IEEE Transactions on Neural Networks and Learning Systems, Volume 31, Issue 11, November 2020, pp. 5005-5013
DOI: 10.1109/TNNLS.2019.2955400

Six-DOF Spacecraft Optimal Trajectory Planning and Real-time Attitude Control: A Deep Neural Network-based Approach

Runqi Chai, *Member, IEEE,* Antonios Tsourdos, *Member, IEEE,* Al Savvaris, *Member, IEEE,* Senchun Chai, *Senior Member, IEEE,* Yuanqing Xia, *Senior Member, IEEE,* and C. L. Philip Chen, *Fellow, IEEE,*

***Abstract*—This paper presents an integrated trajectory planning and attitude control framework for six-degree-of-freedom (6-DOF) hypersonic vehicle (HV) reentry flight. The proposed framework utilizes a bilevel structure incorporating desensitized trajectory optimization and deep neural network (DNN)-based control. In the upper level, a trajectory dataset containing optimal system control and state trajectories is generated, while in the lower-level control system, DNNs are constructed and trained using the pregenerated trajectory ensemble in order to represent the functional relationship between the optimized system states and controls. These well-trained networks are then used to produce optimal feedback actions online. A detailed simulation analysis was performed to validate the real-time applicability and the optimality of the designed bilevel framework. Moreover, comparative analysis was also carried out between the proposed DNN-driven controller and other optimization-based techniques existing in related works. Our results verify the reliability of using the proposed bilevel design for the control of HV reentry flight in real time.**

***Index Terms*—Trajectory planning, attitude control, 6-DOF hypersonic vehicle, bilevel structure, deep neural network.**

I. Introduction

THE design of guidance and control systems for HVs has been extensively investigated over the last couple of decades due to its increasing importance in a wide variety of real-world applications [1]–[3]. In this work, a particular focus in industry is the development of advanced guidance and control systems for reentry flight [4]–[7]. The reentry mission scenario contains both atmospheric and exoatmospheric flight phases [8]. Consequently, an almost immediate change in the aerodynamic environment must be considered, thus making reentry design a challenging task. In addition to the complex flying environment, other theoretical or practical challenges may also exist, such as the inherent nonlinearities of the system model, disturbances in the initial entry conditions, and high demands of online computation [9]. As a result, it is still difficult to design a reentry guidance and control system that can offer a reliable solution in real time.

A reentry guidance and control system is usually divided into two subsystems, namely, an trajectory control system and an attitude control system [10]. Trajectory control is often fulfilled by carrying out two major steps: trajectory optimization [11] and trajectory tracking guidance [12]. Since a well-planned trajectory is a key for enhanced trajectory control of the HV, a large amount of research on developing trajectory optimization methods can be found [13], [14]. For example, in [13] the authors developed a bioinspired optimization strategy to calculate the crossrange optimal trajectories for hypersonic glide vehicles. In addition, the authors in [14] designed a unified multiobjective optimization technique in order to optimize the reentry trajectory with the consideration of multiple performance indices. Although a large amount of results were provided in the aforementioned works to test and validate the effectiveness of different trajectory optimization methods, the dynamics of the HV were mainly modeled as a point mass. Few attempts have been made to find optimal flight trajectories for six-degree-of-freedom (6-DOF) HVs. It should be highlighted that attitude rates and angles might have non-negligible impacts on the HV motion in real flight scenarios. Algorithms developed using a 3-DOF model may not fully exploit the relationships between the rotational and translational dynamics. As a result, the accuracy of the simulated state and control trajectories might be degraded.

Apart from the optimal flight path generation, the design of advanced trajectory or attitude control systems has also been extensively investigated [15]–[17]. For instance, in [15] a nonsingular fast terminal SMC controller equipped with a finite-time disturbance observer was designed to fulfill the tracking control problem by nullifying both the disturbance observation and trajectory errors. In the research reported by Xiao et al. [18], a sliding mode control (SMC)-based attitude tracking controller was developed. One unique feature of this design is that it has a strong fault-tolerance capability with respect to different types of actuator faults. Similarly, a reliable fuzzy tracking attitude control scheme was constructed in [19], wherein stochastic actuator failures were considered and modeled as a Markov chain.

Other effective algorithms have also been designed, such as that by Zong et al. [20], [21], who proposed specific disturbance attenuation and rejection-based methods. These methods aim to asymptotically stabilize the system under certain conditions and have been successfully applied to attitude tracking control of spacecraft [22].

Nevertheless, the core idea of most trajectory or attitude control designs is to apply the estimation technique as a compensator to allow adjustment in advanced control methods (e.g., the SMC-based control [23], the fault-tolerant control [24], [25], or the output-feedback control [26]). Using these strategies, a robust control performance can be acquired. However, less attention has been paid to the optimality of the tracked reference signal or the derived control law.

To optimally produce the control command, the possibility of applying model predictive control (MPC) has been investigated. For instance, a heuristic approach-based MPC was developed by Tang et al. [27]. Then this controller was applied to steer the flight of a reduced-order HV. Similarly, in [28] a robust MPC method was proposed to control a longitudinal model of the HV in the presence of control input delays. The suitability of this approach was also

R. Chai, A. Tsourdos and A. Savvaris are with the School of Aerospace, Transport and Manufacturing, Cranfield University, UK, e-mail: (r.chai@cranfield.ac.uk), (a.tsourdos@cranfield.ac.uk), and (a.savvaris@cranfield.ac.uk).

S. Chai and Y. Xia are with the school of Automation, Beijing Institute of Technology, Beijing, China, e-mail: (chaisc97@163.com), (xia_yuanqing@bit.edu.cn).

C. L. P. Chen is with the Faculty of Science and Technology, University of Macau, Macau 999078, China, and also with the Department of Navigation, Dalian Maritime University, Dalian 116026, China, and with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China, e-mail: (philip.chen@ieee.org).

validated by conducting a number of simulations as reported in the original paper. However, a critical problem of applying MPC-based approaches is that the consideration of the full system model and multiple path constraints tends to result in a large computational burden on the optimization process, which cannot be afforded in real time.

Recently, growing interest in researching integrated guidance and control (IGC) schemes for HVs has been witnessed. A large amount of research on designing or implementing an IGC method for HVs has been reported in the literature [29], [30]. Among these works, the investigation carried out by Tian et al. [30] is of particular importance. Specifically, the IGC system designed in [30] produced the guidance command via real-time trajectory replanning. Subsequently, the optimized guidance command was offered to the inner loop, where a bilevel SMC controller was implemented to compute the attitude angle profiles. However, two main drawbacks may exist. First, the rotational effects are not considered when the guidance command is produced. Moreover, the SMC-based attitude controller does not take into account the optimality of the state feedback. Although there are still some open issues in applying the IGC system, it was illustrated that the IGC scheme can have the potential to be utilized in near-real time. Consequently, attention is given to the IGC design in the present study.

In this paper, we design and suggest a new reentry IGC framework that can effectively tackle the issues faced by existing methods. Compared with the results of traditional methods, one advantage of applying the proposed control system is that well-researched trajectory optimization and deep learning approaches are fused together to build an integrated framework, thereby allowing additional DOFs and more system state information available. In addition, the proposed control algorithm is able to produce optimal feedback actions online under the consideration of large initial state perturbations. The contributions of this article are summarized as follows.

1) A 6-DOF version of the reentry trajectory optimization problem is formulated by extending the traditional 3-DOF model to consider rotational effects.
2) A desensitized trajectory optimization algorithm is iteratively utilized to address the optimal 6-DOF trajectory design problem with the consideration of noise-perturbed initial states, thereby producing a large optimal trajectory dataset.
3) A simple yet effective DNN-based online feedback action producer is designed by training DNNs on the optimal trajectory dataset to learn the inherent relation with respect to the state-control actions.

It is important to remark that some preliminary results were reported in [31]. Compared to this preliminary version, this work presents detailed problem formulation, methodology design and additional experimental results, thus making the paper more comprehensive. In addition, this paper can also be treated as an extension work in order to address some remaining problems in [31]. In particular, a desensitized optimization strategy is embedded in the original algorithm framework to enhance the robustness of the solution-finding process, and a systematic strategy is suggested to select the network structural parameters such that the network prediction ability can be improved.

The remainder of this article is organized as follows. Sec II gives the mathematical formulation of the considered 6-DOF reentry trajectory optimization mission. In Sec III, a desensitized trajectory design algorithm applied to form the training dataset, along with the DNN-based real-time feedback action producer, is outlined. Numerical and semiphysical studies testing the control performance and the real-time implementation of the proposed strategy are demonstrated in Sec IV. Finally, concluding remarks are drawn in Sec V.

II. Six-DOF HV Reentry Trajectory Optimization

In this section, the 6-DOF HV reentry trajectory optimization model considered in this research is outlined. Specifically, Sec II.A presents the nonlinear translational and rotational dynamics of the reentry vehicle. Subsequently, Sec II.B defines various system-related constraints. According to the constructed system model and mission constraints, a 6-DOF reentry trajectory planning formulation with accumulated heating load minimization is presented in Sec II.C.

A. Translational and Rotational Equations of Motion

Typical 3-DOF reentry trajectory planning models consider the vehicle as a point mass only. In this work, alternatively, we take into account the coupling between the translational and rotational equations of motion (EOMs). More precisely, the translational EOMs of the HV can be described by Eq. (1), whereas the rotational EOMs of the HV are formulated by Eq. (2).

$$
\begin{aligned}
\dot{r} &= V\sin\gamma \\
\dot{\theta} &= \tfrac{V\cos\gamma\sin\psi}{r\cos\phi} \\
\dot{\phi} &= \tfrac{V\cos\gamma\cos\psi}{r} \\
\dot{V} &= -\tfrac{D}{m} - g\sin\gamma \\
&\quad +\Omega^2 r\cos\phi(\cos\phi\sin\gamma - \cos\gamma\sin\psi\cos\psi) \\
\dot{\gamma} &= \tfrac{L\cos\sigma}{mV} + (\tfrac{V^2-gr}{rV})\cos\gamma + 2\Omega\sin\psi\cos\phi \\
&\quad +\Omega^2 r\cos\phi(\cos\phi\cos\gamma + \sin\phi\cos\psi\sin\gamma) \\
\dot{\psi} &= \tfrac{L\sin\sigma}{mV\cos\gamma} + \tfrac{V}{r}\cos\gamma\sin\psi\tan\phi \\
&\quad +\tfrac{\Omega^2 r\cos\phi\sin\phi}{\cos\gamma} - 2\Omega(\tan\gamma\cos\psi\cos\phi - \sin\phi)
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
\dot{\alpha} &= q - p\cos\alpha\tan\beta - \nu\sin\alpha\tan\beta + \tfrac{\sin\sigma}{\cos\beta}(\cos\gamma\dot{\psi} - \\
&\quad \dot{\phi}\sin\gamma\sin\psi + (\Omega+\dot{\theta})(\sin\gamma\cos\psi\cos\phi - \cos\gamma\sin\phi)) \\
&\quad -\tfrac{\cos\sigma}{\cos\beta}(\dot{\gamma} - \dot{\phi}\cos\psi - (\dot{\theta}+\Omega)\cos\phi\sin\psi) \\
\dot{\sigma} &= -p\cos\beta\cos\alpha - q\sin\beta + \dot{\alpha}\sin\beta - \nu\cos\beta\sin\alpha \\
&\quad -\dot{\phi}\cos\gamma\sin\psi - \dot{\psi}\sin\gamma \\
&\quad +(\Omega+\dot{\theta})(\sin\gamma\sin\phi + \cos\gamma\cos\phi\cos\psi) \\
\dot{\beta} &= -\nu\cos\alpha + p\sin\alpha + \sin\sigma(\dot{\gamma} - \dot{\phi}\cos\psi + (\Omega+\dot{\theta})\sin\psi \\
&\quad \cos\phi) + \cos\sigma(-\dot{\phi}\sin\gamma\sin\psi + \dot{\psi}\cos\gamma - (\Omega+\dot{\theta}) \\
&\quad (\sin\gamma\cos\psi\cos\phi - \cos\gamma\sin\phi)) \\
\dot{p} &= \tfrac{M_x I_{zz}}{I_{xx}I_{zz}-I_{xz}^2} + \tfrac{M_z I_{xz}}{I_{xx}I_{zz}-I_{xz}^2} \\
&\quad + \tfrac{pq(I_{xx}+I_{zz}-I_{yy})I_{xz}}{I_{xx}I_{zz}-I_{xz}^2} + \tfrac{q\nu((I_{yy}-I_{zz})I_{zz}-I_{xz})}{I_{xx}I_{zz}-I_{xz}^2} \\
\dot{q} &= \tfrac{I_{xz}}{I_{yy}}(\nu^2 - p^2) + \tfrac{M_y}{I_{yy}} + \tfrac{p\nu(I_{zz}-I_{xx})}{I_{yy}} \\
\dot{\nu} &= \tfrac{M_x I_{xz}}{I_{xx}I_{zz}-I_{xz}^2} + \tfrac{M_z I_{xx}}{I_{xx}I_{zz}-I_{xz}^2} \\
&\quad + \tfrac{pq((I_{xx}-I_{yy})I_{xx}+I_{xz}^2)}{I_{xx}I_{zz}-I_{xz}^2} + \tfrac{q\nu(I_{yy}-I_{xx}-I_{zz})I_{xz}}{I_{xx}I_{zz}-I_{xz}^2}
\end{aligned}
\tag{2}
$$

The definitions of variables appearing in Eq. (1) and Eq. (2) are tabulated in Table I.

TABLE I: Notations for variables

r, V:	Radial distance and velocity
θ, ϕ:	Longitude and latitude
γ, ψ:	Flight path angle (FPA) and heading angle
α, β, σ:	Angle of attack (AOA), sideslip angle and bank angle
p, q, ν:	Angular rates for roll, pitch, and yaw angles
I_{ij} (i, j=x, y, z):	Moment of inertia
M_x, M_y, M_z:	Roll, pitch, and yaw moments
L, D:	Lift and drag forces
m, Ω:	Mass of HV and earth's rotation rate
g, ρ:	The gravity and density of the atmosphere
x, u, t	The state, control, and time variables
t_0, t_f	The initial and terminal time

In Table I, D is calculated via $D = \frac{1}{2}\rho V^2 S C_D$, where S and C_D represent the reference area of the HV and drag coefficient, respectively. $L = (C_L/C_D)D$, where C_L is the lift coefficient.

$\rho = \rho_0 \exp{(r - R_e)/h_s}$, where R_e and h_s are respectively the radius of the earth and the density scale height. The states x and controls u are abbreviated as

$$\begin{aligned} x &= [r, \theta, \phi, V, \gamma, \psi, \alpha, \sigma, \beta, p, q, \nu] \\ u &= [M_x, M_y, M_z] \end{aligned}$$

As a result, the system dynamics can be written as $\dot{x} = f(x, u)$.

B. System-related Constraints

Some state/control-dependent constraints are required to be considered during the reentry flight. For instance, to successfully complete the mission, boundary constraints are imposed on the initial and terminal state values. These requirements can be modeled as:

$$\begin{array}{lll} r(t_0) = r_0 & \theta(t_0) = \theta_0 & \phi(t_0) = \phi_0 \\ V(t_0) = V_0 & \gamma(t_0) = \gamma_0 & \psi(t_0) = \psi_0 \\ \alpha(t_0) = \alpha_0 & \sigma(t_0) = \sigma_0 & \beta(t_0) = \beta_0 \\ p(t_0) = p_0 & q(t_0) = q_0 & \nu(t_0) = \nu_0 \\ r(t_f) = r_f & V(t_f) = V_f & \gamma(t_f) = \gamma_f \\ p(t_f) = p_f & q(t_f) = q_f & \nu(t_f) = \nu_f \end{array} \tag{3}$$

where the initial time is set to $t_0 = 0$. Apart from the boundary limitations, path constraints are also imposed on the control variables such that they could vary in desirable regions:

$$\begin{cases} M_x^{\min} \leq M_x \leq M_x^{\max} \\ M_y^{\min} \leq M_y \leq M_y^{\max} \\ M_z^{\min} \leq M_z \leq M_z^{\max} \end{cases} \tag{4}$$

For safety reasons, the flight trajectory is restricted to a narrow corridor such that three important safety factors (i.e., the aerodynamic heat, dynamic pressure and normal load) cannot violate their allowable ranges. To describe these requirements, three state path constraints are constructed:

$$\begin{bmatrix} \dot{Q}(x) \\ P_d(x) \\ n_L(x) \end{bmatrix} = \begin{bmatrix} K_Q \rho^{0.5} V^3 \\ \frac{1}{2}\rho V^2 \\ \frac{\sqrt{L^2+D^2}}{mg} \end{bmatrix} \leq \begin{bmatrix} \dot{Q}^{max} \\ P^{max} \\ N^{max} \end{bmatrix} \tag{5}$$

C. Overall Optimization Formulation

For the considered problem, the objective function applied for analysis is minimizing the accumulated aerodynamic heating. This can be modeled as:

$$J = \min \int_0^{t_f} \dot{Q}(x) dt \tag{6}$$

Based on the objective and constraints constructed earlier, the overall optimization formulation for the considered mission is established below:

$$\begin{array}{lll} \text{minimize} & \text{Eq.}(6) & \text{(mission objective)} \\ \text{subject to} & \text{Eq.}(1) & \text{(translational EOMs)} \\ & \text{Eq.}(2) & \text{(rotational EOMs)} \\ & \text{Eq.}(3) & \text{(boundary conditions)} \\ & \text{Eq.}(4) & \text{(control limitations)} \\ & \text{Eq.}(5) & \text{(state limitations)} \end{array} \tag{7}$$

III. Bilevel Trajectory and Attitude Control Method

In this paper, we aim to design an IGC method that can steer the HV to fly along the simulated optimal path in real time. Motivated by related works, a bilevel control structure is designed. An advantage of utilizing the designed control scheme is that the structure of the optimized control signal can be maximally preserved, while the recursive online replanning process is no longer necessary. The structure of the proposed framework, along with some key application aspects, is outlined in the subsequent subsections.

A. Offline Trajectory Dataset Construction

In the upper level, a set of optimized trajectories for the HV reentry flight are generated to form a trajectory ensemble. Suppose that x_0 (the initial condition) is perturbed by the stochastic parameters ξ_x. The stochastic parameters are sampled via $\{\xi^{(k)_x}\}_{k=1}^N \sim R(\xi_x)$, where $R(\cdot)$ denotes the probability density function of ξ_x. Subsequently, the trajectory planning model (7) with noise-perturbed x_0 is written as:

$$\begin{array}{ll} \text{minimize} & J^{(k)} \\ \text{subject to} & \dot{x}^{(k)} = f(x^{(k)}, u^{(k)}) \\ & x_0^{(k)} = x_0 + \xi_x^{(k)} \\ & \Phi(x_f^{(k)}) = 0 \\ & h(x^{(k)}, u^{(k)}) = 0 \\ & g(x^{(k)}, u^{(k)}) \leq 0 \end{array} \tag{8}$$

in which $h(\cdot,\cdot)$ and $g(\cdot,\cdot)$ are respectively the compressed forms of equality and inequality constraints defined in Sec II.B. $\Phi(\cdot)$ is the compressed form of the terminal conditions given by Eq. (3). Consequently, the optimal solution set $\{(x^{(k)}, u^{(k)})\}_{k=1}^N$ can be obtained by iteratively solving Eq. (8) until k reaches N. Problem (8) can be solved via standard optimal control methods such as the direct collocation method [9], the multiple shooting method [9], or the pseudospectral method [11]. These methods all apply a "discretization + optimization" mode to address the problem. More precisely, $\{t_i\}_{i=1}^{N_c}$ temporal nodes are utilized to parameterize the continuous-time optimization model (8). Subsequently, nonlinear programming (NLP) algorithms are applied to optimize the static optimization model.

1) Initialization: Note that one critical process that could have an impact on the convergence performance of the NLP algorithms is the assignment of initial guess values. Based on the problem formulation, it is obvious that the high nonlinearity and coupling of the 6-DOF dynamic model might significantly increase the sensitivity of the solution with respect to the initial guess values. Moreover, the consideration of various path constraints might further restrict the feasible region, thus increasing the computational burden on the optimization algorithm.

To overcome these difficulties, an initial guess generator as suggested in [32] is applied. This generator is based on an improved particle swarm optimization (IPSO) algorithm and is used mainly to desensitize the effects caused by active mission constraints and poor initial guess values. Compared with the results of the original PSO, the local exploitation ability of the IPSO is further improved by using a local gradient operation. For completeness reasons, a brief description including the implementation steps of the algorithm is recalled as follows [32]:

Step 1. Initialize a set of particles with a position vector $u_z(G) = [u_{z,1}(G), ..., u_{z,d}(G)]$ and a velocity vector $v_z(G) = [v_{z,1}(G), ..., v_{z,d}(G)]$. Here, z and d are the index and dimension of the particle, respectively, while G denotes the index of generation.

Step 2. Compute the fitness value for the particle via:

$$J_z(G) = \begin{cases} J_z(G), & \text{if } Vo_z(G) = 0; \\ \bar{J}(G) + \bar{J}(G) Vo_z(G), & \text{if } Vo_z(G) > 0. \end{cases}$$

where $\bar{J}(G)$ denotes the worst fitness value among the generation, whereas $V_o \in [0, 1]$ represents the magnitude of constraint violation. The calculation approach can be found in [8].

Step 3. Perform a local gradient operation to update u_z via:

$$\begin{aligned} \bar{u}_z(G) &= u_z(G) + e_z \\ e_z &= -\left(\tfrac{\nabla_u J_z}{\|\nabla_u J_z\|} + \tfrac{\nabla_u Vo_z}{\|\nabla_u Vo_z\|}\right) \end{aligned}$$

where $\nabla.$ is the partial differentiation operation.

Step 4. Search the global best position $g(G)$ and the best position of the zth particle $p_z(G)$.

Step 5. Update the position and velocity vectors via:

$$\begin{aligned} v_{z,d}(G+1) = &\quad w v_{z,d}(G) + c_1(p_{z,d}(G) - u_{z,d}(G)) \\ & \quad c_2(g(G) - u_{z,d}(G)) \\ u_{j,d}(G+1) = &\quad u_{z,d}(G) + v_{z,d}(G+1) \end{aligned}$$

where c_1 and c_2 are two uniformly distributed random numbers [9], [32].

Step 6. Set $G = G + 1$, and go back to Step 2. Terminate the process when G reaches G_{max}.

Step 7. Output the best solution among the G_{max}th generation, and use this solution as the initial reference solution.

2) Computational Consideration: According to experimental tests shown in previously published works [30], another important process that could influence the algorithm convergence ability is scaling. Suitable scaling can increase the convergence rate, convergence speed and algorithm robustness. To desensitize the negative effect caused by poor scaling, a nondimensionalization process should be performed by regulating the system variables via:

$$\begin{array}{llll} \bar{r} = r/r_0 & \bar{\theta} = \theta/\pi & \bar{\phi} = \phi/\pi & \bar{V} = V/V_0 \\ \bar{\gamma} = \gamma/\pi & \bar{\psi} = \psi/\pi & \bar{\alpha} = \alpha/\pi & \bar{\sigma} = \sigma/\pi \\ \bar{\beta} = \beta/\pi & \bar{p} = p/\pi & \bar{q} = q/\pi & \bar{\nu} = \nu/\pi \end{array} \tag{9}$$

The generated initial guess value and regulated variables are then provided to a newly proposed NLP solver that addresses the static optimization problem via a dual-loop improved gradient method [33]. It was shown in [33] and [34] that this method can be effective for solving this type of problem and can significantly improve the success rate of finding optimal solutions. As a result, the optimizer developed in [33] is well suited to solve the considered problem, and we iteratively use it to create the optimal trajectory ensemble.

B. DNN-based Feedback Action Producer

Following the discussion stated in Sec III.A, we assumed that the optimized HV reentry trajectory dataset has been generated. To effectively control the vehicle in real time, the core idea is to build multiple DNNs on the preconstructed trajectory ensemble in order to learn and represent the optimal state-control actions. Then, the trained DNNs is used as the optimal feedback controller in real time. That is, $u^{(k)} = \mathscr{N}(x^{(k)})$.

Generally speaking, a DNN consists of three components: an input layer, multiple hidden layers, and an output layer (as demonstrated in Fig. 1). Typically, each layer contains multiple function units. Their outputs are written as:

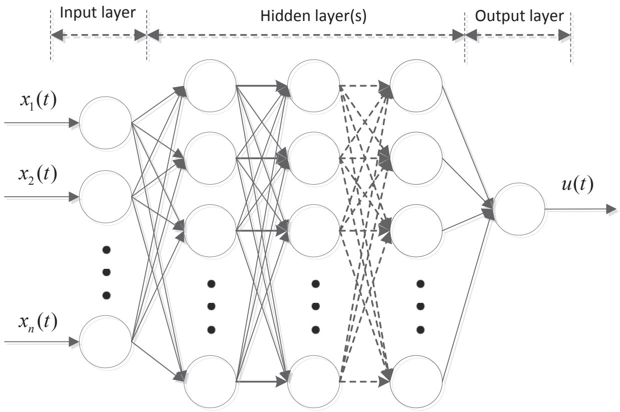

Fig. 1: DNN illustration

$$\begin{cases} o_{ij} = c_j(\omega_{ij} o_{j-1} + b_{ij}) & \text{(10a)} \\ o_{j-1} = \sum_{i=1}^{N_o^{j-1}} c_{j-1}(\omega_{i,j-1} o_{i,j-1} + b_{i,j-1}) & \text{(10b)} \end{cases}$$

In Eq. (10), $i = 1, 2, ..., N_o$ and $j = 1, 2, ..., N_L$ respectively represent the indices of the unit and layer. ω_{ij} and b_{ij} are the weight and bias parameters, respectively. c_j stands for the activation function of the jth layer, which is written as:

$$\begin{cases} c_j = \max(x^{(k)}, 0) & \text{(11a)} \\ c_{N_L} = \max(\bar{u}, \min(x^{(k)}), \underline{u}) & \text{(11b)} \end{cases}$$

in which $\bar{u}$ and $\underline{u}$ are respectively the upper and lower bounds of the controls. In the constructed DNN, ReLU activation functions are applied in hidden layers, while the linear bounded function is utilized in the output layer. The motivation for the use of a linear bounded function relies on its capability in accommodating cases when a bang-bang optimal control structure appears. Once a DNN is structured, the stochastic gradient descent method is used to train the network and adjust $[\omega, b]$ such that the mapping accuracy can be increased. More precisely, a loss function in the form of (12) is used in the training process.

$$E = \frac{1}{N_b} \sum_{i=1}^{N_b} (D(x_i) - y(x_i))^2 \tag{12}$$

where N_b, $D(\cdot)$, and $y(x_i)$ denote the batch size, the final output of the network, and the target output values, respectively. Subsequently, the weight parameters can be updated via:

$$\omega_i = \omega_i + \Delta\omega_i \tag{13}$$

where $\Delta\omega_i$ is given by:

$$\Delta\omega_i = -\frac{\zeta}{N_b} \frac{\partial \sum_{i=1}^{N_b} (D(x_i) - y(x_i))^2}{\partial \omega_i} \tag{14}$$

C. Overall Algorithm Framework

A graphical illustration of the overall framework of the proposed bilevel control strategy is shown in Fig. 2. To further clarify the online control design, the lower-level DNN-based control scheme illustrated in Fig. 2 is extracted and depicted in Fig. 3.

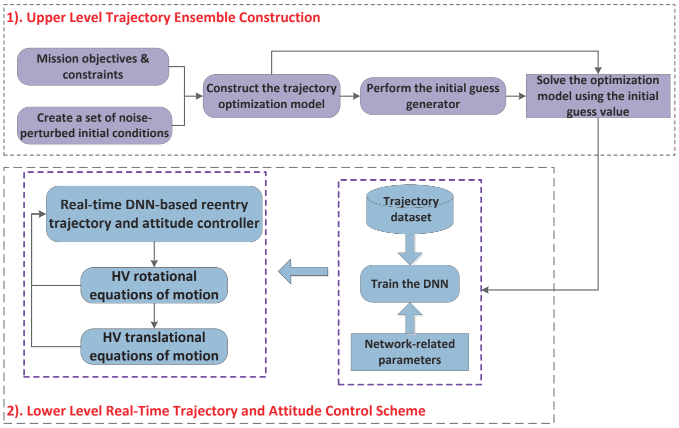
1). Upper Level Trajectory Ensemble Construction
Mission objectives & constraints
Create a set of noise-perturbed initial conditions
Construct the trajectory optimization model
Perform the initial guess generator
Solve the optimization model using the initial guess value
Real-time DNN-based reentry trajectory and attitude controller
HV rotational equations of motion
HV translational equations of motion
Trajectory dataset
Train the DNN
Network-related parameters
2). Lower Level Real-Time Trajectory and Attitude Control Scheme

Fig. 2: Graphical illustration of the bilevel strategy

IV. Numerical Results

Simulation results and experimental studies of applying the bilevel trajectory and attitude control strategy to address the considered problem are presented in this section. Several goals and objectives for carrying out the experiments are listed below:

- Validating the effectiveness and reliability of using the DNN-based feedback action producer to steer the flight of the HV.
- Comparing the control performance between existing control methods and the proposed strategy.

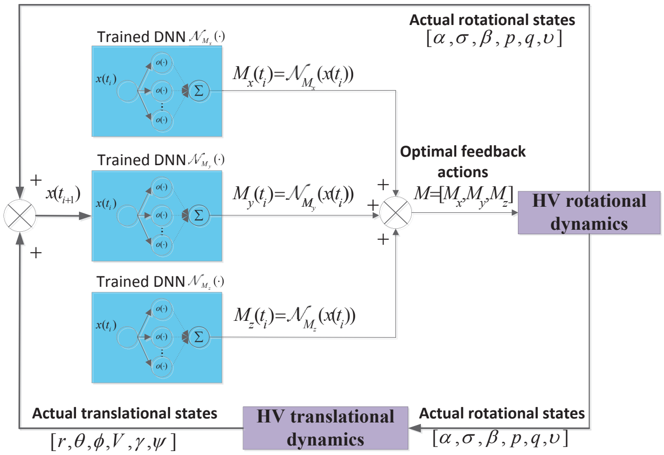
Actual rotational states
$[\alpha,\sigma,\beta,p,q,\upsilon]$
Trained DNN $\mathcal{N}_{M_x}(\cdot)$
$x(t_i)$
$o(\cdot)$
Σ
$M_x(t_i)=\mathcal{N}_{M_x}(x(t_i))$
Trained DNN $\mathcal{N}_{M_y}(\cdot)$
$M_y(t_i)=\mathcal{N}_{M_y}(x(t_i))$
Trained DNN $\mathcal{N}_{M_z}(\cdot)$
$M_z(t_i)=\mathcal{N}_{M_z}(x(t_i))$
Optimal feedback actions
$M=[M_x,M_y,M_z]$
HV rotational dynamics
$x(t_{i+1})$
+
+
Actual translational states
$[r,\theta,\phi,V,\gamma,\psi]$
HV translational dynamics
Actual rotational states
$[\alpha,\sigma,\beta,p,q,\upsilon]$

Fig. 3: Overall framework of the DNN-based control scheme

- Studying the real-time performance of the proposed control scheme.

Before the obtained results are presented in detail, some environment/vehicle-related parameters are first specified.

A. Scenario/Vehicle-related Parameter Assignment

The values of mission-related parameters are tabulated in Table II, while the vehicle-related parameters are specified in Table III. The state path constraints are assigned as $[Q^{max}, P^{max}, N^{max}] = [200, 280, 2.5]$, whereas the control constraints $[M_x, M_y, M_z$ are limited in $[-10^5, 10^5]$. ζ is set as 0.001. The numerical studies were performed under Windows 10 and an Intel (R) i7-3520M CPU running at 2.90 GHz with 8.00 GB RAM.

TABLE II: Assignment of mission-related parameters

Parameter	Value	Parameter	Value
$r(t_0)$,	21162900	$q(t_0)$,	$0^\circ/s$
$\theta(t_0)$,	0°	$\nu(t_0)$,	$0^\circ/s$
$\phi(t_0)$,	0°	r_f,	21066900
$V(t_0)$,	25600 ft/s	V_f,	16400 ft/s
$\gamma(t_0)$,	-1°	γ_f,	0°
$\psi(t_0)$,	90°	β_f,	0°
$\alpha(t_0)$,	17°	p_f,	$0^\circ/s$
$\sigma(t_0)$,	-75°	q_f,	$0^\circ/s$
$\beta(t_0)$,	0°	ν_f,	$0^\circ/s$
$p(t_0)$,	$0^\circ/s$	K_q,	$9.289e^{-9}$

TABLE III: Assignment of vehicle-related parameters

Parameter	Value	Parameter	Value
R_e, ft	20902900	h_s, ft	23800
ρ_0, sl/ft^3	2.378×10^{-3}	Ω, rad/s	$7.29{\times}10^{-5}$
m, sl	6209.43	S, ft^2	2690
I_{xx}, sl/ft^2	434270	I_{xz}, sl/ft^2	17880
I_{yy}, sl/ft^2	961200	I_{zz}, sl/ft^2	1131541

B. Trajectory Ensemble Construction

We generate the optimal flight trajectory dataset used to train the DNNs in this subsection. As stated in Sec III.A, it is assumed that the stochastic disturbances ξ_x act on the initial state conditions x_0. Here, $\xi_x = [\xi_{r_0}, \xi_{\theta_0}, \xi_{\phi_0}, \xi_{V_0}, \xi_{\gamma_0}, \xi_{\psi_0}, \xi_{\alpha_0}, \xi_{\sigma_0}]^T$ and their values are uniformly distributed on the following regions:

$$\begin{array}{ll} \xi_{r_0} \in [-2500, 2500] \text{ ft} & \xi_{\theta_0} \in [-0.25^\circ, -0.25^\circ] \\ \xi_{\phi_0} \in [-1^\circ, 1^\circ] & \xi_{V_0} \in [-350, 350] \text{ ft/s} \\ \xi_{\gamma_0} \in [-0.5^\circ, 0.5^\circ] & \xi_{\psi_0} \in [-0.5^\circ, 0.5^\circ] \\ \xi_{\alpha_0} \in [-1^\circ, 1^\circ] & \xi_{\sigma_0} \in [-2^\circ, 2^\circ] \end{array}$$

A number of typical trajectory optimization methods are available in the literature [9]. However, due to the existence of stochastic disturbances, a direct implementation of these methods for the considered problem might result in poor convergence performance and local/infeasible solution detection [34]. To increase the success rate for finding optimal solutions, the desensitized trajectory optimization method established in Sec III.A is used to produce a set of optimal reentry trajectories. To highlight its advantages, a Monte Carlo (MC) simulation was executed to assess the convergence performance by using different trajectory optimization techniques. By setting the maximum iteration number to 5000, 3000 MC trials were performed. Convergence results obtained via the proposed method and the other two widely applied trajectory optimization algorithms are tabulated in Table IV. Three primary indicators $[I_s, I_i, I_m]$, representing times of optimal/infeasible solution found and times of maximum iteration reached, are presented in Table IV. Pr_s is the success rate for finding the optimal solution.

TABLE IV: Convergence results via different methods

Indicator	Proposed method	Method reported in [11]	Method reported in [35]
I_s	2772	2007	2098
I_i	0	220	79
I_m	228	773	823
Pr_s	92.40%	66.90%	69.93%

Using the IPSO algorithm to favorably start the gradient-based optimization process can improve the success rate for finding the optimal solution in the presence of initial condition disturbances. This confirms the benefit from applying the IPSO-based desensitization strategy as well as the scaling process. With the desensitized optimization method, 2×10^3 optimal trajectories for the considered problem are collected. $N_c = 100$ state-control pairs are extracted from each flight path, thereby resulting in 2×10^6 pairs.

C. Selection of DNN Structural Parameters

To steer the HV reentry flight, three DNNs (represented as $\mathcal{N}_{M_x}$, $\mathcal{N}_{M_y}$ and $\mathcal{N}_{M_z}$) are established. The networks are trained on the constructed trajectory dataset to approximate the inherent relation of the optimized state-control action. Within the dataset, 50% of the data are utilized for training, while 25% of the data are utilized for testing and validation.

As for the network structural parameters, the number of layers, units, and batches should be determined. Experiments were carried out by specifying four levels for each parameter, as detailed in Table V. Then, an orthogonal array can be constructed to describe different parameter combinations (as illustrated in Table VI). The DNN is trained for each parameter combination. The obtained average testing mean absolute error (Te-MAE) values for different experiments are also reported in Table VI. Based on the results from Table VI, the level trends of the DNN structural parameters are shown in Fig. 4, from where it can be observed that the best performance is acquired if these three parameters are set as $N_L = 5$, $N_o = 64$ and $N_b = 8$.

TABLE V: Parameter specification

Parameter	Level			
	1	2	3	4
N_L	3	4	5	6
N_o	16	32	64	128
N_b	6	8	10	12

Remark 1. Since the trained DNN acts as the optimal feedback action generator, there is a concern regarding the tradeoff between its control

TABLE VI: Results of different parameter combinations

Test No.	Level			Result
	N_L	N_o	N_b	Te-MAE
Experiment No. 1	1	1	1	0.1085
Experiment No. 2	1	2	2	0.0853
Experiment No. 3	1	3	3	0.0908
Experiment No. 4	1	4	4	0.0842
Experiment No. 5	2	1	2	0.0644
Experiment No. 6	2	2	1	0.0703
Experiment No. 7	2	3	4	0.0652
Experiment No. 8	2	4	3	0.0715
Experiment No. 9	3	1	3	0.0458
Experiment No. 10	3	2	4	0.0473
Experiment No. 11	3	3	1	0.0427
Experiment No. 12	3	4	2	0.0482
Experiment No. 13	4	1	4	0.0539
Experiment No. 14	4	2	3	0.0496
Experiment No. 15	4	3	2	0.0511
Experiment No. 16	4	4	1	0.0513

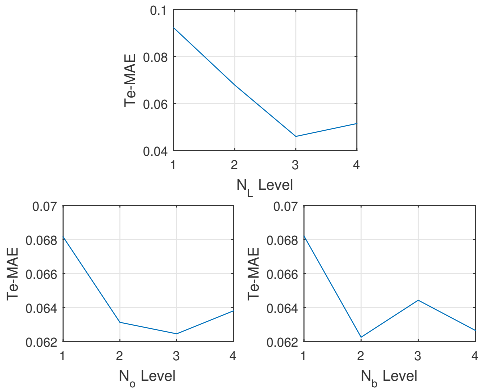

Fig. 4: Level trends of the DNN parameters

accuracy and online computational burden. From Table VI and Fig. 4, it can be expected that a more deeply trained network may result in a higher control accuracy. Notably, the computational complexity of the online process requires only a modest CPU effort. More precisely, to calculate the optimal feedback actions, we only need a finite number of forward operations influenced by the number of the network layer N_L. Trajectory ensemble generation and network training, on the other hand, are performed offline.

D. DNN-based Control Performance

The performance of applying the lower-level DNN-based feedback action generator is validated in this subsection. Denoting the trained DNNs as $\mathscr{N} = [\mathscr{N}_{M_x}, \mathscr{N}_{M_y}, \mathscr{N}_{M_z}]$, the online optimal feedback actions $M = [M_x, M_y, M_z]$ can be computed via $M(t_i) = \mathscr{N}(x(t_i))$. A sample test was executed, and the simulated optimal results and the DNN-driven results are visualized in Fig. 5. The characteristic of the simulated optimal state/control trajectories is first analyzed. The translational state profiles are demonstrated in Figs. 5(a)-(c). These state trajectories display a relatively smooth trend and can vary in their allowable regions as prespecified in the optimization model. On the other hand, Figs. 5(d)-(i) illustrate the time evolution of the rotational states during the reentry phase. Different from the translational state, the rotational states exhibit higher-frequency dynamics. The angular rate profiles are relatively small, which guarantees that unfeasibly sudden movements can be avoided during the flight.

In addition, attention is given to the effect of the accumulated heating load over the obtained solution profiles. Specifically, as can be seen from Fig. 5(d), the AOA profile displays a steadily increasing trend in order to slow down the HV (see Fig. 5(b)). This aspect guarantees that the accumulated heating load does not increase significantly. In addition, according to the definition of the objective function (6), it is obvious that the total amount of heating load depends strongly on the mission duration time. Therefore, we can expect that minimizing the accumulated heating load will indirectly shrink the flight duration to some extent.

The actual flight trajectories driven by the proposed DNN-based control scheme are now studied. From Fig. 5, we can see that the DNN-driven control scheme is able to generate almost identical solutions as the simulated optimal results. Hence, it can be concluded that the structure of the optimized control signal is maximally preserved. It is noteworthy that although the state and control evolutions illustrated in Fig. 5 are used only as an instance to verify the control performance of the proposed DNN-based method, other test cases among the test set can display similar results.

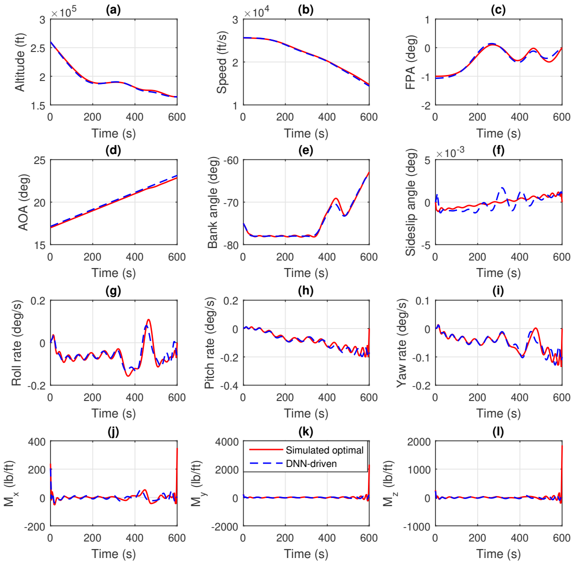

Fig. 5: Results produced via the DNN-driven approach

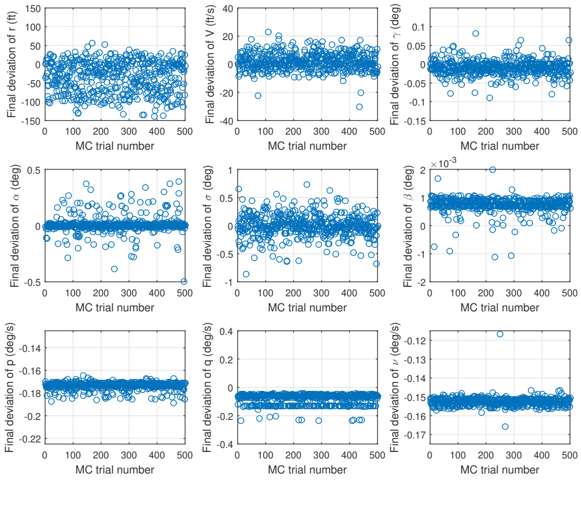

Fig. 6: Final state deviations

Results regarding the terminal state distribution $e_f = x_f - x(t_f)$ for 500 test cases are visualized in Fig. 6, where the terminal state errors are denoted as:

$$\begin{cases} e_{r_f} = r(t_f) - r_f & e_{V_f} = V(t_f) - V_f \\ e_{\gamma_f} = \gamma(t_f) - \gamma_f & e_{p_f} = p(t_f) - p_f \\ e_{q_f} = q(t_f) - q_f & e_{\nu_f} = \nu(t_f) - \nu_f \end{cases}$$

From Fig. 6, it is obvious that e_{r_f}, e_{V_f} and e_{γ_f} can be successfully steered into small neighborhoods around zero; specifically, their values can be steered into $[-150, 150]$ ft, $[-25, 25]$ ft/s and $[-0.1^\circ, 0.1^\circ]$, respectively. Hence, the effectiveness and reliability of applying the proposed bilevel control scheme can be confirmed.

E. Comparative Analysis

Comparative case studies were executed to demonstrate the control performance of implementing different IGC methods in real time. For instance, one comparative study is performed between the DNN-based approach and the IGC method designed in [30]. Five hundred MC test trials were executed for the considered problem. The statistical performance regarding the terminal state dispersions and the mean CPU processing time t_p (for every control action) are shown in Table VII.

TABLE VII: Statistical performance regarding terminal state values

	Final state value	DNN-driven approach	Approach [30]
Mean value	e_{r_f}	-34.3550	22.5621
	e_{V_f}	1.8614	0.5427
	e_{γ_f}	-0.0093	-0.0142
	e_{p_f}	-0.1737	-0.1421
	e_{q_f}	-0.0829	-0.2235
	e_{ν_f}	-0.1530	0.3142
Standard deviation	e_{r_f}	41.8435	13.7935
	e_{V_f}	6.0374	0.4431
	e_{γ_f}	0.0195	0.0144
	e_{p_f}	0.0034	0.0179
	e_{q_f}	0.0379	0.0568
	e_{ν_f}	0.0078	0.0131
Average CPU time	t_p	9.7579 ms	0.6049s

From Table VII, we can see that the DNN-based feedback action generator is able to acquire comparable control performance with the controller suggested in [30]. Using the proposed approach, we are able to achieve a microsecond-class processing ability. This is much more efficient than using the controller designed in [30] (second-class processing ability). This performance can be attributed by the fact that in the proposed bilevel structure, there is no need for the time-consuming online optimization process. To better demonstrate the online processing performance of the DNN-based algorithm, a histogram of the average CPU time of the 500 MC tests is depicted in Fig. 7. Based on the computational results, the strong performance of the online implementation of an optimal feedback controller modeled via DNNs is further verified.

F. Experimental Study

This subsection describes experimental studies carried out on a practical testing framework to further validate the effectiveness of the designed strategy. The testing system consists of a server (Dell EMC PowerEdge R930 rack), an industrial PC (IPC-610MB-30LDE/I5-2400/DDR3 8 GB) and an embedded controller (NI PXI-8820) as shown in Fig. 8. Specifically, the server is applied to train DNNs, while the PC is used to convert the real-time simulation to executable files via LabVIEW Real-Time Module. Following that, the performance is tested on the embedded controller.

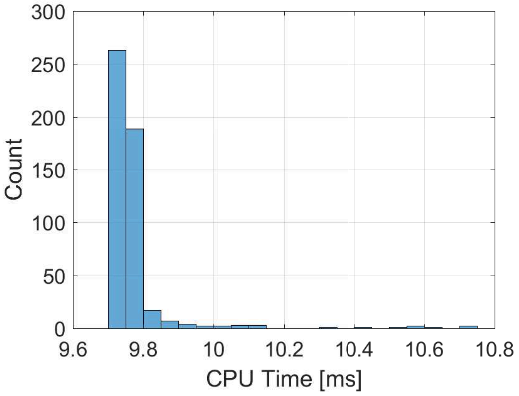

Fig. 7: Histogram of MC tests (Average CPU time)

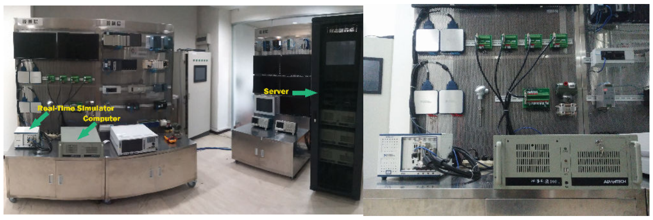

Fig. 8: Testing framework

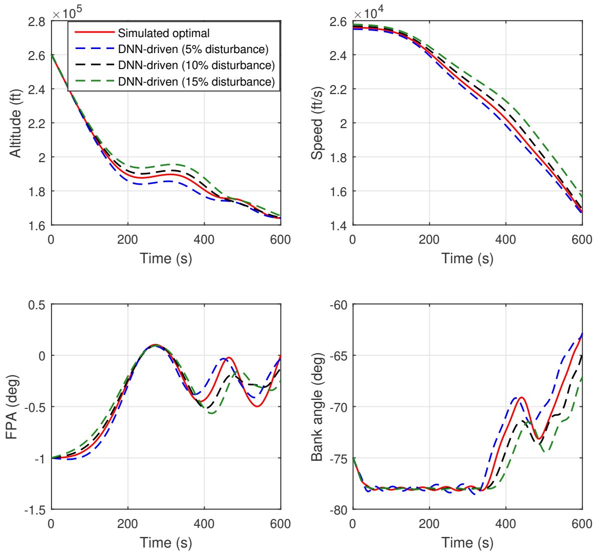

Fig. 9: DNN-driven performance

The DNN-driven results, together with the simulated optimal flight trajectories, are depicted in Fig. 9, from where it is obvious that the proposed strategy can steer the 6-DOF HV model to fly along the optimal solution. Supplementary works were performed to study the impact of disturbances/uncertainties on the control performance. For example, it is assumed that the inertia moment I_{ij} has some variations (e.g., 5%, 10%, and 15%). The resulting DNN-driven results are presented in Fig. 9. We can observe that the DNN-driven results start deviating from the simulated optimal solution as the uncertainty increases. Actually, a potential way to strengthen the online control accuracy is to expand the training dataset such that it can cover more uncertain cases. For instance, the original trajectory optimization model can be extended by including a variety of uncertainties or noise-perturbed dynamics. As a result, the trained networks will have an enhanced capability in order to deal with the disturbances/uncertainties, thus further improving the robustness as well as the practicability of the proposed approach.

V. Conclusion

In this work, a bilevel control scheme incorporating an improved trajectory optimization method and DNNs was designed to steer HV reentry flight. By taking into consideration the rotational effects, the typical 3-DOF HV model was extended to a 6-DOF version in the trajectory optimization phase. Furthermore, DNNs were established to study the simulated optimal state-control actions such that they can generate optimal feedback actions in real time. Numerical studies were performed to assess the real-time capability of the DNN-based control scheme. Experimental studies were executed to confirm the reliability of the proposed method. According to the obtained results, we can conclude that in comparison to other IGC approaches reported in the literature, the new design tends to be easier and more straightforward to implement. Moreover, this design has the potential to be implemented in real HV reentry flight situations. As a result, we believe the suggested bilevel approach and obtained numerical results are of practical interest to communities that are involved with deep neural network-based control applications and spacecraft guidance and control systems.

Some future research directions can be inspired from the study carried out in this paper. For example, it would be worthwhile to test the proposed strategy on solving other trajectory planning and attitude control applications. Efforts can also be made to investigate the possibility of applying reinforcement learning approaches to learn the optimal feedback actions in a highly stochastic environment.

References

[1] A. Zou and K. D. Kumar, "Neural network-based distributed attitude coordination control for spacecraft formation flying with input saturation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1155–1162, 2012.

[2] C. Mu, Z. Ni, C. Sun, and H. He, "Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 584–598, 2017.

[3] B. Xu, D. Yang, Z. Shi, Y. Pan, B. Chen, and F. Sun, "Online recorded data-based composite neural control of strict-feedback systems with application to hypersonic flight dynamics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3839–3849, 2018.

[4] M. C. W. Bergsma and E. Mooij, "Application of taylor-series integration to reentry problems with wind," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 10, pp. 2324–2335, 2016.

[5] A. Rahimi, K. Dev Kumar, and H. Alighanbari, "Particle swarm optimization applied to spacecraft reentry trajectory," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 1, pp. 307–310, 2012.

[6] L. Mu, X. Yu, Y. M. Zhang, P. Li, and X. Wang, "Onboard guidance system design for reusable launch vehicles in the terminal area energy management phase," *Acta Astronautica*, vol. 143, pp. 62–75, 2018.

[7] R. Chai, A. Tsourdos, A. Savvaris, Y. Xia, and S. Chai, "Real-time reentry trajectory planning of hypersonic vehicles: A two-step strategy incorporating fuzzy multi-objective transcription and deep neural network," *IEEE Transactions on Industrial Electronics*, to be published.

[8] R. Chai, A. Savvaris, and A. Tsourdos, "Violation learning differential evolution-based hp-adaptive pseudospectral method for trajectory optimization of space maneuver vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 4, pp. 2031–2044, 2017.

[9] B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *Journal of Optimization Theory and Applications*, vol. 152, no. 2, pp. 271–306, 2012.

[10] C. Sanchez-Sanchez and D. Izzo, "Real-time optimal control via deep neural networks: Study on landing problems," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 5, pp. 1122–1135, 2018.

[11] T. Guo, J. Li, H. Baoyin, and F. Jiang, "Pseudospectral methods for trajectory optimization with interior point constraints: Verification and applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 2005–2017, 2013.

[12] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Optimal tracking guidance for aeroassisted spacecraft reconnaissance mission based on receding horizon control," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1575–1588, 2018.

[13] J. Zhao and R. Zhou, "Pigeon-inspired optimization applied to constrained gliding trajectories," *Nonlinear Dynamics*, vol. 82, no. 4, pp. 1781–1795, 2015.

[14] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Unified multiobjective optimization scheme for aeroassisted vehicle trajectory planning," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 7, pp. 1521–1530, 2018.

[15] N. Wang, H. R. Karimi, H. Li, and S. Su, "Accurate trajectory tracking of disturbed surface vehicles: A finite-time control approach," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1064–1074, 2019.

[16] N. Wang, S. Su, X. Pan, X. Yu, and G. Xie, "Yaw-guided trajectory tracking control of an asymmetric underactuated surface vehicle," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3502–3513, 2019.

[17] N. Wang, G. Xie, X. Pan, and S. Su, "Full-state regulation control of asymmetric underactuated surface vehicles," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8741–8750, 2019.

[18] B. Xiao, Q. Hu, and Y. Zhang, "Finite-time attitude tracking of spacecraft with fault-tolerant capability," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1338–1350, 2015.

[19] Y. Wang, X. Yang, and H. Yan, "Reliable fuzzy tracking control of near-space hypersonic vehicle using aperiodic measurement information," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9439–9447, 2019.

[20] H. Sun, Y. Li, G. Zong, and L. Hou, "Disturbance attenuation and rejection for stochastic markovian jump system with partially known transition probabilities," *Automatica*, vol. 89, pp. 349–357, 2018.

[21] D. Yang, G. Zong, and H. R. Karimi, "h_∞ refined anti-disturbance control of switched lpv systems with application to aero-engine," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2019.

[22] H. Sun, L. Hou, G. Zong, and X. Yu, "Fixed-time attitude tracking control for spacecraft with input quantization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 1, pp. 124–134, 2019.

[23] P. Li, X. Yu, Y. Zhang, and X. Peng, "Adaptive multivariable integral tsmc of a hypersonic gliding vehicle with actuator faults and model uncertainties," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2723–2735, 2017.

[24] X. Yu, P. Li, and Y. Zhang, "The design of fixed-time observer and finite-time fault-tolerant control for hypersonic gliding vehicles," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4135–4144, 2018.

[25] L. Mu, L. Li, X. Yu, Y. Zhang, P. Li, and X. Wang, "Observer-based fault-tolerant control of hypersonic scramjet vehicles in the presence of actuator faults and saturation," *International Journal of Robust and Nonlinear Control*, vol. 0, no. 0, 2017.

[26] X. Li, C. K. Ahn, D. Lu, and S. Guo, "Robust simultaneous fault estimation and nonfragile output feedback fault-tolerant control for markovian jump systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–8, 2018.

[27] W. Tang, W. Long, and H. Gao, "Model predictive control of hypersonic vehicles accommodating constraints," *IET Control Theory & Applications*, vol. 11, no. 15, pp. 2599–2606, 2017.

[28] Q. Weiwei, H. Bing, L. Gang, and Z. Pengtao, "Robust model predictive tracking control of hypersonic vehicles in the presence of actuator constraints and input delays," *Journal of the Franklin Institute*, vol. 353, no. 17, pp. 4351–4367, 2016.

[29] J. D. Schierman, D. G. Ward, J. R. Hull, N. Gandhi, M. Oppenheimer, and D. B. Doman, "Integrated adaptive guidance and control for re-entry vehicles with flight test results," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 6, pp. 975–988, 2004.

[30] B. Tian, W. Fan, R. Su, and Q. Zong, "Real-time trajectory and attitude coordination control for reusable launch vehicle in reentry phase," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1639–1650, 2015.

[31] R. Chai, A. Savvaris, A. Tsourdos, and S. Chai, *Real-Time Optimal Guidance and Control Strategies for Space Maneuver Vehicles*. Singapore: Springer Singapore Press, 2020, pp. 133–161.

[32] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, and Y. Xia, “Two-stage trajectory optimization for autonomous ground vehicles parking maneuver,” *IEEE Transactions on Industrial Informatics*, pp. 1–11, 2018.

[33] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, “Improved gradient-based algorithm for solving aeroassisted vehicle trajectory optimization problems,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 8, pp. 2093–2101, 2017.

[34] ——, “Trajectory optimization of space maneuver vehicle using a hybrid optimal control solver,” *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 467–480, 2019.

[35] D. Gonzalez-Arribas, M. Soler, and M. Sanjurjo-Rivo, “Robust aircraft trajectory planning under wind uncertainty using optimal control,” *Journal of Guidance, Control, and Dynamics*, pp. 1–16, 2017.

Six-DOF spacecraft optimal trajectory planning and real-time attitude control: a deep neural network-based approach

Chai, Runqi

IEEE