energies

MDPI

Article

An Online Data-Driven LPV Modeling Method for Turbo-Shaft Engines

Ziyu Gu, Shuwei Pang, Wenxiang Zhou, Yuchen Li and Qiuhong Li *

Jiangsu Province Key Laboratory of Aerospace Power System, College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; guziyu@nuaa.edu.cn (Z.G.); sw.pang@nuaa.edu.cn (S.P.); zhouwx@nuaa.edu.cn (W.Z.); yuchenli@nuaa.edu.cn (Y.L.)
* Correspondence: lqh203@nuaa.edu.cn

Abstract: The linear parameter-varying (LPV) model is widely used in aero engine control system design. The conventional local modeling method is inaccurate and inefficient in the full flying envelope. Hence, a novel online data-driven LPV modeling method based on the online sequential extreme learning machine (OS-ELM) with an additional multiplying layer (MLOS-ELM) was proposed. An extra multiplying layer was inserted between the hidden layer and the output layer, where the hidden layer outputs were multiplied by the input variables and state variables of the LPV model. Additionally, the input layer was set to the LPV model's scheduling parameter. With the multiplying layer added, the state space equation matrices of the LPV model could be easily calculated using online gathered data. Simulation results showed that the outputs of the MLOS-ELM matched that of the component level model of a turbo-shaft engine precisely. The maximum approximation error was less than 0.18%. The predictive outputs of the proposed online data-driven LPV model after five samples also matched that of the component level model well, and the maximum predictive error within a large flight envelope was less than 1.1% with measurement noise considered. Thus, the efficiency and accuracy of the proposed method were validated.

Keywords: turbo-shaft engine; linear parameter-varying model; data-driven method; online sequential extreme learning machine (OS-ELM)

Citation: Gu, Z.; Pang, S.; Zhou, W.; Li, Y.; Li, Q. An Online Data-Driven LPV Modeling Method for Turbo-Shaft Engines. *Energies* **2022**, *15*, 1255. https://doi.org/10.3390/en15041255

Academic Editor: Yolanda Vidal

Received: 7 December 2021
Accepted: 7 February 2022
Published: 9 February 2022

1. Introduction

The behavior of an aero engine can be described by a linear parameter-varying (LPV) model, which can approximate a nonlinear system or a time-varying linear system with a combination of linear time-invariant (LTI) models [1–3]. The parameters of LPV models are expressed as a function of the scheduling variables or the operating point variables, capturing the dynamic changes of the nonlinear system accurately [4,5]. The main advantage of LPV models is that they allow the application of well-established linear design techniques to complex nonlinear systems based on the linear matrix inequality [6,7] and linear quadric methods [8]. For example, the online optimization of model predictive control (MPC) or performance seeking control (PSC) can be reduced to a linear program or a quadratic program, thereby achieving high real-time property when an LPV model is used as a predictive model [9,10]. LPV models preserve the advantageous properties of LTI models, such as low modeling cost, while being able to represent a large class of nonlinear systems with high accuracy [11]. Therefore, the LPV model has been widely used in convex system controller design [12,13] and model-based approaches [14,15], such as robust control, fault diagnosis and optimal control.

The LPV and quasi-LPV identification method has attracted much attention in recent years [16–18]. Quasi- means that the scheduling parameters of the LPV model include system variables [19]. There are two main methods of LPV (or quasi-LPV) modeling which have been researched in the literature: the global method and the local method

[20,21]. In the global method, the LPV model of the system can be derived from one experiment. Additionally, both control inputs and scheduling parameters are excited simultaneously in this experiment, which is not realizable in aero engine LPV modeling applications [20]. This is because the dynamic of the engine is a continuous function of the wide flying envelope and the various operation states, which can hardly be covered by finite-time experiments. Furthermore, the state variable of an aero engine is often directly selected as the scheduling parameter, which is dependent on control inputs and cannot be used to excite the system in a global modeling experiment. In the local method, a certain number of working states in the scheduling space are selected as the LPV model base point. A local LTI model is established at each point, and the LPV model is based on the interpolation of the local LTI systems [1–3,22,23]. The use of LTI models renders the local LPV modeling method simple and easily realized in complicated nonlinear systems.

However, for aero engines, it is difficult to build LTI models in the full flying envelope and variety of working states; also, the regularity of the parameters' variation is poor [24,25]. The interpolation of the LTI parameters can be very complicated and time-consuming in the full envelope, and even fail to stabilize the system [26]. A common solution involves only modeling the aero engine at sea level with zero flying Mach numbers [27,28]. Models on other conditions are acquired through similarity transformation. Unfortunately, similarity transformation is carried out with many hypotheses which may not strictly hold in real operating states [29,30], especially for turbo-shaft engines. Therefore, the accuracy of the LPV model based on the similarity transformation may not satisfy the model-based control requirement of the turbo-shaft engine.

In recent years, data-driven methods have been employed as a powerful tool for LPV modeling since they can be carried out without mechanism models. In a data-driven modeling method, only the input, state and output trajectory information are used to identify the parameters [31]. The least square criterion and gradient descent algorithm are adopted to optimize the parameters of the LPV model [31–33]. A statistical approach was used in data-driven LPV system identification in [34]. However, the previously mentioned data-driven modeling methods are inefficient because they must be carried out offline [31,32,34] or carry online identification at the cost of reduced accuracy with the stochastic gradient descent algorithm. For aero engines, a more engaging, newly developed data-driven modeling method is the equilibrium manifold expansion (EME) model [35–37]. Differing from traditional LPV models of aero engines, the EME model considers the mapping between the equilibrium points of the engine and the scheduling parameters [36]. Thus, the EME can achieve high accuracy with less data. However, it can sometimes be difficult to find enough steady-state points to meet the needs of EME modeling, and then a similarity transformation must be conducted [37]. The similarity transformation may cause inaccuracies in the turbo-shaft engine. With the spread of intelligent technology, there have been efforts to build an LPV model by using support vector machines (SVM) [38,39], a promising approach to data-driven identification of LPV models [40]. Fényes et al. [41] presented a data-driven LPV modeling method for the control design of autonomous vehicle systems through machine learning, but it still required a reasonably large data set for training. Compared to the previous data-driven LPV modeling methods, the intelligent method can become more efficient by taking advantage of its online learning ability.

Hence, a novel online data-driven LPV (DD-LPV) modeling method for turbo-shaft engines based on a special neural network was proposed in this paper. It derived the DD-LPV model from an online sequential extreme learning machine (OS-ELM), with an extra multiplying layer (MLOS-ELM). It was able to achieve state space models at any operating point, static or dynamic, and modeling without the companion of the component-level model (CLM) of aero engines. Utilizing the online DD-LPV model derived by MLOS-ELM, a state space model was generated at every sample period based on the instant online gathered data, which made this model represent the current property of the individual engine accurately in any flight condition.

Compared with existing studies, the main contribution of this paper can be summarized as follows: (1) A special structure OS-ELM named MLOS-ELM was proposed in order to build a state space model directly and conveniently in the form of an artificial neural network. With the multiplying layer added, the neural network model and the LPV model were linked closely in the same output form. (2) A novel DD-LPV online modeling method was proposed which could linearize the dynamic system of the turbo-shaft engine at any working state online with the instantly gathered data. (3) The DD-LPV online modeling method provided a high-accuracy online predictive model for model-based approaches, such as MPC, and fault diagnoses of the individual aero-engine.

The paper is organized as follows: Section 2 introduces the description of the turbo-shaft engine's LPV model. Section 3 presents the special structure of MLOS-ELM for DD-LPV modeling. Section 4 introduces the online updating algorithm of the MLOS-ELM. Section 5 discusses the simulation results and Section 6 concludes this paper.

2. LPV Model for Turbo-Shaft Engine

2.1. Brief Introduction of Turbo-Shaft Engine

The sketch of the bi-rotor turbo-shaft engine researched in this paper is shown in Figure 1 [42]. During the operation, the compressed air is drawn from the inlet and flows through the compressor to the combustor where fuel is introduced, mixed, and ignited. Then, the hot gas flows through the gas turbine which extracts a portion of the energy to drive the compressor. Finally, the gas exiting the gas turbine flows through the power turbine which extracts the remaining energy to drive the main rotor [43,44]. All the components work cooperatively to provide power to the helicopter.

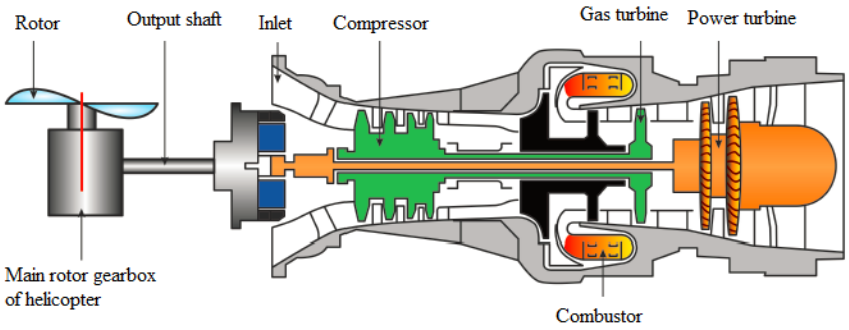


Figure 1. Structural sketch of turbo-shaft engine.

For smooth and reliable operation of the helicopter, the rotor speed of the power turbine was always kept at 100% to maintain the main rotor speed constant [42]. This was achieved by adjusting the fuel flow input to the engine. When a flight condition of the helicopter changed, such as forward speed, the required power of the main rotor of the helicopter varied accordingly. This meant that the power turbine needed to provide more or less power to meet the requirement of the main rotor and reach a newly balanced state.

2.2. LPV Model Description

The general discrete state-space equation of LPV model can be described as:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}(\alpha)\boldsymbol{x}(k) + \boldsymbol{B}(\alpha)\boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C}(\alpha)\boldsymbol{x}(k) + \boldsymbol{D}(\alpha)\boldsymbol{u}(k) \end{aligned} \tag{1}$$

where α is the scheduling parameter, $\boldsymbol{u}(k)$ is the input vector of the LPV model, $\boldsymbol{x}(k)$ is the state variable vector, $\boldsymbol{y}(k)$ is the output vector of the LPV model and $\boldsymbol{A}(\alpha)$, $\boldsymbol{B}(\alpha)$, $\boldsymbol{C}(\alpha)$ and $\boldsymbol{D}(\alpha)$ are time-varying matrices scheduled by α.

For turbo-shaft engine, its main dynamic is derived from the cooperative working of the rotor systems [44], which can be described as:

$$\begin{aligned}\frac{dn_g}{dt} &= (W_g * \eta_g - W_c) / [n_g * J_g * (\frac{30}{\pi})^2] \\ \frac{dn_p}{dt} &= (W_p * \eta_p - W_r) / [n_p * J_p * (\frac{30}{\pi})^2]\end{aligned} \tag{2}$$

where n_g is the rotor speed of gas turbine, n_p is the rotor speed of power turbine, W_c, W_g, W_p, W_r are the power values of the compressor, gas turbine, power turbine and main rotor, respectively, η_g and η_p are the shaft efficiency values of the gas turbine and power turbine, respectively and J_g and J_p are the rotor inertia values of the gas turbine and power turbine, respectively.

Equation (2) is a basic equation for the turbo-shaft engine's dynamic calculation in its mechanism model (for more detail, refer to [42–44]). Thus, in research relevant to the state space model of turbo-shaft engines, n_g and n_p are most frequently selected as the state variables [42–46]. In addition, n_g can reflect both the output power and the dynamic property of the engine [47,48]. Therefore, the LPV model of turbo-shaft engine can be described in a quasi-LPV form:

$$\begin{aligned}\boldsymbol{x}(k+1) &= \begin{pmatrix} n_g(k+1) \\ n_p(k+1) \end{pmatrix} = \boldsymbol{A}(n_g(k)) \begin{pmatrix} n_g(k) \\ n_p(k) \end{pmatrix} + \boldsymbol{B}(n_g(k)) W_f(k) \\ \boldsymbol{y}(k) &= \begin{pmatrix} n_g(k) \\ n_p(k) \\ P_3(k) \\ T_{44}(k) \end{pmatrix} = \boldsymbol{C}(n_g(k)) \begin{pmatrix} n_g(k) \\ n_p(k) \end{pmatrix} + \boldsymbol{D}(n_g(k)) W_f(k)\end{aligned} \tag{3}$$

where n_g is selected as both the state variable and the scheduling parameter of the LPV equation, which has $\alpha(k) = n_g(k)$. Additionally, n_p is the controlled variable and another state variable; thus, we have $\boldsymbol{x}(k) = [n_g(k)\ n_p(k)]$. The fuel flow W_f is the main control input $u(k) = W_f(k)$. P_3 is the total pressure at the compressor exit and T_{44} is the total temperature at the power turbine inlet. P_3, T_{44} together with n_g, are all limit parameters for turbo-shaft engine security which should be taken into consideration in the control system design process to avoid overspeed, overpressure and overtemperature. Therefore, the output vector of the LPV model was set to $\boldsymbol{y}(k) = [y_1(k)\ y_2(k)\ y_3(k)\ y_4(k)]^T = [n_g(k)\ n_p(k)\ P_3(k)\ T_{44}(k)]^T$.

Note that n_g and n_p were already chosen as the states, so its corresponding entries in $\boldsymbol{C}$ and $\boldsymbol{D}$ had a constant value of 0 and 1. For P_3 and T_{44}, corresponding entries in matrix $\boldsymbol{C}$ and $\boldsymbol{D}$ varied with the scheduling parameter [15].

In the proposed method, the LPV model was built based on the online training neural network. However, the order of magnitude of the parameters in Equation (3) differed from each other considerably. Therefore, all the data used in the model should be normalized into the order of magnitude of proximity to enhance the accuracy of the model and avoid data overwhelming. The corresponding normalized LPV model is:

$$\begin{aligned}\overline{\boldsymbol{x}}(k+1) &= \begin{pmatrix} \overline{n}_g(k+1) \\ \overline{n}_p(k+1) \end{pmatrix} = \overline{\boldsymbol{A}}(\overline{n}_g(k)) \begin{pmatrix} \overline{n}_g(k) \\ \overline{n}_p(k) \end{pmatrix} + \overline{\boldsymbol{B}}(\overline{n}_g(k)) \overline{W}_f(k) \\ \overline{\boldsymbol{y}}(k) &= \begin{pmatrix} \overline{n}_g(k) \\ \overline{n}_p(k) \\ \overline{P}_3(k) \\ \overline{T}_{44}(k) \end{pmatrix} = \overline{\boldsymbol{C}}(\overline{n}_g(k)) \begin{pmatrix} \overline{n}_g(k) \\ \overline{n}_p(k) \end{pmatrix} + \overline{\boldsymbol{D}}(\overline{n}_g(k)) \overline{W}_f(k)\end{aligned} \tag{4}$$

where the bar over the variable means that the variable is a normalized parameter, the $\overline{\boldsymbol{A}}$, $\overline{\boldsymbol{B}}$, $\overline{\boldsymbol{C}}$, $\overline{\boldsymbol{D}}$ are normalized system and output matrices.

The normalization was carried out as follows:

$$\bar{d} = \frac{2d - d_{\max} - d_{\min}}{d_{\max} - d_{\min}} \tag{5}$$

where $d_{\max}$ and $d_{\min}$ are the maximum and minimum values of any given variable d.

3. Special Structure of MLOS-ELM for DD-LPV Modeling

The online sequential extreme learning machine (OS-ELM) is a kind of single-hidden-layer feedforward neural network (SLFN). It was adopted to train neural networks due to its faster online learning and universal approximation capabilities. In OS-ELM, the training sets of input vector and target vector come one by one or chunk by chunk. For one training data set $\{(\boldsymbol{x}_{in}, \boldsymbol{t})\}$, the OS-ELM model with L additive hidden nodes and activation function $f(\cdot)$ can be formulated as:

$$\hat{\boldsymbol{y}} = \sum_{i=1}^{L} \boldsymbol{\beta}_i f(\boldsymbol{W}_i \boldsymbol{x}_{in} + b_i) \tag{6}$$

where $\hat{\boldsymbol{y}}$ is OS-ELM model's output, $\boldsymbol{W}_i$ is the weight from the input layer to the i^{th} hidden node, b_i is the bias of the i^{th} hidden node, $\boldsymbol{\beta}_i$ is the weight from the i^{th} hidden node to the output layer.

For DD-LPV modeling, a special layer called multiplying layer was inserted between the hidden layer and the output layer based on common OS-ELM, as shown in Figure 2. The inputs of the multiplying layer were the outputs of the hidden layer, the normalized states, and the normalized inputs of the LPV model. The output of the multiplying layer node was the product of its inputs.

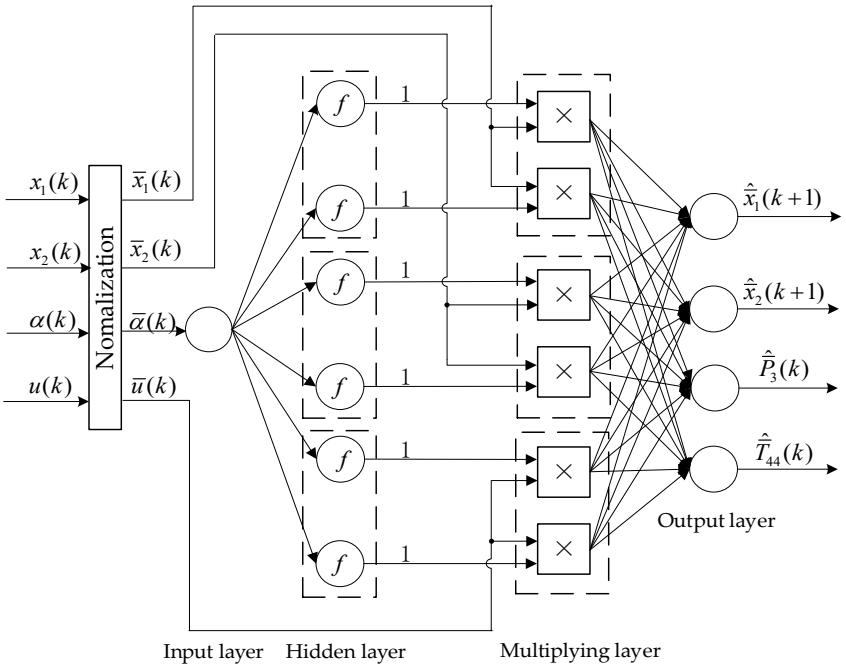


Figure 2. MLOS-ELM structure for DD-LPV modeling of the turbo-shaft engine.

There were one-to-one local connections between the hidden layer and the multiplying layer, whose weight was set to 1. The nodes of the hidden layer were divided into 3

groups according to the total dimension of $\boldsymbol{x}$ (2-dimension) and u (1-dimension), and each group had 2 nodes, determined by heuristics. Therefore, the outputs of the MLOS-ELM become:

$$\begin{aligned}\hat{y}_m &= \sum_{j=1}^{2}\beta_{jm}(k)f(W_{1j}\bar{\alpha}(k)+b_j)\bar{x}_1(k)+\sum_{j=3}^{4}\beta_{jm}(k)f(W_{1j}\bar{\alpha}(k)+b_j)\bar{x}_2(k)\\ &\quad+\sum_{j=5}^{6}\beta_{jm}(k)f(W_{1j}\bar{\alpha}(k)+b_j)\bar{u}(k)\\ m &= 1,2,3,4\end{aligned} \tag{7}$$

where $\hat{\boldsymbol{y}}=\left[\hat{\bar{x}}_1(k+1)\ \hat{\bar{x}}_2(k+1)\ \hat{\bar{P}}_3(k)\ \hat{\bar{T}}_{44}(k)\right]^T$, W_{1j} is the weight from the input node to the j^{th} hidden node, b_j is the bias of the j^{th} hidden node, β_{jm} is the weight from the j^{th} multiplying layer node to the m^{th} output node. (β_{jm} is related to time k because it is updated online, while W_{1j} and b_j are invariant, as detailed in Section 4).

The normalized LPV model of a turbo-shaft engine described by Equation (4), can be written in the form of matrix entries as:

$$\begin{aligned}\bar{\boldsymbol{x}}(k+1) &= \begin{bmatrix}\bar{x}_1(k+1)\\ \bar{x}_2(k+1)\end{bmatrix}=\begin{bmatrix}\bar{a}_{11}(\bar{\alpha}) & \bar{a}_{12}(\bar{\alpha})\\ \bar{a}_{21}(\bar{\alpha}) & \bar{a}_{22}(\bar{\alpha})\end{bmatrix}\begin{bmatrix}\bar{x}_1(k)\\ \bar{x}_2(k)\end{bmatrix}+\begin{bmatrix}\bar{b}_1(\bar{\alpha})\\ \bar{b}_2(\bar{\alpha})\end{bmatrix}\bar{u}(k)\\ \bar{\boldsymbol{y}}(k) &= \begin{bmatrix}\bar{x}_1(k)\\ \bar{x}_2(k)\\ \bar{P}_3(k)\\ \bar{T}_{44}(k)\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\\ \bar{c}_{31}(\bar{\alpha}) & \bar{c}_{32}(\bar{\alpha})\\ \bar{c}_{41}(\bar{\alpha}) & \bar{c}_{42}(\bar{\alpha})\end{bmatrix}\begin{bmatrix}\bar{x}_1(k)\\ \bar{x}_2(k)\end{bmatrix}+\begin{bmatrix}0\\ 0\\ \bar{d}_3(\bar{\alpha})\\ \bar{d}_4(\bar{\alpha})\end{bmatrix}\bar{u}(k)\end{aligned} \tag{8}$$

According to Equation (7), the entries of $\bar{\boldsymbol{A}}(\bar{\alpha})$, $\bar{\boldsymbol{B}}(\bar{\alpha})$, $\bar{\boldsymbol{C}}(\bar{\alpha})$, $\bar{\boldsymbol{D}}(\bar{\alpha})$ in Equation (8) were obtained:

$$\begin{aligned}\bar{\boldsymbol{A}}(\bar{\alpha}) &= \begin{bmatrix}\sum_{j=1}^{2}\beta_{j1}(k)f(W_{1j}\bar{\alpha}(k)+b_j) & \sum_{j=3}^{4}\beta_{j1}(k)f(W_{1j}\bar{\alpha}(k)+b_j)\\ \sum_{j=1}^{2}\beta_{j2}(k)f(W_{1j}\bar{\alpha}(k)+b_j) & \sum_{j=3}^{4}\beta_{j2}(k)f(W_{1j}\bar{\alpha}(k)+b_j)\end{bmatrix}\\ \bar{\boldsymbol{B}}(\bar{\alpha}) &= \begin{bmatrix}\sum_{j=5}^{6}\beta_{j1}(k)f(W_{1j}\bar{\alpha}(k)+b_j) & \sum_{j=5}^{6}\beta_{j2}(k)f(W_{1j}\bar{\alpha}(k)+b_j)\end{bmatrix}^T\\ \bar{\boldsymbol{C}}(\bar{\alpha}) &= \begin{bmatrix}1 & 0\\ 0 & 1\\ \sum_{j=1}^{2}\beta_{j3}(k)f(W_{1j}\bar{\alpha}(k)+b_j) & \sum_{j=3}^{4}\beta_{j3}(k)f(W_{1j}\bar{\alpha}(k)+b_j)\\ \sum_{j=1}^{2}\beta_{j4}(k)f(W_{1j}\bar{\alpha}(k)+b_j) & \sum_{j=3}^{4}\beta_{j4}(k)f(W_{1j}\bar{\alpha}(k)+b_j)\end{bmatrix}\\ \bar{\boldsymbol{D}}(\bar{\alpha}) &= \begin{bmatrix}0 & 0 & \sum_{j=5}^{6}\beta_{j3}(k)f(W_{1j}\bar{\alpha}(k)+b_j) & \sum_{j=5}^{6}\beta_{j4}(k)f(W_{1j}\bar{\alpha}(k)+b_j)\end{bmatrix}^T\end{aligned} \tag{9}$$

As can be seen from Equation (9), the non-constant elements of $\bar{\boldsymbol{A}}$, $\bar{\boldsymbol{B}}$, $\bar{\boldsymbol{C}}$, $\bar{\boldsymbol{D}}$ in Equation (4) were derived from the parameters of the MLOS-ELM, varying with the scheduling parameter α and output weight $\boldsymbol{\beta}$. By inserting a multiplying layer between the hidden

layer and the output layer, the output function of the MLOS-ELM was presented in the mathematical description form of the LPV model. That meant that the weights and the input scheduling parameter of the MSOL-ELM were given the ability to derive the parameters of the LPV model. In other words, Equation (9) linked the neural network model and the LPV model closely, which made the LPV model able to be derived directly and conveniently by the network's inner parameters at every sample time *k*.

Remark 1. The numbers of the hidden nodes and the multiplying nodes were both set to 6 by heuristics in this research, which was sufficient to achieve an accurate MLOS-ELM model. The smallest number of nodes in each group was 2, to ensure the accuracy of the MLOS-ELM in this research. If each group had only 1 node, MLOS-ELM's accuracy would have been negatively affected. However, if each group had more than 2 nodes, its accuracy would not be improved but the computational burden would increase rapidly. Moreover, they could be set to different numbers (no less than 3) separately. Owing to there being 3 independent state and input variables $\overline{x}_1, \overline{x}_2, \overline{u}$ in turbo-shaft engine's LPV model, the nodes in the multiplying layer must be divided into 3 groups, evenly or not. Then, the output of the network would always have the expression in form of Equation (7) which in turn had an LPV form. The even structure presented in the paper was simple and easily described.

Remark 2. MLOS-ELM's input layer had only one input (the 1-dimensional scheduling parameter $\overline{\alpha}(k) = \overline{n}_g(k)$) which was enough to express the variation of the matrices of the DD-LPV model. Other factors that could have affected the matrices, such as Mach number *Ma* or altitude *H*, were not used as the inputs because the online training method was adopted. Therein, the output weight of the network was adapted to the current working condition, including the Mach number and the altitude change, so the entries in Equation (9) could automatically adjust according to the flying condition. Moreover, taking more parameters as the inputs of MLOS-ELM could have affected the accuracy of the DD-LPV.

4. Online Updating Algorithm of MLOS-ELM

OS-ELM was utilized to build the network for its high real-time properties and competence in recursive learning. The convergence of OS-ELM was proven in [49]. At each sample time *k*, the newly gathered data set was used to update OS-ELM's output weight $\boldsymbol{\beta}$.

According to the MLOS-ELM described in Figure 2. the output vector of the hidden layer at sample time *k* is:

$$\begin{aligned} \boldsymbol{H}(k) &= [h_1(k) \quad h_2(k) \quad \cdots \quad h_6(k)] \\ &= [f(W_{11}\overline{\alpha}(k)+b_1) \quad f(W_{12}\overline{\alpha}(k)+b_2) \quad \cdots \quad f(W_{16}\overline{\alpha}(k)+b_6)] \end{aligned} \tag{10}$$

where $\boldsymbol{W} \in R^{1\times 6}$ and $\boldsymbol{b} \in R^{1\times 6}$ are randomly generated at the first time step and kept constant, $f(\boldsymbol{W}\overline{\alpha}+\boldsymbol{b}) = \dfrac{1}{1+e^{-(\boldsymbol{W}\overline{\alpha}+\boldsymbol{b})}}$.

Then the output vector of the multiplying layer is:

$$\boldsymbol{O}(k) = \left[h_1(k)\overline{x}_1(k) \;\; h_2(k)\overline{x}_1(k) \;\; h_3(k)\overline{x}_2(k) \;\; h_4(k)\overline{x}_2(k) \;\; h_5(k)\overline{u}(k) \;\; h_6(k)\overline{u}(k)\right] \tag{11}$$

The outputs of MLOS-ELM model of Equation (7) can be rewritten in a compact form:

$$\hat{\boldsymbol{y}}(k) = \boldsymbol{O}(k)\boldsymbol{\beta}(k) \to \boldsymbol{t}(k) \tag{12}$$

where $\boldsymbol{\beta} \in R^{6\times 4}$, $\boldsymbol{t}(k) = \left[\overline{x}_1(k+1) \;\; \overline{x}_2(k+1) \;\; \overline{P}_3(k) \;\; \overline{T}_{44}(k)\right]^T$ is the target vector.

If $k = 1$, the weight $\boldsymbol{\beta}$ can be calculated by least square method:

$$\boldsymbol{\beta}(1) = (\boldsymbol{O}^T(1)\boldsymbol{O}(1))^{-1}\boldsymbol{O}^T(1)\boldsymbol{t}(1) \tag{13}$$

However, $\boldsymbol{O}(1) \in R^{1\times 6}$ has rank 1 (not full column rank), thus $\boldsymbol{O}(1)^T\boldsymbol{O}(1)$ is singular. Then the ridge regression theory is used to rewrite Equation (13):

$$\boldsymbol{\beta}(1) = (\boldsymbol{O}(1)^T\boldsymbol{O}(1) + \lambda\mathbf{I})^{-1}\boldsymbol{O}(1)^T\boldsymbol{t}(1) \tag{14}$$

where λ is the generalized regularization parameter.

If $k > 1$, the output weight $\boldsymbol{\beta}$ was updated recursively online every sampling period to adapt to the new working state of the engine. However, in some instances, the outputs of MLOS-ELM already matched the target outputs well with previous output weight, especially at a steady state. Under such conditions, there was no need to update the output weight at every sampling time, which increased the computational burden. Therefore, a criterion was set to judge whether the output weight needed to be updated according to the output error of the network.

The output error was defined by relative percentage error:

$$err_i(k) = \frac{\hat{y}_i(k) - t_i(k)}{t_i(k)} \times 100\% \tag{15}$$

If one of MLOS-ELM's output errors err_i, $i = 1,2,3,4$ exceeded the update criterion, the recursive least square method was used to update the weight according to the new gathered data. Otherwise, the weight maintained the same value as the last sampling time.

Then we have the output weight $\boldsymbol{\beta}(k)$ $(k > 1)$ as:

If $\left|err_i(k)\right| < \delta_i$, $i = 1,2,3,4$

$$\boldsymbol{\beta}(k) = \boldsymbol{\beta}(k-1) \tag{16}$$

else

$$\begin{cases} \boldsymbol{\beta}(k) = \boldsymbol{\beta}(k-1) + \boldsymbol{P}(k)\boldsymbol{O}^T(k)(\boldsymbol{t}(k) - \hat{\boldsymbol{y}}(k)) \\ \boldsymbol{P}(k) = \boldsymbol{P}(k-1) - \dfrac{\boldsymbol{P}(k-1)\boldsymbol{O}(k)\boldsymbol{O}^T(k)\boldsymbol{P}(k-1)}{1 + \boldsymbol{O}^T(k)\boldsymbol{P}(k-1)\boldsymbol{O}(k)} \end{cases} \tag{17}$$

where $\delta_i > 0$, $i = 1,2,3,4$ is the maximum allowable output error of MLOS-ELM, $\boldsymbol{P}(1)$ is defined as $\boldsymbol{P}(1) = (\boldsymbol{O}^T(1)\boldsymbol{O}(1) + \lambda\mathbf{I})^{-1}$.

Combining Sections 3 and 4, the proposed online DD-LPV modeling method of turbo-shaft engine based on MLOS-ELM can be summarized in the following steps:

- **Step 1**. Randomly generate the hidden layer parameters W and b, set δ and λ;
- **Step 2**. Update the output weight $\boldsymbol{\beta}(k)$: If $k = 1$, Equation (14) is used. Otherwise, Equation (16) or Equation (17) is used;
- **Step 3**. Calculate the LPV system matrices $\bar{\boldsymbol{A}}(\bar{\alpha})$, $\bar{\boldsymbol{B}}(\bar{\alpha})$, $\bar{\boldsymbol{C}}(\bar{\alpha})$, $\bar{\boldsymbol{D}}(\bar{\alpha})$ according to Equation (9);
- **Step 4**. Set $k = k + 1$, go back to **Step 2**.

Remark 1. To be consistent with the description of the discrete state space equation in Equation (4), the outputs of MLOS-ELM and the target vector in Equation (12) included states at time $k + 1$. Since the data at time $k + 1$ could not be collected online at time k in real applications, the DD-LPV model at time k was obtained by the data of the states at time k and the outputs at time $k - 1$.

5. Simulation and Discussions

To verify the accuracy of the MLOS-ELM network and the online DD-LPV model derived from it, simulations in the flight envelope were carried out based on the component level model (CLM) of the UH-60 helicopter and T700 engine system platform for lack of rig test data. The accuracy of this integrated UH-60 helicopter and T700 engine model was checked and verified in [44,50] by a great number of tests. The test results prove that the integrated model not only had good reliability, but was able to carry out digital flight simulations of routine missions [44]. The data generated by the integrated model had high fidelity. For more detailed information, refer to [43,44,51].

The output weight update criteria of MLOS-ELM were $\delta_i = 0.005$, $i = 1,2,3,4$ and $\lambda = 10^{-8}$. The anti-normalized outputs were adopted to evaluate the accuracy of the MLOS-ELM and the DD-LPV model with physical value.

The true value of the outputs was attained by anti-normalization according to Equation (5):

$$d = \frac{\bar{d}(d_{\max} - d_{\min}) + d_{\max} + d_{\min}}{2} \tag{18}$$

In Sections 5.1 and 5.2, all comparisons and relative errors were calculated by anti-normalized data.

5.1. Approximation Ability Verification of the MLOS-ELM

First, the effectiveness of the proposed MLOS-ELM was tested through simulation. The target values of $\boldsymbol{t}$ were gathered online from the component level model (CLM) of the turbo-shaft engine. The purpose of this section was to validate whether the output of MLOS-ELM $\hat{\boldsymbol{y}}$ could track the output of CLM accurately.

The simulation was first carried out at altitude 0 m, with the trajectory of the forward speed of the helicopter increasing from 0 m/s to 30 m/s in the first 50 s, remaining invariant for 60 s and then decreasing to 0 m/s in the last 70 s, as shown in Figure 3. The turbo-shaft engine was controlled to keep the power turbine rotor speed at 100% during the whole process.

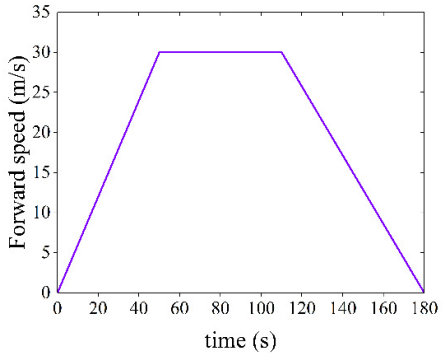
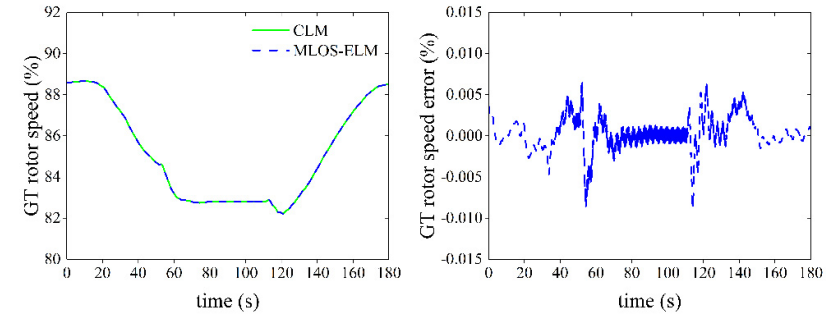
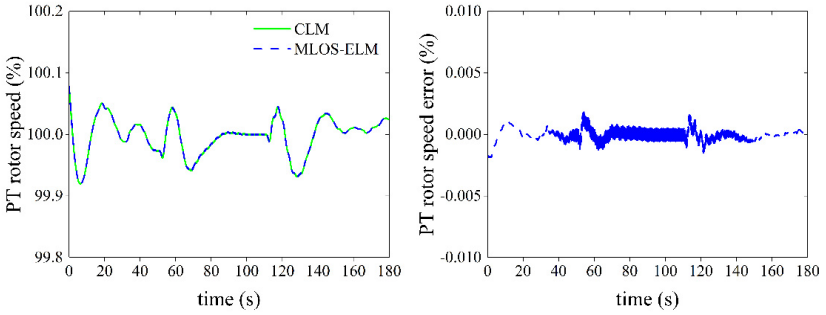


Figure 3. The forward speed of the helicopter in simulation.

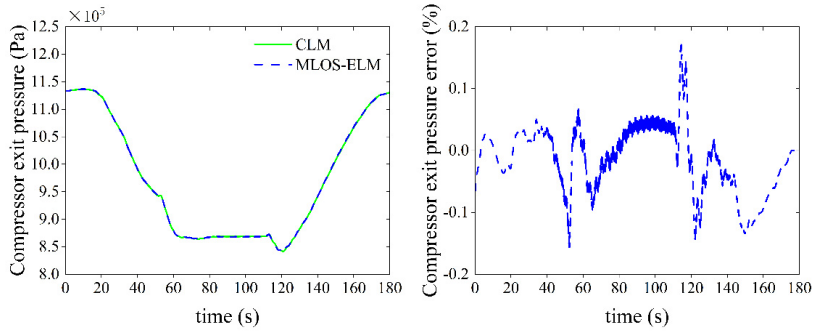
The comparisons of the MLOS-ELM network's output and the CLM's output are shown in Figure 4. In the figure, GT represents the gas turbine and PT represents the power turbine.


CLM
MLOS-ELM
GT rotor speed (%)
time (s)
GT rotor speed error (%)
time (s)

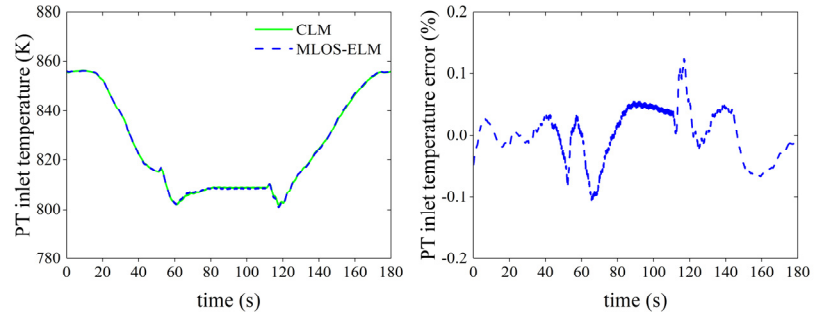
(**a**) (**b**)


CLM
MLOS-ELM
PT rotor speed (%)
time (s)
PT rotor speed error (%)
time (s)

(**c**) (**d**)


×10^5
CLM
MLOS-ELM
Compressor exit pressure (Pa)
time (s)
Compressor exit pressure error (%)
time (s)

(**e**) (**f**)


CLM
MLOS-ELM
PT inlet temperature (K)
time (s)
PT inlet temperature error (%)
time (s)

(**g**) (**h**)

Figure 4. Output comparison of MLOS-ELM and CLM (**a**) Responses of GT rotor speed. (**b**) Relative error of GT rotor speed. (**c**) Responses of PT rotor speed. (**d**) Relative error of PT rotor speed. (**e**) Responses of compressor exit pressure. (**f**) Relative error of compressor exit pressure. (**g**) Responses of PT inlet temperature. (**h**) Relative error of PT inlet temperature.

It can be seen from Figure 4 that the rotor speed response of the MLOS-ELM was able to match that of the CLM accurately during both the steady state and the dynamic process. The relative error of MLOS-ELM outputs to those of CLM was less than 0.009% for GT rotor speed and 0.003% for PT rotor speed. The error of compressor exit pressure and PT inlet temperature was also very small, which was within 0.18% and 0.12%, respectively. When the helicopter switched from acceleration to constant speed at 50 s and from constant speed to deceleration at 110 s, the network could still track the outputs well (though a sudden change of the forward speed trajectory occurred). The simulation results showed that the special OS-ELM, with an additional multiplying layer, achieved a high degree of approximation accuracy, which implied the effectiveness of the online DD-LPV model derived by MLOS-ELM.

5.2. Prediction Ability Validation of the Online DD-LPV Model in Flight Envelope

The online DD-LPV model was designed to be used in MPC or PSC, whose model was to predict the future outputs of the next few sample times. Therefore, in this section, the prediction ability of the proposed DD-LPV model was validated.

For the LPV model derived from the MLOS-ELM at time k, its state space equation matrices were $\bar{\boldsymbol{A}}(k), \bar{\boldsymbol{B}}(k), \bar{\boldsymbol{C}}(k), \bar{\boldsymbol{D}}(k)$. Its prediction outputs at time $k + p$ were obtained with the input series of the CLM over time k to $k + p$, and compared with the CLM output at time $k + p$ to verify the prediction ability of the LPV model derived at time k. As k moved along the test horizon, the prediction ability of the models derived at each sampling time were tested.

The prediction outputs of the LPV model were acquired by the response calculation of the discrete time system:

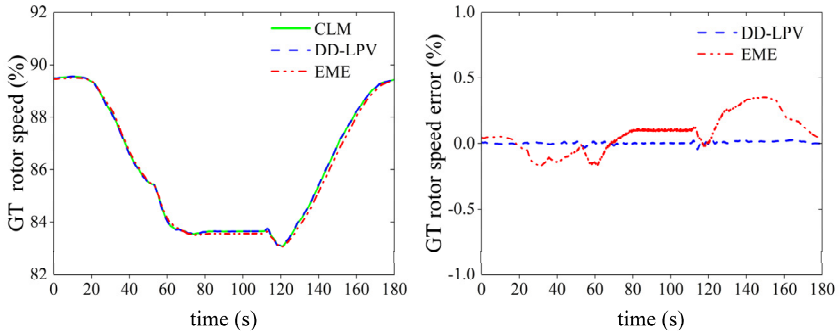
$$\begin{aligned} \bar{\boldsymbol{x}}(k+1) &= \bar{\boldsymbol{A}}(k)\bar{x}_{\mathrm{CLM}}(k) + \bar{\boldsymbol{B}}(k)\bar{u}_{\mathrm{CLM}}(k) \\ \bar{\boldsymbol{x}}(k+2) &= \bar{\boldsymbol{A}}(k)\bar{x}(k+1) + \bar{\boldsymbol{B}}(k)\bar{u}_{\mathrm{CLM}}(k+1) \\ &= \bar{\boldsymbol{A}}^2(k)\bar{x}_{\mathrm{CLM}}(k) + \bar{\boldsymbol{A}}(k)\bar{\boldsymbol{B}}(k)\bar{u}_{\mathrm{CLM}}(k) + \bar{\boldsymbol{B}}(k)\bar{u}_{\mathrm{CLM}}(k+1) \end{aligned} \tag{19}$$

Proceeding forward from Equation (19), we can obtain $\bar{\boldsymbol{x}}(k+p)$ and then $\bar{\boldsymbol{y}}(k+p)$:

$$\begin{aligned} \bar{\boldsymbol{x}}(k+p) &= \bar{\boldsymbol{A}}^p(k)\bar{x}_{\mathrm{CLM}}(k) + \sum_{m=k}^{k+p-1} \bar{\boldsymbol{A}}^{k+p-1-m}(k)\bar{\boldsymbol{B}}(k)u_{\mathrm{CLM}}(m) \\ \bar{\boldsymbol{y}}(k+p) &= \bar{\boldsymbol{C}}(k)\bar{\boldsymbol{A}}^p(k)\bar{x}_{\mathrm{CLM}}(k) + \sum_{m=k}^{k+p-1} \bar{\boldsymbol{C}}(k)\bar{\boldsymbol{A}}^{k+p-1-m}(k)\bar{\boldsymbol{B}}(k)u_{\mathrm{CLM}}(m) + \bar{\boldsymbol{D}}(k)u_{\mathrm{CLM}}(k+p) \end{aligned} \tag{20}$$

As a recently proposed data-driven modeling method for aero engines, an equilibrium manifolds expansion (EME) model [35–37] was also built on the UH-60/T700 system for prediction ability comparison. The EME model for the turbo-shaft engine was built at an altitude of 1000 m with different power requirements of the helicopter and was extended to the entire flight envelope through similarity transformation [37].

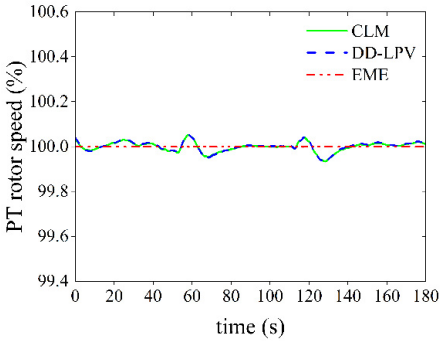
The simulations were carried out at altitudes of 1000 m and 3000 m with the same forward speed variation in Figure 3. The prediction results of $k + 5$ at 1000 m are shown in Figure 5 and the variations of the DD-LPV model parameters during this process are shown in Figure 6. Figure 7 shows the prediction results of $k + 5$ at 3000 m and the variation of the corresponding DD-LPV model parameters are shown in Figure 8. Random noises (magnitude 0.2%) were added to the data gathered from the CLM at 3000 m to imitate the measurement noise, which meant that the MLOS-ELM model at 3000 m was trained by the data with noise—and so was the DD-LPV model derived from it.


GT rotor speed (%)
time (s)
CLM
DD-LPV
EME

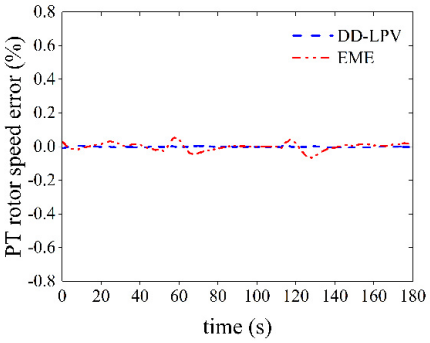
(**a**)

GT rotor speed error (%)
time (s)
DD-LPV
EME

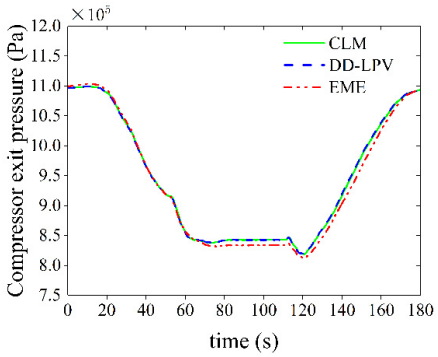
(**b**)


PT rotor speed (%)
time (s)
CLM
DD-LPV
EME

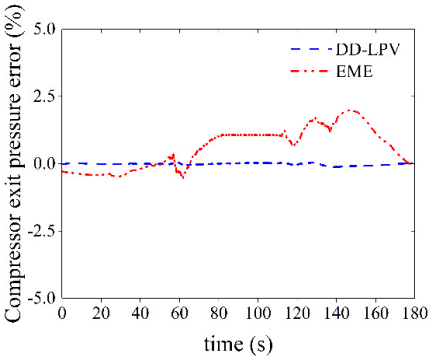
(**c**)


PT rotor speed error (%)
time (s)
DD-LPV
EME

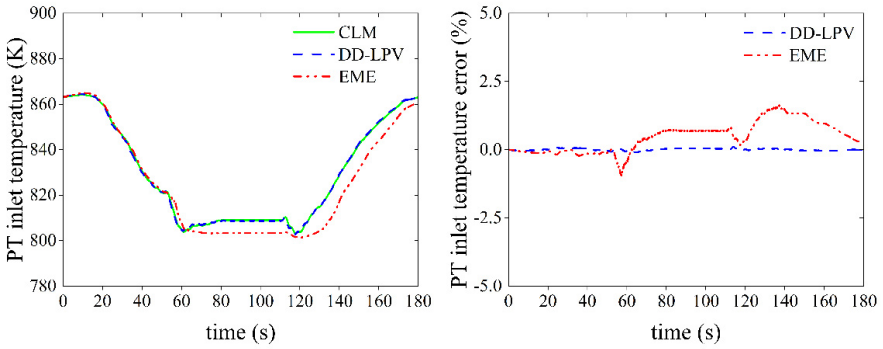
(**d**)


×10⁵
Compressor exit pressure (Pa)
time (s)
CLM
DD-LPV
EME

(**e**)

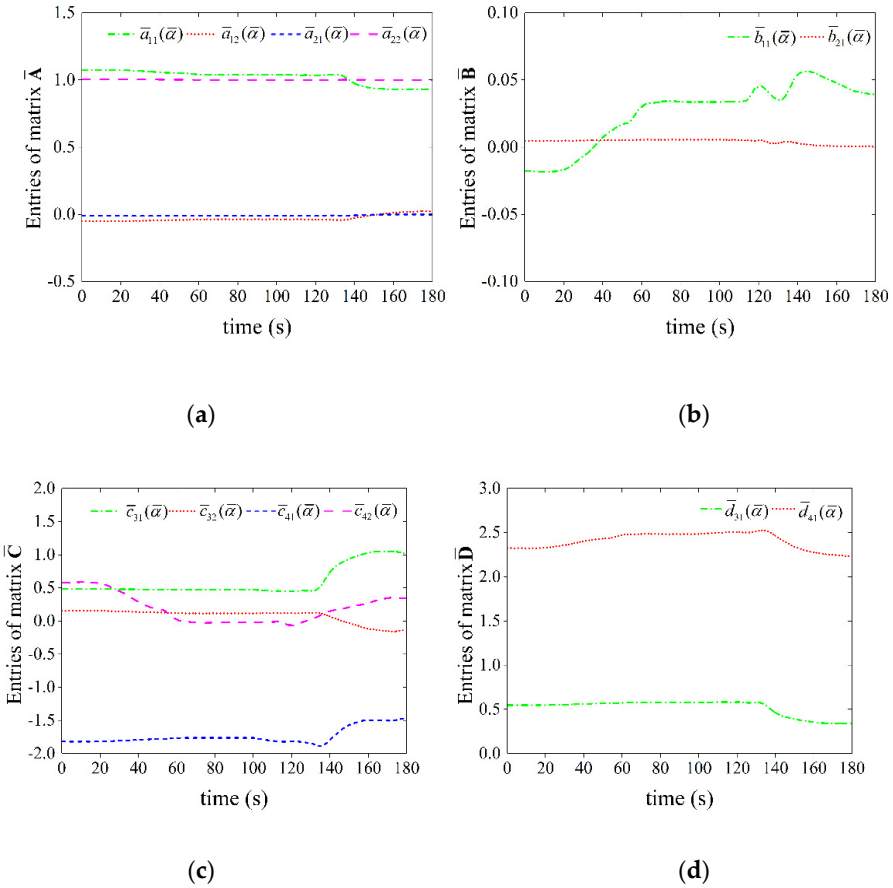

Compressor exit pressure error (%)
time (s)
DD-LPV
EME

(**f**)



(**g**) (**h**)

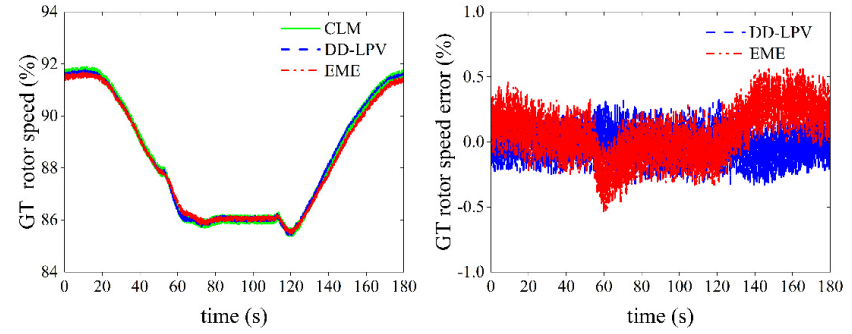
Figure 5. Prediction comparison of DD-LPV model and EME model to CLM at time $k+5$ (at 1000 m, without measurement noise). (**a**) Prediction of GT rotor speed. (**b**) Relative error of GT rotor speed prediction. (**c**) Prediction of PT rotor speed. (**d**) Relative error of PT rotor speed prediction. (**e**) Prediction of compressor exit pressure. (**f**) Relative error of compressor exit pressure prediction. (**g**) Prediction of PT inlet temperature. (**h**) Relative error of PT inlet temperature prediction.



(**a**) (**b**)

(**c**) (**d**)

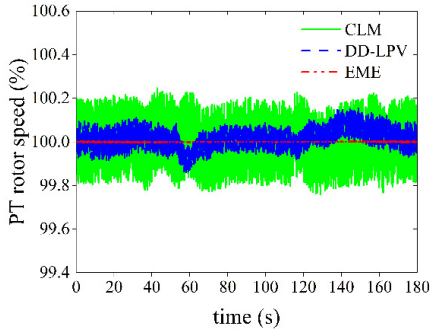
Figure 6. Parameter variations of DD-LPV model (at 1000 m, without measurement noise). (**a**) Entries of matrix $\bar{A}$. (**b**) Entries of matrix $\bar{B}$ (**c**) Entries of matrix $\bar{C}$. (**d**) Entries of matrix $\bar{D}$.


GT rotor speed (%)
time (s)
CLM
DD-LPV
EME

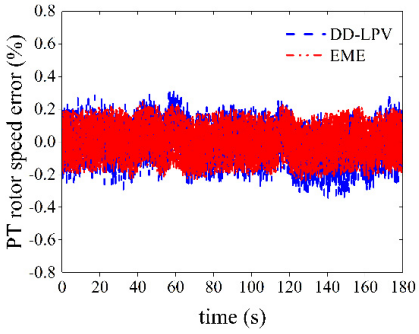
(**a**)

GT rotor speed error (%)
time (s)
DD-LPV
EME

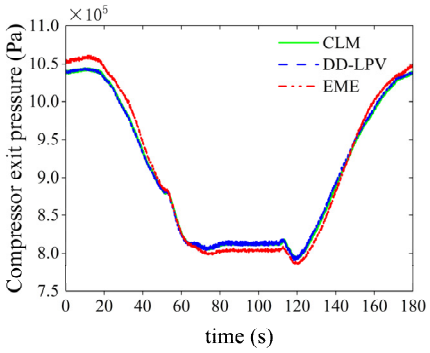
(**b**)


PT rotor speed (%)
time (s)
CLM
DD-LPV
EME

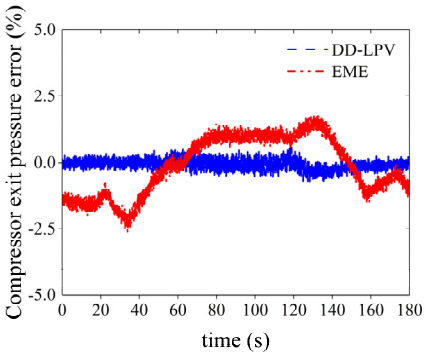
(**c**)


PT rotor speed error (%)
time (s)
DD-LPV
EME

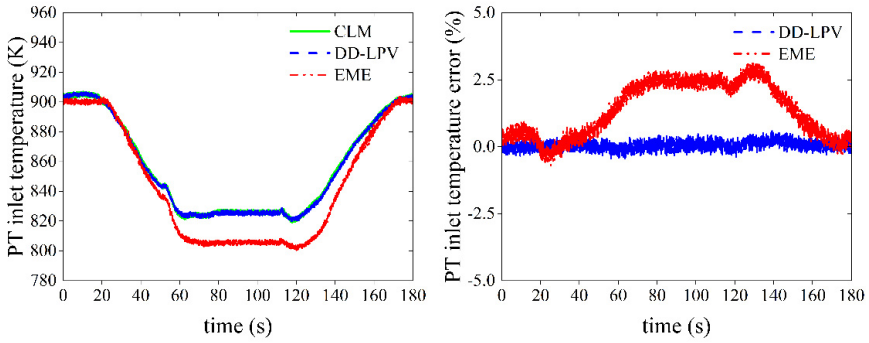
(**d**)


×10^5
Compressor exit pressure (Pa)
time (s)
CLM
DD-LPV
EME

(**e**)

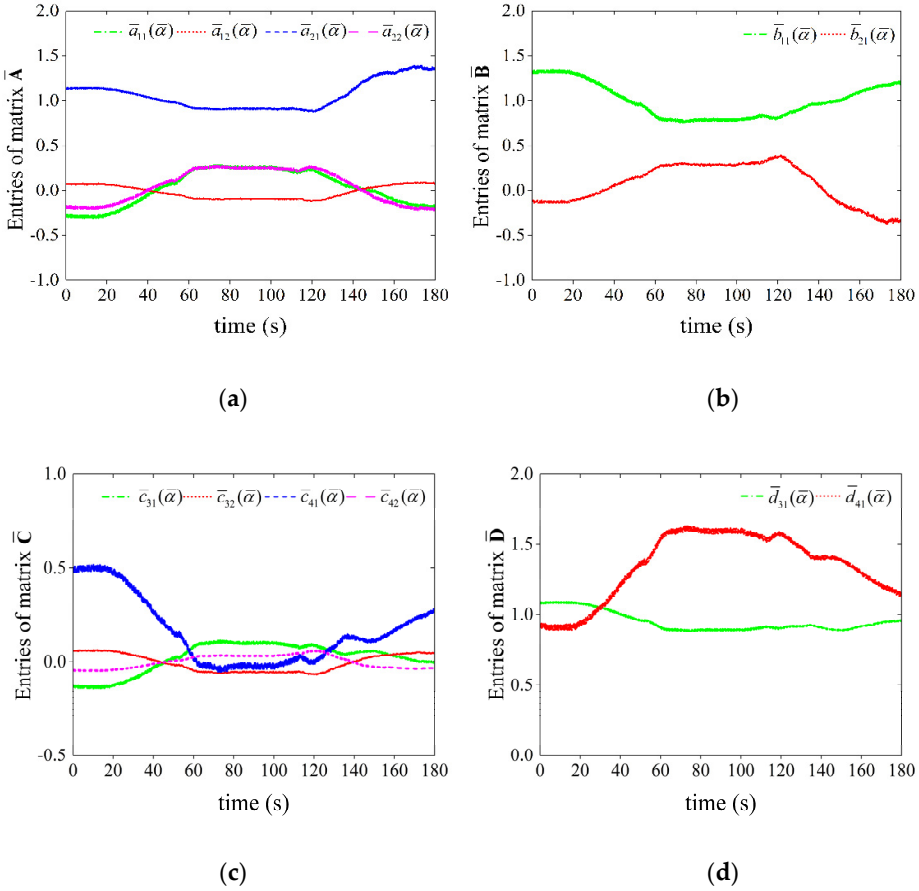

Compressor exit pressure error (%)
time (s)
DD-LPV
EME

(**f**)



(**g**) (**h**)

Figure 7. Prediction comparison of DD-LPV model and EME model to CLM at time $k + 5$, (at 3000 m, with measurement noise) (**a**) Prediction of GT rotor speed. (**b**) Relative error of GT rotor speed prediction. (**c**) Prediction of PT rotor speed. (**d**) Relative error of PT rotor speed prediction. (**e**) Prediction of compressor exit pressure. (**f**) Relative error of compressor exit pressure prediction. (**g**) Prediction of PT inlet temperature. (**h**) Relative error of PT inlet temperature prediction.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. Parameter variations of DD-LPV model (at 3000 m, with measurement noise). (**a**) Entries of matrix $\bar{A}$. (**b**) Entries of matrix $\bar{B}$ (**c**) Entries of matrix $\bar{C}$. (**d**) Entries of matrix $\bar{D}$.

Figure 5 shows that future output prediction at time $k+5$ of the online DD-LPV model was able to match that of the CLM more accurately than the EME model when the altitude was set to 1000 m. The relative predictive errors at time $k+5$ of DD-LPV model rotor speed to that of CLM were less than 0.05% for GT rotor speed and less than 0.01% for PT rotor speed. The maximum errors of compressor exit pressure and PT inlet temperature were less than 0.13% and 0.12% respectively, which meant that the DD-LPV model achieved high accuracy in the prediction of all output parameters. At 50 s and 110 s, when the forward speed switched, the fluctuation ranges of the prediction error were also very small. Meanwhile, as shown in Figure 5, the EME model's relative predictive errors of these four outputs were less than 0.36%, 0.07%, 2.0% and 1.7%, respectively. Although the accuracy of the EME model, whose similarity transformation was computed at the similar inlet pressure and temperature, was still acceptable at 1000 m, its prediction error was still much greater than that of the online identified DD-LPV model.

Figure 6 shows that some of the parameters, such as $\bar{a}_{12}(\bar{\alpha})$, $\bar{a}_{21}(\bar{\alpha})$ and $\bar{b}_{21}(\bar{\alpha})$, changed very slightly, even their corresponding scheduling parameter $\bar{\alpha}(k)=\bar{n}_{\mathrm{g}}(k)$ changed within a relatively wide range. Thus, the output weight $\boldsymbol{\beta}(k)$ of the MLOS-ELM changed accordingly because the parameters of the DD-LPV model were changed not only with the scheduling parameter, but also with the online updated output weight. All the parameters of the LPV model varied smoothly during the forward speed increase and decrease process which guaranteed the DD-LPV model the ability of smooth prediction over the test horizon. ($\bar{c}_{11}(\bar{\alpha})\sim\bar{c}_{22}(\bar{\alpha})$, $\bar{d}_{11}(\bar{\alpha})\sim\bar{d}_{21}(\bar{\alpha})$ are not shown in the figure since they were fixed values of 1 or 0).

It can be seen from Figure 7 that all the future outputs prediction of the online DD-LPV model was able to track that of the CLM at time $k+5$ very well with measurement noise considered. Most of the relative errors fluctuated around the magnitude of 0.2% which was consistent with the magnitude of the measurement noise. The rotor speed prediction error of the LPV model at time instant $k+5$ was less than 0.35%. Most compressor exit pressure prediction errors of the online DD-LPV model at time instant $k+5$ were less than 0.6%, and the maximum error was less than 0.81%. All the GT inlet temperature prediction errors were less than 0.6%. By contrast, the maximum prediction errors of P_3 and T_{44} of the EME model increased greatly, up to 2.62% and 3.17%, respectively, when the flying altitude changed to 3000 m.

Figure 8 shows that the parameters of the DD-LPV model varied in a wider range than that at altitude 1000 m with the same forward speed variation and a much similar scheduling parameter variation range. If the measurement noise was not considered, similar parameter variations resulted, as Figure 6 illustrates. This meant that the changes were aroused by the measurement noise added to the online training data of the MLOS-ELM, which in turn affected the parameters of the DD-LPV model. Even so, all the parameters of the DD-LPV model still changed smoothly during the dynamic process.

To further verify the effectiveness of the proposed online DD-LPV model, online prediction comparations of the DD-LPV model and the EME model (at different altitudes with and without measurement noise) were conducted. The results are listed in Table 1.

Table 1. Online prediction comparations of DD-LPV model and EME model at time $k+5$ in flight envelope.

Altitude	Measurement Noise	Error Type	Prediction Error of DD-LPV Model (%)				Prediction Error of EME Model (%)			
			n_{g}	n_{p}	P_3	T_{44}	n_{g}	n_{p}	P_3	T_{44}
0 m	no	Average error	0.0115	0.0016	0.0415	0.0471	0.1703	0.0222	0.6233	0.7414
		Maximum error	0.0509	0.0092	0.1442	0.2306	0.4203	0.0801	2.5217	1.8090
		Standard deviation	0.0105	0.0016	0.0360	0.0355	0.1139	0.0195	0.6301	0.3995
	yes	Average error	0.1202	0.1125	0.2320	0.1895	0.1974	0.1034	0.6494	0.7525
		Maximum error	0.4447	0.3760	0.9075	0.8425	0.6847	0.2763	2.7619	2.1970
		Standard deviation	0.0847	0.0751	0.1787	0.1512	0.1392	0.0622	0.6236	0.4280

1000 m	no	Average error	0.0071	0.0011	0.0316	0.0287	0.1298	0.0157	0.7899	0.5714
		Maximum error	0.0422	0.0066	0.1210	0.1110	0.3540	0.0664	1.9774	1.6577
		Standard deviation	0.0069	0.0011	0.0282	0.0247	0.0970	0.0145	0.5461	0.4464
	yes	Average error	0.1152	0.1115	0.2240	0.1655	0.1692	0.1016	0.8017	0.5943
		Maximum error	0.3945	0.3721	1.0303	0.6901	0.6595	0.2549	2.3002	1.9504
		Standard deviation	0.0806	0.0733	0.1786	0.1254	0.1270	0.0593	0.5608	0.4489
2000 m	no	Average error	0.0085	0.0012	0.0172	0.0165	0.1367	0.0150	0.9661	1.0330
		Maximum error	0.0398	0.0065	0.0624	0.1024	0.3419	0.0542	2.1097	2.2256
		Standard deviation	0.0079	0.0011	0.0140	0.0185	0.0848	0.0155	0.5126	0.6554
	yes	Average error	0.1122	0.1114	0.1976	0.1512	0.1682	0.1018	0.9677	1.0394
		Maximum error	0.3900	0.3825	0.9232	0.7013	0.6564	0.2521	2.4632	2.5868
		Standard deviation	0.0765	0.0723	0.1573	0.1123	0.1225	0.0601	0.5234	0.6617
3000 m	no	Average error	0.0060	0.0015	0.0230	0.0215	0.1258	0.0154	1.0008	1.4259
		Maximum error	0.0385	0.0086	0.0850	0.1108	0.3158	0.0554	2.2672	2.8193
		Standard deviation	0.0058	0.0014	0.0219	0.0156	0.1028	0.0142	0.4752	0.9727
	yes	Average error	0.1092	0.1072	0.1727	0.1430	0.1679	0.1027	1.0016	1.4306
		Maximum error	0.3454	0.3489	0.8034	0.5814	0.5835	0.2473	2.6232	3.1719
		Standard deviation	0.0720	0.0717	0.1304	0.1036	0.1236	0.0588	0.4954	0.9749
4000 m	no	Average error	0.0086	0.0025	0.0896	0.0307	0.2134	0.0205	1.1080	1.4431
		Maximum error	0.0376	0.0141	0.5005	0.1255	0.5116	0.0837	2.6658	2.4238
		Standard deviation	0.0087	0.0022	0.1255	0.0311	0.1563	0.0209	1.0029	0.6846
	yes	Average error	0.1040	0.1077	0.2251	0.1278	0.2387	0.1018	1.1380	1.4401
		Maximum error	0.3071	0.3531	0.9292	0.5040	0.8281	0.2616	3.0371	2.7334
		Standard deviation	0.0643	0.0710	0.1762	0.0908	0.1754	0.0611	0.9832	0.7082

It can be seen from Table 1 that the online DD-LPV model was able to predict future outputs more accurately than the EME model at different altitudes within the flight envelope.

Without measurement noise considered, the average prediction errors of N_g of the DD-LPV model were between 0.0071% and 0.0115%—much smaller than those of the EME model, which were between 0.1258% and 0.2134%. For N_p, the range of the average errors of DD-LPV model was between 0.0011% and 0.0025%, and was also smaller than that of EME model which was between 0.0150% and 0.0222%. For P_3 and T_{44}, much better prediction results could be achieved by the online DD-LPV model. For P_3, the average errors of DD-LPV model were between 0.0172% and 0.0896% while those of EME model were between 0.6233% and 1.1080%. For T_{44}, the average errors of DD-LPV model were between 0.0165% and 0.0471% while those of EME model were between 0.5714% and 1.4431%. Additionally, the maximum errors of N_g, N_p, P_3 and T_{44} of DD-LPV model were also obviously better than the EME model. Moreover, the performance of output prediction of the DD-LPV model was more stable, with the most standard deviation lower by more than an order of magnitude, compared with the EME model. Furthermore, the DD-LPV model had no obvious prediction error fluctuations when the altitude and the forward flying speed changed. It identified those changes instantly, according to the online gathered data. On the contrary, from 1000 m to 4000 m, the EME model's average prediction errors of P_3 and T_{44} showed a clearly increasing trend from 0.7899% and 0.5714% to 1.1080% and 1.4431%, respectively, mainly contributing to the turbo-shaft engine's poor applicability of similarity transformation.

With measurement noise considered, the average prediction errors of the online DD-LPV model's four outputs N_g, N_p, P_3 and T_{44} were within 0.1202%, 0.1125%, 0.2320%, and 0.1895%, respectively, and the maximum errors were less than 0.4447%, 0.3825%, 1.0303% and 0.8425%, obviously better than the EME model. However, it was notable that the average prediction errors of the EME model were not greatly changed with or without the measurement noise, while, for the DD-LPV model, the prediction errors increased greatly with noise considered. This was mainly because the original error of the DD-LPV model without noise considered was much smaller than the magnitude of the noise. As an online

training method, noise will affect the update of the output weight, which in turn will affect the accuracy of MLOS-ELM and the DD-LPV model derived from it.

6. Conclusions

This paper proposed an online data-driven LPV modeling method for turbo-shaft engines using the online sequential extreme learning machine with an additional multiplying layer (MLOS-ELM). Through simulations conducted in a wide fight envelope, several conclusions were drawn.

First, the outputs of the proposed MLOS-ELM were able to track the outputs of the CLM online precisely, which demonstrated that the MLOS-ELM was able to model the turbo-shaft engine dynamic accurately. The insertion of the multiplying layer made the output of the network become a state space LPV form without affecting the approximation accuracy. Hence, the MLOS-ELM could represent a new thought for data-driven modeling methods that deserves further research. It has the potential to adapt to the properties of other individual engine or nonlinear systems.

When the traditional state space modeling methods of for aero engines were carried out based on applying small perturbations on the CLM, the DD-LPV modeling method based on MLOS-ELM provided a CLM-free online linear model identification method. It was able to avoid the modeling errors of the CLM and enhance the accuracy of the LPV model, especially at the dynamic state.

The DD-LPV modeling method was very efficient and convenient. It was able to derive an accurate LTI state space model instantly at each sampling time with the individual engine online gathered data, which made the LPV model free from the influence of engine degradation and usable as an adaptive LPV model. Additionally, it could lower the mechanism modeling burden of physical systems for linearization purposes.

Differing from the EME model, which needed the similarity transformation to be applied to the whole flying envelope, the proposed DD-LPV model was derived online, and was able to match variations in flying conditions well. Future output prediction simulation results showed that the online DD-LPV model could be used to predict the output of the aero engine over the next several steps much more precisely than the EME model, with or without the measurement noise considered.

Therefore, the proposed modeling method based on MLOS-ELM represents a new and promising data-driven online modeling method that is worthy of further research in control system designs where LPV models are widely used, such as MPC, fault diagnosis, performance seeking control, and so on.

Author Contributions: Conceptualization, Q.L.; methodology, Z.G. and S.P.; software, Z.G.; validation, Z.G., S.P. and Y.L.; formal analysis, Z.G.; investigation, Z.G.; resources, Z.G. and Y.L.; data curation, Z.G. and Q.L.; writing—original draft preparation, Z.G.; writing—review and editing, Z.G., S.P. and Q.L.; visualization, Z.G.; supervision, Q.L.; project administration, W.Z.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Science and Technology Major Project (J2019-I-0010-0010).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lacerda, M.J.; Palhares, R.M. On discrete-time LPV control using delayed Lyapunov functions. *Asian J. Control* **2021**, *23*, 2359–2369.
2. Lao-atiman, W.; Olaru, S.; Diop, S.; Skogestad, S.; Arpornwichanop, A.; Cheacharoen, R.; Kheawhom, S. Linear parameter-varying model for a refuellable zinc–air battery. *R. Soc. Open Sci.* **2020**, *7*, 201107.

3. Zhao, M.; Jiang, C.C.; Tang, X.M.; She, M.H. Interpolation Model Predictive Control of Nonlinear Systems Described by Quasi-LPV Model. *Autom. Control Comput. Sci.* **2018**, *52*, 354–364.
4. Liu, X.; Han, G.; Yang, X. Robust Global Identification of LPV Errors-in-Variables Systems with Incomplete Observations. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, 1–9. https://doi.org/10.1109/TSMC.2021.3071137
5. Yue, X.; Su, B. Predictive Functional Control of Nonlinear Systems Based on Multiple LPV Models. In Proceedings of the 2019 Chinese Automation Congress, Hangzhou, China, 1 November 2019.
6. Atam, E. Advanced Air Path Control in Diesel Engines Accounting for Variable Operational Conditions. *IEEE Access* **2018**, *6*, 42165–42176.
7. Samadzadeh, S.; Mazinan, A.H. LMI-based LPV control strategy considering UAV systems. *Spat. Inf. Res.* **2019**, *27*, 425–431.
8. Hang, P.; Chen, X. Path tracking control of 4-wheel-steering autonomous ground vehicles based on linear parameter-varying system with experimental verification. *Proc. Inst. Mech. Engineers. Part I J. Syst. Control. Eng.* **2021**, *235*, 411–423.
9. Alcalá, E.; Puig, V.; Quevedo, J.; Rosolia, U. Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC). *Control Eng. Pract.* **2020**, *95*, 104270.
10. Lim, J.; Kirchen, P.; Nagamune, R. LPV Controller Design for Diesel Engine SCR Aftertreatment Systems Based on Quasi-LPV Models. *IEEE Control Syst. Lett.* **2020**, *5*, 1807–1812.
11. Gidon, D.; Abbas, H.S.; Bonzanini, A.D.; Graves, D.B.; Velni, J.M.; Mesbah, A. Data-driven LPV model predictive control of a cold atmospheric plasma jet for biomaterials processing. *Control. Eng. Pract.* **2021**, *109*, 104725.
12. López-Estrada, F.R.; Rotondo, D.; Valencia-Palomo, G. A review of convex approaches for control, observation and safety of linear parameter varying and Takagi-Sugeno systems. *Processes* **2019**, *7*, 814.
13. Hoffmann, C.; Werner, H. A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Trans. Control Syst. Technol.* **2014**, *23*, 416–433.
14. Sun, T.; Sun, X.M.; Sun, A. Optimal Output Tracking of Aircraft Engine Systems: A Data-driven Adaptive Performance Seeking Control. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, 1–1. https://doi.org/10.1109/TCSII.2021.3115777.
15. Pang, S.W.; Li, Q.H.; Zhang, H.B. A new online modelling method for aircraft engine state space model, *Chin. J. Aeronaut* **2020**, *33*, 1756–1773.
16. Cox, P.B.; Tóth, R. Linear parameter-varying subspace identification: A unified framework. *Automatica* **2021**, *123*, 109296.
17. Guangdeng, Z.; Yang, D.; Lam, J.; Song, X. Fault-tolerant control of switched LPV systems: A bumpless transfer approach. *IEEE/ASME Trans. Mechatron.* **2021**,1–1. https://doi.org/10.1109/TMECH.2021.3096375.
18. Salt Ducajú, J.M.; Salt Llobregat, J.J.; Cuenca, Á.; Tomizuka, M. Autonomous ground vehicle lane-keeping LPV model-based control: Dual-rate state estimation and comparison of different real-time control strategies. *Sensors* **2021**, *21*, 1531.
19. Rotondo, D.; Nejjari, F.; Puig, V. Quasi-LPV modeling, identification and control of a twin rotor MIMO system. *Control Eng. Pract.* **2013**, *21*, 829–846.
20. Ferranti, F.; Rolai. Modeling of linear parameter-varying systems using interpolation of root macromodels and scaling coefficients. *Mech. Syst. Signal Processing* **2015**, *60*, 836–852.
21. Ferranti, F.; Rolain, Y. A local approach for the modeling of linear parameter-varying systems based on transfer function interpolation with scaling coefficients. In Proceedings of the Instrumentation & Measurement Technology Conference, IEEE, Pisa, Italy, 11–14 May 2015; pp. 606–611.
22. Marcos, A.; Balas. G.J. Development of Linear-Parameter-Varying Models for Aircraft. *J. Guid. Control Dyn.* **2004**, *27*, 218–228.
23. De Caigny, J.; Camino, J.F.; Swevers, J. Interpolating model identification for SISO linear parameter-varying systems. *Mech. Syst. Signal Processing* **2009**, *23*, 2395–2417.
24. Chen, Q.J.; Huang, J.Q.; Pan, M.X.; Lu, F. A Novel Real-Time Mechanism Modeling Approach for Turbofan Engine. *Energies* **2019**, *12*, 3791.
25. Sun, H.B. *Design of Linear Parameter Varying Controllers for Turbo-fan Engines*; Nanjing University of Aeronautics and Astronautics: Nanjing, China, 2018. (In Chinese)
26. Liu, J.F.; Ma, Y.J.; Zhu, L.H.; Zhao, H.; Liu, H.P.; Yu, D.R. Improved. Gain Scheduling Control and Its Application to Aero-Engine LPV Synthesis. *Energies* **2020**, *3*, 5976.
27. Sun, H.B.; Pan, M.X.; Huang, J.Q. Switching control for turbofan engine based on Double-Layer LPV Model. *J. Propuls. Technol.* **2018**, *39*, 2828–2838.
28. Lv, C.K.; Chang, J.T.; Yu, D.R. Feedback Linearized Sliding Mode Control of Turbofan Engine Based on Multiple Input Multiple Output Equilibrium Manifold Expansion Model. *J. Propuls. Technol.* **2021**, *42*, 1681–1689.
29. Liu, Z.H.; Zheng, Q.G.; Liu, M.L.; Hu, C.P.; Zhang, H.B. Research on the Improved Method of Full-Enveloped Acceleration Control Plan for Turbofan Engine. *J. Propuls. Technology*. **2022**, 43, 346-353. https://doi.org/10.13675/ j.cnki.tjjs.200416.
30. Xia, C.J.; Wang, D. Improved correction methods of aircraft engine fan speed based on similarity theory. *J. Aerosp. Power* **2016**, *31*, 941–947.
31. Iannelli, A.; Fasel, U.; Smith, R.S. The balanced mode decomposition algorithm for data-driven LPV low-order models of aeroservoelastic systems. *Aerosp. Sci. Technol.* **2021**, *115*, 106821.
32. Rödönyi, G.; Tóth, R.; Pup, D.; Kisari, Á.; Vígh, Z.; Kőrös, P.; Bokor, J. Data-driven linear parameter-varying modelling of the steering dynamics of an autonomous car. *IFAC-Pap.* **2021**, *54*, 20–26.
33. Chen, Y.; Andrey, B. Data-Driven Linear Parameter-Varying Modeling and Control of Flexible Loads for Grid Services. In Proceedings of the 2020 American Control Conference, Denver, CO, USA, 1–3 July 2020.

34. Yu, M.; Yang, X.; Liu, X. LPV system identification with multiple-model approach based on shifted asymmetric laplace distribution. *Int. J. Syst. Sci.* **2021**, *52*, 1452–1465.
35. Liu, X.; Zhang, L.; Luo, C. Model reference adaptive control for aero-engine based on system equilibrium manifold expansion model. *Int. J. Control.* **2021**. https://doi.org/10.1080/00207179.2021.2016979.
36. Lv, C.; Wang, Z.; Dai, L.; Liu, H.; Chang, J.; Yu, D. Control-Oriented Modeling for Nonlinear MIMO Turbofan Engine Based on Equilibrium Manifold Expansion Model. *Energies*. **2021**, *14*, 6277.
37. Zhu, L.; Liu, J.; Ma, Y.; Zhou, W.; Yu, D. A Corrected Equilibrium Manifold Expansion Model for gas Turbine System Simulation and Control. *Energies* **2020**, *13*, 4904.
38. Tóth, R.; Laurain, V.; Zheng, W.X.; Poolla, K. Model structure learning: A support vector machine approach for LPV linear-regression models. In Proceedings of the IEEE Conference on Decision and Control, Orlando, FL, USA, 12–15 December 2011; pp. 3192–3197.
39. Feng, K.; Lu, J.G.; Chen, J.S. Identification and model predictive control of LPV models based on LS-SVM for MIMO system. *CIESC J.* **2015**, *66*, 197–205.
40. Cavanini, L.; Ferracuti, F.; Longhi, S.; Monteriù, A. LS-SVM for LPV-ARX Identification: Efficient Online Update by Low-Rank Matrix Approximation. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020.
41. Fényes, D.; Németh, B.; Gáspár, P. A Novel Data-Driven Modeling and Control Design Method for Autonomous Vehicles. *Enegies* **2021**, *14*, 517.
42. Gu, N.N.; Wang, X.; Lin, F.Q. Design of Disturbance Extended State Observer (D-ESO)-Based Constrained Full-State Model Predictive Controller for the Integrated Turbo-Shaft Engine/Rotor System. *Energies* **2019**, *12*, 4496.
43. Ballin, M.G. *A High-Fidelity Real-Time Simulation of a Small Turboshaft Engine*; National Aeronautics and Space Administration, Ames Research Center: Mountain View, CA, USA, 1988.
44. Zheng, Q.; Xu, Z.; Zhang, H.; Zhu, Z. A turboshaft engine NMPC scheme for helicopter autorotation recovery maneuver. *Aerosp. Sci. Technol.* **2018**, *76*, 421-432.
45. Zhang, M.; Lin, Z.; Huang, H.; Zhang, T. Design and verification of model predictive control for micro-turboshaft engine. *Adv. Mech. Eng.* **2019**, *11*, 1687814019890198.
46. Wang, Y.; Zheng, Q.; Zhang, H.; Chen, H. Study on inversion control for integrated helicopter/engine system with variable rotor speed based on state variable model. *Int. J. Turbo Jet-Engines* **2020**, 000010151520200018. https://doi.org/10.1515/tjj-2020-0018.
47. Lin, Z.X. *Design and Verification of a Micro Turboshaft Engine Control System*; Nanjing University of Aeronautics and Astronautics: Nanjing, China, 2019.
48. Gu, N.; Wang, X. Model predictive controller design based on the linear parameter varying model method for a class of turboshaft engines. In Proceedings of the 2018 Joint Propulsion Conference, Cincinnati, OH, USA, 9–11 July 2018; p. 4620.
49. Liang, N.; Huang, G.; Saratchandran, P.; Sundararajan, N. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Trans. Neural Netw.* **2006**, *17*, 1411–1423.
50. Yan, C.K.; Zhou, X.; Zhang, H.B. Design and simulation of a control scheme for turbo-shaft engine in helicopter autorotation training process, *J. Aerosp. Power* **2014**, *29*, 1744–1751.
51. Jianguo, S.; Vasilyev, V.; Ilyasov, B. *Advanced Multivariable Control System of Aeroengines*; Beijing University of Aeronautics and Astronautics: Beijing, China, 2005.