

## Article

# An Online Generation Method of Terminal-Area Trajectories for Wave-Rider Using Deep Neural Networks

Zhe Liu <sup>1</sup>, Jie Yan <sup>1,\*</sup>, Bangcheng Ai <sup>2</sup>, Yonghua Fan <sup>1</sup>, Kai Luo <sup>2</sup>, Guodong Cai <sup>2</sup> and Jiankai Qin <sup>2</sup>

<sup>1</sup> School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China; neo\_csdn@mail.nwpu.edu.cn (Z.L.)

<sup>2</sup> China Academy of Aerospace Aerodynamics, Beijing 100074, China; 2016302078@mail.nwpu.edu.cn (G.C.)

\* Correspondence: jyan@nwpu.edu.cn

**Abstract:** This paper presents a deep neural network-based online trajectory generation method for the aerodynamic characteristic description and terminal-area energy management of wave-rider aircrafts. First, the flight dynamics equations in the energy domain are linearized and discretized to generate numerous aircraft trajectory samples with sequential convex optimization (SCO) methods. Then, an optimization objective function is designed to promote the smoothness of the control variables and improve the trajectory similarity. Compared to the nonlinear programming (NLP), the proposed trajectory sample generation method is more suitable for the training of deep neural networks (DNNs). Finally, deep neural networks are formulated and trained for the control variables and state variables, using the generated obtained trajectory samples, so that the reference trajectories can be obtained online during the energy management process of the wave-rider's terminal phase. Numerical simulations validate the high accuracy of the trajectories generated with the deep neural network. Meanwhile, this proposed method enables smaller storage usage, which is highly suitable for integration into on-board flight control systems.

**Keywords:** sequential convex optimization; deep neural networks; online generation; terminal-area; wave-rider



**Citation:** Liu, Z.; Yan, J.; Ai, B.; Fan, Y.; Luo, K.; Cai, G.; Qin, J. An Online Generation Method of Terminal-Area Trajectories for Wave-Rider Using Deep Neural Networks. *Aerospace* **2023**, *10*, 654. <https://doi.org/10.3390/aerospace10070654>

Academic Editors: Won-Sang Ra, Shaoming He and Ivan Masmitja

Received: 17 May 2023

Revised: 16 July 2023

Accepted: 22 July 2023

Published: 23 July 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

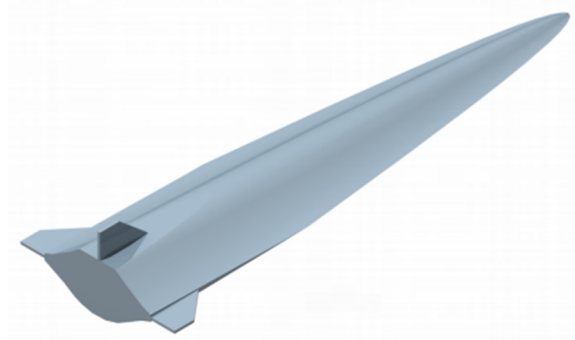
### Background

The concept of wave-rider was first proposed by Nonweiler [1] in 1959. A wave-rider is generally designed for hypersonic flight conditions. During the hypersonic flight, an aircraft rides on top of shock waves, creating a high-pressure flow field on the lower surface, which generates significant lift. The body of the aircraft is typically flat, resulting in low drag. Therefore, a wave-rider aircraft generally has a high lift-to-drag ratio at the design point for hypersonic flight. However, for the off-design points, such as in subsonic conditions, the aerodynamic performance will deteriorate with a lower lift-to-drag ratio. Figure 1 shows a typical wave-rider aircraft.

The research on the design method of wave-riders has made significant progress. In order to solve the problems of excessive negative dihedral angles and insufficient volume ratios, the theory of a cone-guided wave-rider is proposed. To further improve the control of the shock wave shape, the theory of a tangential wave-rider is proposed. Subsequently, based on the tangential theory, some scholars domestically and internationally have obtained a series of new tangential-type wave-rider configurations with more complex configurations and specific purposes [2], such as the variable Mach number wave-rider, double-sweep wave-rider and dual/multistage wave-rider.

The wave-rider aircraft requires energy management techniques to adjust its flight state during the terminal area before approaching and landing. The United States' space shuttle was the first unit which employed the energy management techniques in the end

phase and successfully conducted 135 unpowered gliding approaches and landings. In 2011, the space shuttle “Atlantis” completed the final flight. Following the space shuttle, the United States’ X-37B utilized terminal-area energy management techniques and successfully completed multiple landing missions at the Vandenberg Air Force Base in California and Kennedy Space Center [3,4].



**Figure 1.** Wave-rider aircraft.

The terminal-area energy management segment of the spacecraft is situated between the reentry phase and landing phase. It typically starts from an altitude of  $H = 26$  km and  $Mach = 2.5$ , and it ends at an altitude of 1.5–3 km [5]. During the reentry phase, the spacecraft experiences significant uncertainties in aerodynamic forces and flight conditions due to traversing a large airspace and speed range. As a result, the carrier vehicle may deviate from the nominal trajectory, leading to significant errors of position, heading, and energy at the end of the reentry phase. After the reentry phase, the energy management segment aims to regulate the energy of the aircraft, guiding it along an appropriate trajectory for descent. This ensures that the spacecraft achieves the required altitude and velocity windows for the approach and landing phase while aligning the flight heading with the centerline of the runway, ensuring the safety of the approach and landing process. The trajectory adjustment is more difficult for wave-rider aircraft compared to space shuttles. A detailed comparative analysis is presented in Section 2.1. Therefore, wave-rider aircraft require more precise reference trajectories in the terminal area to reduce the additional trajectory adjustment required by deviations.

The most typical method for designing reference trajectories is the altitude-dynamic pressure profile method [6,7]. However, this method often leads to oscillations or even abrupt changes of the attack angle during the descent trajectory [7]. This is because the traditional altitude-dynamic pressure profile method relies on empirical assumptions, whereas the altitude-dynamic pressure profile is typically designed as a constant or linear profile. The reference trajectory is then obtained by using inverse calculation with the designed profile. However, the assumed profile cannot guarantee the existence of a solution, and the solution process often encounters situations where no solution can be found. Even when a solution exists, it is challenging to ensure the smoothness of the attack angle.

Deep neural networks have gradually been used for the analysis or generation of motion trajectories. For example, in [8], a deep convolutional neural network was employed to classify the trajectories of ships. Reference [9] used a deep neural network to obtain trajectories of high-speed aircraft, and the training samples for the network were obtained through a nonlinear programming method. However, extensive trajectory planning practices have shown that there still exist two unresolved issues with nonlinear programming methods. Firstly, the complex aerodynamic force model of the aircraft brings difficulty of algorithm convergence. Therefore, nonlinear programming methods are more suitable for solving trajectories of space vehicles without aerodynamic forces in the exosphere [10,11], or trajectories with well-behaved aerodynamic models, such as the reentry trajectories of hypersonic aircraft [12–17]. However, in the terminal-area energy management segment, the aircraft passes through the transonic regime where the lift-to-drag ratio shows sig-

nificant changes. That is, nonlinear programming methods converge slowly and require longer computational time in this transonic regime. Secondly, trajectories obtained with nonlinear programming methods may lead to oscillations of control and state variables. The oscillations in trajectories can reduce the fitting accuracy of deep neural networks to the samples. In conclusion, there have been challenges such as long computational time, low efficiency, and low fitting accuracy when using nonlinear programming methods to generate a large number of terminal-area energy management trajectory samples.

Motivated by the above observations, this paper proposes a deep neural network-based online trajectory generation method for wave-rider aircraft in the terminal-area energy management. The main contributions of this work can be summarized as follows.

(a) The trajectory samples are generated based on sequential convex optimization (SCO). Compared to the most commonly used nonlinear programming (NLP) methods, sequential convex optimization (SCO) methods can converge quickly and obtain a large number of trajectory samples with less computation time and higher efficiency, especially when dealing with complex aerodynamic models. Moreover, we incorporate a term in the optimized objective function to suppress oscillations of attack angle. This ensures that the control variables obtained from the sequential convex optimization are smoother, ensuring stronger shape similarity to trajectory samples. Consequently, blending sequential convex optimization (SCO) with deep neural networks (DNNs) enables a more effective sample generation scheme.

(b) We establish deep neural networks separately for the control variable of angle of attack and the state variables of flight path angle, altitude, and velocity. With this method, each deep neural network has only one output: either the control variable of angle of attack or a specific state variable. This simplifies the network structure, reduces the network complexity, and improves the fitting accuracy of network. Moreover, the number of hidden layers in each deep neural network is designed according to the complexity of the sample trajectories. This approach effectively avoids excessive hidden layers, which would waste storage space and potentially lead to overfitting. As a result, this method requires less storage space, lower computational requirements, and thus provides highly accurate trajectories which are suitable for use in onboard computers.

The rest of this paper is organized as follows. Section 1 provides a detailed introduction to the sequential convex optimization (SCO) method used for generating trajectory samples. Section 2 generates trajectory samples using the nonlinear programming (NLP) in comparison with samples generated using sequential convex optimization (SCO) through a comparative analysis. Section 3 establishes the deep neural networks (DNNs) and presents the training method. Section 4 introduces the online trajectory generation method. Finally, Section 5 provides the concluding remarks.

## 2. Sequential Convex Optimization (SCO) Method

When the optimization objective function and inequality constraint functions of an optimization problem are convex with linear equality constraint functions, the optimization problem is referred to as a convex optimization problem. This problem can converge and be solved quickly. It is insensitive to initial guesses, and it typically obtains the global optimal solution within 10 to 100 steps. Therefore, they are suitable for online applications. This method has explicit function type restrictions on the objective and constraint functions and needs to be transformed into a convex optimization problem before solving [18]. Such convex optimization is usually solved using primal–dual interior point methods [19–24].

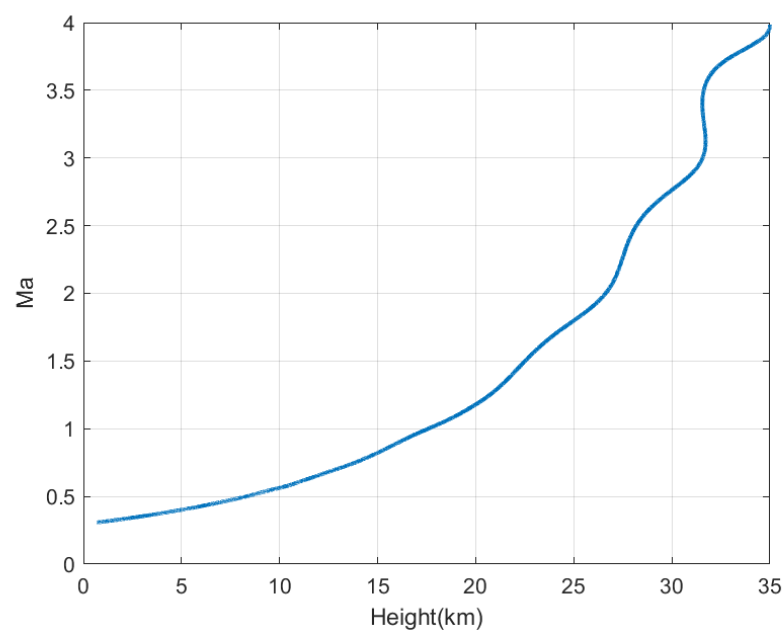
Flight dynamics equations have highly nonlinear equality constraints, but convex optimization methods require the equality constraints to be linear functions. Thus, linearization approximation is usually applied to the flight dynamics equations, only keeping the linear parts as equality constraints. The linearization of the dynamics equations is performed within the neighborhood of the optimized trajectory obtained in the previous optimization iteration. Solving a trajectory optimization problem requires conducting multiple rounds of convex optimization sequentially, which are named sequential convex optimization [25–37].

In this work, considering the need for efficient and fast optimization to generate a large number of trajectory samples for training deep neural networks, the sequential convex optimization method is employed.

### 2.1. Wave-Rider Aircraft Aerodynamics and Modeling

In terms of aerodynamic performance, compared to the space shuttles, wave-rider aircraft has a higher lift-to-drag ratio during the hypersonic flight regime, but their lift-to-drag ratio is similar to the space shuttle at low speeds. Therefore, in terms of body lift and drag characteristics, a wave-rider aircraft has similar energy management capabilities to the space shuttle. However, in terms of aerodynamic layout, there are more differences. On one hand, the space shuttle is equipped with dedicated speed brakes, which provide strong speed control capabilities. On the other hand, the large-sized split rudder can also contribute to the deceleration. In contrast, a wave-rider aircraft has a flatter body shape and a smaller tail section, making it challenging to install a servo system for speed brakes. Moreover, the weight of the speed brake servo system would shift the center of mass rearward and increase the static instability of the wave-riding aircraft. The vertical tail of the wave-rider aircraft is much smaller in size compared to the space shuttle, resulting in weaker speed control capabilities. These aerodynamic limitations make the trajectory adjustment more difficult for wave-rider aircraft compared to space shuttles.

In this paper, a type of typical wave-rider aircraft is chosen as the research object, and this wave-rider aircraft is designed for flight testing. The shape of the wave-rider is mainly designed based on the shock-fitting method, and the volume ratio is improved [38,39]. The designed flight condition of the wave-rider is less than 35 km, and the Mach number is less than 4. The designed range is less than 300 km. Furthermore, the aircraft is unpowered and is delivered to the desired flight condition by a booster rocket. A typical flight profile of the aircraft is shown in Figure 2.

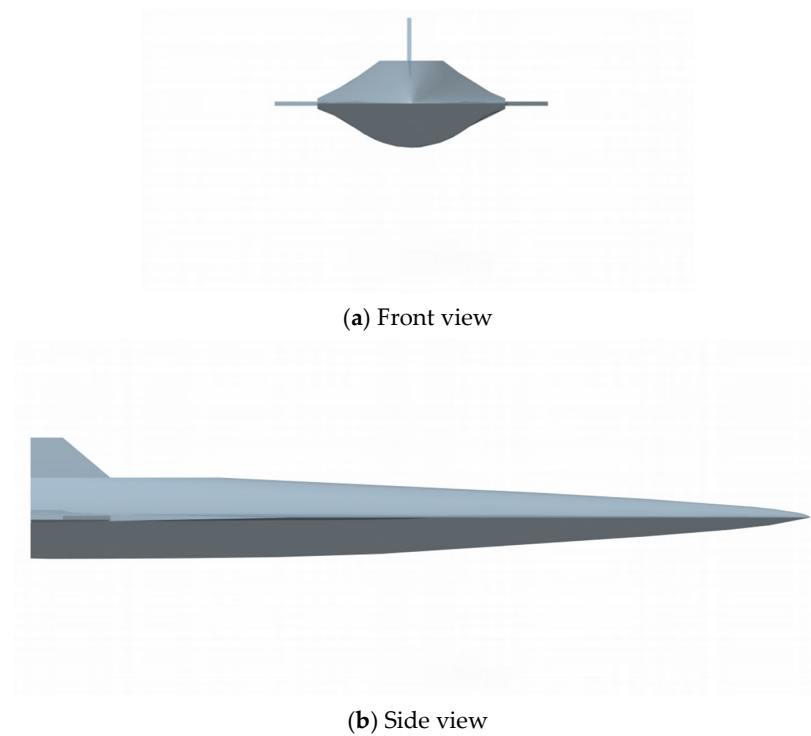


**Figure 2.** Typical flight profile.

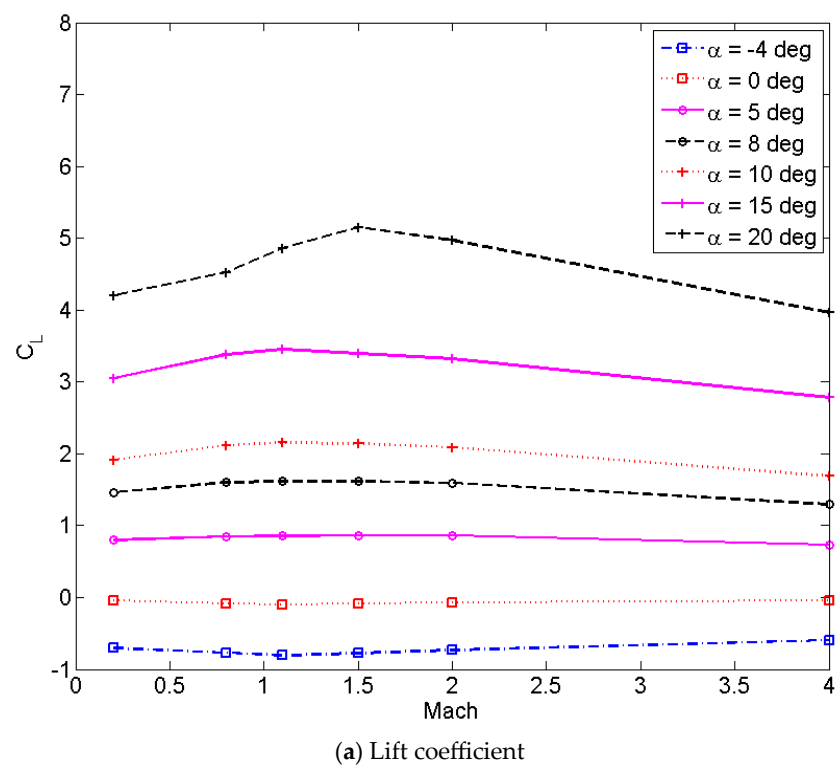
The 3D view of the wave-rider is as shown in Figure 1. The 2D view of the aircraft is shown in Figure 3a,b. The mass is  $m = 300$  kg, and the reference area is  $S_{ref} = 0.3$  m<sup>2</sup>. The axial length is chosen as the reference length,  $L_{ref} = 3$  m, and the span is  $L_{span} = 1$  m.

The aerodynamic performance evaluation was conducted using the numerical simulation software developed independently. The main algorithm of the software is based on reference [40]. The non-structured Cartesian grid was selected, and the grid was refined near the surface of the wave-rider body with a grid count of approximately 6.2 million.

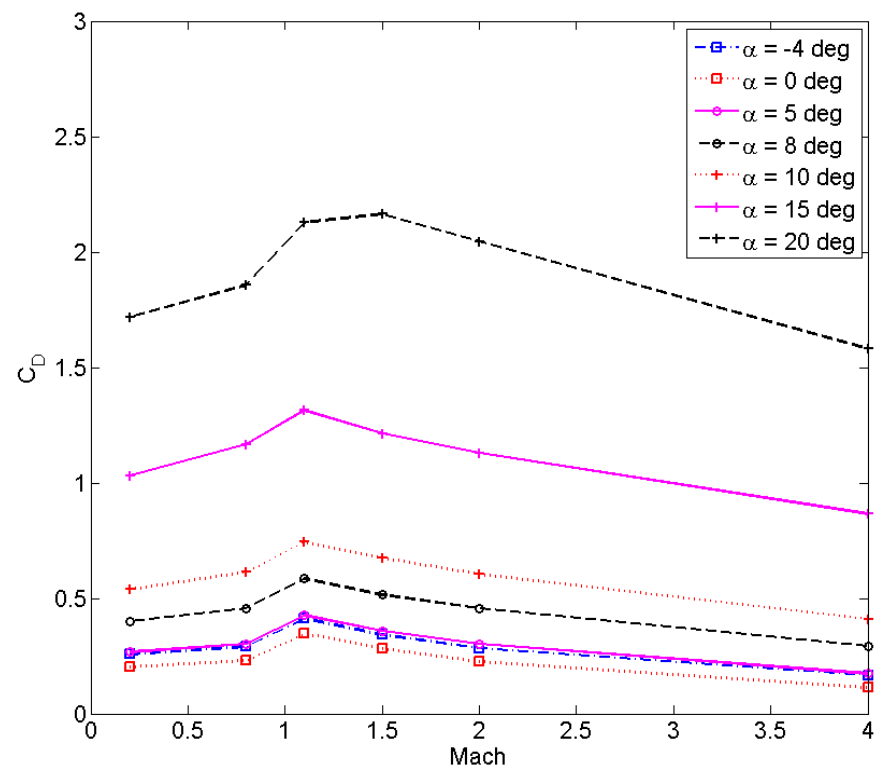
The spatial discretization was implemented using the SD-SLAU numerical scheme, and the time discretization was carried out using the LU-SGS implicit method. The variable reconstruction employed the MUSCL method. The lift coefficient, drag coefficient, and lift-to-drag ratio of this aircraft are shown in Figure 4a–c respectively, where the unit of the angle of attack is degree.



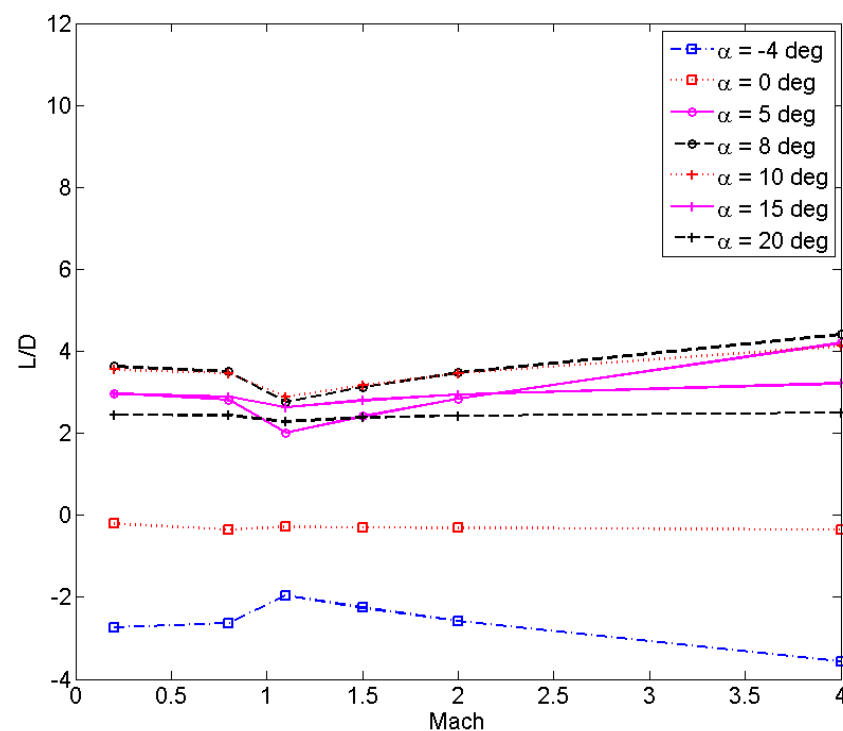
**Figure 3.** Two-dimensional (2D) view of the aircraft.



**Figure 4.** Cont.



(b) Drag coefficient



(c) Lift-to-drag ratio

**Figure 4.** Aerodynamic force coefficients.

## 2.2. The Longitudinal Dynamic Equations in the Energy Domain

Trajectory planning is performed in the longitudinal plane. A two-degree-of-freedom point mass dynamic model is adopted in the launching coordinate system. The optimization model is dimensionless. The dimensionless dynamic model is shown in Equations (1)–(4).

The dimensionless quantities for distance are denoted as  $l_s = R_0$ , while those for velocity are denoted as  $V_s = \sqrt{R_0 g_0}$ , and those for time are denoted as  $t_s = \sqrt{\frac{R_0}{g_0}}$ . The dimensionless variables  $\bar{x}, \bar{y}, \bar{r}, \tau, \bar{V}, \gamma, \bar{L}, \bar{D}$  are used. Specifically,  $x = l_s \bar{x}$ ,  $y = l_s \bar{y}$ ,  $\bar{r} = 1 + \bar{y}$ ,  $t = t_s \tau$ ,  $V = V_s \bar{V}$ ,  $\gamma = \bar{\gamma}$ ,  $\bar{L} = \frac{1}{2} R_0 \rho \bar{V}^2 S C_{L_m}$ ,  $\bar{D} = \frac{1}{2} R_0 \rho \bar{V}^2 S C_{D_m}$ . The dimensionless energy  $e$  of the aircraft is defined in Equation (5), with which Equations (6) and (7) are obtained. By combining Equations (1)–(4) and (7), the dimensionless dynamic equations in the energy domain can be obtained as Equations (8)–(10).

$$\frac{d\bar{x}}{d\tau} = \bar{V} \cos \gamma \quad (1)$$

$$\frac{d\bar{y}}{d\tau} = \bar{V} \sin \gamma \quad (2)$$

$$\frac{d\bar{V}}{d\tau} = -\bar{D} - \frac{\sin \gamma}{\bar{r}^2} \quad (3)$$

$$\frac{d\gamma}{d\tau} = \frac{\bar{L}}{\bar{V}} - \frac{\cos \gamma}{\bar{V} \bar{r}^2} \quad (4)$$

$$e = \frac{1}{\bar{r}} - \frac{1}{2} \bar{V}^2 \quad (5)$$

$$\bar{V} = \sqrt{2 \left( \frac{1}{\bar{r}} - e \right)} \quad (6)$$

$$\frac{de}{d\tau} = \bar{D} \bar{V} \quad (7)$$

$$\frac{d\bar{x}}{de} = \frac{\cos \gamma}{\bar{D}} \quad (8)$$

$$\frac{d\bar{y}}{de} = \frac{\sin \gamma}{\bar{D}} \quad (9)$$

$$\frac{d\gamma}{de} = \frac{\bar{L}}{\bar{D} \bar{V}^2} - \frac{\cos \gamma}{\bar{D} \bar{V} \bar{r}^2} \quad (10)$$

### 2.3. Equation Constraints

Equations (8)–(10) can be rewritten as

$$\frac{d\mathbf{x}}{de} = \mathbf{f}(\mathbf{x}, u) \quad (11)$$

where  $\mathbf{x} = [\bar{x}, \bar{y}, \gamma]^T$ , and  $u = \alpha$ . The optimal trajectory obtained from the previous optimization round is denoted as  $(\mathbf{x}^k, u^k)$ , where  $k$  represents the optimization round.  $(\mathbf{x}^{k+1}, u^{k+1})$  represents the trajectory to be optimized in the current round. In Equation (12), the right-hand side term  $\mathbf{f}(\mathbf{x}, u)$  in Equation (11) is the Taylor expanded along the trajectory  $(\mathbf{x}^k, u^k)$ , and the left-hand side term in Equation (11) is discretized in the energy domain. Here,  $j$  denotes the index of the discrete points,  $\Delta e = (e_f - e_0)/N$ , where  $N$  is the number of discrete

energy segments.  $e_0$  represents the initial energy, and  $e_f$  represents the terminal energy.  $v_j$  represents a virtual control term with unrestricted magnitude.  $v_j$  ensures the feasibility of Equation (12) throughout the optimization process and should be minimized.  $E_j$  is the coefficient matrix for  $v_j$ , which allows full controllability of  $x^k$  and is typically chosen as the identity matrix. Matrices  $A$ ,  $B$ , and  $C$  are defined as shown in Equation (13). The specific calculation methods for matrices  $A$  and  $B$  are shown in Equations (14) to (30).

$$\frac{2(x_{j+1}^{k+1} - x_j^{k+1})}{\Delta e} = A(x_j^k, u_j^k)x_j^{k+1} + B(x_j^k, u_j^k)u_j^{k+1} + C(x_j^k, u_j^k) + A(x_{j+1}^k, u_{j+1}^k)x_{j+1}^{k+1} + B(x_{j+1}^k, u_{j+1}^k)u_{j+1}^{k+1} + C(x_{j+1}^k, u_{j+1}^k) + E_j v_j \quad (12)$$

$$j = 1, 2, \dots, N$$

$$A(x, u) = \frac{\partial f}{\partial x}, \quad B(x, u) = \frac{\partial f}{\partial u}, \quad C(x, u) = f(x, u) - A(x, u)x - B(x, u)u \quad (13)$$

$$A = \begin{bmatrix} 0 & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \end{bmatrix} \quad (14)$$

$$a_{12} = -\frac{\cos \gamma}{\bar{D}} \frac{\partial \bar{D}}{\partial \bar{y}}, \quad a_{13} = -\frac{\sin \gamma}{\bar{D}} \quad (15)$$

$$a_{22} = -\frac{\sin \gamma}{\bar{D}} \frac{\partial \bar{D}}{\partial \bar{y}}, \quad a_{23} = \frac{\cos \gamma}{\bar{D}} \quad (16)$$

$$a_{32} = \frac{\partial \bar{L}}{\partial \bar{y}} \frac{1}{\bar{D} \bar{V}^2} - \frac{\bar{L}}{\bar{D}^2 \bar{V}^2} \frac{\partial \bar{D}}{\partial \bar{y}} - \frac{\bar{L}}{\bar{D} \bar{V}^4} \frac{\partial \bar{V}^2}{\partial \bar{y}} - \cos \gamma \left( -\frac{1}{\bar{D}^2 \bar{V}^2} \frac{\partial \bar{D}}{\partial \bar{y}} - \frac{1}{\bar{D} \bar{V}^4} \frac{\partial \bar{V}^2}{\partial \bar{y}} - \frac{1}{\bar{D} \bar{V}^2} \frac{\partial \bar{r}}{\partial \bar{y}} \right) \quad (17)$$

$$a_{33} = \frac{\sin \gamma}{\bar{V}^2 (1 + \bar{y})^2 \bar{D}} \quad (18)$$

$$b_{11} = -\frac{\cos \gamma}{\bar{D}} \frac{\partial \bar{D}}{\partial \alpha} \quad (19)$$

$$b_{21} = -\frac{\sin \gamma}{\bar{D}} \frac{\partial \bar{D}}{\partial \alpha} \quad (20)$$

$$b_{31} = \frac{1}{\bar{V} \bar{D}} \frac{\partial \bar{L}}{\partial \alpha} - \frac{\bar{L}}{\bar{V} \bar{D}^2} \frac{\partial \bar{D}}{\partial \alpha} + \cos \gamma \frac{1}{\bar{V} \bar{r}} \frac{1}{\bar{D}^2} \frac{\partial \bar{D}}{\partial \alpha} \quad (21)$$

where,

$$\frac{\partial \bar{D}}{\partial \bar{y}} = \frac{1}{2m} R_0 \frac{\partial \rho}{\partial \bar{y}} \bar{V}^2 S_{CD} + \frac{1}{2m} R_0 \rho \frac{\partial \bar{V}}{\partial \bar{y}} S_{CD} + \frac{1}{2m} R_0 \rho \bar{V}^2 S \frac{\partial C_D}{\partial Ma} \frac{\partial Ma}{\partial \bar{V}} \frac{\partial \bar{V}}{\partial \bar{y}} \quad (22)$$

$$\frac{\partial \bar{L}}{\partial \bar{y}} = \frac{1}{2m} R_0 \frac{\partial \rho}{\partial \bar{y}} \bar{V}^2 S_{CL} + \frac{1}{2m} R_0 \rho \frac{\partial \bar{V}}{\partial \bar{y}} S_{CL} + \frac{1}{2m} R_0 \rho \bar{V}^2 S \frac{\partial C_L}{\partial Ma} \frac{\partial Ma}{\partial \bar{V}} \frac{\partial \bar{V}}{\partial \bar{y}} \quad (23)$$



$$\frac{\partial \rho}{\partial \bar{y}} = -\frac{l_s}{H} \rho \quad (24)$$

$$\frac{\partial Ma}{\partial \bar{V}} = -\frac{V_s}{V_{sound}} \quad (25)$$

$$\frac{\partial \bar{V}^{-2}}{\partial \bar{y}} = -\frac{2}{(1 + \bar{y})^2} \quad (26)$$

$$\frac{\partial \bar{V}}{\partial \bar{y}} = -\left(\frac{2}{(1 + \bar{y})} - 2e\right)^{(-\frac{1}{2})} \left(\frac{1}{(1 + \bar{y})^2}\right) \quad (27)$$

$$\frac{\partial \bar{r}^{-2}}{\partial \bar{y}} = 2(1 + \bar{y}) \quad (28)$$

$$\frac{\partial \bar{L}}{\partial \alpha} = \frac{1}{2m} R_0 \rho \bar{V}^{-2} S \frac{\partial C_L}{\partial \alpha} \quad (29)$$

$$\frac{\partial \bar{D}}{\partial \alpha} = \frac{1}{2m} R_0 \rho \bar{V}^{-2} S \frac{\partial C_D}{\partial \alpha} \quad (30)$$

In Equations (22) and (23),  $\frac{\partial C_D}{\partial Ma}$  and  $\frac{\partial C_L}{\partial Ma}$  are calculated using central differencing based on aerodynamic data. In Equation (24),  $H$  is a constant with a value of  $H = 7254.24$  m.  $V_{sound}$  represents the speed of sound.  $\rho$  and  $V_{sound}$  are calculated using the 1976 U.S. Standard Atmosphere.

In Equations (29) and (30),  $\frac{\partial C_D}{\partial \alpha}$  and  $\frac{\partial C_L}{\partial \alpha}$  are calculated using central differencing based on aerodynamic data.

During the optimization process, the lift and drag coefficients, as well as their derivatives with respect to Mach number and angle of attack, are directly calculated using the original aerodynamic data table. There is no need for data fitting, and this ensures better adaptation to aerodynamic characteristics with sudden changes in the transonic region and thus enables higher trajectory accuracy. The initial states of the trajectory should satisfy the equality constraint as shown in the following equation:

$$\mathbf{x}_1^{k+1} = [\bar{x}_0, \bar{y}_0, \gamma_0]^T \quad (31)$$

#### 2.4. Path Constraints

To maintain the convergence of the sequential convex optimization, the current optimized trajectory should not deviate far from the trajectory obtained in the previous iteration. Therefore, it is necessary to satisfy the inequality constraint (32). Here,  $\varepsilon$  can be chosen as a constant parameter based on experience.

$$|\mathbf{x}^{k+1} - \mathbf{x}^k| \leq \varepsilon \quad (32)$$

During the flight, the rate of change of the angle of attack should be kept below a certain threshold to maintain the stable flight attitude. The constraint on the angle of attack rate

can be described as Equation (33). Here,  $\Delta t = \Delta \tau (R_0/g_0)^{0.5}$ . From Equations (7) and (33), Equation (34) can be obtained.

$$\frac{|u_{j+1}^{k+1} - u_j^{k+1}|}{\Delta t} \leq \dot{\alpha}_{max}, j = 1, 2, \dots, N \quad (33)$$

$$|u_{j+1}^{k+1} - u_j^{k+1}| \leq \dot{\alpha}_{max} \sqrt{\frac{R_0}{g_0} \frac{\Delta e}{DV}}, j = 1, 2, \dots, N \quad (34)$$

In Equation (34), the calculation of  $\Delta e$  is given as (35)–(37).

$$\Delta e = \frac{e_f - e_0}{N} \quad (35)$$

$$e_0 = \frac{1}{r_0} - \frac{1}{2} \bar{V}_0^2 \quad (36)$$

$$e_f = \frac{1}{r_f} - \frac{1}{2} \bar{V}_f^2 \quad (37)$$

In addition to limiting the angle of attack rate, the magnitude of the angle of attack should also be restricted. Excessive negative or positive angles of attack can cause the failure of aircraft control. The angle of attack constraint is shown in the following equation:

$$\alpha_{min} \leq u_j^{k+1} \leq \alpha_{max}, j = 1, 2, \dots, N \quad (38)$$

During the flight, the aircraft should not descend below a certain altitude:

$$\bar{y}_j \geq \bar{y}_{min}, j = 1, 2, \dots, N + 1 \quad (39)$$

The trajectory of the aircraft during the flight can be designed below a certain limit value  $\gamma_{max}$ , as shown in the following equation:

$$\gamma_j \leq \gamma_{max}, j = 1, 2, \dots, N + 1 \quad (40)$$

## 2.5. Optimization Objective Function

The optimization objective is represented as a weighted sum of five terms, as shown in Equation (41). The first three terms aim to make the final state variables  $\bar{x}$ ,  $\bar{y}$  and  $\gamma$  of the optimal trajectory as close as possible to their design values. These terms are included in the objective function as soft constraints to seek the closest feasible solution when the desired values cannot be achieved exactly. The fourth term aims to minimize the virtual control variable  $v$  as much as possible, where  $v = [v_1, v_2, \dots, v_N]$ . The fifth term effectively reduces the accumulated rate of change of the control variables along the entire trajectory. When the final state variables reach the design values and the virtual control variable is small, the fifth term becomes the primary optimization objective in the objective function. It helps to suppress the oscillation of control variables. In Equation (41),  $C_y$ ,  $C_\gamma$ ,  $C_u$ , and  $C_v$  are positive real numbers that can be adjusted based on the optimization results.

$$J = \bar{x}_{tol} + C_y \bar{y}_{tol} + C_\gamma \gamma_{tol} + C_u \bar{u}_{tol} + C_v v_{tol} \quad (41)$$

In Equation (41),  $\bar{x}_{tol}$ ,  $\bar{y}_{tol}$ ,  $\gamma_{tol}$ ,  $\bar{u}_{tol}$ , and  $v_{tol}$  satisfy the following constraints.

$$|\bar{x}_{N+1}^{k+1} - \bar{x}_f| \leq \bar{x}_{tol} \quad (42)$$

$$|\bar{y}_{N+1}^{k+1} - \bar{y}_f| \leq \bar{y}_{tol} \quad (43)$$

$$|\gamma_{N+1}^{k+1} - \gamma_f| \leq \gamma_{tol} \quad (44)$$

$$\sum_{j=1}^N |u_{j+1}^{k+1} - u_j^{k+1}| \leq u_{tol} \quad (45)$$

$$\|v_j\|_1 \leq v_{tol}, j = 1, \dots, N \quad (46)$$

So far, Equations (12) to (46) formulate the complete sequential convex optimization (SCO) problem. This problem can be solved using the CVX software package.

## 2.6. Trajectory Samples Design

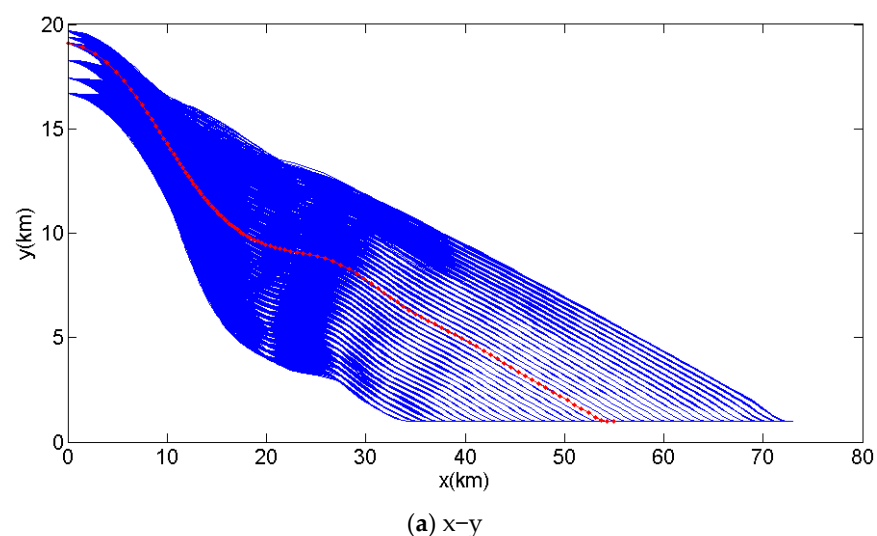
This work focuses on the trajectory samples design for a wave-rider aircraft.

By determining the initial energy  $e_0$ , initial velocity  $V_0$ , initial flight path angle  $\gamma_0$ , final height  $y_f$ , final velocity  $V_f$ , and final flight path angle  $\gamma_f$ , and specifying different final ranges  $x_f$ , a set of trajectories can be obtained. In this scenario, the initial and final states are shown in Table 1.

**Table 1.** Initial and terminal conditions.

Initial Conditions	$e_0$	$\gamma_0$ (°)	$V_0$ (m/s)	
Value	0.996297	0, −1, −2, −3, −4, −5	280, 290, 300, 325, 350, 370	
Terminal conditions	$y_f$ (km)	$\gamma_f$ (°)	$V_f$ (m/s)	$x_f$ (km)
Value	1.0	0.0	90.0	[35, 36, ..., 73]

With Table 1, a set of trajectories is obtained, as shown in Figure 5, with each trajectory consisting of 101 nodes. The red curve represents the trajectory for  $V_0 = 300$ ,  $\gamma_0 = -2^\circ$ , and  $x_f = 55$  km. As shown in Figure 5a, the curves of the x-y profile have a similar shape. Similarly, the curves of the e-y, e-v, e-gamma, and e-alpha profiles also have a similar shape, as depicted in Figure 5b–d, respectively. These characteristics make the trajectories suitable for training samples in deep neural networks.



**Figure 5.** Cont.

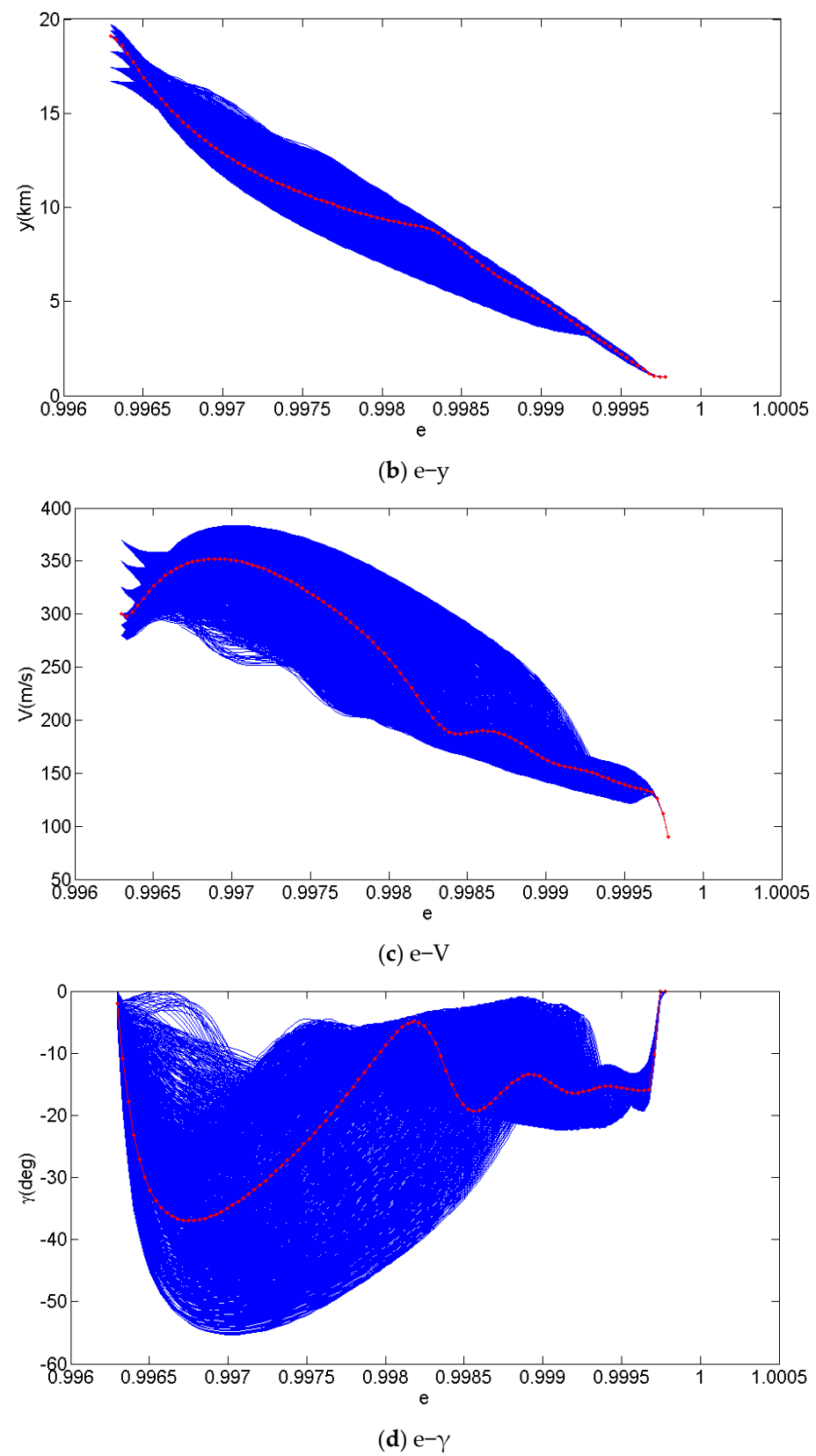
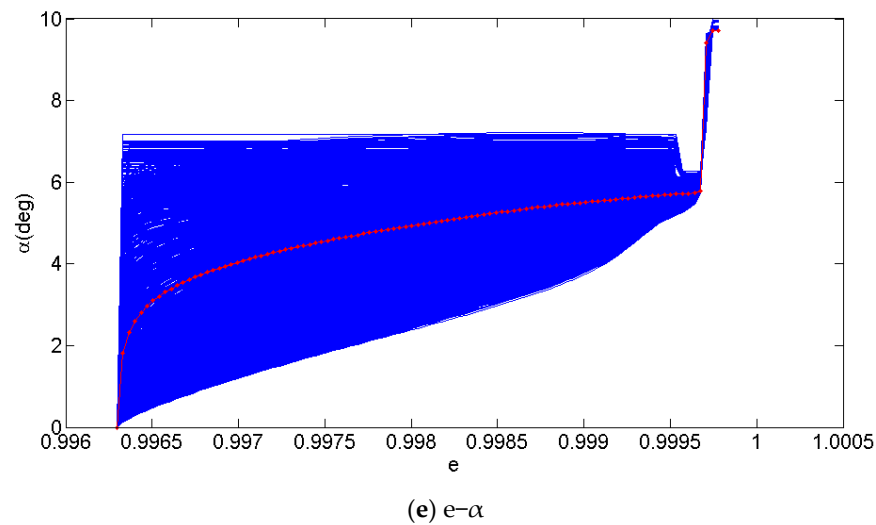


Figure 5. Cont.



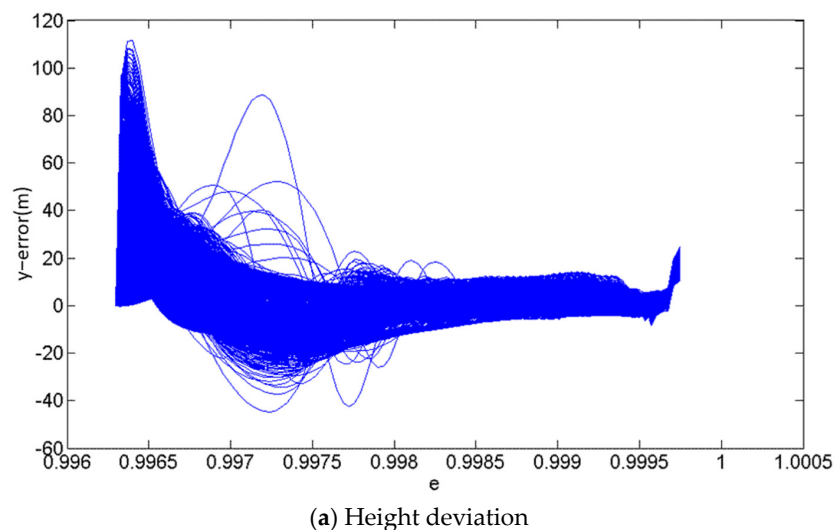
**Figure 5.** Trajectory samples obtained by sequential convex optimization (SCO) method.

### 2.7. Accuracy and Convergence Analysis of the Optimization Method

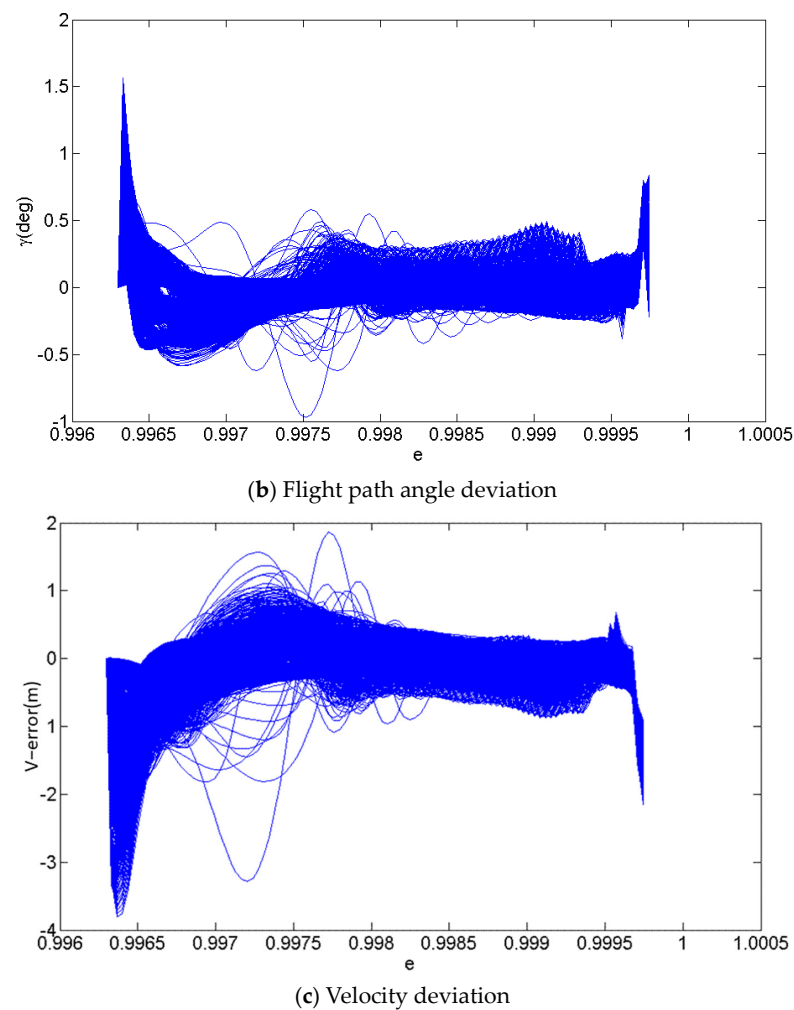
In this section, numerical methods are used to analyze the accuracy of the sequential convex optimization algorithm. Using the sequential convex optimization algorithm with a trajectory discretization of  $N = 100$ , the optimized trajectory is obtained. The simulated trajectory is obtained by numerically integrating Equations (1) to (4), using the energy-angle of attack profile  $[e-\alpha]_{\text{opt}}$  of the optimized trajectory. The deviation of the optimization algorithm is defined as the difference between the states of the optimized trajectory and the simulated trajectory at the same  $x$ -coordinate as:

$$\begin{aligned} y_{\text{error}} &= y_{\text{SCO}}(x) - y_{\text{sim}}(x) \\ \gamma_{\text{error}} &= \gamma_{\text{SCO}}(x) - \gamma_{\text{sim}}(x) \\ V_{\text{error}} &= V_{\text{SCO}}(x) - V_{\text{sim}}(x) \end{aligned} \quad (47)$$

Figure 6a–c illustrate the height deviation, flight path angle deviation, and velocity deviation of the optimized trajectory, respectively. As shown in Figure 6a–c, the optimization algorithm shows high accuracy. The height deviation is within 110 m, the trajectory inclination angle deviation is within 1.5 degrees, and the velocity deviation is within 4 m/s. These results meet the requirements for engineering applications.



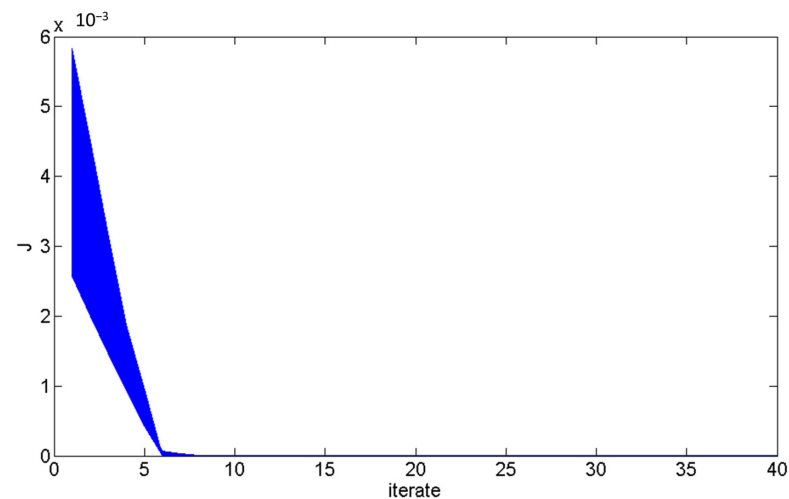
**Figure 6.** Cont.



**Figure 6.** Deviation of the optimized trajectory.

### 2.8. Convergence Analysis

Figure 7 illustrates the convergence process of the optimization objective function  $J$ . The definition of  $J$  is given in Equation (41). From Figure 7, it can be observed that the values of  $J$  for all trajectories converge within 25 iterations. After convergence,  $J$  is kept below  $1 \times 10^{-5}$ .



**Figure 7.** Convergence process of optimization objective function  $J$ .

### 3. The Nonlinear Programming (NLP) Method

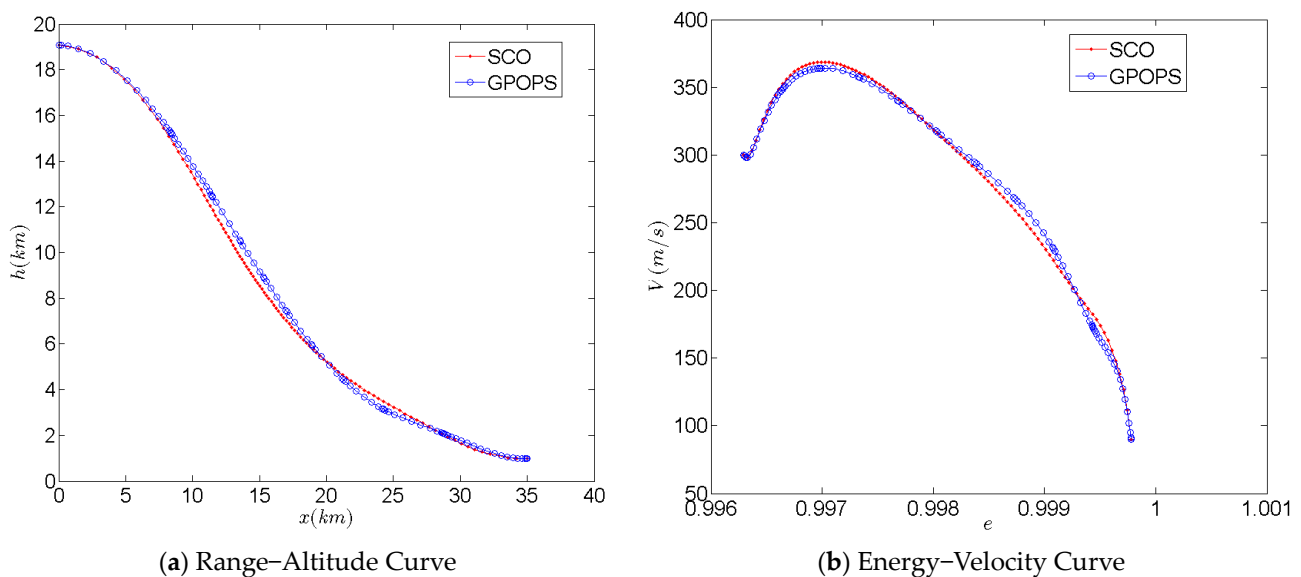
To demonstrate the superiorities of the proposed trajectory generation method using sequential convex optimization (SCO), this section provides a comparative study with the nonlinear programming (NLP) approach. The NLP approach is implemented using the GPOPS software. The dynamic equality constraints, path constraints, and optimization objective function for SCO are described in Section 2. The NLP approach also has similar constraint functions and optimization objective functions but with some differences. Firstly, in the equality constraints, the NLP approach directly discretizes the nonlinear dynamic equations without linearization. Secondly, the path constraints in the NLP approach do not require Equation (32). This is because the NLP approach does not need to linearize the dynamic equations and therefore does not rely on the results of the previous trajectory optimization. The remaining path constraints are kept the same. Thirdly, the optimization objective function in the NLP approach retains the first three terms of Equation (41). The fourth term is not necessary, since the NLP approach does not linearize the trajectory. The fifth term is updated as Equation (48). The reason is that using Equation (45) in GPOPS tests leads to difficult convergence, whereas using Equation (48) achieves better convergence and smoother control variables.

$$\sum_{j=1}^N (u_{j+1}^{k+1} - u_j^{k+1})^2 \leq u_{tol} \quad (48)$$

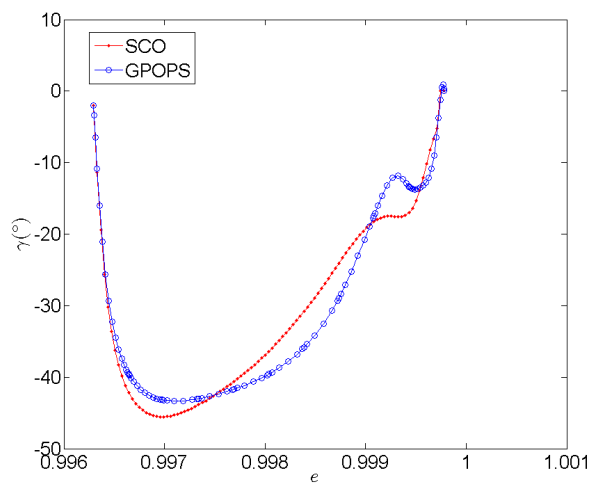
The GPOPS software is used to generate an equal number of trajectories with the initial and terminal trajectory conditions shown in Table 2. Three representative trajectories are selected for comparison. The initial conditions for all trajectories are the same: energy  $e_0 = 0.996297$ ,  $V_0 = 300$ ,  $\gamma_0 = -2^\circ$ , and  $\alpha_0 = 0^\circ$ . The values of  $x_f$  are chosen as 35 km, 55 km, and 75 km. Figures 8–10 illustrate the comparison of the results obtained using sequential convex optimization (SCO) and GPOPS for these three trajectories.

**Table 2.** Comparison of solution time between SCO and GPOPS.

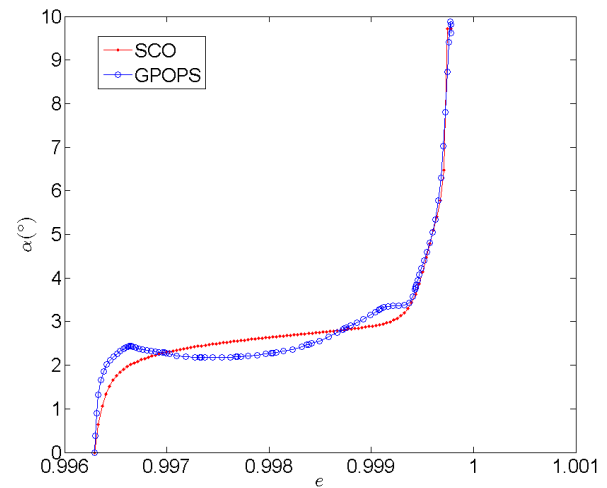
$x_f$ (km)	SCO (s)	GPOPS (s)
35	40.6	108.5
55	25.5	135.7
75	41.7	90.2



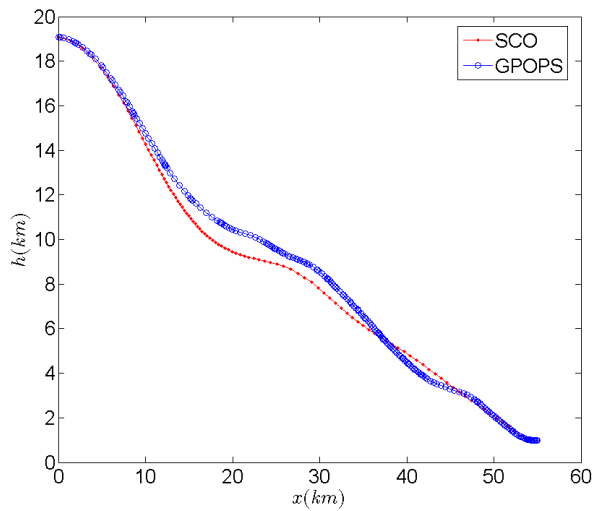
**Figure 8.** Cont.



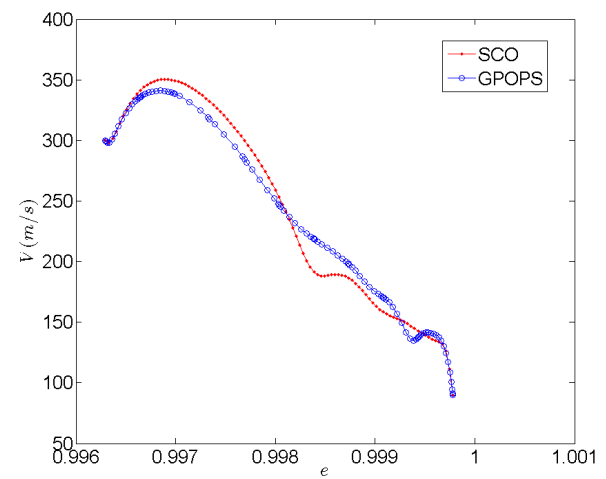
(c) Energy–Trajectory Angle Curve



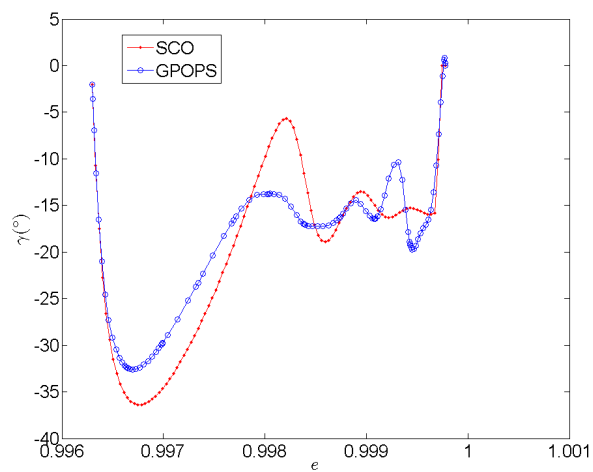
(d) Energy–Angle of Attack Curve

**Figure 8.** Comparison of convex optimization and GPOPS results— $x_f = 35$  km.

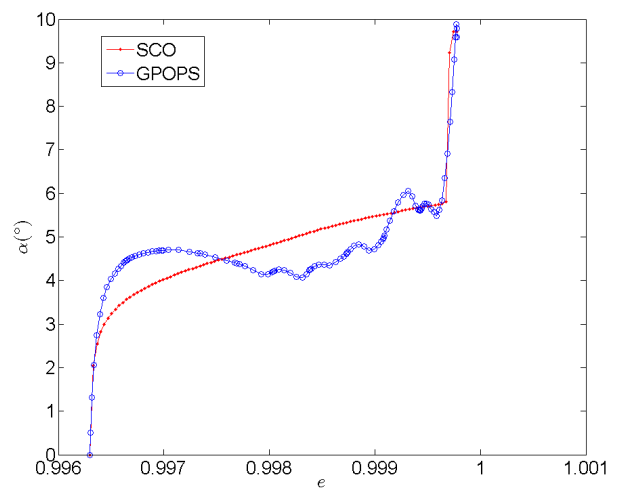
(a) Range–Altitude Curve



(b) Energy–Velocity Curve



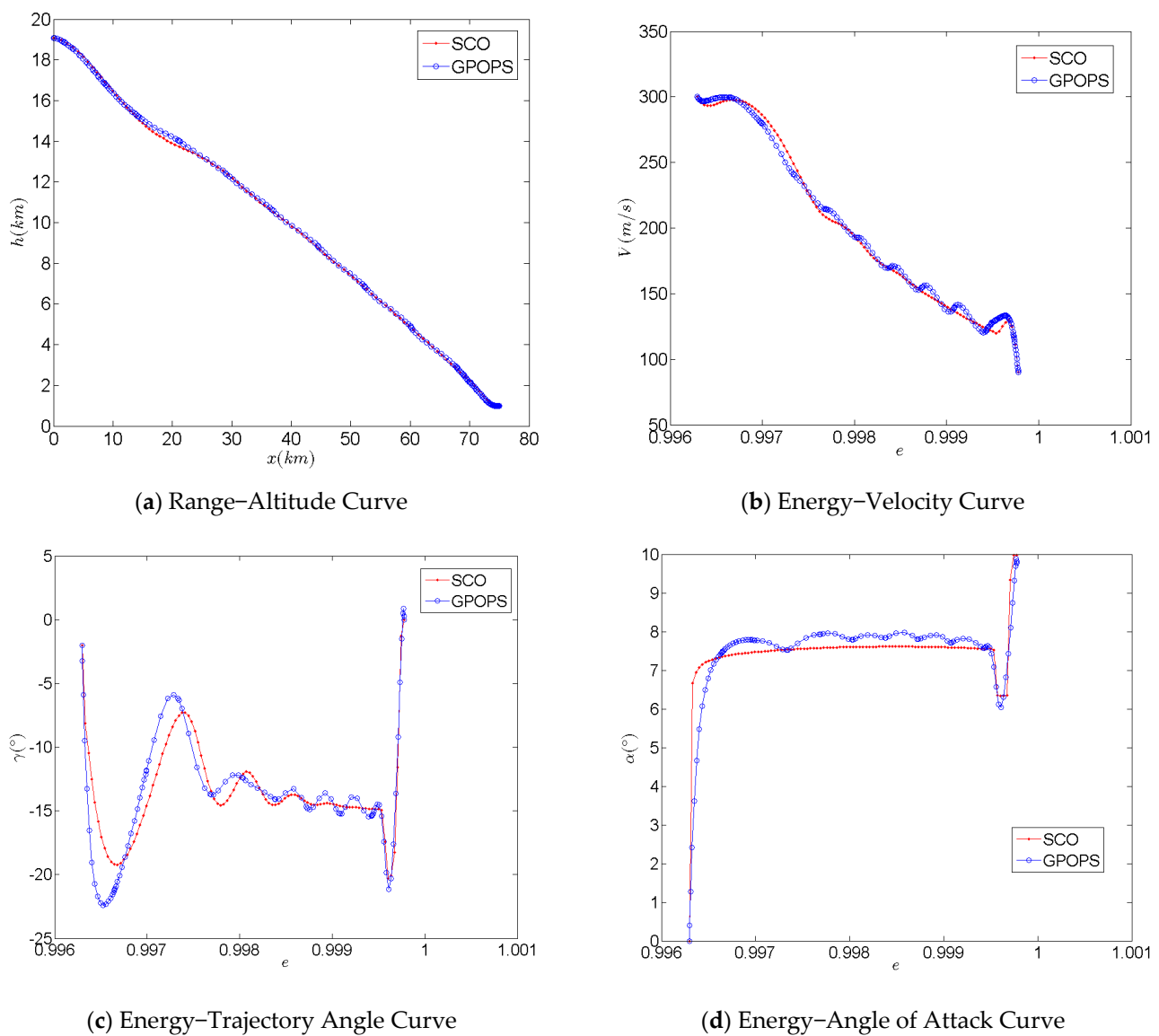
(c) Energy–Trajectory Angle Curve



(d) Energy–Angle of Attack Curve

**Figure 9.** Comparison of convex optimization and GPOPS results— $x_f = 55$  km.





**Figure 10.** Comparison of convex optimization and GPOPS results— $x_f = 75$  km.

From Figures 8–10, it is seen that the results obtained from SCO and GPOPS are quite similar, and their trends are consistent. However, compared to SCO, the oscillations in the angle of attack curve are more obvious for GPOPS. Additionally, for the curve with  $x_f = 75$  km obtained by GPOPS, the oscillations in the angle of attack lead to significant fluctuations in velocity. Table 2 provides a comparison of the time taken by SCO and GPOPS to solve the three trajectories. It is evident from Table 2 that SCO demonstrates superior efficiency compared to GPOPS.

#### 4. Building and Training Deep Neural Networks

To ensure more storage of trajectory samples in a limited storage space for onboard computer applications, this work utilizes the deep neural networks (DNNs) to fit the trajectory samples obtained in Section 2.

Four DNNs are designed to fit the flight path angle, angle of attack, velocity, and altitude samples depicted in Figure 5b–e, respectively. Considering the complex shapes of the flight path angle curves, angle of attack curves, velocity curves, and altitude curves, the DNNs adopts multiple hidden layers to achieve accurate fitting. The number of layers is properly tuned for balanced efficiency and effectiveness.

Due to the significant fluctuations in the flight path angle curve, a DNN with 10 hidden layers is selected to fit the flight path angle curve. For the angle of attack curve and velocity curve with relatively moderate fluctuations, the DNNs with five hidden layers are chosen for fitting. As for the altitude curve, which exhibits the slowest variation, a DNN with three hidden layers is sufficient to achieve high accuracy. The first hidden layer of each network consists of 10 neurons, and the remaining hidden layers contain five neurons.

The input of the deep neural network is designed to consist of four variables: initial velocity  $V_0$ , initial trajectory angle  $\gamma_0$ , range  $x_f$ , and energy  $e$ . The output is a single variable, which corresponds to the angle of attack for the angle of attack network, flight path angle for the flight path angle network, velocity for the velocity network, and height for the height network. Therefore, each training sample for the deep neural network consists of five values: the first four are inputs, and the last one is the output, as shown in the following equation:

$$[V_0, \gamma_0, x_f, e, \alpha/\gamma/V/y] \quad (49)$$

The specific configurations and training methods for the four deep neural networks are shown in Table 3. The network structures for the four networks are depicted in Figure 11. According to Table 3, the flight path angle network only requires storing the data of 306 weights, while the angle of attack and velocity networks require storing 201 weights. The altitude network only needs to store 141 weights. Therefore, the deep neural network database requires less storage space, making it suitable for online applications in on-board computers.

**Table 3.** Configuration of deep neural networks.

Networks	Flight Path Angle Network	Angle of Attack Network	Velocity Network	Altitude Network
Inputs	$V_0, \gamma_0, x_f, e$	$V_0, \gamma_0, x_f, e$	$V_0, \gamma_0, x_f, e$	$V_0, \gamma_0, x_f, e$
Outputs	$\gamma$	$\alpha$	$V$	$y$
Number of Hidden Layers	10	5	5	3
Number of Neurons per Hidden Layer	[10,5,5,...,5]	[10,5,5,5,5]	[10,5,5,5,5]	[10,5,5]
Activation Function	tansig	tansig	tansig	tansig
Number of Weights	306	201	201	141
Training Method	Levenberg–Marquardt	Levenberg–Marquardt	Levenberg–Marquardt	Levenberg–Marquardt

All four deep neural networks are trained using the Levenberg–Marquardt method. The performance function used is the mean square error function, as defined in Equation (50), which calculates the mean squared error between the target values and the network output for all training samples. In Equation (50),  $N$  represents the number of samples. For each network, 70% of the samples are randomly selected as the training set, 15% are randomly selected as the validation set, and 15% are randomly selected as the test set. The four networks are trained five times, and the results of the performance function  $F$  for each training time are shown in Figure 12a–d. A smaller value of the performance function  $F$  indicates a better fit of the deep neural network.

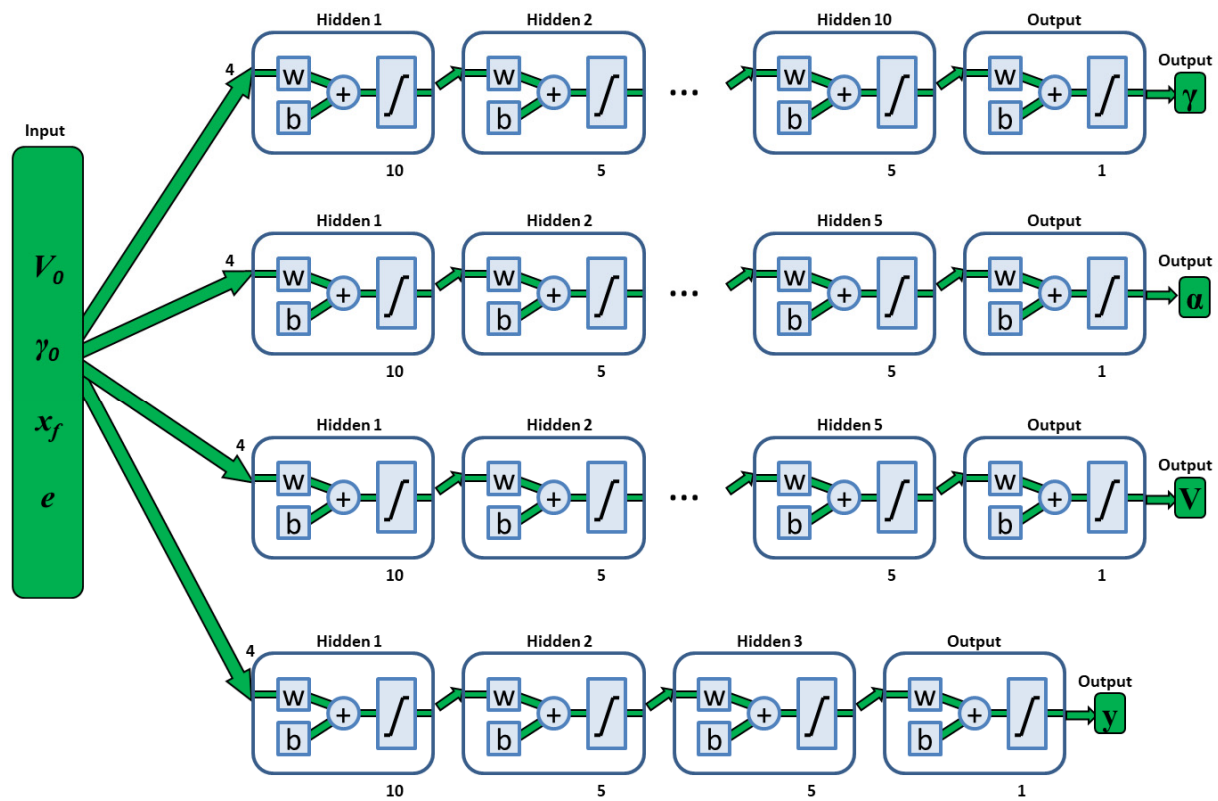


Figure 11. The structures of four deep neural networks.

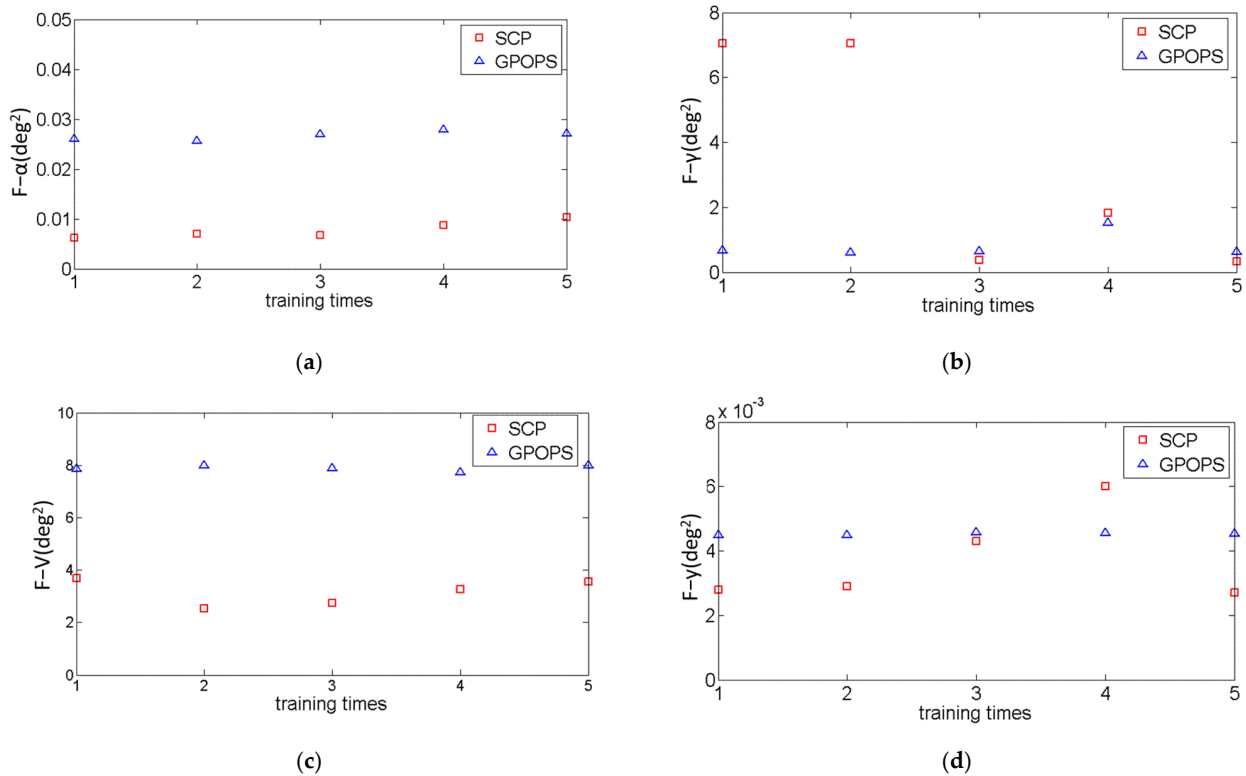


Figure 12. Training performance function  $F$  values of the four deep neural networks. (a) Performance function for angle of attack network training, (b) Performance function for flight path angle network training, (c) Performance function for velocity network training, (d) Performance function for altitude network training.

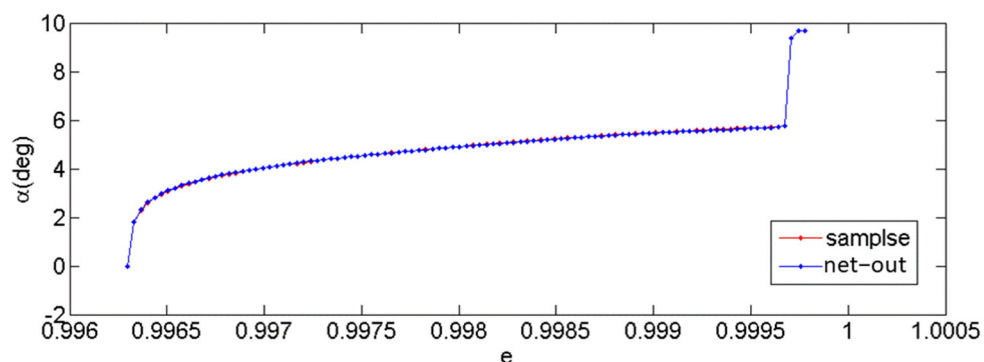
In Figure 12, the training samples used for each network were generated by the sequential convex optimization (SCO) method and the GPOPS software. From Figure 12, it can be observed that in all five trainings of the angle of attack network, the SCO samples outperformed the GPOPS samples. For two times of training, for the flight path angle network, the SCO samples performed better than the GPOPS samples. For all five times of training of the velocity network, the SCO samples show superiority to the GPOPS samples. For four times of training for the altitude network, the SCO samples performed better than the GPOPS samples.

Therefore, compared to the GPOPS software, using the reference trajectory samples generated by the SCO method for training the deep neural networks consistently resulted in smaller values of the performance function  $F$ . This indicates that the reference trajectory samples generated by SCO achieve better fitting performance.

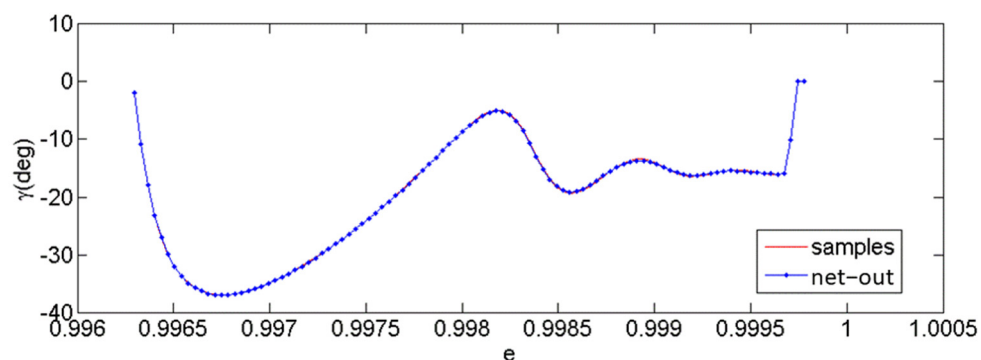
$$F = \frac{1}{N} \sum_{i=1}^N (\text{target}_i - \text{output}_i)^2 \quad (50)$$

### 5. Online Trajectory Generation

Online trajectory generation is performed using the deep neural networks trained with SCO samples. The angle of attack network, flight path angle network, velocity network, and altitude network are selected based on the smallest  $F$  values shown in Figure 12. The angle of attack network is selected from the first training, the flight path angle network is selected from the third training, the velocity network is selected from the second training, and the altitude network is selected from the first training. Figure 13a–d provide a comparison between the outputs of the deep neural networks and the sample outputs for a typical operating condition with  $e_0 = 0.996297$ ,  $V_0 = 300$ ,  $\gamma_0 = -2^\circ$ , and  $x_f = 55$  km. It can be observed that the designed multi-layer deep neural networks ensure better fitting for curves with complex shapes.

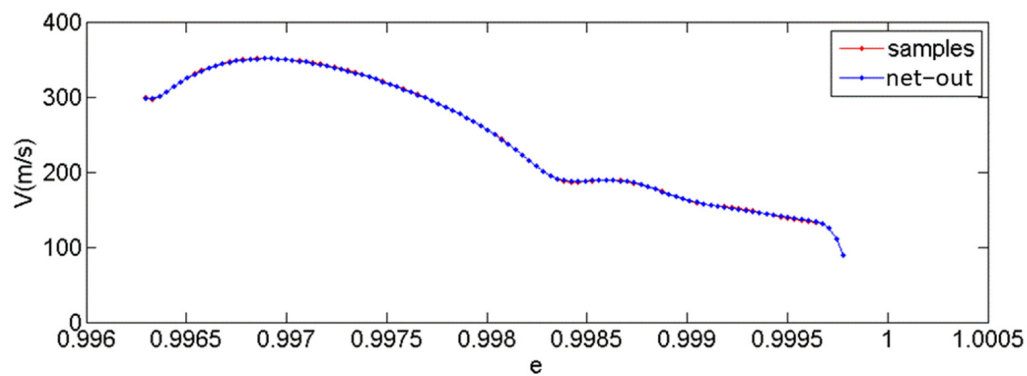


(a) Comparison between the outputs of the angle of attack network and the sample outputs

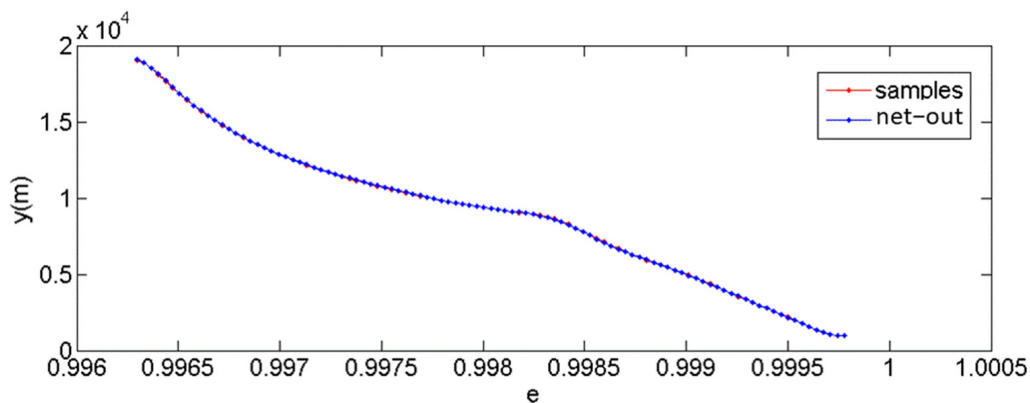


(b) Comparison between the outputs of the trajectory inclination network and the sample outputs

Figure 13. Cont.



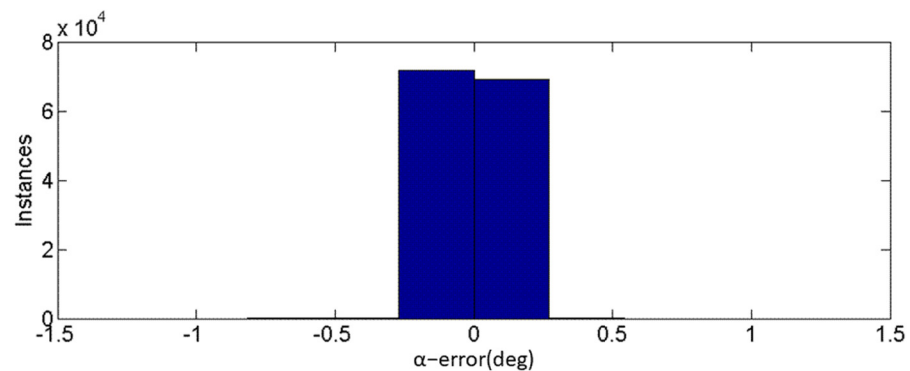
(c) Comparison between the outputs of the velocity network and the sample outputs



(d) Comparison between the outputs of the altitude network and the sample outputs

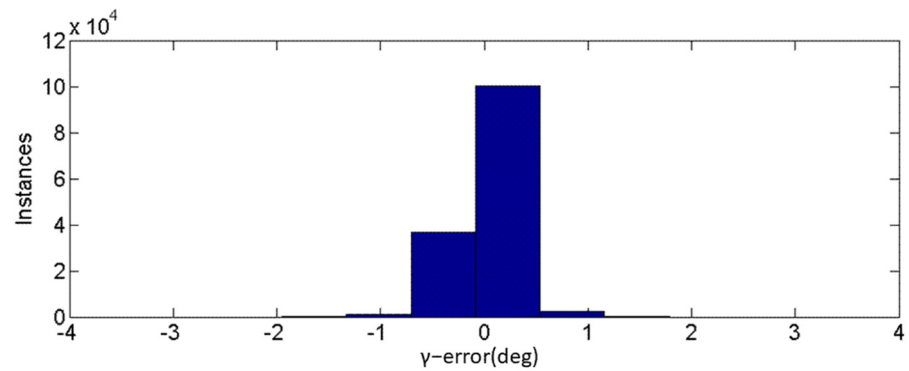
**Figure 13.** The comparison between the outputs of the deep neural network and the sample outputs.

Figure 14 shows the errors between the outputs of the four deep neural networks and the sample outputs under the same set of 141,804 sample inputs. From Figure 14, we can see that for each deep neural network, the majority of the outputs have relatively small errors. Approximately 99.9% of the angle of attack network output errors are kept within  $0.5^\circ$ ; 99.1% of the flight path angle network's output errors are within  $1^\circ$ ; 99.6% of the velocity network's output errors are within 5 m/s, and 99.6% of the altitude network's output errors are within 150 m. This indicates that the deep neural networks achieve higher fitting accuracy and have promising potentiality in engineering applications.

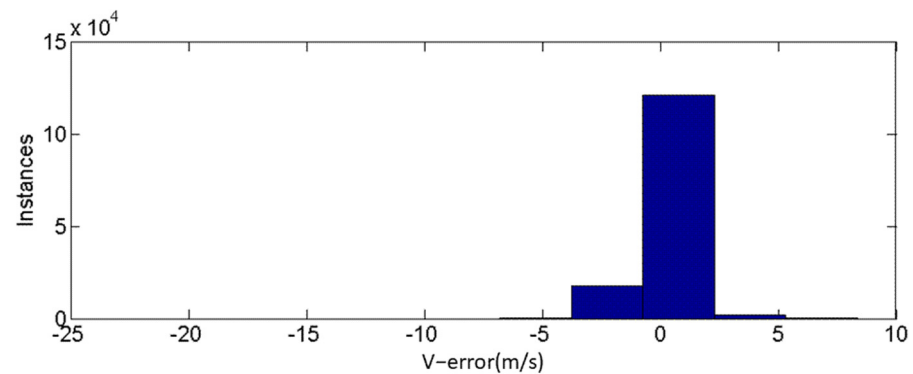


(a) The output errors of the angle of attack network

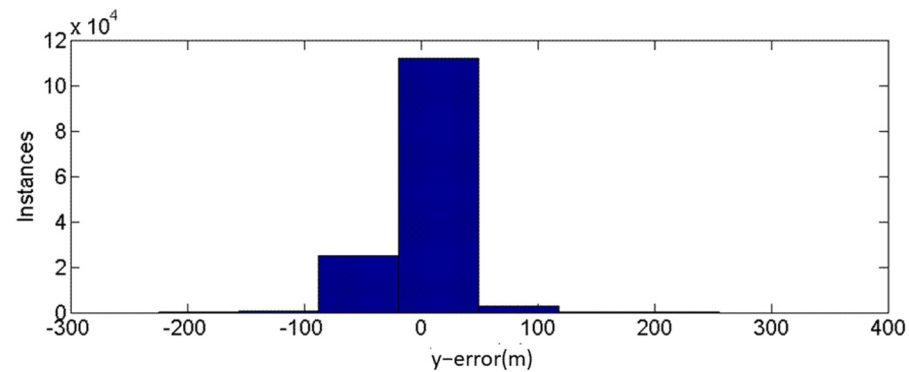
**Figure 14.** Cont.



(b) The output errors of the flight path angle network



(c) The output errors of the velocity network



(d) The output errors of the altitude network

**Figure 14.** Output errors of the four deep neural networks.

## 6. Summary

This paper presents an online trajectory generation framework for the terminal-area energy management of wave rider aircraft based on sequential convex optimization (SCO) and deep neural networks (DNNs). The sequential convex programming is used to generate massive training samples for the deep neural networks. The deep neural networks are constructed separately for each control and state variable to simplify the structure and improve fitting accuracy.

The superiority of sequential convex optimization (SCO) is demonstrated through comparative studies. Training samples for the deep neural networks are obtained using both the sequential convex programming (SCO) and nonlinear programming software GPOPS. The optimization criteria for both methods are designed to ensure that the terminal states of the trajectories meet the design values and to minimize oscillations in the control variable  $\alpha$ . Compared to GPOPS software, the trajectories obtained using sequential convex

optimization (SCO) provide smoother control variable  $\alpha$ , similar shapes among trajectories for the same state variable, and smaller solution time with higher computational efficiency. These characteristics make the trajectories obtained from sequential convex optimization suitable for training the deep neural networks.

A separate deep neural network is established for each control and state variable, including the flight path angle network, angle of attack network, velocity network, and altitude network. The number of hidden layers and neurons in each layer of the deep neural networks is designed based on the complexity of the trajectory curves to achieve higher fitting accuracy with minimal storage space usage. The calculation of output errors for the networks indicates that the generation method based on deep neural networks achieves high trajectory accuracy, with low storage space requirements, so it is more suitable for online applications in onboard computers.

**Author Contributions:** Conceptualization, Z.L.; Methodology, Z.L. and G.C.; Software, Z.L. and G.C.; Validation, J.Y. and B.A.; Formal Analysis, Y.F.; Investigation, Z.L.; Resources, K.L.; Data Curation, J.Q.; Writing—Original Draft Preparation, Z.L., G.C. and J.Q.; Writing—Review and Editing, J.Y., B.A., Y.F. and K.L.; Visualization, G.C.; Supervision, J.Y. and B.A.; Project Administration, K.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** All data used during the study appear in the submitted article.

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

## References

- Shen, H.-L.; Gong, Z. Methodology of Onboard Trajectory Design for Space Shuttle Terminal Area Energy Management Phase. *J. Astronaut.* **2008**, *29*, 430–433.
- Zhao, Z.-T.; Huang, W.; Yan, L.; Yang, Y.-G. An overview of research on widespeed range waverider configuration. *Prog. Aerosp. Sci.* **2020**, *113*, 100606. [\[CrossRef\]](#)
- Corraro, F.; Morani, G.; Nebula, F.; Cuciniello, G.; Palumbo, R. GN&C Technology Innovations for TAEM USV DTFT2 Mission Results. In Proceedings of the 17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, San Francisco, CA, USA, 11–14 April 2011.
- Corraro, F.; Cuciniello, G.; Morani, G.; Nebula, F.; Palumbo, R.; Vitale, A. Advanced GN&C Technologies for TAEM: Flight Test Results of the Italian Unmanned Space Vehicle. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, OR, USA, 8–11 August 2011.
- Moore, T.E. *Space Shuttle Entry Terminal Area Energy Management*; NASA Johnson Space Center: Houston, TX, USA, 1991.
- Sun, C.Z. Research on Terminal Area Energy Management and Autoland Technology for Reusable Launch Vehicle. Ph.D. Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, January 2008.
- Fang, G.C. Guidance Law Design of Terminal Area Energy Management for Reusable Launch Vehicle. Master Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, January 2013.
- Guo, T.; Xie, L. Research on Ship Trajectory Classification Based on a Deep Convolutional Neural Network. *J. Mar. Sci. Eng.* **2022**, *10*, 568. [\[CrossRef\]](#)
- Wang, J.; Wu, Y.; Liu, M.; Yang, M.; Liang, H. A Real-Time Trajectory Optimization Method for Hypersonic Vehicles Based on a Deep Neural Network. *Aerospace* **2022**, *9*, 188. [\[CrossRef\]](#)
- Hart, S.T.; Ayoubi, M.A.; Naseradinmousavi, P. Comparative Study of Pseudospectral Methods for Spacecraft Optimal Attitude Maneuvers. *J. Spacecr. Rocket.* **2022**, *59*, 178–189. [\[CrossRef\]](#)
- Haiping, R.A.O.; Zhong, R.; Pengjie, L.I. Fuel-optimal deorbit scheme of space debris using tethered space-tug based on pseudospectral method. *Chin. J. Aeronaut.* **2021**, *34*, 210–223.
- Zong, Q.; Li, Z.Y.; Ye, L.Q.; Tian, B. Variable trust region sequential convex programming for RLV online reentry trajectory reconstruction. *J. Harbin Inst. Technol.* **2020**, *52*, 147–155.
- An, K.; Guo, Z.-Y.; Xu, X.-P.; Huang, W. A framework of trajectory design and optimization for the hypersonic gliding vehicle. *Aerosp. Sci. Technol.* **2020**, *106*, 106110. [\[CrossRef\]](#)
- Liu, Z.; Lu, H.R.; Zheng, W.; Wen, G.; Wang, Y.; Zhou, X. Rapid time-coordination trajectory planning method for multi-glide vehicles. *Acta Aeronaut. Astronaut. Sin.* **2021**, *42*, 524479.
- Wei, J.-H.; Wang, Z.-K. Glide-Cruise Trajectory Optimization for Hypersonic Vehicles. *J. Command. Control.* **2021**, *7*, 249–256.
- Li, Z.; Zhou, X.; Zhang, H.; Tang, G. Analysis of Entry Footprint Based on Pseudospectral Method. *J. Shanghai Jiao Tong Univ.* **2022**, *56*, 1470–1478.



17. Ren, P.F.; Wang, H.B.; Zhou, G.F. Reentry trajectory optimization for hypersonic vehicle based on adaptive pseudospectral method. *J. Beijing Univ. Aeronaut. Astronaut.* **2019**, *45*, 2257–2265.
18. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
19. Wang, B. Implementation of Interior Point Methods for Second Order Conic Optimization. Ph.D. Thesis, McMaster University, Hamilton, ON, Canada, 2003.
20. Andersen, E.D.; Roos, C.; Terlaky, T. On implementing a primal-dual interior-point method for conic quadratic optimization. *Math. Program.* **2003**, *95*, 249–277. [\[CrossRef\]](#)
21. Andersen, E.D.; Andersen, K.D. *The MOSEK Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm*; High Performance Optimization; Springer US: New York, NY, USA, 1996; pp. 197–232.
22. Vandenberghe, L. The CVXOPT Linear and Quadratic Cone Program Solvers. 20 March 2010. Available online: <http://www.seas.ucla.edu/~vandenbe/publications/coneprog.pdf> (accessed on 3 January 2023).
23. Domahidi, A.; Chu, E.; Boyd, S. ECOS: An SOCP Solver for Embedded Systems. In Proceedings of the 2013 European Control Conference (ECC), Zürich, Switzerland, 17–19 July 2013.
24. Domahidi, A. Methods and Tools for Embedded Optimization and Control. Ph.D. Thesis, RWTH Aachen University, Aachen, Germany, 2013.
25. Liu, X.; Shen, Z.; Lu, P. Entry Trajectory Optimization by Second-Order Cone Programming. *J. Guid. Control. Dyn.* **2015**, *39*, 227–241. [\[CrossRef\]](#)
26. Liu, X.; Shen, Z.; Lu, P. Exact convex relaxation for optimal flight of aerodynamically controlled missiles. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 1881–1892. [\[CrossRef\]](#)
27. Mao, Y.; Szmuk, M.; Açıkmeşe, B. Successive Convexification of Non-Convex Optimal Control Problems and Its Convergence Properties. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016.
28. Liu, X.; Lu, P.; Pan, B. Survey of convex optimization for aerospace applications. *Astrodynamics* **2017**, *1*, 23–40. [\[CrossRef\]](#)
29. Szmuk, M.; Açıkmeşe, B. Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time. In Proceedings of the 2018 AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 8–12 January 2018.
30. Mao, Y.; Szmuk, M.; Açıkmeşe, B. Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems. *arXiv* **2018**, arXiv:1804.06539.
31. Szmuk, M.; Reynolds, T.P.; Açıkmeşe, B. Successive Convexification for Real-Time 6-Dof Powered Descent Guidance with State-Triggered Constraints. *arXiv* **2018**, arXiv:1811.10803.
32. Liu, X. Fuel-Optimal Rocket Landing with Aerodynamic Controls. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Grapevine, TX, USA, 9–13 January 2017.
33. Reynolds, T.P.; Szmuk, M.; Malyuta, D.; Mesbahi, M.; Açıkmeşe, B. Dual Quaternion Based Powered Descent Guidance with State-Triggered Constraints. *J. Guid. Control. Dyn.* **2019**, *43*, 1584–1599. [\[CrossRef\]](#)
34. Wang, Z.; Grant, M.J. Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization. In Proceedings of the AIAA SciTech Forum, San Diego, CA, USA, 7–11 January 2019.
35. Han, H.; Qiao, D.; Chen, H.; Li, X. Rapid planning for aerocapture trajectory via convex optimization. *Aerosp. Sci. Technol.* **2019**, *84*, 763–775. [\[CrossRef\]](#)
36. Pei, P.; Wang, J. Near-Optimal Guidance with Impact Angle and Velocity Constraints Using Sequential Convex Programming. *Math. Probl. Eng.* **2019**, 2019, 2065730. [\[CrossRef\]](#)
37. Malyuta, D.; Reynolds, T.P.; Szmuk, M.; Açıkmeşe, B. Fast Trajectory Optimization via Successive Convexification for Spacecraft Rendezvous with Integer Constraints. *arXiv* **2019**, arXiv:1906.04857.
38. Chen, B.; Liu, C.; Ji, C. Waverider design and analysis based on shock-fitting method. *ACTA Aerodyn. Sin.* **2017**, *35*, 421–428.
39. Liu, C.; Bai, P.; Chen, B.; Ji, C. Rapid Design and Multi-Object Optimization for Waverider from 3D Flow. *J. Astronaut.* **2016**, *29*, 535–543.
40. Li, D.; Ji, C.; Ma, H. Three dimensions unstructured Cartesian grid for Navier-Stokes equation. *ACTA Aerodyn. Sin.* **2006**, *24*, 6.

**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.