JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 24, No. 4, July–August 2001

Adaptive Fuzzy Gain Scheduling in Guidance System Design

Chun-Liang Lin*
Feng Chia University, Taichung 40724, Taiwan, Republic of China
and
Huai-Wen Su†
National Tsing Hua University, Hsinchu 30043, Taiwan, Republic of China

A dynamic backpropagation training algorithm for an adaptive fuzzy gain scheduling feedback control scheme was developed. This novel design methodology uses a Takagi–Sugeno fuzzy system to represent the fuzzy relationship between the scheduling variables and controller parameters (Takagi, T., and Sugeno, M., "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 15, No. 1, 1985, pp. 116–132). Direct realistic extension applicable to the guidance system design is introduced. This application relates to terminal guidance design for guided missiles. Mach number, altitude, and time to go are used as measured, time-varying exogenous scheduling variables injected into the guidance law. Results from homing-loop simulations show that the presented approach offers better terminal guidance performance than the conventional proportional navigation guidance design, that is, less control effort and a smaller miss distance.

Nomenclature

A_c = missile acceleration command
A_m = missile lateral acceleration
A_t = target lateral acceleration
H_m = missile altitude
K_α = aerodynamic turning rate constant
N = proportional navigation ratio
p = differential operator, d/dt
RR = radome refraction slope error
T_a = autopilot time constant
T_c = guidance filter time constant
T_{go} = time to go
T_s = seeker tracking loop time constant
V_m = missile speed
ε = boresight error
η = learning rate of the parameter update algorithm
λ = line-of-sight (LOS) angle
$\dot{\lambda}$ = LOS rate command
$\mu(\cdot)$ = membership function
ρ = weighting factor
σ_{ij} = adjustable variable
τ_s = sampling period

I. Introduction

THE concept of gain scheduling, originated in connection with the development of flight control systems, is an effective way to control systems whose dynamics change with the operating conditions. This technique uses process variables related to dynamics that compensate for the effect of working in different operating regions. It is normally used in the control of nonlinear plants where the relationship between the plant dynamics and operating conditions is known and for which a single linear time-invariant model is insufficient.[1–3]

Gain scheduling design involves three main issues: operating region partitioning into several approximately linear regions, local controller design in each linear region, and controller parameter interpolation between the linear regions. A gain-scheduled control system can be viewed as a feedback control system in which the feedback gains are adjusted using feedforward compensation. The main advantage of gain scheduling is that controller parameters can be adjusted very quickly in response to changes in the plant dynamics. Conventional autopilot gain-scheduled designs for tactical missiles may be found in Refs. 4–7. The H_∞ methods can also be applied to treat the autopilot gain-scheduled design problems.[8,9]

The main drawback of most conventional gain scheduling (CGS) is that the parameter change may be rather abrupt across the region boundaries, which may result in unacceptable or even unstable performance. Another problem is that accurate linear time-invariant models at various operating points may be difficult or impossible to obtain. As a solution to this problem, a fuzzy gain scheduling (FGS) method has been proposed that utilizes a fuzzy reasoning technique to determine the controller parameters.[10] With this method, human expertise in the linear control design and CGS can be represented by fuzzy rules with a fuzzy inference mechanism used to interpolate the controller parameters in the transition regions.[3,11]

Takagi–Sugeno[10] fuzzy models provide an effective representation of complex nonlinear systems in terms of fuzzy sets and fuzzy reasoning applied to a set of linear input–output submodels. Based on each model, FGS controllers are facilitated using linear matrix inequality methods.[12] An H_∞ gain schedule using fuzzy rules was also proposed in Ref. 13 to ensure stability and performance robustness. As for applications, the FGS technique has been used in vehicle guidance design[14] and aircraft flight control design.[15,16] A robust fuzzy gain scheduler has also been designed for an aircraft autopilot system.[17]

Neural network gain scheduling (NNGS) could incorporate learning ability into gain scheduling control.[3] The training example consists of the operating variables and control gains obtained at various operating points and their corresponding desired output. Compared to the FGS technique, the main advantage of NNGS is that it avoids the need to manually design a scheduling program or determine a suitable inferencing system. NNGS techniques may be found in various fields such as hydroelectric generation[18] and aircraft flight control systems.[19]

In summary, the major difficulties in applying FGS consist of 1) difficulty in specifying appropriate fuzzy rules and membership functions and 2) dependence of performance of the gain schedule on the accuracy of the missile guidance and control system model under each flight condition. For NNGS, the major difficulties are 1) appropriate network size selection and 2) difficulty in adding the designer's expertise to the existing network when necessary.

This paper proposes a methodology for controlling a class of dynamic systems using an adaptive fuzzy gain scheduling (AFGS) technique. This novel design methodology uses a Tagaki–Sugeno[10] fuzzy system to represent the fuzzy relationship between the scheduling variables and controller parameters. A particular

Received 27 August 1999; revision received 15 August 2000; accepted for publication 2 October 2000.

*Professor, Institute of Automatic Control Engineering; chunlin@fcu.edu.tw.
†Graduate Student, Institute of Electrical Engineering.

compensation strategy is used to design the AFGS guidance law, which is a variant from the conventional proportional navigation guidance (PNG) law. Traditionally, the guidance and control gains are switched along the missile trajectories according to a function or a table lookup built in the computer. Unlike conventional designs, the proposed novel technique offers the advantage of performance improvement for ill-defined flight dynamics through learning using an adaptive fuzzy inferencing technique. Rapid adaptivity to environmental changes makes this technique appropriate for the guidance and control systems in coping with aerodynamic changes during flight.

II. AFGS Control Scheme

Consider the general AFGS control scheme shown in Fig. 1. The dynamic nonlinear system to be controlled is described using the following state-space equations:

$$\dot{\boldsymbol{x}} = f[\boldsymbol{x}(t), u(t)] \tag{1a}$$

$$y = g[\boldsymbol{x}(t)] \tag{1b}$$

$$\boldsymbol{w} = h[\boldsymbol{x}(t)] \tag{1c}$$

where $\boldsymbol{x}$ is the state variable vector, u is the control command, y is the plant output, $\boldsymbol{w} = [w_1 \ \cdots \ w_m]^T$ is the scheduling variable vector that correlates well with the change in the plant dynamics. In Fig. 1, r is the tracking command, $e = r - y$ is the tracking error. We consider an adaptive gain-scheduled controller described by

$$u(p) = C(p)e(p) \tag{2}$$

where the transfer function $C(p)$ is defined by

$$C(p) = \frac{b_n p^n + b_{n-1} p^{n-1} + \cdots + b_0}{p^n + a_{n-1} p^{n-1} + \cdots + a_1 p + a_0} \tag{3}$$

The objective of this controller is to minimize a predefined performance measure for the closed-loop system based on the information provided by the scheduling variables. The controller is trained using a set of controller parameters $(a_0, a_1, \ldots, a_{n-1}, b_0, b_1, \ldots, b_n)$ tuned at various operating points determined by the variable $\boldsymbol{w}$. A fuzzy logic-based, adaptive parameter adjustment rule will be proposed in the following paragraphs to train the controller.

For the continuous-time control scheme shown in Fig. 1, the performance measure is defined by integrating the combination of the tracking error and control command over $[t_0, t]$ as follows:

$$J = \frac{1}{2}\int_{t_0}^{t} [e^2(\tau) + \rho u^2(\tau)]\, \mathrm{d}\tau \tag{4}$$

where the scalar ρ is included to weight the relative importance of the tracking error and the control effort.

To characterize the adaptive, fuzzy logic-based controller, a group of adjustable variables σ_{ij} are introduced. These variables are used as the consequent function parameters for the fuzzy rules. The gradient descent algorithm is then used to derive a generalized backpropagation training algorithm:

$$\Delta\sigma_{ij} = -\eta \nabla_{\sigma_{ij}} J, \qquad \forall i, j \tag{5}$$

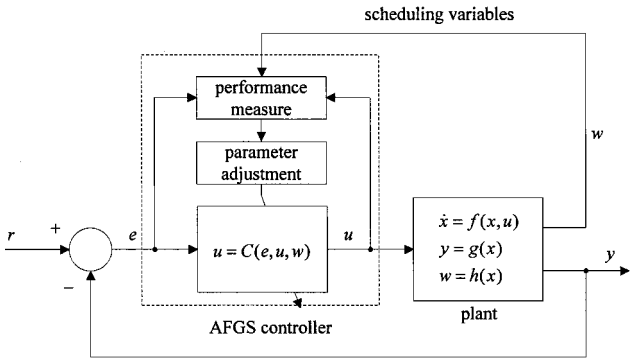


Fig. 1 AFGS control scheme.

where $\Delta\sigma_{ij} = \sigma_{ij}(t + \tau_s) - \sigma_{ij}(t)$ and the gradient $\nabla_{\sigma_{ij}} J = \partial J/\partial\sigma_{ij}$ is calculated using the chain rule:

$$\frac{\partial J}{\partial \sigma_{ij}} = \frac{\partial J}{\partial y}\frac{\partial y}{\partial \sigma_{ij}} + \frac{\partial J}{\partial u}\frac{\partial u}{\partial \sigma_{ij}} = \int_{t_0}^{t} \left[-e(\tau)\frac{\partial y}{\partial \sigma_{ij}} + \rho u(\tau)\frac{\partial u}{\partial \sigma_{ij}} \right] \mathrm{d}\tau \tag{6}$$

To complete the derivations, $\partial y/\partial\sigma_{ij}$ and $\partial u/\partial\sigma_{ij}$ must be found. For the nonlinear dynamic system (1), we have

$$\frac{\partial y}{\partial \sigma_{ij}} = \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \sigma_{ij}} \tag{7}$$

Note that if the precise plant model is not known, the gradient vector $\partial g(\boldsymbol{x})/\partial\boldsymbol{x}$ can usually be approximately calculated using the perturbation method, that is, $\partial g(\boldsymbol{x})/\partial\boldsymbol{x} \approx \Delta g(\boldsymbol{x})/\Delta\boldsymbol{x}$. From Eq. (7), the problem is transformed to obtain the derivative $\partial\boldsymbol{x}/\partial\sigma_{ij}$. From Eq. (1a), taking the derivative $\partial\boldsymbol{x}/\partial\sigma_{ij}$ with respect to t produces the gradient dynamics

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \boldsymbol{x}}{\partial \sigma_{ij}}\right) = \frac{\partial}{\partial \sigma_{ij}}\dot{\boldsymbol{x}} = \frac{\partial f(\boldsymbol{x}, u)}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \sigma_{ij}} + \frac{\partial f(\boldsymbol{x}, u)}{\partial u}\frac{\partial u}{\partial \sigma_{ij}} \tag{8}$$

The values of the Jacobian matrix $\partial f(\boldsymbol{x}, u)/\partial\boldsymbol{x}$ and the gradient vector $\partial f(\boldsymbol{x}, u)/\partial u$ can also be approximated using the perturbation method. Next, we evaluate $\partial u/\partial\sigma_{ij}$ and then proceed with solving the differential equation (7) to obtain $\partial\boldsymbol{x}/\partial\sigma_{ij}$. Since $u = u(e, \boldsymbol{w}, \sigma_{ij})$, thus

$$\frac{\partial u}{\partial \sigma_{ij}} = \frac{\partial u(\sigma_{ij})}{\partial \sigma_{ij}} + \frac{\partial u}{\partial \boldsymbol{w}}\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \sigma_{ij}} + \frac{\partial u}{\partial e}\frac{\partial e}{\partial y}\frac{\partial y}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \sigma_{ij}} = \frac{\partial u(\sigma_{ij})}{\partial \sigma_{ij}} + \left[\frac{\partial u}{\partial \boldsymbol{w}}\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} - \frac{\partial u}{\partial e}\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]\frac{\partial \boldsymbol{x}}{\partial \sigma_{ij}} \tag{9}$$

Combining Eqs. (8) and (9) yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \boldsymbol{x}}{\partial \sigma_{ij}}\right) = \left\{\frac{\partial f(\boldsymbol{x}, u)}{\partial \boldsymbol{x}} + \frac{\partial f(\boldsymbol{x}, u)}{\partial u}\left[\frac{\partial u}{\partial \boldsymbol{w}}\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} - \frac{\partial u}{\partial e}\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]\right\} \times \frac{\partial \boldsymbol{x}}{\partial \sigma_{ij}} + \frac{\partial f(\boldsymbol{x}, u)}{\partial u}\frac{\partial u(\sigma_{ij})}{\partial \sigma_{ij}} \tag{10}$$

For this differential equation, $\partial\boldsymbol{x}/\partial\sigma_{ij}$ is viewed as a state variable vector.

A Takagi–Sugeno[10] fuzzy model is used next to represent the fuzzy conditional statement between the scheduling variables and the controller parameters. This model provides an effective representation of the complex nonlinear relations in terms of fuzzy sets and fuzzy reasoning applied to a set of input–output data. In this approach, the scheduling variables $\boldsymbol{w}_i$, $i = 1, \ldots, m$, are used as the inputs, and the outputs are the parameters of the linear dynamic controller, that is, a_i, $i = 0, 1, \ldots, n-1$, and b_i, $i = 0, 1, \ldots, n$. The complex relationship between these factors is constructed through a set of fuzzy rules.

To facilitate the adaptation of the Takagi–Sugeno fuzzy model, the fuzzy model is placed into the framework of adaptive networks so that gradient vectors can be computed systematically. For a first-order Takagi–Sugeno model, the ith fuzzy rule takes the following form.

If $\boldsymbol{w}_1$ is $\overline{LW}_1^i$ and $\cdots$ and $\boldsymbol{w}_m$ is $\overline{LW}_m^i$, then

$$\begin{aligned} \boldsymbol{a}^i &= \boldsymbol{\alpha}_1^i w_1 + \boldsymbol{\alpha}_2^i w_2 + \cdots + \boldsymbol{\alpha}_m^i w_m + \boldsymbol{\alpha}_{m+1}^i \\ \boldsymbol{b}^i &= \boldsymbol{\beta}_1^i w_1 + \boldsymbol{\beta}_2^i w_2 + \cdots + \boldsymbol{\beta}_m^i w_m + \boldsymbol{\beta}_{m+1}^i \end{aligned} \tag{11}$$

where $\boldsymbol{\alpha} = [\alpha_{1j}^i \ \alpha_{2j}^i \ \cdots \ \alpha_{nj}^i]^T$, $\boldsymbol{\beta} = [\beta_{1j}^i \ \beta_{2j}^i \ \cdots \ \beta_{(n+1)j}^i]^T$, $\overline{LW}_j^i$ is the notation for the fuzzy sets defined over the domain of the scheduling variable $\boldsymbol{w}_j$, and a_j^i and b_j^i, defined next, are the linear

combinations of $w_1, \ldots, w_m$ with the coefficients α^i_{pq} and β^i_{rs}, respectively,

$$\boldsymbol{a}^i = \begin{bmatrix} a^i_0 \\ a^i_1 \\ \vdots \\ a^i_{n-1} \end{bmatrix} = \begin{bmatrix} \alpha^i_{11} & \alpha^i_{12} & \cdots & \alpha^i_{1m} & \alpha^i_{1(m+1)} \\ \alpha^i_{21} & \alpha^i_{22} & \cdots & \alpha^i_{2m} & \alpha^i_{2(m+1)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \alpha^i_{n1} & \alpha^i_{n2} & \cdots & \alpha^i_{nm} & \alpha^i_{n(m+1)} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \\ w_{m+1} \end{bmatrix}$$

$$\boldsymbol{b}^i = \begin{bmatrix} b^i_0 \\ b^i_1 \\ \vdots \\ b^i_n \end{bmatrix} = \begin{bmatrix} \beta^i_{11} & \beta^i_{12} & \cdots & \beta^i_{1m} & \beta^i_{1(m+1)} \\ \beta^i_{21} & \beta^i_{22} & \cdots & \beta^i_{2m} & \beta^i_{2(m+1)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \beta^i_{(n+1)1} & \beta^i_{(n+1)2} & \cdots & \beta^i_{(n+1)m} & \beta^i_{(n+1)(m+1)} \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \\ w_{m+1} \end{bmatrix}$$

in which $i = 1, \ldots, l$, $w_{m+1} = 1$, $a^i_j(b^i_j)$ indicates the firing strength contributed by the ith rule. In this rule, elements of the consequent parameter vectors $\boldsymbol{\alpha}^i_1, \ldots, \boldsymbol{\alpha}^i_m$ and $\boldsymbol{\beta}^i_1, \ldots, \boldsymbol{\beta}^i_m$ are viewed as the adjustable variable σ_{ij} in Eq. (5). Note that the use of a functional consequence simplifies the inferencing stage because the defuzzification process is avoided.

An equivalent representation for the controller $C(s)$ is shown in Fig. 2. In Fig. 2, $a_0, \ldots, a_{n-1}$ and $b_0, \ldots, b_n$ are the controller parameters to be adjusted such that the performance measure J is minimized. Using the Takagi–Sugeno inference mechanism, the fuzzified parameters a_j, $j = 0, 1, \ldots, n-1$, and b_j, $j = 0, 1, \ldots, n$, can be expressed by the weighted sum of all strengths of the fired rules, which takes the following forms:

$$a_j = \sum_{i=1}^{l} \bar{w}^i a^i_j, \qquad b_j = \sum_{i=1}^{l} \bar{w}^i b^i_j \tag{12}$$

where $\bar{w}^i$ denotes the normalized firing strength.

We now proceed to find the derivative $\partial u(\sigma_{ij})/\partial \sigma_{ij}$ of Eq. (9). Referring to Eq. (12) and Fig. 2, the derivative can be found using repeated applications of the chain rule. The derivations for $\sigma = \alpha, \beta$ can be generalized as

$$\frac{\partial u}{\partial \alpha^i_{pq}} = \frac{\partial u}{\partial \zeta_p}\frac{\partial \zeta_p}{\partial a_{p-1}}\frac{\partial a_{p-1}}{\partial a^i_{p-1}}\frac{\partial a^i_{p-1}}{\partial \alpha^i_{pq}} = \left(b_{p-1} + b_0 t^{p-1}\right)\frac{\partial \zeta_p}{\partial a_{p-1}}\bar{w}^i w_q$$

$$p = 1, \ldots, n, \qquad q = 1, \ldots, m+1, \qquad i = 1, \ldots, l \tag{13a}$$

$$\frac{\partial u}{\partial \beta^i_{rs}} = \frac{\partial u}{\partial b_{r-1}}\frac{\partial b_{r-1}}{\partial b^i_{r-1}}\frac{\partial b^i_{r-1}}{\partial \beta^i_{rs}} = \frac{\partial u}{\partial b_{r-1}}\bar{w}^i w_s$$

$$r = 1, \ldots, n+1, \qquad s = 1, \ldots, m+1, \qquad i = 1, \ldots, l \tag{13b}$$

Dynamic controllers are composed using proportional terms, integration terms, or derivative terms. The compensation approach just developed can be used especially for closed-loop systems with lower-order compensators such as proportional integral derivative or lead–lag networks. For these situations, the parameter update algorithm would be simpler.

The adaptive nature of AFGS renders it fundamentally different from the traditional gain scheduling schemes. The major advantage is that it avoids the need to manually design a scheduling program or to determine a suitable fuzzy inferencing system. Moreover, because the parameters are adaptable, AFGS is more flexible than the traditional gain scheduling in the sense that it is not tied to a particular system dynamics and, therefore, does not require significant modification if the plant is altered. It can, thus, be expected that the controller becomes more robust and more insensitive to plant parameter variations. In addition, unlike the traditional FGS, the nonsmoothness of the membership function will not degrade the performance significantly because we are changing the parameters of the dynamic controller, not the equivalent fuzzy gain.

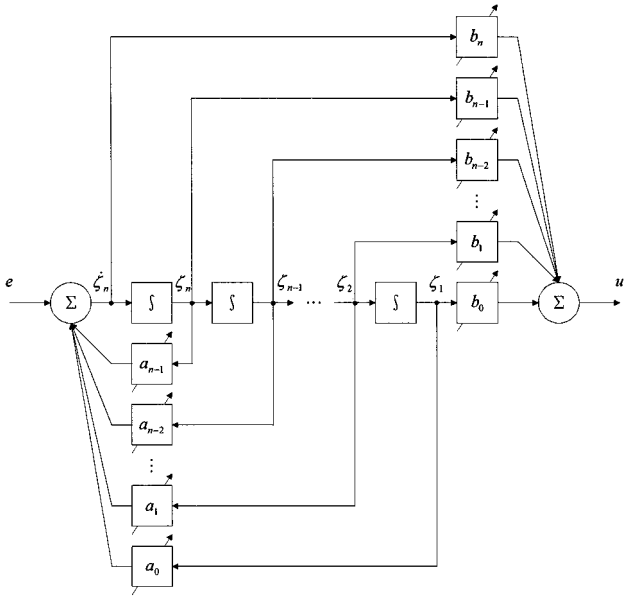


Fig. 2 Schematic of AFGS controller.

III. AFGS Guidance

FGS Guidance Law

The development of an appropriate gain-scheduling law is crucial to the performance of a scheduled system. It is quite often either difficult to find or too complicated to design CGS law. However, this problem can be resolved using fuzzy inference techniques. In FGS guidance design, the scheduling variables are taken as V_m, H_m, and T_{go}. The inference engine generates the proportional navigation ratio N and guidance filter constant T_c. They are used as a function of T_{go} to optimize the tradeoff between engagement performance and homing-loop stability.

In the following application, three term sets describing for the scheduling variables V_m, H_m, and T_{go} are, respectively, defined in the following:

$$T_{V_m} = \{A_1, A_2, A_3, A_4, A_5\}, \qquad T_{H_m} = \{B_1, B_2, B_3, B_4, B_5\}$$
$$T_{T_{\text{go}}} = \{C_1, C_2, C_3\} \tag{14}$$

where A_i, B_i, and C_i are the fuzzy sets that are characterized by certain membership functions. For the purpose of theoretical studies, the physical domain over which the scheduling variable V_m takes its crisp value is supposed to be [1.6, 2.2] Mach. The domain for H_m is [5.0, 12.5] km and the domain for $T_{T_{\text{go}}}$ is [0, 10] s.

Based on the notation introduced earlier, the AFGS rule base contains a set of FGS rules in the following form.

If V_m is A^l_i, and H_m is B^l_j, and T_{go} is C^l_k, then

$$N^l = p^l_1 V_m + q^l_1 H_m + r^l_1 T_{\text{go}} + s^l_1$$
$$T^l_c = p^l_2 V_m + q^l_2 H_m + r^l_2 T_{\text{go}} + s^l_2, \qquad l = 1, \ldots, 75 \tag{15}$$

where $(V_m, H_m, T_{\text{go}})$ is the scheduling vector that best fits the description in the premise part of the rule, and $p^l_1, q^l_1, r^l_1, s^l_1, p^l_2, q^l_2, r^l_2$, and s^l_2 are the consequent function parameters. The outputs N^l and T^l_c for this rule are the inferenced guidance parameters.

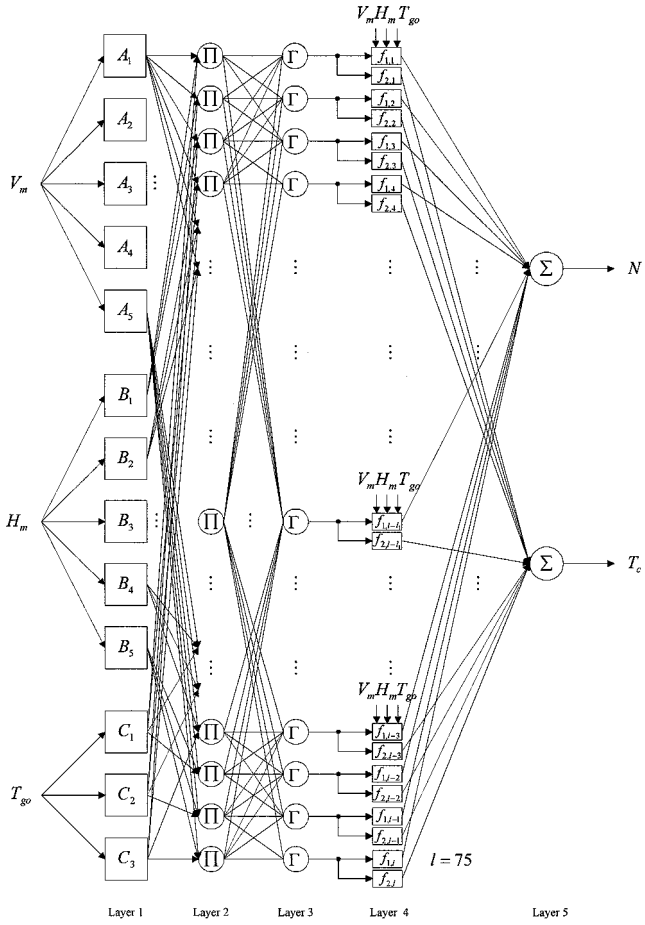


Fig. 3 Structure of FGS guidance parameters.

To facilitate the adaptation of the gain scheduling guidance model, it is convenient to put the fuzzy model into the framework of adaptive networks that can compute gradient vectors systematically. The resultant network architecture is shown in Fig. 3. We denote ${}^{j}O_i$ as the output of the ith node in layer j. The node functions of each layer are described as follows.

1) Each node in layer 1 generates the membership grades for the scheduling variables V_m, H_m, and T_{go}:

$$ {}^{1}O_i = \mu_{A_i}(V_m), \qquad {}^{1}O_j = \mu_{B_j}(H_m), \qquad {}^{1}O_k = \mu_{C_k}(T_{\text{go}}) $$
$$ i, j = 1, \ldots, 5, \qquad k = 1, \ldots, 3 \tag{16} $$

where the membership functions $\mu_{A_i}(\cdot)$, $\mu_{B_j}(\cdot)$, and $\mu_{C_k}(\cdot)$ are defined as follows:

$$ \mu_{A_i}(V_m) = 1 \Bigg/ \left\{ 1 + \left[\left(\frac{V_m - c_i}{a_i} \right)^2 \right]^{b_i} \right\} $$

$$ \mu_{B_j}(H_m) = 1 \Bigg/ \left\{ 1 + \left[\left(\frac{H_m - c_j}{a_j} \right)^2 \right]^{b_j} \right\}, \qquad i, j = 1, \ldots, 5 $$

$$ \mu_{C_k}(T_{\text{go}}) = 1 \Bigg/ \left\{ 1 + \left[\left(\frac{T_{\text{go}} - c_k}{a_k} \right)^2 \right]^{b_k} \right\}, \qquad k = 1, \ldots, 3 $$

in which the parameters a and c are, respectively, the width and center of the membership function and b is the slope of the membership function at the crossover point. A desired membership function is obtained using a proper parameter set (a, b, c). Figure 4 shows the bell-shaped membership functions considered in this research.

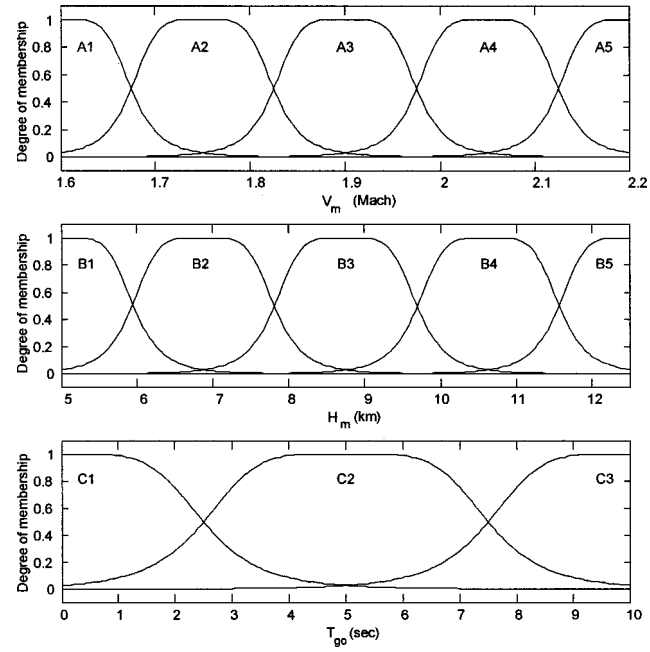


Fig. 4 Membership functions describing the scheduling variables V_m, H_m, and T_{go}.

2) Each node in layer 2 calculates the firing strength of each rule via multiplication:

$$ {}^{2}O^l = w^l = \prod_{i,j,k} \mu_{A_i}(V_m)\mu_{B_j}(H_m)\mu_{C_k}(T_{\text{go}}), \qquad l = 1, \ldots, 75 \tag{17} $$

3) The ith node in layer 3 calculates the relative firing strength of the ith rule to the sum of all rules' firing strengths:

$$ {}^{3}O^l = \Gamma(w^l) = \bar{w}^l = w^l \Big/ \sum_l w^l, \qquad l = 1, \ldots, 75 \tag{18} $$

4) The ith node in layer 4 has the following node function:

$$ {}^{4}O_1^l = \bar{w}^l f_1^l = \bar{w}^l \left(p_1^l V_m + q_1^l H_m + r_1^l T_{\text{go}} + s_1^l \right), \qquad l = 1, \ldots, 75 $$
$$ {}^{4}O_2^l = \bar{w}^l f_2^l = \bar{w}^l \left(p_2^l V_m + q_2^l H_m + r_2^l T_{\text{go}} + s_2^l \right), \qquad l = 1, \ldots, 75 \tag{19} $$

5) The nodes in layer 5 yield the guidance parameters N and T_c:

$$ N = {}^{5}O_1 = \sum_l {}^{4}O_1^l, \qquad T_c = {}^{5}O_2 = \sum_l {}^{4}O_2^l, \qquad l = 1, \ldots, 75 \tag{20} $$

We have constructed an adaptive network that is functionally equivalent to a Takagi–Sugeno fuzzy model. The consequent function parameters are regulated by the learning algorithm presented in Sec. II when the input information is provided.

Adaptive Guidance Parameter Update Law

It was known that the effective navigation ratio N, which varies due to changes in the flight conditions, is one major determinant for missile homing performance. Another major determinant is the guidance filter time constant T_c. These determinants are usually selected to compensate for the stability problem caused by radome error and for various flight conditions to generate the best engagement performance.

In general, the guidance system designer attempts to keep the effective navigation ratio as small as possible to meet the stability requirements and yet large enough so that homing will be effective. To cope with the stability problem caused by negative radome error, the guidance system time constant is generally made larger at

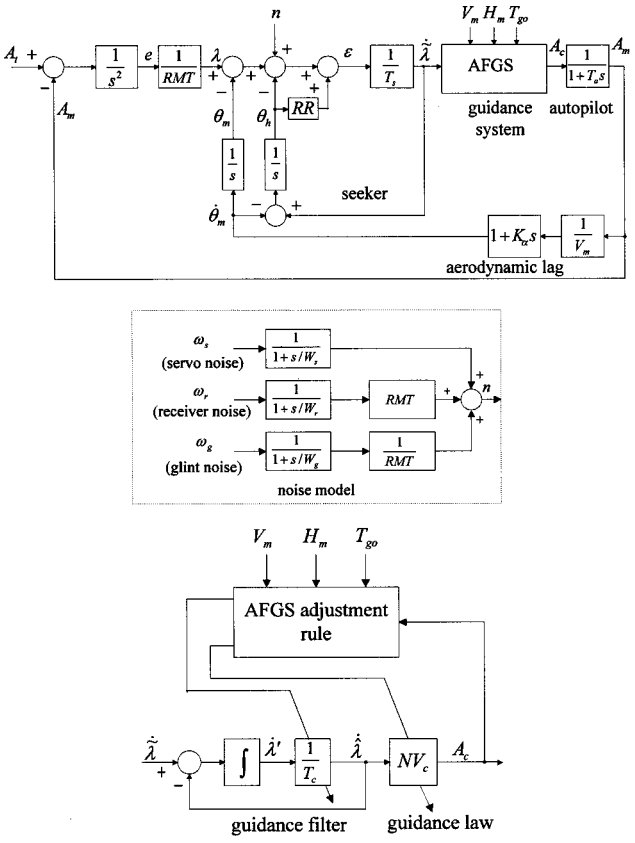


Fig. 5 Simplified homing loop with AFGS guidance law.

higher altitudes. The penalty for such a decision is that the miss distances tend to increase as the guidance system time constant increases. From another viewpoint, the guidance filtering must be heavy enough to smooth the missile's flight. It reduces the general motion due to large random accelerations responding to the noise. Yet the filtering must be light enough to allow quick missile response when needed to correct for heading errors or to chase a maneuvering target.

The preceding developed result can be extended to design an AFGS guidance law. This is a variant from the conventional PNG law described by

$$A_c = [NV_c/(1 + T_c s)]\dot{\tilde{\lambda}} \tag{21}$$

with N and T_c being functions of the scheduling variables. The learning rule for this guidance law is the backpropagation gradient descent algorithm described in Sec. II. The principal elements of a simplified missile guidance system model with an AFGS guidance law is shown in Fig. 5. In Fig. 5, the external noise n is composed of glint, receiver, and fading noises. The seeker model is completed by developing a simplified tracking and stabilization control system in which θ_m is the missile body angle and θ_h is the seeker gimbal angle. For more details about this missile system model, see Refs. 20 and 21.

To minimize the kinetic energy loss over the homing phase and, hence, to maximize the terminal speed, the guidance command policy is derived by selecting the commanded acceleration to minimize the following performance index:

$$J\left(p_1^l, q_1^l, r_1^l, p_2^l, q_2^l, r_2^l\right) = \frac{1}{2}\int_{t_0}^{t} A_c^2\left(p_1^l, q_1^l, r_1^l, p_2^l, q_2^l, r_2^l\right) \mathrm{d}\tau \tag{22}$$

The AFGS guidance command can be written as

$$A_c\left(p_1^l, q_1^l, r_1^l, p_2^l, q_2^l, r_2^l\right) = \left\{1\Big/\left[1 + T_c\left(p_2^l, q_2^l, r_2^l\right)p\right]\right\} \times N\left(p_1^l, q_1^l, r_1^l\right)V_c\dot{\tilde{\lambda}} \tag{23}$$

With reference to Eq. (20), the fuzzy gain-scheduled navigation ratio and guidance filter time constant are, respectively, given by

$$N\left(p_1^l, q_1^l, r_1^l\right) = N_{\text{bias}} + \sum_l \bar{w}^l\left(p_1^l V_m + q_1^l H_m + r_1^l T_{\text{go}}\right) \tag{24a}$$

$$T_c\left(p_2^l, q_2^l, r_2^l\right) = T_{c_{\text{bias}}} + \sum_l \bar{w}^l\left(p_2^l V_m + q_2^l H_m + r_2^l T_{\text{go}}\right) \tag{24b}$$

To simplify the adaptation, the biased terms have been adopted in Eq. (24) to replace the terms including s_1^l and s_2^l in Eq. (19). Because K_α varies with different velocities and altitudes, V_m and H_m are used as the scheduling variables in the AFGS guidance law. For notation simplicity, the arguments for N and T_c will be omitted in the sequel.

To construct the parameter update law, the autopilot is viewed as a dynamic plant to be controlled by the guidance law. The state-space representation for a first-order autopilot can be expressed as

$$\dot{x} = -(1/T_a)x + A_c, \qquad A_m = (1/T_a)x$$

Figure 5 is a schematic diagram of the AFGS guidance system. Based on these expressions we proceed to derive the parameter update equations. First, the partial derivative of J with respect to the parameters p_1^l, q_1^l, and r_1^l are, respectively, obtained as

$$\frac{\partial J}{\partial p_1^l} = \int_{t_0}^{t} A_c\frac{\partial A_c}{\partial p_1^l}\,\mathrm{d}\tau, \qquad \frac{\partial J}{\partial q_1^l} = \int_{t_0}^{t} A_c\frac{\partial A_c}{\partial q_1^l}\,\mathrm{d}\tau$$

$$\frac{\partial J}{\partial r_1^l} = \int_{t_0}^{t} A_c\frac{\partial A_c}{\partial r_1^l}\,\mathrm{d}\tau \tag{25}$$

where $\partial A_c/\partial p_1^l$ is determined via the following derivatives:

$$\frac{\partial A_c}{\partial p_1^l} = \frac{\partial A_c\left(p_1^l\right)}{\partial p_1^l} + \frac{\partial A_c}{\partial \dot{\tilde{\lambda}}}\frac{\partial \dot{\tilde{\lambda}}(\lambda, \dot{\theta}_m)}{\partial p_1^l} = \dot{\tilde{\lambda}}\bar{w}^l V_m V_c$$

$$+ \frac{\partial A_c}{\partial \dot{\tilde{\lambda}}}\left(\frac{\partial \dot{\tilde{\lambda}}}{\partial \lambda}\frac{\partial \lambda}{\partial e}\frac{\partial e}{\partial A_m} + \frac{\partial \dot{\tilde{\lambda}}}{\partial \dot{\theta}_m}\frac{\partial \dot{\theta}_m}{\partial A_m}\right)\frac{\partial A_m}{\partial x}\frac{\partial x}{\partial p_1^l}$$

$$= \dot{\tilde{\lambda}}\bar{w}^l V_m V_c + \xi(x)\frac{\partial x}{\partial p_1^l}$$

with

$$\xi(x) = \frac{1}{T_a}\frac{\partial A_c}{\partial \dot{\tilde{\lambda}}}\left(-\frac{t^2}{2\text{RMT}}\frac{\partial \dot{\tilde{\lambda}}}{\partial \lambda} + \frac{\partial \dot{\tilde{\lambda}}}{\partial \dot{\theta}_m}\frac{\partial \dot{\theta}_m}{\partial A_m}\right)$$

and $\partial x/\partial p_1^l$ being the solution to the following differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial x}{\partial p_1^l}\right) = -\frac{1}{T_a}\frac{\partial x}{\partial p_1^l} + \frac{\partial A_c}{\partial p_1^l} = \left[-\frac{1}{T_a} + \xi(x)\right]\frac{\partial x}{\partial p_1^l} + \dot{\tilde{\lambda}}\bar{w}^l V_m V_c$$

and RMT is missile–target relative range.

Similarly, $\partial A_c/\partial q_1^l$ and $\partial A_c/\partial r_1^l$ are determined from

$$\frac{\partial A_c}{\partial q_1^l} = \dot{\tilde{\lambda}}\bar{w}^l H_m V_c + \xi(x)\frac{\partial x}{\partial r_1^l}, \qquad \frac{\partial A_c}{\partial r_1^l} = \dot{\tilde{\lambda}}\bar{w}^l T_{\text{go}} V_c + \xi(x)\frac{\partial x}{\partial r_1^l}$$

with $\partial \boldsymbol{x}/\partial q_1^l$ and $\partial \boldsymbol{x}/\partial r_1^l$ being, respectively, the solutions for the following differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial x}{\partial q_1^l}\right) = \left[-\frac{1}{T_a} + \xi(x)\right]\frac{\partial x}{\partial q_1^l} + \dot{\tilde{\lambda}}\bar{w}^l H_m V_c$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial x}{\partial r_1^l}\right) = \left[-\frac{1}{T_a} + \xi(x)\right]\frac{\partial x}{\partial r_1^l} + \dot{\tilde{\lambda}}\bar{w}^l T_{\text{go}} V_c$$

Similarly, the parameter update equations for the guidance filter time constant T_c are determined from the following derivatives:

$$\frac{\partial J}{\partial p_2^l} = \int_{t_0}^{t} A_c\frac{\partial A_c}{\partial p_2^l}\,\mathrm{d}\tau, \qquad \frac{\partial J}{\partial q_2^l} = \int_{t_0}^{t} A_c\frac{\partial A_c}{\partial q_2^l}\,\mathrm{d}\tau$$

$$\frac{\partial J}{\partial r_2^l} = \int_{t_0}^{t} A_c\frac{\partial A_c}{\partial r_2^l}\,\mathrm{d}\tau \tag{26}$$

where $\partial A_c/\partial p_2^l$, $\partial A_c/\partial q_2^l$, and $\partial A_c/\partial r_2^l$ are determined via

$$\frac{\partial A_c}{\partial p_2^l} = -\frac{NV_cV_m}{T_c^2}\bar{w}^l\dot{\lambda}' + \xi(x)\frac{\partial x}{\partial p_2^l}$$

$$\frac{\partial A_c}{\partial q_2^l} = -\frac{NV_cH_m}{T_c^2}\bar{w}^l\dot{\lambda}' + \xi(x)\frac{\partial x}{\partial q_2^l}$$

$$\frac{\partial A_c}{\partial r_2^l} = -\frac{NV_cT_{\text{go}}}{T_c^2}\bar{w}^l\dot{\lambda}' + \xi(x)\frac{\partial x}{\partial r_2^l}$$

with $\partial x/\partial p_2^l$, $\partial x/\partial q_2^l$, and $\partial x/\partial r_2^l$ being the solutions to the following differential equations, respectively:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial x}{\partial p_2^l}\right) = \left[-\frac{1}{T_a} + \xi(x)\right]\frac{\partial x}{\partial p_2^l} - \frac{NV_cV_m}{T_c^2}\bar{w}^l\dot{\lambda}'$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial x}{\partial q_2^l}\right) = \left[-\frac{1}{T_a} + \xi(x)\right]\frac{\partial x}{\partial q_2^l} - \frac{NV_cH_m}{T_c^2}\bar{w}^l\dot{\lambda}'$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial x}{\partial r_2^l}\right) = \left[-\frac{1}{T_a} + \xi(x)\right]\frac{\partial x}{\partial r_2^l} - \frac{NV_cT_{\text{go}}}{T_c^2}\bar{w}^l\dot{\lambda}'$$

Constrained Parameter Update Algorithm

To avoid causing the stability problem and to not override the flight control system, the elements used to construct the guidance system should be strictly limited. For example, given K_α and randome reflection slope error (RR), the range for the achievable guidance time constant can be determined on the basis of useful stability criteria.[20] Suppose here those reliable values of N and T_c are restricted in the following constraint sets:

$$N_{\min} \le N(\sigma_{ij}) \le N_{\max}, \qquad T_{c_{\min}} \le T_c(\sigma_{ij}) \le T_{c_{\max}}$$

The two inequalities can be equivalently expressed as

$$|N(\sigma_{ij}) - N_{\text{mid}}| \le (N_{\max} - N_{\min})/2$$

$$\left|T_c(\sigma_{ij}) - T_{c_{\text{mid}}}\right| \le \left(T_{c_{\max}} - T_{c_{\min}}\right)/2 \tag{27}$$

where $N_{\text{mid}} = 0.5(N_{\max} + N_{\min})$ and $T_{c_{\text{mid}}} = 0.5(T_{c_{\max}} + T_{c_{\min}})$. From Eq. (27), the consequent parameters p_1^l, q_1^l, r_1^l, p_2^l, q_2^l, and r_2^l for the AFGS guidance law should be restricted to the following convex sets:

$$S_N = \left\{\sigma_{ij} \;\middle|\; g_N(\sigma_{ij}) = [N(\sigma_{ij}) - N_{\text{mid}}]^2 - [(N_{\max} - N_{\min})/2]^2 \le 0\right\} \tag{28a}$$

$$S_{T_c} = \left\{\sigma_{ij} \;\middle|\; g_{T_c}(\sigma_{ij}) = \left[T_c(\sigma_{ij}) - T_{c_{\text{mid}}}\right]^2 - \left[\left(T_{c_{\max}} - T_{c_{\min}}\right)/2\right]^2 \le 0\right\} \tag{28b}$$

A simple saturation of just the guidance parameters N and T_c alone could provide an absolute guarantee that they would stay below the given boundary. However, because the output of AFGS is saturated, it will no longer reflect the changes in the consequent function parameters. Therefore, the adjustment laws will keep adjusting the parameters with no apparent improvement on the performance cost minimization. To manage the constraints while minimizing the cost function, a confinement algorithm based on the gradient projection algorithm[22] is derived in the following.

First, as the current parameter σ_{ij} is in the interior of $S_N(S_{T_c})$, the unconstrained algorithm described by Eq. (5) is used. If the current parameter is on the boundary of $S_N(S_{T_c})$, that is, $g_N(\sigma_{ij}) = 0\,[g_{T_c}(\sigma_{ij}) = 0]$ and the direction of search given by the unconstrained algorithm is still pointing inside $S_N(S_{T_c})$, that is, $-\nabla J_N^T(\sigma_{ij})\nabla g_N(\sigma_{ij}) \le 0\,[-\nabla J_{T_c}^T(\sigma_{ij})\nabla g_{T_c}(\sigma_{ij}) \le 0]$, then we keep the algorithm. If the searching direction is pointing away from $S_N(S_{T_c})$ then the gradient projection algorithm is used. The parameter update process for constrained optimization is summarized in the following.

1) If $\sigma_{ij} \in S_N$ or $g_N(\sigma_{ij}) = 0$ and $-\nabla J_N^T(\sigma_{ij})\nabla g_N(\sigma_{ij}) \le 0$, then the following gradient descent algorithm is applied:

$$\begin{bmatrix} p_1^l(t+\tau_s) \\ q_1^l(t+\tau_s) \\ r_1^l(t+\tau_s) \end{bmatrix} = \begin{bmatrix} p_1^l(t) \\ q_1^l(t) \\ r_1^l(t) \end{bmatrix} - \eta_N \begin{bmatrix} \dfrac{\partial J}{\partial p_1^l(t)} \\ \dfrac{\partial J}{\partial q_1^l(t)} \\ \dfrac{\partial J}{\partial r_1^l(t)} \end{bmatrix} \tag{29}$$

where η_N is the learning rate; $\partial J/\partial p_1^l$, $\partial J/\partial q_1^l$, and $\partial J/\partial r_1^l$ are determined from Eq. (25).

2) If condition 1 is not satisfied, then the following the gradient projection algorithm is used:

$$\begin{bmatrix} p_1^l(t+\tau_s) \\ q_1^l(t+\tau_s) \\ r_1^l(t+\tau_s) \end{bmatrix} = \begin{bmatrix} p_1^l(t) \\ q_1^l(t) \\ r_1^l(t) \end{bmatrix} - \eta_N \begin{bmatrix} \dfrac{\partial J}{\partial p_1^l(t)} \\ \dfrac{\partial J}{\partial q_1^l(t)} \\ \dfrac{\partial J}{\partial r_1^l(t)} \end{bmatrix} + \varphi_N \frac{\nabla g_N \nabla g_N^T}{\nabla g_N^T \nabla g_N} \begin{bmatrix} \dfrac{\partial J}{\partial p_1^l(t)} \\ \dfrac{\partial J}{\partial q_1^l(t)} \\ \dfrac{\partial J}{\partial r_1^l(t)} \end{bmatrix} \tag{30}$$

where the gradient of $g_N(\sigma_{ij})$ with respect to σ_{ij} can be obtained as follows:

$$\nabla g_N = 2(N - N_{\text{mid}})\bar{w}_l[V_m \quad H_m \quad T_{\text{go}}]^T$$

The parameter φ_N is used as a scaling factor for modifying the speed of constraint satisfaction.

Similarly, for the guidance filter time constant T_c, the parameter update law is given as follows.

1) If $\sigma_{ij} \in S_{T_c}$ or $g_{T_c}(\sigma_{ij}) = 0$ and $-\nabla J_{T_c}^T(\sigma_{ij})\nabla g_{T_c}(\sigma_{ij}) \le 0$, then

$$\begin{bmatrix} p_2^l(t+\tau_s) \\ q_2^l(t+\tau_s) \\ r_2^l(t+\tau_s) \end{bmatrix} = \begin{bmatrix} p_2^l(t) \\ q_2^l(t) \\ r_2^l(t) \end{bmatrix} - \eta_{T_c} \begin{bmatrix} \dfrac{\partial J}{\partial p_2^l(t)} \\ \dfrac{\partial J}{\partial q_2^l(t)} \\ \dfrac{\partial J}{\partial r_2^l(t)} \end{bmatrix} \tag{31}$$

where η_{T_c} is the learning rate; $\partial J/\partial p_2^l$, $\partial J/\partial q_2^l$, and $\partial J/\partial r_2^l$ are determined from Eq. (26).

2) If condition 1 is not satisfied then

$$\begin{bmatrix} p_2^l(t+\tau_s) \\ q_2^l(t+\tau_s) \\ r_2^l(t+\tau_s) \end{bmatrix} = \begin{bmatrix} p_2^l(t) \\ q_2^l(t) \\ r_2^l(t) \end{bmatrix} - \eta_{T_c} \begin{bmatrix} \dfrac{\partial J}{\partial p_2^l(t)} \\ \dfrac{\partial J}{\partial q_2^l(t)} \\ \dfrac{\partial J}{\partial r_2^l(t)} \end{bmatrix} + \varphi_{T_c} \frac{\nabla g_{T_c} \nabla g_{T_c}^T}{\nabla g_{T_c}^T \nabla g_{T_c}} \begin{bmatrix} \dfrac{\partial J}{\partial p_2^l(t)} \\ \dfrac{\partial J}{\partial q_2^l(t)} \\ \dfrac{\partial J}{\partial r_2^l(t)} \end{bmatrix} \tag{32}$$

where the gradient ∇g_{T_c} is defined by

$$\nabla g_{T_c} = 2\left(T_c - T_{c_{\text{mid}}}\right)\bar{w}_l[V_m \quad H_m \quad T_{\text{go}}]^T$$

In summary, the proposed AFGS guidance law is given by Eq. (23) with the adaptive guidance parameters N and T_c determined from Eq. (24) in which the consequent parameters are updated via Eqs. (29–32).

IV. Results and Analysis

The nominal parameters used for homing-loop simulations are listed in Table 1. In Table 2, 12 considered flight conditions are given. The values for K_α corresponding to 12 flight conditions were also tabulated. In general, K_α increases as the missile altitude increases, and it decreases as the missile speed decreases. To maintain consistent performance, the navigation ratio and guidance time constant must be varied under various flight conditions. In the simulation studies we first let $N_{\min} = 3$ and $N_{\max} = 6$, and $T_{c_{\min}} = 0.1$ and $T_{c_{\max}} = 0.2$ s.

The performance measure considered in the simulation study is the minimization of the control effort defined in Eq. (22). To achieve better training results, two phases for the guidance parameter adaptation learning were conducted.

Offline Training

The purpose for the offline training is to provide a better initial setting for the consequent parameters p_1^l, q_1^l, r_1^l, p_2^l, q_2^l, and r_2^l. During this phase, the external noise source n is temporally removed from the homing-loop simulation scheme and a 3 g ($g = 9.8$ m/s^2) step input is used to simulate the target lateral acceleration command A_t.

The training parameters were initialized at zero values. While the training progresses, the learning rates were chosen to speed up convergence and to avoid overshooting the training results. The chosen learning rates were suitable for the homing-loop simulation under all flight conditions. In the simulation studies $\eta_N = 1 \times 10^{-8}$, $\varphi_N = 3 \times 10^{-7}$, $\eta_{T_c} = 1 \times 10^{-11}$, and $\varphi_{T_c} = 3 \times 10^{-10}$ were chosen. A batch of parameter learning processes includes 12 homing-loop simulations with each case corresponding to a flight condition. Figure 6 shows profiles for N and T_c with missile speeds of 1.6 and 2.2 Mach and altitudes of 7.5–12.5 km.

Table 1 Nominal homing-loop parameters

Parameter	Value	Parameter	Value
t_0	0 s	RMT_{x0}	10,000 m
RR	−0.02 rad/rad	RMT_{y0}	0 m
T_s	0.1 s	N_{bias}	4.5
W_x	100 rad/s	$N_{\min}$	3
W_g	10 rad/s	$N_{\max}$	6
W_r	100 rad/s	$T_{c_{\text{bias}}}$	0.15 s
v_t	300 m/s	$T_{c_{\min}}$	0.1 s
A_t	3 g	$T_{c_{\max}}$	0.2 s

Table 2 Aerodynamic turning rate constant for 12 flight conditions

Flight condition				
Number	H_m, km	V_m, Mach	K_α, s	$T_c^{(,)}$ and $N^{(,)}$
1	7.5	1.6	1.0	(1, 1)
2	10.0	1.6	1.4	(2, 1)
3	12.5	1.6	1.8	(3, 1)
4	7.5	1.8	0.9	(1, 2)
5	10.0	1.8	1.3	(2, 2)
6	12.5	1.8	1.7	(3, 2)
7	7.5	2.0	0.8	(1, 3)
8	10.0	2.0	1.2	(2, 3)
9	12.5	2.0	1.6	(3, 3)
10	7.5	2.2	0.7	(1, 4)
11	10.0	2.2	1.1	(2, 4)
12	12.5	2.2	1.5	(3, 4)

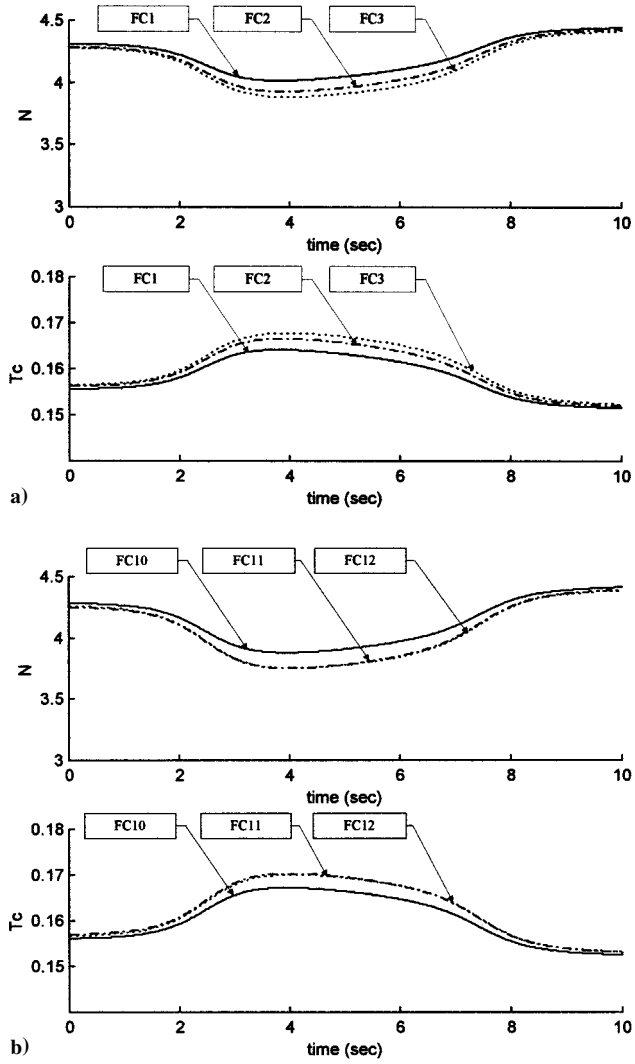


Fig. 6 Transients of the effective navigation ratio and guidance filter time constant during offline training for a) flight conditions 1–3 and b) flight conditions 10–12.

Online Training

While online training progresses, the external noise n is included in the homing-loop simulation scheme. A band-limited Gaussian noise model is used to emulate the Poisson jinking target maneuver.[21] The model generates a specific lateral acceleration profile with the 3-g variance that corresponds to an evasive trajectory. This is used to simulate the operational environment while the missile engages an evasive target. The learning rates used in this phase remain the same as those in offline training.

Analysis for Online Operation

After the consequent function parameters of the guidance law have been set up, they are then prepared for online operation. The parameters are adaptively tuned to minimize the control effort and, hence, to avoid the stability problem. Figure 7 shows transients for N and T_c for online homing-loop simulations with missile speeds of 1.6 and 2.2 Mach and altitudes of 7.5–12.5 km. It can be observed that the navigation ratio tends to decrease and the guidance filter time constant tends to increase as the altitude increases. This tendency coincides with the conventional guidance system design in which the guidance parameters were appropriately designed for individual flight conditions to meet the engagement performance and stability requirement caused by radome errors. The missile altitude is fixed next with changes in the speed. Figure 8 shows the results for homing-loop simulations with missile altitudes fixed at 7.5 and 12.5 km and speed varied from 1.6 to 2.2 Mach. From Figs. 6–8, we also see that the equivalent navigation ratio over the engagement process lies within 3.5 and 4.5 and the equivalent guidance time

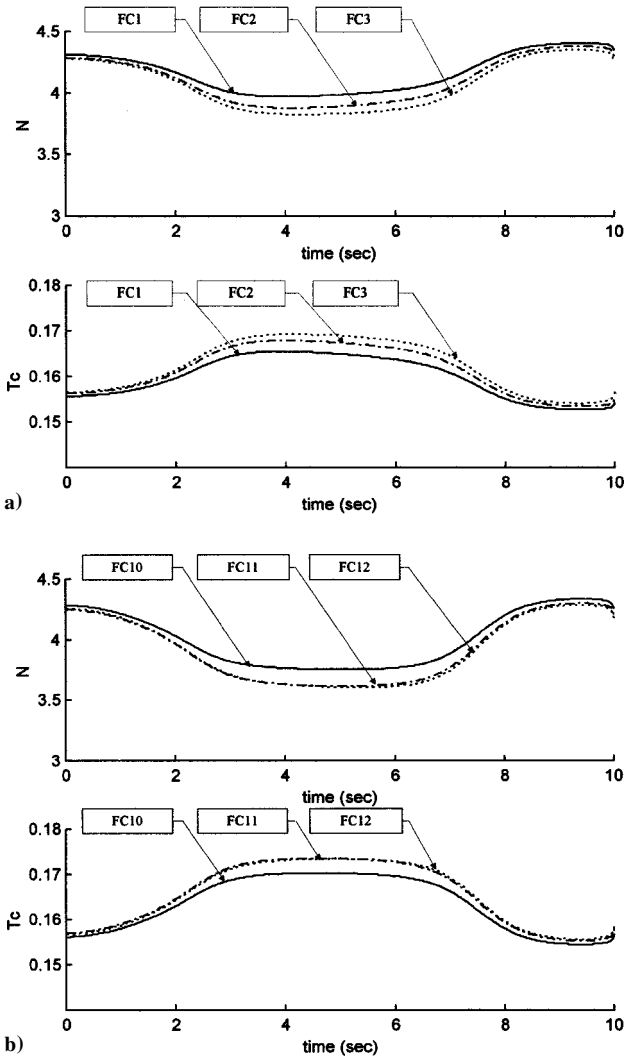


Fig. 7 Transients of the effective navigation ratio and guidance filter time constant during online operation for a) flight conditions 1–3 and b) flight conditions 10–12.

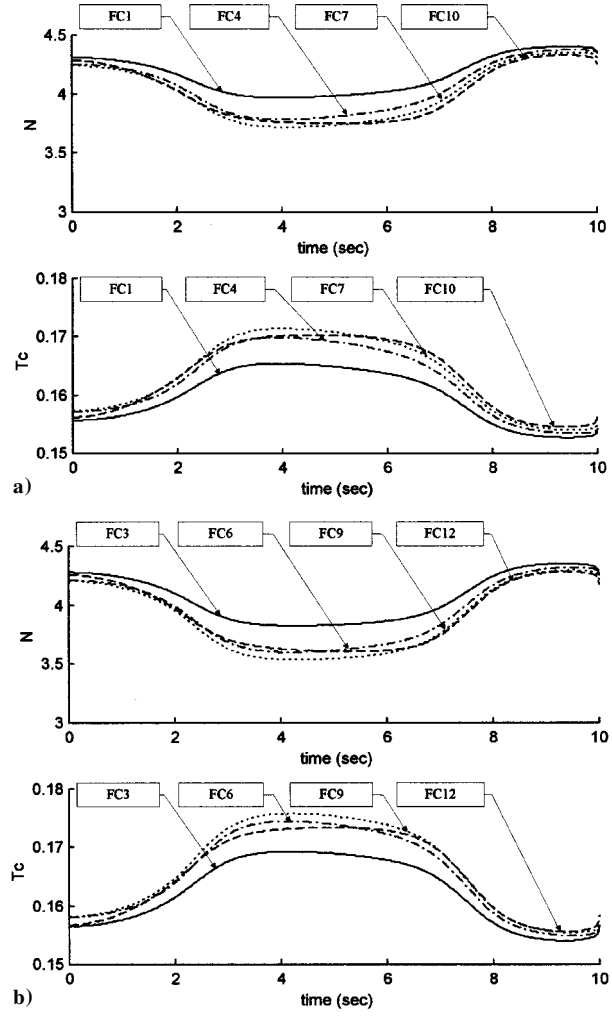


Fig. 8 Transients of the effective navigation ratio and guidance filter time constant for a) flight conditions 1, 4, 7, and 10 and b) flight conditions 3, 6, 9, and 12.

constant lies within 0.15 and 0.18 s. Clearly, theses values meet the constraint described by Eq. (26).

Figure 9 shows the flight trajectories over all flight conditions, where RMT_x and RMT_y are missile–target relative range on the horizontal and vertical axes, respectively. The initial relative range on the horizontal axis is denoted by −10 and 0 km on the vertical axis. It is seen that the missile can effectively engage the target under 12 flight conditions. These results demonstrate tracking performance as the system is operated over a significant portion of the operating regions. That is, the guidance parameter adaptations have effectively captured the effect of plant variations.

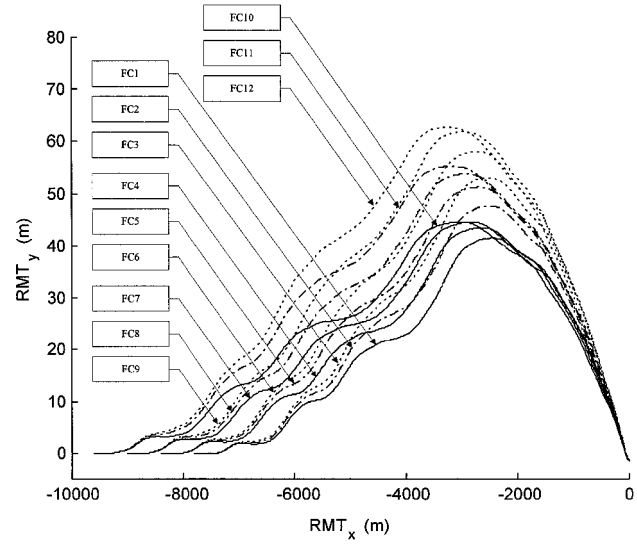


Fig. 9 Flight trajectories corresponding to all flight conditions.

Performance Evaluation

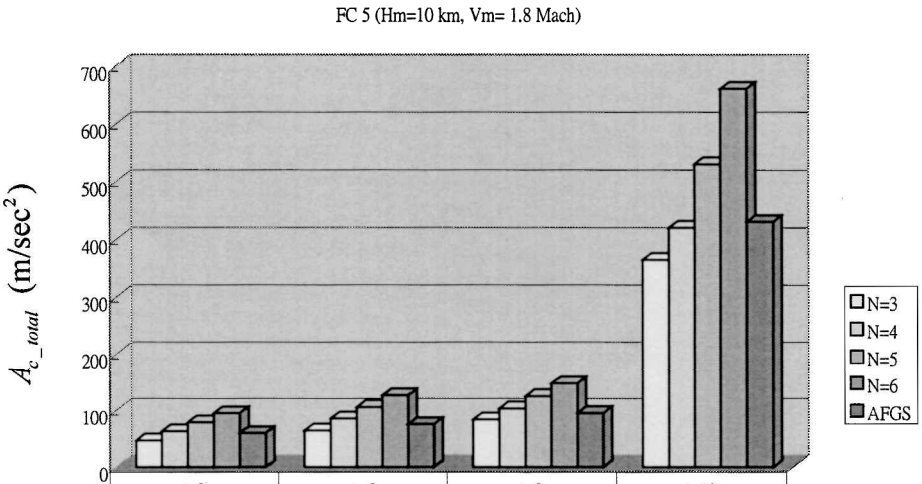
To evaluate the control energy consumption, we define a cost measure given by

$$A_{c_{\text{total}}} = \sqrt{\int_{t_0}^{t_1} A_c^2(t)\,\mathrm{d}t}$$

To verify the effectiveness of our proposed approach, the accumulated control effort was calculated for four periods: $t = 0$–3, 0–6, 0–9, and 0–10 s with altitude 10 km and speed 1.8 Mach (Fig. 10). It can be found that control effort of the AFGS guidance law is, in general, less than the conventional PNG design with $N = 4, 5, 6$. This implies that the AFGS guidance system possesses higher kinetic energy to engage an evasive target.

With the missile engaging a 3-g random target maneuver, 12 homing-loop simulations were performed. Table 3 summarizes the resultant miss distances with the missile guided by the conventional PNG [i.e., Eq. (21)] and the proposed AFGS guidance.

From the extensive simulation studies, we may conclude that the AFGS guidance law is distinctive from the conventional approach because of wider design freedom and higher flexibility in the controller architecture. A finely tuned AFGS guidance law uses less control effort during flight and, therefore, possesses the potential to increase the terminal speed. As we have expected, performance of



	t=3	t=6	t=9	t=10
N=3	46.6704	63.8786	82.9825	364.6468
N=4	62.3818	84.6485	102.2796	420.6291
N=5	78.1674	105.1484	123.8978	530.7863
N=6	94.0188	125.5181	146.5939	661.3985
AFGS	59.5032	75.3008	94.3997	430.5578

Fig. 10 Comparison for energy consumption within four periods (H_m = 10 km, V_m = 1.8 Mach).

Table 3 Comparison of miss distances, m

Flight condition			PNG[a]				
Number	H_m, km	V_m, Mach	$N=3$	$N=4$	$N=5$	$N=6$	AFGS
1	7.5	1.6	1.7	1.37	1.39	1.42	1.34
2	10.0	1.6	1.75	1.38	1.39	1.4	1.34
3	12.5	1.6	1.78	1.39	1.38	1.39	1.36
4	7.5	1.8	1.69	1.37	1.4	1.44	1.34
5	10.0	1.8	1.75	1.38	1.4	1.43	1.35
6	12.5	1.8	1.78	1.39	1.39	1.42	1.36
7	7.5	2.0	1.62	1.35	1.42	1.5	1.33
8	10.0	2.0	1.69	1.37	1.41	1.47	1.34
9	12.5	2.0	1.78	1.38	1.4	1.44	1.36
10	7.5	2.2	1.62	1.36	1.44	1.53	1.32
11	10.0	2.2	1.69	1.37	1.43	1.5	1.25
12	12.5	2.2	1.74	1.38	1.42	1.48	1.34

[a] $T_c = 0.15$ s.

the guidance system under individual flight conditions also agrees with the family of all flight conditions.

V. Conclusions

This study developed a novel AFGS control design method for a class of nonlinear dynamic systems. The systematic design methodology uses a Takagi–Sugeno fuzzy system to represent the fuzzy relationship between the scheduling variables and controller parameters. Based on this control design scheme, an AFGS guidance law was developed. This novel technique offers the advantage of performance improvement for ill-defined flight dynamics through learning using the adaptive fuzzy inferencing mechanism. Compared to the conventional PNG design, the presented guidance law possesses a higher degree of automation and results in better terminal guidance performance, that is, less control effort and a smaller miss distance. Results from extensive homing-loop simulations verify the effectiveness of the proposed design.

Acknowledgment

This research was sponsored by National Science Council, Taiwan, Republic of China, under the Grant NSC 88-2213-E-035-031.

References

[1]Rugh, W. J., "Analytical Framework for Gain Scheduling," *IEEE Control Systems Magazine*, Vol. 11, No. 1, 1991, pp. 79–84.

[2]Hualin, T., and Rugh, W. J., "Overtaking Optimal Control and Gain Scheduling," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, 1997, pp. 1273–1277.

[3]Tan, S., Hang, C. C., and Chai, J. S., "Gain Scheduling: From Conventional to Neuro-Fuzzy," *Automatica*, Vol. 33, No. 3, 1997, pp. 411–419.

[4]Balas, G. J., and Packard, A. K., "Design of Robust Time-Varying Controllers for Missile Autopilot," *Proceedings of the First IEEE Conference on Control Applications*, Inst. of Electrical and Electronics Engineers, New York, 1992, pp. 104–110.

[5]Shamma, J. S., and Cloutier, J. R., "Trajectory Scheduled Missile Autopilot Design," *Proceedings of the First IEEE Conference on Control Applications*, Inst. of Electrical and Electronics Engineers, New York, 1992, pp. 237–242.

[6]White, D. P., Wozniak, J. G., and Lawrence, D. A., "Missile Autopilot Design Using a Gain Scheduling Technique," *Proceedings of the 26th Southeastern Symposium on System Theory*, IEEE Publications, Piscataway, NJ, 1994, pp. 606–610.

[7]Piou, J. E., and Sobel, K. M., "Application of Gain Scheduling Eigenstructure Assignment to Flight Control Design," *Proceedings of 1996 IEEE International Conference on Control Applications*, Inst. of Electrical and Electronics Engineers, New York, 1996, pp. 101–106.

[8]Nichols, R. A., Reichert, R. T., and Rugh, W. J., "Gain Scheduling for H_∞ Controllers: A Flight Control Example," *IEEE Transactions on Control Systems Technology*, Vol. 1, No. 2, 1993, pp. 69–79.

[9]Schumacher, C., and Khargonekar, P. P., "Missile Autopilot Designs Using H_∞ Control with Gain Scheduling and Dynamic Inversion," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 234–243.

[10]Takagi, T., and Sugeno, M., "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 15, No. 1, 1985, pp. 116–132.

[11]Ling, C., and Edgar, T. F., "A New Fuzzy Gain Scheduling Algorithm for Process Control," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, 1992, pp. 2284–2290.

[12]Driankov, D., Hellendoorn, H., and Reinfrank, M., *An Introduction to Fuzzy Control*, Springer, Berlin, 1993, pp. 186–195.

[13]Yang, C. D., Kuo, T. M., and Tai, H. C., "H_∞ Gain Scheduling Using Fuzzy Rules," *Proceedings of the 35th Conference on Decision and Control*, IEEE Publications, Piscataway, NJ, 1996, pp. 3794–3799.

[14]Hessburg, T., "Fuzzy Logic Control for Lateral Vehicle Guidance," *Proceedings of the Second IEEE Conference on Control Applications*, Inst. of Electrical and Electronics Engineers, New York, 1993, pp. 581–586.

[15]Gonsalves, P. G., and Zacharias, G. L., "Fuzzy Logic Gain Scheduling for Flight Control," *Proceedings of the Third IEEE Conference on Fuzzy Systems*, Inst. of Electrical and Electronics Engineers, New York, 1994, pp. 952–957.

[16]Adams, R. J., Sparks, A. G., and Banda, S. S., "A Gain Scheduled Multivariable Design for a Manual Flight Control System," *Proceedings of the First IEEE Conference on Control Applications*, Inst. of Electrical and Electronics Engineers, New York, 1992, pp. 584–589.

[17]Tanaka, T., and Aizawa, Y., "A Robust Gain Scheduler Interpolated into Multiple Models by Membership Functions," AIAA Paper 92-4553, Aug. 1992.

[18]Liang, R. H., and Hsu, Y. Y., "Scheduling of Hydroelectric Generations Using Artificial Neural Networks," *IEE Proceedings: Generation, Transmission, and Distribution*, Vol. 141, No. 5, 1994, pp. 452–458.

[19]Jonckheere, E. A., Yu, G. R., and Chen, C. C., "Gain Scheduling for Lateral Motion of Propulsion Controlled Aircraft Using Neural Networks," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, 1997, pp. 3321–3325.

[20]Nesline, W., and Zarchan, P., "Radome Induced Miss Distance in Aerodynamically Controlled Homing Missiles," *Proceedings of 1984 AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 1984, pp. 99–115.

[21]Lin, C. F., *Modern Navigation, Guidance, and Control Processing*, Prentice–Hall, Englewood Cliffs, NJ, 1991, pp. 13–72.

[22]Luenberger, D. G., *Optimization by Vector Space Methods*, Wiley, New York, 1969, pp. 297–299.