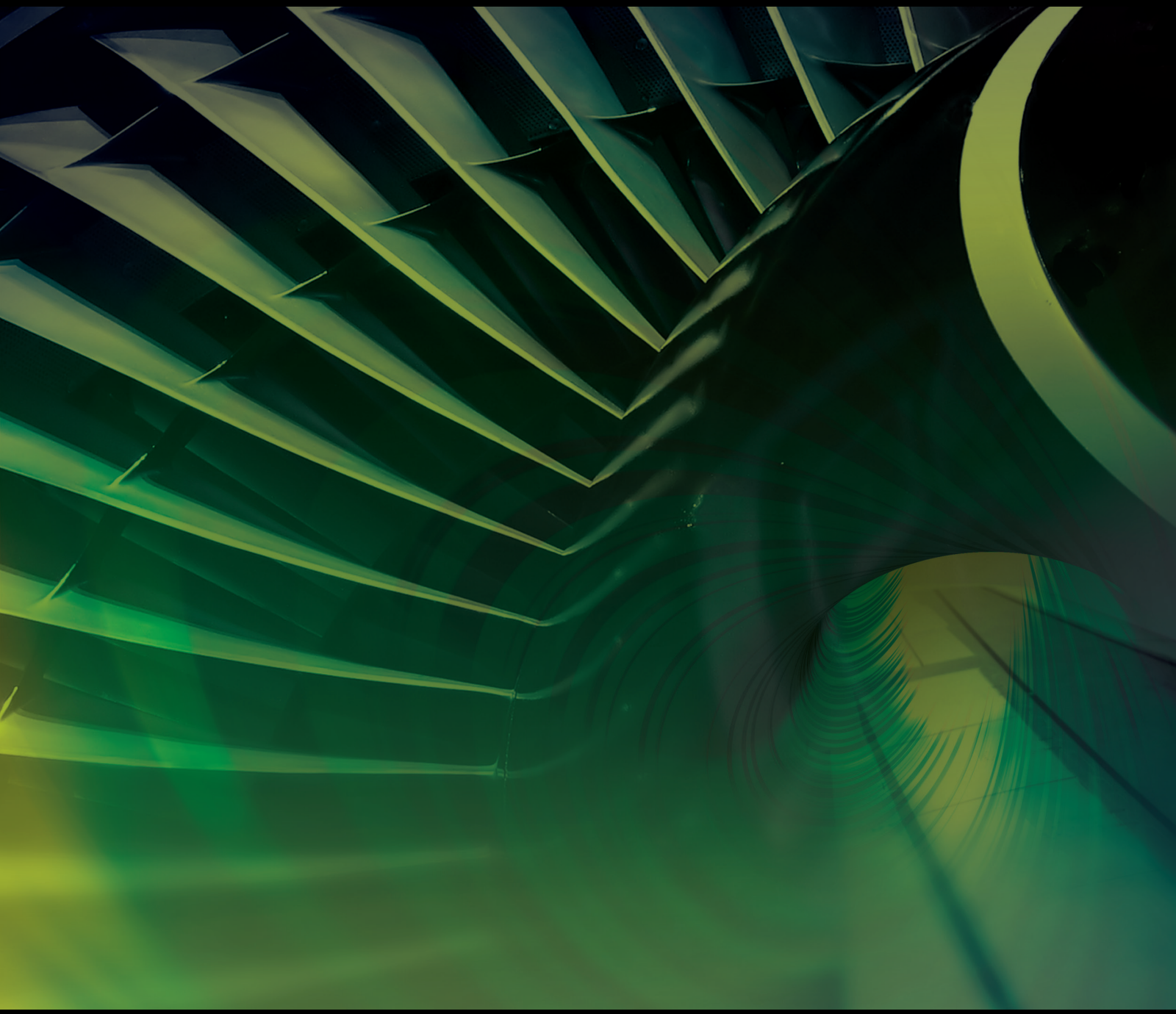
International Journal of Aerospace Engineering

Emerging Techniques in Aerospace Controls

Lead Guest Editor: Mahmut Reyhanoglu
Guest Editors: Erdal Kayacan and William MacKunis



Emerging Techniques in Aerospace Controls

International Journal of Aerospace Engineering

Emerging Techniques in Aerospace Controls

Lead Guest Editor: Mahmut Reyhanoglu
Guest Editors: Erdal Kayacan and William MacKunis

Contents

Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 7570059, 12 pages
https://doi.org/10.1155/2021/7570059

Research Article

Robust Controller Designing for an Air-Breathing Hypersonic Vehicle with an HOSVD-Based LPV Model

Xing He,[1] **Wei Jiang**,[2] **and Caisheng Jiang**[2]

[1]*School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an, Shaanxi 710055, China*
[2]*Xi'an Research Institute of High-Tech, Xi'an, Shaanxi 710025, China*

Correspondence should be addressed to Xing He; trees241@163.com

Received 4 May 2021; Revised 14 September 2021; Accepted 13 November 2021; Published 3 December 2021

Academic Editor: Mahmut Reyhanoglu

This paper focuses on the linear parameter varying (LPV) modeling and controller design for a flexible air-breathing hypersonic vehicle (AHV). Firstly, by selecting the measurable altitude and velocity as gain-scheduled variables, the original longitudinal nonlinear model for AHV is transformed into the LPV model via average gridding division, vertex trimming, Jacobian linearization, and multiple linear regression within the entire flight envelope. Secondly, using the tensor product model transformation method, the obtained LPV model is converted into the polytopic LPV model via high-order singular value decomposition (HOSVD). Third, the validity and applicability of the HOSVD-based LPV model are further demonstrated by designing a robust controller for command tracking control during maneuvering flight over a large envelope.

1. Introduction

Since the 50s and 60s of the 20th century, with the series of events such as the advent of long-range ballistic missiles, the successful return of manned spacecraft, and the X-15 test aircraft velocity faster than Mach 6, it marked that mankind formally entered the era of hypersonic velocity. The hypersonic vehicle has taken advantages of both spacecraft and aircraft and has become the new technological commanding point in the 21st century aerospace field. It has been hailed as the third epoch-making milestone in the history of aviation in the world after the invention of aircraft and the breakthrough of the sound barrier [1–3].

Compared with a conventional aircraft, the complex aerodynamic characteristics of hypersonic vehicles and the application of technologies such as scramjet engines and integrated airframe/engine configuration make the coupling more powerful between elastic bodies, propulsion systems, aerodynamics, and structural dynamics of hypersonic vehicles; besides, the physical model is more nonlinear. At present, the main means of dynamic modeling for hypersonic vehicles are computational fluid dynamics (CFD), wind tunnel tests, aircraft tests, and other test methods. Because of the imperfection of hypersonic aerodynamic database and CFD software, the low degree of wind tunnel test approximation, the high cost and risk of flight tests, and the limitation of hypersonic vehicles modeling methods, the system modeling has great structural and parameter uncertainties. Second, the large-span flight envelope of hypersonic vehicles not only requires the flight control system to adapt to different flight environments but also requires to satisfy the strict constraints such as more complex dynamic pressure, overload, flight envelopes, and actual physical characteristics (e.g., actuator saturation). In summary, a hypersonic vehicle is a highly dynamic, strongly coupled, fast time-varying, and multivariable nonlinear system. Its control system design not only faces the influence of model uncertainty and external environment but also faces the constraints of actual existence and the time-delay dynamic limitation, which makes the control, stability, safety, rapidity, and accuracy

of hypersonic vehicle systems face great challenges. It has become one of the key issues to be solved and is also a hot issue in the field of flight control, such as adaptive control strategies based on the multiple Lyapunov function method or using an auxiliary reaction control system for air-breathing hypersonic vehicles [4–8].

Recently, the LPV controller has emerged as a reliable alternative to classical gain scheduling approaches and provides guaranteed stability, robustness, or performance [9, 10]. Gain-scheduled control strategies for LPV systems have been developed intensively and bring promising solutions in many fields, such as missiles, aircrafts, spacecrafts, winding systems, automated vehicles, and robotic systems [11–13, 14]. Zhang handled the reference tracking for a class of flexible hypersonic vehicles using a novel switched LPV framework [15]. Yang proposed a fault-tolerant controller based on robust model-predictive control and A polytopic LPV model for A hypersonic vehicle with external disturbances and actuator loss of effectiveness faults [16]. Hu proposes a novel passive fault-tolerant control method using weighted tube-based model predictive control via polytopic LPV for air-breathing hypersonic vehicles [17]. However, the formulation of the polytopic LPV model for nonlinear systems and controller design for LPV systems with uncertainty and external disturbance remain difficult, which motivates our study.

This paper takes the flight control system design of the hypersonic vehicle as the background. In the process of modeling, it is necessary to deal with various factors that affect the performance of the control system. Aircraft modeling errors, flight parameter changes of large package flight, elastic vibration deformation of aircraft, and other factors are regarded as internal factors leading to system uncertainty. Factors such as the complex flight environment, dynamic pressure effect, aerodynamic thermal effect, and the frictional resistance effect are regarded as external disturbances. Then, the attitude stability and command tracking issues during the hypersonic vehicle maneuvering flight in large envelope are transformed into the robust controller synthesis and guaranteed performance control of the uncertain disturbance system. Firstly, for a nonlinear longitudinal elasticity model of air-breathing hypersonic vehicles, the altitude and velocity are selected as scheduling parameters, uniform meshing is performed within the range of parameter variation, and then, the LPV model of the aircraft is obtained by Jacobi linearization and multivariate linear fitting. Secondly, the LPV model of the aircraft is transformed into a polytopic LPV model by the tensor product model change method, which lays a foundation for the subsequent aircraft controller design. Finally, a robust controller is designed based on the obtained aircraft polytopic LPV model, which verifies the effectiveness of the aircraft polytopic LPV model in the command tracking control of aircraft maneuvering flight in large envelopes.

2. Problem Formulation

2.1. Control-Oriented Model. The longitudinal elastic nonlinear model of the hypersonic vehicle used in this paper is mainly based on the winged-cone model provided by NASA's Langley Research Center [18–20], which takes into account the influence of the elastic modality of the aircraft. The simplified longitudinal elastic nonlinear model of the aircraft is as follows:

$$\begin{aligned}
\dot{h} &= V\sin(\theta-\alpha),\\
\dot{V} &= \frac{T\cos\alpha - D}{m} - g\sin(\theta-\alpha),\\
\dot{\alpha} &= \frac{-T\sin\alpha - L}{mV} + Q + \frac{g\cos(\theta-\alpha)}{V},\\
\dot{\theta} &= Q,\\
\dot{Q} &= \frac{M}{I_{yy}},\\
\ddot{\eta}_i &= -2\varsigma_i\omega_i\dot{\eta}_i - \omega_i^2\eta_i + N_i, \quad i = 1, 2.
\end{aligned} \tag{1}$$

In equation (1), h represents the aircraft altitude, V represents the aircraft velocity, α indicates the angle of attack, θ denotes the pitch angle, Q represents the pitch rate, g indicates the gravitational constant, M is the pitching moment, I_{yy}represents the moment of inertia around the body axis, T is the engine thrust, D represents resistance, L represents lift, $\eta_i(i = 1, 2)$ represents the elastic mode, $\omega_i(i = 1, 2)$ represents the natural frequency of the elastic mode, $\varsigma_i(i = 1, 2)$ represents the damping coefficient of the elastic mode, and $N_i$denotes the generalized force.

The approximate expressions of the above aerodynamics, moments, thrusts, and generalized forces fitted near the equilibrium point are shown in equation (2). In equation (2), ρindicates air density, S is the reference area, $C_L(\alpha, \delta_e)$ represents the lift coefficient related to the angle of attack and elevator deflection, $C_D(\alpha, \delta_e)$ indicates the drag coefficient related to the angle of attack and elevator deflection, $C_{M,\alpha}(\alpha)$ indicates the torque coefficient related to the angle of attack, $C_{M,\delta_e}(\delta_e)$ indicates the torque coefficient related to the elevator deflection, z_T represents the coupling coefficient between thrust and pitching moment, $C_T^{\alpha^j}$ represents the thrust coefficient related to $\alpha^j(i = 1, 2, 3)$, $N_i^{\alpha^j}$ represents the generalized force coefficients related to α^j, and $N_2^{\delta_e}$ denotes the influence coefficient of elevator declination to generalized force N_2, and C_T^0 and N_i^0 represent constant items in thrust and generalized forces, respectively.

$$\begin{aligned}
L &\approx \frac{1}{2}\rho V^2 S C_L(\alpha, \delta_e),\\
D &\approx \frac{1}{2}\rho V^2 S C_D(\alpha, \delta_e),\\
M &\approx z_T T + \frac{1}{2}\rho V^2 S\bar{c}\left(C_{M,\alpha}(\alpha) + C_{M,\delta_e}(\delta_e)\right),\\
T &\approx C_T^{\alpha^3}\alpha^3 + C_T^{\alpha^2}\alpha^2 + C_T^{\alpha}\alpha + C_T^0,\\
N_1 &\approx N_1^{\alpha^2}\alpha^2 + N_1^{\alpha}\alpha + N_1^0,\\
N_2 &\approx N_2^{\alpha^2}\alpha^2 + N_2^{\alpha}\alpha + N_2^{\delta_e}\delta_e + N_2^0,
\end{aligned} \tag{2}$$

$$\begin{aligned}
\rho &= \rho_0 \exp\left(\frac{-(h-h_0)}{h_s}\right),\\
C_L &= C_L^{\alpha}\alpha + C_L^{\delta_e}\delta_e + C_L^0,\\
C_D &= C_D^{\alpha^2}\alpha^2 + C_D^{\alpha}\alpha + C_D^{\delta_e^2}\delta_e^2 + C_L^{\delta_e}\delta_e + C_D^0,\\
C_{M,\alpha} &= C_{M,\alpha}^{\alpha^2}\alpha^2 + C_{M,\alpha}^{\alpha}\alpha + C_{M,\alpha}^0,\\
C_{M,\delta_e} &= c_e\delta_e,\\
\bar{q} &= 0.5\rho V^2,\\
C_T^{\alpha^3} &= \beta_1(h,\bar{q})\Phi + \beta_2(h,\bar{q}),\\
C_T^{\alpha^2} &= \beta_3(h,\bar{q})\Phi + \beta_4(h,\bar{q}),\\
C_T^{\alpha} &= \beta_5(h,\bar{q})\Phi + \beta_6(h,\bar{q}),\\
C_T^0 &= \beta_7(h,\bar{q})\Phi + \beta_8(h,\bar{q}).
\end{aligned} \tag{3}$$

The relevant force and moment coefficients are shown in equation (3), where $1/h_s$is the air density decay rate, C_L^0 and C_D^0 are the constant term in $C_L(\alpha,\delta_e)$and $C_D(\alpha,\delta_e)$, respectively, $C_L^{\alpha_i}$ and $C_D^{\alpha_i}$ are the ith-order coefficient of α contribution to $C_L(\alpha,\delta_e)$ and $C_D(\alpha,\delta_e)$, respectively, $C_L^{\delta_e}$ is the coefficient of δ_e contribution to $C_L(\alpha,\delta_e)$, $C_D^{\delta_e^i}$ is the ith-order coefficient of δ_e contribution to $C_D(\alpha,\delta_e)$, $C_{M,\alpha}^{\alpha_i}$ is the ith order coefficient of α contribution to $C_{M,\alpha}(\alpha)$, $C_{M,\alpha}^0$ is the constant term in $C_{M,\alpha}(\alpha)$, $\bar{q}$ is the dynamic pressure, and Φ is the stoichiometrically normalized fuel-to-air ratio, $\beta_i(h,\bar{q})$ is the ith thrust fit parameter. The parameter reference values of the elastic nonlinear model are detailed in Hu et al. [21] and listed as Table 1.

Select $\mathbf{x}(t) = [h, V, \alpha, \theta, Q, \eta_1, \dot{\eta}_1, \eta_2, \dot{\eta}_2]^T$ as state variables, select $\mathbf{u}(t) = [\Phi, \delta_e]^T$ as system inputs, and select $\mathbf{z}(t) = [V, h]^T$ as system outputs. Substitute equations (2) and (3) into (1) to obtain the affine nonlinear model of the system as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}, \mathbf{u}(t), t) = [f_1(\mathbf{x},t), f_2(\mathbf{x},t), f_3(\mathbf{x},t), \cdots, f_9(\mathbf{x},t)]^T. \tag{4}$$

In equation (4), $f_i(\mathbf{x},t)$ $(i = 1, 2, \cdots, 9)$ is shown on the right side of equation (1). The equilibrium state of a hypersonic vehicle is a specific flight condition under which the line velocity, and angular velocity are constant or zero and the acceleration is zero under the given constraints. Therefore, the system equilibrium point sequence can be solved by equation (5) as follows:

$$f_i(\mathbf{x}, t) = 0, \quad i = 1, 2, \cdots, 9. \tag{5}$$

Within the permissible range of parameters, the velocity and altitude space of the selected scheduling variables are gridded and each point on the grid is calculated and balanced according to equation (5) to obtain the equilibrium working point of the aircraft. A set of LTI systems is obtained by using the Jacobian linearization method. Finally, these linearized local systems are multivariate linearly fitted to obtain the LPV model. The equidistant gridding method is chosen in this paper. Five points are selected at equal intervals in the velocity interval [7200 9600] ft/s, and four points are selected at equal

Table 1: Nomenclature and nominal values.

Symbol	Nominal value
m	300 slug/ft
I_{yy}	5×10^5 lb · ft
$\bar{c}$	17 ft
ρ_0	6.7429×10^{-5} slugs · ft^{-3}
h_0	85000 ft
h_s	2.1358×10^4 ft
g	31.92 ft·s^{-2}
S	17 ft^2·ft^{-1}
z_T	8.36 ft
C_L^{α}	4.6773 rad^{-1}
$C_L^{\delta_e}$	0.76224 rad^{-1}
C_L^0	-1.8714×10^{-2}
$C_D^{\alpha^2}$	5.8224 rad^{-2}
C_D^{α}	-4.5315×10^{-2} rad^{-1}
$C_D^{\delta_e^2}$	8.1993×10^{-1} rad^{-2}
$C_D^{\delta_e}$	2.7699×10^{-4} rad^{-1}
C_D^0	1.0131×10^{-2}
$C_{M,\alpha}^{\alpha^2}$	6.2926 rad^{-2}
$C_{M,\alpha}^{\alpha}$	2.1335 rad^{-1}
$C_{M,\alpha}^0$	0.18979
c_e	−1.2897 rad^{-1}
β_1	-3.7693×10^5 lb · ft^{-1} · rad^{-3}
β_2	-3.7225×10^4 lb · ft^{-1} · rad^{-3}
β_3	2.6814×10^4 lb · ft^{-1} · rad^{-3}
β_4	$^{-}1.7277 \times 10^4$ lb · ft^{-1} · rad^{-3}
β_5	3.5542×10^4 lb · ft^{-1} · rad^{-3}
β_6	-2.4216×10^3 lb · ft^{-1} · rad^{-3}
β_7	6.3785×10^3 lb · ft^{-1}
β_8	-1.0090×10^2 lb · ft^{-1}
$N_1^{\alpha^2}$	1.4013×10^3 lb · ft^{-1} · slug$^{-0.5}$ · rad^{-2}
N_1^{α}	4.5737×10^3 lb · ft^{-1} · slug$^{-0.5}$ · rad^{-1}
N_1^0	1.1752×10^2 lb · ft^{-1} · slug$^{-0.5}$
$N_2^{\alpha^2}$	-5.0227×10^3 lb · ft^{-1} · slug$^{-0.5}$ · rad^{-2}
N_2^{α}	2.8633×10^3 lb · ft^{-1} · slug$^{-0.5}$ · rad^{-1}
$N_2^{\delta_e}$	1.2465×10^3 lb · ft^{-1} · slug$^{-0.5}$ · rad^{-1}
N_2^0	-44.201×10^2 lb · ft^{-1} · slug$^{-0.5}$

intervals in the altitude interval [80000 110000] ft, totaling 20 points, as shown in Figure 1.

By means of parameter meshing, Jacobian linearization, and multivariate linear fitting, the LPV model of the hypersonic vehicle with respect to the scheduling parameters is shown in equation (6). The fitting results of partial LPV model parameters are shown in Figure 2, and its corresponding fitting results are listed as follows.

$$\begin{bmatrix} \dot{h} \\ \dot{V} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{Q} \\ \dot{\eta}_1 \\ \ddot{\eta}_1 \\ \dot{\eta}_2 \\ \ddot{\eta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -V & V & 0 & 0 & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & -g & 0 & 0 & 0 & 0 & 0 \\ A_{31} & A_{32} & A_{33} & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ A_{51} & A_{52} & A_{53} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & A_{73} & 0 & 0 & -\omega_1^2 & -2\varsigma_1\omega_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & A_{93} & 0 & 0 & 0 & 0 & -\omega_2^2 & -2\varsigma_2\omega_2 \end{bmatrix} \begin{bmatrix} h \\ V \\ \alpha \\ \theta \\ Q \\ \eta_1 \\ \dot{\eta}_1 \\ \eta_2 \\ \dot{\eta}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ B_{21} & B_{22} \\ B_{31} & B_{32} \\ 0 & 0 \\ B_{51} & B_{52} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & N_2^{\delta_e} \end{bmatrix} \begin{bmatrix} \Phi \\ \delta_e \end{bmatrix}, \tag{6}$$

$$\begin{aligned}
\mathbf{A}_{21} &= 0.005869328270053 + 0.000000350162259V \\
&\quad - 0.000000174371733h - 0.000000000008442V \cdot h \\
&\quad + 0.000000000022808V^2 + 0.000000000001476h^2, \\
\mathbf{A}_{22} &= -0.023088159404555 - 0.000002004091776V \\
&\quad + 0.000000734297646h + 0.000000000065894V \cdot h \\
&\quad - 0.000000000208165V^2 - 0.000000000007750h^2, \\
\mathbf{A}_{23} &= -4.281455848712256 \times 10^2 + 0.000203656246563 \\
&\quad \times 10^2 V + 0.000109797017027 \times 10^2 h \\
&\quad - 0.000000002665318 \times 10^2 V \cdot h + 0.000000008045619 \\
&\quad \times 10^2 V^2 - 0.000000000711388 \times 10^2 h^2, \\
\mathbf{A}_{31} &= 0.108230120558901 \times 10^{-5} + 0.000004011899188 \\
&\quad \times 10^{-5} V - 0.000003114516949 \times 10^{-5} h \\
&\quad - 0.000000000147221 \times 10^{-5} V \cdot h + 0.000000000376981 \\
&\quad \times 10^{-5} V^2 + 0.000000000030249 \times 10^{-5} h^2, \\
\mathbf{A}_{32} &= -0.457306127416146 \times 10^{-5} + 0.000041821380304 \\
&\quad \times 10^{-5} V + 0.000004391002007 \times 10^{-5} h \\
&\quad + 0.000000000788219 \times 10^{-5} V \cdot h - 0.000000005149628 \\
&\quad \times 10^{-5} V^2 - 0.000000000076799 \times 10^{-5} h^2, \\
\mathbf{A}_{33} &= -0.003787681755457 - 0.000001119856725V \\
&\quad + 0.000000058391695h + 0.000000000008971V \cdot h \\
&\quad - 0.000000000041242V^2 - 0.000000000000762h^2, \\
\mathbf{A}_{51} &= 0.296461521297026 \times 10^{-4} + 0.000017734787744 \\
&\quad \times 10^{-4} V - 0.000008809870838 \times 10^{-4} h \\
&\quad - 0.000000000429950 \times 10^{-4} V \cdot h + 0.000000001166841 \\
&\quad \times 10^{-4} V^2 + 0.000000000074731 \times 10^{-4} h^2, \\
\mathbf{A}_{52} &= -0.116805126192769 \times 10^{-3} - 0.000010123132197 \\
&\quad \times 10^{-3} V + 0.000003712711998 \times 10^{-3} h \\
&\quad + 0.000000000334612 \times 10^{-3} V \cdot h - 0.000000001060704 \\
&\quad \times 10^{-3} V^2 - 0.000000000039242 \times 10^{-3} h^2, \\
\mathbf{A}_{53} &= 5.802500675695122 - 0.000054761923165V \\
&\quad - 0.000140973420646h + 0.000000002348128V \cdot h \\
&\quad + 0.000000030407243V^2 + 0.000000000863671h^2, \\
\mathbf{A}_{73} &= 4.841368923917495 \times 10^3 - 0.000005151942352 \\
&\quad \times 10^3 V - 0.000006725356453 \times 10^3 h \\
&\quad - 0.000000001382188 \times 10^3 V \cdot h + 0.000000005761907 \\
&\quad \times 10^3 V^2 + 0.000000000136397 \times 10^3 h^2, \\
\mathbf{A}_{93} &= 1.903890377392848 \times 10^3 + 0.000018466182009 \\
&\quad \times 10^3 V + 0.000024105793090 \times 10^3 h \\
&\quad + 0.000000004954198 \times 10^3 V \cdot h - 0.000000020652486 \\
&\quad \times 10^3 V^2 - 0.000000000488889 \times 10^3 h^2, \\
\mathbf{B}_{21} &= 29.298643129461226 - 0.000509535839514V \\
&\quad - 0.000182979829453h - 0.000000042211072V \cdot h \\
&\quad + 0.000000177655838V^2 + 0.000000004424883h^2, \\
\mathbf{B}_{22} &= -2.492498015279544 \times 10^2 - 0.000058628507583 \\
&\quad \times 10^2 V + 0.000067800689280 \times 10^2 h + 0.000000001251828 \\
&\quad \times 10^2 V \cdot h - 0.000000006600229 \times 10^2 V^2 \\
&\quad - 0.000000000504206 \times 10^2 h^2, \\
\mathbf{B}_{31} &= -0.416859626570442 \times 10^{-3} + 0.000021770424271 \\
&\quad \times 10^{-3} V + 0.000008073449568 \times 10^{-3} h \\
&\quad + 0.000000004341692 \times 10^{-3} V \cdot h - 0.000000020925782 \\
&\quad \times 10^{-3} V^2 - 0.000000000294067 \times 10^{-3} h^2, \\
\mathbf{B}_{32} &\approx -0.000000000019683 \times 10^{-5} - 0.145625062719996 \times 10^{-5} V, \\
\mathbf{B}_{51} &= 0.149606059591807 - 0.000002276698805V \\
&\quad - 0.000001007292996h - 0.000000000224446V \cdot h \\
&\quad + 0.000000000937140V^2 + 0.000000000023361h^2, \\
\mathbf{B}_{52} &= -0.000000167960940 \times 10^{-7} + 0.000000000036265 \\
&\quad \times 10^{-7} V + 0.000000000000177 \times 10^{-7} h \\
&\quad - 0.251323593957002 \times 10^{-7} V^2.
\end{aligned} \tag{7}$$

2.2. Polytopic LPV Model Transformation. Based on the LPV model of the hypersonic vehicle longitudinal elastic nonlinear model, the stability analysis and controller synthesis problem of the system can be transformed into a convex optimization problem within a set of LMI frameworks which depend on the scheduling parameters. If it can be ensured that the LMI is feasible on the entire continuous variable parameter trajectory, the resulting controller can guarantee the global stability of the closed-loop system and meet the desired performance index. Since the spatial continuity of

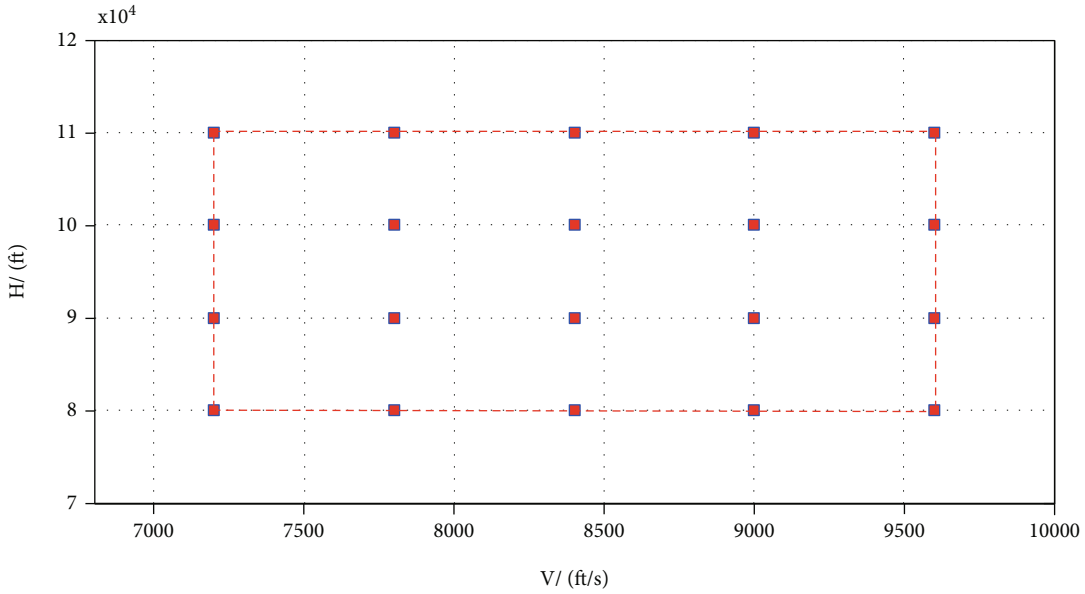


Figure 1: Scheduling parameter mesh division.

the scheduling parameters means that an infinite number of LMIs are to be solved over the entire parameter trajectory, which is obviously difficult to achieve, the scheduling parameters can be spatially meshed to solve the problem of LMI at a finite number of mesh vertices but the performance of this method is more dependent on the mesh density; besides, the determination of the mesh density not only lacks theoretical proof but also is time-consuming. The polytopic set is a convex set, which could describe the entire system. Through transformation of the polytopic model, the problem of solving infinitely many LMI problems can be attributed to solving the LMI problem at a limited number of vertices, which can greatly reduce the amount of computation. The controller obtained by this method has continuous gain throughout the workspace and has global action features ([11] [12, 22]).

The system equation (6) can be expressed as follows.

$$S(\rho(t)) \coloneqq [\mathbf{A}(\rho(t)) \quad \mathbf{B}(\rho(t))]. \tag{8}$$

where $S(\rho(t))$ is a parameter-varying object and $\rho(t) \in \Omega$ is a time-varying N dimensional parameter vector within the closed hypercube $\Omega = [a_1\ b_1] \times [a_2\ b_2] \times \cdots \times [a_N\ b_N] \subset \mathbb{R}^N$. Parameter $\rho(t)$ can also include some elements of $\mathbf{x}(t)$. The $(N+2)$-dimensional tensor S is constructed from LTI vertex systems $S_{i_1 i_2, \cdots, i_N}$.

$$\dot{\mathbf{x}}(t) = \widehat{S} \overset{N}{\underset{n=1}{\otimes}} A_n(\rho_n(t)) \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix}, \tag{9}$$

where the row vector $A_n(\rho_n(t)) \in \mathbb{R}^{I_n}$ $(n = 1, \cdots, N)$ contains one bounded variable and continuous weighting functions $\alpha_{n,i_n}(\rho_n)$ $(i_n = 1 \cdots I_n)$. The weighting function $\alpha_{n,i_n}(\rho_n(t)) \in [0, 1]$ is the i_n weighting function defined on the nth dimension of Ω, and $\rho_n(t)$ is the nth element of vector $\rho(t)$ and $\sum_{i_n=1}^{I_n} \alpha_{n,i_n}(\rho_n(t)) = 1$. I_n $(n = 1, \cdots, N)$ denotes the number of the weighting functions used in the nth dimension of Ω. The symbol $\otimes$ denotes the Kronecker product. The tensor product model transformation method based on high-order singular value decomposition is an effective LPV system polytopic model transformation method. The brief process is as follows. Firstly, mesh the variable parameters in the LPV model, and then, discretize the system to form tensors. Finally, we perform high-order singular value decomposition. The related theory of high-order singular value decomposition can refer to the literature [23–25].

On the basis of the LPV model of the hypersonic vehicle in equation (6), the tensor product model transformation method is used to obtain the polyhedral description of the LPV model of the hypersonic vehicle. By uniformly dividing the altitude variable h and velocity variable V in the LPV model described in equation (6), the original system is approached in the specified area (velocity interval $[7200 \quad 9600]$ ft/s, altitude interval $[80000 \quad 110000]$ ft), where the number of sampling points is set to 200×200. Using the high-order singular value method decomposition, the singular value decomposition of the expansion matrix is arranged in a descending order, where $N = 2$, $I_1 = I_2 = 3$. Table 2 shows the top 5 largest singular values. All 6 (the remaining singular values are relatively small or equal to 0) of the singular values are preserved. The resulting numerical model is a convincing approximation of the original model, which has 9 polytopic vertex systems. The resulting approximation model is as follows:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \sum_{i=1}^{9} w_i(V, h) \mathbf{S}_i \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \\ &= \sum_{i=1}^{3} \sum_{j=1}^{3} w_i(V) w_j(h) \left(\mathbf{A}_{i,j} \mathbf{x}(t) + \mathbf{B}_{i,j} \mathbf{u}(t) \right) \\ &= \sum_{k=1}^{9} w_k(V, h) (\mathbf{A}_k \mathbf{x}(t) + \mathbf{B}_k \mathbf{u}(t)). \end{aligned} \tag{10}$$

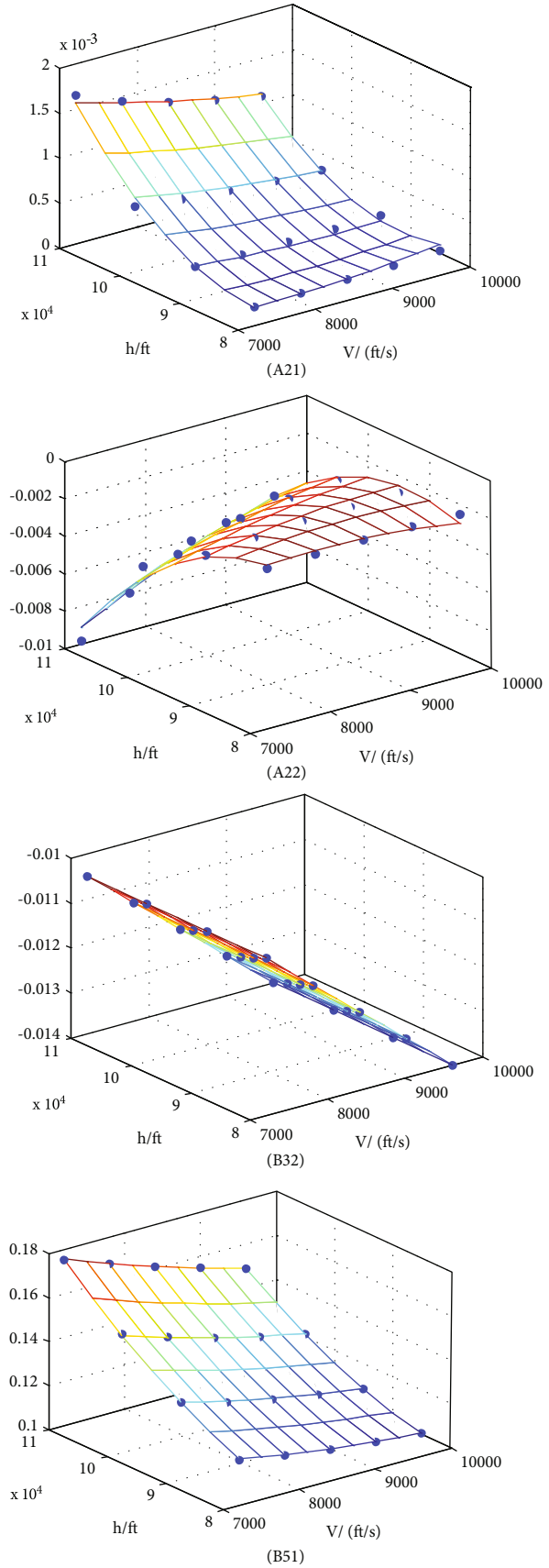


Figure 2: The fitting effect of LPV model parameter.

Table 2: HOSVD decomposition results of the LPV model.

Expand matrix	The 1st-order singular value	The 2nd-order singular value	The 3rd-order singular value	The 4th-order singular value	The 5th-order singular value
$\mathbf{A}_{(1)}$	$2.62023e+06$	81445.3	1850.49	$6.95697e-09$	$3.80140e-10$
$\mathbf{A}_{(2)}$	$2.62101e+06$	50610.4	916.243	$5.56458e-10$	$3.59899e-10$

In equation (10), $w_k(V,h) = w_i(V)w_j(h)$, $k \in \{1,2,\cdots,9\}$, $i,j \in \{1,2,3\}$. Based on the high-order singular value decomposition results, the maximum modeling error of the approximation model described in equation (8) can be calculated as 0.003679 and the average modeling error is 0.002499.

In order to verify the approximate equivalence between the hypersonic elastic nonlinear model described in equation (1) and the polytopic LPV model obtained by the transformation of the tensor product model described in equation (8), an equilibrium state in the parameter variation interval in Table 3 is used as an example. The system approximation matrix at the equilibrium point is obtained by the Jacobian linearization method which can locally approximate the elastic nonlinear model and the tensor product model transformation method in equation (8). The zero-pole distribution map of the system is shown in Figure 3.

Comparing the zero-pole positions of the two methods at the same equilibrium point, we can find that the zero-pole positions of the two methods are very close, which indicates that the polytopic LPV model based on the transformation of the tensor product model accurately reflects the dynamic characteristics of longitudinal elasticity nonlinear models of the hypersonic vehicle.

The shape of the time-varying weight function $w_k(V,h)$ ($k \in \{1,2,\cdots,9\}$) is shown in Figure 4.

The general distribution of the nine time-varying weight functions shows that when any time-varying weight is 1, the rest of the weights are 0, which indicates that the LPV model weight function obtained by the tensor product model transformation satisfies the requirement of convex coordinates.

3. Robust Controller Design

The LPV model of a hypersonic vehicle is as follows.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(\theta(t))\mathbf{x}(t) + \mathbf{B}(\theta(t))\mathbf{u}(t) + \mathbf{D}\mathbf{w}(t),\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t).\end{aligned} \tag{11}$$

In equation (11), $\theta(t) = [V\ h]^T$ denotes the scheduling variables, $A(\theta(t))$ and $B(\theta(t))$ both are the corresponding parameter-dependent system matrix, and the numerical expression is described in equation (6). $\mathbf{C}$ is the observation matrix, $\mathbf{D}$ is the interference matrix, and $\mathbf{w}(t)$ denotes time-varying exogenous disturbance. The reference velocity and altitude of the aircraft are recorded as $r(t) = [V_{\text{com}} \quad h_{\text{com}}]^T$ and the control output is recorded as $y(t) = [V_t \quad h_t]^T$. The controller design goal of the hypersonic vehicle not only needs to ensure the stability of the aircraft but also requires that the output of the aircraft can track a given command signal in real time. Therefore, the design goal of the hybrid target robust controller is to design an effective controller within the whole flight envelope ensuring that the output tracking error of the aircraft is 0.

Table 3: Trim condition of the hypersonic vehicle.

Status	Value	Status	Value	Status	Value
h	85000 ft	η_1	1.5122	Φ	0.2514
V	7202.0808 ft/s	$\dot{\eta}_1$	0	δ_e	11.463 deg
α	1.5153 deg	η_2	1.2144		
θ	1.5153 deg	$\dot{\eta}_2$	0		
Q	0 deg/s				

$$\lim_{t\longrightarrow\infty}[\mathbf{y}(t) - \mathbf{r}(t)] = 0. \tag{12}$$

In order to evaluate the system's steady-state error more effectively, the following tracking error integral term is introduced to eliminate the steady-state error in the feedback control strategy.

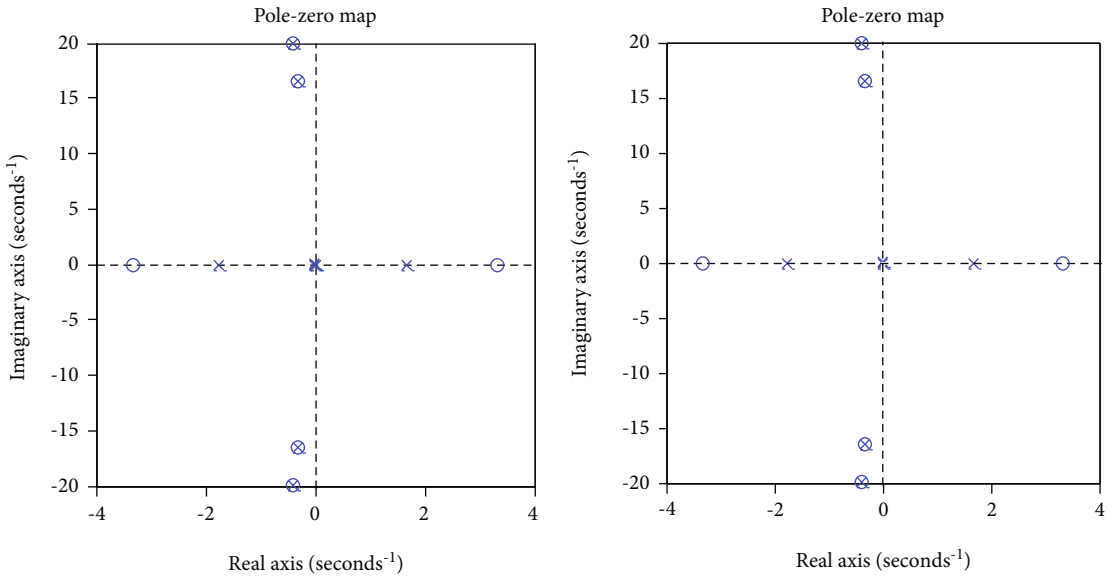
$$\mathbf{x}_e(t) = \int_0^t [\mathbf{y}(\tau) - \mathbf{r}(\tau)]\, d\tau. \tag{13}$$

According to the tracking error integral term and combining polytopic description and the LPV model of the hypersonic vehicle, the extended system can be obtained as follows.

$$\begin{aligned}\dot{\mathbf{X}}(t) &= \hat{\mathbf{A}}(\theta(t))\mathbf{X}(t) + \mathbf{B}_1\mathbf{r}(t) + \hat{\mathbf{B}}(\theta(t))\mathbf{u}(t) + \hat{\mathbf{D}}\mathbf{w}(t),\\ \mathbf{Y}(t) &= \hat{\mathbf{C}}\mathbf{X}(t),\end{aligned} \tag{14}$$

where

$$\begin{aligned}\mathbf{X}(t) &= [\mathbf{x}(t)\ \mathbf{x}_e(t)]^T,\\ \hat{\mathbf{A}}(\theta(t)) &= \begin{bmatrix}\mathbf{A}(\theta(t)) & 0\\ \mathbf{C} & 0\end{bmatrix} = \sum_{k=1}^{9} w_k(V,h)\hat{\mathbf{A}}_k,\\ \hat{\mathbf{B}}(\theta(t)) &= \begin{bmatrix}\mathbf{B}(\theta(t))\\ 0\end{bmatrix} = \sum_{k=1}^{9} w_k(V,h)\hat{\mathbf{B}}_k,\\ \hat{\mathbf{A}}_k &= \begin{bmatrix}\mathbf{A}_k & 0\\ \mathbf{C} & 0\end{bmatrix},\\ \hat{\mathbf{B}}_k &= \begin{bmatrix}\mathbf{B}_k\\ 0\end{bmatrix},\\ \hat{\mathbf{C}} &= [\mathbf{C} \quad 0]\\ \mathbf{B}_1 &= [0 \quad -\mathbf{I}]^T,\\ \hat{\mathbf{D}} &= [\mathbf{D}^T \quad 0]^T.\end{aligned} \tag{15}$$



(a) Jacobi linearization (b) Polytopic LPV model

Figure 3: Zero-pole comparison diagram.

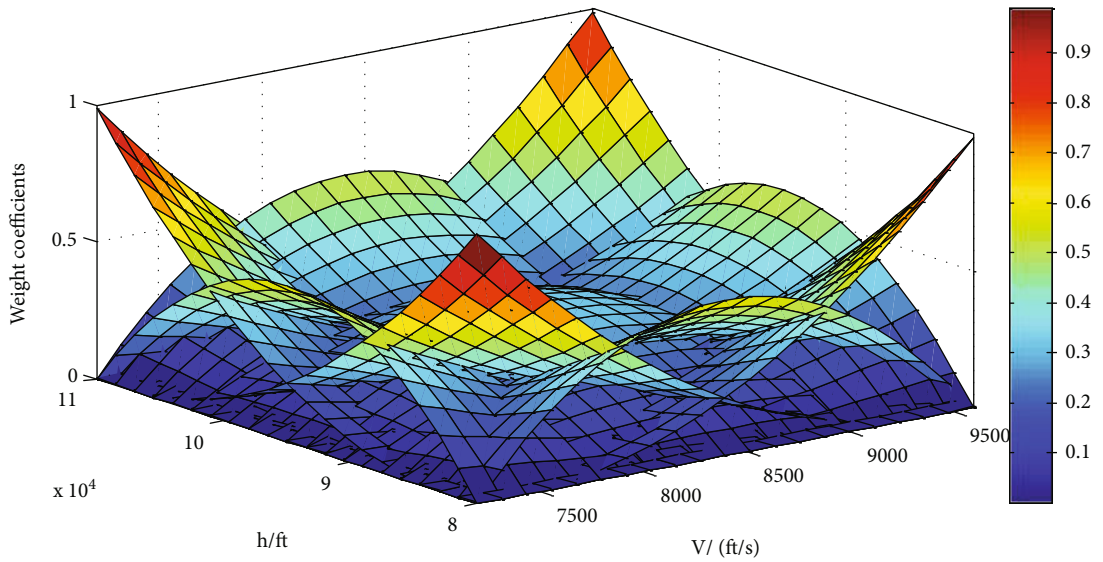


Figure 4: Time-varying weight function.

According to the parallel distributed compensation (PDC), the designed state feedback controller is as follows.

$$\mathbf{u}(t) = \sum_{k=1}^{9} w_k(V,h)\mathbf{K}_k\mathbf{X}(t) = \mathbf{K}(\theta(t))\mathbf{X}(t). \tag{16}$$

In equation (16), $\mathbf{K}(\theta(t)) = \sum_{k=1}^{9} w_k(V,h)\mathbf{K}_k$. Substituting equation (16) into (14), the expression for the closed-loop system of a hypersonic vehicle can be obtained as follows.

$$\begin{aligned}\dot{\mathbf{X}}(t) &= \left(\widehat{\mathbf{A}}(\theta(t)) + \widehat{\mathbf{B}}(\theta(t))\mathbf{K}(\theta(t))\right)\mathbf{X}(t) + \mathbf{B}_1\mathbf{r}(t) + \widehat{\mathbf{D}}\mathbf{w}(t)\\ &= \sum_{i=1}^{9}\sum_{j=1}^{9} w_i(V,h)w_j(V,h)\left(\widehat{\mathbf{A}}_i + \widehat{\mathbf{B}}_i\mathbf{K}_j\right)\mathbf{X}(t) + \mathbf{B}_1\mathbf{r}(t) + \widehat{\mathbf{D}}\mathbf{w}(t).\end{aligned} \tag{17}$$

The output tracking problem of the hypersonic vehicle is transformed into the robust stability problem of the above extended system. When the closed-loop system described in equation (17) is robustly stable, the command tracking target of the original system described in equation (9) can be achieved.

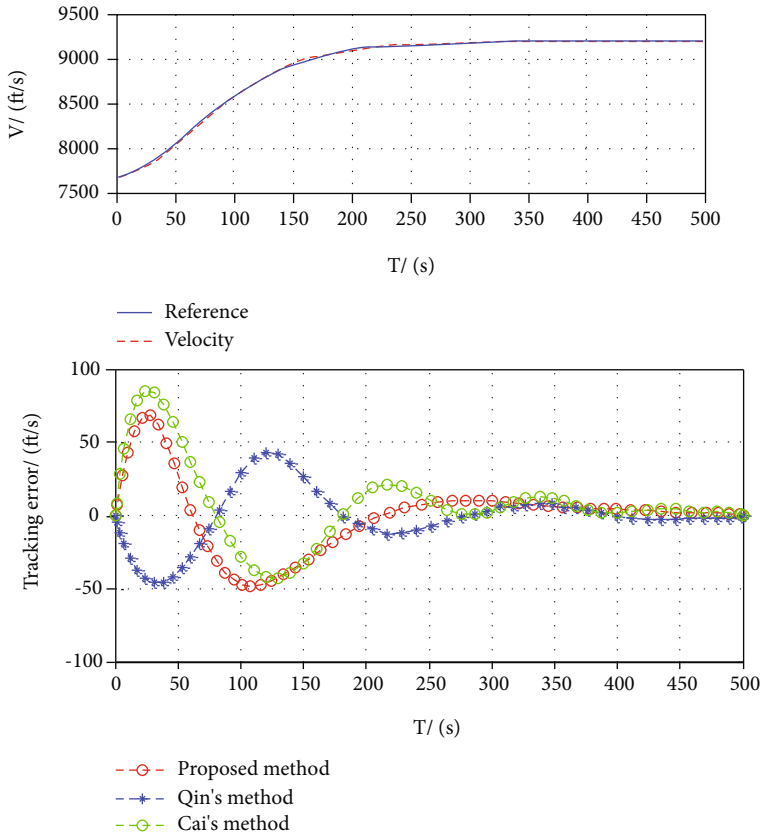


Figure 5: Velocity tracking performance and tracking error.

4. Numerical Simulation

Under conditions of cruising flight of the hypersonic vehicle (altitude 85000 ft, velocity 7702 ft/s), a robust model predictive controller for state feedback of mixed performance indicators was designed to achieve tracking control of velocity commands and altitude commands [26]. In the actual control of the aircraft, the command must comprehensively consider the physical constraints of the aircraft such as structural stiffness and payload. In general, it is assumed that the given velocity and altitude reference command signals are generated by the second-order filters.

$$\mathbf{H}(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2}. \tag{18}$$

In equation (18), the velocity instruction signal filtering system natural frequency $w_V = 0.0195$ and damping coefficient $\zeta_V = 0.9$. The altitude command signal filtering system natural frequency $w_h = 0.0219$ and damping coefficient $\zeta_h = 0.9$. It is assumed that the initial equilibrium state of the longitudinal dynamics of the aircraft is as follows. The rigid state $V_0 = 7702$ft/s, $h_0 = 85000$ft, $\alpha_0 = 1.6262$ deg, $\theta_0 = 1.6262$ deg, and $Q = 0$. The elastic mode $\eta_1 = 0.9127$, $\dot{\eta}_1 = 0$, $\eta_2 = 0.7623$, and $\dot{\eta}_2 = 0$. The control input $\Phi = 0.2666$ and $\delta_e = 12.4978$ deg. The control input satisfies the hard constraints of the actuator; the engine fuel equivalence ratio and the range of the elevator declination angle are $0.1 \le \Phi \le 1.4$ and $-30 \le \delta_e \le 30$.

The discretization sampling time $\mathbf{T}_s$ of the continuous system is 0.01 s. The aerodynamic parameters in the aircraft model are mainly obtained by computational fluid dynamics simulations or wind tunnel experiments. Besides, the unknown external environment and complex dynamic influences such as dynamic pressure effects, viscous effects, lean gas effects, and atmospheric turbulence may be encountered in the flight process. Therefore, the abovementioned factors are assumed as uncertainties of model inertial parameters and aerodynamic parameters. Similar to the literature [27], the value of parameters lies within ±10% of the nominal value such as $m = m(1 + \triangle m)$ where $|\triangle m| \le 0.07$ and $\rho = \rho(1 + \triangle \rho)$ where $|\triangle \rho| \le 0.08$. Compared with the ones in Qin et al. [27], the uncertainties are more aggressive and energy-bounded external disturbance is considered. When the desired velocity $V_e = 9200$ft/s and desired altitude $h_e = 104000$ ft, the simulation results using the state feedback controller design method are shown in Figures 5–9. The simulation results are compared with the results obtained through Qin's method [27] and Cai's method [28]. In the sequence, results obtained through the proposed method, Qin's method, and Cai's method are depicted by a red line marked by circles, blue dashed line marked by circles, and green dashed line marked by asterisks. Since the tracking curves are quite close, thus, only the tracking errors are given.

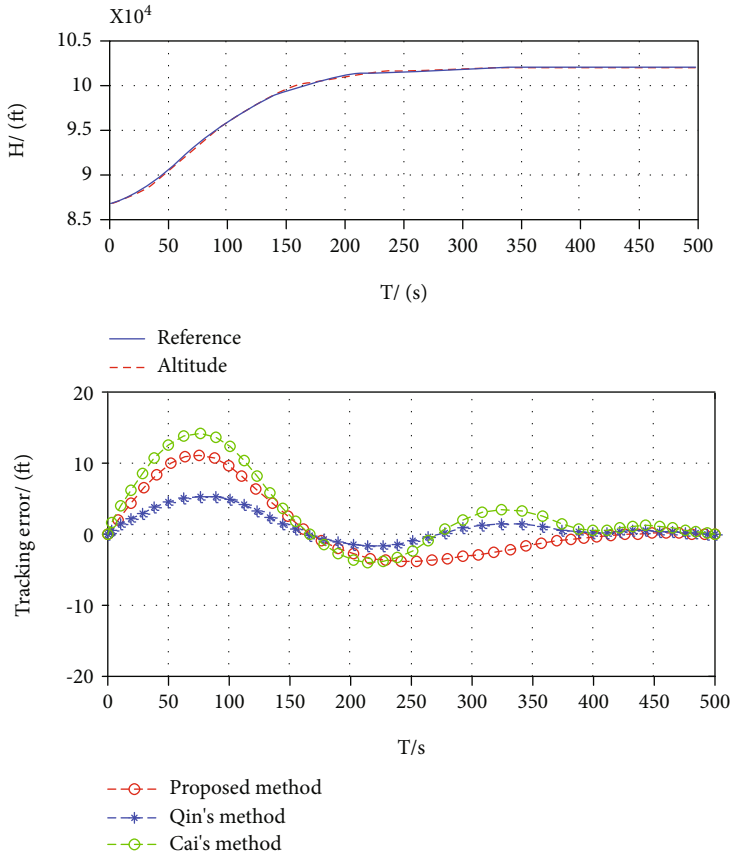


Figure 6: Altitude tracking performance and tracking error.

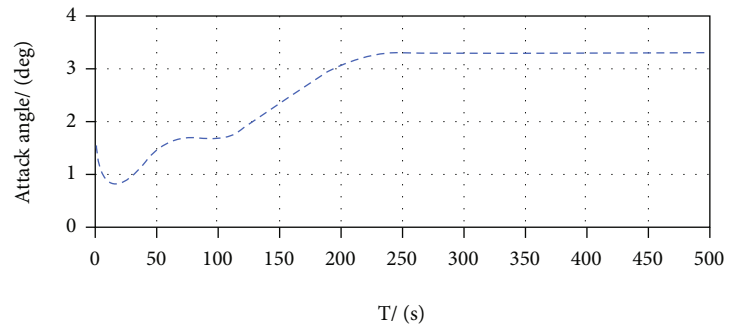


Figure 7: Angle of attack.

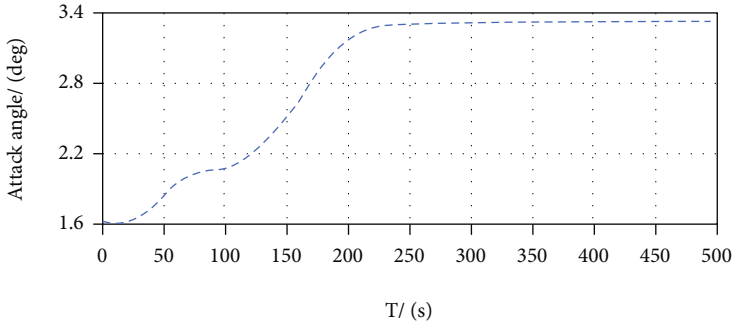


Figure 8: Pitch angle.

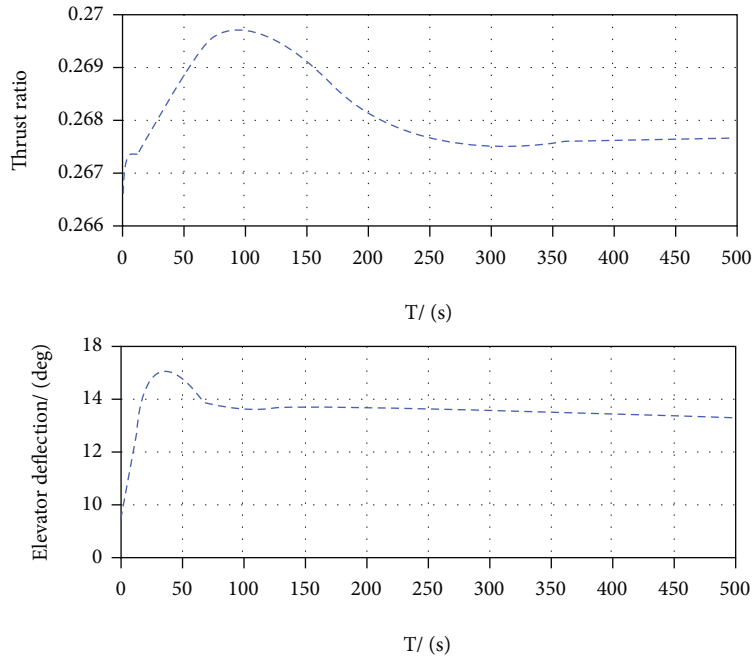


Figure 9: Control input.

In Figures 5 and 6, it can be seen that the robust controller can ensure that a hypersonic vehicle achieves the fast and accurate velocity and altitude command tracking during the maneuvering flight of the large flight envelope. The above controller has a smaller tracking error and a relatively smooth state, which can effectively suppress the influence of serious aerodynamic parameter uncertainties and elastic modes in the flight process. Figures 7 and 8 show the variation curves of the angle of attack and pitch angle in the whole process, respectively. In order to achieve a high degree of climb while increasing the velocity, the hypersonic vehicle in the initial trim state adds engine fuel input to the controller output signal and the elevator is pulled up until the stable command tracking is achieved again to reach the equilibrium state. The angle of attack and the pitch angle will be stabilized in the new trim state. The curve of the control input shown in Figure 9 indicates that the engine fuel equivalence ratio and the elevator deflection angle all satisfy the system control input hard constraints.

In summary, in the case of external disturbances, parameter uncertainties, and elastic modal couplings during the maneuvering flight, the designed controller has strong robustness and ensures that the system can quickly track the command signal.

5. Conclusion

Aiming at a type of hypersonic vehicle longitudinal elasticity model disclosed in the existing literature, altitude and velocity are selected as the gain-scheduled variables, through a series of sequential steps such as meshing within the range of parameter variation, finding equilibrium points, Jacobian linearization, and multivariate linear fitting; the continuous-time LPV model of the aircraft is established. Then, using the tensor product model transformation method, the general LPV model is transformed into a polytopic LPV system model with 9 vertices and the rationality of the LPV model is verified. At the same time, regarding the maneuvering flight command tracking problem for hypersonic vehicles, a state feedback controller is used to translate the command tracking control problem into a robust stability problem for the corresponding extended closed-loop system combined with the obtained polytopic LPV system model. Simulation results demonstrate that the obtained polytopic LPV model based on high-order singular value decomposition in this paper is effective in solving the maneuvering flight command tracking and attitude stabilization control of general complex nonlinear systems, such as hypersonic vehicles. However, since the linearization process lose some beneficial nonlinearities, some gain perturbations due to finite word length or time delays of state or input might bring unpredictable complexities. LPV nonfragile control of air-breathing hypersonic vehicles with time delays needs further studies.

Data Availability

The longitudinal elastic nonlinear model of the hypersonic vehicle used in this paper is mainly based on the winged-cone model provided by NASA's Langley Research Center [18, 19, 20], which takes into account the influence of the elastic modality of the aircraft.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgments

The research is partially supported by the Key Laboratory Fund under grant number 6142003190204 and the Special Scientific Research Program of the Department of Education of Shaanxi Province (Grant no. 20JK0728).

References

[1] L. Cao, S. Tang, and D. Zhang, "Fractional-order sliding mode control of air-breathing hypersonic vehicles based on linear-quadratic regulator," *Journal of Aerospace Engineering*, vol. 31, no. 3, 2018.

[2] Y. B. Ding, X. G. Wang, Y. L. Bai, and N. G. Cui, "An improved continuous sliding mode controller for flexible air-breathing hypersonic vehicle," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 14, pp. 5751–5772, 2020.

[3] L. G. Mu, X. M. Wang, R. Xie, Y. M. Zhang, B. Li, and J. Wang, "A survey of the hypersonic flight vehicle and its guidance and control technology," *Journal of Harbin Institute of Technology*, vol. 51, no. 3, pp. 1–14, 2019.

[4] X. Bu, "Envelope-constraint-based tracking control of air-breathing hypersonic vehicles," *Aerospace Science and Technology*, vol. 95, 2019.

[5] J. G. Guo, T. B. Zhang, C. Cheng, and J. Zhou, "Model reference adaptive attitude control for near space hypersonic vehicle with mismatched uncertainties," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 5, pp. 1301–1312, 2019.

[6] W. Jiang and H. L. Wang, *Recent Progress of Control Approaches for Air-Breathing Hypersonic Vehicles*, CAC, Wuhan, China, 2015.

[7] H. An, Q. Q. Wu, and C. H. Wang, "Scramjet operation guaranteed longitudinal control of air-breathing hypersonic vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 6, pp. 2587–2598, 2020.

[8] H. An, Q. Wu, G. Wang, Z. Guo, and C. Wang, "Simplified longitudinal control of air-breathing hypersonic vehicles with hybrid actuators," *International Journal of Control*, vol. 104, 2019.

[9] J. Mohammadpour and C. W. Scherer, *Control of linear parameter varying systems with applications*, Science & Business Media, New York, 2012.

[10] F. Wu, *Control of Linear Parameter Varying Systems*, University of California, Berkeley, 1995.

[11] Y. Q. Huang, C. Y. Sun, C. S. Qian, and L. Wang, "Non-fragile switching tracking control for a flexible air-breathing hypersonic vehicle based on polytopic LPV model," *Chinese Journal of Aeronautics*, vol. 26, no. 4, pp. 948–959, 2013.

[12] Y. Q. Huang, C. Y. Sun, C. Qian, J. Zhang, and L. Wang, "Polytopic LPV modeling and gain-scheduled switching control for a flexible air-breathing hypersonic vehicle," *Journal of Systems Engineering and Electronics*, vol. 24, no. 1, pp. 118–127, 2013.

[13] A. Nguyen, P. Chevrel, and F. Claveau, "LPV static output feedback for constrained direct tilt control of narrow tilting vehicles," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 2, pp. 661–670, 2020.

[14] D. Muniraj, J. M. Fry, and M. Farhood, "LPV control design for autonomous underwater vehicles using robustness analysis tools," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 236–241, 2019.

[15] L. X. Zhang, L. Nie, B. Cai, S. Yuan, and D. Wang, "Switched linear parameter-varying modeling and tracking control for flexible hypersonic vehicle," *Aerospace Science and Technology*, vol. 95, 2019.

[16] X. H. Yang, W. J. Lv, X. F. Wei, and C. F. Hu, "Fault-tolerant control based on LPV-robust model predictive control for hypersonic vehicle," in *Chinese Intelligent Systems Conference*, pp. 159–168, Springer, 2020.

[17] C. F. Hu, X. F. Wei, and Y. L. Ren, "Passive fault-tolerant control based on weighted LPV tube-MPC for air-breathing hypersonic vehicles," *International Journal of Control, Automation and Systems*, vol. 17, no. 8, pp. 1957–1970, 2019.

[18] R. F. Chavez and D. K. Schmidt, "Analytical aeropropulsive-aeroelastic hypersonic vehicle model with dynamic analysis," *Journal of Guidance, Control and Dynamics*, vol. 17, no. 6, pp. 1308–1319, 1994.

[19] M. A. Bolender and D. B. Doman, "A non-linear model for the longitudinal dynamics of a hypersonic air-breathing vehicle," *Journal of Spacecraft and Rockets*, vol. 44, no. 2, pp. 374–387, 2007.

[20] X. B. Zhang, Q. Zong, and F. L. Zeng, "Control-oriented modeling and analysis of hypersonic vehicle with coupled aerodynamic-propulsion- elastic," *Journal of Astronautics*, vol. 35, no. 5, pp. 528–536, 2014.

[21] X. X. Hu, L. G. Wu, C. H. Hu, and H. J. Gao, "A daptive sliding mode tracking control for a flexible air-breathing hypersonic vehicle," *Journal of the Franklin Institute*, vol. 349, no. 2, pp. 559–577, 2012.

[22] U. Shaked, "Improved LMI representation for the analysis and the design of continuous-time systems with polytopic type uncertainty," *IEEE Transactions on Automatic Control*, vol. 46, no. 4, pp. 652–656, 2001.

[23] B. Péter, Y. Yam, and V. Péter, *Tensor Product Model Transformation in Polytopic Model-Based Control*, CRC Press, Taylor & Francis Group, Boca Raton, 2014.

[24] B. Péter, *TP-Model-Transformation-Based-Control Design Frame-Works*, Springer, New York Dordrecht London, 2016.

[25] B. Takarics and V. B. Bálint, "Tensor product model-based robust flutter control design for the FLEXOP aircraft," *IFAC-Papers OnLine*, vol. 52, no. 12, pp. 134–139, 2019.

[26] W. Jiang, H. L. Wang, J. H. Lu, and Z. Xie, "HOSVD-based LPV modeling and mixed robust H_2/H_∞ control for air-breathing hypersonic vehicle," *Journal of Systems Engineering and Electronics*, vol. 27, no. 1, pp. 183–191, 2016.

[27] W. W. Qin, J. Y. Liu, G. Liu, B. He, and L. Wang, "Robust parameter dependent receding HorizonH∞Control of flexible air-breathing hypersonic vehicles with input constraints," *Asian Journal of Control*, vol. 17, no. 2, pp. 508–522, 2015.

[28] G. B. Cai, C. H. Hu, B. J. Yin, H. He, and X. Han, "Gain-scheduled controller synthesis for continuous-time polytopic LPV systems," *Mathematical Problems in Engineering*, vol. 2014, Article ID 972624, 14 pages, 2014.

Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 8609211, 20 pages
https://doi.org/10.1155/2021/8609211

Research Article

Interpolation-Based Modeling Methodology for Efficient Aeroelastic Control of a Folding Wing

Chengyu Yue and Yonghui Zhao

State Key Laboratory of Mechanics and Control of Mechanical Structures, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Correspondence should be addressed to Yonghui Zhao; zyhae@nuaa.edu.cn

Received 11 August 2021; Accepted 28 October 2021; Published 20 November 2021

Academic Editor: Erdal Kayacan

The aeroelastic model of a folding wing varies with different configurations, so it actually represents a parameter-varying system. Firstly, a new approach based on interpolation of local models is proposed to generate the linear parameter-varying model of a folding wing. This model is capable of predicting the aeroelastic responses during the slow morphing process and is suitable for subsequent control synthesis. The underlying inconsistencies among local linear time-invariant (LTI) models are solved through the modal matching of structural modes and the special treatment of the rational functions in aerodynamic models. Once the local LTI models are represented in a coherent state-space form, the aeroservoelastic (ASE) model at any operating point can be immediately generated by the matrix interpolation technique. Next, based on the present ASE model, the design of a parameterized controller for suppressing the gust-induced vibration is studied. The receptance method is applied to derive fixed point controllers, and the effective independence method is adopted and modified for optimal sensor placement in variable configurations, which can avoid solving ill-conditioned feedback gains. Numerical simulation demonstrates the effectiveness of the proposed interpolation-based modeling approach, and the parameterized controller exhibits a good gust mitigation effect within a wide parameter-varying range. This paper provides an effective and practical solution for modeling and control of the parameterized aeroelastic system.

1. Introduction

Morphing aircraft has the ability to significantly change the shape or structure during flight so that a single aircraft can adapt to various mission scenarios [1]. Over the past decades, morphing aircraft has received widespread attention in the aviation industry and aircraft manufacturers. A variety of morphing concepts have emerged, including the changing airfoil camber, wingspan and twist morphing, and variable sweep angle [2, 3]. In 2003, the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory (AFRL), and the National Aeronautics and Space Administration (NASA) jointly launched the Morphing Aircraft Structures Program. Three contractors of this program, Lockheed Martin, NextGen Aeronautics, and Raytheon Missile Systems, respectively, proposed the concepts of a folding wing, flexible skin morphing wing, and telescoping wing [1]. Among them, the folding wing proposed by Lockheed Martin is an innovative morphing design that allows the structure to switch smoothly between the unfolded and folded configurations so that the optimal configuration can be adopted according to different mission requirements: the unfolded configuration is for efficient cruising, and the folded configuration is for high-speed diving [4].

Substantial changes in the structure and aerodynamic shape of the folding wings introduce specific aeroelastic behaviors that do not occur on traditional fixed wings [5]. Previous studies have provided a variety of modeling and analysis methods to investigate these aeroelastic behaviors. Lee and Weisshaar [6] used ZAERO software to generate a linear aeroelastic model of a folding wing and investigated the hinge stiffness effect on flutter dynamic pressure. Later on, Lee and Chen [7] considered freeplay nonlinearity at

the wing-fold hinge in this model and performed nonlinear flutter analysis to predict the limit cycle oscillation. Wang et al. [8] presented a general aeroelastic modeling method that uses the strip theory unsteady aerodynamic model and simplified structural model. This method can perform flutter analysis on folding wings with any number of wing segments. Zhao and Hu [9] proposed a parameterized aeroelastic modeling method, which enables rapid flutter analysis of a folding wing with different configurations. Tang and Dowell [10] introduced the component modal analysis method to efficiently and accurately derive the folding wing model. In order to explain the limit cycle oscillation measured in the experiment, Attar et al. [11] extended this model to include the geometric nonlinear effect. The amplitude and dominant response frequency of the limit cycle oscillation obtained by the computational analysis were in good agreement with the experimental results. The aforementioned parametric studies on folding wings were only applicable to the case of slow morphing, and the dynamic response calculation during the rapid morphing process needs to further consider the time-varying effects on the structural dynamic characteristics. Zhao and Hu [12] combined the Craig-Bampton substructure synthesis technique with the flexible multibody dynamic approach to investigate the transient responses of a folding wing during rapid morphing processes. Hu et al. [13] proposed an integrated model by incorporating the Kriging agent model of the unsteady aerodynamic force in the time domain with the flexible multibody structural model and calculated the aeroelastic responses of a folding wing during quasi-steady morphing processes. Later on, they used this technique to study the nonlinear aeroelastic response characteristics of a folding wing with cubic stiffness [14]. The shortcoming of the Kriging model they used is that the rational function expressions of the unsteady aerodynamic force are not unique, which brings risks to the use of agent-based interpolation. Recently, Verstraete et al. [15] built a simulating system of a folding wing, which used the unsteady vortex lattice method and the finite element method (FEM) to carry out the nonlinear aeroelastic analyses in multiple flight configurations.

It can be seen that a large body of work on folding wings focuses on flutter and dynamic response predictions. The involved modeling methods can perform aeroelastic analyses under fixed or time-varying parameters and can account for miscellaneous nonlinear effects. Due to the large variation of the folding wing configuration, the time-varying morphing process, and the nonlinearity in the structure or aerodynamics, most of the time-varying nonlinear modeling techniques are quite complicated and generally far away from the control-oriented modeling. Some studies on the morphing aircraft control trend to use fixed configuration controllers as a compromise [16–18]. When the wing shape changes, the controller needs to switch online between different parameters to ensure the stability and performance of the closed-loop system. Another promising solution is to describe the nonlinear aeroelastic system as a linear parameter-varying (LPV) model that approximately captures the complex behavior during the morphing process. The LPV model simplifies the nonlinear dynamic equations of the folding wing, especially the controller can be designed in the linear system theory framework [19, 20]. Theoretically, linearizing a nonlinear model around equilibrium points in the parameter space can directly generate an LPV model [21] or a set of linear time-invariant (LTI) models for interpolation [22]. In practice, however, the original nonlinear model may be completely opaque or overly complex, making this method difficult to implement [23]. Another approach is to use the global identification or local modeling technique [24]. The former allows to generate the LPV model in a single step, but it requires the control inputs and operating conditions to be constantly changing in one experiment; such experimental conditions do not always exist in practice. The latter is based on a set of LTI models that are estimated under different fixed parameters, and the LPV model is obtained by interpolating these local LTI models. Since interpolation can be efficiently implemented in real time, the local modeling technique is currently the most convenient and effective method to establish the LPV model of the aeroelastic system [25].

When using the local modeling technique, direct interpolation of the system matrices is usually infeasible. This is because the state-space representation of a certain system is not unique, which means that the state-space matrices may be expressed in an incoherent form at different parameter points. The common solution is to convert the state-space models into a coherent form through the state coordinate transformation. Two possible canonical forms can be chosen: the companion form [23] and the modal form [26]. However, the companion transformation requires the controllability of the system inputs, and this form is known to be poorly conditioned for large-scale systems [27]. In another solution, using the modal form requires pairing the decomposed modal matrices of the local models. Most existing methods cannot handle systems with more than one parameter and require the additional assumption that the number of complex and real poles does not change over the parameter range [28, 29]. There are other solutions for coherent representation of the local state-space models, including the balanced realization [30], the least-squares approximation [31], and the SMILE technique proposed by de Caigny et al. [32]. These methods more or less have problems such as relying on experience, being difficult to implement, or having harsh conditions.

A suitable aeroservoelastic open-loop model is of great significance for studying the aeroelastic control of folding wings. In order to solve the LPV modeling problem, this paper proposes a practical local modeling technique for a typical folding wing. This approach does not require the difficult coordinate transformation of state-space LTI models (usually necessary for interpolation) but deals with the structural finite element model and the doublet lattice-based aerodynamic model in a targeted manner. Firstly, a general algorithm for structure modal matching is presented, which converts the modal matching problem into a standard linear sum assignment problem (LSAP). LSAPs are solved immediately by the Hungarian algorithm so that the local structural modes are aligned to continuously change with the folding angle. Then, the rational function approximation (RFA) results of the generalized aerodynamic force (GAF)

matrices are scaled to transform the local aerodynamic models into a coherent form suitable for interpolation. The above steps eliminate all possible inconsistencies in local models and ensure that the local system matrices are continuously dependent on the scheduling parameters. In this way, whether the scheduling parameters are folding angle or flight parameters (e.g., flow speed), the state-space representation naturally has a coherent form.

As the second task of this paper, the closed-loop analysis of active aeroelastic control for the folding wing is also studied through the present aeroservoelastic model. In order to suppress the gust-induced vibration at different configurations, a parameterized controller with a multi-input multioutput (MIMO) static output feedback structure is designed by using a receptance-based method. The receptance method was originally proposed by Ram and Mottershead [33, 34]. It provides a straightforward way for vibration control of linear systems through partial pole placement. Some studies have successfully used this method to suppress the unstable aeroelastic responses due to the flutter instability [35, 36], but its control effect on external disturbances has not been evaluated. The advantage of the receptance method is that the controller can be achieved only by transfer functions from the available sensors and actuators. Therefore, the tedious tasks of model order reduction and state observer design in modern control can be avoided. Moreover, the controller in the form of static output feedback has a simple structure, so it is easy to extend to the parameterized controller through interpolation of local controllers. In this work, an additional step of optimal sensor placement is introduced to find a proper sensor layout suitable for the variable configuration of the folding wing. For this purpose, we employed the effective independence method and modified it so that it can be used in the parameter-varying system. The sensor layout is optimized by iteratively constructing the independence distribution vector and eliminating the insignificant locations. As a result, the proper sensor layout avoids solving the ill-conditioned equations of the receptance method within the parameter range. Numerical examples demonstrate that the proposed modeling and control methods are effective and reliable for the parameterized aeroelastic system in variable configurations.

2. Description of the Parameterized Aeroelastic Modeling

The schematic diagram of a folding wing geometry is shown in Figure 1(a). The folding wing structure consists of three components: the fuselage, the inboard wing, and the outboard wing. The inboard wing and the outboard wing each have a trailing-edge control surface (see Figure 1(b)). These three substructures are connected by rotating hinges driven by the servomechanism. The folding angle θ between the inboard and the fuselage can be varied from 0 deg (fully unfolded configuration) to 120 deg (fully folded configuration). During the folding process, the outboard wing remains parallel to the x–y plane, as shown in Figure 1(c).

Obviously, the equations of motion of the folding wing depend on the parameter θ. Conventional nonparameterized FEM is only applicable for fixed structural configurations. When the folding angle changes, the finite element model has to be reestablished, which leads to a huge amount of repetitive modeling work. To solve this problem, Zhao and Hu [9] proposed an efficient substructure synthesis method to construct finite element models under different folding angles. It only requires modeling each substructure once. This paper adopts this method to establish the parameterized finite element model under arbitrary folding angles. To this end, the folding wing is divided into three substructures: the fuselage, the inboard wing, and the outboard wing. Finite element models of the substructures are established in their respective local coordinate systems, as shown in Figure 2. The four-node CQUAD4 elements in NASTRAN are used to discretize each substructure. For an arbitrary folding angle θ, a coordinate transformation is used to express each substructural model in the global coordinate system. Substructure synthesis is performed according to the compatibility conditions for the forces and displacements at the interfaces of the substructures. In this way, the finite element model of the entire structure at an arbitrary folding angle θ is quickly obtained. The synthesized structural model of the folding wing is expressed exactly as a regular FEM equation as follows:

$$\mathbf{M}_s\ddot{\mathbf{x}}_s + \mathbf{C}_s\dot{\mathbf{x}}_s + \mathbf{K}_s\mathbf{x}_s = \mathbf{f}_s + \mathbf{f}_g, \tag{1}$$

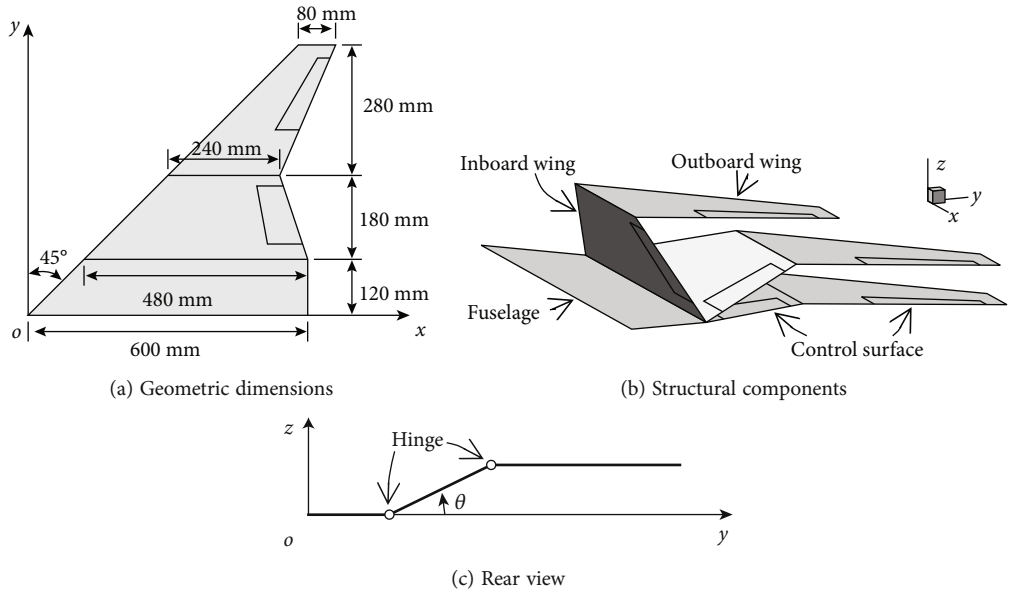
where $\mathbf{x}_s$ is the displacement vector in global coordinates. $\mathbf{M}_s$, $\mathbf{C}_s$, and $\mathbf{K}_s$ are the mass matrix, damping matrix, and stiffness matrix, respectively, all of which depend on the parameter θ. $\mathbf{f}_s$ and $\mathbf{f}_g$ are the unsteady aerodynamic forces induced by structural motion and gust disturbance, respectively.

The unsteady aerodynamic model is established by using the doublet lattice method (DLM) [37]. In general, the aerodynamic boxes are independent of the finite element meshes, but in this paper, the aerodynamic boxes of the folding wing coincide with the finite element meshes. The aerodynamic influence coefficient (AIC) matrix generated by DLM is used to calculate the pressure coefficients distributed on the lifting surface under simple harmonic motion, and then the equivalent aerodynamic forces acting on the structure are derived from the force and displacement transfer relationships between structural nodes and aerodynamic grid points. Based on this, the unsteady aerodynamic forces in Equation (1) have the following form [38]:

$$\mathbf{f}_s = q_\infty \mathbf{G}_{ks}^{\mathrm{T}} \mathbf{S}_{kj} \mathbf{A}_{aic}^{-1}(k, M_\infty) \mathbf{D}_{jk}(k) \mathbf{G}_{ks} \mathbf{x}_s, \tag{2}$$

$$\mathbf{f}_g = q_\infty \mathbf{G}_{ks}^{\mathrm{T}} \mathbf{S}_{kj} \mathbf{A}_{aic}^{-1}(k, M_\infty) \mathbf{\Phi}_{jg}(k) \frac{w_g}{U_\infty}, \tag{3}$$

where $\mathbf{A}_{aic}$ is the AIC matrix, which is a function of the reduced frequency $k = \omega b_R / U_\infty$ and the Mach number M_∞. ω is the angular frequency of harmonic vibration, b_R is the reference half chord length, and U_∞ is the flow speed. $q_\infty = 0.5\rho_\infty U_\infty^2$ is the dynamic pressure, and ρ_∞ is the air

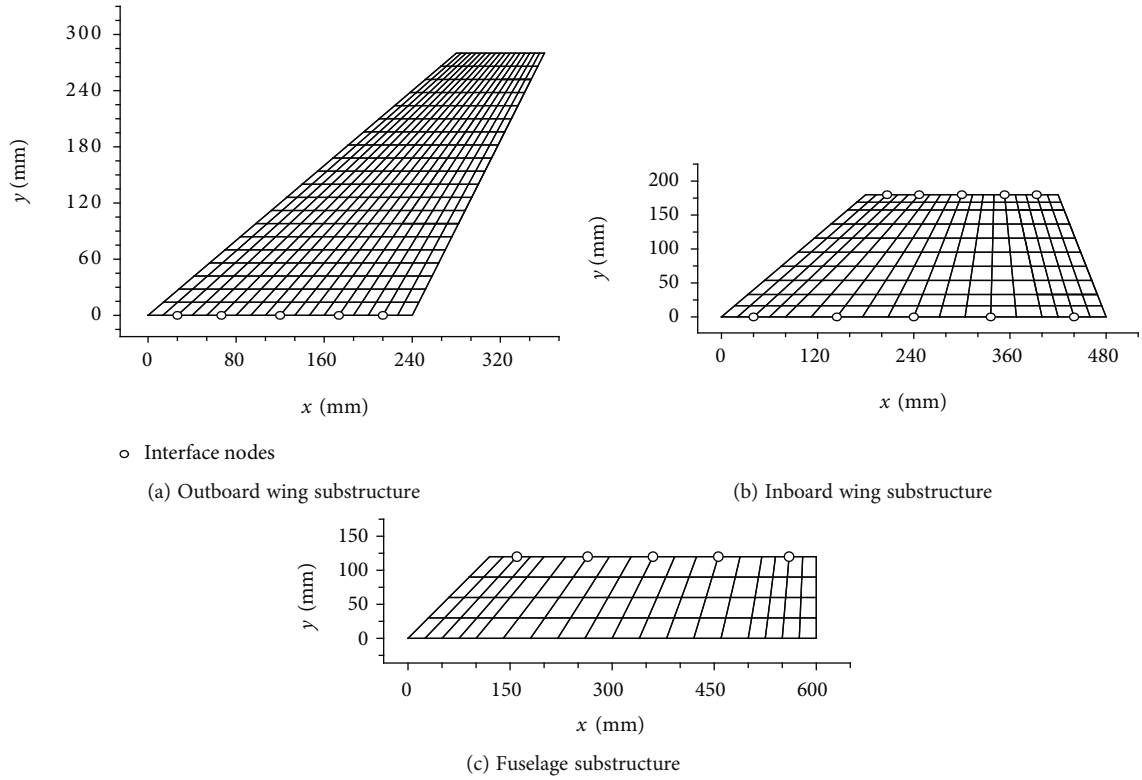


(a) Geometric dimensions

(b) Structural components

(c) Rear view

Figure 1: Schematic diagram of the folding wing.



(a) Outboard wing substructure

(b) Inboard wing substructure

(c) Fuselage substructure

Figure 2: Finite element models of the three substructures.

density. $\mathbf{S}_{kj}$ is the integration matrix used to concentrate the distributed pressures to the aerodynamic grid points. $\mathbf{D}_{jk}$ is the substantial differentiation matrix derived from the airflow boundary conditions. $\mathbf{G}_{ks}$ is the spline matrix obtained by the infinite plate spline method [39], which is used for mesh deformation interpolation and equivalent aerodynamic force calculation. $\mathbf{\Phi}_{jg}$ is the gust mode vector, including the time delay and amplitude ratio of the harmonic gust excitation on each aerodynamic box [40]. w_g is the gust velocity disturbance at the gust reference point, and the gust reference point is selected at the leading-edge of the fuselage, i.e., $x = 0$. It should be noted that in Equations (2) and (3), except that $\mathbf{D}_{jk}$ and $\mathbf{S}_{kj}$ are not related to the folding angle θ, the rest of the matrices depend on θ.

The structural motion can be expressed as the superposition of each modal motion. Taking the control surface deflection modes into consideration, the following modal coordinate transformation is introduced:

$$\mathbf{x}_s = [\boldsymbol{\Phi}_h \quad \boldsymbol{\Phi}_c] \begin{Bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\delta} \end{Bmatrix}, \tag{4}$$

where $\boldsymbol{\Phi}_h$ is the structure modal matrix and $\boldsymbol{\xi}$ is the vector of corresponding modal displacements. $\boldsymbol{\Phi}_c$ is the matrix of control surface modes, which contains rigid displacements due to the unit deflection of each control surface. $\boldsymbol{\delta}$ is the vector of actual deflections of the control surfaces.

Combining Equations (1)–(4), the aeroelastic equation of the folding wing in modal coordinates is obtained as follows:

$$\begin{aligned} &\mathbf{M}_{hh}\ddot{\boldsymbol{\xi}} + \mathbf{C}_{hh}\dot{\boldsymbol{\xi}} + \mathbf{K}_{hh}\boldsymbol{\xi} + \mathbf{M}_{hc}\ddot{\boldsymbol{\delta}} \\ &\quad = q_\infty \mathbf{Q}_{hh}\boldsymbol{\xi} + q_\infty \mathbf{Q}_{hc}\boldsymbol{\delta} + q_\infty \mathbf{Q}_{hg}\frac{w_g}{U_\infty}, \end{aligned} \tag{5}$$

where $\mathbf{M}_{hh}$, $\mathbf{C}_{hh}$, and $\mathbf{K}_{hh}$ are the modal mass, modal damping, and modal stiffness matrices in the diagonal form, respectively. $\mathbf{M}_{hc}$ is the mass coupling matrix, which represents the inertial effect induced by the control surfaces. $\mathbf{Q}_{hh}$, $\mathbf{Q}_{hc}$, and $\mathbf{Q}_{hg}$ are the frequency-domain GAF matrices due to structural modes, control surface modes, and gust modes, respectively. Matrices in Equation (5) are expressed as follows:

$$\begin{aligned} \mathbf{M}_{hh} &= \boldsymbol{\Phi}_h^T \mathbf{M}_s \boldsymbol{\Phi}_h, \\ \mathbf{C}_{hh} &= \boldsymbol{\Phi}_h^T \mathbf{C}_s \boldsymbol{\Phi}_h, \\ \mathbf{K}_{hh} &= \boldsymbol{\Phi}_h^T \mathbf{K}_s \boldsymbol{\Phi}_h, \\ \mathbf{M}_{hc} &= \boldsymbol{\Phi}_h^T \mathbf{M}_s \boldsymbol{\Phi}_c, \\ \mathbf{Q}_{hh} &= \boldsymbol{\Phi}_h^T \mathbf{G}_{ks}^T \mathbf{S}_{kj} \mathbf{A}_{aic}^{-1}(k, M_\infty) \mathbf{D}_{jk}(k) \mathbf{G}_{ks} \boldsymbol{\Phi}_h, \\ \mathbf{Q}_{hc} &= \boldsymbol{\Phi}_h^T \mathbf{G}_{ks}^T \mathbf{S}_{kj} \mathbf{A}_{aic}^{-1}(k, M_\infty) \mathbf{D}_{jk}(k) \mathbf{G}_{ks} \boldsymbol{\Phi}_c, \\ \mathbf{Q}_{hg} &= \boldsymbol{\Phi}_h^T \mathbf{G}_{ks}^T \mathbf{S}_{kj} \mathbf{A}_{aic}^{-1}(k, M_\infty) \boldsymbol{\Phi}_{jg}(k). \end{aligned} \tag{6}$$

So far, the parameterized aeroelastic model of the folding wing in modal coordinates has been initially established. All the structural and aerodynamic matrices in Equation (5) depend on parameter θ. For any fixed θ, system analysis and response calculation are easy to carry out. Note that the aerodynamic forces in the current model are expressed in the frequency domain. If time-domain aeroelastic analyses are desired, the GAF matrices can be transformed into the time domain through the RFA technique [41]. However, the current model is not suitable for simulation under the time-varying folding angle. It is extremely inefficient to perform online substructure synthesis, structure modal decomposition, and aerodynamic matrix calculation at each time step. Therefore, a modeling method for quickly obtaining an aeroelastic model at any point in the parameter space is desired.

3. Interpolation-Based Modeling Methodology

For the folding wing system, both the folding angle and the aerodynamic parameters (e.g., flow speed) will undergo variations. The response analysis and control synthesis are expected to be carried out under the LPV framework. Using a set of local models to compute an interpolating LPV model is a practical and efficient modeling strategy [25]. This method begins with the discretization of a given parameter space, which generates a set of grid points called operating points; then, the LTI models (local models) are prebuilt for fixed operating conditions at each operating point; finally, the LPV model is computed by interpolating these local LTI models. Although the local LTI models are obtained for fixed parameters and do not incorporate the time-varying effects, in case the parameter variations are slow relative to the dynamic characteristics of the system, the dynamic parameter-dependent part in the system can be ignored [32, 42]. Therefore, in this paper, while applying the local modeling technique, we assume that the folding process and flight conditions undergo slow and smooth variations.

Model interpolation requires that all local models have a coherent form; that is, the system matrix should change continuously over the considered parameter range. Note that Equation (1) is expressed in the physical coordinate system. The matrices in Equation (1) are continuously dependent on the folding angle θ. However, this is not the case in the modal coordinate system of Equation (5). It can be inferred that only the structure modal matrix $\boldsymbol{\Phi}_h$ may cause inconsistency. Since the mode crossing is a common phenomenon in the parameter-varying system, the structural modes with the same order cannot be directly interpolated because they often do not change continuously (do not belong to the same mode branch). Besides, the mode shapes may exhibit abrupt sign changes, which will also make interpolation impossible. Another inconsistency occurs in the subsequent RFA procedure for transforming the frequency-domain aerodynamic forces into the time domain. The commonly used minimum-state (MS) method generates nonunique coefficient matrices that cannot be directly interpolated. This section presents a structure modal matching algorithm and a coherent RFA representation method, which eliminates the underlying inconsistencies in local models and paves the way for model interpolation.

3.1. Modal Matching and Alignment. To perform the model interpolation, a set of local aeroelastic models of the folding wing are established under the n_t folding angles $\theta_1, \cdots, \theta_{n_t}$, and the following structural eigenvalue problems are solved:

$$\mathbf{K}_{s,l}\boldsymbol{\Phi}_{h,l} = \mathbf{M}_{s,l}\boldsymbol{\Phi}_{h,l}\boldsymbol{\Omega}_{h,l}^2, \quad l = 1, \cdots, n_t, \tag{7}$$

where

$$\begin{aligned} \boldsymbol{\Omega}_{h,l} &= \operatorname{diag}\left(\omega_{1,l} \quad \cdots \quad \omega_{n_h,l}\right), \\ \boldsymbol{\Phi}_{h,l} &= \left[\boldsymbol{\varphi}_{1,l} \quad \cdots \quad \boldsymbol{\varphi}_{n_h,l}\right]. \end{aligned} \tag{8}$$

$\mathbf{\Omega}_{\mathrm{h},l}$ is a diagonal matrix whose elements are angular natural frequencies of the structure. The column vectors in matrix $\mathbf{\Phi}_{\mathrm{h},l}$ are structural mode shapes normalized for unit-generalized masses. n_{h} is the number of the retained structural modes.

Typical eigenvalue solvers generally sort structural modes by natural frequencies in ascending order. When the mode crossing occurs, modes in matrix $\mathbf{\Phi}_{\mathrm{h},l}$ should be rearranged before interpolation to ensure continuous change between parameter points. Considering that the structural modes in the same branch have similar vibration patterns, we can compare the natural frequencies and mode shapes under the adjacent parameters and rearrange the orders to match these dynamic properties. For the two adjacent folding angles θ_{l-1} and θ_l, define the following distance metric to measure the dynamic similarity between the two modes:

$$d_{i,j} = \left|\omega_{i,l-1} - \omega_{j,l}\right| \cdot \left(1 - \sqrt{\mathrm{MAC}_{i,j}}\right), \quad i,j = 1, \cdots, n_{\mathrm{h}}, \quad (9)$$

where the modal assurance criterion (MAC) is calculated as

$$\mathrm{MAC}_{i,j} = \frac{\left|\boldsymbol{\varphi}_{i,l-1}^{\mathrm{T}} \boldsymbol{\varphi}_{j,l}\right|^2}{\left(\boldsymbol{\varphi}_{i,l-1}^{\mathrm{T}} \boldsymbol{\varphi}_{i,l-1}\right) \cdot \left(\boldsymbol{\varphi}_{j,l}^{\mathrm{T}} \boldsymbol{\varphi}_{j,l}\right)}. \quad (10)$$

The distance metric uses the linear distance of two natural frequencies and is weighted by the MAC. MAC takes value in the interval $[0, 1]$; a larger MAC value indicates a higher degree of linear correlation between the associated mode shapes. Assuming that the modal data at point $l-1$ has been sorted in a proper order, the goal of modal matching is to compare the modal data at point l with point $l-1$ and give a correct pairing by minimizing the total distance metric. From the perspective of the graph theory, the structural modes to be investigated can be naturally divided into the following two sets of vertices:

$$\begin{aligned} V_{l-1} &= \left\{ \omega_{1,l-1} \quad \cdots \quad \omega_{n_{\mathrm{h}},l-1} \right\}, \\ V_l &= \left\{ \omega_{1,l} \quad \cdots \quad \omega_{n_{\mathrm{h}},l} \right\}. \end{aligned} \quad (11)$$

Define the edge set $E_l = V_{l-1} \times V_l$; that is, any two vertices in V_{l-1} and V_l are connected by a unique edge, thus forming a complete bipartite graph $G_l = (V_{l-1}, V_l; E_l)$. The distance metric $d_{i,j}$ given in Equation (9) specifies the cost of each edge. The modal matching problem is reformulated as finding a perfect matching in the graph G_l, which minimizes the sum of the costs. This problem is a typical LSAP and can be solved by the Hungarian algorithm, which is a mature solution in the field of multiobject tracking [43]. The Hungarian algorithm was originally presented by Kuhn [44]. An improved version based on the shortest augmenting path technique [45] is currently more popular because it achieves an $O(n_{\mathrm{h}}^3)$ worst-case time complexity. Therefore, the modal matching problem can be efficiently and completely solved.

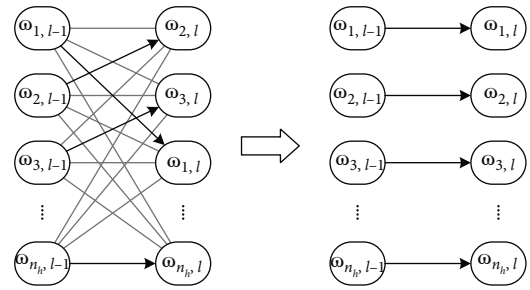


Figure 3: Illustration of the modal matching and alignment procedure. Gray lines: distance metrics between each adjacent mode. Black arrow lines: mode pairing with minimum total cost.

Figure 3 briefly illustrates the modal matching and alignment procedure. According to the modal matching results, the precalculated natural frequencies and modal vectors are aligned (reordered) successively for $l = 2, \cdots, n_{\mathrm{t}}$. In order to ensure numerical continuity, each aligned mode shape vector should be adjusted to $c_{i,l} \cdot \boldsymbol{\varphi}_{i,l}$, where the coefficient $c_{i,l}$ is calculated by a signum function as

$$c_{i,l} = \mathrm{sign}\left(\boldsymbol{\varphi}_{i,l-1}^{\mathrm{T}} \boldsymbol{\varphi}_{\mathrm{i},l}\right), \quad i = 1, \cdots, n_{\mathrm{h}}. \quad (12)$$

After modal matching and alignment, the set of aeroelastic models established by Equation (5) have a coherent form under all folding angles. Next, a method to construct coherent time-domain models is given.

3.2. Coherent RFA Representation. The GAF matrices in Equation (5) are expressed in the frequency domain. To obtain the time-domain aeroelastic model, the frequency-domain aerodynamic forces need to be converted to the Laplace domain through the RFA technique. The RFA technique uses the tabulated GAF data at several discrete reduced frequencies to fit a specific rational function of the Laplace variable s. The most commonly used MS method gives a general expression of RFA as [41]

$$\mathbf{Q}(p) = \mathbf{A}_0 + \mathbf{A}_1 p + \mathbf{A}_2 p^2 + \mathbf{D}(p\mathbf{I} - \mathbf{R})^{-1}\mathbf{E}p, \quad (13)$$

where $p = sb_{\mathrm{R}}/U_\infty$ is the nondimensional Laplace variable and $\mathbf{Q}$ is the combination of the following three GAF terms:

$$\mathbf{Q}(p) = \begin{bmatrix} \mathbf{Q}_{\mathrm{hh}}(p) & \mathbf{Q}_{\mathrm{hc}}(p) & \mathbf{Q}_{\mathrm{hg}}(p) \end{bmatrix}. \quad (14)$$

Matrix $\mathbf{R}$ in Equation (13) is a diagonal matrix containing manually specified negative numbers called the aerodynamic lag roots, and the other coefficient matrices are solved by the nonlinear least-squares iterative procedure. Although the fitting algorithm (irrespective of the way it is implemented) gives an explicit RFA expression, the $\mathbf{D}$ and $\mathbf{E}$ matrices in the lag term are not unique. Specifically, for any nonsingular matrix $\mathbf{T}$, the following equality always holds:

$$\mathbf{D}(p\mathbf{I} - \mathbf{R})^{-1}\mathbf{E} = \mathbf{D}\mathbf{T}\left(p\mathbf{I} - \mathbf{T}^{-1}\mathbf{R}\mathbf{T}\right)^{-1}\mathbf{T}^{-1}\mathbf{E}. \quad (15)$$

When building the parameterized system model, RFAs are performed independently at each parameter point. The nonuniqueness of the RFA expression brings inconsistency between local aerodynamic models, which will lead to incorrect interpolation results. In fact, the right side of Equation (15) contains all possible expressions. It can be further proved that $\mathbf{T}^{-1}\mathbf{R}\mathbf{T} = \mathbf{R}$ if and only if $\mathbf{T}$ is a diagonal matrix. Therefore, considering the prescribed matrix $\mathbf{R}$, the uncertainty of the RFA expression is limited to the case where $\mathbf{T}$ is a diagonal matrix, and the nonuniqueness is manifested in the arbitrary scaling of the column vectors of $\mathbf{D}$ and the row vectors of $\mathbf{E}$. Assuming that the matrix $\mathbf{R}$ is consistent or changes continuously with the parameters, then a unique scaling matrix $\mathbf{T}$ can be constructed according to the RFA results under adjacent parameters so that the unique and coherent RFA expression can be obtained.

Consider the case where the folding angle is the variable parameter. For the two adjacent folding angles θ_{l-1} and θ_l, take the matrix $\mathbf{D}_{l-1}$ at point $l-1$ as the standard reference, and the diagonal matrix $\mathbf{T}_l$ at point l is constructed as follows:

$$\mathbf{T}_l = \operatorname{diag}\left(t_{1,l} \quad \cdots \quad t_{n_a,l}\right), \tag{16}$$

where n_a is the number of aerodynamic lag roots and the diagonal element $t_{i,l}$ is calculated as follows:

$$t_{i,l} = \operatorname{sign}\left(\mathbf{d}_{i,l-1}^{\mathrm{T}}\mathbf{d}_{i,l}\right) \cdot \frac{\left\|\mathbf{d}_{i,l-1}\right\|_2}{\left\|\mathbf{d}_{i,l}\right\|_2}, \quad i = 1, \cdots, n_a, \tag{17}$$

where $\mathbf{d}_{i,l-1}$ and $\mathbf{d}_{i,l}$ are the i-th column of matrices $\mathbf{D}_{l-1}$ and $\mathbf{D}_l$, respectively. After obtaining $\mathbf{T}_l$, adjust the matrices $\mathbf{D}_l$ and $\mathbf{E}_l$ according to Equation (15). The corresponding column vectors of the scaled matrix $\mathbf{D}_l$ and the reference matrix $\mathbf{D}_{l-1}$ thus have the same 2-norm and consistent direction.

When the folding angle and Mach number are variable parameters, the RFA data should be generated on the two-dimensional parameter grid points. In this case, the processing method for coherent RFA representation should be as follows: first, fix the first Mach number and successively adjust the coefficient matrices for the sequential folding angles; then, fix each folding angle and successively adjust the coefficient matrices for the sequential Mach numbers in the same way.

After obtaining a set of coherent local RFAs, the state-space representation of the LPV system can be generated by combining Equation (13) with the aeroelastic equation given in Equation (5), as shown below:

$$\dot{\mathbf{x}} = \mathbf{A}(\boldsymbol{\rho})\mathbf{x} + \mathbf{B}(\boldsymbol{\rho})\mathbf{u} + \mathbf{B}_{\mathrm{w}}(\boldsymbol{\rho})\tilde{\mathbf{w}}_{\mathrm{g}}, \tag{18}$$

$$\mathbf{y} = \mathbf{C}(\boldsymbol{\rho})\mathbf{x} + \mathbf{D}_{\mathrm{w}}(\boldsymbol{\rho})\tilde{\mathbf{w}}_{\mathrm{g}}, \tag{19}$$

where $\boldsymbol{\rho}$ is the vector of scheduling parameters including the folding angle θ. The state vector $\mathbf{x}$ contains modal displacement vector $\boldsymbol{\xi}$ and its time derivative $\dot{\boldsymbol{\xi}}$, as well as the augmenting aerodynamic states introduced by the RFA. The motion of the control surfaces is generally determined by the actuator model, and the above state-space equation also integrates the actuator equation of the state-space form [46] so that the control input vector $\mathbf{u}$ represents the control surface deflection commands, while the actual deflections in $\boldsymbol{\delta}$ are incorporated into $\mathbf{x}$ as the augmenting states. The gust input vector $\tilde{\mathbf{w}}_{\mathrm{g}}$ contains gust velocity w_{g} and its time derivative $\dot{w}_{\mathrm{g}}$. There are various standard methods for generating gust disturbances [47]. Two types of gust models, the 1-cos discrete gust and the Dryden continuous gust, are used in the following time-domain simulations. The system output vector $\mathbf{y}$ may include responses at the FEM nodes (monitoring points) and/or internal loads (bending moment, etc.). For more details about constructing the state-space matrices, see Ref. [48].

Taking the fixed altitude flight as an example, the scheduling parameters in $\boldsymbol{\rho}$ thus consist only of the folding angle θ and the flow speed U_∞. Depending on the local modeling approach, the LPV model shown in Equations (18) and (19) is established by local LTI models on the two-dimensional parameter grid. The methods to deal with structural modes and RFA coefficients proposed in this section successfully solved the possible inconsistencies in the local models. Whether it is the change of the folding angle or the flow speed, the local models are guaranteed to be numerically continuous. In this way, by interpolating the state-space matrices, the model at any point in the parameter space can be quickly obtained, and the LPV simulation and control synthesis can also be performed efficiently.

4. Receptance-Based Active Aeroelastic Control

In this section, the receptance method [33, 34] is applied to design a parameterized controller for the folding wing, which is expected to reduce the structural vibration and additional loads induced by gust disturbances. This method uses the receptance transfer function extracted from the analysis model or the identified model to design the controller and achieves the active vibration control through partial pole placement. The receptance-based controller is theoretically solved under fixed parameters, and the control strategy under variable parameters can be realized by interpolation of the local controllers. Compared with the standard LPV control design such as gain-scheduled control [49, 50], interpolation of the fixed point controllers is easy to implement, but the system performance cannot be guaranteed when parameters change rapidly. However, under the assumption of slow parameter variation, the interpolation approach can avoid introducing the conservativeness involved in the LPV approaches and thus obtain a better control effect than the gain-scheduled control [51].

In order to avoid blindly selecting the sensor locations, an additional step of optimal sensor placement is introduced in the control design strategy. As will be seen below, this step also avoids solving the ill-conditioned equations of the receptance method.

4.1. Receptance-Based Control. The receptance-based controller is more suitable to be derived using the frequency-

domain equation. Converting Equation (5) into the Laplace domain, we obtain

$$\begin{aligned}&\left(s^2\mathbf{M}_{\mathrm{hh}}+s\mathbf{C}_{\mathrm{hh}}+\mathbf{K}_{\mathrm{hh}}-q_\infty\mathbf{Q}_{\mathrm{hh}}(s)\right)\boldsymbol{\xi}(s)\\&=\left(-s^2\mathbf{M}_{\mathrm{hc}}+q_\infty\mathbf{Q}_{\mathrm{hc}}(s)\right)\boldsymbol{\delta}(s)+\mathbf{d}(s),\end{aligned}\tag{20}$$

where $\mathbf{d}(s)\in\mathbb{C}^{n_{\mathrm{h}}}$ denotes the gust disturbances. Generally, the dynamic model of the actuators driving the control surfaces is represented by a specific transfer function as

$$\boldsymbol{\delta}(s)=\mathbf{H}_{\mathrm{c}}(s)\mathbf{u}(s),\tag{21}$$

where $\mathbf{u}(s)\in\mathbb{C}^{n_{\mathrm{c}}}$ denotes the actuator commands and $\boldsymbol{\delta}(s)\in\mathbb{C}^{n_{\mathrm{c}}}$ stands for the actual deflections of the control surfaces. The output equation of the system is

$$\mathbf{y}(s)=\boldsymbol{\Phi}_{\mathrm{s}}\boldsymbol{\xi}(s),\tag{22}$$

where $\mathbf{y}(s)\in\mathbb{C}^{n_{\mathrm{s}}}$ is the output vector and $\boldsymbol{\Phi}_{\mathrm{s}}\in\mathbb{R}^{n_{\mathrm{s}}\times n_{\mathrm{h}}}$ is the sensor modal matrix. Let $\mathbf{G}_{\mathrm{d}}$ and $\mathbf{G}_{\mathrm{v}}\in\mathbb{R}^{n_{\mathrm{c}}\times n_{\mathrm{s}}}$ be the proportional displacement and velocity output feedback gains to be solved, respectively. The feedback control law can be written as

$$\mathbf{u}(s)=(s\mathbf{G}_{\mathrm{v}}+\mathbf{G}_{\mathrm{d}})\mathbf{y}(s).\tag{23}$$

Set the control inputs and external disturbances to be zero and solve Equation (20) to obtain the eigenvalues λ_i and eigenvectors $\mathbf{w}_i$ of the open-loop system, expressed as

$$\left(\lambda_i^2\mathbf{M}_{\mathrm{hh}}+\lambda_i\mathbf{C}_{\mathrm{hh}}+\mathbf{K}_{\mathrm{hh}}-q_\infty\mathbf{Q}_{\mathrm{hh}}(\lambda_i)\right)\mathbf{w}_i=0,\quad i=1,\cdots,2n_{\mathrm{h}}.\tag{24}$$

Due to the nonlinearity of the $\mathbf{Q}_{\mathrm{hh}}$ term, Equation (24) does not express a typical quadratic eigenvalue problem. The number of eigenvalues is generally not equal to twice the number of the structural modes. Consider that Equation (24) is also the equation used in linear flutter analysis. The classical *p-k* method only finds n_{h} solutions (ignoring the complex conjugate) of the equation under each dynamic pressure, which is enough to reveal the governing modes of the aeroelastic system [52]. This paper uses the RFA method instead of the flutter analysis method to calculate the open-loop eigenvalues (more accurate unsteady aerodynamic forces), but only the $2n_{\mathrm{h}}$ eigenvalues closely related to the structural motion are retained.

The target for partial pole placement is to assign a part of closed-loop eigenvalues to the desired values μ_i, $i=1,\cdots m$, through the feedback gains, where the assigned eigenvalues are distinct from all open-loop eigenvalues; meanwhile, the rest of the closed-loop eigenvalues are unchanged, i.e., $\mu_i=\lambda_i$, $i=m+1,\cdots,2n_{\mathrm{h}}$. Substituting Equations (21)–(23) into Equation (20), the assigned eigenvalues μ_i and eigenvectors $\mathbf{v}_i$ of the closed-loop system should satisfy the following equation:

$$\mathbf{v}_i=\mathbf{H}(\mu_i)\mathbf{P}(\mu_i)(\mu_i\mathbf{G}_{\mathrm{v}}+\mathbf{G}_{\mathrm{d}})\boldsymbol{\Phi}_{\mathrm{s}}\mathbf{v}_i,\quad i=1,\cdots,m,\tag{25}$$

where $\mathbf{H}(\mu_i)$ and $\mathbf{P}(\mu_i)$ are the values of the receptance transfer function $\mathbf{H}(s)$ and the control channel transfer function $\mathbf{P}(s)$, respectively, defined as

$$\begin{aligned}\mathbf{H}(s)&=\left(s^2\mathbf{M}_{\mathrm{hh}}+s\mathbf{C}_{\mathrm{hh}}+\mathbf{K}_{\mathrm{hh}}-q_\infty\mathbf{Q}_{\mathrm{hh}}(s)\right)^{-1},\\\mathbf{P}(s)&=\left(-s^2\mathbf{M}_{\mathrm{hc}}+q_\infty\mathbf{Q}_{\mathrm{hc}}(s)\right)\mathbf{H}_{\mathrm{c}}(s).\end{aligned}\tag{26}$$

The following in this subsection describes how to solve the feedback gains. Introduce the weighting parameter vector $\boldsymbol{\alpha}_i\in\mathbb{C}^{n_{\mathrm{c}}}$ as follows:

$$\boldsymbol{\alpha}_i=(\mu_i\mathbf{G}_{\mathrm{v}}+\mathbf{G}_{\mathrm{d}})\boldsymbol{\Phi}_{\mathrm{s}}\mathbf{v}_i,\quad i=1,\cdots,m.\tag{27}$$

The value in vector $\boldsymbol{\alpha}_i$ indicates the degree of participation of each control input for controlling the *i*-th mode. Equation (25) can therefore be written as

$$\mathbf{v}_i=\mathbf{H}(\mu_i)\mathbf{P}(\mu_i)\boldsymbol{\alpha}_i,\quad i=1,\cdots,m.\tag{28}$$

When $\boldsymbol{\alpha}_i$ is given in advance, the closed-loop eigenvector $\mathbf{v}_i$ is determined by Equation (28), and the pole placement problem is transformed into finding feedback gains satisfying Equation (27). In addition, the unchanged closed-loop eigenvalues also satisfy the following condition:

$$\mathbf{P}(\lambda_i)(\lambda_i\mathbf{G}_{\mathrm{v}}+\mathbf{G}_{\mathrm{d}})\boldsymbol{\Phi}_{\mathrm{s}}\mathbf{w}_i=0,\quad i=m+1,\cdots,2n_{\mathrm{h}}.\tag{29}$$

Since $\mathbf{P}(s)$ can take various forms, Equation (29) is modified to the following sufficient condition:

$$(\lambda_i\mathbf{G}_{\mathrm{v}}+\mathbf{G}_{\mathrm{d}})\boldsymbol{\Phi}_{\mathrm{s}}\mathbf{w}_i=0,\quad i=m+1,\cdots,2n_{\mathrm{h}}.\tag{30}$$

Combining Equations (27) and (30), the following linear equation is deduced to solve the feedback gain matrix:

$$\mathbf{G}\mathbf{Y}=\mathbf{X},\tag{31}$$

where

$$\begin{aligned}\mathbf{G}&=\begin{bmatrix}\mathbf{G}_{\mathrm{v}} & \mathbf{G}_{\mathrm{d}}\end{bmatrix},\\\mathbf{X}&=\begin{bmatrix}\boldsymbol{\alpha}_1 & \cdots & \boldsymbol{\alpha}_m & 0 & \cdots & 0\end{bmatrix},\\\mathbf{Y}&=\begin{bmatrix}\mu_1\boldsymbol{\Phi}_{\mathrm{s}}\mathbf{v}_1 & \cdots & \mu_m\boldsymbol{\Phi}_{\mathrm{s}}\mathbf{v}_m & \lambda_{m+1}\boldsymbol{\Phi}_{\mathrm{s}}\mathbf{w}_{m+1} & \cdots & \lambda_{2n_{\mathrm{h}}}\boldsymbol{\Phi}_{\mathrm{s}}\mathbf{w}_{2n_{\mathrm{h}}}\\\boldsymbol{\Phi}_{\mathrm{s}}\mathbf{v}_1 & \cdots & \boldsymbol{\Phi}_{\mathrm{s}}\mathbf{v}_m & \boldsymbol{\Phi}_{\mathrm{s}}\mathbf{w}_{m+1} & \cdots & \boldsymbol{\Phi}_{\mathrm{s}}\mathbf{w}_{2n_{\mathrm{h}}}\end{bmatrix}.\end{aligned}\tag{32}$$

In case the number of sensors n_{s} is greater than or equal to the number of structural modes n_{h}, the proper selection of sensor location will make the sensor modal matrix $\boldsymbol{\Phi}_{\mathrm{s}}$ have a full column rank. Thus, the coefficient matrix $\mathbf{Y}$ in Equation (31) is also a full column rank, and the solution of the equation must exist. That is to say, in theory, the weighting parameters $\boldsymbol{\alpha}_i$ can be arbitrarily chosen (appear in conjugate pairs), and the real feedback gains that meet the pole placement conditions can always be obtained by Equation (31).

According to the discussion in Ref. [53], the minimum control effort can be achieved by choosing the closed-loop

eigenvectors $\mathbf{v}_i$ equal to the open-loop eigenvectors $\mathbf{w}_i$. When the number of control surfaces n_c is less than the number of structural modes n_h, Equation (28) shows that it is unlikely to find an $\boldsymbol{\alpha}_i$ that exactly meets this condition. Therefore, in this study, the following formula is used to approximately obtain the $\boldsymbol{\alpha}_i$ so that the closed-loop eigenvectors $\mathbf{v}_i$ and the open-loop eigenvectors $\mathbf{w}_i$ are as close as possible.

$$\boldsymbol{\alpha}_i = \left(\mathbf{L}_i^{\mathrm{T}}\mathbf{L}_i\right)^{-1}\mathbf{L}_i^{\mathrm{T}}\mathbf{w}_i, \quad i = 1, \cdots, m, \tag{33}$$

where $\mathbf{L}_i$ is the value of the transfer function $\mathbf{H}(s)\mathbf{P}(s)$ at point μ_i.

4.2. Optimal Sensor Placement. The number and placement of sensors are critical to determining the existence of control gains. Based on the locations of FEM nodes where sensors are placed, the sensor modal matrix $\boldsymbol{\Phi}_s$ in Equation (22) is formed from the structure modal matrix $\boldsymbol{\Phi}_h$. Normally, it is easy to select a satisfactory $\boldsymbol{\Phi}_s$ to meet the control requirements through trial and error. However, for the folding wing model, the $\boldsymbol{\Phi}_s$ corresponding to a certain sensor layout varies with the folding angle. A sensor layout suitable for one fixed folding angle may not meet the control requirements at other folding angles.

In order to solve the sensor placement problem of the morphing structure, based on the effective independence method proposed by Kammer [54], this paper generalizes the independence distribution vector to a function form. Then, the problem of optimal sensor placement is settled by optimizing the modified distribution vector in an iterative manner. The optimization strategy of the effective independence method is to quantitatively evaluate all possible sensor locations and iteratively eliminate insignificant locations to obtain the final sensor layout. First, define the Fisher information matrix $\mathbf{Q}_s \in \mathbb{R}^{n_h \times n_h}$ as a performance index of the sensor distribution as

$$\mathbf{Q}_s = \boldsymbol{\Phi}_s^{\mathrm{T}}\boldsymbol{\Phi}_s. \tag{34}$$

Only if the information matrix has a relatively small condition number, the problem of observing target structural modes through the sensor outputs is well-conditioned, and Equation (31) does not appear to be ill-conditioned, and the numerical solution exists. Moreover, a larger value of $\mathbf{Q}_s$ means that the signal energy output by sensors is larger, which is beneficial for improving the sensor noise immunity and implementing active control. It is commonly suggested to use the trace norm or determinant as an overall index of $\mathbf{Q}_s$. In order to evaluate the contribution of each sensor individually, the independence distribution vector $\mathbf{e}_D \in \mathbb{R}^{n_s}$ is defined as the diagonal elements of the projection matrix formulated by $\boldsymbol{\Phi}_s$, as shown below:

$$\mathbf{e}_D = \mathrm{diag}\left[\boldsymbol{\Phi}_s\left(\boldsymbol{\Phi}_s^{\mathrm{T}}\boldsymbol{\Phi}_s\right)^{-1}\boldsymbol{\Phi}_s^{\mathrm{T}}\right]. \tag{35}$$

Kammer [54] pointed out that each element in $\mathbf{e}_D$ represents the fractional contribution of the corresponding sensor location to the linear independence of modes in $\boldsymbol{\Phi}_s$ so that removing sensor locations with smaller independence contributions can maintain the determinant of the information matrix. Poston and Tolson [55] later demonstrated that the i-th element e_{Di} of the vector $\mathbf{e}_D$ has the following explicit expression:

$$e_{Di} = \frac{\det\left(\mathbf{Q}_s\right) - \det\left(\mathbf{Q}_{si}\right)}{\det\left(\mathbf{Q}_s\right)}, \quad i = 1, \cdots, n_s, \tag{36}$$

where $\mathbf{Q}_{si}$ is the information matrix constructed after removing the i-th row of $\boldsymbol{\Phi}_s$. From the above formula, it can be seen that $0 \le e_{Di} \le 1$, and e_{Di} equal to 0 indicates that the determinant of the information matrix remains unchanged after removing the corresponding sensor location, and e_{Di} equal to 1 indicates that the information matrix is singular after removing the corresponding sensor location. Generally, removing the sensor location with the smallest independence distribution value will maximize the determinant of the information matrix while reducing the number of sensors.

In the folding wing model, $\boldsymbol{\Phi}_s$ is a function of the folding angle θ. In order to apply the effective independence method to the morphing structure, the independence distribution vector defined in Equation (35) should also generalize to the function $\mathbf{e}_D(\theta)$. This paper adopts a harmonic mean fashion to construct the mean value $\bar{\mathbf{e}}_D$ of the function $\mathbf{e}_D(\theta)$ as an integrated performance index for the sensor placement, where the i-th element $\bar{e}_{Di}$ is calculated as

$$\bar{e}_{Di} = 1 - \left(\theta_{\max} - \theta_{\min}\right)\left(\int_{\theta_{\min}}^{\theta_{\max}} \frac{1}{1 - e_{Di}(\theta)}\, d\theta\right)^{-1}, \quad i = 1, \cdots, n_s, \tag{37}$$

where $\theta_{\min}$ and $\theta_{\max}$ are the lower and upper bounds of the folding angle range, respectively. Compared with the arithmetic mean, the above formula is more biased toward larger values in the function curve. This is because the independence distribution close to 1 means that the corresponding sensor location is of great significance; the above index ensures that as long as the function $e_{Di}(\theta)$ has a value close to 1, then its mean value $\bar{e}_{Di}$ is also close to 1. In practice, the structural models are established at n_t discrete folding angles, and the definite integral in Equation (37) is approximated by a finite sum as

$$\bar{e}_{Di} = 1 - n_t\left(\sum_{l=1}^{n_t} \frac{1}{1 - e_{Di,l}}\right)^{-1}, \quad i = 1, \cdots, n_s, \tag{38}$$

where $e_{Di,l}$ represents the i-th element of the independence distribution vector at the l-th folding angle. To ensure numerical accuracy, it is recommended to increase the number of modal matrices $\boldsymbol{\Phi}_s$ by interpolating the coherent local models.

The procedure for optimal sensor placement based on the effective independence method is as follows: (1) select an initial candidate set of sensor locations and construct

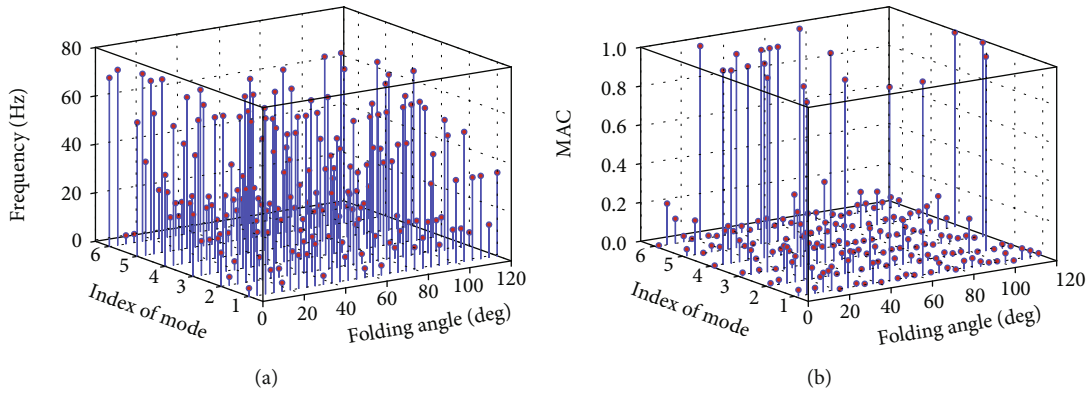


Figure 4: The scrambled structural modes (a) and the MAC between every two adjacent folding angles (b).

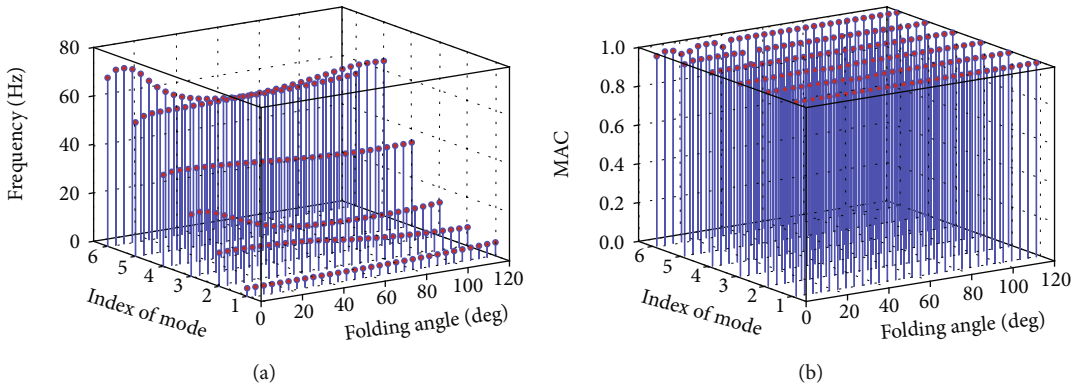


Figure 5: The matched structural modes (a) and the MAC between every two adjacent folding angles (b).

independence distribution vectors at each folding angle; (2) construct the integrated distribution vector according to Equation (38); (3) delete the sensor location with the smallest value in the integrated distribution vector and then reconstruct it; and (4) repeat the above process, deleting one location each time until the required number of sensors is reached.

5. Numerical Results and Discussions

5.1. Validation of the Interpolation-Based Modeling. In this subsection, the effectiveness of the developed interpolation-based modeling method is validated, and the LPV baseline model for control synthesis and response prediction is established. As mentioned above, the procedure of interpolation-based modeling includes two steps: the structure modal matching and the treatment of the RFA coefficient matrices.

In the simulation, a set of 61 local aeroelastic models are generated according to fixed folding angles, ranging from 0 deg to 120 deg with 2 deg intervals. The initial aeroelastic model shown in Equation (5) includes the structural matrices and the GAF matrices, and the coherency of these matrices cannot be guaranteed. To verify the proposed modal matching method, orders of the first 6 structural modes at each parameter point are randomly assigned, as shown in Figure 4(a). The MAC value in Figure 4(b) reveals that the mode shapes between adjacent folding angles have extremely low linear correlations. Hence, all the system matrices in

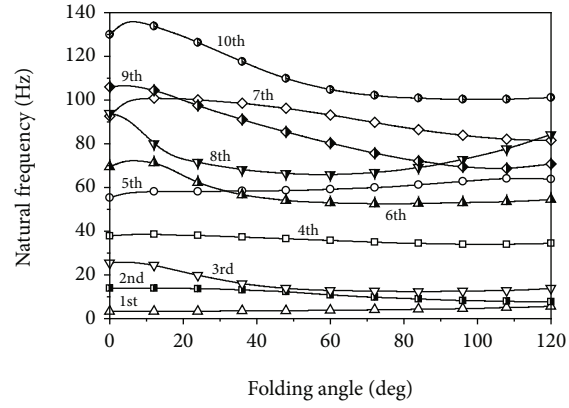


Figure 6: Variations of the first ten natural frequencies with the folding angle.

Equation (5) exhibit discontinuities. After implementing the modal matching algorithm described in Section 3.1, it can be seen from Figure 5 that the scrambled structural modes are successively aligned. Thus, the obtained coherent structure modal matrices are continuously dependent on the folding angle.

Figure 6 exhibits the evolution of the first ten mode branches. It is clear that the structural modes are significantly affected by the folding angle, and the proposed modal matching algorithm exactly tracks each mode branch. Complex mode crossing phenomena occur in the high-order

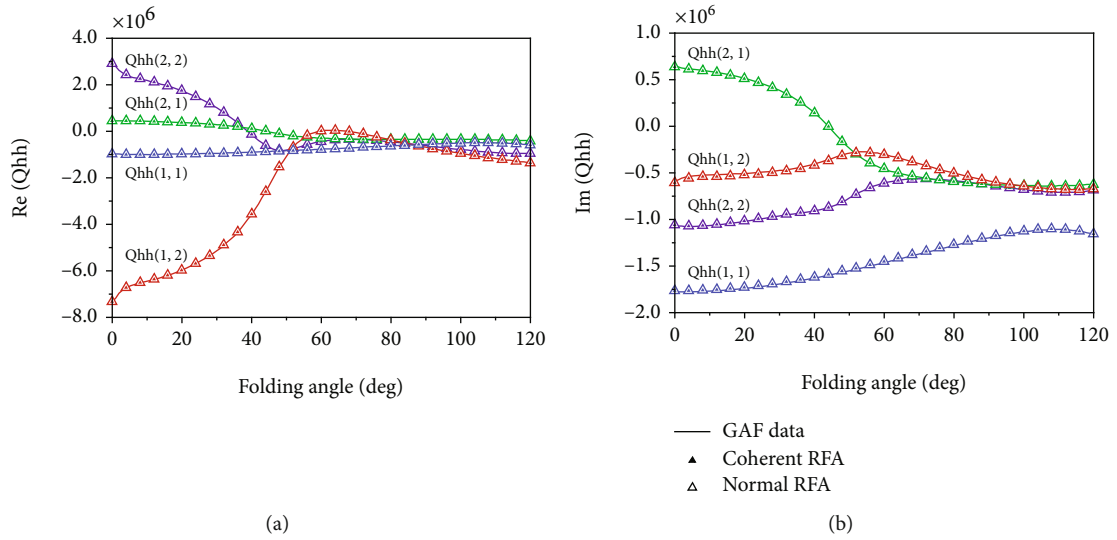


FIGURE 7: Comparison of approximated GAF obtained from normal RFA and coherent RFA with the original GAF data ($k = 0.25$). (a) Real part. (b) Imaginary part.

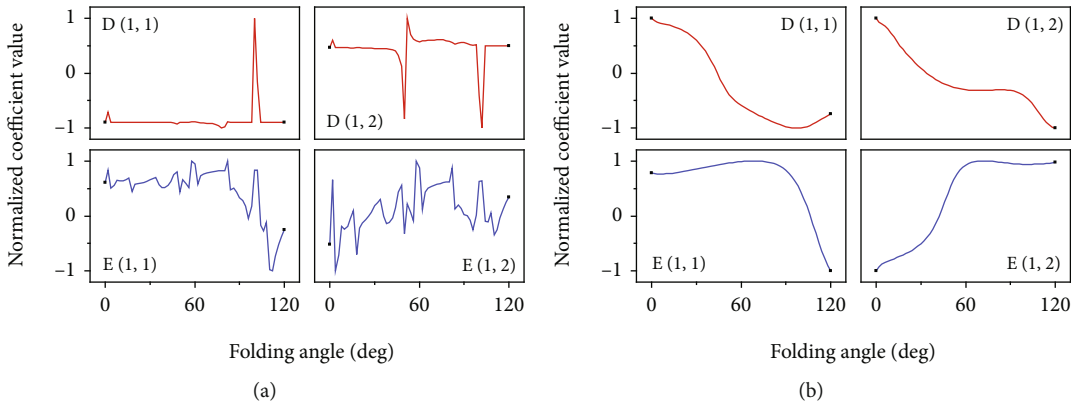


FIGURE 8: Variations of coefficient matrices **D** and **E** in the RFA representation. (a) Normal RFA. (b) Coherent RFA.

modes. Even so, the algorithm still gives the correct matching result.

In the following, retain the first 6 matched structural modes and continue to construct the state-space aeroelastic model of the folding wing. At the incompressible flow condition, the GAF matrices are computed at 16 reduced frequencies in the range of 0 to 1.5. Then, the MS method is applied to fit the GAF matrices to the RFA expression shown in Equation (13). The normal RFA is performed independently at each folding angle, and the aerodynamic lag roots required in the MS method are obtained by the following empirical formula:

$$R_i = -1.7k_{\max}\left(\frac{i}{n_a + 1}\right)^2, \quad i = 1, \cdots, n_a, \tag{39}$$

where the number of aerodynamic lag roots n_a is set to 6 and the maximum reduced frequency $k_{\max}$ is 1.5. In the MS procedure, a constant initial matrix **D** and 300 iterations are used for all parameter points.

After the normal RFA procedure is completed, the coefficient matrices are adjusted to a coherent form according to the method described in Section 3.2. All results shown below use the matched structural modes, and the verification work focuses on the influence of the coherent and normal RFA representations on the modeling results. As shown in Figure 7, the directly computed GAF data (solid lines) vary smoothly with the folding angle. The normal RFA (open triangles) and the coherent RFA (filled triangles) are in good agreement with the directly computed GAF data. Although the normal RFA gives accurate fitting results at scattered parameter points, the coefficient matrices **D** and **E** obtained by normal RFA have dramatic jumps as the parameter varies (see Figure 8(a)), which makes the aeroelastic state-space interpolation impossible. By comparison, it can be concluded from Figure 8(b) that the proposed method for coherent RFA representation successfully adjusts the RFA coefficient matrices to a coherent form without changing the fitting results. The continuously varying system matrices pave the way for state-space model interpolation.

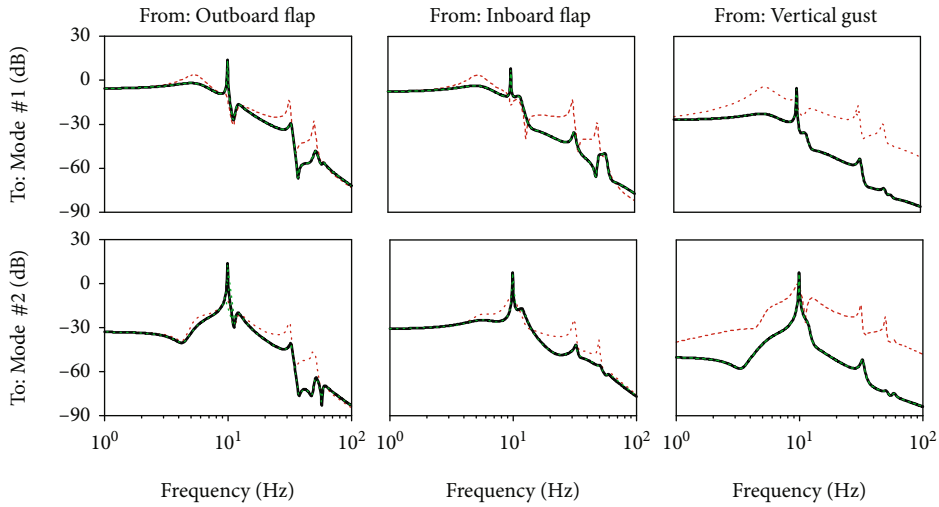
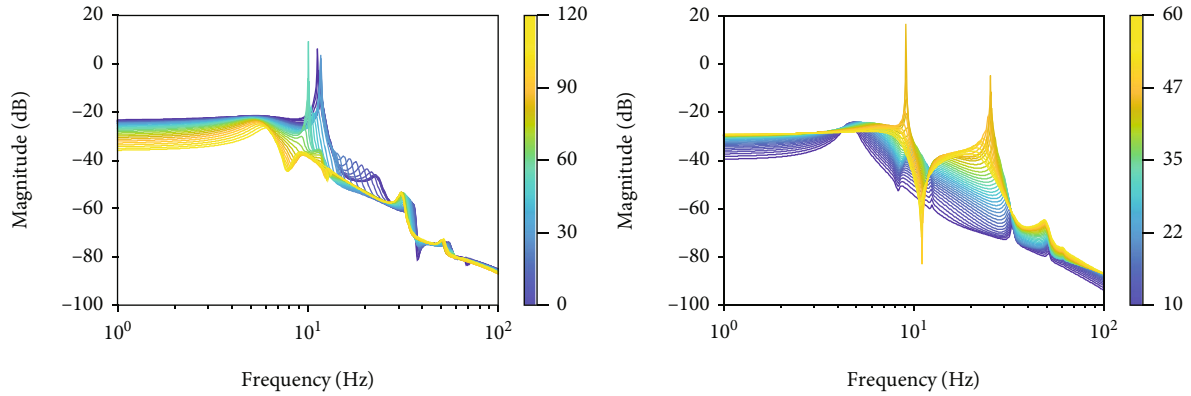


FIGURE 9: Comparison of frequency responses of the exact model (solid black, $\theta = 65$ deg, $U_\infty = 30$ m/s) and the interpolated model using incoherent (dashed red) and coherent (dashed green) system matrices.



(a) $\theta = 0$~120 deg, $U_\infty = 30$ m/s

(b) $\theta = 95$ deg, $U_\infty = 10 \sim 60$ m/s

FIGURE 10: Frequency responses for varying parameters. (a) Variation with respect to the folding angle. (b) Variation with respect to the flow speed.

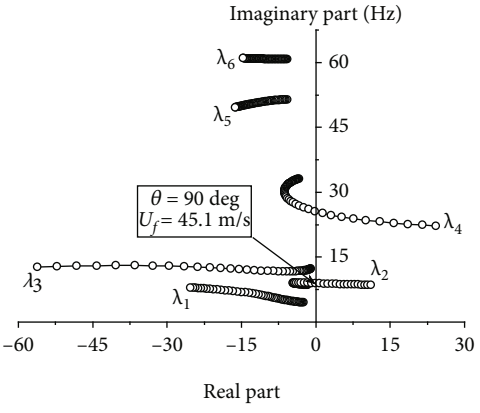


FIGURE 11: Evolution of open-loop eigenvalues with respect to the flow speed ($\theta = 90$ deg).

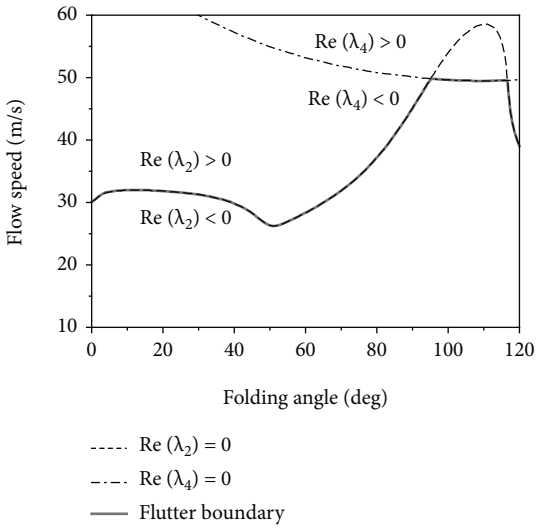


FIGURE 12: Contour lines of critical eigenvalues and the flutter boundary.

Figure 9 is the comparison of the frequency response functions of the interpolated model and the exact model. In the figure, the flight conditions are fixed at flow speed of 30 m/s and sea level, and the system model at $\theta = 65$ deg

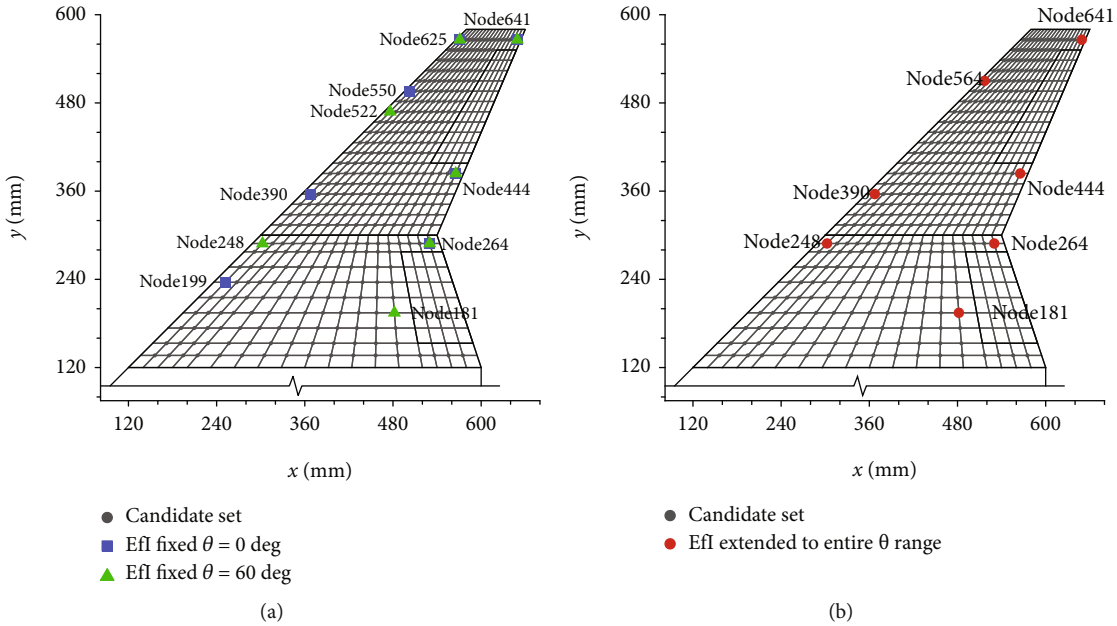


Figure 13: Optimal sensor placement layouts by the effective independence (EfI) method. (a) Fixed configurations. (b) Integrating all configurations.

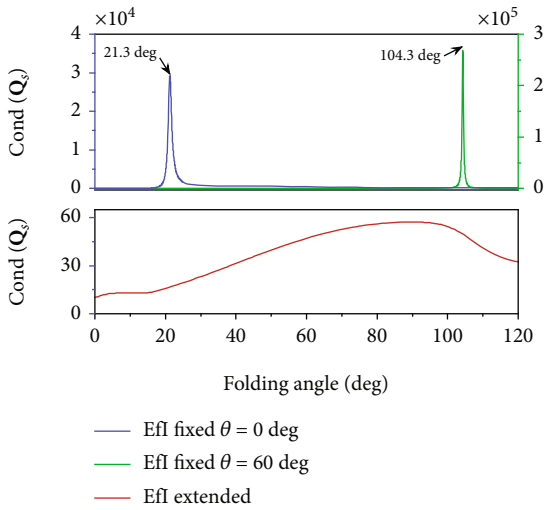


Figure 14: Condition number of the information matrix varying with the folding angle under different sensor layouts.

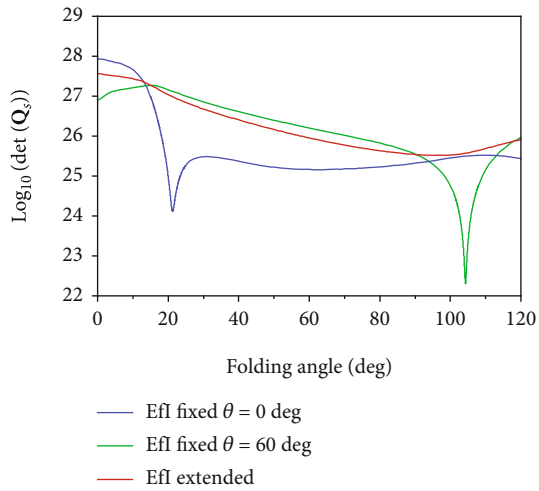


Figure 15: Determinant of the information matrix varying with the folding angle under different sensor layouts.

is obtained by linear interpolation of the local models at $\theta = 64$ deg and $\theta = 66$ deg. The system inputs are the deflections of the outboard and inboard control surfaces as well as the vertical gust disturbance, and the outputs are the first two modal displacements of the structure (numbered according to the mode branches in Figure 6). It can be seen that the coherent system matrices construct an accurate interpolated model. However, for the incoherent system matrices constructed by unprocessed RFA coefficient matrices, the interpolated model here brings serious modeling errors, as shown by red dashed lines in Figure 9.

Using the parameterized modeling approach proposed in this paper, an aeroelastic LPV model with two scheduling parameters is established as the baseline model for the subsequent simulation and control implementation. Here, the folding angle and the flow speed are taken as the scheduling parameters. In order to generate the local models, the two-dimensional parameter space is divided into regular grids, in which the folding angle parameter is still in the range of 0 deg to 120 deg with 2 deg intervals, and the flow speed is in the range of 10 m/s to 60 m/s with 1 m/s intervals. Figure 10 shows the frequency response curves for varying folding angles and flow speeds, in which the system input and output are the vertical gust disturbance and the first structure modal displacement, respectively.

5.2. Implementation of the Receptance-Based Control. The controller is designed to work at subcritical conditions, so

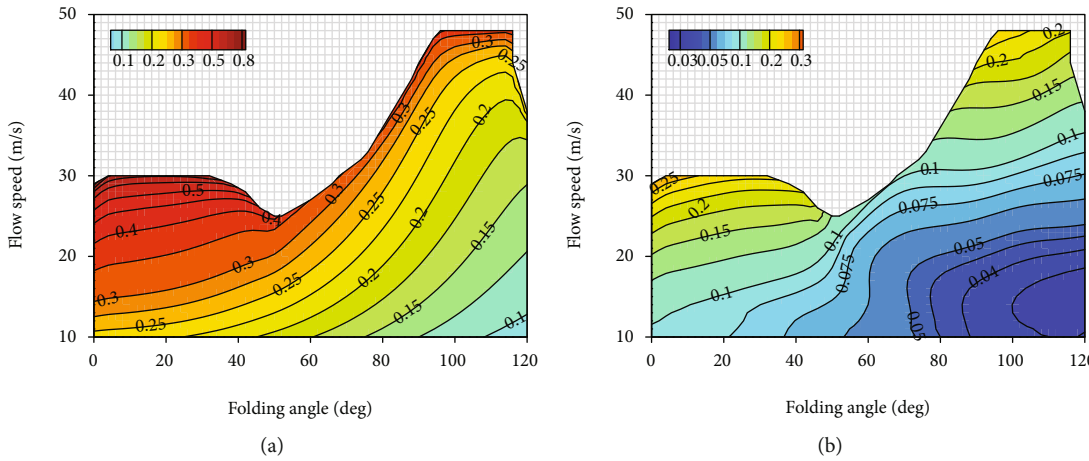


FIGURE 16: H_2-norm comparison of the (a) open-loop and (b) closed-loop systems in the design parameter region.

only parameters below the flutter speed need to be considered. The first step in control design is to obtain the eigenvalues and eigenvectors of the open-loop system. Since the GAF matrices have been converted to the RFA representation, solving the characteristic equation of Equation (24) is equivalent to solving the eigenvalues of the state-space matrix $\mathbf{A}$ in Equation (18). Similar to the structure modal data, the initial eigenvalues at discrete parameter points may not have a coherent order. Obviously, the structure modal matching algorithm in Section 3.1 is also applicable to the aeroelastic systems, and it only needs to replace the natural frequency ω and the mode shape $\boldsymbol{\varphi}$ in Equation (9) with the eigenvalue λ and the eigenvector $\mathbf{w}$ of the aeroelastic system. Through the modal matching procedure, the eigenvalues that vary smoothly with the parameter can be obtained. This is useful to compute a continuous parameterized controller through uniform pole placement.

According to the number of structural modes, 6 pairs of complex conjugate eigenvalues are calculated. Based on this, the flutter characteristics of the folding wing within the folding angle range are also obtained. As an example for $\theta = 90$ deg, Figure 11 shows the root locus of the matched eigenvalues with respect to the flow speed. The flutter speed is identified by the critical point at which the real part of an eigenvalue trajectory changes from negative to positive. At $\theta = 90$ deg, the 2nd mode firstly becomes unstable, and the corresponding flutter speed $U_f = 45.1$ m/s. Figure 12 is a contour map where the real part of the eigenvalues is equal to zero on the two-dimensional parameter plane. In the entire folding angle range, only the 2nd and 4th modes have eigenvalues with nonnegative real parts. Combining the critical contour lines of these two modes, the flutter boundary and the stable region in the parameter space can be obtained.

The sensor layout should be determined before calculating the control gains. In the current example, the necessary condition for the existence of the receptance controller is that the number of sensors is greater than or equal to 6. Therefore, we intend to find 7 sensor locations that maximize the information matrix $\mathbf{Q}_s$ defined in Section 4.2. To apply the effective independence method, the initial candidate set of sensor locations is selected as the normal direction of the structural nodes on the inboard and outboard wings. The 7 optimal locations are iteratively retained from 388 initial locations. Figure 13 shows the final sensor layouts obtained by the standard method (fixed configuration) and the modified method (parameter-varying version). Figure 13(a) shows the optimal locations at fixed configurations for $\theta = 0$ deg and $\theta = 60$ deg, and Figure 13(b) shows the optimal locations obtained by integrating all folding angles.

The information matrices related to the above three sensor layouts are compared and shown in Figures 14 and 15. As shown in Figure 14, the sensor layout obtained at fixed $\theta = 0$ deg causes a large condition number of the information matrix at 21.3 deg; the sensor layout obtained at fixed $\theta = 60$ deg also encounters the same problem at 104.3 deg. Either of these two sensor layouts will lead to ill-conditioned solutions of Equation (31), and the resulting control gains will produce high control effort or invalid control at certain folding angles. As a comparison, the modified method proposed in this paper provides an improved sensor layout, and the condition number of the information matrix has a relatively small value in the entire folding angle range. Figure 15 also shows the determinant of the information matrix varying with the folding angle. It can be seen that under fixed folding angles of $\theta = 0$ deg and $\theta = 60$ deg, the sensor layouts obtained by the standard method are optimal at the each given folding angle, but it cannot guarantee performances for other angles. The modified method does not produce the maximum determinant at most folding angles, but the obtained sensor layout achieves good comprehensive performance in the entire folding angle range.

The next step in control design is to manually assign the eigenvalues of the closed-loop system. The system response induced by gust disturbances is dominated by low-frequency modes. Therefore, the first 4 eigenvalues of the open-loop system are intended to be assigned to specified values, while the 5th and 6th eigenvalues remain unchanged. The assigned closed-loop eigenvalues are specified by the known open-loop eigenvalues plus the real and imaginary

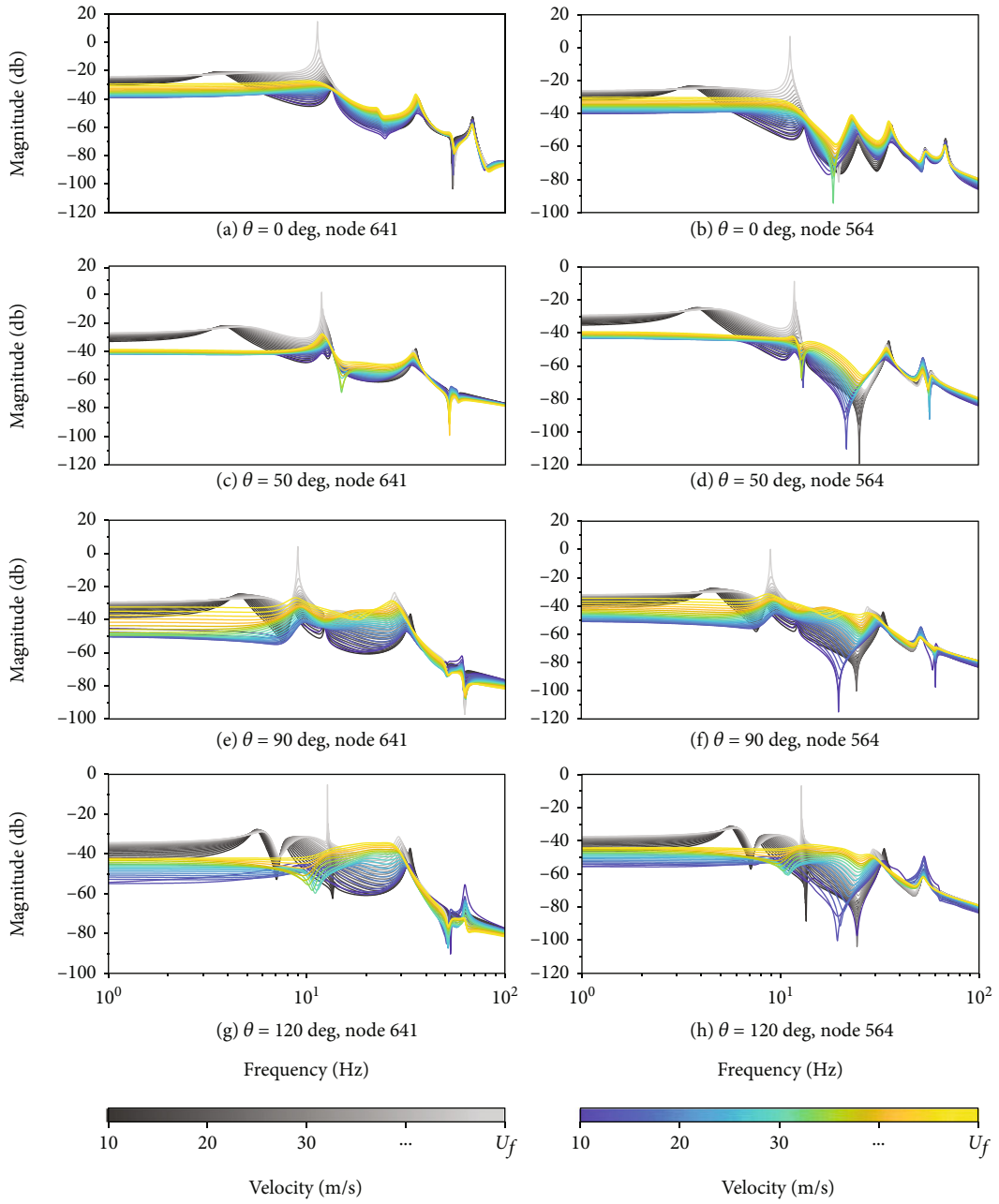


FIGURE 17: Frequency responses of the open-loop (black line/gray line) and closed-loop (blue line/yellow line) systems from the vertical gust disturbance to the displacements of structural nodes.

part increments, so there are 8 increments that need to be given at each parameter point. In the two-dimensional parameter space, the vector Δ composed of the increments of the open-loop eigenvalues is uniformly given by the following polynomial:

$$\Delta(x, y) = \sum_{i+j=0}^{n} \mathbf{p}_{ij} x^i y^j, \tag{40}$$

where x and y are the normalized parameters of θ and U_∞, $\mathbf{p}_{ij}$ is the coefficient vector, and n is the degree of the polynomial. The way of specifying the closed-loop eigenvalues by an explicit expression can make the controller change continuously with the parameters. Set $n = 3$, and then all undetermined coefficients constitute 80 free variables, and these variables will determine the final control gains and closed-loop models. The influence of pole placement on disturbance rejection can be evaluated by calculating the H_2 -norm or H_∞ -norm of the closed-loop system. Unfortunately, analytically assigning poles to minimize the system norm is still an open problem, so this study uses the optimization method to obtain the coefficients in Equation (40). The optimization is implemented by using the Nelder-Mead simplex algorithm [56], which is a direct search method that does not use gradients. The objective function

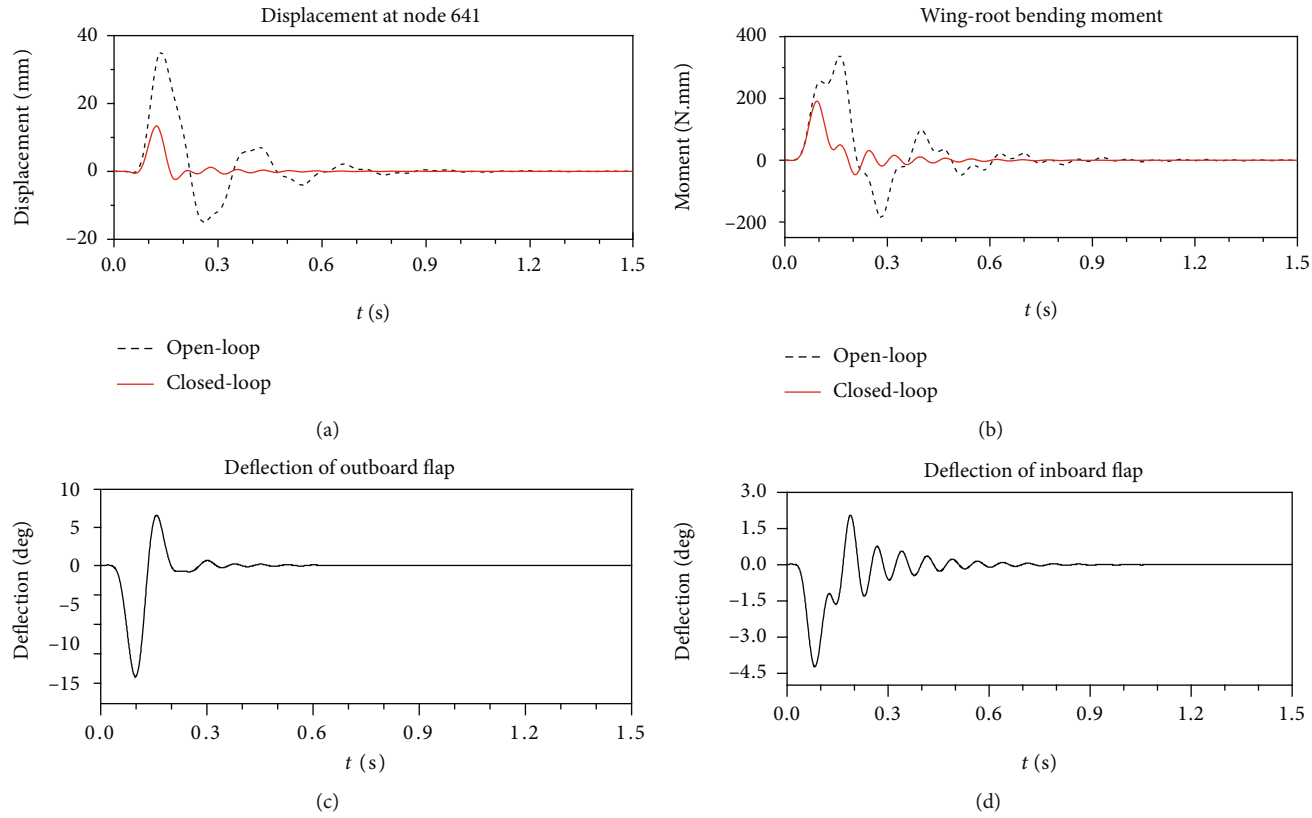


Figure 18: Dynamic responses of the open and closed-loop systems to the 1-cos discrete gust ($\theta = 0$ deg, $U_\infty = 15$ m/s).

is set as the average value of the H_2-norm of all local closed-loop systems within the flutter boundary.

According to the above steps, the control gains are calculated one by one at each design parameter point. Based on the interpolation of the local controllers, a parameterized gust alleviation controller for the folding wing is finally established. Figure 16 compares the H_2-norm distribution of the open-loop and closed-loop systems in the parameter space under the vertical gust disturbance. The figure uses different colors to indicate the system norm. The H_2 -norm of the open-loop system at the flutter boundary approaches infinity, so only the gust alleviation results within the flutter boundary are investigated. In general, increasing the folding angle can passively reduce the influence of the vertical gust disturbance. After the active control system works, the response energy of the closed-loop system is significantly reduced, and the controller shows a powerful control capability within a wide parameter-varying range.

Figure 17 shows the frequency responses of the open-loop and closed-loop systems at different folding angles. In each subfigure, the folding angle is fixed, and a series of curves represent the frequency response amplitude at different flow speeds from 10 m/s to the flutter speed U_f. The open-loop and closed-loop frequency responses are represented by gray and colored curves, respectively. It can be seen that changing the first 4 eigenvalues of the system can effectively reduce the low-frequency responses. The first-order bending vibration of the wing is greatly reduced by the parameterized controller.

In order to further verify the controller performance, the time-domain dynamic responses of the folding wing to gust excitations are calculated. Figures 18 and 19 show the dynamic responses of the folding wing to the 1-cos discrete gust under two sets of simulation parameters. The frequency of the 1-cos gust is set to 10 Hz, and the maximum gust velocity is 1 m/s. The simulation results show that the displacement and wing-root bending moment of the folding wing are significantly reduced due to the driving of the inboard and outboard control surfaces. In the flight condition of $\theta = 0$ deg and $U_\infty = 15$ m/s, the maximum value of the wing-tip displacement at node 641 is reduced by 62.00%, while the maximum value of the wing-root bending moment is reduced by 43.88%. In the condition of $\theta = 90$ deg and $U_\infty = 30$ m/s, the maximum values of the displacement and the bending moment are reduced by 79.99% and 60.34%, respectively. Table 1 lists the detailed data of the open-loop and closed-loop responses at four different folding angles.

For the continuous gust control, Table 2 shows an overview of the folding wing responses to the Dryden gust, as well as the statistical comparisons of the open-loop and closed-loop responses at different folding angles. The scale of the Dryden gust is set to 5 m, and the root mean square (RMS) value of the gust velocity is 0.5 m/s. The RMS values of the wing-tip displacement, the wing-root bending

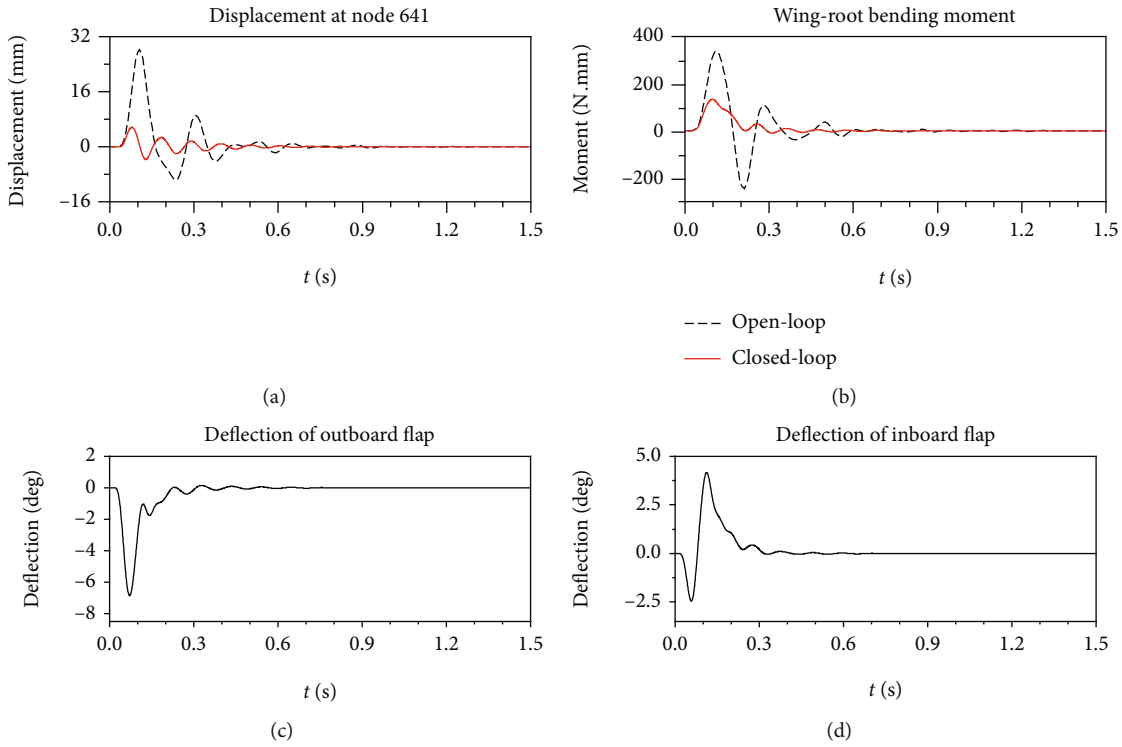


Figure 19: Dynamic responses of the open and closed-loop systems to the 1-cos discrete gust ($\theta = 90$ deg, $U_\infty = 30$ m/s).

Table 1: Maximum values of the wing-tip displacement (D_n), wing-root bending moment (M_x), and outboard flap deflection (δ) in discrete gust responses.

θ (deg)	Variable	Maximal value of the variable					
		$U_\infty = 15$ m/s			$U_\infty = 30$ m/s		
		Open-loop	Closed-loop	Reduction of the maximal value	Open-loop	Closed-loop	Reduction of the maximal value
0	D_n (mm)	35.21	13.38	62.00%	77.57	35.08	54.77%
	M_x (N•mm)	340.86	191.28	43.88%	731.16	618.80	15.37%
	δ (deg)	—	14.62	—	—	12.18	—
50	D_n (mm)	27.00	7.99	70.40%	Inf.	12.21	—
	M_x (N•mm)	292.28	117.68	59.74%	Inf.	294.88	—
	δ (deg)	—	10.98	—	—	9.21	—
90	D_n (mm)	16.57	3.11	81.21%	28.24	5.65	79.99%
	M_x (N•mm)	205.04	48.23	76.48%	338.76	134.35	60.34%
	δ (deg)	—	11.31	—	—	6.86	—
120	D_n (mm)	11.82	2.88	75.65%	19.26	4.79	75.15%
	M_x (N•mm)	125.75	14.93	88.12%	199.03	19.75	90.08%
	δ (deg)	—	8.73	—	—	4.66	—

moment, and the outboard control surface deflection are listed in Table 2.

Through the fixed parameter simulations, it is clearly seen that the controller is valid at all parameter points and has good alleviation effects in both the discrete and continuous gusts. Next, the time-varying system simulation and control for the folding wing system are performed. As shown in Figure 20, during the simulation time of 60 seconds, both the folding angle and the flow speed change slowly and uniformly, where the folding angle changes from 0 deg to 120 deg and the flow speed changes from 10 m/s to 30 m/s. The time-varying system simulation shows that the structural vibration and additional loads are alleviated in each time period. In conclusion, the above results verify that the

Table 2: RMS values of the wing-tip displacement (D_n), wing-root bending moment (M_x), and outboard flap deflection (δ) in continuous gust responses.

θ (deg)	Variable	RMS value of the variable					
		U_∞ = 15 m/s			U_∞ = 30 m/s		
		Open-loop	Closed-loop	Reduction of the RMS value	Open-loop	Closed-loop	Reduction of the RMS value
0	D_n (mm)	22.47	5.90	73.75%	67.73	16.67	75.39%
	M_x (N•mm)	272.53	111.24	59.18%	634.32	303.91	52.09%
	δ (deg)	—	5.82	—	—	4.77	—
50	D_n (mm)	17.01	3.95	76.80%	Inf.	7.14	—
	M_x (N•mm)	217.84	67.33	69.09%	Inf.	160.25	—
	δ (deg)	—	4.18	—	—	3.94	—
90	D_n (mm)	9.60	1.50	84.33%	15.43	2.44	84.18%
	M_x (N•mm)	125.33	36.05	71.24%	209.29	114.84	45.13%
	δ (deg)	—	4.80	—	—	4.02	—
120	D_n (mm)	5.72	1.23	78.53%	9.29	3.01	67.56%
	M_x (N•mm)	58.12	6.20	89.33%	102.92	10.04	90.25%
	δ (deg)	—	3.34	—	—	2.00	—

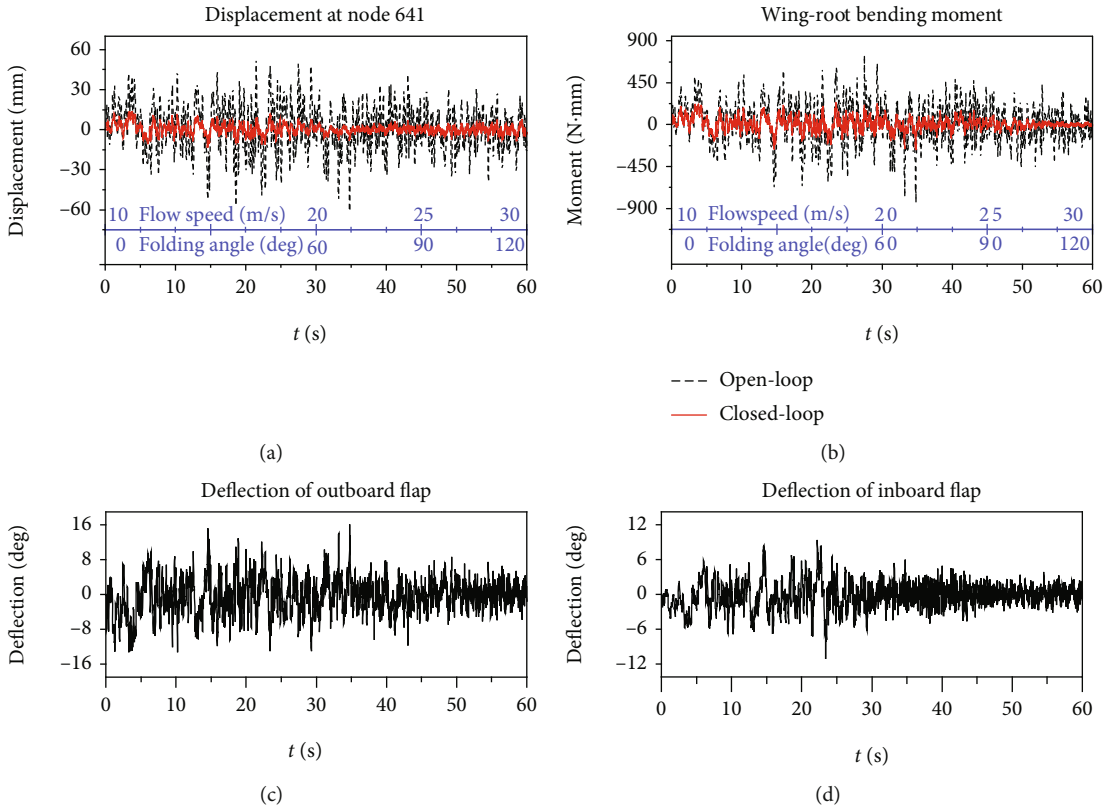


Figure 20: Dynamic responses of the open and closed-loop systems to continuous gust under the time-varying folding angle and flow speed.

proposed parameterized controller is valid for a wide range of parameters. Under the assumption of slow parameter variation, extending the receptance method to the parameter-varying system is able to design a reliable and effective gust alleviation controller.

6. Conclusions

To efficiently investigate the aeroelasticity and control of a folding wing, this paper presents an interpolation-based modeling strategy for parameterized aeroelastic systems.

The key steps involved are the structure modal matching and the manipulation of the RFA coefficient matrices. The main advantage of the proposed method is that the coherent local models are obtained before constructing the aeroelastic state-space matrices, which makes the coherent state-space representation much easier. Based on the developed modeling strategy, the LPV model of the folding wing system is constructed within the local modeling framework. Numerical examples demonstrate that the interpolation of incoherent local models brings serious modeling errors, while the proposed coherent representation method gives accurate modeling results. Next, the aeroelastic control for the folding wing under various flight conditions and structural configurations is studied. For this purpose, the original receptance-based control method for fixed configuration is extended to the parameter-varying system, and a modified version of the effective independence method is derived to select an optimal sensor layout suitable for all folding angles. The simulation results show that the designed parameterized controller for gust alleviation achieves satisfactory closed-loop performance in the given parameter space.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 11472128.

References

[1] T. A. Weisshaar, "Morphing aircraft systems: historical perspectives and future challenges," *Journal of Aircraft*, vol. 50, no. 2, pp. 337–353, 2013.

[2] S. Barbarino, O. Bilgen, R. M. Ajaj, M. I. Friswell, and D. J. Inman, "A review of morphing aircraft," *Journal of Intelligent Material Systems and Structures*, vol. 22, no. 9, pp. 823–877, 2011.

[3] R. M. Ajaj, C. S. Beaverstock, and M. I. Friswell, "Morphing aircraft: the need for a new design philosophy," *Aerospace Science and Technology*, vol. 49, pp. 154–166, 2016.

[4] M. H. Love, P. S. Zink, R. L. Stroud, D. R. Bye, and C. Chase, "Impact of actuation concepts on morphing aircraft structures," in *Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Palm Springs, California, USA, April 2004.

[5] I. Wang, "Aeroelastic and flight dynamics analysis of folding wing systems, [Ph.D. thesis]," Duke University, 2013.

[6] D. H. Lee and T. A. Weisshaar, "Aeroelastic studies on a folding wing configuration," in *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Austin, Texas, USA, April 2005.

[7] D. H. Lee and P. C. Chen, "Nonlinear aeroelastic studies on a folding wing configuration with free-play hinge nonlinearity," in *Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Newport, Rhode Island, USA, May 2006.

[8] I. Wang, S. C. Gibbs, and E. H. Dowell, "Aeroelastic model of multisegmented folding wings: theory and experiment," *Journal of Aircraft*, vol. 49, no. 3, pp. 911–921, 2012.

[9] Y. Zhao and H. Hu, "Parameterized aeroelastic modeling and flutter analysis for a folding wing," *Journal of Sound and Vibration*, vol. 331, no. 2, pp. 308–324, 2012.

[10] D. Tang and E. H. Dowell, "Theoretical and experimental aeroelastic study for folding wing structures," *Journal of Aircraft*, vol. 45, no. 4, pp. 1136–1147, 2008.

[11] P. J. Attar, D. Tang, and E. H. Dowell, "Nonlinear aeroelastic study for folding wing structures," *AIAA Journal*, vol. 48, no. 10, pp. 2187–2195, 2010.

[12] Y. Zhao and H. Hu, "Prediction of transient responses of a folding wing during the morphing process," *Aerospace Science and Technology*, vol. 24, no. 1, pp. 89–94, 2013.

[13] W. Hu, Z. Yang, and Y. Gu, "Aeroelastic study for folding wing during the morphing process," *Journal of Sound and Vibration*, vol. 365, pp. 216–229, 2016.

[14] W. Hu, Z. Yang, Y. Gu, and X. Wang, "The nonlinear aeroelastic characteristics of a folding wing with cubic stiffness," *Journal of Sound and Vibration*, vol. 400, pp. 22–39, 2017.

[15] M. L. Verstraete, B. A. Roccia, D. T. Mook, and S. Preidikman, "A co-simulation methodology to simulate the nonlinear aeroelastic behavior of a folding-wing concept in different flight configurations," *Nonlinear Dynamics*, vol. 98, no. 2, pp. 907–927, 2019.

[16] T. M. Seigler, D. A. Neal, J. S. Bae, and D. J. Inman, "Modeling and flight control of large-scale morphing aircraft," *Journal of Aircraft*, vol. 44, no. 4, pp. 1077–1087, 2007.

[17] B. Yan, Y. Li, P. Dai, and S. Liu, "Aerodynamic analysis, dynamic modeling, and control of a morphing aircraft," *Journal of Aerospace Engineering*, vol. 32, no. 5, article 04019058, 2019.

[18] R. Huang, Z. Yang, X. Yao, Y. Zhao, and H. Hu, "Parameterized modeling methodology for efficient aeroservoelastic analysis of a morphing wing," *AIAA Journal*, vol. 57, no. 12, pp. 5543–5552, 2019.

[19] D. H. Baldelli, D. H. Lee, R. S. S. Pena, and B. Cannon, "Modeling and control of an aeroelastic morphing vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1687–1699, 2008.

[20] T. Yue, L. Wang, and J. Ai, "Gain self-scheduled _H_ $_\infty$ control for morphing aircraft in the wing transition process based on an LPV model," *Chinese Journal of Aeronautics*, vol. 26, no. 4, pp. 909–917, 2013.

[21] A. Marcos and G. J. Balas, "Development of linear-parameter-varying models for aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 2, pp. 218–228, 2004.

[22] H. Pfifer and S. Hecker, "Generation of optimal linear parametric models for LFT-based robust stability analysis and control design," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 1, pp. 118–131, 2011.

[23] G. Ferreres, "Computation of a flexible aircraft LPV/LFT model using interpolation," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 1, pp. 132–139, 2011.

[24] M. Lovera, M. Bergamasco, and F. Casella, "LPV modelling and identification: an overview," in *Robust Control and Linear*

Parameter Varying Approaches, O. Sename, P. Gaspar, and J. Bokor, Eds., Springer, Berlin, Heidelberg, 2013.

[25] C. Poussot-Vassal and C. Roos, "Generation of a reduced-order LPV/LFT model from a set of large-scale MIMO LTI flexible aircraft models," *Control Engineering Practice*, vol. 20, no. 9, pp. 919–930, 2012.

[26] C. Roos, "Generation of flexible aircraft LFT models for robustness analysis," *IFAC Proceedings Volumes*, vol. 42, no. 6, pp. 349–354, 2009.

[27] C. Poussot-Vassal and F. Demourant, "Dynamical medium (large)-scale model reduction and interpolation with application to aircraft systems," *The French Aerospace Lab*, vol. 4, pp. 1–11, 2012.

[28] J. Theis, B. Takarics, H. Pfifer, G. Balas, and H. Werner, "Modal matching for LPV model reduction of aeroservoelastic vehicles," in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference*, Kissimmee, Florida, USA, January 2015.

[29] A. K. Al-Jiboory, G. Zhu, S. S. M. Swei, W. Su, and N. T. Nguyen, "LPV modeling of a flexible wing aircraft using modal alignment and adaptive gridding methods," *Aerospace Science and Technology*, vol. 66, pp. 92–102, 2017.

[30] M. Lovera and G. Mercere, "Identification for gain-scheduling: a balanced subspace approach," in *Proceedings of the 2007 American Control Conference*, New York, NY, USA, July 2007.

[31] F. Ferranti and Y. Rolain, "A local identification method for linear parameter-varying systems based on interpolation of state-space matrices and least-squares approximation," *Mechanical Systems and Signal Processing*, vol. 82, pp. 478–489, 2017.

[32] J. de Caigny, R. Pintelon, J. F. Camino, and J. Swevers, "Interpolated modeling of LPV systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2232–2246, 2014.

[33] Y. M. Ram and J. E. Mottershead, "Receptance method in active vibration control," *AIAA Journal*, vol. 45, no. 3, pp. 562–567, 2007.

[34] Y. M. Ram and J. E. Mottershead, "Multiple-input active vibration control by partial pole placement using the method of receptances," *Mechanical Systems and Signal Processing*, vol. 40, no. 2, pp. 727–735, 2013.

[35] K. V. Singh, L. A. McDonough, R. Kolonay, and J. E. Cooper, "Receptance-based active aeroelastic control using multiple control surfaces," *Journal of Aircraft*, vol. 51, no. 1, pp. 335–342, 2014.

[36] H. Liu, X. Gao, and X. Wang, "Parametric active aeroelastic control of a morphing wing using the receptance method," *Journal of Fluids and Structures*, vol. 98, article 103098, 2020.

[37] E. Albano and W. P. Rodden, "A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows," *AIAA Journal*, vol. 7, no. 2, pp. 279–285, 1969.

[38] Y. Zhao and R. Huang, *Advanced Aeroelasticity and Control*, Science Press, Beijing, China, 2015.

[39] R. L. Harder and R. N. Desmarais, "Interpolation using surface splines," *Journal of Aircraft*, vol. 9, no. 2, pp. 189–191, 1972.

[40] M. Karpel, B. Moulin, and P. C. Chen, "Dynamic response of aeroservoelastic systems to gust excitation," *Journal of Aircraft*, vol. 42, no. 5, pp. 1264–1272, 2005.

[41] M. Karpel, "Design for active flutter suppression and gust alleviation using state-space aeroelastic modeling," *Journal of Aircraft*, vol. 19, no. 3, pp. 221–227, 1982.

[42] Z. Alkhoury, M. Petreczky, and G. Mercere, "Comparing global input-output behavior of frozen-equivalent LPV state-space models," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9766–9771, 2017.

[43] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*, SIAM, Philadelphia, USA, 2009.

[44] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 2, no. 1–2, pp. 83–97, 1955.

[45] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, 1987.

[46] E. Livne, "Alternative approximations for integrated control/-structure aeroservoelastic synthesis," *AIAA Journal*, vol. 31, no. 6, pp. 1100–1108, 1993.

[47] F. M. Hoblit, *Gust Loads on Aircraft: Concepts and Applications*, AIAA Education Series, Washington, USA, 1988.

[48] Y. Zhao, C. Yue, and H. Hu, "Gust load alleviation on a large transport airplane," *Journal of Aircraft*, vol. 53, no. 6, pp. 1932–1946, 2016.

[49] P. Apkarian and P. Gahinet, "A convex characterization of gain-scheduled H_∞ controllers," *IEEE Transactions on Automatic Control*, vol. 40, no. 5, pp. 853–864, 1995.

[50] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled H_∞ control of linear parameter-varying systems: a design example," *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.

[51] D. J. Stilwell and W. J. Rugh, "Stability preserving interpolation methods for the synthesis of gain scheduled controllers," *Automatica*, vol. 36, no. 5, pp. 665–671, 2000.

[52] H. J. Hassig, "An approximate true damping solution of the flutter equation by determinant iteration," *Journal of Aircraft*, vol. 8, no. 11, pp. 885–889, 1971.

[53] B. Mokrani, F. Palazzo, J. E. Mottershead, and S. Fichera, "Multiple-input multiple-output experimental aeroelastic control using a receptance-based method," *AIAA Journal*, vol. 57, no. 7, pp. 3066–3077, 2019.

[54] D. C. Kammer, "Sensor placement for on-orbit modal identification and correlation of large space structures," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 2, pp. 251–259, 1991.

[55] W. L. Poston and R. H. Tolson, "Maximizing the determinant of the information matrix with the effective independence method," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 6, pp. 1513-1514, 1992.

[56] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 4163766, 11 pages
https://doi.org/10.1155/2021/4163766

Research Article

Passive Sonar Multiple-Target Tracking with Nonlinear Doppler and Bearing Measurements Using Multiple Sensors

Xiaohua Li,[1,2] **Bo Lu**,[1,2] **Wasiq Ali**,[3] **Jun Su**,[3] **and Haiyan Jin**[1,2]

[1]*School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China*
[2]*Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an 710048, China*
[3]*School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China*

Correspondence should be addressed to Xiaohua Li; lixiaohua@xaut.edu.cn

Received 22 June 2021; Accepted 21 September 2021; Published 11 October 2021

Academic Editor: Mahmut Reyhanoglu

The major advantage of the passive multiple-target tracking is that the sonars do not emit signals and thus they can remain covert, which will reduce the risk of being attacked. However, the nonlinearity of the passive Doppler and bearing measurements, the range unobservability problem, and the measurement to target data association uncertainty make the passive multiple-target tracking problem challenging. To deal with the target to measurement data association uncertainty problem from multiple sensors, this paper proposed a batch recursive extended Rauch-Tung-Striebel smoother- (RTSS-) based probabilistic multiple hypothesis tracker (PMHT) algorithm, which can effectively handle a large number of passive measurements including clutters. The recursive extended RTSS which consists of a forward filter and a backward smoothing is used to deal with the nonlinear Doppler and bearing measurements. The target range unobservability problem is avoided due to using multiple passive sensors. The simulation results show that the proposed algorithm works well in a passive multiple-target tracking system under dense clutter environment, and its computing cost is low.

1. Introduction

Passive multiple-target tracking has gained more and more attention in the fields of military and civilian, such as navigation, monitoring and early warning, and salvage [1–3]. How to discover targets timely and to track targets accurately becomes one of the hot topics. The aim of the multiple-target tracking is to estimate the expected targets' states, such as position, velocity, and acceleration, from the linear or nonlinear measurements [4–6]. The advantage of the passive target tracking is that the sonar can remain covert, which will reduce the risk of being attacked. The challenges for underwater passive multiple-target tracking include that the measurements are usually nonlinear, the target range may be unobservable, and the measurement to target data association is complex [5, 6].

In general, the passive measurements include bearing, Doppler, and bearing rate. So the passive multiple-target tracking is a typically nonlinear tracking problem [7–15]. One method to handle the nonlinear measurements is the pseudo linearization estimation (PLE) algorithm [16–18]. The other approach is the recursive nonlinear Bayesian filter and smoother, such as the extended Kalman filter (EKF) [19–22], the unscented Kalman filter (UKF) [23, 24], the cubature Kalman filter (CKF) [11, 12], and the particle filter (PF) [16]. EKF locally linearizes the targets' state and measurement equations using the first-order Taylor series expansion of the nonlinear transformations around the predicted target state. The merit of the EKF is that it is very easy to understand and apply to target tracking problem duo to its linear approximations to the nonlinear system or measurement function, and its computational cost is lower than that of other nonlinear filter, such as UKF and PF. The disadvantage of the EKF is that the tracking performance may be not good if the system model is seriously nonlinear or non-Gaussian.

The UKF uses a deterministic sampling method to capture targets' posterior distribution of mean and covariance based on the unscented transform. So the UKF has better tracking capability than the EKF to some extent, but its

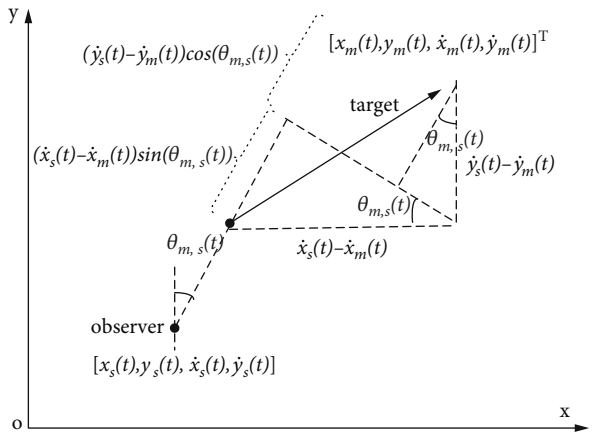


Figure 1: Multiple-target tracking scenario using Doppler and bearing measurement.

computational cost is larger than that of the EKF. The PF algorithm forms the Monte Carlo approximation to the solution of the Bayesian filter. It uses a set of particle samples to approximate the target distribution. The PF is used in the cases that the tracking system is highly nonlinear or non-Gaussian. The Bayesian filters only estimate the state of the targets from the history of measurements. On the contrary, the Bayesian smoother can use the forthcoming measurements to estimate the current state of targets. Corresponding to the Bayesian filter, the Bayesian smoother includes Rauch-Tung-Striebel smoother (RTSS), extended RTSS, unscented RTSS, and particle smoother [25].

The other challenge for passive sonar multiple-target tracking is the complexity and particularity of the underwater environment, which is characterized by many of false alarms caused by reverberation and multipath effect [3, 4]. Furthermore, the low target detection will cause tracking uncertainty in target location. All of those will cause a data association uncertainty problem. To handle the data association uncertainty problem, some algorithms are proposed, such as the multiple hypothesis trackers (MHT) [26], the joint PDA filter (JPDA) [27], the probabilistic MHT (PMHT) [28, 29], the random finite set framework-based probability hypothesis density (PHD) [30], the cardinalized PHD (CPHD) [31–33], and multi-Bernoulli filter (MBF). The MHT makes all the data association hypotheses probability optimal. The JPDA makes multiple hypotheses into a single hypothesis and performs the Kalman update with composite measurements. The PMHT is based on the expectation maximization (EM), which optimizes the multiple-target states' maximum a posteriori (MAP) estimation [34–36]. Different from the data association algorithms, the PHD, CPHD, and MBF are based on the random finite set theory, which makes all the measurements a measurement set and all the targets a target set.

The information entropy theories are also used to estimate the target states. The fuzzy c-means clustering method based on maximum information entropy and the probabilistic data association filter (PDA) is proposed in [37], which uses a value optimized by the maximum information entropy to represent the measurement to target association probability. The multiple-target tracking problem is also solved by the maximum entropy intuitionistic fuzzy data association [38], cross entropy [39], maximum fuzzy entropy-based Gaussian clustering algorithm [40], entropy distribution and game theory based on the probability hypothesis density (PHD) method [41], maximum entropy fuzzy based on the fire-fly and PF [42], and the distributed cross entropy-based δ-generalized labelled multi-Bernoulli filter [43].

Table 1: Batch recursive multiple-sensor PMHT.

1. Initialization
Initialize the target states $\hat{\mathbf{x}}_m(0 \mid 0)$ and $\mathbf{P}_m(0 \mid 0)$.
2. Set the EM iteration $n = 1$. Calculate the target state prediction $\mathbf{x}_m^{(n)}(t \mid t-1)$ and state covariance $\mathbf{P}_m^{(n)}(t \mid t-1)$.
3. Calculate the posterior association probabilities $w_{m,r}^{(n)}(t,s)$ in (15).
4. Evaluate the synthetic Doppler and bearing measurements $\tilde{\mathbf{z}}_{m,s}(t)$ and covariance $\tilde{\mathbf{R}}_{m,s}(t)$ in (20) and (21).
5. Evaluate the innovation covariance and filter gain for each target and passive sensor.
6. Update the target state $\mathbf{x}_m^{(n)}(t \mid t)$ and state prediction covariance $\mathbf{P}_m^{(n)}(t \mid t)$ for each target m according to the extended RTSS.
7. Forward $n = n + 1$. Repeat the EM algorithm of steps 3 to 6 until the iteration convergent.

As for the target range unobservability problem, the target range is observable from bearing and Doppler measurements if and only if the bearings are not constant [44, 45]. So in order to avoid the range unobservability problem, this paper uses multiple passive sensors to track targets. In addition, the target tracking accuracy of using multiple sensors is better than that of the single sensor generally.

The most commonly used passive measurement is bearing. In this paper, in order to improve the multiple targets' range observability and tracking performance, we introduce the nonlinear Doppler measurement and use multiple sensors. The extended RTSS method is used to deal with the nonlinear Doppler and bearing measurements. The batch recursive multiple-sensor PMHT algorithm is used to handle the measurement to target data association complexity problem.

The remainder of this paper is as follows. The passive multiple-target tracking system model and measurement model are given in Section 2. Section 3 develops the multiple-sensor PMHT algorithm which is suitable for multiple-target tracking under dense clutter environment. The simulation result is given in Section 4. At last, a summary is given in Section 5.

2. System Model and Measurement Model

We consider the passive multiple-target tracking problem in a two-dimensional space.

2.1. Tracking System Model. Assume there are M targets in the tracking space. The $m^{\text{th}}(m = 1, 2, \cdots, M)$ target's state is $\mathbf{x}_m(t) = (x_m(t), \dot{x}_m(t), y_m(t), \dot{y}_m(t))^T$, where $x_m(t)$ and $y_m(t)$ are the location of target m in the x and y coordinate, respectively,

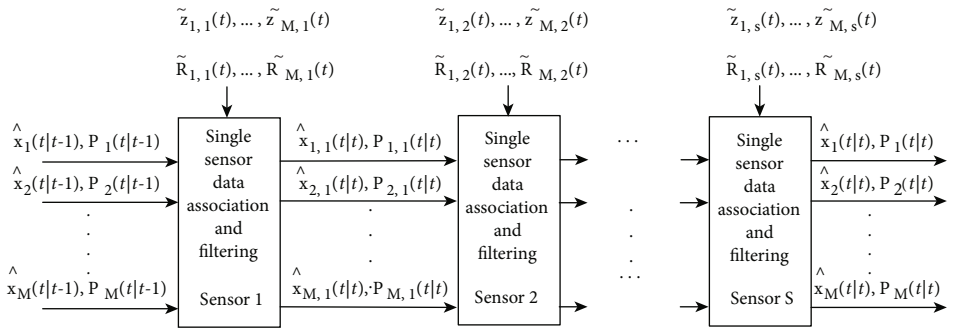


Figure 2: Sequential fusion implementation of multiple sensors.

Table 2: The five targets' initial position and velocity.

Target	Position	Velocity
1	(800, 2000) m	(-22, -20) m/s
2	(1500, 1500) m	(20, -30) m/s
3	(500, -500) m	(23, 17) m/s
4	(0, 0) m	(-13, -27) m/s
5	(500, -1000) m	(13, -20) m/s

and $\dot{x}_m(t)$ and $\dot{y}_m(t)$ are the velocity of target m. Assume the tracking time is from $t = 1$ to $t = N$. The overview of the multiple-target tracking using Doppler and bearing measurements is shown in Figure 1.

Assume that all the targets move according to the following constant velocity (CV) model or constant acceleration (CA) model [46].

$$\mathbf{x}_m(t) = \mathbf{F}\mathbf{x}_m(t-1) + \mathbf{u}_m(t), \tag{1}$$

where $\mathbf{u}_m(t)$ is the system process noise which is assumed as Gaussian white noise. $\mathbf{F}$ is the system state transition matrix. For the CV and CA models, $\mathbf{F}$ is denoted as $\mathbf{F}_{\mathrm{CV}}$ and $\mathbf{F}_{\mathrm{CA}}$, respectively, and

$$\mathbf{F}_{\mathrm{CV}} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

$$\mathbf{F}_{CA} = \begin{bmatrix} 1 & \Delta t & 0.5\Delta t^2 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & 0.5\Delta t^2 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where Δt is the sampling interval.

The process noise covariance matrices for the CV and CA models are denoted as $\boldsymbol{Q}_{\mathrm{CV}}$ and $\boldsymbol{Q}_{\mathrm{CA}}$, respectively, which are given by

$$\boldsymbol{Q}_{\mathrm{CV}} = \delta_p^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^2}{2} & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t^2 & 0 & 0 \\ 0 & 0 & \frac{\Delta t^4}{4} & \frac{\Delta t^2}{2} \\ 0 & 0 & \frac{\Delta t^2}{2} & \Delta t^2 \end{bmatrix}, \tag{4}$$

$$\boldsymbol{Q}_{\mathrm{CA}} = \delta_p^2 \begin{bmatrix} \frac{\Delta t^5}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} & 0 & 0 & 0 \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\Delta t^5}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} \\ 0 & 0 & 0 & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ 0 & 0 & 0 & \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}, \tag{5}$$

where δ_p^2 is the process noise intensity.

2.2. Measurement Model for Multiple Sensors. Assume there are S passive sensors in the target tracking space, and the s^{th} sensor's states is $\mathbf{x}_s(t) = (x_s(t), \dot{x}_s(t), y_s(t), \dot{y}_s(t))^T$, in which $x_s(t)$ and $y_s(t)$ are the location of the sensor s and $\dot{x}_s(t)$ and $\dot{y}_s(t)$ are the velocity of passive sensor s.

The Doppler and bearing measurements are nonlinear with respect to targets' states and sensors' states, which are given by

$$\mathbf{z}_{m,s}(t) = \mathbf{h}_{m,s}(t) + \mathbf{w}_{m,s}(t), \tag{6}$$

in which $\mathbf{w}_{m,s}(t)$ is the measurement noise with a covariance matrix $\mathbf{R}_{m,s}(t)$.

The Doppler and bearing measurement function $\mathbf{h}_{m,s}(t)$ is given by

$$\mathbf{h}_{m,s}(t) = \left[\left[1 - \frac{(\dot{x}_m(t) - \dot{x}_s(t))\sin\theta_{m,s}(t) + (\dot{y}_m(t) - \dot{y}_s(t))\cos\theta_{m,s}(t)}{c}\right] f_0 \arctan\left(\frac{x_m(t) - x_s(t)}{y_m(t) - y_s(t)}\right)\right], \tag{7}$$

where c is the sound speed in water, f_0 is targets' radiation frequency, and $\theta_{m,s}(t)$ is bearing from target m and sensor s.

3. Multiple-Sensor PMHT

3.1. PMHT for Multiple Sensors. The following notations are used in this section.

S is the total number of passive observers.

M is the total number of targets.

n_t is the number of measurements.

T is the total number of tracking time.

$\mathbf{X}(t) = \{\mathbf{x}_{m,s}(t)\}$ is the set of target states at time t.

$\mathbf{Z}(t) = \{\mathbf{z}_{m,s}(t)\}$ is the set of measurements at time t.

$\mathbf{K}(t) = \{k_{r,s}(t)\}$ is the set of measurement-target assignments at time t.

$\mathbf{X} = \{\mathbf{X}(t)\}$ is the set of the target state for time $1, 2, \cdots, T$.

$\mathbf{Z} = \{\mathbf{Z}(t)\}$ is the set of measurements for time $1, 2, \cdots, T$.

$\mathbf{K} = \{\mathbf{K}(t)\}$ is the set of measurement-target assignments for time $1, 2, \cdots, T$.

The PMHT is a Bayesian framework-based batch recursive algorithm, which obtains the maximum a posteriori estimation of the target states based on the expectation maximization (EM) method [47].

Let $k_{r,s}(t) = m$ indicate that measurement r from sensor s at time t is associated with target m. Assume the measurement to target assignment is independent from each other.

Define the prior probability of the r^{th} measurement from the s^{th} sensor and m^{th} target as

$$p(k_{r,s}(t) = m) = \pi_m(t). \tag{8}$$

The MAP estimate of $\mathbf{X}$ is

$$\widehat{\mathbf{X}}_{\text{MAP}} = \arg \max_{\mathbf{X}} \mathrm{E}\{\log(p(\mathbf{X} \mid \mathbf{Z}))\}. \tag{9}$$

In order to calculate the MAP, define the following auxiliary function as

$$Q\left(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}\right) = \int_{\mathbf{K}} \log\left(p\left(\mathbf{X}^{(n+1)}, \mathbf{K} \mid \mathbf{Z}\right)\right) \cdot p\left(\mathbf{K} \mid \mathbf{X}^{(n)}, \mathbf{Z}\right) d\mathbf{K}, \tag{10}$$

where n is the number of EM iteration.

The goal is to maximize the auxiliary function over $\mathbf{X}^{(n+1)}$ using an initialized target state $\mathbf{X}^{(0)}$. In each EM iteration step, the goal of the PMHT is to solve the following equation.

$$\mathbf{X}^{(n+1)} = \arg \max_{\mathbf{X}^{(n+1)}} Q\left(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}\right). \tag{11}$$

We have

$$p(\mathbf{X}, \mathbf{Z}) = \prod_{m=1}^{M} p(\mathbf{x}_m(1)) \prod_{t=2}^{\mathrm{T}} \prod_{m=1}^{M} p(\mathbf{x}_m(t) \mid \mathbf{x}_m(t-1)) \cdot \prod_{s=1}^{S} \prod_{t=1}^{\mathrm{T}} \prod_{r=1}^{n_t} \left[\sum_{p=1}^{P} \pi_p N\left\{\mathbf{z}_{r,s}(t); \widehat{\mathbf{z}}_{m,s}(t), \mathbf{R}_{m,s}(t)\right\}\right], \tag{12}$$

$$p(\mathbf{K}, \mathbf{Z}, \mathbf{X}) = \prod_{m=1}^{M} p(\mathbf{x}_m(1)) \prod_{t=2}^{\mathrm{T}} \prod_{m=1}^{M} p(\mathbf{x}_m(t) \mid \mathbf{x}_m(t-1)) \cdot \prod_{s=1}^{S} \prod_{t=1}^{\mathrm{T}} \prod_{r=1}^{n_t} \pi_m N\left\{\mathbf{z}_{r,s}(t); \widehat{\mathbf{z}}_{m,s}(t), \mathbf{R}_{m,s}(t)\right\}. \tag{13}$$

According to the conditional probabilistic theory,

$$p(\mathbf{K} \mid \mathbf{Z}, \mathbf{X}) = \frac{p(\mathbf{K}, \mathbf{Z}, \mathbf{X})}{p(\mathbf{X}, \mathbf{Z})} = \prod_{s=1}^{S} \prod_{t=1}^{N} \prod_{r=1}^{n_t} w_{m,r}^{(n)}(t, s). \tag{14}$$

From (12), (13), and (14), the posterior association probability is

$$w_{m,r}^{(n)}(t, s) \triangleq p(k_m(t) \mid \mathbf{x}_m(t), \mathbf{z}_{r,s}(t)) = \frac{\pi_m N\left\{\mathbf{z}_{r,s}(t); \widehat{\mathbf{z}}_{m,s}(t), \mathbf{R}_{m,s}(t)\right\}}{\sum_{m=1}^{M} \pi_m N\left\{\mathbf{z}_{r,s}(t); \widehat{\mathbf{z}}_{m,s}(t), \mathbf{R}_{m,s}(t)\right\}}, \tag{15}$$

in which

$$\widehat{\mathbf{z}}_{m,s}(t) = h(\mathbf{x}_m(t), \mathbf{x}_s(t)). \tag{16}$$

According to the conditional probabilistic theory,

$$p(\mathbf{K}, \mathbf{X} \mid \mathbf{Z}) = \frac{p(\mathbf{K}, \mathbf{X}, \mathbf{Z})}{p(\mathbf{Z})}. \tag{17}$$

Substituting equation (17) into equation (10), the defined auxiliary function is calculated as

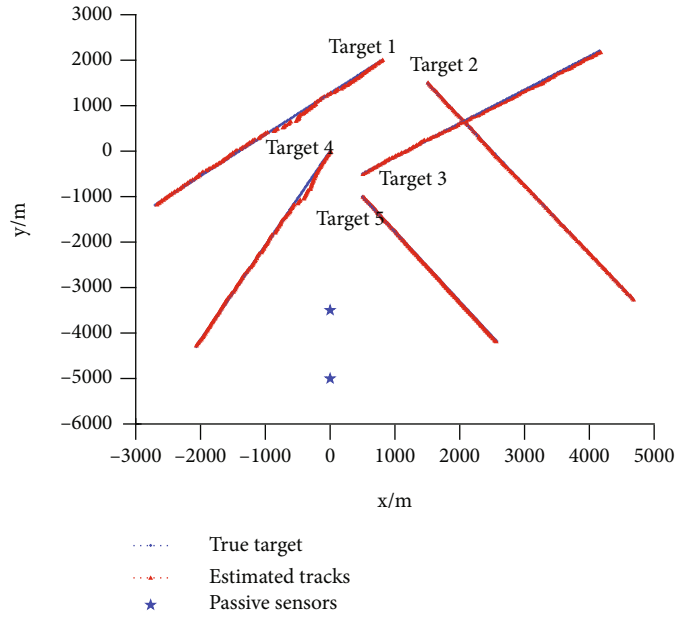


Figure 3: The tracking scenario of multiple targets, given two passive static sensors, true target trajectories, and estimated tracks.

$$Q\left(\mathbf{X}^{(n+1)};\mathbf{X}^{(n)}\right)=\int_{\mathbf{K}}\log\left(p\left(\mathbf{K},\mathbf{X}^{(n+1)},\mathbf{Z}\right)\right)\cdot p\left(\mathbf{K}\mid\mathbf{X}^{(n)},\mathbf{Z}\right)d\mathbf{K}-\log p(\mathbf{Z}). \tag{18}$$

The $\log p(\mathbf{Z})$ has no effect with regard to maximizing the auxiliary function $Q(\mathbf{X}^{(n+1)};\mathbf{X}^{(n)})$ in (18). So it can be removed from (18).

So the auxiliary function is given by

$$Q\left(\mathbf{X}^{(n+1)};\mathbf{X}^{(n)}\right)=\sum_{s=1}^{S}\sum_{t=1}^{\mathrm{T}}\sum_{r=1}^{n_t}\sum_{m=1}^{Q_{\Pi}=M}\log\pi_m w_{m,r}^{(n)}(t,s)+\underbrace{\log\left(\prod_{m=1}^{M}p\left(\mathbf{x}_m^{(n+1)}(1)\right)\prod_{t=2}^{\mathrm{T}}p\left(\mathbf{x}_m^{(n+1)}(t)\mid\mathbf{x}_m^{(n+1)}(t-1)\right)\right)+\sum_{s=1}^{S}\sum_{t=1}^{\mathrm{T}}\sum_{r=1}^{n_t}\sum_{m=1}^{M}\log p\left(\mathbf{z}_r(t)\mid\mathbf{x}_m^{(n)}(t)\right)w_{m,r}^{(n)}(t,s)}_{Q_{\mathbf{X}}}. \tag{19}$$

In order to maximize the auxiliary function, we compute its derivative and set the derivative to zero.

It turns out that the $\mathbf{X}^{(n+1)}$ is given by applying the extended RTSS method, in which the synthetic measurement $\tilde{\mathbf{z}}_{m,s}(t)$ and $\tilde{\mathbf{R}}_{m,s}(t)$ are calculated by

$$\tilde{\mathbf{z}}_{m,s}(t)=\frac{\sum_{r=1}^{n_t}w_{m,r}^{(n)}(t,s)\mathbf{z}_{r,s}(t)}{\sum_{r=1}^{n_t}w_{m,r}^{(n)}(t,s)}, \tag{20}$$

$$\tilde{\mathbf{R}}_{m,s}(t)\frac{\mathbf{R}_{m,s}(t)}{\sum_{r=1}^{n_t}w_{m,r}^{(n)}(t,s)}. \tag{21}$$

The implementation of the batch recursive multiple-sensor extended RTSS-based PMHT is depicted in Table 1.

3.2. Recursive Extended RTSS. The recursive extended RTSS is a linearized RTS smoother, which is based on analogous approximation to the EKF. It is a Gaussian approximation to the Bayesian smoother for the nonlinear target state and measurement model. The extended RTSS firstly performs the extended Kalman filter; then, a Kalman smoother is applied. That is, the extended RTSS consists of a forward filter and a backward smoother, which is summarized as follows:

Forward filter:

For $i=1,\cdots,T$,

Time update

$$\hat{\mathbf{x}}_m(t\mid t-1)=\mathbf{F}\hat{\mathbf{x}}_m(t-1\mid t-1), \tag{22}$$

$$\mathbf{P}_m(t\mid t-1)=\mathbf{F}\mathbf{P}_m(t-1\mid t-1)\mathbf{F}^{\mathrm{T}}+\mathbf{Q}_m(t). \tag{23}$$

Measurement update

$$\hat{\mathbf{x}}_m(t \mid t) = \hat{\mathbf{x}}_m(t \mid t-1) + \mathbf{G}_{m,s}(t)[\mathbf{z}_{m,s}(t) - \mathbf{h}_{m,s}(t \mid t-1)], \tag{24}$$

$$\mathbf{P}_m(t \mid t) = (\mathbf{I} - \mathbf{G}_{m,s}(t)\mathbf{H}_{m,s}(t)) \cdot \mathbf{P}_m(t \mid t-1) \cdot (\mathbf{I} - \mathbf{G}_{m,s}(t)\mathbf{H}_{m,s}(t))^{\mathrm{T}} + \mathbf{G}_{m,s}(t)\mathbf{R}_{m,s}(t)\mathbf{G}_{m,s}^{T}(t), \tag{25}$$

where the filter gain is

$$\mathbf{G}_{m,s}(t) = \mathbf{P}_m(t \mid t-1)\mathbf{H}_{m,s}^{T}(t) \cdot \left[\mathbf{H}_{m,s}(t)\mathbf{P}_m(t \mid t-1) \cdot \mathbf{H}_{m,s}^{T}(t) + \mathbf{R}_{m,s}(t)\right]^{-1}, \tag{26}$$

$$\mathbf{H}_{m,s}(t) = \left.\frac{\partial \mathbf{h}_{m,s}(t)}{\partial \mathbf{x}_m(t)}\right|_{\mathbf{x}_m(t)=\hat{\mathbf{x}}_m(t|t-1)}. \tag{27}$$

Backward smoothing:

For $i = T\text{-}1, \cdots, 1$,

$$\hat{\mathbf{x}}_m(t \mid \mathrm{T}) = \hat{\mathbf{x}}_m(t \mid t) + \mathbf{C}_m(t)[\hat{\mathbf{x}}_m(t+1 \mid \mathrm{T}) - \hat{\mathbf{x}}_m(t+1 \mid t)], \tag{28}$$

$$\mathbf{P}_m(t \mid \mathrm{T}) = \mathbf{P}_m(t \mid t) + \mathbf{C}_m(t)[\mathbf{P}_m(t+1 \mid \mathrm{T}) - \mathbf{P}_m(t+1 \mid t)]\mathbf{C}_m^{T}(t), \tag{29}$$

where

$$\mathbf{C}_m(t) = \mathbf{P}_m(t \mid t)\mathbf{F}^{T}\mathbf{P}_m^{-1}(t+1 \mid t). \tag{30}$$

For the Doppler and bearing measurement, the measurement Jacobian matrix is given by

$$\mathbf{H}_{m,s}(t) = \left[\frac{\partial \mathbf{h}_{m,s}(t)}{\partial x_m(t)} \; \frac{\partial \mathbf{h}_{m,s}(t)}{\partial \dot{x}_m(t)} \; \frac{\partial \mathbf{h}_{m,s}(t)}{\partial y_m(t)} \; \frac{\partial \mathbf{h}_{m,s}(t)}{\partial \dot{y}_m(t)}\right] = \begin{bmatrix} \dfrac{y_m(t) - y_s(t)}{r^2} & 0 & -\dfrac{x_m(t) - x_s(t)}{r^2} & 0 \\ 0 & \dfrac{-f_0[x_m(t) - x_s(t)]}{cr} & 0 & \dfrac{-f_0[y_m(t) - y_s(t)]}{cr} \end{bmatrix}, \tag{31}$$

where $r = \sqrt{(x_m(t) - x_s(t))^2 + (y_m(t) - y_s(t))^2}$.

3.3. Data Fusion for Multiple Sensors. For the tracking problem using multiple sensors, we need to fuse the target and measurement information from multiple sensors. The most used data fusion methods are parallel fusion and sequential fusion [48].

In this paper, we use the sequential data fusion for multiple sensors which updates the target state and covariance by each sensor sequentially. The multiple sensors deal with the measurements one after another with an intermediate target state and covariance at one time.

Let $\hat{\mathbf{x}}_{m,s}(t \mid t)$ and $\mathbf{P}_{m,s}(t \mid t)$ denote the target state estimation and corresponding covariance processed by the sensor s. The Doppler and bearing measurements from the first passive sensor are used to calculate the first intermediate target state $\hat{\mathbf{x}}_{m,1}^{(n)}(t \mid t)$ and covariance $\mathbf{P}_{m,1}^{(n)}(t \mid t)$ for each of the targets using a single sensor filter. Then, the measurements from the next passive sensor are used to update the intermediate target state and covariance as follows:

$$\hat{\mathbf{x}}_{m,s}(t \mid t) = \hat{\mathbf{x}}_{m,s-1}(t \mid t) + \mathbf{G}_{m,s}(t)[\mathbf{z}_{m,s}(t) - \mathbf{h}_{m,s}(t \mid t)], \tag{32}$$

$$\mathbf{P}_{m,s}(t \mid t) = (\mathbf{I} - \mathbf{G}_{m,s}(t)\mathbf{H}_{m,s}(t)) \cdot \mathbf{P}_{m,s-1}(t \mid t) \cdot (\mathbf{I} - \mathbf{G}_{m,s}(t)\mathbf{H}_{m,s}(t))^{\mathrm{T}} + \mathbf{G}_{m,s}(t)\mathbf{R}_{m,s}(t)\mathbf{G}_{m,s}^{T}(t), \tag{33}$$

where $\mathbf{G}_{m,s}(t)$ is the filter gain, and

$$\hat{\mathbf{x}}_{m,0}(t \mid t) = \hat{\mathbf{x}}_m(t \mid t-1), \mathbf{P}_{m,0}(t \mid t) = \mathbf{P}_m(t \mid t-1), \tag{34}$$

$$\hat{\mathbf{x}}_m(t \mid t) = \hat{\mathbf{x}}_{m,\mathrm{S}}(t \mid t), \mathbf{P}_m(t \mid t) = \mathbf{P}_{m,\mathrm{S}}(t \mid t). \tag{35}$$

The sequential fusion implementation of multiple sensors is shown in Figure 2.

4. Simulation

Assume the targets move in the two-dimensional space, and there are two static passive sensors. The two passive sensors deal with Doppler and measurements from targets and clutter. The two passive sensors located at (0, -3500) m and (0, -5500) m, respectively.

Assume that the clutter number is a Poisson distribution, and they are uniformly distributed in the Doppler and bearing measurement space. The average number of clutter in

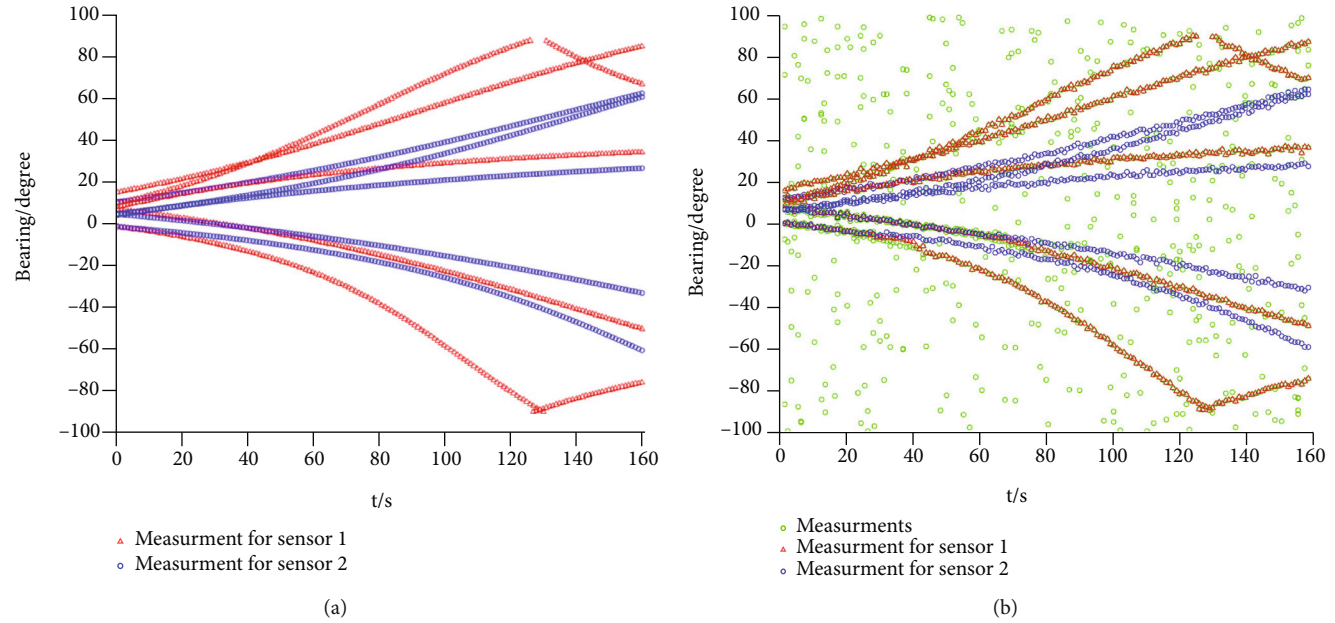


Figure 4: Bearing measurements for two static passive observers: (a) the true bearing measurements without clutter; (b) synthetic bearing measurements with clutter.

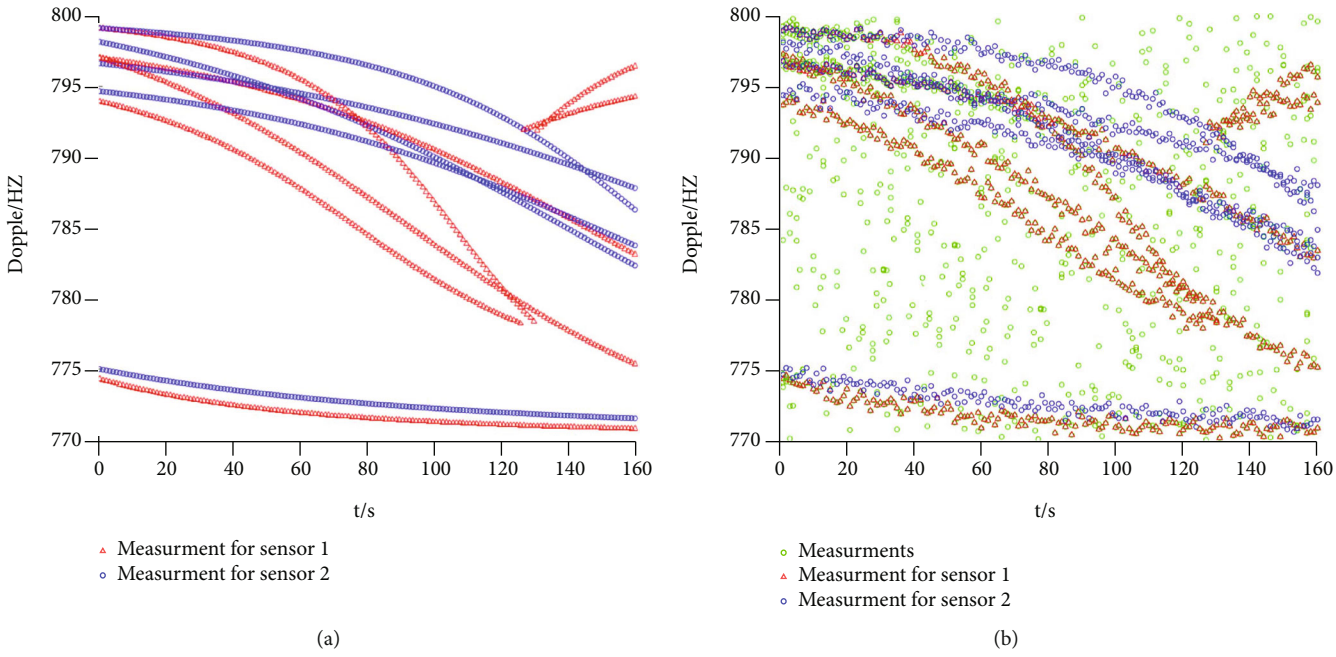


Figure 5: Doppler measurements for two static passive sensors: (a) the true Doppler measurements without clutter; (b) synthetic Doppler measurements with clutter.

each sampling scan is 20 in the Doppler and bearing measurements domain. The target detection probability is the same for all targets; here, we set it to 0.8. The tracking time is 160 s with a sampling interval 1 s. The Monte Carlo run is 200. The process noise intensity is 1 m. The Doppler measurement noise variance is 1 Hz, and the bearing noise variance is 0.8°.

4.1. Case of the CV Model. The five targets' initial position and velocity for the CV model are given in Table 2.

For the CV model, the simulated initial position $\mathbf{p}$ and velocity $\mathbf{v}$ are generated from the position $\mathbf{p}_0$ and velocity $\mathbf{v}_0$ of ground truth with an estimation error such that $\mathbf{p} = \mathbf{p}_0 + \mathbf{e}_1$ and $\mathbf{v} = \mathbf{v}_0 + \mathbf{e}_2$, where $\mathbf{e}_1 = 30$ m and $\mathbf{e}_2 = 2$ m/s.

The tracking scenario of true target trajectories and PMHT-estimated trajectories for the CV model is shown in Figure 3. The true Doppler and measurements for two static passive sensors without clutter and the PMHT synthetic Doppler and bearing measurements under dense clutter are given in Figures 4 and 5, respectively.

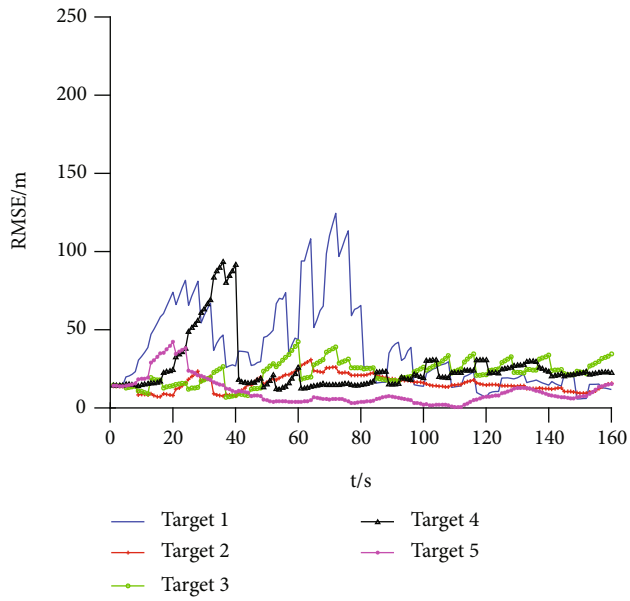


Figure 6: The position RMSE versus time scans of five targets.

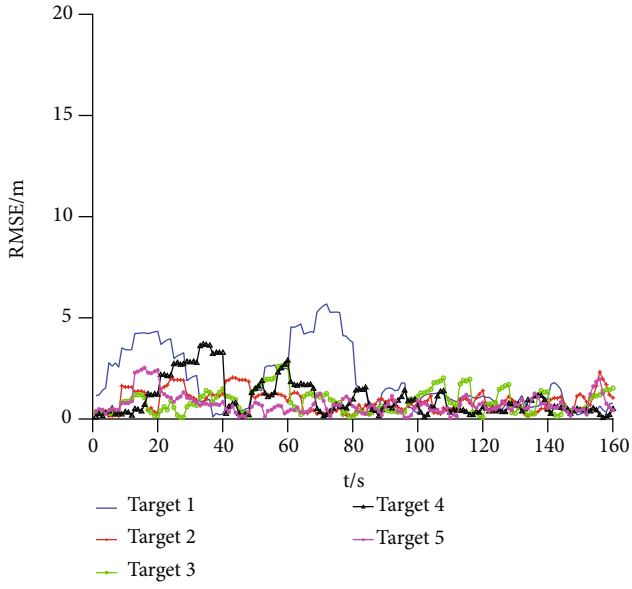


Figure 7: The velocity RMSE versus time scans of five targets.

In Figure 3, the extended RTSS-based PMHT algorithm can track all the five targets effectively; even if the target 1 and target 4 tracks are away from the true trajectory, the proposed algorithm can get the true target track after several time scans in the middle tracking time scan.

As seen in Figures 4 and 5, the density of Doppler and bearing measurement from clutter is high, and the synthetic Doppler and bearing measurement of the proposed algorithm broadly consists of the true Doppler and bearing measurements without clutter, which means that the proposed algorithm has good declutter ability.

The position and velocity RMSE of the five targets are given in Figures 6 and 7. Accordingly, the average position and velocity RMSE are shown in Tables 3 and 4. As shown

Table 3: The average position RMSE of five targets.

Targets	1	2	3	4	5
RMSE (m)	34.94	9.93	23.71	20.23	13.25

Table 4: The average velocity RMSE of five targets.

Targets	1	2	3	4	5
RMSE (m/s)	2.116	0.991	1.216	1.002	0.590

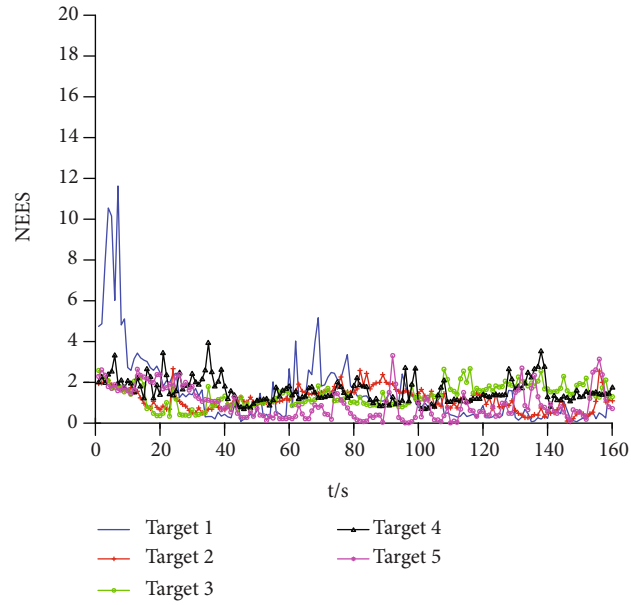


Figure 8: The average NEES versus time scans for five targets.

in Figures 6 and 7, the position RMSE and velocity RMSE for target 1 are increasing in the middle sampling scans, and as time goes on, the RMSE is decreased to a low lever, which is similar to other targets. This also can be seen from Tables 3 and 4. The average position and velocity RMSE is small which can meet the tracking accuracy requirement.

This paper uses the average normalized estimation error squared (ANEES) to evaluate the consistency of the proposed algorithm. For one target, the ANEES is defined as follows:

$$\mathrm{ANEES}(t) = \frac{1}{N_m}\sum_{i=1}^{N_m}\left(\mathbf{x}^i(t) - \hat{\mathbf{x}}^i(t)\right)^T \mathbf{P}^i(t)^{-1}\left(\mathbf{x}^i(t) - \hat{\mathbf{x}}^i(t)\right), \tag{36}$$

where N_m is Monte Carlo runs, $\mathbf{x}^i(t)$ is the true target state, $\hat{\mathbf{x}}^i(t)$ is the estimated target state, and $\mathbf{P}^i(t)$ is the target state covariance.

The ANEES for fives targets is shown in Figure 8. As can be seen in Figure 8, the consistency of the proposed algorithm is good.

4.2. Case of the CA Model. The four targets' initial position, velocity, and acceleration for the CA model are given in Table 5.

Table 5: The five targets' initial position and velocity.

Target	Position	Velocity	Acceleration
1	(800, 2000) m	(-22, -20) m/s	(0.3, 0) m/s^2
2	(1500, 1500) m	(20, -30) m/s	(0, -0.3) m/s^2
3	(500, -500) m	(23, 17) m/s	(0.4, 0.4) m/s^2
4	(0, 0) m	(-13, -27) m/s	(0.1, 0.1) m/s^2

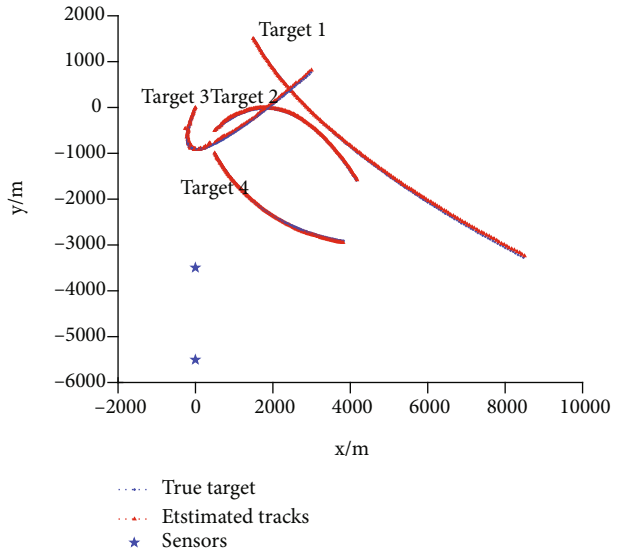


Figure 9: The tracking scenario of multiple targets, given two passive static sensors, CA model, true target trajectories, and estimated tracks.

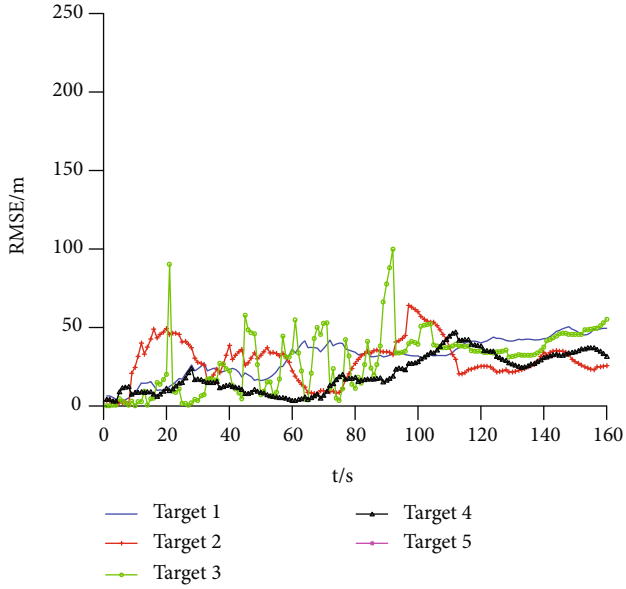


Figure 10: The position RMSE for four targets (CA model).

For the CA model, the simulated initial position $\mathbf{p}$, velocity $\mathbf{v}$, and acceleration $\mathbf{a}$ are generated from the position $\mathbf{p}_0$, velocity $\mathbf{v}_0$, and acceleration $\mathbf{a}_0$ of ground truth with an estimation error such that $\mathbf{p} = \mathbf{p}_0 + \mathbf{e}_1$, $\mathbf{v} = \mathbf{v}_0 + \mathbf{e}_2$, and $\mathbf{a} = \mathbf{a}_0 + \mathbf{e}_3$, where $\mathbf{e}_1 = 30$ m, $\mathbf{e}_2 = 2$ m/s, and $\mathbf{e}_3 = 0.5$ m/s^2.

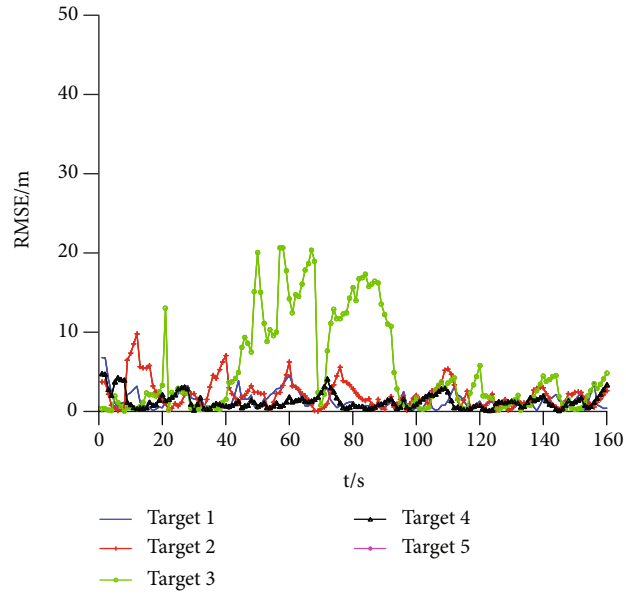


Figure 11: The velocity RMSE for four targets (CA model).

The tracking scenario of the CA model for target true trajectories and the estimated tracks is shown in Figure 9. Similar to the CV model, the estimated tracks are consistent with the true targets' trajectories.

The position RMSE and velocity RMSE for the CA model are shown in Figures 10 and 11. As shown in Figures 10 and 11, the position RMSE and velocity RMSE for target 3 are increasing in the middle sampling scans, and as time goes on, the RMSE is decreased to a low lever. The average position and velocity RMSE for the CA model is small which can meet the tracking accuracy requirement.

5. Conclusion

The major advantage of the passive sonars multiple-target tracking is that the sonars do not emit signals, and thus they can remain covert, which will reduce the risk of being attacked. But there are also challenges. Firstly, the Doppler and bearing measurements are nonlinear which makes the multiple-target tracking difficult. Secondly, the target states may be unobservable. Thirdly, the underwater environment is with dense clutter which will cause the measurement to target data association uncertainty problem. To deal with those problems, this paper proposed the extended RTSS-based batch PMHT method for multiple sensors and applied it to the passive multiple-sensor tracking system under dense clutter environment. This paper uses the extended RTSS algorithm to handle the nonlinear Doppler and bearing measurements. Multiple passive sonars are used to avoid the target state range unobservable problem. The multiple-sensor batch PMHT is used to deal with the data association uncertainty problem under dense clutter. The experiment results demonstrated that the proposed extended RTSS-based multiple-sensor PMHT algorithm can track multiple targets efficiently in the dense clutter environment, and the computing time is low.

Data Availability

The simulation condition data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (61703333, 62076201, and U1934222), the Natural Science Basic Research Plan in Shaanxi Province of China (2019JQ-746, 2019-740), the Key Laboratory of Shaanxi Provincial Department of Education (20JS088), and the Science and Technology Project of Beilin District (GX2017).

References

[1] M. S. Arulampalam, B. Ristic, N. Gordon, and T. Mansell, "Bearings-only tracking of maneuvering targets using particle filters," *EURASIP J. Appl. Signal Process.*, vol. 15, pp. 2351–2365, 2004.

[2] B. Ristic and M. S. Arulampalam, "Tracking a manoeuvring target using angle-only measurements: algorithms and performance," *Signal Processing*, vol. 83, no. 6, pp. 1223–1238, 2003.

[3] X. Chen, W. Li, Q. Lu, P. Willett, and Q. Zhang, "Underwater acoustic channel tracking by multi-Bernoulli filter," in *IEEE 2018 OCEANS*, pp. 1–8, Kobe, Japan, 2018.

[4] W. H. Foy, "Position-location solutions by Taylor series estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-12, no. 2, pp. 187–194, 1976.

[5] M. Ye, B. D. O. Anderson, and C. Yu, "Bearing-only measurement self-localization, velocity consensus and formation control," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 2, pp. 575–586, 2017.

[6] T. L. Song, "Observability of target tracking with bearings only measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1468–1472, 1996.

[7] I. Arasaratnam, S. Haykin, and R. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.

[8] R. Zanetti, "Recursive update filtering for nonlinear estimation," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1481–1490, 2012.

[9] R. Van Der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, pp. 3461–3464, Salt Lake City, UT, USA, 2001.

[10] B. Coraluppi, M. S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter*, Artech House, Boston, MA, 2004.

[11] P. H. Leong, S. Arulampalam, T. A. Lamahewa, and T. D. Abhayapala, "A Gaussian-sum based cubature Kalman filter for bearings-only tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1161–1176, 2013.

[12] C. Hu, H. Lin, Z. Li, B. He, and G. Liu, "Kullback–Leibler divergence based distributed cubature Kalman filter and its application in cooperative space object tracking," *Entropy*, vol. 20, no. 2, p. 116, 2018.

[13] S. Konatowski, P. Kaniewski, and J. Matuszewski, "Comparison of estimation accuracy of EKF, UKF and PF filters," *Annual of Navigation*, vol. 23, no. 1, pp. 69–87, 2016.

[14] M. Arulampalam, S. Maskell, and N. Gordon, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[15] Z. M. Chen, Y. X. Qu, and B. Liu, "Improved multiple model target tracking algorithm based on particle filter," in *Proc. ImechE, Part G: J. Aerospace Engineering*, pp. 1–13, 2015.

[16] A. Miller and B. Miller, "Tracking of the UAV trajectory on the basis of bearing-only observations," in *Proc. 53rd IEEE Conf. Decision Control*, pp. 4178–4184, Los Angeles, CA, USA, 2014.

[17] A. Miller, "Developing algorithms of object motion control on the basis of Kalman filtering of bearing-only measurements," *Automation and Remote Control*, vol. 76, no. 6, pp. 1018–1035, 2015.

[18] A. Miller and B. Miller, "Underwater target tracking using bearing-only measurements," *Journal of Communications Technology and Electronics*, vol. 63, no. 6, pp. 643–649, 2018.

[19] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*, John Wiley & Sons, 2004.

[20] H. L. Van Trees and K. L. Bell, *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*, Wiley, Hoboken, NJ, USA, 2007.

[21] A. Haug, *Bayesian Estimation and Tracking: A Practical Guide*, John Wiley & Sons, Hoboken, NJ, USA, 2012.

[22] P. Stano, Z. Lendek, J. Braaksma, R. Babuska, C. de Keizer, and A. J. den Dekker, "Parametric Bayesian filters for nonlinear stochastic dynamical systems: a survey," *IEEE Transactions on Systems, Man, and Cybernetics Part B, Cybernetics*, vol. 43, no. 6, pp. 1607–1624, 2013.

[23] R. Zhan and J. Wan, "Iterated unscented Kalman filter for passive target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 1155–1163, 2007.

[24] S. Särkkä, *Bayesian Filtering and Smoothing*, Cambridge Univ. Press, Cambridge, U.K., 2013.

[25] T. Lang and G. Hayes, "Evaluation of an MHT-enabled tracker with simulated multistatic sonar data," Proc. IEEE Oceans Conf, Aberdeen, Scotland, Jun, 2007.

[26] S. Coraluppi, C. Carthel, and M. Micheli, "DMHT-based undersea surveillance: insights from MSTWG analysis and recent sea-trial experimentation," in *Proc. 11th Int. Conf. Infor. Fusion*, pp. 1783–1790, 2008.

[27] W. Blanding, P. Willett, and Y. Bar-Shalom, "ML-PDA: advances and a new multitarget approach," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, 2007.

[28] M. Wieneke and W. Koch, "The PMHT algorithm: solutions for some of its problems," in *Proc. SPIE Conf.: signal data process. Small targets Conf*, vol. 6699, pp. 1–12, 2007.

[29] C. Rago, P. Willett, and R. Streit, "A comparison of the JPDAF and PMHT tracking algorithms," in *Proc Int. Conf. Acoustics, Speech, Signal Proc*, pp. 3571–3574, 1995.

[30] M. Beard and S. Arulampalam, "Performance of PHD and CPHD filtering versus JIPDA for bearings-only multitarget tracking," in *Proc. 15th Int. Conf. Infor. Fusion*, pp. 542–549, Singapore, 2012.

[31] O. Erdinc, P. Willett, and C. S. The, "Gaussian mixture cardinalized PHD tracker on MSTWG and SEABAR'07 datasets," in *Proc. 11th Int. Conf. Infor. Fusion*, pp. 1791–1798, Cologne, Germany, 2008.

[32] K. Pikora and F. Ehlers, "Analysis of the FKIE passive radar data set with GMPHD and GMCPHD," in *Proc. 16th Int. Conf. Infor. Fusion*, pp. 272–279, Istanbul, Turkey, 2013.

[33] R. Georgescu and P. Willett, *The GM-CPHD applied to the corrected TON-blind, adjusted SEABAR07 and Metron multistatic sonar datasets*, Proc. SPIE, 2010.

[34] C. Rago, P. Willett, and R. Streit, "Direct data fusion using the PMHT," *Proc. American Control Conf.*, vol. 3, pp. 1698–1702, 1995.

[35] S. Schoenecker, P. Willett, and Y. Bar-Shalom, "Maximum likelihood probabilistic multi-hypothesis tracker applied to multistatic sonar data sets," Proc. SPIE, 2011.

[36] Q. Lu, K. Domrese, P. Willett, Y. Bar-Shalom, and K. Pattipati, "A bootstrapped PHMT with feature measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 5, pp. 2559–2571, 2017.

[37] X. Chen, Y. Li, J. Yu, and Y. Li, "Developing the fuzzy c-means clustering algorithm based on maximum entropy for multitarget tracking in a cluttered environment," *Journal of Applied Remote Sensing*, vol. 12, no. 1, article 016019, 2018.

[38] X. Zhan, X. Wang, and L. Li, "Online multi-object tracking via maximum entropy intuitionistic fuzzy data association," in *2018 14th IEEE International Conference on Signal Processing (ICSP)*, pp. 803–806, Beijing, China, 2018.

[39] M. Shozo and C. Chee, "Cross-entropy method for K-best dependent-target data association hypothesis selection," in *13th International Conference on Information Fusion*, Edinburgh, UK, 2010.

[40] X. Lian and A. Hamdulla, "A maximal fuzzy entropy based Gaussian clustering algorithm for tracking dim moving point targets in image sequences," in *2008 International Conference on Computer Science and Software Engineering*, pp. 54–57, Wuhan, China, 2008.

[41] X. Zhou, Y. Li, B. He, and T. Bai, "GM-PHD-based multitarget visual tracking using entropy distribution and game theory," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1064–1076, 2014.

[42] Y. Guo and J. Gong, "Group targets tracking using maximum entropy fuzzy based on fire-fly algorithm and particle filter," in *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*, pp. 937–942, Hefei, China, 2020.

[43] A. Saucan and P. Varshney, "Distributed cross-entropy δ-GLMB filter for multi-sensor multi-target tracking," in *2018 21st International Conference on Information Fusion (FUSION)*, pp. 1559–1566, Cambridge, UK, 2018.

[44] K. Becker, "A general approach to TMA observability from angle and frequency measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 1, pp. 487–494, 1996.

[45] C. Jauffret and D. Pillon, "Observability in passive target motion analysis," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1290–1300, 1996.

[46] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking . part I: dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[47] D. Crouse, M. Guerriero, and P. Willet, "A critical look at the PMHT," *J. Adv. Infor. Fusion*, vol. 4, pp. 93–116, 2009.

[48] W. Saidani, Y. Morsly, and M. S. Djouadi, "Sequential versus parallel architecture for multiple sensors multiple target tracking," in *IEEE Conference on Human System Interactions*, pp. 904–908, Krakow, Poland, 2008.

Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 2526916, 11 pages
https://doi.org/10.1155/2021/2526916

Research Article

Trajectory Optimization of Hypersonic Periodic Cruise Using an Improved PSO Algorithm

Hesong Li, Yi Wang, Shangcheng Xu, Yunfan Zhou, and Dan Su

College of Aerospace Science, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Yi Wang; 1262581502@qq.com

Received 4 July 2021; Revised 5 August 2021; Accepted 9 September 2021; Published 27 September 2021

Academic Editor: Mahmut Reyhanoglu

Periodic cruise has the potential to improve the fuel-saving efficiency of hypersonic cruise vehicles but is difficult to optimize. In this paper, hypersonic periodic cruise trajectory is analyzed theoretically and optimized by an improved Particle Swarm Optimization algorithm. Firstly, through theoretical analysis, it is determined that the optimal throttle curve can be parameterized as a switching function. Considering the optimization direction of algorithm, a new penalty function for the constraints of periodic cruise is proposed. Then, PSO algorithm is improved and applied in periodic cruise trajectory optimization. Numerical results demonstrate that optimized periodic cruise trajectory costs less fuel compared with steady-state cruise trajectory, and without computing gradient information, the proposed method is also robust. Finally, the fuel-saving mechanism of periodic cruise is explored by comparing with steady-state cruise, which reveals that periodic cruise trajectory has higher impulse and lift-drag ratio, but lower mechanical energy loss rate.

1. Introduction

Hypersonic vehicle has a series of advantages and is attractive to researchers all over the world [1–3]. In hypersonic flight, in order to satisfy multifarious constraints and reduce fuel consumption at the same time, trajectory design is necessary and significant [4, 5].

As far as hypersonic cruise vehicle is concerned, the whole trajectory of is generally divided into ascending stage, cruising stage, and gliding stage [6]. To carry out different missions in different stages, the flight modes of hypersonic vehicle are also quite different, which has been widely studied [7–11]. Generally speaking, the cruising stage accounts for a large proportion in the whole trajectory, and it decides the flight range to a great extent. To make the flight range longer, improving the fuel efficiency is an effective method and finding the cruise trajectory with higher fuel efficiency has been the focus of many researchers [12].

In cruising stage, there are two main cruise modes: steady-state cruise and periodic cruise. Steady-state cruise means that aircraft cruises at a constant altitude and velocity [13]. Thus, steady-state cruise trajectory has only two degree-of-freedoms (DOF) and is relatively easy to optimize [14]. However, it has been demonstrated that steady-state cruise trajectory is not the optimal in reducing fuel consumption [15]. Theoretically, the optimal trajectory is a curve with infinite DOF, and as a Two-Point Boundary Value Problem (TPBVP), it is quite difficult to be solved [16]. Therefore, to simplify the problem, periodic cruise was raised by dividing the whole cruise trajectory into a few phases. During periodic cruise, the curves of altitude and velocity approximate a form of periodic function. And trajectories in neighboring periods are nearly identical. At the end of a cycle, flight states are the same as the initial states [17].

In the analysis of hypersonic periodic cruise trajectory, angles of attack and throttle are control variables, which are also functions about time and need to be determined to minimize the objective, while these two variables are nonlinear and discontinuous sometimes, which increases the difficulty of obtaining the optimal solution [18]. From last century, a few theoretic analysis methods for trajectory design have been proposed [19, 20], but simplification and approximation methods were widely adopted in these analysis; otherwise, it was difficult to carry out because of the complexity of hypersonic flight. With the development of

computational science and intelligent algorithm, many optimization methods have been proposed and applied in the study of aerospace [21–23]. In recent years, evolution-based algorithms and pseudo-spectral method are widely applied in trajectory optimization [24–28], and it is demonstrated that optimization method is effective to solve trajectory optimization problems [29, 30].

Based on optimization methods, many researches about periodic cruise trajectory have been carried out. Kang et al. [31] studied the optimal periodic cruise trajectory by combining genetic algorithm and direct shooting method; then, a method of two-level optimization was developed to deal with the parameters of initial state in the outer loop and the control variables in the inner loop, respectively. Chen et al. [32] divided periodic cruise trajectory into a boost phase where engine was working and a coasting phase where engine was closed. Then, different constraints were proposed, and gradients of the cost were determined numerically. OTIS (Optimal Trajectories Implicit Simulation) was also applied in the problem. In [33], periodic cruise trajectory was divided into four parts, and different constraints were added separately in different parts. The trajectories in the four parts were optimized by GPOPS (Gauss Pseudospectral OPtimization Software) and fitted into a harmonic curve; then, the whole trajectory was obtained. Gao et al. [17] regarded the highest point of periodic cruise trajectory as the starting point and assumed that the curve of altitude was approximately a cosine curve which was regarded as a path constraint; then, the optimal control problem was solved by GPOPS. In [18], trajectories in accelerating phase and gliding phase were optimized separately; then, periodic cruise trajectory with less fuel consumed was obtained.

However, most of the optimization methods proposed for periodic cruise are based on gradient-based algorithm or pseudo-spectral method, which are sensitive to the initial guess value. If the initial guess is not suitable, the optimal solution may be missed. On the contrary, evolutionary optimization method does not need initial guess because the initial population is generated randomly, and it has more advantages in global search ability [34]. Furthermore, the derivative information is not required, which means that the difficulty to construct Jacobian and Hessian matrix can be avoided [35]. So far, there are few researches about the direct application of evolutionary method in the optimization of periodic cruise trajectory without combined with gradient-based algorithm or pseudo-spectral method. Therefore, this paper tries to directly apply evolutionary optimization method in periodic cruise trajectory design, which can optimize the trajectory robustly without initial guess or derivative information computed.

In this paper, firstly, theoretical analysis for periodic cruise is carried out, and it is determined that the optimal throttle curve is a switching function; then, the time when the engine switches on is regarded as an optimization variable. To deal with the constraint of periodic cruise and considering the optimization direction of algorithm, a new penalty function is proposed in the calculation of cost function. Then, to solve the problem effectively and efficiently, Particle Swarm Optimization (PSO) algorithm is improved in several aspects and applied in periodic cruise trajectory optimization. Finally, the difference of trajectory between steady-state cruise and periodic cruise is discussed, and the fuel-saving mechanism of periodic cruise is explored by contrast with steady-state cruise.

2. Models

The HL-20 aircraft model [36] is employed whose aerodynamic coefficients are parameterized in (1) [37]. α means the angle of attack, and M denotes Mach number. C_L and C_D denote the coefficient of lift and drag, respectively. The coefficient of drag at zero attack of angle, denoted by C_{D0}, is equal to 0.008 when $M > 10$.

$$\begin{cases} C_L(M,\alpha) = C_{L0}(M) + C_{L\alpha}(M)\alpha, \\ C_D(M,\alpha) = C_{D0}(M) + K(M){C_L}^2, \\ C_{L0}(M) = \dfrac{1}{20\pi}\arctan\left[10(M-1)\right] - 0.035, \\ C_{L\alpha}(M) = 0.057\exp\left(-0.654M\right) + 0.014, \\ K(M) = 1.85[1 - \exp\left(-0.2356M\right)]. \end{cases} \tag{1}$$

Thrust is calculated by thrust coefficient in (2) [37] and is proportional to throttle denoted by s in (3). q means the dynamic pressure, and S_e is the area of engine whose value is $9.02\,\mathrm{m}^2$.

$$C_{T\max} = \begin{cases} 0.4736M^{1.5} + 1.6947M^{-2} & (M < 4) \\ \dfrac{15(\alpha+5)^{0.25}}{M^{1.15}} \cdot \exp\left[-\dfrac{M^{0.08}}{200} \times \left(\alpha + 5 - \dfrac{35}{M^{0.6}}\right)^2\right] & (M \geq 4) \end{cases}, \tag{2}$$

$$T = sqC_{T\max}S_e. \tag{3}$$

Impulse is calculated by (4), and fuel consumed per second is calculated by (5), where h denotes flight altitude and g denotes the acceleration of gravity whose value is $9.8\,\mathrm{m/s}^2$.

$$I_{sp} = \begin{cases} 4500 - 10(h-20) & (M < 4) \\ -245M + 5480 - 10(h-20) & (M \geq 4) \end{cases}, \tag{4}$$

$$\frac{dm}{dt} = -\frac{T}{gI_{sp}}. \tag{5}$$

The 1976 U.S. Standard Atmosphere Model is used. An altitude factor is defined by (6) [17], where R_e means the radius of earth.

$$H = \frac{h}{1 + h/R_e}. \tag{6}$$

When flight altitude is in the range of 32 to 47 km, the atmospheric density, denoted by ρ, is calculated in (7),

where $\rho_0 = 1.225\,\text{kg/m}^3$.

$$W = 1 + \frac{H - 39.7499}{89.4107},$$
$$\rho = 3.2618 \times 10^{-3} \rho_0 W^{-13.2011}. \tag{7}$$

The earth is considered to be a homogeneous sphere. Sound velocity, which is denoted by a, is also regarded as a constant whose value is 340.294 m/s [37]. Then, the dynamic model is described in (8). The flight-path angle is denoted by γ. T, L, and D are thrust, lift, and drag, respectively; m means the mass of aircraft whose initial value is 89930 kg, and r denotes the flight range. Based on (8), the parameters of trajectory can be computed by Four Order Runge-Kutta method (RK-4), a numerical method for Ordinary Differential Equation (ODE).

$$\begin{cases} \dfrac{dh}{dt} = M \cdot a \cdot \sin\gamma, \\ \dfrac{dM}{dt} = \dfrac{T\cos\alpha - D - mg\sin\gamma}{m \cdot a}, \\ \dfrac{d\gamma}{dt} = \dfrac{T\sin\alpha + L}{mM \cdot a} + \cos\gamma\left(\dfrac{M \cdot a}{R_e + h} - \dfrac{g}{M \cdot a}\right), \\ \dfrac{dr}{dt} = M \cdot a \cdot \cos\gamma\left(\dfrac{R_e}{R_e + h}\right). \end{cases} \tag{8}$$

Trajectory optimization is a classic optimal control problem. h, M, and γ are regarded as state variables which is formulated as $\mathbf{x} = (h, M, \gamma)^T$. α and s are regarded as control variables which is formulated as $\mathbf{u} = (\alpha, s)^T$. The limits of α and s are showed in Table 1. The optimization of trajectory is to determine the change of α and s to minimize the objective.

3. Theoretical Analysis and Optimization Method

3.1. Theoretical Analysis for Periodic Cruise. In order to simplify the problem theoretically, the values of α and γ are regarded relatively small whose cosine value is set to 0 and sine value is set to 1. Then, considering the directions of altitude and range, the dynamic equations can be expressed in the two directions as shown in (9) [38].

$$\begin{cases} \dfrac{d^2r}{dt^2} = \dfrac{dv}{dt} = \dfrac{1}{m}(T - D), \\ \dfrac{d^2h}{dt^2} = \dfrac{1}{m}(L - mg). \end{cases} \tag{9}$$

Make the transformation as shown in (10).

$$x_1 = r, x_2 = h, x_3 = \frac{dr}{dt}, x_4 = \frac{dh}{dt}, x_5 = m. \tag{10}$$

The control variables, α and s, are denoted by u_1 and u_2; thus, $0 \le u_1 \le 15^\circ$, and $0 \le u_2 \le 1$. Then, L, D, and I_{sp} are all relative to x_2, x_3, x_4, and u_1, while T is relative to x_2, x_3, x_4, u_1, and proportional to u_2. So the fuel consumption per second can be expressed briefly as shown in (11), where k is a lumped parameter of those which are relative to x_2, x_3, x_4, and u_1.

Table 1: The limits of control variables.

Variable	Lower limit	Upper limit
α (°)	0	15
s	0	1

$$\frac{dm}{dt} = -\frac{T}{gI_{sp}} = -\frac{k_T(x_2, x_3, x_4, u_1) \cdot u_2}{gI_{sp}(x_2, x_3, x_4, u_1)} = -k(x_2, x_3, x_4, u_1) \cdot u_2. \tag{11}$$

Then, the dynamic equations in (9) can be described as:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}) = \begin{cases} \dot{x}_1 = x_3, \\ \dot{x}_2 = x_4, \\ \dot{x}_3 = \dfrac{1}{x_5}(k_T u_2 - D), \\ \dot{x}_4 = \dfrac{1}{x_5} L - g, \\ \dot{x}_5 = -k u_2. \end{cases} \tag{12}$$

The constraints are as shown in (13).

$$\begin{cases} x_2(t_0) = x_2(t_f), \\ x_3(t_0) = x_3(t_f), \\ x_4(t_0) = x_4(t_f). \end{cases} \tag{13}$$

The objective of optimal control is to minimize the fuel consumption averaged by range in a whole period. In order to construct the Hamilton function conveniently, it is regarded as a multiobjective optimization problem, whose objectives are maximum range and minimum fuel consumption. So the performance index is as given in (14), where J_1 reflects the range and J_2 reflects the fuel consumption. ω is a weight coefficient.

$$\begin{aligned} J = J_1 + \omega \cdot J_2 &= \int_{t_0}^{t_f} (-x_3 - \omega \cdot \dot{x}_5) dt \\ &= \int_{t_0}^{t_f} (-x_3 + \omega \cdot k u_2) dt. \end{aligned} \tag{14}$$

Therefore, the Hamilton function can be defined as (15), and λ is the Lagrange multipliers.

$$\begin{aligned} H &= L + \boldsymbol{\lambda}^T \cdot \mathbf{f} = -x_3 + \omega k u_2 + \lambda_1 x_3 + \lambda_2 x_4 \\ &\quad + \frac{\lambda_3}{x_5}(k_T u_2 - D) + \frac{\lambda_4}{x_5} L - \lambda_4 g - \lambda_5 k u_2 \\ &= (\lambda_1 - 1) x_3 + \lambda_2 x_4 + \frac{\lambda_3}{x_5}(k_T u_2 - D) \\ &\quad + \frac{\lambda_4}{x_5} L - \lambda_4 g + (\omega - \lambda_5) k u_2. \end{aligned} \tag{15}$$

And the costate function can be expanded as (16).

$$\dot{\boldsymbol{\lambda}} = \begin{bmatrix} 0 \\ -\frac{\lambda_3}{x_5}\frac{\partial k_T}{\partial x_2} u_2 + \frac{\lambda_3}{x_5}\frac{\partial D}{\partial x_2} - \frac{\lambda_4}{x_5}\frac{\partial L}{\partial x_2} + (\lambda_5 - \omega)\frac{\partial k}{\partial x_2} u_2 \\ 1 - \lambda_1 - \frac{\lambda_3}{x_5}\frac{\partial k_T}{\partial x_3} u_2 + \frac{\lambda_3}{x_5}\frac{\partial D}{\partial x_3} - \frac{\lambda_4}{x_5}\frac{\partial L}{\partial x_3} + (\lambda_5 - \omega)\frac{\partial k}{\partial x_3} u_2 \\ -\lambda_2 - \frac{\lambda_3}{x_5}\frac{\partial k_T}{\partial x_4} u_2 + \frac{\lambda_3}{x_5}\frac{\partial D}{\partial x_4} - \frac{\lambda_4}{x_5}\frac{\partial L}{\partial x_4} + (\lambda_5 - \omega)\frac{\partial k}{\partial x_4} u_2 \\ \frac{\lambda_3}{x_5^2}(k_T u_2 - D) + \frac{\lambda_4}{x_5^2} L \end{bmatrix}. \tag{16}$$

According to the optimality condition of Hamilton function:

$$H(\mathbf{x}^*(t), \boldsymbol{\lambda}(t), \mathbf{u}^*(t)) = \min_{\mathbf{u}(t) \in \mathbf{U}} H(\mathbf{x}^*(t), \boldsymbol{\lambda}(t), \mathbf{u}(t)), \tag{17}$$

due to

$$\frac{\partial H}{\partial u_2} = \frac{\lambda_3}{x_5} k_T + (\omega - \lambda_5) k, \tag{18}$$

let

$$\xi = \frac{\partial H}{\partial u_2} = \frac{\lambda_3}{x_5} k_T + (\omega - \lambda_5) k, \tag{19}$$

thus when $\xi > 0$, H is positively correlated with u_2. According to the minimum principle, to minimize the value of H, when $\xi > 0$, u_2 should be equal to its minimum value at 0; similarly, when $\xi < 0$, u_2 should be equal to its maximum value at 1. Thus, (20) is obtained.

$$u_2 = \begin{cases} 0, & \text{when } \xi \geq 0, \\ 1, & \text{when } \xi < 0. \end{cases} \tag{20}$$

Therefore, it can be concluded that the optimal throttle is either open or close totally, which is consistent with the results obtained by optimization method in [17, 19]. Then, the change of throttle can be parameterized by a switching function, whose coefficients are also regarded as optimization variables in this paper.

Based on the theoretical analysis, the throttle is parameterized by a switching function, and the time when engine starts and the duration when engine works are regarded as variables, while the change of angle of attack is not clear. Thus, the angle of attack needs to be parameterized as well. Then, the optimal control problem of periodic cruise trajectory design is transformed into a parameter optimization problem, and optimization algorithm can be employed to solve it. PSO algorithm is relatively easy to program, and it is reported well suited for finding the optimal solution for nonlinear system. In this paper, PSO algorithm is improved firstly and then applied to solve the parameter optimization problem.

3.2. Optimization Problem of Periodic Cruise. In a periodic cruise whose period is donated by t_c, the terminal constraints are illustrated in (21).

$$\begin{cases} h(0) = h(t_c), \\ v(0) = v(t_c), \\ \gamma(0) = \gamma(t_c). \end{cases} \tag{21}$$

The objective of optimization is to minimize the fuel consumption averaged by flight range in a cruise period, which is as given in (22).

$$J = \frac{\int_0^{t_c} | \dot{m} | \, dt}{\int_0^{t_c} \dot{r} dt} = \frac{m(0) - m(t_c)}{r_c}. \tag{22}$$

Therefore, the optimization problem of periodic cruise trajectory can be formulated as (23).

$$\begin{aligned} &\text{minimize } J = \frac{m(t_c) - m(0)}{r_c} \\ &\text{subject to} \begin{cases} \alpha(t) \in [5, 20] \\ s(t) \in [0, 1] \\ h(0) = h(t_c) \\ v(0) = v(t_c) \\ \gamma(0) = \gamma(t_c) \end{cases} . \end{aligned} \tag{23}$$

To parameterize the angle of attack, the Lagrange interpolation method is employed. The values of α when $t = 0$, $t = 1/3t_c$, and $t = 2/3t_c$ are selected as control points, and the value at the end of period is set equal to that in the initial time to ensure continuity in neighboring periods. Thus, there are four control points totally so that cubic Lagrange interpolation can be achieved to calculate the value of α at a certain moment. Based on the conclusion in last section, the throttle is open or closed totally; thus, the engine switch-on time t_b and working duration t_d are also regarded as optimization variables. Therefore, there are 5 variables in all, and the optimization objective is to minimize the fuel consumption averaged by range in a period.

Periodic cruise requires that the initial state $\mathbf{x}_0 = (h_0, M_0, \gamma_0)^T$ is the same as the final state $\mathbf{x}_f = (h_f, M_f, \gamma_f)^T$, which is a multiconstrained optimization

problem. To deal with the constraints of periodic cruise, generally the traditional strategy is to require $|h_0 - h_f| < h_\varepsilon$ and $|M_0 - M_f| < M_\varepsilon$, where h_ε and M_ε are acceptable small difference. However, when h_0 and M_0 are determined, if h_f and M_f are lower, it costs less fuel because less energy is needed to recover the altitude and Mach number back to the initial state. Thus, an optimization algorithm aiming at fuel-saving trends to decrease h_f and M_f, and the extreme case is $h_f = h_0 - h_\varepsilon$ and $M_f = M_0 - M_\varepsilon$ when $|h_0 - h_f| < h_\varepsilon$ or $|M_0 - M_f| < M_\varepsilon$ is required; that is, the altitude and velocity in final state are lower than those in initial state. It is also subjective to determine the value of h_ε and M_ε, which results in affecting the accuracy of fuel consumption computation. To avoid this, in this paper, $h_f \geq h_0$ and $M_f \geq M_0$ are required instead. Now that the optimization algorithm aiming at fuel-saving can naturally reduce h_f and M_f in the optimization process, if $h_f \geq h_0$ and $M_f \geq M_0$ are required, the extreme case is $h_f = h_0$ and $M_f = M_0$, which is exactly what is required for periodic cruise.

Therefore, there are three constraints that $h_f \geq h_0$, $M_f \geq M_0$, and $|\gamma_0 - \gamma_f| < \gamma_\varepsilon$, which are transformed into penalty functions and included in the computation of cost function. Thus, the cost function is as shown in (24)

$$\text{cost} = F_h + F_M + F_\gamma + \frac{m_0 - m_f}{r_c}. \tag{24}$$

F_h, F_M, and F_γ are penalty functions for flight altitude, Mach number, and flight-path angle, respectively, as given in (25), where λ_1, λ_2, and λ_3 are large positive numbers.

$$F_h = \begin{cases} \lambda_1 \cdot \dfrac{h_0 - h_f}{h_0}, & \text{if } h_0 > h_f \\ 0, & \text{else} \end{cases},$$

$$F_M = \begin{cases} \lambda_2 \cdot \dfrac{M_0 - M_f}{M_0}, & \text{if } M_0 > M_f \\ 0, & \text{else} \end{cases}, \tag{25}$$

$$F_\gamma = \begin{cases} \lambda_3 \cdot \dfrac{|\gamma_0 - \gamma_f|}{\gamma_\varepsilon}, & \text{if } |\gamma_0 - \gamma_f| > \gamma_\varepsilon \\ 0, & \text{else} \end{cases}.$$

Based on the description above, constraints are formulated in a new form, and cost function is established; thus, the periodic cruise problem is transformed into an optimization problem as follows:

$$\text{minimize cost} = F_h + F_M + F_\gamma + \frac{m_0 - m_f}{r_c}$$

$$\text{subjective to} \begin{cases} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} \alpha_{1,2,3} \\ t_b \\ t_d \end{bmatrix} \leq \begin{bmatrix} 15 \\ t_c \\ t_c - t_b \end{bmatrix}. \end{cases} \tag{26}$$

Then, the optimization can be carried out by optimization algorithm, which is illustrated next.

3.3. PSO Algorithm and Improvement. Due to there are 5 optimization variables in total, PSO algorithm, which is demonstrated suitable for optimization problem with high dimensions [39, 40] and easy to program, is employed to deal with the optimization problem. In order to improve the speed and effect of the algorithm, several improvements are implemented.

The core of PSO algorithm is to update the particle velocity, whose formula is shown as (27), where r_1 and r_2 are both random numbers between 0 and 1 and k is the number of iteration.

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot r_1 \cdot \left(p_i^k - x_i^k\right) + c_2 \cdot r_2 \cdot \left(p_g^k - x_i^k\right),$$
$$x_i^{k+1} = x_i^k + v_i^{k+1}. \tag{27}$$

Inertia weight, denoted by w, is used to adjust the global and local search ability of algorithm. A larger inertia weight could enhance the global search ability, while a smaller weight could enhance the local search ability [41]. Therefore, in this paper, w varies nonlinearly with k as shown in (28), where $k_{\max}$ is the maximum number of iteration and its value is 100. The values of $w_{\min}$ and $w_{\max}$ are 0.4 and 0.8, respectively [42].

$$w = w_{\min} + (w_{\max} - w_{\min})\sqrt{1 - \frac{k}{k_{\max}}}. \tag{28}$$

c_1 and c_2 are learning factors which reflect the ability of learning from individual and swarm, respectively [43]. In the early stage, a larger c_1 and a smaller c_2 could improve the global search performance. In the later stage, a smaller c_1 and a larger c_2 could make more particles close to the optimal position and is conducive to accelerate convergence [44]. Therefore, the learning factors are adjusted by linear strategy in (29), where c_{1f} and c_{10} equal to 0.8 and 1.5, while c_{2f} and c_{20} equal to 2.5 and 0.5.

$$\begin{cases} c_1 = \left(c_{1f} - c_{10}\right)\dfrac{k}{k_{\max}} + c_{10}, \\ c_2 = \left(c_{2f} - c_{20}\right)\dfrac{k}{k_{\max}} + c_{20}. \end{cases} \tag{29}$$

In order to reduce the possibility of falling into the local optimal solution, a hybrid strategy is also implemented. Based on the concept of genetic algorithm, there are a certain amount of particles selected to implement hybridization according to a determined probability. The position and velocity of the offspring particles can be obtained by random pairwise hybridization of the parent particles [45, 46]. Thus, the position and velocity of particles in the next iteration are obtained by (30), where r_3 is a random number between 0 and 1.

$$x_j^{k+1} = r_3 \cdot x_m^k + (1 - r_3) \cdot x_n^k,$$
$$v_j^{k+1} = \frac{v_m^k + v_n^k}{\left| v_m^k + v_n^k \right|} \cdot \left| v^k \right|. \tag{30}$$

In addition, the maximum velocity is dynamically limited: the range of updating velocity is allowed to be relatively larger at the initial stage of the algorithm to enhance swarm diversity and to be slightly smaller in the middle and later stages to prevent the updating velocity too large and away from the optimal position. Maximum velocity of k-th iteration is obtained by (31).

$$v_{\max}^k = \left[\left(1 - 0.9 \sin \left(\frac{\pi}{2} \frac{k}{k_{\max}} \right) \right) \right] v_{\max}^0. \tag{31}$$

Therefore, the process of the improved PSO algorithm is as illustrated in Figure 1, and the details are as follows:

Step 1. Generate the initial position and velocity of swarm randomly.

Step 2. Compute the cost value of each particle in the swarm.

Step 3. Select the particles to implement hybridization.

Step 4. Update the position and velocity of particle: the position and velocity of selected particles are updated by (30), while others are updated according to (27).

Step 5. Repeat Step 2 to Step 4 until the terminal condition is satisfied.

In this paper, the swarm size is 800, and the maximum iteration is 100. Then, based on the optimization problem and improved PSO algorithm, the optimization for periodic cruise trajectory can be carried out.

4. Results and Analysis

4.1. Optimization Results. To confirm the effect of the proposed method and explore the performance of periodic cruise, take the initial altitude of 45 km and the initial Mach number of 14 as an example, numerical simulation is carried out based on the improve PSO algorithm. The calculation is implemented based on Intel Core i5-11300H CPU whose main frequency is 3.10 GHz.

Set the initial altitude h_0 at 45 km initial Mach number M_0 at 14, the cruise period is roughly determined at 200 s according to [17, 19]. Figure 2 shows the change of cost function in the optimization process. With identical initiation and population size, the changes of cost function by classic PSO algorithm, genetic algorithm (GA) and differential evolution algorithm (DE) are also displayed to validate the effect of improvements. It can be seen that the classic PSO, GA, and DE all converge after approximately 80 iterations, while the improved PSO algorithm needs less than 40 iterations. And the classic PSO and GA as well as DE are more likely to be stuck in a local optimal solution.

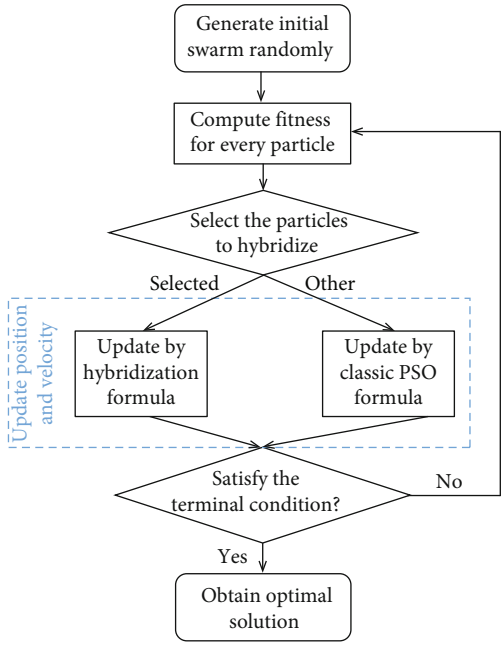


Figure 1: The progress of improved PSO algorithm.

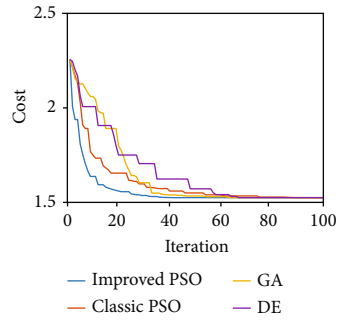


Figure 2: The value of cost function.

Figure 3 shows the optimized trajectory parameters of periodic cruise (PC) in red solid lines. In addition, parameters of steady-state cruise (SC), which is computed by the method in [6] at altitude of 45 km and Mach number of 14, are also displayed in dotted lines. It can be seen that periodic cruise is demonstrated to be realized because state variables return to its initial value at the end of period; thus, the new penalty function for constraints is validated as well. The fuel consumption averaged by range of periodic cruise is 1.5251 kg/km, while that value of steady-state cruise is 1.6855 kg/km; thus, periodic cruise is more fuel-efficient, and 9.51% fuel can be saved by periodic cruise. The proposed method is effective to optimize periodic cruise trajectory.

Compared with optimized results in [17, 18], fuel-saving performance of the proposed method in this paper is slightly higher. And the reason should be that there are constraints that the altitude curve is approaching a cosine function in [17], and engine starts at the lowest points in [18]. Without these additional constraints, the trajectory obtained by the proposed method can be more flexible; thus, the fuel-saving percentage is higher.

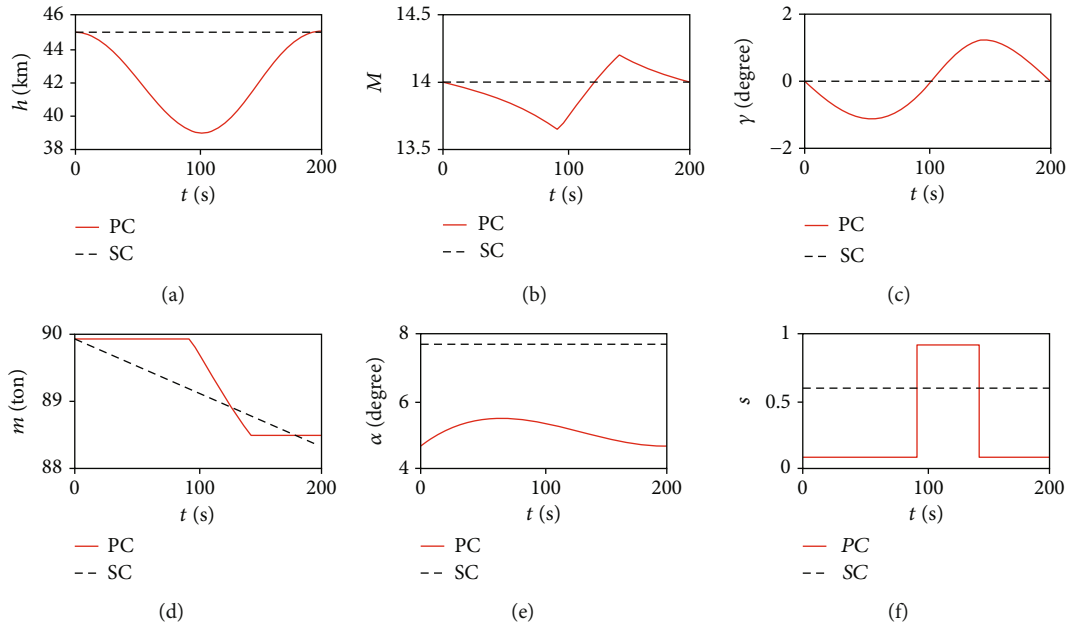


Figure 3: Flight parameters in cruise: (a) altitude (km); (b) Mach number; (c) fight-path angle (degree); (d) mass (ton); (e) angle of attack (degree); (f) throttle.

Table 2: Dispersions in the entry initial conditions.

State/parameter	Distribution	Mean	Standard deviation
h_0 (km)	Normal	45	0.45
M_0	Normal	14	0.14
γ_0 (degree)	Normal	0	0.1
m_0 (ton)	Normal	89.93	0.8993

To illustrate the stability of the optimization method, dispersion simulations are carried out, and 100 Monte-Carlo simulations are performed based on the case where $h_0 = 45$ km and $M_0 = 14$. The random initialization data used in the dispersion model are tabulated in Table 2.

The histories of 100 dispersed trajectories obtained using the proposed optimization method are plotted in Figure 4. Simulation results show that most of the cases can successfully converge to the optimal solution, and their fuel-saving percentages are all close to 9%. Thus, the proposed optimization method is not sensitive with a random initialization.

In order to further verify the robustness and feasibility of the proposed method, several cases whose initial altitudes and Mach numbers are around the optimal values of steady-state cruise are selected to be optimized. The optimal altitude of steady-state cruise is 42.6 km, and the optimal Mach number is 14.38. Then, 6 points, whose altitudes are 40 km and 45 km and Mach numbers are 14, 14.38, and 15, respectively, are selected as the initial points of periodic cruise. Figure 5 displays the trajectories of these cases, and their fuel-saving percentages are showed in Figure 6. It can be seen that periodic cruise is realized in all the 6 cases. When the initial altitude is relatively low, the altitude increases firstly and then decreases, and the altitude variation range increases with Mach number. And the reason should be that with a larger initial Mach number, a higher

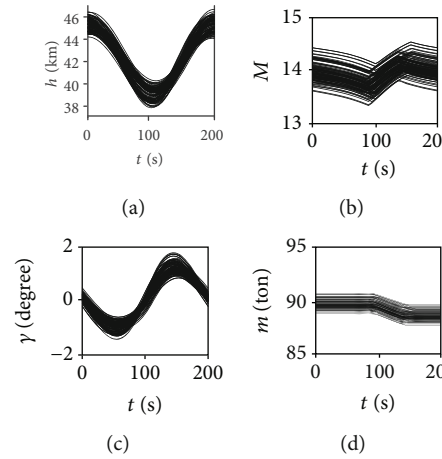


Figure 4: One hundred dispersed trajectories for the state variables: (a) altitude (km); (b) Mach number; (c) fight-path angle (degree); (d) mass (ton).

altitude can be achieved to reduce atmospheric density and then reduce drag. While when the initial altitude is relatively high, the altitude decreases firstly and then increases, and the altitude variation range decreases with Mach number. And the reason is that if initial Mach number is large, it is not appropriate to largely decrease altitude because drag will increase a lot. From Figure 6, it can be seen that the periodic cruise trajectories obtained in the all 6 cases consume less fuel than corresponding steady-state cruise trajectories. Therefore, the proposed method is effective to optimize periodic cruise trajectory robustly.

4.2. Analysis of Periodic Cruise Trajectory. In the case of $h_0 = 45$ km and $M_0 = 14$ above, a periodic cruise trajectory with fuel saving rate of nearly 10% compared to that of steady-

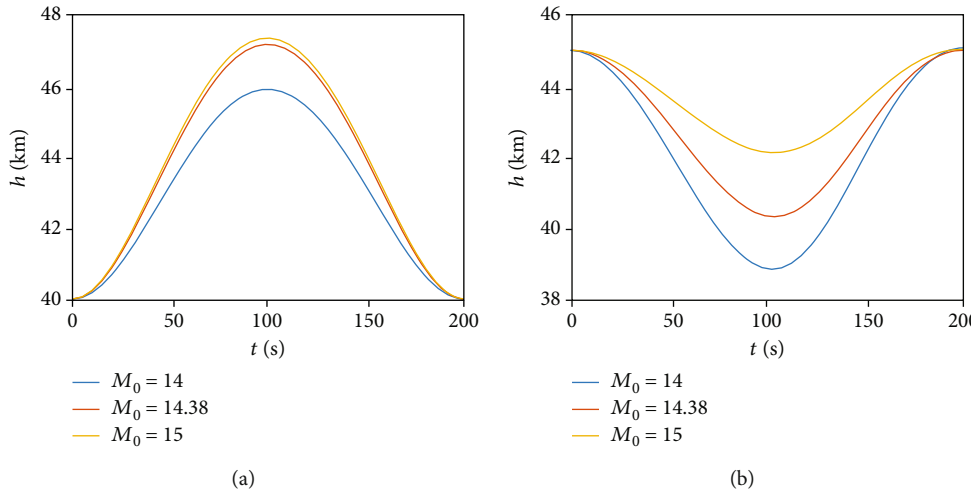


Figure 5: Trajectories at different initial Mach number: (a) $h_0 = 40$ km; (b) $h_0 = 45$ km.

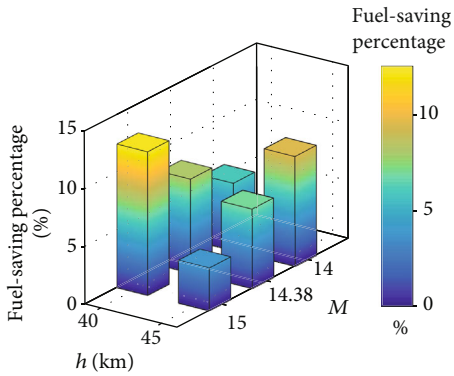


Figure 6: Distribution of fuel-saving percentage of periodic cruise at different initial points.

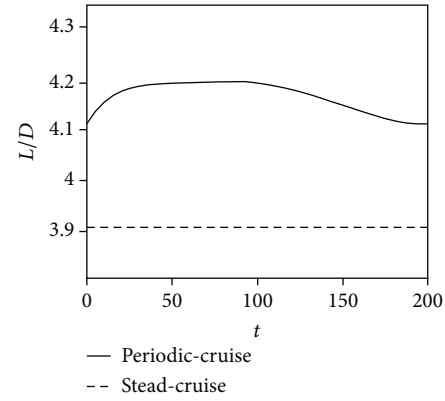


Figure 7: The change of lift-drag ratio.

state cruise is obtained. Based on this case, the fuel-saving mechanism of periodic cruise is analyzed in detail.

It can be seen from Figure 3 that the aircraft firstly decelerates and descends without power. When the flight altitude approaches the lowest point, the engine ignites and starts to accelerate. At this time, due to the low altitude and high atmospheric density, the engine can capture more airflow for combustion. Mach number is small at the same time. According to (4), the specific impulse is larger with lower altitude and smaller Mach number, which means a unit mass of fuel could generate larger thrust. Therefore, when engine is working, periodic cruise makes better use of impulse performance.

Figure 7 shows the curve of L/D in steady-state cruise and periodic cruise within a cycle. It can be found that L/D of periodic cruise is in the range of 4.1 to 4.2, which is greater than that of steady-state cruise at 3.9. Based on the Breguet Range Equation [47], large L/D is beneficial to enhance flight range. Thus, it reveals that periodic cruise makes better use of aerodynamic characteristics of aircraft to maintain high L/D, which is also a reasons for improving fuel efficiency during cruise.

In Figure 8, the curves of drag and thrust of steady-state cruise and periodic cruise are given, respectively. Thrust of steady-state cruise is not shown in the figure because the thrust and drag of steady-state cruise are roughly equal. Combined with the Figure 3, it can be seen that the drag of periodic cruise is the largest near the position where flight altitude and Mach number reach the lowest point. After the engine switches on, Mach number and altitude increase gradually, and drag decreases conversely. It is interesting that the drag of periodic cruise is not always lower than that of steady-state cruise. There is still a large span when the drag of periodic cruise is larger. However, from the perspective of energy conversion, the advantage of periodic cruise can be revealed. Since the initial and final states of a cycle are the same, if the mass change is ignored, the kinetic energy and gravitational potential energy keep unchanged in initial state and final state, which means that all the energy generated by fuel combustion is used to overcome drag and finally convert into heat energy. According to [48], the mechanical energy loss rate could be evaluated by (32), and its curve is displayed in Figure 9. The total mechanical energy loss of cruise could be obtained by integrating the curves. It is obvious that the total energy loss of periodic cruise is smaller than that of steady-state cruise, whose reason should be that from Figures 3 and 8, periodic cruise Mach number is lower than that of steady-state cruise in

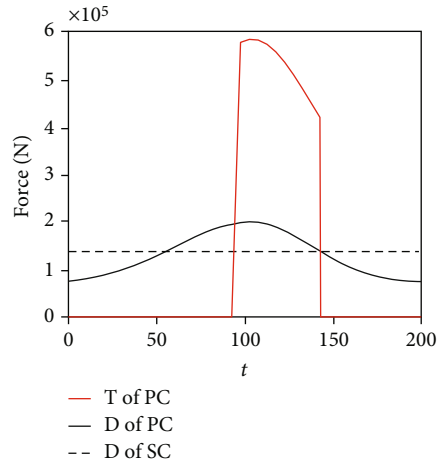


Figure 8: The change of drag and thrust.

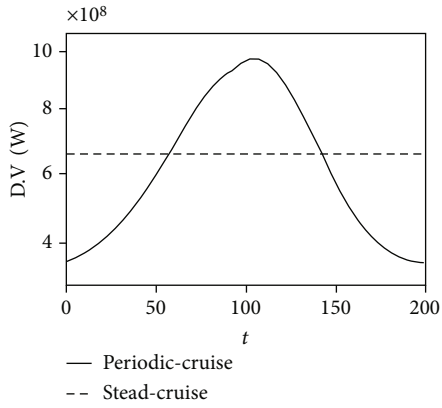


Figure 9: The change of mechanical energy loss rate.

the span when the drag of periodic cruise is larger than that of steady-state cruise, which plays a role in reducing the mechanical energy loss rate. Thus, periodic cruise needs less energy supplement, and it is more fuel-efficient.

$$\frac{dE_{\text{loss}}}{dt} = D \cdot V = D \cdot M \cdot a. \tag{32}$$

Therefore, the trajectory of periodic cruise utilizes the change of atmospheric density to reduce drag and enhance impulse and adopts suitable flight states to achieve higher lift-drag ratio and reduce the whole mechanical energy loss. Therefore, less fuel is consumed compared with that of steady-state cruise.

5. Conclusions

In this paper, based on the theoretical analysis, the design problem of periodic cruise trajectory is transformed into optimization problem by parameterizing the throttle and angle of attack. An improved PSO algorithm is applied to solve the optimization problem, and results with less fuel consumed are obtained. Finally, the fuel-saving mechanism of periodic cruise is analyzed. Conclusions can be drawn as follows:

(1) The change of throttle is a switching function; thus, the throttle is open totally when engine works in periodic cruise

(2) The improved PSO algorithm can robustly optimize trajectory of periodic cruise with less fuel consumed, and the new form of constraints is effective for periodic cruise

(3) Periodic cruise trajectory can enhance the impulse and L/D and reduce the loss of mechanical energy by matching velocity and drag properly

Data Availability

The image data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors want to express their thanks for the support from Tao Tang. This research was funded by the Postgraduate Scientific Research Innovation Project of Hunan Province (Grant number: CX20200084).

References

[1] Z. Chen and Y. Zhao, "Aerothermoelastic analysis of a hypersonic vehicle based on thermal modal reconstruction," *International Journal of Aerospace Engineering*, vol. 2019, Article ID 8384639, 13 pages, 2019.

[2] R. Chai, A. Tsourdos, A. Savvaris, Y. Xia, and S. Chai, "Trajectory planning for hypersonic reentry vehicle satisfying deterministic and probabilistic constraints," *Acta Astronautica*, vol. 177, pp. 30–38, 2020.

[3] Y. Fan, P. Yan, F. Wang, and H. Xu, "Discrete sliding mode control for hypersonic cruise missile," *International Journal of Aerospace Engineering*, vol. 2016, Article ID 2402794, 9 pages, 2016.

[4] G. N. Kumar, M. Ikram, and A. K. Sarkar, "Hypersonic flight vehicle trajectory optimization using pattern search algorithm," *Optimization and Engineering*, vol. 19, no. 1, pp. 125–161, 2018.

[5] L. Ma, Y. Tian, S. Yang, C. Xu, and A. Hao, "A scheme of sustainable trajectory optimization for aircraft cruise based on comprehensive social benefit," *Discrete Dynamics in Nature and Society*, vol. 2021, no. 2, Article ID 7629203, 15 pages, 2021.

[6] H. Li, Y. Wang, Y. Zhou, S. Xu, K. An, and X. Fan, "Optimization and analysis for hypersonic steady-state cruise trajectory," in *Proceedings of the 33rd Chinese Control and Decision Conference*, pp. 7555–7560, Kunming, 2021.

[7] X. Liu, Y. Wang, L. Liu, and J. Huang, "Optimization on steady-state cruise for a hypersonic vehicle," in *Proceedings*

of the 33rd Chinese Control Conference, pp. 8935–8940, Nanjing, 2014.

[8] P. Pei, S. Fan, W. Wang, and D. Lin, "Online reentry trajectory optimization using modified sequential convex programming for hypersonic vehicle," *IEEE Access*, vol. 9, pp. 23511–23525, 2021.

[9] H. Zhou, X. Wang, and N. Cui, "Glide trajectory optimization for hypersonic vehicles via dynamic pressure control," *Acta Astronautica*, vol. 164, pp. 376–386, 2019.

[10] Z. Huang, Y. Zhang, and Y. Liu, "Research on state estimation of hypersonic glide vehicle," *Conference series*, vol. 1060, no. 1, article 012088, 2018.

[11] Z. Wang, "Optimal trajectories and normal load analysis of hypersonic glide vehicles via convex optimization," *Aerospace Science & Technology*, vol. 87, pp. 357–368, 2019.

[12] P. F. Gath and A. J. Calise, "Optimization of launch vehicle ascent trajectories with path constraints and coast arcs," *Journal of Guidance Control and Dynamics*, vol. 24, no. 2, pp. 296–304, 2001.

[13] D. E. Koelle, "On the optimum cruise speed of a hypersonic aircraft," *IEEE Aerospace and Electronic Systems Magazine*, vol. 4, no. 5, pp. 13–16, 1989.

[14] K. Wang, B. Zhang, and Y. Hou, "Multiobjective optimization of steady-state cruise trajectory for a hypersonic vehicle," in *Proceedings of 2017 3rd IEEE international conference on control science and systems engineering (ICCSSE)*, pp. 130–135, Beijing, 2017.

[15] J. R. Irons, R. L. Schultz, and N. R. Zagalsky, "Energy state approximation and minimum-fuel fixed-range trajectories," *Journal of Aircraft*, vol. 8, pp. 488–490, 1971.

[16] R. H. Chen, W. R. Williamson, J. L. Speyer, H. Youssef, and R. Chowdhry, "Optimization and implementation of periodic cruise for a hypersonic vehicle," *Journal of Guidance Control & Dynamics*, vol. 29, no. 5, pp. 1032–1040, 2006.

[17] H. Gao, Z. Chen, M. Sun, Z. Wang, and Z. Chen, "General periodic cruise guidance optimization for hypersonic vehicles," *Applied Sciences*, vol. 10, no. 8, p. 2898, 2020.

[18] H. Li, Y. Wang, Y. Zhou, S. Xu, and D. Su, "A two-level optimization method for hypersonic periodic cruise trajectory," *International Journal of Aerospace Engineering*, vol. 2021, Article ID 9975007, 11 pages, 2021.

[19] C. Chuang and H. Morimoto, "Optimal periodic cruise for a hypersonic vehicle with constraints," in *Proceedings of Guidance, Navigation, and Control Conference*, San Diego, CA, 1996.

[20] P. Menon, G. Sweriduk, and A. Bowers, "A study of near-optimal endurance-maximizing periodic cruise trajectories," in *Proceedings of AIAA guidance, navigation, and control conference and exhibit*, p. 6153, San Francisco, 2005.

[21] D. Spiller, L. Ansalone, and F. Curti, "Particle swarm optimization for time-optimal spacecraft reorientation with keep-out cones," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 2, pp. 312–325, 2016.

[22] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "A review of optimization techniques in spacecraft flight trajectory design," *Progress in Aerospace Sciences*, vol. 109, no. 100543, pp. 1–15, 2019.

[23] H. Zhang, H. Wang, L. Na, Y. Yue, and Y. Liu, "Time-optimal memetic whale optimization algorithm for hypersonic vehicle reentry trajectory optimization with no-fly zones," *Neural Computing and Applications*, vol. 32, no. 2, pp. 2735–2749, 2020.

[24] G. Huang, Y. Lu, and Y. Nan, "A survey of numerical algorithms for trajectory optimization of flight vehicles," *Science China-Technological Sciences*, vol. 55, no. 9, pp. 2538–2560, 2012.

[25] K. S. Sana and W. Hu, "Hypersonic reentry trajectory planning by using hybrid fractional-order particle swarm optimization and gravitational search algorithm," *Chinese Journal of Aeronautics*, vol. 34, no. 1, pp. 50–67, 2020.

[26] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Trajectory optimization of space maneuver vehicle using a hybrid optimal control solver," *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 467–480, 2017.

[27] 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*, vol. 15, no. 7, pp. 3899–3909, 2019.

[28] 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*, vol. 67, no. 8, pp. 6904–6915, 2020.

[29] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance Control and Dynamics*, vol. 21, pp. 193–207, 1998.

[30] W. Wang, Z. Hou, S. Shan, and L. Chen, "Optimal periodic control of hypersonic cruise vehicle: trajectory features," *IEEE Access*, vol. 7, pp. 3406–3421, 2019.

[31] B. N. Kang, D. B. Spencer, S. Tang, and D. Jordan, "Optimal periodic cruise trajectories via a two-level optimization method," *Journal of Spacecraft and Rocket*, vol. 47, no. 4, pp. 597–613, 2010.

[32] R. H. Chen, W. R. Williamson, J. L. Speyer, H. Youssef, and R. Chowdhry, "Optimization and implementation of periodic cruise for a hypersonic vehicle," *Journal of Guidance, Control and Dynamics*, vol. 29, no. 5, pp. 1032–1040, 2006.

[33] H. Gao, M. Sun, and Z. Chen, "Hypersonic periodic cruise trajectory optimization based on flexible use of pseudo-spectral method," in *Proceedings of 2019 Chinese Intelligent Systems Conference*, Haikou, China, 2020.

[34] Y. Gong, J. Li, Y. Zhou et al., "Genetic learning particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2277–2290, 2016.

[35] R. Chai, A. Tsourdos, A. Savvaris, Y. Xia, and S. Chai, "Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1630–1643, 2020.

[36] G. M. Ware and C. I. Cruz, "Aerodynamic characteristics of the HL-20," *Journal of Spacecraft and Rockets*, vol. 30, no. 5, pp. 529–536, 1993.

[37] C. H. Chuang and H. Morimoto, "Periodic optimal cruise for a hypersonic vehicle with constraints," *Journal of Spacecraft and Rockets*, vol. 34, no. 2, pp. 165–171, 1997.

[38] R. Li and Y. Shi, "The fuel optimal control problem of a hypersonic aircraft with periodic cruising mode," *Mathematical and Computer Modelling*, vol. 55, no. 11-12, pp. 2141–2150, 2012.

[39] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 303–308, Indianapolis, IN, 1997.

[40] P. Mauro and A. C. Bruce, "Particle swarm optimization applied to space trajectories," *Journal of Guidance Control and Dynamics*, vol. 33, no. 5, pp. 1429–1441, 2010.

[41] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.

[42] C. Jiang, S. Zhao, and S. Shen, "Particle swarm optimization algorithm with sinusoidal changing inertia weight," *Computer Engineering and Applications*, vol. 48, no. 8, pp. 40–42, 2012.

[43] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[44] G. Ma, R. Li, and L. Liu, "Particle swarm optimization algorithm of learning factors and time factors adjusting to weights," *Application Research of Computers*, vol. 31, no. 11, pp. 3291–3294, 2014.

[45] P. Kaur and P. Kaur, "A hybrid PSO-GA approach to solve vehicle routing problem," *International Journal of Engineering Development and Research*, vol. 3, no. 3, pp. 1–5, 2015.

[46] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp. 292–305, 2016.

[47] L. Bréguet, "Calcul du poids de combustible consummé par un avionen vol ascendant," *Comptes Rendus del'Academie des Sciences*, vol. 177, pp. 870–872, 1923.

[48] Z. Wang, X. Cheng, and H. Li, "Hypersonic skipping trajectory planning for high L/D gliding vehicles," in *21st AIAA International Space Planes and Hypersonics Technologies Conference*, Beijing, China, 2017.

Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 2734230, 13 pages
https://doi.org/10.1155/2021/2734230

Research Article

Fuel-Optimal Ascent Trajectory Problem for Launch Vehicle via Theory of Functional Connections

Shiyao Li[1], **Yushen Yan**[1], **Kun Zhang**[2] and **Xinguo Li**[1,3]

[1]*School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China*
[2]*Xi'an Modern Control Technology Research Institute, Xi'an 710065, China*
[3]*National Key Laboratory of Aerospace Flight Dynamics, Northwestern Polytechnical University, Xi'an 710072, China*

Correspondence should be addressed to Shiyao Li; lishiyao@mail.nwpu.edu.cn

Received 3 June 2021; Revised 16 August 2021; Accepted 25 August 2021; Published 13 September 2021

Academic Editor: Erdal Kayacan

In this study, we develop a method based on the Theory of Functional Connections (TFC) to solve the fuel-optimal problem in the ascending phase of the launch vehicle. The problem is first transformed into a nonlinear two-point boundary value problem (TPBVP) using the indirect method. Then, using the function interpolation technique called the TFC, the problem's constraints are analytically embedded into a functional, and the TPBVP is transformed into an unconstrained optimization problem that includes orthogonal polynomials with unknown coefficients. This process effectively reduces the search space of the solution because the original constrained problem transformed into an unconstrained problem, and thus, the unknown coefficients of the unconstrained expression can be solved using simple numerical methods. Finally, the proposed algorithm is validated by comparing to a general nonlinear optimal control software GPOPS-II and the traditional indirect numerical method. The results demonstrated that the proposed algorithm is robust to poor initial values, and solutions can be solved in less than 300 ms within the MATLAB implementation. Consequently, the proposed method has the potential to generate optimal trajectories on-board in real time.

1. Introduction

With the recent development in space exploration, launch vehicles are very important as they are the only means for humans to explore space from the earth. In general, a launch vehicle mission has been planned over a long period, and the trajectory was designed in advance, and it cannot be updated during flight, which means it is not robust or flexible. The fast launch and trajectory reconstruction are the main research of the guidance system, and both need rapid trajectory planning technology. Rapid trajectory planning can shorten the launch mission cycle and quickly update the trajectory in case of thrust failure during the flight of the vehicle to ensure the success of the mission.

The primary aim of the trajectory planning algorithm is to solve the optimal control problem that is generally based on nonlinear dynamics, which achieves specific performance indicators under the constraints of state and control variables. The solution of such problems is mainly achieved using the indirect method [1–3] and the direct method [4–6]. The direct method transforms the optimal control problem of continuous space into a nonlinear programming problem and uses a numerical method to directly optimize the performance index [7–9]. Although the direct method, represented by a sequential quadratic programming algorithm with the pseudospectral discrete, has advanced a lot over a period of time, it still has considerable issues between the on-board application. The algorithm for the general nonlinear programming, for example, the famous sequential quadratic programming algorithm, had low algorithm efficiency and low sensitivity to the initial value and was unable to guarantee convergence in the past, but now, these issues have easily resolved by using a convex optimization method. Ralph Rockafellar, a renowned mathematician, pointed out that the key determinant of the performance of a numerical optimization algorithm is neither the linearity nor nonlinearity of the problem, but the convexity or nonconvexity of the problem [10]. In 2007, JPL proposed lossless convexity

technology for the dynamic descent guidance of the Mars lander [11]. After that, a systematic summary of the research and development of lossless convexity technology is presented in [12]. The applicability of the lossless convexity method is extended to general linear systems with multiple state constraints using the concepts of control theory. Unfortunately, only a few nonconvex constraints can be used for lossless convexification, and there is no analytical convexification method for the system dynamic constraints in the trajectory optimization problem of aircraft. A method based on the Newton–Kantorovich/pseudospectral and sequence convexification is used for the ascending phase of the launch vehicle [13]. However, the sequential convex optimization algorithm is a convexification method based on linearization, which increases the dependence on the reference trajectory. This not only offers higher requirements for the reference trajectory but also annihilates the advantage of the convex optimization algorithm that does not rely on the good initial value. Nevertheless, considering the rapidity of the convex optimization algorithm in solving convex problems, in recent years, trajectory planning based on the convex optimization algorithm, such as planetary landing [14–16], rocket ascent guidance [17], and entry guidance [18], has been widely studied.

The indirect method solves the optimal control problem by using the classical variational method and the Pontryagin Minimum Principle, derives the first-order necessary conditions of the optimal control, and transforms the optimal control problem into a two-point boundary value problem (TPBVP) [19] that is comprised of initial conditions, Hamiltonian differential equations, optimal conditions, and terminal boundary conditions (including terminal transversal conditions and terminal constraints). However, since the convergence radius of the indirect method is small, and the convergence of the numerical iteration is extremely sensitive to the initial value estimation, which requires a higher accuracy of the initial value estimation, determination of the initial value is highly difficult. To overcome this problem, the deep learning algorithm is used to obtain a higher accuracy of initial value estimation and a better target shooting success rate [20–22]. In general, the higher sensitivity of TPBVP to the initial value makes the problem difficult to solve. Therefore, although the indirect method yields a more accurate solution, it is rarely used in practice.

Recently, a mathematical framework called the Theory of Functional Connections (TFC) has been proposed in [23] to derive the expressions with embedded constraints. The expressions, called constrained expressions, are composed of functionals and functions of functions. The constrained expression is written as

$$y(t) = g(t) + \sum_{n}^{k=1} \Phi_k(t) p_k(t), \tag{1}$$

where $g(t)$ is a free function and $\Phi_k(t)$ are the switch functions composed of the support function $s_k(t)$ with unknown coefficients α_k. The support function is a set of linearly independent functions. If one of the switch functions is equal to 1, the constraint it is referencing is evaluated; otherwise, that is, if it is equal to 0, all other constraints are evaluated. The switch functions can be expressed as $\Phi_k(t) = s_k(t)\alpha_k$. The projection functionals $p_k(t)$ are derived by constraint functions. The constrained optimization problem is transformed into an unconstrained one using the constrained expressions, which reduces the search space of the solution to the admissible solutions that satisfy all constraints. Finally, using common basis functions such as Chebyshev polynomials or Legendre polynomials to express the free function and then using the least-square method to find its unknown coefficient, the solution of the problem can be found. In [24, 25], the TFC algorithm quickly solves the nonlinear differential equations and obtains high-precision solutions. In [26, 27], it is applied to fixed-time asteroid landing and optimal energy landing. The results show that the solution time is basically less than 100 ms, which proves that the algorithm has real-time application potential.

This paper is organized as follows. Section 2 gives a brief description of the TFC mathematical framework to solving TPBVP. In Section 3, the fuel-optimal ascent trajectory problem is described in detail, and the necessary conditions are derived. In Section 4, the fuel-optimal problem in the ascending phase of the launch vehicle is formulated using the TFC framework. Finally, the result and discussion are provided in Section 5.

2. Theory of Functional Connections

In this section, we present an outline of the TFC mathematical framework and a method for solving second-order TPBVP with TFC.

2.1. TFC for TPBVP. In general, trajectory optimization problems are second-order TPBVP [28], which is expressed as

$$F(t, y(t), \dot{y}(t), \ddot{y}(t)) = 0 \quad \text{subject to :} \begin{cases} y(t_0) = y_0, \\ y(t_f) = y_f, \\ \dot{y}(t_0) = \dot{y}_0, \\ \dot{y}(t_f) = \dot{y}_f, \end{cases} \tag{2}$$

where t_0 and t_f represent the initial time and the terminal time, respectively, and $y_0, y_f, \dot{y}_0, \dot{y}_f$ are the initial and terminal constraints, respectively. As mentioned earlier, the constraints are expressed by (1), and then, (2) is simply rewritten as

$$y(t) = g(t) + \Phi_1(t)p_1 + \Phi_2(t)p_2 + \Phi_3(t)p_3 + \Phi_4(t)p_4, \tag{3}$$

where $p_k(t)$ is expressed as

$$\begin{aligned} p_1(t, g_i(t)) &= y_{0_i} - g_i(t_0), \\ p_2(t, g_i(t)) &= y_{f_i} - g_i(t_f), \\ p_3(t, g_i(t)) &= \dot{y}_{0_i} - \dot{g}_i(t_0), \\ p_4(t, g_i(t)) &= \dot{y}_{f_i} - \dot{g}_i(t_f). \end{aligned} \tag{4}$$

The algorithm to derive the term of the constrained expression is given as follows:

(1) Choose $s_k(t)$, which are k linearly independent support functions

(2) Calculate switching functions $\Phi_k(t)$ as a linear combination of the support functions with k unknown coefficients

(3) Formulate a system of equations to solve for the unknown coefficients based on $\Phi_k(t)$

The support functions are defined as $s_k(k) = t^{k-1}$, according to [28]. The switching function is then obtained by solving equations. When the first switch function is activated, the equations are

$$\begin{cases} \Phi_1(t_0) = 1, \\ \Phi_1(t_f) = 0, \\ \Phi_1'(t_0) = 0, \\ \Phi_1'(t_f) = 0. \end{cases} \tag{5}$$

The equations of the first switch functions are combined into the matrix form as

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \alpha_{31} \\ \alpha_{41} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6}$$

Similarly, application of this method to the other three switch functions yields the following matrix:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} \alpha_{11} & \alpha_{21} & \alpha_{31} & \alpha_{41} \\ \alpha_{21} & \alpha_{22} & \alpha_{32} & \alpha_{42} \\ \alpha_{31} & \alpha_{23} & \alpha_{33} & \alpha_{43} \\ \alpha_{41} & \alpha_{24} & \alpha_{34} & \alpha_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

The coefficient α_k is obtained by matrix inversion, and the expression of the switch functions is obtained as

$$\begin{aligned} \Phi_1(t) &= \frac{1}{(t_f - t_0)^3}\left(-t_f^2(3t_0 - t_f) + 6t_0t_ft - 3(t_0 + t_f)t^2 + 2t^3\right), \\ \Phi_2(t) &= \frac{1}{(t_f - t_0)^3}\left(-t_0^2(t_0 - 3t_f) - 6t_0t_ft + 3(t_0 + t_f)t^2 - 2t^3\right), \\ \Phi_3(t) &= \frac{1}{(t_f - t_0)^2}\left(-t_0t_f^2 + t_f(2t_0 + t_f)t - (t_0 + 2t_f)t^2 + t^3\right), \\ \Phi_4(t) &= \frac{1}{(t_f - t_0)^2}\left(-t_0^2t_f + t_0(t_0 + 2t_f)t - (2t_0 + t_f)t^2 + t^3\right). \end{aligned} \tag{8}$$

The boundary conditions of (2) are effectively embedded within the constrained expression by substituting (4) and (8) into (3). Then, by substituting (3) into the vector differential equation $F(t, y(t), \dot{y}(t), \ddot{y}(t))$, the constrained TPBVPs are transformed into an unconstrained problem. According to (3), $y(t)$ is replaced by $g(t)$; thus, the original vector differential equation $F(t, y(t), \dot{y}(t), \ddot{y}(t))$ is transformed into $\widehat{F}(t, g(t), \dot{g}(t), \ddot{g}(t))$, which is only a function of t, the free functions $g(t)$, and their derivatives

$$\widehat{F}(t, g(t), \dot{g}(t), \ddot{g}(t)) = 0. \tag{9}$$

As mentioned earlier, $\widehat{F}(t, g(t), \dot{g}(t), \ddot{g}(t))$ is unconstrained because the boundary is represented by the switch function $\Phi_k(t)$ and the projection function $p_k(t)$.

After determining the switch function and the projection function, we next discuss the free function $g(t)$.

2.2. Definition of the Free Function. In selecting a free function, we are essentially looking for the best function approximator. A natural choice for the free function is a linear combination of basis functions, as this is capable of spanning the entire function space that the basis spans, as the number of basis functions approaches infinity. The free function is expressed as

$$g(t) = \xi^T h, \tag{10}$$

where ξ are $m \times 1$ unknown coefficients and h are m basis functions.

Next, the problem domain t is mapped to the domain of the basis functions z, and Chebyshev and Legendre polynomials are commonly used, the domains of which are defined in $[-1, 1]$. To implement the basis functions, a map between t and z is defined as

$$z = z_0 + \frac{z_f - z_0}{t_f - t_0}(t - t_0) \leftrightarrow t = t_0 + \frac{t_f - t_0}{z_f - z_0}(z - z_0). \tag{11}$$

By using (11), the derivatives of $g(t)$ are computed as

$$\frac{\mathrm{d}^k g}{\mathrm{d}t^k} = c^k \xi^T \frac{\mathrm{d}^k h(z)}{\mathrm{d}z^k}, \quad \text{where } k \in [0, n], \tag{12}$$

where $c \coloneqq \mathrm{d}z/\mathrm{d}t = (z_f - z_0)/(t_f - t_0)$.

2.3. Domain Discretization. For solving TPBVPs numerically, the domain $t \in [t_0, t_f]$ must be discretized by $N + 1$ points. The common method is uniform distribution, but the advantage of Chebyshev–Gauss–Lobatto collocation points is that when the number of basis functions increases, the condition number should also increase slowly, which is useful for improving computational efficiency. The Chebyshev–Gauss–Lobatto collocation points are defined as

$$z_k = -\cos\left(\frac{k\pi}{N}\right) \quad \text{for } k = 1, 2, \cdots, N. \tag{13}$$

Thus, the new vector differential equation $\widehat{F}(t, g(t), \dot{g}(t), \ddot{g}(t))$ becomes $\tilde{F}(z, \xi)$, where the unknown coefficient ξ is the variable that needs to be solved. $\tilde{F}(z, \xi)$ is expressed in the form of loss functions at each discrete point

$$\mathscr{L}_i(\xi) = \begin{Bmatrix} \tilde{F}_i(z_0, \xi) \\ \vdots \\ \tilde{F}_i(z_d, \xi) \\ \vdots \\ \tilde{F}_i(z_N, \xi) \end{Bmatrix}. \tag{14}$$

By setting $\mathscr{L}_i = 0$, the unknown coefficient ξ is solved using optimization schemes such as iterative least-squares.

To solve the nonlinear least-square problem, we need the Jacobian matrix of the loss function, which is written as

$$J(\xi) = \begin{bmatrix} \dfrac{\partial \tilde{F}_i(z_0, \xi)}{\partial \xi_1} & \dfrac{\partial \tilde{F}_i(z_0, \xi)}{\partial \xi_2} & \cdots \dfrac{\partial \tilde{F}_i(z_0, \xi)}{\partial \xi_j} \\ \vdots & \vdots & \vdots \\ \dfrac{\partial \tilde{F}_i(z_d, \xi)}{\partial \xi_1} & \dfrac{\partial \tilde{F}_i(z_d, \xi)}{\partial \xi_2} & \cdots \dfrac{\partial \tilde{F}_i(z_d, \xi)}{\partial \xi_j} \\ \vdots & \vdots & \vdots \\ \dfrac{\partial \tilde{F}_i(z_N, \xi)}{\partial \xi_1} & \dfrac{\partial \tilde{F}_i(z_N, \xi)}{\partial \xi_2} & \cdots \dfrac{\partial \tilde{F}_i(z_N, \xi)}{\partial \xi_j} \end{bmatrix}. \tag{15}$$

The estimation is updated by

$$\xi^{k+1} = \xi^k - \Delta\xi, \tag{16}$$

where $\Delta\xi = (J(\xi)^T J(\xi))^{-1} J(\xi)^T \mathscr{L}(\xi)$.

The iterative process stops when the convergence tolerance is met:

$$L_2[\mathscr{L}(\xi)] < \varepsilon, \tag{17}$$

where ε is the stopping criterion that is defined by the user.

Figure 1 shows the outline of the TFC framework.

Recently, the position of numerical calculation in the current guidance and control field is emphasized in [29], which, based on numerical calculation, is called as the Computational Guidance and Control (CG&C). The CG&C replaces offline planning and closed-loop guidance with on-board computing, which is more robust, more accurate, and more flexible and can adapt to more complex environments and missions, but offers high requirements for computational efficiency. As mentioned earlier, the algorithms used in the CG&C are basically divided into two: direct and indirect. In this study, we used the indirect method, because the optimal control problem is transformed into TPBVP, and then, the TFC method is used to transform and solve the problem.

We found that the TFC method is quite similar to the collocation method in which all three methods using orthogonal polynomials over the global domain, according to the different method. In the collocation method, the states and costates are expanded by using orthogonal polynomials, and the boundary conditions are considered as part of the optimization scheme [30]. It is similar that the pseudospectral method used orthogonal polynomials like Chebyshev or Legendre polynomials to approximate the states and costates, and the boundary conditions are also considered as part of the optimization scheme. The TFC method may use a similar operation, but the fundamental difference between the TFC method and the other two methods lies in handling the constraints of the problem. The TFC method uses orthogonal polynomials to expand the free function $g(t)$ in a constrained expression and then expresses the problem constraints analytically step by step as mentioned above, which can reduce the search space of the solution and thus improve the computational efficiency. In fact, the advantages of the TFC method are presented in [27]; the results in [27] show that the TFC method is two orders of magnitude faster than the pseudospectral method in a fixed time optimal control problem and one order of magnitude faster than the pseudospectral method in a free time optimal control problem.

3. Fuel-Optimal Problem in the Ascending Phase of the Launch Vehicle

In this section, the problem is transformed into a TPBVP, and the first-order necessary conditions and transversal conditions of the problem are derived using the Pontryagin Minimum Principle.

3.1. Dynamical Model. In this section, the last stage of the launching vehicle is studied. The dimensionless equations of motion of a three-dimensional (3-D) launch vehicle can be expressed in the Earth Center Inertial Coordinate System as follows:

$$\begin{aligned} \dot{r} &= v, \\ \dot{v} &= \frac{T}{m} + a_g, \\ \dot{m} &= \frac{T}{I_{sp} g_0}, \end{aligned} \tag{18}$$

where r is the inertial position, which is normalized by the radius of the Earth $R_0 = 6378145$ m. v is the velocity, which is normalized by $\sqrt{R_0 g_0}$, in which $g_0 = 9.81$ m/s^2 represents the gravitational acceleration magnitude on the surface of the Earth. The mass of the launch vehicle is denoted by m. The thrust is denoted by $T = TI_b$, where I_b is the unit vector of the body axis satisfying

$$\|I_b\| \equiv 1. \tag{19}$$

For most launch vehicles, the mass flow is uncontrollable; thus, the thrust magnitude $T = \|T\|$ is constant and uncontrollable during the same flight phase. The gravity acceleration

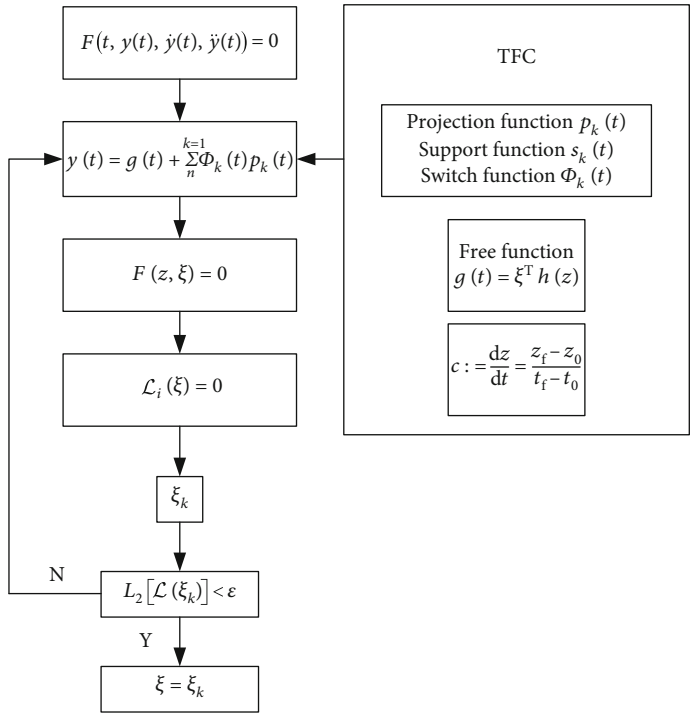


Figure 1: TFC framework.

$a_g = -r/r^3$, where $r = \|r\|$. The specific impulse of the engine is denoted by I_{sp}. The differentiation of the equations in (18) is with respect to dimensionless time normalized by $\sqrt{R_0/g_0}$.

As the mass flow is constant, the optimal ascent problem is treated as a minimum-time problem:

$$\min\ J = t_f, \tag{20}$$

subject to

$$\begin{gathered} \dot{r} = v, \quad \dot{v} = \frac{T}{m} + a_g, \quad \dot{m} = \frac{T}{I_{sp} g_0}, \\ r(0) = r_0, \quad v(0) = v_0, \quad m(0) = m_0, \\ r(t_f) = r_f, \quad v(t_f) = v_f. \end{gathered} \tag{21}$$

3.2. First-Order Necessary Conditions. On the basis of the Pontryagin Minimum Principle, the mass of the vehicle is treated as a prescribed function of time instead of a state variable. The Hamiltonian function is written as

$$H = 1 + \lambda_r^T v + \lambda_v^T \left(\frac{T}{m} I_b - \frac{r}{r^3} \right), \tag{22}$$

where $I_b = -\lambda_v/\lambda_v$ is called the primer's vector, according to Lawden's theory [31]. Thus, (22) is rewritten as

$$H = 1 + \lambda_r^T v - \frac{T}{m} \|\lambda_v\| - \frac{\lambda_v^T r}{r^3}. \tag{23}$$

The first-order necessary conditions for optimality then give the differential equations of the costate variables:

$$\begin{aligned} \dot{\lambda}_r &= -\frac{\partial H}{\partial r} = \frac{\lambda_v^T r}{r^5}, \\ \dot{\lambda}_v &= -\frac{\partial H}{\partial v} = -\lambda_r. \end{aligned} \tag{24}$$

The transversality condition is expressed as

$$H(t_f) = 0. \tag{25}$$

4. Solution via TFC

4.1. TPBVP in TFC Framework. As mentioned in the previous section, to find the optimal state, the following nonlinear TPBVPs must be solved:

$$\dot{r} = v, \tag{26}$$

$$\dot{v} = -\frac{T}{m} \frac{\lambda_v}{\lambda_v} - \frac{r}{r^3}, \tag{27}$$

$$\dot{m} = \frac{T}{I_{sp} g_0}, \tag{28}$$

$$\dot{\lambda}_r = -\frac{\partial H}{\partial r} = \frac{\lambda_v^T r}{r^5}, \tag{29}$$

$$\dot{\lambda}_v = -\frac{\partial H}{\partial v} = -\lambda_r, \tag{30}$$

$$H(t_f) = 1 + \lambda_r^T(t_f)v(t_f) - \frac{T}{m}\|\lambda_v(t_f)\| - \frac{\lambda_v^T(t_f)r(t_f)}{r^3(t_f)} = 0, \tag{31}$$

where (26) and (27) are subject to

$$\begin{aligned} r(0) &= r_0, \quad v(0) = v_0, \quad m(0) = m_0, \\ r(t_f) &= r_f, \quad v(t_f) = v_f. \end{aligned} \tag{32}$$

Additionally, there is a redundant equation and can be removed by the TFC constraints. The derivative of $r(t)$ is exactly the function $v(t)$ because TFC constraints are analytical expressions; thus, (26) can be disregarded. The problem is now reduced. The new equations are expressed as

$$\begin{aligned} \mathscr{L}_a(t) &= a - \dot{\lambda}_r - \frac{\lambda_v^T r}{r^5}, \\ \mathscr{L}_r(t) &= \dot{\lambda}_r - \frac{\lambda_v^T r}{r^5}, \\ \mathscr{L}_v(t) &= \dot{\lambda}_v + \lambda_r, \\ \mathscr{L}_H &= 1 + \lambda_r^T v - \frac{T}{m}\|\lambda_v\| - \frac{\lambda_v^T r}{r^3}. \end{aligned} \tag{33}$$

To solve the above equations, the TFC constraints with $r(t), \lambda_r(t), \lambda_v(t)$ need to be constructed by the TFC method. The unknown coefficients in the TFC constraint expressions are expressed as $\xi_a, \xi_{\lambda_r}, \xi_{\lambda_v}$, and $v(t), a(t), \dot{\lambda}_r(t), \dot{\lambda}_v(t)$ can be obtained by taking the derivative of the TFC constraint expression $r(t), \lambda_r(t), \lambda_v(t)$, respectively.

The initial and terminal constraints of the problem discussed in this paper are position and velocity constraints, respectively. The TFC constraint expressions of $r(t), v(t), a(t)$ are written as

$$\begin{aligned} r(t) &= g(t) + \Phi_1(t)p_1 + \Phi_2(t)p_2 + \Phi_3(t)p_3 + \Phi_4(t)p_4, \\ v(t) &= g(t) + \dot{\Phi}_1(t)p_1 + \dot{\Phi}_2(t)p_2 + \dot{\Phi}_3(t)p_3 + \dot{\Phi}_4(t)p_4, \\ a(t) &= g(t) + \ddot{\Phi}_1(t)p_1 + \ddot{\Phi}_2(t)p_2 + \ddot{\Phi}_3(t)p_3 + \ddot{\Phi}_4(t)p_4, \end{aligned} \tag{34}$$

where the projection function is written as

$$\begin{aligned} p_1(t) &= r_0 - g(t_0), \\ p_2(t) &= r_f - g(t_f), \\ p_3(t) &= v_0 - \dot{g}(t_0), \\ p_4(t) &= v_f - \dot{g}(t_f). \end{aligned} \tag{35}$$

Next, consider constructing a free function. According to (10)–(12), the time domain is mapped to the Chebyshev domain. However, it should be noted that in the time-free TPBVP, the parameter c is a function of t_f. Combined with the TFC method, the new unknown variable ξ_t is used to represent the parameter c, and the optimal time is obtained by solving ξ_t. In addition, to ensure the solved final time is positive, b is used instead of b, where $b^2 = c$. Then, $r(t)$ is rewritten as

$$\begin{aligned} r(t) = r(z) &= \Phi_1(z)r_0 + \Phi_2(z)r_f + \Phi_3(z)\frac{v_0}{b^2} + \Phi_4(z)\frac{v_f}{b^2} \\ &\quad + \left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T \xi_a, \\ v(t) = b^2 v(z) &= b^2\left[\Phi_1'(z)r_0 + \Phi_2'(z)r_f + \Phi_3'(z)\frac{v_0}{b^2} + \Phi_4'(z)\frac{v_f}{b^2}\right. \\ &\quad \left. + \left(h'(z) - \Phi_1'(z)h_0 - \Phi_2'(z)h_f - \Phi_3'(z)h_0' - \Phi_4'(z)h_f'\right)^T \xi_a\right], \\ a(t) = b^4 r(z) &= b^4\left[\Phi_1''(z)r_0 + \Phi_2''(z)r_f + \Phi_3''(z)\frac{v_0}{b^2} + \Phi_4''(z)\frac{v_f}{b^2}\right. \\ &\quad \left. + \left(h''(z) - \Phi_1''(z)_0 - \Phi_2''(z)h_f - \Phi_3''(z)h_0' - \Phi_4''(z)h_f'\right)^T \xi_a\right], \end{aligned} \tag{36}$$

where $h_0 = h(z_0)$ and $h_f = h(z_f)$.

The expression of the switch function is similar to (8):

$$\begin{aligned} \Phi_1(z) &= \frac{1}{(z_f - z_0)^3}\left(-z_f^2(3z_0 - z_f) + 6z_0 z_f z - 3(z_0 + z_f)z^2 + 2z^3\right), \\ \Phi_2(z) &= \frac{1}{(z_f - z_0)^3}\left(-z_0^2(z_0 - 3z_f) - 6z_0 z_f z + 3(z_0 + z_f)z^2 - 2z^3\right), \\ \Phi_3(z) &= \frac{1}{(z_f - z_0)^2}\left(-z_0 z_f^2 + z_f(2z_0 + z_f)z - (z_0 + 2z_f)z^2 + z^3\right), \\ \Phi_4(z) &= \frac{1}{(z_f - z_0)^2}\left(-z_0^2 z_f + z_0(z_0 + 2z_f)z - (2z_0 + z_f)z^2 + z^3\right). \end{aligned} \tag{37}$$

Next, the TFC constraint expressions of the costates are constructed by the same steps as above:

$$\begin{aligned} \lambda_r(t) &= \lambda_r(z) = h(z)^T \xi_r, \\ \dot{\lambda}_r(t) &= b^2 \lambda_r'(z) = b^2 h'(z)^T \xi_r, \\ \lambda_v(t) &= \lambda_v(z) = h(z)^T \xi_v, \\ \dot{\lambda}_v(t) &= b^2 \lambda_v'(z) = b^2 h'(z)^T \xi_v. \end{aligned} \tag{38}$$

Substitution of the above TFC constraint expression into the loss function yields the loss function with respect to unknown coefficient ξ. Then, the solution of the problem is obtained using the nonlinear least-square method. The unknown coefficient ξ is expressed as

$$\xi = \left\{\xi_{a_1}^T \xi_{a_2}^T \xi_{a_3}^T \xi_{\lambda_{r1}}^T \xi_{\lambda_{r2}}^T \xi_{\lambda_{r3}}^T \xi_{\lambda_{v1}}^T \xi_{\lambda_{v2}}^T \xi_{\lambda_{v3}}^T \xi_t\right\}^T \in \mathbb{R}^{9m+1}. \tag{39}$$

The loss function can be expressed as

$$\mathbb{L} = \left\{\mathscr{L}_{a_1}^T \mathscr{L}_{a_2}^T \mathscr{L}_{a_3}^T \mathscr{L}_{\lambda_{r1}}^T \mathscr{L}_{\lambda_{r2}}^T \mathscr{L}_{\lambda_{r3}}^T \mathscr{L}_{\lambda_{v1}}^T \mathscr{L}_{\lambda_{v2}}^T \mathscr{L}_{\lambda_{v3}}^T \mathscr{L}_H\right\}^T. \tag{40}$$

To use the nonlinear least-square method, the partial derivative of the loss function needs to be calculated as

$$a(t) = b^4 r(t) = b^4 \left[\Phi_1'(z) r_0 + \Phi_2'(z) r_f + \Phi_3'(z) \frac{v_0}{b^2} + \Phi_4'(z) \frac{v_f}{b^2} + \left(h''(z) - \Phi_1'(z) h_0 - \Phi_2'(z) h_f - \Phi_3'(z) h_0' - \Phi_4'(z) h_f' \right)^T \xi_a \right]. \tag{41}$$

The terms of (41) are defined by

$$J_{a,\xi_a} = \begin{bmatrix} \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{a_3}} \\ \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{a_3}} \\ \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{a_3}} \end{bmatrix}, \quad J_{a,\xi_v} = \begin{bmatrix} \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{v_1}} & \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{v_2}} & \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{v_3}} \\ \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{v_1}} & \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{v_2}} & \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{v_3}} \\ \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{v_1}} & \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{v_2}} & \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{v_3}} \end{bmatrix},$$

$$J_{a,\xi_t} = \begin{bmatrix} \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_t} & \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_t} & \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_t} \end{bmatrix}^T,$$

$$J_{\lambda_r,\xi_a} = \begin{bmatrix} \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_{a_3}} \\ \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_{a_3}} \\ \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_{a_3}} \end{bmatrix}, \quad J_{\lambda_r,\xi_r} = \begin{bmatrix} \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_{r_1}} & 0 & 0 \\ 0 & \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_{r_2}} & 0 \\ 0 & 0 & \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_{r_3}} \end{bmatrix},$$

$$J_{\lambda_r,\xi_v} = \begin{bmatrix} \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_{v_1}} & 0 & 0 \\ 0 & \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_{v_2}} & 0 \\ 0 & 0 & \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_{v_3}} \end{bmatrix}, \quad J_{\lambda_r,\xi_b} = \begin{bmatrix} \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_t} \\ \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_t} \\ \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_t} \end{bmatrix},$$

$$J_{\lambda_v,\xi_r} = \begin{bmatrix} \frac{\partial \mathscr{L}_{v_1}}{\partial \xi_{r_1}} & 0 & 0 \\ 0 & \frac{\partial \mathscr{L}_{v_2}}{\partial \xi_{r_2}} & 0 \\ 0 & 0 & \frac{\partial \mathscr{L}_{v_3}}{\partial \xi_{r_3}} \end{bmatrix}, \quad J_{\lambda_v,\xi_v} = \begin{bmatrix} \frac{\partial \mathscr{L}_{v_1}}{\partial \xi_{v_1}} & 0 & 0 \\ 0 & \frac{\partial \mathscr{L}_{v_2}}{\partial \xi_{v_2}} & 0 \\ 0 & 0 & \frac{\partial \mathscr{L}_{v_3}}{\partial \xi_{v_3}} \end{bmatrix},$$

$$J_{\lambda_v,\xi_t} = \begin{bmatrix} \frac{\partial \mathscr{L}_{v_1}}{\partial \xi_t} & \frac{\partial \mathscr{L}_{v_2}}{\partial \xi_t} & \frac{\partial \mathscr{L}_{v_3}}{\partial \xi_t} \end{bmatrix}^T,$$

$$J_{H,\xi_a} = \begin{bmatrix} \frac{\partial \mathscr{L}_H}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{a_3}} \end{bmatrix},$$

$$J_{H,\xi_r} = \begin{bmatrix} \frac{\partial \mathscr{L}_H}{\partial \xi_{r_1}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{r_2}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{r_3}} \end{bmatrix},$$

$$J_{H,\xi_v} = \begin{bmatrix} \frac{\partial \mathscr{L}_H}{\partial \xi_{v_1}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{v_2}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{v_3}} \end{bmatrix},$$

$$J_{H,\xi_t} = \frac{\partial \mathscr{L}_H}{\partial \xi_t}, \tag{42}$$

where all partial derivatives are provided in the Appendix.

4.2. Initialization. When using the iterative least-square method to calculate, some parameters need to be estimated reasonably, so that the iterative process of the algorithm can go on without violating the basic mathematical principles. The simplest initialization is to set them equal to zero. This is equivalent to connecting the boundary value problem with the simplest interpolating expression. However, the λ_v is related to the thrust direction; initialization of $\lambda_v = 0$ will cause issues in the TFC method because $I_b = -\lambda_v/\lambda_v$ and $\| I_b \| \equiv 1$. Thus, the coefficient ξ_v is initialized using

$$\xi_{v_0} = \frac{v_0}{\|v_0\|}, \quad \xi_{v_f} = -\frac{r_0}{\|r_0\|}. \tag{43}$$

In addition, the first guess of ξ_r, ξ_a is set equal to zero, and for setting the first guess of ξ_t, an estimate of the final time t_f is needed; in this paper, the initial t_f is set to be 300, and the initial ξ_t can be expressed as

$$\xi_t = \sqrt{\frac{2}{t_f - t_0}}. \tag{44}$$

These are uniformly discrete according to the number of polynomials m.

5. Simulations

In this section, we apply the proposed algorithm to the ascent problem of the launch vehicle to verify the feasibility of the algorithm, and the results are compared with those of the GPOPS-II and classical indirect method solutions of the other two algorithms. All numerical results are obtained on a desktop with Intel Xeon E3-1230 3.4 GHz. Table 1 lists the parameters of the launch vehicle and mission in the numerical simulations.

Table 2 shows the initial and terminal parameters of the experiment, where the orbital elements corresponding to the terminal position and velocity are also given because the launch vehicle generally uses the orbital elements for the target.

To prove the validity and effectiveness of the algorithm proposed in this study, the results of the algorithm proposed in this paper are compared with those obtained by the traditional indirect method and GPOPS-II. The final time calculated by GPOPS-II is 300.97 s, that obtained by the single shooting method is 301.01 s, and the final time obtained by TFC is 301.25 s. The locations of the vehicle and the velocity vector are provided in Figures 2 and 3, respectively. Figure 4 shows the thrust vector of the launch vehicle. In the bottom part of Figure 4, the purple line representing the sum of squares of thrust directions is equal to 1, which also indicates the validity of the TFC solution, and the other three lines are the vector of the body axis. Considering that the pitch angle and yaw angle are generally used as the guidance command of the launch

Table 1: Parameters for launch vehicle.

Parameter	Value
Longitude of launch point (°)	110.95
Latitude of launching point (°)	19.61
Launch azimuth (°)	90
The specific impulse, I_{sp} (m/s)	3365
Thrust, T (N)	2843599.98
Mass rate, $\dot{m}$ (kg/s)	845.052

Table 2: Parameters of the boundary conditions.

Parameter	Value
Initial position, r_0 (m)	[371973.739, 6493779.849, -13899.978]
Initial velocity, v_0 (m/s)	[3652.033, 556.843, -2.666]
Initial mass, m_0 (kg)	350306
Semimajor (m)	6595487
Eccentricity	0.0053
Inclination (°)	20.009
Ascending node (°)	17.250
Argument of perigee (°)	32.390
True anomaly (°)	77.969
Terminal position, r_f (m)	[1912866.558, 6304148.648, 2551.256]
Terminal velocity, v_f (m/s)	[7457.930, -2220.619, 178.661]

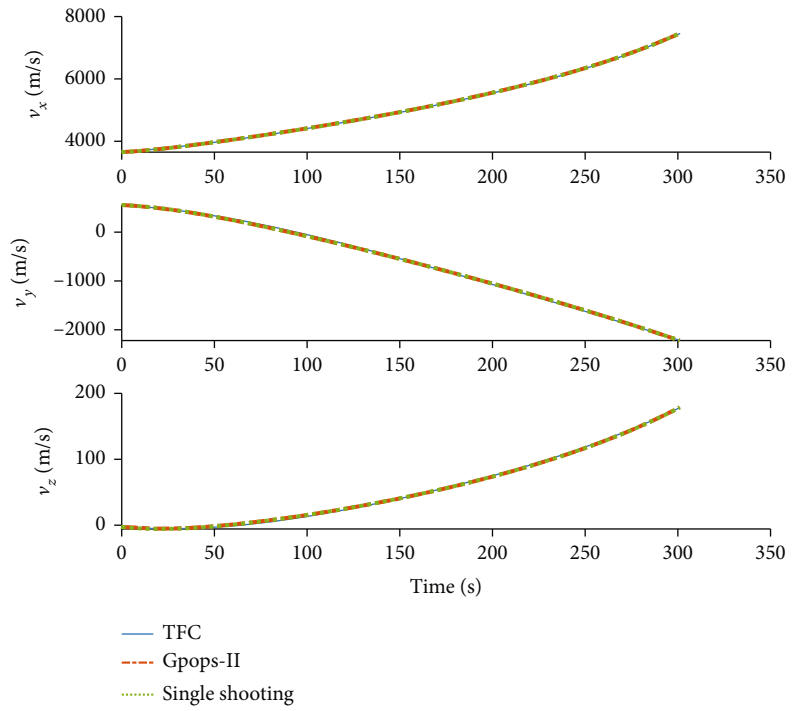


Figure 2: Vector of velocity solution by TFC, GPOPS-II, and single shooting.

vehicle, the results of pitch angle and yaw angle are also given here. Figures 5 and 6 show the results of the height and the velocity, respectively. It can be seen that the results solved by the three methods are basically the same, which again shows the validity of the TFC algorithm proposed in this study. To quantify the accuracy of the TFC method,

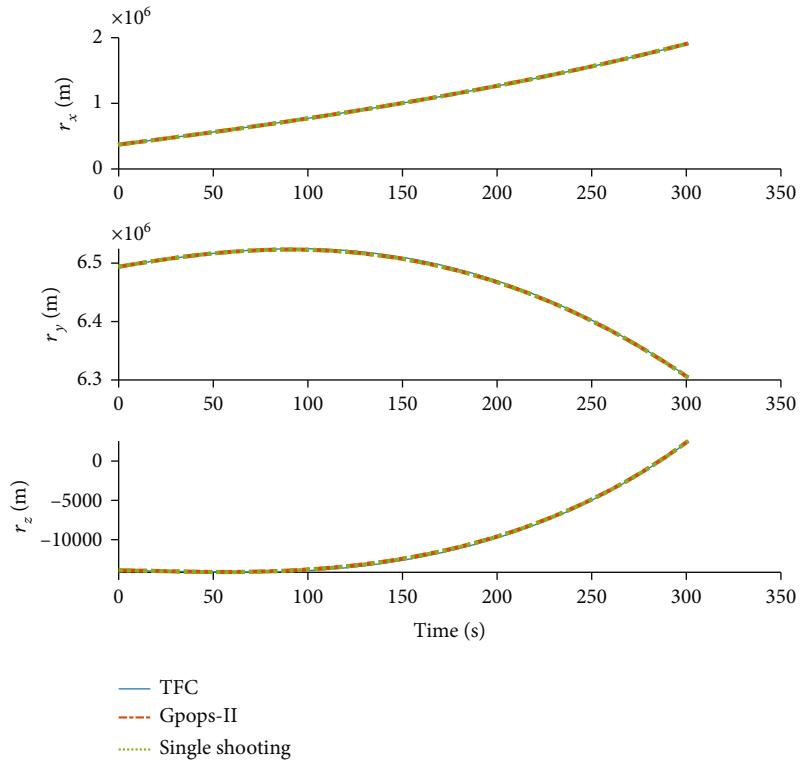


Figure 3: Vector of position solution by TFC, GPOPS-II, and single shooting.

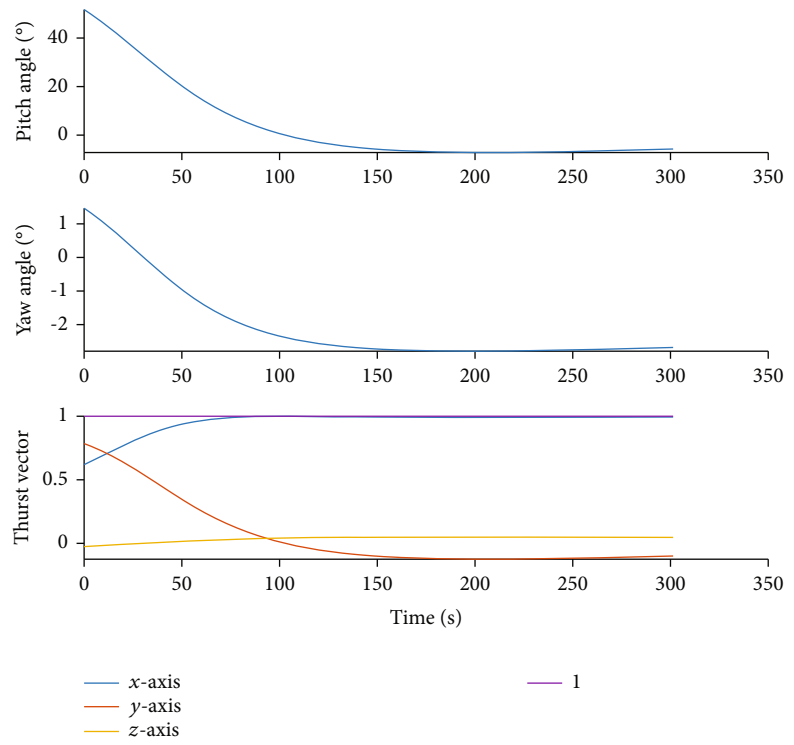


Figure 4: Vector of thrust solution by TFC.

Figure 7 shows the residual of acceleration, and it can be seen that the residual of TFC is about 10^{-14} or less for the whole solution domain.

In this simulation, the cost time of the proposed method is 0.23 s and those of GPOPS-II and the single shooting method are 3.88 and 0.34 s, respectively. It is also known that

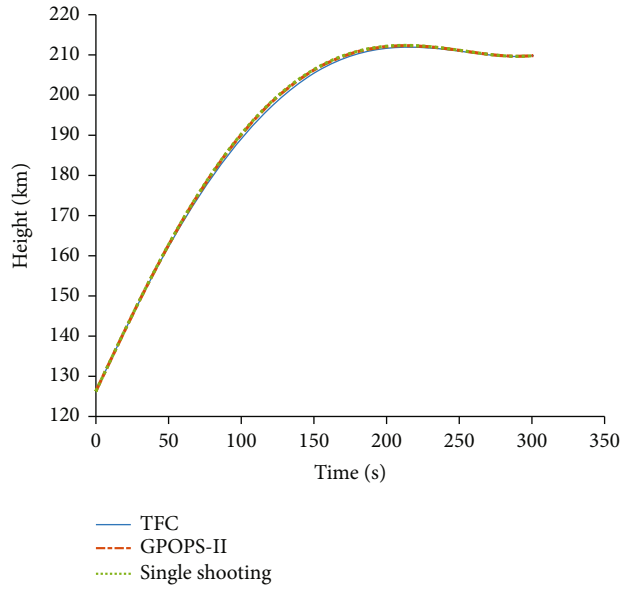


Figure 5: Height solution by TFC, GPOPS-II, and single shooting.

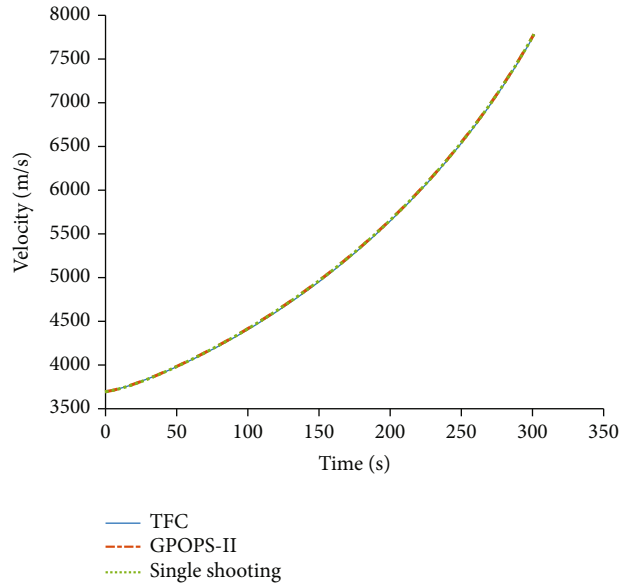


Figure 6: Velocity solution by TFC, GPOPS-II, and single shooting.

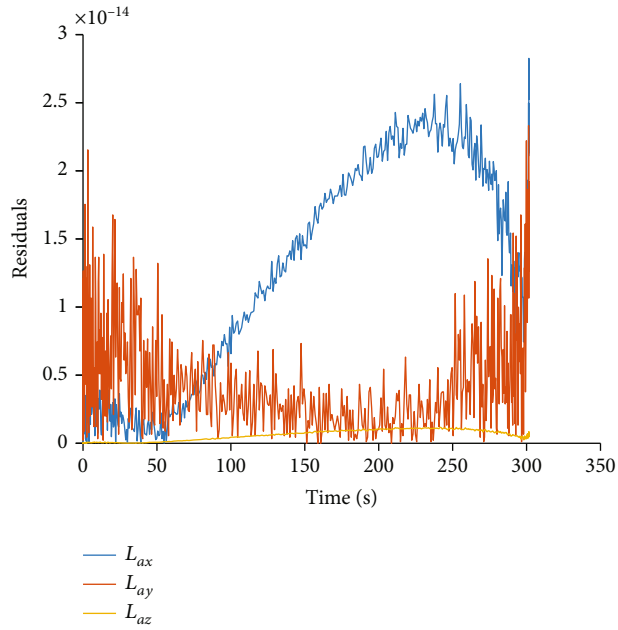


Figure 7: Residual by TFC.

Table 3: Solutions of different discrete points and polynomial numbers.

Discrete points	State and costate polynomial numbers	First-order optimality	Cost time (s)	Iteration number
100	10, 4	0.414	0.15	61
100	30, 4	0.00138	0.16	14
100	60, 4	2.09*e*-06	0.23	15
100	80, 4	0.00526	0.29	14
100	80, 10	2.13*e*-05	0.42	29
100	60, 60	50.3	—	Max
50	10, 4	0.349	0.16	54
50	30, 4	5.94*e*-05	0.23	45
50	50, 4	0.000555	0.25	26

the MATLAB programming language is 10 times slower than C++; thus, the algorithm proposed in this study has online application potential. In terms of solution accuracy, the TFC method is not dominant among the three methods, but combined with the analysis of calculation efficiency, it shows that the proposed method is an effective method.

The above results show the comparison between the results of the proposed algorithm, GPOPS-II, and single shooting method, which verifies the validity of the proposed algorithm. Next, the effect of the number of discrete points and polynomials on the algorithm is studied. Table 3 shows that the excessive number of discrete points will not only reduce the calculation efficiency, but also reduce the accuracy of the solution. In addition, the selection of the number of polynomials is also analyzed in this paper. In [28], the number of state polynomials and costate polynomials is the same and not studied separately. In our simulation, the results show that the selection of the number of state polynomials and costate polynomials can be different, and a better result can be obtained. If the number of the state polynomials is selected too much, then not only the calculation performance will be degraded but also the accuracy of the solution will be greatly affected. It is seen from Table 3 that when the number of costate polynomials is as large as the number of state polynomials, the allowed iteration number is reached and the iteration progress will not converge; it means the solution of the problem cannot be solved. In addition, appropriately increasing the number of costate polynomials can increase the accuracy of the solution, but it will reduce the computational efficiency. Thus, the number of discrete points and polynomials should be select carefully.

6. Conclusion

In this study, we proposed a new approach to solve the fuel-optimal problem in the ascending phase of the launching vehicle using TFC. The main conclusions can be summarized as follows:

(1) The first-order necessary condition of the optimization problem is constructed by the indirect method; the problem's constraints are embedded in the expression by using the TFC method

(2) The constrained optimization problem is transformed into an unconstrained optimization problem by using the TFC method, which reduces the search space of the solution, and a simple root-finding algorithm can be used to obtain the solution of the problem

(3) The residual of the solution is about 10^{-14} or less; for obtaining more accurate numerical solutions, the number of discrete points and polynomials should be selected carefully

(4) The proposed algorithm has the potential for online application. The calculation time of the algorithm is within 300 ms with MATLAB programming

Appendix

Partial Derivative of Loss Functions

The partial derivative of loss functions is

$$\frac{\partial \mathscr{L}_{a_i}}{\partial \xi_{v_j}} = \begin{cases} \dfrac{T}{m}\left(\dfrac{h_v^T(z)}{\|\lambda_v\|} - \dfrac{h_v^T(z)\lambda_{v_j}^2}{\|\lambda_v\|^3}\right), & i = j, \\ -\dfrac{h_v^T(z)\lambda_{v_i}\lambda_{v_j}}{\|\lambda_v\|^3}, & i \neq j, \end{cases} \tag{A.1}$$

$$\begin{aligned} \frac{\partial \mathscr{L}_{a_i}}{\partial \xi_t} &= 4b^3\left(h''(z) - \Phi_1''(z)h_0 - \Phi_2''(z)h_f - \Phi_3''(z)h_0' - \Phi_4''(z)h_f'\right)^T \xi_{a_i} \\ &\quad + 2b\left(\Phi_3''(z)v_{0_i} + \Phi_4''(z)v_{f_i}\right) - \frac{-2T}{\left(m_0 - (z+1)\dot{m}/b^2\right)^2}\frac{(z+1)\dot{m}}{b^3} \\ &\quad + \frac{-2\left(\Phi_3(z)v_{0_i} + \Phi_4(z)v_{f_i}\right)}{b^3\|r\|^3} + \frac{6\|r\|\left(\Phi_3(z)v_{0_i} + \Phi_4(z)v_{f_i}\right)}{b^3\|r\|^5}, \end{aligned} \tag{A.2}$$

$$\frac{\partial \mathscr{L}_{r_i}}{\partial \xi_{v_j}} = \begin{cases} h_v^T\dfrac{r_1}{\|r\|^3}, & i = j, \\ 0, & i \neq j, \end{cases} \quad \frac{\partial \mathscr{L}_{r_i}}{\partial \xi_t} = 2bh'^T_r\xi_{r_i}, \tag{A.3}$$

$$\frac{\partial \mathscr{L}_{v_i}}{\partial \xi_{r_j}} = \begin{cases} h'^T_r, & i = j, \\ 0, & i \neq j, \end{cases} \quad \frac{\partial \mathscr{L}_{v_i}}{\partial \xi_{v_j}} = \begin{cases} b^2h'^T_v, & i = j, \\ 0, & i \neq j, \end{cases} \tag{A.5}$$

$$\frac{\partial \mathscr{L}_{v_i}}{\partial \xi_t} = 2bh'^T_v\xi_{v_i}, \quad \frac{\partial \mathscr{L}_H}{\partial \xi_{r_i}} = h_{rf}^T v_{\mathrm{if}}, \tag{A.6}$$

$$\frac{\partial \mathscr{L}_H}{\partial \xi_{v_i}} = \left(-\frac{T}{m_0 - \left(\left(z_f - z_0\right)/b^2\right)\dot{m}}\frac{\lambda_{v_{\mathrm{if}}}}{\left\|\lambda_{vf}\right\|} - \frac{r_f}{\left\|r_f\right\|^3}\right)h_{vf}^T, \tag{A.7}$$

$$\frac{\partial \mathscr{L}_H}{\partial \xi_t} = -2T\left\|\lambda_{vf}\right\|\frac{-2\dot{m}}{b^3\left(m_0 - \left(\left(z_f - z_0\right)/b^2\right)\dot{m}\right)^2}, \tag{A.8}$$

$$\begin{aligned} \frac{\partial \mathscr{L}_H}{\partial \xi_{ai}} &= -\lambda_v\frac{\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^{\mathrm{T}}}{\|r\|^3} \\ &\quad - \frac{3r_i^2\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^5}, \end{aligned} \tag{A.8}$$

$$\frac{\partial \mathscr{L}_{a_i}}{\partial \xi_{a_j}} = \begin{cases} b^4\left(h''(z) - \Phi_1''(z)h_0 - \Phi_2''(z)h_f - \Phi_3''(z)h_0' - \Phi_4''(z)h_f'\right) + \dfrac{\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^3} - \dfrac{3r_i^2\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{r^5}, & i = j, \\ -\dfrac{3r_i r_j\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^5}, & i \neq j, \end{cases} \tag{A.9}$$

$$\frac{\partial \mathscr{L}_{r_i}}{\partial \xi_{a_j}} = \begin{cases} \lambda_{v_i}^T \left[\dfrac{\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^3}, \right. & \\ \left. -\dfrac{3r_i^2\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^5} \right], & i = j, \\ -\lambda_{v_i}^T \dfrac{3r_i r_j\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^5}, & i \neq j. \end{cases} \tag{A.10}$$

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] P. Lu, H. Sun, and B. Tsai, "Closed-loop endoatmospheric ascent guidance," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 283–294, 2003.

[2] P. Lu and B. Pan, "Highly constrained optimal launch ascent guidance," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 404–414, 2010.

[3] M. J. Grant and R. D. Braun, "Rapid indirect trajectory optimization for conceptual design of hypersonic missions," *Journal of Spacecraft and Rockets*, vol. 52, no. 1, pp. 177–182, 2014.

[4] F. Fahroo and I. M. Ross, "Advances in pseudospectral methods for optimal control," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, 2008.

[5] M. A. Patterson and A. V. Rao, "GPOPS-II: a MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming," *Acm Transactions on Mathematical Software*, vol. 41, no. 1, pp. 1–37, 2014.

[6] I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: from theory to flight," *Annual Reviews in Control*, vol. 36, no. 2, pp. 182–197, 2012.

[7] H. Peng, X. Wang, M. Li, and B. Chen, "An hp symplectic pseudospectral method for nonlinear optimal control," *Communications in Nonlinear Science and Numerical Simulation*, vol. 42, pp. 623–644, 2017.

[8] X. Wang, H. Peng, S. Zhang, B. Chen, and W. Zhong, "A symplectic pseudospectral method for nonlinear optimal control problems with inequality constraints," *ISA Transactions*, vol. 68, pp. 335–352, 2017.

[9] H. Peng, X. Wang, S. Zhang, and B. Chen, "An iterative symplectic pseudospectral method to solve nonlinear state-delayed optimal control problems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 48, pp. 95–114, 2017.

[10] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.

[11] B. Acikmese and S. R. Ploen, "Convex programming approach to powered descent guidance for Mars landing," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.

[12] M. W. Harris, *Lossless convexification of optimal control problems*, Doctor of Philosophy, The University of Texas at Austin, 2014, https://repositories.lib.utexas.edu/handle/2152/24904.

[13] X. Cheng, H. Li, and R. Zhang, "Efficient ascent trajectory optimization using convex models based on the Newton–Kantorovich/pseudospectral approach," *Aerospace Science and Technology*, vol. 66, pp. 140–151, 2017.

[14] F. F. Al-Bakri, A. F. Al-Bakri, and C. A. Kluever, "Approach and landing guidance for an unpowered gliding vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 12, pp. 2366–2371, 2020.

[15] Z. Y. Song, C. Wang, S. Theil et al., "Survey of autonomous guidance methods for powered planetary landing," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 5, pp. 652–674, 2020.

[16] B. Acikmese, J. M. Carson, and L. Blackmore, "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2104–2113, 2013.

[17] Y. Li, B. Pang, C. Wei, N. Cui, and Y. Liu, "Online trajectory optimization for power system fault of launch vehicles via convex programming," *Aerospace Science and Technology*, vol. 98, article 105682, 2020.

[18] Z. Wang and M. J. Grant, "Constrained trajectory optimization for planetary entry via sequential convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 10, pp. 2603–2615, 2017.

[19] A. E. Bryson and Y. Ho, *Applied Optimal Control*, Technometrics, 1975.

[20] L. Cheng, F. Jiang, Z. Wang, and J. Li, "Multi-constrained real-time entry guidance using deep neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 325–340, 2021.

[21] D. Izzo and E. Öztürk, "Real-time guidance for low-thrust transfers using deep neural networks," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 2, pp. 315–327, 2021.

[22] L. Cheng, H. Li, Z. Wang, and F. Jiang, "Fast solution continuation of time-optimal asteroid landing trajectories using deep neural networks," *Acta Astronautica*, vol. 167, pp. 63–72, 2020.

[23] D. Mortari, "The theory of connections: connecting points," *Mathematics*, vol. 5, no. 4, p. 4, 2017.

[24] D. Mortari, "Least-squares solutions of linear differential equations," 2017, http://arxiv.org/abs/1702.08437.

[25] H. Johnston and D. Mortari, "Least-squares solutions of boundary-value problems in hybrid systems," *Journal of Computational and Applied Mathematics*, vol. 393, article 113524, 2021.

[26] E. Schiassi, A. D'Ambrosio, A. Scorsoglio, R. Furfaro, and F. Curti, *Class of optimal space guidance problems solved via indirect methods and physics-informed neural networks*, 2021.

[27] E. Schiassi, A. D'Ambrosio, H. Johnston, R. Furfaro, F. Curti, and D. Mortari, *Complete energy optimal landing on small and large planetary bodies via theory of functional connections*, 2020.

[28] K. Drozd, R. Furfaro, E. Schiassi, H. Johnston, and D. Mortari, "Energy-optimal trajectory problems in relative motion solved via theory of functional connections," *Acta Astronautica*, vol. 182, pp. 361–382, 2021.

[29] D. Mortari, H. Johnston, and L. Smith, "High accuracy least-squares solutions of nonlinear differential equations," *Journal of Computational and Applied Mathematics*, vol. 352, pp. 293–307, 2019.

[30] L. Ping, "Introducing computational guidance and control," *Journal of Guidance Control & Dynamics*, vol. 40, no. 2, pp. 193–193, 2017.

[31] D. E. Rutherford and D. F. Lawden, "Optimal trajectories for space navigation," *The Mathematical Gazette*, vol. 48, no. 366, p. 478, 1964.

Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 9975007, 11 pages
https://doi.org/10.1155/2021/9975007

Research Article

A Two-Level Optimization Method for Hypersonic Periodic Cruise Trajectory

Hesong Li, Yi Wang, Yunfan Zhou, Shangcheng Xu, and Dan Su

College of Aerospace Science, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Yi Wang; 1262581502@qq.com

Received 23 March 2021; Revised 3 June 2021; Accepted 17 June 2021; Published 30 June 2021

Academic Editor: William MacKunis

Periodic cruise has the potential to improve the fuel-saving efficiency of hypersonic cruise vehicle but is difficult to optimize. In this paper, a two-level optimization method for the trajectory of periodic cruise is proposed. Due to that the periodic cruise trajectory can be divided into an acceleration phase where engine works and a glide phase where engine is off, the two-level optimization method is proposed to optimize the trajectory in each phase by the corresponding level. In the first level, Downhill Simplex Method (DSM) is employed to find an optimal angle of attack in the acceleration phase. Subsequently, the optimal trajectory in glide phase is obtained by the Pseudo-Spectral Method (PSM) in the second optimization level. Numerical results demonstrate the effectiveness of the proposed method. Finally, through comparing with steady-state cruise, it is concluded that periodic cruise makes full use of the change of atmospheric density and lift-drag ratio; thus, fuel saving is achieved.

1. Introduction

Hypersonic vehicle generally refers to the aircraft flying at Mach number above 5 [1], which has a series of advantages such as high flight altitude, fast speed, and strong penetration ability, and it has a far-reaching impact on the development of aerospace technology in the foreseeable future [2]. Therefore, the research on hypersonic technology has been widely concerned by researchers all over the world [3, 4].

As one of the important components of hypersonic flight technology, the design of flight trajectory is based on the aerodynamic characteristics of hypersonic vehicle and the complex flight environment, and it is aimed at minimum fuel consumption, longest range, or shortest flight time, with satisfying the complex path constraints including heating rate, acceleration load, dynamic pressure, and other terminal constraints [5]. Through reasonable design, the optimal flight trajectory can improve the efficiency of subsystem in the aircraft and reduce the cost of the whole flight. At the same time, it can provide guidance in the structural and material selection of the aircraft, which is conducive to the overall design of aircraft.

The whole trajectory of hypersonic cruise vehicle is generally divided into ascending stage, cruising stage, and gliding stage. Due to the difference of mission performed by hypersonic vehicle in each stage, the flight modes are also quite different. Therefore, it is necessary to establish appropriate performance indices for different stages, which makes the research face different problems [6]. The direction and magnitude of aerodynamic force in the ascending stage are closely related to the direction of thrust, and it is necessary to control the load caused by aerodynamic force [7]; in the cruising stage, it is difficult for hypersonic vehicle to maintain high maneuverability [8], while the constraints of terminal velocity and flight-path angle are very important to the gliding stage [9, 10]. Therefore, many researches can be carried out to solve these problems. Generally speaking, the cruising stage accounts for a large proportion in the whole flight process of hypersonic cruise vehicle, and the flight range depends on the cruising stage to a great extent. Therefore, it is of great significance to study the design of trajectory in the cruising stage.

To make the range of aircraft longer, an effective method is to improve the fuel efficiency of aircraft in the cruising stage. Finding the cruise trajectory with the highest fuel efficiency has been the focus of many researchers [11]. In these researches, steady-state cruise and periodic cruise are two main cruise modes. Steady-state cruise refers to the cruise

with constant altitude and speed [12]. Thus, it is relatively easy to optimize the trajectory of steady-state cruise because it is a two-degree-of-freedom (DOF) problem [13], while Irons et al. [14] have demonstrated that this trajectory is not optimal. The theoretically optimal trajectory is an infinite DOF curve, whose theoretical analysis is quite difficult as a two-point boundary value problem (TPBVP). Therefore, researchers tried to find a trajectory with better performance by dividing the cruise trajectory into a few phases which were uniform, and then, the concept of periodic cruise was raised. During periodic cruise, the trajectory approximates the form of a periodic function. The engine switches on and off according to a periodic law, which makes the powered propulsion and unpowered glide alternate, and the altitude, velocity, and other state variables change periodically. At the end of a cycle, the parameters of flight state are the same as that of the initial state [15].

There are many researches about the optimization of trajectory by theoretic analysis [16], but many of them are based on simplification; otherwise, it is difficult to carry out because of the complex characteristics of hypersonic flight. With the development of computational science and intelligent algorithm, many researchers have introduced optimization method to the field of aerospace [17, 18]. A number of trajectory optimization methods were proposed, which were well reviewed by [19], and it is validated that optimization method is effective to solve trajectory optimization problems [20]. In many researches about the optimization of cruise trajectory, it is generally considered that the trajectory of steady-state cruise is a suboptimal initial guess for the optimization of trajectory in periodic cruise [21], and the performance index of them is always used for comparison to draw relevant conclusions [22]. Many researches find that a certain amount of fuel can be saved through periodic cruise compared with steady-state cruise, and the trajectory of periodic cruise is more flexible. However, due to the complex characteristics of hypersonic flight, trajectory optimization is a multiobjective, multiconstraint, strong-coupling, and highly nonlinear optimization problem [23], which increases the difficulty of obtaining the optimal solution. In the optimization of periodic cruise trajectory, angle of attack and throttle are the two control variables, which are also functions about time and need to be determined to minimize the objective. While these two variables are nonlinear and discontinuous sometimes, thus, it is not easy to deal with them at the same time in an optimization algorithm unless new constraints are added. So far, there is no effective indirect numerical method and professional software that can deal with the problem of periodic cruise trajectory design in a unified framework [15]. Therefore, it is significant to explore new method to optimize the trajectory of periodic cruise.

In order to obtain the optimization result of periodic cruise trajectory quickly and robustly, many researchers introduced the idea of hierarchical optimization to separate optimization variables and optimize them, respectively, by different algorithms. Earlier, Subbarao and Shippey [24] proposed a trajectory optimization method by combining the collocation method with the genetic algorithm. The initial values of variables were selected by genetic algorithm to improve the optimization efficiency, and then, the trajectory was optimized by the collocation method. Kang et al. [25] studied the optimal periodic cruise trajectory by combining genetic algorithm and direct shooting method; then, a method of two-level optimization was developed to deal with the parameters of initial state in the outer loop and the control variables in the inner loop, respectively. In recent years, many new effective two-level trajectory optimization methods have been proposed and applied in trajectory optimization. Chai et al. [26] formulated and solved a constrained space maneuver vehicle trajectory optimization problem using a three-layer-hybrid optimal control solver, and good performance was obtained. Liu et al. [6] developed a two-level optimization algorithm to solve the optimal steady-state cruise trajectory by combining PSO algorithm with sequential quadratic programming. In [27], deep neural network was trained by generated trajectories from fuzzy multiobjective transcription method; then, a two-step strategy for real-time trajectory planning was proposed with feasibility and reliability confirmed. In this paper, due to the complexity of periodic cruise trajectory design, the concept of hierarchical optimization is also employed.

Considering that there are an acceleration phase and a glide phase in a period of periodic cruise, which can provide convenience to optimize the trajectory if the two phases can be optimized separately, a two-level optimization method which deals with the optimization problems in different phases by different levels, respectively, is proposed in this paper. The first level optimizes the acceleration phase, and the glide phase is optimized by the second level, and it is required that the fuel consumption averaged by range is lowest in the period. The second part of this paper introduces the models including parameterized aircraft model and dynamic equations. The description of optimization problem as well as method is also illustrated in detail. The solution of steady-state cruise and optimized result of periodic cruise are displayed in the third part. Finally, the difference of trajectory between steady-state cruise and periodic cruise is discussed, and the fuel-saving mechanism of periodic cruise is explored by contrast with steady-state cruise.

2. Models and Methods

2.1. Model of Hypersonic Vehicle. The HL-20 aircraft model [28] is widely employed in the research of trajectory optimization [29], and the aerodynamic coefficients are given in (1) [30]. Mach number is denoted by M. The coefficient of lift and drag are denoted by C_L and C_D, respectively. C_{D0} means the drag coefficient at zero attack of angle whose value is 0.008 when $M > 10$.

$$\begin{cases} C_L(M,\alpha) = C_{L0}(M) + C_{L\alpha}(M)\alpha, \\ C_D(M,\alpha) = C_{D0}(M) + K(M){C_L}^2, \\ C_{L0}(M) = \dfrac{1}{20\pi}\arctan\left[10(M-1)\right] - 0.035, \\ C_{L\alpha}(M) = 0.057\exp\left(-0.654M\right) + 0.014, \\ K(M) = 1.85[1 - \exp\left(-0.2356M\right)]. \end{cases} \tag{1}$$

Thrust coefficient is calculated in (2) [30], and the value of thrust force is calculated by (3), where s denotes the throttle, q means the aerodynamic pressure, and S_e is the reference area of engine whose value is 9.02 m^2. Angle of attack is denoted by α, and the unit is degree.

$$C_{T\max} = \begin{cases} 0.4736M^{1.5} + 1.6947M^{-2} \ (M < 4), \\ \dfrac{15(\alpha+5)^{0.25}}{M^{1.15}} \cdot \exp\left[-\dfrac{M^{0.08}}{200} \times \left(\alpha + 5 - \dfrac{35}{M^{0.6}}\right)^2\right] (M \geq 4), \end{cases} \tag{2}$$

$$T = sqC_{T\max}S_e. \tag{3}$$

The specific impulse of engine is computed by (4), and fuel consumption per second is calculated by (5), where h denotes flight altitude and g denotes the acceleration of gravity.

$$I_{\text{sp}} = \begin{cases} 4500 - 10(h-20) & (M < 4), \\ -245M + 5480 - 10(h-20) & (M \geq 4), \end{cases} \tag{4}$$

$$\frac{dm}{dt} = -\frac{T}{gI_{\text{sp}}}. \tag{5}$$

The 1976 U.S. Standard Atmosphere Model is used. An altitude factor is defined by [21], where R_e means the radius of earth.

$$H = \frac{h}{1 + h/R_e}. \tag{6}$$

When flight altitude is in the range of 32 to 47 km, the atmospheric density, denoted by ρ, is calculated in (7), where $\rho_0 = 1.225$ kg/m^3.

$$\begin{aligned} W &= 1 + \frac{H - 39.7499}{89.4107}, \\ \rho &= 3.2618 \times 10^{-3}\rho_0 W^{-13.2011}. \end{aligned} \tag{7}$$

For the sake of simplicity, the earth is considered to be a homogeneous sphere, and the acceleration of gravity is a constant at 9.8 m/s^2. The velocity of sound, which is denoted by a, can be regarded as a constant whose value is 340.294 m/s [30]. Then, the dynamic model is described in (8). The flight-path angle is denoted by γ. T, L, and D are thrust, lift, and drag, respectively; m means the mass of aircraft whose value is 89930 kg, and r denotes the flight range. Based on (8), the trajectory of aircraft can be simulated.

Table 1: The limits of control variables.

Variable	Lower limit	Upper limit
α (°)	5	20
s	0	1

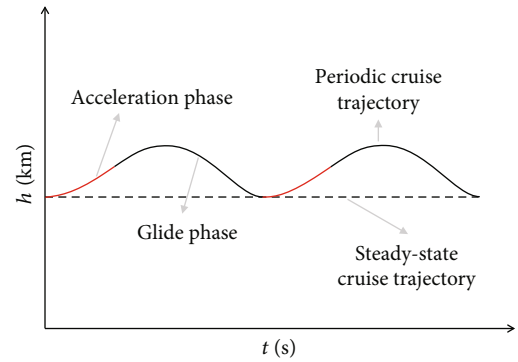


Figure 1: The flight plan of periodic cruise.

$$\begin{cases} \dfrac{dh}{dt} = M \cdot a \cdot \sin\gamma, \\ \dfrac{dM}{dt} = \dfrac{T\cos\alpha - D - mg\sin\gamma}{m \cdot a}, \\ \dfrac{d\gamma}{dt} = \dfrac{T\sin\alpha + L}{mM \cdot a} + \cos\gamma\left(\dfrac{M \cdot a}{R_e + h} - \dfrac{g}{M \cdot a}\right), \\ \dfrac{dr}{dt} = M \cdot a \cdot \cos\gamma\left(\dfrac{R_e}{R_e + h}\right). \end{cases} \tag{8}$$

The optimization of aircraft trajectory is a kind of classic optimal control problem. For a periodic dynamic system whose period is t_c, the dynamic equation can be formulated as (9), where x denotes the array of state variables and u means the array of state variables.

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

The function of $u(t)$ needs to be determined in order to minimize the index function in (10) with the terminal constraints in (11) satisfied.

$$J = \int_0^{t_c} g(x(t), u(t))dt, \tag{10}$$

$$x(0) = x(t_c). \tag{11}$$

Therefore, in the optimization of trajectory, h, M, and γ are regarded as state variables, which is formulated as $x = (h, M, \gamma)^T$. α and s are regarded as control variables, which is formulated as $u = (\alpha, s)^T$. And the optimization of trajectory is to determine the function of α and s to minimize the objective which is relative to the parameters of trajectory.

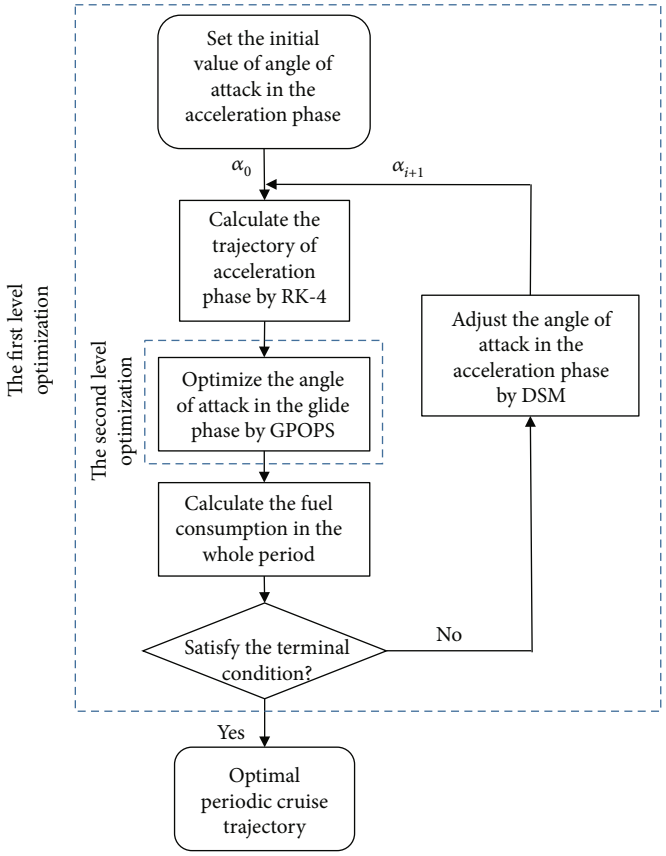


Figure 2: The process of two-level optimization.

2.2. Method for Steady-State Cruise. To compare with the result of periodic cruise, the trajectory of steady-state cruise needs to be solved first.

In steady-state cruise, γ is 0 and h and M remain unchanged. The first three equations in (8) should be equal to 0. While γ is 0 so that the first formula is naturally satisfied, thus, (12) is obtained. L and D are related to h, M, and α, while T is related to h, M, α, and s. Overall, there are two equations with four variables.

$$\begin{cases} \dfrac{dM}{dt} = \dfrac{T\cos\alpha - D}{m \cdot a} = 0, \\ \dfrac{d\gamma}{dt} = \dfrac{T\sin\alpha + L}{mM \cdot a} + \dfrac{M \cdot a}{R_e + h} - \dfrac{g}{M \cdot a} = 0. \end{cases} \tag{12}$$

Given the value of h and M, the other two variables can be solved and the fuel consumption averaged by range can also be obtained. Firstly, T can be eliminated by (12), and then, (13) can be obtained.

$$D\tan\alpha + L - mg + m\frac{(M \cdot a)^2}{(R_e + h)} = 0. \tag{13}$$

Then, a transcendental equation which is only related to α can be obtained by substituting (14) into (13). After the value of α is obtained, s can be also calculated by (12).

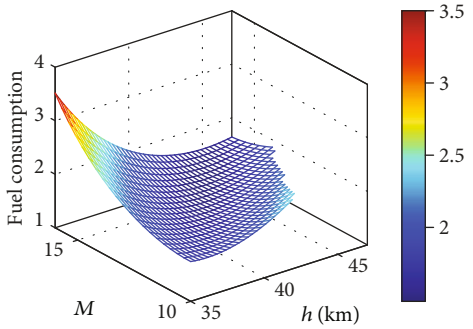


Figure 3: The distribution of fuel consumption averaged by range at different flight altitudes and Mach numbers.

$$\begin{aligned} L &= C_L \cdot \frac{1}{2}\rho(M \cdot a)^2 \cdot S, \\ D &= C_D \cdot \frac{1}{2}\rho(M \cdot a)^2 \cdot S. \end{aligned} \tag{14}$$

Owning to that γ is 0 while h and M remain unchanged. The fuel consumption averaged by flight range can be simplified as in (15) and [6]. Thus, all parameters in the trajectory of steady-state cruise can be solved.

$$J = \frac{T}{gI_{sp}M \cdot a}\left(1 + \frac{h}{R_e}\right). \tag{15}$$

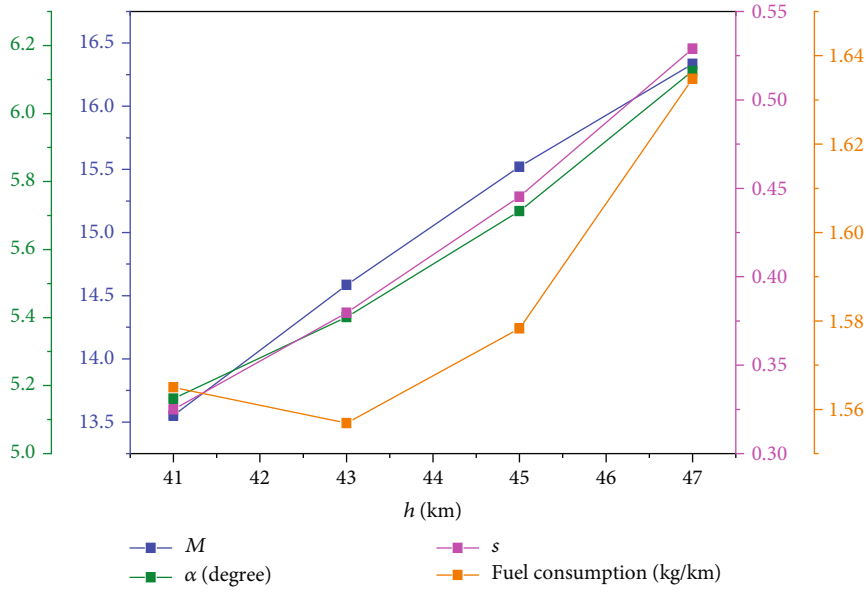


Figure 4: The local optimal parameters of steady-state cruise at different flight altitudes.

2.3. Optimization Problem of Periodic Cruise. In a periodic cruise whose period is donated by t_c, the terminal constraints are illustrated in (16).

$$\begin{cases} h(0) = h(t_c), \\ v(0) = v(t_c), \\ \gamma(0) = \gamma(t_c). \end{cases} \tag{16}$$

The objective of optimization is to minimize the fuel consumption averaged by flight range in a cruise period, which is

$$J = \frac{\int_0^{t_c} |\dot{m}|\, dt}{\int_0^{t_c} \dot{r} dt} = \frac{m(0) - m(t_c)}{r_c}. \tag{17}$$

The limits of α and s are determined according to the original data benchmark [15], as shown in Table 1.

Therefore, the optimization problem of periodic cruise trajectory can be formulated as (18).

$$\begin{aligned} &\text{Minimize} \quad J = \frac{m(t_c) - m(0)}{r_c}, \\ &\text{subject to} \begin{cases} \alpha(t) \in [5, 20], \\ s(t) \in [0, 1], \\ h(0) = h(t_c), \\ v(0) = v(t_c), \\ \gamma(0) = \gamma(t_c). \end{cases} \end{aligned} \tag{18}$$

In [31], based on a simplified dynamic model, the value of throttle in periodic cruise was solved by the minimum principle. It proved that the value of throttle was only 0 or 1, and the engine only started once in a cycle. Gao et al. [15] regarded the highest point of periodic cruise trajectory as the starting point and assumed that the curve which depicted the change of altitude was approximately a cosine curve with the increase of range. The cosine curve was introduced into the optimal control problem as a path constraint, and the parameters which determined the shape of cosine curve were regarded as the augmented state. The optimized results were similar to that in [31]: the throttle switched between 0 and 1. Based on the results of above references, the flight plan of periodic cruise selected in this paper is as shown in Figure 1: the engine switches on when γ is equal to 0, and it lasts for 60 seconds; then, the aircraft begins to glide. The throttle is 1 when starting, and the rest is 0.

2.4. Two-Level Optimization Method for Periodic Cruise. As shown in Figure 1, during the period, the engine switches on and off periodically; thus, the whole trajectory in a period can be divided into two phases. One is the acceleration phase when engine switches on and aircraft speeds up, and the other is the glide phase when engine switches off and aircraft glides without propulsion. Due to a relatively small proportion of acceleration phase [31], the angle of attack in this phase is regarded unchanged, and its value is regarded as a variable which needs to be optimized. Therefore, when studying the acceleration phase, it is a kind of parameter optimization problem. In this paper, the optimization about the angle of attack in the acceleration phase is carried out in the first level of the two-level method by the Downhill Simplex Method (DSM). The DSM is a geometrically intuitive algorithm. In two dimensions, the simplex is a triangle and it is a tetrahedron in three dimensions. As the algorithm proceeds, the simplex makes way downward toward the location of the minimum through series of steps.

The glide phase accounts for a large proportion in the whole flight period, and the angle of attack in this phase cannot be regarded as a constant. Therefore, it is a dynamic optimal control problem whose variables change with the increase of time. In the glide phase, there is no propulsion

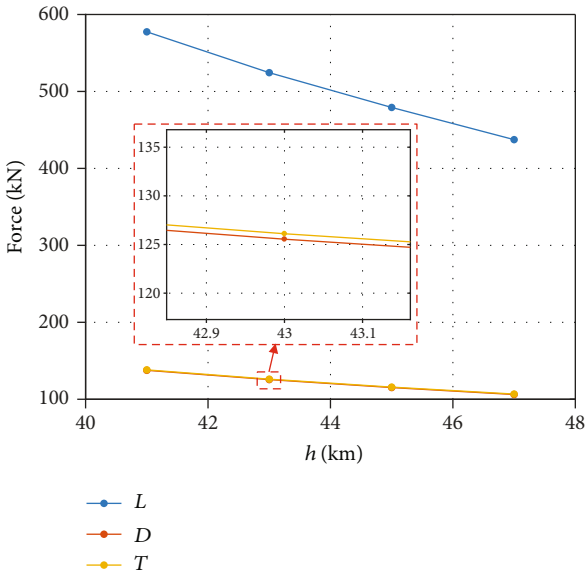


Figure 5: The value of lift, drag, and thrust of steady-state cruise at different flight altitudes.

and the throttle is 0; thus, the angle of attack is the only one control variable. In recent years, direct methods, especially Pseudo-Spectral Methods (PSMs), have become increasingly mature with the development of computers and optimization algorithm. A number of improvements for PSM and combinations with other algorithms were proposed [32]. The GPOPS (Gauss Pseudo-spectral Optimization Software), which is based on PSM and some effective nonlinear programming methods such as sequential quadratic programming (SQP), has been successfully applied in trajectory optimization problems. To optimize the angle of attack in the glide phase quickly and accurately, the GPOPS is employed as the second-level optimization, whose target is the longest flight range in the glide phase and the terminal constraint is (16).

Therefore, due to the division of the whole trajectory, the whole optimization is divided into a parameter optimization problem and an optimal control problem, which are solved in different levels, respectively. Thus, the two levels of the optimization method are mainly divided by different phases of trajectory. Aiming at the minimal fuel consumption averaged by flight range, the process of two-level optimization is shown in Figure 2 and the details are as follows:

Step 1. Set the initial value of angle of attack in the acceleration phase, and calculate the trajectory by the forth order Runge-Kutta (RK-4) method.

Step 2. Take the terminal point of acceleration phase as the starting point, and optimize the angle of attack in glide phase by the GPOPS, whose target is the longest flight range.

Step 3. Calculate the fuel consumption averaged by range in the whole period, and the result is fed back to the Downhill Simplex Method.

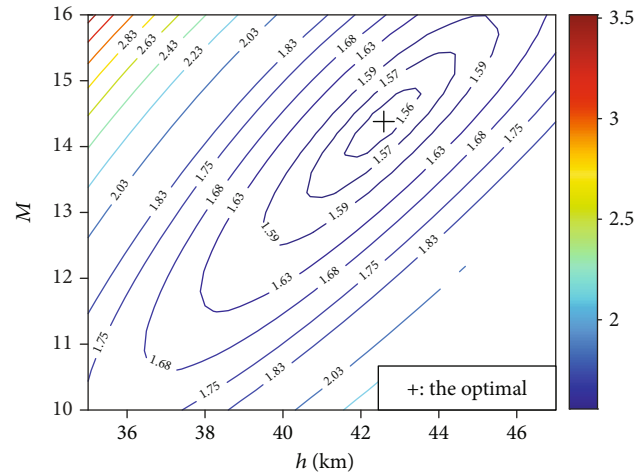


Figure 6: The contour of fuel consumption at different flight altitudes and Mach numbers.

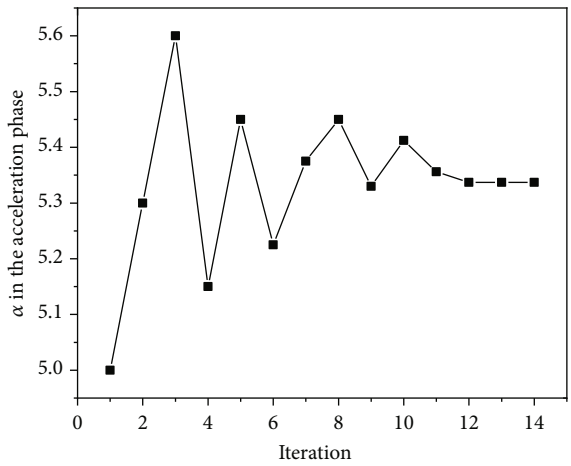


Figure 7: The change of angle of attack in the acceleration phase during the optimization process.

Step 4. Adjust the angle of attack in the acceleration phase by Downhill Simplex Method, and calculate the trajectory.

Step 5. Repeat Step 2 to Step 4 until the algorithm converges.

3. Results and Analysis

3.1. Result of Steady-State Cruise. In steady-state cruise, the distribution of fuel consumption averaged by range at different flight altitudes and Mach numbers is displayed in Figure 3. It can be seen that there is a minimal fuel consumption, whose value is 2 kg/km lower than the maximum. Thus, the optimization of trajectory is significant to reduce fuel consumption.

At a constant flight altitude, with the change of Mach number, there is an optimal flight state whose fuel consumption is the minimum, and it is named the local optimum, while the state with the minimal fuel consumption at all flight altitudes is named the global optimum. Some parameters of local optimums at different flight altitudes are displayed in

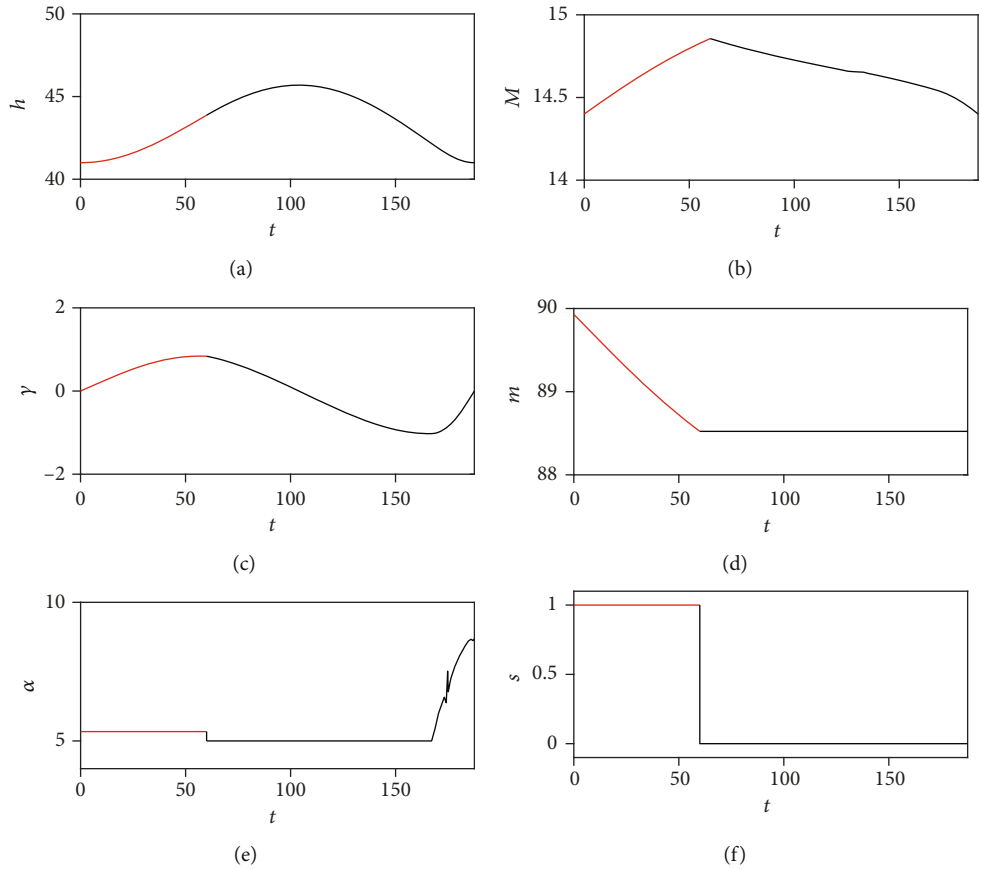


FIGURE 8: The flight parameters in the whole periodic cruise at 41 km: (a) flight altitude (km); (b) Mach number; (c) fight-path angle (degree); (d) mass (ton); (e) angle of attack (degree); (f) throttle.

Figure 4. With the enhancement of flight altitude, the local optimal Mach number and angle of attack as well as throttle increase, while the fuel consumption averaged by range decreases firstly and then increases. Thus, it can be concluded that the global optimal flight altitude of steady-state cruise is around 43 km.

Figure 5 displays the change of lift, drag, and thrust with the increase of flight altitude. It can be seen that all forces decease if flight altitude is higher; thus, the increase of flight altitude tends to reduce the value of drag, which is also beneficial to reduce fuel consumption. However, from Figure 4, a higher flight altitude is accompanied with a higher flight Mach number, which will cause the decrease of impulse according to (4) and thus the efficiency of engine reduces. Therefore, there is an intervening flight altitude between the two aspects to obtain a relatively lower drag and a relatively higher impulse, and that is the point where the global optimum locates. Overall, the global optimal parameters of steady-state cruise trajectory are a result of the combination of the two aspects; thus, minimal fuel is consumed.

By the method of traversal, the flight altitude of the global optimum in steady-state cruise is at 42.6 km and the Mach number is 14.4. The fuel consumption averaged by range is 1.556 kg/km. Figure 6 shows the contour of fuel consumption at different flight altitudes and Mach numbers. It can be seen that if one of the flight altitudes or Mach numbers is away from the global optimal point, more fuel will be consumed, and the total flight range will decrease. However, a hypersonic vehicle usually cannot always work at a state around the optimal point because of the constraints of structure and the requirements of mission. Therefore, it is significant to explore periodic cruise whose trajectory is more flexible with the fuel consumption reduced.

3.2. Optimization Results of Periodic Cruise. Based on the two-level optimization method, the starting point is set in the flight altitude at 41 km and Mach number at 14.4. The initial value of optimization variable in the first level is set as 5°; then, the whole two optimization levels are implemented. Figure 7 shows the change of angle of attack in the acceleration phase during the optimization process, which is convergent at 5.337° after 12 iterations of optimization. The complete trajectory and flight parameters in a whole period are displayed in Figure 8, where the trajectory of acceleration phase is depicted by red lines while that of glide phase is described by black lines. The fuel consumption averaged by range in the whole period is 1.514 kg/km, whose fuel-saving efficiency is 5.1% compared with that of steady-state cruise at 1.596 kg/km at the same flight altitude and Mach number. Therefore, periodic cruise has more advantages in fuel saving, and the two-level optimization method is validated as well.

In last section, it is found that the flight altitude of optimal steady-state cruise point with the minimal fuel

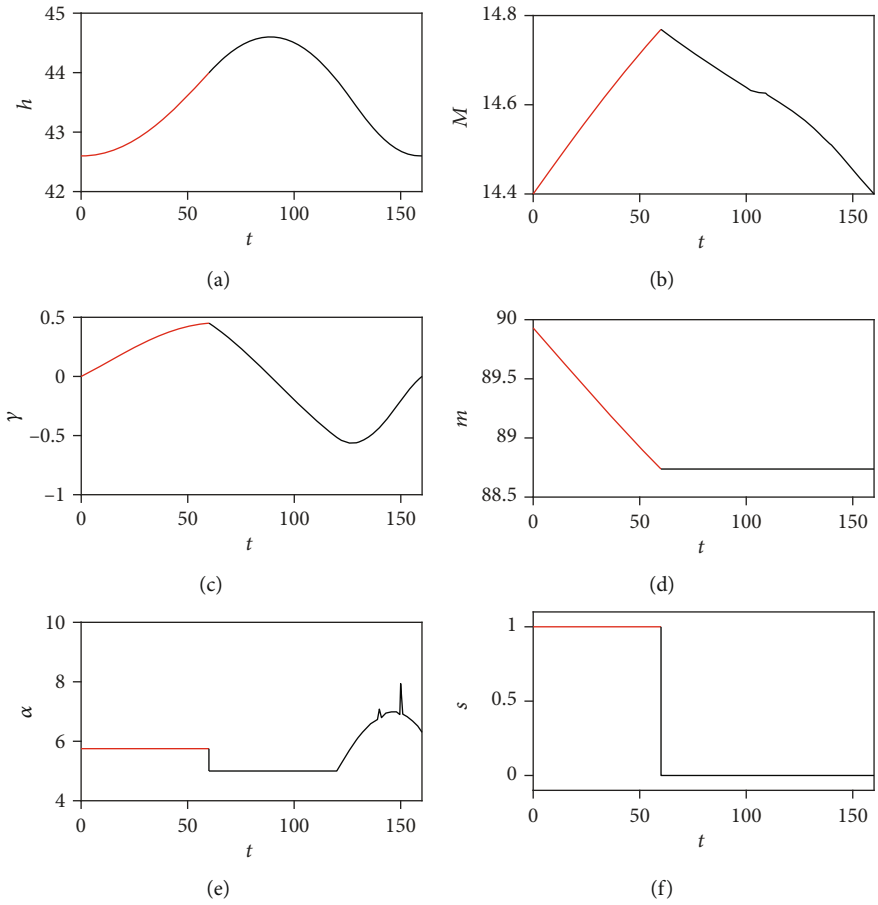


Figure 9: The flight parameters in the whole periodic cruise at 42.6 km: (a) flight altitude (km); (b) Mach number; (c) fight-path angle (degree); (d) mass (ton); (e) angle of attack (degree); (f) throttle.

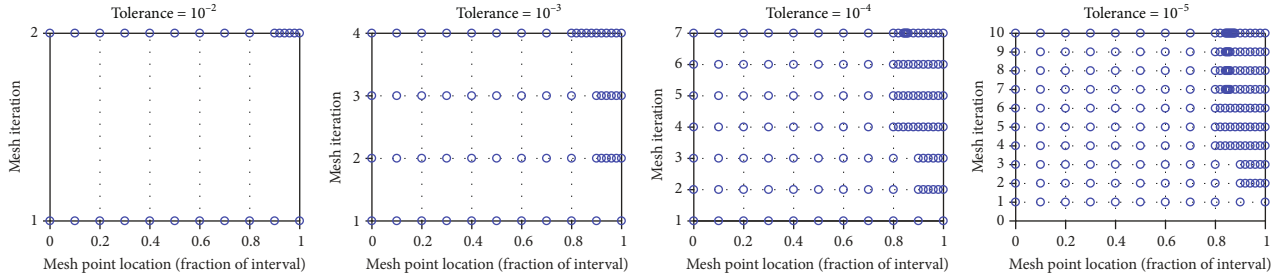


Figure 10: Mesh grid distribution with different accuracy tolerances.

consumption is at 42.6 km and the Mach number is 14.4. To further validate the two-level optimization method and explore the performance of periodic cruise, optimization of periodic cruise is carried out at the optimal point of steady-state cruise. The initial value of angle of attack in the acceleration phase is set as 5° as well. Figure 9 shows the optimized trajectory and flight parameters in the whole periodic cruise. The optimal value of angle of attack in the acceleration phase is 5.751°, and the fuel consumption averaged by range in the whole period is 1.511 kg/km, whose fuel-saving efficiency is 2.92% compared with that of steady-state cruise at 1.556 kg/km. A result with less fuel consumption can be obtained by the two-level optimization even at the optimal steady-state cruise point; thus, the effectiveness of optimization method proposed is proved further.

3.3. Analysis of Parameters in Method. To detail the influence of the GPOPS on the whole method, the distribution of mesh points with different tolerance is displayed in Figure 10. It can be seen that more mesh points are needed to achieve a higher accuracy. It is worth mention that the angle of attack in the acceleration phase changes in the optimization process, which generates different trajectories of the acceleration phase. Thus, the optimization of the trajectory in glide phase by the GPOPS is implemented in different starting points during the optimization process, while generally the GPOPS

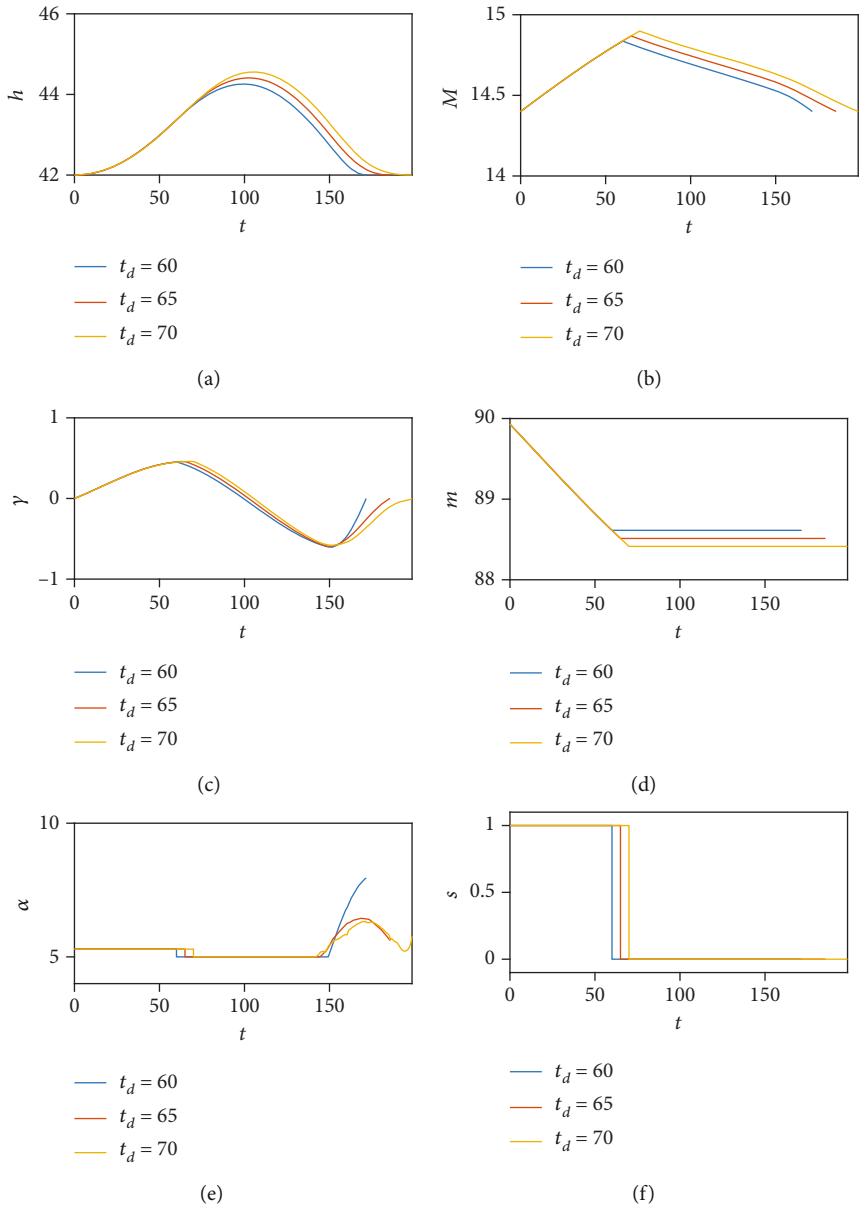


FIGURE 11: The flight parameters in the whole periodic cruise under different t_d: (a) flight altitude (km); (b) Mach number; (c) fight-path angle (degree); (d) mass (ton); (e) angle of attack (degree); (f) throttle.

needs a lot of time to obtain a robust result with a small tolerance at most starting points. Thus, the optimization would cost more time if smaller tolerance is required, and the tolerance should be selected to balance computational accuracy and speed.

In addition, the length of duration when engine works, denoted by t_d, is set at 60 s in previous study. To detail the effect of t_d on the method, a parameter variation of t_d is carried out. When that value of t_d is 60 s, 65 s, and 70 s, the optimal angle of attack in the acceleration phase is obtained by the proposed method and the value is 5.295°, 5.291°, and 5.288°, respectively, which decrease slightly. And the reason should be that a high velocity can be guaranteed at the end of acceleration phase if t_d is large, while a small angle of attack is beneficial to enhance the impulse and reduce the drag. As shown in Figure 11, different optimal trajectories are obtained by the proposed method, which confirm the applicability and feasibility.

3.4. Analysis of Periodic Cruise Trajectory. When the flight altitude of starting point is at 41 km and the Mach number is 14.4, a fuel-saving efficiency more than 5% is achieved by periodic cruise. Take this as an example to explore the mechanism of fuel saving in periodic cruise.

Figure 12 shows the change of drag during periodic cruise and steady-state cruise, respectively. Combined with Figure 8, it can be seen that the aircraft accelerates and ascents with the propulsion provided by engine. Then, the engine switches off, and the aircraft begins to glide without power. At this time, the flight altitude is relatively high and

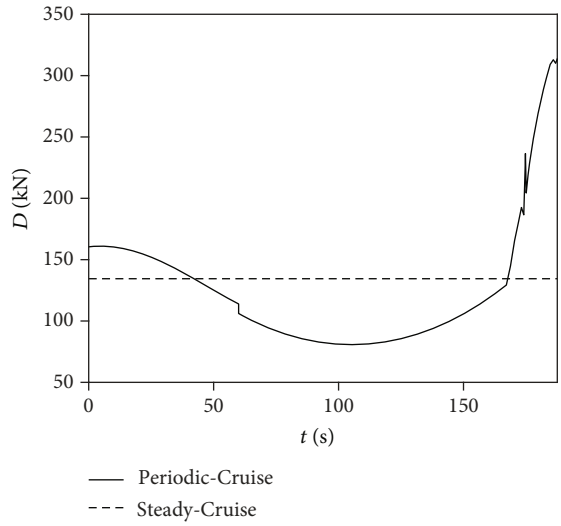


Figure 12: The change of drag in periodic cruise and steady-state cruise.

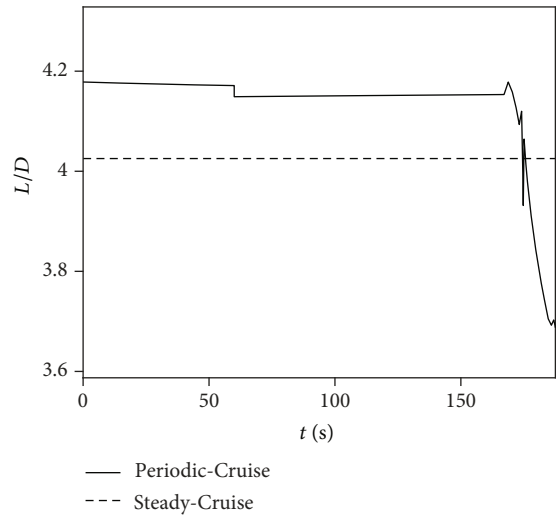


Figure 13: The change of lift-drag ratio in periodic cruise and steady-state cruise.

the atmospheric density is low, which is suitable for glide because of the small drag as shown in Figure 12. When the flight altitude is relatively low, the engine switches on again and the acceleration phase starts. At this time, the atmospheric density is large, which is beneficial for combustion in engine, and larger impulse is achieved according to (4). Therefore, the whole trajectory of periodic cruise shuttles between an atmosphere layer with relatively high density and an atmosphere layer with relatively low density, and it is similar to the ducks and drakes. The trajectory of periodic cruise makes full use of the change of atmospheric density to reduce drag in the glide phase and enhance impulse in the acceleration phase; thus, less fuel is consumed compared with that of steady-state cruise.

According to the Breguet Range Equation [33], a large lift-drag ratio is beneficial for a longer flight range. The change of lift-drag ratio in periodic cruise and steady-state cruise is displayed in Figure 13. It can be seen that the value of lift-drag ratio in periodic cruise is mostly between 4.1 and 4.2, while that of steady-state cruise is around 4.0, which is also a reason why fuel saving is achieved in periodic cruise.

Therefore, the trajectory of periodic cruise utilizes the change of atmospheric density to reduce drag and enhance impulse and adopts flight states with higher lift-drag ratio to enhance flight range. Therefore, less fuel is consumed compared with that of steady-state cruise.

4. Conclusions

Based on the characteristic of periodic cruise trajectory that the whole trajectory can be divided into an acceleration phase and a glide phase, a two-level optimization method which combines the Downhill Simplex Method with the Pseudo-Spectral Method is proposed in this paper; then, the trajectory of periodic cruise for hypersonic vehicle is optimized and analyzed. Conclusions can be drawn as follows:

(1) The proposed optimization method can deal with the design problem of periodic cruise trajectory with feasibility confirmed

(2) The fuel consumption averaged by range in periodic cruise trajectory is less than that of steady-state cruise trajectory, which means periodic cruise can save a certain amount of fuel and make range longer

(3) The trajectory of periodic cruise makes more use of atmospheric density and lift-drag ratio of aircraft, and therefore, fuel saving is achieved

In future work, the length of duration when engine works can be also regarded as an optimization variable, while more improvements are needed to make sure credible results can be obtained by PSM. To enhance the accuracy of results and make the method more robust, new method, such as initial guess generator technique, can be introduced.

Data Availability

The image data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the Postgraduate Scientific Research Innovation Project of Hunan Province (Grant number: CX20200084). The authors want to express their thanks for the support from Tao Tang.

References

[1] Z. Chen and Y. Zhao, "Aerothermoelastic analysis of a hypersonic vehicle based on thermal modal reconstruction,"

International Journal of Aerospace Engineering, vol. 2019, 13 pages, 2019.

[2] J. Zhao and R. Zhou, "Reentry trajectory optimization for hypersonic vehicle satisfying complex constraints," *Chinese Journal of Aeronautics*, vol. 26, no. 6, pp. 1544–1553, 2013.

[3] X. Zhang, J. Huang, G. Wang, and L. Li, "Hypersonic target tracking with high dynamic biases," *IEEE Transactions on Aerospace & Electronic Systems*, vol. 55, no. 1, pp. 506–510, 2019.

[4] Y. Fan, P. Yan, F. Wang, and H. Xu, "Discrete sliding mode control for hypersonic cruise missile," *International Journal of Aerospace Engineering*, vol. 2016, 9 pages, 2016.

[5] G. N. Kumar, M. Ikram, and A. K. Sarkar, "Hypersonic flight vehicle trajectory optimization using pattern search algorithm," *Optimization and Engineering*, vol. 19, no. 1, pp. 125–161, 2018.

[6] X. Liu, Y. Wang, L. Liu, and J. Huang, "Optimization on steady-state cruise for a hypersonic vehicle," in *Proceedings of the 33rd Chinese Control Conference*, pp. 8935–8940, Nanjing, China, 2014.

[7] B. Pan and P. Lu, "Improvements to optimal launch ascent guidance," in *AIAA Guidence, Navigation, and Control Conference*, vol. 8174, pp. 2–5, Toronto, Ontario, Canada, 2012.

[8] G. Cai, G. Duan, C. Hu, and B. Zhou, "Tracking control for air-breathing hypersonic cruise vehicle based on tangent linearization approach," *Journal of Systems Engineering and Electronics*, vol. 21, no. 3, pp. 469–475, 2010.

[9] Z. Huang, Y. Zhang, and Y. Liu, "Research on state estimation of hypersonic glide vehicle," *Journal of Physics: Conference Series*, vol. 1060, no. 1, article 012088, 2018.

[10] M. Xu, L. Liu, G. Tang, and K. Chen, "Study on guidance approaches of hypersonic glide-cruise vehicle during cruise and attack phases," in *Proceedings of 3rd International Symposium on Systems and Control in Aeronautics and Astronautics*, pp. 1491–1495, Harbin, China, 2010.

[11] P. F. Gath and A. J. Calise, "Optimization of launch vehicle ascent trajectories with path constraints and coast arcs," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 296–304, 2001.

[12] D. E. Koelle, "On the optimum cruise speed of a hypersonic aircraft," *IEEE Aerospace and Electronic Systems Magazine*, vol. 4, no. 5, pp. 13–16, 1989.

[13] K. Wang, B. Zhang, and Y. Hou, "Multiobjective optimization of steady-state cruise trajectory for a hypersonic vehicle," in *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)*, pp. 130–135, Beijing, 2017.

[14] J. R. Irons, R. L. Schultz, and N. R. Zagalsky, "Energy state approximation and minimum-fuel fixed-range trajectories," *Journal of Aircraft*, vol. 8, no. 6, pp. 488–490, 1971.

[15] H. Gao, Z. Chen, M. Sun, Z. Wang, and Z. Chen, "General periodic cruise guidance optimization for hypersonic vehicles," *Applied Sciences*, vol. 10, no. 8, p. 2898, 2020.

[16] P. Menon, G. Sweriduk, and A. Bowers, "A study of near-optimal endurance-maximizing periodic cruise trajectories, AIAA 2005-6153," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California, 2005.

[17] C. F. Ollivier-Gooch and M. Azab, "High-order aerodynamic optimization using new hybrid sequential quadratic programing-particle swarm intelligence technique," in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2012.

[18] D. Spiller, L. Ansalone, and F. Curti, "Particle swarm optimization for time-optimal spacecraft reorientation with keep-out cones," *Journal of Guidance, Control, and Dynamics*, vol. 39, pp. 312–325, 2016.

[19] 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.

[20] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, pp. 193–207, 1998.

[21] W. Wang, Z. Hou, S. Shan, and L. Chen, "Optimal periodic control of hypersonic cruise vehicle: trajectory features," *IEEE Access*, vol. 7, pp. 3406–3421, 2019.

[22] P. H. Carter, D. J. Pines, and L. V. Rudd, "Approximate performance of periodic hypersonic cruise trajectories for global reach," *IBM Journal of Research and Development*, vol. 35, no. 5, pp. 703–714, 2000.

[23] G. Huang, Y. Lu, and Y. Nan, "A survey of numerical algorithms for trajectory optimization of flight vehicles," *Science China Technological Sciences*, vol. 55, no. 9, pp. 2538–2560, 2012.

[24] K. Subbarao and B. M. Shippey, "Hybrid genetic algorithm collocation method for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1396–1403, 2009.

[25] B. N. Kang, D. B. Spencer, S. Tang, and D. Jordan, "Optimal periodic cruise trajectories via a two-level optimization method," *Journal of Spacecraft and Rockets*, vol. 47, no. 4, pp. 597–613, 2015.

[26] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Trajectory optimization of space maneuver vehicle using a hybrid optimal control solver," *IEEE Transactions on Cybernetics*, vol. 49, pp. 467–480, 2017.

[27] 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*, vol. 67, pp. 6904–6915, 2019.

[28] C. Chuang and H. Morimoto, "Optimal periodic cruise for a hypersonic vehicle with constraints," in *Guidance, Navigation, and Control Conference*, San Diego, CA, U.S.A., 1996.

[29] G. M. Ware and C. I. Cruz, "Aerodynamic characteristics of the HL-20," *Journal of spacecraft and rockets*, vol. 30, no. 5, pp. 529–536, 1993.

[30] C. H. Chuang and H. Morimoto, "Periodic optimal cruise for a hypersonic vehicle with constraints," *Journal of spacecraft and rockets*, vol. 34, no. 2, pp. 165–171, 1997.

[31] R. Li and Y. Shi, "The fuel optimal control problem of a hypersonic aircraft with periodic cruising mode," *Mathematical and Computer Modelling*, vol. 55, no. 11-12, pp. 2141–2150, 2012.

[32] D. Christopher, H. William, and R. Anil, "An improved adaptive hp algorithm using pseudospectral methods for optimal control," in *AIAA/AAS Astrodynamics Specialist Conference*, Toronto, Ontario, Canada, 2010.

[33] L. Bréguet, "Calcul du poids de combustible consummé par un avionen vol ascendant," *Comptes Rendus del'Academie des Sciences*, vol. 177, pp. 870–872, 1923.

Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 9977262, 11 pages
https://doi.org/10.1155/2021/9977262

Research Article

Modelling and Simulation of Distributed UAV Swarm Cooperative Planning and Perception

Haifeng Ling,[1] **Hongchuan Luo**,[1,2] **Haisong Chen**,[1] **Linyuan Bai**,[1] **Tao Zhu**,[1] **and Yanjun Wang**[2]

[1]*College of Field Engineering, Army Engineering University of PLA, Nanjing 210003, China*
[2]*No. 94789 Unit of PLA, Nanjing 210003, China*

Correspondence should be addressed to Hongchuan Luo; luohc2004@outlook.com

Received 29 March 2021; Revised 14 May 2021; Accepted 19 May 2021; Published 30 May 2021

Academic Editor: Mahmut Reyhanoglu

As an emerging topic, the swarm of autonomous unmanned aerial vehicles (UAVs) has been attracting great attention. Due to the indeterminacy of sensors, distributed cooperative swarms have been considered to be efficient and robust but challenging to design and test. To facilitate the development of distributed swarms, it has been proposed to utilise a simulation platform for cooperative UAVs using imperfect perception. However, the existing simulation platforms cannot satisfy this demand due to a few reasons. First, they are designed for a specific purpose, and their functionalities are difficult to extend. Second, the existing platforms lack compatibility to be applied to different types of scenarios. Third, the modelling of these platforms is too simplified to simulate flight motion dynamic and noisy communication accurately, which may cause a difference in performance between the simulation and real-world application. To address the mentioned issues, this paper models the problem and proposes a simulation platform for distributed swarm cooperative perception, which addresses software engineering concerns and provides a set of extendable functionalities of a cooperative swarm, including communication, estimation, perception fusion, and path planning. The applicability of the proposed platform is verified by simulations with the real-world application. The simulation results demonstrate that the proposed system is viable.

1. Introduction

A swarm of autonomous unmanned aerial vehicles (UAVs) has been an emerging topic in the fields of manufacture, disaster rescue, and the military [1]. Autonomous unmanned swarms are expected to outperform a single complex UAV in terms of flexibility and robustness, thus offering enhanced adaptability, survivability, and fault tolerance. For instance, in strike coordination and reconnaissance missions, a swarm of UAVs can cover a given area in a shorter time than a single one, and a failure of any swarm member will not cause a failure of the entire mission, as shown in Figure 1. However, due to the complexity of the joint decision, the larger the swarm is, the higher the demand for cooperative planning and perception algorithm will be. To achieve the aforementioned benefits, the interswarm cooperative planning and perception algorithm, as well as many other factors, need to be studied in-depth.

First, the interswarm planning and perception algorithm should be fully distributed. Otherwise, the swarm is vulnerable to the failure of certain member. Take a centralised swarm as a counterexample, where a swarm is controlled by a central member (leader), who fuses the perceptions of members and makes a planning decision for them. Once the leader fails, the swarm also fails; thus, the swarm is considered to be nonresilient. Second, the interswarm planning and perception algorithm should be intelligent. The swarm performance should increase with the swarm scale. Lastly, to be applicable in the real-world, physical constraints of the swarm, such as packet loss in communication and noise influenced sensor perception, need to be taken into account.

Studies on the swarm cooperative planning and perception have been limited by many factors, such as the difficulty of a real-world experiment due to the lack of funds to build a UAV swarm, a risk of damage caused by a fall down [2] of a

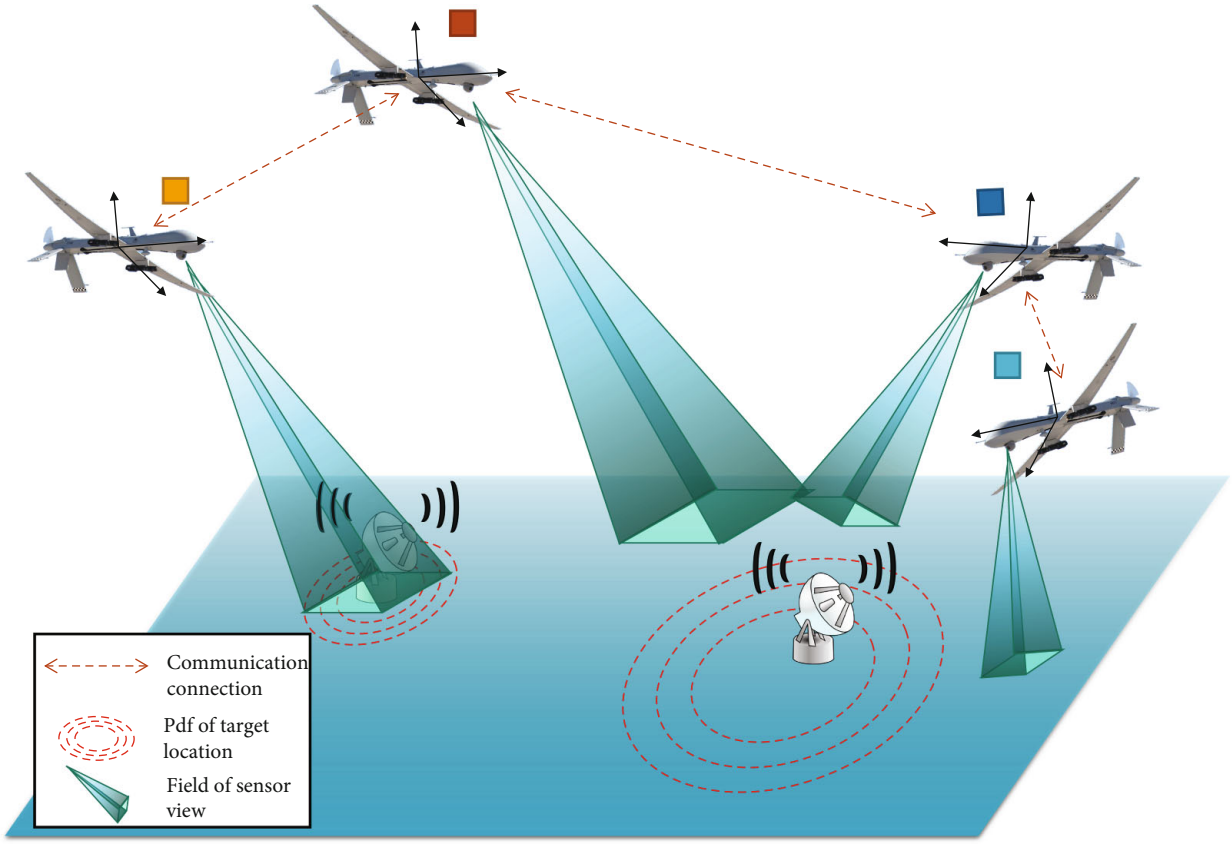


Figure 1: Illustration of a swarm of UAVs executing strike coordination and reconnaissance missions.

UAV in the swarm, and the regulation that constrains UAV flight in urban areas. Therefore, simulation-based validation and testing can significantly facilitate the study of multiple UAV swarm planning and control. Vásárhelyi et al. [3] optimised the parameters of the proposed decentralised guiding algorithm, which enabled large swarms of autonomous drones to navigate in confined spaces seamlessly. The optimised algorithm was tested in real-world applications using a swarm of 30 drones, and the obtained test results coincided well with the expectations. Researchers in the ETH-MAV team in the 2017 Mohamed Bin Zayed International Robotics Challenge tested their swarming algorithm for competition [4] before a real flight using Gazebo [5] as the simulation environment. In most studies, the proposed algorithms have been verified only by numerical simulation. Researchers from the US Air Force Research Laboratory [6] proposed a multiple autonomous vehicle visiting routine planning (VRP) algorithm and validated it by numerical simulations. A framework for generating feasible trajectories in motion coordination problems proposed in [7] was also validated by simulations. Shao et al. [8, 9] developed a state estimator-based minimal learning parameter (SE-MLP) observer using simulation verification to deal with uncertainties appearing in individual quadrotors of the swarm.

Currently, there are many commercial and open-source simulation platforms for multi-UAV planning and control. The MultiUAV2 [10] released by the US Air Force Research Laboratory, which is part of the MATLAB and Simulink software, represents a simulation platform capable of simulating multiple UAVs that cooperate to accomplish tactical missions. However, this platform may cause a copyright problem to the research group without a commercial license. The CoUAV [11] enables rapid implementation of simulation for multiple UAVs, and its source code and demo are available for public access, but in this platform, a member in a UAV swarm is controlled by a ground control centres, which forms a centralised rather than a distributed scheme. The "Infoplanner" simulation platform that was open-sourced by Schlotfeldt et al. [12] is capable of simulating decentralised cooperative planning using an imperfect sensor. However, due to its specific design and lack of documentation, extensions to the system, such as an extension of a sensor model or communication setting, are challenging. An expandable, maintainable, and commercial license-free simulation platform is unavailable nowadays.

To address the mentioned challenges, this paper proposes a simulation platform that provides flexibility in defining varieties of motion dynamics, sensor models, and communication settings; convenience in integration different objectivities with planning algorithm; out-of-the-box trajectory plotting and multiround Monte Carlo simulation; and no requirement of commercial license. The proposed simulation platform is developed with software engineering concerns, such as maintainability and expansibility, and is applied in a military operation study [13, 14].

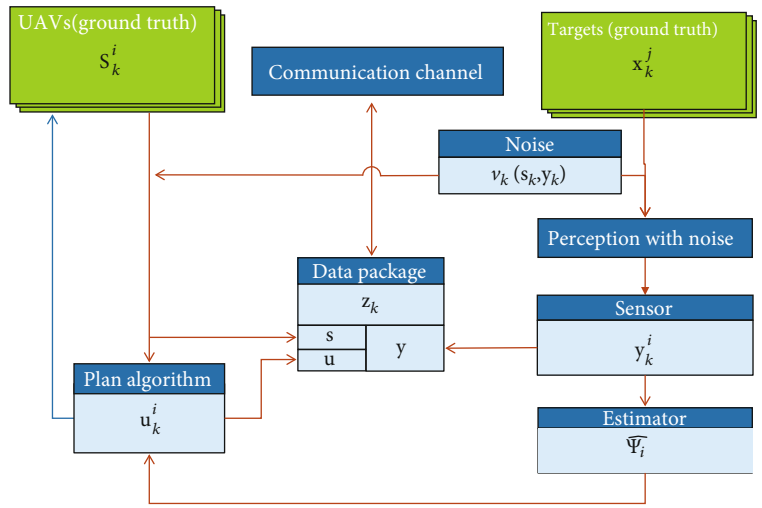


Figure 2: Overall structure of the problem formulation.

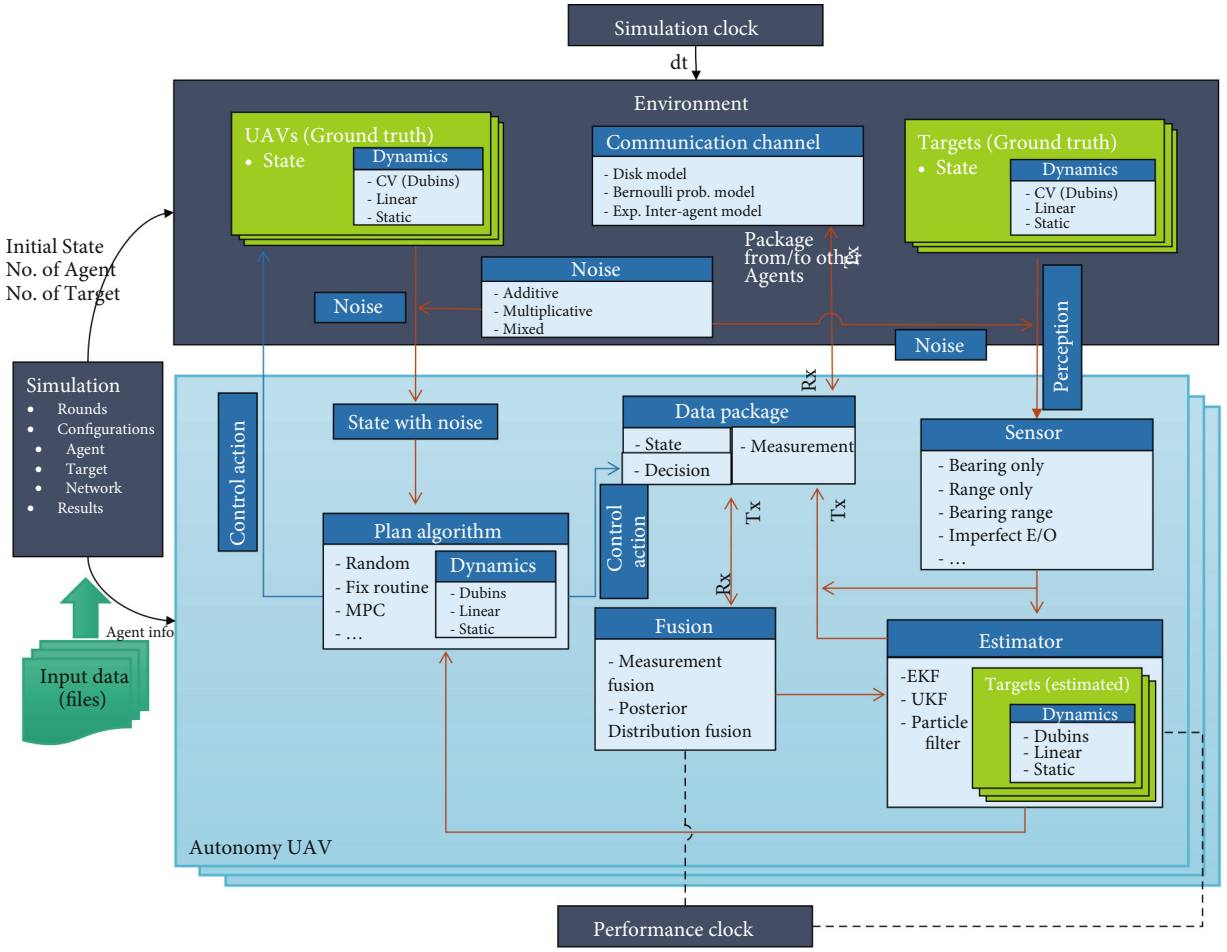


Figure 3: The structure of the proposed domain model.

Table 1: Simulation parameters.

Parameter	Value	Parameter	Value
Area of interest	100 km × 100 km	Target start position	(200, 500)
Unit length	100 m	Unit logical step	1 min
True target motion model $\mathbb{P}_t$	$\mathbb{P}_t(\mathbf{x}_{k+1} \mid \mathbf{x}_k) = f(\boldsymbol{x}_k) + \text{noise}_1$ $\text{noise}_0 \sim N(0, D)$	Determined part of target state transition $f(x_k)$	$\begin{bmatrix} x_k^E + 6\cos 9^\circ \\ x_k^N + 6\sin 9^\circ \\ \varphi_k + 9^\circ \\ \alpha \end{bmatrix}$
D in noise_0	$\begin{bmatrix} 3 & & & \\ & 3 & & \\ & & 1 & \\ & & & 0 \end{bmatrix}$	Planning target motion model $\mathbb{P}_t'$	$\mathbb{P}_t'(\mathbf{x}_{k+1} \mid \mathbf{x}_k) = \boldsymbol{x}_k + \text{noise}_1$ $\text{noise}_1 \sim N(0, 15)$
UAV start position	(-170, -400) (-300, -550) (-430, -4.00) (-300, -250)	UAV start heading	$\varphi_0 = 0$
UAV linear speed	$v = 90$ km/h	UAV start energy amount (fuel)	$\gamma_0 = 1500$ ml
Constant fuel consumption	$c = 3$ ml	Control space (turning option)	$u = \{-15^\circ, 0^\circ, 15^\circ\}$
Turning fuel consumption	$d = 2$ ml	Number of samples for planning	$n_s = 200$
Number of particles	$n_p = 200$	Sensor parameter φ_{fov}	$\varphi_{fov} = 30^\circ$
Sensor parameter μ	$\mu = 0.8$	Sensor parameter ξ	$\xi = 0.25$
Sensor parameter α	$\alpha \sim \text{Rayleigh}(0.96)$	Sensor parameter $g_G(\psi)$	$g_G(\psi) \sim N(0, 2)$
Sensor parameter $g_t(\psi)$	$g_t(\psi) \sim t(1)$	Objective weight $\eta_k(\gamma)$	$3.5 \times \gamma$

The remainder of this paper is organised as follows. Section 2 introduces the system modelling. Section 3 presents simulation cases of cooperative locating radio emitter with an airborne radio receiver to demonstrate the usefulness of the proposed platform. Lastly, Section 4 concludes the paper and presents future work directions.

2. System Modelling

2.1. Overview. In this section, the swarm planning and perception procedure is modelled in a general pattern, as shown in Figure 2. Also, the system design and architecture of the simulation platform which are aimed at supporting rapid and robust development and implementation are introduced, and they are shown in Figure 3. To apply the simulation platform to different scenarios, UAV platforms, perception apparatus, and target motion characteristics, the system modelling adopts a highly abstract way which highlights the interaction among the UAVs, targets, perception, and communication. Every part of the modelling can be further replaced with a specific detailed model. The replacement is also known as concrete implementation of an abstract representation in software engineering.

2.2. Problem Formulation. The problem is formulated in a lockstep discrete way wherein the states of each module are computed synchronically. This design enabled that the simulation can run faster or slower than real time and pause at any time. Consider a swarm consisting of n_a UAVs with a discrete motion dynamic. Then, it can be written that

$$\mathbf{s}_{k+1}^i = f^i\left(\mathbf{s}_k^i, \mathbf{u}_k^i\right), \quad i \in A = \{1, \cdots, n_a\}, \tag{1}$$

where vector $\mathbf{s}_k^i \in \mathcal{S}^i \cong \mathbb{R}^{n_s}$ represents the n_s dimensional state of UAV i at time k and vector $\mathbf{u}_k^i \in \mathcal{U}^i$ is the control action applied to the UAV i at time k, which is one of all possible decisions $\mathcal{U}^i$.

Further, consider n_t targets with Markovian state transition so that

$$\mathbf{x}_{k+1}^j \sim \mathbb{P}_t\left(\mathbf{X}_{k+1}^j | \mathbf{X}_k^j\right), \quad j \in T = \{1, \cdots, n_t\}, \tag{2}$$

where $\mathbf{x}_k^j$ is the state vector of a target j at time k, $\mathbf{X}_k^j$ is the set of all possible states of the target j, and $\mathbb{P}_t(\cdot)$ is the transition probability of the target j. It should be noted that the semi-Markovian system can be converted into the Markovian system by extending the state. According to real-world application experience, it is assumed that a target has no awareness of the existence of the swarm and that its own underlay state transition model is unavailable for the swarm. Considering the Bayesian interference, a swarm model target state transition can be expressed as $\mathbf{x}_{k+1}^j \sim \mathbb{P}_t'(\mathbf{X}_{k+1}^j, | , \mathbf{X}_k^j), j \in T = \{1, \cdots, n_t\}$, where $\mathbb{P}'_t$ is derived from the swarm's prior knowledge, and in the most accurate case, it holds that that $\mathbb{P}'_t = \mathbb{P}_t$.

In a realistic application, perceptions of a UAV are imperfect. The perception result obtained based on the states

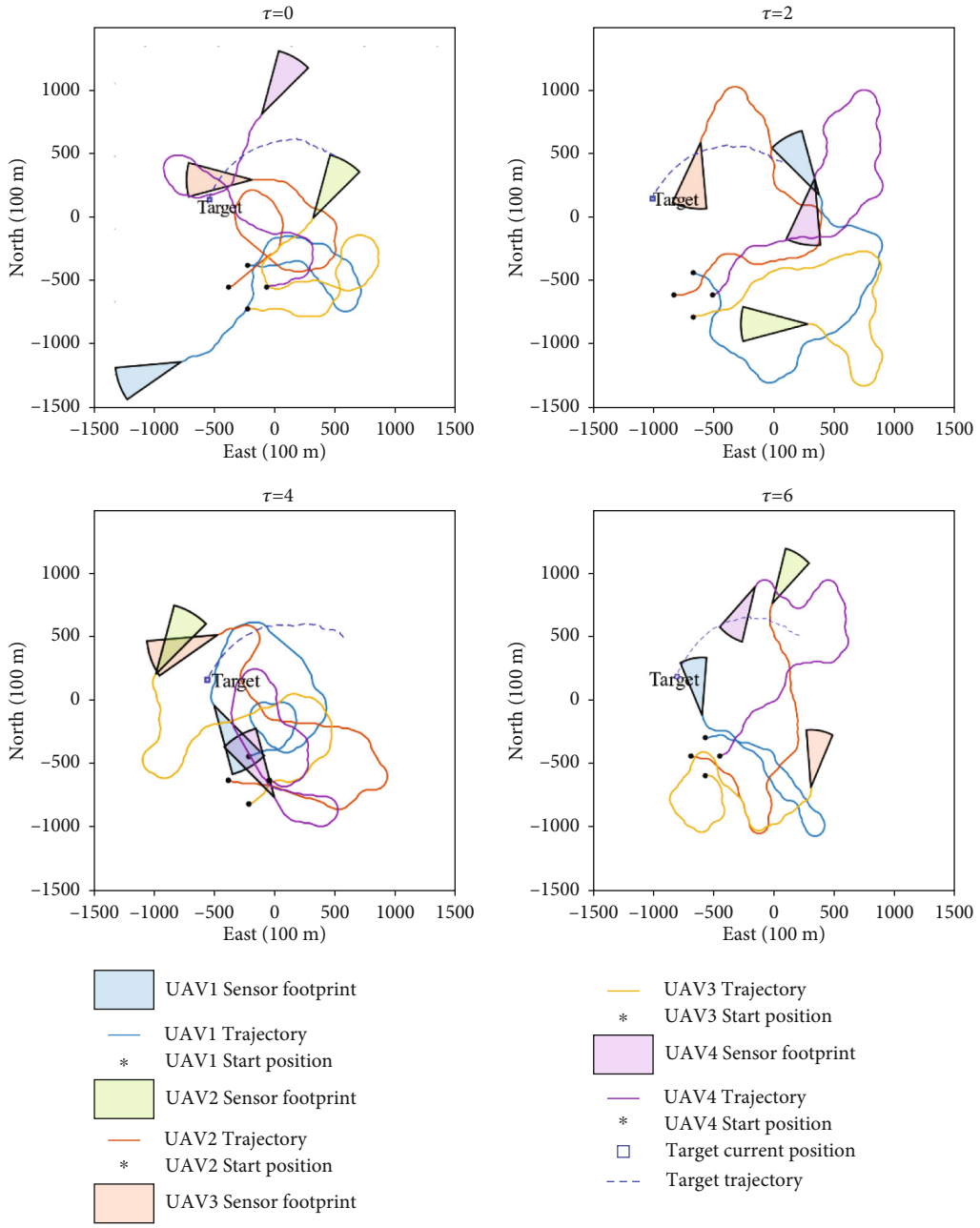


FIGURE 4: The simulated trajectory. In all four cases, the initial conditions except for the receding horizon step τ were the same. In the upper-left figure, $\tau = 0$ stands for greedy exploit.

of UAVs i and target j and environmental noise can be modelled as follows:

$$ {}^{i}\mathbf{y}_k^j = h\left(\mathbf{s}_k^i, \mathbf{x}_k^j, \mathbf{v}_k\left(\mathbf{s}_k^i, \mathbf{x}_k^j\right)\right), \tag{3} $$

where ${}^{i}\mathbf{y}_k^j \in \mathbb{R}^{n_y}$ denotes the perception result of a target j observed by a UAV i at time k, v_k is the noise dependent on the state and environment, and $h(\cdot)$ models the properties of perception.

This work considers an imperfect interswarm communication network, where every two members $l, m \in A$, can share their states, perceptions, and control decisions if needed. Due to the environment interference and packet loss, the communication between UAV can be modelled as a probabilistic model of the received packet. Assume ${}_m^l\mathbf{z}_k$ is the received packet sent by UAV m to UAV l at time k, $\mathbf{Y}_k^m$ is the UAV m's perception result of all targets and ${}_m^l\mathbf{Z}_k$ is the set of all cases of ${}_m^l\mathbf{z}_k$. In addition, denote the probability function of communication distribution as $\mathbb{P}_c(\cdot)$. Then,

$$ {}_m^l\mathbf{z}_k \sim \mathbb{P}_c\left({}_m^l\mathbf{Z}_k | \mathbf{Y}_k^m; \mathbf{s}_k^l, \mathbf{s}_k^m, \mathbf{u}_k^l, \mathbf{u}_k^m\right), \quad l, m \in T = \{1, \cdots, n_a\}. \tag{4} $$

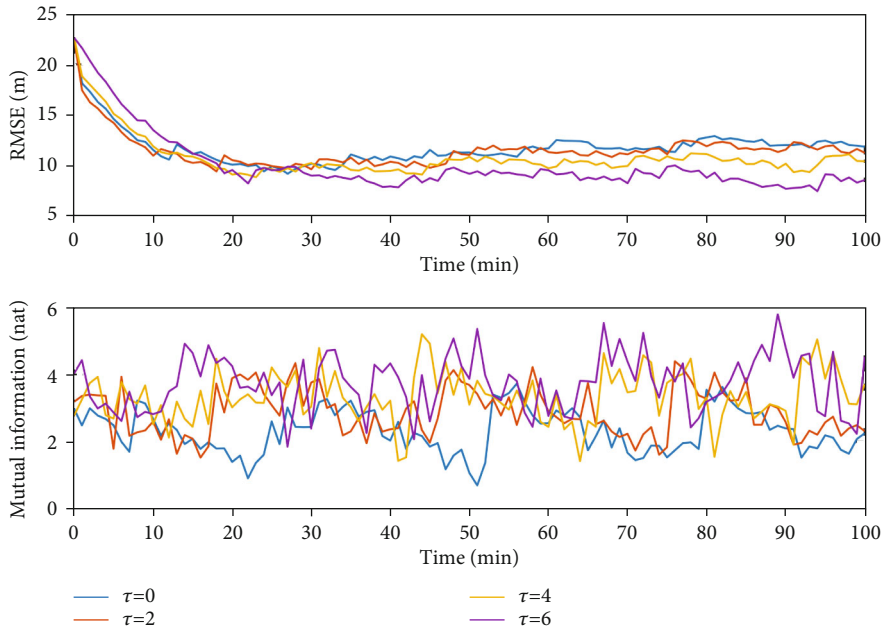


Figure 5: Average RMSE and mutual information of 100 simulation runs.

Table 2: Mean and standard deviation of RMSE at the end of simulation and mutual information during the simulation from 100 simulation runs.

Scenario	Mean and standard deviation of RMSE at the end of simulation and mutual information during the simulation			
	RMSE mean (m)	RMSE Stdev	Mutual information mean (nat)	Mutual information Stdev
$\tau = 0$	11.871	0.3922	2.322	0.4538
$\tau = 2$	11.200	0.3948	2.893	0.4209
$\tau = 4$	10.360	0.3454	3.348	0.4702
$\tau = 6$	8.505	0.3512	3.746	0.4317

In this work, the decision process on cooperative planning and perception is considered as a distributed optimal control problem with the above constraints. Assume that a quantity of interest Ψ represents the yields of the behaviours of UAV members; then, it can be written that

$$U^* = \underset{U}{\operatorname{argmax}} \Psi(\mathbf{s}; \mathbf{x}; \mathbf{y}; \mathbf{z}),$$
$$\text{s.t.} \begin{cases} (1), (2), (3), (4), \\ \mathbf{u}^i \in \mathcal{U}^i, \end{cases} \tag{5}$$

where U^* denotes the optimal control decision of all UAV members in a swarm, U is the set of all control decisions available for UAVs, and $(\mathbf{s}; \mathbf{x}; \mathbf{y}; \mathbf{z})$ is a set consisting of the UAV state, target state, perception result, and communication packet. The underlying form of Ψ is determined by the mission of a swarm. For instance, in the motion planning problem, Ψ can be defined as a negative integration of the throttle along the path under the preference of saving energy. A distributed control decision is made individually by a swarm member aiming to maximise the union quantity of yields of the entire team. The decision can be made by one of the three possible approaches of the control theory: closed-loop control (feedback rule), open-loop control (planning once), and open-loop feedback control (planning and replanning along the receding horizon). Due to the distributed scheme of the problem, every UAV makes a decision based on only its own information and received packet. Let $\widehat{\Psi}_i$ be the local estimation of the union yield of a swam by UAV i. The problem can be rewritten as

$$\mathbf{u}_i^* = \underset{\boldsymbol{u}_i}{\operatorname{argmax}} \widehat{\Psi}_i(\mathbf{s}_i; \mathbf{y}_i; \mathbf{z}_i),$$
$$\text{s.t.} \begin{cases} (1), (2), (3), (4), \\ u^i \in U^i, \\ i \in \{1, \cdots, n_a\}, \end{cases} \tag{6}$$

where $(\mathbf{s}_i; \mathbf{y}_i; \mathbf{z}_i)$ is the set consisting of the ith UAV's state, perception results of UAV i, and all communication packets received by UAV i.

2.3. Platform Implementation. During the implementation of the simulation platform, issues concerning maintainability and functional encapsulation were addressed. Since the simulation platform development is a complex task for a small research group, the main purpose of the system design process is to manage complexity and reuse codes while ensuring that the previously formulated problem can be implemented [15].

The Python programming language is chosen as the primary programming language because its popularity ensures

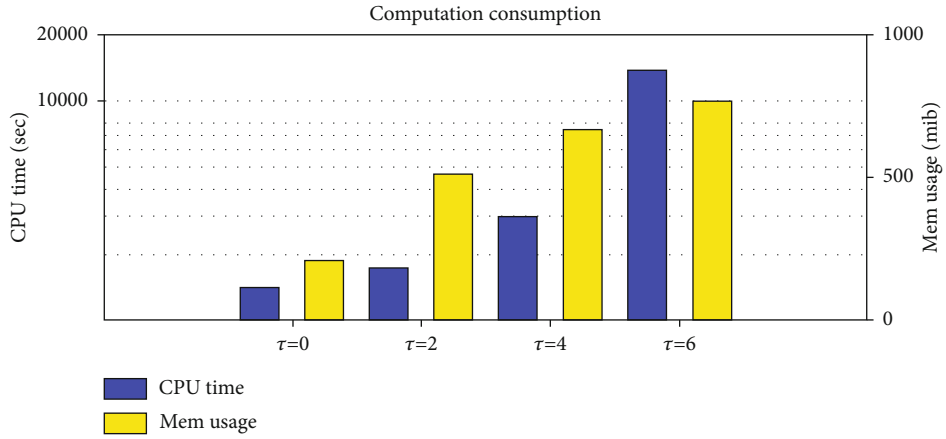


Figure 6: Computation cost of 100 simulation runs. The CPU time (blue) is presented in the logarithmic scale, and memory usage (yellow) is displayed in the linear scale.

the availability to developers in small research groups, development productivity, an abundance of relevant packages, and ampleness of reference cases. Namely, Python is a developer-friendly language, which is easy to learn and code even for a beginner, and a novice researcher can learn Python basics in a relatively short time. In contrast, the other alternative high-level languages, such as C++ and Java, and emerging programming languages, such as Go, requires much more time to master, so a small research group may lack a developer with enough experience in programming in these languages. In addition, unlike MATLAB that requires a paid license, Python has a prosperous open-source community providing many free-of-charge implementations of an interpreter. The features of Python language, including dynamic typing, being object-oriented, and coroutine supporting, which are included in all Python versions since version 3.7, can greatly improve development productivity. Being well-known, one of the main Python tendencies is to writes less code with the help of concise and expressive grammar. The packages managed in PyPI or other source code hosting sites such as GitHub also promote Python, thus increasing its popularity. The SciPy adds efficient matrix and statistics manipulation function to Python, while the SimPy makes building a discrete-time simulation much easier for Python using coroutine.

The key to managing complexity is constructing a domain model representing the essentials of a problem to be addressed [16]. As the distributed cooperative planning and perception problem has been formulated before, the next step is to refine concepts from the problem formulation and to design the information flow between components. The overall structure of a domain model is displayed in Figure 3. The presented structure uses ubiquitous language during the implementation of the proposed platform and bridges the gap between the formula and code.

A class, which originates from object-oriented programming (OOP), helps a developer to integrate data with function, enabling structure reuse and keeping the interface intact while preserving the ability to modify details and implementations. Being well-known, the scattered duplicated codes can significantly jeopardise the maintainability of software and lead to certain defects; in particular, codes cannot be reused directly before certain modifications. To prevent the deterioration of code in advance, the structure presented in Figure 3 is encapsulated in a class in the proposed simulation platform. It should be noted that the object-oriented syntax of Python is concise and expressive.

The proposed platform consists of two major top-level components, *Environments* and *Autonomy UAV*. The *Environments* component enacts the pivotal role as a simulator of the real world and preserves the whole ground truth data in its subcomponents. Being isolated from the ground truth data in the *Environment*, *Autonomy UAV* has multiple instances enacting each UAV in a swarm and retrieving noise-contaminated data generated by the *Noise* component and ground truth data from *Environment*. Subcomponents of *Environment* are *UAVs*, *Targets*, *Communication Channel*, and *Noise*. Information exchange between UAVs is facilitated by introduction of the *Communication Channel* component, which judges the availability of packets according to the communication model. The subcomponents of *Environment*, *UAVs*, and *Targets* compute the state evolution using the dynamics assigned by *Simulation*. Note that in the proposed solution, the dynamic can be different from that in *Plan Algorithm* in that a simplified proxy model can be used in planning. *Sensor* component in *Autonomy UAV* defines the format of perception data and provides a likelihood model for the *Fusion* component and *Plan Algorithm* component. Combining the perception data and *Fusion* model, the *Estimator* component estimates the target state using the target model assigned to *Autonomy UAV* component, which is also not necessarily the same as that in *Targets*. The *Plan Algorithm* provides the control based on the UAV state, estimation result, and packed data obtained from *Communication Channel* component. The control action, UAV state, and estimation are gathered as packed data and sent to the other *Autonomy UAV* components, as well as to the *Environment* component, to change the UAV state.

The remaining three components of the top level are *Simulation*, *Simulation Clock*, and *Performance Clock*.

Simulation loads parameters in *config* files, calls functions to initialize the simulation, and saves simulation data for plotting. *Simulation Clock* maintains a discrete event priority queue and drives the simulation forward. *Performance Clock* records the computation resource usage data when a feature is enabled. This component is implemented in the aspect-oriented scheme using "Wrapper" (a.k.a. decorator), a tool in Python, which allows performance monitoring without modification in a simulation code.

3. Cooperative Locating Radio Emitter with Airborne Radio Receiver

This section introduces the simulation platform configured to model a simulation case of real-world applications, where a swarm of UAVs equipped with an airborne radio receiver attempts to locate a moving ground target emitting radio signals with limited airborne fuel. The main properties of this task are as follows. First, and most importantly, the swarm is completely autonomous and distributed, so trajectories of UAVs in the swarm are planned online according to the distributed cooperative decision of the swarm rather than predefined by operators. Second, due to the nonlinearity and presence of non-Gaussian noise in sensor perception, the posterior distribution is highly nonunimodal, and the standard deviation criteria have a limited guiding significance in the planning of swarm trajectories. Thus, a quantity of interest must be designed to indicate the location of a moving target, as well as energy conservation. Lastly, due to the manoeuvrability constraint of a fixed-wing UAV, the planning of swam trajectories must consider future returns in advance.

3.1. Energy-Aware Motion Model of UAV in Swarm. It is assumed that during the locating process, swarm members are flying at fixed speed and height [17] and action candidates are at finite discrete yaw angle, which is expressed as

$$\mathbf{s}_{k+1} = \left[s^E_{k+1}\ s^N_{k+1}\ \varphi_{k+1}\ \gamma_{k+1}\right]^T = \begin{bmatrix} s^E_k + v\cos(\varphi_{k+1})\Delta t \\ s^N_k + v\sin(\varphi_{k+1})\Delta t \\ \varphi_k \\ \gamma_k - c\cdot\Delta t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \Delta t \\ d\cdot\Delta t \end{bmatrix}\mathbf{u}_k, \tag{7}$$

where s^E and s^N denote a swarm member's coordinates in a north-east coordinate plane, which is defined by the air zone; v is a fixed velocity; φ is the yaw angle; $\gamma \geq 0$ is a real number representing the remaining energy or power of a UAV (once $\gamma = 0$, a UAV fails); $c > 0$ is a fix energy consumption ratio for flying; and, lastly, $d = \eta(\mathbf{u})$ denotes the steering cost.

3.2. Radio Emitter Target Model. In the simulation scenario, two different models of a radio emitter are used in the *Environment* and *Planning* components of the platform for computing the ground truth and the planning of UAV, respectively. The state vector of a moving target can be expressed as $\mathbf{x} = [x^E \quad x^N \quad \alpha]^T$, where x^E, x^N are the target coordinates in the north-east coordinate plane, the same as that of swarm members, and $\alpha > 0$ is an unknown constant relating to the transmit power of the radio emission. It should be noted that the swarm has no prior knowledge about $\mathbb{P}_t$.

3.3. Sensor Model of Airborne Radio Receiver. The airborne radio receiver is assumed to precept the range and bearing of a radio emitter based on the emission signal's strength and direction-of-arrival. The perceptions are polluted by noise, and in accordance with reality, the bearing perception is polluted by non-Gaussian noise.

The strength perception of radio emission can be modelled using the relative emission electronic power as follows:

$$\text{rss} = \mu\log_{10}\left(\left(s^E - x^E\right)^2 + \left(s^N - x^N\right)^2\right) - \alpha, \tag{8}$$

where $\mu > 0$ is a constant related to the performance of a signal receiver and α is a Rayleigh distribution noise introduced by the environmental and thermal noise in the radio receiver [18]. It should be noted that radio emission is undetectable when $\text{rss} \leq 0$, representing the fact that the sensitivity of a radio receiver is limited.

The bearing perception is modelled as follows:

$$\text{bear} = \tan^{-1}\left(\frac{x^N - s^N}{x^E - s^E}\right) + \omega, \tag{9}$$

where w denotes the non-Gaussian noise introduced by a random distortion of the radio waveform. The random variable ω is generated from a weighted mixture of Gaussian distribution and a "long-trailing" noise distribution (Student t distribution) [19], which can be expressed as

$$\omega = \xi g_t(\psi) + (1-\xi)g_G(\psi), \tag{10}$$

where g_t is the probability density function of the t distribution, g_G is the probability density function of the Gaussian distribution, and $\xi \in [0, , ,1]$ is the weight representing the non-Gaussian degree. It should be noted that, in real-world scenarios, radio emission can be detected only in a limited range of direction-of-arrival due to the structural characteristics of a radio receiver antenna.

Thus, the sensor model of an airborne radio receiver can be defined as

$$\mathbf{y}_k = \begin{cases} \begin{bmatrix} \text{rss} \\ \text{bear} \end{bmatrix} & \text{if } \text{rss} \geq 0, \quad |\text{bear} - \varphi_k| \leq \varphi_{fov}, \\ \varnothing, & \text{else}, \end{cases} \tag{11}$$

where φ_{fov} is a constant representing the maximum angle of the field of view with respect to the yaw of the UAV.

3.4. Channel Model of Interswarm Communication. The detection, a.k.a. perception results, are sent with the UAV states $\mathbf{s}$ to other UAV members via a packet loss channel. The packet loss probability is directly proportional to the distance between the two communicating UAVs. The two outcomes, *succeed to reach* and *fail to reach*, follow the

Bernoulli distribution, whose parameters can be obtained from empirical data [20]. Assume ${}^{l}_{m}\beta_k(\mathbf{s}^l_k, \mathbf{s}^m_k)$, where m represents a UAV and l represents the ground vehicle, and ${}^{l}_{m}\beta_k(\mathbf{s}^l_k, \mathbf{s}^m_k)$ denotes the probability distribution of the Bernoulli distribution at time k; then, it can be expressed as

$$ {}^{l}_{m}\beta_k\left(\mathbf{s}^l_k, \mathbf{s}^m_k\right) = 0.5 \times \operatorname{erfc}\left(\frac{\sqrt{(\Delta s^E)^2 + (\Delta s^N)^2} - \alpha_0}{\alpha_1}\right) - \alpha_2, \tag{12} $$

where α_i represents constant experimental parameter.

Then, (9) can be written as

$$ \mathbb{P}_c\left({}^{l}_{m}\mathbf{Z}_k | \mathbf{s}^l_k, \mathbf{s}^m_k\right) = \begin{cases} {}^{l}_{m}\beta_k\left(\mathbf{s}^l_k, \mathbf{s}^m_k\right), & \text{if } {}^{l}_{m}\mathbf{Z}_k = \{\mathbf{y}^m_k, \mathbf{s}^m_k, \mathbf{u}^m_k\}, \\ 1 - {}^{l}_{m}\beta_k\left(\mathbf{s}^l_k, \mathbf{s}^m_k\right), & \text{if } {}^{l}_{m}\mathbf{Z}_k = \varnothing. \end{cases} \tag{13} $$

3.5. Bayesian Inference-Based Site Locating. There are many uncertainty sources along with the perception information flow from a UAV to the ground vehicle, including noise, fault detection, and communication loss. To estimate the location of a lost pilot, as well as the estimation accuracy, a distributed iterative particle filter is utilised. Assuming there is poor prior information about the location of the lost pilot, the prior distribution $P(s_t)$ is chosen as a uniform distribution on the map. Particle sets with weights $\langle \widehat{s^k_t}, w^k \rangle$ are sampled from the prior distribution. The particle set evolves with each perception result or packet reception by the forward state prediction (13) and weight update, which can be expressed as

$$ w^{k+1} \propto w^k P(Z_k \mid s_t). \tag{14} $$

In order to mitigate the particle degeneracy problem, a standard importance resampling step is introduced after the particles' weights are updated.

3.6. Quantity of Interest for Cooperative Decision. As a characterization of uncertainty, the mutual information of the radio emitter state and accumulated perceptions is chosen as one of the two quantities of interest used to evaluate perception quantity. Due to the imperfectness and nonlinearity of UAV perceptions, the radio emitter location acquired by the Bayesian inference is always a nonunimodal probability distribution. In such a situation, other measures like covariance are not appropriate [21]. Another quantity of interest is the sum of distances between the target and UAVs with a negative coefficient related to energy γ. Assume $\mathrm{dis}_i = \sum_i^{n_a} \sqrt[2]{(\mathbf{s}^N_i - \mathbf{x}^N)^2 + (\mathbf{s}^E_i - \mathbf{x}^E)^2}$ represents the sum of distances between the UAVs and the target; then, the objective Ψ in (10) can be expressed as

$$ \Psi = \sum_{\mathrm{i}}^{n_a} [I(\mathbf{Y}, \mathscr{Z}) + \eta(\gamma_i) \cdot \mathrm{dis}_i], \tag{15} $$

where $I(\cdot)$ denotes the mutual information that can be derived from [22]; $\mathscr{Z}$ is the union set of perceptions of a UAV member perceived by itself and received from other members, which can be further decomposed as $\mathscr{Z} = \{{}_1\mathbf{Z}, {}_2\mathbf{Z}, \cdots, {}_{n_a}\mathbf{Z}\}$; and $\eta(\gamma) < 0$ is the coefficient that characterizes the tendency to approach the target with respect to the remaining energy.

3.7. Distributed Cooperative Planning. It is hard to obtain a solution to (15) in the closed form. First, except for a special form of a dynamic system, it is hard to obtain an optimal feedback law of U^* as a closed form even in a centralised scheme [23]. In this study, the global optimal control law is approximated with an open-loop independent decision series in a time receding horizon τ, which is resolved in each time step. Assume $\bar{\Psi}_k = I_k(\mathbf{Y}, \mathscr{Z}) + \eta_k(\gamma_i) \cdot \mathrm{dis}_k$ represents the quantity of interest at time k; then, Ψ in (10) can be approximated by

$$ \Psi \approx \sum_{k}^{k+\tau} \bar{\Psi}_k, \tag{16} $$

where $\mathbf{u}^i_{k+1}, \mathbf{u}^i_{k+2}, \cdots, \mathbf{u}^i_{k+\tau}$ are the independent decision series of a UAV i. A detailed explanation of coordinate descent was given in [12], and the receding horizon control in cooperative perception was presented in detail in [24].

The last problem to solve is the estimation of the quantity of interest in the receding horizon $k+1, k+2, \cdots, k+\tau$. In this work, the sampling-based method [25] is used. Assume that samples of the target state are $\hat{\mathbf{x}}^{(q)}_k \sim \hat{p}(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ and $\hat{\mathbf{y}}^{(q)}_k \sim \hat{p}(\mathbf{y}_k \mid \mathbf{x}_k, \mathbf{u}_k)$, where $\hat{p}(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ and $\hat{p}(\mathbf{y}_k \mid \mathbf{x}_k, \mathbf{u}_k)$ are derived from the weighted particle set generated by the Bayesian inference; then, an approximation of $\bar{\Psi}_{\mathrm{k}}$ is expressed as

$$ \bar{\Psi}_{\mathrm{k}} \approx \tilde{\Psi}_k = \frac{1}{N} \sum_{m}^{N} \Psi\left(\hat{\mathbf{x}}^{(q)}_k, \hat{\mathbf{y}}^{(q)}_k, \mathbf{u}_k\right). \tag{17} $$

Inspired by the motion primitive method, a motion primitive graph can be constructed by taking s_k as the root node, selecting or sampling $\mathbf{u}^i_k$ from the available control action space $\mathscr{U}^i$ as the edge, connecting edge $\mathbf{u}^i_k$ to the candidate node $\hat{s}^i_{k+1}$, and calculating the quantity of interest $\tilde{\Psi}_k$ in each candidate node $\hat{s}^i_{k+1}$ as the weight of the edge. Because the decision space is discrete and $\tilde{\Psi}_k$ can be enumerating and evaluated, the scale of the motion primitive graph is limited. The optimal control action sequence can be obtained by a search algorithm like A$*$.

3.8. Simulation Results. The simulation is conducted in a square area of interest of 10,000 km^2. The UAV swarm is formed by four small fixed winged UAVs with a two-cylinder propeller. The target is moving in a constant linear velocity of 18 km/h and a turning rate of 9 degrees/min. The target's motion is also disturbed in speed and turning rate (see noise_0 in Table 1). The UAVs have no prior

knowledge of the target's motion and assume the target is randomly walking (see planning target motion model $\mathbb{P}_t'$ in Table 1). The parameters of the airborne sensor for perception and other parameters for simulation initialization are given in Table 1. To eliminate the randomness and explore the effect of τ, four simulation scenarios with different τ were designed, and simulations were conducted 100 times for each scenario.

The trajectory snaps of one of the simulations four different τ values are presented in Figure 4. Due to the uncertainty property, the trajectories of both the target and the swarm are different in each run. Thus, the overall estimation performances, including accuracy and credibility, are depicted in an average performance of 100 simulation runs (see Figure 5 and Table 2). The accuracy was measured by the average root mean square error relating to the target true position (the lower the error was, the better the performance was), which was calculated by $\mathrm{RMSE}_k = \sqrt{(\bar{x}_k^E - x_k^E)^2 + (\bar{x}_k^N - x^N)^2}$, where $\bar{x}_k^E, \bar{x}_k^N$ are the maximum a posteriori estimation values. The credibility was measured by the average mutual information derived from the particle set of the swarm member; the higher the amount of the average mutual information was, the better the credibility was.

The results in Figure 5 and Table 2 show that the higher τ resulted in higher mutual information and lower RMSE. This indicates that the increase in the time horizon τ can help to improve the performance of the swarm in the way of earlier turning for gaining more quantity of interest and avoiding extra turns.

The computation resource consumption of the simulation is plotted in Figure 6. Based on the results in Figure 6, the improvements in accuracy and credibility were obtained at the cost of an exponential increase in the computation cost. In addition, under the exhausting energy (see the end part of Figure 5), the swarm tended more to approaching the target than gathering the perception, which coincides well with the designer expectation.

4. Conclusion

This paper presents an extensible and maintainable simulation platform for distributed swarm cooperative perception planning considering the uncertainty in communication and perception. Simulation cases of evaluating the Bayesian inference-based estimation under imperfect perception and evaluating distributed cooperative planning are considered to demonstrate the operating principle and usefulness of the proposed simulation platform.

Since the modelling and implementation of the simulation consider the real-world constraint, such as unstable communication and noisy perception, the proposed solution can be beneficial to distributed cooperative swarm development and application.

However, the proposed simulation platform can be further improved by implementing additional mode functionalities, such as a new sensor model, communication model in complex terrain, and cooperative planning algorithm based on reinforcement learning, which will be part of our future work.

Data Availability

The simulation condition data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declared that they have no conflicts of interest.

Acknowledgments

This work was supported by the Military Science Research under Grant JY2019B0XX and Frontier Innovation of National Defence under Grant 17-163-11-ZT-XXX-XX.

References

[1] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.

[2] X. Shao, Y. Shi, and W. Zhang, "Fault-tolerant quantized control for flexible air-breathing hypersonic vehicles with appointed-time tracking performances," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 2, 2021.

[3] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, 2018.

[4] R. Bähnemann, M. Pantic, M. Popović et al., "The ETH-MAV team in the MBZ International Robotics Challenge," *Journal of Field Robotics*, vol. 36, no. 1, pp. 78–103, 2019.

[5] *Gazebo*Available from: http://gazebosim.org/.

[6] S. Rasmussen, K. Kalyanam, S. Manyam, D. Casbeer, and C. Olsen, "Practical considerations for implementing an autonomous, persistent, intelligence, surveillance, and reconnaissance system," in *IEEE Conference on Control Technology and Applications (CCTA)*, vol. 2017, pp. 1847–1854, 2017.

[7] H. Jaleel and J. S. Shamma, "Distributed submodular minimization and motion planning over discrete state space," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 932–943, 2020.

[8] X. Shao, Z. Cao, and H. Si, "Neurodynamic formation maneuvering control with modified prescribed performances for networked uncertain quadrotors," *IEEE Systems Journal*, pp. 1–12, 2020.

[9] X. Shao, Y. Shi, W. Zhang, and H. Cao, "Neurodynamic approximation-based quantized control with improved transient performances for MEMS gyroscopes: theory and experimental results," *IEEE Transactions on Industrial Electronics*, p. 1, 2020.

[10] S. Rasmussen, J. W. Mitchell, P. Chandler, C. J. Schumacher, and A. L. Smith, "Introduction to the MultiUAV2 simulation and its application to cooperative control research," in *Proceedings of the 2005, American Control Conference*, vol. 2005, pp. 4490–4501, IEEE, 2005.

[11] W. Wu, Z. Huang, F. Shan et al., "CoUAV: a cooperative UAV fleet control and monitoring platform," 2019, http://arxiv.org/abs/190404046.

[12] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *Ieee Robotics and Automation Letters*, vol. 3, no. 2, pp. 1025–1032, 2018.

[13] H. Luo, H. Ling, T. Zhu, and Z. Kang, "Complete coverage path planning of UUV for marine mine countermeasure using grid division and spanning tree," in *2019 Chinese Control And Decision Conference (CCDC)*, pp. 5016–5021, IEEE, 2019.

[14] H. Ling, H. Luo, L. Bai, T. Zhu, Q. Wang, and L. Yu, "Cooperative loitering munition swarm online patrolling route planning with nonlinear seeker measurement," *Mathematical Problems in Engineering*, vol. 2020, 10 pages, 2020.

[15] S. McConnell, *Code Complete*, Wash, Microsoft Press, Redmond, 2004.

[16] E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*, Addison-Wesley, Boston, 2004.

[17] G. Gramajo and P. Shankar, "An efficient energy constraint based UAV path planning for search and coverage," *International Journal of Aerospace Engineering*, vol. 2017, 13 pages, 2017.

[18] N. H. Nguyen, K. Doğançay, and W. Wang, "Adaptive estimation and sparse sampling for graph signals in alpha-stable noise," *Digital Signal Processing*, vol. 105, p. 102782, 2020.

[19] D. Middleton, "Non-Gaussian noise models in signal processing for telecommunications: new methods an results for class A and class B noise models," *IEEE Trans Inform Theory*, vol. 45, no. 4, pp. 1129–1149, 1999.

[20] M. Stachura and E. Frew, "Communication-aware information-gathering experiments with an unmanned aircraft system," *Journal of Field Robotics*, vol. 34, no. 4, pp. 736–756, 2017.

[21] Y. Boers, H. Driessen, A. Bagchi, and P. Mandal, "Particle filter based entropy," in *2010 13th International Conference on Information Fusion*, pp. 1–8, IEEE, Edinburgh, 2010.

[22] J. Ajgl and M. Šimandl, "Differential entropy estimation by particles," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11991–11996, 2011.

[23] A. Bagchi, *Optimal Control of Stochastic Systems*, Prentice Hall, New York, 1993.

[24] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically optimal planning for non-myopic multirobot information gathering," in *Robotics: Science and Systems XV*, Robotics: Science and Systems Foundation, 2019.

[25] E. H. Aoki, A. Bagchi, P. Mandal, and Y. Boers, *Particle Filter Approximations for General Open Loop and Open Loop Feedback Sensor Management*, Memorandum/Department of Applied Mathematics, 2011.