2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Data Learning Based Hypersonic Flight Control Using ELM

Shangmin Zhang, Shixing Wang, Yongquan Zhang, Yu Zhang and Jinrui Ren

***Abstract*— This paper is towards the controller design using extreme learning machine (ELM) for the longitudinal dynamics of a generic hypersonic flight vehicle (HFV). The basic idea is to train the data learning from previous controller and then obtain the optimal weight. In the first step, the existed the back-stepping controller with high order neural networks (HONNs) is borrowed to collect the required data. The "adaptive behavior" of existed the back-stepping controller is trained and tested by batch learning of ELM. Then the optimal parameters obtained from ELM are used as initialization to construct the feedback design for controller. In this way, the prior information of nominal design is not needed and there is no need of online learning for the neural networks (NNs). The simulation study is presented to show the effectiveness of the proposed control approach.**

I. INTRODUCTION

Hypersonic flight refers to the flight speed larger than 5 Mach. Due to the high speed, HFV provides more efficient way access the space and make the military purpose for "global attack" available. Recent decades numerous study has been posed on hypersonic flight control since the control technology is essential for the whole system. Due to the large flight envelope, nonlinear control is winning more and more attention such as sliding mode control[1], back-stepping control[2][3].

Among the different methods, due to the universal approximation ability, intelligent control using fuzzy logic systems[4][5][6] or NN is with great importance [7], [8]. For hypersonic controller, to tackle the continuous system uncertainty, the radial basis function (RBF) neural network is employed to approximate the uncertainty in [9][10]. In [11], the dynamic surface control with fuzzy logic system is proposed. Furthermore, the back-stepping control and singularly perturbed system approach with RBF NN approximating the unknown hypersonic dynamics [12]. Focused on discrete time design, the adaptive NN back-stepping HFV control [13] is studied to deal with the system uncertainty. In [14][15], the Kriging system is used to approximated the uncertainty. However, one concern is the computation burden. In [16], the NN improved method with "minimal learning parameter" design [17] is employed for NN weights updating while the nominal value of control gain is used to construct the controller.

ELM works for generalized single-hidden layer feed-forward networks (SLFNs). ELM was firstly proposed in [18] in view of machine learning and the essence of ELM is that the hidden layer of SLFNs need not to be tuned. Now the method has attracted widespread concern in recent years [19] and extended to more general cases. Also in view of developing new ELM updating law, it provides new method updating the weights with Lyapunov approach. The discrete ELM based design is studied in [20] and in the design the weights need updating during every step.

It has been widely analyzed that ELM provides fast way training data. So in this paper, instead of updating the SLFN every step, we employ ELM to learn the "adaptive behavior" quickly. It means the controller already proposed by other papers could be used to run the process then we can collect the required data that we are interested in. In this paper, the initialization phase is conducted by employing the controller from [16] by minor modification of the adaption item to obtain the chunk of initial training data. In the next step, the optimal weights are used for approximation directly during the implementation of the controller.

This paper is organized as follows. Section II describes the longitudinal dynamics of a generic hypersonic flight vehicle. The strict-feedback form is formulated and the discrete analysis model is obtained in Section III. SLFNs based on ELM are illustrated in Section IV. Section V presents the adaptive controller design based on ELM. The simulation result is included in Section VI. Section VII presents several comments and final remarks.

II. HYPERSONIC FLIGHT VEHICLE MODEL

The longitudinal model of HFV is comprised of five state variables $X = [V, h, \alpha, \gamma, q]^T$ and two control inputs $U_c = [\delta_e, \beta]^T$ where V is the velocity, γ is the flight path angle, h is the altitude, α is the attack angle, q is the pitch rate, δ_e is elevator deflection and β is the throttle setting. The dynamics are described by the following nonlinear equations:

$$\dot{V} = \frac{T\cos\alpha - D}{m} - \frac{\mu\sin\gamma}{r^2} \quad (1)$$

$$\dot{h} = V\sin\gamma \quad (2)$$

$$\dot{\gamma} = \frac{L + T\sin\alpha}{mV} - \frac{(\mu - V^2 r)\cos\gamma}{Vr^2} \quad (3)$$

$$\dot{\alpha} = q - \dot{\gamma} \quad (4)$$

$$\dot{q} = \frac{M_{yy}}{I_{yy}} \quad (5)$$

More detail of the model can refer to [21].

Shangmin Zhang and Shixing Wang are with Department of Computer Science, Tsinghua University (wangsx07@mails.tsinghua.edu.cn).

Yongquan Zhang is with Systems Engineering Research Institute, China State Shipbuilding Corporation.

Yu Zhang is with School of Aeronautics and Astronautics, Zhejiang University (zhangyu80@zju.edu.cn).

Jinrui Ren is with School of Automation, Northwestern Polytechnical University

This work was supported by by National Science Foundation of China (Grant No: 61134004, 61304098, 61273153), Fundamental Research Funds for the Central Universities (2012QNA4024), and Natural Science Basic Research Plan in Shanxi Province (Grant No: 2014JQ8326).

III. System Transformation

A. Strict-Feedback Formulation

Referred to [13], the formulation of the subsystems is presented in (6) and (8).

The velocity subsystem (1) can be rewritten as follows:

$$\begin{aligned} \dot{V} &= f_v + g_v u_v \\ u_v &= \beta \\ y_v &= V \end{aligned} \tag{6}$$

The tracking error of the altitude is defined as $\tilde{h} = h - h_d$ and the flight path command is chosen as

$$\gamma_d = \arcsin\left[\frac{-k_h\left(h-h_d\right) - k_I \int \left(h-h_d\right)dt + \dot{h}_d}{V}\right] \tag{7}$$

if $k_h > 0$ and $k_I > 0$ are chosen and the flight path angle is controlled to follow γ_d, the altitude tracking error is regulated to zero exponentially[22].

Assumption 1: The thrust term $T\sin\alpha$ in (3) is neglected because it is generally much smaller than L.

Define $X_A = [x_1, x_2, x_3]^T, x_1 = \gamma, x_2 = \theta_p, x_3 = q$ where $\theta_p = \alpha + \gamma$. Then the strict-feedback form equations of the attitude subsystem (3)-(5) are written as

$$\begin{aligned} \dot{x}_1 &= f_1\left(x_1\right) + g_1\left(x_1\right)x_2 \\ \dot{x}_2 &= f_2\left(x_1, x_2\right) + g_2\left(x_1, x_2\right)x_3 \\ \dot{x}_3 &= f_3\left(x_1, x_2, x_3\right) + g_3\left(x_1, x_2, x_3\right)u_A \\ u_A &= \delta_e \\ y &= x_1 \end{aligned} \tag{8}$$

The related definition of f_i and g_i, $i = 1, 3, v$ can be referred to [13], [16] and can be found in Appendix.

Assumption 2: f_i and g_i are unknown smooth functions and can be decomposed into the nominal part f_{iN}, g_{iN} and the unknown part $\Delta f_i, \Delta g_i$. There exist constants $\bar{g}_i$ and $\underline{g}_i$ such that $\bar{g}_i \geq g_i \geq \underline{g}_i > 0$, $i = 1, 3, v$.

B. Control Goal

The goal pursued in this study is to design a dynamic controller δ_e and β to steer system altitude and velocity from a given set of initial values to desired trim conditions with the tracking reference h_d and V_d. With the command transformation (7), the control objective of system (8) is to design an adaptive controller, which makes $\gamma \to \gamma_d$, further $h \to h_d$ and all the signals involved are bounded.

C. Discrete-time Model

By Euler expansion with sample time T_s, systems (6) and (8) can be approximated as

$$V(k+1) = V(k) + T_s[f_v(k) + g_v(k)u_v(k)] \tag{9}$$

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s[f_1(k) + g_1(k)x_2(k)] \\ x_2(k+1) &= x_2(k) + T_s[f_2(k) + g_2(k)x_3(k)] \\ x_3(k+1) &= x_3(k) + T_s[f_3(k) + g_3(k)u_A(k)] \end{aligned} \tag{10}$$

For a desired function U^*, it is assumed there exists an ideal weight vector ω^* such that the smooth function vector can be approximated by an ideal NN on a compact set

$$U^* = \omega^{*T}\theta(X) + \varepsilon(X), \|\varepsilon(X)\| < \varepsilon_M \tag{11}$$

where $\varepsilon(X)$ is the bounded NN approximation error vector and ε_M is the supreme of $\varepsilon(X)$.

Assumption 3: [23] For each $i = 1, 2, \cdots, n$ on the compact set Ω_i, ω_i^* satisfies

$$\|\omega_i^*\| \leq \phi_i \tag{12}$$

IV. SLFNs based on ELM

For N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i)$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in R^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^T \in R^m$, standard SLFN with $\tilde{N}$ hidden neurons can be expressed as follows:

$$\sum_{i=1}^{\tilde{N}} \beta_i G(\mathbf{x}_j; \mathbf{w}_i, b_i) = \mathbf{o}_j, j = 1, \cdots, N \tag{13}$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, \cdots, w_{in}]^T$ and b_i are the learning parameters of hidden nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$ is the weight vector connecting the ith hidden neuron and the output neurons and $G(\mathbf{x}_j; \mathbf{w}_i, b_i)$ is the output of the ith hidden node with respect to input $\mathbf{x}_j$.

Let $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$, $\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{\tilde{N}}]$ and $\mathbf{b} = [b_1, b_2, \cdots, b_{\tilde{N}}]$. The standard SLFN with $\tilde{N}$ hidden neurons each with function $g(x)$ can approximate these N samples with zero error means that

$$\sum\nolimits_{j=1}^{N} \|\mathbf{o}_j - \mathbf{t}_j\| = 0, i.e. \tag{14}$$

there exist β_i, $\mathbf{w}_i$ and b_i such that

$$\mathbf{H}\left(\mathbf{x}; \mathbf{w}, \mathbf{b}\right)\beta = \mathbf{T} \tag{15}$$

in which

$$\begin{aligned} &\mathbf{H}(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{w}_1, \cdots, \mathbf{w}_{\tilde{N}}, b_1, \cdots, b_{\tilde{N}}) \\ &= \begin{bmatrix} G(\mathbf{x}_1; \mathbf{w}_1, b_1) & \cdots & G(\mathbf{x}_1; \mathbf{w}_{\tilde{N}}, b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ G(\mathbf{x}_N; \mathbf{w}_1, b_1) & \cdots & G(\mathbf{x}_N; \mathbf{w}_{\tilde{N}}, b_{\tilde{N}}) \end{bmatrix}_{N\times\tilde{N}}, \end{aligned}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N}\times m} \text{ and } \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N\times m}$$

In the case of $\tilde{N} \ll N$ and $\mathbf{H}$ being a nonsquare matrix, one may interest to find $\hat{\mathbf{w}}_i, \hat{b}_i, \hat{\beta}_i (i = 1, \cdots, \tilde{N})$ such that

$$\begin{aligned} &\left\|\mathbf{H}(\hat{\mathbf{w}}_1, \cdots, \hat{\mathbf{w}}_{\tilde{N}}, b_1, \cdots, b_{\tilde{N}})\hat{\beta} - \mathbf{T}\right\| \\ &= \min_{\mathbf{w}_i, b_{i,\beta}} \|\mathbf{H}(\mathbf{w}_1, \cdots, \mathbf{w}_{\tilde{N}}, b_1, \cdots, b_{\tilde{N}})\beta - \mathbf{T}\| \end{aligned} \tag{16}$$

For fixed input weights $\mathbf{w}_i$ and the hidden layer biases b_i, to train a SLFN is simply equivalent to finding a least-squares solution β of the linear system $\mathbf{H}\beta = \mathbf{T}$:

$$\begin{aligned} & \left\| \mathbf{H}(\mathbf{w}_1, \cdots, \mathbf{w}_{\widetilde{N}}, b_1, \cdots, b_{\widetilde{N}})\beta - \mathbf{T} \right\| \\ = \ & \min_{\beta} \left\| \mathbf{H}(\mathbf{w}_1, \cdots, \mathbf{w}_{\widetilde{N}}, b_1, \cdots, b_{\widetilde{N}})\beta - \mathbf{T} \right\| \end{aligned} \quad (17)$$

The unique smallest norm least-squares solution of the above linear system is:

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \quad (18)$$

V. Adaptive Control with ELM

The idea in this paper is to find the easy way to control a hypersonic aircraft. As we know, if the system states for NN inputs are within a domain and the unknown function can be approximated exactly by NN, then the function can be substituted by the NN output with obtained NN weights and the system states. So in this part, we have the following design procedure:

1) Borrowing the controller from [16], the training data including the nominal value, HONN approximation and NN inputs vector are extracted for ELM.
2) With the optimal ELM training result, we applied the ELM neural feedback directly. Finally the method is tested for hypersonic flight control.

In order to show the idea, we recall the direct NN back-stepping design [16].

Firstly, the error definition is presented

$$z_1(k) = x_1(k) - x_{1d}(k) \quad (19)$$
$$z_2(k) = x_2(k) - x_{2d}(k) \quad (20)$$
$$z_3(k) = x_3(k) - x_{3d}(k) \quad (21)$$

where $x_{1d}(k)$ is derived from (7), $x_{2d}(k)$, $x_{3d}(k)$ are the virtual control inputs to be designed.

For simplicity, we define $G_i(k) = T_s g_i(k)$, $\bar{G}_i = T_s \bar{g}_i$, $G_{iN}(k) = T_s g_{iN}(k)$, $\varepsilon_i(k) = \varepsilon_i(X_i(k))$, $\theta_i(k) = \theta_i(X_i(k))$, $i = 1, 2, 3, v$.

Secondly, the related NN design is illustrated in the following 3 steps:

Step 1. From the definition of $z_1(k)$ in (19), we have

$$z_1(k+1) = x_1(k) + T_s[f_1(k) + g_1(k)x_2(k)] - x_{1d}(k+1) \quad (22)$$

where $x_{1d}(k+1)$ is acquired from (7).

Define $X_1(k) = [V(k), x_1(k), h_d(k+1)]^T$.

Take $x_2(k)$ in (22) as the virtual control input and design its desired value as

$$\begin{aligned} x_{2d}(k) &= \frac{-x_1(k) - T_s f_{1N}(k) + c_1 z_1(k) + x_{1d}(k)}{G_{1N}(k)} \\ &+ \frac{\phi_{s1}^2(k)}{\phi_{s1}(k) + \tau_1} \end{aligned} \quad (23)$$

with $0 < c_1 < 1$, $\phi_{s1}(k) = \hat{\phi}_1(k)\|\theta_1(k)\|$ where $\hat{\phi}_1 > 0$ is the estimation of ϕ_1 $f_{1N}(k)$ is the nominal part of $f_1(k)$,.

The adaptive law is

$$\begin{aligned} \hat{\phi}_1(k+1) &= \hat{\phi}_1(k) - \lambda_1\delta_1\hat{\phi}_1(k) \\ &- \lambda_1\|\theta_1(k)\| z_1(k+1) \end{aligned} \quad (24)$$

In the next steps, for simplicity, only the controller design and adaption law are presented. The analysis in detail could be found in [16].

Step 2. From (20),

$$z_2(k+1) = x_2(k) + T_s[f_2(k) + g_2(k)x_3(k)] - x_{2d}(k+1) \quad (25)$$

Define $X_2(k) = [V(k), x_1(k), x_2(k), h_d(k+1), h_d(k+2)]^T$.

Take $x_3(k)$ in (25) as the virtual control input and design its desired value as

$$\begin{aligned} x_{3d}(k) &= \frac{-T_s f_{2N}(k) + (c_2 - 1)z_2(k)}{G_{2N}(k)} \\ &+ \frac{\phi_{s2}^2(k)}{\phi_{s2}(k) + \tau_2} \end{aligned} \quad (26)$$

with $0 < c_2 < 1$, $\phi_{s2}(k) = \hat{\phi}_2(k)\|\theta_2(k)\|$.

The adaptive law is

$$\begin{aligned} \hat{\phi}_2(k+1) &= \hat{\phi}_2(k) - \lambda_2\delta_2\hat{\phi}_2(k) \\ &- \lambda_2\|\theta_2(k)\| z_2(k+1) \end{aligned} \quad (27)$$

Step 3. From (21),

$$z_3(k+1) = x_3(k) + T_s[f_3(k) + g_3(k)u_A(k)] - x_{3d}(k+1) \quad (28)$$

Define $X_3(k) = [V(k), X_A^T(k), h_d(k+1), h_d(k+2), h_d(k+3)]^T$.

The elevator deflection is designed as

$$\begin{aligned} u_A(k) &= \frac{-x_3(k) - T_s f_{3N}(k) + c_3 z_3(k) + x_{3d}(k)}{G_{3N}(k)} \\ &+ \frac{\phi_{s3}^2(k)}{\phi_{s3}(k) + \tau_3} \end{aligned} \quad (29)$$

where $0 < c_3 < 1$, $\phi_{s3}(k) = \hat{\phi}_3(k)\|\theta_3(k)\|$ $f_{3N}(k)$ is the nominal part of $f_3(k)$.

The adaptive law is

$$\hat{\phi}_3(k+1) = \hat{\phi}_3(k) - \lambda_3\left[\|\theta_3(k)\| z_3(k+1) + \delta_3\hat{\phi}_3(k)\right] \quad (30)$$

For the velocity, define $X_v(k) = [V(k), X_A^T(k), V_d(k+1)]^T$ and $z_v(k) = V(k) - V_d(k)$

$$\begin{aligned} z_v(k+1) &= V(k+1) - V_d(k+1) \\ &= V(k) + T_s[f_v(k) + g_v(k)u_v(k)] \\ &- V_d(k+1) \end{aligned} \quad (31)$$

The throttle setting is designed as

$$\begin{aligned} u_v(k) &= \frac{-V(k) - T_s f_{vN}(k) + c_v z_v(k) + V_d(k+1)}{G_{vN}(k)} \\ &+ \frac{\phi_{sv}^2(k)}{\phi_{sv}(k) + \tau_v} \end{aligned} \quad (32)$$

with $0 < c_v < 1$, $\phi_{sv}(k) = \hat{\phi}_v(k)\|\theta_v(k)\|$.

The robust updating law for NN weights is as

$$\hat{\phi}_v\left(k+1\right) = \hat{\phi}_v\left(k\right) - \lambda_v\left[\left\|\theta_v(k)\right\| z_v(k+1) + \delta_v \hat{\phi}_v\left(k\right)\right] \quad (33)$$

Define

$$F_1(k) = \frac{-T_s f_{1N}(k)}{G_{1N}\left(k\right)} + \frac{\phi_{s1}^2\left(k\right)}{\phi_{s1}\left(k\right) + \tau_1}$$

$$F_2(k) = \frac{-T_s f_{2N}(k)}{G_{2N}\left(k\right)} + \frac{\phi_{s2}^2\left(k\right)}{\phi_{s2}\left(k\right) + \tau_2}$$

$$F_3(k) = \frac{-T_s f_{3N}(k)}{G_{3N}\left(k\right)} + \frac{\phi_{s3}^2\left(k\right)}{\phi_{s3}\left(k\right) + \tau_3}$$

$$F_v(k) = \frac{-T_s f_{vN}(k)}{G_{vN}\left(k\right)} + \frac{\phi_{sv}^2\left(k\right)}{\phi_{sv}\left(k\right) + \tau_v}$$

Now we obtain the training data with X_1, X_2, X_3, X_v as inputs and F_1, F_2, F_3, F_v as outputs.

In the next step, the ELM is employed for data training. Referred to Section IV, the function of $g(w_i \cdot x_j + b_i)$ is selected as

$$g(w_i \cdot x_j + b_i) = \frac{1}{1 + e^{-(w_i \cdot x_j + b_i)}} \quad (34)$$

In this way, the parameters of w_i, b_i and β_i are obtained and we denote the ELM estimation of $F_i(k)$ as $\hat{F}_i(k)$ where $i = 1, 2, 3, v$. Finally, the controller with ELM estimation is formulated as

$$x_{2d}(k) = \frac{-x_1(k) + c_1 z_1(k) + x_{1d}\left(k\right)}{G_{1N}\left(k\right)} + \hat{F}_1(k) \quad (35)$$

$$x_{3d}(k) = \frac{-x_2(k) + c_2 z_2(k) + x_{2d}\left(k\right)}{G_{2N}\left(k\right)} + \hat{F}_2(k) \quad (36)$$

$$u_A(k) = \frac{-x_3(k) + c_3 z_3(k) + x_{3d}\left(k\right)}{G_{3N}\left(k\right)} + \hat{F}_3(k) \quad (37)$$

$$u_v(k) = \frac{-V(k) + c_v z_v(k) + V_d\left(k+1\right)}{G_{vN}\left(k\right)} + \hat{F}_v(k) \quad (38)$$

The ELM estimation $\hat{F}_i(k)$ is the function of related NN inputs and the trained weights. With the NN approximation ability, the value will change according to the system states and provide the uncertainty information for the feedback design.

VI. Simulations

The flight of the vehicle is at the condition $M = 15$, $V = 15,060$ft/s, $h = 110,000$ft, $\alpha = 0.1$rad, $\gamma = 0$, $q = 0$.

Reference commands are generated by the filter:

$$\frac{h_d}{h_c} = \frac{\omega_{n1}\omega_{n2}^2}{(s + \omega_{n1})(s^2 + 2\varepsilon_c\omega_{n2}s + \omega_{n2}^2)}$$

$$\frac{V_d}{V_c} = \frac{\omega_{v1}}{(s + \omega_{v1})}$$

where $\omega_{n1} = 0.7$, $\omega_{n2} = 0.3$, $\varepsilon_c = 0.7$, $\omega_{v1} = 0.1$. The perturbation is set to be 3% for the parameter set (m, μ, Iyy, S).

The parameters for the ELM controller are selected as $k_h = 0.3$, $k_I = 0.1$, $T_s = 0.05s$, $c_1 = 0.95$, $c_2 = 0.85$, $c_3 = 0.7$, $c_v = 0.75$.

With the same parameters for NN adaption design [16], the data of the four groups is collected with step command of 500ft and 100ft/s for altitude and velocity separately. The ELM training and testing result is illustrated with Fig.1-Fig.8. From the high accuracy of both training and testing results, it indicates that the ELM has the favorable generalization capability in the hypersonic uncertainty modeling. The training result of the weights and biases is then applied directly to formulate the adaption item for hypersonic flight controller.

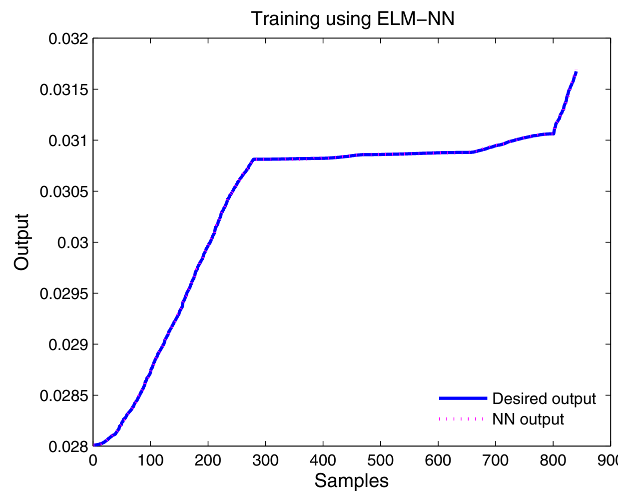

Fig. 1. Data Training:$X_1 - F_1$

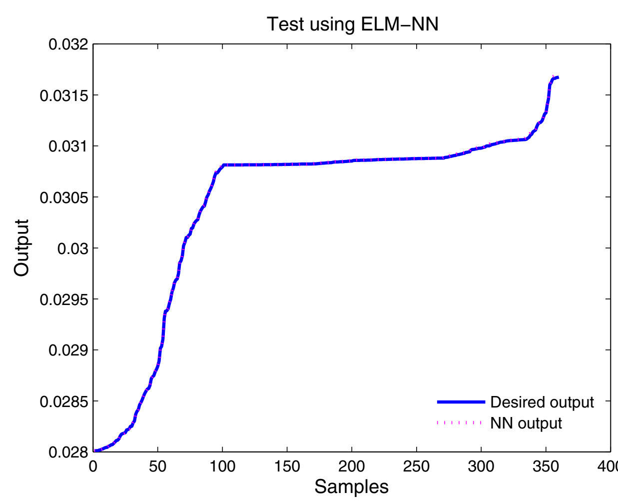

Fig. 2. Data Test:$X_1 - F_1$

Fig.9 depicts the response performance that the altitude controller tracks the step change with magnitude 500ft while the velocity steps from 15060ft/s to 15160ft/s. The control inputs of elevator deflection and throttle setting are illustrated in Fig.10. From the simulation study, ELM demonstrates good capability to learn from the system information. In this way, the controller is greatly simplified and the good performance is guaranteed.

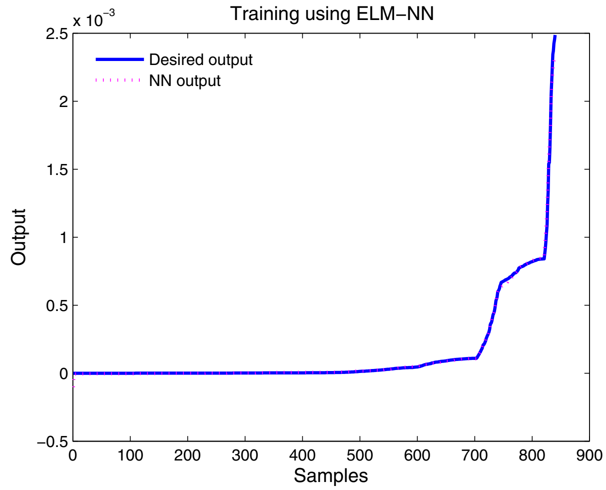

Fig. 3. Data Training: $X_2 - F_2$

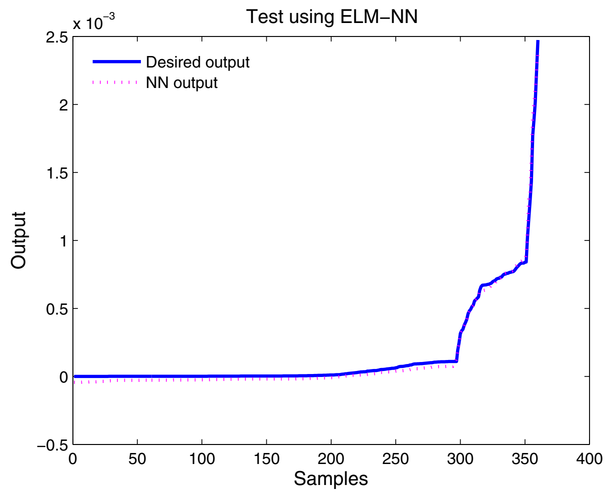

Fig. 4. Data Test: $X_2 - F_2$

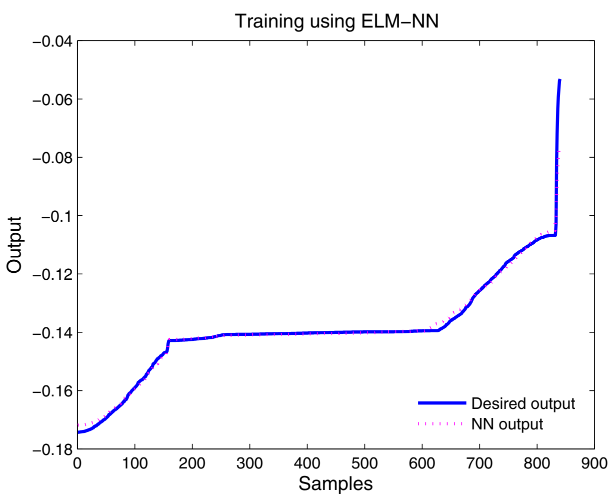

Fig. 5. Data Training: $X_3 - F_3$

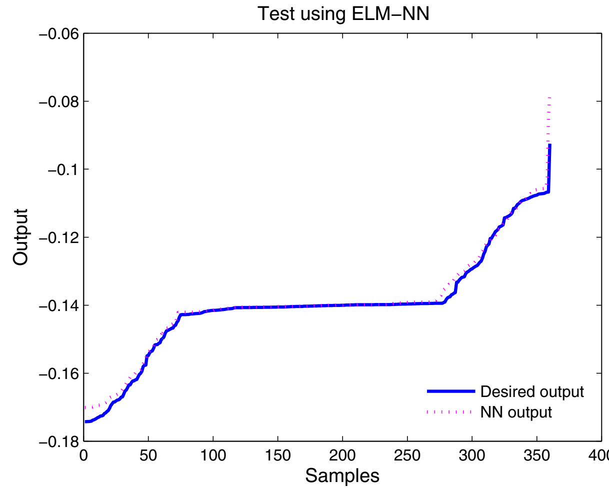

Fig. 6. Data Test: $X_3 - F_3$

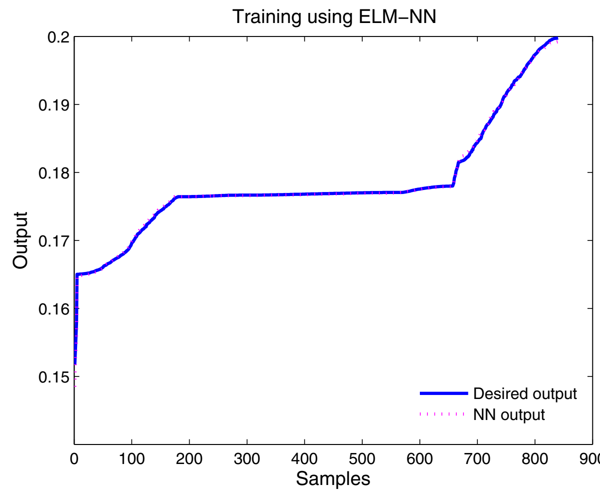

Fig. 7. Data Training: $X_v - F_v$

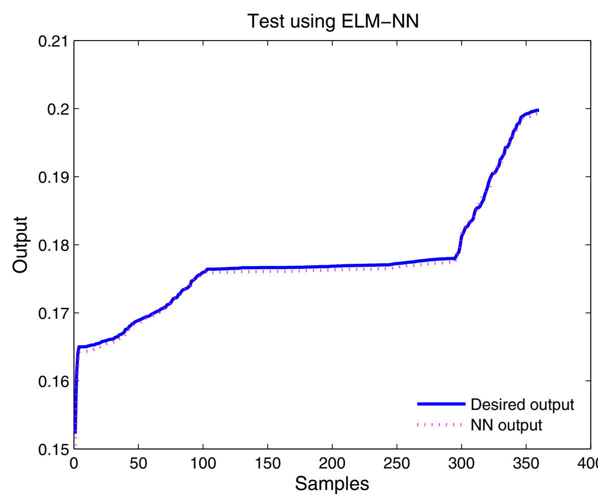

Fig. 8. Data Test: $X_v - F_v$

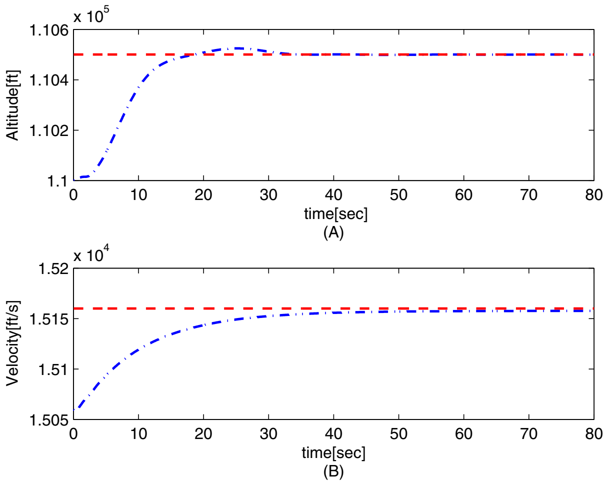

Fig. 9. Altitude and Velocity Tracking

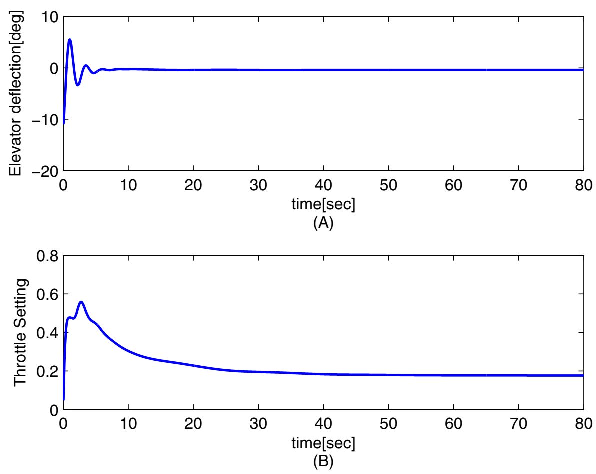

Fig. 10. Control Inputs

VII. Conclusions and Future Work

The ELM based discrete controller via back-stepping design is applied on hypersonic flight. Learning from the nominal and neural approximation information from the previous controller [16], the controller is easy to implement. The data is trained and tested by ELM to obtain the optimal weights. Finally the effectiveness of the method is verified by simulation. It shows the controller can achieve good performance in the presence of system uncertainty.

For future work, three concerns will be analyzed. One problem is that though the optimal weights are derived in the design, to accommodate more complicated situation, we will consider how to enhance the robustness of the controller. Furthermore, it is important to ensure closed-loop stability in case the system states are out of the approximation region. The other concern is that the similar idea could be tried on the prediction model based design[15][17][20] to simplify the design procedure.

Appendix

(A) Hypersonic Aircraft Model Description

The related definition of the hypersonic aircraft model is as: $r = h + R_E$, $\bar{q} = \frac{1}{2}\rho V^2$, $L = \bar{q}SC_L$, $D = \bar{q}SC_D$, $T = \bar{q}SC_T$, $M_{yy} = \bar{q}S\bar{c}[C_M(\alpha) + C_M(\delta e) + C_M(q)]$, $C_L = 0.6203\alpha$, $C_D = 0.6450\alpha^2 + 0.0043378\alpha + 0.003772$, $C_M(\alpha) = -0.035\alpha^2 + 0.036617\alpha + 5.3261\times 10^{-6}$, $C_M(q) = (q\bar{c}/2V)\times(-6.796\alpha^2 + 0.3015\alpha - 0.2289)$.

The control inputs related definition is as

$$\begin{aligned} C_T &= \begin{cases} 0.02576\beta, \text{if } \beta < 1 \\ 0.0224 + 0.00336\beta, \text{otherwise} \end{cases} \\ C_M(\delta e) &= 0.0292(\delta e - \alpha) \end{aligned}$$

where ρ denotes the air density, S is the reference area, $\bar{c}$ is the reference length and R_E is the radius of the Earth. $C_x, x = L, D, T, M$ are the force and moment coefficients.

(B) Definition for Nonlinear Function f_i and g_i, $i = 1, 2, 3, v$

$f_1 = -(\mu - V^2 r)\cos\gamma/(Vr^2) - 0.6203\bar{q}S\gamma/(mV)$, $g_1 = 0.6203\bar{q}S/(mV)$, $f_2 = 0$, $g_2 = 1$, $f_3 = \bar{q}S\bar{c}[C_M(\alpha) + C_M(q) - 0.0292\alpha]/I_{yy}$, $g_3 = 0.0292\bar{q}S\bar{c}/I_{yy}$.

$f_v = -(D/m + \mu\sin\gamma/r^2) + \bar{q}S \times 0.0224\cos\alpha/m$, $g_v = \bar{q}S \times 0.00336\cos\alpha/m$ if $\beta > 1$. Otherwise $f_v = -(D/m + \mu\sin\gamma/r^2)$, $g_v = \bar{q}S \times 0.02576\cos\alpha/m$.

References

[1] H. Xu, M. Mirmirani, and P. Ioannou, “Adaptive sliding mode control design for a hypersonic flight vehicle,” *Journal of guidance, control, and dynamics*, vol. 27, no. 5, pp. 829–838, 2004.
[2] B. Xu, S. Wang, D. Gao, Y. Zhang, and Z. Shi, “Command filter based robust nonlinear control of hypersonic aircraft with magnitude constraints on states and actuators,” *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 233–247, 2014.
[3] B. Xu, D. Wang, H. Wang, and S. Zhu, “Adaptive neural control of a hypersonic vehicle in discrete time,” *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 219–231, 2014.
[4] Y. Pan, M. Er, D. Huang, and Q. Wang, “Adaptive fuzzy control with guaranteed convergence of optimal approximation error,” *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 807–818, 2011.
[5] Y. Pan, Y. Zhou, T. Sun, and M. Er, “Composite adaptive fuzzy H^∞ tracking control of uncertain nonlinear systems,” *Neurocomputing*, vol. 99, pp. 15–24, 2013.
[6] W. Chen, L. Jiao, R. Li, and J. Li, “Adaptive backstepping fuzzy control for nonlinearly parameterized systems with periodic disturbances,” *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 4, pp. 674–685, 2010.
[7] B. Xu, C. Yang, and Z. Shi, “Reinforcement learning output feedback nn control using deterministic learning technique,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 635–641, 2014.
[8] B. Xu, Z. Shi, C. Yang, and F. Sun, “Composite neural dynamic surface control of a class of uncertain nonlinear systems in strict-feedback form,” *IEEE Transactions on Cybernetics, DOI:10.1109/TCYB.2014.2311824*.
[9] M. Chen, R. Mei, and B. Jiang, “Sliding mode control for a class of uncertain mimo nonlinear systems with application to near-space vehicles,” *Mathematical Problems in Engineering*, vol. 2013, 2013.
[10] M. Chen, C.-S. Jiang, and Q.-X. Wu, “Disturbance-observer-based robust flight control for hypersonic vehicles using neural networks,” *Advanced Science Letters*, vol. 4, no. 4-5, pp. 4–5, 2011.
[11] D. Gao, Z. Sun, and T. Du, “Dynamic surface control for hypersonic aircraft using fuzzy logic system,” in *IEEE International Conference on Automation and Logistics*, pp. 2314–2319, 2007.

[12] D. Gao, S. Wang, and H. Zhang, "A singularly perturbed system approach to adaptive neural back-stepping control design of hypersonic vehicles," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 249–259, 2014.

[13] B. Xu, F. Sun, C. Yang, D. Gao, and J. Ren, "Adaptive discrete-time controller design with neural network for hypersonic flight vehicle via back-stepping," *International Journal of Control*, vol. 84, no. 9, pp. 1543–1552, 2011.

[14] B. Xu, F. Sun, H. Liu, and J. Ren, "Adaptive kriging controller design for hypersonic flight vehicle via back-stepping," *IET control theory & applications*, vol. 6, no. 4, pp. 487–497, 2012.

[15] B. Xu and Z. Shi, "Universal kriging control of hypersonic aircraft model using predictor model without back-stepping," *IET control theory & applications*, vol. 7, no. 4, pp. 573–583, 2013.

[16] B. Xu, D. Wang, F. Sun, and Z. Shi, "Direct neural discrete control of hypersonic flight vehicle," *Nonlinear Dynamics*, vol. 70, no. 1, pp. 269–278, 2012.

[17] B. Xu, Z. Shi, C. Yang, and S. Wang, "Neural control of hypersonic flight vehicle model via time-scale decomposition with throttle setting constraint," *Nonlinear Dynamics*, vol. 73, no. 3, pp. 1849–1861, 2013.

[18] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 2, pp. 985–990, Ieee, 2004.

[19] G. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1, pp. 155–163, 2010.

[20] B. Xu, D. Wang, F. Sun, and Z. Shi, "Direct neural control of hypersonic flight vehicles with prediction model in discrete time," *Neurocomputing*, vol. 115, no. 4, pp. 39–48, 2013.

[21] Q. Wang and R. Stengel, "Robust nonlinear control of a hypersonic aircraft," *Journal of Guidance Control and Dynamics*, vol. 23, no. 4, pp. 577–585, 2000.

[22] B. Xu, D. Gao, and S. Wang, "Adaptive neural control based on hgo for hypersonic flight vehicles," *SCIENCE CHINA Information Sciences*, vol. 54, no. 3, pp. 511–520, 2011.

[23] Y. Liu, G. Wen, and S. Tong, "Direct adaptive nn control for a class of discrete-time nonlinear strict-feedback systems," *Neurocomputing*, vol. 73, no. 13-15, pp. 2498–2505, 2010.