcal{L}_m &= \tilde{\mathbf{S}}_{k+1}^T \tilde{\mathbf{S}}_{k+1} - \tilde{\mathbf{S}}_k^T \tilde{\mathbf{S}}_k \\ &+ \frac{1}{l_m} \text{tr}\{\tilde{\mathcal{W}}_{m,k+1}^T \tilde{\mathcal{W}}_{m,k+1} - \tilde{\mathcal{W}}_{m,k}^T \tilde{\mathcal{W}}_{m,k}\} \end{aligned} \quad (14)$$

Substituting (6) and (10) into (14), we can obtain

$$\begin{aligned} \Delta \mathcal{L}_m &= (\tilde{\mathcal{W}}_m^T \Psi(\Phi(\mathbf{T}_k)))^T \tilde{\mathcal{W}}_m^T \Psi(\Phi(\mathbf{T}_k)) + \epsilon_{mk}^T \epsilon_{mk} \\ &- 2(\tilde{\mathcal{W}}_m^T \Psi(\Phi(\mathbf{T}_k)))^T \epsilon_{mk} - \tilde{\mathbf{S}}_k^T \tilde{\mathbf{S}}_k \\ &+ l_m \Psi(\Phi(\mathbf{T}_k))^T \Psi(\Phi(\mathbf{T}_k)) \tilde{\mathbf{X}}_{k+1}^T \tilde{\mathbf{X}}_{k+1} \end{aligned} \quad (15)$$

According to the Cauchy–Schwarz inequality, we have

$$\begin{aligned} \Delta \mathcal{L} &\leq (\tilde{\mathcal{W}}_m^T \Psi(\Phi(\mathbf{T}_k)))^T \tilde{\mathcal{W}}_m^T \Psi(\Phi(\mathbf{T}_k)) + \epsilon_{mk}^T \epsilon_{mk} \\ &- \tilde{\mathbf{S}}_k^T \tilde{\mathbf{S}}_k + 2l_m \Psi(\Phi(\mathbf{T}_k))^T \Psi(\Phi(\mathbf{T}_k)) \\ &\times ((\tilde{\mathcal{W}}_m^T \Psi(\Phi(\mathbf{T}_k)))^T \tilde{\mathcal{W}}_m^T \Psi(\Phi(\mathbf{T}_k)) + \epsilon_{mk}^T \epsilon_{mk}) \end{aligned} \quad (16)$$

Based on the Assumption 2, we can obtain

$$\begin{aligned} \Delta \mathcal{L} &\leq - (1 - 2l_m \alpha_M^2) \|\tilde{\mathcal{W}}_m^T \Psi(\Phi(\mathbf{T}_k))\|^2 \\ &- (1 - \gamma_m - 2l_m \gamma_m \alpha_M^2) \|\tilde{\mathbf{S}}_k\|^2 \end{aligned} \quad (17)$$

Select l_m to satisfy the condition as

$$l_m < \min\left\{\frac{1}{2\alpha_M^2}, \frac{1 - \gamma_m}{2\gamma_m \alpha_M^2}\right\} \quad (18)$$

Then, we have $\Delta \mathcal{L}_m < 0$, which means the system identification error $\tilde{\mathbf{S}}_k$ is asymptotically stable and the error matrices $\tilde{\mathcal{W}}_m$ and both converge to zero, as $k \rightarrow \infty$. \square

3.3. Predictive Experiments

After reconstructing the actual system, predictive experiments are executed to gain the optimal control of the artificial system (5). In the predictive experiments, the neural networks are applied to predict the state of the artificial system in future N steps. Thus, the system (5) can be rewritten as

$$\mathbf{Z}_{k+1} = \mathcal{F}(\mathbf{Z}_k, \mathbf{U}_k) \quad (19)$$

where \mathbf{Z}_k denote the predictive state of the system (5).

Let the control sequence \mathcal{U}_k^{N-1} defined as $\mathcal{U}_k^{N-1} = \{\mathbf{U}_k, \mathbf{U}_{k+1}, \dots, \mathbf{U}_{k+N-1}\}$ to be the predictive control in future N steps. Furthermore, the utility function is defined as

$$\Gamma(\mathbf{U}_k, e_k) = \mathbf{U}_k^T \mathbf{Q} \mathbf{U}_k + e_k^T \mathbf{R} e_k \quad (20)$$

where $e_k = \mathbf{S}_k - \mathbf{S}_k^d$ with \mathbf{S}_k^d denote the target state at k instant; \mathbf{Q} and \mathbf{R} are semi-positive definite matrices with suitable dimensions. Thus, the cost function can be defined as

$$\mathcal{J}_k = \sum_{i=0}^{N-1} \Gamma(\mathbf{U}_{k+i}, e_{k+i}) \quad (21)$$

Then, the optimal control sequence can be obtained as

$$\mathcal{U} = \left\{ \arg \min_{\mathbf{U}_i} \sum_{j=0}^{N-1} \Gamma(\mathbf{U}_{i+j}, e_{i+j}) \right\}_{i=k}^{\infty} \quad (22)$$

However, to maintain the safety and stability of the flight process, the following constraint should be applied in the optimization.

3.3.1. Input Constraint

In the parallel control scheme, the limits of the input of the Q-UAV is considered for the safety of the flight process. The constraint for the Q-UAV can be defined as

$$\mathbf{U}_{min} \leq \mathbf{U}_k \leq \mathbf{U}_{max} \quad (23)$$

where the \mathbf{U}_{min} and \mathbf{U}_{max} denote the minimum and maximum of the input vector at the instant k .

3.3.2. Velocity Constraint

In the algorithm proposed in this article, the velocity of the Q-UAV is limited, which can be defined as

$$\begin{aligned} \dot{x}_{min} &\leq \dot{x} \leq \dot{x}_{max} \\ \dot{y}_{min} &\leq \dot{y} \leq \dot{y}_{max} \\ \dot{z}_{min} &\leq \dot{z} \leq \dot{z}_{max} \end{aligned} \quad (24)$$

where $\dot{\zeta}_{max}$ and $\dot{\zeta}_{min}$ denote the minimum and maximum of the velocity of the Q-UAV in ζ axis.

3.3.3. Angle Velocity Constraint

In this article, the limits of the angular velocities are also considered in the proposed algorithm, which can be defined as follows.

$$\begin{aligned} \dot{\phi}_{min} &\leq \dot{\phi} \leq \dot{\phi}_{max} \\ \dot{\theta}_{min} &\leq \dot{\theta} \leq \dot{\theta}_{max} \\ \dot{\psi}_{min} &\leq \dot{\psi} \leq \dot{\psi}_{max} \end{aligned} \quad (25)$$

Similarly to (24), $\dot{\phi}_{min}$, $\dot{\theta}_{min}$, and $\dot{\psi}_{min}$ denote the minimum angle velocity of three Euler angles, respectively; and $\dot{\phi}_{max}$, $\dot{\theta}_{max}$ and $\dot{\psi}_{max}$ denote the maximum angle velocity of three Euler angles, respectively.

Then, the parallel predictive experiment method under the constraints can be expressed as Algorithm 2.

Algorithm 2 Parallel Experiment Algorithm

Initialization:

- 1: Select the semi-positive matrices P and Q to construct the cost function as (21).
- 2: Select the positive constant N as the control horizon.

Rolling Prediction:

- 3: Collect data with the artificial system (6) based on current control law.
 - 4: Calculate the cost function by (21).
 - 5: Obtain the optimized control sequence \mathcal{U} according to the function (22) with the constraint (23).
 - 6: Implement the first element of the control sequence into the artificial system (6).
-

3.4. Parallel Execution

In the above section, the predictive experiments are executed to gain the optimal control of the artificial system (5), which aims to control the discrete-time dynamic model (3) of the Q-UAVs. However, for the real Q-UAV system, it is difficult to rebuild the system dynamics in the whole time horizon by a single artificial system, because the system function is complex and unknown. Thus, to keep the accuracy of the model, the dataset in (7) is updated with time, which means the dataset always stores the latest \mathcal{N} data.

Combining the above sections, the Parallel MPC method can be expressed as Algorithm 3.

Algorithm 3 Parallel MPC algorithm

Initialization:

- 1: Select the positive integer \mathcal{N} as the maximum length of the experience pool.
- 2: Select the semi-positive matrices P and Q .
- 3: Select the positive integer N as the control horizon.
- 4: Select the computation precision ζ .
- 5: Select the initial instant $k = 0$.
- 6: Select the positive integer l as the length of the experience pool.

Parallel Predictive Control:

- 7: **if** $l < \mathcal{N}$ **then**
 - 8: Store the last state \mathbf{X}_{k-1} , control input \mathbf{U}_{k-1} and the system state \mathbf{X}_k , as a data pair, into the experience pool.
 - 9: **else**
 - 10: Remove the first data pair from the experience pool.
 - 11: Store the newest data pair in the experience pool.
 - 12: **end if**
 - 13: Training the artificial system with the experience pool until $\|(\hat{\mathcal{W}}_{m,k-1} - \hat{\mathcal{W}}_{m,k})^T(\hat{\mathcal{W}}_{m,k-1} - \hat{\mathcal{W}}_{m,k}) + \tilde{\mathbf{X}}_k^T \tilde{\mathbf{X}}_k\| \leq \zeta$ is satisfied.
 - 14: Predict the system state based on the artificial system in the control horizon N .
 - 15: Calculate the cost value based on the function (21) for every predictive instant.
 - 16: Calculate the optimal control based on the cost value.
 - 17: Applying the first optimal control into the actual system.
-

4. Simulation Results

In this section, to demonstrate the effectiveness of the PMPC algorithm for UAVs, a comparison with traditional MPC is performed. The initial mass of the Q-UAV is chosen as $m = 6.98 \times 10^{-2}$ kg; the moment of the inertia of the Q-UAV around the axes are chosen as $I_x = I_y = I_z = 3.4 \times 10^{-3}$ kg·m², respectively; the moments of inertia of the propeller blades are chosen as $J_R = 1.302 \times 10^{-6}$ kg·m²; the length from the rotors to the center of mass is chosen as $L = 1.17 \times 10^{-1}$ m; and the acceleration of gravity is chosen as 9.8 m/s².

The parameters in this simulation are chosen as

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (26)$$

and the learning rate of the modelNN is chosen as $lr = 0.01$, the activation function $\Psi(\cdot)$ and $\Phi(\cdot)$ are defined as sigmoid function. To reflect the internal dynamic uncertainties of the UAV dynamic model, 5% parameter inaccuracies are assumed during the model prediction process. Some constraints in this simulation are shown as Figure 3.

The target trajectory is chosen as

$$\begin{aligned} x &= 2\cos(0.05\pi t) \\ y &= 2\sin(0.05\pi t) \\ z &= 0.2t \end{aligned} \quad (27)$$

It is worth mentioning that the target Euler angles are defined to keep track of the target trajectory in this simulation. Furthermore, the constraints of the position velocity in this simulation are chosen as 1 m/s and the angular velocity is chosen as $\pi/15$ rad/s. The rotating speed of rotors ω is bounded in the range of $[110, 860]$ r/s.

As shown in Figure 4, under the PMPC algorithm, the UAV tracked the target trajectory in 15 seconds. However, when it is under traditional MPC control, it cannot follow the target trajectory if the dynamics model is changeable. This means the PMPC algorithm has better performance when dealing with the tracking control problem with a dynamic dynamics model. To demonstrate the robustness of PMPC algorithm, the external disturbances are added at 20 s, and the results are shown in Figure 5. From Figure 5, the system can track the target trajectory in several seconds under the PMPC algorithm.

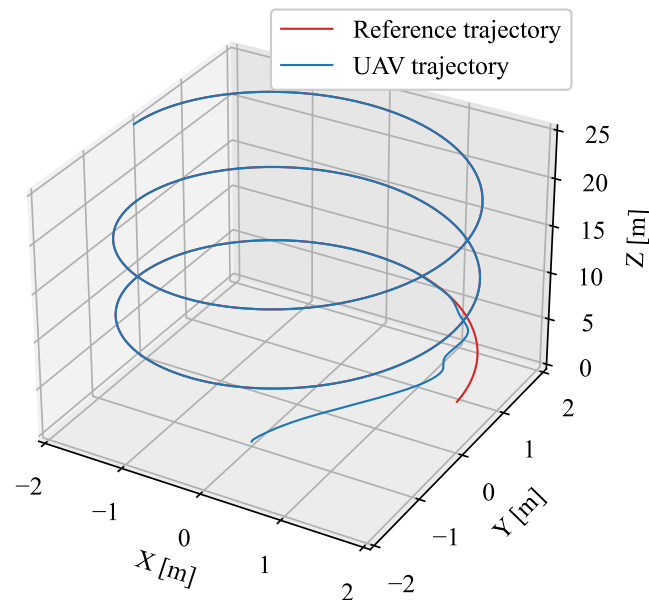


Figure 3. The tracking histories of the UAV.

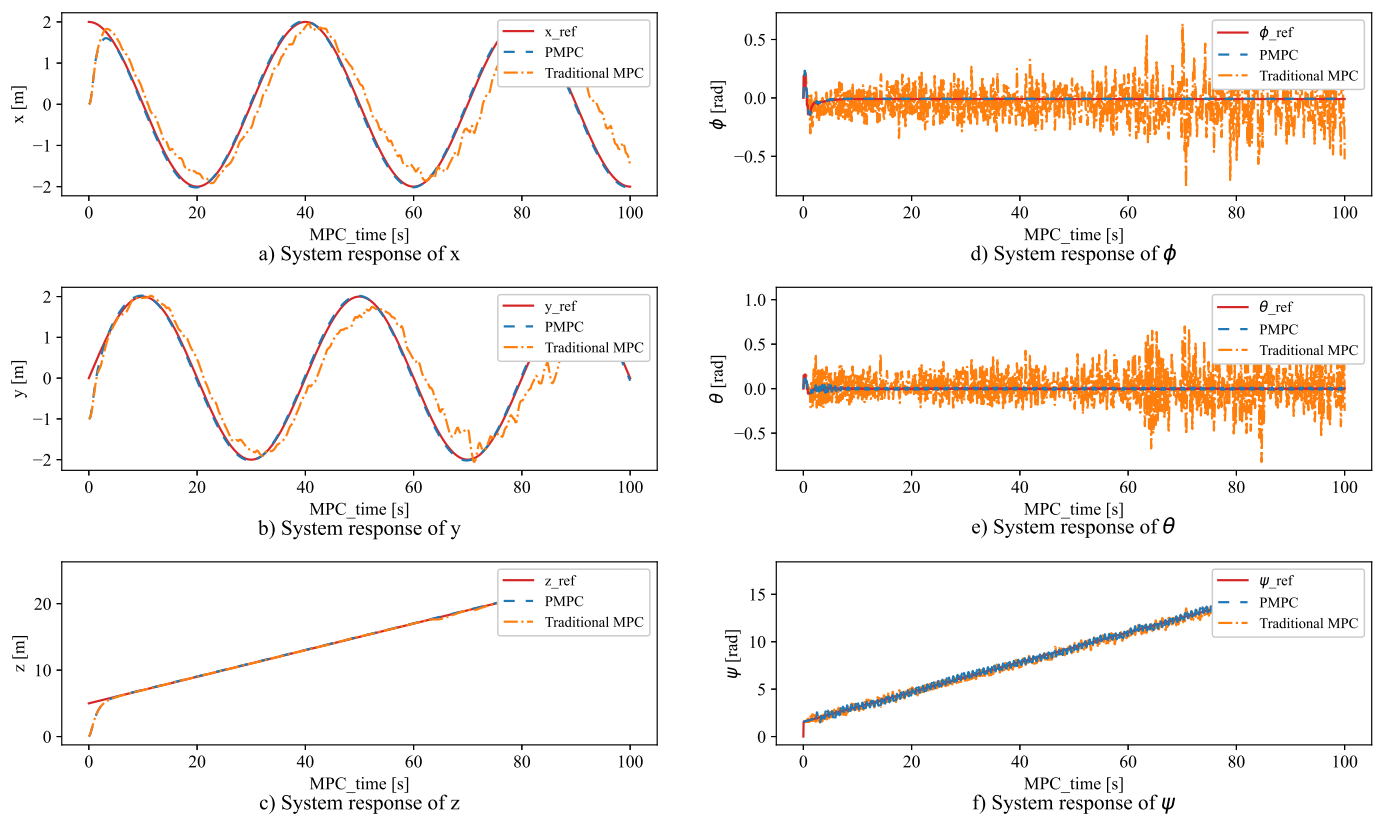


Figure 4. System response of $x, y, z, \phi, \theta,$ and ψ .

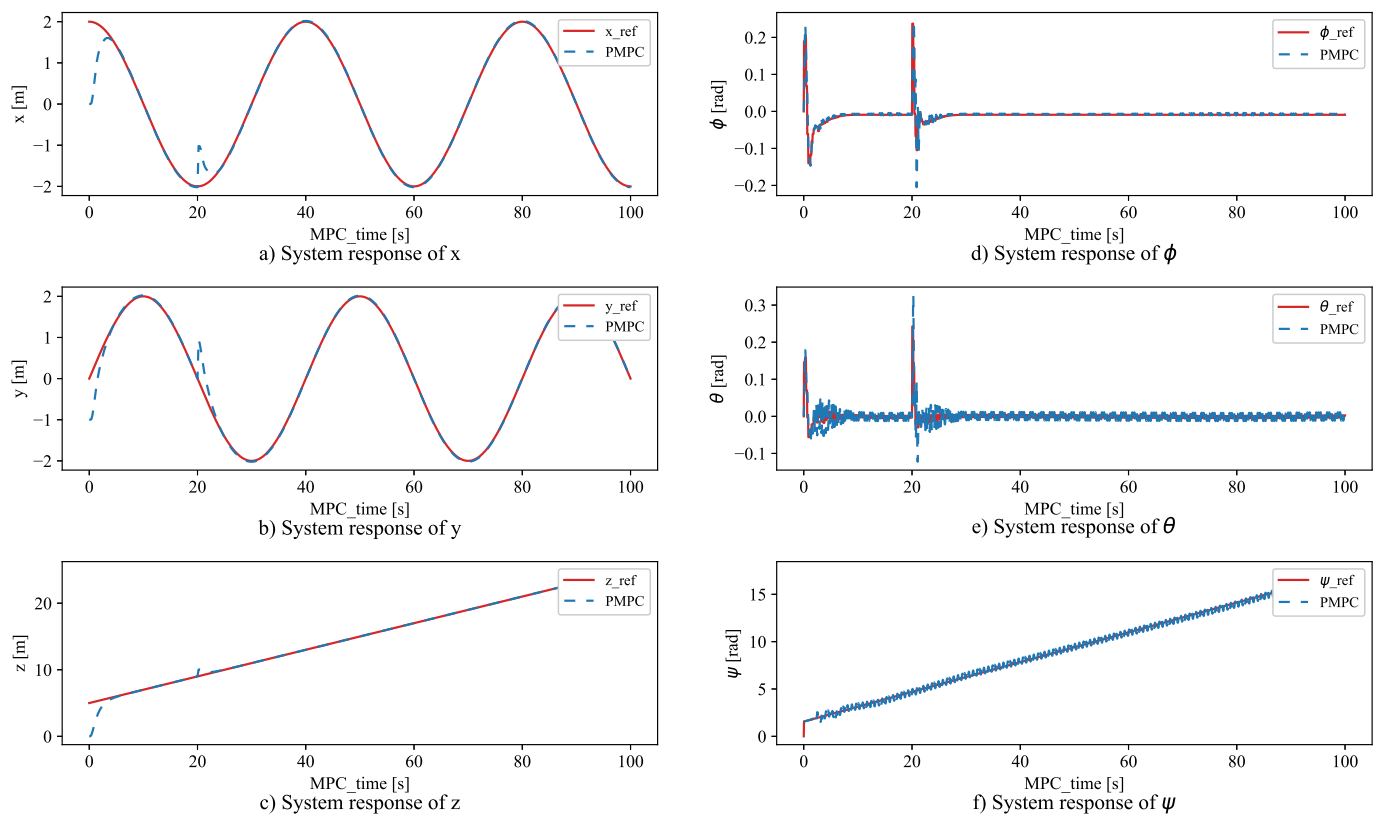


Figure 5. Tracking error of $x, y, z, \phi, \theta,$ and ψ .

5. Conclusions

In this article, a novel NN-based PMPC optimal tracking control algorithm is proposed for the UAV under a dynamic dynamics model. Due to the dynamics model of the UAV being changeable, which is difficult to deal with for traditional MPC algorithms, a neural network with experiment reply technology is applied to approximate the dynamic model as the system runs. Then, a parallel structure is constituted by the NN-based artificial system and the real system. Under the parallel structure, the MPC algorithm is applied to predict the future states of the artificial system and obtain the optimal control for the tracking control problem. Finally, a simulation is applied to demonstrate the effectiveness of the PMPC algorithm.

Author Contributions: Software, J.Q.; Investigation, J.C.; Writing—original draft, C.H.; Writing—review & editing, Z.X.; Funding acquisition, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Sichuan Province (2023NSFSC0872).

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the Natural Science Foundation of Sichuan Province (2023NSFSC0872).

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

Nomenclature

All symbols and their meanings are shown in the following Table

Symbols	Meanings of the symbols
x	The position of the UAVs on the axis x
y	The position of the UAVs on the axis y
z	The position of the UAVs on the axis z
θ	The roll angle of the UAVs
ϕ	The pitch angle of the UAVs
ψ	The yaw angle of the UAVs
g	Acceleration of gravity
I_x	The moment of inertia of the UAVs around the axis x
I_y	The moment of inertia of the UAVs around the axis y
I_z	The moment of inertia of the UAVs around the axis z
J_R	The moments of inertia of the propeller blades
L	The length from the rotors to the center of mass
Ω	The moments of angular velocity of the propeller blades
\mathbf{S}_k	The actual state of the real system
\mathbf{U}_k	The actual control input of the real system
$\hat{\mathbf{S}}_k$	The state of the artificial system
\mathbf{T}_k	The input of the artificial system
$\Psi(\cdot), \Phi(\cdot)$	The activation functions of the NN
\mathbf{Z}_k	The prediction state of the artificial system
\mathcal{N}	The length of the experiment pool

References

- Han, J.; Xu, Y.; Di, L.; Chen, Y. Low-cost multi-UAV technologies for contour mapping of nuclear radiation field. *J. Intell. Robot. Syst.* **2013**, *70*, 401–410. [[CrossRef](#)]
- Jiang, G.; Voyles, R.M.; Choi, J.J. Precision fully-actuated uav for visual and physical inspection of structures for nuclear decommissioning and search and rescue. In Proceedings of the 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Philadelphia, PA, USA, 6–8 August 2018; pp. 1–7.
- Casbeer, D.W.; Beard, R.W.; McLain, T.W.; Li, S.M.; Mehra, R.K. Forest fire monitoring with multiple small UAVs. In Proceedings of the American Control Conference, Portland, OR, USA, 8–10 June 2005; pp. 3530–3535.
- Ollero, A.; Merino, L. Unmanned aerial vehicles as tools for forest-fire fighting. *For. Ecol. Manag.* **2006**, *234*, S263. [[CrossRef](#)]

5. Jiménez-Jiménez, S.I.; Ojeda-Bustamante, W.; Marcial-Pablo, M.d.J.; Enciso, J. Digital terrain models generated with low-cost UAV photogrammetry: Methodology and accuracy. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 285. [[CrossRef](#)]
6. Tariq, A.; Osama, S.M.; Gillani, A. Development of a low cost and light weight uav for photogrammetry and precision land mapping using aerial imagery. In Proceedings of the 2016 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 19–21 December 2016; pp. 360–364.
7. Fang, Z.; Zhi, Z.; Jun, L.; Jian, W. Feedback linearization and continuous sliding mode control for a quadrotor UAV. In Proceedings of the 2008 27th Chinese Control Conference, Kunming, China, 16–18 July 2008; pp. 349–353.
8. Choi, I.; Bang, H. Quadrotor-tracking controller design using adaptive dynamic feedback-linearization method. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2014**, *228*, 2329–2342. [[CrossRef](#)]
9. Gao, H.; Liu, C.; Guo, D.; Liu, J. Fuzzy adaptive PD control for quadrotor helicopter. In Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 8–12 June 2015; pp. 281–286.
10. Yacef, F.; Bouhali, O.; Hamerlain, M. Adaptive fuzzy backstepping control for trajectory tracking of unmanned aerial quadrotor. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 920–927.
11. Xiong, J.J.; Zhang, G. Discrete-time sliding mode control for a quadrotor UAV. *Optik* **2016**, *127*, 3718–3722. [[CrossRef](#)]
12. Sun, L.; Zheng, Z. Finite-time sliding mode trajectory tracking control of uncertain mechanical systems. *Asian J. Control* **2017**, *19*, 399–404. [[CrossRef](#)]
13. Salih, A.L.; Moghavvemi, M.; Mohamed, H.A.; Gaeid, K.S. Flight PID controller design for a UAV quadrotor. *Sci. Res. Essays* **2010**, *5*, 3660–3667.
14. Khebbache, H.; Tadjine, M. Robust fuzzy backstepping sliding mode controller for a quadrotor unmanned aerial vehicle. *J. Control. Eng. Appl. Inform.* **2013**, *15*, 3–11.
15. Zheng, E.H.; Xiong, J.J.; Luo, J.L. Second order sliding mode control for a quadrotor UAV. *ISA Trans.* **2014**, *53*, 1350–1356. [[CrossRef](#)]
16. Incremona, G.P.; Ferrara, A.; Magni, L. MPC for robot manipulators with integral sliding modes generation. *IEEE/ASME Trans. Mechatronics* **2017**, *22*, 1299–1307. [[CrossRef](#)]
17. Sun, Z.; Xia, Y.; Dai, L.; Liu, K.; Ma, D. Disturbance rejection MPC for tracking of wheeled mobile robot. *IEEE/ASME Trans. Mechatronics* **2017**, *22*, 2576–2587. [[CrossRef](#)]
18. Kong, J.; Pfeiffer, M.; Schildbach, G.; Borrelli, F. Kinematic and dynamic vehicle models for autonomous driving control design. In Proceedings of the 2015 IEEE intelligent vehicles symposium (IV), Seoul, Republic of Korea, 28 June–1 July 2015; pp. 1094–1099.
19. Qian, X.; Navarro, I.; de La Fortelle, A.; Moutarde, F. Motion planning for urban autonomous driving using Bézier curves and MPC. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 826–833.
20. Borhan, H.; Vahidi, A.; Phillips, A.M.; Kuang, M.L.; Kolmanovsky, I.V.; Di Cairano, S. MPC-based energy management of a power-split hybrid electric vehicle. *IEEE Trans. Control. Syst. Technol.* **2011**, *20*, 593–603. [[CrossRef](#)]
21. Di Cairano, S.; Bernardini, D.; Bemporad, A.; Kolmanovsky, I.V. Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management. *IEEE Trans. Control. Syst. Technol.* **2013**, *22*, 1018–1031. [[CrossRef](#)]
22. Lindqvist, B.; Mansouri, S.S.; Agha-mohammadi, A.a.; Nikolakopoulos, G. Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6001–6008. [[CrossRef](#)]
23. Liu, Y.; Van Schijndel, J.; Longo, S.; Kerrigan, E.C. UAV energy extraction with incomplete atmospheric data using MPC. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 1203–1215. [[CrossRef](#)]
24. Fresk, E.; Nikolakopoulos, G. A generalized Frame Adaptive MPC for the low-level control of UAVs. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; pp. 1815–1820. [[CrossRef](#)]
25. Song, S.; Zhu, M.; Dai, X.; Gong, D. Model-Free Optimal Tracking Control of Nonlinear Input-Affine Discrete-Time Systems via an Iterative Deterministic Q-Learning Algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–14. [[CrossRef](#)]
26. Bai, W.; Zhou, Q.; Li, T.; Li, H. Adaptive reinforcement learning neural network control for uncertain nonlinear system with input saturation. *IEEE Trans. Cybern.* **2019**, *50*, 3433–3443. [[CrossRef](#)]
27. Liu, Y.J.; Zeng, Q.; Tong, S.; Chen, C.P.; Liu, L. Adaptive neural network control for active suspension systems with time-varying vertical displacement and speed constraints. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9458–9466. [[CrossRef](#)]
28. Ni, J.; Shi, P. Global predefined time and accuracy adaptive neural network control for uncertain strict-feedback systems with output constraint and dead zone. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 7903–7918. [[CrossRef](#)]
29. Jiang, B.; Li, B.; Zhou, W.; Lo, L.Y.; Chen, C.K.; Wen, C.Y. Neural Network Based Model Predictive Control for a Quadrotor UAV. *Aerospace* **2022**, *9*, 460. [[CrossRef](#)]
30. Bauersfeld, L.; Kaufmann, E.; Foehn, P.; Sun, S.; Scaramuzza, D. Neurobem: Hybrid aerodynamic quadrotor model. *arXiv* **2021**, arXiv:2106.08015.
31. Bouffard, P.; Aswani, A.; Tomlin, C. Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012; pp. 279–284.

32. Saviolo, A.; Frey, J.; Rathod, A.; Diehl, M.; Loianno, G. Active Learning of Discrete-Time Dynamics for Uncertainty-Aware Model Predictive Control. *arXiv* **2022**, arXiv:2210.12583.
33. Belkhale, S.; Li, R.; Kahn, G.; McAllister, R.; Calandra, R.; Levine, S. Model-Based Meta-Reinforcement Learning for Flight With Suspended Payloads. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1471–1478. [[CrossRef](#)]
34. O’Connell, M.; Shi, G.; Shi, X.; Azizzadenesheli, K.; Anandkumar, A.; Yue, Y.; Chung, S.J. Neural-fly enables rapid learning for agile flight in strong winds. *Sci. Robot.* **2022**, *7*, eabm6597. [[CrossRef](#)] [[PubMed](#)]
35. Zhou, Y.; Tian, Z.; Lin, H. UAV based adaptive trajectory tracking control with input saturation and unknown time-varying disturbances. *IET Intell. Transp. Syst.* **2023**, *17*, 780–793. [[CrossRef](#)]
36. Lu, J.; Han, L.; Wei, Q.; Wang, X.; Dai, X.; Wang, F.Y. Event-Triggered Deep Reinforcement Learning Using Parallel Control: A Case Study in Autonomous Driving. *IEEE Trans. Intell. Veh.* **2023**, *8*, 2821–2831. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.