A NOLINEAR MODEL PREDICTIVE CONTROL METHOD FOR AIRBREATHING HYPERSONIC VEHICLE BASED ON KOOPMAN OPERATOR

Chaoran Li[1], Weiqiang Li[1], Yi Li[1], Shuo Tang[1*]

[1]1 School of Astronautics NPU, Northwestern Polytechnic University, Xian, China

Abstract

In this paper, a model predictive control method for Airbreathing Hypersonic Vehicles based on Koopman operator theory is proposed. Simulation experiments are carried out with the First Principal model as the controlled object, and it is compared with the two other Koopman-MPC based on DMDc and EDMD to demonstrate the effectiveness of the method.

Keywords: model predictive control, koopman operator, airbreathing hypersonic vehicles

Introduction

In recent years, the development of material technology makes it possible to adopt a more radical aerodynamic layout and structural design scheme on a new generation of hypersonic aircraft, which will predictably improve the flight performance of the aircraft, but also reduce the natural frequency of the elastic mode of the aircraft, and the elastic phenomenon becomes more significant. This change means that due to the intensification of the aeroelastic coupling phenomenon, the nonlinear and uncertainty of the aerodynamic model of the aircraft will be further increased. Especially for the air-breathing hypersonic vehicle, not only the aerodynamic force, but also the thrust depends on the rigid body and elastic state of the vehicle. Because the thrust performance of the airbody-engine integrated ramjet is related to the pressure at the air intake, and it is affected by the shape of the lower surface of the aircraft precursor. This coupling characteristics of aerodynamics, thrust and elasticity make it very difficult to design the controller of the elastic aspirated hypersonic vehicle.

In order to solve this problem, many scholars have proposed controller design methods based on different control methods, such as reverse step control, synovial control, etc. However, most of these methods are model-based control methods, whose controller performance depends on the accuracy of the model, and it is difficult to achieve the expected control effect in the face of the problem of parameter uncertainty and model uncertainty.

In recent years, with the development of big data technology, data-driven control methods have begun to enter the field of vision of researchers. The controller design of data-driven control method does not rely entirely on the mathematical model information of the controlled object. Instead, it relies on the data of the controlled system and the implicit knowledge obtained by processing the data offline or online. As an effective data-driven control method, data-driven Model Predictive Control (MPC) has been widely used in aerospace field.

In this paper, the data-driven MPC method is introduced into the control problem of the elastic air-breathing hypersonic vehicle, and the Koopman Neural Network (KNN) is used to fit the linear model of the vehicle and design the controller. In Chapter 1, the mathematical model for controller design and verification of the elastic aspirated hypersonic vehicle is introduced, which includes two parts: dynamic model and static model. In the second chapter, we briefly introduce the Koopman operator theory and two traditional numerical approximation methods of Koopman operator, and propose a neural network form for approximating Koopman operator. In Chapter 3, three Koopman operator numerical approximation methods described in Chapter 2 are used to gain the linear model of the aircraft (based on different observation spaces), and the controller of Koopman model predictive control (K-MPC) is designed. Finally, in Chapter 4, this paper carries out simulation verification, and compares the MPC control performance based on the above three methods, showing the feasibility

and advancement of K-MPC based on KNN.

1. Mathematics Model

So far, a large number of scholars have conducted researches on the dynamics modeling of hypersonic vehicles, among which a series of mature models have been born, such as the Winged-cone model proposed by NASA Langley Research Center[1] and the CSULA-GHV model proposed by California State University, Los Angeles[2]. Considering the requirement of controller design and verification for air-breathing elastic hypersonic vehicle, the First Principal model of Air Force Laboratory is adopted in this paper, which regards the vehicle as a beam for analysis and has aerodynamic-thrust-elastic coupling characteristics.

1.1 First-Principal dynamic model

Considering that the content of this paper is limited to the pitch channel of the aircraft and the vibration mode is assumed to be orthogonal mode, the inertial coupling term in the elastic vibration equation is ignored and the First Principal dynamic model is simplified. The simplified dynamic equation is as follows:

$$\begin{aligned}
\dot{V} &= \frac{T\cos\alpha - D}{m} - g\sin\theta \\
\dot{\theta} &= \frac{L + T\sin\alpha}{mV} - \frac{g\cos\theta}{V} \\
\dot{h} &= V\sin\theta \\
\ddot{\vartheta} &= \frac{M_{yy} + \psi_i \ddot{\eta}_i}{I_{yy}} \\
\ddot{\eta}_i &= -2\zeta_i \omega_i \dot{\eta}_i - \omega_i^2 \eta_i + N_i
\end{aligned} \tag{1.1}$$

Where η_i is the generalized coordinates of the *i* order elastic vibration of the aircraft. And the elastic motion of the aircraft is simplified by the assumed mode method, which regards the vibration of the aircraft in a certain plane as the superposition of a series of vibration modes:

$$y(x) = \sum_{i=1}^{N} \varphi_i(x)\eta_i(t) \tag{1.2}$$

Where $y(x)$ is the distribution function of the displacement in the y direction along x axis and ϕ_i is the mode function of the *i* order vibration mode. In this paper, the assumed mode functions are selected as follows:

$$\varphi_i(x) = \cos(i\pi \frac{x}{L}) \tag{1.3}$$

Where L is the longitude of the aircraft. In this paper, $i \in [1,3]$. And the inherent frequency ω_i of each mode and the structural damping e are listed in table 1-1.

table 1-1 inherent frequency of mode 1~3

i	1	2	3
inherent frequency ω	218.74	281.65	384.22
damping ratio ζ	0.1	0.1	0.1

In the First Principal model, the impact of elasticity on aircraft can be seen as reflected in two aspects, namely, its direct impact and indirect impact on aircraft. The direct influence, that is, the influence of the elastic state of the aircraft on its thrust and aerodynamic force, is the main concern of this paper, which will be introduced in Section 1.2. The indirect effect is the effect of the elastic state on the aircraft sensor, which can be described as:

$$\begin{aligned}\vartheta_s &= \vartheta + \sum_{i=-2}^{4} \varphi_i'(x_s)\eta_i \\ \dot{\vartheta}_s &= \dot{\vartheta} + \sum_{i=-2}^{4} \varphi_i'(x_s)\dot{\eta}_i\end{aligned} \tag{1.4}$$

Where $\vartheta_s, \dot{\vartheta}_s$ are the pitch angle signal and pitch angle velocity signal output by the sensor. It acts indirectly on the rigid motion of the aircraft through the controller, which often makes it difficult to design the controller of the elastic aircraft. But up to now, a large number of studies have proposed more mature suppression methods using notch filters. Therefore, this part is not the main focus of this article, although it will still be considered in the controller design below.

1.2 Mechanical model of aircraft

Since the 1990s, many researches have tried to establish aerodynamic models of air-breathing hypersonic vehicles.By CFD simulation, a mathematical model of air vehicle force with air-push coupling in the form of discrete data table has been presented[3]. Further, Mirmirani et al.[2] fitted the mathematical expression of force and torque coefficients based on CFD simulation data, in which the engine thrust is coupled with the rigid body state of the aircraft:

$$\begin{aligned}C_T &= 0.0069 - 0.0438\alpha - 0.1409\phi + 0.0046\delta_e - 0.0016M \\ &\quad -0.2107\alpha\phi - 0.0266\alpha\delta_e + 0.0057\alpha M + 0.0111\phi M \\ &\quad -0.0001\delta_e M + 0.1287\alpha^2 + 0.0001M^2\end{aligned} \tag{1.5}$$

Where $\alpha, \delta_e, \phi, M$ are respectively aircraft attack Angle, elevator deflection Angle, throttle opening and Mach number. This expression only takes the coupling of thrust with the rigid state of the vehicle into consideration. In this paper, the coupling of thrust and elastic state is reflected by the accessional attack angle in the following form:

$$\begin{aligned}\alpha &= \alpha_0 + \Delta\alpha, \\ \Delta\alpha &= \sum_{i=-1}^{3} \varphi_i'(x_f)\eta_i(t)\end{aligned} \tag{1.6}$$

Similarly, the lift coefficient, drag coefficient, and pitching moment coefficient are expressed as follows:

$$\begin{aligned}C_L &= -0.0236 + 0.9661\alpha + 0.2494\phi - 0.240\delta_e + 0.0061M \\ &\quad +0.5989\alpha\phi + 0.0656\alpha\delta_e - 0.0486\alpha M - 0.0164\phi M \\ &\quad -0.0012\delta_e M + 0.1954\alpha^2 - 0.0003M^2\end{aligned} \tag{1.7}$$

$$\begin{aligned}C_D &= 0.0039 + 0.0314\alpha - 0.0478\phi + 0.0022\delta_e - 0.0002M \\ &\quad -0.0210\alpha\phi - 0.05\alpha\delta_e + 0.0006\alpha M + 0.0039\phi M \\ &\quad -0.001\delta_e M + 0.6802\alpha^2\end{aligned} \tag{1.8}$$

$$\begin{aligned}C_{M_y} &= 0.0273 - 0.233\alpha - 0.137\phi + 0.04\delta_e - 0.0054M \\ &\quad -0.2505\alpha\phi - 0.0538\alpha\delta_e + 0.0182\alpha M + 0.0098\phi M \\ &\quad -0.0016\delta_e M + 0.2057\alpha^2 + 0.0003M^2\end{aligned} \tag{1.9}$$

The aerodynamic coefficient polynomial includes 12 coefficients with the highest degree is 2, and there is cross-coupling between the Angle of attack, the elevator deviation and the throttle opening term, indicating that the aerodynamic-thrust-elastic coupling of the model is serious, which brings great difficulties to the design of the controller.Koopman Operator theory for control

1.3 Introduction of Koopman Operator theory

Given a general discrete dynamic system:

$$x_{t+1} = F(x_t) \tag{2.1}$$

Where $x_t \in \mathrm{M}$ is the state vector of the system at time *t*, M is the state space of the system. koopman operator theory holds that there is an infinite dimensional space G and the relevant transformation $g : \mathrm{M} \to \mathrm{R}$, which make the linear operator $\mathcal{K}$ satisfying the following formula exists：

$$\mathcal{K}g = g \circ F \tag{2.2}$$

Where $\mathcal{K}$: $\mathrm{G} \to \mathrm{G}$ is the Koopman Operator. Defining $z_t = g(x_t)$, the lifted system can be expressed as:

$$z_{k+1} = g(F(x_t)) = g \circ F(x_t) = \mathcal{K}g(x_t) = \mathcal{K}z_k \tag{2.3}$$

Videlicet, the lifted system is a linear system in space G. This method realizes the global linearization of the system (2.1). However, $\mathcal{K}$ is an infinite dimensional operator, which cannot be calculated and utilized in practical engineering. Therefore, finite-dimensional approximations of Koopman operators need to be carried out by various methods to obtain the linear model of the system which is engineering available.

1.4 Numerical approximation of Koopman Operator

In this section, three approaches to the numerical approximation of the Koopman operator are introduced, the first two of which, Dynamic Mode Decomposition with control (DMDc) and Extended Dynamic Mode Decomposition (EDMD), are traditional numerical methods, and the third requires a neural network. However, all three methods allow online calculation of the Koopman operator, even though the third method has two different update periods, namely a short operator calculation period and a relatively long neural network training period (minute level), it is still enough to support real-time updates during flight of the aircraft.

1.4.1 Gain Koopman Operator via DMDc

Given a discrete linear system with control input shown as:

$$x_{k+1} = Ax_k + Bu_k \tag{2.4}$$

where $x \in \mathrm{R}^n, u \in \mathrm{R}^m, A \in \mathrm{R}^{n\times n}, B \in \mathrm{R}^{n\times m}$, and data snapshots $U_k = \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix}$, $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_{k+1} \end{bmatrix}$, which can be decomposed to:

$$\begin{cases} X_k = \begin{bmatrix} x_1 & x_2 & \cdots & x_k \end{bmatrix} \in \mathrm{R}^{n\times k} \\ X_{k+1} = \begin{bmatrix} x_2 & x_3 & \cdots & x_{k+1} \end{bmatrix} \in \mathrm{R}^{n\times k} \end{cases} \tag{2.5}$$

Hence, (2.4) can be expressed as:

$$X_{k+1} = \hat{A}\hat{X}_k \tag{2.6}$$

Where $\hat{A} = \begin{bmatrix} A & B \end{bmatrix}, \hat{X}_k = \begin{bmatrix} X_k \\ U_k \end{bmatrix}$. It seems that matrix $\hat{A}$ can be calculated easily by least square:

$$\min_{\hat{A}\in\mathrm{R}^{n\times(m+n)}} \left\| X_{k+1} - \hat{A}\hat{X}_k \right\|_2^2 \tag{2.7}$$

However, in practical the values of *k* and *n* are generally large, which makes the solution of the pseudo-inverse of $\hat{X}_k$ difficult to achieve. In the DMD method, the singular value decomposition of X needs to be performed and the low-order truncation $\hat{X}_k$ is taken, so that:

$$\tilde{\hat{X}}_k = U\Sigma V^T \approx \hat{X}_k \tag{2.8}$$

Then *A* can be reduced to obtain its lower order approximation:

$$\tilde{A} = YV\Sigma^{-1}U^T \tag{2.9}$$

It is worth mentioning that $\tilde{A}$ in the other Koopman Operator numerical approximation methods mentioned below are all finally calculated by (2.8) and (2.9).

1.4.2 Gain Koopman Operator via EDMD

Given the nonlinear system as (2-2), in order to gain the high dimension observation space in which Koopman operator works, a basis function group Ψ is defined:

$$\Psi(x(k),u(k)) = \begin{bmatrix} \psi_1, & \psi_2, & \cdots & \psi_N \end{bmatrix} : \mathrm{R}^{n+m} \to \mathrm{R}^N \tag{2.10}$$

which makes:

$$\Psi(x(k+1),u(k+1)) = \mathcal{K}\Psi(x(k),u(k)) \tag{2.11}$$

By least square, EDMD can find matrix $K \in \mathrm{R}^{\mathrm{N}\times\mathrm{N}}$ as a numerical approximate of $\mathcal{K}$:

$$\min_{K\in\mathrm{R}^{N\times N}} \sum_{k=1}^{P-1} \left\| \Psi(x(k+1),u(k+1)) - K\Psi(x(k),u(k)) \right\|_2^2 \tag{2.12}$$

Where P is the length of the data snapshot.

However, there is no data-based approach to the basis function group Ψ in EDMD, instead, the basis function group is assumed or selected though experience. In general, the basis function group $\Psi(x(k),u(k))$ can be decompose to $\left[\Psi_x^{\mathrm{T}}(x(k)),\Psi_u^{\mathrm{T}}(u(k))\right]^{\mathrm{T}}$, where $\Psi_u(u(k))=u(k)$. Hence, (2.12) can be expressed as:

$$\min_{K\in \mathrm{R}^{N\times N}} \sum_{k=1}^{P-1} \left\| \Psi_x(x(k+1)) - A\Psi_x(x(k)) - Bu(k) \right\|_2^2 \tag{2.13}$$

And in practical, a common constructor of EDMD divides Ψ_x into two parts:

1.4.3 Gain Koopman Operator via machine learning

Although some data-driven numerical approximation methods of Koopman Operator have been mentioned above, it has remained a challenge to obtain representations of Koopman operator with high precision and representability that can characterize as many dynamic features of the system as possible to achieve global linearization. Even for relatively simple dynamical systems, the Koopman operator of these systems may be arbitrarily complex and will only be approximately represented in a finite basis. Machine learning is well-suited for representing such arbitrarily complex functions, and has recently shown tremendous promise for discovering and representing Koopman embeddings and Koopman forecasts[4]

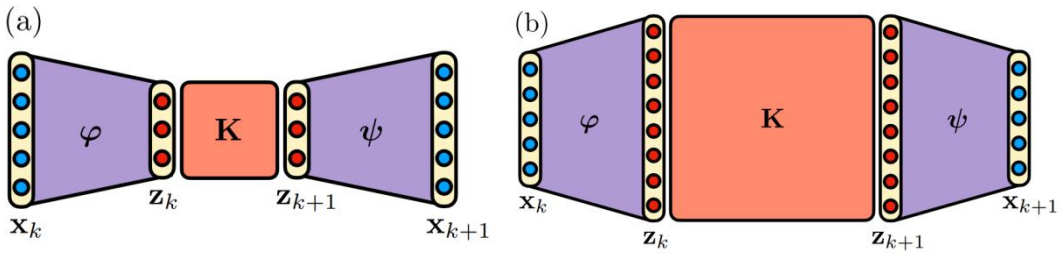

Figure 2.1 Two neural network architectures to approximate the Koopman Operator[5]

There are two basic neural network architectures that have been proposed for Koopman numerical approximation[5], shown in Figure 2.1. In the first architecture, the auto-encoder architecture extracts a few key latent variables z = ϕ(x) to parameterize the dynamics. And in the second one, the input state vectors are lifted to an higher dimension, where the evolution is approximately linear.
However, both types of networks face two practical problems. The first is the loss of dynamic system characteristic information caused by encoder network and anti-encoder network, and the second is the extra computing power and time cost required by the iterative training of encoder and anti-encoder network. With this in mind, it is not always worthwhile to use KNN as an alternative to the EDMD method. When there is sufficient prior knowledge about the controlled system, it is easy to obtain better results by using EDMD, while KNN may not be worth the loss.

1.5 Koopman neural network

In this paper, a Koopman neural network for dynamic systems with relatively low dimensional state space and high dimensional observation space is proposed based on the idea of extending the state space with EDMD and introducing nonlinear observation function to form the observation space. In this structure, the autodecoder network structure (shown in Figure 2.1) of the neural network described above is omitted, and the transformation from the observation space to the original state space is directly given in the form of a matrix.

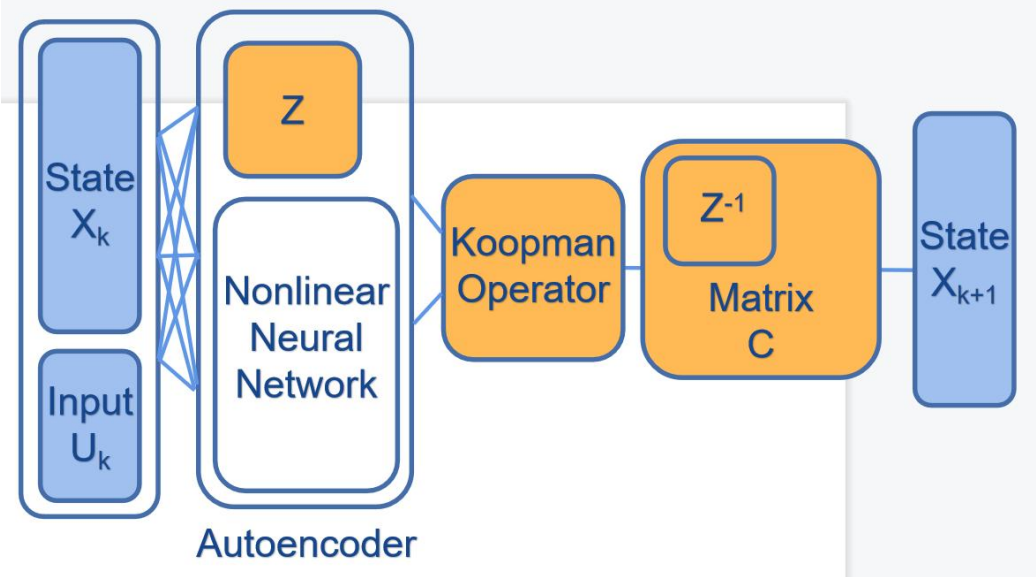

Figure 2.2 The architecture of Koopman Neural Network (KNN)

1.5.1 The architecture of Koopman neural network

The Koopman neural network structure proposed in this paper is shown in Figure 2.2. the autoencoder network completes the fitting of the dimensional-raising function:

$$z_k = \varphi(x_k) \tag{2.14}$$

Which makes $z_{k+1} = Kz_k$ exists. Where $z(x) \in \mathrm{R}^N$ is the observation function, which consists of two parts:

$$z(x_k) = \left[(Zx_k)^T, \Psi^T(x_k) \right]^T \tag{2.15}$$

Where $Z \in \mathrm{R}^{n\times n}$, $\Psi(x) \in \mathrm{R}^{N-n}$ and n is the dimension of state vector x. The matrix Z is nonsingular and is fitted through a fully connected layer using a linear activation function, and the lifting function $\Psi : \mathrm{R}^n \to \mathrm{R}^{N-n}$ is fitted by a fully connected neural networks using nonlinear activation functions. In the training of neural networks, the state vector z, which is restricted by the loss function, runs linearly in the observation space. And this loss function takes the form of the Koopman operator:

$$\mathcal{L}_{\text{pred}} = \frac{1}{k} \left\| Z_{k+1} - KZ_k \right\|_{\text{MSE}} \tag{2.16}$$

Where $Z_k = \left[z(x_1), z(x_2), \cdots, z(x_k) \right]$ is the lifted data snapshot and $K \in \mathrm{R}^{N\times N}$ is the finite-dimensional approximation of Koopman Operator, which is calculated though (2.8) and (2.9). This loss function represents the difference between the observed vector at this time and the predicted one at the next step, and can reflect the linearity of the lifted system. Furthermore, considering the high sampling frequency of the data snapshot, the actual one-step fitting error of the model may be overwhelmed by the numerical error generated by the calculation of k during the calculation of the loss function. Therefore, in order to enhance the accuracy of network training, the loss function is selected to change from a single-step form to a multi-step form, that is:

$$\mathcal{L}_{\text{pred}} = \frac{1}{k-m} \sum_{n=1}^{k-m} \left\| z(x_{n+m}) - K^m z(x_n) \right\|_{\text{MSE}} \tag{2.17}$$

Where m is the number of predicted steps. The larger the m value, the more the network tends to compare slices of data that are further apart in time, which is beneficial to extract the long period dynamic characteristics of the system. But in practice, a large value of parameter m will cause the computational complexity of the loss function to increase, and then slow down the training speed of the Koopman neural network.

1.5.2 Koopman neural network with control

The neural network mentioned above is for the unforced dynamical system, and does not introduce the function of the control quantity u. It is not difficult to extend this network structure to forced dynamical systems. As in the form of (2.13), the form of loss function is simply extend to:

$$\mathcal{L}_{\text{pred}} = \frac{1}{k-m}\sum_{n=1}^{k-m}\left\| z(x_{n+m}) - \left(A^m z(x_n) + \sum_{i=0}^{m-1} A^i B u_{m-i}\right)\right\|_{\text{MSE}} \tag{2.18}$$

Where $\begin{bmatrix} A & B \end{bmatrix} = Z_{k+1}\begin{bmatrix} Z_k \\ U_k \end{bmatrix}^{\dagger}$. The training of the network itself is not difficult, the difficulty lies in how to select the data snapshot Z, U so that the network can get more dynamic system characteristic information in the training process. Because the dynamic system fitted by the neural network is a finite dimensional truncation of the infinite dimensional system corresponding to the original system, and this truncation is not unique. If the data used in the training of the neural network is insufficient, it may fail to capture the main dynamic features (or one of them) of the corresponding system, resulting in poor generalization performance of the trained observation function φ and failure to achieve the goal of global linearization.

In this paper, in order to ensure that the data snapshot contains as much information of the forced dynamic system as possible, the control input sequence U which is used for generating data slices is superimposed on the original control sequence with the swept-frequency cosine (chirp) signal:

$$u_k = u_{k0} + u_{\text{chirp0}} e^{2\pi\left(f_0 t + \frac{1}{2}u_0 t^2\right)} \tag{2.19}$$

However, this does not mean that the calculation of the Koopman operator needs to use such a complex data snapshot. Shorter slices can be used to achieve similar results to other methods, because part of the dynamical system information is already stored in the autoencoder network, and there is no need for the data snapshot used to compute the operator to contain all the dynamical characteristics of the system. For the detailed data snapshot generation of aircraft and the numerical approximation of Koopman Operator, this paper will be introduced in detail in Section 3.1.

2. Controller Design

2.1 Model Predictive Controller based on Koopman Operator

For linear MPC, it is the key problem that how to obtain the linear model for designing the controller. However, in MPC based on Koopman Operator, this problem is not changed, and it is transformed into two problems, namely, how to obtain data snapshots for approximating operators, and how to perform numerical approximations to operators through data snapshots. For the second question, this paper has been elaborated in the second chapter, and the first question is the main content of the first section of this chapter. The state vector of the control object can be expressed as $x = \begin{bmatrix} x_r^T & x_q^T \end{bmatrix}^T$,where x_r, x_q are the rigid body state vector and the elastic state vector of aircraft respectively:

$$\begin{aligned} x_r &= \left[v, \theta, h, \vartheta, \dot{\vartheta}\right] \\ x_q &= \left[\eta_1, \dot{\eta}_1, \eta_2, \dot{\eta}_2, \eta_3, \dot{\eta}_3\right] \end{aligned} \tag{3.1}$$

Where $v, \theta, h, \vartheta, \eta_i$ are the velocity, ballistic inclination, altitude, pitch angle and the generalized coordinates of the *i* order elastic vibration of the aircraft. And $u = \left[\vartheta, \phi\right]^T$ is the control vector of the aircraft. Hence, substituting in (2.6), the data snapshot can be expressed as:

$$\hat{X}_{k+1} = \begin{bmatrix} X_r \\ X_q \\ U \end{bmatrix} = \begin{bmatrix} x_{r,1} & x_{r,2} & \cdots & x_{r,k} & x_{r,k+1} \\ x_{q,1} & x_{q,2} & \cdots & x_{q,k} & x_{q,k+1} \\ u_1 & u_2 & \cdots & u_k & 0 \end{bmatrix} \tag{3.2}$$

The data slice is generated by simulation, the simulation length is 20s, the sampling frequency is 200Hz, and the control frequency is 50Hz. The simulated control output sequence is given by the elevator deviation data output by the attitude PID controller and the sinusoidal throttle opening.

It is worth mentioning that, considering the influence of elastic vibration on the sensor as (1.4), all the controllers in this paper have the sliding window filtering of the pitch angle velocity signal. This point will not be further discussed in the controller design and simulation verification.

In addition, the training of KNN also requires an additional and longer data snapshot $\hat{X}_{KNN}$ that can represent as many dynamic features of the system as possible, and its structure and generation

method are exactly the same as $\hat{X}_{k+1}$. In this paper, the length of $\hat{X}_{KNN}$ is 30s, and the control output form is shown in (2.19).

After the data snapshot is obtained, three different low-dimensional approximate $\tilde{K}_{DMDc}$, $\tilde{K}_{EDMD}$ and $\tilde{K}_{KNN}$ of the Koopman operator is obtained by (2.7), (2.13) and (2.18) respectively, and $\tilde{K} = \begin{bmatrix} \tilde{A} & \tilde{B} \end{bmatrix}$.

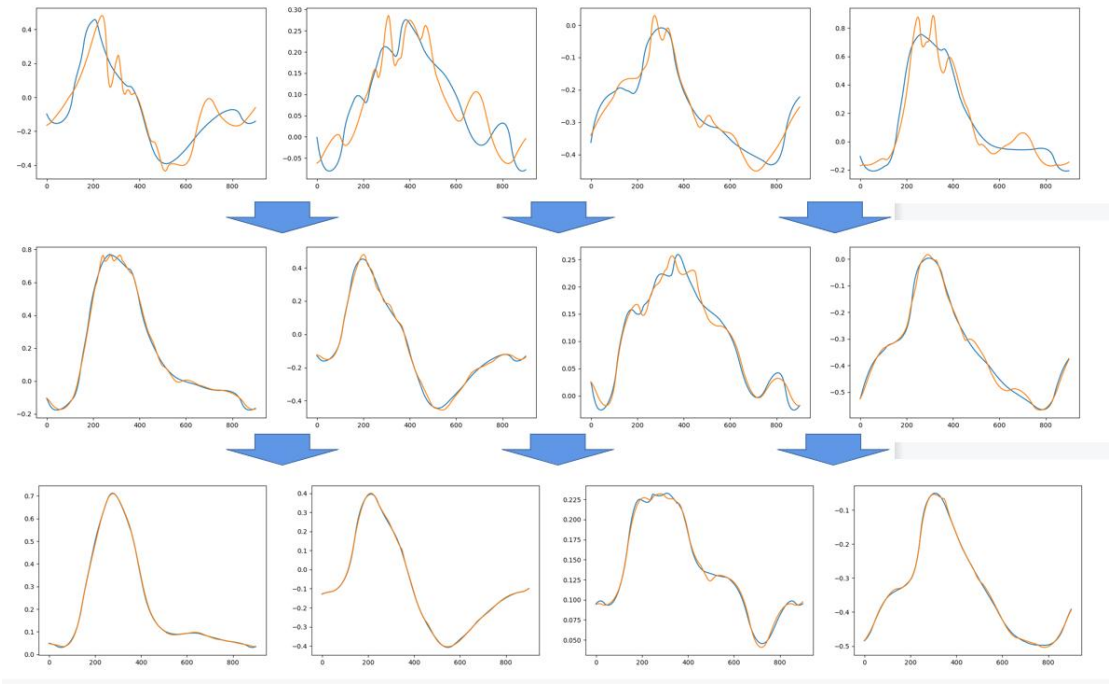

Figure 3.1 The change of network prediction curve fitting with the number of training

In this process, the fitting of each observation in the training process of KNN are shown in Figure 3.1. So far, the linear model of the aircraft required for linear model predictive controller design has been obtained:

$$\begin{aligned} x_{DMDc,k+1} &= \tilde{A}_{DMDc} x_k + \tilde{B}_{DMDc} u_k \\ x_{EDMD,k+1} &= \tilde{A}_{EDMD} x_k + \tilde{B}_{EDMD} u_k \\ x_{KNN,k+1} &= \tilde{A}_{KNN} x_k + \tilde{B}_{KNN} u_k \end{aligned} \tag{3.3}$$

The subsequent controller design will be completed according to these three linear models.

2.2 Aircraft K-MPC controller design

For the data-driven linear dynamics model of air-breathing hypersonic vehicle(3.3) obtained above, assuming that the output of MPC controller at time k is $u_{mpc}(k)$, and substituting it into the system model, the following forced dynamic model can be obtained:

$$\begin{aligned} z(k+1) &= Az(k) + Bu_{mpc}(k) \\ x(k) &= Cz(k) \\ y(k) &= Dz(k) \end{aligned} \tag{3.4}$$

Define N_p, N_c as prediction time domain and control time domain respectively, and $z(k+n|k)$ is the observation state in *k+i* time predicted from the observation state in time *k* and the input sequence from *k* to *k+i*. Hence, the prediction function of the high dimensional model in observation space can be expressed as:

$$\begin{aligned} z(k+1|k) &= Az(k) + Bu_{mpc}(k) \\ z(k+2|k) &= Az(k+1|k) + Bu_{mpc}(k+1) \\ &\vdots \\ z(k+N_c|k) &= Az(k+N_c-1|k) + Bu_{mpc}(k+N_c-1) \\ &\vdots \\ z(k+N_p|k) &= Az(k+N_c-1|k) + Bu_{mpc}(k+N_c-1) \end{aligned} \tag{3.5}$$

Substituted into (3.4), the prediction function can be expressed in the following compact form:

$$Y(k) = Fz(k) + \Phi U(k) \tag{3.6}$$

Where vectors and matrices defined as follow:

$$\begin{aligned} Y(k) &= \begin{bmatrix} y(k|k) & y(k+1|k) & \cdots & y(k+N_p|k) \end{bmatrix} \\ U(k) &= \begin{bmatrix} u_{mpc}(k) & u_{mpc}(k+1) & \cdots & u_{mpc}(k+N_c-1) \end{bmatrix} \\ F &= \begin{bmatrix} DA & DA^2 & \cdots & DA^{N_p} \end{bmatrix}^T \\ \Phi &= \begin{bmatrix} DB & & & \\ DAB & DB & & \\ \vdots & \vdots & \ddots & \\ DA^{N_p-1}B & DA^{N_p-2}B & \cdots & DA^{N_p-N_c}B \end{bmatrix} \end{aligned} \tag{3.7}$$

In order to achieve smooth tracking of the altitude and velocity commands of the aircraft, the reference trajectory $Y_d(k) \in \mathbb{R}^{q \times N_p}$ is defined as:

$$Y_d(k+i|k) = \alpha^i Y(k) + (1-\alpha^i) Y_{t\,\mathrm{arg}et}(k+i) \tag{3.8}$$

And the performance index function is selected as:

$$J = (Y - Y_d)^T Q (Y - Y_d) + U(k)^T R U(k) \tag{3.9}$$

Where Q, R are the control gain matrix. Substituted into (3.6), (3.9) can be expressed as:

$$J = H^T U(k) + \frac{1}{2} U(k)^T E U(k) \tag{3.10}$$

Where $H = \Phi^T Q(Fz(k) - Y_d(k)), E = \Phi^T Q \Phi + R$. And in order to ensure the closed-loop stability of the system, terminal constraints must be met:

$$y(k+N_p|k) = y_d(k+N_p) \tag{3.11}$$

In this case, the problem is transformed to solving optimization problems with terminal constraints. After the solution is completed, the first control vector in the optimal control sequence is taken as the controller output at this time:

$$u_{mpc}(k) = G(x(k|k)) \tag{3.12}$$

Where G is a function mapping, which represents solving the minimization problem defined by (3.10) and (3.11).

3. Simulation

3.1 Control capability verification of the controller

In the simulation, the dynamic model of the elastic aspirated hypersonic vehicle given in the first chapter is used, in which the parameter assignment and vibration mode function selection are consistent with the first chapter. Now consider the altitude and speed command control targets given step function forms:

$$\begin{aligned} h_{aim,step} &= 500 \\ v_{aim,step} &= \begin{cases} 4000, t = 0s \\ 4250, t > 0s \end{cases} \end{aligned} \tag{3.13}$$

According to this, reference trajectories $Y_{d,step}, Y_{d,\sin}$ are generated and simulated respectively. Simulation parameters, such as control frequency f_c and sampling frequency f_s, are consistent with those used in Section 3.1 to generate data snapshots. The controller parameters are set to $N_p = N_c = 10$ with the aircraft initial state $x_0 = [4000, 0, 0, 0, 0, x_{q0}^T]^T$, where the Initial elastic state $x_{q0} = [0, \cdots, 0]^T, x_{q0} \in \mathbb{R}^6$. The linear model used by the controller is obtained from the DMDc method.

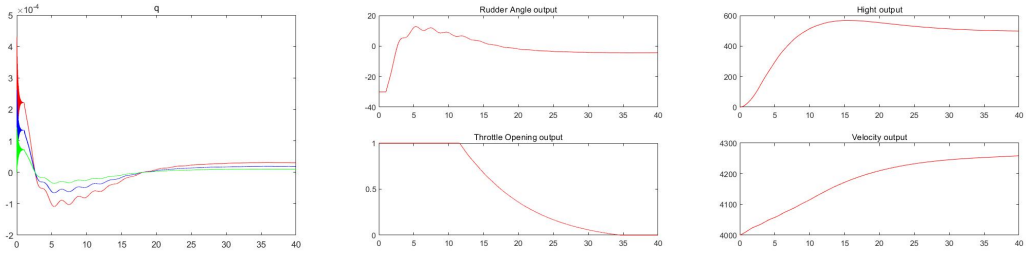

Figure 4.1 Simulation result.
(a)Elastic mode generalized coordinates (b)Control input (c)System output

The tracking effect of K-MPC on step signal is shown in Figure 4.1, which intuitively shows that K-MPC has a good control ability in the face of the air-breathing hypersonic vehicle model with strong nonlinear and aerodynamic-thrust-elastic coupling characteristics.

3.2 Comparison between K-MPC controllers based on different numerical approximations

K-MPC based on DMDc, EDMD and KNN linear models were simulated and compared respectively, and the simulation conditions were the same as described in section 4.1.

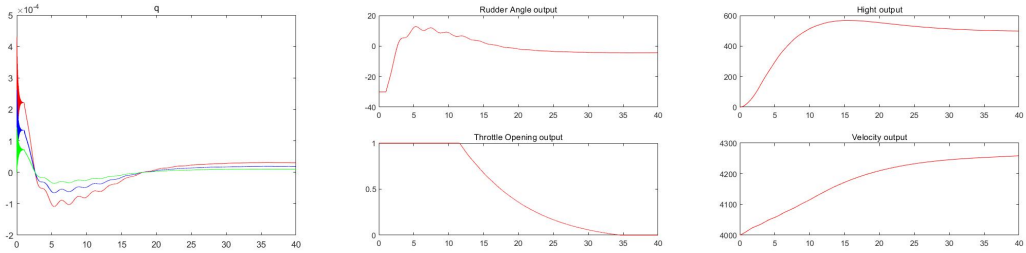

Figure 4.2 Simulation result of DMDc Koopman-MPC

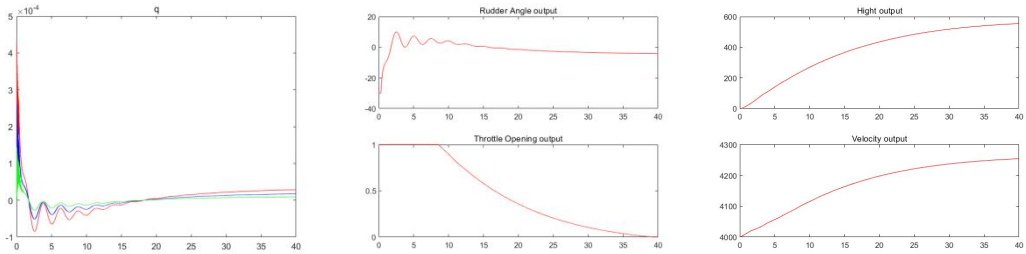

Figure 4.3 Simulation result of EDMD Koopman-MPC

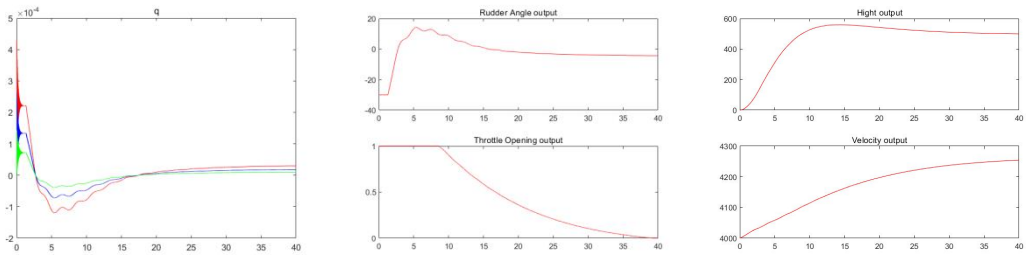

Figure 4.4 Simulation result of KNN Koopman-MPC

The calculation results of simulation comparison diagram are shown in Figure 4.2 to Figure 4.4 respectively, indicating that KNN-MPC has certain advantages in control performance.

4. Conclusion

In this paper, a Koopman model predictive controller based on Koopman Neural Network is proposed for the control of elastic hypersonic vehicle, and its feasibility and performance are verified by simulation. In fact, many parts mentioned in this question are still worth further discussion, such as the feasibility of using semi-online learning KNN-MPC (both neural network and Koopman operator are updated online, but the updating of neural network will have a relatively longer period) for aircraft control. In addition, because this paper mainly carries out some feasibility analysis and research, it does not use some advanced control methods in recent years, only the simplest linear MPC is used as the controller. In the future work, the author will improve this part and compare its performance with the current mainstream advanced control methods.

References

[1] Shaughnessy D, Pinckney Z, McMinn D, et al. Hypersonic vehicle simulation model:Winged-cone configuration. Tech Rep,NASA TM-102610,1990.
[2] Mirmirani M, Wu C, Clark A. Modeling for control of a generic airbreathing hypersonic vehicle. AIAA Guidance, Navigation and Control Conference and Exhibit,San Francisco,2005,2005-6256:1-19.
[3] Clark D, Wu C, Mirmirani D. Development of an airframe-propulsion integrated generic hypersonic vehicle model. 44th AIAA Aerospace Science Meeting and Exhibit,Reno,2006.
[4] N. Takeishi, Y. Kawahara, and T. Yairi, Learning Koopman invariant subspaces for dynamic mode decomposition, in Advances in Neural Information Processing Systems, 2017, pp. 1130–1140.
[5] Brunton, S.L., Budišić, M., Kaiser, E., & Kutz, J.N. (2021). Modern Koopman Theory for Dynamical Systems. SIAM Rev., 64, 229-340.

Copyright Statement