applied sciences

MDPI

Article

Hybrid Grey Wolf Optimization Nonlinear Model Predictive Control for Aircraft Engines Based on an Elastic BP Neural Network

Lingfei Xiao *, Min Xu, Yuhan Chen and Yusheng Chen

College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; xumin_nuaa@163.com (M.X.); 15234257364@163.com (Y.C.); 1762698394@foxmail.com (Y.C.)

* Correspondence: lfxiao@nuaa.edu.cn

Received: 14 January 2019; Accepted: 21 March 2019; Published: 25 March 2019

Featured Application: This paper provides a novel hybrid grey wolf optimization algorithm and a new nonlinear model predictive control method for aircraft engines. These contributions can also be used for other kind of systems with constraints and performance optimization requirements.

Abstract: In order to deal with control constraints and the performance optimization requirements in aircraft engines, a new nonlinear model predictive control method based on an elastic BP neural network with a hybrid grey wolf optimizer is proposed in this paper. Based on the acquired aircraft engines data, the elastic BP neural network is used to train the prediction model, and the grey wolf optimization algorithm is applied to improve the selection of initial parameters in the elastic BP neural network. The accuracy of network modeling is increased as a result. By introducing the logistics chaotic sequence, the individual optimal search mechanism, and the cross operation, the novel hybrid grey wolf optimization algorithm is proposed and then used in receding horizon optimization to ensure real-time operation. Subsequently, a nonlinear model predictive controller for aircraft engine is obtained. Simulation results show that, with constraints in the control signal, the proposed nonlinear model predictive controller can guarantee that the aircraft engine has a satisfactory performance.

Keywords: aircraft engines; hybrid grey wolf optimizer; nonlinear model predictive control; elastic BP neural network

1. Introduction

Predictive control is a model-based advanced control technique [1]. The working principle mainly includes three modules: the predictive model, receding horizon optimization, and feedback correction. In each control interval, the MPC algorithm attempts to optimize system behaviors, computes a future manipulated variable adjustment sequence, then sends the first input to the system in an optimal order, and repeats the entire calculation in subsequent control intervals. MPC can be divided into linear MPC and nonlinear MPC. Linear MPC is an effective control strategy, widely used in the process industry. However, many systems are essentially nonlinear. Nonlinear model predictive control (NMPC) is the corresponding predictive control scheme for nonlinear systems [2]. Aircraft engines are a class of complicated aerodynamic thermodynamic systems with strong nonlinearity, systems that have a broad working range, complex working conditions, and inevitably many uncertainties such as disturbance and unmodeled dynamics of actual situations. Therefore, aircraft engines are subject to constraints during operation, and model predictive control can deal with constraints excellently. Therefore, an increasing amount of attention has been paid to the application of NMPC in aircraft engines [3–5].

Traditional NMPC is based on a nonlinear programming (NP) method to solve control law, such as the sequential quadratic programming (SQP) method [6–8], but the initial condition of an NP method is limited in its solution, and the algorithm easily falls into a local optimum. Existing intelligent algorithms such as the genetic algorithm (GA) [9], the artificial bee colony algorithm (ABC), and the particle swarm optimization algorithm (PSO) have been successfully used for many optimization problems and have achieved results far superior to those of the traditional NP method. However, they suffer from a slow convergence rate and a low computational accuracy for complex problems, and they also easily fall into a local optimum [10–12]. Therefore, it is necessary to study an optimization method with a strong searching ability, a high convergence speed, a high calculation accuracy, and a good ability to handle constraints.

Some bio-heuristic optimization algorithms have been proposed in recent years, including the grey wolf algorithm [13], the whale algorithm [14], the spider monkey algorithm [15], and the polar bear algorithm [16]. The common feature of these newly bio-heuristics is that they optimize by simulating the survival or predation mechanisms of biological populations. The grey wolf optimization (GWO) algorithm is a new intelligent algorithm that was proposed in 2014. Compared with other intelligent algorithms, the GWO has a faster convergence speed and a stronger search ability in solving function optimization problems, so it has been applied to many optimization problems [17–19]. However, as a swarm intelligence algorithm, it is inevitable that it still has the risk of falling into a local optimum. Therefore, in order to ensure an optimization ability, many scholars have made improvements on the GWO. In [20], EPD (Population Dynamic Evolution Operator) was integrated into the population updating process of the GWO algorithm by combining self-organizing criticality. The search range of the wolf group in the GWO algorithm was extended to the entire solution space in each iteration process, thus increasing the probability of obtaining a global optimal solution. In [21], the population was divided into a local optimization group and a global optimization group, which increased the diversity of the population and improved the search ability. However, the global search was blind and lacked norms, which increased the complexity of the algorithm. In [22], a differential evolution optimization algorithm was combined with the GWO algorithm to eliminate inferior individuals and accelerate the optimization speed, but it is still easily fell into a local optimum because the fundamental idea of the algorithm was not considered. In [23], an embedded optimization operator was used to update the population position, and a differential evolution strategy was used with adaptive parameters to further improve the optimization functions. The performance of the hybrid algorithm was improved by introducing the worst individual, but the role of other individuals was not considered. Although these improved algorithms optimized the performance of the GWO algorithm in some aspects, it was still difficult to achieve a balance between convergence speed, local optimum, and search accuracy.

The aim of this paper is to provide an effective control method for aircraft engine systems with constraints.

The primary contribution of this paper is as follows:

(1) By introducing logistic and chaotic mapping, an individual optimal search mechanism, and cross operation, a novel hybrid grey wolf optimization (HGWO) algorithm is proposed. The presented HGWO algorithm can provide faster convergence speed and better search accuracy and avoids falling into a local optimum.
(2) The GWO is used to improve the selection of initial parameters in an elastic BP neural network. The accuracy of network modeling is increased, and the established engine neural network model has a better prediction performance as a result.
(3) A nonlinear model predictive control method based on a hybrid grey wolf optimizer is proposed for the control of aircraft engines with constraints.

The remainder of this paper is organized as follows. Section 2 describes the considered aircraft engine and the corresponding GWO-based elastic BP neural network prediction model. Section 3 presents the HGWO algorithm. In Section 4, the design of the NMPC controller is shown. Section 5 verifies the simulation of the designed NMPC controller. Section 6 draws conclusions.

2. Aircraft Engine and Its Prediction Model

2.1. Brief Description of the Aircraft Engine

The dynamic model of the aircraft engine is highly complicated and nonlinear [24,25], which can be described as

$$\dot{x} = f(x, u)y = g(x, u) \tag{1}$$

where x is the state vector, $\dot{x}$ is the state derivative vector, u is the input vector, y the output vector, and $f(\cdot)$ and $g(\cdot)$ are nonlinear functions.

For a twin-shaft aircraft engine whose generalized schematic is shown in Figure 1, the state vector x can include a low-pressure (LP) spool speed and a high-pressure (HP) spool speed; the control vector u may contain the main fuel flow, the nozzle throat area, the inlet guide vane angle, and the compress guide vane angle; the output vector y may be comprised of low-pressure spool speeds, high-pressure spool speeds, a low-pressure turbine total temperature and total pressure, a pressure ratio of turbine, a low pressure surge margin, and a gross thrust.

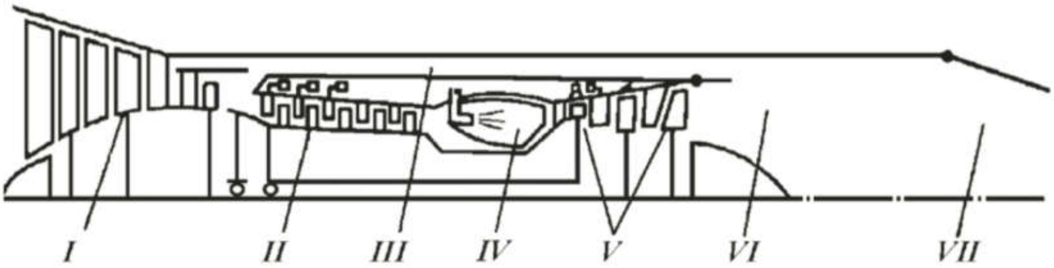


Figure 1. Generalized schematic of the twin-shaft aircraft engine [19]: I: fan (LP compressor); II: HP compressor; III: bypass duct; IV: main combustion chamber; V: HP and LP turbines; VI: mixing chamber; VII: nozzle.

2.2. A Prediction Model of an Aircraft Engine

A neural network [26–28] has excellent predictive ability and has been used by researchers in multi-step predictive control as early as the 1990s. Generally, it is employed as a predictive model to predictoutput values for rolling optimization.

In this section, a modified elastic BP neural network whose initial parameters selection is improved by a GWO algorithm is presented to construct a prediction model of the aircraft engine.

2.2.1. Elastic BP Neural Network (EBPNN)

Because a BP neural network usually uses a sigmoid function as an activation function, the network output may enter a flat area when the net input of each node is too large. The flat area causes a small partial derivative of the error with respect to the weight value, which results in a correcting process which stagnates and slows down the convergence rate. In order to reduce the influence of the small change in gradient amplitude on the weight and threshold correction, the elastic BP algorithm is used to adjust the weight and threshold by judging the change in the positive and negative directions of the gradient [29].

For convenience of description, we take the training of the weights and thresholds of a node in the network as an example to briefly summarize the main correction formulas of the conventional BP neural network and the elastic BP neural network.

The conventional BP neural network weight and threshold correction formula obtained by the gradient descent method is as follows:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)(1) \tag{2}$$

$$\Delta w_{ij}(t) = -\eta_1 \frac{\partial E}{\partial w_{ij}} \tag{3}$$

$$b_j(t+1) = b_j(t) + \Delta b_j(t) \tag{4}$$

$$\Delta b_j(t) = -\eta_2 \frac{\partial E}{\partial b_j} \tag{5}$$

where w_{ij} is the weight of the network node, b_j is the threshold value of the network node, Δw_{ij} and Δb_j are the correction values of the node weight and the threshold value, respectively, t is the number of iterations, η_1 and η_2 are the learning rates of the weight and the threshold value, respectively, and E_{nn} is the error function.

From Equations (3) and (5), it can be seen that the weight and threshold values of the conventional BP algorithm mainly depend on the change in gradient amplitude. When entering the flat region, the gradient amplitude is basically zero, the weight and threshold value no longer change, and the algorithm easily falls into a local optimum.

Therefore, the elastic BP neural network modifies the conventional BP algorithm. The correction formula is as follows:

$$x_{TD}(t) = \frac{\partial E(t)}{\partial w_{ij}} \frac{\partial E(t-1)}{\partial w_{ij}} \tag{6}$$

$$\Delta w_{ij}(t) = \alpha^{sign(x_{TD})} \Delta w_{ij}(t-1) \tag{7}$$

$$\Delta b_j(t) = \alpha^{sign(x_{TD})} \Delta b_j(t-1) \tag{8}$$

where x_{TD} is the gradient product of two adjacent iterations.

From Equations (6)–(8), it can be seen that the correction of weights and thresholds only considers the direction change of the gradient and not the change of the gradient amplitude, thus avoiding entering the flat area, preventing reaching the local convergence area, and improving the convergence speed and the training ability of the network.

2.2.2. The Grey Wolf Optimization (GWO) Algorithm

The GWO algorithm is proposed by imitating the leadership level and predation mechanism of the natural grey wolf population. According to the leadership level, the grey wolf population is divided into four types: alpha (α), beta (β), delta (δ), and omega (ω). The α grey wolf is called the ruling level: the whole pack must obey its order. This level of grey wolf is mainly responsible for drawing up the predation, rest, forward, stop, and other decision-making. The primary responsibility of the β grey wolf is to help the α to formulate a variety of decisions and participate in other collective activities. The δ grey wolf is mainly responsible for the execution of α and β decisions, as well as ordering the ω grey wolf. The ω grey wolf mainly follows the command of the three high-level grey wolves. The predation mechanism is divided into three predation steps: hunting for prey, encircling prey, and attacking prey.

When the GWO algorithm is used to solve the optimization problem, it is assumed that the number of grey wolves in the wolf population is n and the search space is d-dimension. The position of the ith grey wolf in the d-dimensional space can be expressed as $X_i = \left[x_i^1, x_i^2, \ldots, x_i^d\right]$. The current optimal individual of population is denoted by α, and the location is recorded as X_α. The corresponding individuals with the fitness values ranked second and third are denoted as β and δ, and the positions are denoted as X_β and X_δ. respectively. The position of prey corresponds to the global optimal solution of the optimization problem. The predatory behavior of grey wolf populations was introduced in Equations (1) and (2):

$$D = |CX_P(t) - X_i(t)| \tag{9}$$

$$X_i(t+1) = X_P(t) - A \cdot D \tag{10}$$

where t is the number of iterations, $X_p(t)$ is the position of the prey in the tth generation, and D is the distance between individual and prey. C is the convergence factor, and A is the wobble factor. The calculation formula is

$$\begin{aligned} C &= 2r_1 \\ A &= 2ar_2 - a \end{aligned} \tag{11}$$

where r_1 and r_2 are random numbers between 0 and 1. a is the distance control parameter, which decreases linearly from 2 to 0 with the increase of iterations. The calculation formula is

$$a = 2 - t/t_{\max}. \tag{12}$$

When the wolf determines the location of prey in the predation process, α commands β, δ, and ω to hunt prey. However, the prey location cannot be determined directly. In the pack, α, β, and δ are the closest to prey, so we can locate prey using the three positions. Ultimately we obtain the following location update formula:

$$\begin{aligned} D_{\alpha i} &= |CX_\alpha(t) - X_i(t)| \\ D_{\beta i} &= |CX_\beta(t) - X_i(t)| \\ D_{\delta i} &= |CX_\delta(t) - X_i(t)| \\ X_{1i} &= X_\alpha(t) - A \cdot D_{\alpha i} \\ X_{2i} &= X_\beta(t) - A \cdot D_{\beta i} \\ X_{3i} &= X_\delta(t) - A \cdot D_{\delta i} \\ X_i(t+1) &= \frac{1}{3}(X_{1i} + X_{2i} + X_{3i}) \end{aligned}$$

where X_{1i}, X_{2i}, and X_{3i} are the positions of the ith grey wolf individual after moving toward α,β, and δ, respectively. $D_{\alpha i}$, $D_{\beta i}$, and $D_{\delta i}$ are the distances of grey wolf individuals to α, β, and δ, respectively, and $X_i(t+1)$ is the updated position of the ith grey wolf individual.

2.2.3. The Aircraft Engine Prediction Model Based on GWO-EBPNN

In this paper, the nonlinear model of aircraft engine from the main fuel flow w_f to low-pressure spool speed n_l is considered, which can be described in the form of the following discrete difference equation:

$$n_l(k) = f\left[n_l(k-1), n_l(k-2), w_f(k), w_f(k-1)\right]. \tag{13}$$

The corresponding neural network model can be expressed in the following form:

$$\hat{n}_l(k) = f_{NN}\left[n_l(k-1), n_l(k-2), w_f(k), w_f(k-1)\right] \tag{14}$$

where $\hat{n}_l$ is the neural network estimator of n_l, and k is the sampling instant.

The input vector of the network is $\left[n_l(k-1), n_l(k-2), w_f(k), w_f(k-1)\right]$, and the output is $\hat{n}_l$; that is, the network contains four input neuron nodes and one output neuron node. The number of hidden layer nodes not only depends on the nonlinearity of the system but also is affected by the excitation function and training accuracy requirements. Therefore, it cannot be randomly selected in setting. In this paper, Kolmogorov's theorem is used to obtain 9 as the number of hidden layer nodes in the elastic BP neural network.

The randomization of initial weights and thresholds is an important factor affecting the training accuracy of the elastic BP neural network algorithm, which easily leads to training results that fall into the local convergence region, making the prediction accuracy of the neural network too low to meet the modeling requirements.

Therefore, in view of the deficiency of the traditional initial value selection, this paper introduces the GWO algorithm in the elastic BP neural network to optimize the initial value selection. The basic idea is to use the GWO algorithm to globally pre-optimize the initial value of the network and then use the optimal solution obtained by the algorithm as the initial weight and threshold of each node of the elastic BP neural network for subsequent training operations.

According to the above elaboration, there are 4 nodes in the input layer, 9 nodes in the hidden layer, and 1 node in the output layer. Therefore, it is known that the whole network includes two threshold matrices, $b^1 = [b^1{}_i]_{9\times 1}$ and $b^2 = [b^2{}_i]_{1\times 1}$, and two weight matrices, $w^1 = [w^1{}_{i\times j}]_{4\times 9}$ and $w^2 = [w^2{}_{j\times 1}]_{9\times 1}$. Before using the GWO algorithm to optimize the initial weights and thresholds, each matrix is sorted to a certain extent and the elements in each matrix are combined into a vector. During optimization, the vector is set as the position vector of the individual population in the GWO algorithm, and the reciprocal of the criterion function of the elastic BP neural network (the mean square error between the expected output and the network output) is taken as the fitness function of the algorithm. After optimization iteration, the obtained optimal solution is applied to the elastic BP neural network initialization. On this basis, network training can be carried out.

The corresponding network initial value optimization process is as follows:

Step 1: Create an elastic BP neural network according to the selected samples and the network structure determined previously.

Step 2: Set the initial value vector of the weights and thresholds of each layer of the elastic BP neural network as the individual position of the population, and use the designed fitness function to calculate the fitness value of each individual. After sorting, carry out global optimization according to the optimization procedure in Section 2.2.2. When the fitness value reaches the condition or the iteration number reaches the maximum value, complete the optimization and save the position of the optimal individual as the optimal initial weight and threshold of the network to be trained.

Step 3: Input the optimal initial value obtained in Step 2 into the network, start training the elastic BP network, and when the optimal conditions and iteration times are met, complete training to obtain the neural network training parameter value.

Step 4: Select sample data to test the trained elastic BP network and verify the network performance.

One can see from Figure 2 clearly that GWO is used to improve the selection of initial parameters in the elastic BP neural network.

The aircraft engine prediction model can now be obtained by the recursive prediction method based on GWO-EBPNN.

The main recurrence process is as follows:

At the k sampling instant, if the actual output value of the engine is known as $n_l(k)$, the predicted output value $\hat{n}_l(k)$ can be described by

$$\hat{n}_l(k) = f_{NN}\Big[n_l(k-1), n_l(k-2), w_f(k), w_f(k-1)\Big]. \tag{15}$$

The predicted output value $\hat{n}_l(k+1)$ at the $k+1$ sampling instant will be

$$\hat{n}_l(k+1) = f_{NN}\Big[\hat{n}_l(k), n_l(k+1), w_f(k+1), w_f(k)\Big]. \tag{16}$$

The predicted output value $\hat{n}_l(k+i)$ at the $k+i$ sampling instant will be

$$\hat{n}_l(k+i) = f_{NN}\left[\hat{n}_l(k+i-1), \hat{n}_l(k+i-2), w_f(k+i), w_f(k+i-1)\right] \tag{17}$$

where $i = 1, 2, \ldots, P$, and P is the prediction horizon.

The disadvantage of using recursive prediction is that there is inevitably an error between the neural network model and the real engine system, because of inevitable uncertainties such as disturbance and the unmodeled dynamic in actual situations. In multi-step prediction, each step of prediction will produce a prediction error, which will be superimposed and amplified with the increase in recursive times.

Therefore, the requirement for model accuracy is relatively high, and a feedback mechanism is needed to correct the prediction value, which will be discussed in Section 4.

After the prediction accuracy is improved by feedback correction, the advantage of recursive prediction in the calculation can be realized. Therefore, the nonlinear prediction model of the engine in this paper will be established by using a recursive neural network.

For convenience of expression, n_l will be expressed by y, w_f will be expressed by u, and $\hat{n}_l$ will be expressed by $\hat{y}$ in the following.

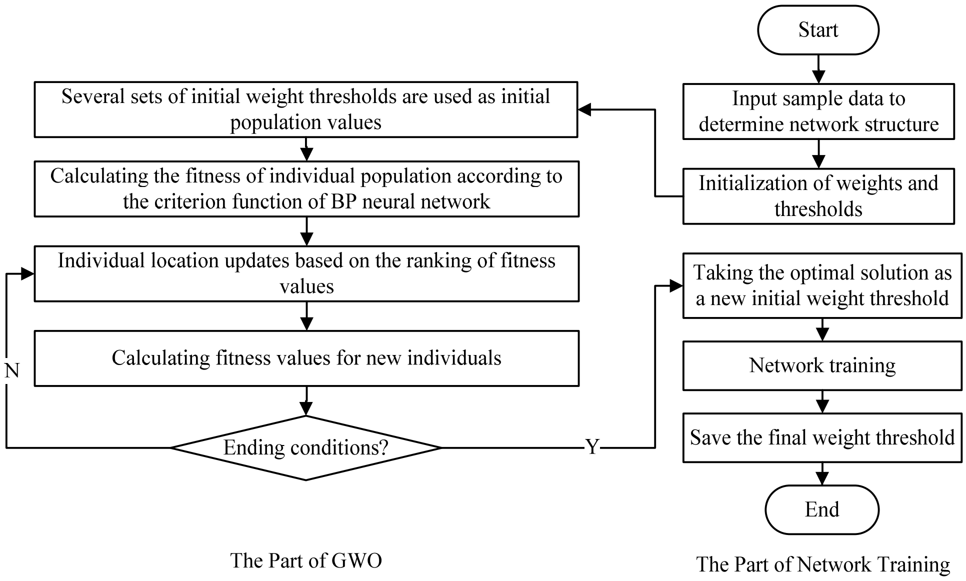

Start
Input sample data to determine network structure
Initialization of weights and thresholds
Several sets of initial weight thresholds are used as initial population values
Calculating the fitness of individual population according to the criterion function of BP neural network
Individual location updates based on the ranking of fitness values
Calculating fitness values for new individuals
N
Ending conditions?
Y
Taking the optimal solution as a new initial weight threshold
Network training
Save the final weight threshold
End
The Part of GWO
The Part of Network Training

Figure 2. The training structure diagram of the elastic BP neural network based on grey wolf optimization (GWO) (GWO-EBPNN).

3. Hybrid Grey Wolf Optimization (HGWO) Algorithm

Because an aircraft engine is complex and requires high real-time control, the optimization algorithm applied in NMPC to obtain the controller should have a satisfactory optimization speed.

Hence, in order to improve the optimization speed of the GWO algorithm and prevent the solution process from a local optimum, a new hybrid grey wolf optimization (HGWO) algorithm is proposed. The HGWO algorithm improves the traditional GWO algorithm in three ways:

1. A logistic chaotic sequence is used to generate the initial individual position. Compared with random initialization, the initial population position distribution is more uniform, and the convergence rate is then improved.
2. An individual optimal retention mechanism is used to increase the ability of the independent evaluation of GWO algorithmand to prevent a local optimum; that is to say, the optimal value of the individual in the optimization process will be retained.

3. Crossover operation is used to increase the diversity of populations, andthen better inividuals are retained by the greedy rule (a selection mechanism).. Superior offspring is thus obtained.

3.1. Logistic Chaotic Mapping

For the swarm intelligence algorithm, the quality of the initial population directly affects the convergence rate and the optimization ability of the algorithm. In solving optimization problems, without any prior knowledge of the global optimal solution, the initial population should be as evenly distributed as possible in the feasible region, and this can prevent a local optimum and lead to high-quality solutions.

The traditional GWO algorithm adopts a stochastic initialization method, which has high randomness and cannot guarantee a global optimal solution. Chaos is a nonlinear phenomenon with traversal uniformity. Chaotic motion can traverse all states in a certain range without repetition according to its own laws [22]. Therefore, if chaotic motion is used to initialize the population, there is no doubt that it is superior to stochastic initialization. In this paper, logistic maps are used to generate chaotic sequences as an initial population. The logistic map expression is [30].

$$l(\mathrm{t}+1) = \mu \cdot l(t) \cdot (1 - l(t)), l \in [0,1],\ \mu \in (0,4]. \tag{18}$$

Suppose the feasible region of the optimization problem is $[a, b]$. The chaotic sequence generated from Equation (18) can be mapped into an initial population in the feasible domain using Equation (19):

$$X(t) = l(t) \cdot (a - b) + b. \tag{19}$$

3.2. Individual Optimal Retention Mechanism

The traditional GWO algorithm only considers the guiding role of α, β, and δ. However, it ignores the search function of the other grey wolf individual. Therefore, we can refer to the description of the individual search function in the particle swarm optimization algorithm and propose using an individual optimal selection mechanism to increase the search function of the grey wolf individual. Thus, a new position updating formula is obtained on the basis of Equation (18):

$$X_i'(t+1) = X_i(t+1) + r[X_{i,best} - X_i(t)] \tag{20}$$

where $X_i'(t+1)$ is the position after updating, r is the random number between 0 and 1, and $X_{i,best}$ is the optimal position of the ith grey wolf individual in the t generations.

3.3. Crossover Operation

In order to increase population diversity and prevent local convergence, crossover operation is introduced. Firstly, a pseudo-random number between 1 and d is generated as the dimension number of grey wolf individuals to be exchanged. Random probability is then compared with crossover probability. If the random probability is less than the crossover probability, the crossover operation is performed. The process is shown in Figure 3.

After the crossover operation obtains a new individual, the greedy rule is used to select a superior individual. The fitness of these individuals is compared with the individual before crossover operation, and the betterindividuals are retainedin the update so that the population evolves in an optimal direction.

Through the improvement of three aspects, a new HGWO algorithm is obtained, as shown in Figure 4.

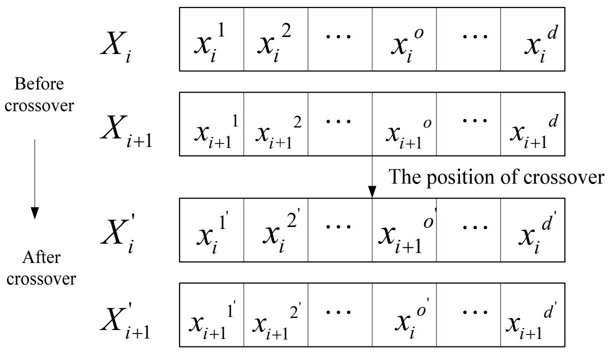

X_i
x_i^1 x_i^2 ⋯ x_i^o ⋯ x_i^d
Before crossover
X_{i+1}
x_{i+1}^1 x_{i+1}^2 ⋯ x_{i+1}^o ⋯ x_{i+1}^d
The position of crossover
X_i'
$x_i^{1'}$ $x_i^{2'}$ ⋯ $x_{i+1}^{o'}$ ⋯ $x_i^{d'}$
After crossover
X_{i+1}'
$x_{i+1}^{1'}$ $x_{i+1}^{2'}$ ⋯ $x_i^{o'}$ ⋯ $x_{i+1}^{d'}$

Figure 3. Crossover operation.

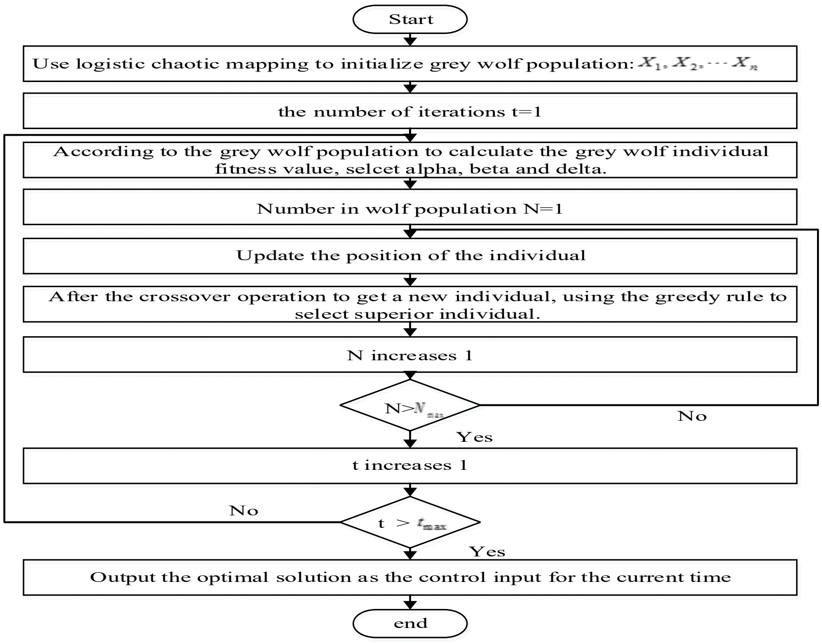

Start
Use logistic chaotic mapping to initialize grey wolf population: $X_1, X_2, \cdots X_n$
the number of iterations t=1
According to the grey wolf population to calculate the grey wolf individual fitness value, selcet alpha, beta and delta.
Number in wolf population N=1
Update the position of the individual
After the crossover operation to get a new individual, using the greedy rule to select superior individual.
N increases 1
N>N_{max}
No
Yes
t increases 1
t > t_{max}
No
Yes
Output the optimal solution as the control input for the current time
end

Figure 4. Flow chart of hybrid grey wolf optimization (HGWO).

4. NMPC Controller Design for Aircraft Engines with HGWO

The schematic diagram of the NMPC controller designed in this paper is shown in Figure 5.

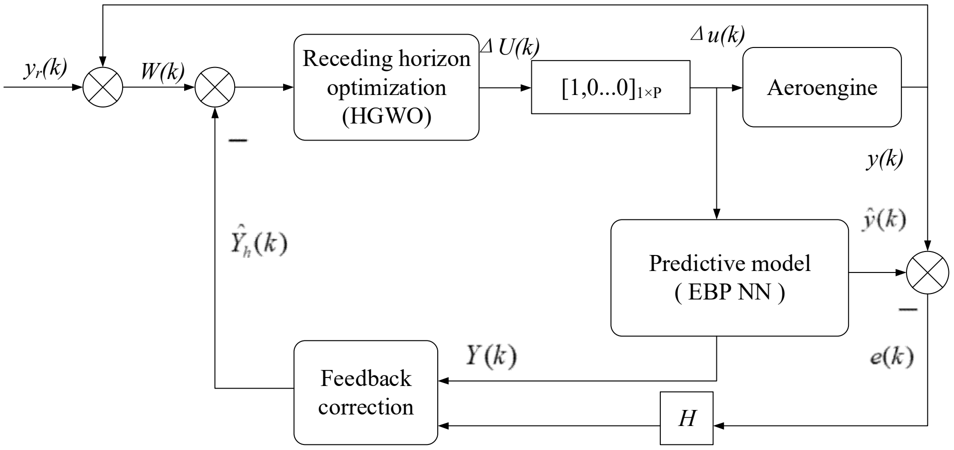


Figure 5. Nonlinear model predictive control (NMPC) schematic diagram.

4.1. Prediction Model

The corresponding output data (low pressure turbine speed) are obtained using Gaussian white noise as input data (fuel quantity). Based on the acquired input and output data, the BP method is used to train the corresponding neural network model off-line, and the recursive method is then used to establish the prediction model.

Where y and u represent n_l and w_f, respectively, i.e., low pressure turbine speed and fuel flow, the BP neural network model is

$$y(k) = f_{nn}[y(k-1), \ldots, y(k-n_1), u(k-1), \ldots, u(k-n_2)]. \tag{21}$$

In this formula, k represents the current moment, and n_1 and n_2 represent the order of output y and input u, respectively.

Based on Equation (21), the dynamic recursive neural network prediction model is established using the recursive method:

$$\begin{aligned} \hat{y}(k+1) &= f_{nn}[y(k), \ldots, y(k-n_1+1), u(k), \ldots, u(k-n_2+1)] \\ \hat{y}(k+2) &= f_{nn}[\hat{y}(k+1), \ldots, y(k-n_1+2), u(k+1), \ldots, u(k-n_2+2)] \\ &\vdots \\ \hat{y}(k+P) &= f_{nn}[y(k+P-1), \ldots, y(k-n_1+P+1), u(k+P-1), \ldots, u(k-n_2+P+1)] \end{aligned} \tag{22}$$

where P is the prediction horizon, and $\hat{y}$ is the output value of the prediction model.

4.2. Feedback Correction

It is inevitable that there is an error between the prediction model's output value and the actual output value. Therefore, we can use the prediction error at time k to correct the output value of the prediction model and improve the prediction accuracy.

The prediction error of the k moment is

$$e(k) = y(k) - \hat{y}(k). \tag{23}$$

Suppose the feedback correction weight matrix is $H = [h_1, h_2, \ldots, h_P]$, and each element of H takes a number between 0 and 1. The corrected prediction output is then

$$\hat{Y}_h = Y(k) + H \cdot e(k) \tag{24}$$

where $Y(k) = [\hat{y}(k+1), \hat{y}(k+2), \ldots, \hat{y}(k+P)]$ is the prediction output matrix before correction, and $\hat{Y}_h(k) = [\hat{y}_h(k+1), \hat{y}_h(k+2), \ldots, \hat{y}_h(k+P)]$ is the corrected prediction output matrix.

4.3. Reference Trajectory

The reference trajectory of the prediction output can be represented as

$$w(k+i) = c^i y(k) + \left(1 - c^i\right) y_r(k) \tag{25}$$

where y_r is the target value, i.e., the expected relative speed of aircraft engines, and c is the soft coefficient, $0 < c < 1, i = 1, 2, \ldots, P$.

4.4. Controller Obtaining

In the NMPC controller design, in order to guarantee the instruction tracking and reduce fuel consumption in the control process, the objective function is

$$\min J = (Y_h - W)^T Q (Y_h - W) + \Delta U(k)^T R \Delta U(k) \tag{26}$$

$$s.t.\ u_{\min} \le u(k+i) \le u_{\max}$$
$$\Delta u_{\min} \le \Delta u(k+i) \le \Delta u_{\max}$$

where $Q = qI_{P\times P}$ is the weighted matrix of the output error, $R = rI_{M\times M}$ is the weighted matrix of the control increment, $I_{P\times P}$ is the identity matrix with P dimension, $I_{M\times M}$ is the identity matrix with M dimension, M is the control horizon, $W(k) = [w(k+1), w(k+1), \ldots, w(k+P)]$ is the reference trajectory output vector, and $\Delta U(k) = [\Delta u(k), \Delta u(k+1), \ldots, \Delta u(k+M-1)]$ is the control increment vector with $\Delta u(k) = u(k) - u(k-1)$.

In each solution process, the receding horizon optimization will obtain an optimal control incremental sequence $\Delta U(k)$ and take the first element of the sequence, and we can then obtain the optimal control input of the aircraft engines at the current time:

$$u(k) = \Delta u(k) + u(k-1).$$

The steps of applying the HGWO algorithm into the receding horizon optimization to solve the controller are as follows:

Step 1: Determine the population size and the maximum number of iterations, set the constraints of the aircraft engines, set the prediction horizon and the control horizon, and set the crossover probability.

Step 2: According to the constraints, use the logistic chaotic sequence to generate the individual location of the grey wolf population.

Step 3: According to the grey wolf position X to calculate the grey wolf individual fitness value, i.e., the reciprocal of the performance index, select α, β, and δ.

Step 4: Based on the current positions of α, β, and δ, update the position of the other individuals in the population using Equation (2).

Step 5: After the crossover operation to obtain new individuals, use the greedy rule to select superior individuals.

Step 6: When either the optimal condition is satisfied or the number of iterations is enough, the iteration is ended and the optimal solution is output as the control input for the current time; otherwise, return to Step 3.

5. Simulations

Based on an aircraft engine component level model, a simulation experiment of the designed controller was carried out.

The working point of the engine was on the ground, namely, height and Mach number were both zero. The input was the fuel supply, and the output was the relative speed of the low-pressure rotating shaft. In order to fully stimulate the dynamic characteristics of the engine, the Pseudo Random Binary Signal (PRBS) was used as the fuel supply to the engine. The sampling time was 0.04 s. The input and output data were obtained. One thousand five hundred sets of input and output data were selected and normalized.

According to the parameters in the elastic BP neural network and the GWO algorithm in Section 2.2, the GWO-EBPNN model of the engine was established. Figures 6–8 show the error curve between the test output and the expected output of the low-pressure spool speed under the GWO-EBPNN, the elastic BP neural network, and the conventional BP neural network, respectively. It can be seen that the average error between test output and expected output in Figure 6 is the smallest, while for elastic BPNN and classic BPNN the results are almost two times larger than those of the GWO-EBPNN.

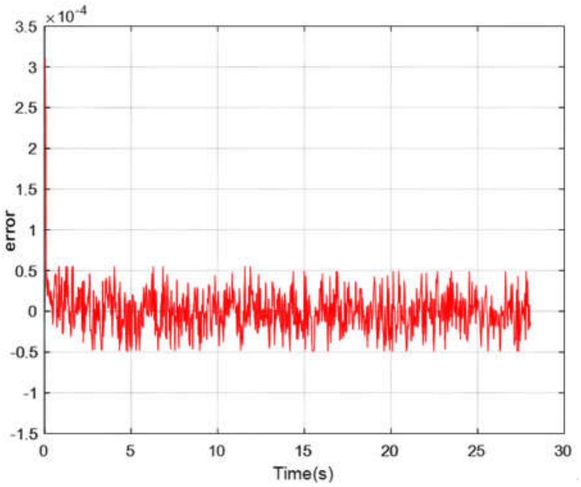


Figure 6. Error curve between test output and expected output of low-pressure spool speed under GWO-EBPNN.

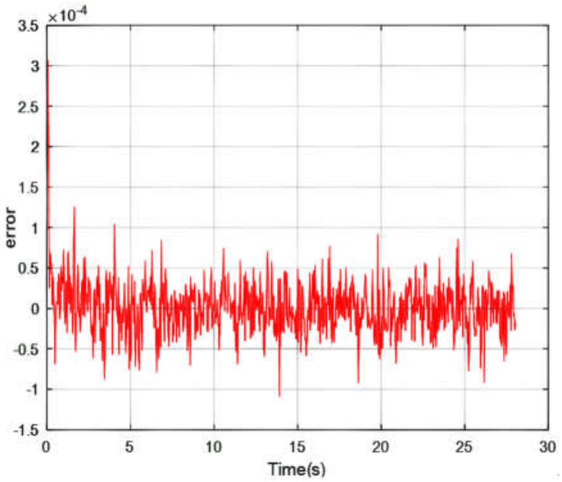


Figure 7. Error curve between test output and expected output of low-pressure spool speed under the elastic BP neural network.

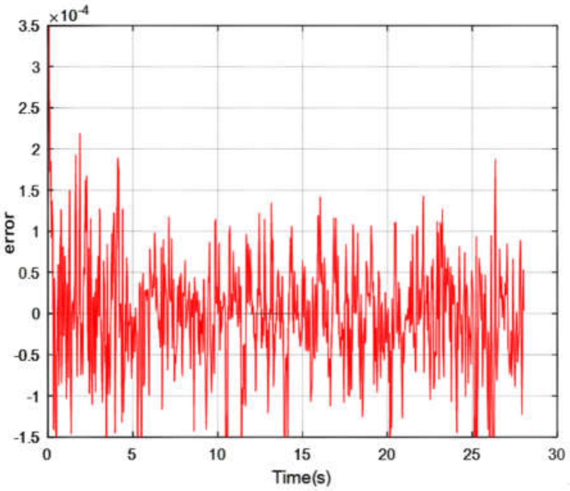


Figure 8. Error curve between test output and expected output of low-pressure spool speed under the conventional BP neural network.

In order to test the ability of the new improved grey wolf optimization (HGWO) algorithm proposed in this paper to solve function optimization problems, two standard optimization test functions Sphere and Rastrigin were tested. Sphere test function is a typical unimodal function, while Rastrigin function is a representative multimodal function. It is of high reference value to evaluate the performance of the algorithm with these two functions. The specific form of the Sphere function is $f_1(x) = \sum_{i=1}^{d} x_i^2$, and the specific form of the Rastrigin function is $f_1(x) = \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$. For the two test functions, the initial parameters of the optimization algorithm are as follows: the dimension is 30, the optimization space is [−50,50], the minimum value is 0, the population is 80, the stagnation algebraic threshold is 10, and the number of iterations is 1000.

To make a comparison, the typical GA is used to optimize the two test functions as well. The GA algorithm adopts binary coding, the chromosome length is 630, and the crossover probability and mutation probability are 0.8 and 0.005, respectively. The optimization accuracy of the test algorithm is set to 1×10^{-8}.

The relationship between the optimal values and the number of iterations is shown in Figures 9 and 10.

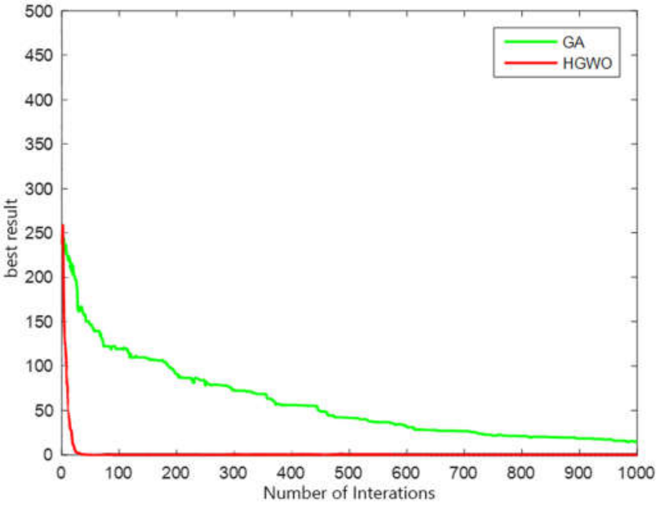


Figure 9. Sphere function—objective function value vs. iteration number.

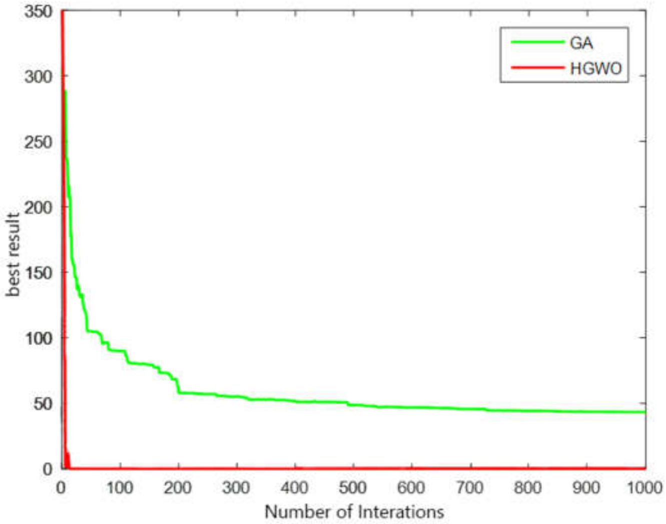


Figure 10. Rastrigin function—objective function value vs. iteration number.

In Figure 9, after 30 iterations, HGWO did not change the best results, while the GA did not arrive at the targeted value within 1000 iterations. In Figure 10, the iterative process in HGWO was easily converged and GA failed to optimize the best result. Therefore, the GA algorithm easily falls into a local extremum and cannot complete the optimization well; it needs to continue iterating or increase the population. The HGWO algorithm can complete the optimization well, has a fast convergence speed, avoids falling into a local optimum, and possesses good optimization performance.

Therefore, the HGWO algorithm is useful to improve the real-time performance of the NMPC controller for aircraft engines.

In the NMPC controller, prediction horizon $P = 10$, control horizon $M = 2$, the feedback correction matrix $H = [1, 1, \ldots, 1]_{1\times 10}$, the soft coefficient $c = 0.3$, the weighted value of the output error $q = 1$, and the weighted value of the control incremental $r = 0.01$. According to the actual engine working condition, the input constraint is taken as $0.05 \leq u \leq 0.32$.

In the HGWO algorithm, the grey wolf individual population is 80, the iteration number is 100, and the crossover probability is 0.2.

For comparison, the NMPC controller design with a GA algorithm is applied to the considered aircraft engine system. The GA algorithm is binary coded. The chromosome length, crossover probability and mutation probability are 630, 0.6 and 0.005, respectively.

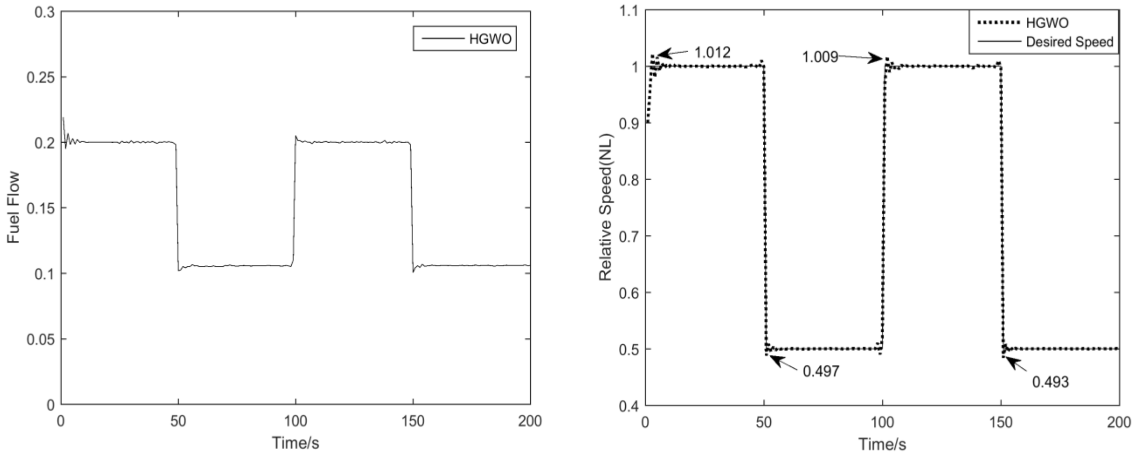
Figures 11 and 12 show the tracking trajectory of the low-pressure spool speed under HGWO and GA optimization algorithms, respectively.

In Figures 10 and 11, one can see that the convergence speed of the HGWO algorithm is higher than that of the traditional GA.

The variation of the GA-NMPC control law is relatively high, and there is a slight vibration, so the control effect is not good, while the control law of HGWO-NMPC has a very good dynamic performance whether in the initial time or at step points.

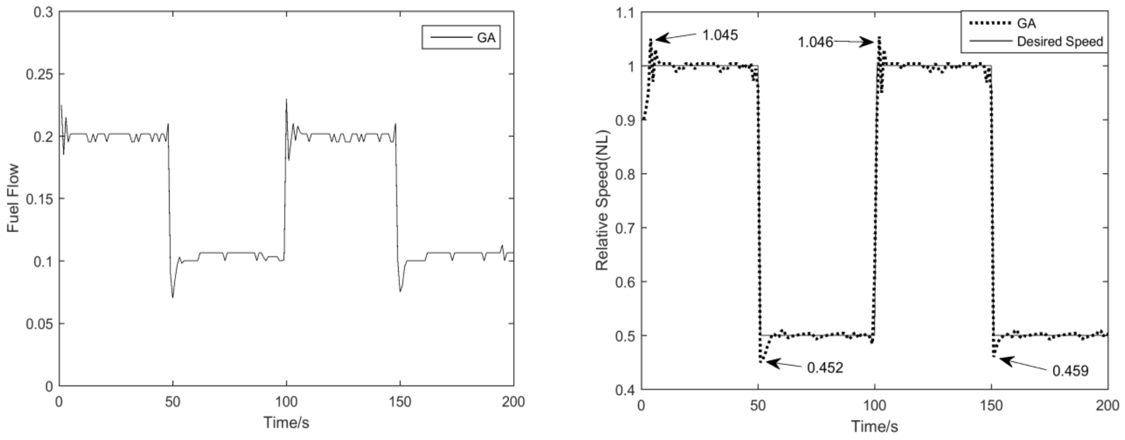
The output of the NMPC system based on the GA algorithm can track the expected response. However, in the acceleration process, the maximum overshoot is 0.045, and the maximum droop value is 0.452 during the deceleration process. The overshoot and the droop amount are relatively large, and in the process of control, local convergence occurs continuously around the expected value. It is impossible to track the expected value accurately, and the control performance is relatively poor.

The overshoot and the droop under the control of HGWO-NMPC are smaller, the maximum overshoot value is 0.012, and the maximum droop value is almost 0. HGWO-NMPC has little oscillation in the initial stage of the accelerating response.



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

Figure 11. The simulation results of NMPC based on HGWO. (**a**) Fuel flow curve; (**b**) relative speed curve.



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

Figure 12. The simulation results of NMPC based on a GA. (**a**) Fuel flow curve; (**b**) relative speed curve.

Therefore, the presented HGWO-NMPC method in this paper shows that the considered aircraft engine control system has fast convergence, high precision, good dynamic performance, and an ability to handle constraints well. It can thus handle the requirement of control constraints and the performance optimization requirements of aircraft engines.

6. Conclusions

In this paper, a new nonlinear model predictive control method based on an elastic BP neural network with hybrid grey wolf optimization (HGWO) is proposed. The elastic BP neural network is used to train a prediction model of aircraft engines, and the grey wolf optimizer (GWO) is used to improve the selection of initial parameters in the elastic BP neural network. The accuracy of the network modeling was increased. By introducing a logistics chaotic sequence, an individual optimal search mechanism, and cross operation into the traditional GWO algorithm, a new HGWO algorithm is achieved, and HGWO is used in relation to nonlinear model predictive control (NMPC) of aircraft engines. Simulation results show that, under the proposed NMPC, aircraft engines have good control performance. Future work can involve analyzing the stability of NMPC, which is an important part of control system design. In terms of the work conditions of aircraft engines, the control lag during the operation should be considered.

Author Contributions: Conceptualization and methodology: L.X.; formal analysis and software: M.X. and Y.C. (Yusheng Chen); investigation and data curation: Y.C. (Yuhan Chen).

Funding: This research was funded by National Natural Science Foundation of China, grant number 51876089.

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

References

1. Ding, B. *Predictive Control Theory and Method*; China Machine Press: Beijing, China, 2008; pp. 11–12.
2. Zou, Z.; Wang, Z. Research Status and Development Trend of Nonlinear Model Predictive Control Technology. *Comput. Appl. Chem.* **2018**, *35*, 7–17.
3. Du, X.; Guo, J. Design of model predictive controller for commercial turbofan engine. *Aeroengine* **2013**, *39*, 27–30.
4. Xiao, L.; Zhu, Y. Combined model based nonlinear predictive control for turboshaft engine. *J. Propuls. Technol.* **2012**, *33*, 283–287.
5. Wiese, A.P.; Blom, M.J. Model reduction and MIMO model predictive control of gas turbine systems. *Control Eng. Pract.* **2015**, *45*, 194–206. [CrossRef]
6. Yao, W.; Sun, J. Nonlinear model predictive control for turboshaft engine. *Acta Aeronaut. Astronaut. Sin.* **2008**, *29*, 776–780.

7. Zhu, Q.; Onori, S.; Prucka, R. Nonlinear economic model predictive control for SI engines based on sequential quadratic programming. In Proceedings of the American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 1802–1807.
8. Zeng, G.; Chen, J.; Dai, Y. Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization. *Neurocomputing* **2015**, *160*, 173–184. [CrossRef]
9. Lapa, K.; Cpalka, K.; Przybyl, A. Genetic Programming Algorithm for Designing of Control Systems. *J. Inf. Technol. Control* **2018**, *47*, 668–683.
10. Yang, J.; Liu, M. Genetic algorithm based nonlinear model predictive control method. *Control Decis.* **2003**, *18*, 141–144.
11. Wang, S.; Shan, S. Nonlinear predictive control based on T-S fuzzy model and particle-swarm optimization. *CIESC J.* **2012**, *63*, 176–187.
12. Zhou, X. Multi-objective predictive control for boiler combustion optimization. *Comput. Simul.* **2013**, *30*, 89–94.
13. Mirjalili, S.; Mirjalili, S.M. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
14. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
15. Bansal, J.C.; Sharma, H.; Jadon, S.S.; Clerc, M. Spider Monkey Optimization algorithm for numerical optimization. *Memet. Comput.* **2014**, *6*, 31–47. [CrossRef]
16. Wozniak, M.; Ksiazek, K.; Marciniec, J.; Polap, D. Heat production optimization using bio-inspired algorithms. *Eng. Appl. Artif. Intell.* **2018**, *76*, 185–201. [CrossRef]
17. Singhal, R.; Kumar, R. Receding horizon based greenhouse air temperature control using Grey wolf optimization algorithm. In Proceedings of the Computer and Electronics Engineering (UPCON), Varanasi, India, 9–11 December 2016; pp. 32–37.
18. Madhiarasan1, M.; Deepa, S.N. Long-term wind speed forecasting using spiking neural network optimized by improved modified grey wolf optimization algorithm. *Int. J. Adv. Res.* **2016**, *4*, 356–368. [CrossRef]
19. Verma, S.K.; Yadav, S.; Nagar, S.K. Optimization of fractional order PID controller using grey wolf optimizer. *J. Control Autom. Electr. Syst.* **2017**, *28*, 314–322. [CrossRef]
20. Xu, D.; Ding, S. Research on improved GWO-optimized SVM-based short-term load prediction for cloud computing. *Comput. Eng. Appl.* **2017**, *53*, 68–73.
21. Luo, J.; Tang, B. Application of novel grey wolf optimization algorithm in function optimization. *J. Lanzhou Univ. Technol.* **2016**, *42*, 96–101.
22. Jitkongchuen, D. A hybrid differential evolution with grey wolf optimizer for continuous global optimization. In Proceedings of the ICITEE, Chiang Mai, Thailand, 29–30 October 2015; pp. 51–54.
23. Zhang, X.; Tu, Q.; Kang, Q.; Cheng, J. Hybrid Optimization Algorithm Based on Grey Wolf Optimization and Differential Evolution for Function Optimization. *Comput. Sci.* **2017**, *44*, 93–98.
24. Xiao, L.; Du, Y.; Hu, J.; Jiang, B. Sliding mode fault tolerant control with adaptive diagnosis for aircraft engines. *Int. J. Turbo Jet Eng.* **2018**, *35*, 49–57. [CrossRef]
25. Lyantsev, O.D.; Breikin, T.V.; Kulikov, G.G.; Arkov, V.Y. On-line performance optimisation of aero engine control system. *Automatica* **2003**, *39*, 2115–2121. [CrossRef]
26. Capizzi, G.; Sciuto, G.L.; Napoli, C.; Shikler, R.; Wozniak, M. Optimizing the Organic Solar Cell Manufacturing Process by Means of AFM Measurements and Neural Networks. *Energies* **2018**, *11*, 1221. [CrossRef]
27. Wozniak, M.; Polap, D. Adaptive neuro-heuristic hybrid model for fruit peel defects detection. *Neural Netw.* **2018**, *98*, 16–33. [CrossRef] [PubMed]
28. Wozniak, M.; Polap, D. Hybrid neueo-heuristic methodology for simulation and control of dynamic systems over time interval. *Neural Netw.* **2017**, *93*, 45–56. [CrossRef]
29. Blum, E.K.; Li, L.K. Approximation theory and feedforward networks. *Neural Netw.* **1991**, *4*, 511–515. [CrossRef]
30. Ye, G.; Huang, X. A self-cited pixel summation based image encryption algorithm. *Chin. Phys. B* **2017**, *26*, 131–138. [CrossRef]