JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 20, No. 2, March–April 1997

Adaptive Robust Bank-to-Turn Missile Autopilot Design Using Neural Networks

Li-Chen Fu,* Wei-Der Chang,† Jung-Hua Yang,‡ and Te-Son Kuo§
National Taiwan University, Taipei 106, Taiwan, Republic of China

An adaptive robust neural-network-based control approach is proposed for bank-to-turn missile autopilot design. Feedforward neural networks with sigmoid hidden units are analyzed in detail for controller design. Without prior knowledge of the so-called optimal neural networks, we design a controller that exploits the advantages of both neural networks and robust adaptive control theory. For this scheme, a stable adaptive law is determined by using the Lyapunov theory, and the boundedness of all signals in the closed-loop system is guaranteed. No prior offline training phase is necessary, and only a single neural network is employed. It is shown that the tracking errors converge to a neighborhood of zero. Performance of the controller is demonstrated by means of simulations.

Nomenclature

A_n = normal acceleration, g
A_{nx}, A_{ny}, A_{nz} = output measurements of the accelerometer mounted at the center of gravity of the missile corresponding to the $\boldsymbol{b}_x, \boldsymbol{b}_y$, and $\boldsymbol{b}_z$ axes, respectively, g
A_x, A_y, A_z = acceleration along the $\boldsymbol{b}_x, \boldsymbol{b}_y$, and $\boldsymbol{b}_z$ axes, respectively, g
a = speed of sound in air, m/s
$\{\boldsymbol{b}_x, \boldsymbol{b}_y, \boldsymbol{b}_z\}$ = right-handed body coordinate frame that is attached to the missile, consisting of three mutually perpendicular unit vectors
$C_{f_x}, C_{f_y}, C_{f_z}$ = aerodynamic force coefficients corresponding to the $\boldsymbol{b}_x, \boldsymbol{b}_y$, and $\boldsymbol{b}_z$ axes, respectively
$C_{m_x}, C_{m_y}, C_{m_z}$ = moment coefficients corresponding to the $\boldsymbol{b}_x$, $\boldsymbol{b}_y$, and $\boldsymbol{b}_z$ axes, respectively
$\bar{c}$ = mean aerodynamic chord or reference length, m
$d_p,\ d_q,\ d_r$ = effective aileron deflection angle, effective elevator deflection angle, and effective rudder deflection angle, respectively, rad
$\{\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z\}$ = right-handed inertial coordinate frame, consisting of three mutually perpendicular unit vectors
F_{gx}, F_{gy}, F_{gz} = gravitational force along $\boldsymbol{b}_x, \boldsymbol{b}_y$, and $\boldsymbol{b}_z$ axes, respectively, N
F_x, F_y, F_z = external forces along $\boldsymbol{b}_x, \boldsymbol{b}_y$, and $\boldsymbol{b}_z$ axes, respectively, N
$\bar{F}_x, \bar{F}_y, \bar{F}_z$ = aerodynamic forces along $\boldsymbol{b}_x$, $\boldsymbol{b}_y$, and $\boldsymbol{b}_z$ axes, respectively, N
I_{xx}, I_{yy}, I_{zz} = moment of inertia of the missile body about the $\boldsymbol{b}_x, \boldsymbol{b}_y$, and $\boldsymbol{b}_z$ axes, respectively, kg·m^2
M_x, M_y, M_z = total moment about $\boldsymbol{b}_x, \boldsymbol{b}_y$ and $\boldsymbol{b}_z$ axes, respectively, N·m
M_∞ = Mach number
m = missile mass, kg
P, Q, R = roll rate, pitch rate, and yaw rate along the $\boldsymbol{b}_x, \boldsymbol{b}_y$, and $\boldsymbol{b}_z$ axes, respectively (+clockwise), rad/s
Q_s = dynamic pressure, N/m^2
S_{ref} = reference wing area, m^2
T = thrust, kg
$(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})^T$ = velocity vector of the missile represented with respect to the body frame, m/s
u_p, u_q, u_r = actuator inputs for aileron deflection, elevator deflection, and rudder deflection, respectively
V_m = magnitude of missile velocity, m/s
$\boldsymbol{V}_m$ = missile velocity vector, m/s
w_a = actuator bandwidth, rad/s
$(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})^T$ = position vector of the center of mass of the missile represented with respect to the inertial frame, m
α = angle of attack, rad
β = sideslip angle, rad
$\delta_1, \delta_2, \delta_3, \delta_4$ = four tail deflection angles that are intermixed to produce the effective aileron, elevator, and rudder deflection angles, respectively, rad
θ = body-axis pitch angle measured from the projection of $\boldsymbol{b}_x$ onto the plane horizontal to $\boldsymbol{b}_x$, rad
ρ = density of air, kg/m^3
ϕ = body-axis bank angle measured from the downward vertical direction to $\boldsymbol{b}_z$ about the axis $\boldsymbol{b}_x$, rad
ψ = body-axis yaw angle measured between a fixed compass bearing and the projection of $\boldsymbol{b}_x$ onto the horizontal plane, rad

I. Introduction

BANK-TO-TURN (BTT) missiles have high maneuverability as compared with skid-to-turn (STT) missiles. To execute a BTT autopilot strategy, these missiles must have the ability to change the orientation of the acceleration rapidly by means of a very high roll rate. However, such a high roll rate will induce cross coupling, resulting in undesirable pitch and yaw motion. Furthermore, the autopilot design will become more difficult because of imprecise knowledge of the aerodynamic parameters.

In the literature, the method of feedback linearization[1] has been applied to the flight control problem.[2–5] To perform exact cancellation, the technique assumes exact knowledge of aerodynamic coefficients and aircraft configuration parameters (e.g., reference wing area, mean aerodynamic chord, mass, moment of inertia) in the entire flight envelope. In practice, this assumption is not valid; hence, further study on robustness is necessary. Hedrick and Gopalswamy[6] used a sliding mode control technique to design a robust pitch axis control system for a high-performance aircraft. By using a rather detailed six-degree-of-freedom (DOF) missile model, Lian et al.[2] proposed an adaptive robust BTT autopilot design to treat the uncertainties efficiently without prior knowledge of the bounds on the uncertainties. Huang and Lin[3] applied sliding mode control to cope with model uncertainty of the BTT missile autopilot design. However, these schemes require a tedious design procedure to perform I/O feedback linearization.

Received Oct. 27, 1995; revision received Aug. 13, 1996; accepted for publication Oct. 28, 1996.

*Professor, Department of Electrical Engineering and Department of Computer Science and Information Engineering.
†M.S., Department of Electrical Engineering.
‡Ph.D., Department of Electrical Engineering.
§Professor, Department of Electrical Engineering.

Recently, neural networks have shown great promise in the realm of nonlinear control problems because of their universal approximation capability. This powerful property has inspired the development of many neural-network-based controllers without significant prior knowledge of the system dynamics. So far, some developed feedforward neural-network-based control schemes, whose network parameters are adapted according to the Lyapunov theory, have provided rigorous theoretical analysis. In particular, Sanner and Slotine[7] utilized the spatial sampling theory to design a Gaussian radial basis function network to deal with plant nonlinearities, which was then applied to the flight control problem[8] and to robotic systems.[9] Unfortunately, this scheme cannot be applied to approximate very-high-dimensional nonlinearities in practice because the total number of Gaussian hidden units will become prohibitively large. In flight control, however, a rather precise aircraft model needs to be considered, so that highly dimensional state-space will become inevitable. Moreover, Chen and Liu[10] used multilayer feedforward networks and a backpropagation (BP) rule with a deadzone to overcome modeling errors; this method, however, required that the initial parameter errors be small enough to ensure convergence. Following a similar line of thought, Lewis et al.[11] also utilized multilayer feedforward networks but applied e_1-modification to the BP rule and added an extra control signal to increase robustness. The drawback of this scheme is that the design approach needs a priori knowledge of the upper bound on the optimal neuron weights, whereas the present work does not require such prior knowledge. Nevertheless, these methods provide a link between neural networks and adaptive control theory.

An adaptive robust neural-network-based controller is proposed for a BTT autopilot design. Feedforward neural networks with sigmoid hidden units are analyzed in detail for controller design. Without prior knowledge of the so-called optimal neural networks, we have designed a controller that exploits the advantages of both neural networks and neural-network-based sliding mode control. For this scheme, a stable adaptive law is devised by using the Lyapunov theory, whereby the boundedness of all the signals as well as the convergence of the tracking errors of the closed-loop system are guaranteed. It is particularly noteworthy that no prior offline training phase is necessary and that only a single neural network is employed.

The organization of the paper is as follows. Section II presents the dynamic model of a missile system. Section III gives a detailed analysis of the multilayer feedforward neural networks that will be used in design. In Sec. IV, the design procedure is presented, and the stability property of the proposed BTT missile autopilot is proven. Simulation results are presented in Sec. V to verify the control approach. Finally, some conclusions are given in Sec. VI.

II. Missile Model

The missile model integrates several parts that perform different functions: 1) The aerodynamics are represented throughout the entire flight envelope using multidimensional tables and linear interpolation, which perform as a nonlinear function generator; 2) an atmospheric data model is given in terms of atmospheric tables, based on which the speed of sound and air density can be calculated; and 3) the actuator dynamics of the missile fins and the equations of motion of the missile are both expressed in a state-space form. As a result, the description of the overall missile model is decomposed into three parts as shown next.

A. Equations of Motion

Suppose the aircraft has a rigid body. Then, the complete six-DOF missile dynamics expressed in a state-space form are as follows:

$$\dot{P} = \frac{I_{yy} - I_{zz}}{I_{xx}} QR + \frac{M_x}{I_{xx}} \qquad \dot{Q} = \frac{I_{zz} - I_{xx}}{I_{yy}} RP + \frac{M_y}{I_{yy}}$$

$$\dot{R} = \frac{I_{xx} - I_{yy}}{I_{zz}} PQ + \frac{M_z}{I_{zz}}$$

$$\dot{\phi} = P + (Q \sin\phi + R \cos\phi) \tan\theta \qquad \dot{\theta} = Q \cos\phi - R \sin\phi$$

$$\dot{\psi} = \frac{(Q \sin\phi + R \cos\phi)}{\cos\theta}$$

$$\dot{U} = RV - QW + \frac{1}{m} F_x \qquad \dot{V} = -RU + PW + \frac{1}{m} F_y$$

$$\dot{W} = QU - PV + \frac{1}{m} F_z$$

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\phi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \begin{pmatrix} U \\ V \\ W \end{pmatrix} \tag{1}$$

where we use the shorthand notation $s_z \equiv \sin(z)$, and $c_z \equiv \cos(z)$, $\forall z = \phi, \theta, \psi$. Further details can be found elsewhere.[12,13]

B. Actuator Model

The missile system has four tail fins (Fig. 1) intermixed to produce the effective aileron, elevator, and rudder deflection angles

$$d_p = \tfrac{1}{4}(-\delta_1 - \delta_2 + \delta_3 + \delta_4), \qquad d_q = \tfrac{1}{4}(\delta_1 + \delta_2 + \delta_3 + \delta_4)$$
$$d_r = \tfrac{1}{4}(-\delta_1 + \delta_2 - \delta_3 + \delta_4) \tag{2}$$

Each fin actuator has a finite bandwidth so that, for simplicity, the action of the effective deflection angles can be modeled by a first-order system with surface position saturation and rate saturation as follows:

$$\dot{d}_p = -w_a d_p + w_a u_p, \qquad \dot{d}_q = -w_a d_q + w_a u_q$$
$$\dot{d}_r = -w_a d_r + w_a u_r \tag{3}$$

C. Aerodynamic Model

The force exerted on an aircraft in flight consists of the aerodynamic force, thrust, and gravitational forces, and the aerodynamic moment is created by the aerodynamic load distribution. The components of the external force and moment are

$$F_x = \bar{F}_x + T + F_{gx}, \qquad F_y = \bar{F}_y + F_{gy}$$
$$F_z = \bar{F}_z + F_{gz}, \qquad M_x = C_{m_x} Q_s S_{\text{ref}} \bar{c} \tag{4}$$
$$M_y = C_{m_y} Q_s S_{\text{ref}} \bar{c}, \qquad M_z = C_{m_z} Q_s S_{\text{ref}} \bar{c}$$

where the gravitational forces and the aerodynamic forces are defined by

$$F_{gx} \equiv -mg \sin\theta, \qquad F_{gy} \equiv mg \cos\theta \sin\phi$$
$$F_{gz} \equiv mg \cos\theta \cos\phi, \qquad \bar{F}_x \equiv C_{f_x} Q_s S_{\text{ref}} \tag{5}$$
$$\bar{F}_y \equiv C_{f_y} Q_s S_{\text{ref}}, \qquad \bar{F}_z \equiv C_{f_z} Q_s S_{\text{ref}}$$

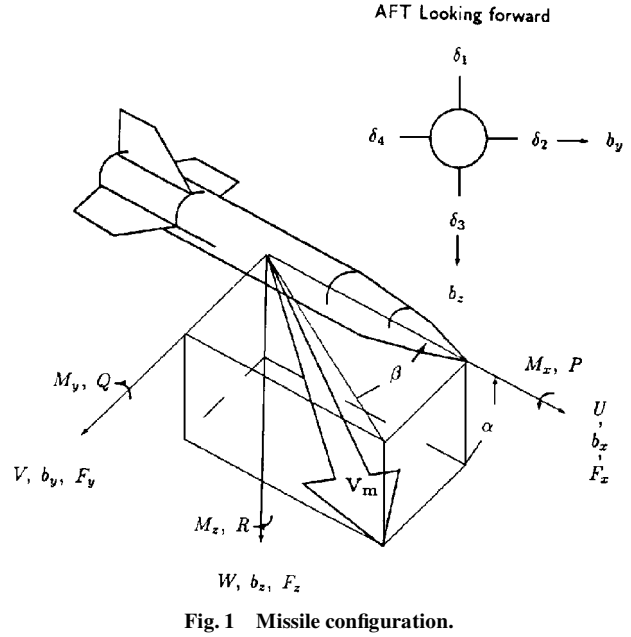

Fig. 1 Missile configuration.

where

$$Q_s = \tfrac{1}{2}\rho V_m^2 \qquad \text{and} \qquad V_m^2 = U^2 + V^2 + W^2 \tag{6}$$

and the model of all aerodynamic force coefficients C_f and the aerodynamic moment coefficients C_m can be expressed as follows:

$$\begin{pmatrix} C_{f_x} \\ C_{f_y} \\ C_{f_z} \end{pmatrix} = \begin{pmatrix} C_1 \\ C_4 \\ C_7 \end{pmatrix} + \begin{pmatrix} 0 & C_2 & C_3 \\ C_5 & 0 & C_6 \\ C_8 & C_9 & 0 \end{pmatrix} \begin{pmatrix} d_p \\ d_q \\ d_r \end{pmatrix} \tag{7}$$

$$\begin{pmatrix} C_{m_x} \\ C_{m_y} \\ C_{m_z} \end{pmatrix} = \begin{pmatrix} C_{10} \\ C_{13} \\ C_{16} \end{pmatrix} + \begin{pmatrix} C_{11} & 0 & C_{12} \\ C_{14} & C_{15} & 0 \\ C_{17} & 0 & C_{18} \end{pmatrix} \begin{pmatrix} d_p \\ d_q \\ d_r \end{pmatrix} \tag{8}$$

where the C_i terms are functions of the attack angle $\alpha \equiv \tan^{-1}(W/U)$, sideslip angle $\beta \equiv \tan^{-1}(V/U)$, Mach number $M_\infty \equiv V_m/a$, body angular velocities P, Q, R, missile altitude $-Z$, and missile velocity V_m and can be calculated via linear interpolation of the tabular data.

In the analysis and design problems concerning missile control, the vehicle body-axis accelerations constitute the set of most commonly adopted observation variables and are defined by

$$A_x = F_x/m, \qquad A_y = F_y/m, \qquad A_z = F_z/m \tag{9}$$

In contrast, the equations concerning the output measurement of the body accelerometers (or the aerodynamic accelerations) mounted at the vehicle center of gravity are

$$A_{nx} = \bar{F}_x/m, \qquad A_{ny} = \bar{F}_y/m, \qquad A_{nz} = \bar{F}_z/m, \qquad A_n = -A_{nz} \tag{10}$$

III. Feedforward Neural Networks

In a neural network, simple nonlinear elements, called neurons, are interconnected, and the strengths of the interconnections are denoted by adjustable parameters called neuron weights. A three-layer feedforward neural network is used to design a robust adaptive neural controller. The structure of the neural network, shown in Fig. 2, performs as an approximator described in matrix form as

$$\hat{h}_n(\bar{x}, \hat{W}_h, \hat{V}_h) = \hat{W}_h \sigma(\hat{V}_h \bar{x}_a) \tag{11}$$

where $\hat{W}_h \in \mathcal{R}^{m \times p}$ and $\hat{V}_h \in \mathcal{R}^{p \times (n+1)}$ are the output-hidden weight matrix and hidden-input weight matrix, respectively; $\bar{x} \in \mathcal{R}^{n \times 1}$ is the input vector; $\bar{x}_a \equiv (\bar{x}^T, -1)^T \in \mathcal{R}^{(n+1) \times 1}$ is the augmented neural input vector (the -1 term denotes the input bias),

$$\sigma_i\left(\hat{V}_{h_i}\bar{x}_a\right) \equiv \frac{1}{1 + \exp\left(-\hat{V}_{h_i}\bar{x}_a\right)} \in \mathcal{R}, \qquad i = 1, \ldots, p \tag{12}$$

is a sigmoid function, and

$$\sigma(\hat{V}_h \bar{x}_a) \equiv \begin{pmatrix} \sigma_1(\hat{V}_{h_1}\bar{x}_a) \\ \vdots \\ \sigma_p\left(\hat{V}_{h_p}\bar{x}_a\right) \end{pmatrix}, \qquad \text{with} \qquad \hat{V}_h \equiv \begin{pmatrix} \hat{V}_{h_1} \\ \vdots \\ \hat{V}_{h_p} \end{pmatrix} \tag{13}$$

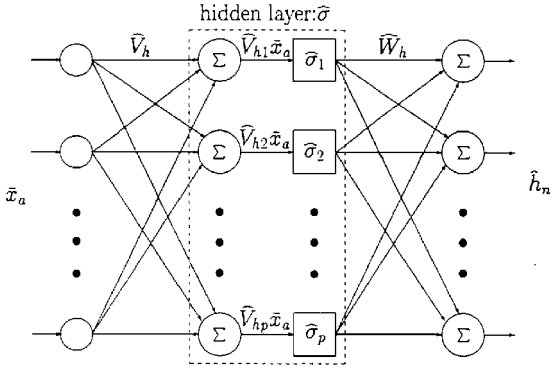

Fig. 2 Multilayer neural network structure.

where $\hat{V}_h$ includes the threshold, $\sigma_i(\hat{V}_{h_i}\bar{x}_a) \equiv \hat{\sigma}_i$ for each $i = 1, \ldots, p$ and $\sigma(\hat{V}_h\bar{x}_a) \equiv \hat{\sigma}$.

Artificial neural networks are promising tools for control applications because multilayer feedforward networks with as few as one hidden layer can approximate any well-behaved nonlinear function to any desired accuracy.[14,15] This universal approximation theory is stated formally in the following theorem.

Theorem 1. Let $\bar{x} \in \mathcal{D}$ (a compact subset of $\mathcal{R}^n$), $\boldsymbol{h}(\bar{x})$: $\mathcal{D} \to \mathcal{R}^m$ be a continuous-function-vector. For an arbitrary constant $\epsilon > 0$, there exists an integer p (the number of hidden neurons) and real constant optimal weight matrices $\boldsymbol{W}_h \in \mathcal{R}^{m \times p}$ and $\boldsymbol{V}_h \in \mathcal{R}^{p \times (n+1)}$, such that

$$\boldsymbol{h}(\bar{x}) = \boldsymbol{h}_n(\bar{x}, \boldsymbol{W}_h, \boldsymbol{V}_h) + \boldsymbol{\epsilon}_h(\bar{x}) \tag{14}$$

where $\boldsymbol{\epsilon}_h(\bar{x})$ is the approximation error vector satisfying $\|\boldsymbol{\epsilon}_h(\bar{x})\| \le \epsilon$, $\forall\, \bar{x} \in \mathcal{D}$. The optimal approximator can be described as

$$h_n(\bar{x}, \boldsymbol{W}_h, \boldsymbol{V}_h) \equiv \boldsymbol{W}_h \sigma(\boldsymbol{V}_h \bar{x}_a) \tag{15}$$

where $\|\cdot\|$ denotes the 2-norm, and $\bar{x}_a \equiv (\bar{x}^T, -1)^T$.

Proof: See Ref. 15.

Remark 1: Given the result of Theorem 1, there exist $\bar{W}_h$ and $\bar{V}_h$ as $\bar{x} \in \mathcal{D}$ such that

$$\|\boldsymbol{W}_h\|_F \le \bar{W}_h \in \mathcal{R}_+ \tag{16}$$

$$\|\boldsymbol{V}_h\|_F \le \bar{V}_h \in \mathcal{R}_+ \tag{17}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\|\boldsymbol{W}_h\|_F^2 \equiv \text{tr}\{\boldsymbol{W}_h^T \boldsymbol{W}_h\}$, $\|\boldsymbol{V}_h\|_F^2 \equiv \text{tr}\{\boldsymbol{V}_h^T \boldsymbol{V}_h\}$.

In general, study of the stability of a multilayer neural-network-based control system is difficult because the corresponding dynamics are nonlinear in adjustable neural network weights. Such a structure generally is not suitable for adaptive control. Fortunately, the approximation error denoted by $\tilde{h}_n$ can be expressed in a linearly parameterized form modulo a residual term. This is stated in the following lemma.

Lemma 1. Define the estimation errors of the weight matrix as

$$\tilde{\boldsymbol{W}}_h \equiv \boldsymbol{W}_h - \hat{\boldsymbol{W}}_h, \qquad \tilde{\boldsymbol{V}}_h \equiv \boldsymbol{V}_h - \hat{\boldsymbol{V}}_h \tag{18}$$

and the estimation error of the hidden-layer vector as $\tilde{\boldsymbol{\sigma}} \equiv \boldsymbol{\sigma}(\boldsymbol{V}_h\bar{x}_a) - \boldsymbol{\sigma}(\hat{\boldsymbol{V}}_h\bar{x}_a) \equiv \boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}$; then, the function approximation error $\boldsymbol{h}_n$ is

$$\begin{aligned} \tilde{\boldsymbol{h}}_n &\equiv \boldsymbol{h}(\bar{x}) - \hat{\boldsymbol{h}}_n(\bar{x}, \hat{\boldsymbol{W}}_h, \hat{\boldsymbol{V}}_h) \\ &= \tilde{\boldsymbol{W}}_h(\hat{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}'\hat{\boldsymbol{V}}_h\bar{x}_a) + \hat{\boldsymbol{W}}_h\hat{\boldsymbol{\sigma}}'\tilde{\boldsymbol{V}}_h\bar{x}_a + \boldsymbol{d}_h \end{aligned} \tag{19}$$

where the residual term $\boldsymbol{d}_h$ can be expressed as

$$\boldsymbol{d}_h = \tilde{\boldsymbol{W}}_h\hat{\boldsymbol{\sigma}}'\boldsymbol{V}_h\bar{x}_a + \boldsymbol{W}_h\boldsymbol{o}(\tilde{\boldsymbol{V}}_h\bar{x}_a) + \boldsymbol{\epsilon}_h(\bar{x}) \tag{20}$$

with $\boldsymbol{o}(\cdot)$ representing a sum of the high-order terms of the argument in a Taylor-series expansion.

Proof: To obtain the linear parameterization structure shown above, we manipulate the following equalities:

$$\begin{aligned} \tilde{\boldsymbol{h}}_n &= \boldsymbol{W}_h\boldsymbol{\sigma}(\boldsymbol{V}_h\bar{x}_a) - \hat{\boldsymbol{W}}_h\boldsymbol{\sigma}(\hat{\boldsymbol{V}}_h\bar{x}_a) + \boldsymbol{\epsilon}_h(\bar{x}) \\ &= \tilde{\boldsymbol{W}}_h\tilde{\boldsymbol{\sigma}} + \hat{\boldsymbol{W}}_h\tilde{\boldsymbol{\sigma}} + \tilde{\boldsymbol{W}}_h\hat{\boldsymbol{\sigma}} + \boldsymbol{\epsilon}_h(\bar{x}) \end{aligned} \tag{21}$$

To deal with $\tilde{\boldsymbol{\sigma}}$, we use the Taylor-series expansion

$$\begin{aligned} \boldsymbol{\sigma}(\boldsymbol{V}_h\bar{x}_a) &= \boldsymbol{\sigma}(\hat{\boldsymbol{V}}_h\bar{x}_a) + \boldsymbol{\sigma}'(\hat{\boldsymbol{V}}_h\bar{x}_a)(\boldsymbol{V}_h\bar{x}_a - \hat{\boldsymbol{V}}_h\bar{x}_a) + \boldsymbol{o}(\tilde{\boldsymbol{V}}_h\bar{x}_a) \\ &= \hat{\boldsymbol{\sigma}} + \hat{\boldsymbol{\sigma}}'\tilde{\boldsymbol{V}}_h\bar{x}_a + \boldsymbol{o}(\tilde{\boldsymbol{V}}_h\bar{x}_a) \end{aligned} \tag{22}$$

$$\Rightarrow \tilde{\boldsymbol{\sigma}} = \hat{\boldsymbol{\sigma}}'\tilde{\boldsymbol{V}}_h\bar{x}_a + \boldsymbol{o}(\tilde{\boldsymbol{V}}_h\bar{x}_a) \tag{23}$$

where $\boldsymbol{o}(\cdot)$ denotes the sum of the high-order terms of the argument in a Taylor-series expansion and

$$\hat{\sigma}_i' \equiv \sigma_i'\left(\hat{\boldsymbol{V}}_{h_i}\bar{x}_a\right) = \left.\frac{\mathrm{d}\sigma_i(Z)}{\mathrm{d}Z}\right|_{Z = \hat{V}_{h_i}\bar{x}_a}, \qquad \forall\, i = 1, \ldots, p \tag{24}$$

$$\hat{\boldsymbol{\sigma}}' = \text{diag}\{\hat{\sigma}_1', \ldots, \hat{\sigma}_p'\} \in \mathcal{R}^{p \times p} \tag{25}$$

Now, we can rewrite $\tilde{\boldsymbol{h}}_n$ as

$$\begin{aligned}\tilde{\boldsymbol{h}}_n &= \tilde{\boldsymbol{W}}_h[\hat{\boldsymbol{\sigma}}'\tilde{\boldsymbol{V}}_h\bar{\boldsymbol{x}}_a + \boldsymbol{o}(\tilde{\boldsymbol{V}}_h\bar{\boldsymbol{x}}_a)] \\ &\quad + \hat{\boldsymbol{W}}_h[\hat{\boldsymbol{\sigma}}'\tilde{\boldsymbol{V}}_h\bar{\boldsymbol{x}}_a + \boldsymbol{o}(\tilde{\boldsymbol{V}}_h\bar{\boldsymbol{x}}_a)] + \tilde{\boldsymbol{W}}_h\hat{\boldsymbol{\sigma}} + \boldsymbol{\epsilon}_h(\bar{\boldsymbol{x}}) \\ &= \tilde{\boldsymbol{W}}_h(\hat{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}'\hat{\boldsymbol{V}}_h\bar{\boldsymbol{x}}_a) + \hat{\boldsymbol{W}}_h\hat{\boldsymbol{\sigma}}'\tilde{\boldsymbol{V}}_h\bar{\boldsymbol{x}}_a \\ &\quad + \underbrace{\tilde{\boldsymbol{W}}_h\hat{\boldsymbol{\sigma}}'\boldsymbol{V}_h\bar{\boldsymbol{x}}_a + \boldsymbol{W}_h\boldsymbol{o}(\tilde{\boldsymbol{V}}_h\bar{\boldsymbol{x}}_a) + \boldsymbol{\epsilon}_h(\bar{\boldsymbol{x}})}_{\boldsymbol{d}_h} \end{aligned} \tag{26}$$

which is exactly the claim of this lemma. This, hence, completes the proof. QED

Lemma 2. The residual term $\boldsymbol{d}_h$ can be bounded by a linear-in-parameter function, i.e.,

$$\|\boldsymbol{d}_h\| < \boldsymbol{\alpha}_h^T\boldsymbol{Y}_\alpha \tag{27}$$

where the unknown parameter vector $\boldsymbol{\alpha}_h \in \mathcal{R}^{4\times 1}$ is composed of optimal weight matrices and some bounded constants and the known function vector $\boldsymbol{Y}_\alpha$ is defined as

$$\boldsymbol{Y}_\alpha = (1,\ \|\bar{\boldsymbol{x}}_a\|,\ \|\bar{\boldsymbol{x}}_a\|\|\hat{\boldsymbol{W}}_h\|_F,\ \|\bar{\boldsymbol{x}}_a\|\|\hat{\boldsymbol{V}}_h\|_F)^T \tag{28}$$

Proof: For the sigmoidal activation function, the high-order terms are bounded by

$$\begin{aligned}\|\boldsymbol{o}(\tilde{\boldsymbol{V}}_h\bar{\boldsymbol{x}}_a)\| &= \|\tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}\tilde{\boldsymbol{V}}_h\bar{\boldsymbol{x}}_a\| \\ &\le \|\tilde{\boldsymbol{\sigma}}\| + \|\hat{\boldsymbol{\sigma}}'\|\|\tilde{\boldsymbol{V}}_h\|_F\|\bar{\boldsymbol{x}}_a\| \\ &\le c_1 + c_2\|\tilde{\boldsymbol{V}}_h\|_F\|\bar{\boldsymbol{x}}_a\| \end{aligned} \tag{29}$$

where c_1, c_2 are some bounded constants because the sigmoid function and its derivative are always bounded by constants. Then, we are ready to state that $\boldsymbol{d}_h$ is bounded by

$$\begin{aligned}\|\boldsymbol{d}_h\| &= \|\tilde{\boldsymbol{W}}_h\hat{\boldsymbol{\sigma}}'\boldsymbol{V}_h\bar{\boldsymbol{x}}_a + \boldsymbol{W}_h\boldsymbol{o}(\tilde{\boldsymbol{V}}_h\bar{\boldsymbol{x}}_a) + \boldsymbol{\epsilon}_h(\bar{\boldsymbol{x}})\| \\ &\le \|\tilde{\boldsymbol{W}}_h\|_F\|\hat{\boldsymbol{\sigma}}'\|_F\|\boldsymbol{V}_h\|_F\|\bar{\boldsymbol{x}}_a\| + \|\boldsymbol{W}_h\|_F(c_1 + c_2\|\tilde{\boldsymbol{V}}_h\|_F\|\bar{\boldsymbol{x}}_a\|) + \epsilon \\ &\le \|\tilde{\boldsymbol{W}}_h\|_F c_2\bar{V}_h\|\bar{\boldsymbol{x}}_a\| + \bar{W}_h(c_1 + c_2\|\tilde{\boldsymbol{V}}_h\|_F\|\bar{\boldsymbol{x}}_a\|) + \epsilon \\ &\le (c_1\bar{W}_h + \epsilon) + c_2\bar{V}_h(\bar{W}_h + \|\hat{\boldsymbol{W}}_h\|_F)\|\bar{\boldsymbol{x}}_a\| \\ &\quad + c_2\bar{W}_h(\bar{V}_h + \|\hat{\boldsymbol{V}}_h\|_F)\|\bar{\boldsymbol{x}}_a\| \\ &\le \underbrace{(c_1\bar{W}_h + \epsilon)}_{\alpha_{h0}} + \underbrace{(c_2\bar{V}_h\bar{W}_h + c_2\bar{W}_h\bar{V}_h)}_{\alpha_{h1}}\|\bar{\boldsymbol{x}}_a\| \\ &\quad + \underbrace{c_2\ \bar{V}_h}_{\alpha_{h2}}\|\hat{\boldsymbol{W}}_h\|_F\|\bar{\boldsymbol{x}}_a\| + \underbrace{c_2\ \bar{W}_h}_{\alpha_{h3}}\|\hat{\boldsymbol{V}}_h\|_F\|\bar{\boldsymbol{x}}_a\| \\ &= (\alpha_{h0}\quad \alpha_{h1}\quad \alpha_{h2}\quad \alpha_{h3})\begin{pmatrix} 1 \\ \|\bar{\boldsymbol{x}}_a\| \\ \|\bar{\boldsymbol{x}}_a\|\|\hat{\boldsymbol{W}}_h\|_F \\ \|\bar{\boldsymbol{x}}_a\|\|\hat{\boldsymbol{V}}_h\|_F \end{pmatrix} \equiv \boldsymbol{\alpha}_h^T\boldsymbol{Y}_\alpha \end{aligned} \tag{30}$$

where we have used the following facts:

$$\|\tilde{\boldsymbol{W}}_h\|_F \le \bar{W}_h + \|\hat{\boldsymbol{W}}_h\|_F, \qquad \|\tilde{\boldsymbol{V}}_h\|_F \le \bar{V}_h + \|\hat{\boldsymbol{V}}_h\|_F \tag{31}$$

This concludes the proof of this lemma. QED

So far, these two lemmas allow us to approximate any continuous-function vector over a compact domain via a neural network architecture as shown in Fig. 2. Their difference can be shown to depend linearly on the errors of the neural weights (i.e., the difference between the weight estimates and the optimal weights) and a residual term that can be upper bounded by some linearly parameterizable term (i.e., an inner product of an unavailable parameter vector and some known function vectors). These results, hence, will help us to apply a robust adaptive control technique and adaptive sliding mode control to overcome the difficulty caused by the approximation process.

IV. Autopilot Design

The control goal of a missile autopilot system is, normally, to drive the missile to track acceleration commands generated by a guidance/navigation system, whose objective is in fact to produce a certain desired motion pattern, including turning. To achieve tracking of the acceleration command, the autopilot has to develop both a side force and a normal force on the missile body. If a missile autopilot exercises a turning motion by means of a side force (in the presence of sideslip), it is called an STT missile. On the other hand, the BTT steering scheme first involves rolling (banking) to a certain angle such that the plane of the maximum aerodynamic normal force is in the desired direction. Next, the magnitude of the normal force can be controlled by adjusting the angle of attack. A BTT missile should maintain a minimal sideslip angle and positive angle of attack as ramjet propulsion technology is applied.

Such a BTT autopilot strategy gives the missile relatively high maneuverability. However, this favorable property means not only high aerodynamic normal acceleration, but also the ability to change the orientation of acceleration rapidly by means of a considerably large roll rate. However, a high roll rate will induce cross coupling, resulting in undesirable pitch and yaw motion. To deal with the effect of cross coupling, the autopilot design must treat the BTT missile as a multi-input/multi-output (MIMO) system, whereas in typical STT technology, this cross coupling is negligible, and the autopilot can be designed with independent pitch, yaw, and roll channels. In the literature, there exist several logical autopilot command options used to achieve the BTT steering policy.[2,3,16,17] These different options nevertheless have the same purpose and are summarized in Table 1 for clarity.[2,3,16–18]

However, output assignment (A_y, A_z) unfortunately leaves the system with an undesirable nonminimum phase phenomenon. To alleviate this problem, the output redefinition has been investigated carefully.[2,6] On the basis of the work of Lian et al.,[2] the output of the BTT system is changed to the following set of variables:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \equiv \begin{pmatrix} \phi \\ V \\ W \end{pmatrix} \tag{32}$$

Note that the effect of the above output definition is somewhat similar to the set $(\phi,\ \beta,\ \alpha)^T$ because $\beta \equiv \tan^{-1}(V/U)$ and $\alpha \equiv \tan^{-1}(W/U)$.

The design procedure of the autopilot is summarized as follows.

Step 1) I/O feedback linearization.

From Eqs. (1), (3), and (32), the complete missile dynamic equations and output equation can be rewritten in the following compact form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \qquad \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{33}$$

with $\boldsymbol{x} = (P,\ Q,\ R,\ \phi,\ \theta,\ \psi,\ U,\ V,\ W,\ X,\ Y,\ Z,\ d_p,\ d_q,\ d_r)^T$ and $\boldsymbol{u} = (u_p,\ u_q,\ u_r)^T$.

Table 1 Options for autopilot output channels

Purpose	Output definition	Notation	Ref.
Certain roll angle (maximum changes ±180 deg)	ϕ	Bank angle	2, 17
	P	Roll rate	18
	$P_s \equiv P\cos\alpha + R\sin\alpha$	Roll rate with respect to the stability axis	3
Suitable normal force	α	Angle of attack	3, 17
	A_z	Normal acceleration along b_z body axis	16
	$\dot{W}$	Time derivative of velocity along b_z body axis	2
Minimal side force	β	Sideslip angle	3, 17, 18
	A_y	Side acceleration along b_y body axis	16
	$\dot{V}$	Time derivative of velocity along b_y body axis	2

It is well known that under certain structural conditions,[1] there exists a state diffeomorphism such that the nonlinear MIMO BTT missile system can be decoupled and linearized by a nonlinear state feedback control law. This condition is stated via the following assumption.

Assumption 1. The BTT missile system has a known strong relative-degree vector $(r_1, \ldots, r_m)$, i.e.,

$$r_i = \inf\{k | L_{g_j} L_f^{k-1}(h_i) \neq 0, \quad \text{for some } j, \quad 1 \leq j \leq 3\}, \quad i = 1, 2, 3 \tag{34}$$

where $L_f(\cdot)$, $L_{g_j}(\cdot)$ denote the Lie derivatives with respect to $\boldsymbol{f}$ and $\boldsymbol{g}_j$, respectively.

Hence, to derive the relative-degree vector of model (34), given the choice of output channels (33), we first arrive at the I/O relationship as follows:

$$\begin{pmatrix} y_1^{(r_1)} \\ y_2^{(r_2)} \\ y_3^{(r_3)} \end{pmatrix} \equiv \begin{pmatrix} \phi^{(3)} \\ \ddot{V} \\ \ddot{W} \end{pmatrix} = \boldsymbol{A}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} \tag{35}$$

and then we can directly read out that the relative-degree vector (r_1, r_2, r_3) is (3, 2, 2) by following the procedure, e.g., in Isidori.[1] Because the preceding equations only refer to the observable part of the transformed system equations, the characteristics of the unobservable part (or internal dynamics) must possess some desirable property in advance. Thus, some relevant system information is stated as follows.

Assumption 2. The zero dynamics are exponentially stable.

Step 2) Linear parameterization of $\boldsymbol{B}$.

If B is completely unknown, the problem becomes too complex to design a suitable controller here. Even if a time-consuming offline training scheme were applied to a neural-network-based controller, it would still require the availability of suitable and complete training patterns with respect to $\boldsymbol{B}(\boldsymbol{x})$. However, training that is almost perfect seems impossible because of the wide operation range and multivariable characteristics of a flight system. If the structure of $\boldsymbol{B}(\boldsymbol{x})$ can be expressed in a linear-in-parameter form, then the effort needed to design a neural-network-based controller will be drastically reduced. Thus, the intrinsic useful property is assumed and is stated in the following:

Assumption 3. The control gain matrix $\boldsymbol{B}$ satisfies the linear-in-parameters property, i.e.,

$$\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{\Theta}_B^T \boldsymbol{\omega}_B \tag{36}$$

where $\boldsymbol{\Theta}_B$ is the unknown parameter matrix and $\boldsymbol{\omega}_B$ is the regressor matrix.

On the basis of some modeling knowledge (refer to Sec. II), the linear-in-parameter form of $\boldsymbol{B}$ can be obtained and expressed in the following equation via Table 2:

$$\boldsymbol{B} \equiv \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \boldsymbol{\Theta}_B^T \boldsymbol{\omega}_B \tag{37}$$

Step 3) Specifying the sliding-surface variable.

Suppose the desired trajectories ϕ_d, V_d, W_d are sufficiently smooth; then, the output tracking error is defined as

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} \equiv \begin{pmatrix} \phi - \phi_d \\ V - V_d \\ W - W_d \end{pmatrix} \tag{38}$$

Therefore, we can define the sliding-surface vector as

$$\boldsymbol{S} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \end{pmatrix} \equiv \begin{pmatrix} \ddot{e}_1 + 2\lambda_1 \dot{e}_1 + \lambda_1^2 e_1 \\ \dot{e}_2 + \lambda_2 e_2 \\ \dot{e}_3 + \lambda_3 e_3 \end{pmatrix} \tag{39}$$

Table 2 Structure of the control gain matrix $\boldsymbol{B}$

Entry	Unknown parameter Θ_B	Known regressor ω_B
b_{11}	$= (0.5 \times w_a S_{\text{ref}} \bar{c} \rho C_{11}/I_{xx}) \times$	V_m^2
	$+ (0.5 \times w_a S_{\text{ref}} \bar{c} \rho C_{14}/I_{yy}) \times$	$V_m^2 \tan(\theta) \sin(\phi)$
	$+ (0.5 \times w_a S_{\text{ref}} \bar{c} \rho C_{17}/I_{zz}) \times$	$V_m^2 \tan(\theta) \cos(\phi)$
b_{12}	$= (0.5 \times w_a S_{\text{ref}} \bar{c} \rho C_{15}/I_{yy}) \times$	$V_m^2 \tan(\theta) \sin(\phi)$
b_{13}	$= (0.5 \times w_a S_{\text{ref}} \bar{c} \rho C_{12}/I_{xx}) \times$	V_m^2
	$+ (0.5 \times w_a S_{\text{ref}} \bar{c} \rho C_{18}/I_{zz}) \times$	$V_m^2 \tan(\theta) \cos(\phi)$
b_{21}	$= (0.5 \times w_a S_{\text{ref}} \rho C_5/m) \times$	V_m^2
b_{22}	$= 0$	
b_{23}	$= (0.5 \times w_a S_{\text{ref}} \rho C_6/m) \times$	V_m^2
b_{31}	$= (-0.5 \times w_a S_{\text{ref}} \rho C_8/m) \times$	V_m^2
b_{32}	$= (-0.5 \times w_a S_{\text{ref}} \rho C_9/m) \times$	V_m^2
b_{33}	$= 0$	

where $\lambda_i \; \forall \, i = 1, 2, 3$ are all positive constants. Then, the derivative of $\boldsymbol{S}$ is

$$\begin{aligned} \dot{\boldsymbol{S}} &= \begin{pmatrix} y_1^{(3)} \\ \ddot{y}_2 \\ \ddot{y}_3 \end{pmatrix} + \underbrace{\begin{pmatrix} -\phi_d^{(3)} & +2\lambda_1 \ddot{e}_1 & +\lambda_1^2 \dot{e}_1 \\ -\ddot{V}_d & +\lambda_2 \dot{e}_2 & \\ -\ddot{W}_d & +\lambda_3 \dot{e}_3 & \end{pmatrix}}_{A_r} \\ &= -\boldsymbol{\Lambda}\boldsymbol{S} + \underbrace{(\boldsymbol{\Lambda}\boldsymbol{S} + \boldsymbol{A} + \boldsymbol{A}_r)}_{\boldsymbol{\psi}(\bar{\boldsymbol{x}})} + \boldsymbol{B}\boldsymbol{u} \\ &= -\boldsymbol{\Lambda}\boldsymbol{S} + \boldsymbol{\psi} + \boldsymbol{B}\boldsymbol{u} \end{aligned} \tag{40}$$

where

$$\bar{\boldsymbol{x}} = \left(\boldsymbol{x}^T, \; e_1, \; \dot{e}_1, \; \ddot{e}_1, \; \phi_d^{(3)}, \; e_2, \; \dot{e}_2, \; \ddot{V}_d, \; e_3, \; \dot{e}_3, \; \ddot{W}_d\right)^T \in \mathcal{R}^{n' \times 1}$$

$$n' = n + \sum_{j=1}^{3} r_j + 3$$

and

$$\boldsymbol{\Lambda} = \text{diag}\{\Lambda_1, \ldots, \Lambda_m\}$$

is a positive definite matrix.

Remark 2. Employment of such a sliding-mode design arises from the desire to achieve output tracking regardless of the approximation error, namely, $\boldsymbol{d}_h$. In other words, the neural network tries to approximate the complex dynamics $\boldsymbol{\psi}(\bar{\boldsymbol{x}})$, whereas the sliding-mode control tries to increase the closed-loop system robustness.

Step 4) Adaptive robust neural-network-based controller structure.

The main idea of the controller is to apply a feedforward neural network to approximate the function $\boldsymbol{\psi}(\bar{\boldsymbol{x}})$ and a matrix

$$\hat{\boldsymbol{B}} \equiv \hat{\boldsymbol{\Theta}}_B^T \boldsymbol{\omega}_B \tag{41}$$

to estimate the original matrix $\boldsymbol{B}$. On the basis of Theorem 1, we know that there exists an optimal neural approximator $\boldsymbol{\psi}_n = \boldsymbol{W}_\psi \boldsymbol{\sigma}(\boldsymbol{V}_\psi \bar{\boldsymbol{x}}_a)$ over a properly defined compact domain, and we design a neural approximator $\hat{\boldsymbol{\psi}}_n(\hat{\boldsymbol{W}}_\psi, \hat{\boldsymbol{V}}_\psi, \bar{\boldsymbol{x}}_a)$ to model the function $\boldsymbol{\psi}(\bar{\boldsymbol{x}})$, given the estimates $\hat{\boldsymbol{W}}_\psi$ and $\hat{\boldsymbol{V}}_\psi$.

Then, the approximation error $\tilde{\boldsymbol{\psi}}_n$ is

$$\begin{aligned} \tilde{\boldsymbol{\psi}}_n(\hat{\boldsymbol{W}}_\psi, \hat{\boldsymbol{V}}_\psi, \bar{\boldsymbol{x}}_a) &\equiv \boldsymbol{\psi} - \hat{\boldsymbol{\psi}}_n \\ &= \tilde{\boldsymbol{W}}_\psi (\hat{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}' \hat{\boldsymbol{V}}_\psi \bar{\boldsymbol{x}}_a) + \hat{\boldsymbol{W}}_\psi \hat{\boldsymbol{\sigma}}' \tilde{\boldsymbol{V}}_\psi \bar{\boldsymbol{x}}_a \\ &\quad + \boldsymbol{d}_\psi \in \mathcal{R}^{3 \times 1} \end{aligned} \tag{42}$$

where $\bar{\boldsymbol{x}}_a \equiv (\bar{\boldsymbol{x}}^T, -1)^T$ is the augmented neural input vector, $\tilde{\boldsymbol{W}}_\psi \equiv \boldsymbol{W}_\psi - \hat{\boldsymbol{W}}_\psi$, $\tilde{\boldsymbol{V}}_\psi \equiv \boldsymbol{V}_\psi - \hat{\boldsymbol{V}}_\psi$ are the estimation errors of $\boldsymbol{W}_\psi$ and $\boldsymbol{V}_\psi$, respectively, and the residual term $\boldsymbol{d}_\psi \in \mathcal{R}^{3 \times 1}$ is bounded by

$$\begin{aligned} \|\boldsymbol{d}_\psi\| &\leq \boldsymbol{\alpha}_\psi^T (1, \|\bar{\boldsymbol{x}}_a\|, \|\bar{\boldsymbol{x}}_a\| \|\hat{\boldsymbol{W}}_\psi\|_F, \|\bar{\boldsymbol{x}}_a\| \|\hat{\boldsymbol{V}}_\psi\|_F)^T \\ &= \boldsymbol{\alpha}_\psi^T \boldsymbol{Y}_\alpha \end{aligned} \tag{43}$$

where $\boldsymbol{\alpha}_\psi \in \mathcal{R}^{4 \times 1}$ is an unknown parameter vector and $\boldsymbol{Y}_\alpha \in \mathcal{R}^{4 \times 1}$ is a known function vector.

According to the preceding steps, we can design an adaptive robust neural-network-based autopilot as follows.

Control law:

$$\begin{aligned} \boldsymbol{u} &= \hat{\boldsymbol{B}}^{-1}(-\hat{\psi}_n + \boldsymbol{v}) & \hat{\psi}_n &= \hat{\boldsymbol{W}}_\psi \boldsymbol{\sigma}(\hat{\boldsymbol{V}}_\psi) \\ \boldsymbol{v} &= -\text{sat}(\boldsymbol{S}/\varepsilon)\hat{\boldsymbol{\alpha}}_\psi^T \boldsymbol{Y}_\alpha & \hat{\boldsymbol{B}} &= \hat{\boldsymbol{\Theta}}_B^T \boldsymbol{\omega}_B \end{aligned} \tag{44}$$

where $\text{sat}(\boldsymbol{S}/\varepsilon) \equiv [\text{sat}(S_1/\varepsilon), \text{sat}(S_2/\varepsilon), \text{sat}(S_3/\varepsilon)]^T$, a linear range ε is used to avoid discontinuous control transitions, and $\text{sat}(\cdot)$ is the saturation function satisfying

$$\text{sat}(\chi) = \begin{cases} \chi & \text{if} \quad |\chi| \le 1 \\ \text{sgn}(\chi) & \text{otherwise} \end{cases} \tag{45}$$

Adaptive law:

$$\begin{aligned} \dot{\hat{\boldsymbol{W}}}_\psi^T &= \boldsymbol{\Gamma}_w(\hat{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}'\hat{\boldsymbol{V}}_\psi \bar{\boldsymbol{x}}_a)\boldsymbol{S}_\Delta^T & \dot{\hat{\boldsymbol{V}}}_\psi^T &= \boldsymbol{\Gamma}_v\left(\bar{\boldsymbol{x}}_a \boldsymbol{S}_\Delta^T \hat{\boldsymbol{W}}_\psi \hat{\boldsymbol{\sigma}}'\right) \\ \dot{\hat{\boldsymbol{\Theta}}}_B^T &= \boldsymbol{\Gamma}_B \boldsymbol{\omega}_B \boldsymbol{u} \boldsymbol{S}_\Delta^T & \dot{\hat{\boldsymbol{\alpha}}}_\psi &= \boldsymbol{\Gamma}_\alpha \|\boldsymbol{S}_\Delta\| \boldsymbol{Y}_\alpha \end{aligned} \tag{46}$$

where the adaptive gain matrices $\boldsymbol{\Gamma}_w$, $\boldsymbol{\Gamma}_v$, and $\boldsymbol{\Gamma}_\alpha$ are all symmetric and positive definite with

$$\boldsymbol{S}_\Delta = \begin{pmatrix} S_{\Delta_1} \\ S_{\Delta_2} \\ S_{\Delta_3} \end{pmatrix} \equiv \begin{bmatrix} S_1 - \varepsilon\ \text{sat}(S_1/\varepsilon) \\ S_2 - \varepsilon\ \text{sat}(S_2/\varepsilon) \\ S_3 - \varepsilon\ \text{sat}(S_3/\varepsilon) \end{bmatrix} \tag{47}$$

used to carry out a smooth adaptive law with a deadzone size of ε. Such a dead-zone technique often has been incorporated to deal with the phenomenon of parameter drift in the robust adaptive control theory.[19,20]

Figure 3 depicts the block diagram of the autopilot developed herein, and the effect of this BTT autopilot controller can be stated via the following theorem.

Theorem 2. Consider the nonlinear BTT missile system (33) satisfying Assumptions 1–3. Define the control law as given in Eq. (44) and the adaptive law in Eq. (46). If $\phi_d, \ldots, \phi_d^{(3)}$, $V_d, \ldots, \dot{V}_d$ and $W_d, \ldots, \ddot{W}_d$ are all bounded, then the tracking errors e_1, e_2, and e_3 defined in Eq. (38) will asymptotically converge to a neighborhood of zero, all with individual size of order ε, and where all adjustable parameters will remain bounded.

Proof: Choose a Lyapunov function $V = V_1 + V_2$, where

$$V_1 = \tfrac{1}{2}\boldsymbol{S}_\Delta^T \boldsymbol{S}_\Delta \tag{48}$$

$$\begin{aligned} V_2 = {} & \tfrac{1}{2}\text{tr}\left\{\tilde{\boldsymbol{W}}_\psi \boldsymbol{\Gamma}_w^{-1} \tilde{\boldsymbol{W}}_\psi^T\right\} + \tfrac{1}{2}\text{tr}\left\{\tilde{\boldsymbol{V}}_\psi \boldsymbol{\Gamma}_v^{-1} \tilde{\boldsymbol{V}}_\psi^T\right\} \\ & + \tfrac{1}{2}\text{tr}\left\{\tilde{\boldsymbol{\Theta}}_B \boldsymbol{\Gamma}_B^{-1} \tilde{\boldsymbol{\Theta}}_B^T\right\} + \tfrac{1}{2}\tilde{\boldsymbol{\alpha}}_\psi^T \boldsymbol{\Gamma}_\alpha^{-1} \tilde{\boldsymbol{\alpha}}_\psi \end{aligned} \tag{49}$$

Hence, the time derivative of V_1 can be derived as follows:

$$\begin{aligned} \dot{V}_1 &= \boldsymbol{S}_\Delta^T(-\boldsymbol{\Lambda}\boldsymbol{S} + \tilde{\psi}_n + \tilde{\boldsymbol{B}}\boldsymbol{u} + \boldsymbol{v}) \\ &= -\boldsymbol{S}_\Delta^T \boldsymbol{\Lambda} \boldsymbol{S}_\Delta - |S_\Delta|\varepsilon + \boldsymbol{S}_\Delta^T\{\tilde{\boldsymbol{W}}_\psi(\hat{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}'\hat{\boldsymbol{V}}_\psi \bar{\boldsymbol{x}}_a) \\ &\quad + \hat{\boldsymbol{W}}_\psi \hat{\boldsymbol{\sigma}}' \tilde{\boldsymbol{V}}_\psi \bar{\boldsymbol{x}}_a + \boldsymbol{d}_\psi\} + \boldsymbol{S}_\Delta^T \boldsymbol{v} + \boldsymbol{S}_\Delta^T \tilde{\boldsymbol{\Theta}}_B^T \boldsymbol{\omega}_B \boldsymbol{u} \\ &\le -\boldsymbol{S}_\Delta^T \boldsymbol{\Lambda} \boldsymbol{S}_\Delta + \text{tr}\left\{\tilde{\boldsymbol{W}}_\psi\ (\hat{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}'\hat{\boldsymbol{V}}_\psi \bar{\boldsymbol{x}}_a)\boldsymbol{S}_\Delta^T\right\} + \text{tr}\left\{\tilde{\boldsymbol{V}}_\psi \bar{\boldsymbol{x}}_a \boldsymbol{S}_\Delta^T \hat{\boldsymbol{W}}_\psi \hat{\boldsymbol{\sigma}}'\right\} \\ &\quad + \text{tr}\left\{\tilde{\boldsymbol{\Theta}}_B^T \boldsymbol{\omega}_B \boldsymbol{u} \boldsymbol{S}_\Delta^T\right\} + \|\boldsymbol{S}_\Delta\| \tilde{\boldsymbol{\alpha}}_\psi^T \boldsymbol{Y}_\alpha \end{aligned} \tag{50}$$

where $\tilde{\boldsymbol{B}} = \boldsymbol{B} - \hat{\boldsymbol{B}}$ and $\tilde{\psi}_n = \psi - \hat{\psi}_n$. Note that, in the preceding, we have investigated the situation where $|S_i| \ge \varepsilon$, $\forall\, i = 1, \ldots, m$, and hence $\dot{\boldsymbol{S}}_\Delta = \dot{\boldsymbol{S}}$. For other situations where some S_i satisfy $|S_i| < \varepsilon$, one only needs to modify the first term on the right-hand side of the preceding inequality as

$$-\sum_i \left(\boldsymbol{S}_\Delta^T \boldsymbol{\Lambda} \boldsymbol{S}_\Delta\right)_i$$

to obtain a new $\dot{V}_1$, where the summation solely includes the terms with $|S_i| \ge \varepsilon$.

Furthermore, we have used the following facts:

$$\text{If} \quad |S_i| \le \varepsilon, \quad \text{then} \quad S_{\Delta_i} = 0 = \dot{S}_{\Delta_i}, \qquad \forall\, i = 1, 2, 3 \tag{51}$$

$$\text{If} \quad |S_i| > \varepsilon, \quad \text{then} \quad |S_{\Delta_i}| = |S_i| - \varepsilon \quad \text{and} \quad \dot{S}_{\Delta_i} = \dot{S}_i \qquad \forall\, i = 1, 2, 3 \tag{52}$$

$$S_{\Delta_i}\ \text{sat}(S_i/\varepsilon) = |S_{\Delta_i}|, \qquad \forall\, i = 1, 2, 3 \tag{53}$$

In addition, the time derivative of V_2 is

$$\begin{aligned} \dot{V}_2 = {} & -\text{tr}\left\{\tilde{\boldsymbol{W}}_\psi \boldsymbol{\Gamma}_w^{-1} \dot{\tilde{\boldsymbol{W}}}_\psi^T\right\} - \text{tr}\left\{\tilde{\boldsymbol{V}}_\psi \boldsymbol{\Gamma}_v^{-1} \dot{\tilde{\boldsymbol{V}}}_\psi^T\right\} \\ & - \text{tr}\left\{\tilde{\boldsymbol{\Theta}}_B \boldsymbol{\Gamma}_B^{-1} \dot{\tilde{\boldsymbol{\Theta}}}_B^T\right\} - \tilde{\boldsymbol{\alpha}}_\psi^T \boldsymbol{\Gamma}_\alpha^{-1} \dot{\tilde{\boldsymbol{\alpha}}}_\psi \end{aligned} \tag{54}$$

Then, applying the adaptive law, we can obtain the following result:

$$\dot{V} \le -\boldsymbol{S}_\Delta^T \boldsymbol{\Lambda} \boldsymbol{S}_\Delta \qquad \text{or} \qquad \dot{V} \le -\sum_i \left(\boldsymbol{S}_\Delta^T \boldsymbol{\Lambda} \boldsymbol{S}_\Delta\right)_i \tag{55}$$

which obviously implies that $\boldsymbol{S}_\Delta$, $\tilde{\boldsymbol{W}}_\psi$, $\tilde{\boldsymbol{V}}_\psi$, and $\tilde{\boldsymbol{\alpha}}_\psi$ are all bounded. However, either one of the above two will imply that $\boldsymbol{S}_\Delta \in L_2$. According to the zero dynamics assumption and the boundedness of all adjustable parameters, it is easy to conclude that $\boldsymbol{Y}_\alpha$, $\boldsymbol{v}$, $\boldsymbol{u}$, and $\dot{\boldsymbol{S}}_\Delta$ are all bounded. Finally, the fact that $\boldsymbol{S}_\Delta \to 0$ as $t \to \infty$ can be concluded by applying Barbalat's Lemma; hence, asymptotic bounds on the actual tracking errors can be achieved. QED

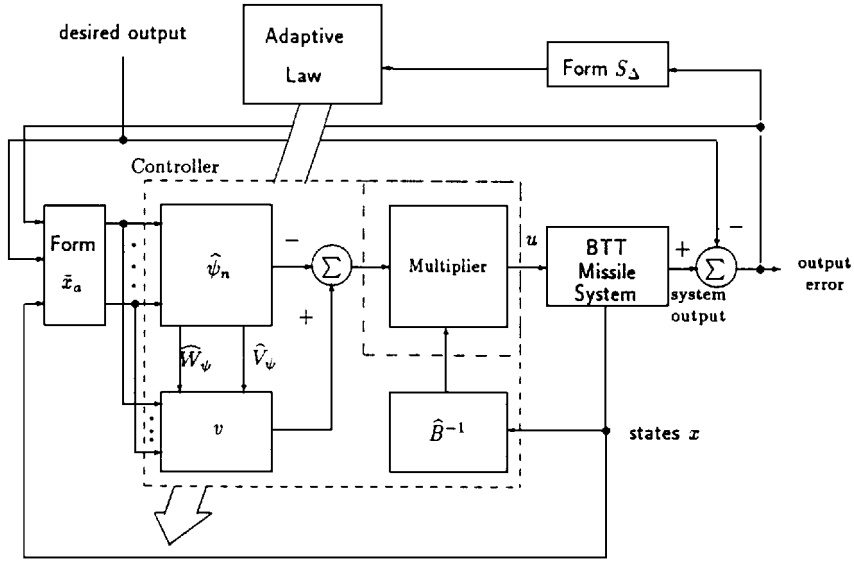

Fig. 3 BTT autopilot architecture.

Remark 3. Because $\hat{\boldsymbol{B}}$ may evolve to be singular, despite the fact that the true matrix $\boldsymbol{B}$ is always nonsingular, an auxiliary technique is necessary to ensure the existence of $\hat{\boldsymbol{B}}^{-1}$. This projection technique is used frequently and is introduced in the adaptive control context to keep the unknown parameters within a certain parameter range that guarantees that $\hat{\boldsymbol{B}}^{-1}$ exists. Because the unknown parameter matrix $\boldsymbol{\Theta}_B$ has physical meaning, this technique can be incorporated easily into the proposed controller, and a prior rough estimate of the range where $\hat{\boldsymbol{\Theta}}_B$ lies (hence, its estimate is invertible) is not difficult to obtain.

Remark 4. Note that the sliding-surface variable is only involved in the adaptive law, and that the neural-network-based sliding-mode control law $\boldsymbol{v}$ incorporates the neural approximator $\hat{\psi}_n$ so as to track the desired trajectories as quickly as possible. Such a controller structure can be quite feasible because the estimated sliding gain $\hat{\boldsymbol{\alpha}}_\psi^T \boldsymbol{Y}_\alpha$ is updated gradually and continuously to avoid undesirable system chattering, instead of directly employing a high sliding gain to overcome large modeling uncertainties or disturbance.

V. Computer Simulation

A. System Constraints

In the BTT autopilot computer simulations, the frequency bandwidth of the actuator, namely w_a, was 20π. Furthermore, the BTT missile was subjected to the following physical constraints:

limit on attack angle α: $\alpha \geq 0$ deg
limit on sideslip angle β: $|\beta| \leq 5$ deg
limit on pitch rate Q: $|Q| \leq 150$ deg/s
limit on yaw rate R: $|R| \leq 150$ deg/s
limit on roll rate P: $|P| \leq 500$ deg/s
actuator position saturation: $|d_p| \leq 21$ deg, $|d_q| \leq 21$ deg, $|d_r| \leq 21$ deg
actuator rate saturation: $|\dot{d}_p|$, $|\dot{d}_q|$, and $|\dot{d}_r| \leq 200$ deg/s

B. Definition of the Desired Output

The original desired ϕ command and A_z command were both square functions, and the two commands were asynchronous. Figure 4 depicts the relationship between them. The leading edge of the desired A_z command lags behind the leading edge of the desired ϕ command, and the lagging edge of the desired A_z command antedates the lagging edge of the desired ϕ command, both by Δ_t, which is equal to 0.5 s. Under this arrangement, the effect of the cross coupling can be reduced to a certain degree. Moreover, to obtain the desired smooth outputs, ϕ_d, A_{zd} are generated by means of the following equations:

$$\ddot{\phi}_d = -225\phi_d - 30\dot{\phi}_d + 225(\text{desired } \phi \text{ command}) \tag{56}$$

$$\dot{A}_{zd} = -10A_{zd} + 10.0(\text{desired } A_z \text{ command}) \tag{57}$$

where the desired ϕ command and A_z command are square functions.

In addition, if we choose $(\phi,\ A_y,\ A_z)$ as the system outputs, the system can very easily become unstable. On the basis of the groundwork of the output-redefinition system,[2] the difficulty with the undesirable nonminimum phase property was alleviated. We also found that the relationship between A_{zd} and W_d could be expressed in a simple fashion. Thus, we designed a proper W_d command:

$$\dot{W}_d = -\eta(A_{zd} - A_z) \tag{58}$$

where W_d is the redefined desired command, A_z is the actual output response, and $\eta(\approx 4.0)$ is a design parameter.

The response of A_y is always constrained to zero to eliminate the side force F_y. If the signal V [or sideslip angle $\beta = \tan^{-1}(V/U)$] is constrained to zero, then the same result can be attained. On this basis, we let the desired command V_d be zero. Because of this setup, it is found that, in fact, the signals $\phi_d^{(3)}$, $\ddot{W}_d$, and $\ddot{V}_d$ are not really needed to form $\bar{\boldsymbol{x}}_a$ (the input vector of the neural network); hence, they are deleted from $\bar{\boldsymbol{x}}_a$ to reduce the computation time.

Table 3 Setup parameters for the BTT missile autopilot

No. of inputs of neural network	21
No. of hidden nodes of neural network	25
No. of outputs of neural network	3
No. of total elements of $\hat{\Theta}_B$	10
No. of total elements of $\hat{\alpha}_\psi$	4
Total adjustable parameters	$614 = 21 \times 25 + 25 \times 3 + 10 + 4$
Sliding surface S_1	$\ddot{e}_1 + 25\,\dot{e}_1 + 10\,e_1$
Sliding surface S_2	$\dot{e}_2 + 2\,e_2$
Sliding surface S_3	$\dot{e}_3 + 2\,e_3$
Dead-zone size ε	0.5
Adaptive gain Γ_w with respect to ϕ	diag{400, . . . , 400}
Adaptive gain Γ_w with respect to V	diag{25, . . . , 25}
Adaptive gain Γ_w with respect to W	diag{200, . . . , 200}
Adaptive gain Γ_v	diag{0.25, . . . , 0.25}
Adaptive gain Γ_α	diag{2, 2, 0.02, 2}

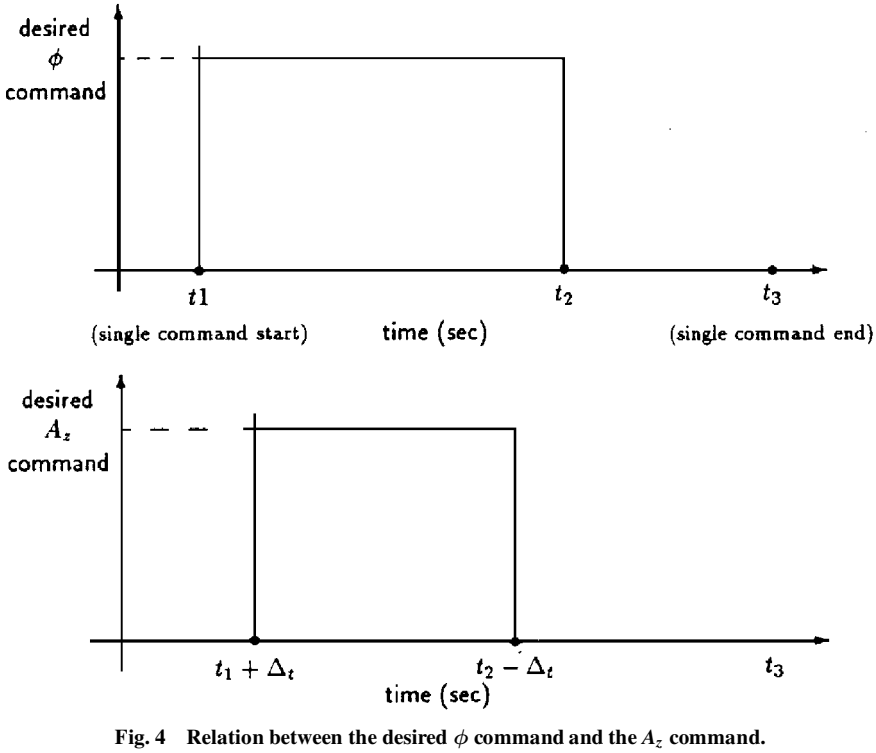

Fig. 4 Relation between the desired ϕ command and the A_z command.

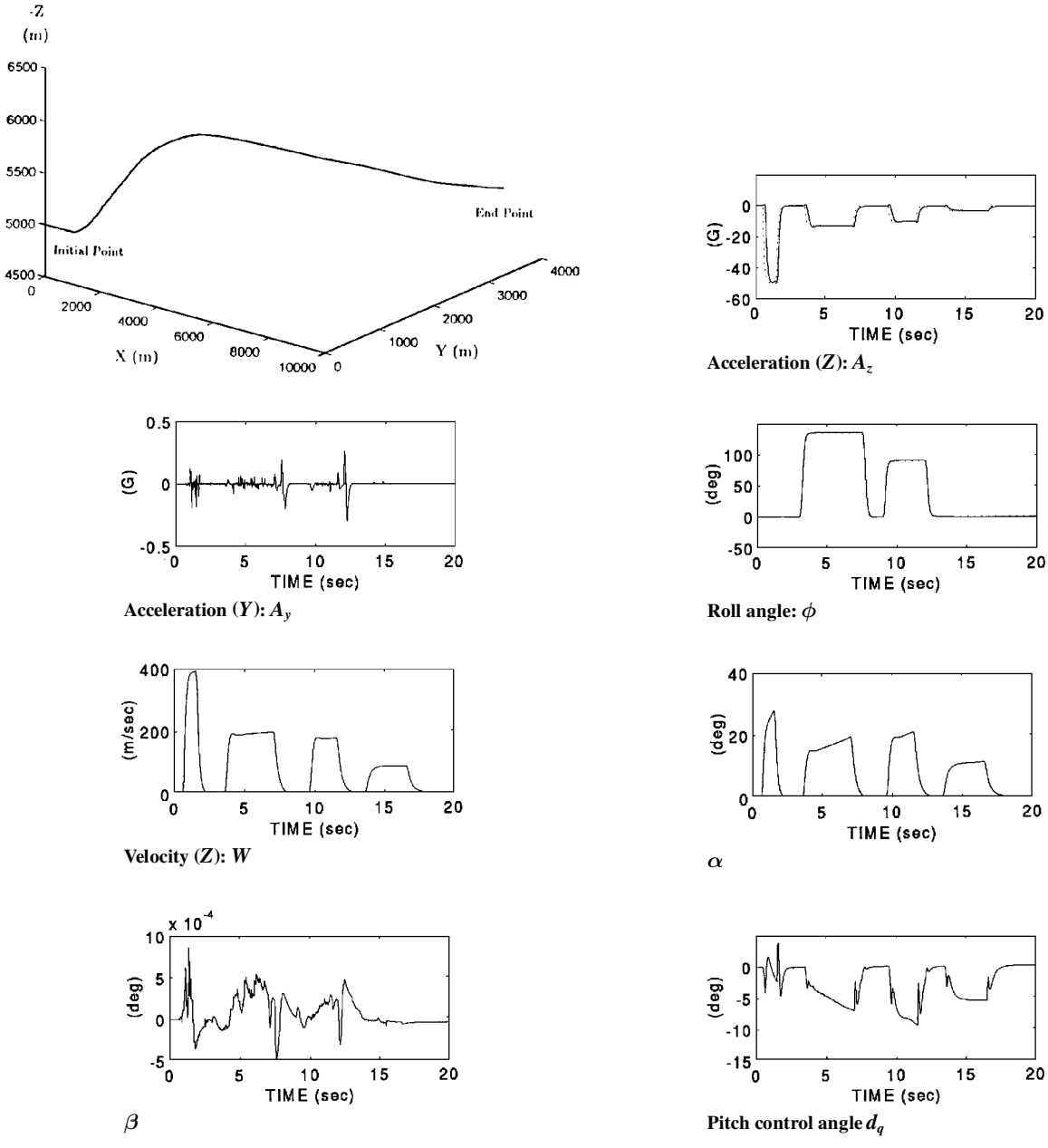

Fig. 5 BTT simulation results.

C. Controller Setup

The design procedure and the detailed definitions for the BTT autopilot are shown in Sec. IV. Now, we give the values of the design parameters of the controller that was used in all simulations, and these design parameters are listed on Table 3. The overall weights of a neural network are initially set to zero. Note that only one single MIMO neural network is used in the autopilot design, and that the adaptive gain Γ_w is separated into three parts with respect to three different output channels. Furthermore, in the BTT missile model, we can find that Z is involved in the aerodynamic model, but that the states X, Y are used only to express the position of a missile. Because X and Y are not involved in the ϕ, V, or W channel, the vector function $\boldsymbol{A}(\cdot)$ is not related to these two states, and so they are excluded from $\bar{\boldsymbol{x}}_a$ as well. Therefore, the total number of neural network inputs is 21.

D. Simulation Results

In the following simulation, the thrust T of the missile was set to zero. The initial flight condition of the missile and the command set for the continuous flight were as follows.

Initial conditions:

$$U = 1020\ \text{m/s}, \qquad W = -0.01\ \text{m/s}, \qquad Z = -5000\ \text{m}$$

and the other states were all set at zero.

Command set:

Command stage	Time, s			Desired command		
	t_1	t_2	t_3	ϕ_d, deg	A_{yd}, g	A_{zd}, g
(1)	0.0	2.0	3.0	0	0.0	−50.0
(2)	3.0	7.5	9.0	135	0.0	−13.0
(3)	9.0	12.0	13.0	90	0.0	−10.0
(4)	13.0	17.0	20.0	0	0.0	−3.0

The detailed descriptions of t_1, t_2, and t_3 can be found in Fig. 4, and the simulation results are shown in Fig. 5.

In the preceding case, we used one controller in the continuous flight simulation, and the setup parameters of this controller were

identical. No tedious prior training was necessary. One can observe that the system outputs tracked the command (ϕ_d, A_{yd}, A_{zd}) and the redefined command (V_d, W_d) quite well, that the value of the sideslip angle β ($<10^{-3}$ deg) was kept very small, and that the states of the missile did not violate the physical limitations.

VI. Conclusions

An on-line tuning multilayer neural-network-basedautopilot has been developed for a BTT missile. The Lyapunov theory is used to complete the closed-loop stability proof. This scheme combines neural networks and the sliding-mode control technique. The former has been applied to model some unknown nonlinear functions, whereas the latter has been used to overcome some modeling residual terms. Especially, the proposed controller has the following features.

1) The controller design never requires a prior time-consuming training phase or unrealistic information about the optimal neural network (e.g., the upper bound on the norm of the optimal weight matrices).

2) The sliding-mode controller employs signals from the neural network, but the parameters of the sliding-mode controller are updated in an online manner so as to avoid excessively high control efforts.

3) Only a single neural network is used to design the autopilot, and the controller no longer requires use of coarse or inaccurate tabular data regarding the aerodynamics of the atmospheric coefficients.

Acknowledgment

This research is sponsored by National Science Council, Republic of China, under Grant NSC 85-2213-E-002-040.

References

[1]Isidori, A., *Nonlinear Control Systems*, Springer–Verlag, 1989.

[2]Lian, K.-Y., Fu, L.-C., Chung, D.-M., and Kuo, T.-S., "Adaptive Robust Autopilot Design for Bank-to-Turn Aircraft," *Proceedings of American Control Conference*, 1993, pp. 1746–1750.

[3]Huang, J., and Lin, C.-F., "Sliding Mode Control of HAVE DASH II Missile Systems," *Proceedings of American Control Conference*, 1993, pp. 183–187.

[4]Lane, S. H., and Stengel, R. F., "Flight Control Design Using Non-Linear Inverse Dynamics," *Automatica*, Vol. 24, No. 4, 1988, pp. 478–483.

[5]Romano, J. J., and Singh, S. N., "I-O Map Inversion, Zero Dynamics and Flight Control," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 26, No. 6, 1990, pp. 1022–1029.

[6]Hedrick, J. K., and Gopalswamy, S., "Nonlinear Flight Control Design via Sliding Methods," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 5, 1990, pp. 850–858.

[7]Sanner, R. M., and Slotine, J. J. E., "Gaussian Network for Direct Adaptive Control," *IEEE Transactions on Neural Networks*, Vol. 3, No. 6, 1992, pp. 837–863.

[8]Singh, S. N., Yim, W., and Wells, W. R., "Direct Adaptive and Neural Control of Wing-Rock Motion of Slender Delta Wings," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 25–30.

[9]Mizerek, R. T., "Gaussian Neural Network for Direct Adaptive Control of Robotic Systems," *IEEE Conference on Decision and Control*, 1994, pp. 3504, 3505.

[10]Chen, F.-C., and Liu, C.-C., "Adaptively Controlling Nonlinear Continuous-TimeSystems Using Multilayer Neural Networks," *IEEE Transactions on Automatic Control*, Vol. 39, No. 6, 1994, pp. 1306–1310.

[11]Lewis, F. L., Yesildirek, A., and Liu, K., "Neural Net Robot Controller: Structure and Stability Proofs," *IEEE Conference on Decision and Control*, 1993, pp. 2785–2791.

[12]Nelson, R. C., *Flight Stability and Automatic Control*, McGraw–Hill, New York, 1989.

[13]Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, Wiley, New York, 1991.

[14]Hornik, K., Stinchcombe, M., and White, H., "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, 1989, pp. 359–366.

[15]Funahashi, K.-I., "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks*, Vol. 2, 1989, pp. 183–192.

[16]Williams, D. E., Friedland, B., and Madiwale, A. N., "Modern Control Theory for Design of Autopilots for Bank-to-Turn Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 378–386.

[17]Arrow, A., "Status and Concerns for Bank-to-Turn Control of Tactical Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 2, 1984, pp. 267–274.

[18]Chuang, D.-M., "Guidance and Nonlinear Control for a Highly Maneuverble Missile," M.S. Thesis, Dept. of Electrical Engineering, National Taiwan Univ., Taiwan, ROC, 1993.

[19]Sastry, S., and Bodson, M., *Adaptive Control: Stability, Convergence, and Robustness*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[20]Narendra, K. S., and Annaswamy, A. M., *Stable Adaptive Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1989.