JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 26, No. 5, September–October 2003

Improving Conventional Longitudinal Missile Autopilot Using Cerebellar Model Articulation Controller Neural Networks

C. C. Lin* and F. C. Chen†
National Chiao-Tung University, Hsinchu, Taiwan, Republic of China

A way to integrate the cerebellar model articulation controller (CMAC) neural network with the missile longitudinal conventional feedback controller (CFC) to compensate for nonlinearities, unmodeled dynamics, parameter variations, etc., is proposed. The inner loop of the CFC will essentially be left unchanged to improve the stability of the missile. The outer loop of CFC, in addition to playing its traditional role, would work with the CMAC to learn quickly to approximate the dynamic inversion from angle of attack to control deflection, to achieve better tracking in normal acceleration. In this arrangement, the well-known CFC acts as a safety net, whereas additional performance is brought about through CMAC learning.

Nomenclature

A_Z	=	normal acceleration, m/s^2
A_{ZC}	=	normal acceleration command, m/s^2
C_N, C_M, $C_{N\delta}$, $C_{M\delta}$	=	aerodynamic coefficients
d	=	reference diameter, cm
I_y	=	pitch moment of inertia, kg · m^2
M_m	=	Mach number
Q	=	dynamic pressure, kg/(m.sec^2)
q	=	pitch rate, deg/s
S	=	reference area, m^2
U	=	$U_C + U_O$
U_C	=	output of cerebellar model articulation controller neural network, deg
U_I	=	command generated by the inner loop of conventional feedback controller, deg
U_O	=	command generated by the outer loop of conventional feedback controller, deg
V_m	=	missile speed, m/s
W	=	missile weight, kg
α	=	angle of attack, deg
δ	=	tail-fin deflection, deg
θ	=	pitch angle, deg

I. Introduction

CLASSICAL design methods, coupled with techniques such as gain scheduling, form some well-tested and widely applied methodologies for control problems in aircraft and missiles.[1] The control designs, performed offline at a limited number of linear time-invariant models representing different flight conditions in the flight envelope, require extensive gain-scheduling computations to ensure a smooth transition between different flight conditions in the flight envelope. Also these approaches are not suitable for highly nonlinear problems or those with inconsistencies between the actual flight dynamics and its mathematical model.

Because advances in control theory, we have seen many new methods proposed for designing longitudinal and lateral control laws for flight control systems. First, during the 1970s, adaptive control schemes were introduced to a few experimental aircraft flight control systems with slow-time-varying dynamics between different regions in the flight envelope.[2,3] However, these algorithms are still not capable of handling dynamic nonlinearities. More recently, some methods were proposed to handle model uncertainties. In particular, H_∞-based control strategies have shown robustness to modeling inaccuracy, but their capability can be limited when facing large nonlinearities in fast missile maneuvering.[4–7]

A more recent approach that has shown excellent potential for the control of highly maneuverable airframes is the robust dynamic inversion (DI) method.[8,9] The control design philosophy is to use an inner-loop dynamic inversion controller and an outer-loop robust controller. The DI controller linearizes the longitudinal dynamics of the airframe. Because model uncertainties prevent exact linearization, there will always be errors associated with this controller. This problem is addressed by synthesizing a robust, outer-loop controller around the dynamically inverted inner loop.

A totally different approach is the use of neural network (NN) technology. A clear description of the basic principles of NN theory and a quite complete classification of different neural architectures and learning algorithms are provided in Ref. 10. The NN has attracted significant attention because of its capability to learn to approximate nonlinear mappings; therefore, the nonlinear dynamics and uncertainties of a system can be learned or identified directly by NNs. Recently, the implementation of neural architectures in flight control laws, both as controllers and as estimators, has been proposed in several investigations.[11–13] Sadhukhan and Feteih presented a neural controller based on the DI approach.[11] The controller is found to provide stable and good performance in the presence of modeling uncertainty of the linearized longitudinal dynamics of an F8 aircraft. Gili and Battipede employ an adaptive neurocontroller for a nonlinear six-degree-of-freedom combat aircraft,[12] taking the advantage of NN online learning ability in dealing with any changes of aircraft dynamics during flight. Calise et al. present an adaptive NN-based control element to overcome highly uncertain nonlinear systems that does not rely on state estimation.[13] Among the available NN paradigms, we will specifically propose to use the cerebellar model articulation controller (CMAC) in this paper.[14,15] The local learning property of the CMAC, together with its parallel-processing power when implemented as very large-scale integrated (VLSI), make the CMAC suitable for real-time, quick maneuvering applications.[16]

Despite the recent developments in autopilot design using DI or using NN as already reported, conventional feedback controllers (together with the gain-scheduling techniques) still dominate in aerospace industries. The purpose of this paper is to propose a way to bring together the maturity of the conventional feedback controller (CFC), the fast learning of the CMAC NN, and the nonlinear control idea of DI into a unique framework for longitudinal missile autopilot design. Under this control scheme, in addition to

Received 13 June 2002; revision received 24 March 2003; accepted for publication 25 April 2003.

*Ph.D. Student, Department of Electrical and Control Engineering; lin.turbo@msa.hinet.net.tw.

†Professor, Department of Electrical and Control Engineering; fcchen@cc.nctu.edu.tw.

playing its traditional role, the CFC would help the CMAC to learn quickly to compensate for nonlinearities, parameter variations, etc., so that a DI from angle of attack (AOA) to control deflection can be achieved, resulting in better tracking in normal acceleration. The advantage of this proposed approach is that one can exploit existing well-tested traditional control designs, while having an additional mechanism for compensating the deficiencies in traditional control designs. Moreover, because the main role of the CFC in the new scheme is in maintaining basic stability, instead of achieving performance, the design cost of CFC can be reduced considerably.

This paper is organized as follows. In Sec. II, we present some general information about CMAC NNs. In Sec. III, the missile dynamics and the traditional control design methods are reviewed. In Sec. IV, the ID from AOA to control deflection involving the inner control loops are derived first. The CMAC is then introduced to work with the outer control loop to achieve better tracking in normal acceleration. Simulations and discussions are included in Sec. V. Section VI concludes the paper.

II. CMAC Control Systems

The CMAC NN was originally conceived by Albus.[14,15] It is well known that many biological sensory-motor control structures are organized using neurons that process locally tuned overlapping receptive fields. The CMAC has been quite successful in representing complex nonlinear functions using this same organizing principle and is a potential tool for real-time control. Because only a small subset of weights are active at each point in the input space, individual weights account for a significant function of the "local" output error. Network parameters that determine the effectiveness of this purely local approach include generalization and quantization, where generalization is the size of the receptive field and quantization indicates how finely an analog input is quantized. The main benefits of using local approximation techniques to represent a complex nonlinear functions are faster learning, compared to the global approaches such as the backpropagation multilayer NN and the ability to train the network in one part of the input space without corrupting what has already been learned in more distant regions. If we compare the CMAC to other local NN such as the radial basis net, the CMAC is simpler and faster. More important, it can be very easily implemented as digital ASICs for parallel processing.

In the following, we use the simple CMAC control system shown in Fig. 1 as an example to give the basic idea of how the CMAC NN is applied in this paper. The system to be controlled is

$$Y(k+1) = G[\boldsymbol{Y}(\boldsymbol{k}), \boldsymbol{U}(\boldsymbol{k})] \tag{1}$$

where G is a continuous nonlinear function, $\boldsymbol{Y}(\boldsymbol{k})$ is the system output, and $\boldsymbol{U}(\boldsymbol{k})$ is the control input. We assume the system is invertible such that

$$U(k) = G^{-1}[Y(k+1), Y(k)] \tag{2}$$

Initially the CMAC table is empty. In each time step k, the CMAC involves a recall and a learning process. In the recall process, $Y_d(k+1)$ and $Y(k)$ are used as an address to generate the control signal $U_c(k)$ from the CMAC, where $Y_d(k+1)$ is the desired system output for the next time step. The control signal $U(k)$ is the sum of this CMAC output $U_c(k)$ and the constant-gain controller output $U_p(k)$. Suppose the actual system output is $Y(k+1)$ when the control $U(k)$ is applied to the system. Then, in the learning process, system input $U(k)$ is treated as the desired CMAC output to modify the CMAC content stored at location $[Y(k+1), Y(k)]$. Notice that what the CMAC learns is the correct inverse mapping of the system. With generalization, each input vector to CMAC for recall or learning will map to a receptive field, which involves a number of memory locations. When two input vectors are closer, their receptive fields will have more overlap.

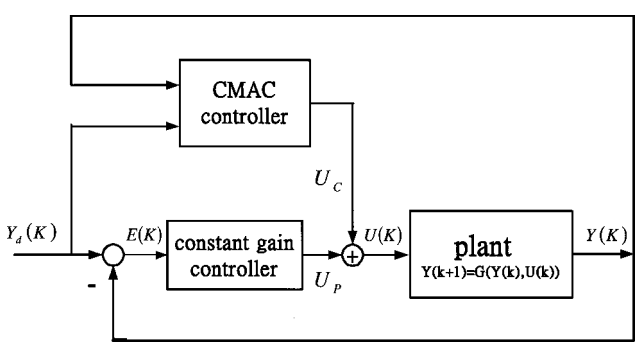

Fig. 1 CMAC control system.

In mathematical terms, during the recall, the control $U(k)$ is generated by

$$U(k) = U_c(k) + U_p(k) \tag{3}$$

where $U_c(k) \equiv CMAC[Y_d(k+1), Y(k)]$ is the output of CMAC module for the input vector $[Y_d(k+1), Y(k)]$.

The CMAC is updated by the following gradient-type learning rule in the learning process:

$$W_i(k+1) = W_i(k) + (\mu/g)\{U(k) - CMAC[Y(k+1), Y(k)]\} \tag{4}$$

where g is the size of generalization, W_i the content of the ith memory location, and μ is the learning rate. (Note that there are g locations to be updated.)

III. Missile Dynamics and Conventional Control Designs

For a missile traveling at Mach 3 at an altitude of 5 km, its longitudinal dynamics can be approximated by

$$\dot{\alpha} = q + (A_Z/V_m)\cos(\alpha) \tag{5}$$

$$\dot{q} = (QSd/I_y)[C_M(\alpha, M_m) + C_{M\delta}(M_m)\cdot\delta] \tag{6}$$

$$A_Z = -(QS/W)[C_N(\alpha, M_m) + C_{N\delta}(M_m)\cdot\delta] \tag{7}$$

where Q is 338,760 kg/(m·s), S is 0.125 m^2, W is 500 kg, d is 40 cm, M_m is 3, V_m is 1000 m/s, I_y is 6500 kg·m^2, and C_N, C_M, $C_{N\delta}$, and $C_{M\delta}$ are aerodynamic coefficients shown in the Appendix. Notice that the missile aerodynamics are highly nonlinear. The aerodynamic models used in Eqs. (5–7), which are identified using experimental data, involve a lot of uncertainty.

Although CFC are widely used, the missile control design problem remains unsolved from a theoretical point of view. A typical three-loop pitch CFC shown in Fig. 2 will be introduced next, which will work with the CMAC in the following sections.[1] The three loops consist of one outer accelerometer loop and two inner loops, the rate loop and the synthetic stability loop. The quadratic model of actuator dynamics considered here is

$$G_A(S) = 1\Big/\left[1 + 2(\xi_A/\omega_A)S + S^2\Big/\omega_A^2\right] \tag{8}$$

where $\xi_A = 0.7$, $\omega_A = 200$ rad/s, and the transfer functions of the rate gyro and accelerometer are neglected. The aerodynamic transfer functions q/δ and A_z/δ are obtained from the linearized longitudinal dynamics.[1]

The linearization of the longitudinal dynamics is carried out at several AOA under trim conditions. A linearized longitudinal model

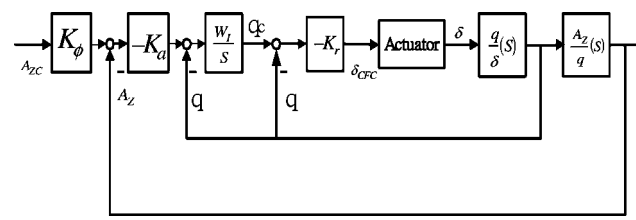

Fig. 2 Configuration of CFC pitch autopilot.

Table 1 Stability derivatives of missile and gains K_a, W_I, and K_r of four flight conditions

Derivatives and gains	$\alpha_0 = 2$ deg	$\alpha_0 = 7$ deg	$\alpha_0 = 12$ deg	$\alpha_0 = 17$ deg
Z_α	−0.8051	−0.9513	−1.6721	−1.9547
Z_δ	−0.2295	−0.2295	−0.2294	−0.2294
M_α	−19.275	7.8000	5.277	−16.566
M_δ	−37.52	−37.224	−37.52	−37.52
K_a	0.401	0.246	0.189	0.223
W_I	3.50	5.11	4.93	4.50
K_r	0.50	0.54	0.54	0.51

is represented as

$$\dot{\alpha} = Z_\alpha \alpha + q + Z_\delta \delta = q + (A_Z/V_m) \tag{9}$$

$$\dot{q} = M_\alpha \alpha + M_\delta \delta \tag{10}$$

$$A_Z = V_m(Z_\alpha \alpha + Z_\delta \delta) \tag{11}$$

where M_α and M_δ are pitch moment derivatives and Z_α and Z_δ z-axis force derivatives. Table 1 gives the stability derivatives at four different AOA. Observe that the linear model undergoes severe variations with respect to flight condition. Especially, notice the variations of M_α, which cause the pitch dynamics to become unstable at flight conditions 2 and 3. Accordingly, aerodynamic transfer functions q/δ and A_z/δ in Fig. 2 are defined for each of these four flight conditions.

In designing the CFC shown in Fig. 2, the method proposed in Ref. 17 can be employed to determine a combination of the parameters K_a, W_I, and K_r, such that the required gain margin (GM) and phase margin (PM) can be satisfied. For the requirement that the GM and PM in the four flight conditions shown in table 1 are GM = 10 dB and PM = 60 deg, the combinations of the gains K_a, W_I, and K_r for the four flight conditions are listed in Table 1. Another parameter K_ϕ shown in Fig. 1 is defined as

$$K_\phi = 1 + (1/K_a V_m) \tag{12}$$

where K_ϕ is a gain to overcome the existence of the steady-state error. Notice that the control gains between two flight conditions are interpolated. In the next section, we will propose a way for the CMAC to work with the CFC to improve control results.

IV. Improving the CFC by Using the CMAC

In this application, the control plant is the missile aerodynamics plus the inner loop of the CFC. The control U applied to the control plant is the sum of the CFC outer loop and the CMAC output. This section is divided into four parts. First, the inverse dynamics of the control plant from AOA α to control U will be derived, so that the architecture of the CMAC can be determined.

The control U is expected to achieve good tracking in AOA α. However, the guidance law is expressed in terms of normal acceleration A_Z. Therefore, in the second part of this section, some mechanism will be introduced to translate the A_Z command into a smooth α command, to enhance CMAC learning and to achieve better tracking in A_Z. Then, in the third and fourth parts, the CMAC online control law and training law are introduced, respectively.

DI of Missile

DI is a systematic method for designing control laws for nonlinear systems and has been proposed for flight control.[8,9] The basic idea is to solve for the control input required from inverse dynamic equations to give a specific output. It relies on knowledge of the nonlinear inverse dynamic equations rather than linearizations about various operating points. Although the normal acceleration A_Z is the variable to be controlled, the dynamics from tail fin δ to A_Z are nonminimum phase. Therefore, in this paper we seek to invert the dynamics from δ to α.

To proceed, we need to transform the conventional missile control system shown in Fig. 2 into an alternative form. In Fig. 2,

$$\delta_{\mathrm{CFC}} = -K_r\{(W_I/S)[-K_a(K_\phi A_{ZC} - A_Z) - q] - q\} \tag{13}$$

From Eq. (9), one has

$$q = \dot{\alpha} - (1/V_m)A_Z \tag{14}$$

Then, Eq. (13) becomes

$$\begin{aligned}\delta_{\mathrm{CFC}} &= -K_r((W_I/S)\{-K_a(K_\phi A_{ZC} - A_Z) - [\dot{\alpha} - (1/V_m)A_Z]\} \\ &\quad - [\dot{\alpha} - (1/V_m)A_Z]) = -K_r((W_I/S)\{-K_a K_\phi(A_{ZC} - A_Z) \\ &\quad + K_a(1 - K_\phi)A_Z - [\dot{\alpha} - (A_Z/V_m)]\} - [\dot{\alpha} - (A_Z/V_m)])\end{aligned} \tag{15}$$

When Eq. (12), then Eq. (15) becomes

$$\begin{aligned}\delta_{\mathrm{CFC}} &= -K_r\{W_I[-K_a K_\phi(A_{ZC} - A_Z) - \dot{\alpha}]/S - [\dot{\alpha} - (1/V_m)A_Z]\} \\ &= -K_r W_I[-K_a K_\phi(A_{ZC} - A_Z)/S + (1/W_I V_m)A_Z] \\ &\quad + K_r(\dot{\alpha} + W_I\alpha) = U_O + U_I\end{aligned} \tag{16}$$

According to the CFC described in Eq. (16), the missile control system shown in Fig. 2 can be rearranged as in Fig. 3. We denote $U_O = -K_r W_I\lfloor -K_a K_\phi(A_{ZC} - A_Z)/S + A_Z/(W_I V_m)\rfloor$ as the outer loop of CFC and $U_I = K_r(\dot{\alpha} + W_I\alpha)$ as the combined control effects of the two inner loops of CFC.

The CMAC missile control system proposed in this paper is shown in Fig. 4. The two CFC inner loops are left unchanged, while the CFC outer loop output U_O adds to the CMAC output U_C to generate the new control U. Denote the new tail-fin command as δ_C. Then $\delta_C = \delta_{\mathrm{CFC}} + U_C = U_I + U_O + U_C = U_I + U$. It is clear from Fig. 4 that the missile dynamics plus the two CFC inner loops become the CMAC control plant and that U is the control input. Notice that when the CMAC weights are zero, then $U_C = 0$ and the control δ_C is equivalent to the δ_{CFC} shown in Fig. 3. Therefore, initially the proposed CMAC controller is just the CFC. Then the CMAC goes through the online learning and control process. The CMAC is designed to learn to work with the outer loop of CFC such that U becomes a dynamic inverting control of the plant. Before this is

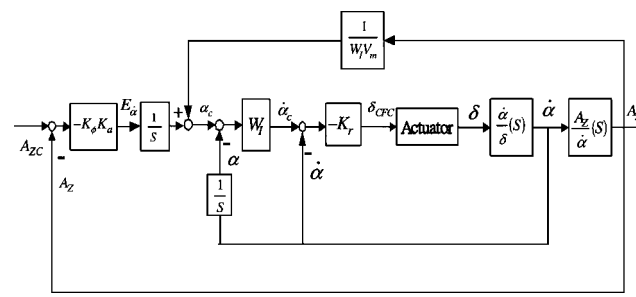

Fig. 3 AOA reconfiguration of the CFC pitch autopilot.

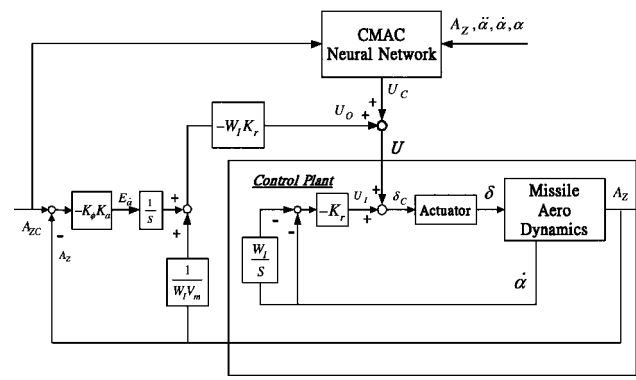

Fig. 4 CMAC NN-based autopilot.

explained, because CMAC learns the nonlinear dynamic mapping of AOA, we need to derive its inverse dynamics.

From Eqs. (5) and (7), we have

$$\dot{\alpha} = q + (QS/V_m W)[C_N(\alpha) + C_{N\delta} \cdot \delta] \cos\alpha \tag{17}$$

Assume the dynamics of the actuator are ignored such that

$$\delta = \delta_C = U + K_r(\dot{\alpha} + W_I \alpha) \tag{18}$$

Differentiating Eq. (17) yields an expression for $\ddot{\alpha}$,

$$\ddot{\alpha} = \frac{\partial \dot{\alpha}}{\partial q}\dot{q} + \frac{\partial \dot{\alpha}}{\partial \alpha}\dot{\alpha} + \frac{\partial \dot{\alpha}}{\partial \delta}\dot{\delta} \tag{19}$$

where

$$\frac{\partial \dot{\alpha}}{\partial q}\dot{q} = \frac{QSd}{I_y}[C_M(\alpha) + C_{M\delta} \cdot \delta] \tag{20}$$

$$\frac{\partial \dot{\alpha}}{\partial \alpha}\dot{\alpha} = \frac{QS}{V_m W}[C_{N\alpha}(\alpha)\cos\alpha - C_N(\alpha)\sin\alpha - C_{N\delta} \cdot \delta \cdot \sin\alpha]\dot{\alpha} \tag{21}$$

$$\frac{\partial \dot{\alpha}}{\partial \delta}\dot{\delta} = \frac{QS}{V_m W}[C_{N\delta} \cdot \cos\alpha]\dot{\delta} \tag{22}$$

$$\dot{\delta} = \dot{U} + K_r(\ddot{\alpha} + W_I \dot{\alpha}) \tag{23}$$

Because the effect of δ on $\dot{\alpha}$, that is, $\partial\dot{\alpha}/\partial\delta$, is small, the third term in the right-hand side of Eq. (19) is neglected.[9] Thus, the second-order dynamic equation for missile AOA (19) is expressed as

$$\begin{aligned}\ddot{\alpha} = {} & (QSd/I_y)[C_M(\alpha) + C_{M\delta} \cdot (U + K_r\dot{\alpha} + K_r W_I \alpha)] \\ & + (QS/V_m W)[C_{N\alpha}(\alpha)\cos\alpha - C_N(\alpha)\sin\alpha \\ & - C_{N\delta} \cdot (U + K_r\dot{\alpha} + K_r W_I \alpha)\sin\alpha]\dot{\alpha}\end{aligned} \tag{24}$$

where U is the control input. After some manipulations of Eq. (24), the control U can be expressed explicitly as

$$\begin{aligned}U = {} & [(QSd/I_y)C_{M\delta} - (QS/V_m W)C_{N\delta} \cdot \sin\alpha \cdot \dot{\alpha}]^{-1} \\ & \times\{\ddot{\alpha} - (QSd/I_y)[C_M(\alpha) + C_{M\delta} \cdot (K_r\dot{\alpha} + K_r W_I \alpha)] \\ & - (QS/V_m W)[C_{N\alpha}(\alpha)\cos\alpha - C_N(\alpha)\sin\alpha \\ & - C_{N\delta} \cdot (K_r\dot{\alpha} + K_r W_I \alpha)\sin\alpha]\dot{\alpha}\}\end{aligned} \tag{25}$$

Equation (25) represents the inverse dynamics from α to the control U, which shows that U is a function of $\ddot{\alpha}$, $\dot{\alpha}$, and α. The discrete-time form of Eq. (25) is

$$U(k) = f[\alpha(k+1), \alpha(k), \alpha(k-1)] \tag{26}$$

This can be justified by applying many existing discretizing rules to Eq. (25), for example, Euler's rule, trapezoid rule, etc. If f in Eq. (26) is known, the inverse control can be implemented by Eq. (26) in the sense that if $\alpha_{\text{des}}(k+1)$ is the desired AOA at time $k+1$, then $\alpha(k+1)$ would be $\alpha_{\text{des}}(k+1)$ if the control applied at time k is

$$U(k) = f[\alpha_{\text{des}}(k+1), \alpha(k), \alpha(k-1)] \tag{27}$$

Because f is unknown or inaccurate, later in this section we will explain how the CMAC is designed and trained to approximate the mapping of Eq. (27).

Path Planning

Typically, a missile guidance law generates a normal acceleration command profile, A_{ZC}, and the purpose of control is for normal acceleration A_Z to track A_{ZC}. The A_Z is closely related to AOA α in the sense that no significant A_Z would appear before a meaningful α is built up. Because the command of the inverting control equation (27) is given in terms of α, we need to transform the normal acceleration command A_{ZC} into the AOA command α_{des}. This transformation would significantly affect the learning of CMAC and the control result of the CMAC controller, and so it is treated here separately as a path-planning problem.

As far as the CMAC learning is concerned, it is desired that the AOA command α_{des} is not too far from the current α, so that the CMAC can quickly generate adequate control from generalization of its learning. On the other hand, the nonminimum phase property of the missile system would dictate an extremely large AOA when precise tracking in normal acceleration is desired. Under these considerations, we will propose to use a low-pass filter to set up a soft limitation to the AOA command α_{des}. The purpose of this low-pass filter is to restrict the bandwidth of the control signal not to exceed that of the control plant, to enhance CMAC learning. We will also use AOA rate-limiting and acceleration-limiting mechanisms to provide a hard limit on the AOA command α_{des}.

The signal $E_{\dot{\alpha}}$ in Fig. 4, which equals $K_a K_\phi(A_{ZC} - A_Z)$, can be considered as the tracking error of the AOA rate. Therefore, a desired rate of AOA is defined as

$$\dot{\alpha}_d = \dot{\alpha} - K_a K_\phi(A_{ZC} - A_Z) \tag{28}$$

Let $\dot{\alpha}_d$ go through a low-pass filter and denote the output as $\dot{\alpha}_{dd}$

$$\dot{\alpha}_{dd} = [1/(1 + aS)]\dot{\alpha}_d \tag{29}$$

The AOA rate limiting is

$$\dot{\alpha}_{\max} = \text{bias} + P_{\text{gain}}|(A_{ZC} - A_z)| \tag{30}$$

Subject to the rate constraint (30), the $\dot{\alpha}_{dd}$ is modified to be $\dot{\alpha}_{dr}$ according to

$$\dot{\alpha}_{dr} = \begin{cases} \dot{\alpha}_{dd}; & \text{if} \quad |\dot{\alpha}_{dd}| < \dot{\alpha}_{\max} \\ \text{sgn}(\dot{\alpha}_{dd})\dot{\alpha}_{\max}; & \text{if} \quad |\dot{\alpha}_{dd}| \geq \dot{\alpha}_{\max} \end{cases} \tag{31}$$

Define $\ddot{\alpha}_{\max}$ as the acceleration limiting of AOA. Let us consider desired acceleration of AOA in discrete time as $\ddot{\alpha}_{\text{des}}(k) = (\dot{\alpha}_{dr}(k) - \dot{\alpha}(k))/\Delta T$, where ΔT is the sampling time interval. Then, the desired rate of AOA $\dot{\alpha}_{\text{des}}(k)$ subject to acceleration constraints can be expressed as

$$\dot{\alpha}_{\text{des}}(k) = \begin{cases} \dot{\alpha}_{dr}(k); & \text{if} \quad |\ddot{\alpha}_{\text{des}}(k)| < \ddot{\alpha}_{\max} \\ \dot{\alpha}(k-1) + \Delta T\ \text{sgn}[\ddot{\alpha}_{\text{des}}(k)]\ddot{\alpha}_{\max}; & \text{if} \quad |\ddot{\alpha}_{\text{des}}(k)| \geq \ddot{\alpha}_{\max} \end{cases} \tag{32}$$

Now the desired AOA at time step $k+1$ can be defined as

$$\alpha_{\text{des}}(k+1) = \alpha(k) + \dot{\alpha}_{\text{des}}(k)\Delta T \tag{33}$$

The $\alpha_{\text{des}}(k+1)$ was introduced in Eq. (27) and will be used in the rest of this section.

CMAC Control

Here we explain how the CMAC control U_C is generated. By using Eq. (29), the expression of U_O in Eq. (16) can be rearranged as follows:

$$\begin{aligned}U_O &= -K_r W_I[-K_a K_\phi(A_{ZC} - A_Z)/S + A_Z/(W_I V_m)] \\ &= -K_r W_I(\dot{\alpha}_d - \dot{\alpha})/S - K_r A_Z/V_m \\ &= -K_r W_I \cdot a \cdot \dot{\alpha}_{dd} - K_r W_I(\dot{\alpha}_{dd} - \dot{\alpha})/S - K_r A_Z/V_m \\ &= -K_r W_I \cdot a \cdot \dot{\alpha}_{dd} + U_G\end{aligned} \tag{34}$$

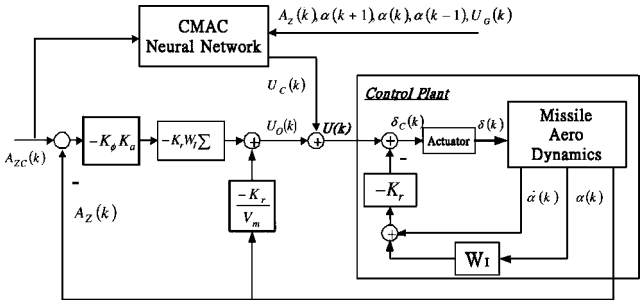

Fig. 5 Discrete-time form of CMAC NN-based autopilot.

where $U_G = -K_r W_I(\dot{\alpha}_{dd} - \dot{\alpha})/S - K_r A_Z/V_m$ includes the integration part. Since $U = U_O + U_C$, one has

$$U = -K_r W_I \cdot a \cdot \dot{\alpha}_{dd} + U_G + U_C \tag{35}$$

To express Eq. (35) in its discrete-time form, defining $A = a(\dot{\alpha}_{dd}/\dot{\alpha}_{\text{des}})$ and making use of Eq. (33), one obtains

$$U(k) = -K_r W_I \cdot A \cdot [\alpha_{\text{des}}(k+1) - \alpha(k)]/\Delta T + U_G(k) + U_C(k) \tag{36}$$

Figure 5 is the discrete-time form of Fig. 4. The $U(k)$ defined in Eq. (36) is the control applied to the plant in Fig. 5. Recall that Eq. (27) is the expression of the inverse dynamics of the control plant, which is the missile plus two CFC inner loops. The CMAC is expected to generate the control $U_C(k)$ such that the right-hand side of Eq. (36) equals the right-hand side of Eq. (27). If this is achieved, then

$$U_C(k) = f[\alpha_{\text{des}}(k+1), \alpha(k), \alpha(k-1)] + K_r W_I \cdot A \cdot [\alpha_{\text{des}}(k+1) - \alpha(k)]/\Delta T - U_G(k) \tag{37}$$

Equation (37) indicates that, when generating control, the input vector to the CMAC is $[\alpha_{\text{des}}(k+1), \alpha(k), \alpha(k-1), U_G(k)]$. At time step k, denote the actual CMAC output as

$$U_C(k) = \text{CMAC}[\alpha_{\text{des}}(k+1), \alpha(k), \alpha(k-1), U_G(k)] \tag{38}$$

Then, as shown in Fig. 5, the control $U(k) = U_O(k) + U_C(k)$ is applied to the control plant, resulting in the plant output $\alpha(k+1)$ at time-step $k+1$. If the CMAC output (38) does not produce the correct mapping as in Eq. (37), then $\alpha(k+1) \neq \alpha_{\text{des}}(k+1)$, and the CMAC goes through the following learning, to reduce the tracking error.

CMAC Learning

The purpose of training the CMAC NN is to approximate the inverse dynamics described by Eq. (25) or Eq. (26), to generate effective inverse control online. Suppose at time step k, the reference command is $\alpha_{\text{des}}(k+1)$, and $\alpha_{\text{des}}(k+1)$ is used in Eqs. (36) and (38) to generate the control $U(k)$. Furthermore, suppose this control results in the output $\alpha(k+1)$. Notice that $U(k)$ and the $\alpha(k+1)$ satisfy Eq. (26) and are used to train the CMAC as follows. The contents of g memory locations addressed by the CMAC input vector $[\alpha(k+1), \alpha(k), \alpha(k-1), U_G(k)]$ are updated according to

$$\begin{aligned} W_i(k+1) &= W_i(k) + (\mu/g)(U(k) - \{-K_r W_I A \cdot [\alpha(k+1) \\ &\quad - \alpha(k)]/\Delta T\} - U_G(k) - \text{CMAC}[\alpha(k+1), \alpha(k), \alpha(k \\ &\quad - 1), U_G(k)]) = W_i(k) + (\mu/g)\{-K_r W_I A \cdot [\alpha_{\text{des}}(k+1) \\ &\quad - \alpha(k+1)]/\Delta T + U_c(k) - \text{CMAC}[\alpha(k \\ &\quad + 1), \alpha(k), \alpha(k-1), U_G(k)]\}, \qquad i = 1, 2, \ldots, g \end{aligned} \tag{39}$$

In other words, the CMAC learns the difference between $U(k)$ and the control that is generated with $\alpha_{\text{des}}(k+1)$ replaced by $\alpha(k+1)$. Thus, CMAC learns $U_C(k)$ plus a correcting term

$$-K_r W_I A \cdot [\alpha_{\text{des}}(k+1) - \alpha(k+1)]/\Delta T \tag{40}$$

After the CMAC is trained at the location $[\alpha(k+1), \alpha(k), \alpha(k-1), U_G(k)]$ by Eq. (39), it is used to generate $U_C(k+1)$ at location $[\alpha_{\text{des}}(k+2), \alpha(k+1), \alpha(k), U_G(k+1)]$. With path-planning Eq. (33), $\alpha_{\text{des}}(k+2)$ will differ only slightly from $\alpha(k+1)$. Therefore, the generalization from the CMAC input vector for recall $[\alpha_{\text{des}}(k+2), \alpha(k+1), \alpha(k), U_G(k+1)]$ would have much overlap with the generalization from the CMAC input vector for learning $[\alpha(k+1), \alpha(k), \alpha(k-1), U_G(k)]$ because the two vectors are close in space. In other words, CMAC learning would immediately help the subsequent control via CMAC generalization. This is the reason the CMAC is a suitable tool for fast missile maneuvering control.

Effectiveness of CMAC Learning and Convergence of Tracking Error

The simulation results in Sec. V show that the proposed CMAC controller can improve the transient behavior and reduce settling time. However, because of highly nonlinear characteristics, both in missile dynamics and in the CMAC net, and because of the control approach adopted in this paper, it can be difficult to obtain an analytical stability result. Nonetheless, we discuss the effectiveness of the CMAC learning and the convergence of the tracking error.

The purpose of the learning rule suggested in Eq. (39) is to train the CMAC net to approximate the inverse dynamics of the "control plant" shown in Fig. 5.

The learning in Eq. (39) is equivalent to a gradient rule, which reduces the cost function

$$\begin{aligned} J &= \tfrac{1}{2}[\alpha_{\text{des}}(k+1) - \alpha(k+1)]^2 = \tfrac{1}{2}(\{[\alpha_{\text{des}}(k+1) - \alpha(k)]/\Delta T \\ &\quad - [\alpha(k) - \alpha(k-1)]/\Delta T\}/\Delta T - \{[\alpha(k+1) - \alpha(k)]/\Delta T \\ &\quad - [\alpha(k) - \alpha(k-1)]/\Delta T\}/\Delta T)^2 \Delta T^4 \\ &\cong (\Delta T^4/2)[\ddot{\alpha}_{\text{des}}(k) - \ddot{\alpha}(k)]^2 \end{aligned} \tag{41}$$

under either of the following two assumptions: 1) when the system is not far away from a target trim flight or 2) when the CMAC net has acquired the inverse mapping to certain extent. In either case, the difference between $\text{CMAC}[\alpha(k+1), \alpha(k), \alpha(k-1), U_G(k)]$ and $U_C(k) = \text{CMAC}[\alpha_{\text{des}}(k+1), \alpha(k), \alpha(k-1), U_G(k)]$ in Eq. (39) is small, and the correcting term (40) becomes dominant.

From Eqs. (24), (41) and because $U = U_O + U_C$, we have

$$\begin{aligned} -\frac{\partial J}{\partial U_C} &= \Delta T^4[\ddot{\alpha}_{\text{des}}(k) - \ddot{\alpha}(k)]\frac{\partial \ddot{\alpha}}{\partial U}\frac{\partial U}{\partial U_C} = \Delta T^4(\{[\alpha_{\text{des}}(k+1) \\ &\quad - \alpha(k)]/\Delta T - [\alpha(k) - \alpha(k-1)]/\Delta T\}/\Delta T - \{[\alpha(k+1) \\ &\quad - \alpha(k)]/\Delta T - [\alpha(k) - \alpha(k-1)]/\Delta T\}/\Delta T)\frac{\partial \ddot{\alpha}}{\partial U}\frac{\partial U}{\partial U_C} \end{aligned} \tag{42}$$

Because the change of U_o is independent of the change of U_C, we may rewrite Eq. (42) as

$$-\frac{\partial J}{\partial U_C} = \frac{\partial \ddot{\alpha}}{\partial U}[\alpha_{\text{des}}(k+1) - \alpha(k+1)]\Delta T^2 \tag{43}$$

Differentiating Eq. (24) with respect to U, one obtains

$$\frac{\partial \ddot{\alpha}}{\partial U} = \frac{QSd}{I_y}C_{M\delta} - \frac{QS}{V_m W}C_{N\delta} \cdot \sin\alpha \cdot \dot{\alpha} \equiv B \cdot (-K_r W_I \cdot A/\Delta T^3) \tag{44}$$

where

$$B = -\Delta T^3[(QSd/I_y)C_{M\delta} - (QS/V_m W)C_{N\delta} \cdot \sin\alpha \cdot \dot{\alpha}]/(K_r W_I \cdot A) \tag{45}$$

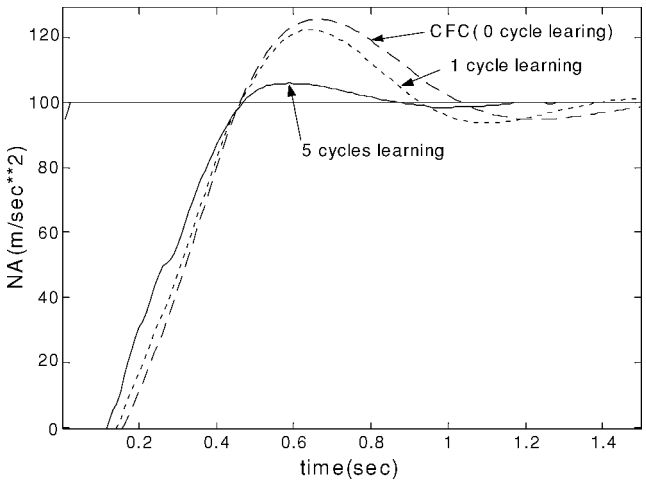

Fig. 6 Output of NA after 0, 1, and 5 cycles learning.

Notice that B is positive if

$$|\dot{\alpha}| < \left| \frac{dV_m W}{I_y \sin\alpha} \cdot \frac{C_{M\delta}}{C_{N\delta}} \right|$$

Since

$$\left| \frac{dV_m W}{I_y \sin\alpha} \cdot \frac{C_{M\delta}}{C_{N\delta}} \right| > \left| \frac{dV_m W}{I_y} \right| = 30.76 \text{ rad/s}$$

(an impossibly large rate of AOA), then we can conclude that $B > 0$. Based on Eqs. (41), (43), and (44), $-(\partial J/\partial U_C)$ is related to the correcting term in Eq. (40) as

$$-\frac{\partial J}{\partial U_C} = B \cdot (-K_r W_I A)[\alpha_{\text{des}}(k+1) - \alpha(k+1)]/\Delta T \qquad (46)$$

Thus, the effect of the CMAC learning modifies CMAC output according to $\delta U_C = -(\mu/B)(\partial J/\partial U_C)$ and results in the gradient descent of cost function J. Figure 6 shows the effectiveness of the CMAC learning in accelerating the convergence of normal acceleration A_Z to its target value.

Robustness Properties

The purpose of online CMAC learning is for it to approximate the following equation:

$$\begin{aligned} U_c = U + K_r W_I \cdot A \cdot \dot{\alpha} - U_G = [(QSd/I_y)C_{M\delta} \\ - (QS/V_m W)C_{N\delta} \cdot \sin\alpha \cdot \dot{\alpha}]^{-1}\{\ddot{\alpha} - (QSd/I_y)[C_M(\alpha) + C_{M\delta} \\ \cdot (K_r\dot{\alpha} + K_r W_I \alpha)] - (QS/V_m W)[C_{N\alpha}(\alpha)\cos\alpha \\ - C_N(\alpha)\sin\alpha - C_{N\delta} \cdot (K_r\dot{\alpha} + K_r W_I \alpha)\sin\alpha]\dot{\alpha}\} \\ + K_r W_I \cdot A \cdot \dot{\alpha} - U_G = f(\alpha, \dot{\alpha}, \ddot{\alpha}, U_G) \end{aligned} \qquad (47)$$

where α, $\dot{\alpha}$, $\ddot{\alpha}$, and U_G are the inputs to the CMAC, so that an inverse control from AOA to control deflection can be achieved. The proposed CMAC control scheme is robust to nonlinearities, parameter uncertainties, and time delays.

1) Instead of the traditional gain-scheduling approach, the CMAC learns to approximate the system nonlinearity f in Eq. (47) directly.

2) Because the CMAC is trained online, when parameters in f such as aerodynamic coefficients C_N, C_M, $C_{N\delta}$, and $C_{M\delta}$ vary, the system control can respond immediately. Figure 7 shows how the CMAC controller handles variations in C_N and C_M.

3) In our control scheme, the CMAC and CFC are integrated as the system controller. Although we do not expect the CMAC to handle time delays, the adequate PM that is designed into the CFC can overcome this problem. Figure 8 shows the response of the system under time delay.

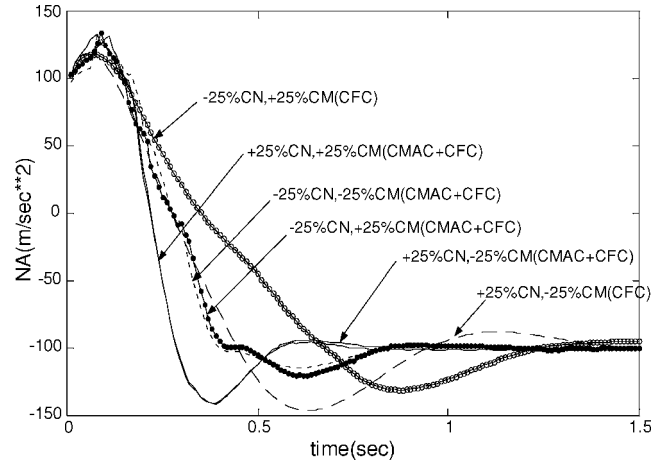

Fig. 7 Robustness test about $\pm 25\%$ variation on C_N and C_M.

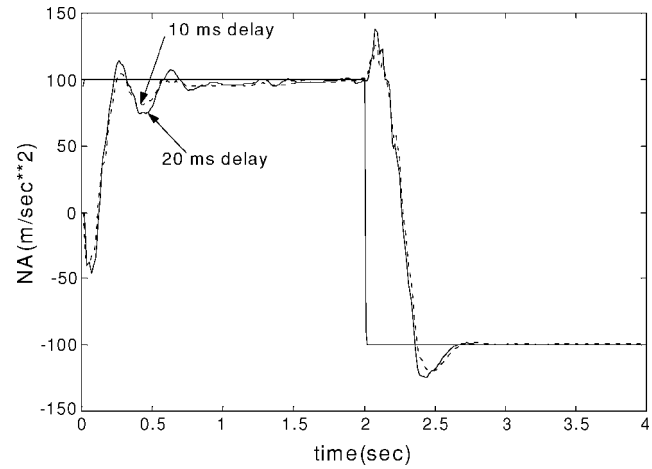

Fig. 8 Robustness with regard to time delay at actuator output.

V. Simulation Case Studies

Simulation studies were performed to validate the CMAC NN-based autopilot. The plant model used in the simulation includes a second-order actuator model, limits on control surface deflections and rates, and aerodynamic coefficients obtained from look-up tables as functions of AOA using linear interpolation.

As described in the preceding sections, path planning based on the desired acceleration of AOA is employed to limit the control bandwidth of the CMAC control system. The AOA acceleration $\ddot{\alpha}$ is related to deflection δ and deflection rate $\dot{\delta}$ by $\ddot{\alpha} = Z_\alpha\dot{\alpha} + M_\alpha\alpha + M_\delta\delta + Z_\delta\dot{\delta}$. Suppose the maximum deflection and the maximum deflection rate of the tail fin are set to be ± 25 deg (0.436 rad) and 500 deg/s (8.72 rad/s). Under the flight condition $M_\delta = -37.52$ 1/s^2 and $Z_\delta = -0.23$ 1/s, the maximum acceleration of AOA (radians per square second) is limited to

$$\ddot{\alpha}_{\max} = \text{maximum}(|\ddot{\alpha}|) = 37.52 \times 0.436 + 0.23 \times 8.72 = 18.36$$

Considering the uncertainties of aerodynamic coefficients and the restoring torque of missile body force on the transient stage, we further reduce the maximum acceleration to be $\ddot{\alpha}_{\max} = 10$ rad/s^2 in our simulation. For the rate limit on AOA in Eq. (30), we set bias $= 0.5$ rad/s and $P_{\text{gain}} = 0.02$ rad $\cdot$ s/m. To determine the time constant a in the design of the low-pass filter described in Eq. (29), we will consider the closed-loop response of the inner loop. From Eq. (5), with α typically small, one has

$$\dot{\alpha}/\delta = (q/\delta) + (1/V_m) \times A_Z/\delta \qquad (48)$$

In Eq. (48), the first term in the right-hand side is the dominant term; thus, we may approximate $\dot{\alpha}/\delta$ as q/δ, whose crossover frequency

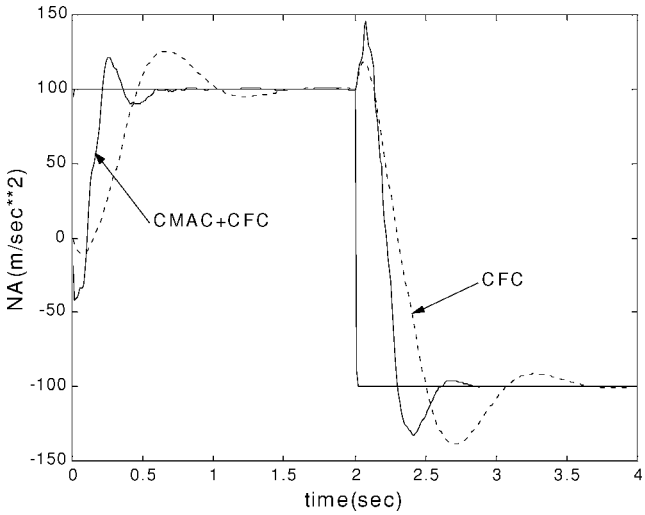

Fig. 9 Performance of CMAC control system and performance of CFC only.

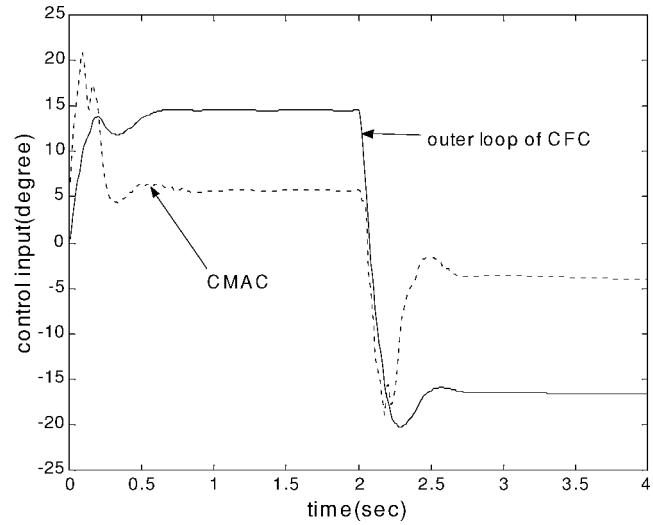

Fig. 10 Control effort generated by CMAC NN and generated by the outer loop of CFC.

is $\omega_{cr} \approx |K_r M_\delta| = 0.5 \times 37.52 = 18.76$ (see Ref. 1). In general, it is reasonable to design the bandwidth of the shaping filter larger than ω_{cr} to generate the desired rate of AOA. However, from the view of CMAC learning, the larger filter bandwidth may degrade learning efficiency. Thus, there is a tradeoff in determining the bandwidth of the low-pass filter. Here the time-constant a is set to be 0.028 s.

In implementing the CFC, the AOA α is used as the scheduling variable. The gains K_a, W_I, and K_r for four different flight conditions are listed in Table 1. Gains in between those listed in Table 1 can be interpolated with respect to α.

According to Eq. (38), the CMAC has four inputs and one output. How precisely the CMAC can approximate a function is mainly determined by the quantization in each dimension of the input vector. Reducing quantization would quickly increase the memory demand for storing the CMAC weights. During our simulation, the generalization technique is employed to speed up learning and control. Because the memory requirement is large, the technique of hash coding is employed to avoid collision.[18] Unless specified otherwise, the parameters of the CMAC neural network used in simulation are learning rate $\mu = 0.2$, low-pass filter time constant $a = 0.028$ s, quantization of AOA $\alpha = 0.1$ deg, quantization of $U_G = 0.08$ deg, and generalization size $g = 120$.

Case 1: Step Normal Acceleration Command Tracking

The CMAC NN-based autopilot provides good performance in step command tracking. Figure 9 shows a 4-s piecewise constant normal acceleration (NA) command $A_{ZC}(t)$, the NA output provided by CFC only, and the NA output provided by the CMAC NN-based autopilot after 15 cycles of training.

The control effort from the CMAC, that is, U_C, and the control effort from the CFC outer loop, that is U_O are shown in Fig. 10. It is indicated that U_C contributes more in the transient stage, but the U_O plays a dominant role when the autopilot is at the trim condition.

Case 2: Convergence and Robustness

In Sec. IV, it is shown that when the system is near a target trim flight condition, then the CMAC learning is of a gradient type. Figure 6 shows 10-g step-command simulation results for the CMAC NN-based autopilot after one and five cycles of learning. The CMAC learning accelerates the convergence of NA A_Z toward its target value.

With regard to robustness, Fig. 7 shows -20-g step-command simulation results for the CMAC NN-based autopilot with four possible combinations of $\pm 25\%$ variations in the two aerodynamic coefficients C_N and C_M, indicating good robustness of this controller to parametric uncertainty. Generally, $\pm 25\%$ variations in the aerodynamic coefficients are an adequate test in missile control design.[19]

With regard to unmodeled of time delay, consider time delays of 10 and 20 ms at the actuator output. The simulation results shown in Fig. 8 indicate that our scheme is also robust to moderate time delays.

VI. Conclusions

In this paper we propose a way to integrate the CMAC with existing CFC to improve performance. Although we proposed a transformation of the CFC, no change in existing CFC design is needed. When the CMAC is disabled, the controller (which includes the CMAC and the CFC) reduces to the original CFC. With basic stability and performance provided by the CFC, the effect of the CMAC can be adjusted by tuning the learning rate μ.

The local learning property of the CMAC allows it to approximate quickly nonlinear mappings, and the generalization property can convert the learned information into effective control. These are the reasons the CMAC controller can quickly improve performance, particularly during the transient period. With its learning capability, the CMAC controller can also tolerate uncertainties and parameter variations.

Because of the nonlinear characteristics of both the missile dynamics and the CMAC net, it is not practical to present analytical stability guarantees for the control approach adopted in this paper, except to state that stability of the CFC is preserved when $\mu = 0$. Nonetheless, when the system is near a target trim flight condition or when the CMAC net represents the inverse mapping to sufficient accuracy, we showed that the CMAC learning is a gradient rule that reduces the tracking error.

VII. Appendix: Aerodynamics Coefficients of C_N, C_M, $C_{N\delta}$, and $C_{M\delta}$

The coefficients are as follows:

$$C_N(\alpha, M_m) = \begin{bmatrix} C_N(-17, 3) \\ C_N(-12, 3) \\ C_N(-7, 3) \\ C_N(-2, 3) \\ C_N(0, 3) \\ C_N(2, 3) \\ C_N(7, 3) \\ C_N(12, 3) \\ C_N(17, 3) \end{bmatrix} = \begin{bmatrix} 5.305 \\ 3.377 \\ 1.730 \\ 0.793 \\ 0.0 \\ -0.793 \\ -1.730 \\ -3.377 \\ -5.305 \end{bmatrix}$$

$$C_M(\alpha, M_m) = \begin{bmatrix} C_M(-17, 3) \\ C_M(-12, 3) \\ C_M(-7, 3) \\ C_M(-2, 3) \\ C_M(0, 3) \\ C_M(2, 3) \\ C_M(7, 3) \\ C_M(12, 3) \\ C_M(17, 3) \end{bmatrix} = \begin{bmatrix} -0.759 \\ -0.206 \\ -0.383 \\ -0.643 \\ 0.0 \\ 0.643 \\ 0.383 \\ 0.206 \\ 0.759 \end{bmatrix}$$

$$C_{N\delta} = -2.636, \qquad C_{M\delta} = -14.35$$

References

[1]Nesline, F. W., and Nesline, M. L., "How Autopilot Requirements Constrain the Aerodynamic Design of Homing Missiles," *Proceedings of American Control Conference*, American Automatic Control Council, Evanston, IL, 1984, pp. 716–730.

[2]Elliott, J. R., "NASA's Advanced Control Law Program for the F4 Digital Fly-by-Wire Aircraft," *IEEE Transactions on Automatic Control*, Vol. AC-22, No. 5, 1977, pp. 753–757.

[3]Athans, M., Castanon, D., Dunn, K. P., Greene, C., Lee, W. H., Sandell, N. R., and Willsky, A. S., "The Stochastic Control of the F-8C Aircraft Uses a Multiple Model Adaptive Control(MMAC) Method—Part I: Equilibrium Flight," *IEEE Transactions on Automatic Control*, Vol. AC-22, No. 5, 1977, pp. 767–780.

[4]Hiret, A., Duc, G., Friang, J. P., and Farret, D., "Linear-Parameter-Varying/Loop-Shaping H_∞ Synthesis for a Missile Autopilot," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 879–886.

[5]Iglesias, P. A., and Urban, T. J., "Loop Shaping Design for Missile Autopilots: Controller Configurations and Weighting Filter Selection," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 3, 2000, pp. 516–525.

[6]Shue, S.-P., and Agarwal, R. K., "Nonlinear H_∞ Method for Control of Wing Rock Motion," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 1, 2000, pp. 60–67.

[7]Lin, C.-F., Cloutier, J. R., and Evers, J. H., "Hight-Performance, Robust, Bank-to-Turn Missile Autopilot Design," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 46–53.

[8]Reiner, J., Balas, G. J., and Garrard, W. L., "Robust Dynamic Inversion for Control of Highly Maneuverable Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 18–24.

[9]Snell, S. A., and Stout, P. W., "Robust Longitudinal Control Design Using Dynamic Inversion and Quantitative Feedback Theory," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, 1997, pp. 933–940.

[10]Simpson, P. K., *Artificial Neural Systems*, 1st ed., Pergamon, Fairview Park, NY, 1990, pp. 30–133.

[11]Sadhukhan, D., and Feteih, S., "F8 Neurocontroller Based on Dynamic Inversion," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 1, 1996, pp. 150–156.

[12]Gili, P. A., and Battipede, M., "Adaptive Neurocontroller for a Nonlinear Combat Aircraft Model," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 910–917.

[13]Calise, A. J., Hovakimyan, N., and Idan, M., "Adaptive Output Feedback Control of Nonlinear Systems Using Neural Networks," *Automatica*, Vol. 37, Aug. 2001, pp. 1201–1211.

[14]Albus, J. S., "Data Storage in the Cerebellar Model Articulation Controller," *Journal of Dynamic Systems, Measurement and Control*, Vol. 97, Sept. 1975, pp. 228–233.

[15]Albus, J. S., "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," *Journal of Dynamic Systems, Measurement and Control*, Vol. 97, Sept. 1975, pp. 220–227.

[16]Ker, J.-S., Kuo, Y.-H., Wen, R.-C., and Liu, B.-D., "Hardware Implementation of CMAC Neural Network with Reduced Storage Requirement," *IEEE Transactions on Neural Networks*, Vol. 8, No. 6, 1997, pp. 1545–1556.

[17]Chang, C.-H., and Han, K.-W., "Gain Margins and Phase Margins for Control Systems with Adjustable Parameters," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, 1990, pp. 404–408.

[18]Ellison, D., "On the Convergence of the Multidimensional Albus Perception," *International Journal of Robotics Research*, Vol. 10, No. 4, 1991, pp. 338–357.

[19]Moore, F. G., "State-of-the-Art Engineering Aero-Prediction Methods with Emphasis on New Semiempirical Techniques for Predicting Nonlinear Aerodynamics on Complete Missile Configurations," U.S. Naval Surface Warfare Center, NSWCDD/TR-93/551, Dahlgren, VA, Nov. 1993.