Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 5910928, 8 pages
http://dx.doi.org/10.1155/2016/5910928

Research Article

Aerodynamic Parameter Estimation of a Symmetric Projectile Using Adaptive Chaotic Mutation Particle Swarm Optimization

Jun Guan,[1] Wenjun Yi,[1] Sijiang Chang,[2] and Xiaoyuan Li[3]

[1] *National Key Laboratory of Transient Physics, Nanjing University of Science & Technology, Nanjing 210094, China*
[2] *School of Power and Engineering, Nanjing University of Science & Technology, Nanjing 210094, China*
[3] *Navy Equipment Research Institute, Beijing 100073, China*

Correspondence should be addressed to Jun Guan; guanjun8710@163.com

Received 27 April 2016; Accepted 7 August 2016

Academic Editor: Zhike Peng

This article details a new optimizing algorithm called Adaptive Chaotic Mutation Particle Swarm Optimization (ACM-PSO). The new algorithm is used to perform aerodynamic parameter estimation on a spinning symmetric projectile. The main creative ideas of this new algorithm are as follows. First, a self-adaptive weight function is used so that the inertial weight can be adjusted dynamically by itself. Second, the initialized particle is generated by chaos theory. Last, a method that can be used to judge whether the algorithm has fallen into a local optimum is established. The common testing function is used to test the new algorithm, and the result shows that, compared with the basic particle swarm optimization (PSO) algorithm, it is more likely to have a quick convergence and high accuracy and precision, leading to extensive application. Simulated ballistic data are used as testing data, and the data are subjected to the new algorithm to identify the aerodynamic parameters of a spinning symmetric projectile. The result shows that the algorithm proposed in this paper can effectively identify the aerodynamic parameters with high precision and a quick convergence velocity and is therefore suitable for use in actual engineering.

1. Introduction

Artillery forms an important wing of the army in providing firepower during both war and cross-border skirmishes with an enemy. The traditional artillery shell aeroballistic theory has been developed over several years, and it is an important part of the field of aerodynamics. The accuracy of an artillery shell is an important indicator that is used to estimate its effectiveness. The mathematical models of an artillery shell have often been used for predicting its behavior and flight performance from a conventional approach [1].

The estimation of aerodynamic parameters in the preliminary design stage for vehicles, such as airplanes, missiles, and gun-launched weapons by theoretical methods is very useful. Computational fluid dynamics has recently [2] positively influenced the analytical scenario by providing numerical solutions of the total configuration via sophisticated Euler and Navier-Stokes flow solvers. Several methods based on experimental approaches are essential to validating the analytical result. Although wind tunnel experiments have been used to improve the precision of the estimated parameters, it is difficult to simulate the right flight conditions and weather circumstances. Moreover, the vehicle model tested in the tunnel is slightly different from the actual model used in flight due to last-minute changes [3]. Consequently, an analysis guided by flight data appears to be the best option [4]. Aerodynamic parameter identification is the most developed field in conventional aircraft system identification and has been successfully applied to aircraft and missiles [5]. Suk et al. [6] used maximum likelihood estimation and the extended Kalman filter to identify the system of a UAV in 2003. Tang et al. [7] used a numerically robust least-squares estimator in the frequency domain to identify the aircraft flutter modal parameters in 2008. Burchett [8] used an improved gradient-based method to estimate the aerodynamic coefficients of a projectile from flight range data. Singh and Ghosh [9] used the neural network method to identify the aerocoefficients. Wu et al. [10] designed signals to excite the longitudinal motion of a fly-by-wire passenger airliner to identify the aerodynamic parameters in 2013. However, there have been few studies on spinning symmetric projectiles in recent years. Intelligent identification algorithms have been widely used in the field of aerodynamic coefficient identification with the development of optimization theories [11].

This paper proposes a new PSO algorithm to strengthen the properties of the basic PSO. The new algorithm is then used to identify the aerocoefficients of a spinning projectile. The paper is organized as follows. Section 1 introduces the background of gun-launched projectile aerodynamic parameter identification. Section 2 presents the mathematical modeling of a spinning projectile with 6 degrees of freedom. Section 3 gives the algorithm for the basic PSO and ACM-PSO. Section 4 uses some testing functions to verify the performance of the ACM-PSO. Section 5 uses ACM-PSO to identify the aerodynamic parameters using simulated data. Section 6 is the conclusion.

2. Modeling

To acquire a more precise description of the motion of a spinning projectile, six-degree-of-freedom flight dynamic equations are used in this paper. The detailed expressions of these equations are as follows:

$$
\begin{aligned}
\frac{dv}{dt} &= \frac{1}{m}\left(-\frac{\rho v_r}{2}Sc_{x_0}\left(v-w_{x_2}\right)-\frac{\rho v_r}{2}Sc_{x_2}\delta_r^2\left(v-w_{x_2}\right)\right. \\
&\quad +\frac{\rho S}{2}c_y\frac{v_r^2\cos\delta_2\cos\delta_1}{\sin\delta_r}-\frac{\rho S}{2}c_y\frac{v_{r\varepsilon}\left(v-w_{x_2}\right)}{\sin\delta_r}-\frac{\rho v_r}{2} \\
&\quad \cdot Sc_z\frac{w_{z_2}\cos\delta_2\sin\delta_1}{\sin\delta_r}+\frac{\rho v_r}{2}Sc_z\frac{w_{y_2}\sin\delta_2}{\sin\delta_r}-mg \\
&\quad \left.\cdot\sin\theta_a\cos\psi_2\right), \\
\frac{d\theta_a}{dt} &= \frac{1}{m}\cdot\left(\frac{\rho v_r Sc_x w_{y_2}}{2v\cos\psi_2}\right. \\
&\quad +\frac{\rho Sc_y\left(v_r^2\cos\delta_2\sin\delta_1+v_{r\zeta}w_{y_2}\right)}{2v\cos\psi_2\sin\delta_r} \\
&\quad -\frac{\rho v_r^2 Sc_y'\delta_N\cos\gamma_1}{2v\cos\psi_2} \\
&\quad +\frac{\rho v_r Sc_z\left[\left(v-w_{x_2}\right)\sin\delta_2+w_{z_2}\cos\delta_2\cos\delta_1\right]}{2v\cos\psi_2\sin\delta_r} \\
&\quad -\frac{mg\cos\theta_a}{v\cos\psi_2}+\frac{2\Omega_E mv}{v\cos\psi_2}\left(\sin\psi_2\cos\theta_a\cos\Lambda\cos\alpha_N\right. \\
&\quad \left.\left.+\sin\theta_a\sin\psi_2\sin\Lambda+\cos\psi_2\cos\Lambda\sin\alpha_N\right)\right), \\
\frac{d\psi_2}{dt} &= \frac{1}{m}\left(\frac{\rho v_r}{2v}Sc_x w_{z_2}+\frac{\rho S}{2v}c_y\frac{1}{\sin\delta_r}\left[v_r^2\sin\delta_2\right.\right. \\
&\quad \left.+v_{r\zeta}w_{z_2}\right]-\frac{\rho v^2 Sc_y'\delta_N\sin\gamma_1}{2v}+\frac{\rho v_r}{2v}Sc_z \\
&\quad \cdot\frac{1}{\sin\delta_r}\left(-w_{y_2}\cos\delta_2\cos\delta_1\right)-\frac{\rho v_r}{2v}Sc_z\frac{1}{\sin\delta_r}\left(v\right. \\
&\quad \left.-w_{x_2}\right)\cos\delta_2\sin\delta_1+\frac{1}{v}mg\sin\theta_a\sin\psi_2 \\
&\quad \left.+2\Omega_E m\left(\sin\Lambda\cos\theta_a-\cos\Lambda\sin\theta_a\cos\alpha_N\right)\right), \\
\frac{d\omega_\varepsilon}{dt} &= \frac{1}{C}\left[-\frac{\rho Sld}{2}m_{xz}'v_r\omega_\varepsilon+\frac{\rho v_r^2}{2}Slm_{xw}'\delta_f\right], \\
\frac{d\omega_\eta}{dt} &= \frac{1}{A}\left[\frac{\rho Sl}{2}v_r m_z\frac{1}{\sin\delta_r}v_{r\zeta}-\frac{\rho Sld}{2}v_r m_{zz}'\omega_\eta-\frac{\rho Sld}{2}\right. \\
&\quad \left.\cdot m_y'\frac{1}{\sin\delta_r}\omega_\varepsilon v_{r\eta}-\frac{\rho v^2 Slm_z'\delta_M\sin\gamma_2}{2}\right]-\frac{C}{A}\omega_\varepsilon\omega_\zeta \\
&\quad +\omega_\zeta^2\tan\varphi_2, \\
\frac{d\omega_\zeta}{dt} &= \frac{1}{A}\left[\frac{\rho Sl}{2}v_r m_z\frac{1}{\sin\delta_r}v_{r\eta}-\frac{\rho Sld}{2}v_r m_{zz}'\omega_\zeta-\frac{\rho Sld}{2}\right. \\
&\quad \left.\cdot m_y'\frac{1}{\sin\delta_r}\omega_\varepsilon v_{r\zeta}+\frac{\rho v^2 Slm_z'\delta_M}{2}\right]+\frac{C}{A}\omega_\varepsilon\omega_\eta-\omega_\eta\omega_\zeta \\
&\quad \cdot\tan\varphi_2, \\
\frac{d\varphi_a}{dt} &= \frac{\omega_\zeta}{\cos\varphi_2}, \\
\frac{d\varphi_2}{dt} &= -\omega_\eta, \\
\frac{d\gamma}{dt} &= \omega_\varepsilon-\omega_\zeta\tan\varphi_2, \\
\frac{dx}{dt} &= v\cos\psi_2\cos\theta_a, \\
\frac{dy}{dt} &= v\cos\psi_2\sin\theta_a, \\
\frac{dz}{dt} &= v\sin\psi_2.
\end{aligned}
\tag{1}
$$

The physical meanings of each variable in (1) and other unlisted equations can be found in [12].

3. Algorithm

3.1. Particle Swarm Optimization (PSO). Particle swarm optimization [13] is an intelligent optimization algorithm that was proposed by Kenny and Eberhart in 1995. PSO is a population-based optimization tool that could be used to solve several function optimization problems or problems that can be transformed to function optimization problems. Compared with other global optimization algorithms, like Genetic Algorithms or Simulated Annealing, the main strength of PSO is its fast convergence. PSO is inspired by the migration and aggregation of bird flocks as they seek foods. PSO is initialized with a population of random solutions [14]. The potential solutions called particles fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution achieved so far. This value is called *pbest*. Another value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the neighborhood of the particle. This value is called *lbest*. The particle also takes the entire population as its topological neighbors, and the best value is a global best, which is called *gbest*. The position of an individual particle is adjusted according to its own previous searching experience. The best solution is determined by its objective function value. The general procedure of PSO is as follows.

Step 1 (determination of necessary parameters). The main parameters of a basic PSO include the population size N, particle dimensions D, inertial weight w, personal cognition coefficient c_1, social cognition coefficient c_2, and maximum iterated generations $k_{\max}$.

Step 2 (initialization). The initialized particles are generated by the methods of randomization. For each particle, both its position and velocity must be initialized.

Step 3 (position and velocity changes). The position of a particle is influenced by its velocity. Let $\mathbf{x}_i(k)$ denote the position of particle i in the solution space at time step k. Through adding the velocity $\mathbf{v}_i(k)$ as shown in formula (2), the position is updated. The velocity-updating law is described in formula (3):

$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + \mathbf{v}_i(k+1), \tag{2}$$

$$\begin{aligned}\mathbf{v}_i(k+1) = {} & w\mathbf{v}_i(k) + c_1 r_1 (\mathbf{p}_i(k) - \mathbf{x}_i(k)) \\ & + c_2 r_2 (\mathbf{p}_g(k) - \mathbf{x}_i(k)),\end{aligned} \tag{3}$$

where $\mathbf{v}_i(k)$ and $\mathbf{x}_i(k)$ represent the particle's previous speed and position, respectively. $\mathbf{v}_i(k+1)$ and $\mathbf{x}_i(k+1)$ are the particle's current speed and position, respectively. w is the inertial weight coefficient; r_1 and r_2 are random numbers that are in the range of $[0, 1]$; and c_1 and c_2 are the personal and social cognitive coefficients, which have fixed values in the basic PSO. $\mathbf{p}_i(k)$ is the previous individual best position of particle i, and $\mathbf{p}_g(k)$ is the previous best particle position for all of the particles.

Step 4 (calculation of the fitness function). Calculate each particle's fitness value, and update $\mathbf{p}_i(k)$ if the current fitness value of the ith particle is better than the previous value. Compare the fitness value of each particle with its $\mathbf{p}_i(k)$, and if the current value is better, update $\mathbf{p}_g(k)$. $\mathbf{p}_g(k)$ can be updated as shown in

$$\mathbf{p}_g(k) = \arg\min \{ f(\mathbf{p}_i(k)) \mid i = 1, 2, \ldots, N \}, \tag{4}$$

where $f(\cdot)$ denotes a fitness function.

Step 5 (termination). Stop the algorithm if the stopping condition is satisfied; if not, go to Step 3.

3.2. Adaptive Chaotic Mutation Particle Swarm Optimization.

The classical PSO is widely used in the area of function optimization, parameter identification, and control system design. PSO has advantages including simple computation and quick convergence but also some disadvantages. For example, long computation times, undispersed initial particles, and easing to fall into local optimum. To address these problems, a modified PSO called Adaptive Chaotic Mutation Particle Swarm Optimization (ACM-PSO) is proposed in this paper. The new algorithm has specific powers, such as short computation time; the generated initial particles will be more dispersed in the solution space and more likely to have a global optimization.

3.2.1. Adaptive Inertial Weight.

The inertial weight w in (3) is employed to manipulate the impact of the previous history of the velocities on the current velocity [15]. The inertial weight resolves the tradeoff between the global and local exploration abilities of the swarm. Many researchers have advocated that the value of w should be large in the exploration (global optimization) state and small in the exploitation (local optimization) state [16]. A large inertial weight encourages global exploration, while a small one enhances local optimization, that is, fine-tuning the current search area. A proper value of the inertial weight supplies the expected balance between the global and local optimization abilities of the particles and consequently improves the effectiveness of the PSO algorithm. It has been shown in many experiments that a large value for the initialized w gives a quick convergence to the global exploration in the solution space, and w gradually decreases to acquire refined solutions. A new adaptive updating law for the inertial weight is proposed in this paper:

$$w_i^k = w_{\max} - (w_{\max} - w_{\min}) \frac{k}{k_{\max}} e^{\|\mathbf{x}_i^k - \mathbf{p}_g^k\|}, \tag{5}$$

where w_i^k denotes the inertial weight of particle i in the kth generation, $w_{\max}$ is the maximum inertial weight, and $w_{\min}$ is the minimum value. Formula (3) can now be rewritten as the following expression:

$$\begin{aligned}\mathbf{v}_i(k+1) = {} & w_i^k \mathbf{v}_i(k) + c_1 r_1 (\mathbf{p}_i(k) - \mathbf{x}_i(k)) \\ & + c_2 r_2 (\mathbf{p}_g(k) - \mathbf{x}_i(k)).\end{aligned} \tag{6}$$

The physical meaning of formula (5) can be described as follows. The larger the generation number k is, the smaller the inertial weight w_i^k becomes. The norm $\|\mathbf{x}_i^k - \mathbf{p}_g^k\|$ denotes the distance between the ith particle and the global best particle. When the value is small and the value of w_i^k is also small, it means that this particle is near the global solution, and a small value of w_i^k enables a refined search to obtain more precise solutions.

3.2.2. Chaotic Initialization. In the PSO algorithm, the initialization of particles is usually randomly generated. The convergence speed of the whole search process and the optimization efficiency of the algorithm are directly affected by the initial position of the spread degree and the uniform characteristic of its position in the search space. It is not likely for the initialization of particles to be a uniform distribution if the solution space has a large dimension or only a small number of particles. Chaos is a common nonlinear phenomenon whose behavior is seemingly complex and random but actually has strong internal rules. Chaos has features of randomness, ergodicity, and sensitivity to initial conditions. An initialization strategy utilizing chaotic theory is used in this paper. Cubical mapping is selected to produce the initial sequence. The mathematical form is as follows:

$$\begin{gathered} x_{n+1} = 4x_n^3 - 3x_n, \\ -1 < x_1 < 1. \end{gathered} \tag{7}$$

3.2.3. Premature Judge Criterion. PSO can easily fall into local optima. In this paper, a mutate operator is introduced to the classical PSO to enhance the global optimization ability. When the criterion of premature is satisfied, the mutation operator is applied to the global optimization particle $\mathbf{p}_g^k$ to make the algorithm jump out of the local convergence. A fitness variance is used to judge whether the local convergence has occurred. The mathematical form of the fitness variance σ^2 is shown in

$$\sigma^2 = \sum_{i=1}^{N} \left(\frac{f\left(\mathbf{x}_i^k\right) - f_{\text{average}}^k}{\max\left(\max\left|f\left(\mathbf{x}_i^k\right) - f_{\text{average}}^k\right|, 1\right)} \right)^2, \tag{8}$$

where f_{average}^k is the average fitness value of all of the particles in the kth generation. σ^2 denotes the convergence degree of the algorithm. If the value of σ^2 is not only smaller than a set threshold but also larger than the smallest theoretical fitness variance, we can judge that the algorithm has fallen into a local convergence. Once the algorithm is in a local convergence, a mutation operator in accordance with a certain probability is applied to $\mathbf{p}_g^k = [p_{g1}^k, p_{g2}^k, \ldots, p_{gD}^k]^{\mathrm{T}}$:

$$p_{gi}^k = p_{gi}^k \left(1 + 0.5\mu\right) \quad (i = 1, 2, \ldots, D), \tag{9}$$

where μ is Gaussian white noise.

3.2.4. Procedure of ACM-PSO

Step 1 (determination of relevant parameters). The main parameters have a population size N, the dimension of each particle is D, the maximum and minimum values of the inertial weight are $w_{\max}$ and $w_{\min}$, respectively, the personal cognitive coefficient is c_1, the social cognitive coefficient is c_2, and the maximum iteration generation number is $k_{\max}$.

Step 2 (initialization of particles). Use the method detailed in Section 3.2.2 to generate initial chaotic particles.

Step 3 (update inertial weight). According to formula (5), update the inertial weight.

Step 4 (update position and velocity). According to formulas (2) and (6), update the position and velocity.

Step 5 (judge whether the algorithm has fallen into a local convergence). Use the method detailed in Section 3.2.3 to judge whether local convergence has occurred. If premature convergence has occurred, formula (9) can be used to do a mutation operation to help the algorithm jump out. Otherwise, go to Step 6.

Step 6 (termination). Stop the algorithm if the stopping criterion is met; otherwise, go to Step 3.

4. Algorithm Validation

To verify the effectiveness of the ACM-PSO proposed in this paper, standard testing functions are used. In this paper, the sphere function f_1, Rastrigin function f_2, and Schaffer function f_3 are used as testing functions. The mathematical formulas of these functions are as follows:

Sphere function:

$$f_1 = \sum_{i=1}^{30} x_i^2, \quad x_i \in [-100, 100] \tag{10}$$

Rastrigin function:

$$f_2 = \sum_{i=1}^{30} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right], \quad x_i \in [-5.12, 5.12] \tag{11}$$

Schaffer function:

$$f_3 = 0.5 + \frac{\left(\sin\sqrt{x_1^2 + x_2^2}\right)^2 - 0.5}{\left(1 + 0.001\left(x_1^2 + x_2^2\right)\right)^2}, \quad x_i \in [-10, 10] \tag{12}$$

The necessary parameters are set as follows: $k_{\max} = 1000$, $v_{\max} = 0.3$, $v_{\min} = -0.3$, $w_{\max} = 0.9$, $w_{\min} = 0.5$, $c_1 = 1.49618$, $c_2 = 1.49618$, and σ^2 threshold of 0.05. The testing result is shown in Table 1.

It can be seen from Table 1 that the performance of ACM-PSO is remarkably better than that of classical PSO.

The solution of ACM-PSO is closer to the true value than that of PSO, and the minimum value, average value, and variance value are also smaller than those of classical PSO. The fitness value change with the generation number

Table 1: Objective values of different testing functions.

Test function	Algorithm	Particle number	Maximum value	Minimum value	Average value	Variance	True value	Computation time consumed (s)
Sphere	PSO	$N = 50$	0.449	0.0321	0.1295	0.0053	0	1.1822
		$N = 100$	0.0507	0.0036	0.0183	$1.13E-04$	0	1.5750
		$N = 200$	0.0089	$1.02E-04$	0.0021	$2.70E-06$	0	2.4435
	ACM-PSO	$N = 50$	$5.89E-37$	$2.37E-52$	$6.46E-39$	$3.48E-75$	0	0.8866
		$N = 100$	$8.77E-44$	$9.60E-58$	$1.60E-45$	$1.13E-88$	0	1.1812
		$N = 200$	$9.81E-49$	$2.56E-60$	$1.35E-50$	$1.01E-98$	0	1.8326
Rastrigin	PSO	$N = 50$	0.0703	0.0189	$4.46E-02$	$2.23E-05$	0	1.3238
		$N = 100$	0.0601	0.0024	0.0182	$1.16E-04$	0	1.7426
		$N = 200$	0.0084	$3.07E-04$	0.0021	$3.12E-06$	0	2.6602
	ACM-PSO	$N = 50$	2.1745	$1.46E-08$	0.6807	0.3825	0	0.9928
		$N = 100$	2.2075	$1.73E-10$	0.5769	0.3643	0	1.3069
		$N = 200$	2.3464	$8.66E-12$	0.4469	0.3093	0	1.9951
Schaffer	PSO	$N = 50$	0.0372	$9.70E-03$	$1.03E-02$	$1.50E-05$	0	1.3453
		$N = 100$	0.0372	0.0097	0.0100	$7.57E-06$	0	1.7610
		$N = 200$	0.0097	$1.11E-16$	0.0096	$9.44E-07$	0	2.7522
	ACM-PSO	$N = 50$	0.0097	$0.00E+00$	$6.35E-04$	$5.58E-06$	0	1.0090
		$N = 100$	0.0097	0.0000	$9.72E-05$	$9.44E-07$	0	1.3207
		$N = 200$	0.0000	0.0000	0.0000	0.0000	0	2.0641

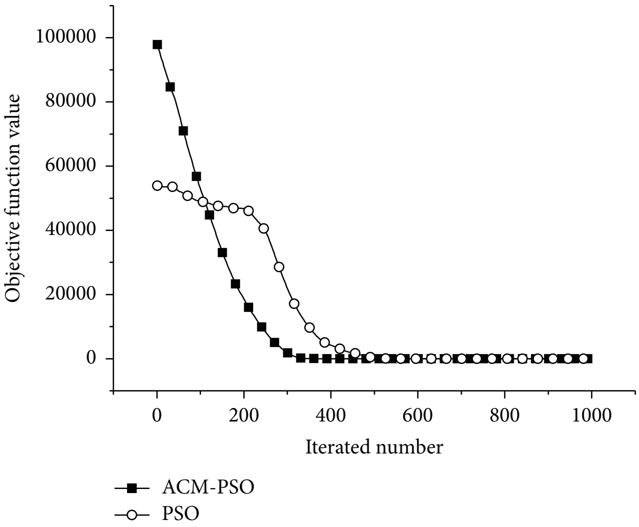


Figure 1: Relationship between objective value and generation number.

is depicted in Figure 1. It can be seen from this figure that ACM-PSO reaches convergence in approximately the 300th generation, while PSO converges in approximately the 500th generation. The convergence velocity of the former is higher than that of the latter. In conclusion, the ACM-PSO proposed in this paper not only can effectively resolve the problem of function optimization but also provides more precise solutions. The algorithm can thus meet the needs of engineering.

5. Projectile Aerodynamic Parameter Identification

In this paper, ACM-PSO is used to identify the main parameters of a spinning projectile. The parameters identified in this paper are the zero-yaw drag coefficient c_{x0}, yaw drag coefficient c_{x2}, linear lift coefficient c'_y, and linear overturning moment coefficient m'_z. The basic task of parameter identification is to find a group of parameters that minimize an objective function.

5.1. Objective Function. The maximum likelihood estimate (MLE) used in parameter identification is asymptotically unbiased, asymptotically uniform, and asymptotically efficient. Practice has also proven that MLE is an effective parameter identification method. Suppose $\boldsymbol{\theta}$ is a parameter vector that is waiting to be estimated:

$$\boldsymbol{\theta} = \left[m'_z \;\; c_{x_0} \;\; c_{x_2} \;\; c'_y \right]^{\mathrm{T}}. \tag{13}$$

$\mathbf{y}_m(i)$ is the measurement vector, $\mathbf{y}(i)$ is the calculated vector, and i denotes the time t_i:

$$\begin{aligned} \mathbf{y}_m(i) &= \left[\delta_{rm}(t_i) \;\; v_m(t_i) \;\; x_m(t_i) \;\; y_m(t_i) \;\; z_m(t_i) \right]^{\mathrm{T}}, \\ \mathbf{y}(i) &= \left[\delta_r(t_i) \;\; v(t_i) \;\; x(t_i) \;\; y(t_i) \;\; z(t_i) \right]^{\mathrm{T}}. \end{aligned} \tag{14}$$

The maximum likelihood criterion is chosen as the objective function, and its mathematical expression is $J(\boldsymbol{\theta})$:

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{M} \left[\mathbf{e}^{\mathrm{T}}(i)\, \mathbf{R}^{-1} \mathbf{e}(i) + \ln |\mathbf{R}| \right], \tag{15}$$

Table 2: Results of identified aerodynamic parameters.

Parameters	True value	ACM-PSO			
		Case 1	Case 2	Case 3	Case 4
c_{x_0}	0.4000	0.4000	0.4000	0.3999	0.4002
c_{x_2}	4.5000	4.5000	4.5002	4.4975	4.4879
c_y'	2.5000	2.5000	2.4961	2.4933	2.4901
m_z'	4.0000	3.999	3.9875	3.9856	3.9833

Case 1: simulated data as radar output directly.
Case 2: add Gaussian white noise to simulated data as radar output with SNR of 30 dB.
Case 3: add Gaussian white noise to simulated data as radar output with SNR of 20 dB.
Case 4: add Gaussian white noise to simulated data as radar output with SNR of 10 dB.

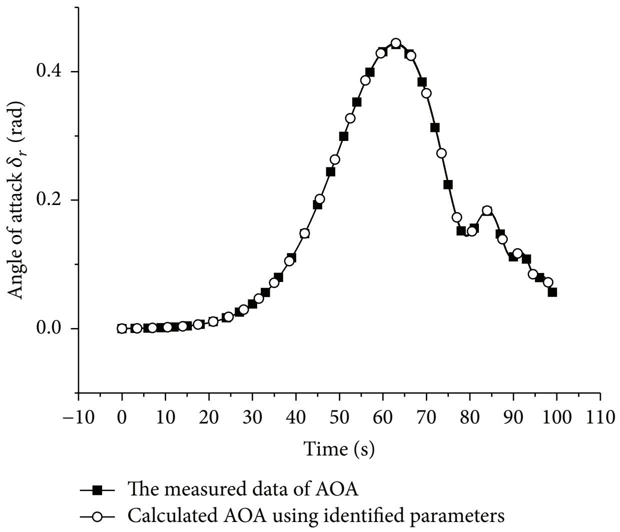


Figure 2: Comparison of angles of attack.

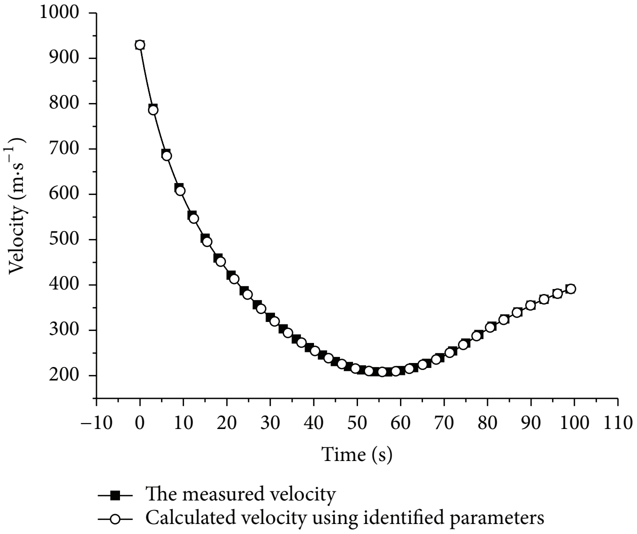


Figure 3: Comparison of velocities.

where $\mathbf{e}$ is the error vector between $\mathbf{y}_m$ and $\mathbf{y}$, which can be described as $\mathbf{e}(i) = \mathbf{y}(i) - \mathbf{y}_m(i)$. M is the number of data points. $\mathbf{R}$ is the covariance matrix of measurement noise. As the statistical property of the measurement noise is unknown, the optimum estimation $\widehat{\mathbf{R}}$ usually replaces $\mathbf{R}$:

$$\mathbf{R} \approx \widehat{\mathbf{R}} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{e}(i)\,\mathbf{e}^{\mathrm{T}}(i). \tag{16}$$

5.2. Simulation Validation. With the background of a high spinning projectile, 6-degree-of-freedom flight equations are used for generating simulated flight test data. We use the simulated data as the output of a radar combined with the PSO algorithm identifying high spinning projectile aerodynamic parameters to validate the performance of the PSO algorithm. The identified result can be seen in Table 2.

It is often necessary to put the identified parameters into flight dynamic equations to examine whether the identified parameters satisfy the accuracy requirements of engineering. This checking method is also used in this paper. Figures 2–6 provide the angle of attack, velocity, distance, height, and side deviation, respectively, which are calculated using the identified parameters that are compared with the original radar data. It can be seen from these figures that the bias between two curves in every figure is very small. From the figure data analyses, we can confirm that the identification

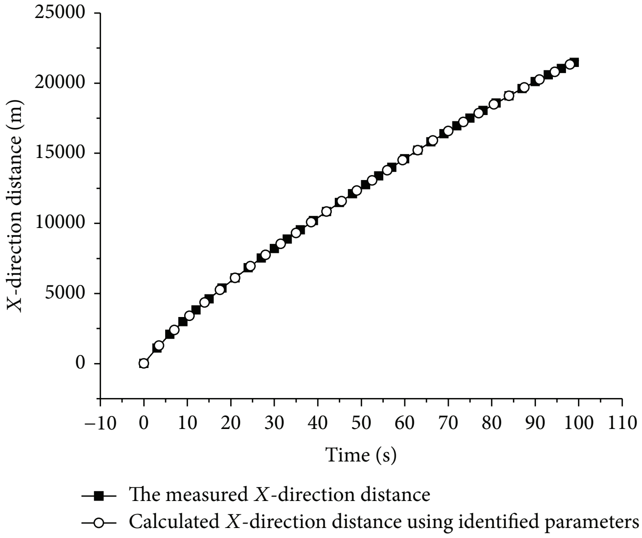


Figure 4: Comparison of x-direction distances.

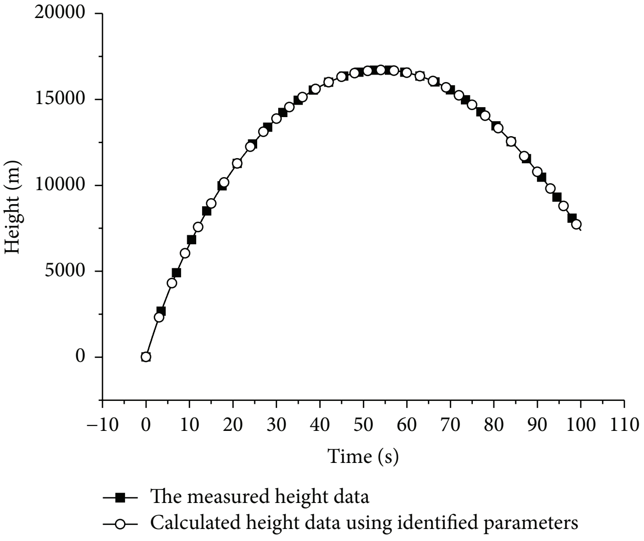


Figure 5: Comparison of the ballistic heights.

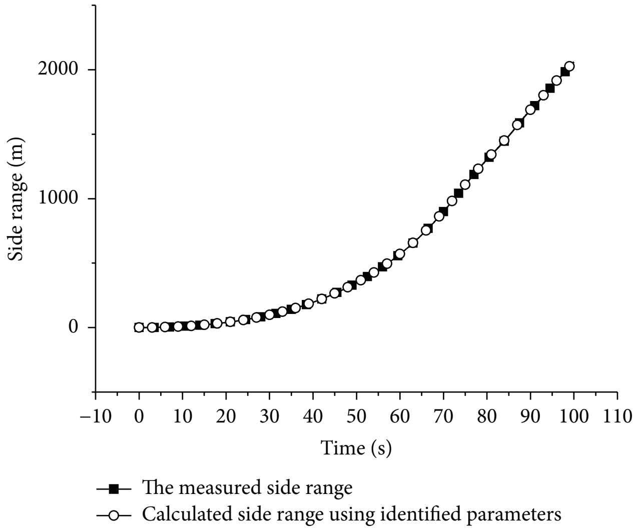


Figure 6: Comparison of the side ranges.

result is very accurate for an uncontrolled gun-launched projectile.

6. Conclusions

The algorithm of ACM-PSO is proposed based on the PSO method and a mathematical model for the 6-degree-of-freedom of spinning projectile flight dynamic equations. ACM-PSO is introduced and applied towards parameter identification for a gun-launched spinning projectile:

(1) The result of simulation validation shows that the mathematical model is correct, and the aerodynamic parameters and coefficients can be identified accurately. This method can be applied in practical engineering.

(2) ACM-PSO has strong global and local search abilities for the spinning projectile aerodynamic parameter identification.

(3) The identification result has some difference from the true value, but the error is allowable.

Competing Interests

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

Acknowledgments

This work was supported by the National Natural Science Fund Project of China (nos. 11472136 and 11402117).

References

[1] G. G. Dutta, A. Singhal, A. Kushari, and A. K. Ghosh, “Estimation of drag coefficient from radar-tracked flight data of a cargo shell,” *Defence Science Journal*, vol. 58, no. 3, pp. 377–389, 2008.

[2] Umit Kutluay, *Aerodynamic parameter estimation using flight test data [thesis]*, Middle East Technical University, Ankara, Turkey, 2011.

[3] V. Condaminet and F. Delvare, “Identification of aerodynamic coefficients of a projectile and reconstruction of its trajectory from partial flight data,” *Computer Assisted Methods in Engineering and Science*, vol. 21, pp. 177–186, 2014.

[4] K. W. Iliff, “Parameter estimation for flight vehicles,” *Journal of Guidance, Control, and Dynamics*, vol. 12, no. 5, pp. 609–622, 1989.

[5] T. Jiang, J. Li, and K. Huang, “Longitudinal parameter identification of a small unmanned aerial vehicle based on modified particle swarm optimization,” *Chinese Journal of Aeronautics*, vol. 28, no. 3, pp. 865–873, 2015.

[6] J. Suk, Y. Lee, S. Kim, H. Koo, and J. Kim, “System identification and stability evaluation of an unmanned aerial vehicle from automated flight tests,” *KSME International Journal*, vol. 17, no. 5, pp. 654–667, 2003.

[7] W. Tang, Z. Shi, and J. Chen, “Aircraft flutter modal parameter identification using a numerically robust least-squares estimator in frequency domain,” *Chinese Journal of Aeronautics*, vol. 21, no. 6, pp. 550–558, 2008.

[8] B. T. Burchett, “Aerodynamic parameter identification for symmetric projectiles: an improved gradient based method,” *Aerospace Science and Technology*, vol. 30, no. 1, pp. 119–127, 2013.

[9] S. Singh and A. K. Ghosh, “Parameter estimation from flight data of a missile using maximum likelihood and neural network method,” in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Keystone, Colo, USA, August 2006.

[10] Z. Wu, L. Wang, Z. Xu, and X. Tan, “Investigation of longitudinal aerodynamic parameters identification method for fly-by-wire passenger airliners,” *Chinese Journal of Aeronautics*, vol. 26, no. 5, pp. 1156–1163, 2013.

[11] X.-S. Gan, J.-S. Duanmu, Y.-B. Meng, and W. Cong, “Wavelet neural network aerodynamic modeling from flight data based on pso algorithm with information sharing and velocity disturbance,” *Journal of Central South University*, vol. 20, no. 6, pp. 1592–1601, 2013.

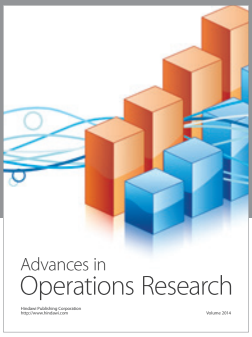
[12] Z. P. Han, *Exterior Ballistics of Projectiles and Rockets*, Beijing Institute of Technology Press, Beijing, China, 2014 (Chinese).

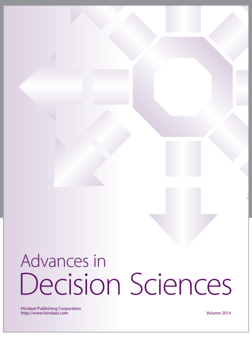
[13] G. Venter, “Particle swarm optimization,” in *Proceedings of the 43rd AIAA/ASME/ASCE/ASC Structure Dynamics, and Materials Conference*, pp. 22–25, Denver, Colo, USA, April 2002.

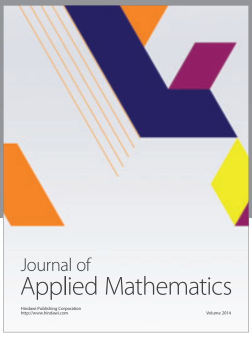
[14] A. T. Chase and R. A. Mcdonald, “Flight testing small UAVs for aerodynamic parameter estimation,” in *Proceedings of the AIAA Atompspheric Flight Mechanics Conference*, National Harbor, Md, USA, 2014.

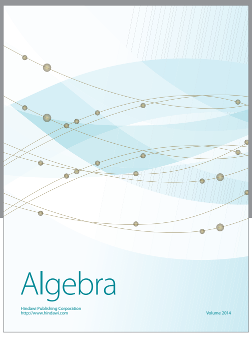
[15] Z. Hou, “Hammerstein model identification based on adaptive particle swarm optimization,” in *Proceedings of the Workshop on Intelligent Information Technology Application*, vol. 52, pp. 137–140, December 2007.

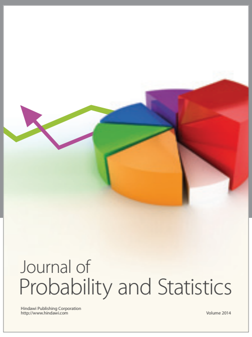
[16] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, “Adaptive particle swarm optimization,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.


Advances in
Operations Research

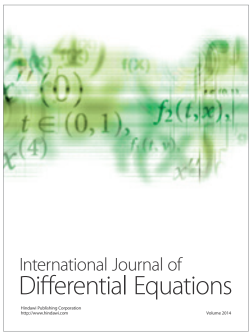

Advances in
Decision Sciences


Journal of
Applied Mathematics

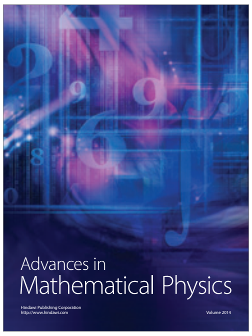

Algebra

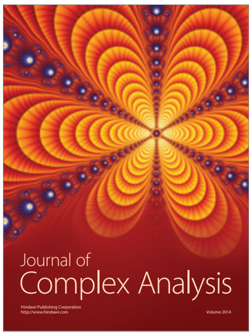

Journal of
Probability and Statistics

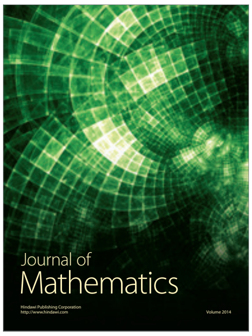
The Scientific
World Journal

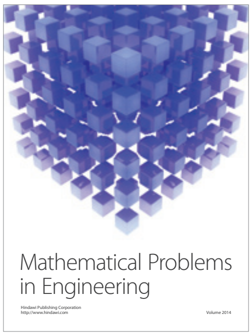

International Journal of
Differential Equations

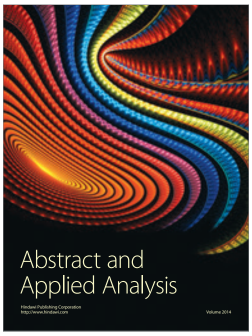
International Journal of
Combinatorics

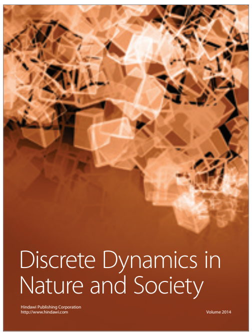

Advances in
Mathematical Physics

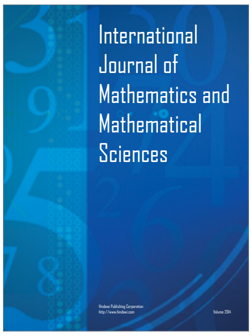

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

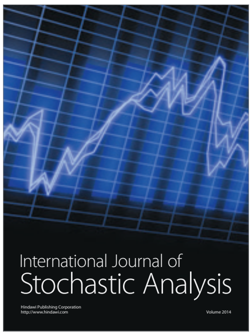

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization