SCIENCE CHINA
Information Sciences

• RESEARCH PAPERS •
Special Focus

March 2011 Vol. 54 No. 3: 448–459
doi: 10.1007/s11432-011-4193-z

Fuzzy dynamic characteristic model based attitude control of hypersonic vehicle in gliding phase

LUO Xiong[1,2]* & LI Jiang[1]

[1]*Department of Computer Science and Technology, Key Laboratory of Advanced Control of Iron and Steel Process (Ministry of Education), University of Science and Technology Beijing, Beijing* 100083*, China;*
[2]*National Laboratory of Space Intelligent Control, Beijing* 100190*, China*

Received May 5, 2010; accepted January 11, 2011

Abstract The intelligent autonomous control of hypersonic vehicles has aroused great interest from the field of spacecraft. To solve the problem of longitudinal attitude control of hypersonic vehicle in gliding phase, a new intelligent controller is proposed in this paper. This new controller is based on the fuzzy dynamic characteristic modeling method. The fuzzy logic is introduced into the characteristic modeling by dividing the whole restriction range into several subspaces. Simulations show that this modification greatly improves the performance of the original method. With the same whole restriction range the fuzzy dynamic characteristic modeling decreases the time of convergence, and at the same time makes the attitude angle tracing more precise and robust. Since the sub-model is a characteristic model that has stronger adaptiveness than a fixed local model, the number of fuzzy rules is greatly reduced. Our model sharply reduces the complexity in constructing a fuzzy dynamic model. Finally, simulation results are given to show the effectiveness of the proposed approach in dealing with the attitude control problem of hypersonic vehicle in gliding phase.

Keywords hypersonic vehicle, gliding attitude control, fuzzy dynamic characteristic modeling

Citation Luo X, Li J. Fuzzy dynamic characteristic model based attitude control of hypersonic vehicle in gliding phase. Sci China Inf Sci, 2011, 54: 448–459, doi: 10.1007/s11432-011-4193-z

1 Introduction

Hypersonic vehicle generally refers to a vehicle flying at Mach 5 or above, and usually far higher than normal plane. Because of its military significance, the hypersonic vehicle has become a hot research point in many countries. Just in 2010, the USA successfully launched the X-37B, a new vehicle of the X-project. If the whole test proves successful, it will be a major step forward to the real space weapon [1].

Because of its high speed and high altitude, it can easily break the enemy's defense line. In addition, it can serve as a platform for the long-range strike weapon. Although it has many advantages and very bright prospect, its high speed makes it very sensitive to the change of flight condition. So in fact, the hypersonic vehicle is a fast time varying, nonlinear and coupling system. It is very hard to build an accurate model for this system. The classical method is to choose several operating points. Then for

*Corresponding author (email: robertxiongluo@gmail.com)

each operating point an approximate model is established, which enables the traditional control method, such as PID, to be fit to use again [2].

To conquer the difficulty, various controllers have been proposed and designed. The X-33 used a control strategy based on nonlinear dynamic inverse with neural network. Although the X-33 project was finally canceled for some reasons, the controller was proved to be robust and error-tolerant [3]. Other control methods such as model predictive control and sliding mode control have also been introduced into the field of hypersonic vehicle control. However, the significant successful practice was the test flight of X-34A that still uses the traditional controller with lots of work points to be set and stored before [2]. Therefore, there is still a lot of work to do to introduce advanced control methods from other control fields and make it fit the control of hypersonic vehicle.

Recently, a new method based on the characteristic modeling was introduced into the control of hypersonic vehicle. The characteristic modeling method was proposed about 20 years ago [4]. Developing for many years, this method has proved successful both in theory and in practice. Moreover, it has been applied successfully to the control of over 400 engineering systems [4]. The characteristic modeling, together with the golden-section control strategy, forms an adaptive control system. To build such a controller, no accurate model is needed, even the approximate model is not very important. A significant success of the method is the application to the entry control of space ship. An adaptive entry guidance and trajectory control law of reentry vehicle was designed based on characteristic model [5]. Some good results were obtained. The same method was applied to the attitude control of a typical hypersonic vehicle X-34 [6]. The experiment demonstrated that the controller worked well during the whole climbing phase. Moreover, throughout the designing the physical model of the plane was not much concerned. Although in climbing phase the characteristic modeling seems to work well, we find that a single set of parameters was unlikely to enable the controller to perform well enough in every condition in the gliding phase.

Indeed, according to our early work, for the rapid tracking nonlinear system, because characteristic modeling needs to compress all the information of a high order model into several given characteristic parameters, it is difficult to meet the control performance requirements by merely using a characteristic model [7]. Thus, in order to describe some complex systems, we should improve the characteristic modeling by combining it with fuzzy logic.

In this paper, a fuzzy dynamic characteristic modeling method is described and applied to the attitude control of hypersonic vehicle in the gliding phase.

This paper is organized as follows. In the next section, the hypersonic vehicle model and attitude control architecture are introduced in brief. In section 3, the fuzzy dynamic characteristic modeling approach is presented. Meanwhile, the universal approximation of general fuzzy dynamic characteristic model system is discussed and proved. In section 4, the corresponding controller based on fuzzy dynamic characteristic model is given. In section 5, a simulation example is illustrated to show the application and the effectiveness of the proposed method. Finally, concluding remarks are presented in section 6.

2 Hypersonic vehicle control

2.1 Hypersonic vehicle model

In gliding phase, the engine is off. The only controllable parameter is δE (elevator deflection). The longitudinal analytical model of the hypersonic vehicle in gliding phase can be described by (see [8])

$$\dot{V} = -\frac{D}{m} - \frac{\mu}{r^2}\sin(\gamma), \tag{1}$$

$$\dot{\gamma} = -\frac{L}{mV} - \frac{\mu - V^2 r}{V r^2}\cos(\gamma), \tag{2}$$

$$\dot{h} = V\sin(\gamma), \tag{3}$$

$$\dot{\alpha} = q - \dot{\gamma}, \tag{4}$$

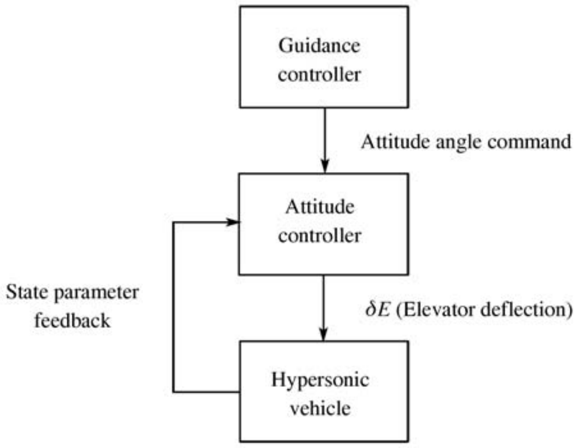

Figure 1 Two steps architecture of hypersonic vehicle controller.

$$\dot{q} = \frac{M_{yy}}{I_{yy}}, \tag{5}$$

$$L = 0.5\rho V^2 SC_L, \tag{6}$$

$$D = 0.5\rho V^2 SC_D, \tag{7}$$

$$M = 0.5\rho V^2 SC_M, \tag{8}$$

where V, q, h, α and γ refer to the velocity, pitch rate, altitude, angle of attack and flight-path angle, respectively. The coefficients C_L, C_D and C_M are functions of V, h, α and δE. But unlike in cruise phase such as work done in [8], which restricts the flight condition in a limit space, in gliding phase the flight conditions such as V and h continously change from time to time. That means although they in fact are functions, we could not describe them analytically. Therefore, in simulation we have to use tables to look for their current values.

2.2 Longitudinal attitude control of hypersonic vehicle

The final orient of hypersonic vehicle control is to make the flight move at required speed and in a certain path by giving a series of time-varying δE. But in fact people have found that there are some states of the vehicle that react very slowly to the change of δE, such as V, γ and h, while others react fastly. Therefore, it is very difficult to manipulate δE in order to make V or h reach an expected value directly. Thus, we divide the control procedure into two steps: guiding and attitude control. Figure 1 shows how the two parts collaborate.

Instead of controlling the "slow state" V or γ, the attitude controller gives δE only to get an expected attitude angle (for longitudinal control, it is α). Then from the viewpoint of guidance controller, the attitude angle is now a parameter to be used as an input. α is far more closely related to the slow states than δE. Therefore, it is now possible to control the whole flight. This paper will focus on the attitude controller.

In order to make the attitude angle controllable, the attitude controller often works at far higher "clock rate" than the guidance controller, and needs to get the flight into target attitude as fast as possible.

3 Fuzzy dynamic characteristic modeling

The fuzzy dynamic characteristic modeling is based on the characteristic modeling. By adding fuzzy logic into the classic characteristic model identifier, we can get a more accurate dynamic model that is fit to be used under more broad conditions.

3.1 Characteristic modeling

The characteristic modeling is to build up a model for target system based on the analysis of the dynamic feature of that system, with emphasis laid on the needs of controller. There are some characters of this model, as listed below [4]:

1) Under the same input, the characteristic model has an output equivalent to target system.

2) The rank of the characteristic model not only depends on the system feature, but more closely associates to the performance required.

3) The characteristic model has less complicated expressions and is more easily to be established in engineering practice.

4) Although characteristic model usually has lower rank than the real system, it is not just a rank-reduced model for it compresses the high rank information into a few characteristic coefficients.

Generally, a characteristic model is described by the 2-order differential equation below:

$$y(k+1) = f_1(k)\, y(k) + f_2(k)\, y(k-1) + g(k)\, u(k), \tag{9}$$

in which, $y(k)$ represents the output and $u(k)$ stands for the input at time step k. The coefficients f and g are slowly time varying, and change in a certain range.

To obtain such a model, instead of deducing from the analytical system equation, an online adaptive algorithm is used. With the online data, we could use any recursive regression method to find the coefficients we need. Here a gradient descent algorithm is given below [6]:

$$\theta(k+1) = \theta(k) + \frac{\lambda_1 \phi(k)}{\lambda_2 + \phi^{\mathrm{T}}(k)\, \phi(k)} \left[y(k+1) - \phi^{\mathrm{T}}(k)\, \theta(k)\right], \tag{10}$$

where $\theta(k) = [f_1(k), f_2(k), g(k)]$, $\phi(k) = [y(k), y(k-1), u(k)]$, λ_1 and λ_2 are two positive constants. After the coefficients of next time step are calculated they should be restricted to the range set before.

Because it is an online adaptive algorithm, it is not necessary to know the exact analytic model before. However, in order to approach the system, we need enough samples. That is why we emphasize that the differential equation should be slowly time-varying. If the coefficients change too fast, the algorithm will never have enough samples before coefficients change apparently. In order to make the model slowly time-varying, a direct method is setting a proper sampling interval. Theoretically, just by reducing this interval, the coefficients will finally get slow enough to be estimated. Although in engineering practice, we do not usually have the freedom to do so. At that time, we have to change our control strategy or employ another adaptive algorithm with higher performance.

3.2 Golden-section controller

The golden-section controller is always used together with the characteristic model. They are both parts of the whole control strategy. Theoretically, if a model we get by characteristic modeling is identical to the real system, the only controller we need is just the inverse of the model:

$$u(k) = \frac{y_r(k+1) - f_1(k)\, y(k) - f_2(k)\, y(k-1)}{g(k)}. \tag{11}$$

Here, y_r means the control target. If the coefficients can fully represent the system, this controller will get the output to the target immediately.

However in fact, there will always be modeling error, and more importantly, the characteristic model is to approach the target system through a series slowly time-varying coefficients. That means that we do not need to analyze the real system. Meanwhile, we only need to choose a lot of work points and build many local models. The characteristic model is in fact a local model of target system at each time step, but of course, automatically online adapted. Therefore, if the current state is far away from the target, the input gets from (11) cannot meet our needs. The whole closed-loop system may become unstable, especially when the coefficients have not get close enough to the “true value”.

To conquer this problem, the golden-section controller is proposed. It can be described by (see [4])

$$u_g(k) = \frac{l_1 \hat{f}_1(k)(y_r(k) - y(k)) + l_2 \hat{f}_2(k)(y_r(k-1) - y(k-1))}{\hat{g}(k)}, \tag{12}$$

where l_1=0.382, l_2=0.618, $\hat{f}_1$, $\hat{f}_2$ and $\hat{g}$ are the identification results of parameters f_1, f_2, and g obtained by using the weighted least squares method. However, because of the too large input, there was unstability [4]. The f_1 and f_2 were adjusted by multiplying the golden-section coefficients. Also it was proved that the golden-section coefficients could make the controller perform best while keeping the system stable at the same time [4].

Together with the golden-section controller, there are several other controllers usually used for the characteristic model. They are u_0 (keep controller), u_i (logical integration controller) and u_d (logical differential controller). They are (see [6])

$$u_0(k) = \frac{y_r(k) - \hat{f}_1(k) y_r(k-1) - \hat{f}_2(k) y_r(k-2)}{\hat{g}(k)}, \tag{13}$$

$$u_i(k) = u_i(k-1) - K_i(e(k)), \tag{14}$$

$$u_d(k) = -K_d(e(k)), \tag{15}$$

$$K_i = \begin{cases} K_{i1}, & e(k)(e(k) - e(k-1)) > 0, \\ K_{i2}, & e(k)(e(k) - e(k-1)) \leqslant 0, \end{cases} \tag{16}$$

$$K_d = C_d \sqrt{\sum_{j=0}^{l_d} |e(k-j)|}, \tag{17}$$

where $e(k) = y(k) - y_r(k)$, $K_{i1} \gg K_{i2}$, and C_d are positive constants. So the whole input is

$$u(k) = u_0(k) + u_g(k) + u_i(k) + u_d(k). \tag{18}$$

According to the requirements of problem, we can choose one or more of the above controllers to form the whole input.

3.3 Fuzzy dynamic characteristic modeling

The fuzzy dynamic characteristic modeling is based on the method introduced above. Characteristic modeling needs to compress all the information of the high order model into several given characteristic parameters. That is to say, because the characteristic modeling is essentially an online adaptive method, it is very important to restrict the result coefficients in a reasonable range in order to make the identifier precise enough.

In fact, the characteristic modeling is a searching process. Within the certain range, the model searching for a group of coefficients represents the local model best automatically by the samples online collected. Therefore, the range we set for the coefficients actually represents the solution space. That can properly explain why the range is important. If the solution space we set does not contain some local model, the algorithm will fail when the system runs into that space. On the contrary, if the solution space is too large it may take too many time steps to get a result precise enough. Sometimes that will finally make the closed-loop system unstable.

In order to construct a proper range, a usual method is to deduce an approximate analytical model and then sets a range according to the approximate range. However, for some problem especially when the system changes drastically, covering all the possible state of the system with a single range will cause an inevitable large solution space.

To solve this problem, we introduce the fuzzy logic to partition the large space into several smaller subspaces, as shown in Figure 2.

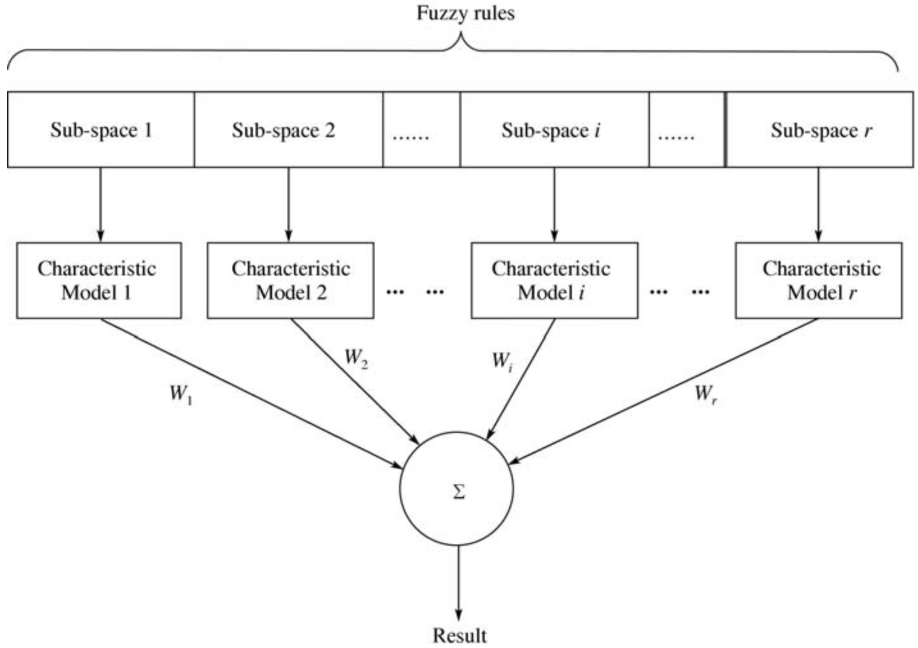

Figure 2 Fuzzy dynamic characteristic model.

Here, the whole solution space is divided into r subspaces. Then for each space, a characteristic model is established by restricting the coefficients in corresponding sub-solution-spaces. Finally, the r results are summed up by multiplying with fuzzy factors W_i that describes how close the current system is to this subspace.

The fuzzy dynamic characteristic model can be given by (see [7])

$$R = \left\{R^i\right\}_{i=1}^{r} = \left\{\begin{array}{ll} \text{IF} & \operatorname*{and}_{j=1}^{p} \ \xi_j(k) \text{ is } M_j^{i_j}, \\ \text{THEN} & \boldsymbol{y}^i(k+1) = \boldsymbol{F}_1^i(k)\boldsymbol{y}(k) + \boldsymbol{F}_2^i(k)\boldsymbol{y}(k-1) + \boldsymbol{G}^i(k)\boldsymbol{u}(k) \end{array}\right\}, \tag{19}$$

where $\xi_1(k) \sim \xi_p(k)(k \geqslant 3)$ are known premise variables (here, let z_j be the number of fuzzy divisions of $\xi_j(k)(j = 1, 2, \ldots, p)$); $M_1^{i_1} \sim M_p^{i_p}$ are the fuzzy sets (here, $i_1 \in \{1, 2, \ldots, z_1\}, \ldots, i_p \in \{1, 2, \ldots, z_p\}$); $r = \prod_{j=1}^{p} z_j$; $\boldsymbol{u} \in \mathbb{R}^m$ is the control input vector; $\boldsymbol{F}_1^i$, $\boldsymbol{F}_2^i \in \mathbb{R}^{p\times p}$, and $\boldsymbol{G}^i \in \mathbb{R}^{p\times m}$ are slowly time-varying parameter matrixes; $\boldsymbol{y}(k) \in \mathbb{R}^p$ are the state vectors.

The fuzzy factor is defined as

$$W_i = \frac{a^i(k)}{\sum_{i=1}^{r} a^i(k)},$$

where $a^i(k) = \prod_{j=1}^{p} M_j^{i_j}(\xi_j(k))$, $M_j^{i_j}(\xi_j(k))$ is the grade of membership of $\xi_j(k)$ in $M_j^{i_j}$.

After introducing the fuzzy logic into modeling, the solution space for each characteristic model is no longer as large as before. By properly partitioning, the too large solution space will finally be transformed into several subspaces that have a reasonable size. Although at some time-steps, there are models (even most of them) that cannot get a reasonable result because their subspaces do not contain the current local model. We should notice that after multiplying with fuzzy factors, the further their subspaces away from the subspace containing the local model, the less their results affect the final answer. This keeps the whole result still close to the local model. Moreover, when the system just enters a subspace, there are not enough valid samples to make the corresponding model get the correct result. At this time with the fuzzy logic, we could use some result from nearby subspace to form an approximate result.

For some large solution space that is hard to search for, the fuzzy dynamic characteristic modeling can effectively reduce the time-steps needed to get a group of coefficients precisely enough by reducing the size of searching space, which will finally make the closed-loop system more stable. Even for some solution spaces that already make the system stable, the introduction of fuzzy logic could further improve the performance by the searching space reduction.

3.4 The general fuzzy dynamic characteristic model system as universal approximators

In this section, we will use the following constructive proof approach to prove that the general fuzzy dynamic characteristic model systems are universal approximators. The key of this approach is to use characteristic model of linear constant-coefficient system as a “bridge” to connect the proof steps.

A generalized controlled object with p_c input and q_c output (c=1,2,…,r) is

$$y_c = G_c(s)u_c. \tag{20}$$

System transfer function $G_c(s)$ can be decomposed into

$$G_c(s) = \sum_{i=1}^{\gamma_1^{(c)}} \frac{l_i}{s^i} + \sum_{i=1}^{\gamma_2^{(c)}}\sum_{j=1}^{v_i^{(c)}} \frac{d_{ij}}{(s-\mu_i^{(c)})^j} + \sum_{i=1}^{\gamma_3^{(c)}}\sum_{j=1}^{\nu_i^{(c)}} \frac{p_{ij1}s+p_{ij0}}{((s-\xi_i^{(c)})^2+\eta_i^{(c)})^j}.$$

Definition 1.

$$\lambda_{\min(\max)}^{(c)} = \min(\max)\left\{\mu_{i_2}^{(c)},\ \frac{\left(\xi_{i_3}^{(c)}\right)^2+\left(\eta_{i_3}^{(c)}\right)^2}{\xi_{i_3}^{(c)}},\quad (i_j=1,2,...,\gamma_j^{(c)},\ \ j=2,3)\right\}.$$

Assumption 1. There exists a sampled data control law $u_c(t)$ such that the output of system $y_c(t)$ is a constant under steady-state condition. And $u_c(t)=u_c(mh_c)$, $\lim_{t\to\infty}u_c(t)=u_c^{(w)}$(constant), where h_c is sampling period.

By Assumption 1, $\dot{y}_c(t)$ is bounded. Let the upper bound of $|\dot{y}_c(t)|$ be M_c.

Assumption 2. The sampling period h_c satisfies condition:

$$h_c \ll \min\left\{\left|\lambda_{\min}^{(c)}\right|,\ \left|\lambda_{\max}^{(c)}\right|,\ \max_{i=1,\dots,\gamma_3^{(c)}}\left(\frac{\pi}{\eta_i^{(c)}},\ \frac{\varepsilon}{M_c}\right)\right\},$$

where ε is a sufficiently small positive number.

Lemma 1[9]. If Assumptions 1 and 2 both hold, then the characteristic model of system (20) can be expressed with the following 2-order difference equation:

$$y_i(k+2) = f_{i1}(k+1)y_i(k+1) + f_{i2}(k)y_i(k) + f_{i3}^T(k+1)u(k+1) + f_{i4}^T(k)u(k), \quad i=1,2,...,p_c.$$

The coefficients of the above equation are slowly time-varying, and the ranges of these coefficients satisfy the following conditions:

$$\begin{cases} f_{i3}(k) = -f_{i4}(k)+O(h_c^2)=O(h_c), \\ (1)\quad \text{if } \lambda_{\min}^{(c)} < \lambda_{\max}^{(c)} < 0, \qquad f_{i1}(k)\in[2+2h_c\lambda_{\min}^{(c)},\ 2],\ \ f_{i2}(k)\in[-1,\ -1-2h_c\lambda_{\min}^{(c)}], \\ (2)\quad \text{if } \lambda_{\max}^{(c)} > \lambda_{\min}^{(c)} > 0, \qquad f_{i1}(k)\in[2,\ 2+2h_c\lambda_{\max}^{(c)}],\ \ f_{i2}(k)\in[-1-2h_c\lambda_{\max}^{(c)},\ -1], \\ (3)\quad \text{otherwise}, \qquad f_{i1}(k)\in[2+2h_c\lambda_{\min}^{(c)},\ 2+2h_c\lambda_{\max}^{(c)}],\ \ f_{i2}(k)\in[-1-2h_c\lambda_{\max}^{(c)},\ -1-2h_c\lambda_{\min}^{(c)}]. \end{cases}$$

Definition 2. Fuzzy dynamic model is shown as an extension of the T-S type fuzzy model:

$$R = \left\{R^i\right\}_{i=1}^r = \left\{\text{IF} \ \operatorname*{and}_{j=1}^{p}\ \xi_j(k)\ \text{is}\ M_j^{i_j}, \qquad \text{THEN}\ \begin{Bmatrix} \dot{\boldsymbol{x}}(k) = \boldsymbol{A}_i\boldsymbol{x}(k)+\boldsymbol{B}_i\boldsymbol{u}(k) \\ \boldsymbol{y}(k)=\boldsymbol{C}_i\boldsymbol{x}(k)\end{Bmatrix}\right\}. \tag{21}$$

Thus, fuzzy dynamic characteristic model (19) is an extension of the fuzzy dynamic model.

Lemma 2 [10, 11]. If the membership functions of the inferred fuzzy set are bell-shaped membership functions (BSMF), for any given real continuous function g on the compact set $U\subset\mathbb{R}^n$ and $\varepsilon>0$, there exists a fuzzy dynamic model (21) such that

$$\sup_{x\in U}|f(x)-g(x)|<\varepsilon.$$

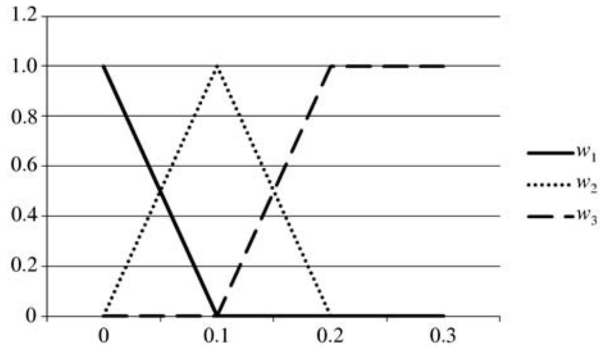

Figure 3 Fuzzy factor.

Theorem 1. If Assumptions 1 and 2 both hold and the membership functions of the inferred fuzzy set are BSMF, for any given real continuous function g on the compact set $U \subset \mathbb{R}^n$ and $\varepsilon > 0$, there exists a fuzzy dynamic characteristic model (19) such that

$$\sup_{x\in U} |f(x) - g(x)| < \varepsilon.$$

Proof. See Appendix.

Remark 1. Theorem 1 shows that any continuous function on a compact set can be approximated by the fuzzy dynamic characteristic model (19) to an arbitrary degree of accuracy.

4 Design of the hypersonic vehicle controller

For the attitude control problem discussed in this paper, we can build a characteristic model as below:

$$\alpha\,(k+1) = f_1\,(k)\,\alpha\,(k) + f_2\,(k)\,\alpha\,(k-1) + g\,(k)\,\delta E\,(k)\,. \tag{22}$$

To make the coefficients slowly time-varying we choose the sampling interval $\Delta t = 0.01$ s. Because hypersonic vehicle is just a system that changes drastically, to make the controller perform better we then split the model into three sub-models, with the same equation as (22) but each with a different restriction range. The split rule is based on $|e(k)|$. That means there will be nine coefficients to be identified, which are: $f_{11}(k), f_{12}(k), g_1(k), f_{21}(k), f_{22}(k), g_2(k), f_{31}(k), f_{32}(k), g_3(k)$.

Then we can design the controller. Here we only employ the u_g and u_i to form our controller as below:

$$u_{ig}(k) = -\frac{l_1\hat{f}_{i1}(k)e(k) + l_2\hat{f}_{i2}(k)e(k-1)}{\hat{g}_i\,(k)}, \quad i = 1, 2, 3, \tag{23}$$

$$u_g\,(k) = \sum_{i=1}^{3} W_i\,(|e\,(k)|)\,u_{ig}(k), \tag{24}$$

$$u(k) = u_g(k) + u_i(k), \tag{25}$$

where u_i is calculated by using (14) and (16). The W_i is a fuzzy factor working as shown in Figure 3.

5 Simulation results

We use Matlab simulink to build a test model for the hypersonic vehicle mentioned in (1)–(8), and load the aerodynamic coefficients from a table. The vehicle starts running from $V = 6000$ m/s, $h = 60000$ m/s, γ= 0 degree, α= 4 degrees and $q = 0$ degree/s.

The target attack angle tracked by us is designed as shown in Figure 4.

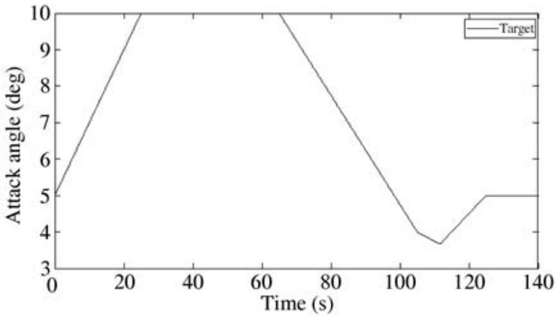

Figure 4 Control target.

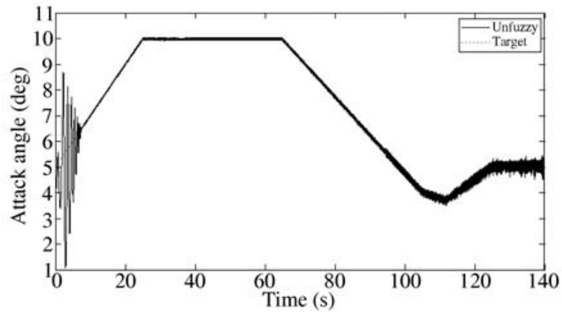

Figure 5 Single characteristic modeling.

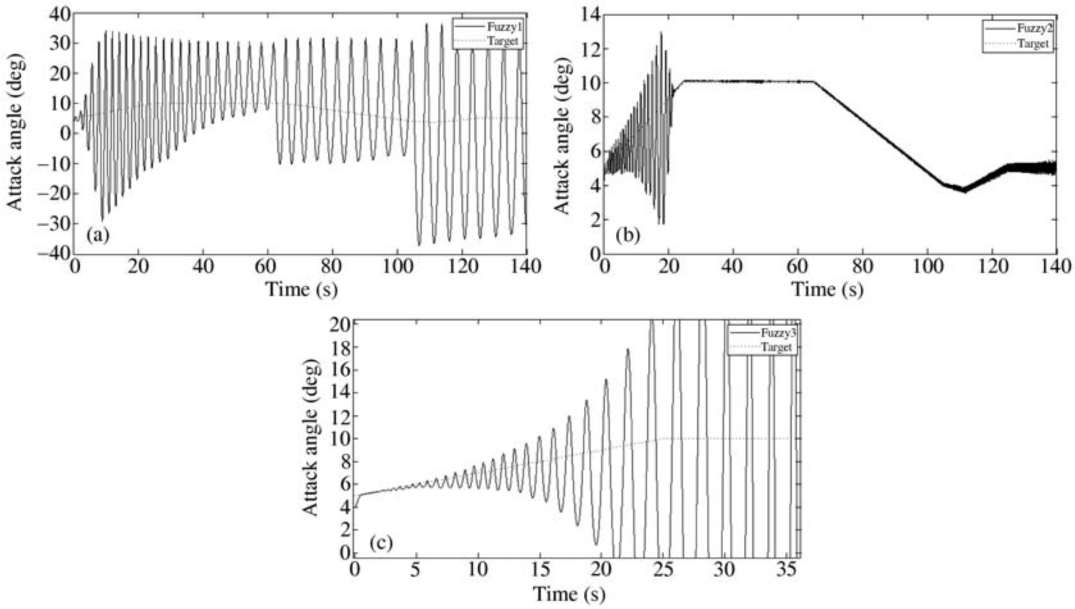

Figure 6 Sub-model using alone. (a) Sub-model Fuzzy 1 using along; (b) sub-model Fuzzy 2 using along; (c) sub-model Fuzzy 3 using along.

If we do not use the fuzzy partition but use the single standard characteristic modeling method, we just search for the result in the whole solution space. We will get an output in Figure 5.

From this figure, we could find that the standard characteristic modeling needs about ten second to trace the target track, and is not very robust while tracing. As we explained above, the reason is that the model needs many time-steps to find the "true value" of the coefficients. That makes it react to the change of flight condition a little slowly. The mean square error (MSE) of this controller is 0.1682.

Now, we use the controller proposed in section 4. The solution space is divided into three subspaces. Then, there are three sub-models. If we use each sub-model alone (i.e. fuzzy 1, fuzzy 2, and fuzzy 3), we will get output results as shown in Figure 6.

It is clear that none of them could match the characteristic modeling that searches in a whole solution space. The reason is apparent. All of them lose some "local model", so it is even not possible for them to get a certain local model no matter how many time-steps we give them. However, we give this simulation result to explain the characteristic of each sub-model.

Fuzzy 3 uses only a little time to get into the target track, because its solution space is fittest to the condition when the output is far away from the target. But after it gets close enough, because it does not have the necessary local model, it quickly becomes divergent. Fuzzy 1 is designed only under the condition that $|e(k)|$ is small enough. But, under this initial condition the local model from the beginning is lacking, thus making the corresponding curve look worst. However, if we change the initial α to 5, its output is as shown in Figure 7.

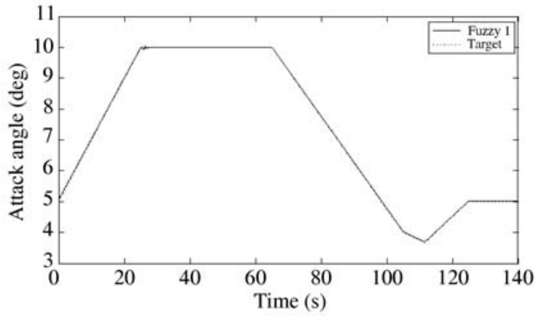

Figure 7 Fuzzy 1 using alone with α start at 5 degree.

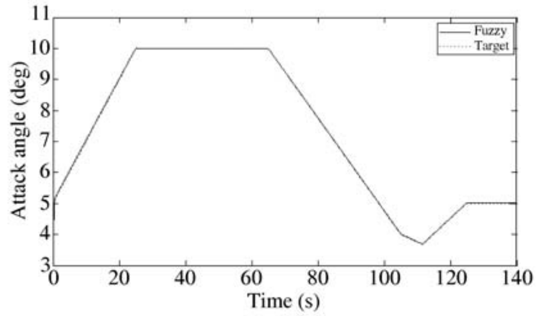

Figure 8 The fuzzy dynamic characteristic modeling.

Table 1 Simulation results of fuzzy dynamic characteristic modeling

Initial α (deg)	Convergence time (s)	MSE
2	0.6	0.0138
4	0.36	0.0018
5	0	0.0002
6	0.27	0.0012
8	0.46	0.0105
10	0.87	0.0337

Table 2 Simulation results of characteristic modeling

Initial α (deg)	Unstable convergence time (s)	MSE
2	52.3	0.1216
4	6.71	0.1682
5	0	0.0112
6	divergent	divergent
8	divergent	divergent
10	divergent	divergent

Because under this condition fuzzy 1 has all local models it needs, and it can keep tracing the target very well.

Fuzzy 2 is between fuzzy 1 and fuzzy 3. So, it can get into the target track finally but slower than fuzzy 3 and at the same time can trace the target but not as closely as fuzzy 1.

By integrating all of three sub-models designed by fuzzy dynamic characteristic modeling, we finally get a result as listed in Figure 8.

At the beginning, because the $|e(k)|$ is far larger than 0.2, fuzzy 3 dominates the whole controller, which makes the output get into the target track as fast as fuzzy 3. Then fuzzy 2 will get domination. With a new set of local models, the output can be controlled around the target like fuzzy 2 rather than diverge like fuzzy 3. When the output gets closer to the target, fuzzy 1 will finally dominate the controller and make the output trace the target stably. The MSE of fuzzy dynamic characteristic modeling is 0.0018, which is almost only 1% of that obtained by a single standard characteristic model.

Because of the solution space partition, the time-steps needed to search for the current local model are greatly reduced. Then after the sub-controllers have finally combined, the fuzzy control system as a whole is able to find all the local models in the solution space.

To further make a more complete comparison, we do a series of simulations with the initial α from 0 to 10 degrees. The results are listed in Tables 1 and 2.

In Table 1, the convergence time means the time after which $|e(k)|$ is always less than 0.2. In Table 2, the unstable convergence time means the time after which $|e(k)|$ is always less than 0.6. Except the initial α, all of other conditions are the same as the simulation before.

From the tables, we can find that after adding fuzzy logic into the characteristic modeling, the performance is greatly improved.

6 Conclusions

In this paper, we design a longitudinal attitude controller for the hypersonic vehicle by using the fuzzy dynamic characteristic modeling. By adding fuzzy logic into the classic characteristic modeling, we greatly improve the performance of the control system. With the same range of coefficients restriction, the fuzzy logic apparently enlarges the robustness region and shortens the time needed to get convergence.

Meanwhile, this method is different from classic fuzzy dynamic modeling approach which often needs to build lots of linear local models. Because the characteristic modeling is an adaptive algorithm, the fuzzy rules needed is far less in number. This character makes the design of the fuzzy controller easier than before.

Additionally, the universal approximation of general fuzzy dynamic characteristic model system is briefly analyzed and proved.

A simulation example is given to show the effectiveness of the proposed approach. Comparing it with the classic characteristic modeling method in simulation figures and tables, we can safely draw a conclusion that the method discussed in this paper is more effective than the conventional method in dealing with attitude control problem of hypersonic vehicle.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61074066, 90716021, 60736023, 61004021), the National Laboratory of Space Intelligent Control and Open Foundation (Grant No. SIC07010202), and the Beijing Key Discipline Development Program (Grant No. XK100080537).

References

1 NASA. News about the space craft X-37. At: Nasa official web site, http://www.msfc.nasa.gov/news/x37news/index.html
2 Liu Y B. Research on modeling and advanced flight control theories for hypersonic vehicle (in Chinese). Ph.D. thesis, Nanjing: Nanjing University of Aeronautics and Astronautics, 2007
3 Lu P, Hanson J M. Entry guidance for the X-33 vehicle. J Spacecraft Rockets, 1998, 35: 342–349
4 Wu H X, Liu Y W, Liu Z H, et al. Characteristic modeling and the control of flexible structure. Sci China Ser F-Inf Sci, 2001, 44: 278–291
5 Yang J C, Hu J, Ni M L. Adaptive guidance law design based on characteristic model for reentry vehicles. Sci China Ser F-Inf Sci, 2008, 51: 2005–2021
6 Meng B, Wu H X, Lin Z L, et al. Characteristic model based control of the X-34 reusable launch vehicle in its climbing phase. Sci China Ser F-Inf Sci, 2009, 52: 2216–2225
7 Luo X, Sun Z Q, Sun F C. A new approach to fuzzy modeling and control for nonlinear dynamic systems: neuro-fuzzy dynamic characteristic modeling and adaptive control mechanism. Int J Control Autom Syst, 2009, 7: 123–132
8 Wang Q, Stengel R F. Robust nonlinear control of a hypersonic aircraft. J Guid Control Dyn, 2000, 23: 577–585
9 Meng B, Wu H X. The proof for characteristic model of linear constant-coefficient system (in Chinese). Sci China Ser E-Inf Sci, 2007, 37: 1258-1271
10 Wang L X, Mendel J M. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. IEEE Trans Neural Networks, 1992, NN-3: 807–814
11 Cao S G, Rees N W, Feng G. Analysis and design for a class of complex control systems (Part I: fuzzy modelling and identification). Automatica, 1997, 33: 1017–1028

Appendix

Proof of Theorem 1. From the default preconditions of this theorem and Lemma 2, any continuous function g on a compact set can be approximated by the fuzzy dynamic model f (as shown in (21)) to an arbitrary degree of accuracy.

For the local dynamic model appeared in the consequent of fuzzy rule (21):

$$\begin{cases} \dot{\boldsymbol{x}}(k) = \boldsymbol{A}_i \boldsymbol{x}(k) + \boldsymbol{B}_i \boldsymbol{u}(k), \\ \boldsymbol{y}(k) = \boldsymbol{C}_i \boldsymbol{x}(k), \end{cases} \tag{A1}$$

it can be equivalently transformed into the corresponding I/O model by selecting the specific values p_c and q_c. Here, the system transfer function can be expressed as

$$G_i(s) = \mathrm{C}_i (s\boldsymbol{I} - \boldsymbol{A}_i)^{-1} \boldsymbol{B}_i.$$

From Assumptions 1 and 2 and Lemma 1, the above I/O model can be expressed with the following characteristic model:

$$y_i(k+2) = f_{i1}(k+1) y_i(k+1) + f_{i2}(k) y_i(k) + f_{i3}^{\mathrm{T}}(k+1) u(k+1) + f_{i4}^{\mathrm{T}}(k) u(k).$$

That is to say, the local dynamic model (A1) can be expressed as the corresponding characteristic model.

Moreover, if the controlled plant is a minimum-phase system, or a weak non-minimum-phase system, in order to make simplification in engineering, only one item $u(k+1)$ is preserved. Then, by means of a simple substitution, we can get characteristic model equation shown in (19).

That is to say, any continuous function g on a compact set can be approximated by the fuzzy dynamic model, which provides a high performance framework for modeling by decomposition of a complex nonlinear system into a collection of local characteristic sub-models, to an arbitrary degree of accuracy.