

Research Article

An Accurate Method for Real-Time Aircraft Dynamics Simulation Based on Predictor-Corrector Scheme

Jiaxin Zhao,^{1,2} Hongwei Wang,³ and Heming Zhang¹

¹National CIMS Engineering Research Centre, Tsinghua University, Beijing 100084, China

²State Key Laboratory of Civil Aircraft Flight Simulation, Shanghai Aircraft Design and Research Institute, Shanghai 201210, China

³School of Engineering, University of Portsmouth, Portsmouth PO1 3DJ, UK

Correspondence should be addressed to Heming Zhang; hzmz@mail.tsinghua.edu.cn

Received 4 June 2014; Revised 15 October 2014; Accepted 28 October 2014

Academic Editor: Kang Li

Copyright © 2015 Jiaxin Zhao 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.

Real-time aircraft dynamics simulation requires very high accuracy and stability in the numerical integration process. Nonetheless, traditional multistep numerical methods cannot effectively meet the new requirements. Therefore, a novel real-time multistep method based on Predict-Evaluate-Correct scheme of three-step fourth-order method (RTPEC-34) is proposed and developed in this research to address the gap. In addition to the development of a highly accurate algorithm based on predictor-corrector, the contribution of this work also includes the analysis of truncation error for real-time problems. Moreover, the parameters for the RTPEC-34 method are optimized using intelligent optimization algorithms. The application and comparison of the optimization algorithms also lead to general guidelines for their applications in the development of improved multistep methods. Last but not least, theoretical analysis is also conducted on the stability of the proposed RTPEC-34 method, which is corroborated in simulation experiments and thus provides general guidelines for the evaluation of real-time numerical methods. The RTPEC-34 method is compared with other multistep algorithms using both numerical experiments and a real engineering example. As shown in the comparison, it achieves improved performance in terms of accuracy and stability and it is also a viable and efficient algorithm for real-time aircraft dynamics simulation.

1. Introduction

Aircraft dynamics simulation is a complex nonlinear process whereby engine, aerodynamic, and atmospheric models are solved simultaneously [1]. Aircraft variables such as aircraft velocity, orientation angles, aerodynamic angles, and angular rates are assembled by ordinary differential equations (ODEs) with initial value problems (IVP). The real-time simulation of aircraft dynamics requires very accurate and stable numerical methods for solving the ODEs.

A numerical solution of an ODE is a table of approximated values of the variables for a set of discrete time points. Multistep methods, as a class of the most widely used numerical solution, use information at more than one previous point to estimate the solution at the next point. Linear multistep methods have the form shown in (1) where

the parameters α_i and β_i are determined using polynomial interpolation [2]

$$y_{n+1} = \sum_{i=1}^k \alpha_i y_{n+1-i} + h \sum_{i=0}^k \beta_i f(x_{n+1-i}, y_{n+1-i}). \quad (1)$$

If $\beta_0 = 0$, the method is explicit; and if $\beta_0 \neq 0$, then the method is implicit. Since multistep methods require several values of variables and derivatives from previous calculations history, they cannot self-start before the values have been obtained. Using a single-step method, which does not require past history, is a common strategy to generate solution values at enough points for starting using a multistep method of a desired order. The Euler method and the Runge-Kutta (RK) method (or variants of them) are popular single-step methods to assist multistep methods in providing initial values. We know that implicit single-step methods are generally more

accurate and stable than the explicit ones. Similarly, implicit multistep methods are usually more accurate and stable than the explicit ones but a nonlinear equation needs to be solved to obtain the derivative values y_{n+1} . On the other hand, explicit linear multistep methods can conveniently make a guess of initial values and have simple and direct formulas. These advantages help meet the requirements of real-time input and calculation. As a result, the explicit and implicit methods are often used together as a predictor-corrector pair and the Predict-Evaluate-Correct (PEC) method has become a very useful scheme for solving ODEs with IVP. When computational complexity and the error constants relative to order are concerned [3], the most popular fourth-order Adams-Bashforth-Moulton (ABM) [2] of predictor-corrector method is widely used. This method, however, does not satisfy the needs of real-time simulation [4]. In addition, although implicit methods have a much greater region of stability than explicit methods, they are still not necessarily unconditionally stable. Explicit methods cannot be A-stable due to the famous Dahlquist barrier, which means it is less useful for solving stiff ODEs [5] and a PEC scheme is actually explicit in a complicated way. On the contrary, a properly designed implicit multistep method is suitable for dealing with stiff problems. Since the accuracy restriction depends on the slowly varying components of the solution and the stability restriction depends on the rapidly varying ones, the Gear method [6] and the Rosenbrock method [7] aim to solve stiff ODEs with long intervals of stability regions, at the cost of accuracy.

According to the different requirements and constraints of a simulation, many researchers improved the multistep methods from different aspects. Huang [8] considered two-stage hybrid methods of seventh order. Gottlieb et al. [9] considered a class of two-step and two-stage methods and Xu and Zhao [3] proposed an estimation of the longest stability interval for three-order four-step methods. In addition, Bulatov and Berghe [10] discussed two-step fourth-order methods of the second order and Bresten et al. [11] focused on strong stability preserving methods. Seong et al. [12] considered the fifth-order multistep method using constant step size and Wang et al. [13] discussed the variable-step interaction algorithm for multidisciplinary collaborative simulation. Although these methods attempted to improve the method by varying the order and number of integration steps, very few of them focused on optimizing parameters for a fixed order. When stability and accuracy of numerical solutions are considered at the same time, a predictor-corrector pair of order four is a very good choice based on which improved methods can be developed.

This research, with a particular focus on the aircraft dynamics ODE problems, aims to improve the accuracy of the real-time predictor-corrector methods and analyze the stability regions of the proposed methods. Both simulation experiments of numerical examples and an engineering example are used to evaluate the performance of the proposed method by comparing its results with those obtained from the classic RK method and the methods for stiff ODEs. It has been shown in the evaluation that these proposed methods are effective. The rest of this paper is organized as follows.

Section 2 details the development of a predictor-corrector method for highly accurate real-time aircraft dynamics simulation as well as the optimization of its key parameters. In Section 3, a theoretical analysis is conducted to evaluate the stability of the RTPEC-34 method. In Section 4, the RTPEC-34 method is evaluated by comparing it with other classic multistep algorithms in several simulation experiments. The main conclusions of this research are drawn in Section 5.

2. Optimization of the Accuracy of Multistep Numerical Methods Using Predictor-Corrector Scheme

Aircraft simulation involves a computation process that is subject to real-time constraints as it is a real-time system. This means the responses or results of the simulation process have a deadline that must be met, regardless of system load, in order for the system considered to be correct [14, 15]. The predictor-corrector multistep formula is rearranged as the real-time form discussed in [16]. In particular, the predictor equation is obtained as follows:

$$y_{n+1}^p = y_n + h \left[\bar{\beta}_0 f(x(t_n), y_n) + \bar{\beta}_1 f(x(t_{n-1}), y_{n-1}) + \bar{\beta}_2 f(x(t_{n-2}), y_{n-2}) + \bar{\beta}_3 f(x(t_{n-3}), y_{n-3}) \right]. \quad (2a)$$

Evaluate $f(x(t_{n+1}), y_{n+1}^p)$ at this predicted value of y_{n+1}^p to obtain y_{n+1} at t_{n+1} . The corrector equation is then obtained as

$$y_{n+1} = y_n + h \left[\beta_v f(x(t_{n+1}), y_{n+1}^p) + \beta_0 f(x(t_n), y_n) + \beta_1 f(x(t_{n-1}), y_{n-1}) + \beta_2 f(x(t_{n-2}), y_{n-2}) \right]. \quad (2b)$$

In order to improve the accuracy of the real-time predictor-corrector multistep methods, the step of the predictor equation needs to be reduced. Consider the fourth-order three-step algorithm as follows:

$$y_{n+v} = y_n + h (\bar{\beta}_0 f_n + \bar{\beta}_1 f_{n-1} + \bar{\beta}_2 f_{n-2}) \quad (3)$$

$$y_{n+1} = y_n + h (\beta_v f_{n+v} + \beta_0 f_n + \beta_1 f_{n-1} + \beta_2 f_{n-2}),$$

where parameters $\bar{\beta}_0$, $\bar{\beta}_1$, $\bar{\beta}_2$, β_v , β_0 , β_1 , and β_2 are yet to be determined. Numerical convergence can be achieved for formula (3) when the following constraint equations are established [17]:

$$\bar{\beta}_0 + \bar{\beta}_1 + \bar{\beta}_2 = v, \quad (4a)$$

$$\beta_v + \beta_0 + \beta_1 + \beta_2 = 1.$$

Additionally, based on the method of undetermined coefficients [18], formula (3) can be further constrained by five equations as follows:

$$\begin{aligned} -\bar{\beta}_1 - 2\bar{\beta}_2 &= \frac{v^2}{2}, \\ \frac{1}{2}\bar{\beta}_1 + 2\bar{\beta}_2 &= \frac{v^3}{6}, \\ v\beta_v - \beta_1 - 2\beta_2 &= \frac{1}{2}, \\ \frac{v^2}{2}\beta_v + \frac{1}{2}\beta_1 + 2\beta_2 &= \frac{1}{6}, \\ \frac{v^3}{2}\beta_v - \frac{1}{6}\beta_1 - \frac{4}{3}\beta_2 &= \frac{1}{24}. \end{aligned} \quad (4b)$$

Thus, (4a) and (4b) include seven undetermined equations which contain eight unknown variables. The multistep numerical methods for solving an ODE suffer from two distinct sources of error. Compared to truncation error, round-off error generally plays only a minor role [19, 20]. Therefore, truncation error is the main focus of this research and the estimation of the p-order truncation error can be done using the following formula:

$$\begin{aligned} \sum_{i=0}^k \alpha_i y(t_{n+k} - ih) - h \sum_{i=0}^k \dot{y}(t_{n+k} - ih) \\ = h^{p+1} C_{p+1} y^{(p+1)}(t_{nk}) + \dots \end{aligned} \quad (5)$$

The local truncation error coefficient is then represented as

$$\begin{aligned} C_{p+1} &= (-1)^{p+1} \frac{1}{(p+1)!} (\alpha_1 + 2^{p+1}\alpha_2 + \dots + k^{p+1}\alpha_k) \\ &+ \frac{1}{p} (\beta_1 + 2^p\beta_2 + \dots + k^p\beta_k). \end{aligned} \quad (6)$$

For (3), the estimation of truncation error coefficient can be represented as

$$C_{p-34} = (-1)^5 \frac{2}{5!} + \frac{1}{4!} [v^4\beta_v + \bar{\beta}_1 + 2^4(\bar{\beta}_2 + \beta_2)]. \quad (7)$$

Based on the above discussion, the problem has now become how to figure out the minimum value of formula (7) with the constraints described in (4a) and (4b).

There are many optimization methods to find out the minimum value with nonlinear constraints, such as feasible direction method (FDM) [21], penalty function method, and quadratic programming (QP) [22]. However, these methods are not accurate enough to solve multivalued equality. The genetic algorithm (GA) [23] has the potential to fix the problem of falling into local optimum compared with the classic methods mentioned above. While the equality nonlinear constraints in the above model are strict, the infeasible solution will become a large proportion in the populations as the genetic algorithm is used separately [24].

Hence, some other optimization methods should be utilized to assist the GA [25]. The penalty function has the advantages of considering the points out of the feasible regions when solving nonlinear constraints problems [26]. In this research, the GA with a punishment strategy is chosen to find solutions outside the feasible regions.

A penalty function is generally constructed using the addition form below:

$$val(x) = f(x) + \sum_{i=1}^n r_i p_i(x), \quad (8)$$

where x is the chromosome, $f(x)$ is the objective function, $p_i(x)$ is i th penalty element, and r_i is the changeable penalty coefficient of the $p_i(x)$. The element $\sum_{i=1}^n r_i p_i(x)$ in formula (8) is for multiplying constraint violations, which will be used to alter feasible region unless the original function is unsolvable. The complete expression of penalty function then can be established as follows:

$$val(x) = \begin{cases} f(x), & x \text{ workable,} \\ f(x) + \sum_{i=1}^n r_i p_i(x), & \text{other.} \end{cases} \quad (9)$$

To simplify the constraint, use v to represent $\bar{\beta}_0$, $\bar{\beta}_1$, and $\bar{\beta}_2$:

$$\begin{aligned} \bar{\beta}_0 &= \frac{1}{6}v^3 + \frac{3}{4}v^2 + v, \\ \bar{\beta}_1 &= -\frac{v^3}{3} - v^2, \\ \bar{\beta}_2 &= \frac{1}{6}v^3 + \frac{1}{4}v^2. \end{aligned} \quad (10)$$

Thus, the objective function (7) can be rearranged as

$$f(x) = C_{p-34} = v^4\beta_v + \frac{7}{3}v^3 + 3v^2 + \beta_1 + 16\beta_2. \quad (11)$$

Then, the following three penalty elements are used for the GA optimization process, which are derived from (4a), (4b), and (10):

$$p_1 = \frac{v^2}{2}\beta_v + \frac{1}{2}\beta_1 + 2\beta_2 - \frac{1}{6} = 0, \quad (12a)$$

$$p_2 = \frac{v^3}{2}\beta_v - \frac{1}{6}\beta_1 - \frac{4}{3}\beta_2 - \frac{1}{24} = 0, \quad (12b)$$

$$p_3 = v\beta_v - \beta_1 - 2\beta_2 - \frac{1}{2} = 0. \quad (12c)$$

The setting for the GA running is as follows: binary code is used; the number of individuals in the population is 1000; the crossover operator is 0.9; and mutation operator is 0.08. A simulation is done using MATLAB software, and then the parameters can be obtained for the minimum truncation error of the RTPEC-34 method.

Compared with the classic optimization methods, it can be found that different parameters are obtained as shown in Table 1.

Apply the parameters to (3); a family of the RTPEC-34 method with the highest accuracy can be obtained. The RTPEC-34 method using FDM optimization is shown as follows:

$$\begin{aligned} y_{n+v} &= y_n + h(0.5239f_n - 0.1775f_{n-1} + 0.0495f_{n-2}), \\ y_{n+1} &= y_n + h(1.5534f_{n+v} - 0.6811f_n \\ &\quad + 0.1403f_{n-1} - 0.0126f_{n-2}). \end{aligned} \quad (13a)$$

The RTPEC-34 method using QP optimization is shown as follows:

$$\begin{aligned} y_{n+v} &= y_n + h(0.7947f_n - 0.3510f_{n-1} + 0.1012f_{n-2}), \\ y_{n+1} &= y_n + h(0.9122f_{n+v} + 0.1233f_n \\ &\quad - 0.0681f_{n-1} + 0.0326f_{n-2}). \end{aligned} \quad (13b)$$

The RTPEC-34 method using GA optimization with a punishment strategy is shown as follows:

$$\begin{aligned} y_{n+v} &= y_n + h(0.7425f_n - 0.3147f_{n-1} + 0.0903f_{n-2}), \\ y_{n+1} &= y_n + h(0.9963f_{n+v} + 0.0125f_n \\ &\quad - 0.0337f_{n-1} + 0.0249f_{n-2}). \end{aligned} \quad (13c)$$

3. Stability Analysis

A widely used approach to determining stability region is to apply the method to the linear ODE $dy/dx = \lambda y$ with initial condition $y(0) = y_0$, and its analytical solution is given as $y(x) = y_0 e^{\lambda x}$. Applying the three-step fourth-order ABM algorithms (12a), (12b), and (12c) to this equation using a fixed step-size h , we have

$$\begin{aligned} y_{n+v} &= y_n + h(\bar{\beta}_0 \lambda y_n + \bar{\beta}_1 \lambda y_{n-1} + \bar{\beta}_2 \lambda y_{n-2}), \\ y_{n+1} &= y_n + h(\beta_v \lambda y_{n+v} + \beta_0 \lambda y_n + \beta_1 \lambda y_{n-1} + \beta_2 \lambda y_{n-2}). \end{aligned} \quad (14)$$

Substituting y_{n+v} into y_{n+1} , we can obtain the following stability polynomial:

$$\begin{aligned} \pi(\xi, \mu) &= \xi^3 - (\beta_v \bar{\beta}_0 \mu^2 + (\beta_v + \beta_0) \mu + 1) \xi^2 \\ &\quad - (\beta_v \bar{\beta}_1 \mu^2 + \beta_1 \mu) \xi - \beta_v \bar{\beta}_2 \mu^2 - \beta_2 \mu, \end{aligned} \quad (15)$$

where $\mu = h\lambda$.

Obtaining RTPEC-34 stability polynomial with the parameters shown in Table 1, we can then use root-locus method to obtain the diagram for the stability regions of this method as shown Figure 1.

The curves C_1 , C_2 , and C_3 are the root locus of RTPEC-34 (13a), (13b), and (13c) stability polynomial, respectively. Since no multistep methods of greater than second order are unconditionally stable, an explicit linear multistep method

TABLE 1: Parameter values obtained using different optimization methods.

Parameter	FDM	QP	GA with a punishment strategy
ν	0.3959	0.5450	0.5181
β_ν	1.5534	0.9122	0.9963
β_0	-0.6811	0.1233	0.0125
β_1	0.1403	-0.0681	-0.0337
β_2	-0.0126	0.0326	0.0249
$\bar{\beta}_0$	0.5239	0.7947	0.7425
$\bar{\beta}_1$	-0.1775	-0.3510	-0.3147
$\bar{\beta}_2$	0.0495	0.1012	0.0903

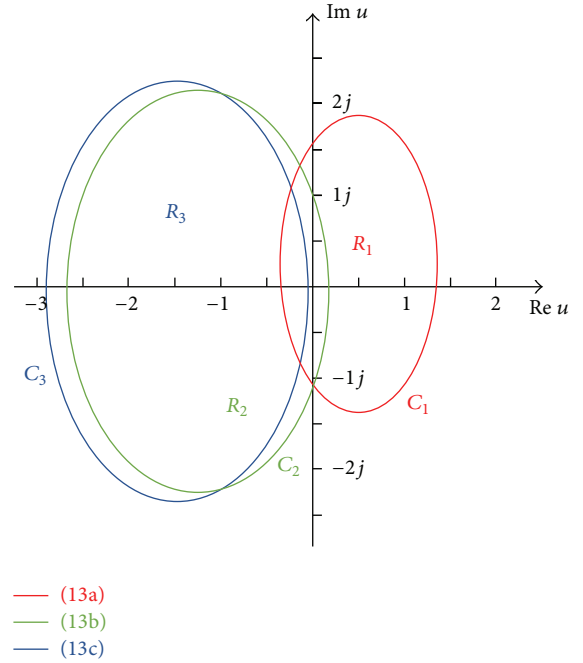


FIGURE 1: The region of absolute stability of the RTPEC-34 method.

cannot be A-stable if it is higher than order two due to the famous Dahlquist barrier theorem. Even if it is implicit, it is still the case. Thus the fourth-order method can be stable only in the finite stability region. From Figure 1, it can be concluded that the highest accuracy RTPEC-34 method (13c) using GA with a punishment strategy has the largest stability region among the three methods.

4. Evaluation and Discussion

4.1. Numerical Experiments. In this section, the numerical results obtained by using the highest accuracy RTPEC-34 method are presented in Example 1. In addition, the method proposed is compared with the Runge-Kutta method of fourth-order (RK4) method to demonstrate its good performance in supporting real-time simulation in Example 2. Furthermore, the Rosenbrock method for stiff ODEs is used to evaluate the accuracy of the proposed method.

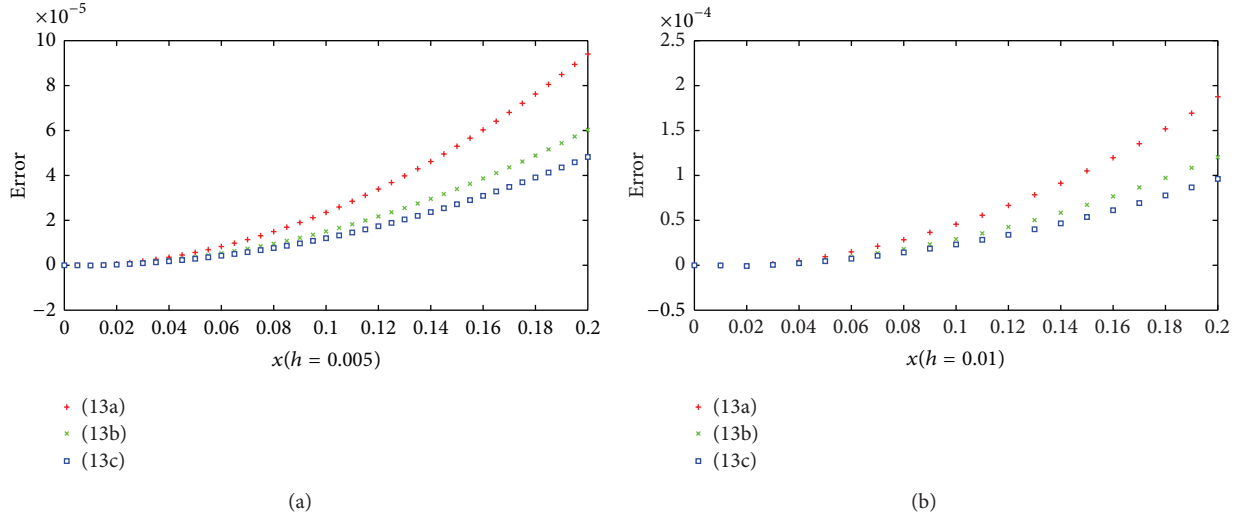


FIGURE 2: Comparison of numerical error for (13a), (13b), and (13c).

Example 1. The ODE with IVP (16) is used to compare the proposed methods (13a), (13b), and (13c) using different parameters. Consider

$$\begin{aligned} \frac{dy}{dx} &= 1 + \cos(x), \\ y(0) &= 0. \end{aligned} \quad (16)$$

The analytical solution of this ODE is $y(x) = \sqrt{1 + 2x}$. Applying (13a), (13b), and (13c) to the ODE (16), the numerical calculation errors in different simulation settings with different step-size h are shown in Table 2. According to Figure 2, the accuracy of (13c) is better than that of (13a) and (13b) under the condition of $h = 0.005$, $h = 0.01$, and $h = 0.02$. The RTPEC-34 methods using GA optimization with a punishment strategy (13c) are much more stable than the other two cases. Thus, we choose (13c) as the method with the highest accuracy after the optimization and comparison. It is used to demonstrate the efficiency of the proposed methods in other simulation experiments.

Example 2. To compare the proposed method (13c) with the RK4 method, the real-time simulation procedure of an aircraft propulsion system [27] is considered in which the motion ODE with IVP has the following form:

$$\begin{aligned} \frac{dy}{dx} &= -10y^2 + 1 + \sin(2\pi x), \quad x \in [0, 1], \\ y(0) &= 0. \end{aligned} \quad (17)$$

Applying (13c) and the RK4 to system (17), numerical solutions can be obtained, as shown in Figure 3. It is shown in the curves in Figure 3 that the solutions obtained using the RTPEC-34 method come earlier in time (for about one) than the RK4 method, which is a significant phenomenon in the real-time simulation. This means the RTPEC-34 method (13c) is much more adaptive than RK4 when the aircraft

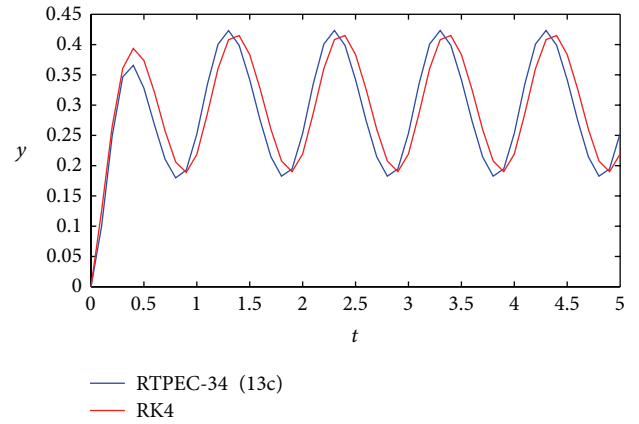


FIGURE 3: The numerical solutions obtained using the RTPEC-34 method and the RK4 method.

dynamics simulation requires the numerical solutions in real time.

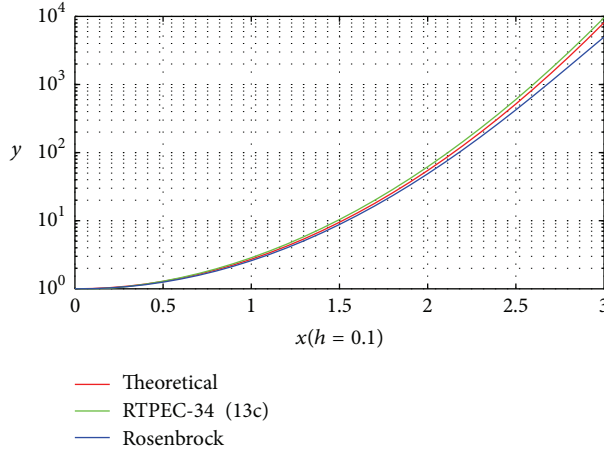
Example 3. Due to the quality of being A-stable, the Rosenbrock method is a half-implicit RK method used for solving stiff ODEs. The Rosenbrock method of order three has the following form:

$$\begin{aligned} y_{n+1} &= y_n + w_1 k_1 + w_2 k_2, \\ k_1 &= h \left[1 - ha_1 f_y(x_n, y_n) \right]^{-1} f(x_n, y_n), \\ k_2 &= h \left[1 - ha_2 f_y(x_n c_1 h, y_n + c_2 k_1) \right]^{-1} \\ &\quad \times f(x_n + d_1 h, y_n + d_2 k_1), \end{aligned} \quad (18)$$

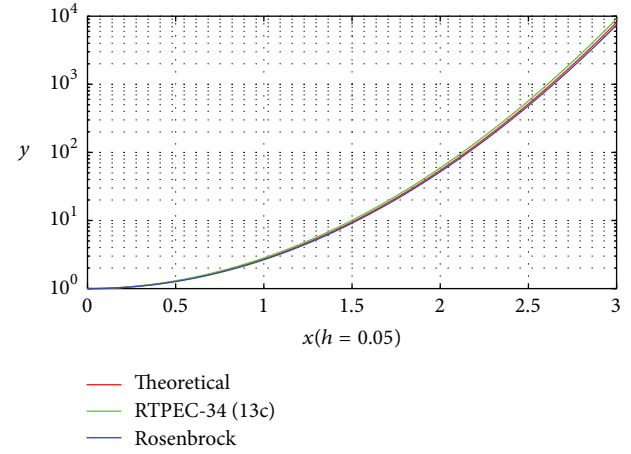
where $a_1 = 1.40824829$, $a_2 = 0.59175171$, $c_1 = c_2 = 0.17378667$, $w_1 = -0.41315432$, $d_1 = d_2 = 0.17378667$, and $w_2 = 0.41315432$. However, it is difficult to estimate the truncation error directly. Hence, the following ODE (19)

TABLE 2: Numerical errors due to different values of the parameters in (13a), (13b), and (13c).

t	Formula (13a) (FDM)			Formula (13b) (QR)			Formula (13c) (GA with a punishment strategy)		
	$h = 0.005$	$h = 0.01$	$h = 0.02$	$h = 0.005$	$h = 0.01$	$h = 0.02$	$h = 0.005$	$h = 0.01$	$h = 0.02$
0.0000	$0e-4$	$0e-3$	$0e-3$	$0e-4$	$0e-3$	$0e-3$	$0e-4$	$0e-4$	$0e-3$
0.0200	$0.0065e-4$	$-0.0008e-3$	$-0.0013e-3$	$0.0038e-4$	$-0.0008e-3$	$-0.0013e-3$	$0.0029e-4$	$-0.0083e-4$	$-0.0013e-3$
0.0400	$0.0355e-4$	$0.0052e-3$	$-0.0067e-3$	$0.0226e-4$	$0.0031e-3$	$-0.0067e-3$	$0.0179e-4$	$0.0230e-4$	$-0.0067e-3$
0.0600	$0.0834e-4$	$0.0149e-3$	$0.0136e-3$	$0.0533e-4$	$0.0094e-3$	$0.0065e-3$	$0.0425e-4$	$0.0734e-4$	$0.0039e-3$
0.0800	$0.1499e-4$	$0.0284e-3$	$0.0413e-3$	$0.0960e-4$	$0.0181e-3$	$0.0245e-3$	$0.0767e-4$	$0.1431e-4$	$0.0184e-3$
0.1000	$0.2352e-4$	$0.0457e-3$	$0.0765e-3$	$0.1507e-4$	$0.0291e-3$	$0.0473e-3$	$0.1205e-4$	$0.2319e-4$	$0.0366e-3$
0.1200	$0.3391e-4$	$0.0666e-3$	$0.1192e-3$	$0.2173e-4$	$0.0426e-3$	$0.0748e-3$	$0.1738e-4$	$0.3398e-4$	$0.0587e-3$
0.1400	$0.4617e-4$	$0.0913e-3$	$0.1693e-3$	$0.2958e-4$	$0.0585e-3$	$0.1071e-3$	$0.2367e-4$	$0.4667e-4$	$0.0846e-3$
0.1600	$0.6028e-4$	$0.1197e-3$	$0.2269e-3$	$0.3862e-4$	$0.0767e-3$	$0.1442e-3$	$0.3090e-4$	$0.6127e-4$	$0.1142e-3$
0.1800	$0.7625e-4$	$0.1519e-3$	$0.2919e-3$	$0.4885e-4$	$0.0973e-3$	$0.1860e-3$	$0.3909e-4$	$0.7776e-4$	$0.1477e-3$
0.2000	$0.9406e-4$	$0.1877e-3$	$0.3642e-3$	$0.6025e-4$	$0.1202e-3$	$0.2325e-3$	$0.4822e-4$	$0.9614e-4$	$0.1850e-3$



(a)



(b)

FIGURE 4: Comparison of the numerical solutions obtained by the RTPEC-34 and Rosenbrock methods.

is used to compare the RTPEC-34 method (13c) with the Rosenbrock method in terms of accuracy:

$$\begin{aligned} \frac{dy}{dx} &= 2xy, \quad x \in [0, 1], \\ y(0) &= 1. \end{aligned} \quad (19)$$

The analytical solution of ODE (18) is $y(x) = e^{x^2}$.

Table 3 shows the exact solutions and numerical solutions using the RTPEC-34 method (13c) and the Rosenbrock method in the interval $[0, 1]$. It is noted that the accuracy of RTPEC-34 method (13c) is much better than that of the Rosenbrock method at $h = 0.1$ in Figure 4. It is indicated in this phenomenon that ODEs can be solved more accurately by the proposed method unless they are extremely stiff.

4.2. Aircraft Dynamics Experiment. The aircraft dynamics experiment is focused on the simulation of the motion model, which is a system of 12 scalar order differential equations [1]. To avoid the singularities of derivatives of roll

angular rate \dot{p} and yaw angular rate \dot{r} when pitch angle θ passes through $\pm(\pi/2)$, quaternion methods are used for the aircraft orientation representation [28]. We have got an aircraft system representation consisting of 13 scalar first order differential equations as follows:

$$\begin{aligned} \dot{u} &= ru - qw + \frac{1}{m} (\bar{X} + F_T) + 2(q_1q_3 - q_0q_2)g, \\ \dot{v} &= pw - ru + \frac{1}{m} \bar{Y} + 2(q_2q_3 + q_0q_2)g, \end{aligned} \quad (20a)$$

$$\begin{aligned} \dot{w} &= qu - pv + \frac{1}{m} \bar{Z} + (q_0^2 - q_1^2 - q_2^2 + q_3^2)g, \\ \dot{p} &= (c_1r + c_2p)q + c_3\bar{L} + c_4(\bar{N} + h_Eq), \\ \dot{q} &= c_5pr - c_6(p^2 - r^2) + c_7(\bar{M} - h_Er), \\ \dot{r} &= (c_8p - c_2r)q + c_4\bar{L} + c_9(\bar{N} + h_Eq), \end{aligned} \quad (20b)$$

TABLE 3: Comparison of numerical solutions obtained with different step sizes.

x	Theoretical value	Rosenbrock method		RTPEC-34 method (13c)	
		$h = 0.05$	$h = 0.1$	$h = 0.05$	$h = 0.1$
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.1	1.0101	1.0075	1.0049	1.0050	1.0100
0.2	1.0408	1.0355	1.0303	1.0405	1.0300
0.3	1.0942	1.0859	1.0777	1.0991	1.0822
0.4	1.1735	1.1617	1.1501	1.1843	1.1714
0.5	1.2840	1.2679	1.2522	1.3019	1.2934
0.6	1.4333	1.4118	1.3910	1.4601	1.4567
0.7	1.6323	1.6038	1.5765	1.6705	1.6737
0.8	1.8965	1.8588	1.8228	1.9498	1.9616
0.9	2.2479	2.1978	2.1501	2.3217	2.3451
1.0	2.7183	2.6511	2.5875	2.8203	2.8599

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad (20c)$$

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{z}_E \end{bmatrix} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad (20d)$$

where

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \pm \begin{bmatrix} \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} - \sin \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \end{bmatrix}. \quad (21)$$

$$u = V_T \cos \alpha \cos \beta,$$

$$v = V_T \sin \beta,$$

$$w = V_T \sin \alpha \cos \beta,$$

The 12 aircraft variables, $[V_T, \alpha, \beta]$, $[p, q, r]$, $[x_E, y_E, z_E]$, and $[\phi, \theta, \psi]$, represent velocity, sideslip angle, attack angle, body-axis rolling rate, pitch rate and yaw rate, x -position, y -position, and z -position with respect to earth, rolling angle, pitch angle, and yaw angle, respectively. For the sake of calculation simplicity, 7 parameters are introduced, where $[u, v, w]$ are velocity in x -axis, y -axis, and z -axis direction and $[q_0, q_1, q_2, q_3]$ are quaternion components which do not have any physical meaning.

To demonstrate the effectiveness and efficiency of the proposed method (13c), it has been applied to a real aircraft dynamics simulation problem. A number of comparisons with other classic linear multistep methods such as the fourth-order Adams-Bashforth method (AB4) have been made in the simulation which has high real-time requirements

$$y_{n+1} = y_n + \frac{1}{24} (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}). \quad (22)$$

The nonlinear dynamics model of the F-16 aircraft [29] is used in the experiment as the simulation in such an engineering problem requires a detailed model and practical data. For the atmospheric data relative to the coefficient of formulas (20a), (20b), (20c), and (20d) an approximation is used based on the guidelines by the International Standard Atmosphere (ISA) [30]. In the experiment, the flight status of the F-16 aircraft with the altitude at 1000 m and the velocity at 200 m/s and the initial values of variables are calculated by using the ISA atmospheric model, aerodynamics model, and engine model. The simulation is performed for the F-16 aircraft dynamical model using different numerical methods, the RTPEC-34 (13c) and the AB4 method. The configuration for the simulation is shown in Table 4.

As demonstrated in these simulation experiments, the aircraft variables curves in Figure 5 can illustrate the real state of a whole fight, where the roll rate, pitch rate, and yaw rate $[p, q, r]$ tend to be stable in Figures 5(a)~5(c). This is also the case for $[\alpha, \beta]$ in Figure 5(d) through to Figure 5(f) and the coordinates $[x_E, y_E, z_E]$ of the fight position in the

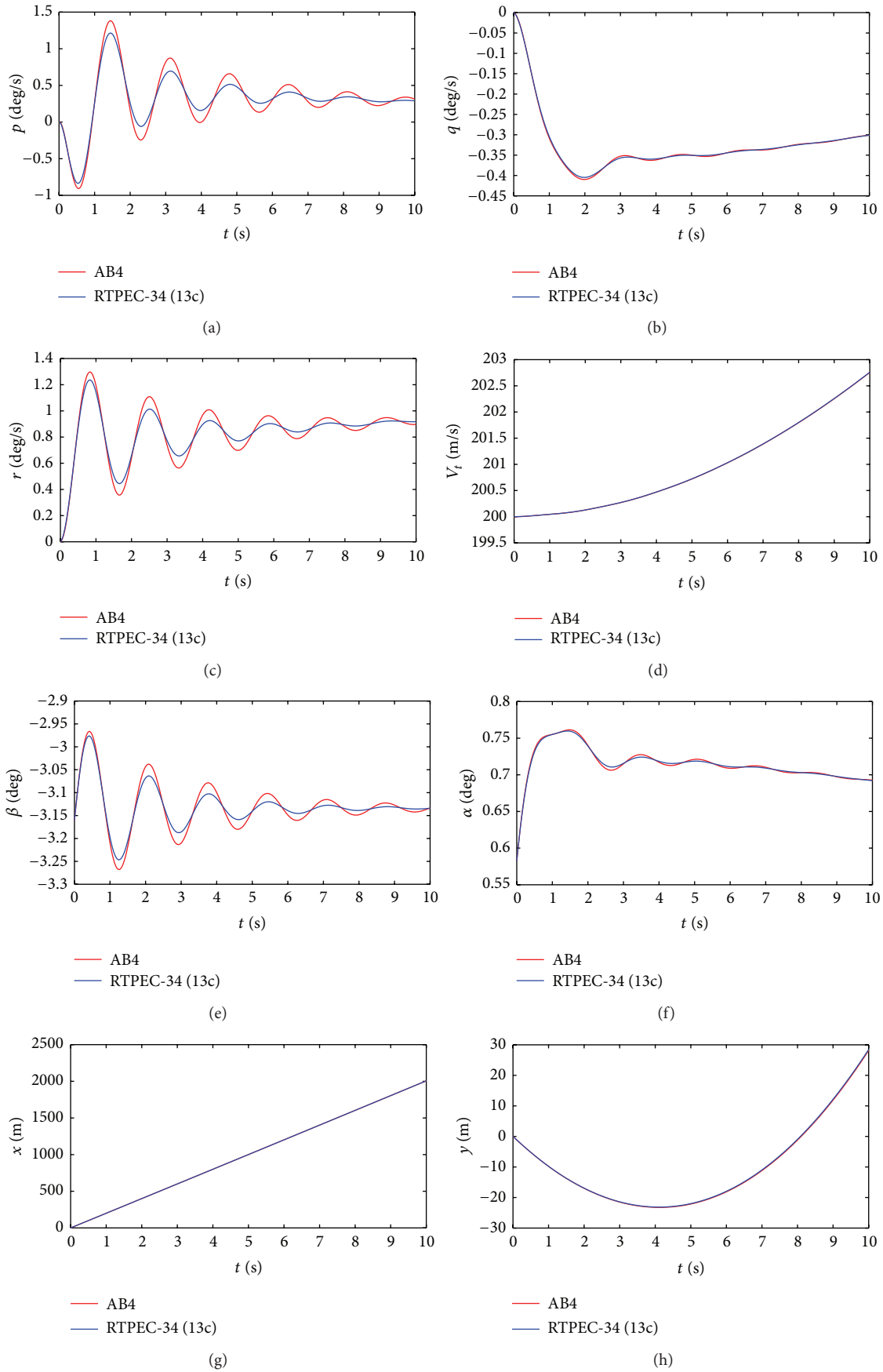


FIGURE 5: Continued.

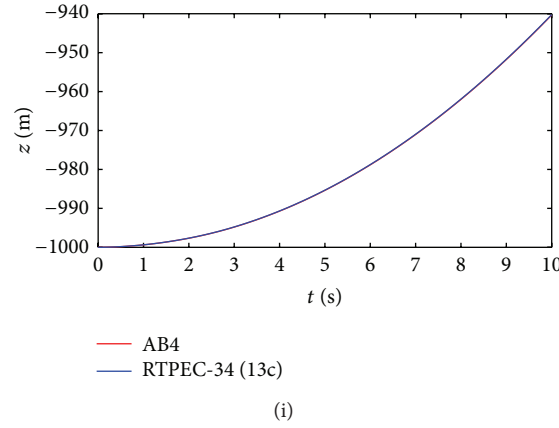


FIGURE 5: Simulation results of different numerical methods for the F-16 aircraft model.

TABLE 4: Simulation configuration for the F-16 aircraft dynamics model.

Initial values of variables	$[V_T, \alpha, \beta] = [200, -0.0552, 0.0102]$
	$[q_0, q_1, q_2, q_3] = [1, 0, 0.0051, 0]$
	$[p, q, r] = [0, 0, 0]$
	$[x_E, y_E, z_E] = [0, 0, -1000]$
Simulation period	0~10 seconds
Step size	0.02 seconds

whole air route in Figure 5(g) through to Figure 5(i). Both of the numerical methods are effective in the simulation of the F-16 aircraft at the altitude of 1000 m; however the curve of RTPEC-34 demonstrates smaller fluctuations than that of AB4 in Figures 5(a) and 5(b), which means that the convergence speed of RTPEC-34 (13c) is faster than that of AB4. With the number of iteration increased, the accuracy of RTPEC-34 will be higher than AB4.

The example of the F-16 aircraft simulation demonstrates that the proposed method is adaptive in the field of dynamic ODEs. The comparison with the classic multistep methods of the same order shows the good accuracy and speed of convergence of the RTPEC-34 (13c) and confirms that the proposed real-time fourth-order three-step predictor-corrector method achieves the highest accuracy.

5. Conclusions

In this paper, a novel real-time multistep method based on Predict-Evaluate-Correct scheme of three-step fourth-order (RTPEC-34) method is proposed and developed for the real-time simulation of aircraft dynamics, which is improved by obtaining the predictor-corrector parameters using optimization algorithms. The method of using GA optimization with a punishment strategy is more stable and accurate than others both theoretically and practically. This work involves theoretical analysis of the truncation error of the predictor-corrector for real-time problems as well as of the stability of the proposed RTPEC-34 method. The analysis work can be

used by other applications that involve improvement of accuracy and stability. Both numerical examples and engineering examples are used in simulation experiments to evaluate the performance of the RTPEC-34 method by comparing it with the RK4 and the Rosenbrock methods. It is shown in the evaluation that the proposed method achieves improved performance in terms of accuracy, stability, and support of real-time simulation. Moreover, the successful application of the proposed method in the F-16 aircraft experiment has shown that the proposed method is adaptive to solve the multivariable ODEs popular in aircraft dynamics simulation.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (no. 61374163), the National High Technology Research and Development Program (863 Program) of China (no. 2013AA041302), and the Aviation Science Foundation of China (no. 20124060179).

References

- [1] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*, vol. 2, John Wiley & Sons, New York, NY, USA, 2003.
- [2] M. T. Heath, *Scientific Computing*, McGraw-Hill, New York, NY, USA, 1997.
- [3] Y. Xu and J. J. Zhao, "Estimation of longest stability interval for a kind of explicit linear multistep methods," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 912691, 18 pages, 2010.
- [4] Z. A. Majid, N. Z. Mokhtar, and M. Suleiman, "Direct two-point block one-step method for solving general second-order ordinary differential equations," *Mathematical Problems in Engineering*, vol. 2012, Article ID 184253, 16 pages, 2012.
- [5] G. G. Dahlquist, "A special stability problem for linear multistep methods," *BIT Numerical Mathematics*, vol. 3, pp. 27–43, 1963.

- [6] L. F. Shampine and C. W. Gear, "A user's view of solving stiff ordinary differential equations," *SIAM Review*, vol. 21, no. 1, pp. 1–17, 1979.
- [7] H. H. Rosenbrock, "Some general implicit processes for the numerical solution of differential equations," *The Computer Journal*, vol. 5, no. 4, pp. 329–330, 1963.
- [8] C. Huang, "Strong stability preserving hybrid methods," *Applied Numerical Mathematics*, vol. 59, no. 5, pp. 891–904, 2009.
- [9] S. Gottlieb, C.-W. Shu, and E. Tadmor, "Strong stability-preserving high-order time discretization methods," *SIAM Review*, vol. 43, no. 1, pp. 89–112, 2001.
- [10] M. V. Bulatov and G. V. Berghe, "Two-step fourth order methods for linear ODEs of the second order," *Numerical Algorithms*, vol. 51, no. 4, pp. 449–460, 2009.
- [11] C. Bresten, S. Gottlieb, Z. Grant, D. Higgs, D. I. Ketcheson, and Németh, "Strong stability preserving multistep Runge-Kutta methods," <http://arxiv.org/abs/1307.8058>.
- [12] H. Y. Seong, Z. A. Majid, and F. Ismail, "Solving second-order delay differential equations by direct Adams-Moulton method," *Mathematical Problems in Engineering*, vol. 2013, Article ID 261240, 7 pages, 2013.
- [13] H. Wang, H. Mao, and H. Zhang, "A variable-step interaction algorithm for multidisciplinary collaborative simulation," *Integrated Computer-Aided Engineering*, vol. 21, no. 3, pp. 263–279, 2014.
- [14] F. Bergero and E. Kofman, "PowerDEVS: a tool for hybrid system modeling and real-time simulation," *Simulation*, vol. 87, no. 1–2, pp. 113–132, 2011.
- [15] X. Wang, J. Zhang, and M. Scalia, "Parallel motion simulation of large-scale real-time crowd in a hierarchical environmental model," *Mathematical Problems in Engineering*, vol. 2012, Article ID 918497, 15 pages, 2012.
- [16] O. B. Widlund, "A note on unconditionally stable linear multistep methods," *BIT Numerical Mathematics*, vol. 7, pp. 65–70, 1967.
- [17] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley & Sons, New York, NY, USA, 2008.
- [18] Q. Y. Li, Z. Guan, and F. S. Bai, *Numerical Calculation Principle*, Tsinghua University Press, Beijing, China, 2000.
- [19] T. E. Hull, W. H. Enright, B. M. Fellen, and A. E. Sedgwick, "Comparing numerical methods for ordinary differential equations," *SIAM Journal on Numerical Analysis*, vol. 9, no. 4, pp. 603–637, 1972.
- [20] H. M. Zhang, S. Liang, S. J. Song, and H. W. Wang, "Truncation error calculation based on Richardson extrapolation for variable-step collaborative simulation," *Science China Information Sciences*, vol. 54, no. 6, pp. 1238–1250, 2011.
- [21] G. Zoutendijk, *Methods of Feasible Directions; A Study in Linear and Nonlinear Programming*, Elsevier, New York, NY, USA, 1960.
- [22] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1, pp. 95–110, 1956.
- [23] C. R. Houck, J. Joines, and M. Kay, "GA genetic algorithm for function optimization: a Matlab implementation," NCSU-IE TR, 1995.
- [24] O. Kramer and H.-P. Schwefel, "On three new approaches to handle constraints within evolution strategies," *Natural Computing*, vol. 5, no. 4, pp. 363–385, 2006.
- [25] F. Zong, H. Lin, B. Yu, and X. Pan, "Daily commute time prediction based on genetic algorithm," *Mathematical Problems in Engineering*, vol. 2012, Article ID 321574, 20 pages, 2012.
- [26] C. Coello and C. Artemio, "Constraint-handling techniques used with evolutionary algorithms," in *Proceedings of the 14th International Conference on Genetic and Evolutionary Computation Conference Companion*, ACM, 2012.
- [27] J. S. Litt, D. L. Simon, S. Garg et al., "A survey of intelligent control and health management technologies for aircraft propulsion systems," *Journal of Aerospace Computing, Information and Communication*, vol. 1, no. 12, pp. 543–563, 2004.
- [28] L. Stevens, *Aircraft Control and Simulation*, John Wiley & Sons, 1992.
- [29] R. S. Russel, "Nonlinear f-16 simulation using Simulink and matlab," Tech. Rep., University of Minnesota, 2003.
- [30] J. A. Mulder, W. H. J. J. van Staveren, and J. C. van der Vaart, "Flight dynamics," Tech. Rep., Delft University of Technology, 2000.

