The Aero-Engine Component-Level Modelling Research Based on NSDE Hybrid Damping Newton Method

Chen Wang, Ximing Sun, and Xian Du

Dalian University of Technology, School of Control Theory and Control Engineering, China

Correspondence should be addressed to Ximing Sun; sunxm@dlut.edu.cn

Received 18 May 2022; Accepted 1 August 2022; Published 10 September 2022

Copyright © 2022 Chen Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Advanced aero-engine component-level models are characterized by strong nonlinearity and multivariate, and traditional iterative algorithms cannot meet the requirements of convergence, real-time, and accuracy at the same time. To improve the convergence and alleviate the initial value dependence, a hybrid damped Newton algorithm based on the neighborhood based speciation differential evolution (NSDE) is proposed in this paper for solving the aero-engine component-level model. The computational efficiency and convergence of the hybrid damped Newton algorithm and NSDE hybrid damped Newton algorithm under four typical steady-state operating point conditions are analyzed, and then, the accuracy of the model is verified. It is demonstrated that the hybrid damped Newton method has the advantage of low initial value sensitivity and high computational efficiency under large deviation conditions. The hybrid damped Newton method is more efficient than the Broyden algorithm in terms of iterative efficiency, faster than the traditional N-R algorithm in terms of computation speed, and has the highest computational convergence rate under the four typical operating conditions, but it cannot eliminate the initial value dependence. The NSDE hybrid damped Newton method offers high simulation accuracy and greatly increases computational real-time performance under large deviation conditions, and the maximum error between the numerical simulation results and the experimental reference value is 8.1%. This study provides advanced theoretical support for component-level modelling and has certain engineering application value.

1. Introduction

At present, nonlinear component-level models of aero-engine are the most commonly utilized in research on analytical residuals, fault diagnostics, aircraft design, and control systems. It is well documented that engine modelling and knowledge of power system characteristics account for 80% of the work in aero-engine control and monitoring system research [1]. Modern advanced engine models have more operating modes and adjustable components, which require higher accuracy and convergence of component-level model solving. A compromise between accuracy and real-time is required in modelling while the increased demand for real-time models in engine semiphysical testing and control systems. Therefore, determining ways to increase the iteration algorithm’s convergence and real-time performance while fulfilling accuracy criteria has become a pressing issue.

Iterative algorithms for aero-engine models have received a lot of attention from academics. Early iterative algorithms included parametric loop methods such as SPEEDY and CARPET and equilibrium loop methods such as AFQUIR and DSPOOL with nested loops [2], which were mainly used for engine simulation under steady-state conditions, but the iteration time was long and could not be applied to dynamic simulation. Gradient-based optimization algorithms, such as the steepest descent method, the N + 1 residual method, the Newton-Raphson (N-R) method, and the Broyden method [3], were later used in the numerical calculation of aero-engine models because they do not require nested loops and are multiple iteration methods based on initial values and are widely used in steady-state and dynamic engine model calculations. The N-R method is characterized by strong second-order convergence and is widely utilized in the USAF general-purpose simulation programs SMOTE, GENENG, and DYENG [4], but it needs
repetitive Jacobi matrix computations and too many component calculations, resulting in poor real-time performance. The Broyden quasi Newton technique updates the correction function using a recursive formula, which eliminates the problem of recurrent Jacobian matrix calculation in the N-R method and considerably increases real-time performance. This approach is used in the primary simulation programs NASA’s NCP and NLR's GSP and terts engine models [5, 6], although the convergence stability is weak. The N-R and Broyden techniques are locally convergent algorithms that rely heavily on the initial value of the iterations, restricting their use to "big deviation" dynamic engine modelling and simulation. Scholars have proposed initial value fitting, finite-domain optimization search, component characteristic expansion, and variable step size [7–12], as well as tried hybrid algorithm schemes of N-R and Broyden method [13, 14] to improve convergence and real-time performance from the perspective of improving initial conditions and algorithm update function, but cannot solve the problem of global convergence from the root.

Many researchers have introduced advanced intelligent optimisation algorithms in recent years. Several scholars have used genetic algorithms [15, 16] (GA) and hybrid genetic algorithms [17] to solve engine models, which can improve global convergence due to the lack of dependence on initial values, but the GA has more algorithmic control parameters and cannot meet the requirements of real-time. Yang et al. [18] coupled the neural network algorithm with the N-R method, utilizing the neural network to approximate the inverse function of the model and the neural network solution result as the initial value of the N-R method, which increased real-time performance and convergence. Several scholars have employed the PSO [19] algorithm, but the computational efficiency of PSO depends on the choice of algorithm parameters and is prone to falling into local optimum solutions. The improved PSO [20] solves the engine model and improves convergence, although it still suffers from the problem of low solution rate. To increase the model's real-time performance, the improved hybrid particle swarm [21] approach combines the advantages of fast calculation efficiency and superior global convergence. However, because the PSO method includes numerous changeable parameters and varied suitable values for different operating situations, it cannot adapt to the computation of a large envelope range. DE algorithm [22] is another evolutionary algorithm after GA with simple structure, few control parameters, and easy implementation using real number encoding and theoretically proven convergence [23]. The nondominated sorting difference algorithm [24] (NSDE) is a kind of evolutionary algorithm, which utilizes the strategy of neighborhood search in evolutionary programming to adjust the scaling factor, improve the global search capability, and maintain the ideal solution efficiency [24].

For the past few years, research into engine modelling has mainly focused on improvements to the modelling of key components, including inlet ejector [25], multiple variable geometries, and combustion chamber design. Scholars have applied various KF-based methods [26–29] to estimate engine performance parameters and using various neural networks [30], support vector machine [31], extreme learning machine [32–34] to increase the accuracy and the real-time performance of the model in full life circle. Less analysis of the accuracy, convergence, and real-time performance of iterative algorithms for engine models, especially the application of hybrid intelligence algorithms. At this stage, NSDE is not used as an iterative algorithm in aero-engine intelligent algorithms. Due to the strong nonlinearity and multi-iterative variables of the aero-engine model, intelligent algorithms suffer from poor real-time performance and computational accuracy. Therefore, combining the traditional gradient-based optimisation algorithm with the intelligent algorithm can combine the advantages of both, while taking into account the problems of convergence, real-time, and computational accuracy.

To resolve the challenges of convergence and real-time performance, a hybrid damped Newton algorithm based on NSDE has been designed to solve the aero-engine component level model. The effectiveness of the hybrid damped Newton algorithm is first validated by comparing it to traditional algorithms in terms of computational efficiency and convergence; second, on the basis of the hybrid damped Newton algorithm, the NSDE algorithm is integrated to further improve the convergence of the engine model; and finally, the simulation results of the engine model are compared to the experimental results to verify its effectiveness.

2. Method

2.1. Integrated Inlet/Engine Component Level Model. The principles of component-level model solution for aero-engine are described in this part. Aero-engine models are multivariable, nonlinear, time-varying systems that frequently employ component-level aero thermodynamic models. Conventional engine models of the inlet system are mostly calculated using idealised models and empirical formulas, which cannot match the calculation accuracy and fidelity criteria. Taking the shock structure and resistance calculation approach into consideration, the inlet is estimated utilizing quasi one-dimensional flow theory and the engine fundamental calculation procedure represented in Figure 1. Flow continuity, static pressure balance, and power balance between components are all required when calculating off-design performance. Flow continuity means that the flow from the upstream component to the current component is the same as the flow calculated by the current component itself; static pressure balance means that when the two incoming flows are mixed, the static pressure should be equal; and power balance means that the power consumed on the same shaft should be equal to the power generated. The residuals of the balancing equations are expressed in terms of $e$. The iterative variables are selected and coworking equations are solved in association expressed as

$$
\begin{align*}
&f_1(x_0, x_1, x_2, \ldots, x_8) = e_1, \\
&f_2(x_0, x_1, x_2, \ldots, x_8) = e_2, \\
&\cdots, \\
&f_8(x_0, x_1, x_2, \ldots, x_8) = e_8.
\end{align*}
$$

(1)
To satisfy the flow balance of the inlet and engine, the residuals of the inlet and fan flow are used as balance equations, and the ratio of the flow coefficient/total pressure recovery factor is chosen as the iteration’s starting guess value. If the number of iterative variables and residuals is both identical, the issue transforms into a nonlinear implicit set of equations with unknown independent variables, and the engine model is considered to have a reliable solution when the residuals tend to zero.

2.2. Basic Principles of the NSDE Hybrid Damped Newton Method. To solve the nonlinear implicit system of the aero-engine model, the hybrid damped Newton method and the adaptive NSDE hybrid damped Newton method are designed, and the principles of both methods are described in this section.

2.2.1. Hybrid Damped Newton Method. The hybrid damped Newton method is an enhanced N-R technique with a damping factor in which the nonlinear equation is enlarged according to the Taylor series and the Newton method is an enhanced N-R technique with a damping factor in which the nonlinear equation is enlarged according to the Taylor series and the adaptive NSDE hybrid damped Newton method is used as the main algorithm, and the Broyden quasi-Newton technique is utilized for computation to minimize the quantity of iterative calculation. Two iterative techniques are utilized alternatively to effectively minimize the number of aero-thermodynamic iterations based on the indices of specified convergence speed and range. If the iteration result of the nth hybrid Newton’s method is $X^{(n)}$, the Broyden quasi-Newton method is used in the intermediate m steps, and the damped Newton’s method is used again from $n + 1$ step, using the following iteration format:

$$
\frac{\partial E^k}{\partial X} = \frac{\Delta E^k}{\Delta X^k}.
$$

(5)

The hybrid damped Newton method indicates that the damped Newton method is used as the main algorithm, and the Jacobian matrix is not adjusted immediately after each iteration, but the Broyden quasi-Newton technique is utilized for computation to minimize the quantity of iterative calculation. Two iterative techniques are utilized alternatively to effectively minimize the number of aero-thermodynamic iterations based on the indices of specified convergence speed and range. If the iteration result of the $n$th hybrid Newton’s method is $X^{(n)}$, the Broyden quasi-Newton method is used in the intermediate m steps, and the damped Newton’s method is used again from $n + 1$ step, using the following iteration format:

$$
X^{(n)} = X^{(n)},
$$

(6)

$$
X^{(n,j)} = X^{(n,j-1)} - \alpha_{j-1} B_{j-1} \cdot F\left(X^{(n,j-1)}\right),
$$

$$
B_j = B_{j-1} + \beta_{j-1} \cdot \frac{\left(y_{j-1} - B_{j-1} s_{j-1}\right) \cdot S_{j-1}^T}{S_{j-1}^T \cdot S_{j-1}},
$$

$$
X^{(n+1)} = X^{(n,m)}.
$$

In the formula, $X^{(n,j)}$ represents the $j$th use of the Broyden quasi-Newton method in the $n$th Newton iteration; $y_{j-1} = F(X^{(n,j-1)}) - F(X^{(n,j-1)})$; $s_{j-1} = X^{(n,j)} - X^{(n,j-1)}$; $\beta_{j-1}$ and $\alpha_{j-1}$ are...
damping factors; and $B_0$ takes the jacob matrix of $X^{(n,0)}$ point.

$$
Eva^k = \sum_{n=1}^{N} \frac{\Delta e(X^k_n)}{\Delta X^k_n},
$$

(7)

$$
Eva^k \geq \lambda_1 \cdot Eva^{k-1},
$$

(8)

$$
Eva^k \geq \lambda_2 \cdot Eva^{k-1}.
$$

(9)

The specific switching conditions are shown in Eqs. (7)–(9), where $Eva$ indicates the update efficiency at each step, and the Broyden algorithm can be used when $Eva$ varies widely, while the N-R algorithm is used when $Eva$ varies less. $\lambda_1$ indicates the condition for switching the Broyden algorithm, and $\lambda_2$ indicates the condition of whether to jump out of the Broyden algorithm. The calculation process is shown in Figure 2.

2.2.2. Adaptive NSDE Algorithm Method. The adaptive NSDE algorithm, which is based on the differential evolution algorithm, can adaptively adjust the scaling factor of the variation process and the crossover rate of the crossover process to improve global convergence.

Initializing a population: determining the value range of each iteration variable in the adaptive differential evolution algorithm based on the characteristics of the engine components and working condition limitations, as the variable value ranges $x_{ji}$ and $x_{ji}'$ of an initial population of the adaptive differential evolution algorithm; setting the initial population number NP of the differential evolution algorithm, and randomly generating the initial population.

$$
x_i(0) = \begin{cases} x_{ji}\left(x_{ji}(0) \leq x_{ji}', \right) \\ j = 1, 2, \cdots, NP; j = 1, 2, \cdots, D \end{cases}
$$

(10)

$$
x_{ji}(0) = x_{ji} + \text{rand} (0, 1) \cdot \left(x_{ji}' - x_{ji}\right),
$$

(11)

wherein $x_{ji}(0)$ represents the $j^{th}$ gene of the $i^{th}$ individual in the 0 generation; NP represents the population size; and rand $(0, 1)$ represents random numbers evenly distributed in an interval of $(0, 1)$; selecting a residual $e[m]$ of each coworking equation (the number of equations is $m$) of the engine model, and taking $O_j = 1/m \sum_{j=1}^{m} e(j)$ as a fitness function to serve as an optimization objective function.

Adaptive mutation strategies: attempting to use two different mutation strategies, and introducing probability $p$ to control the selection; conducting self-adapting by $p$ according to the learning experience in the calculation process, and obtaining a scaling factor $F$ based on a Gaussian distribution function; initializing $p = 0.5$; after the population fully evolves in the current round, recording the number $n$s1 of individuals which enter the next generation and the number $n$f1 of individuals not entering the next generation under the condition of $U_i(0, 1) < p$ from $v_i$; and the number $n$s2 of individuals which enter the next generation and the number $n$f2 of individuals not entering the next generation under the condition of $U_i(0, 1) \geq p$ from $v_i$, wherein $x_{best}$ represents a current optimal individual, respectively, recording the two groups of numbers for 50 generations, called as a "learning cycle"; and when the probability $p$ is updated after the learning cycle, resetting the values of $n$s1, $n$s2, $n$f1, and $n$f2, with the formulas of the adaptive mutation strategies shown as follows:

$$
v_i = \begin{cases} x_{i1} + F_i \cdot (x_{i2} - x_{i3}), U_i(0, 1) < p, \\ x_{best} + F_i \cdot (x_{i1} - x_{i2}) \

F_i = N_i(0.5, 0.3),
$$

(12)

$$
p = \frac{n1 \cdot (ns2 + nf2)}{ns2 \cdot (ns1 + nf1) + ns1 \cdot (ns2 + nf2)}.
$$

(13)

Crossover operation: the adaptive evolution crossover operation is conducted for dimensions. The new individual has the probability of $CR$ to select the dimension in $v_i(j)$, and other dimensions select $x_i(j)$, wherein the adaptive crossover rate $CR$ allocates a crossover rate $CR$, for each individual, $CR_{ns}$ is initialized as 0.5, and $CR_{ns}$ is updated every 5 generation. In each generation, the value of $CR_{ns}$ and a subgeneration successfully enter the next generation; corresponding $CR$ enters an array $CR_{rec}$ and is updated every 25 generations; and after the update, $CR_{rec}$ is emptied. In Eq. (17), $|CR_{rec}|$ represents the length of the array, and when a better individual appears in each generation, the result is inserted into $CR_{rec}$, and eventually, the mean is calculated by summing from 1 to $|CR_{rec}|$, and the value of $CR_{ns}$ is updated.

$$
u_j = \begin{cases} v_j(j), U_i(0, 1) < CR, \text{or } j = j_{rand}, \\ x_j(j), \

CR_i = N_i(CRm, 0.1)
$$

(15)

(16)

$$
CRm = \sum_{k=1}^{CR_{rec}} CR_{rec}(k).
$$

(17)

Selection operation: the selection operation is selection of a better individual from a mutated individual $u_i$ and an old individual $x_i$ by using a greedy algorithm to generate a new individual $x_i'$.

$$
x_i' = \begin{cases} u_i, f(u_i) \leq f(x_i), \\ x_i,

2.2.3. The NSDE Hybrid Damped Newton Method. The NSDE hybrid damped Newton method combines the hybrid damped Newton method and the NSDE algorithm, with the number of iterations and the size of the residuals calculated in real time used as a basis for judgement, switching between the two algorithms and thus improving computational efficiency and solution accuracy. The NSDE hybrid damped Newton method's computational flow is depicted in Figure 2. If the hybrid damped Newton method fails to converge within the maximum number of iterations $N_{\text{max}}$, or if the iterations diverge, the calculation is passed to the NSDE algorithm until the
equilibrium residuals are reduced to a certain error value $E_{\text{max}}$. Returning to the hybrid damped Newton method, the iteration ends when the equilibrium equation’s residuals fulfill the error criterion. $N_{\text{loop}}$ represents the actual number of iterations of the hybrid damped Newton method, $N_{2}^{\text{loop}}$ and $N_{2}^{\text{max}}$ denote the actual number of iterations and the maximum number of iterations of the NSDE algorithm, respectively.

\[
\sum_{n=1}^{8} e^{k}[n] \geq \alpha \cdot \sum_{n=1}^{8} e^{k-1}[n] \text{or } N_{\text{loop}}^{1} \geq N_{\text{max}}^{1}, \quad (19)
\]

\[
\max \{ e[k], k = 1, 2, \cdots, 8 \} \leq E_{\text{max}} \text{or } N_{\text{loop}}^{2} \geq N_{\text{max}}^{2}. \quad (20)
\]

### 3. Results and Discussion

#### 3.1. Computational Efficiency and Convergence Analysis of the Hybrid Damped Newton Method

##### 3.1.1. Hybrid Damped Newton Method for Computational Efficiency Analysis

This paper compares the hybrid damped Newton algorithm with the N-R method and Broyden method under different computational initial value conditions and analyzes the computational efficiency by comprehensively analyzing the number of iteration steps and component calculations. In this section, a typical supersonic condition ($\text{Alt} = 12.47, \text{Ma} = 2.0$) is selected, the initial deviation conditions are determined by the deviation rate, and the algorithm iterations’ default step size is set to 0.5.

Figures 3–5 depict the fluctuation of the iterative equations’ equilibrium residuals with the number of component computations and iteration steps for different deviation conditions. This section has defined initial deviation conditions of 0.1, 0.4, and 0.5 as small deviations, medium deviations, and large deviations, respectively. The values of eight variables are determined when the equilibrium equations are calculated to converge, and the value deviation is set as the rate of deviation done relative to the initial values of the eight variables taken. The formula for initial value deviation should be

\[
\sum_{k=1}^{8} \left| \frac{X_{\text{init}} - X_{\text{final}}}{X_{\text{final}}} \right| \times 100\%. \quad (21)
\]
When shown in Figure 3, as the number of component calculations increases, the equilibrium residuals of the three iterative algorithms decline to zero and all achieve a state of convergence. The N-R method requires more component computations than the others since it requires iterative solution of the differential terms of the Jacobi matrix at each iteration step. The Broyden method has the fewest model back iterations calculated, with just 28. As seen in Figure 3(b), the hybrid damped Newton method has the highest number of iterations at 53, whereas the N-R and Broyden methods have a comparable amount of iterations. Under the small deviation condition, all three iterative methods may converge in a limited number of iteration steps, with the Broyden method possessing the fastest computational speed and the N-R and Broyden algorithms being highly efficient in terms of actual iterations.

Figure 4(a) shows that the three algorithms still converge for an initial deviation of 0.4, but the number of component calculations increases, and the equilibrium residuals of the three iterative algorithms decline to zero and all achieve a state of convergence. The N-R method requires more component computations than the others since it requires iterative solution of the differential terms of the Jacobi matrix at each iteration step. The Broyden method has the fewest model back iterations calculated, with just 28. As seen in Figure 4(b), the hybrid damped Newton method has the highest number of iterations at 53, whereas the N-R and Broyden methods have a comparable amount of iterations. Under the small deviation condition, all three iterative methods may converge in a limited number of iteration steps, with the Broyden method possessing the fastest computational speed and the N-R and Broyden algorithms being highly efficient in terms of actual iterations.
computations grows dramatically, with the Broyden method having the lowest number. Figure 4(b) shows that the Broyden method has the highest number of iteration steps, 107, while the hybrid damped Newton method has the lowest number of iteration steps, 37. The Broyden method has the quickest computing speed but the lowest iteration efficiency under the condition of considerable departure from the initial value, while the hybrid damped Newton method has the maximum iteration efficiency.

Figure 5(a) shows that the Broyden method fails to converge for an initial deviation of 0.5, whereas the N-R method converges although the equation residuals are slow to reduce. The number of iterations for the hybrid damped Newton method is less than that of the N-R method, as seen in Figure 5(b). When the initial deviation is large, the Broyden algorithm can no longer guarantee convergence, and the N-R algorithm can converge but at a slow rate. The hybrid damped Newton method’s iteration efficiency and computational speed remain steady at the optimal level, and the computational effect is the best.

To examine the computational speed and iteration efficiency of the three different algorithms under different initial deviation and step size settings, four different flight conditions (Case1: \( \text{Alt} = 0 \text{ km}, \text{Ma} = 0 \), Case2: \( \text{Alt} = 0.06 \text{ km}, \text{Ma} = 0.3 \), Case3: \( \text{Alt} = 9.96 \text{ km}, \text{Ma} = 1.5 \), Case4: \( \text{Alt} = 12.47 \text{ km}, \text{Ma} = 2.0 \)) were used, as shown in Figure 6.

The initial deviations were set to 0-0.9, and the step sizes were set to 0.1-1.0, with each box-line plot reflecting the set of data for each flight condition. As shown in Figure 6(a), the hybrid damped Newton method has the lowest mean values for the number of model returns and iteration steps in Case1, 85.778 and 17.667, indicating that the algorithm has the fastest average computation speed and the highest iteration efficiency. The hybrid damped Newton method has the smallest box line plot area for the number of model returns and iteration steps compared to the other two algorithms, implying that the median and mean results of the hybrid damped Newton method are close to each other. The mean values of the other two algorithms are larger than the median, which is caused by the high number of outliers in the Broyden algorithm and the excessive number of iterations back in the N-R method in the large deviation condition.

The results for the Case2-4 conditions are shown in Figures 6(b)–6(d). The mean values of the number of component calculations and iteration steps of the three algorithms gradually increase as the flight altitude and Mach number increase, and the difference between the average and the median value gradually increases, indicating that the overall computational efficiency decreases and stability become less stable. In terms of computing speed, the Broyden method is comparable to the hybrid damped Newton method, but its iterative efficiency is lower and the number of outliers is highest. In summary, the hybrid damped Newton method can show the ideal calculation speed and the best iteration efficiency in the four operating conditions.

The above analysis demonstrates that the hybrid damped Newton method may maintain high computing efficiency and convergence even under significant deviation conditions, demonstrating that the algorithm is less sensitive to different initial conditions. The computational speed of the hybrid damped Newton method is comparable to the Broyden method, but the iteration efficiency is higher; when compared to the traditional N-R method, the computational speed of the hybrid damped Newton method is significantly improved, and the iteration efficiency is also optimized.
3.1.2. Hybrid Damped Newton Method for Convergence Analysis. This section examines the convergence of the three iterative algorithms under four typical operating settings with varying initial conditions. The same data set as in Figure 6 is still used in Figure 7 to characterize the convergence of the various iterative algorithms using hotspot plots. The heat map shows the iteration residuals from tiny to large, with red indicating that the residuals are too large to converge. The horizontal coordinate represents the initial deviation rate, while the vertical coordinate is the maximum step size for just a single iteration.

As the initial deviation rate and maximum iteration step size increase, the algorithm eventually converges to a non-converging state, as seen in Figure 7. Case 1 indicates that the experimental deviation rate of the Broyden method is more than 0.2 and that it does not converge at all when the deviation rate is greater than 0.7. The convergence of the hybrid damped Newton method and the N-R method is similar, but the N-R method shows partial nonconvergence at a maximum step size of 0.1 in a single iteration.

This is due to the fact that the maximum number of iterations is exceeded. For the four operating conditions, the
The iterative convergence patterns of the three algorithms were similar, with convergence rates of 75.31%, 75.31%, 76.54%, and 90.12% for the hybrid damped Newton method; 67.90%, 70.37%, 72.84%, and 81.48% for the N-R method; and 49.38%, 49.38%, 59.26%, 55.56%, and 46.91% for the Broyden method, indicating that the convergence rates of the hybrid damped Newton method were the highest, and the convergence rates of the Broyden method were the worst. 59.26%, 55.56%, and 46.91% indicate that the mixed damped Newton method has the highest convergence rate, and the Broyden method has the lowest. All three iterative algorithms did not converge when the initial value deviation rate was at 0.9.

The preceding analysis demonstrates that the initial deviation rate and the maximum step size of a single iteration have a significant impact on the convergence of the iterative algorithm, and too large a parameter will lead to nonconvergence. Under the four typical operating conditions, the hybrid damped Newton method has the highest convergence rate, indicating strong convergence and good stability, but it cannot completely eliminate the dependence on the initial value of the iteration.

### 3.1.3. Dynamic Process Analysis of the Hybrid Damped Newton Method

The simulation verifies the variation of the number of component calculations and dynamic residuals during a typical acceleration process, as shown in Figures 8 and 9, to affirm the effectiveness of the three iterative algorithms in the dynamic calculation. Due to the slow change in the computational deviation of the iterative algorithm, the acceleration process is in a small deviation state and all three algorithms converge. The N-R method consistently has a higher number of component computations than the Broyden and hybrid damped Newton methods.
suggested a slower computational time, as seen in Figure 8. The computing speed and equilibrium times of the Broyden and hybrid damped Newton methods are comparable. Shown in Figure 9, the dynamic residuals of the N-R method fluctuate markedly, taking 13 s to reach equilibrium, whereas the dynamic residuals of the Broyden and hybrid damped Newton methods are similar, taking 8 s to reach equilibrium, indicating that both have a clear advantage in terms of dynamic computational speed.

The results show that the hybrid damped Newton method, which incorporates the concept of damping update and the integration of Broyden and traditional N-R algorithms, has ideal performance in dynamic process simulation while ensuring better convergence in large deviation states, providing a good theoretical basis and application value for engine modelling techniques.

3.2. Analysis of the NSDE Hybrid Damped Newton Method. The preceding study validates the benefits of the hybrid damped Newton method in terms of computing efficiency and stability, although the gradient algorithm is not totally free of the reliance of iterative beginning values. As a heuristic algorithm, the swarm intelligence algorithm may avoid the dependency on beginning values. To validate the effectiveness of the model calculation, the NSDE algorithm, which has high global convergence and computational efficiency, is integrated with the hybrid damped Newton method in this paper.

The population size of NSDE is 30, where ns1, ns2, nf1, and nf2 are dynamically changing data for each generation. Figure 10 depicts the fluctuation of the computational residuals of the NSDE algorithm, the hybrid damped Newton method, and the NSDE hybrid damped Newton method with the number of model iterations under Alt = 12.47 km, Ma = 2.0, and an initial deviation rate of 0.9. The Alt = 12.47 km, Ma = 2.0 condition was chosen because it provided the best convergence of the hybrid damped Newton method. The initial residuals of the hybrid damped Newton method are larger, while the residuals of the NSDE algorithm are lower in the first iteration due to the initial population update in the first step, indicating that the NSDE algorithm optimises better in the first iteration; the hybrid damped Newton method model terminates after 10 iterations when the calculation is scattered, whereas the NSDE algorithm can calculate until the model converges. The NSDE algorithm requires 81 model iterations for computational convergence, and the optimization efficiency at the late iterative stage is quite sluggish, failing to fulfill the real-time requirement. It is demonstrated that the NSDE damped Newton method can calculate convergence in the presence of a large deviation of the initial deviation rate of 0.1, while being free of initial value sensitivity and maintaining the ideal computational speed.

Figure 11 represents the variation pattern of the calculated residuals of the NSDE algorithm, the hybrid damped Newton method and the NSDE hybrid damped Newton method with the number of component calculations for Alt = 9.96 km, Ma = 1.5 working conditions and an initial deviation rate of 0.5. The hybrid damped Newton method was able to sustain convergence under typical large deviation conditions with 47 model iterations, but the NSDE algorithm required 97 model iterations, demonstrating that the NSDE algorithm is computationally sluggish. In addition, the number of NSDE component calculations is 2910. The NSDE hybrid damped Newton method requires 23 component calculations, which is fewer than the hybrid damped Newton method, indicating that this method has the quickest computational speed. The above analysis shows that in the large deviation state (initial deviation rate of 0.5), both the NSDE hybrid damped Newton method and the hybrid...
damped Newton method can ensure computational convergence, but the NSDE hybrid damped Newton method is faster. The NSDE hybrid damped Newton method triggers the NSDE algorithm only under conditions where the initial conditions are unknown or deviate significantly, more suitable for applications in steady-state models and large deviations from dynamic calculations to improve convergence.

In summary, the NSDE hybrid damped Newton method designed in this paper can get rid of the dependence on the initial value of the iterations, maintain the model convergence, and guarantee the computational speed under the condition that the initial deviation rate is 0.9. The simple NSDE method is not suitable for the solving of the coworking equation because every individual needs to call the model to calculate its fitness. Furthermore, under the large deviation condition (initial deviation rate of 0.5), the computational speed of the NSDE hybrid damped Newton method is faster than that of the hybrid damped Newton method, greatly improving the real-time performance, which is useful for the development of real-time aero-engine models.

3.3. Experimental Validation of Model Calculation Results Based on the NSDE Hybrid Damped Newton Method. The accuracy of the model must be validated in order to further evaluate the validity of the simulation results based on the NSDE hybrid damped Newton method. The numerical simulation results of the component-level model are compared to the steady-state reference data of the engine test at four typical operating points. Figure 12 represents the error analysis graph of the numerical calculation results of the model and the test reference data, comparing the parameters including engine thrust $F_n$, low pressure rotor speed $N_l$, high pressure rotor speed $N_h$, fan outlet pressure $P_3$, outlet temperature $T_3$, low-pressure turbine outlet pressure $P_6$, and temperature $T_6$. It can be seen that the maximum error between the numerical simulation results of the four operating conditions and the test reference value is 8.1%. The rest of the errors are controlled within 7.5%.

The model calculation errors are mainly caused by the accuracy of the component characteristics and the convergence residuals of the model equations. The overall model errors in this paper are maintained within a reasonable range, which indicates that the NSDE hybrid damped Newton method based model can maintain a good simulation accuracy because the gradient method is still used in the later iteration to convergence. Therefore, the NSDE hybrid damped Newton method can be applied to the simulation of aero-engine component-level models.

4. Conclusion

(1) The hybrid damped Newton method offers significant advantages in terms of iteration efficiency and computational speed; it is less sensitive to initial conditions and may keep strong computing efficiency and convergence, particularly in the high deviation state. Under the four typical operating conditions, the hybrid damped Newton method improves the comprehensive computational speed and iteration efficiency compared to the Broyden method and improves the comprehensive computational speed and iteration efficiency compared to the N-R method.

(2) The initial deviation rate and the maximum step size of a single iteration have a substantial influence on the iterative method’s convergence, and too large a parameter in both situations leads to nonconvergence. For the four typical operating conditions, the
hybrid damped Newton method has a substantially greater convergence rate than the N-R and Broyden methods, and it has the best convergence performance. The hybrid damped Newton method considerably improves convergence, but cannot avoid the dependence on beginning values.

(3) The NSDE hybrid damped Newton method is able to further get rid of the dependence, maintaining the convergence and guarantee the computational speed under the initial deviation rate of 0.9; under the large deviation condition, the computational speed of the NSDE hybrid damped Newton method is better than that of the hybrid damped Newton method, which greatly improves the real-time performance and provides engineering application value. The maximum error between the NSDE hybrid damping Newton method numerical simulation results and the experimental reference value is 8.1%, demonstrating positive simulation accuracy and engineering application value.

Data Availability

The data used to support the findings of this study are included within the article, which can available from the author [Chen Wang], upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by National Natural Science Foundation of China (NNSF) under Grant 61890921 and 61903059 and Aviation Science Foundation of China under Grant 2019ZB063001.

References

[1] L. Jaw and J. Mattingly, Aircraft Engine Controls, American Institute of Aeronautics and Astronautics Virgin, 2009.
[2] J.-h. Huang and G.-t. Feng, "Development and expectation of performance simulation in aviation engine," Journal of Propulsion Technology-Beijing, vol. 23, no. 4, pp. 346–356, 2002.
[3] L. Jiarui, S. Jianguo, and A. Zhang Shaoji, "Convergence algorithm of aeroengine performance mathematical model," Aeronautics, vol. 31, no. 4, pp. 48–50, 2005.
[4] R. Ashleman Jr., T. Lavelle, and F. Parsons, "The national cycle program-a flexible system modeling architecture," in 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, p. 3114, Cleveland, OH, USA, 1998.
[5] F. Ghiglizzeta, A. Traverso, M. Pascenti, and A. F. Massardo, "Micro gas turbine real-time modeling: test rig verification," Turbo Expo: Power for Land, Sea, and Air, vol. 48869, pp. 29–36, 2009.
[6] J. Kurzke, Calculation of Installation Effects within Performance Computer Programs, Motoren-Und Turbinen-Union Gmbh Munich, Germany, 1992.
[7] J. Biazar and B. Ghanbary, "A new technique for solving systems of nonlinear equations," Applied Mathematical Sciences, vol. 2, no. 55, pp. 2699–2703, 2008.
[8] M. A. Noor and M. Waseem, "Some iterative methods for solving a system of nonlinear equations," Computers & Mathematics with Applications, vol. 57, no. 1, pp. 101–106, 2009.
[9] A. Dhamacharoeen, "An efficient hybrid method for solving systems of nonlinear equations," Journal of Computational and Applied Mathematics, vol. 263, pp. 59–68, 2014.
[10] J. Wang and Y. Sui, "Study on convergence and improvement of iteration methods in aero-engine mathematical models," Journal of System Simulation, vol. 26, no. 2, pp. 310–314, 2014.
[11] Y. C. Chen, S. Y. Xu, Y. K. Yang, Q. Y. Tu, and X. S. Shang, "Research on the method to solve convergence problem in aero turbo-engine performance computation," Journal of Aerospace Power, vol. 12, 2008.
[12] X. Huang, Z. X. Wang, and X. B. Zhang, "Research of a corrected Broyden’s method on the aero-engine performance simulation," Science Technology Engineering, vol. 21, pp. 103–114, 2012.
[13] L. Jun, Y. Guo, and S. Zhang, "Research on the iteration methods in aero-engine non-linear model real-time computation," Journal of Aerospace Power, vol. 25, no. 3, pp. 681–686, 2010.
[14] Y. Wang, Q. Li, and X. Huang, "Numerical calculation of aeroengine model based on self-tuning Broyden quasi-newton method," Journal of Aerospace Power, vol. 31, no. 1, pp. 249–256, 2016.
[15] S.-L. Li and J.-G. Sun, "Application of genetic algorithm to solving nonlinear model of aeroengines," Chinese Journal of Aeronautics, vol. 16, no. 2, pp. 69–72, 2003.
[16] S. M. Su and X. C. Lian, "Application of genetic algorithm in aeroengine nonlinear mathematical models," Journal of Propulsion Technology-Beijing, vol. 25, pp. 237–240, 2004.
[17] S. M. Su and Y. Q. Chen, "Hybrid genetic algorithm in solving aeroengine nonlinear mathematical model," Journal of Propulsion Technology-Beijing, vol. 28, no. 6, p. 661, 2007.
[18] W. Yang, L. X. Feng, J. B. Peng, and H. T. Wang, "An intelligent algorithm for solution of nonlinear mathematical model for aeroengine," Journal of Propulsion Technology, vol. 5, 2008.
[19] H. Y. Qian, P. Y. Yang, and S. L. Xu, "Application of particle swarm optimization in obtaining solution of aeroengine component-level model," Journal of Propulsion Technology, vol. 6, 2012.
[20] Y. Bai, L.-M. Duan, L. Liu, F.-L. Zhou, and Y. Wang, "Solving variable cycle engine model based on improved hybrid particle swarm optimization," Taijin Jishu Journal of Propulsion Technology, vol. 35, no. 12, pp. 1694–1700, 2014.
[21] H. Xiao, H. Li, J. Li, S. Wang, and K. Peng, "Modeling method of variable cycle engine based on qpsd hybrid algorithm," Journal of Beijing University of Aeronautics and Astronautics, vol. 44, no. 2, p. 305, 2018.
[22] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.
[23] S. Ghosh, S. Das, A. V. Vasilakos, and K. Suresh, "On convergence of differential evolution over a class of continuous functions with unique global optimum," IEEE Transactions on Systems, Man, and Cybernetics, vol. 42, no. 1, pp. 107–124, 2011.
[24] R. Angira and B. Babu, “Non-dominated sorting differential evolution (NSDE): an extension of differential evolution for multi-objective optimization,” Indian International Conference on Artificial Intelligence, pp. 1428–1443, 2005.

[25] H. Chen, C. Cai, S. Jiang, and H. Zhang, “Numerical modeling on installed performance of turbofan engine with inlet ejector,” Aerospace Science and Technology, vol. 112, article 106590, 2021.

[26] F. Lu, H. Ju, and J. Huang, “An improved extended Kalman filter with inequality constraints for gas turbine engine health monitoring,” Aerospace Science and Technology, vol. 58, pp. 36–47, 2016.

[27] B. Pourbabae, N. Meskin, and K. Khorasani, “Sensor fault detection, isolation, and identification using multiple-model-based hybrid Kalman filter for gas turbine engines,” IEEE Transactions on Control Systems Technology, vol. 24, no. 4, pp. 1184–1200, 2015.

[28] F. Lu, T. Gao, J. Huang, and X. Qiu, “Nonlinear Kalman filters for aircraft engine gas path health estimation with measurement uncertainty,” Aerospace Science and Technology, vol. 76, pp. 126–140, 2018.

[29] F. Lu, T. Gao, J. Huang, and X. Qiu, ”A novel distributed extended Kalman filter for aircraft engine gas-path health estimation with sensor fusion uncertainty,” Aerospace Science and Technology, vol. 84, pp. 90–106, 2019.

[30] I. Loboda, Y. Feldshteyn, and V. Ponomaryov, ”Neural networks for gas turbine fault identification: multilayer perceptron or radial basis network?,” International Journal of Turbo & Jet Engines, vol. 29, no. 1, 2012.

[31] L.-H. Ren, Z.-F. Ye, and Y.-P. Zhao, ”A modeling method for aero-engine by combining stochastic gradient descent with support vector regression,” Aerospace Science and Technology, vol. 99, article 105775, 2020.

[32] F. Lu, J. Wu, J. Huang, and X. Qiu, ”Aircraft engine degradation prognostics based on logistic regression and novel os-elm algorithm,” Aerospace Science and Technology, vol. 84, pp. 661–671, 2019.

[33] Y.-P. Zhao, G. Huang, Q.-K. Hu, J.-F. Tan, J.-J. Wang, and Z. Yang, ”Soft extreme learning machine for fault detection of aircraft engine,” Aerospace Science and Technology, vol. 91, pp. 70–81, 2019.

[34] M. Xu, J. Wang, J. Liu et al., ”An improved hybrid modeling method based on extreme learning machine for gas turbine engine,” Aerospace Science and Technology, vol. 107, article 106333, 2020.