

Article

# Adaptive Trajectory Tracking Algorithm for the Aerospace Vehicle Based on Improved T-MPSP

Chao Ou <sup>1,2</sup>, Chengjun Shan <sup>1</sup>, Zhongtao Cheng <sup>1</sup> and Yaosong Long <sup>1,\*</sup><sup>1</sup> School of Aerospace Engineering, Huazhong University of Science and Technology, Wuhan 430074, China<sup>2</sup> Aerospace Technology Institute of China Aerodynamics Research and Development Center, Mianyang 621000, China

\* Correspondence: longyaosong@hust.edu.cn

**Abstract:** To deal with the uncertainty and disturbance that exist in the tracking system of an aerospace vehicle, an adaptive trajectory-tracking method based on a novel tracking model predictive static programming (T-MPSP) is proposed. Firstly, to make the proposed method more adaptive to uncertain parameter deviations, an extended Kalman filter (EKF) parameter correction strategy is designed. Then, the control constraints are considered to form a novel T-MPSP algorithm. By combining the parameter correction strategy with the improved T-MPSP algorithm, a novel adaptive tracking guidance scheme is presented. Finally, simulations are carried out to demonstrate the effectiveness of the proposed method.

**Keywords:** trajectory tracking; uncertainty; extended kalman filter; adaptive; control constraints; T-MPSP

**MSC:** 90C29

**Citation:** Ou, C.; Shan, C.; Cheng, Z.; Long, Y. Adaptive Trajectory Tracking Algorithm for The Aerospace Vehicle Based on Improved T-MPSP. *Mathematics* **2023**, *11*, 2160. <https://doi.org/10.3390/math11092160>

Academic Editors: Haizhao Liang, Jianying Wang, Chuang Liu and António M. Lopes

Received: 18 January 2023

Revised: 25 April 2023

Accepted: 2 May 2023

Published: 4 May 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

For the past few decades, aerospace vehicles have been applied in both military and civilian fields, and their performance has been very compelling. For missions involving aerospace vehicles, a good trajectory-tracking ability is the essential prerequisite for the successful application of the vehicles [1]. In a traditional trajectory-tracking process for an aerospace vehicle, the dynamic model of the vehicle and the desired trajectory satisfying all the constraints are given in advance. Then, the control methods are designed to guide the vehicle to track the desired trajectory. However, the trajectories of aerospace vehicles usually cover a wide range of altitudes, which may lead to a dramatic change in the atmospheric environment during the trajectory-tracking process. In addition to the disturbance brought about by the dramatic change in the external environment, there are also uncertainties about the dynamic system of the vehicles [2,3]. Thus, the precise model information is not always available, and the traditional trajectory-tracking method cannot achieve satisfactory performance under these circumstances. As a result, the design of the novel trajectory-tracking control method to cope with disturbance and uncertainty is very important.

One of the most commonly used control architectures for trajectory-tracking control is the proportional-integral-derivative (PID) controller and its variants [4–7], which are known for the simplicity of their framework. To improve the tracking performance, many modern control theories, including intelligent optimization [4], feedback control [5], and fuzzy control [6,7], have been applied to the trajectory-tracking method design process. Another popular control architecture for trajectory-tracking control derives from the sliding mode control theory. In [8], the sliding mode control theory is applied to make the tracking error converge to zero in a finite time period. As an improvement of the method

in [8], the fixed-time control theory was combined with the sliding mode control theory. The robustness of the resulting tracking laws was further enhanced [9,10]. Recently, the neural network and adaptive updating laws were applied to the sliding mode control framework. The convergence of the tracking error was ensured by the sliding mode control approach, and any system uncertainties were handled by the neural network [11].

In addition, the nonlinear model predictive control (NMPC) was applied to the trajectory-tracking control [12–15]. However, several issues exist in the implementation of the NMPC-based tracking methods, which are summarized as follows: (1) a universal and exact model of the whole tracking system is difficult to acquire, (2) the calculation speed of the NMPC cannot meet the requirement in practice. A feasible way to cope with these issues is to use model predictive static programming (MPSP) [16], which combines the characteristics of approximate dynamic programming [17] and NMPC. The MPSP has been proven to be an effective method to cope with two-point boundary value problems with terminal constraints, which has significant advantages: (1) the terminal constraints are transformed into linear equality constraints with only the control variable to be optimized, (2) the closed analytical expression of the appropriate objective function is acquired from the static algorithm, (3) the sensitivity matrix of the algorithm can be calculated skillfully using recursion, which improves the calculation speed. The MPSP method has been used in many applications, such as in the trajectory control of launch vehicles, reentry guidance, cooperative control, etc. [18–20]. Typically, the MPSP method aims to improve the terminal accuracy by iteratively discretizing and updating. For a trajectory-tracking problem, only the terminal-tracking accuracy can be guaranteed using the MPSP method, and its computing time increases exponentially with the number of discrete nodes that are tracked. To deal with this drawback, the T-MPSP is put forward to track a trajectory over a receding horizon window. Meanwhile, the predicted time horizon can be set manually to balance the computational efficiency and the precision, which is another advantage of the T-MPSP algorithm [21,22].

Some trajectory-tracking methods are based directly on the input and output data of the system. For example, a robust model-free controller for trajectory-tracking control is proposed in Ref. [23]. No dynamic model information of the controlled system is needed, and the resulting controller is a combination of the PID controller and the sliding mode control. As an improvement of the work in Ref. [23], a forecasting-based data-driven model-free adaptive sliding mode attitude control method is proposed for the post-capture combined spacecraft with unknown inertial properties and external disturbances [24]. To cope with the unknown dynamics of the system, a model-free control method is proposed via the time-varying compensation of the un-modeled system [25]. An iterative sliding mode control technique is utilized to design the adaptive model-free trajectory-tracking method in Ref. [26]. Although these model-free control methods can deal with model uncertainties, the operation data require a huge memory size, and the computing burden is heavy.

Hence, considering the uncertainty and disturbance that exist in the tracking system of the aerospace vehicle, an adaptive trajectory-tracking method based on the novel T-MPSP is proposed. Firstly, to cope with the uncertain parameter deviations, an EKF parameter correction strategy is designed. Then, the control constraints are considered to form a novel T-MPSP algorithm. By combining the parameter correction strategy with the improved T-MPSP algorithm, an adaptive tracking guidance scheme is presented. Finally, simulations are carried out under various deviation conditions to verify the reliability and robustness of the proposed method.

The main contributions of this paper are summarized as follows:

- (1) To our best knowledge, no existing methods have applied the EKF with the T-MPSP to solve the trajectory problems of aerospace vehicles.
- (2) Compared with the MPSP method in [18–20], the proposed method has a fast computing speed and high accuracy.
- (3) Compared with [21–23], the proposed scheme can cope with the control constraints.

The rest of this paper is organized as follows: Section 2 presents the model of the aerospace vehicle. Section 3 presents the online parameter identification method and the improved T-MPSP algorithm. The simulations are presented in Section 4. Finally, the conclusion is presented in Section 5.

## 2. Model of the Aerospace Vehicle

In this section, first, the dynamic model describing the motion of the aerospace vehicle is presented. Then, the constraints that should be satisfied are introduced.

### 2.1. Dynamic Equations

The three-dimensional mass point dynamic equations [22,27] of the aircraft in the longitudinal plane are

$$\begin{cases} \frac{dh}{dt} = v \sin \gamma \\ \frac{dv}{dt} = \frac{T \cos \alpha - D}{m} - g \sin \gamma \\ \frac{d\gamma}{dt} = \frac{T \sin \alpha + L}{mv} - \frac{g}{v} \cos \gamma + \frac{v}{r} \cos \gamma \\ \frac{dm}{dt} = -\frac{T}{g_0 I_{sp}} \end{cases} \quad (1)$$

where  $h$ ,  $v$ ,  $\gamma$ ,  $\alpha$  and  $m$  represent the altitude, velocity, flight path angle, angle of attack (AOA), and mass, respectively.  $g$  and  $g_0$  represent the gravitational acceleration at the current altitude and on the earth’s surface, respectively.  $I_{sp}$  represents the specific impulse of the engine.  $T$ ,  $L$  and  $D$  denote the engine thrust, aerodynamic lift, and drag, respectively, which are defined as

$$\begin{cases} T = 0.029 \phi I_{sp} \rho g_0 v C_T A_C \\ L = \frac{1}{2} \rho v^2 S_{ref} C_L(\text{Ma}, \alpha) \\ D = \frac{1}{2} \rho v^2 S_{ref} C_D(\text{Ma}, \alpha) \end{cases} \quad (2)$$

where  $C_T$ ,  $C_L$ , and  $C_D$  denote the thrust, lift and drag coefficients, respectively.  $\phi$  denotes the throttle.  $\rho$ ,  $S_{ref}$  and  $\text{Ma}$  denote the atmospheric density, reference area, and Mach number, respectively.

### 2.2. Flight Constraints

To ensure flight safety, the dynamic pressure constraint is considered, which is defined as

$$q = \frac{1}{2} \rho v^2 \leq q_{\max} \quad (3)$$

In addition, the aircraft must satisfy the terminal constraints, which are described as

$$\begin{cases} |h_f - h_f^*| \leq \varepsilon_h \\ |v_f - v_f^*| \leq \varepsilon_v \\ |\gamma_f - \gamma_f^*| \leq \varepsilon_\gamma \end{cases} \quad (4)$$

where the subscript  $f$  refers to the final value and the superscript  $*$  refers to the desired value.  $\varepsilon_h$ ,  $\varepsilon_v$ , and  $\varepsilon_\gamma$  represent the deviation thresholds of the terminal altitude, velocity, and flight path angle, respectively.

The control variables considered are  $\alpha$  and the throttle  $\phi$ , the constraints of which are as follows:

$$\begin{cases} \alpha_{\min} \leq \alpha \leq \alpha_{\max} \\ \phi_{\min} \leq \phi \leq \phi_{\max} \end{cases} \tag{5}$$

Additionally, to prevent the control variable from changing too drastically, the rate of AOA must satisfy the following constraint:

$$\dot{\alpha}_{\min} \leq \dot{\alpha} \leq \dot{\alpha}_{\max} \tag{6}$$

### 3. The Trajectory-Tracking Strategy

In this section, we will introduce an online parameter identification method and an improved T-MPSP algorithm. By combining these two methods, an adaptive trajectory-tracking guidance algorithm based on an improved T-MPSP will be presented.

#### 3.1. Online Parameter Identification Method

Since an accurate model can improve the trajectory-tracking accuracy, a parameter correction strategy based on the EKF is designed in this subsection. In the actual tracking process, there are deviations in the parameters of atmospheric density, thrust coefficient, lift coefficient, and drag coefficient, which will make the model inaccurate. We define these parameters as

$$\rho = \rho^* (1 + \Delta\rho), C_T = C_T^* (1 + \Delta C_T), C_L = C_L^* (1 + \Delta C_L), C_D = C_D^* (1 + \Delta C_D) \tag{7}$$

where  $\rho^*$ ,  $C_T^*$ ,  $C_L^*$  and  $C_D^*$  are the desired values.  $\rho$ ,  $C_T$ ,  $C_L$  and  $C_D$  are the actual values.  $\Delta\rho$ ,  $\Delta C_T$ ,  $\Delta C_L$ , and  $\Delta C_D$  respectively represent the unknown deviation of each parameter. These unknown deviations are written in an overall form as

$$\beta = [\Delta\rho \quad \Delta C_T \quad \Delta C_L \quad \Delta C_D]^T \tag{8}$$

Defining  $x = [h \quad v \quad \gamma \quad m]$ , the following dynamic equations are obtained from (1) and (7) as

$$\dot{x} = f(x, u, \beta) \tag{9}$$

in which

$$f(x, u, \beta) = \begin{bmatrix} v \sin \gamma \\ \frac{T^* (1 + \Delta\rho)(1 + \Delta C_T) \cos \alpha - D^* (1 + \Delta\rho)(1 + \Delta C_D)}{m} - g \sin \gamma \\ \frac{T^* (1 + \Delta\rho)(1 + \Delta C_T) \sin \alpha + L^* (1 + \Delta\rho)(1 + \Delta C_L)}{mv} - \frac{g}{v} \cos \gamma + \frac{v}{r} \cos \gamma \\ - \frac{T^* (1 + \Delta\rho)(1 + \Delta C_T)}{g_0 I_{sp}} \end{bmatrix} \tag{10}$$

Since the EKF algorithm [28] needs to augment the unknown parameters into the state variables of the system, the new augmented dynamic equation is defined as

$$\dot{x}_a = f_a(x_a, u) + \omega_a = \begin{bmatrix} f(x, u, \beta) \\ 0 \end{bmatrix} + \omega_a \tag{11}$$

where

$$\begin{cases} x_a = [h \quad v \quad \gamma \quad m \quad \Delta\rho \quad \Delta C_T \quad \Delta C_L \quad \Delta C_D]^T \\ u = [\alpha \quad \phi]^T \end{cases} \tag{12}$$

$\omega_a$  represents the uncorrelated zero-mean white Gaussian noise. To identify the unknown parameters, the EKF algorithm also requires the measurement information of the unknown parameters. The measurement equation is expressed as

$$y = \begin{bmatrix} x_m \\ a_{xm} \\ a_{zm} \\ q_m \end{bmatrix} = h(x_a, u) + v_a = \begin{bmatrix} x \\ \frac{\rho v^2}{2m} C_x + \frac{T}{m} \\ \frac{\rho v^2}{2m} C_z \\ \frac{\rho v^2}{2} \end{bmatrix} + v_a \tag{13}$$

where  $x_m$ ,  $a_{xm}$ ,  $a_{zm}$  and  $q_m$  denote the state values, axial acceleration, normal acceleration and dynamic pressure measured by the sensor, respectively.  $v_a$  represents the uncorrelated zero-mean Gaussian white noise.  $C_x$  and  $C_z$  refer to axial and normal force coefficients, respectively. The calculation formulas of  $C_x$  and  $C_z$  are

$$\begin{cases} C_x = C_L \sin \alpha - C_D \cos \alpha \\ C_z = -C_L \cos \alpha - C_D \sin \alpha \end{cases} \tag{14}$$

Then, the two-step online parameter identification method will be introduced as follows:

a. Prediction

First, the prior estimate of the state at the current time instant  $k$  is

$$\hat{x}_{a_k}^- = \hat{x}_{a_{k-1}} + f(\hat{x}_{a_{k-1}}) \Delta t + \frac{F_k f(\hat{x}_{a_{k-1}})}{2} \Delta t^2 \tag{15}$$

where  $F_k$  represents the Jacobian matrix of the augmented state equation  $f_a(\hat{x}_{a_{k-1}})$  to estimate the state  $\hat{x}_{a_{k-1}}$  at the previous time instant and  $\Delta t$  represents the sampling time interval. Then, the error covariance matrix of the current time instant  $k$  according to the equation is acquired as

$$P_k^- = \Phi_k P_{k-1} \Phi_k^T + Q_k \tag{16}$$

where  $\Phi_k = I + F_k \Delta t$  represents the state transition matrix, and  $Q_k$  represents the noise covariance matrix.

b. Update

Using the error covariance matrix  $P_k^-$ , we update the Kalman filter gain coefficient  $K_k$  at the current time instant  $k$  as

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k)^{-1} \tag{17}$$

where  $V_k$  represents the measurement noise covariance matrix, and  $H_k$  represents the Jacobian matrix of the prior estimate  $\hat{x}_{a_k}^-$  of the state from the measurement equation  $y(\hat{x}_{a_k}^-)$ . Subsequently, the error covariance matrix is updated as

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \tag{18}$$

The measurement correction is updated as

$$\Delta y_k = y_k - y(\hat{x}_{a_k}^-) \tag{19}$$

where  $y_k$  represents the actual measurement value.

Finally, we update the posterior estimate of the state at the current time instant  $k$  using

$$\hat{x}_{a_k} = \hat{x}_{a_k}^- + K_k \Delta y_k \tag{20}$$

$\hat{x}_{a_k}$  of the augmented state has been obtained through the EKF algorithm, from which the estimated value of the corresponding unknown parameter  $\beta$  is also obtained. Thus, the new model that accounts for uncertain derivations is obtained by substituting the estimated value of  $\beta$  into (10).

**Remark 1.** An imprecise model will make it difficult for the trajectory-tracking method design, and uncertain parameter deviations will affect the accuracy of the model. Hence, a parameter correction strategy based on the EKF is designed in this subsection. By applying the EKF algorithm, the augmented state  $\hat{x}_{a_k}$  is obtained, from which the estimated value of the corresponding unknown parameter  $\beta$  is also obtained. Then, the exact model that considers uncertain derivations is obtained.

### 3.2. Improved T-MPSP Algorithm

According to (1), the state variables are  $X = [h, v, \gamma, m]^T$ , the control variables are  $U = [\alpha, \phi]^T$ , and the output variables are  $Y = [h, v, \gamma]^T$ . The discrete form of the state equation and output equation of a continuous nonlinear system is expressed as

$$X_{k+1}^i = F_k(X_k^i, U_k^i) \tag{21}$$

$$Y_k^i = h(X_k^i) \tag{22}$$

where  $X \in \mathbb{R}^n$ ,  $U \in \mathbb{R}^m$  and  $Y \in \mathbb{R}^p$  denote the state, control, and output variables, respectively.  $k = 1, 2, \dots, N_y$  represents the  $k$  th sampling point, and  $N_y$  represents the total length of the prediction time domain.  $k = 1$  and  $k = N_y$  represent the starting and ending points of the prediction time domain, respectively, and  $i$  represents the number of iterations.

The main objective of the trajectory-tracking algorithm is to find the suitable control variables  $U_k^{i+1}$  to make the output  $Y_k^{i+1}$  as close to the desired output  $Y_k^*$  ( $k = 2, 3, \dots, N_y$ ) as possible at each sampling time; that is,  $Y_k^{i+1} \rightarrow Y_k^*$ , where  $U_k^i$  represents the control variables at the current  $i$  th iteration, and  $U_k^{i+1}$  represents the control variables for the next iteration.

Similarly,  $Y_k^i$  is the current output, and  $Y_k^{i+1}$  is the output of the next iteration. We should also add a performance index that minimizes the deviation of the control variables to avoid overly drastic changes in the control variables. Therefore, the following objective function is proposed:

$$J = \frac{1}{2} \sum_{k=2}^{N_y} (Y_k^{i+1} - Y_k^*)^T Q_k^i (Y_k^{i+1} - Y_k^*) + \frac{1}{2} \sum_{k=1}^{N_y-1} (U_k^{i+1} - U_k^i)^T R_k^i (U_k^{i+1} - U_k^i) \tag{23}$$

where  $Q_k^i$  and  $R_k^i$  are both the positive definite weight matrix at the  $i$  th iteration.

The deviation vectors between two consecutive iterations at the same time are defined as follows:

$$Y_k^{i+1} = Y_k^i + \Delta Y_k^i \tag{24}$$

$$X_k^{i+1} = X_k^i + \Delta X_k^i \tag{25}$$

$$U_k^{i+1} = U_k^i + \Delta U_k^i \tag{26}$$

By expanding the Taylor series of  $\Delta Y_k^i$  and ignoring its higher-order terms, we obtain

$$\Delta Y_k^i \approx dY_k^i = \left[ \frac{\partial Y_k}{\partial X_k} \right] dX_k^i \tag{27}$$

Similarly, we obtain

$$\Delta X_{k+1}^i \approx dX_{k+1}^i = \left[ \frac{\partial F_k}{\partial X_k} \right] dX_k^i + \left[ \frac{\partial F_k}{\partial U_k} \right] dU_k^i \tag{28}$$

where  $dX_k^i$  and  $dU_k^i$  represent deviations in the state variables and control variables at the  $k$  th time instant, respectively. By substituting (27) into (26), we obtain

$$dY_k^i = \left[ \frac{\partial Y_k}{\partial X_k} \right] \left[ \frac{\partial F_{k-1}}{\partial X_{k-1}} \right] dX_{k-1}^i + \left[ \frac{\partial Y_k}{\partial X_k} \right] \left[ \frac{\partial F_{k-1}}{\partial U_{k-1}} \right] dU_{k-1}^i \tag{29}$$

where  $dX_{k-1}^i$  is also expanded into an equation composed of  $dX_{k-2}^i$  and  $dU_{k-2}^i$ , and the corresponding equation is substituted into (29). By successively substituting  $dX_k, dX_{k-1}, \dots, dX_2$  into the expression of  $dY_k^i$ , we obtain

$$dY_k^i = [A^k]^i dX_1^i + [B_1^k]^i dU_1^i + [B_2^k]^i dU_2^i + \dots + [B_{k-1}^k]^i dU_{k-1}^i \tag{30}$$

It is obvious that  $dX_1^i = 0$ . Then, (30) is simplified to

$$dY_k^i = \sum_{j=1}^{k-1} [B_j^k]^i dU_j^i \tag{31}$$

$$[B_j^k]^i = \left[ \frac{\partial Y_k}{\partial X_k} \right] \left[ \frac{\partial F_{k-1}}{\partial X_{k-1}} \right] \dots \left[ \frac{\partial F_{j+1}}{\partial X_{j+1}} \right] \left[ \frac{\partial F_j}{\partial U_j} \right], \quad k = 2, 3, \dots, N_y \tag{32}$$

where  $[B_j^k]^i$  is called the sensitivity matrix. Then, the deviations in the output variables and control variables at each sampling time are formed into a linear equation, in which each  $dU_j^i$  is a variable to be optimized. (32) is calculated recursively in the following way:

$$\begin{aligned} [A_k^k]^i &= I_{n \times n} \\ [A_j^k]^i &= [A_{j+1}^k]^i \left[ \frac{\partial F_j}{\partial X_j} \right] \\ [B_j^k]^i &= \left[ \frac{\partial Y_k}{\partial X_k} \right] [A_{j+1}^k]^i \left[ \frac{\partial F_j}{\partial U_j} \right] \end{aligned} \tag{33}$$

where  $k = 2, 3, \dots, N_y, j = (k-1), (k-2), \dots, 1$ , when  $j \geq k, [B_j^k]^i = 0_{p \times m}$ .

According to (24) and (26), and considering the small approximation  $\Delta Y_k^i \approx dY_k^i, \Delta U_k^i \approx dU_k^i$ , we obtain

$$\begin{cases} (Y_k^{i+1} - Y_k^*) = \Delta Y_k^i + (Y_k^i - Y_k^*) = \Delta Y_k^i - \Delta Y_k^{*i} = dY_k^i - \Delta Y_k^{*i} \\ (U_k^{i+1} - U_k^i) = \Delta U_k^i = dU_k^i \end{cases} \tag{34}$$

Hence, the objective function in (23) is rewritten as

$$J = \frac{1}{2} \sum_{k=2}^{N_y} (dY_k^i - \Delta Y_k^{*i})^T Q_k^i (dY_k^i - \Delta Y_k^{*i}) + \frac{1}{2} \sum_{k=1}^{N_y-1} (dU_k^i)^T R_k^i (dU_k^i) \tag{35}$$

In the traditional T-MPSP algorithm, the increment of the control variables is added to the performance index to indirectly constrain the control variables. However, this method cannot strictly constrain the control variables, which may fail to satisfy the constraints. Therefore, we propose an improved T-MPSP algorithm by adding control variable constraints.

The control variable constraints (5) and (6) are expressed as

$$\begin{aligned}
 U_{\min} &\leq U_k^{i+1} \leq U_{\max} \\
 \dot{U}_{\min} &\leq \dot{U}_k^{i+1} = \frac{U_{k+1}^{i+1} - U_k^{i+1}}{h} \leq \dot{U}_{\max}
 \end{aligned}
 \tag{36}$$

where  $h$  represents the sampling step size,  $U \in \mathbb{R}^m$ . According to (34), we obtain

$$U_{\min} \leq U_k^i + dU_k^i \leq U_{\max} \Rightarrow \begin{cases} dU_k^i \leq U_{\max} - U_k^i \\ -dU_k^i \leq U_k^i - U_{\min} \end{cases}
 \tag{37}$$

which is simplified to

$$\begin{cases} dU_k^i \leq U_{\max} - U_k^i = C1_k^i \\ -dU_k^i \leq U_k^i - U_{\min} = C2_k^i \end{cases}
 \tag{38}$$

The control magnitude constraint is thus transformed into a linear inequality constraint with the only unknown  $dU_k^i$ . Then,  $\dot{U}_k^{i+1}$  is converted into

$$\dot{U}_k^{i+1} = \frac{U_{k+1}^{i+1} - U_k^{i+1}}{h} = \frac{(U_{k+1}^i + dU_{k+1}^i) - (U_k^i + dU_k^i)}{h}
 \tag{39}$$

Thereby,

$$\dot{U}_{\min} \leq \frac{(U_{k+1}^i + dU_{k+1}^i) - (U_k^i + dU_k^i)}{h} \leq \dot{U}_{\max} \Rightarrow \begin{cases} -dU_k^i + dU_{k+1}^i \leq h\dot{U}_{\max} - U_{k+1}^i + U_k^i \\ dU_k^i - dU_{k+1}^i \leq -h\dot{U}_{\min} + U_{k+1}^i - U_k^i \end{cases}
 \tag{40}$$

which is simplified to

$$\begin{cases} \begin{bmatrix} -I_m & I_m \end{bmatrix} \begin{bmatrix} dU_k^i \\ dU_{k+1}^i \end{bmatrix} \leq h\dot{U}_{\max} - U_{k+1}^i + U_k^i = C3_k^i \\ \begin{bmatrix} I_m & -I_m \end{bmatrix} \begin{bmatrix} dU_k^i \\ dU_{k+1}^i \end{bmatrix} \leq -h\dot{U}_{\min} + U_{k+1}^i - U_k^i = C4_k^i \end{cases}
 \tag{41}$$

where  $I_m$  is an identity matrix of order  $m \times m$ . In this way, the change rate constraint is transformed into linear inequality constraints with unknowns  $dU_k^i$  and  $dU_{k+1}^i$ . In order to facilitate the subsequent solution, (41) is rewritten in the following form:

$$\begin{cases} PdU^i = C3^i \\ -PdU^i = C4^i \end{cases}
 \tag{42}$$

where  $P = \begin{bmatrix} -I_m & I_m & 0 & \cdots & 0 \\ 0 & -I_m & I_m & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -I_m & I_m \end{bmatrix}$ ,  $dU^i = \begin{bmatrix} dU_1^i \\ dU_2^i \\ \vdots \\ dU_{N-2}^i \\ dU_{N-1}^i \end{bmatrix}$ ,  $C3^i = \begin{bmatrix} C3_1^i \\ C3_2^i \\ \vdots \\ C3_{N-2}^i \\ C3_{N-1}^i \end{bmatrix}$ ,  $C4^i = \begin{bmatrix} C4_1^i \\ C4_2^i \\ \vdots \\ C4_{N-2}^i \\ C4_{N-1}^i \end{bmatrix}$ .

Therefore, combining (35) with the inequality constraints in (38) and (42) yields

$$\begin{aligned}
 \min J &= \frac{1}{2} \sum_{k=2}^{N_y} (dY_k^i - \Delta Y_k^{*i})^T Q_k^i (dY_k^i - \Delta Y_k^{*i}) + \frac{1}{2} \sum_{k=1}^{N_y-1} (dU_k^i)^T R_k^i (dU_k^i) \\
 \text{s.t.} & \begin{cases} dU_k^i \leq C1_k^i \\ -dU_k^i \leq C2_k^i \end{cases}, \quad \begin{cases} PdU^i = C3^i \\ -PdU^i = C4^i \end{cases}
 \end{aligned}
 \tag{43}$$

In this paper, the Lagrangian multiplier method and the penalty function are applied to solve the NLP problem; thus, we have

$$\begin{aligned}
 L = & \frac{1}{2} \sum_{k=2}^{N_y} \left( \sum_{j=1}^{k-1} B_j^k dU_j - \Delta Y_k^* \right)^T Q_k \left( \sum_{j=1}^{k-1} B_j^k dU_j - \Delta Y_k^* \right) \\
 & + \frac{1}{2} \sum_{k=1}^{N_y-1} (dU_k)^T R_k (dU_k) \\
 & + \frac{1}{2} \sum_{k=1}^{N_y-1} (dU_k - C1_k)^T \sigma 1_k (dU_k - C1_k) \\
 & + \frac{1}{2} \sum_{k=1}^{N_y-1} (-dU_k - C2_k)^T \sigma 2_k (-dU_k - C2_k) \\
 & + \frac{1}{2} \sum_{k=1}^{N_y-2} \left( \sum_{j=1}^{N_y-1} P_{kj} dU_j - C3_k \right)^T \sigma 3_k \left( \sum_{j=1}^{N_y-1} P_{kj} dU_j - C3_k \right) \\
 & + \frac{1}{2} \sum_{k=1}^{N_y-2} \left( \sum_{j=1}^{N_y-1} -P_{kj} dU_j - C4_k \right)^T \sigma 4_k \left( \sum_{j=1}^{N_y-1} -P_{kj} dU_j - C4_k \right)
 \end{aligned} \tag{44}$$

where  $P_{kj}$  represents the element of the  $k$  th row and the  $j$  th column of the matrix  $P$ . The data in the following equations are all in the same iteration  $i$ , so the superscript  $i$  is omitted for the convenience of subsequent derivation. According to the necessary conditions for first-order optimality, we obtain

$$\begin{aligned}
 \frac{\partial L}{\partial dU_l} = & 0 \\
 \Rightarrow & R_l dU_l + \sum_{k=2}^{N_y} \left( (B_l^k)^T Q_k \sum_{j=1}^{k-1} B_j^k dU_j \right) - \sum_{k=2}^{N_y} (B_l^k)^T Q_k \Delta Y_k^* + \sigma 1_l (dU_l - C1_l) + \\
 & \sigma 2_l (dU_l + C2_l) + \sum_{k=2}^{N_y-2} \left( (P_{kl})^T \sigma 3_k \sum_{j=1}^{N_y-1} P_{kj} dU_j \right) - \sum_{k=2}^{N_y-2} (P_{kl})^T \sigma 3_k C3_k + \\
 & \sum_{k=2}^{N_y-2} \left( (P_{kl})^T \sigma 4_k \sum_{j=1}^{N_y-1} P_{kj} dU_j \right) + \sum_{k=2}^{N_y-2} (P_{kl})^T \sigma 4_k C4_k = 0
 \end{aligned} \tag{45}$$

where  $\sigma 1_k, \sigma 2_k, \sigma 3_k$  and  $\sigma 4_k$  are all penalty factors. By changing the positions of the terms in Equation (45), we obtain

$$\begin{aligned}
 & (R_l + \sigma 1_l + \sigma 2_l) dU_l + \sum_{k=2}^{N_y} \left( (B_l^k)^T Q_k \sum_{j=1}^{k-1} B_j^k dU_j \right) + \\
 & \sum_{k=2}^{N_y-2} \left( (P_{kl})^T \left( \sigma 3_k \sum_{j=1}^{N_y-1} P_{kj} dU_j + \sigma 4_k \sum_{j=1}^{N_y-1} P_{kj} dU_j \right) \right) \\
 & = \sum_{k=2}^{N_y} (B_l^k)^T Q_k \Delta Y_k^* + \sum_{k=2}^{N_y-2} P_{kl}^T (\sigma 3_k C3_k - \sigma 4_k C4_k) + \sigma 1_l C1_l - \sigma 2_l C2_l
 \end{aligned} \tag{46}$$

All  $dU_l (l = 1, 2, \dots, N_y - 1)$  in (46) are expressed in a matrix form as

$$\begin{bmatrix} M_{11} + T_1 & \cdots & M_{1(N_y-2)} & M_{1(N_y-1)} \\ M_{21} & M_{22} + T_2 & \cdots & M_{2(N_y-1)} \\ \vdots & \cdots & \ddots & \vdots \\ M_{(N_y-1)1} & \cdots & \cdots & M_{(N_y-1)(N_y-1)} + T_{N_y-1} \end{bmatrix} \begin{bmatrix} dU_1 \\ dU_2 \\ \vdots \\ dU_{N_y-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N_y-1} \end{bmatrix} \tag{47}$$

where

$$\begin{cases} M_{ij} = \sum_{l=(j+1)}^{N_y} B_i^{lT} Q_l B_j^l + \sum_{k=2}^{N_y-2} (P_{ki}^T \sigma 3_k P_{kj} + P_{ki}^T \sigma 4_k P_{kj}) \\ b_i = \sigma 1_i C 1_i - \sigma 2_i C 2_i + \sum_{l=2}^{N_y} B_i^{lT} Q_l \Delta Y_l^* + \sum_{k=2}^{N_y-2} P_{ki}^T (\sigma 3_k C 3_k - \sigma 4_k C 4_k) \\ T_i = R_i + \sigma 1_i + \sigma 2_i \end{cases} \quad (48)$$

By solving (47), we obtain the corrections for all control variables  $dU = [dU_1, dU_2, \dots, dU_{N_y-1}]$ . Finally, the updated control variables are calculated as

$$U^{i+1} = U^i + dU \quad (49)$$

### 3.3. Overall Structure and Operating Steps

In this subsection, we combine the online parameter identification method with the improved T-MPSP algorithm to propose an adaptive trajectory tracking guidance algorithm based on an improved T-MPSP. By applying the first-order Euler method, we obtain

$$X_{k+1} = X_k + hf(X, U) = F_k(X_k, U_k) \quad (50)$$

where  $h$  represents the sampling step size.  $X_k = [h_k, v_k, \gamma_k, m_k]^T$ ,  $U_k = [\alpha_k, \phi_k]^T$  and  $Y_k = [h_k, v_k, \gamma_k]^T$  denote the state, control, and output variables of the aircraft, respectively.  $f(X, U)$  represents the state differential equation of (1). The comprehensive block diagram of the proposed method is presented in Figure 1, and the operating steps of the adaptive tracking guidance algorithm based on the improved T-MPSP are as follows:

Step 1: Parameter initialization, such as initializing the prediction time domain, sampling step size, EKF parameters, etc.

Step 2: Use the EKF algorithm to identify the parameter deviation online, which is used to correct the prediction model for the T-MPSP algorithm.

Step 3: In the improved T-MPSP algorithm, use the revised model to update the control variables until they converge or the algorithm reaches the maximum number of iterations.

Step 4: Use the first control values of the prediction time-domain window of the T-MPSP algorithm as the control variables of the current time.

Step 5: Integrate the dynamic equations using fourth-order Runge-Kutta to the next time instant.

Step 6: Determine whether the terminal time is reached. If it is reached, stop the operation. Otherwise, go to step 2 and continue with the same steps.

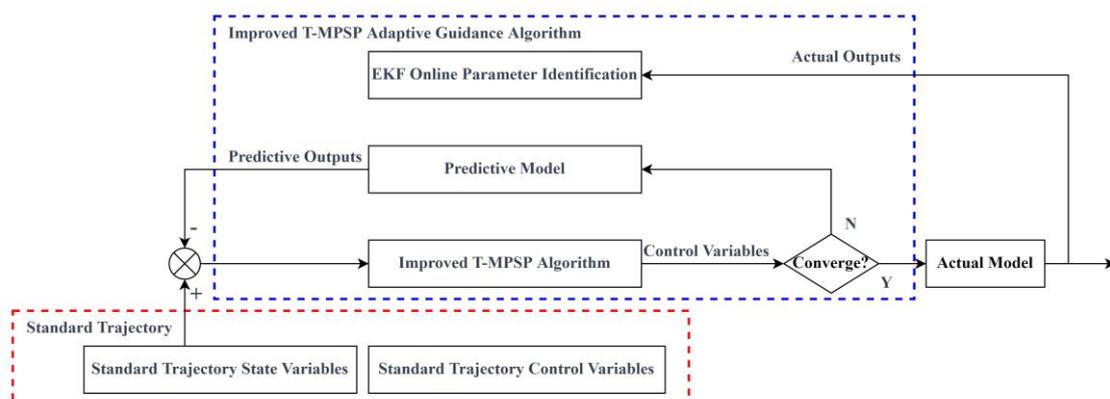


Figure 1. The block diagram of the proposed method.

### 4. Simulations

In this section, simulations are carried out to show the effectiveness of the proposed method. First, comparison simulations against the existing T-MPSP tracking method are presented. Then, several extreme combinations of parameter deviations are considered in the simulation. Finally, the Monte Carlo simulations are carried out to verify the robustness of the proposed method.

The control parameters are initialized as

$$\begin{cases} N_y = 7 \\ Q_k = \begin{bmatrix} 1.11 & 0 & 0 \\ 0 & 1.96 & 0 \\ 0 & 0 & 5000 \end{bmatrix} \\ R_k = \begin{bmatrix} 1000 \\ 90000 \end{bmatrix} \end{cases} \tag{51}$$

And the constraints considered in this simulation are tabulated in Table 1.

**Table 1.** Constraints.

Process constraints	$q_{\max}$	150 (kPa)
Control constraints	$[\alpha_{\min}, \alpha_{\max}]$	$[-3, 21]$ ( $^{\circ}$ )
	$[\phi_{\min}, \phi_{\max}]$	$[0, 2]$
	$[\dot{\alpha}_{\min}, \dot{\alpha}_{\max}]$	$[-5, 5]$ ( $^{\circ}/s$ )
Terminal constraint	$\epsilon_h$	500 (m)
	$\epsilon_v$	50 (m/s)
	$\epsilon_{\gamma}$	0.5 ( $^{\circ}$ )

#### 4.1. Comparison Simulations

To demonstrate the effectiveness of the proposed method, both the open-loop tracking method and the MPSP tracking law presented in [22] are considered as comparison methods in this subsection. The detailed simulation results are presented in Table 2 and Figure 1.

In Figure 2, the solid lines in blue represent the tracking results of the proposed method, dash lines in blue refer to the tracking results of the open-loop control, dot lines in pink denote the tracking results of the comparison T-MPSP method, and the reference curves are presented in dash-dot form.

**Table 2.** Comparison results.

	Terminal Height Deviations (m)	Terminal Velocity Deviations (m/s)	Terminal Flight Path Angle Deviations ( $^{\circ}$ )
Proposed method	32.95	-12.53	0.279
Open-loop tracking method	-444.49	272.40	1.124
T-MPSP method	-242.36	-15.48	0.025

For the open-loop control scenario, the tracking performance is far from satisfactory. Moreover, the maximum dynamic pressure violates the constraint, which will cause

damage to the vehicle. As for the comparison T-MPSP method, the terminal accuracy is acceptable, while the overall tracking performance is not so good. In the proposed method, the terminal deviations are much smaller than in the comparison law, which confirms the effectiveness and superiority of the proposed method. The optimization process of the proposed method and the T-MPSP method requires about 1.7 s on our laptop with an AMD 1.8 GHz CPU. However, it needs about 2.8 s for the MPSP tracking method.

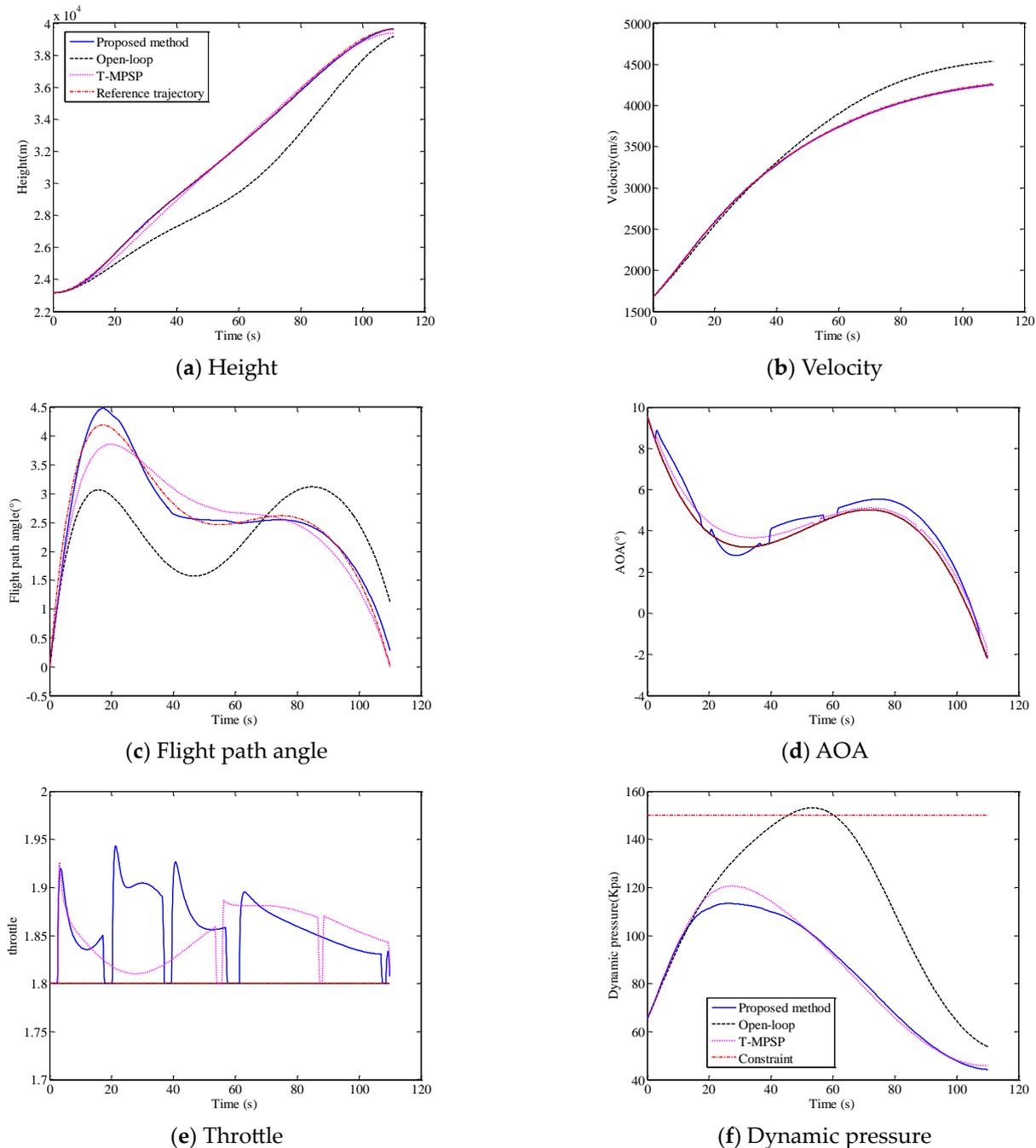


Figure 2. Comparison results.

#### 4.2. The Monte Carlo Simulations

In this subsection, trajectory tracking is carried out under 300 sets of deviations sampled through the LHS method. Furthermore, the distribution intervals of the four uncertain conditions are as follows:

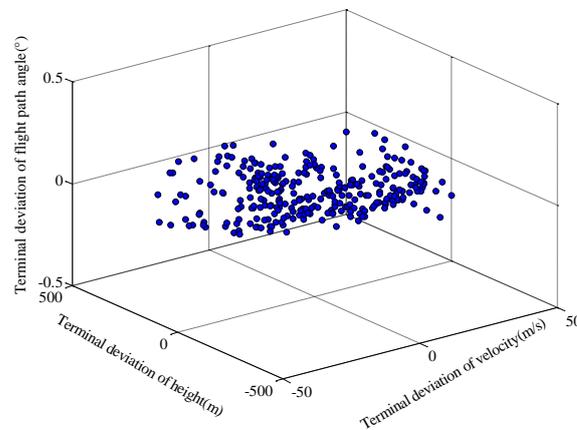
$$\begin{cases} \Delta\rho \in [-0.1, 0.1] \\ \Delta C_r \in [-0.05, 0.05] \\ \Delta C_L \in [-0.1, 0.1] \\ \Delta C_D \in [-0.1, 0.1] \end{cases} \quad (52)$$

The terminal state deviations under the 300 groups of random deviations are presented in Figures 3–5. The statistical maximum deviations in the terminal states are recorded in Table 3.

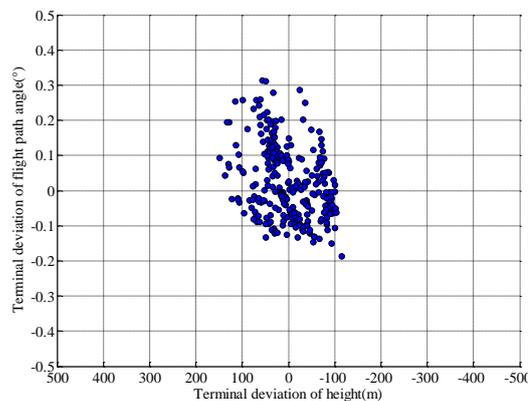
**Table 3.** Maximum tracking deviations.

Terminal Height Deviations (m)	Terminal Velocity Deviations (m/s)	Terminal Flight Path Angle Deviations (°)	Dynamic Pressure (kPa)
148.73	47.26	0.313	142.12

The maximum tracking deviation in the terminal height is less than 150 m, which is far lower than the requirements of terminal height accuracy. At the same time, the maximum tracking deviation in the terminal speed is less than 50 m/s, which also meets the requirements. In addition, the absolute value of terminal flight path angle deviation is also within the allowable range of 0.5°, and the maximum dynamic pressure also meets the constraints. These simulation results show that the novel T-MPSP method proposed in this paper has a good anti-interference ability against coefficient deviations caused by complex flight environments.



**Figure 3.** Terminal state deviation distributions.



**Figure 4.** Terminal state deviation distributions (left view).

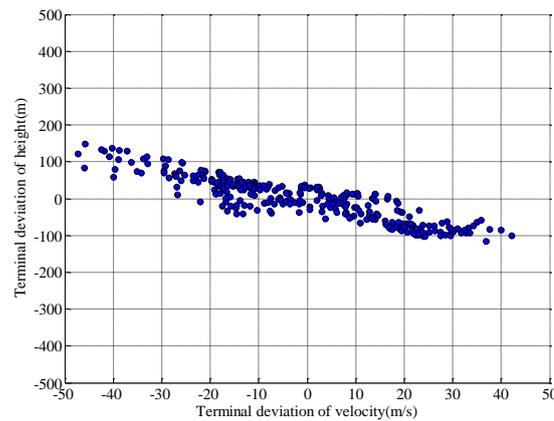


Figure 5. Terminal state deviation distributions (vertical view).

### 5. Conclusions

An adaptive trajectory-tracking method based on a novel tracking model predictive static programming (T-MPSP) was proposed. The proposed method had advantages in computing efficiency and tracking accuracy. Further, it was more adaptive to uncertain parameter deviations due to the parameter correction strategy. Firstly, a parameter correction strategy was designed. Then, the control constraints were considered to form a novel T-MPSP algorithm. By combining the parameter correction strategy with the improved T-MPSP algorithm, an adaptive tracking guidance scheme was presented. Finally, simulations were carried out to show the effectiveness of the proposed method. In related future research, the saturation of the input should be considered in the design of the trajectory-tracking method.

**Author Contributions:** Conceptualization, C.O. and Z.C.; Methodology, C.O., Z.C. and Y.L.; Investigation, Z.C. and Y.L.; Data curation, C. S.; Writing—original draft, Z.C.; Supervision, Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was co-supported in part by the National Natural Science Foundation of China (No. 61903146).

**Data Availability Statement:** Not applicable.

**Acknowledgments:** This study has been co-supported in part by the National Natural Science Foundation of China (No. 61903146 and 61873319).

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

### Nomenclature

*Scalar*

$h$	altitude	$v$	velocity
$\gamma$	flight path angle	$\alpha$	angle of attack (AOA)
$m$	mass	$g$	gravitational acceleration
$I_{sp}$	the specific impulse of the engine	$T$	the engine thrust
$L$	the aerodynamic lift	$D$	the aerodynamic drag
$C_T$	the thrust coefficient	$C_L$	the lift coefficient
$C_D$	the drag coefficient	$\rho$	the atmospheric density
$\phi$	the throttle	$S_{ref}$	the reference area
$Ma$	mach number	$\varepsilon$	the terminal deviation thresholds
$\omega_a$	uncorrelated white Gaussian noise	$C_z$	normal force coefficients
$C_x$	axial force coefficients	$k$	the current time instant

*Matrix*

$F$	the Jacobian matrix	$P$	the error covariance matrix
$\Phi$	the state transition matrix	$Q_i$	the noise covariance matrix
$K$	the Kalman filter gain coefficient matrix	$R_i$	the noise covariance matrix
<b>Subscripts</b>			
max	the maximum value	min	the minimum value
$f$	the final value and the superscript	*	the desired value
$a$	the new augmented state		

## References

- Chai, R.; Tsourdos, A.; Savvaris, A.; Chai, S.; Chen, C. Review of advanced guidance and control algorithms for space/aerospace vehicles. *Prog. Aerosp. Sci.* **2021**, *122*, 100696.
- Gharib, M.R.; Koochi, A.; Ghorbani, M. Path tracking control of electromechanical micro-positioner by considering control effort of the system. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2021**, *235*, 984–991.
- Salehi Kolahi, M.R.; Gharib, M.R.; Heydari, A. Design of a non-singular fast terminal sliding mode control for second-order nonlinear systems with compound disturbance. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2021**, *235*, 7343–7352.
- Abdalla, T.Y.; Abdulkarem, A. PSO-based optimum design of PID controller for mobile robot trajectory tracking. *Int. J. Comput. Appl.* **2012**, *47*, 30–35.
- Madhushani, T.; Maithripala, D.S.; Berg, J.M. Feedback regularization and geometric PID control for trajectory tracking of mechanical systems: Hoop robots on an inclined plane. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 3938–3943.
- Rabah, M.; Rohan, A.; Han, Y.-J.; Kim, S.-H. Design of fuzzy-PID controller for quadcopter trajectory-tracking. *Int. J. Fuzzy Log. Intell. Syst.* **2018**, *18*, 204–213.
- Dang, T.S.; Duong, D.T.; Le, V.C.; Banerjee, S. A combined backstepping and adaptive fuzzy PID approach for trajectory tracking of autonomous mobile robots. *J. Braz. Soc. Mech. Sci. Eng.* **2021**, *43*, 156.
- Shen, G.; Xia, Y.; Ma, D.; Zhang, J. Adaptive sliding-mode control for Mars entry trajectory tracking with finite-time convergence. *Int. J. Robust Nonlinear Control* **2019**, *29*, 1249–1264.
- Ai, X.; Yu, J. Fixed-time trajectory tracking for a quadrotor with external disturbances: a flatness-based sliding mode control approach. *Aerosp. Sci. Technol.* **2019**, *89*, 58–76.
- Sun, L.; Liu, Y. Fixed-time adaptive sliding mode trajectory tracking control of uncertain mechanical systems. *Asian J. Control* **2020**, *22*, 2080–2089.
- Zhou, B.; Huang, B.; Su, Y.; Zheng, Y.; Zheng, S. Fixed-time neural network trajectory tracking control for underactuated surface vessels. *Ocean Eng.* **2021**, *236*, 109416.
- Shen, C.; Shi, Y.; Buckham, B. Nonlinear model predictive control for trajectory tracking of an AUV: A distributed implementation. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 5998–6003.
- Nascimento, T.P.; Dórea, C.E.T.; Gonçalves, L.M.G. Nonlinear model predictive control for trajectory tracking of nonholonomic mobile robots: A modified approach. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1729881418760461.
- Chai, J.; Medagoda, E.; Kayacan, E. Adaptive and Efficient Model Predictive Control for Booster Reentry. *J. Guid. Control Dyn.* **2020**, *43*, 2372–2382.
- Emami, S.A.; Banazadeh, A. Fault-tolerant predictive trajectory tracking of an air vehicle based on acceleration control. *IET Control Theory Appl.* **2019**, *14*, 750–762.
- Padhi, R.; Kothari, M. Model predictive static programming: a computationally efficient technique for suboptimal control design. *Int. J. Innov. Comput. Inf. Control* **2009**, *5*, 399–411.
- Powell, W.B. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*; John Wiley & Sons: Hoboken, NJ, USA, 2007; Volume 703.
- Halbe, O.; Raja, R.G.; Padhi, R. Robust reentry guidance of a reusable launch vehicle using model predictive static programming. *J. Guid. Control Dyn.* **2014**, *37*, 134–148.
- Zheng, H.; Hong, H.; Tang, S. Model predictive static programming rendezvous trajectory generation of unmanned aerial vehicles. In Proceedings of the 2019 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 17–19 October 2019; pp. 415–420.
- Wang, Y.; Hong, H.; Tang, S. Geometric control with model predictive static programming on SO (3). *Acta Astronaut.* **2019**, *159*, 471–479.
- Tripathi, A.K.; Padhi, R. Autonomous landing for uavs using t-mpsp guidance and dynamic inversion autopilot. *IFAC-PapersOnLine* **2016**, *49*, 18–23.
- Wang, M.; Zhang, S. Adaptive trajectory tracking algorithm based on tracking model-predictive-static-programming. *Acta Aeronaut. Astronaut. Sin.* **2018**, *39*, 322105–322113.
- Kara, T.; Mary, A. Robust trajectory tracking control of robotic manipulators based on model-free PID-SMC approach. *J. Eng. Res.* **2018**, *6*, 170–188.

24. Emami, S.A.; Banazadeh, A. Intelligent trajectory tracking of an aircraft in the presence of internal and external disturbances. *Int. J. Robust Nonlinear Control*. **2019**, *29*, 5820–5844. <https://doi.org/10.1002/rnc.4698>.
25. Dou, L.; Su, X.; Zhao, X.; Zong, Q.; He, L. Robust tracking control of quadrotor via on-policy adaptive dynamic programming. *Int. J. Robust Nonlinear Control*. **2021**, *31*, 2509–2525. <https://doi.org/10.1002/rnc.5419>.
26. Qiu, X.; Hua, C.; Chen, J.; Zhang, L.; Guan, X. Model-free adaptive iterative sliding mode control for a robotic exoskeleton trajectory tracking system. *Int. J. Syst. Sci.* **2020**, *51*, 1782–1797. <https://doi.org/10.1080/00207721.2020.1778114>.
27. Murillo, O.J. A Fast Ascent Trajectory Optimization Method for Hypersonic Air-Breathing Vehicles. Ph.D. Thesis, Iowa State University, Ames, IA, USA, 2010.
28. Lopez-Sanchez, I.; Montoya-Chairez, J.; Perez-Alcocer, R.; Moreno-Valenzuela, J. Experimental Parameter Identifications of a Quadrotor by Using an Optimized Trajectory. *IEEE Access* **2020**, *8*, 167355–167370.

**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.