ELSEVIER

Fuzzy Sets and Systems 96 (1998) 281–306

FUZZY sets and systems

A self-organizing fuzzy control approach for bank-to-turn missiles

Chuan-Kai Lin, Sheng-De Wang*

EE Building, Room 441, Department of Electrical Engineering, National Taiwan University, 1 Roosevelt Rd., Sec. 4, Taipei 106, Taiwan

Received March 1996; revised October 1996

Abstract

In this paper, an adaptive fuzzy autopilot is developed for bank-to-turn (BTT) missiles. We propose a self-organizing rotated fuzzy basis function control system that is more flexible than fuzzy basis function expansion in that not only the output weights but also the shapes of fuzzy basis functions can be tuned. Another characteristic of this autopilot is to incorporate the terminal attractor controller into the autopilot to speed up the convergence rate. The global stability analysis established in the Lyapunov sense shows that the states and tracking errors of the BTT missile are uniformly bounded. Finally, there are two simulation examples to demonstrate the effectiveness of the proposed autopilot. The purpose of the first example is to indicate the better performance including faster convergence speed and lower steady-state error obtained by autopilots with terminal attractors as compared with that autopilot without terminal attractor. Another simulation result proves that BTT missile can manipulate under various flight conditions in our scenarios and shows the superior tracking performance. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Fuzzy basis function; Neural networks; Engineering; Terminal attractor

1. Introduction

Fuzzy control is an approach to nonlinear and complex control design which has attracted a great deal of research interest in the past decade. The basic idea of the approach is to incorporate fuzzy IF–THEN rules into the control design, that is, fuzzy control combines two resources: input–output data and the experts' experience expressed by rules. Therefore, fuzzy control is always applied to the system which is too complex to get the mathematical model precisely as chemical process. The autopilot design for the BTT missile, which is a highly nonlinear coupled system, is a challenging task and it is a novel application to BTT missiles for fuzzy control.

The characteristics of BTT missiles are high maneuverability and high aerodynamic acceleration. High maneuverability will require large roll rates which will in turn introduce cross-coupled nonlinear dynamics to affect the motion of pitch and yaw. Furthermore, the asymmetric cross section of BTT missile results in the high acceleration capability in pitch plane restrict the acceleration in the yaw plane. On the other hand, the performance of the autopilot design should comply with the limits of positive attack angle and small sideslip angle.

A wide variety of approaches have been used successfully to address the autopilot design for missiles. Lin and Yueh [21] neglected the nonlinear terms to apply classical SISO methods to the pitch and yaw channels which are simplified to be independent. The conventional approach for designing BTT autopilot

* Corresponding author. E-mail: sdwang@star.ee.ntu.edu.tw.

PII S0165-0114(96)00320-X

based on linear approximation of the cross-coupling at each design point has been widely investigated [7]. Many research works apply the optimal control theory to the autopilot design; for example, Williams have proposed an LQG/LQR-based control methodology to cope with the nonlinear and coupled equations of motion in [33, 34]. Bossi and Langehough [1] also proposed an LQG/LQR scheduling approach based on the linearized dynamic models which are derived from different flight conditions. By satisfying performance and stability robustness requirements, Wise [35] used loop transfer recovery to design autopilots. Under the constraints of the small attack angle and fixed flight condition, Chang and Yuan [4] developed a nonlinear polynomial feedback control law for BTT asymmetric missiles. Recently, some studies, concerning adaptive robust control based on nonlinear geometric theory to achieve the satisfactory tracking performance, have been presented in [17, 18].

In the past, the fuzzy control approach has not been viewed as a rigorous science due to a lack of formal synthesis techniques that guarantee the very basic requirements of global stability and acceptable performance [30]. Adaptive fuzzy controllers, which are proved to be globally stable, are developed in sequence [29, 30]. This kind of controller expresses the fuzzy model as a series expansion of basis functions named fuzzy basis functions (FBF). The fuzzy controller, used to approximate an optimal controller or unknown part of the plant, is adjusted by an adaptive law based upon a Lyapunov synthesis approach.

Motivated by the work in [5], we are inspired to apply the fuzzy control theory to perform the attitude control for BTT aircrafts. In this paper, a self-organizing rotated fuzzy basis function expansion is used to approximate partial unknown parts of the plant. As in [12, 15], the weighting matrix scales and rotating input vectors produce elliptic isocontours for each fuzzy basis function to add more flexibility in approximating a function. However, the scales and rotating angles of the isocontours are tuned by learning in this paper rather than fixed in [15, 12]. The self-organizing rotated fuzzy basis function (RFBF) expansion can construct all free parameters: centers and radii of FBFs. The approximation errors, unmodeled dynamics and disturbances are inevitable. Therefore, a robust learning algorithm as the one proposed [6] is used to tune the parameters of the network. In addition, a tactical missile should track target as soon as possible. Hence, the concept of terminal attractors, which is introduced by Zak [36], is added to the controller to improve the convergence rate.

The remainder of this paper is organized as follows. In Section 2, the control objective is of BTT missiles. Section 3 is dedicated to introducing the rotated fuzzy basis function. The rotated fuzzy basis function network is compared with the fuzzy basis function network and adaptive-network-based fuzzy inference system (ANFIS) [9]. Section 4 describes the self-organizing rotated fuzzy basis function network based control approach in detail. A global stable adaptive fuzzy autopilot in a constructive manner based on the Lyapunov synthesis technique is developed. In Section 5, the faster convergence rate of autopilot with terminal attractor is demonstrated in the first simulation example. Another simulation example is to demonstrate that only one autopilot can operate at various flight conditions. At last, Section 6 concludes the paper.

2. Problem statement

The detailed dynamic equations and state notations of BTT missiles is available in the Appendix. The overall BTT missile dynamic system including actuators has been shown to be not an affine system [18]. The output signals Φ, A_y and A_z are chosen to track the desired trajectories Φ_c, A_{yc} and A_{zc} respectively. However, a vast number of numerical simulations and numerical calculation of the system eigenvalues of the locally linearized version of the dynamic system verify that BTT missiles have significant nonminimum phase phenomenon. In [18], the undesirable nonminimum phase property is avoided by an output-redefinition method such that the nominal model of a BTT missile can be viewed as a weakly nonminimum phase system. The new output signals Φ, V and W are chosen according to the profile of the desired trajectories.

This paper is devoted to the design of a fuzzy adaptive control algorithm for a class of BTT missiles. The nominal plant of BTT missiles can be rewritten by the input–output feedback linearization technique [28] in the following form:

$$\begin{bmatrix} y_1^{(n_1)} \\ y_2^{(n_2)} \\ y_3^{(n_3)} \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ f_3(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} g_{11}(\boldsymbol{x}) & g_{12}(\boldsymbol{x}) & g_{13}(\boldsymbol{x}) \\ g_{21}(\boldsymbol{x}) & g_{22}(\boldsymbol{x}) & g_{23}(\boldsymbol{x}) \\ g_{31}(\boldsymbol{x}) & g_{32}(\boldsymbol{x}) & g_{33}(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = f(x) + G(x)\boldsymbol{u} \qquad (2.1)$$

where

$$y_1 = \Phi,\ y_2 = V,\ y_3 = W,\ [u_1\ u_2\ u_3]^{\mathrm{T}} = [\delta_p\ \delta_q\ \delta_r]^{\mathrm{T}}$$

and

$$\boldsymbol{x} = [P\ Q\ R\ \Phi\ \Theta\ \Psi\ U\ V\ W\ X\ Y\ Z]^{\mathrm{T}}.$$

The relative degree (n_1, n_2, n_3) for the three input–output channels is found to be $(3, 2, 2)$. In the above equation, $f(\boldsymbol{x})$ is an unknown function vector, $G(\boldsymbol{x})$ is a known gain matrix as given in the Appendix and $\boldsymbol{u}$ is the input vector.

Now, the problem is how to develop an adaptive fuzzy controller to achieve the control objective. The control objective is to force the plant states, $\Phi, \dot{\Phi}, \ddot{\Phi}, V, \dot{V}, W$ and $\dot{W}$, to follow the specified trajectories, $\Phi_c, \dot{\Phi}_c, \ddot{\Phi}_c, V_c, \dot{V}_c, W_c$ and $\dot{W}_c$, and output signals, y_1, y_2, y_3, to follow the desired output trajectories, $y_{1\mathrm{d}}, y_{2\mathrm{d}}, y_{3\mathrm{d}}$. For the simplicity of discussion, three tracking error vectors are defined as follows:

$$\boldsymbol{e}_1 = \begin{bmatrix} y_1 \\ \dot{y}_1 \\ \ddot{y}_1 \end{bmatrix} - \begin{bmatrix} y_{1\mathrm{d}} \\ \dot{y}_{1\mathrm{d}} \\ \ddot{y}_{1\mathrm{d}} \end{bmatrix} = \begin{bmatrix} \Phi \\ \dot{\Phi} \\ \ddot{\Phi} \end{bmatrix} - \begin{bmatrix} \Phi_c \\ \dot{\Phi}_c \\ \ddot{\Phi}_c \end{bmatrix},$$

$$\boldsymbol{e}_2 = \begin{bmatrix} y_2 \\ \dot{y}_2 \end{bmatrix} - \begin{bmatrix} y_{2\mathrm{d}} \\ \dot{y}_{2\mathrm{d}} \end{bmatrix} = \begin{bmatrix} V \\ \dot{V} \end{bmatrix} - \begin{bmatrix} V_c \\ \dot{V}_c \end{bmatrix}, \qquad (2.2)$$

$$\boldsymbol{e}_2 = \begin{bmatrix} y_3 \\ \dot{y}_3 \end{bmatrix} - \begin{bmatrix} y_{3\mathrm{d}} \\ \dot{y}_{3\mathrm{d}} \end{bmatrix} = \begin{bmatrix} W \\ \dot{W} \end{bmatrix} - \begin{bmatrix} W_c \\ \dot{W}_c \end{bmatrix}.$$

The control problem is thus to develop a control law that guarantees the tracking error vectors will approach zero.

For the purpose of alleviating the nonminimum phase phenomenon, an output–redefinition method is applied such that the new command signals should be transformed from the original commands, or trajectories [4, 18, 20]. The command generator performs the task. The inputs of the command generator are original desired trajectories, Φ_c, A_{yc} and A_{zc}, velocity of the missile and the height of the missile; the outputs are Φ_c, V_c and W_c. In practice, Φ_c and V_c are not changed; as to W_c, it is proportional to $A_z - A_{zc}$. In other words, $W_c = K_{Wc}(A_z - A_{zc})$.

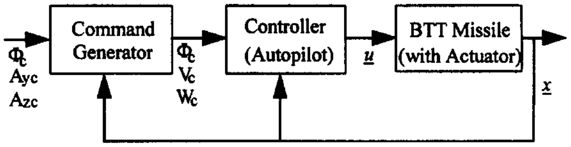


Fig. 1. The block diagram of the closed loop system.

Since the outputs are redefined, it is necessary to generate the corresponding redefined tracking signals by a command generator as shown in Fig. 1. If $f(x)$ is known and the gain matrix $G(x)$ is also known and invertible, then the control law can be of the form:

$$\boldsymbol{u}(t) = G^{-1}\left(\begin{bmatrix} y_{1\mathrm{d}}^{(3)} \\ y_{2\mathrm{d}}^{(2)} \\ y_{3\mathrm{d}}^{(2)} \end{bmatrix} - v_{\mathrm{lf}}(t) - f(x) \right), \qquad (2.3)$$

where

$$v_{\mathrm{lf}}(t) = [v_{\mathrm{lf1}}(t)\ v_{\mathrm{lf2}}(t)\ v_{\mathrm{lf3}}(t)]^{\mathrm{T}} = [\boldsymbol{k}_1^{\mathrm{T}}\boldsymbol{e}_1\ \boldsymbol{k}_2^{\mathrm{T}}\boldsymbol{e}_2\ \boldsymbol{k}_3^{\mathrm{T}}\boldsymbol{e}_3]^{\mathrm{T}}$$

is a linear feedback control law. The control law (2.3) will lead to the following error dynamics:

$$e_i^{(n_i)} + k_{i1} e_i^{(n_i-1)} + \cdots + k_{in_i} = 0. \qquad (2.4)$$

If we choose $\boldsymbol{k}_i = [k_{in_i}\ \ldots\ k_{i1}]^{\mathrm{T}}$ such that all roots of the polynominal $p_i(s) = s^{n_i} + k_{i1}s^{n_i-1} + \cdots + k_{in_i}$ are in the open left half of the complex plane, the closed loop system will be asymptotically stable. Unfortunately, $f(\boldsymbol{x})$ is unknown and the control law (2.3) should be modified. For theoretical and practical reasons stated in [30], we can use a fuzzy basis function network to approximate $f(\boldsymbol{x})$. And it has been shown that, with linguistic fuzzy description, functions f_i can be approximated using a fuzzy model called fuzzy basis function expansion [32]. Suppose that the control law is modified as

$$\boldsymbol{u}(t) = G^{-1}\left(\begin{bmatrix} y_{1\mathrm{d}}^{(3)} \\ \ddot{y}_{2\mathrm{d}} \\ \ddot{y}_{3\mathrm{d}} \end{bmatrix} - \boldsymbol{v}_{\mathrm{lf}}(t) - \boldsymbol{v}_{\mathrm{RFBF}}(t)\right), \qquad (2.5)$$

where $v_{\mathrm{RFBF}}(t) = [v_{\mathrm{RFBF1}}(t)\ \ v_{\mathrm{RFBF2}}(t)\ \ v_{\mathrm{RFBF3}}(t)]^{\mathrm{T}}$ is the fuzzy basis function expansion used to cancel out the unknown nonlinear function $f(\boldsymbol{x})$. With the control law (2.5), the closed-loop system becomes

$$\dot{\boldsymbol{e}}_i = A_i \boldsymbol{e}_i + \boldsymbol{b}_i(f_i(\boldsymbol{x}) - v_{\mathrm{RFBFi}}(t)), \quad i = 1,2,3 \quad (2.6)$$

where $b_1 = [0\ \ 0\ \ 1]^{\mathrm{T}}$, $b_2 = b_3 = [0\ \ 1]^{\mathrm{T}}$, and

$$A_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -k_{11} & -k_{12} & -k_{13} \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 1 \\ -k_{21} & -k_{22} \end{bmatrix} \quad \text{and} \quad A_3 = \begin{bmatrix} 0 & 1 \\ -k_{31} & -k_{32} \end{bmatrix}$$

are matrices in the canonical form, with eigenvalues at the roots of the Hurwitz polynomials which are determined by the linear feedback control law $\boldsymbol{v}_{\mathrm{lf}}(t)$ [27].

However, the approximation errors and disturbances are unavoidable. One approach to solve the control problem is to define an error metric $\boldsymbol{s}$ as follows:

$$\boldsymbol{s}(t) = \begin{bmatrix} (\mathrm{d}/\mathrm{d}t + \lambda_1)^2(y_1 - y_{1\mathrm{d}}) \\ (\mathrm{d}/\mathrm{d}t + \lambda_2)(y_2 - y_{2\mathrm{d}}) \\ (\mathrm{d}/\mathrm{d}t + \lambda_3)(y_3 - y_{3\mathrm{d}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda}_1^{\mathrm{T}} \boldsymbol{e}_1 \\ \boldsymbol{\lambda}_2^{\mathrm{T}} \boldsymbol{e}_2 \\ \boldsymbol{\lambda}_3^{\mathrm{T}} \boldsymbol{e}_3 \end{bmatrix}, \quad (2.7)$$

where $\boldsymbol{\lambda}_1 = [\lambda_1^2\ \ 2\lambda_1\ \ 1]^{\mathrm{T}}$, $\boldsymbol{\lambda}_2 = [\lambda_2\ \ 1]^{\mathrm{T}}$, $\boldsymbol{\lambda}_3 = [\lambda_3\ \ 1]^{\mathrm{T}}$, and $\lambda_1, \lambda_2, \lambda_3$ are constants. The equation $\boldsymbol{s}(t) = 0$ defines three time-varying hyperplanes on which the tracking error vectors decay exponentially to zero, so that perfect tracking can be asymptotically obtained by maintaining this condition [27]. From Eqs. (2.6) and (2.7), the time derivative of the metric can then be written as

$$\dot{s}_i(t) = -k_{D_i} s_i(t) + f_i(t) - v_{\mathrm{RFBFi}}(t), \quad i = 1,2,3, \quad (2.8)$$

where k_{D_i} satisfies the relationship $k_{D_i}^n - k_{in} k_{D_i}^{n-1} - k_{in-1} k_{D_i}^{n-2} - \cdots - k_{i1} = 0$ [20]. If the magnitude of s_i can be shown to be bounded by a constant Δ_i, then the actual tracking errors can be shown to be asymptotically bounded by

$$\left| \frac{\mathrm{d}^j}{\mathrm{d}t^j}(y_i - y_{i\mathrm{d}}) \right| \leqslant 2^j \lambda_i^{n_i - i + 1} \Delta_i,$$

$$j = 0,\ldots,n_i - 1, \quad i = 0,1,2. \quad (2.9)$$

Deadzones can be incorporated into error metrics by defining continuous functions $s_{i\Delta}$ as:

$$s_{i\Delta}(t) = s_i(t) - \Delta_i\, \mathrm{sat}(s_i(t)/\Delta_i), \quad i = 0,1,2, \quad (2.10)$$

where sat is the saturation function:

$$\mathrm{sat}(z) = \begin{cases} 1, & z > 1, \\ z, & |z| \leqslant 1, \\ -1, & z < -1. \end{cases} \quad (2.11)$$

Deadzone functions, which are specified around the zero of their corresponding error metrics, will be used in the adaptation law to tolerate the parameter errors, unknown dynamics approximation errors and disturbances.

Appropriately choosing parameter k_i in (2.4) implies that the tracking errors will asymptotically converge to zero, that is, they approach zero at infinite time. As we know, the missile should attack the target in a short time. For this control objective, the concept of terminal attractors is used to reduce the tracking errors to zero in finite time. The detailed controller architecture and stability analysis will be discussed in Section 4.

3. Rotated fuzzy basis function network

Fuzzy basis function network is an approach to integrating fuzzy reasoning systems with radial basis function networks. The differences and similarities between fuzzy basis functions and radial basis functions has been discussed [9]. In this paper, we propose an RFBF network to improve the fuzzy basis function network in modeling capability and convergence speed. There are three major reasons for presenting the RFBF networks. The first, FBFs and radial basis functions have the same problem that they may not adequately capture the relative "scales" between inputs of different types [12]. Poggio and Girosi [26] have proposed using a single weighted norm instead of an Euclidean norm to overcome this problem [12].

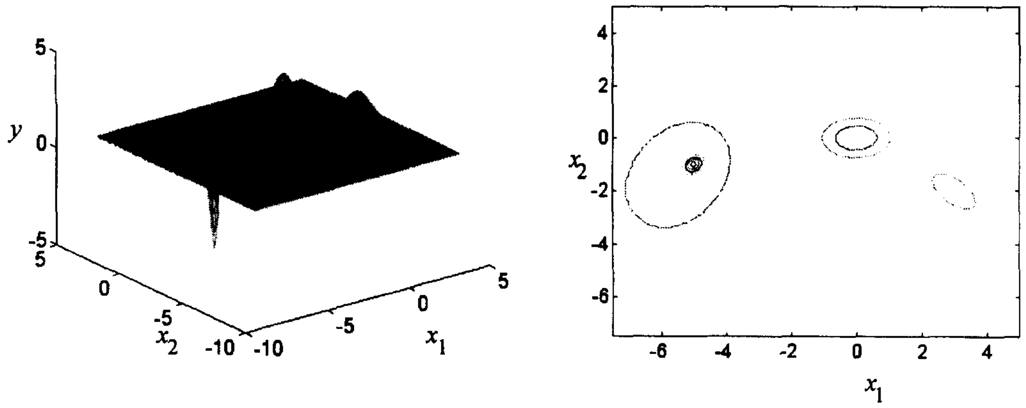


Fig. 3. Surface of function y and contours of RFBFs in Example 3.2.

or

IF $(\eta_j, \sigma_j, \Re_j)$ THEN c_j,

where $\Re_j$ is the rotational matrix. This the basic idea of RFBF. For explaining the rule format more clearly, we take two examples as illustration.

Example 3.1. A function

$$y = 3.0\mathrm{e}^{-x_1^2-2x_2^2} + 2.0\mathrm{e}^{-4(x_1-3)^2-(x_2+2)^2}$$
$$-4.5\mathrm{e}^{-16(x_1+5)^2-9(x_2+1)^2}$$

can be approximated by three FBFs expansion exactly:

IF ([0 0], [1 $1/\sqrt{2}$]) THEN y is ([3.0]),

IF ([3 −2], [0.5 1]) THEN y is ([2.0]),

IF ([−5 −1], [1/4 1/3]) THEN y is ([−4.5]).

It can also be expanded by RFBFs exactly as

IF ([0 0], [1 $1/\sqrt{2}$], 0°) THEN y is ([3.0]),

IF ([3 −2], [0.5 1], 0°) THEN y is ([2.0]),

IF ([−5 −1], [1/4 1/3], 0°) THEN y is ([−4.5]).

Example 3.2. A function

$$y = 3.0\mathrm{e}^{-x_1^2-2x_2^2} + 2.0\mathrm{e}^{-2.5(x_1-3)^2-3(x_1-3)(x_2+2)-2.5(x_2+2)^2}$$
$$+2.5\mathrm{e}^{-14.25(x_1+5)^2+3.5\sqrt{3}(x_1+5)(x_2+1)-10.75(x_2+1)^2}$$

can be expanded by three RFBFs exactly:

IF ([0 0], [1 $1/\sqrt{2}$], 0°) THEN y is ([3.0]),

IF ([3 −2], [0.5 1], 45°) THEN y is ([2.0]),

IF ([−5 −1], [1/4 1/3], -30°)
THEN y is ([−4.5]).

Fig. 3, which shows the three-dimensional diagram of function y and the contours of RFBFs, gives a real feeling of RFBFs. If one uses traditional FBFs to approximate the function, it needs more rules and gets worse results. Therefore, this function can be approximated exactly by RFBFs that are rotated FBFs.

General speaking experts can state IF–THEN rules and rough membership functions, however, the exact membership functions or FBFs are designed or modified by engineering or tuned by algorithms. Based on the facts, the initial rotation angles are 0° (rotational matrices are identity matrices) and the rotational matrices can be tuned by learning algorithms. For instance, experts may observe that the centers of rules should be at [0 0], [3 −2] and [−5 −1] in the Example 3.2. However, they do not capture the "scale" and "orientation" very well. Therefore, experts may not specify the rotational angles or rotational matrices. Of course, as the above example, the initial rotational matrices may not be specified as identity matrices for some experts who have the idea about the "scale" and "orientation". For a two-inputs fuzzy rules, the

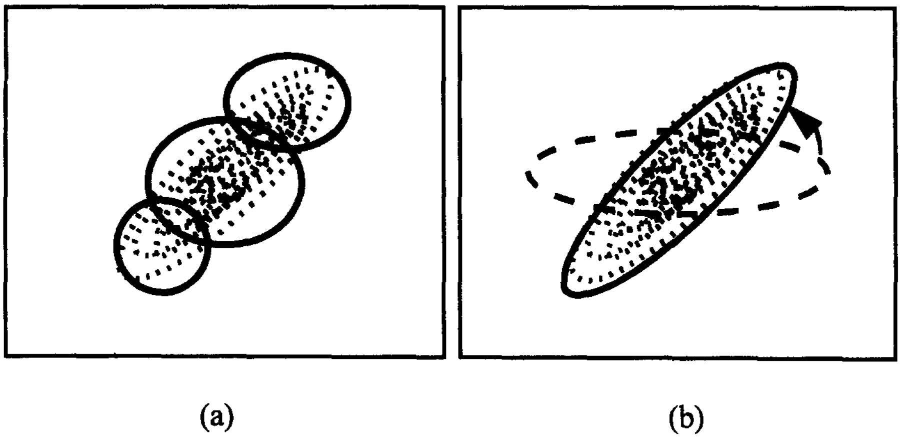


Fig. 2. (a) Dots represent data and solid elliptic are FBFs' contours. (b) Dots represent data and the RFBF (solid elliptic) is rotated by the FBF (dashed elliptic).

However, the freedom is restricted that each node in radial basis function network has the same aspect ratio and is rotated by the same amount. Therefore, Lee and Kil [15] used a weighted norm for each node to improve the flexibility. Based on Lee and Kil's study, we propose a rotated FBF network that is more flexible than an FBF network and still posesses the advantages of FBF. We find that the RFBF network reduces to FBF network if there is no rotation. So, the FBF network is a special case of the RFBF network. Secondly, the RFBF network can tune not only the parameters of consequent part of fuzzy rules but also the widths, centers, scales and orientations of FBFs. The third reason is that the parameters of the RFBF network is fewer than the self-organizing FBF system [24] that can also tune membership functions with the same number of fuzzy rules. Furthermore, for evaluating the performance of the RFBF network, we also compare it with the famous adaptive fuzzy systems, ANFIS [8], and FBF network.

3.1. Rule format of the RFBF network

From the viewpoint of conventional fuzzy systems, it is surprising that FBFs can be rotated. Assume that the fuzzy rule base consists r linguistic rules in the IF–THEN form as follows:

IF x_1 is $\tilde{A}_{j1}$ AND x_2 is $\tilde{A}_{j2}$ AND…AND x_n is $\tilde{A}_{jn}$
THEN y_1 is $\tilde{B}_{j1}$ AND y_2 is $\tilde{B}_{j2}$ AND…AND
y_m is $\tilde{B}_{jm}$,

where $j=1,2,\ldots,r$, x_i $(i=1,2,\ldots,n)$ are the input variables to the fuzzy system, y_k $(k=1,2,\ldots,m)$ are output variables of the fuzzy system, and $\tilde{A}_{ji}$ and $\tilde{B}_{jk}$ are linguistic terms characterized by their corresponding fuzzy membership functions $\mu_{\tilde{A}_{ji}}(x_i)$ and $\mu_{\tilde{B}_{jk}}(y_k)$, respectively. From geometrical point of view [24], the fuzzy rules can be rewritten as

IF (η_{j1},σ_{j1}) AND (η_{j2},σ_{j2}) AND…AND (η_{jn},σ_{jn})
THEN $(c_{j1},\ldots,c_{jm})$,

where η_{ji} and σ_{ji} are the center and width of $\mu_{\tilde{A}_{ji}}(x_i)$, and c_{jk} is the center of the membership function $\mu_{\tilde{B}_{jk}}(y_k)$ of the fuzzy singleton set $\tilde{B}_{jk}$. Further, the rule format can be simplified as

IF $(\boldsymbol{\eta}_j,\boldsymbol{\sigma}_j)$ THEN $\boldsymbol{c}_j$,

where $\boldsymbol{\eta}_j=[\eta_{j1}\ \ldots\ \eta_{jn}]^{\mathrm{T}}$, $\boldsymbol{\sigma}_j=[\sigma_{j1}\ \ldots\ \sigma_{jn}]^{\mathrm{T}}$ and $\boldsymbol{c}_j=[c_{j1}\ \ldots\ c_{jm}]^{\mathrm{T}}$. In other words, the FBF can be characterized by the center vector $\boldsymbol{\eta}_j$ and width vector $\boldsymbol{\sigma}_j$. Take Fig. 2, which shows data distribution, into consideration, experts who are familiar with FBF expansion may use three rules to cover the whole data. However, experts may observe that only one FBF can cover the whole data by rotating the FBF. Hence, experts may state a rule as

IF $(\boldsymbol{\eta}_j,\boldsymbol{\sigma}_j)$ AND rotation angle is $\boldsymbol{\theta}_j$ THEN $\boldsymbol{c}_j$, rotational matrix $\mathfrak{R}_j$ is

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

where θ is the rotation angle. As for a three-inputs fuzzy rule, the rotational matrix $\mathfrak{R}_j$ is defined by $\mathfrak{R}_j = R_{1j}(\theta_{1j})R_{2j}(\theta_{2j})R_{3j}(\theta_{3j})$ where rotation angle $\boldsymbol{\theta}_j = [\theta_{1j}\ \theta_{2j}\ \theta_{3j}]^{\mathrm{T}}$ and $R_{1j}(\theta_{1j})$, $R_{2j}(\theta_{2j})$, and $R_{3j}(\theta_{3j})$ are defined as follows:

$$R_{1j}(\theta_{1j}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{1j} & -\sin\theta_{1j} \\ 0 & \sin\theta_{1j} & \cos\theta_{1j} \end{bmatrix},$$

$$R_{2j}(\theta_{2j}) = \begin{bmatrix} \cos\theta_{2j} & 0 & \sin\theta_{2j} \\ 0 & 1 & 0 \\ -\sin\theta_{2j} & 0 & \cos\theta_{2j} \end{bmatrix},$$

$$R_{3j}(\theta_{3j}) = \begin{bmatrix} \cos\theta_{3j} & -\sin\theta_{3j} & 0 \\ \sin\theta_{3j} & \cos\theta_{3j} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We have provided an intuitive method to find rotational matrix $\mathfrak{R}_j$ for two-dimensional and three-dimensional input space. For conventional fuzzy control, experts only design a two-dimensional lookup table and most higher-dimensional control rules are generated by self-organizing or adaptive fuzzy systems. So, it is enough for experts to use the above methods to describe their experience and knowledge with the RFBF rule's form. As for higher dimension control rules, they can be resorted to the learning algorithms. From the above discussion, it is believed that experts' knowledge can be incorporated into the RFBF network. Although the RFBFs take advantage over FBFs, the cost is the additional computational efforts of rotational matrices.

3.2. Architecture of the RFBF network

The RFBF network is a three-layer neural network architecture as shown in Fig. 4. Each node in hidden layer of the RFBF network represents a fuzzy rule and the output of the node is the firing strength of the fuzzy rule. Assume a fuzzy system has n inputs, m outputs, and r fuzzy rules, that is, there are only $(n+1) \times r + r \times m$ parameters to be tuned for an RFBF network and $2n \times r + r \times m$ parameters for three-layer self-organizing FBF system [24]. In general, n is always greater than one such that the number of parameters to be tuned of the RFBF network is fewer. As for performing fuzzy inference by the three-layer multilayer neural network with Gaussian activation functions, it would be the topic of this subsection.

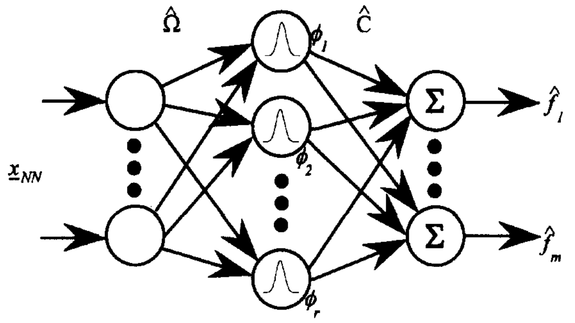


Fig. 4. Network representation of an RFBF expansion system.

The output of the fuzzy logic system with center-average defuzzification, product inference, and singleton fuzzification is of the following form:

$$y_k = \frac{\sum_{j=1}^{r} c_{jk}(\prod_{i=1}^{n} \mu_{\tilde{A}_{ji}}(x_i))}{\sum_{j=1}^{r}(\prod_{i=1}^{n} \mu_{\tilde{A}_{ji}}(x_i))}. \tag{3.1}$$

As in [29], the defuzzification of RFBF network is a weighted sum of firing strengths of rules.

For fuzzy basis function expansion, the basis function is Gaussian function defined as follows [32]:

$$\mu_{\tilde{A}_{ji}}(x_i) = \mathrm{e}^{-((x_i-\eta_{ji})/\sigma_{ji})^2}, \tag{3.2}$$

where η_{ji} is the center of the Gaussian function and σ_{ji} determines the width of the function. The firing strength of each rule is denoted as the fuzzy basis function. For ease of representation, we define the FBF and RFBF as follows.

Definition 1 (*Fuzzy basis function*). Define fuzzy basis functions (FBFs) by

$$\phi_j(\|\boldsymbol{x} - \boldsymbol{\eta}_j\|, \boldsymbol{\sigma}_j) = \prod_{i=1}^{n} \mu_{\tilde{A}_{ji}}(x_i) = \mathrm{e}^{-(x-\boldsymbol{\eta}_j)^{\mathrm{T}} S_j (x-\boldsymbol{\eta}_j)},$$
$$j = 1, 2, \ldots, r, \tag{3.3}$$

where $\mu_{A_{ji}}(x_i)$ are Gaussian membership functions (3.2), $\boldsymbol{x} = [x_1 \ \ldots \ x_n]^{\mathrm{T}} \in U$, $\boldsymbol{\eta}_j = [\eta_{j1} \ \ldots \ \eta_{jn}]^{\mathrm{T}} \in U$ and $S_j = \mathrm{diag}[1/\sigma_{j1}^2, \ldots, 1/\sigma_{jn}^2)$.

Definition 2 (*Rotated fuzzy basis function*). Define rotated fuzzy basis functions (RFBFs) by

$$\phi_j(\|\boldsymbol{x} - \boldsymbol{\eta}_j\|, \boldsymbol{\sigma}_j) = \mathrm{e}^{-(\boldsymbol{x}-\boldsymbol{\eta}_j)^{\mathrm{T}} \Re_j^{\mathrm{T}} S_j \Re_j (\boldsymbol{x}-\boldsymbol{\eta}_j)} = \mathrm{e}^{-\boldsymbol{z}_j^{\mathrm{T}} S_j \boldsymbol{z}_j},$$
$$j = 1, 2, \ldots, r, \tag{3.4}$$

where $\Re_j$ is the rotational matrix and $\boldsymbol{z}_j = \Re_j(\boldsymbol{x} - \boldsymbol{\eta}_j)$.

The RFBF expansion can be defined in the following definition.

Definition 3 (*Rotated fuzzy basis function expansion*). The output of a fuzzy system with rotated fuzzy basis functions can be expressed in the form

$$y_k = \sum_{j=1}^{r} c_{jk} \phi_j = \sum_{j=1}^{r} c_{jk} \mathrm{e}^{-\boldsymbol{z}_j^{\mathrm{T}} S_j \boldsymbol{z}_j}, \quad k = 1, \ldots, m, \tag{3.5}$$

where c_{jk}'s are weights for defuzzification, ϕ_j's are rotated fuzzy basis functions.

In adaptive FBF systems [29, 30], the parameters of $\mu_{\tilde{A}_{ji}}(x_i)$'s are fixed and the weights c_{jk}'s are adjustable. However, in this paper, the parameters of $\mu_{\tilde{A}_{ji}}(x_i)$'s, η_{ji} and σ_{ji} are also adjustable and the parameter updating law will be stated in later section.

To facilitate easy computation of rotated fuzzy basis function, we adopt the following form for the basis function:

$$\phi_j = \mathrm{e}^{-(\omega_{j0} + \omega_{j1} x_1 + \cdots + \omega_{jn} x_n)^2} = \mathrm{e}^{-\boldsymbol{x}_{NN}^{\mathrm{T}} \boldsymbol{\omega}_j \boldsymbol{\omega}_j^{\mathrm{T}} \boldsymbol{x}_{NN}},$$
$$j = 1, 2, \ldots, r, \tag{3.6}$$

where $\omega_{j0}, \ldots, \omega_{jn}$ are adjustable parameters, $\boldsymbol{\omega}_j = [\omega_{j0} \ \ldots \ \omega_{jn}]^{\mathrm{T}}$ and $x_{NN} = [1 \ x_1 \ \ldots \ x_n]^{\mathrm{T}}$. In the following, we will prove (3.6) can be equivalent to (3.4), i.e., the three-layer RFBF network can perform fuzzy inference.

It is easy to show, using Definition 2, Eq. (3.6) is an RFBF, since there exists some matrix Γ_j such that

$$\boldsymbol{\omega}_j \boldsymbol{\omega}_j^{\mathrm{T}} = \Gamma_j^{\mathrm{T}} \Lambda_j \Gamma_j, \quad j = 1, 2, \ldots, r, \tag{3.7}$$

where $\Lambda_j = \mathrm{diag}(0, \lambda_{j1}, \ldots, \lambda_{jn})$. Because the symmetric matrix $\boldsymbol{\omega}_j \boldsymbol{\omega}_j^{\mathrm{T}}$ is semi-positive definite, all diagonal elements of Λ_j are nonnegative. The matrix Γ_j can be divided into four blocks:

$$\Gamma_j = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix}, \tag{3.8}$$

where Γ_{11} is 1×1, Γ_{12} is $1 \times n$, Γ_{21} is $n \times 1$ and Γ_{22} is $n \times n$ matrix. Thus Eq. (3.6) can be rewritten as

$$\phi_j = \mathrm{e}^{-(\Gamma_{21} + \Gamma_{22}\boldsymbol{x})^{\mathrm{T}} \Lambda_j' (\Gamma_{21} + \Gamma_{22}\boldsymbol{x})}, \quad j = 1, 2, \ldots, r, \tag{3.9}$$

where $S_j = \Lambda_j' = \mathrm{diag}(\lambda_{j1}, \ldots, \lambda_{jn})$, $\boldsymbol{\eta}_j = -\Gamma_{22}^{-1}\Gamma_{21}$ and $\Re_j = \Gamma_{22}$. Therefore, Eq. (3.6) defines a class of rotated fuzzy basis functions.

In Fig. 4, the architecture of RFBF network as an approximator with $(n+1)$ inputs, r rules and m outputs can be denoted as follows:

$$\boldsymbol{\hat{f}}(\boldsymbol{x}_{NN}, \hat{\Omega}, \hat{C}) = \hat{C}\phi(\hat{\Omega}\boldsymbol{x}_{NN}), \tag{3.10}$$

where $\hat{\Omega} \in \mathbb{R}^{r \times (n+1)}$ and $\hat{C} \in \mathbb{R}^{m \times r}$ are the RFBF weight matrix and defuzzification weight matrix respectively, and the RFBF vector is given by

$$\phi(\hat{\Omega}\boldsymbol{x}_{NN}) = \begin{bmatrix} \phi(\hat{\omega}_1^{\mathrm{T}} \boldsymbol{x}_{NN}) \\ \vdots \\ \phi(\hat{\omega}_r^{\mathrm{T}} \boldsymbol{x}_{NN}) \end{bmatrix} \quad \text{and} \quad \hat{\Omega} = \begin{bmatrix} \hat{\omega}_1^{\mathrm{T}} \\ \vdots \\ \hat{\omega}_r^{\mathrm{T}} \end{bmatrix}. \tag{3.11}$$

For simplicity, we define $\hat{\boldsymbol{\phi}} = \phi(\hat{\Omega}\boldsymbol{x}_{NN})$. In this paper, the RFBF network is used to be the basic component of an adaptive fuzzy controller to approximate the unknown functions. It has been proved in [32] that for any given real function h over U, there exists a fuzzy system in the fuzzy basis function expansion form of (3.3) such that it can uniformly approximate any function on the compact set U to arbitrary accuracy. Accordingly, we have the following assumption.

Assumption 1. There exist matrices Ω^* and C^* such that $\boldsymbol{\hat{f}}$ approximates $\boldsymbol{f}$ with arbitrary accuracy ε, over a compact set U, i.e.

$$\exists \Omega^* \text{ and } C^* \text{ s.t. } |\boldsymbol{f}(\boldsymbol{x}(t)) - \boldsymbol{\hat{f}}(\Omega^*, C^*, \boldsymbol{x}(t))| \leqslant \varepsilon.$$

Hence (2.8) can be rewritten as

$$\dot{s}(t) = -Ks(t) + \boldsymbol{\hat{f}}(\Omega^*, C^*, \boldsymbol{x}(t)) - v_{\mathrm{FBF}}(t) + \boldsymbol{d}(t), \tag{3.12}$$

where $K = \mathrm{diag}(k_{D1}, k_{D2}, k_{D3})$ and the disturbance $\boldsymbol{d}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) - \hat{\boldsymbol{f}}(\Omega^*, C^*, \boldsymbol{x}(t))$ satisfies $|d_i(t)| \leqslant \varepsilon_i$.

3.3. Comparisons with the FBF network and ANFIS

This subsection presents two examples, which can be found in [9], to compare the proposed RFBF network with an adaptive fuzzy system, ANFIS, and the FBF network. The data for training and checking are all retrieved via anonymous ftp user/ai/areas/fuzzy/systems/anfis at ftp.cs.cmu.edu.

Example 3.3. A three-input nonlinear function

$$\text{output} = (1 + x^{0.5} + y^{-1} + z^{-1.5})$$

is used to be modeled by three fuzzy approaches. The 216 training data is used for training the ANFIS, FBF and RFBF approaches, and 125 checking data are used for verifying the generalization capability. A performance index, average percentage error (APE), for comparison is defined as

$$\text{APE} = \frac{1}{P}\sum_{i=1}^{P} \frac{|T(i) - O(i)|}{|T(i)|} \times 100\%,$$

where P is the number of data pairs; $T(i)$ and $O(i)$ are ith desired output and calculated output, respectively.

The RFBF network uses 9 rules (45 parameters) and 16 rules (80 parameters) for simulation, the FBF network uses 64 (125) rules with 4 (5) membership functions being assigned to each input variable. The ANFIS contains 8 rules with 50 parameters. We initialize the parameters of the RFBF network randomly. For FBF network, the membership functions are equally spaced along [1.0, 6.0] of each input variable and $\sigma_{ji} = 1.08$ for 64 rules case and $\sigma_{ji} = 0.81$ for 125 rules case. The learning rule adopted by the RFBF network is backpropagation algorithm and the learning rule of the FBF network is

$$\dot{c}_i = k_i e_i \phi_i, \quad i = 1, \ldots, N_i,$$

where N_i is the rule number, ϕ_i is the firing strength of the ith rule, k_i is a constant and ϕ_i is the output error $(T(i) - O(i) = T(i) - \sum_{i=1}^{N_i} c_i\phi_i)$. After 200 epochs, the results are listed in Table 1. It is worth noting that one rule of the RFBF network is in the following form:

Table 1
Comparisons with ANFIS and FBF network

Model	APE_{trn} (%)	APE_{chk} (%)	Parameter number
RFBF	0.448	1.064	45
RFBF	0.441	1.057	80
FBF	4.170	5.162	64
FBF	0.630	4.956	125
ANFIS	0.043	1.066	50

$$\text{If}\left([-0.1466\ 0.4407\ \ -0.5053], [\infty\ \infty\ 5.47], \begin{bmatrix} -0.3299 & 0.3548 & -0.1307 \\ 0.4854 & 0.8561 & 0.0907 \\ -0.8092 & 0.3660 & 0.1419 \end{bmatrix}\right)$$

$$\text{THEN } ([19.127621]),$$

that is, the rule is reduced to one dimension and the RFBF is

$$\mathrm{e}^{-((w+0.5053)/5.47)^2},$$

where $w = -0.8092x + 0.336y + 0.14198z$. In this example, the RFBF network has the best generalization capability.

Example 3.4. The chaotic Mackey–Glass differential delay equation defined as

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t)$$

is used to generate time series of simulation ($\tau = 17$). The 500 input–output data pair for training and 500 input–output data pairs for checking, which have the format

$$[x(t-18)x(t-12)x(t-6)x(t)\,|\,x(t+6)],$$

are the same as the data pairs in [9]. The learning algorithms of the FBF and RFBF are the same ones in Example 3.3. The FBF and RFBF network use 81 rules (81 adjustable parameters) for simulation, and each rule with 3 membership functions being assigned

Table 2
Generalization result comparison

Method	Training cases	NDEI
RFBF	500	0.022
FBF	500	0.026
ANFIS	500	0.007
AR Model	500	0.19
Cascaded-correlation NN	500	0.06
Back-prop NN	500	0.02
Sixth-order polynomial	500	0.04
Linear prediction method	2000	0.55

Note: The last six rows are from [9].

to each input variable. While ANFIS comprises 16 rules with 104 adjustable parameters. For FBF network, the membership functions are equally spaced along [0.4, 1.4] of each input variable and $\sigma_{ji} = 0.27$. All rotational matrices of the RFBF network are

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.9924 & -0.0792 & 0.0940 \\ 0 & 0.0868 & 0.9931 & -0.07923 \\ 0 & -0.0871 & 0.0868 & 0.9924 \end{bmatrix}.$$

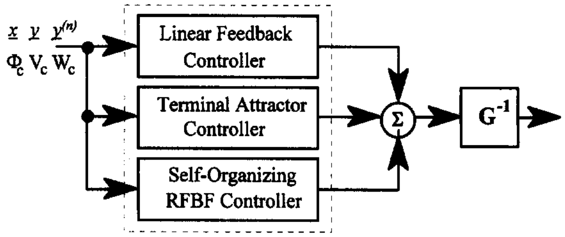
This is a special type of RFBF network. A performance index, nondimensional error index (NDEI), for comparison is defined as the root mean square error divided by the standard deviation of the target series. Table 2 provides simulation results of many methods. In this example, ANFIS has the best generalization capability, however, the RFBF network has good performance, too.

From Examples 3.3 and 3.4, the performance of the RFBF network is better than that of the FBF network. The ANFIS is not suitable for controlling BTT missiles because the learning algorithm of ANFIS cannot guarantee the stability and learning epoch by epoch will not occur in controlling BTT missiles.

4. Adaptive fuzzy autopilot design

4.1. Architecture of fuzzy autopilot

The block diagram of the controller architecture is shown in Fig. 5. Through the command generator, a set of command signals, Φ_c, V_c and W_c are generated for tracking. Thus, the control objective is to design an autopilot (controller) such that the tracking errors are as small as possible, the tracking errors converge faster and the closed system must be globally stable. For achieving the goals, the architecture of the controller consists of three components: a linear feedback controller, a terminal attractor controller and a fuzzy controller.



$\underline{x}$ $\underline{y}$ $\underline{y}^{(n)}$
Φ_c V_c W_c
Linear Feedback Controller
Terminal Attractor Controller
Self-Organizing RFBF Controller
Σ
G^{-1}

Fig. 5. The block diagram of autopilot.

The dynamic equations of BTT missiles are described in (2.1), where the gain matrix G can be derived by input–output feedback linearization method. Further, it is necessary to assume that G is invertible for our control law. Then, the overall control law is given as follows:

$$\boldsymbol{u} = G^{-1}(-\boldsymbol{v}_{\mathrm{lf}} - \boldsymbol{v}_{\mathrm{TA}} - \boldsymbol{v}_{\mathrm{FBF}}), \tag{4.1}$$

where

$$\boldsymbol{v}_{\mathrm{lf}} = \begin{bmatrix} v_{\mathrm{lf}1} \\ v_{\mathrm{lf}2} \\ v_{\mathrm{lf}3} \end{bmatrix} = \begin{bmatrix} -y_{1\mathrm{d}}^{(3)} + k_{11}(y_1 - y_{1\mathrm{d}}) + k_{12}(\dot{y}_1 - \dot{y}_{1\mathrm{d}}) + k_{13}(\ddot{y}_1 - \ddot{y}_{1\mathrm{d}}) \\ -y_{2\mathrm{d}}^{(2)} + k_{21}(y_2 - y_{2\mathrm{d}}) + k_{22}(\dot{y}_2 - \dot{y}_{2\mathrm{d}}) \\ -y_{3\mathrm{d}}^{(2)} + k_{31}(y_3 - y_{3\mathrm{d}}) + k_{32}(\dot{y}_3 - \dot{y}_{3\mathrm{d}}) \end{bmatrix}, \tag{4.2}$$

$$\boldsymbol{v}_{\mathrm{TA}} = \begin{bmatrix} v_{\mathrm{TA}1} \\ v_{\mathrm{TA}2} \\ v_{\mathrm{TA}3} \end{bmatrix} = \begin{bmatrix} k_{\mathrm{ta}1} s_{1\Delta}^{1/a_1} \\ k_{\mathrm{ta}2} s_{2\Delta}^{1/a_2} \\ k_{\mathrm{ta}3} s_{3\Delta}^{1/a_3} \end{bmatrix}, \tag{4.3}$$

$$\boldsymbol{v}_{\mathrm{RFBF}} = \begin{bmatrix} v_{\mathrm{RFBF1}} \\ v_{\mathrm{RFBF2}} \\ v_{\mathrm{RFBF3}} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{N_1} \hat{c}_{1j}\phi_j + \hat{d}_1 \\ \sum_{j=1}^{N_2} \hat{c}_{2j}\phi_j + \hat{d}_2 \\ \sum_{j=1}^{N_3} \hat{c}_{3j}\phi_j + \hat{d}_3 \end{bmatrix}, \qquad (4.4)$$

where α_1, α_2, and α_3 are positive odd integers, $k_{\mathrm{ta1}}, k_{\mathrm{ta2}}$, and k_{ta3} are positive constants. In the numerical simulation in next section, we choose $\alpha_1 = \alpha_2 = \alpha_3 = 3$. The RFBF can be expressed by Taylor series expansion as

$$\boldsymbol{\phi}^* = \hat{\boldsymbol{\phi}} + \hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN} + \mathrm{O}((\Omega\boldsymbol{x}_{NN})^2) \qquad (4.5)$$

$$\tilde{\boldsymbol{\phi}} = \hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN} + \mathrm{O}((\tilde{\Omega}\boldsymbol{x}_{NN})^2), \qquad (4.6)$$

where $\tilde{\boldsymbol{\phi}} = \boldsymbol{\phi}^* - \hat{\boldsymbol{\phi}}$, $\tilde{\Omega} = \Omega^* - \hat{\Omega}$ and

$$\hat{\phi}' = \begin{bmatrix} \hat{\phi}_1 & & O \\ & \ddots & \\ O & & \hat{\phi}_r \end{bmatrix}.$$

Therefore, the approximation error can be written in the vector form:

$$\begin{aligned} \tilde{\boldsymbol{f}} &= \boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}(\boldsymbol{x}_{NN}, \hat{\Omega}, \hat{C}) \\ &= \tilde{C}\tilde{\boldsymbol{\phi}} + \tilde{C}\hat{\boldsymbol{\phi}} + \hat{C}\tilde{\boldsymbol{\phi}} + \boldsymbol{d}(t). \end{aligned} \qquad (4.7)$$

Substitute (4.6) into (4.7), we can get

$$\begin{aligned} \tilde{\boldsymbol{f}} &= \tilde{C}\hat{\boldsymbol{\phi}} + \hat{C}\hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN} \\ &\quad + (\tilde{C}\hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN} + \Omega^*\mathrm{O}((\tilde{\Omega}\boldsymbol{x}_{NN})^2) + \boldsymbol{d}(t)) \\ &= \tilde{C}\hat{\boldsymbol{\phi}} + \hat{C}\hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN} + \boldsymbol{d}_{NN}, \end{aligned} \qquad (4.8)$$

where

$$\begin{aligned} \tilde{C} &= C^* - \hat{C}, \boldsymbol{d}_{NN} \\ &= \tilde{C}\hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN} + C^*\mathrm{O}((\tilde{\Omega}\boldsymbol{x}_{NN})^2) + \boldsymbol{d}(t). \end{aligned}$$

In the following, norm on vector or matrix, $\|\bullet\|$, is the Frobenius norm [16]. The upper bound of the norms of $\mathrm{O}((\tilde{\Omega}\boldsymbol{x}_{NN})^2)$ is given by

$$\begin{aligned} \|\mathrm{O}((\tilde{\Omega}\boldsymbol{x}_{NN})^2)\| &= \|\tilde{\boldsymbol{\phi}} - \hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN}\| \\ &\leqslant \|\tilde{\boldsymbol{\phi}}\| + \|\hat{\phi}'\|\|\tilde{\Omega}\|\|\boldsymbol{x}_{NN}\| \\ &\leqslant \kappa_1 + \kappa_2\|\tilde{\Omega}\|\|\boldsymbol{x}_{NN}\|, \end{aligned} \qquad (4.9)$$

where $\|\tilde{\boldsymbol{\phi}}\| \leqslant \kappa_1$ and $\|\hat{\boldsymbol{\phi}}'\| \leqslant \kappa_2$. And, the upper bound of the norm of the ith element of $\boldsymbol{d}_{NN}$ is as follows:

$$\begin{aligned} |\boldsymbol{d}_{NNi}| &= \|\boldsymbol{d}_{NN}\| \\ &\leqslant \|\tilde{C}\hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN} + C^*\mathrm{O}((\tilde{\Omega}\boldsymbol{x}_{NN})^2) + \boldsymbol{d}(t)\| \\ &\leqslant \|\tilde{C}\hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN}\| + \|C^*\|(\kappa_1 + \kappa_2\|\tilde{\Omega}\|\|\boldsymbol{x}_{NN}\|) \\ &\leqslant \|\tilde{C}\|\kappa_2\|\tilde{\Omega}\|\|\boldsymbol{x}_{NN}\| + \kappa_C(\kappa_1 + \kappa_2\|\tilde{\Omega}\|\|\boldsymbol{x}_{NN}\|) \\ &\leqslant (\kappa_C + \|\hat{C}\|)\kappa_2(\kappa_\Omega + \|\hat{\Omega}\|)\|\boldsymbol{x}_{NN}\| \\ &\quad + \kappa_C(\kappa_1 + \kappa_2(\kappa_\Omega + \|\hat{\Omega}\|)\|\boldsymbol{x}_{NN}\|) \\ &= H_i\zeta, \end{aligned} \qquad (4.10)$$

where H_i is the ith row of H, $\|\tilde{\Omega}\| \leqslant \|\Omega^*\| + \|\hat{\Omega}\|$, $\|\tilde{C}\| \leqslant \|C^*\| + \|\hat{C}\|$, $\|\Omega^*\| \leqslant \kappa_\Omega$, $\|C^*\| \leqslant \kappa_C$, $\|\tilde{\boldsymbol{\phi}}\| \leqslant \kappa_1$, $\|\hat{\boldsymbol{\phi}}'\| \leqslant \kappa_2$ and $\boldsymbol{\zeta}^{\mathrm{T}} = [1\ \|\boldsymbol{x}_{NN}\|\ \|\hat{\Omega}\|\|\boldsymbol{x}_{NN}\|\ \|\hat{C}\|\|\boldsymbol{x}_{NN}\|\ \|\hat{\Omega}\|\|\hat{C}\|\|\boldsymbol{x}_{NN}\|]$ ($\kappa_1, \kappa_2, \kappa_\Omega, \kappa_C$ are constants). The robustifying term estimating $\|\boldsymbol{d}_{NN}\|$ in (4.4) is as

$$\hat{\boldsymbol{d}} = Z\hat{H}\zeta, \qquad (4.11)$$

where $Z = \mathrm{diag}(\mathrm{sgn}(s_{1\Delta}), \mathrm{sgn}(s_{2\Delta}), \mathrm{sgn}(s_{3\Delta}))$ and sgn is a sign function:

$$\mathrm{sgn}(x) = \begin{cases} 1, & x \geqslant 0 \\ -1, & x < 0. \end{cases}$$

The parameters $\hat{C}$, $\hat{\Omega}$ and $\hat{H}$ are updated by the following adaptation law:

$$\dot{\hat{C}}^{\mathrm{T}} = K_C\hat{\phi}\boldsymbol{s}_\Delta^{\mathrm{T}} \qquad (4.12)$$

$$\dot{\hat{\Omega}}^{\mathrm{T}} = K_\Omega\boldsymbol{x}_{NN}\boldsymbol{s}_\Delta^{\mathrm{T}}\hat{C}\hat{\phi}' \qquad (4.13)$$

$$\dot{\hat{H}}^{\mathrm{T}} = K_H\boldsymbol{\zeta}\boldsymbol{s}_\Delta^{\mathrm{T}}Z, \qquad (4.14)$$

where K_C, K_Ω and K_H are positive symmetric constant matrices determining the adaptation rate. Hence, the time derivative of error metric (3.12) can be rewritten as

$$\dot{\boldsymbol{s}}_i(t) = -K\boldsymbol{s}_i(t) + \tilde{C}\hat{\boldsymbol{\phi}} + \hat{C}\hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN} - \boldsymbol{v}_{\mathrm{TA}} + \tilde{\boldsymbol{d}}(t), \qquad (4.15)$$

where $\tilde{\boldsymbol{d}}(t) = \boldsymbol{d}_{NN}(t) - \hat{\boldsymbol{d}}(t)$.

4.2. Stability analysis

A stability theorem is presented for the control law (4.2)–(4.4).

Theorem 1. *Consider the dynamic equations of a BTT missile* (2.1) *with the control law* (4.2)–(4.4). *All states in the* (2.1) *will remain bounded and the tracking errors will approach zero.*

Proof. Consider the Lyapunov function candidate

$$V(t)=\tfrac{1}{2}(s_\Delta^{\mathrm{T}}s_\Delta+\mathrm{Tr}(\tilde{\Omega}^{\mathrm{T}}K_\Omega^{-1}\tilde{\Omega}) \\ +\mathrm{Tr}(\tilde{C}^{\mathrm{T}}K_C^{-1}\tilde{C})+\mathrm{Tr}(\tilde{H}^{\mathrm{T}}K_H^{-1}\tilde{H})). \quad (4.16)$$

Evaluating the time derivative of $V(t)$ along the trajectories of the adaptive learning laws of Eqs. (4.12)–(4.14), $\dot{V}=0$, when $|s_i|\leqslant\Delta_i$, $i=1,2,3$. When $|s_i|>\Delta_i$ for each i, since

$$\frac{\mathrm{d}}{\mathrm{d}t}s_{i\Delta}^2=2s_{i\Delta}\dot{s}_i \quad \text{and} \quad s_{i\Delta}\,\mathrm{sat}(s_i/\Delta_i)=|s_{i\Delta}|,$$

the time derivative of $V(t)$ is given by

$$\begin{aligned}\dot{V}&=\boldsymbol{s}_\Delta^{\mathrm{T}}(-K\boldsymbol{s}+\tilde{C}\hat{\boldsymbol{\phi}}+\hat{C}\hat{\phi}'\tilde{\Omega}\boldsymbol{x}_{NN}-\boldsymbol{v}_{\mathrm{TA}}+\tilde{\boldsymbol{d}})\\ &\quad-\mathrm{Tr}(\tilde{C}\hat{\boldsymbol{\phi}}\boldsymbol{s}_\Delta^{\mathrm{T}})-\mathrm{Tr}(\tilde{\Omega}\boldsymbol{x}_{NN}\boldsymbol{s}_\Delta^{\mathrm{T}}\hat{C}\hat{\phi}')-\mathrm{Tr}(\tilde{H}^{\mathrm{T}}\boldsymbol{\zeta}\boldsymbol{s}_\Delta^{\mathrm{T}}Z)\\ &\leqslant-\boldsymbol{s}_\Delta^{\mathrm{T}}K\boldsymbol{s}_\Delta-|\boldsymbol{s}_\Delta^{\mathrm{T}}K\boldsymbol{\Delta}|-\boldsymbol{s}_\Delta^{\mathrm{T}}\boldsymbol{v}_{\mathrm{TA}}+\boldsymbol{s}_\Delta^{\mathrm{T}}Z\tilde{H}\boldsymbol{\zeta}\\ &\quad-\mathrm{Tr}(\tilde{H}^{\mathrm{T}}\boldsymbol{\zeta}\boldsymbol{s}_\Delta^{\mathrm{T}}Z)\\ &=-\boldsymbol{s}_\Delta^{\mathrm{T}}K\boldsymbol{s}_\Delta-|\boldsymbol{s}_\Delta^{\mathrm{T}}K\boldsymbol{\Delta}|-\boldsymbol{s}_\Delta^{\mathrm{T}}\boldsymbol{v}_{\mathrm{TA}}\leqslant 0,\end{aligned} \quad (4.17)$$

where $\boldsymbol{\Delta}=[\Delta_1\ \ \Delta_2\ \ \Delta_3]^{\mathrm{T}}$.

Therefore, if $s_{i\Delta}$, all $\tilde{c}_{ij}$'s and all $\tilde{\omega}_{ij}$'s are bounded at initial time $t=0$, they will remain bounded for all time $t>0$. If $\tilde{y}_i(0)$ is bounded, then $\tilde{y}_i(t)$ is also bounded for all time t, and since $y_{id}(t)$ is bounded specified, $y_i(t)$ is as well. Next, we will show that $s_{i\Delta}\to 0$ as $t\to\infty$. It is easy to show by Barbalat's lemma:

$$V_1(t)=V(t) \\ -\int_0^t[\dot{V}(\tau)+(\boldsymbol{s}_\Delta^{\mathrm{T}}K\boldsymbol{s}_\Delta+\boldsymbol{s}_\Delta^{\mathrm{T}}K\boldsymbol{\Delta}+\boldsymbol{s}_\Delta^{\mathrm{T}}\boldsymbol{v}_{\mathrm{TA}})]\,\mathrm{d}\tau \quad (4.18)$$

with

$$\dot{V}_1(t)=-\boldsymbol{s}_\Delta^{\mathrm{T}}K\boldsymbol{s}_\Delta+\boldsymbol{s}_\Delta^{\mathrm{T}}K\boldsymbol{\Delta}+\boldsymbol{s}_\Delta^{\mathrm{T}}\boldsymbol{v}_{\mathrm{TA}}. \quad (4.19)$$

Thus, one has that every term in (3.34) is bounded, hence $s_{i\Delta}$ is bounded, s_i is as well. This implies that $\dot{V}_1(t)$ is a uniformly continuous function of time. Since V_1 is bounded by 0, and $\dot{V}_1\leqslant 0$ for all time t, Barbalat's lemma is applied to prove that $\dot{V}_1\to 0$ and hence $s_{i\Delta}\to 0$ as $t\to\infty$. □

4.3. Terminal attractor

So far, we have discussed the autopilot (controller) architecture and the stability issue. However, the terminal attractors have not been mentioned yet. The concept of terminal attractors can be easily explained by the equation $\dot{x}=ax^b$. An equilibrium point at $x=0$ will be approached by transients in finite time under the conditions: $a<0$ and $0<b<1$, and repellers appear when $a>0$. In this section, we will use a theorem to state the existence of terminal attractors in our autopilot. Based upon Theorem 1, we can make the following assumption:

Assumption 2. Assume that there are N rules in fuzzy rule base, $|\tilde{c}_{ij}|\leqslant M_{c1}$, $|\hat{c}_{ij}|\leqslant M_{c2}$, and $|\sum_{j=1}^N\tilde{c}_{ij}\phi_j^*+\sum_{j=1}^N\hat{c}_{ij}\hat{\phi}_j+\tilde{d}_i|\leqslant N(M_{c1}+M_{c2})+\varepsilon_i$.

Eq. (3.12) can be of the form

$$\dot{\boldsymbol{s}}(t)=-K\boldsymbol{s}(t)+\tilde{C}\boldsymbol{\phi}^*+\hat{C}\tilde{\boldsymbol{\phi}}-\boldsymbol{v}_{\mathrm{TA}}+\tilde{\boldsymbol{d}}. \quad (4.20)$$

Because $0\leqslant\phi_j\leqslant 1$ and all $\tilde{c}_{ij}$'s are bounded as stated in Theorem 1, Assumption 2 is reasonable. Define the squared error $E=\frac{1}{2}s_{i\Delta}^2(t)$ as the error measure for the control task. Now, a theorem is presented for explaining the existence of the terminal attractor in $\mathrm{d}E/\mathrm{d}t$.

Theorem 2. *Applying the control law* (4.1)–(4.4) *to the dynamic equations* (2.1), $s_{i\Delta}=0$ *is a terminal attractor of* $\dot{E}$.

Proof. If $|s_i|\leqslant\Delta_i$ or $s_{i\Delta}=0$, then

$$\frac{\mathrm{d}}{\mathrm{d}t}s_{i\Delta}^2=2\dot{s}_is_{i\Delta}=0.$$

Next, we will show that when $s_{i\Delta} \neq 0$,

$$\frac{\mathrm{d}}{\mathrm{d}t} s_{i\Delta}^2 = -\eta_i (s_{i\Delta}^2)^{1/2} - k_{\mathrm{tai}} (s_{i\Delta}^2)^{(1+\alpha_i)/2\alpha_i}.$$

When $|s_i| > \Delta_i$ or $s_{i\Delta} \neq 0$, Assumption 2 is satisfied and there exist some k_{D_i} and Δ_i such that

$$|k_{D_i} s_i| > k_{D_i} \Delta_i > N(M_{c1} + M_{c2}) + \varepsilon_i$$
$$\geqslant \left| \sum_{j=1}^{N} \tilde{c}_{ij} \phi_j^* + \sum_{j=1}^{N} \hat{c}_{ij} \hat{\phi}_j + \tilde{d}_i \right|. \qquad (4.21)$$

The equation

$$|k_{D_i} s_i| > \left| \sum_{j=1}^{N} \tilde{c}_{ij} \phi_j^* + \sum_{j=1}^{N} \hat{c}_{ij} \hat{\phi}_j + \tilde{d}_i \right|$$

implies that

$$\mathrm{sgn}(s_i) = \mathrm{sgn}(s_{i\Delta})$$
$$= \mathrm{sgn}\left(k_{D_i} s_i + \sum_{j=1}^{N} \tilde{c}_{ij} \phi_j^* + \sum_{j=1}^{N} \tilde{c}_{ij} \phi_j^* + \tilde{d}_i \right),$$

where sgn is a sign function. If the sign is different, then the terminal attractor becomes repeller.

$$\frac{\mathrm{d}}{\mathrm{d}t} s_{i\Delta}^2 = 2 s_{i\Delta} \dot{s}_i$$
$$= -s_{i\Delta} \left(k_{D_i} s_i + \sum_{j=1}^{N} \tilde{c}_{ij} \phi_j^* + \sum_{j=1}^{N} \hat{c}_{ij} \tilde{\phi}_j + \tilde{d}_i \right)$$
$$- k_{\mathrm{tai}} s_{i\Delta}^{(1+\alpha_i)/\alpha_i}$$
$$= -\theta_i (s_{i\Delta}^2)^{1/2} - k_{\mathrm{tai}} (s_{i\Delta}^2)^{(1+\alpha_i)/2\alpha_i}, \qquad (4.22)$$

where $\theta_i = k_{D_i} s_i + \sum_{j=1}^{N} \tilde{c}_{ij} \phi_j^* + \sum_{j=1}^{N} \hat{c}_{ij} \tilde{\phi}_j + \tilde{d}_i$. Therefore, $s_{i\Delta} = 0$ is a terminal attractor. □

5. Simulations

In this section, the adaptive fuzzy autopilot, which is developed in last section, is applied to BTT missiles. There are two simulation results, which take the practical restrictions into consideration, to demonstrate the effectiveness of the autopilot.

Example 5.1. The main purpose of this example is to show the power of the terminal attractor controller. In this example, we compare three autopilots: one is our autopilot, another is our autopilot without terminal attractor controller and the other is Su's adaptive FBF controller [29]. Su's controller is a similar approach; however, Su's controller is only capable of tuning the weights of defuzzifier.

In this simulation, during the interval between 0 and 2 s, the desired output signals are $\Phi_c = 135°$, $A_{yc} = 0\mathrm{G}$, and $A_{zc} = -15\mathrm{G}$, that is, the missile rolls 135° and is given a force to climb. After 2 s for climbing, the time interval between 2 and 4 s, the force for climbing is released and the desired command is $\Phi_c = 0°$, $A_{yc} = 0\mathrm{G}$, and $A_{zc} = 0\mathrm{G}$. The specifications are rise time $\leqslant 0.5$ s and overshoot $\leqslant 10\%$. The simulation results are shown in Fig. 6. The rise time of the autopilot with terminal attractor is significantly better than the other two controllers. The number of rules in the RFBF network is only 50 and each rule has 12 input variables (700 adjustable parameters). As to the Su's adaptive FBF controller, there are 6795 control rules [20]. Therefore, in comparison with the Su's controller, our autopilot can use smaller size to achieve better performance from the results of simulation. All simulations run on a Pentium 75 MHz PC. The RFBF simulations (with and without terminal attractor) take about 1 min, while the Su's controller takes about 8 min. Control rules of the RFBF network, which have the same format as described in Section 3, are not listed since each rule has too many variables. For each rule, the variance matrix and the center vector can be derived by $S_j = \Lambda_j' = \mathrm{diag}(\lambda_{j1}, \ldots, \lambda_{jn})$ and $\boldsymbol{\eta}_j = -\Gamma_{22}^{-1} \Gamma_{21}$.

Example 5.2. In traditional gain scheduling controller, the controller is highly dependent on the LTI controller at each fixed operating point. In our approach, complexity is reduced for only one autopilot is used. Another autopilot design in [17], the design methodology is highly dependent upon the geometric parameters and the aerodynamics conditions. Our approach is more simpler and easier to design.

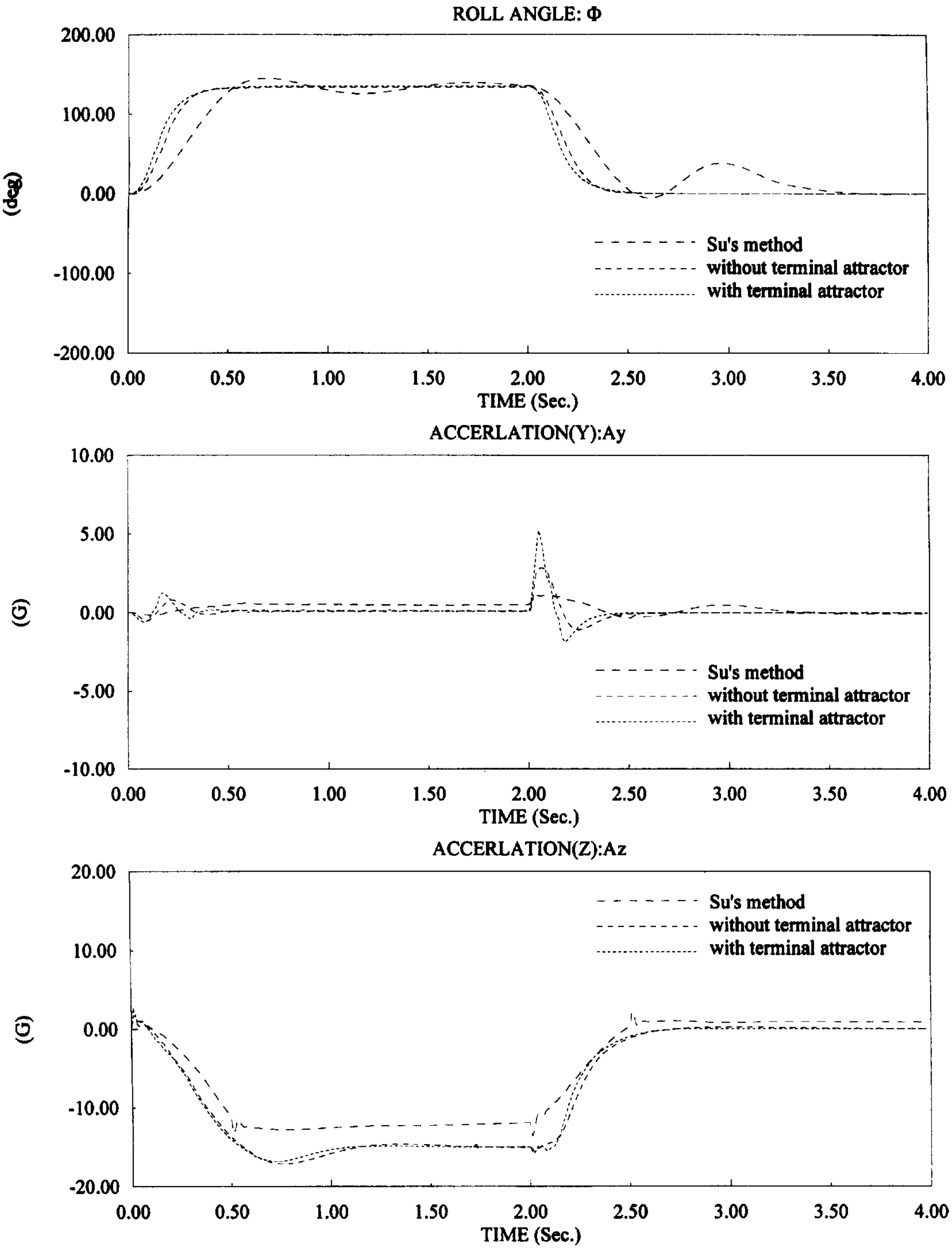


Fig. 6. Simulation results of Example 5.1.

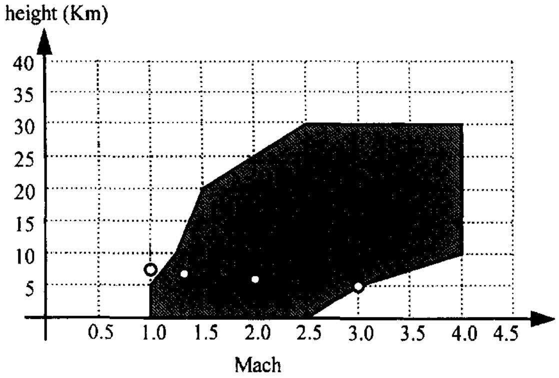


Fig. 7. The flight envelop figure (each circle represents a flight condition).

Consider that aerodynamic pressure $QS = 0.5\rho\|V_{\mathrm{m}}\|$, where ρ is the air density, is affected by air density and velocity of the missile. However, the air density is determined by the height of missile. Therefore, the flight condition can be programmed by the two chosen flight conditions chosen in this example. The desired operating range is for Mach number from 0.5 to 4.5 and the height is from 0 to 35 km. The angle of attack, α, is from 0 to 40° and sideslip angle, β, is as small as possible. In this example, numerical simulation results are conducted to evaluate the performance of the proposed autopilot under various operating ranges. The desired commands of the four flight conditions are (Fig. 7):

(I) $\Phi_c = 0°$, $A_{yc} = 0\mathrm{G}$, and $A_{zc} = -50\mathrm{G}$,

(II) $\Phi_c = 135°$, $A_{yc} = 0\mathrm{G}$, and $A_{zc} = -20\mathrm{G}$,

(III) $\Phi_c = 90°$, $A_{yc} = 0\mathrm{G}$, and $A_{zc} = -10\mathrm{G}$,

(IV) $\Phi_c = 0°$, $A_{yc} = 0\mathrm{G}$, and $A_{zc} = -5\mathrm{G}$.

The size of the RFBF network is the same as Example 1; that is, the number of rules is also 50 and 700 adjustable parameters which are initialized randomly. The simulation result is shown in Fig. 8. This simulation takes about 15 min on Pentium 75 MHz PC. The expected flight attitude is that the missile rolls and climbs until about 6800 km, and then keeps the height and roll angle is zero; finally, the missile climbs again. Fig. 9 shows that the trajectories of states meet the specified trajectories.

6. Conclusions

In this paper, an adaptive fuzzy autopilot is developed for BTT missile. The autopilot design is based on the proposed RFBF network which is capable of incorporating experts' experience into the autopilot design. The self-organizing rotated fuzzy basis function control system used a smaller network size than other neural fuzzy systems to achieve the desired performance. Taking the uncertainties into consideration, a robust control technique is employed to reject the internal and external disturbances. Further, we also combine the terminal attractor controller into the adaptive fuzzy controller that provides a way to accelerate the convergence rate successfully. As a result, the autopilot does not need large amount of rules to track the desired commands using the proposed RFBF network controller.

Appendix A

A.1. Dynamic equations of BTT missiles with actuators

The complete 6-DOF missile dynamic equations, which include attitude dynamics and translational dynamics, are derived in [35]. The attitude dynamics dominate the orientation and the angular velocity of the body of the missile. As to translational dynamics, they dominate the position and the translational velocity of the center of mass of the missile. Fig. 10 shows the relationship between inertial frame and missile body frame. The notations used in the mathematical dynamic model are as follows. The lowercase letters are used to denote variables in inertial coordinates, whereas capital letters are used to denote variables in missile body coordinates.

Notations

$\{b_x, b_y, b_z\}$	a right-handed orthonormal basis of body coordinate frame which is attached to the center of mass, C, of the missile, where b_x, b_y are on the longitudinal and lateral axis, respectively
$\{e_x, e_y, e_z\}$	a right-handed orthonormal basis of inertial coordinate frame

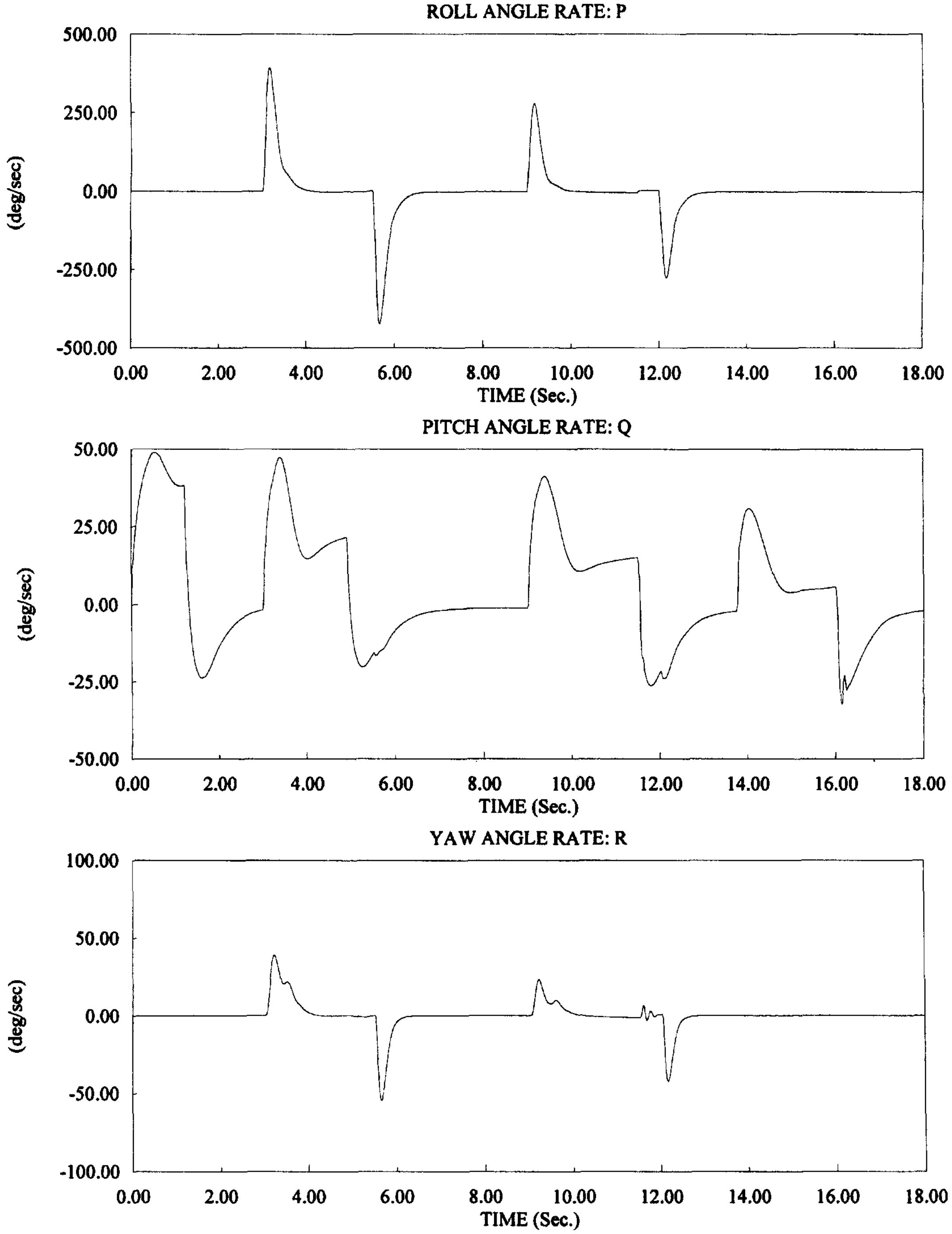
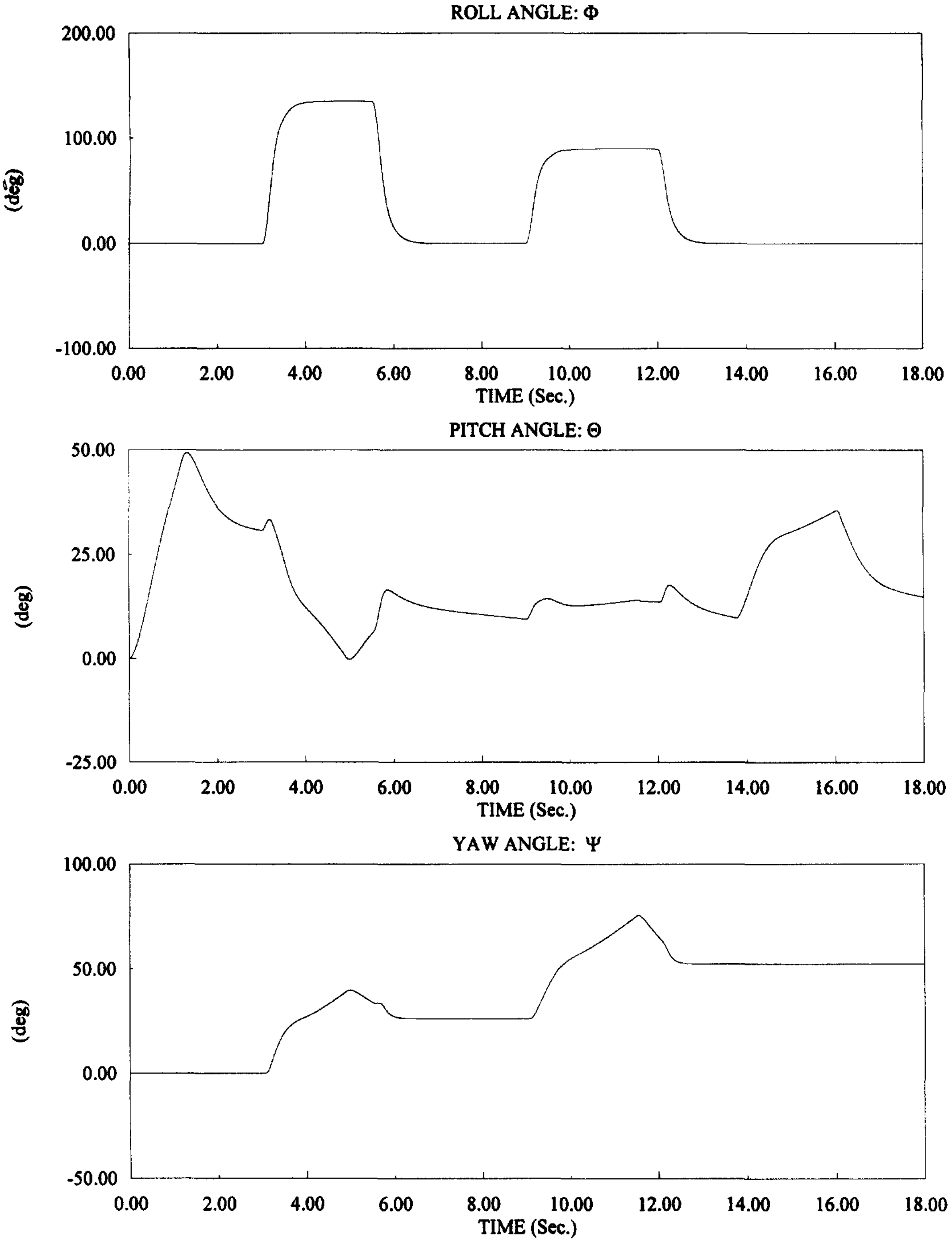


Fig. 8. States of BTT missiles of Example 5.2.


ROLL ANGLE: Φ

PITCH ANGLE: Θ

YAW ANGLE: Ψ

Fig. 8. (cont.).

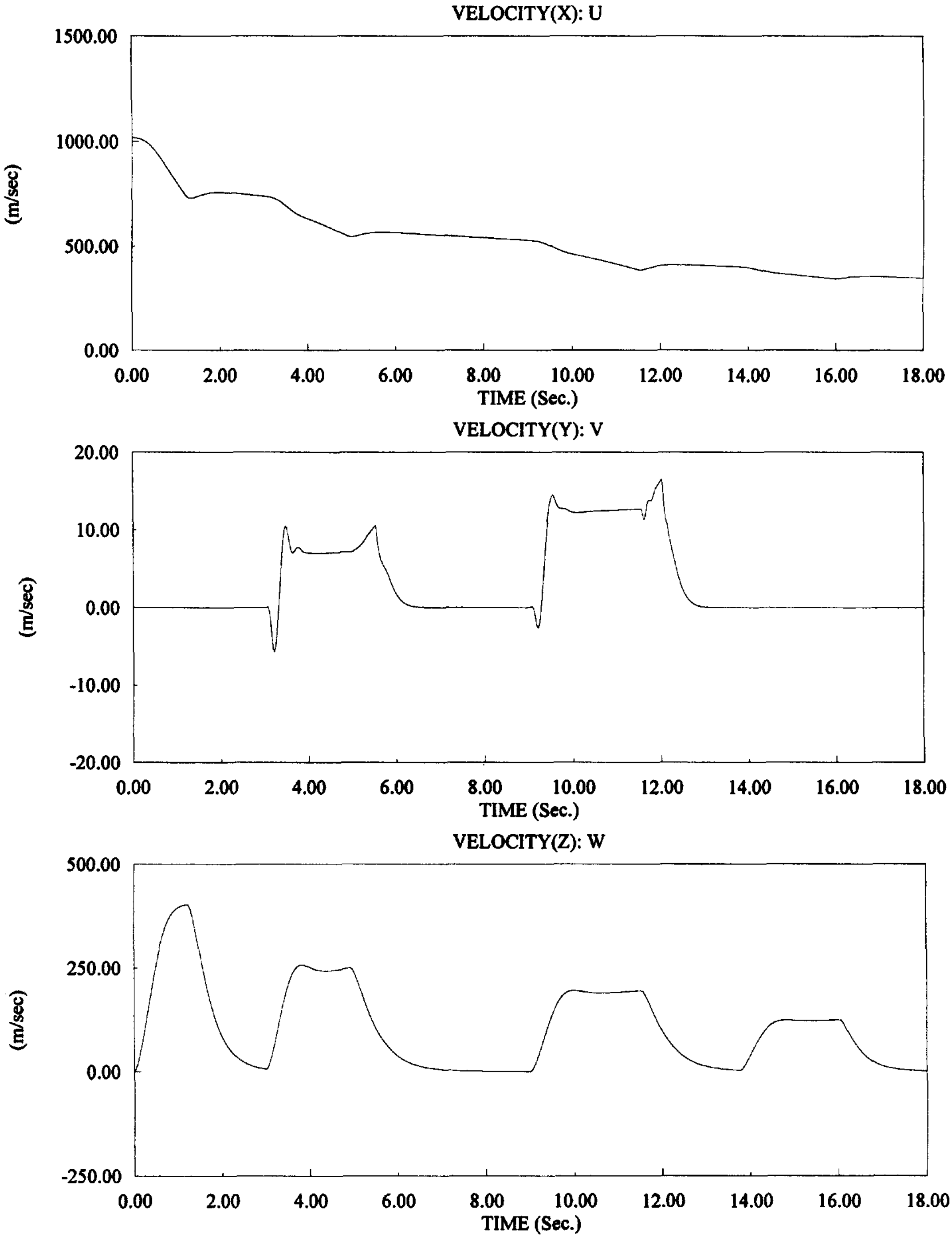

VELOCITY(X): U
1500.00
1000.00
500.00
0.00
(m/sec)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)
VELOCITY(Y): V
20.00
10.00
0.00
-10.00
-20.00
(m/sec)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)
VELOCITY(Z): W
500.00
250.00
0.00
-250.00
(m/sec)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)

Fig. 8. (cont.).

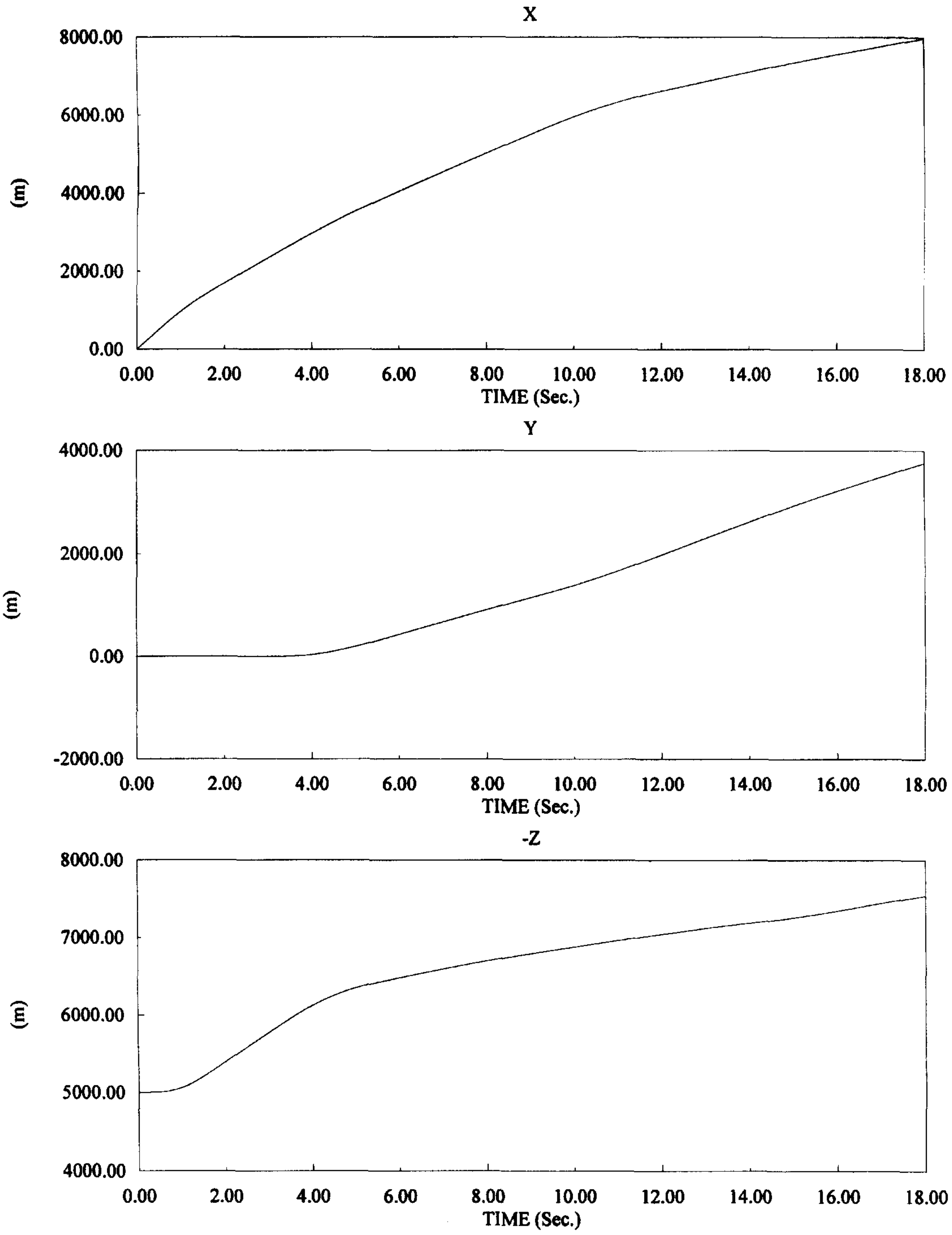

X
8000.00
6000.00
4000.00
2000.00
0.00
(m)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)
Y
4000.00
2000.00
0.00
-2000.00
(m)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)
-Z
8000.00
7000.00
6000.00
5000.00
4000.00
(m)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)

Fig. 8. (cont.).

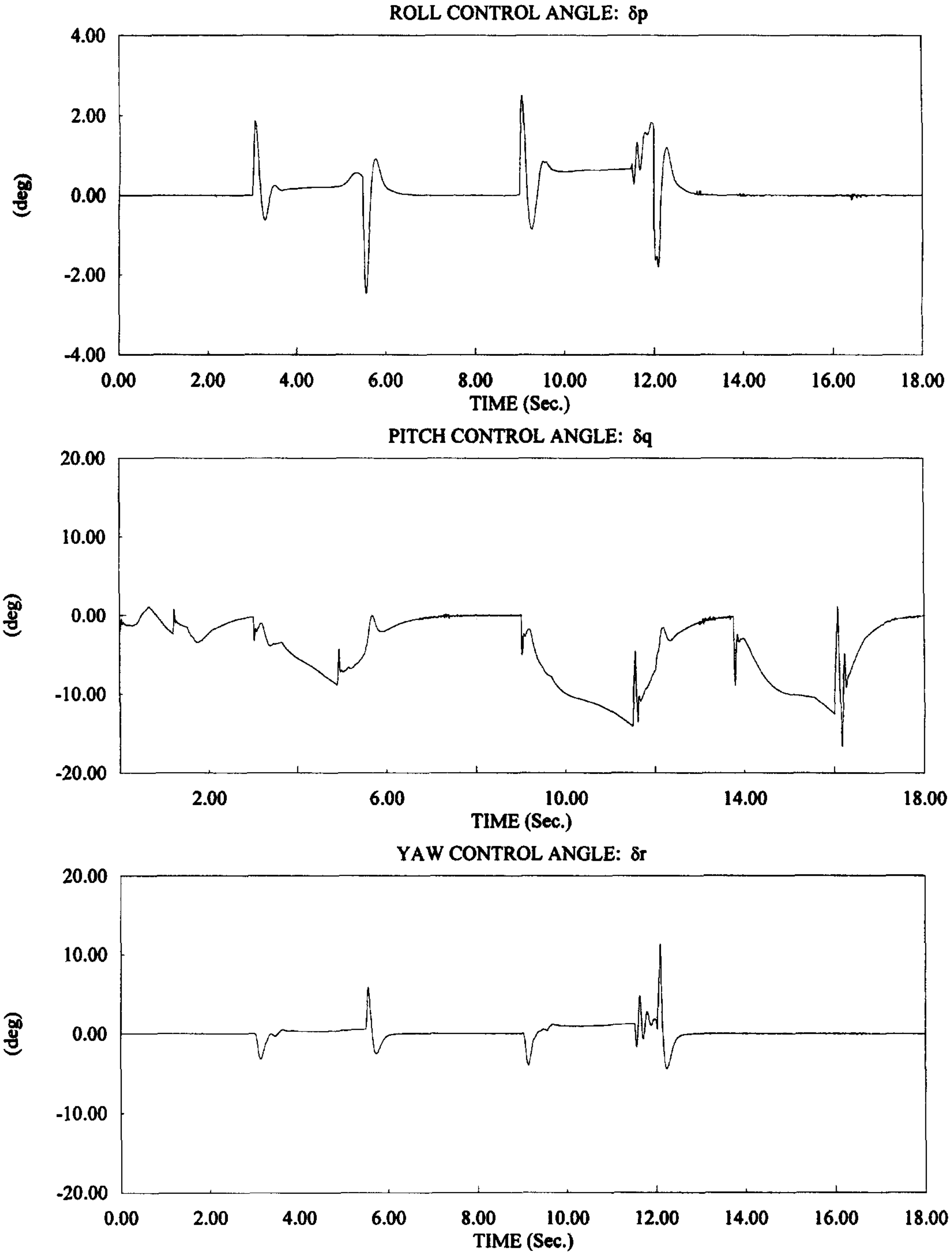

ROLL CONTROL ANGLE: δp
4.00
2.00
0.00
-2.00
-4.00
(deg)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)
PITCH CONTROL ANGLE: δq
20.00
10.00
0.00
-10.00
-20.00
(deg)
2.00
6.00
10.00
14.00
18.00
TIME (Sec.)
YAW CONTROL ANGLE: δr
20.00
10.00
0.00
-10.00
-20.00
(deg)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)

Fig. 8. (cont.).

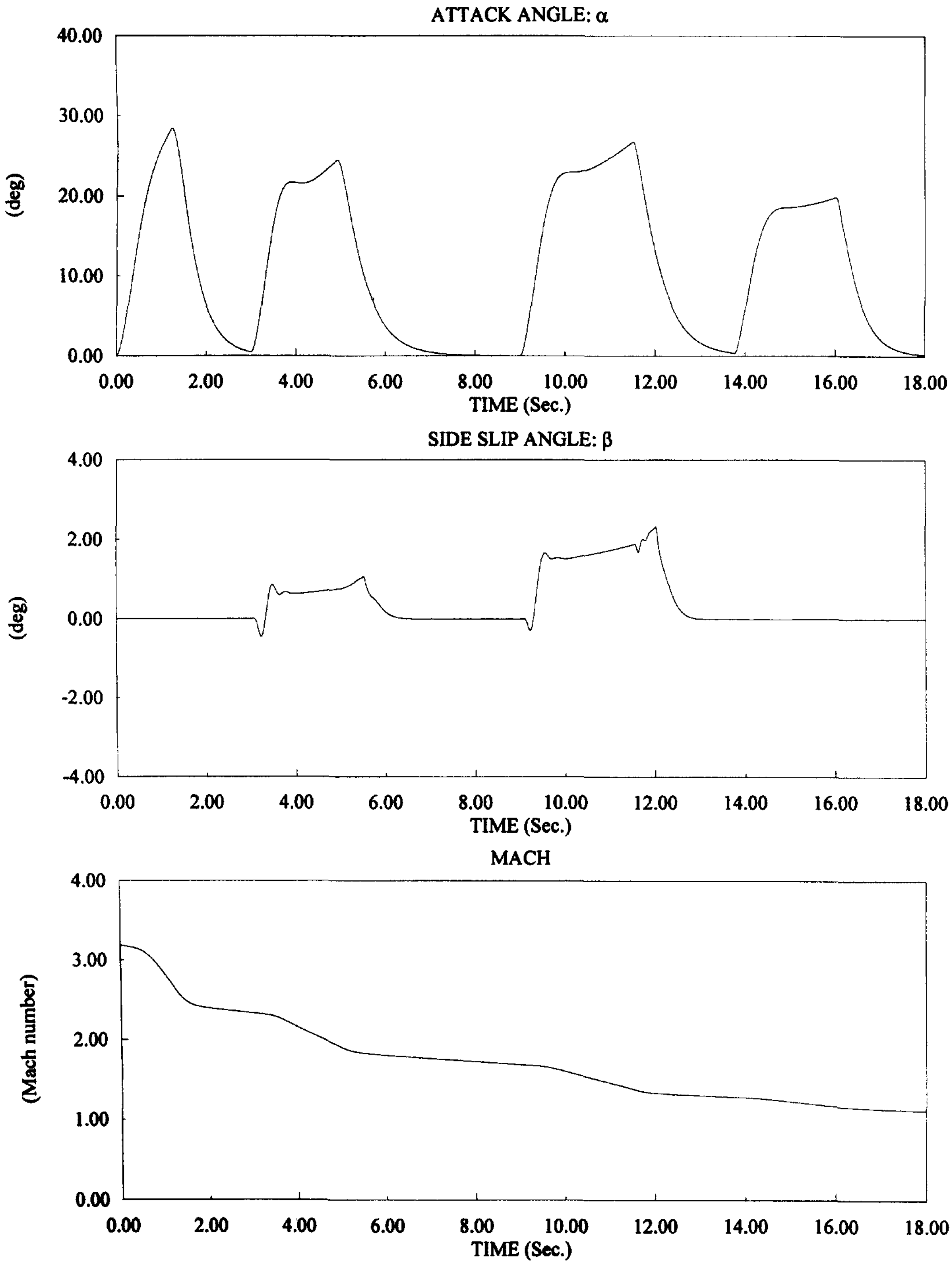

ATTACK ANGLE: α
40.00
30.00
20.00
10.00
0.00
(deg)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)
SIDE SLIP ANGLE: β
4.00
2.00
0.00
-2.00
-4.00
(deg)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)
MACH
4.00
3.00
2.00
1.00
0.00
(Mach number)
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)

Fig. 8. (cont.).

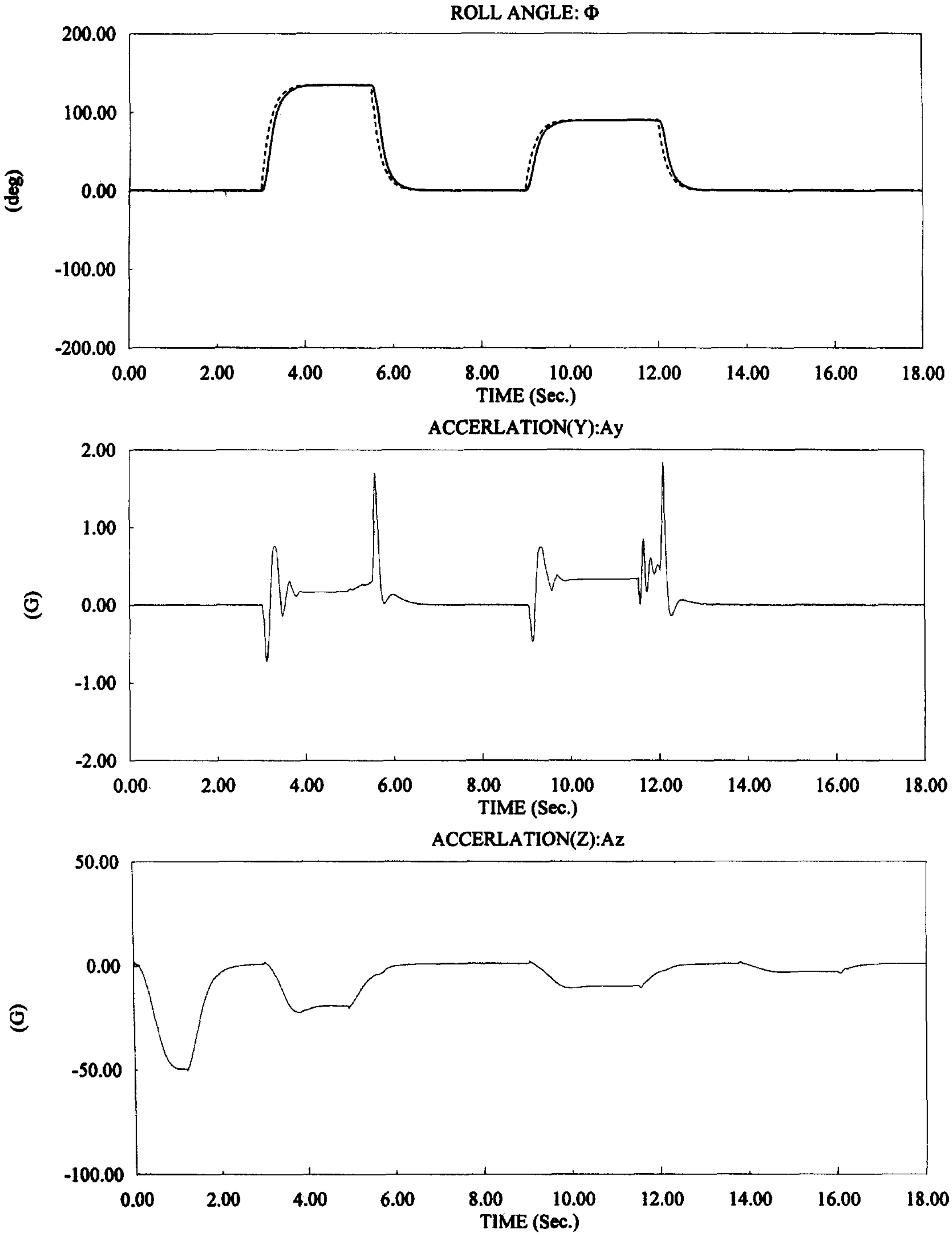

ROLL ANGLE: Φ
(deg)
200.00
100.00
0.00
-100.00
-200.00
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)
ACCERLATION(Y):Ay
(G)
2.00
1.00
0.00
-1.00
-2.00
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)
ACCERLATION(Z):Az
(G)
50.00
0.00
-50.00
-100.00
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
TIME (Sec.)

Fig. 8. (cont.).

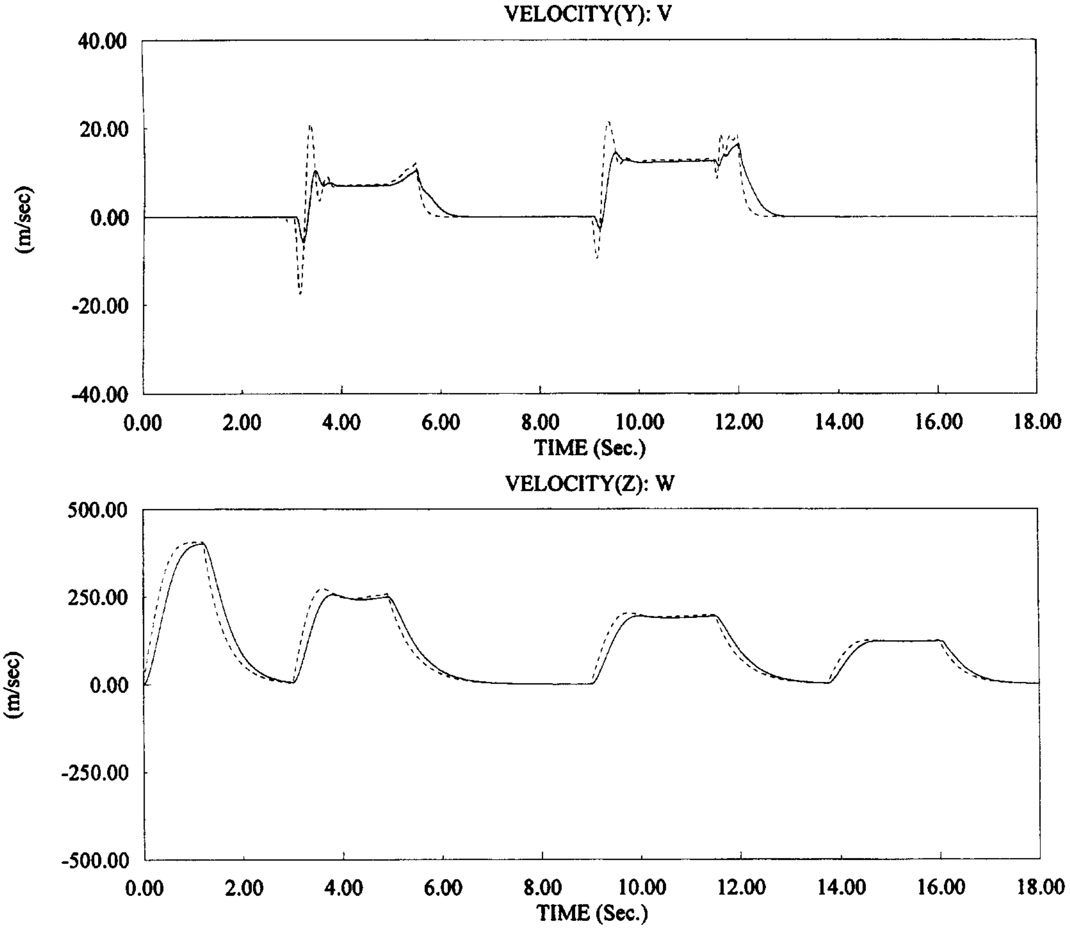


Fig. 9. Desired trajectory (dashed line) and trajectory with proposed autopilot (solid line) of Example 5.2.

P, Q, R roll rate, pitch rate and yaw rate corresponding to the axes b_x, b_y and b_z, respectively (clockwise), (rad/s)

Φ body-axis roll angle measured from the downward vertical to b_z about the axis b_x (rad)

Θ body-axis pitch angle measured from the projection of b_x onto the horizontal plane to b_x (rad)

Ψ body-axis yaw angle measured between a fixed compass bearing and the projection of b_x onto the horizontal plane to b_x (rad)

$(X, Y, Z)^{\mathrm{T}}$ position vector of the center of mass of the missile transformed with respect to the inertial frame (m)

$(U, V, W)^{\mathrm{T}}$ velocity vector of the missile transformed with respect to the body frame (m/s)

$\delta_p, \delta_q, \delta_r$ aileron deflation angle, elevator deflation angle and rudder deflation angle, respectively (rad)

α, β attack angle and sideslip angle, respectively (rad)

T thrust (kg)

V_{m} magnitude of the missile velocity

M_x, M_y, M_z external torques corresponding to the directions b_x, b_y, b_z, respectively

F_x, F_y, F_z external forces corresponding to the directions b_x, b_y, b_z, respectively

$C_{F_x}, C_{F_y}, C_{F_z}$ total aerodynamic force coefficients corresponding to the directions b_x, b_y and b_z, respectively

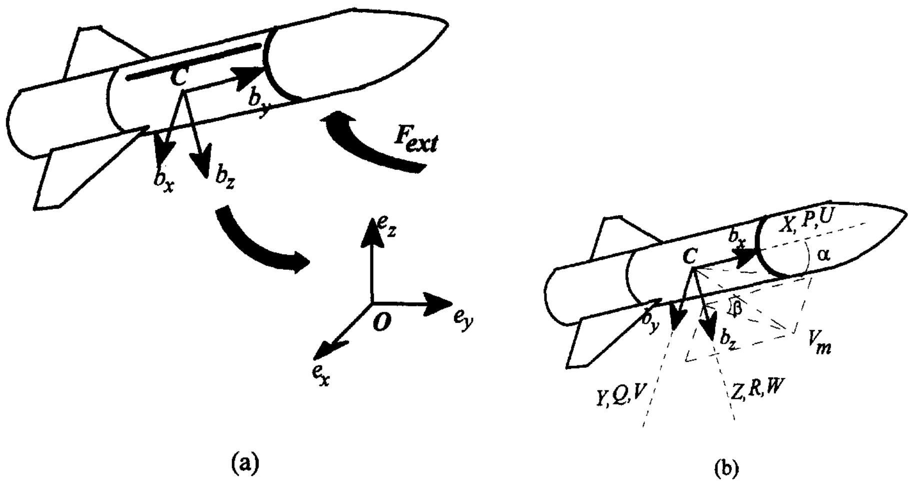


Fig. 10. (a) The relationship between the inertial frame and body frame. (b) The BTT missile diagram.

$C_{M_x}, C_{M_y}, C_{M_z}$ total moment coefficients corresponding to the directions b_x, b_y and b_z, respectively

A_x, A_y, A_z acceleration along the directions b_x, b_y, b_z at center of mass, respectively

G_x, G_y, G_z gravitational force corresponding to the directions b_x, b_y and b_z, respectively

I I is the moment of inertial tensor of the missile, where $I = \text{diag}\{I_{xx}, I_{yy}, I_{zz}\}$

M Mach number

m missile mass (kg)

S reference wing area (m^2)

d reference length (m)

QS dynamic pressure, i.e. $QS = 0.5\rho \| V_m \|$, where ρ is the air density

The detailed process of derivation, which can be obtained from [18], is omitted. The 6-DOF missile dynamic equations are written in the following state space form:

$$\dot{P} = \frac{I_{yy} - I_{zz}}{I_{xx}} QR + \frac{M_x}{I_{xx}},$$

$$\dot{Q} = \frac{I_{zz} - I_{yy}}{I_{yy}} RP + \frac{M_y}{I_{yy}},$$

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

$$\dot{\Phi} = P + (Q \sin \Phi + R \cos \Phi) \tan \Theta,$$

$$\dot{\Theta} = Q \cos \Phi - R \sin \Phi,$$

$$\dot{\Psi} = (Q \sin \Phi + R \cos \Phi)/\cos \Theta,$$

$$\dot{U} = RV - QW + \frac{1}{m} F_x,$$

$$\dot{V} = -RU + PW + \frac{1}{m} F_y, \qquad \text{(A.1)}$$

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

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = R_1(\Phi) R_2(\Theta) R_3(\Psi) \begin{bmatrix} U \\ V \\ W \end{bmatrix},$$

where R_1, R_2 and R_3, which are rotational matrices, are defined as follows:

$$R_1(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi & \cos \Phi \end{bmatrix},$$

$$R_2(\Theta) = \begin{bmatrix} \cos \Theta & 0 & \sin \Theta \\ 0 & 1 & 0 \\ -\sin \Theta & 0 & \cos \Theta \end{bmatrix},$$

$$R_3(\Psi) = \begin{bmatrix} \cos \Psi & -\sin \Psi & 0 \\ \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The external forces and torques caused by the actuators, aerodynamic and gravitational forces are specified as follows:

$$\begin{aligned}
F_x &= C_{F_x}\, QS\, S + T + G_x,\\
F_y &= C_{F_y}\, QS\, S + T + G_y,\\
F_z &= C_{F_z}\, QS\, S + T + G_z,\\
M_x &= C_{M_x}\, QS\, Sd,\\
M_y &= C_{M_y}\, QS\, Sd,\\
M_z &= C_{M_z}\, QS\, Sd.
\end{aligned} \tag{A.2}$$

Taking the actuator and throttle dynamics into consideration, the dynamics are written in the first-order form:

$$\begin{aligned}
\dot{\delta}_p &= -\omega_c \delta_p + \omega_c \delta_{pc},\\
\dot{\delta}_q &= -\omega_c \delta_q + \omega_c \delta_{qc},\\
\dot{\delta}_r &= -\omega_c \delta_r + \omega_c \delta_{rc},\\
\dot{T} &= -\omega_t T + \omega_t T_c,
\end{aligned} \tag{A.3}$$

where the constants ω_c and ω_t are the bandwidths of the actuator and throttle dynamic equations, respectively.

The nine elements of the gain matrix derived by input–output linearization technique are given as follows:

$$\begin{aligned}
g_{11}(\boldsymbol{x}) &= QS\, Sd \left(\frac{C_1}{I_{xx}} + \Theta \sin \Phi \frac{C_2}{I_{yy}} + \Theta \cos \Phi \frac{C_3}{I_{zz}} \right),\\
g_{12}(\boldsymbol{x}) &= QS\, Sd \Theta \sin \Phi \frac{C_4}{I_{yy}},\\
g_{13}(\boldsymbol{x}) &= QS\, Sd \left(\frac{C_5}{I_{xx}} + \Theta \cos \Phi \frac{C_6}{I_{zz}} \right),\\
g_{21}(\boldsymbol{x}) &= QS\, Sd \frac{C_7}{m},\\
g_{22}(\boldsymbol{x}) &= 0,\\
g_{23}(\boldsymbol{x}) &= QS\, Sd \frac{C_8}{m},\\
g_{31}(\boldsymbol{x}) &= QS\, Sd \frac{C_9}{m},\\
g_{32}(\boldsymbol{x}) &= QS\, Sd \frac{C_{10}}{m},\\
g_{33}(\boldsymbol{x}) &= 0,
\end{aligned} \tag{A.4}$$

where C_i's are aerodynamic coefficients which are complex functions of M and x and are not given here.

References

[1] J.A. Bossi and M.A. Langehough, Multivariable autopilot design for a bank-to-turn missile, *Proc. Amer. Control Conf.* (1988) 567–572.

[2] J.J. Buckley, Numerical relationships between neural networks, continuous functions, and fuzzy systems, *Fuzzy Sets and Systems* **60** (1993) 1–8.

[3] J.J. Buckley and Y. Hayashi, Hybrid neural nets can be fuzzy controllers and fuzzy expert systems, *Fuzzy Sets and Systems* **60** (1993) 135–142.

[4] S.K. Chang and K. Yuan, Optimal nonlinear tracking control for bank-to-turn asymmetric missiles using polynomial feedback, *Proc. Amer. Control Conf.* (1991) 2852–2857.

[5] S. Daley and K.F. Gill, Attitude control of a spacecraft using an extensive self-organizing fuzzy logic controller, *Proc. Inst. Mech. Engrs.* **201** (1987) 97–106.

[6] R. Fierro and F.L. Lewis, Control of a nonholonomic mobile robot using neural networks, *IEEE Internat. Symp. on Intelligent Control* (1995) 415–421.

[7] G.W. Irwin, Design of controller for bank-to-turn CLOS guidance using optimal control, *Proc. Amer. Control Conf.* (1986) 1143–1148.

[8] J.-S. Jang, Self-learning fuzzy controllers based on temporal backpropagation, *IEEE Trans. Neural Networks* **3** (1992) 714–723.

[9] J.-S. Jang, ANFIS: Adaptive-network-based fuzzy inference system, *IEEE Trans. Systems Man Cybernet.* **23** (1993) 665–685.

[10] J.M. Keller, R.R. Yager and H. Tahani, Neural network implementation of fuzzy logic, *Fuzzy Sets and Systems* **45** (1992) 1–12.

[11] H.M. Kim and J.M. Mendel, Fuzzy basis function: Comparisons with other basis functions, *IEEE Trans. Fuzzy Systems* **1** (1993) 158–168.

[12] S.J. Koffman and P.H. Meckl, Gaussian network variants: a preliminary study, *Proc. Internat. Joint Conf. Neural Networks* (1993) 523–528.

[13] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller, part I, *IEEE Trans. Systems Man Cybernet.* **20** (1990) 404–418.

[14] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller, part II, *IEEE Trans. Systems Man Cybernet.* **20** (1990) 419–435.

[15] S. Lee and R.M. Kil, A gaussian potential function network with hierarchically self-learning, *Neural Networks* **4** (1991) 207–224.

[16] F.L. Lewis, K. Liu and A. Yesildirek, Neural net robot controller with guaranteed tracking performance, *IEEE Trans. Neural Networks* **6** (1995) 703–715.

[17] K.-Y. Lian, L.-C. Fu, D.-M. Chuang and T.-S. Kuo, Nonlinear autopilot and guidance for a highly maneuverable missile, *Proc. Amer. Control. Conf.* (1991) 2293–2297.

[18] K.-Y. Lian, L.-C. Fu, D.-M. Chuang and T.-S. Kuo, Adaptive robust autopilot design for bank-to-turn aircraft, *Proc. Amer. Control Conf.* (1993) 1746–1750.

[19] C.T. Lin and C.S.G. Lee, Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems, *IEEE Trans. Fuzzy Systems* **2** (1995) 46–63.

[20] C.-K. Lin and S.-D. Wang, Adaptive fuzzy control of bank-to-turn missiles, *Proc. 22nd IEEE IECON* (Taipei, Taiwan, August 1996) 596–601.

[21] C.F. Lin and W.R. Yueh, Coordinated bank-to-turn autopilot design, *Proc. Amer. Control Conf.* (1985) 498–507.

[22] C.-C. Liu and F.-C. Chen, Adaptive control of non-linear continuous-time systems using neural networks–general relative degree and MIMO cases, *Internat. J. Control* **58** (1993) 317–335.

[23] K.S. Narendra and A.M. Annaswamy, *Stable Adaptive Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1989).

[24] J. Nie and D.A. Linkens, Learning control using fuzzified self-organizing radial basis function network, *IEEE Trans. Fuzzy Systems* **1** (1993) 280–287.

[25] W. Pedrycz, Fuzzy neural networks and neurocomputations, *Fuzzy Sets and Systems* **56** (1993) 1–28.

[26] T. Poggio and F. Girosi, Networks for approximation and learning, *Proc. IEEE* **78** (1990) 1481–1497.

[27] R.M. Sanner and J.-J.E. Slotine, Gaussian networks for direct adaptive control, *IEEE Trans. Neural Networks* **3** (1992) 837–863.

[28] J.-J.E. Slotine and W. Li, *Applied Nonlinear Control* (Prentice-Hall, Englewood Cliffs, NJ, 1991).

[29] C.-Y. Su and Y. Stepanenko, Adaptive fuzzy control of a class of nonlinear systems with fuzzy logic, *IEEE Trans. Fuzzy Systems* **2** (1994) 285–294.

[30] L.X. Wang, Stable adaptive fuzzy control of nonlinear systems, *IEEE Trans. Fuzzy Systems* **1** (1993) 146–155.

[31] S.D. Wang and C.H. Hsu, Terminal attractor learning algorithms for backpropagation neural networks, *Proc. Int. Joint Conf. Neural Networks*, Singapore (1991) 183–189.

[32] L.X. Wang and J.M. Mendel, Fuzzy basis functions, universal approximation, and orthogonal least square learning, *IEEE Trans. Neural Networks* **3** (1992) 807–814.

[33] D.E. Williams, A computer-aided autopilot design procedure for a high performance bank-to-turn controlled tactical missile, *Proc. AIAA Guidance, Navigation and Control Conf.* (1987) 498–507.

[34] D.E. Williams, B. Friedland and A.N. Madiwale, Modern control theory for design of autopilots for bank-to-turn missiles, *J. Guidance Control Dynam.* **10** (1987) 378–386.

[35] K.A. Wise, Bank-to-turn missile autopilot design using loop transfer recovery, *Journal Guidance Control Dynam.* **13** (1990) 145–152.

[36] M. Zak, Terminal attractors in neural networks, *Neural Networks* **2** (1989) 259–274.