

Article

Nonlinear Affine System Identification and Feedforward–Feedback Control for Turbofan Engines Based on Polynomial Feature Enhanced Multi-Layer Perceptron

Pengpeng Li ^{1,2}, Penghui Sun ^{2,*} and Fengling Zhang ^{1,*} ¹ School of Aero-Engine, Shenyang Aerospace University, Shenyang 110136, China; blueskysophia@sina.com² State Owned Chuanxi Machinery Factory, Chengdu 610041, China

* Correspondence: ph_sun@outlook.com (P.S.); fling707@163.com (F.Z.)

Abstract

Turbofan engines exhibit complex nonlinear dynamics across the entire flight envelope, which cannot be captured by explicit mathematical models, posing significant challenges for engine controller design. Traditional control designs often rely on multiple linearized models covering the entire operating range, which require designing multiple linear controllers and gain-scheduling strategy to switch these sub-controllers. This article aims to establish a global nonlinear affine model-based feedforward control method for turbofan engines, and the control input can be efficiently obtained through simple algebraic calculations with given reference signal. In this method, the nonlinear affine model via a multi-layer perceptron (MLP) combined with polynomial feature expansion is constructed based on engine model simulation data. With this affine model, the need for the cumbersome design process of multiple linear controllers and their sub-controller switching can be avoided. Additionally, a Proportional–Integral (PI) feedback controller is integrated with the feedforward controller to eliminate tracking errors caused by model mismatches and disturbances. Numerical simulation results verify that the proposed MLP model has high identification accuracy, and the composite control strategy can simplify the control design. During the acceleration/deceleration process between intermediate and idle state, the high-pressure rotor speed overshoot is less than 0.01%, and the settling time is less than 2.6 s, outperforming the 4.1 s of the gain-scheduled PI controllers.

Keywords: turbofan engine; inversion system-based feedforward control; nonlinear affine system; polynomial feature expansion; Salp Swarm Algorithm

1. Introduction

As a type of aircraft engines, turbofan engine is a complex thermodynamic system with nonlinear dynamics across the entire flight envelope and operating range. To achieve safety and fast thrust response across a wider flight envelope, the engine controller, which prevents the engine from surge, over-temperature, and overspeed while minimizing the transient response time, plays an essential role during engine operation [1–3]. However, as aerodynamic coupling exists between different components and the component operating parameters cannot be calculated directly, an accurate engine model cannot be obtained. Hence, the engine controller design is becoming more challenging [4]. As an alternative, linearized models, which approximate the engine nonlinear dynamics around steady-state points, are employed for controller design. In recent decades, many classical control



Received: 11 May 2026

Revised: 1 June 2026

Accepted: 2 June 2026

Published: 21 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

methods and advanced control theories have been used to design controller for increasingly complex turbofan engines based on a set of linear models, aiming to improve thrust response [2,5–8].

However, a series of linear models, which works well in certain operating range, has to sacrifice control performance to enhance robustness and system stability [9,10]. A linear model cannot address the fundamental limitation that it is only valid in the vicinity around specific operating points. A further limitation of such methodologies is the requirement for gain-scheduling or sub-controller switching mechanisms to enable transitions between different sub-controllers throughout the engine operating range. The gain-scheduling or sub-controller switching mechanism can degrade the control performance, and the closed-loop system stability can only be proved under specific operating conditions [11,12].

Hence, the sub-controller gain-scheduling or switching logic design is quite important. The logic requires selection of appropriate strategy and parameters, and the system output chattering caused by frequent switching should be avoided. Therefore, switching action complicates the engine control system design. Furthermore, several advanced control methods, such as sliding mode control and H_∞ control, are not applicable to switching between different sub-controllers due to their structural characteristics, which may directly result in closed-loop system instability [12]. These factors constraint the application of advanced methods. Additionally, the control input calculation of feedback control relies on the deviation between reference signal and the measured system output. However, for a system with strong nonlinear dynamics, feedback control struggles to provide fast response speed [13–15].

Therefore, the feedforward control is considered for the design of turbofan engine controller, as it leverages the advantages of predictive control while enabling multivariable control along the reference signal of system output [16–18]. In this method, feedforward control computes the control input signal under given system output trajectory by precise mathematical dynamic models. Feedforward control possesses strong predictive capability and anticipatory nature, which can significantly improve thrust response [19–21]. In addition, feedforward control methods based on models have been used for other nonlinear system, such as nonlinear time-delay system, non-minimum phase systems, and selective catalytic reduction denitrification systems [22], to improve control performance. Martin Grotjahn has implemented a feedforward control method based on robot dynamic model for the robot control system of the KUKA KR15 industrial robot, achieving precise pose control under certain operating conditions [23]. Furthermore, neural networks and a deep learning method have been integrated with feedforward control strategies to design feedforward controllers [24].

For turbofan engine with strong nonlinear dynamics, feedforward controller can enhance the thrust response by calculating the control input based on the reference signal and the nonlinear system model [20,21]. The short take-off and vertical landing (STOVL) propulsion system of the F-35B aircraft employs a nonlinear dynamic inversion (NDI) control methodology, which is a multivariable control approach incorporating characteristics of both dynamic inversion and predictive control, enabling 5-DOF thrust control [18,19]. In addition, feedforward controller can be integrated with feedback controller to create a complementary control strategy. The response delay caused by pure feedback controller can be compensated by feedforward strategy, and the system disturbance rejection, control accuracy, and robustness under complex operating conditions can be enhanced by feedback control. A feedforward–feedback control method is proposed by Gao to improve the stability of the stick–slip piezoelectric actuator, and the linearity of the actuator’s displacement output has been effectively improved [25]. Cheng has proposed an adaptive feedforward control strategy based on Radial Basis Function neural network for turboshaft engine

to enhance its disturbance rejection capability in conventional integrated control methods [26]. The feedforward–feedback composite control strategy integrates the advantages of feedforward controller’s rapid response and feedback controller’s strong disturbance rejection capability.

This article aims to establish a nonlinear affine model for turbofan engines using polynomial feature augmented MLP model and operational data. In this method, polynomial feature expansion is employed to convert the strong nonlinearity of complex systems into weak nonlinearity in a higher-dimensional space. Accordingly, the nonlinear engine model can be accurately approximated by an MLP model with fewer neurons. MLP generates an approximate engine model with a nonlinear affine system structure, and this structure generally cannot be directly obtained from physical modeling of turbofan engines. Then, the feedforward control input can be obtained from the nonlinear affine model and give reference signal without heavy complex computations, which improves the turbofan engine thrust response. A series of polynomial feature terms are introduced and employed to expand the dimensionality of MLP inputs, and improve the improve its fitting ability for turbofan engines. Meanwhile, Principal Component Analysis (PCA) and Bayesian Information Criterion (BIC) are used to determine the optimal terms of the polynomial feature expansion. Based on the selected feature expansion terms and the MLP model, an optimization algorithm combining the Salp Swarm Algorithm (SSA) with Sequential Quadratic Programming (SQP) has been employed to determine the optimal polynomial feature terms. The multi-loop PI controller is tuned through the direct synthesis method. Finally, the proposed feedforward controller and feedforward–feedback controller are verified through a numerical simulation of turbofan engine model.

This paper is organized as follows: In Section 2, a nonlinear model of dual-spool turbofan engine nonlinear model is presented. In Section 3, a polynomial feature expansion-based MLP model for turbofan engine is constructed, and the nonlinear affine model-based feedforward controller design method is given. In Section 4, numerical simulations are applied to verify the effective of the feedforward–feedback control method. Finally, conclusions are given in Section 5.

2. Turbofan Engine and Component-Level Model

The structure of a twin-spool turbofan engine is illustrated in Figure 1, which consists of a fan, a compressor, a combustor, a high-pressure turbine (HPT), a low-pressure turbine (LPT), a mixer, and a nozzle. In this engine, the low-pressure shaft and high-pressure shaft are connected to a compressor component for air compression and a turbine for power generation, respectively. The compressors and turbines, which are fixed on two different shafts, are aerodynamically coupled. The air entering the fan is compressed and then flows separately into the engine core and bypass duct. The core airflow is further compressed and mixes with fuel and combusts, and the resulting high-temperature, high-pressure gas sequentially drives the HPT and LPT to rotate, and thereby drives the compressors. The bypass airflow enters the mixer, and then mixes with the core stream and is expelled through the nozzle to produce thrust. Hence, a component-level model for turbofan engine can be established.

According to the operating cycle, component maps, and other parameters, a nonlinear component-level turbofan engine model incorporating the aforementioned major components can be developed. Based on the given control input parameters, the working process of turbofan engine components can be calculated sequentially according to the computation flowchart shown in Figure 1 [25].

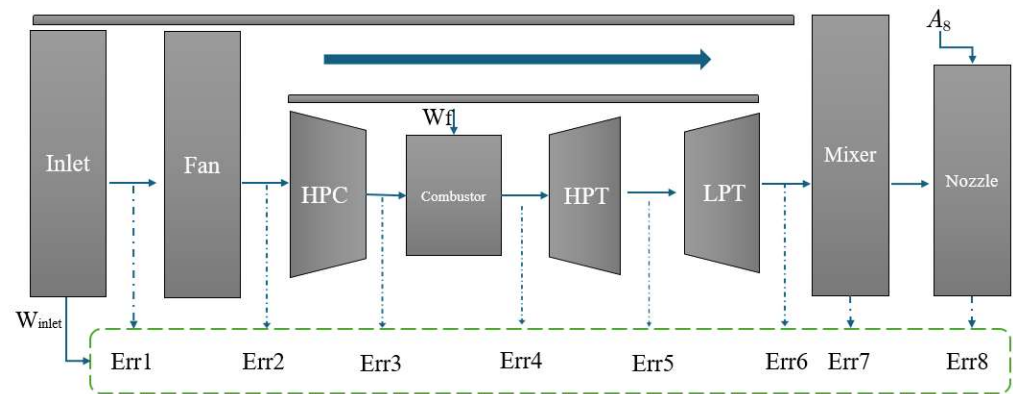


Figure 1. Turbofan engine structure and its component-level model.

$$\begin{cases} f_1(\mathbf{x}) = (W_{\text{inlet}} - W_{\text{fan},f} - W_{\text{fan},a}) / W_{\text{inlet}} = \text{Err}_1 \\ f_2(\mathbf{x}) = (W_{\text{fan},a} - W_{\text{HPC}}) / W_{\text{HPC}} = \text{Err}_2 \\ f_3(\mathbf{x}) = (W_{\text{comb}} - W_{\text{HPC}}) / W_{\text{HPC}} = \text{Err}_3 \\ f_4(\mathbf{x}) = (W_{\text{comb}} - W_{\text{HPT}}) / W_{\text{HPT}} = \text{Err}_4 \\ f_5(\mathbf{x}) = (W_{\text{HPT}} - W_{\text{LPT}}) / W_{\text{LPT}} = \text{Err}_5 \\ f_6(\mathbf{x}) = (W_{\text{mix}} - W_{\text{duct}} - W_{\text{LPT}}) / W_{\text{mix}} = \text{Err}_6 \\ f_7(\mathbf{x}) = (P_6 - P_{16}) / P_6 = \text{Err}_7 \\ f_8(\mathbf{x}) = (P_8 - P_{\text{nozzle}}) / P_8 = \text{Err}_8 \end{cases} \quad (1)$$

where W_{inlet} denotes the inlet air flow, $W_{\text{fan},f}$ and $W_{\text{fan},a}$ represent the fan outer duct airflow and core airflow, W_{HPC} refers to the compressor inlet flow, W_{comb} denotes the main combustion chamber inlet flow, W_{HPT} denotes the HPT inlet flow, and W_{LPT} is the LPT inlet flow. L_{HPC} and L_{fan} refer to the power of the HPC and fan, respectively. N_2 , N_1 denote the compressor rotor speed and fan rotor speed, respectively, P_6 , P_{16} are the inner duct outlet static pressure and outer duct outlet static pressure, respectively, and P_8 , P_{nozzle} indicate the total pressure before the nozzle and total pressure after the nozzle, respectively.

The dynamic of the two-spool turbofan engine model can be expressed as Equation (2).

$$\begin{cases} \frac{dN_2}{dt} = \frac{(30/\pi)^2}{I_H \cdot N_2} (\eta_H \cdot L_{\text{HPT}} - L_{\text{HPC}} + L_{\text{add}}) \\ \frac{dN_1}{dt} = \frac{(30/\pi)^2}{I_L \cdot N_1} (\eta_L \cdot L_{\text{LPT}} - L_{\text{fan}}) \end{cases} \quad (2)$$

where N_1 , N_2 denote the fan speed and HP rotation speed, respectively, I_H and I_L are the rotational inertia of the HP rotor and LP rotor, respectively, L_{HPT} and L_{HPT} denote the power of the HPT and LPT, respectively, L_{add} denotes the additional power of engine starting, and η_H , η_L denote the mechanical efficiency of HP shaft and LP shaft, respectively.

The nonlinear functions $f_i(z)$, $i \in \{1, 2, \dots, 8\}$ denote the matching equation of different component in Figure 1, and $z = [z_1, z_2, \dots, z_8]$ denotes the iterative variable in this model [25]. According to the work process of each component, iterative computation is carried out according to the initial guess values of the selected iterative variables until the norm of errors in Equation (1) is less than a specified threshold ε (where ε is typically set to 1×10^{-6}). In other words, the matching equation in Equation (1) is used to ensure component matching among different engine component.

To obtain the data-driven model that relies on engine simulation data, simulation data are first generated using this component-level model for turbofan engines. The data encompass engine operating conditions and control inputs at various engine stations.

3. Feedforward Control Based on Nonlinear Affine Model

3.1. Inversion System-Based Feedforward Control

As engine thrust cannot be directly measured during flight, fan speed N_1 and engine pressure ratio (EPR) can be selected as controlled variables to represent the engine operating performance. Hence, the system state, control input, and system output can be selected as $x = [x_1, x_2]^T = [N_1, N_2]^T$, $u = [W_f, A_8]^T$, and $y = [N_1, EPR]^T$, respectively. Hence, the turbofan engines can be represented by the nonlinear model in Equation (3).

$$\begin{aligned}\dot{x} &= f(x, u) \\ y &= h(x, u)\end{aligned}\quad (3)$$

where $x \in R^n$, $y \in R^p$, $u \in R^m$ denote system state, output, and control input, respectively, and $f(\cdot)$, $h(\cdot)$ denote nonlinear function with appropriate dimensions.

For the nonlinear system presented in Equation (3), the stable inversion problem can be defined as follows:

Stable Inversion Problem [26]: Given an output trajectory $y_d \in L_1 \cap L_\infty$, the stable inversion problem is considered solvable if a control input and a continuous state trajectory exist satisfying the following three conditions: (1) u_d, x_d satisfy the differential equation $\dot{x}_d = f(x_d, u_d)$; (2) accurate output tracking $y_d = h(x_d, u_d)$ can be achieved by the control input; (3) the control input and state trajectory are bounded and satisfy the Lipschitz continuity condition $\lim_{t \rightarrow \pm\infty} u_d(t) \rightarrow 0$, $\lim_{t \rightarrow \pm\infty} x_d(t) \rightarrow 0$.

The system state trajectory and control input satisfying the aforementioned conditions are referred to as the desired state trajectory and the nominal control input, respectively. Furthermore, the nominal control input can be utilized as the feedforward control input. For nonlinear systems described by explicit algebraic equations, the stable inversion problem can be solved via its inversion system, which yields the bounded desired system state trajectory x_d and the control input u_d corresponding to the desired output trajectory. For a kind of nonlinear model of actual physical systems, it is assumed that the system does not contain unstable states, and the control input can be obtained according to Equation (4) based on the desired output trajectory and the given system state.

$$u_d(t) = f^{-1}\left(h^{-1}(y_d), \dot{x}_d\right) \quad (4)$$

However, for a nonlinear system whose inverse system cannot be directly obtained by solving algebraic equations, the nominal control input satisfying the desired trajectory cannot be derived.

Due to the unavailability of accurate nonlinear models for engine in Equation (3), deriving its inverse system is intractable. To address this problem, an approximate model for the turbofan engine can be derived based on the operation data.

$$\begin{aligned}\dot{\hat{x}} &= \hat{f}_\sigma(x, u, z) \\ \hat{y} &= \hat{h}_\sigma(x, u)\end{aligned}\quad (5)$$

where $\hat{x}$, $\hat{y}$ denote the estimated value of system state derivative and system output, respectively, the $\hat{f}_\sigma(\cdot)$, $\hat{h}_\sigma(\cdot)$ denote the approximation of nonlinear function $f(\cdot)$, $h(\cdot)$, respectively. To simplify the governing equation, the discrete-time state-space equation is employed, as shown in Equation (6).

$$\begin{aligned}\hat{x}_{k+1} &= \hat{f}(x_k, u_k) \\ \hat{y}_k &= \hat{h}(x_k, u_k)\end{aligned}\quad (6)$$

where $\hat{x}_{k+1}$ denotes the next system state, $\hat{y}_k$ denotes the output variable, and x_k, u_k denote the current system state and control input, respectively.

In view of the operating characteristics of turbofan engines, the system state and output maintain specific values when the control input is held constant. For the transient process between different operating points, the engine regulation time can be set by defining the control input trajectory. Consequently, the calculation of the feedforward control input can be transformed to an algebraic problem by solving Equation (7).

$$\hat{x}_{k+1} = \hat{f}(\hat{h}^{-1}(y_k, u_k), u_k) + \epsilon_i = F(x_k, u_k) + \epsilon_i \quad (7)$$

where $F(\cdot)$ denotes nonlinear vector function with suitable dimension, ϵ_i denotes the uncertainty between nonlinear system and the approximation model.

Hence, the feedforward control input under a given system output trajectory can be formulated as Equation (8).

$$u_{d,k}^f = F^{-1}(y_{d,k}, x_{d,k+1}) \quad (8)$$

By transforming the nonlinear system in Equation (8) into the nonlinear affine system structure shown in Equation (9), the feedforward control input can be directly calculated by Equation (10), where $g_2^{-1}(\cdot)$ denotes the inverse matrix of the matrix function $g_2(\cdot) \in R^{2 \times 2}$.

$$x_{d,k+1} = g_1(x_k) + g_2(x_k) \cdot u_k \quad (9)$$

$$u_k^{ff} = g_2^{-1}(x_k) \cdot (x_{d,k+1} - g_1(x_k)) \quad (10)$$

where u_k^{ff} denotes the feedforward control input signal at the k control loop.

In this paper, an MLP model is utilized to approximate the nonlinear engine system over a specific operating range, and the MLP input and output are selected as $X = [x_k, u_k]^T$ and $Y = x_{k+1}$, respectively.

3.2. Multi-Layer Perceptron Model Based on Polynomial Feature Expansion

The calculation of the feedforward control input in Equation (10) relies on the engine nonlinear model; however, the accurate explicit expression of the nonlinear model in Equation (3) is practically infeasible. To address this problem, an MLP model is employed as a nonlinear function approximator to fit the unknown autonomous dynamics term $g_1(x)$ and the input gain matrix term $g_2(x)$ of the nonlinear affine system in Equation (9). It approximates the mapping of $g_1(x)$ and $g_2(x)$ solely through the current system state x_k . The MLP consists of an input layer, hidden layers, and an output layer, with fully connected architectures between layers. By employing nonlinear activation functions to approximate complex nonlinear mappings, it achieves high-precision fitting of the nonlinear dynamics x_k relying only on input $\{u_k, x_k\}$. Hence, the structure of the MLP model can be simplified.

Furthermore, to approximate the nonlinear systems, the polynomial feature expansion is employed to map original low-dimensional input data into a high-dimensional polynomial feature space. For a two-dimensional state system with an original state vector $x \in R^{2 \times 1}$, polynomial feature expansion generates a high-dimensional feature vector by constructing various powers and cross-terms of the original features. This process captures the nonlinear coupling relationships between different variables, providing richer feature information for the MLP model. The advantages of polynomial feature expansion lie in its simplicity, strong interpretability, and low computational complexity. The basic functions of polynomial feature expansion are expressed as Equation (11).

$$\phi(x) = [f_1(x), f_2(x), f_3(x), \dots, f_k(x)]^T \quad (11)$$

By incorporating operating data into approximate the nonlinear function $g_1(x)$ and $g_2(x)$, the approximation functions can be expressed as Equation (12).

$$\begin{aligned}\hat{g}_1(x) &= \sum_{i=1,2,3,\dots}^k f_1(\theta_{1,i} \cdot \phi_i(x) + b_{3,i}) \\ \hat{g}_2(x) &= \sum_{i=1,2,3,\dots}^k f_2(\theta_{2,i} \cdot \phi_i(x) + b_{4,i})\end{aligned}\quad (12)$$

Assume that the input matrix and output matrix of the training set with Q samples and p -dimensional features are X and Y respectively.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,Q} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,Q} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p,1} & x_{p,2} & \cdots & x_{p,Q} \end{bmatrix}_{p \times Q}, Y = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,Q} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,Q} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m,1} & y_{m,2} & \cdots & y_{m,Q} \end{bmatrix}_{m \times Q}$$

Using the normalized polynomial features as the input of MLP model, the mathematical expression $y = f(w^T x + b)$ is employed in the model, where w , x and b denote the weight vector, input vector, and bias, respectively, and $f(\cdot)$ represents the activation function. Consequently, the MLP output can be calculated as Equation (13).

$$T = \begin{bmatrix} t_1 & t_2 & \cdots & t_Q \end{bmatrix}_{m \times Q} \quad (13)$$

where

$$t_j = \begin{bmatrix} t_{1,j} \\ t_{2,j} \\ \vdots \\ t_{m,j} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^l \beta_{i,1} \cdot f(w_i^T x_j + b_i) \\ \sum_{i=1}^l \beta_{i,2} \cdot f(w_i^T x_j + b_i) \\ \vdots \\ \sum_{i=1}^l \beta_{i,m} \cdot f(w_i^T x_j + b_i) \end{bmatrix}$$

To simplify the equation, Equation (13) can be transformed to Equation (14).

$$H\beta = T^T \quad (14)$$

where,

$$H = \begin{bmatrix} f(w_1^T x_1 + b_1) & f(w_2^T x_1 + b_2) & \cdots & f(w_l^T x_1 + b_l) \\ f(w_1^T x_2 + b_1) & f(w_2^T x_2 + b_2) & \cdots & f(w_l^T x_2 + b_l) \\ \vdots & \vdots & \ddots & \vdots \\ f(w_1^T x_Q + b_1) & f(w_2^T x_Q + b_2) & \cdots & f(w_l^T x_Q + b_l) \end{bmatrix}$$

Hence, in order to determine the connection weights β and hidden layer parameters w, b of the MLP such that the model output aligns as closely as possible with the engine simulation data, the parameter estimation problem of β, w, b can be formulated as the optimization problem shown in Equation (15).

$$\min_{\theta, w, b} \left\{ \sum_{i=1,2,\dots,Q} (\|x_{i+1} - \hat{g}_1(x_i) - \hat{g}_2(x_i)\|) \right\} \quad (15)$$

An MLP model can be trained by engine operating data to approximate the nonlinear functions $g_1(x)$ and $g_2(x)$. The suitable k_{poly} (the terms of polynomial feature) can be selected through Principal Component Analysis (PCA) and Bayesian Information Criterion (BIC). Hence, the fitting accuracy can be ensured by effective features and MLP model with simple structure.

Firstly, the PCA is employed to extract the most significant features from a list. The first few principal components that account for the vast majority of data variability should be selected, and the feature dimensionality can be reduced. The steps for PCA dimensionality reduction are as follows:

1. Calculate the covariance matrix $S = 1/(n - 1) \cdot XX^T$ based on the training data and the selected basis functions;
2. Perform eigenvalue decomposition on the covariance matrix S to obtain the eigenvalues λ_i and their corresponding eigenvectors v_i , for $i = 1, 2, \dots, m$;
3. Sort the eigenvalues in descending order and select the polynomial features corresponding to the first k_{PCA} largest eigenvalues.

Then, the BIC calculation shown in Equation (16) is employed to validate the selected polynomial feature terms. The effectiveness of the selected polynomial features could be verified, and the terms can be expanded according to the BIC calculation to enhance the accuracy of the MLP model. Hence, the suitable number of polynomial feature would be defined as k_{poly} .

$$BIC = k_b \cdot \ln(n) - 2 \cdot \ln(\hat{L}) \quad (16)$$

where $\ln(\hat{L})$ denotes the log-likelihood of the model, n denotes the number of training instances, and k_b denotes the number of the parameters to be estimated.

3.3. Hybrid Optimization Strategy Based on Salp Swarm Algorithm

In the MLP approximation of nonlinear systems in Equation (9), the objective function often exhibits multimodal, non-convex, and high-dimensional characteristics. Consequently, single gradient-based algorithms are prone to getting trapped in local optima. To address this challenge, a hybrid optimization strategy is employed to optimize the objective function Equation (15). Proposed by Mirjalili et al. [27] in 2017, the Salp Swarm Algorithm (SSA) offers distinct advantages over other algorithms, such as Particle Swarm Optimization (PSO). When solving complex optimization problems, SSA can effectively escape local optima and achieve superior global parameter optimization. To mathematically model the foraging process, the population is divided into two groups: the leader and the followers. The leader is the salp located at the front of the chain, while the rest serve as followers [27].

In each iteration, the leader performs global exploration, moving toward the food source in a chain-like manner and guiding the followers, while the followers focus on intensive local exploration. This mechanism significantly reduces the likelihood of getting trapped in local optima. Consequently, SSA possesses robust capabilities in both local and global search and has been widely applied to various engineering problems.

During the optimization process, the population is divided into leader and followers.

The position of salps in an n -dimensional search space, where n is the number of variables in a given problem, can be defined firstly. The position of all salps in a two-dimensional matrix x can be stored, and the food source can be designated as F in the search space. In the salp chain, salps are divided into leader salp and follower salps. The leader moves toward the food and guides the movement of the followers. The leader is responsible for global exploration, while the followers are responsible for local exploitation.

Position updating mechanism—leader salp: The leader salp is located at the front of the salp chain and is responsible for exploring new food sources (i.e., the global optimum). The position update law is as Equation (16), and the leader salp only updates its position with respect to the food source.

$$x_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j), & c_3 \geq 0.5 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j), & c_3 < 0.5 \end{cases} \quad (17)$$

where x_j^1 denotes the position of the leader salp in the j -th dimension, and F_j denotes the position of the food source in the j -th dimension, and ub_j , lb_j denotes the upper and lower bounds of the j -th dimension, respectively. c_1 can be updated as Equation (18) to balance the exploration and exploitation process of the salp swarm algorithm, and c_2, c_3 are random numbers uniformly selected from in the set $[0, 1]$.

$$c_1 = 2 \cdot e^{-\left(\frac{4n}{L_{max}}\right)^2} \quad (18)$$

where n, L_{max} denote current iteration numbers and maximum iteration numbers, respectively.

The follower salps update their position with respect to each other. The follower salps' position can be updated by Newton's laws of motion $x_j^i = 1/2 \cdot at^2 + v_0t$, where t, v_0, a denote time, initial speed, and acceleration, respectively. When v_0 is considered to be zero, the position update of the followers can be conducted as Equation (19).

$$x_j^i = \frac{1}{2} (x_j^i + x_j^{i-1}), \quad i \geq 2 \quad (19)$$

where x_j^i denotes the position of the i -th follower salp in the j -th dimension.

As shown in Figure 2, a hybrid optimization strategy, which integrates the global exploration capability of the Salp Swarm Algorithm with the refined exploitation ability of local search algorithms, is proposed to optimize the parameters in the MLP model.

The core philosophy is as follows: initially, SSA is employed to conduct a coarse search across the vast solution space to rapidly locate the basin of attraction for the global optimum. Subsequently, the optimal solution identified by SSA serves as the initial value for the local search algorithm to perform a fine search, thereby achieving fast convergence with high precision.

The parameters of the MLP model, which approximate the nonlinear system in Equation (2), solved using the SSA algorithm, are denoted by θ_{SSA} . Following the global exploration phase of SSA, the algorithm proceeds to the local search stage. During this stage, gradient-based optimization algorithms is leveraged to conduct precise search within the θ_{SSA} solution neighborhood, which is formulated as a nonlinear least squares optimization problem with the objective of minimizing the squared residual sum between MLP predictions and the actual Y matrix. It means that the optimal solution generated by SSA is adopted as the initialization parameter of gradient-based optimization to circumvent potential convergence failures and local optima traps associated with random initialization. The optimized weight coefficients are derived as presented in Equation (20).

$$\min_{\theta} S(\theta) = \|x_{k+1} - f_{\theta}(x_k, u_k)\| \quad (20)$$

where θ denotes the parameters of the MLP model to be optimized.

Subsequently, the feedforward control input can be computed based on the fitted MLP model, as illustrated in Equation (21).

$$u_k^{ff} = \hat{g}_2^{-1}(x_k) \cdot (y_{d,k+1} - \hat{g}_1(x_k)) \quad (21)$$

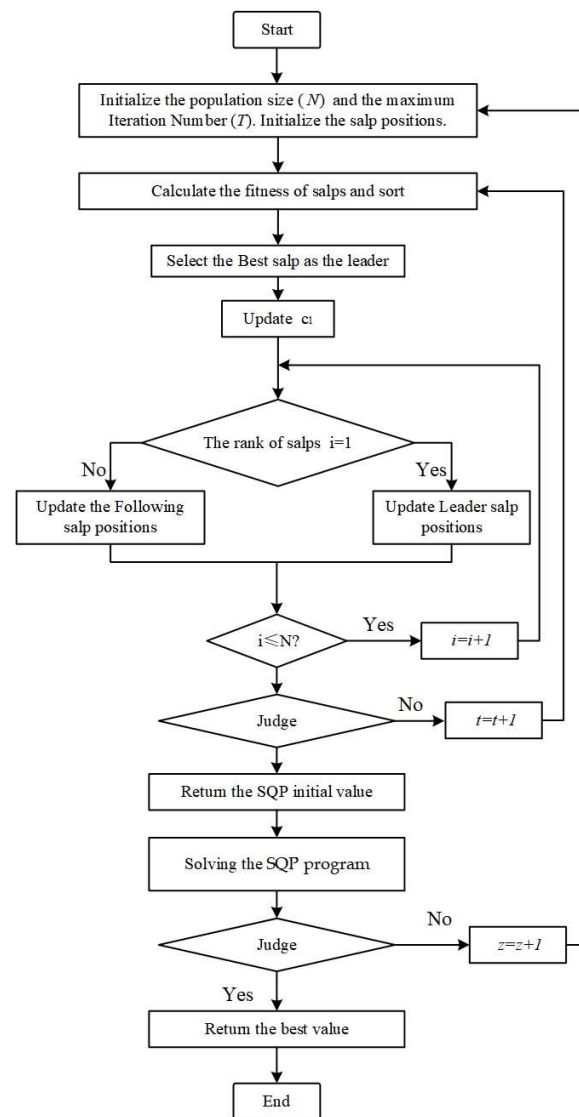


Figure 2. The flowchart of hybrid optimization strategy.

4. Multi-Loop PI Controller Design Using Direct Synthesis Approach

Due to the inevitable discrepancy between the MLP model and the actual nonlinear system in Equation (2), as well as the disturbances encountered during engine operation, feedforward control alone is insufficient to completely eliminate the tracking error. Simultaneously, a multi-loop PI controller as shown in Figure 3 is employed to eliminate tracking errors and suppress disturbances. Hence, the model uncertainties can be addressed.

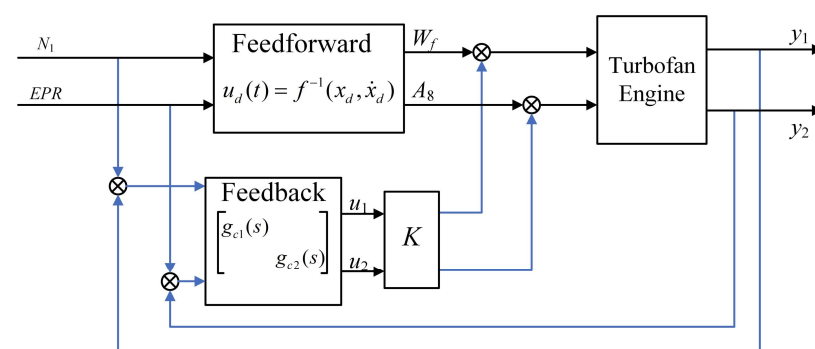


Figure 3. The feedforward control combined with multi-loop PI controller.

A multi-loop PI controller can be derived using direct synthesis method by inversely solving the closed-loop relationship between the desired closed-loop system dynamics and the plant model. For multi-loop control systems with severe coupling between different control loops, decoupling or reduction of coupling degree must be performed under the action of a pre-compensator K , resulting in a weakly coupled controlled system as shown in Equation (22). By reducing the coupling effects between different control loops, the multi-loop system is decomposed into multiple single-loop systems. Controllers are then designed separately for each loop using the direct synthesis method, and finally integrated into a multi-loop PI control system.

$$G_p(s) = G(s) \cdot K \quad (22)$$

The core of the Internal Model Control (IMC) method lies in introducing an ‘internal model’ of the controlled plant (i.e., an approximate model of the plant transfer function), transforming controller design into the matching design between the internal model and the feedback loop. For PI controllers, the IMC method can directly derive tuning formulas for the proportional gain K_p and integral time T_i in a PI controller $u(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot \int_0^t e(\tau) d\tau$ based on the plant model. PI controllers are designed separately for each control loop, thereby forming a decentralized control system. The multi-loop PI feedback controllers can be expressed as Equation (23).

$$G_c(s) = \text{diag}\{g_{c,i}(s)\}, i = 1, 2, \dots, p \quad (23)$$

The closed-loop system transfer function matrix from the system reference command value r to the system output y can be expressed as the Equation (24).

$$H(s) = (I + G(s)KG_c(s))^{-1}G(s)KG_c(s) \quad (24)$$

The desired closed-loop transfer function matrix is selected as a diagonal matrix as shown in Equation (25), with elements chosen as $h_{1,d} = 1/T_{c,i} \cdot s + 1$.

$$H_d(s) = \begin{bmatrix} h_{1,d}(s) & 0 & \cdots & 0 \\ 0 & h_{12,d}(s) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{p,d}(s) \end{bmatrix} \quad (25)$$

where $T_{c,i}$ represents the closed-loop time constant of the i -th control loop, with a larger value leading to a more stable system response. Thus, the Equation (24) can be transformed to Equation (26).

$$H_d(s) = \text{diag}\left\{(I + G(s)KG_c(s))^{-1}G(s)KG_c(s)\right\} = \text{diag}\left\{\left(G^{-1}(s)KG_c^{-1}(s) + I\right)^{-1}KG_c(s)\right\} \quad (26)$$

The system internal model adopts a reduced-order model or approximate model of the controlled plant, and is taken as a first-order inertial element $g_i(s) = k_1/(k_2 s + 1)$. Given that $D(s) = G(s)K$ is a diagonal matrix or weakly diagonal matrix, the inverse of the closed-loop system transfer function matrix can thus be written in the form shown in Equation (27).

$$H_d(s) \approx \text{diag}\left\{\frac{d_i(s) \cdot g_{c,i}(s)}{1 + d_i(s) \cdot g_{c,i}(s)}\right\}, i = 1, 2, \dots, p \quad (27)$$

To obtain the PI controller parameter, $g_{c,i}(s)$ can be compared with the standard PI controller, and by setting $d_i(s) = K_i / (T_i \cdot s + 1)$, and hence the controller in Equation (29) can be obtained.

$$g_{c,i}(s) = \frac{h_{d,i}(s)}{d_i(s)(1 - h_{d,i}(s))} = \frac{T_i \cdot s + 1}{K_i T_{c,i} \cdot s} \quad (28)$$

$$g_{c,i}(s) = K_P + \frac{1}{s} \cdot K_I \quad (29)$$

where, $K_P = T_i / K_i T_{c,i}$, $K_I = K_P / T_i$.

5. Numerical Simulation and Verification

In this section, the performance of the feedforward–feedback control strategy (as shown in Figure 3) based on inverse system control is validated with a dual-spool turbofan engines, and the step responses of acceleration from IDLE to maximum rotation speed and deceleration from maximum rotation speed to IDLE state are presented. The data used for training the MLP model are derived from this nonlinear model, with a sampling frequency of 100 Hz. This nonlinear turbofan engine model is established by solving the matching Equation (1) and dynamic Equation (2) using Newton–Raphson iteration algorithm. For the given turbofan engine, the nonlinear system state in Equation (3) is selected as the fan speed N_1 and high-pressure compressor rotational speed N_2 , i.e., $x = [x_1, x_2]^T = [N_1, N_2]^T$. The fan speed N_1 and engine pressure ratio EPR of the engine are selected as the controlled output variable $y = [N_1, EPR]^T$ corresponding to the desired thrust. The control inputs u is selected as the fuel flow w_f and the main engine nozzle throat area A_8 , i.e., $u = [u_1, u_2]^T = [w_f, A_8]^T$. The maximum fuel flow rate is 1.129 kg/s, the maximum fan speed is 8000 RPM, and the maximum high-pressure compressor speed is 14,000 RPM. Variations in fuel flow and nozzle throat area both affect the system state x and output y . Consequently, the regulation of any single control loop inevitably influences the others. Hence, the control system consists two mutually coupled loops.

The MLP model approximates the engine's nonlinear affine system model and serves as the basis for feedforward control input calculation. The MLP model is used to approximate the engine's nonlinear affine system model and serves as the basis for feedforward control input. The MLP input is a polynomial feature expansion vector of the system state variables, rather than the system state itself, and the dimension of the expansion vector is determined by the Bayesian Information Criterion and Principal Component Analysis methods. To validate the effectiveness of the proposed feedforward control method based on the approximate nonlinear affine system's MLP model, numerical simulations for turbofan engine control were conducted using solely the feedforward control structure shown in Figure 2. Then, the effectiveness of the feedforward–feedback composite control for turbofan engine was verified.

5.1. Multi-Layer Perceptron for the Engine's Nonlinear Affine Model

The feedforward control input is calculated based on the MLP model. To achieve better engine dynamic prediction performance while reducing computational complexity, the MLP model employs a single hidden layer containing only 10 neurons. The MLP input uses a polynomial feature expansion vector of the system state x , therefore it is necessary to first determine the optimal polynomial feature expansion order. The appropriate order is selected based on the BIC as shown in Equation (16) and the prediction error results is shown in Figure 4.

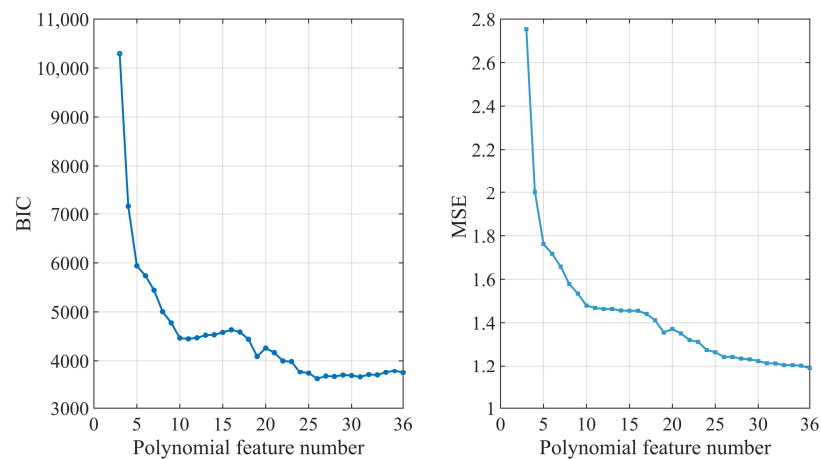


Figure 4. Polynomial feature expansion term selection criteria.

As illustrated in Figure 4, the BIC values calculated with varying numbers of polynomial feature terms show a significant decrease when the term number exceeds 10. When the term number surpasses 20, the BIC values further diminish to approximately 3700 and tend to stabilize. Figure 4 also presents the mean square error (MSE) for different values of features number. Figure 5 shows the prediction error of the trained MLP model. The prediction error gradually decreases as the number of polynomial feature terms increases, indicating that increasing the number of terms significantly improves fitting accuracy. Although PCA results suggest that retaining only the first three components preserves over 95% of the information, the actual fitting accuracy remains relatively low. Consequently, 12 polynomial terms as Equation (30) were selected for the feature expansion and MLP model training. The resulting MLP model yields an average prediction error of less than 0.1% and a maximum prediction error of less than 0.26%.

$$\phi(x) = \left[1, x_1, x_2, x_1^2, x_2^2, x_1 \cdot x_2, x_1^3, x_2^3, \sqrt{x_1}, \sqrt{x_2}, 1/x_1, 1/x_2, x_1^{0.3} \right]^T \quad (30)$$

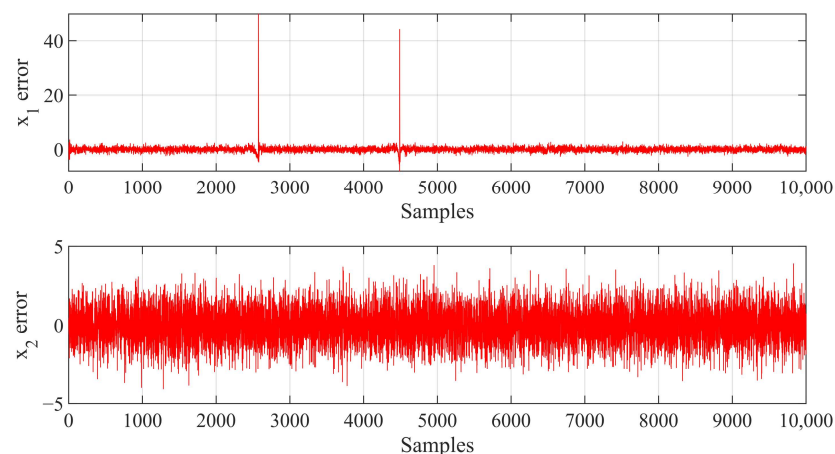


Figure 5. Prediction error of the trained MLP model output between the simulation data.

Taking into account the system's nonlinear complexity and training data, a 2nd or 3rd-order polynomial is typically selected, and an appropriate expansion dimension is chosen to avoid overfitting.

5.2. Multi-Loop PI Controller Design

By linearizing the nonlinear model in Equation (2) at a specific operating point, the linearized model shown in Equation (31) can be obtained.

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \quad (31)$$

where

$$A = \begin{bmatrix} -9.4788 & 6.2643 \\ 0.4839 & -4.5995 \end{bmatrix}, B = \begin{bmatrix} 5757.9968 & 10,116.9426 \\ 3447.6876 & -138.0476 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0.06234 & 0.08969 \end{bmatrix}$$

According to the decoupling compensator design methodology, the frequency range was $\omega < 5$ rad/s selected. The pre-compensator matrix K shown in Equation (32) was obtained using the YALMIP optimization toolbox.

$$K = \begin{bmatrix} -0.009989 & 0.013907 \\ -0.000630 & -0.197646 \end{bmatrix} \quad (32)$$

Based on the diagonally dominant controlled system within the specified frequency range and under the action of the pre-compensator, a multi-loop PI controller is designed using the direct synthesis method. The structure of the PI controller is given by the transfer function, and the resulting parameters for the different control loops are listed in Table 1.

$$G_{c,i}(s) = K_{P,i} + K_{I,i} \cdot \frac{1}{s}, i = 1, 2 \quad (33)$$

Table 1. Multi-loop PI controller parameters.

Loop	K_P	K_I
W_f	−0.2853	0.3031
A_8	−0.0415	0.3318

5.3. Simulation Results with Feedforward Control and Multi-Loop PI Control

During the operation of the turbofan engine, the feedforward control inputs are calculated using the MLP model and the nonlinear affine system. The desired engine thrust is shown in Figure 6, while the corresponding controlled output parameters—fan speed and EPR—are presented in Figure 7. In this control process, the fuel flow and nozzle throat area serve as the control inputs; these inputs and the aforementioned outputs constitute two mutually coupled control loops. The MLP model which approximate the nonlinear affine system in Equation (9) is employed to calculate the feedforward control inputs.

The tracking control performance for the N_1 and the engine pressure ratio desired signals under the action of the feedforward control input is illustrated in Figures 8–10. During the simulation, the maximum value of the main engine pressure ratio is 3.56. The corresponding feedforward control inputs for this operating process are shown in Figure 10, where the control inputs vary relatively smoothly during the acceleration process. The maximum fuel flow is 1.4 kg/s, and the maximum nozzle throat area is 0.468 m². The feedforward control input offered by the MLP model.

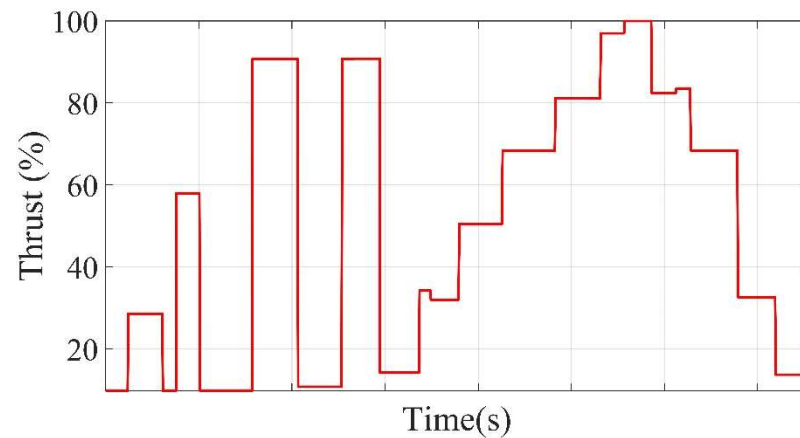


Figure 6. Desired thrust response curve.

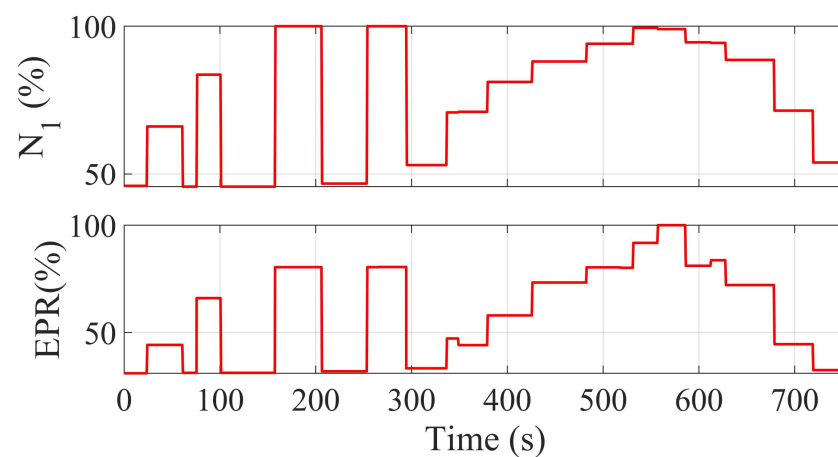


Figure 7. Desired system output curve.

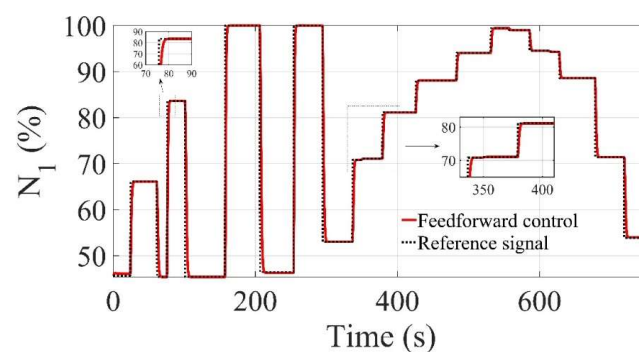


Figure 8. Turbofan engine fan speed during the feedforward control.

The simulation results are summarized in Table 2, where ARE and MRE denote the Average Relative Error and the Maximum Relative Error of the steady-state error, respectively. The steady-state errors and the settling times for acceleration to the maximum operating state are listed in Table 2. The steady-state errors for both controlled parameters are less than 0.2%, and the settling times are under 2.72 s. This indicates that the time required to accelerate from idle speed to the maximum operating state is short. Consequently, the feedforward control provides sufficiently fast system response and high-precision output control. The feedforward control inputs calculated using the MLP model can effectively achieve the desired control objectives.

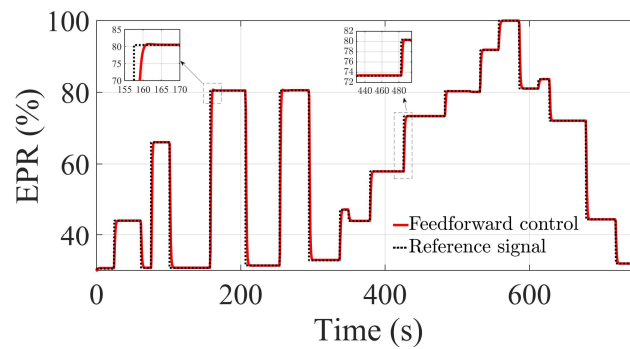


Figure 9. Turbofan engine EPR during the feedforward control.

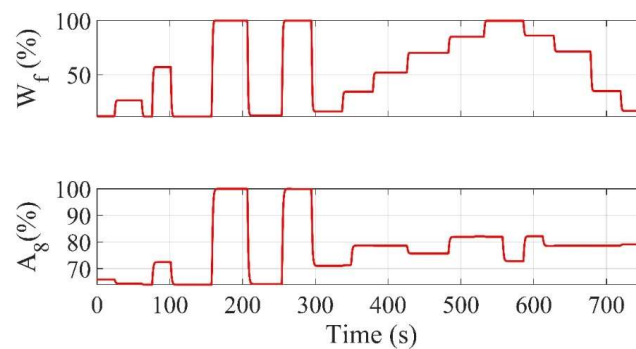


Figure 10. Feedforward control input signal profile.

Table 2. Control performance of feedforward control.

Index	N_1	EPR
ARE	0.11%	0.15%
MRE	0.94%	0.53%
Settling time	2.60 s	2.71 s

The simulation results demonstrate that, provided the MLP model is sufficiently accurate and the prediction error is minimal, the feedforward control based on the nonlinear affine system can achieve rapid and precise engine acceleration control and thrust regulation at the whole operating range.

However, since the MLP model cannot perfectly match the actual nonlinear system, invariant steady-state errors inevitably arise during the control process. Specifically, the maximum steady-state error for the EPR can reach 0.532917% and that for N_1 can reach 0.936759%. As both tracking errors far exceed the acceptable tolerance, feedback control must be incorporated into the control scheme to improve control performance.

To verify the performance of the feedforward–feedback composite control in the presence of discrepancies between the MLP model and the controlled engine, a performance-degraded engine model was constructed. This was achieved by applying the following reductions to the nonlinear engine model: the high-pressure compressor pressure ratio and efficiency were reduced by 7.75% and 5.5%, respectively; the fan pressure ratio and efficiency were reduced by 7.82% and 5.5%, respectively; and the high-pressure turbine pressure ratio and efficiency were reduced by 2% and 3.1%, respectively. Numerical simulations were conducted using this degraded nonlinear model of the turbofan engine, employing the same desired system output signal shown in Figures 11 and 12.

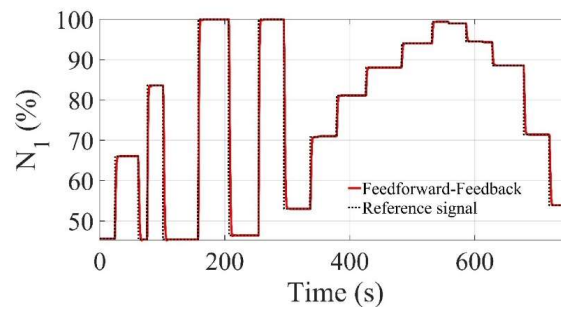


Figure 11. Fan speed tracking performance under the feedforward–feedback composite control.

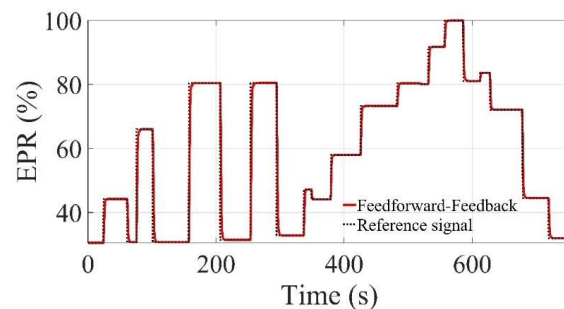


Figure 12. EPR tracking performance under the feedforward–feedback composite control.

A multi-loop PI controller is adopted to compensate for the MLP model-based feedforward control, aiming to address the lack of robustness against model uncertainties and external disturbances. In the simulation of the feedforward–feedback composite control, the engine accelerates from a low speed to the maximum speed, followed by thrust regulation at the maximum operating state.

The tracking performance for the engine fan speed and EPR using the feedforward–feedback composite control is illustrated in Figures 11 and 12, covering acceleration, deceleration, and steady-state operation processes. The combined control input for this process is shown in Figure 13, with the specific feedforward and PI feedback control inputs detailed in Figures 14 and 15, respectively.

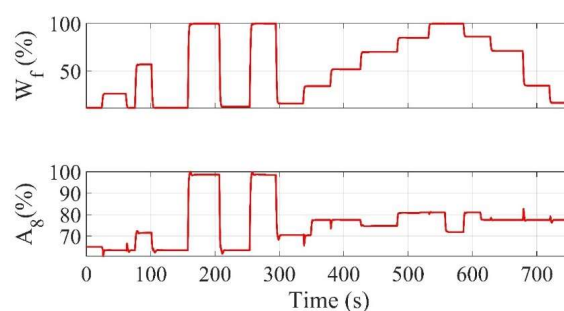


Figure 13. Feedforward–feedback composite control input signal profile.

The simulation results presented in Figures 10 and 11 demonstrate that, compared with the feedforward-only control, the multi-loop PI feedback control effectively eliminates the steady-state tracking errors caused by model discrepancies, reducing them to less than 0.001%. Therefore, relying on multi-loop PI feedback control to enhance the robustness of the control system is proven to be effective.

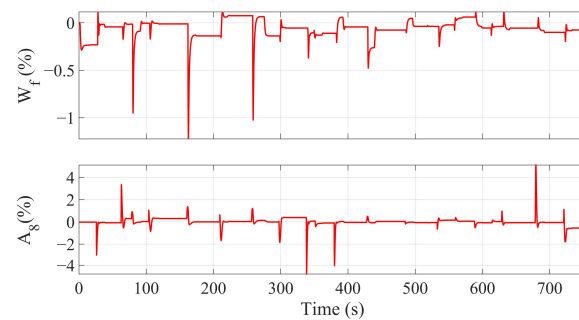


Figure 14. Feedback control input signal of the feedforward–feedback composite controller.

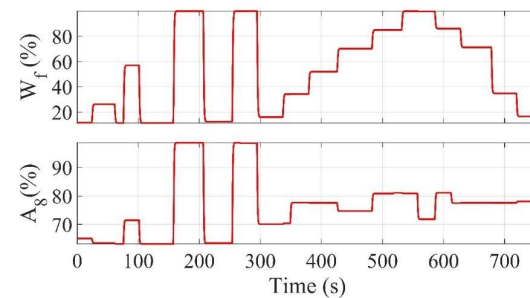


Figure 15. Feedforward control input signal of the feedforward–feedback composite controller.

6. Conclusions

This paper presents a feedforward control strategy for turbofan engine thrust regulation throughout the whole operating range. The feedforward control method is built upon an MLP model that adopts a nonlinear affine system architecture to approximate the turbofan engine dynamics, and the feedforward control input in the whole operating range can be obtained by simple algebraic calculation of the MLP model output at each control step. Thus, this method can overcome the limitation of conventional engine controller design, which requires obtaining linearized models around multiple steady-state operating points, designing a controller for each linear model, and then integrating all controllers into a single supervisory controller. In contrast to model predictive control, which requires solving computationally intensive optimization problems online at each control step, the proposed method enables the feedforward control input calculation at each step through simple algebraic computations. The turbofan engine performance of the designed feedforward controller is verified under chosen conditions. The simulation results show that the multi-loop systems controlled by the feedforward controller can achieve the expected control performance in transient and steady-state control process, and the system's settling time is less than 2.6 s, and the maximum steady-state error between the reference signal and the controlled system outputs is less than 0.97%. Additionally, the steady-state tracking errors resulting from MLP model errors and disturbances can be effectively eliminated using a multi-loop PI controller, achieving a steady-state error reduction to 0.01% without increasing the settling time.

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

Funding: This research was funded by Liaoning Provincial Science and Technology Department Project—Joint Plan Project, grant number 2025JH2/101800122.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author Penghui Sun.

Conflicts of Interest: Authors Pengpeng Li and Penghui Sun were employed by State Owned Chuanxi Machinery Factory. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Jaw, L.C.; Mattingly, J. *Aircraft Engine Controls: Design, System Analysis, and Health Monitoring*; AIAA: Reston, VA, USA, 2009.
2. Richter, H. *Advanced Control of Turbofan Engines*; National Defense Industry Press: Beijing, China, 2013.
3. Jaw, L.C.; Garg, S. *Propulsion Control Technology Development in the United States—A Historical Perspective*; NASA—Glenn Research Center: Cleveland, OH, USA, 2005.
4. McIsaac, B.; Langton, R. *Gas Turbine Propulsion Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
5. Bei, Y.; Xi, W.; Sun, P. Non-affine parameter dependent LPV model and LMI based adaptive control for turbofan engines. *Chin. J. Aeronaut.* **2019**, *32*, 585–594. [[CrossRef](#)]
6. Sun, P.; Wang, X.; Yang, S.; Yang, B.; Chen, H.; Wang, B. Bumpless Transfer of Uncertain Switched System and Its Application to Turbofan Engines. *Energies* **2021**, *14*, 5204. [[CrossRef](#)]
7. Gu, N.; Wang, X.; Lin, F. 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. [[CrossRef](#)]
8. Yang, S.; Wang, X.; Sun, P. Design of LPV-Based Sliding Mode Regulator for Limit Protection of Aero-engines. *J. Dyn. Syst. Meas. Control* **2020**, *142*, 091007. [[CrossRef](#)]
9. Imani, A.; Montazeri-Gh, M. Improvement of Min–Max limit protection in aircraft engine control: An LMI approach. *Aerosp. Sci. Technol.* **2019**, *68*, 214–222.
10. Gu, B.B. Robust Fuzzy Control for Aeroengines. Master's thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2018.
11. Shamma, J.S.; Athans, M. Gain scheduling: Potential hazards and possible remedies. *IEEE Control Syst. Mag.* **1992**, *12*, 101–107. [[CrossRef](#)]
12. Sun, P.; Wang, X.; Yang, B.; Zeng, D. Model reference adaptive control with smooth switching scheme for piecewise linear systems and its application in turbofan engine control. *J. Aerosp. Power* **2019**, *34*, 1615–1623.
13. Liu, S.; Wei, X.; Fang, X.; Ji, W.; Pan, S. Study on the dynamic characteristics of the coupling feedback controlling vibration system with delay. *J. Vib. Shock* **2017**, *36*, 177–183.
14. Miao, K.; Wang, X.; Zhu, M.; Yang, S.; Pei, X.; Jiang, Z. Transient Controller Design Based on Reinforcement Learning for a Turbofan Engine with Actuator Dynamics. *Symmetry* **2022**, *14*, 684. [[CrossRef](#)]
15. Wang, Y.; Zheng, Q.; Xu, Z.; Zhang, H. A novel control method for turboshaft engine with variable rotor speed based on the Ngdot estimator through LQG/LTR and rotor predicted torque feedforward. *Chin. J. Aeronaut.* **2020**, *33*, 1867–1876. [[CrossRef](#)]
16. Bai, G.; Elxat, E.; Fu, X.; Meng, Y.; Liu, L.; Gu, Q. Path tracking of car-like robots based on feed-forward model predictive control. *Chin. J. Eng.* **2024**, *46*, 1130–1139.
17. Liu, Q.; Sun, D.; Chen, H.; Li, D.; Wang, P. Design of energy-saving driving strategy based on proximal policy optimization considering urban transport information. *Control Theory Technol.* **2024**, *23*, 74–90. [[CrossRef](#)]
18. Ji, C. Analysis to the Development of Advanced Technology for Integrated Flight-Propulsion Control. *Aerosp. Power* **2019**, *4*, 32–38.
19. Wurth, S.P.; Smith, M.S. F-35 Propulsion System Integration, Development, and Verification. In Proceedings of the 2018 Aviation Technology, Integration, and Operations Conference, Atlanta, GA, USA, 25–29 June 2018.
20. Zhang, W.J.; Pan, W.H.; Zhang, X.F.; Liu, Q. Predictor-based sampled-data output-feedback control for feedforward nonlinear time-delay systems. *Control Theory Technol.* **2025**, *23*, 105–117.
21. Ji, X.; Zhu, S.; Xu, Y.; Longman, R.W. Lifted time stable inversion based feedforward control for linear non-minimum phase systems. *Automatica* **2025**, *171*, 111979. [[CrossRef](#)]
22. Chen, P.; Xu, B.; He, W.; Zhang, X. Transformer-Based Model Predictive Control with Feedforward Prediction for Ammonia Injection Optimization in SCR Denitrification Systems. *J. Environ. Chem. Eng.* **2025**, *202*, 107804. [[CrossRef](#)]
23. Martin, G.; Bodo, H. Model-based Feedforward Control in Industrial Robotics. *Int. J. Robot. Res.* **2002**, *21*, 45–60. [[CrossRef](#)]

24. Li, S.C.; Wang, Y.; Zhang, H.B. Research on adaptive feedforward control method for Tiltrotor Aircraft/Turboshaft engine system based on radial basis function neural network. *Aerosp. Sci. Technol.* **2024**, *150*, 109180. [[CrossRef](#)]
25. Yu, Z.Y.; Wang, C.; Du, X.; Nie, L.C.; Sun, X.M. An integrated inlet/engine performance seeking control model based on salp swarm algorithm extreme learning machine. *J. Propuls. Technol.* **2024**, *45*, 2302042.
26. Hunt, L.R.; Meyer, G. Stable inversion for nonlinear systems. *Automatica* **1997**, *33*, 1549–1554. [[CrossRef](#)]
27. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.