Research Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
January-February 2018: 1–11

DOI: 10.1177/1729881418755246
journals.sagepub.com/home/arx
SAGE

Neural network–based nonaffine control of air-breathing hypersonic vehicles with prescribed performance

Xiangwei Bu and Qing Wang

Abstract

This article investigates a novel nonaffine control strategy using neural networks for an air-breathing hypersonic vehicle. Actual actuators are regarded as additional state variables and virtual control inputs are derived from low-computational cost neural approximations, while a new altitude control design independent of affine models is addressed for air-breathing hypersonic vehicles. To further reduce the computational load, an advanced regulation algorithm is applied to devise adaptive laws for neural estimations. Moreover, a new prescribed performance mechanism is exploited, which imposes preselected bounds on the transient and steady-state tracking error performance via developing new performance functions, capable of guaranteeing altitude and velocity tracking errors with small overshoots. Unlike some existing neural control methodologies, the proposed prescribed performance-based nonaffine control approach can ensure tracking errors with preselected transient and steady-state performance. Meanwhile, the complex design procedure of backstepping is also avoided. Finally, simulation results are presented to validate the design.

Keywords

Air-breathing hypersonic vehicle, neural network, nonaffine control, prescribed performance

Date received: 1 July 2017; accepted: 18 December 2017

Topic: Robot Manipulation and Control
Topic Editor: Andrey V Savkin
Associate Editor: Alexander Pogromsky

Introduction

Air-breathing hypersonic vehicles (AHVs) have been viewed as a critical solution to achieving reliable affordable access to near-space for both commercial and military applications.[1–3] However, the special peculiarities of AHVs' dynamic characteristics and aerodynamic effects make the control design highly challenging.[4,5] Furthermore, the motion model constructed for AHVs must be highly nonlinear and coupled owing to the airframe-integrated scramjet engines and time-varying flight conditions. Thus, the vehicle model is completely nonaffine in the control inputs. Besides, excellent transient performance and real-time performance are also required for AHVs' control systems because of the hypersonic speed and super maneuver.

Recently, flight control designs for AHVs have been given special considerations, and various control methodologies have been addressed based on simplified affine models of AHVs instead of nonaffine ones. For the longitudinal dynamics of an AHV subject to parametric uncertainties, a robust controller with L_∞ performance is developed utilizing a disturbance observer, which provides robust tracking of velocity and altitude reference trajectories.[6] To cope with conservatism disadvantage derived from robust control, a terminal sliding mode control (SMC)

Air and Missile Defense College, Air Force Engineering University, Xi'an, China

Corresponding author:
Xiangwei Bu, Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China.
Email: buxiangwei1987@126.com

approach is presented for an AHV and furthermore a sliding mode disturbance observer is introduced to enhance the controller's robustness against system uncertainties and external disturbances.[7] Tian et al.[8] proposed a multiple-time scale-based second order SMC scheme for a flexible AHV, while the chattering problem connected with traditional SMC is handled. By combining fractional order proportion integral and derivative with active disturbance rejection control method, a nonlinear flight controller is devised for an AHV to provide accurate and fast tracking of desired attitude signals.[9] In addition, fault-tolerant controllers,[10,11] predictive control law,[12] and constrained control methodologies[13–15] are also widely studied for AHVs.

Under strict assumptions, the altitude dynamics of AHVs can be rewritten as a strict-feedback system of affine formulation. On this basis, several backstepping control methods are investigated for AHVs employing disturbance observer,[16] projection algorithm,[17,18] perturbed system,[19] and neural network.[19–21] It is noted that traditional backstepping control needs a complicated recursive design procedure, yielding a series of virtual control laws and causing a problem of "explosion of terms." In each virtual controller, there is a neural network utilized to approach the subsystem uncertainties. Moreover, to stabilize the closed-loop control system, massive online learning parameters are required for neural approximations. For this reason, the real-time performance of those controllers[19–21] may not meet requirements. Thereby, simplified neural control approaches are studied for AHVs, and online learning parameters are reduced via advanced learning algorithms.[22–24]

Owing to the lack of technical tools, few works that concern transient performance of control systems have been proposed in the above literature. Recently, guaranteeing transient performance–based prescribed performance control methods have been presented for AHVs to provide robust tracking of reference commands with the tracking errors satisfying the predesigned transient and steady-state performance.[25–28] Unfortunately, none of them can guarantee the tracking error with small even zero overshoot. In this article, a novel prescribed performance control approach using nonaffine models is proposed for an AHV based on neural approximation. The special contributions are summarized as follows:

1. Compared with the existing affine control schemes,[4,6,7] the presented nonaffine control strategy exhibits higher fidelity due to the avoidance of model simplifications.
2. Different from traditional neural control strategies,[19,21] the system uncertainties are lumped together and advanced regulation schemes are developed to directly estimate the norm of neural networks, on the basis of which both the required neural networks and online learning parameters are reduced greatly, yielding a low-computational burden design.
3. By comparison with traditional prescribed performance control approaches,[28,29] better transient performance guaranteeing small overshoot can be imposed on tracking errors based on a newly constructed prescribed performance mechanism.

The rest of this article is structured as follows. The vehicle model and preliminaries are shown in the second section. Third section presents the control design process. Simulation results are drawn in the fourth section and the final section draws the conclusions of this article.

AHV model and preliminaries

Model description

The AHV's motion model adopted in this study is expressed as[30]

$$\dot{V} = T\cos(\theta-\gamma)/m - D/m - g\sin\gamma \tag{1}$$

$$\dot{h} = V\sin\gamma \tag{2}$$

$$\dot{\gamma} = L/(mV) + T\sin(\theta-\gamma)/(mV) - g\cos\gamma \tag{3}$$

$$\dot{\theta} = Q \tag{4}$$

$$\dot{Q} = (M + \tilde{\psi}_1\ddot{\eta}_1 + \tilde{\psi}_2\ddot{\eta}_2)/I_{yy} \tag{5}$$

$$k_1\ddot{\eta}_1 = -2\zeta_1\omega_1\dot{\eta}_1 - \omega_1^2\eta_1 + N_1 - \tilde{\psi}_1 M/I_{yy} - \tilde{\psi}_1\tilde{\psi}_2\ddot{\eta}_2/I_{yy} \tag{6}$$

$$k_2\ddot{\eta}_2 = -2\zeta_2\omega_2\dot{\eta}_2 - \omega_2^2\eta_2 + N_2 - \tilde{\psi}_2 M/I_{yy} - \tilde{\psi}_2\tilde{\psi}_1\ddot{\eta}_1/I_{yy} \tag{7}$$

where the five rigid-body states V, h, γ, θ, and Q denote velocity, altitude, flight-path angle, pitch angle, and pitch rate, respectively; the flexible states η_1 and η_2 are the first two generalized elastic coordinates. The thrust force T, the drag force D, the lift force L, the pitching moment M, the first generalized force N_1, and the second generalized force N_2 are given by[30]

$$T \approx \beta_1(h,\bar{q})\Phi\alpha^3 + \beta_2(h,\bar{q})\alpha^3 + \beta_3(h,\bar{q})\Phi\alpha^2 + \beta_4(h,\bar{q})\alpha^2 + \beta_5(h,\bar{q})\Phi\alpha + \beta_6(h,\bar{q})\alpha + \beta_7(h,\bar{q})\Phi + \beta_8(h,\bar{q})$$

$$D \approx \bar{q}SC_D^{\alpha^2}\alpha^2 + \bar{q}SC_D^{\alpha}\alpha + \bar{q}SC_D^{\delta_e^2}\delta_e^2 + \bar{q}SC_D^{\delta_e}\delta_e + \bar{q}SC_D^0$$

$$M \approx z_T T + \bar{q}S\bar{c}C_{M,\alpha}^{\alpha^2}\alpha^2 + \bar{q}S\bar{c}C_{M,\alpha}^{\alpha}\alpha + \bar{q}S\bar{c}C_{M,\alpha}^0 + \bar{q}S\bar{c}c_e\delta_e$$

$$L \approx \bar{q}SC_L^{\alpha}\alpha + \bar{q}SC_L^{\delta_e}\delta_e + \bar{q}SC_L^0, \quad N_1 = N_1^{\alpha^2}\alpha^2 + N_1^{\alpha}\alpha + N_1^0$$

$$N_2 = N_2^{\alpha^2}\alpha^2 + N_2^{\alpha}\alpha + N_2^{\delta_e}\delta_e + N_2^0, \quad \bar{q} = \bar{\rho}V^2/2,$$

$$\bar{\rho} = \bar{\rho}_0\exp\left(-(h-h_0)/h_s\right)$$

where the control inputs Φ and δ_e, occurring implicitly in equations (1) to (7), stand for fuel equivalence ratio and elevator angular deflection, respectively. For more detailed definitions of other variables and coefficients, the reader could refer to Parker et al.[30]

The control objective is to let h and V follow their reference trajectories h_{ref} and V_{ref} in the presence of

parametric uncertainties, while altitude tracking error and velocity tracking error are limited within preselected bounds such that the desired transient performance and steady-state performance are guaranteed.

New prescribed performance mechanism

By prescribed performance, it denotes that the tracking error $e(t)$ is strictly limited within an arbitrarily small residual set. Meantime, the convergence rate isn't less than a given value and the maximum overshoot is less than a chosen constant. The mathematical expression of traditional prescribed performance is formulated as[29]

$$\begin{cases} -\delta\rho(t) < e(t) < \rho(t), \text{if } e(0) > 0 \\ -\rho(t) < e(t) < \delta\rho(t), \text{if } e(0) < 0 \end{cases} \tag{8}$$

where $0 \leq \delta \leq 1$ is a design parameter. The performance function $\rho(t)$ is defined as[29]

$$\rho(t) = (\rho_0 - \rho_\infty)e^{-lt} + \rho_\infty \tag{9}$$

where $\rho_0 > \rho_\infty > 0$, $l > 0$ are design parameters. Moreover, it is noticed that the smooth function $\rho(t)$ is bounded and strictly positive decreasing with the property that $\lim_{t\to\infty} \rho(t) = \rho_\infty$, $\rho(0) = \rho_0$.

The aforementioned prescribe performance is clearly illustrated in Figure 1. It can be seen that there are two serious shortcomings for the traditional prescribe performance (8). The first one is that the behavioral bound (8) on the tracking error $e(t)$ relies on the exact initial value $e(0)$. Based on (8), the subsequent control design and stability analysis have to be implemented based on different initial conditions ($e(0)$ is positive or negative), which is complicated and even unpractical. The second shortage is that we cannot insure the overshoot to be a small value, as shown in Figure 1.

To deal with the aforesaid shortcomings, we exploit a novel formulation of prescribed performance

$$P_l(t) < e(t) < P_r(t) \tag{10}$$

The new performance functions $P_l(t)$ and $P_r(t)$ are defined as

$$\begin{cases} P_l(t) = [\text{sign}(e(0)) - \delta_1]\rho(t) - \rho_\infty \text{sign}(e(0)) \\ P_r(t) = [\text{sign}(e(0)) + \delta_2]\rho(t) - \rho_\infty \text{sign}(e(0)) \end{cases} \tag{11}$$

with $0 \leq \delta_1 \leq 1$, $0 \leq \delta_2 \leq 1$.

The newly defined prescribe performance (10) is exhibited in Figure 2. It is observed from Figure 2, equation (10), and equation (11) that the proposed prescribe performance formulation (10) is more concise than (8). Based on (10), the subsequent control developments and stability proof become much simpler than by employing (8). Furthermore, a small overshoot convergence of $e(t)$ can be achieved by (10) when we select appropriate design parameters for $P_l(t)$ and $P_r(t)$, as depicted in Figure 2.

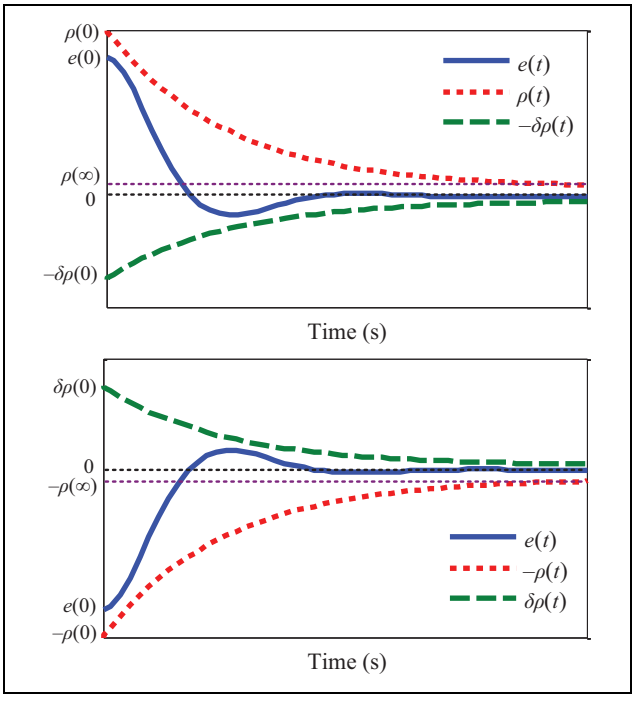

Figure 1. Graphical illustration of the prescribed performance definition (8).

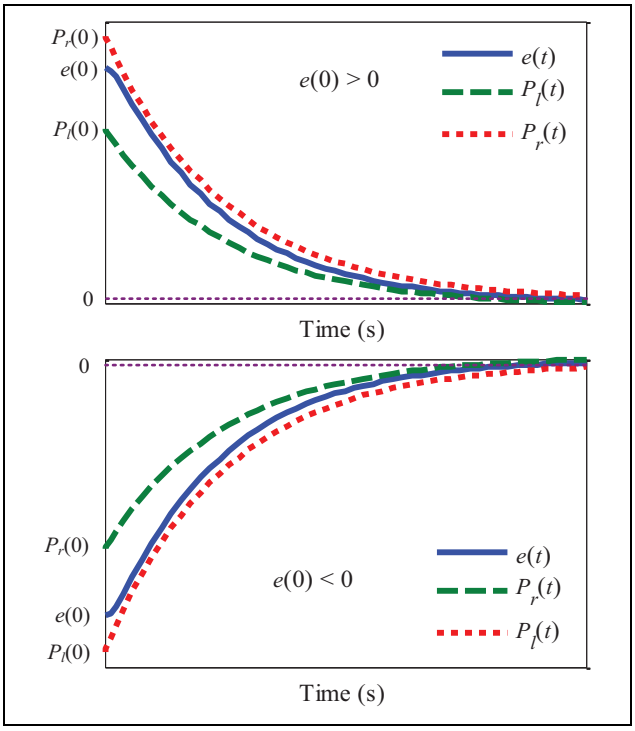

Figure 2. Graphical illustration of the newly defined prescribed performance (10).

Remark 1. For the rare condition of $e(0) = 0$, inequality (10) is also doable. When $e(0) = 0$, inequality (10) becomes $-\delta_1\rho(t) < e(t) < \delta_2\rho(t)$.

It is hard to directly devise controllers using the inequality (10). Thus, we define the following transformed error $\varepsilon(t)$

$$\begin{cases} \varepsilon(t) = \ln\left(\dfrac{\vartheta(t)}{1-\vartheta(t)}\right) \\ \vartheta(t) = \dfrac{e(t)-P_l(t)}{P_r(t)-P_l(t)} \end{cases} \tag{12}$$

Theorem 1. If $\varepsilon(t)$ is bounded, the tracking error $e(t)$ can be limited to the constructed prescribed behavior bound (10).

Proof. The inverse transformation of (12) is given by

$$e^{\varepsilon(t)} = \frac{\vartheta(t)}{1-\vartheta(t)} \tag{13}$$

From (13), we further get

$$\vartheta(t) = \frac{e^{\varepsilon(t)}}{1+e^{\varepsilon(t)}} \tag{14}$$

The boundedness of $\varepsilon(t)$ leads to that there exists a positive constant ε_{M} such that $|\varepsilon(t)| \le \varepsilon_{\mathrm{M}}$. That is, $-\varepsilon_{\mathrm{M}} \le \varepsilon(t) \le \varepsilon_{\mathrm{M}}$. Hence, equation (14) becomes

$$0 < \frac{e^{-\varepsilon_{\mathrm{M}}}}{1+e^{-\varepsilon_{\mathrm{M}}}} \le \vartheta(t) \le \frac{e^{\varepsilon_{\mathrm{M}}}}{1+e^{\varepsilon_{\mathrm{M}}}} < 1 \tag{15}$$

Noting that $\vartheta(t) = [e(t)-P_l(t)]/[P_r(t)-P_l(t)]$, we obtain

$$0 < \frac{e(t)-P_l(t)}{P_r(t)-P_l(t)} < 1 \tag{16}$$

Finally, we have

$$P_l(t) < e(t) < P_r(t) \tag{17}$$

The proof is completed. □

Remark 2. In what follows, the controller will be explored using the transformed error $\varepsilon(t)$ instead of the tracking error $e(t)$.

Controller design

The pursued control objective in this section is to let $h \to h_{\mathrm{ref}}$ by developing a neural prescribed performance controller δ_{e} for altitude subsystems (2) to (5).

The altitude tracking error $\tilde{h}$ and its transformed error $\varepsilon_1(t)$ are defined as

$$\begin{cases} \tilde{h} = h - h_{\mathrm{ref}} \\ \varepsilon_1(t) = \ln\left(\dfrac{\vartheta_1(t)}{1-\vartheta_1(t)}\right) \\ \vartheta_1(t) = \dfrac{\tilde{h}-P_{l1}(t)}{P_{r1}(t)-P_{l1}(t)} \\ P_{l1}(t) = [\,\mathrm{sign}(\tilde{h}(0)) - \delta_{11}]\rho_1(t) - \rho_{1\infty}\,\mathrm{sign}(\tilde{h}(0)) \\ P_{r1}(t) = [\,\mathrm{sign}(\tilde{h}(0)) + \delta_{12}]\rho_1(t) - \rho_{1\infty}\,\mathrm{sign}(\tilde{h}(0)) \\ \rho_1(t) = (\rho_{10}-\rho_{1\infty})e^{-l_1 t} + \rho_{1\infty} \end{cases} \tag{18}$$

with $0 \le \delta_{11} \le 1$, $0 \le \delta_{12} \le 1$, $\rho_{10} > \rho_{1\infty} > 0$, $l_1 > 0$.

The reference command of γ is chosen as

$$\gamma_{\mathrm{d}} = \arcsin\left(\frac{-k_I\varepsilon_1(t)+\dot{h}_{ref}}{V} - \frac{P_{l1}(t)\dot{P}_{r1}(t) - \dot{P}_{l1}(t)P_{r1}(t) - \tilde{h}\left(\dot{P}_{r1}(t)-\dot{P}_{l1}(t)\right)}{V\left(P_{r1}(t)-P_{l1}(t)\right)}\right) \tag{19}$$

where $\dot{P}_{l1}(t) = (\,\mathrm{sign}(\tilde{h}(0)) - \delta_{11})\dot{\rho}_1(t)$, $\dot{P}_{r1}(t) = (\,\mathrm{sign}(\tilde{h}(0)) + \delta_{12})\dot{\rho}_1(t)$, $\dot{\rho}_1 = -l_1(\rho_{10}-\rho_{1\infty})e^{-l_1 t}$, $k_I > 0$.

If $\gamma \to \gamma_{\mathrm{d}}$, the corresponding dynamics of $\varepsilon_1(t)$ satisfies $\dot{\varepsilon}_1(t) + k_I\varepsilon_1(t) = 0$. That is, $\varepsilon_1(t)$ will be bounded if $\gamma \to \gamma_{\mathrm{d}}$.[25,31] We further conclude that the prescribed performance of $\tilde{h}$ can be guaranteed according to theorem 1.

Model transformation

The starting point is to transform the original nonaffine models (3) to (5) into a norm output feedback formulation.

Define

$$\begin{cases} x_1 = \gamma, \ x_2 = \theta \\ x_3 = Q, \ x_4 = \delta_{\mathrm{e}} \end{cases} \tag{20}$$

where the control input δ_{e} is viewed as a newly defined state to deal with the nonaffine feature.

By introducing a virtual control input u_h, the nonaffine systems (3) to (5) can be expressed as an affine one, which facilitates the altitude control design

$$\begin{cases} \dot{x}_1 = \Gamma_1(x_1, x_2) \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = \Gamma_3(x_1, x_2, x_3, x_4) \\ \dot{x}_4 = -\kappa_4 x_4 + u_h \end{cases} \tag{21}$$

where $\kappa_4 > 0$, the system "$\dot{x}_4 = -\kappa_4 x_4 + u_h$" is input-to-state stable, $\Gamma_1(x_1, x_2)$ and $\Gamma_3(x_1, x_2, x_3, x_4)$ are completely unknown functions.

Remark 3. From equations (3) to (5), (21), and previous studies,[23,30] we have

$$\begin{cases} \dfrac{\partial \Gamma_1(x_1,x_2)}{\partial x_2} \approx \dfrac{\bar{q}SC_L^{\alpha} + T\cos(\theta-\gamma)}{mV} > 0 \\ \dfrac{\partial \Gamma_3(x_1,x_2,x_3,x_4)}{\partial x_4} \approx \dfrac{\bar{q}S\bar{c}c_e + \tilde{\psi}_2 N_2^{\delta_e}/k_2}{I_{yy}} > 0 \end{cases}$$

Then, we make the following model transformation.

Step 1. Define $z_1 = x_1 = \gamma$, $z_2 = \dot{z}_1 = \Gamma_1(x_1,x_2)$. Invoking (21), $\dot{z}_2$ is derived as

$$\begin{aligned} \dot{z}_2 &= \frac{\partial \Gamma_1(x_1,x_2)}{\partial x_1}\dot{x}_1 + \frac{\partial \Gamma_1(x_1,x_2)}{\partial x_2}\dot{x}_2 \\ &= \frac{\partial \Gamma_1(x_1,x_2)}{\partial x_1}\Gamma_1(x_1,x_2) + \frac{\partial \Gamma_1(x_1,x_2)}{\partial x_2}x_3 \\ &\triangleq H_1(x_1,x_2,x_3) \end{aligned} \tag{22}$$

It is easy to note that $\frac{\partial H_1(x_1,x_2,x_3)}{\partial x_3} = \frac{\partial \Gamma_1(x_1,x_2)}{\partial x_2} > 0$.

Step 2. We further define $z_3 = \dot{z}_2 = H_1(x_1,x_2,x_3)$. Taking time derivative of z_3 and substituting (21), we have

$$\begin{aligned} \dot{z}_3 &= \frac{\partial H_1(x_1,x_2,x_3)}{\partial x_1}\dot{x}_1 + \frac{\partial H_1(x_1,x_2,x_3)}{\partial x_2}\dot{x}_2 \\ &\quad + \frac{\partial H_1(x_1,x_2,x_3)}{\partial x_3}\dot{x}_3 \\ &= \frac{\partial H_1(x_1,x_2,x_3)}{\partial x_1}\Gamma_1(x_1,x_2) + \frac{\partial H_1(x_1,x_2,x_3)}{\partial x_2}x_3 \\ &\quad + \frac{\partial H_1(x_1,x_2,x_3)}{\partial x_3}\Gamma_3(x_1,x_2,x_3,x_4) \\ &\triangleq H_2(x_1,x_2,x_3,x_4) \end{aligned} \tag{23}$$

Step 3. Finally, we define $z_4 = \dot{z}_3 = H_2(x_1,x_2,x_3,x_4)$. Using (14), the time derivative of z_3 is given by

$$\begin{aligned} \dot{z}_4 &= \frac{\partial H_2(x_1,x_2,x_3,x_4)}{\partial x_1}\dot{x}_1 + \frac{\partial H_2(x_1,x_2,x_3,x_4)}{\partial x_2}\dot{x}_2 \\ &\quad + \frac{\partial H_2(x_1,x_2,x_3,x_4)}{\partial x_3}\dot{x}_3 + \frac{\partial H_2(x_1,x_2,x_3,x_4)}{\partial x_4}\dot{x}_4 \\ &= \frac{\partial H_2(x_1,x_2,x_3,x_4)}{\partial x_1}\Gamma_1(x_1,x_2) + \frac{\partial H_2(x_1,x_2,x_3,x_4)}{\partial x_2}x_3 \\ &\quad + \frac{\partial H_2(x_1,x_2,x_3,x_4)}{\partial x_3}\Gamma_3(x_1,x_2,x_3,x_4) \\ &\quad - \frac{\partial H_2(x_1,x_2,x_3,x_4)}{\partial x_4}\kappa_4 x_4 + \frac{\partial H_2(x_1,x_2,x_3,x_4)}{\partial x_4}u_h \\ &\triangleq H_3(x_1,x_2,x_3,x_4) + \vartheta_h u_h \end{aligned} \tag{24}$$

Notice that $\vartheta_h = \frac{\partial H_2(x_1,x_2,x_3,x_4)}{\partial x_4} = \frac{\partial H_1(x_1,x_2,x_3)}{\partial x_3}\frac{\partial \Gamma_3(x_1,x_2,x_3,x_4)}{\partial x_4} > 0$. With the above transformation, system (21) becomes

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = z_4 \\ \dot{z}_4 = H_3(x_1,x_2,x_3,x_4) + \vartheta_h u_h \end{cases} \tag{25}$$

Table 1. The number of neural network and learning parameter.

Reference	Neural network number	Learning parameter number
Butt et al.[21]	4	>200
Bu et al.[20]	3	>1200
This article	1	1

where $H_3(x_1,x_2,x_3,x_4)$ is an unknown function.

Altitude controller design

By the merit of a high-order SMC technique, a simplified adaptive neural control law is designed for (25) without utilizing backstepping.

Define flight-path angle tracking error e_h and error function E_h as

$$\begin{cases} e_h = \gamma - \gamma_{\mathrm{d}} = z_1 - \gamma_{\mathrm{d}} \\ E_h = \left(\dfrac{\mathrm{d}}{\mathrm{d}t} + \mu\right)^4 \displaystyle\int_0^t e_h d\tau \\ \quad = \dddot{e}_h + 4\mu\ddot{e}_h + 6\mu^2\dot{e}_h + 4\mu^3 e_h + \mu^4 \displaystyle\int_0^t e_h \,\mathrm{d}\tau \end{cases} \tag{26}$$

where $\mu > 0$ and the polynomial $(s+\mu)^4$ is Hurwitz.

Taking time derivative of E_h and invoking (25) and (26) lead to

$$\begin{aligned} \dot{E}_h &= \ddddot{e}_h + 4\mu\dddot{e}_h + 6\mu^2\ddot{e}_h + 4\mu^3\dot{e}_h + \mu^4 e_h \\ &= H_3(x_1,x_2,x_3,x_4) + \vartheta_h u_h - \dddot{\gamma}_{\mathrm{d}} + 4\mu(z_4 - \ddot{\gamma}_{\mathrm{d}}) \\ &\quad + 6\mu^2(z_3 - \dot{\gamma}_{\mathrm{d}}) + 4\mu^3(z_2 - \dot{\gamma}_{\mathrm{d}}) + \mu^4 e_h \\ &= \vartheta_h(F_h + u_h) \end{aligned} \tag{27}$$

with $F_h = H_3(x_1,x_2,x_3,x_4)/\vartheta_h - \dddot{\gamma}_{\mathrm{d}}/\vartheta_h + 4\mu(z_4 - \ddot{\gamma}_{\mathrm{d}})/\vartheta_h + 6\mu^2(z_3 - \ddot{\gamma}_{\mathrm{d}})/\vartheta_h + 4\mu^3(z_2 - \dot{\gamma}_{\mathrm{d}})/\vartheta_h + \mu^4 e_h/\vartheta_h$ being estimated by a neural network, formulated as

$$F_h = \boldsymbol{W}^{*\mathrm{T}}\boldsymbol{h}(\boldsymbol{X}) + \varepsilon, |\varepsilon| \le \varepsilon_{\mathrm{M}} \tag{28}$$

where $\boldsymbol{X} = [\gamma, \theta, Q, \delta_{\mathrm{e}}, z_2, z_3, z_4]^{\mathrm{T}}$ is the input vector of neural network; ε and $\varepsilon_{\mathrm{M}} > 0$ are the approximation error and its upper bound, respectively; $\boldsymbol{W}^* = [w_1^*, w_2^*, \cdots, w_m^*]^{\mathrm{T}}$ is an ideal weight vector, $\boldsymbol{h}(\boldsymbol{X}) = [h_1(\boldsymbol{X}), h_2(\boldsymbol{X}), \cdots, h_m(\boldsymbol{X})]^{\mathrm{T}}$, and $h_j(\boldsymbol{X})$ is selected as the following Gaussian function

$$h_j(\boldsymbol{X}) = \exp\left(-\frac{||\boldsymbol{X} - \boldsymbol{c}||^2}{2b_j^2}\right), \mathrm{j} = 1, 2, \ldots, m \tag{29}$$

where m is the node number; $\boldsymbol{c} = [c_1, c_2, \ldots, c_7]^{\mathrm{T}}$ and $\boldsymbol{b} = [b_{j1}, b_{j2}, \cdots, b_{j7}]^{\mathrm{T}}$ are the center vector and the width vector of $h_j(\boldsymbol{X})$, respectively.

Traditionally, the elements of $\boldsymbol{W}^*$ are required to be updated via exploiting an adaptive law based on Lyapunov stability theory for the sake of achieving desired neural

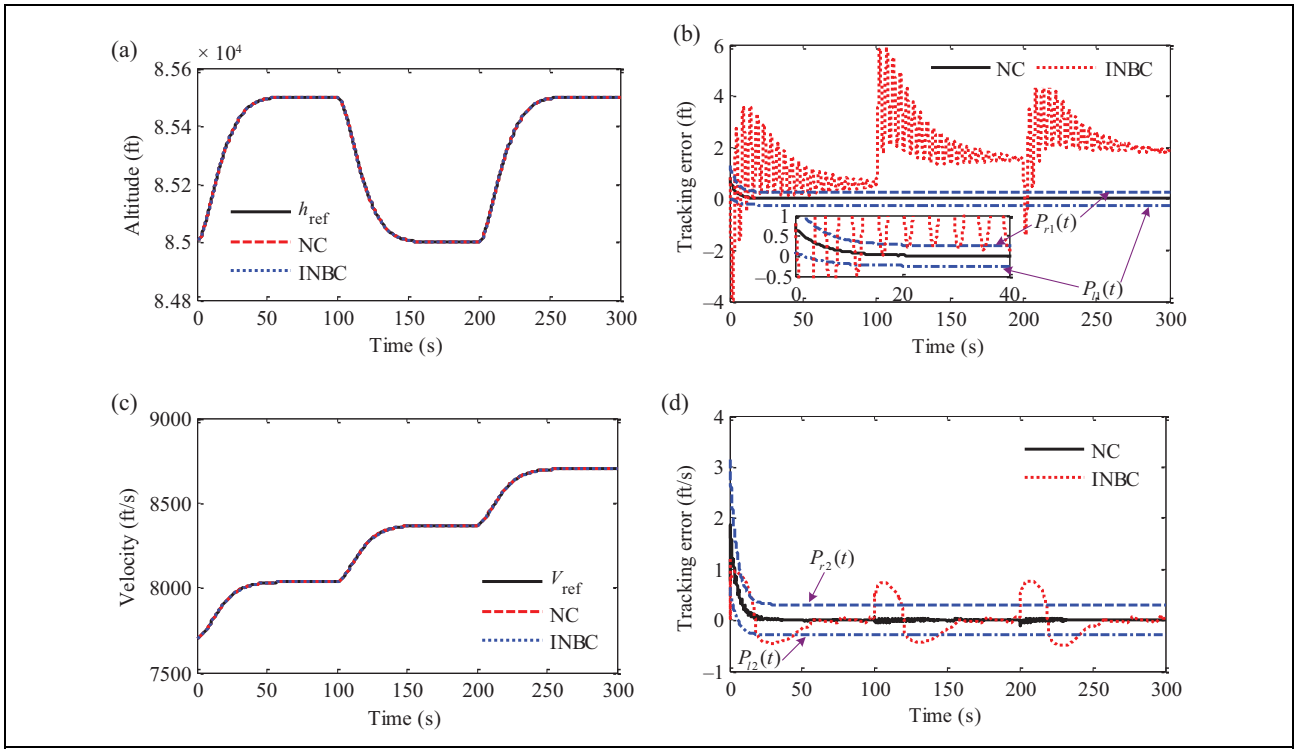

Figure 3. Altitude and velocity tracking in example 1. (a) Altitude tracking performance; (b) altitude tracking error; (c) velocity tracking performance; and (d) velocity tracking error.

approximation performance. In this way, there will be too many online learning parameters (i.e. $w_1^*, w_2^*, \cdots, w_m^*$) and the computational burden is high. In this subsection, an advanced regulation scheme is applied to reduce online learning parameters.

Define $\varphi = ||\boldsymbol{W}^*||^2$ and devise the following neural control law

$$u_h = -k_h E_h - \frac{E_h \hat{\varphi} \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X}) \boldsymbol{h}(\boldsymbol{X})}{2} \tag{30}$$

where $k_h > 0$ is a chosen constant; $\hat{\varphi}$ is the estimation of φ, and its adaptive law is developed as

$$\dot{\hat{\varphi}} = \frac{\lambda_h E_h^2 \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X}) \boldsymbol{h}(\boldsymbol{X})}{2} - 2\hat{\varphi} \tag{31}$$

with $\lambda_h > 0$.

Theorem 2. Consider the closed-loop system consisting of plant (25) with controller (30) and adaptive law (31). Then, all the signals involved are semi-globally uniformly ultimately bounded. Based on theorem 1, it is further concluded that $P_{l1}(t) < \tilde{h} < P_{r1}(t)$.

Proof. Define the estimation error as

$$\tilde{\varphi} = \hat{\varphi} - \varphi \tag{32}$$

Consider the following Lyapunov function candidate

$$W_h = \frac{E_h^2}{2\vartheta_h} + \frac{\tilde{\varphi}^2}{2\lambda_h} \tag{33}$$

Invoking (27), (28), (30) to (32), the time derivative of W_h is derived as

$$\begin{aligned}
\dot{W}_h &= \frac{E_h \dot{E}_h}{\vartheta_h} + \frac{\tilde{\varphi}\dot{\hat{\varphi}}}{\lambda_h} = \frac{E_h \vartheta_h (F_h + u_h)}{\vartheta_h} + \frac{\tilde{\varphi}}{\lambda_h}\left[\frac{\lambda_h E_h^2 \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X})\boldsymbol{h}(\boldsymbol{X})}{2} - 2\hat{\varphi}\right] \\
&= E_h(F_h + u_h) + \frac{E_h^2 \tilde{\varphi} \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X})\boldsymbol{h}(\boldsymbol{X})}{2} - \frac{2\hat{\varphi}\tilde{\varphi}}{\lambda_h} \\
&= E_h\left(\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{h}(\boldsymbol{X}) + \varepsilon - k_h E_h - \frac{E_h \hat{\varphi} \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X})\boldsymbol{h}(\boldsymbol{X})}{2}\right) + \frac{E_h^2 \tilde{\varphi} \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X})\boldsymbol{h}(\boldsymbol{X})}{2} - \frac{2\hat{\varphi}\tilde{\varphi}}{\lambda_h} \\
&= -k_h E_h^2 + E_h \boldsymbol{W}^{*\mathrm{T}}\boldsymbol{h}(\boldsymbol{X}) + E_h\varepsilon - \frac{E_h^2 \hat{\varphi} \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X})\boldsymbol{h}(\boldsymbol{X})}{2} + \frac{E_h^2 \tilde{\varphi} \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X})\boldsymbol{h}(\boldsymbol{X})}{2} - \frac{2\hat{\varphi}\tilde{\varphi}}{\lambda_h} \\
&= -k_h E_h^2 + E_h \boldsymbol{W}^{*\mathrm{T}}\boldsymbol{h}(\boldsymbol{X}) + E_h\varepsilon - \frac{E_h^2 (\tilde{\varphi} + \varphi) \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X})\boldsymbol{h}(\boldsymbol{X})}{2} + \frac{E_h^2 \tilde{\varphi} \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X})\boldsymbol{h}(\boldsymbol{X})}{2} - \frac{2\hat{\varphi}\tilde{\varphi}}{\lambda_h} \\
&= -k_h E_h^2 + E_h \boldsymbol{W}^{*\mathrm{T}}\boldsymbol{h}(\boldsymbol{X}) + E_h\varepsilon - \frac{E_h^2 \varphi \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X})\boldsymbol{h}(\boldsymbol{X})}{2} - \frac{2\hat{\varphi}\tilde{\varphi}}{\lambda_h}
\end{aligned} \tag{34}$$

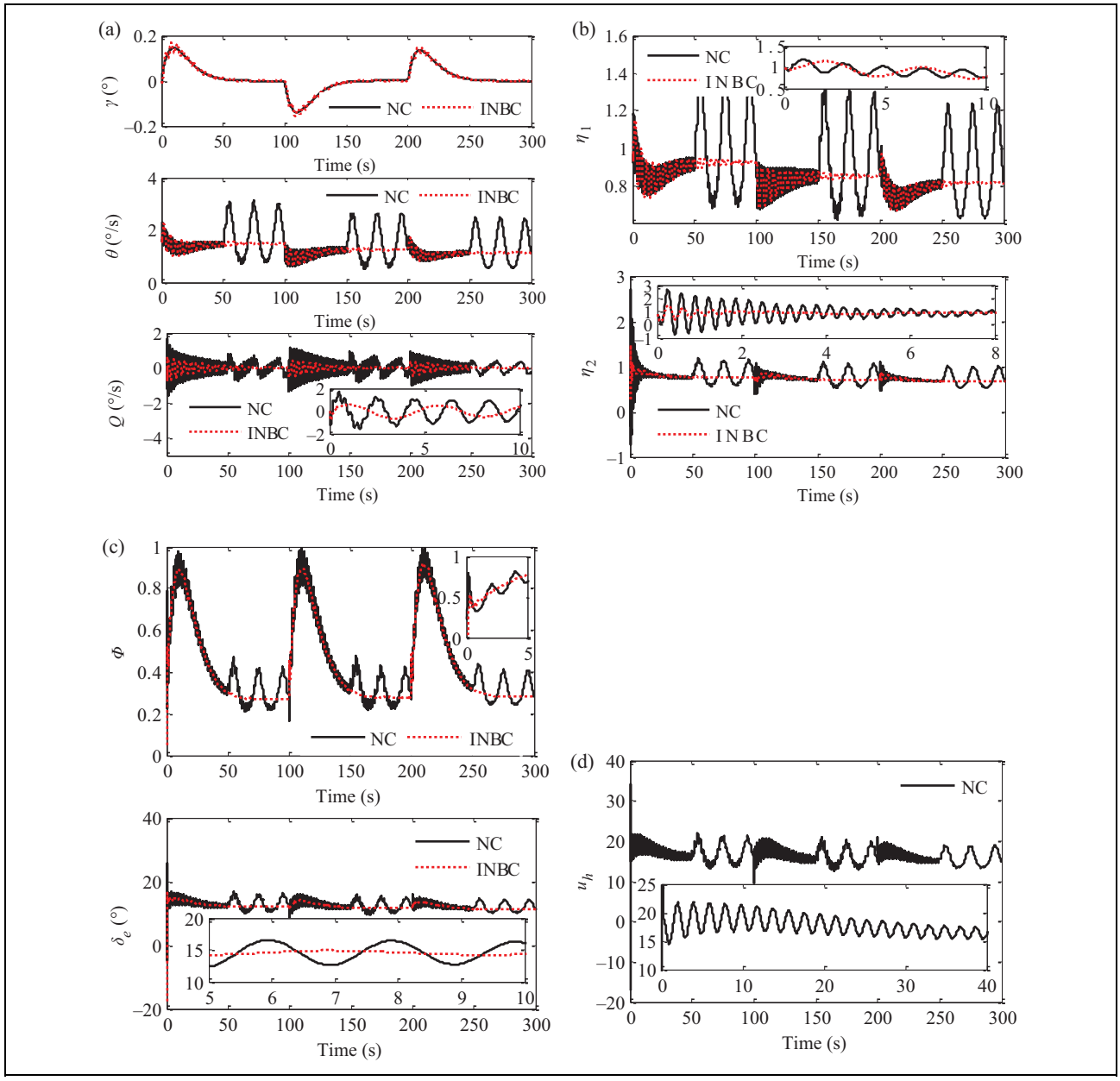

Figure 4. The attitude angles, flexible states, and the control inputs in example 1. (a) attitude angles; (b) flexible states; (c) control inputs Φ and δ_e; and (d) virtual control input u_h.

From the fact that $\tilde{\varphi}^2 + 2\tilde{\varphi}(\hat{\varphi} - \tilde{\varphi}) + \varphi^2 = \tilde{\varphi}^2 + 2\tilde{\varphi}\varphi + \varphi^2 = (\tilde{\varphi} + \varphi)^2 \geq 0$, we obtain $2\tilde{\varphi}\hat{\varphi} \geq \tilde{\varphi}^2 - \varphi^2$. Thereby, equation (34) becomes

$$\dot{W}_h \leq -k_h E_h^2 - \frac{\tilde{\varphi}^2}{\lambda_h} + E_h \boldsymbol{W}^{*\mathrm{T}} \boldsymbol{h}(\boldsymbol{X}) + E_h \varepsilon - \frac{E_h^2 \varphi \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X}) \boldsymbol{h}(\boldsymbol{X})}{2} + \frac{\varphi^2}{\lambda_h} \tag{35}$$

Notice that

$$E_h \boldsymbol{W}^{*\mathrm{T}} \boldsymbol{h}(\boldsymbol{X}) \leq \frac{E_h^2 ||\boldsymbol{W}^{*\mathrm{T}} \boldsymbol{h}(\boldsymbol{X})||^2}{2} + \frac{1}{2} = \frac{E_h^2 ||\boldsymbol{W}^*||^2 ||\boldsymbol{h}(\boldsymbol{X})||^2}{2} + \frac{1}{2} = \frac{E_h^2 \varphi \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{X}) \boldsymbol{h}(\boldsymbol{X})}{2} + \frac{1}{2}, \; E_h \varepsilon \leq \frac{E_h^2 \varepsilon_{\mathrm{M}}^2}{2} + \frac{1}{2}$$

Thus the following inequality holds

$$\dot{W}_h \leq -\left(k_h - \frac{\varepsilon_{\mathrm{M}}^2}{2}\right) E_h^2 - \frac{\tilde{\varphi}^2}{\lambda_h} + 1 + \frac{\varphi^2}{\lambda_h} \tag{36}$$

Let $k_h > \varepsilon_M^2/2$ and define the following compact sets

$$\Omega_{E_h} = \left\{ E_h \middle| |E_h| \le \sqrt{\left(1+\frac{\varphi^2}{\lambda_h}\right) \Big/ \left(k_h - \frac{\varepsilon_M^2}{2}\right)} \right\}$$

$$\Omega_{\tilde{\varphi}} = \left\{ \tilde{\varphi} \middle| |\tilde{\varphi}| \le \sqrt{\left(1+\frac{\varphi^2}{\lambda_h}\right) \Big/ \left(\frac{1}{\lambda_h}\right)} \right\}$$

If $E_h \notin \Omega_E$ or $\tilde{\varphi} \notin \Omega_{\tilde{\varphi}}$, we have $\dot{W}_h < 0$. Hence, E_h and $\tilde{\varphi}$ are semi-globally uniformly ultimately bounded and can be adequately small by choosing appropriate k_h and λ_h. Since the polynomial $(s+\mu)^4$ is Hurwitz, also e_h is bounded. That is, $\gamma \to \gamma_d$. Thus the altitude tracking error $\tilde{h}$ can be limited within $P_{l1}(t) < \tilde{h} < P_{r1}(t)$. By selecting certain design parameters for the performance functions $P_{l1}(t)$ and $P_{r1}(t)$, satisfactory transient performance of $\tilde{h}$ can be achieved. This is the end of the proof. □

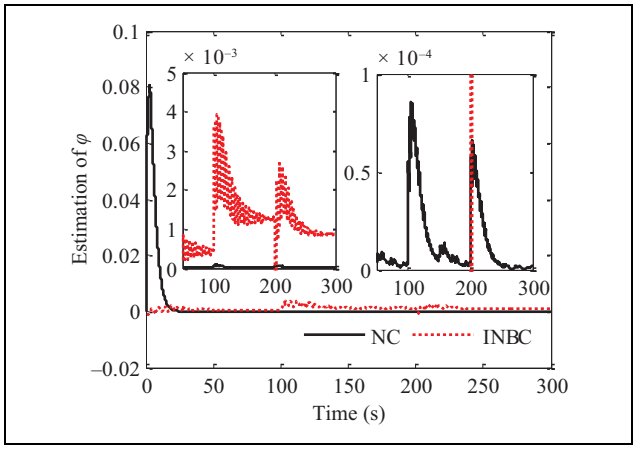

Figure 5. The estimation of φ in example 1.

Remark 4. Unlike the neural backstepping control methodologies addressed in literature,[19–21] in this article, there is no need of complicated recursive design procedure of backstepping. Moreover, the presented control approach shows lower computational load than the ones of earlier studies[19–21] since only one neural network is used and meanwhile only an online learning parameter is required for neural approximation.

Remark 5. In contrast to the neural controllers (NCs) of earlier studies,[20,21] the presented control method exhibits much lower computational load since only one neural network and one learning parameter $\hat{\varphi}$ are need in this study, as shown in Table 1.

Velocity controller design

Noting that velocity dynamics of AHVs is simple, a proportion and integral controller is adopted.

$$\Phi = -k_{V1}\varepsilon_2(t) - k_{V2}\int_0^t \varepsilon_2(\tau)\,\mathrm{d}\tau \tag{37}$$

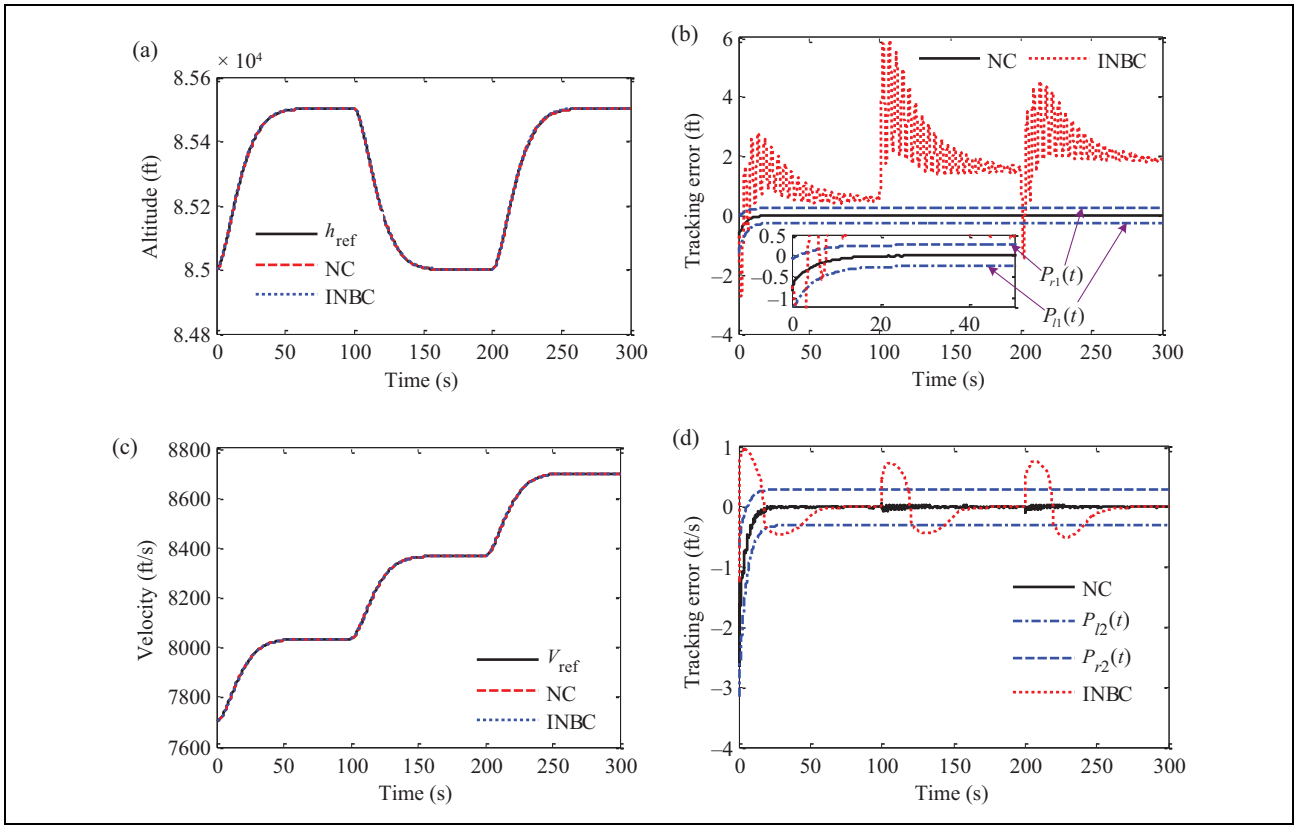

Figure 6. Altitude and velocity tracking in example 2. (a) altitude tracking performance; (b) altitude tracking error; (c) velocity tracking performance; and (d) velocity tracking error.

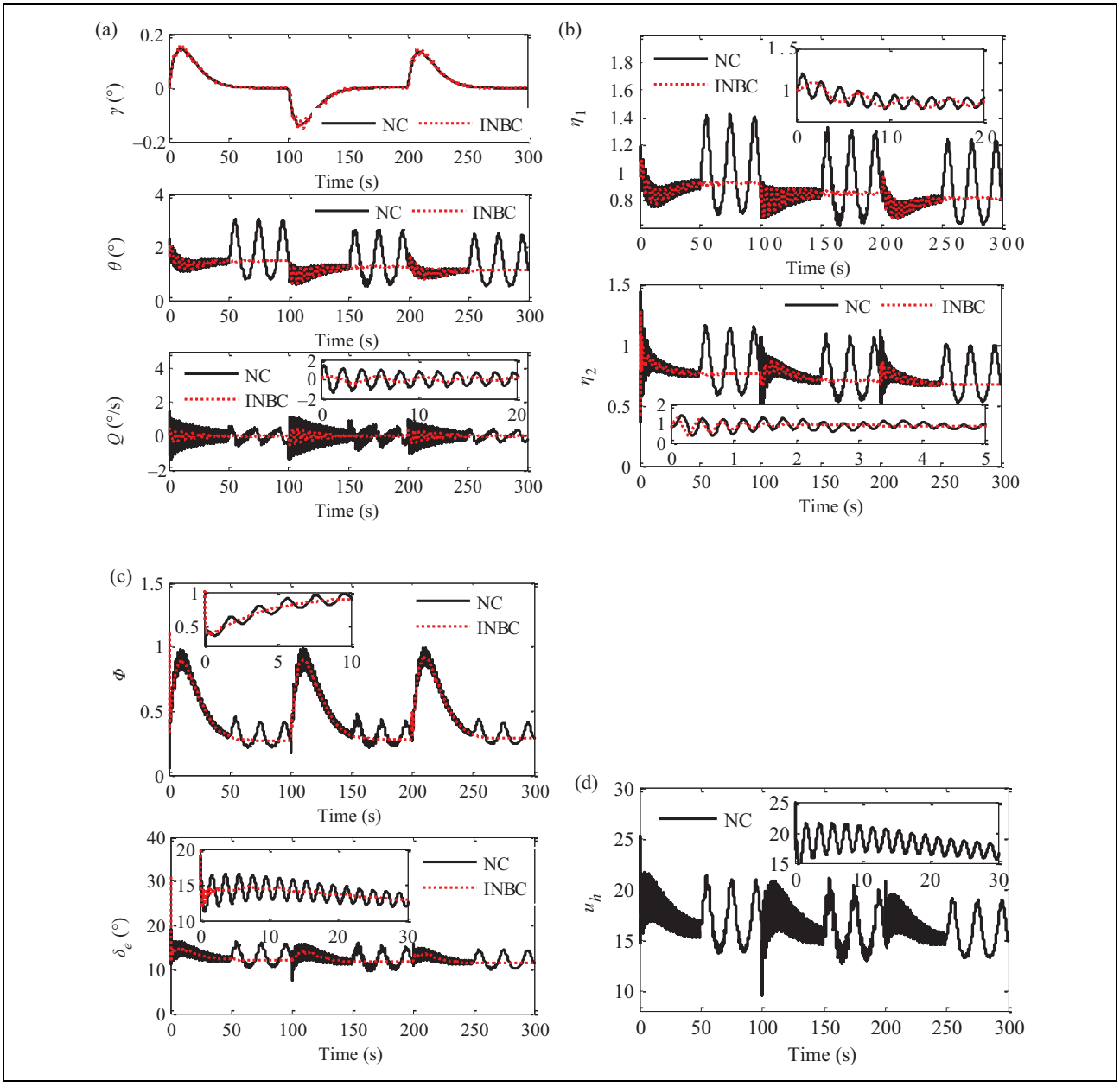

Figure 7. The attitude angles, flexible states, and the control inputs in example 2.

with

$$\begin{cases} \varepsilon_2(t) = \ln\left(\dfrac{\vartheta_2(t)}{1-\vartheta_2(t)}\right) \\ \vartheta_2(t) = \dfrac{\tilde{V} - P_{l2}(t)}{P_{r2}(t) - P_{l2}(t)} \\ \tilde{V} = V - V_{\text{ref}} \\ P_{l2}(t) = [\text{sign}(\tilde{V}(0)) - \delta_{21}]\rho_2(t) - \rho_{2\infty}\,\text{sign}(\tilde{V}(0)) \\ P_{r2}(t) = [\text{sign}(\tilde{V}(0)) + \delta_{22}]\rho_2(t) - \rho_{2\infty}\,\text{sign}(\tilde{V}(0)) \\ \rho_2(t) = (\rho_{20} - \rho_{2\infty})e^{-l_2 t} + \rho_{2\infty} \end{cases} \tag{38}$$

where $0 \le \delta_{21} \le 1$, $0 \le \delta_{22} \le 1$, $\rho_{20} > \rho_{2\infty} > 0$, and $l_2 > 0$ are design parameters.

Simulation results

This section presents numerical simulation results to verify the efficiency of the proposed control strategy. The input vector of neural network is $\boldsymbol{X} = [\gamma, \theta, Q, \delta_{\text{e}}, z_2, z_3, z_4]^{\text{T}}$ with $\gamma \in [-1\text{ deg}, 1\text{ deg}]$, $\theta \in [0\text{ deg}, 5\text{ deg}]$, $Q \in [-5\text{ deg/s}, 5\text{ deg/s}]$, $\delta_{\text{e}} \in [-20\text{ deg}, 20\text{ deg}]$, $z_2 \in [-2 \times 10^{-4}\text{ rad/s}, 4 \times 10^{-4}\text{ rad/s}]$, $z_3 \in [-2 \times 10^{-5}\text{ rad/s}^2, 10^{-4}\text{ rad/s}^2]$, and $z_4 \in [-2 \times 10^{-6}\text{ rad/s}^3, 10^{-5}\text{ rad/s}^3]$. The node number $m = 30$. Moreover, to test the proposed controller's robustness with respect to parametric uncertainties, we suppose

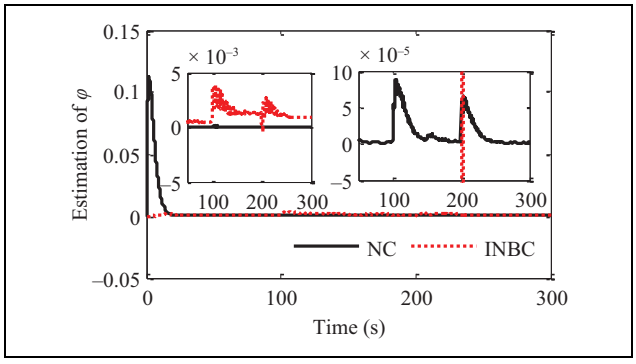

Figure 8. The estimation of φ in example 2.

that all the coefficients in equations (1) to (7) are uncertain. Define

$$C = \begin{cases} C_0, & 0\ \text{s} \le t < 50\ \text{s} \\ C_0[1 + 0.4\sin(0.1\pi t)], & 50\ \text{s} \le t < 100\ \text{s} \\ C_0, & 100\ \text{s} \le t < 150\ \text{s} \\ C_0[1 + 0.4\sin(0.1\pi t)], & 150\ \text{s} \le t < 200\ \text{s} \\ C_0, & 200\ \text{s} \le t < 250\ \text{s} \\ C_0[1 + 0.4\sin(0.1\pi t)], & 250\ \text{s} \le t \le 300\ \text{s} \end{cases} \tag{39}$$

where C means the value of uncertain coefficient and C_0 denotes the normal value of C.

The design parameters are chosen as follows: $k_{V1} = 0.2$, $k_{V2} = 0.8$, $k_I = 2$, $k_h = 5$, $\kappa_4 = 75$, $\mu = 20$, $\lambda_h = 0.05$. The adopted performance functions are given by

$$\begin{aligned} P_{l1}(t) &= [\text{sign}(\tilde{h}(0)) - 0.5]\rho_1(t) - 0.5\,\text{sign}(\tilde{h}(0)), \\ P_{r1}(t) &= [\text{sign}(\tilde{h}(0)) + 0.5]\rho_1(t) - 0.5\,\text{sign}(\tilde{h}(0)), \\ \rho_1(t) &= (1.2 - 0.5)e^{-0.2t} + 0.5 \end{aligned}$$

$$\begin{aligned} P_{l2}(t) &= [\text{sign}(\tilde{V}(0)) - 0.5]\rho_2(t) - 0.6\,\text{sign}(\tilde{V}(0)), \\ P_{r2}(t) &= [\text{sign}(\tilde{V}(0)) + 0.5]\rho_1(t) - 0.6\,\text{sign}(\tilde{V}(0)), \\ \rho_2(t) &= (2.5 - 0.6)e^{-0.2t} + 0.6 \end{aligned}$$

The proposed NC is compared with an improved neural backstepping control (INBC) strategy (When using the INBC method, parametric uncertainties aren't considered.) to show its superiority in tracking performance. Two different examples are taken into consideration. In example 1, the initial tracking errors are chosen as positive values, that is, $\tilde{h}(0) = 0.8\ ft$, $\tilde{V}(0) = 1.2$ ft/s. In the second example, we select negative initial tracking errors, that is, $\tilde{h}(0) = -0.8\ ft$, $\tilde{V}(0) = -1.2$ ft/s.

The simulation results are depicted in Figures 3 to 8. Figures 3 and 6 reveal that the addressed NC scheme can provide robust tracking of altitude and velocity commands in the presence of uncertain parameters, and moreover the altitude tracking error and velocity tracking error provided by the addressed NC have better transient and steady-state performance with small overshoot than by the INBC approach. For both control methodologies, attitude angles, flexible states, and control inputs, shown in Figures 4 and 7, are bounded and smooth (without high-frequency chattering). The estimation performance of φ is presented in Figures 5 and 8.

Conclusions

In this article, a novel proscribed performance control approach utilizing nonaffine models is exploited for an AHV via neural approximation. By introducing a virtual control input and making a model transformation, a simple adaptive NC with prescribed performance is proposed and the complex design process of backstepping is eliminated. Further, an advanced learning algorithm is applied to devise adaptive law for online learning parameter, based on which not only the control law's robustness is guaranteed but also the computational cost is quite low. Besides, a new prescribed performance is presented to constraint the convergence overshoot of tracking errors. Finally, simulation results indicate that the addressed approach can guarantee altitude tracking error and velocity tracking error with excellent transient and steady-state performance.

Declaration of conflicting interests

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

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by National Natural Science Foundation of China (grant nos 61603410, 61703424, and 61703421) and Young Talent Fund of University Association for Science and Technology in Shaanxi, China (grant no 20170107)

References

1. Bu X. Guaranteeing prescribed output tracking performance for air-breathing hypersonic vehicles via nonaffine back-stepping control design. *Nonlinear Dyn* 2018; 91: 525–538. DOI: 10.1007/s11071-017-3887-1.
2. Voland RT, Huebner LD, and McClinton CR. X-43A Hypersonic vehicle technology development. *Acta Astronaut* 2006; 59: 181–191.
3. Bu X, Wu X, Tian M, et al. High-order tracking differentiator based adaptive neural control of a flexible air-breathing hypersonic vehicle subject to actuators constraints. *ISA Trans* 2015; 58: 237–247.
4. Sigthorsson DO, Jankovsky P, Serrani A, et al. Robust linear output feedback control of an airbreathing hypersonic vehicle. *J Guid Control Dyn* 2008; 31(4): 1052–1066.
5. Morelli EA. Flight-test experiment design for characterizing stability and control of hypersonic vehicles. *J Guid Control Dyn* 2009; 32(3): 949–959.
6. Wu HN, Liu ZY, and Guo L. Robust L_∞-gain fuzzy disturbance observer-based control design with adaptive bounding

for a hypersonic vehicle. *IEEE Trans Fuzzy Syst* 2014; 22(6): 1401–1412.
7. Wang J, Wu Y, and Dong X. Recursive terminal sliding mode control for hypersonic flight vehicle with sliding mode disturbance observer. *Nonlinear Dyn* 2015; 81: 1489–1510.
8. Tian B, Su R, and Fan W. Multiple-time scale smooth second order sliding mode controller design for flexible hypersonic vehicles. *Proc IMechE Part G: J Aerosp Eng* 2015; 229(5): 781–791.
9. Song J, Wang L, Cai G, et al. Nonlinear fractional order proportion-integral-derivative active disturbance rejection control method design for hypersonic vehicle attitude control. *Acta Astronaut* 2015; 111: 160–169.
10. He H, Qi R, Jiang B, et al. Adaptive output feedback fault-tolerant control design for hypersonic flight vehicles. *J Franklin Instit* 2015; 352: 1811–1835.
11. Xu B, Guo Y, Yuan Y, et al. Fault-tolerant control using command-filtered adaptive back-stepping technique: application to hypersonic longitudinal flight dynamics. *Int J Adapt Control Signal Process* 2015; 25: 1301–1326. DOI: 10.1002/acs.2596.
12. Tao X, Li N, and Li S. Multiple model predictive control for large envelope flight of hypersonic vehicle systems. *Inform Sci* 2016; 328: 115–126.
13. Bu X, Wu X, Wei D, et al. Neural-approximation-based robust adaptive control of flexible air-breathing hypersonic vehicles with parametric uncertainties and control input constraints. *Inform Sci* 2016; 346–347: 29–43.
14. Bu X, Wu X, He G, et al. Novel adaptive neural control design for a constrained flexible air-breathing hypersonic vehicle based on actuator compensation. *Acta Astronaut* 2016; 120: 75–86.
15. Bu X, Wu X, Ma Z, et al. Novel auxiliary error compensation design for the adaptive neural control of a constrained flexible air-breathing hypersonic vehicle. *Neurocomputing* 2016; 171: 313–324.
16. Bu X, Wu X, Zhang R, et al. Tracking differentiator design for the robust backstepping control of a flexible air-breathing hypersonic vehicle. *J Franklin Instit* 2015; 352: 1739–1765.
17. Xu B, Huang X, Wang D, et al. Dynamic surface control of constrained hypersonic flight models with parameter estimation and actuator compensation. *Asian J Control* 2014; 16(1): 162–174.
18. Fiorentini L, Serrani A, Bolender MA, et al. Nonlinear robust adaptive control of flexible air-breathing hypersonic vehicles. *J Guid Control Dyn* 2009; 32(2): 401–416.
19. Gao D, Wang S, and Zhang H. A singularly perturbed system approach to adaptive neural back-stepping control design of hypersonic vehicles. *J Intell Robot Syst* 2014; 73: 249–259.
20. Bu X, Wu X, Ma Z, et al. Nonsingular direct neural control of air-breathing hypersonic vehicle via back-stepping. *Neurocomputing* 2015l; 153: 164–173.
21. Butt WA, Yan L, and Kendrick AS. Adaptive integral dynamic surface control of a hypersonic flight vehicle. *Int J Syst Sci* 2015; 46(10): 1717–1728.
22. Bu X, Wu X, Huang J, et al. Minimal-learning-parameter based simplified adaptive neural back-stepping control of flexible air-breathing hypersonic vehicles without virtual controllers. *Neurocomputing* 2016; 175: 816–825.
23. Bu X, Wu X, Ma Z, et al. Novel adaptive neural control of flexible air-breathing hypersonic vehicles based on sliding mode differentiator. *Chin J Aeronaut* 2015; 28(4): 1209–1216.
24. Xu B, Fan Y, and Zhang S. Minimal-learning-parameter technique based adaptive neural control of hypersonic flight dynamics without back-stepping. *Neurocomputing* 2015; 164: 201–209.
25. Bu X, Wu X, Huang J, et al. A guaranteed transient performance-based adaptive neural control scheme with low-complexity computation for flexible air-breathing hypersonic vehicles. *Nonlinear Dyn* 2016; 84(4): 2175–2194.
26. Bu X, Wu X, Zhu F, et al. Novel prescribed performance neural control of a flexible air-breathing hypersonic vehicle with unknown initial errors. *ISA Trans* 2015; 59: 149–159.
27. Bu X, Wei D, Wu X, et al. Guaranteeing preselected tracking quality for air-breathing hypersonic non-affine models with an unknown control direction via concise neural control. *J Franklin Instit* 2016; 353: 3207–3232.
28. Bu X, Wu X, Huang J, et al. Robust estimation-free prescribed performance back-stepping control of air-breathing hypersonic vehicles without affine models. *Int J Control* 2016; 89(11): 2185–2200.
29. Bechlioulis C and Rovithakis G. Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance. *IEEE Trans Autom Control* 2008; 53(9): 2090–2099.
30. Parker JT, Serrani A, Yurkovich S, et al. Control-oriented modeling of an air-breathing hypersonic vehicle. *J Guid Control Dyn* 2007; 30(3): 856–869.
31. Zong Q, Wang F, Tian B, et al. Robust adaptive dynamic surface control design for a flexible air-breathing hypersonic vehicle with input constraints and uncertainty. *Nonlinear Dyn* 2014; 78: 289–315.