DOI 10.1007/s10846-012-9656-y

Chaos-Genetic Algorithm for the System Identification of a Small Unmanned Helicopter

Tianmiao Wang · Yang Chen · Jianhong Liang · Yongliang Wu · Chaolei Wang · Yicheng Zhang

Received: 9 June 2011 / Accepted: 25 January 2012 / Published online: 8 February 2012
© Springer Science+Business Media B.V. 2012

Abstract This paper focuses on the system identification of a small unmanned helicopter in hover or low-speed flight conditions. A novel genetic algorithm including chaotic optimization operation named chaos-genetic algorithm (CGA) is proposed to identify the linear helicopter model. Based on the input-output data collected from real flight tests, the identification performance of CGA is compared with those calculated by the traditional genetic algorithm (TGA) and the prediction error method (PEM). The accuracy of the identified model is verified by simulation in time domain. Additionally, the small unmanned helicopter is stabilized by a linear quadratic Gaussian (LQG) regulator based on the proposed identified model. In the automatic flight experiments, the achievement of automatic take-off and landing, hovering performance within a 1.2 m diameter circle and point-to-point horizontal polyline flight also demonstrates the accuracy of the identified model and the effectiveness of the proposed method.

Keywords Small unmanned helicopter · Chaos-genetic algorithm · Identification · Flight tests

1 Introduction

With the advantages relative to fixed-wing aircraft that the helicopter can fly vertically and maneuver in tight spaces, especially its ability of hovering over interesting areas, small unmanned helicopter has been much attractive both in the military and the civilian application domains recently [1].

In terms of academic study, the complex nonlinear system is still drawing many researchers to take the challenge of realizing the automation of a small unmanned helicopter, especially the design strategies of motion controllers based on modern control theory to achieve the required specification. And an appropriate system model capable of capturing the main characteristics of the helicopter is the basis for the design of a satisfactory controller.

During recent years, the approaches to system identification of a small unmanned helicopter appear on the frequency domain and the time domain. A frequency domain method developed by the US Army and NASA named CIFER is the high-quality extraction of a complete multi-input/multi-output (MIMO) set of non-parametric input-to-output frequency responses. The iden-

T. Wang · Y. Chen (✉) · J. Liang · Y. Wu ·
C. Wang · Y. Zhang
Robotics Institute, Beijing University of Aeronautics
and Astronautics, Beijing 100191, China
e-mail: chenyang4117@yahoo.com

tification of the linear models for R50/RMAX helicopter in hover and cruise flight conditions using CIFER is discussed in [2, 5]. Adiprawita et al. [6] identified a simple model for a helicopter in X-Plane simulation. The secant method in the frequency domain [7] is used to obtain three semi-decoupled MIMO models for a small-size helicopter at hover. However, the data for frequency domain identification need expert pilots to do the frequency sweep inputs to obtain the desired frequency spectrum, which is difficult for a normal pilot to control the helicopter due to its instability. Additionally, for MIMO state space model identification, the conversion of time domain measurement data to frequency domain data requires a considerable amount of data used for removing the contaminating effects of partially correlated control inputs from the extracted frequency responses [8].

In the time domain system identification, a multivariable output-error state space subspace-based system identification [8] was used to produce a model that described the input-output relationship of the original system model, where the identified model is a "black-box" structure with a reliable order. Morris et al. [9] used PEM to identify a small scale helicopter placed on a 3DOF stand, and the deriving model is used for the LQG controller design at hover. Cai et al. [10] applied the method PEM to identify the model of the yaw channel of a helicopter which was represented by a black-box state space model. Schafroth et al. [11] identified four decoupled linear subsystems of the muFly micro helicopter with PEM. By applying the least squares method, six SISO models were derived to identify the helicopter's physical parameters in [12]. Raptis et al. [13, 14] used the recursive least squares algorithm to identify the model of a small unmanned helicopter which was described by a traditional 6DOF rigid body nonlinear model. Chen et al. [15] identified a model helicopter's yaw dynamics by using the least squares method with the weighted factor allocated by a single neuron. Bottasso et al. [16] presented recursive and batch procedures to estimate the nonlinear model of a rotorcraft which would be used to formulate trajectory optimization problems.

It can be found that the above related works mostly use local optimization algorithms based on gradient search methods to find a local minimum of the cost function. Recently, the evolutionary computations technique based global optimization algorithms are taken into consideration. Especially, the genetic algorithm (GA) is used to identify the helicopter model. The linear models for the dynamics of a helicopter in level flight with constant speed were identified with the combination of GA and Levenberg–Marquardt optimization algorithms [17]. By using GA method, two decoupled linear helicopter models in hover flight were identified in [18], and Del et al. [19] identified the parameters of the hybrid analytic-empiric model of a helicopter. However, while applying GA to solve large-scale and complex parameters identification problems, the flaw that the premature convergence will make it stuck at a local optimum.

For overcoming the premature convergence of GA, Yan [21] combined the GA and chaotic variable to search the optimization of the operational conditions based on RBF-PLS model. Although the method had taken the ergodicity and irregularity of the chaotic variable [21, 22] to make the individuals of sub-generations distributed ergodically in the defined space to avoid from the premature phenomenon, it did not effectively combine the spatial search advantage of GA and chaotic variable. In this paper, in order to get the global optimal solution of the state space helicopter dynamic model in hover or low-speed flight conditions, a chaos-genetic algorithm which integrates GA with chaotic optimization operation was applied. The novel CGA applied here adopts both the chaotic mapping of each generation of population and the chaotic optimization of each generation of outstanding individuals so that the population diversity will be improved to avoid premature convergence. The mechanism of the GA is still kept so that the convergence characteristic of GA will overcome the randomness of the chaotic process. Based on the above properties, the novel CGA will find the global optimal solution in great probability. The identification was implemented on a Raptor-90 RC model helicopter with a set of avionics system. Three different identification methods CGA, TGA and PEM [20] were used to derive the helicopter models, and the comparison of their performance based on the verification results illustrated the advantages of the method CGA.

The remainder of the paper is organized as follows. In Section 2, the nonlinear and linear helicopter models including the rigid body dynamics, the rotor /stabilizer-bar dynamics and the artificial yaw damping system are presented. In Section 3, the chaos-genetic algorithm is developed to estimate the parameters of the linear helicopter model. The flight platforms, experiment setup, identified results and automatic flight controlled by an LQG regulator based on the identified model are presented in Section 4. The concluding remarks are finally provided in Section 5.

2 Dynamics of a Small Unmanned Helicopter

The dynamic behavior of a small helicopter is very complex with strong inter-axis and fuselage-rotor coupling as well as inherent nonlinearities. Nonetheless, it is suitable to treat the overall helicopter's dynamics as a nonlinear MIMO state space model, and the derivation of the equation of motion by treating the helicopter as a rigid body with 12 state space variables required can form the basic model [2]. In addition, for the small model helicopter used in this paper, there are two main characteristics need to be considered comparing with the normal size helicopter. Firstly, a stabilizer-bar mounted on the main rotor is used to improve the stability of the pitching and rolling movements. Secondly, the model helicopter stabilize the yaw axes with another electronic gyro applied on the tail rotor. In the following, the model consisting of rigid body dynamics, the rotor/stabilizer-bar dynamics and the artificial yaw damping system will be used for the system identification of the helicopter in hover or low-speed flight conditions.

2.1 Rigid Body Dynamics

The standard rigid body dynamical equations derived from the Newton's second law are used to model the motion of the helicopter as:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m}\mathbf{F} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{1}$$

and

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{I}^{-1}\left(\mathbf{M} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I}\begin{bmatrix} p \\ q \\ r \end{bmatrix}\right) \tag{2}$$

Where **F** and **M** denote the external forces and moments acting on the center of gravity of the helicopter respectively, m is the mass, **I** is the rotational inertial matrix of the helicopter with respect to the body-fixed reference frame, $[u, v, w]^T$ and $[p, q, r]^T$ are the translational and rotational velocities in the body-fixed coordinate frame respectively as shown in Fig. 1. In this study, the external forces **F** and the moments **M** are the sum of the forces and moments due to the main rotor, tail rotor, gravity and the fuselage aerodynamic forces. The details of them can be found in [2, 13, 14]. While the small helicopter works in hover or low-speed conditions, the force and moment caused by the fuselage can be ignored [2]. Considering the pilot's inputs, there are four actuators used to change the pitch angles of the main and tail rotor blades to generate forces and

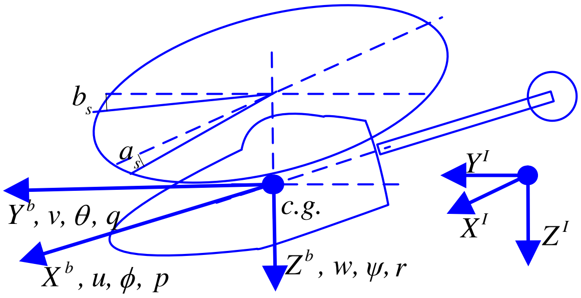


Fig. 1 The states in the body-fixed coordinates of the helicopter

moments on the helicopter. So the changes of **F** and **M** are mainly caused by the tuning of the actuators. In this paper, the control input to the four actuators is defined as $\mathbf{u} = [u_{\text{lat}}, u_{\text{lon}}, u_m, u_t]^T$, where u_m and u_t are collective controls of the main and tail rotor respectively, u_{lat} and u_{lon} are the cyclic controls for pitching and rolling correspondingly.

The rotation matrix $\mathbf{R}_{I2B}$ transforming the inertial coordinates into the body-fixed coordinates using yaw-pitch-roll Euler angles is given as:

$$\mathbf{R}_{I2B} = \begin{bmatrix} c\psi c\theta & s\psi c\theta & -s\theta \\ c\psi s\theta s\phi - s\psi c\phi & s\psi s\theta s\phi + c\psi c\phi & c\theta s\phi \\ c\psi s\theta c\phi + s\psi s\phi & s\psi s\theta c\phi - c\psi s\phi & c\theta c\phi \end{bmatrix} \tag{3}$$

Here, $c\alpha$ and $s\alpha$ are used as the abbreviations for $\cos(\alpha)$ and $\sin(\alpha)$. The variables ϕ, θ and ψ represent the roll, pitch and yaw angles of the helicopter in the body-fixed coordinates. The differential equation relating $[p, q, r]^T$ with the Euler angles $[\phi, \theta, \psi]^T$ is:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \frac{s\phi s\theta}{c\theta} & \frac{c\phi s\theta}{c\theta} \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4}$$

The derivative of position with respect to the earth coordinates is:

$$\dot{\mathbf{p}} = \mathbf{R}_{I2B}^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{5}$$

Where $\mathbf{p} = [p_x, p_y, p_z]^T$ is the position vector of the center of gravity of the helicopter in the earth coordinates. The combination of Eqs. 1–5 shows the dynamical equations of the rigid body.

2.2 Rotor/Stabilizer-Bar Dynamics

The main rotor is the principal source of control forces and moments along the pitching and rolling axes. In real flight, since the small helicopters have faster dynamics compared to normal scale ones, the stabilizer bar is used to act as a lagged-rate feedback system to help them fly. As a consequence, additional dynamics considering the stabilizer bar effects need to be added to the whole dynamics of the helicopter. According to [4], the rotor/stabilizer-bar dynamics can be accurately approximated by a first-order system. In this paper, the additional dynamics reads as follows:

$$\begin{bmatrix} \dot{a}_s \\ \dot{b}_s \end{bmatrix} = \begin{bmatrix} -a_s/\tau_s + k_1 p - q + A_{\text{lat}} u_{\text{lat}} \\ -b_s/\tau_s - k_1 q - p + B_{\text{lon}} u_{\text{lon}} \end{bmatrix} \tag{6}$$

Where a_s and b_s represent the dynamics of the flapping angles that influence the pitching and rolling motion respectively. τ_s is a time constant considering the effects of the stabilizer bar. k_1, A_{lat}, B_{lon} are just gains.

2.3 Artificial Yaw Damping System

In the real control of a small helicopter, an artificial yaw damping system with a yaw-rate gyro is used to attenuate the effect of the anti-torque fluctuation on the yaw response, which makes the pilot control the vehicle easily. The introduced yaw rate feedback can be regarded as a simple first order system [3]:

$$\frac{r_{rfb}}{r} = \frac{k_r}{s + k_{rfb}} \tag{7}$$

Here, r_{rfb} is the feedback gyro system state. k_r and k_{rfb} are parameters to be identified. s is the Laplace operator in the frequency domain.

2.4 Nonlinear and Linear State Space Model

With the combination of the rigid body motion shown as Eqs. 1–5, the dynamics of the rotor/stabilizer bar shown as Eq. 6 and the artificial yaw damping system shown as Eq. 7, a nonlinear helicopter system is defined as:

$$\dot{\mathbf{x}}_{\text{non}} = \mathbf{f}(\mathbf{x}_{\text{non}}, \mathbf{u}) \tag{8}$$

with $\mathbf{x}_{\text{non}} = [p_x, p_y, p_z, u, v, w, \theta, \phi, \psi, p, q, r, r_{rfb}, a_s, b_s]^T$, and $\mathbf{u} = [u_{\text{lat}}, u_{\text{lon}}, u_m, u_t]^T$.

Due to the complexity of the external forces and moments acting on the helicopter, especially the situation that the relative aerodynamic forces are hard to be described totally and explicitly, it is difficult to model the nonlinear helicopter system fully. Usually, a linear model derived from the nonlinear model by using the small perturbation theory is able to capture the main characteristics of the helicopter dynamic behavior near the trim condition, such as hover or low-speed flight. Furthermore, a linear model is often used for the design of helicopter flight control system. For a small helicopter working in hover or low-speed flight regimes, it has the true that $\dot{\psi} \approx r$ and ψ has little couple relationship with other states in the vector $\mathbf{x}_{\text{non}}$. Considering the Eq. 8, it gets that the states p_x, p_y, p_z and ψ do not influence the dynamics of p, q and r. So while linearizing the nonlinear Eq. 8, the number of state variables is reduced to 11.

By applying the small perturbation theory, the linear model of the helicopter can be derived. In addition, there is little coupling between the horizontal and vertical dynamics while helicopter works in hover or low-speed flight conditions, the system can be splitted into two subsystems for horizontal and vertical motions. and they are derived as shown in Eqs. 9 and 10.

$$\delta\dot{\mathbf{x}}_{\text{hor}} = \mathbf{A}_{\text{hor}}\delta\mathbf{x}_{\text{hor}} + \mathbf{B}_{\text{hor}}\delta\mathbf{u}_{\text{hor}} \tag{9}$$

$$\delta\dot{\mathbf{x}}_{\text{ver}} = \mathbf{A}_{\text{ver}}\delta\mathbf{x}_{\text{ver}} + \mathbf{B}_{\text{ver}}\delta\mathbf{u}_{\text{ver}} \tag{10}$$

Where δ denotes the perturbation from the trim condition, and

$$\begin{gathered}
\mathbf{x}_{\text{hor}} = [u, v, \theta, \phi, q, p, a_s, b_s]^T, \quad \mathbf{u}_{\text{hor}} = [u_{\text{lat}}, u_{\text{lon}}]^T, \\
\mathbf{x}_{\text{ver}} = [w, r, r_{rfb}]^T, \qquad \mathbf{u}_{\text{ver}} = [u_m, u_t]^T, \\
\mathbf{A}_{\text{hor}} = \begin{bmatrix} X_u & 0 & -g & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ M_u & M_v & 0 & 0 & M_q & 0 & M_a & M_b \\ L_u & L_v & 0 & 0 & 0 & L_p & L_a & L_b \\ 0 & 0 & 0 & 0 & -1 & k_1 & -1/\tau_s & 0 \\ 0 & 0 & 0 & 0 & -k_1 & -1 & 0 & -1/\tau_s \end{bmatrix}, \\
\mathbf{B}_{\text{hor}} = \begin{bmatrix} X_{\text{lat}} & 0 \\ 0 & Y_{\text{lon}} \\ 0 & 0 \\ 0 & 0 \\ M_{\text{lat}} & M_{\text{lon}} \\ L_{\text{lat}} & L_{\text{lon}} \\ A_{\text{lat}} & 0 \\ 0 & B_{\text{lon}} \end{bmatrix}, \\
\mathbf{A}_{\text{hor}} = \begin{bmatrix} Z_w & Z_r & 0 \\ N_w & N_r & N_{rfb} \\ 0 & k_r & -k_{rfb} \end{bmatrix}, \quad \mathbf{B}_{\text{hor}} = \begin{bmatrix} Z_m & Z_t \\ N_m & N_t \\ 0 & 0 \end{bmatrix}.
\end{gathered} \tag{11}$$

As shown above, there are thirty-three unknown parameters in the system matrixes and control matrixes. Some parameters can not be directly identified from the input-output dada. The assumptions [4] that $N_{rfb} = -N_t$ and $k_{rfb} = -2N_r$ were adopted in this paper. Consequently, there are 22 and 9 parameters to be identified in the linear models for the horizontal and vertical dynamics respectively.

3 Chaos-Genetic Algorithm for the Continuous LTI Helicopter Model Identification

3.1 One-Dimensional Logistic Map

In this paper, a well-known one-dimensional logistic map [21] was used to modify the traditional genetic algorithm which would be applied to the continuous LTI helicopter model identification. The one-dimensional logistic map equation is defined as:

$$x_{k+1} = \mu x_k(1 - x_k) \tag{12}$$

Where μ is a control parameter, $k = 1, 2 \cdots, n$, x_k is the value of variable x at the k_{th} iteration, $0 \le x_1 \le 1$. Suppose that $0 < \mu \le 4$, it can derive that the series $\{x_k\}$ is bounded with $0 \le x_k \le 1$. The behavior of the system (Eq. 12) is greatly changed with the variation of μ. When $\mu = 4$ and $x_1 \in (0, 1) - \{0.25, 0.5, 0.75\}$, the system exhibits chaotic dynamics that very small change in the initial value of x would cause very large difference in its long-term behavior. In this case, the variable x is called as chaotic variable. As n increases, the track of chaotic variable can travel ergodically over the whole search space, and it will never repeat a value having appeared already. Figure 2 shows the chaotic dynamics of the logistic map, where $n = 500$, $\mu = 4$, $x_1 = 0.2$. It indicates the three other main properties of the chaotic variable, i.e., ergodicity, pseudo-randomness and irregularity [22].

3.2 Chaotic Optimization Operation

Considering an optimization problem of searching maximum described as:

$$\max_{x_i \in (a_i, b_i)} f(x_1, x_2, \cdots, x_N), i = 1, \cdots, N \tag{13}$$

Let $\mathbf{x} = [x_1, x_2, \cdots, x_N]^T$ be the vector consisting of N parameters to be optimized, then f is an objective function with respect to the parameter vector $\mathbf{x}$. Define the restricted space S_r and the normalized chaotic space S_c as $\{(x_1, x_2, \cdots, x_N)|a_i < x_i < b_i, i = 1, 2, \cdots, N\}$ and $\{(chx_1, chx_2, \cdots, chx_N)|0 < chx_i < 1, i=1, 2, \cdots, N\}$ respectively. The evolution process of the chaotic optimization operator is defined as follows:

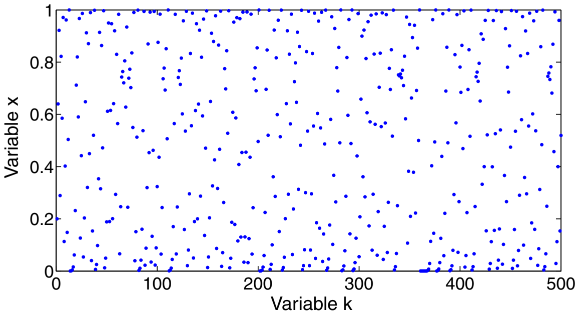


Fig. 2 The dynamics of the one-dimensional logistic map with $n = 500$, $\mu = 4$ and $x_1 = 0.2$

Step 1 Introduce the initial vector $\mathbf{x}^0$ for the objective function f and add little random perturbation to every parameter in $\mathbf{x}^0$, and then a new vector $\mathbf{x}^1$ is derived with $x_j^1 = (1 + \Delta \times (0.5 - rand_j))x_j^0$, $j = 1, 2, \cdots, N$. Here, Δ is a given positive constant, $rand_j$ is a random number in the range (0, 1).

Step 2 Adopt the one-dimensional logistic map with $\mu = 4$:

$$chx_i^{k+1} = 4chx_i^k(1 - chx_i^k), i = 1, \cdots, N \tag{14}$$

Where chx_i^k denotes the i_{th} chaotic variable after k_{th} iteration.

Step 3 Map the parameters $x_1^1, x_2^1, \cdots, x_N^1$ from the restricted space S_r to the normalized chaotic space S_c:

$$chx_i^1 = \frac{x_i^1 - a_i}{b_i - a_i}, i = 1, \cdots, N \tag{15}$$

Step 4 Calculate the second iteration chaotic variables chx_i^2 by applying Eq. 14.

Step 5 Calculate the parameters x_i^2 by back-mapping chx_i^2 from S_c to S_r as:

$$x_i^2 = a_i + chx_i^2(b_i - a_i), i = 1, \cdots, N \tag{16}$$

Therefore, all parameters of the k_{th} iteration $\mathbf{x}^k$ will be chaotically mapped forward shown as Eq. 15 and backward shown as Eq. 16 to produce the $(k + 1)_{th}$ iteration $\mathbf{x}^{k+1}$.

In this paper, the maximum iterations $k_{\max}$ is used to indicate the termination criterion for the chaotic optimization operation. During the above process including the initialization and every iteration, the objective function f with respect to $\mathbf{x}^k$, $k = 0, 1, \cdots, k_{\max}$ is calculated immediately.

Finally, the solution to the problem Eq. 13 would be selected from $\mathbf{x}^k, k = 0, 1, \cdots, k_{\max}$, and it is denoted as $\mathbf{x}^{\text{opt}}$.

The above procedure as a whole is named as chaotic optimization operator for maximum represented by *choo* as:

$$\mathbf{x}^{\text{opt}} = choo(\mathbf{x}^0, k_{\max}), s.t. S_r \tag{17}$$

3.3 Chaos-Genetic Algorithm

In this paper, the real-coded chaos-genetic algorithm is used for the identification of the LTI helicopter model, and it combines the traditional GA operators and the chaotic optimization operator. As an important part of the genetic algorithm, the fitness function should be defined firstly. Since the LTI model of the small helicopter is divided into two decoupled parts, i.e., the horizontal model and the vertical model, the parameters in Eqs. 9 and 10 can be identified separately. The cost function related to the error between the in-flight measured responses and the relative simulated responses obtained from the identified model with the same measured inputs has the following forms:

$$f(\Theta) = \sum_{i=1}^{M} [\mathbf{y}_i - \mathbf{y}_{\text{sim}}(\delta\mathbf{u}_i, \Theta)]^T \mathbf{M}_{\text{cov}}^{-1} [\mathbf{y}_i - \mathbf{y}_{\text{sim}}(\delta\mathbf{u}_i, \Theta)] \tag{18}$$

Where Θ is the vector of parameters with N elements to be identified, $\mathbf{y}$ is the vector of observations from the flight test, $\mathbf{y}_{\text{sim}}$ is the simulated outputs from the proposed model with the parameter vector Θ. M is the simulation length. $\mathbf{M}_{\text{cov}}$ is the covariance matrix given as:

$$\mathbf{M}_{\text{cov}} = \frac{1}{M} \sum_{i=1}^{M} [\mathbf{y}_i - \mathbf{y}_{\text{sim}}(\delta\mathbf{u}_i, \Theta)][\mathbf{y}_i - \mathbf{y}_{\text{sim}}(\delta\mathbf{u}_i, \Theta)]^T \tag{19}$$

In order to simplify the calculation, a diagonal matrix considering different weights to each state

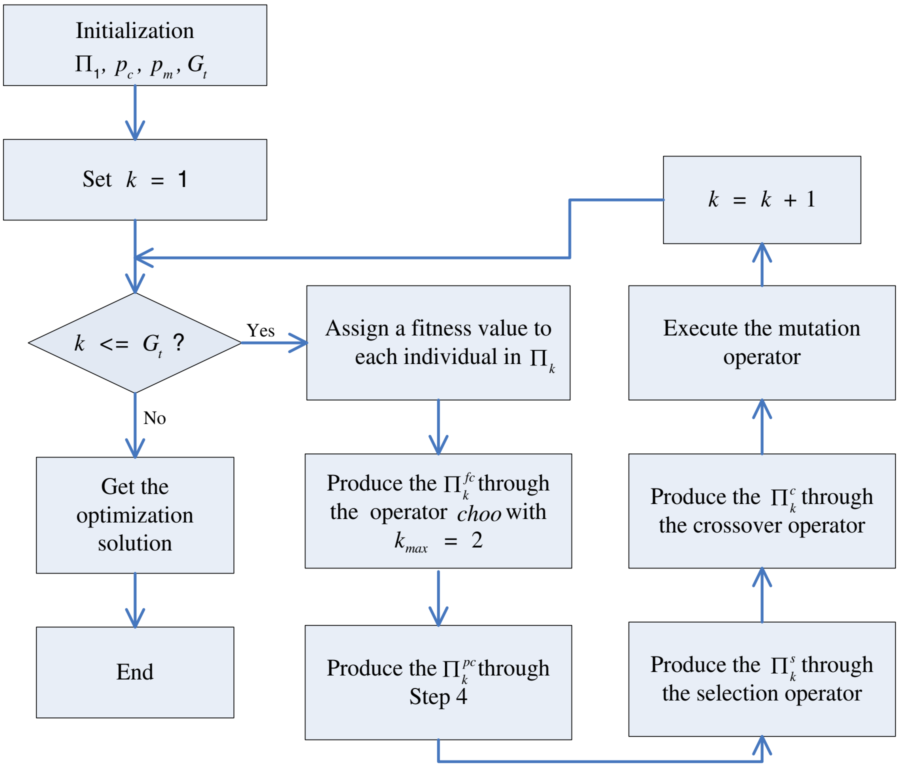

Initialization
Π_1, p_c, p_m, G_t
Set $k = 1$
$k = k + 1$
$k <= G_t$?
Yes
No
Assign a fitness value to each individual in Π_k
Execute the mutation operator
Get the optimization solution
Produce the Π_k^{fc} through the operator *choo* with $k_{max} = 2$
Produce the Π_k^{c} through the crossover operator
End
Produce the Π_k^{pc} through Step 4
Produce the Π_k^{s} through the selection operator

Fig. 3 The flow chart of the chaos-genetic algorithm

is applied to replace the matrix $\mathbf{M}_{\text{cov}}$. And the fitness function can be designed as:

$$F(\Theta) = \frac{1}{f(\Theta) + 10^{-5}} \tag{20}$$

It indicates that the larger value $F(\Theta)$ is, the more possible the parameters will match the dynamic model of the helicopter. The detailed procedure of executing the chaos-genetic algorithm is described as follows (Fig. 3).

Step 1 Initialization. The CGA optimization process begins by determining the crossover probability p_c, the mutation probability p_m and the maximum number of generation m_t. The initial population Π_1 is created with n_p individuals $\Theta_{1,1}, \cdots, \Theta_{n_p,1}$ which are randomly produced from the parameters restricted space S_r.

Step 2 Calculate the fitness function. Assign a fitness value to each individual in the k_{th} generation population Π_k by using Eq. 20. The k_{th} generation population is denoted as $\Pi_k = \{\Theta_{1,k}, \Theta_{2,k}, \cdots, \Theta_{n_p,k}\}$, where $\Theta_{i,k} = [\vartheta_{i,k,1}, \cdots, \vartheta_{i,k,N}], i = 1, \cdots, n_p$.

Step 3 Execute the chaotic optimization operator *choo* over the full population Π_k with $k_{\max} = 2$, and it derived that

$$\Theta^{\text{fc}}_{i,k} = choo(\Theta_{i,k}, 2), s.t.S_r, i = 1, \cdots, n_p \tag{21}$$

The resulting population is denoted as $\Pi^{\text{fc}}_k = \{\Theta^{\text{fc}}_{1,k}, \Theta^{\text{fc}}_{2,k}, \cdots, \Theta^{\text{fc}}_{n_p,k}\}$.

Step 4 Rank the individuals of the population Π_k in descending order of fitness, and then select the top two percent of the sorted individuals. For every selected individual $\Theta^{\text{fc}}_{\text{select}_i,k}$(select$_i$ indicates the i_{th} selected individual), the corresponding sub-restricted space is defined as:

$$S^{\text{fc}}_{r,\text{select}_i,k} = \left\{ \Theta_{\text{select}_i,k} \;\middle|\; \begin{array}{l} |\vartheta_{\text{select}_i,k,j} - \vartheta^{\text{fc}}_{\text{select}_i,k,j}| < 0.2 \times |\vartheta^{\text{fc}}_{\text{select}_i,k,j}| \\ and, a_j < \vartheta_{\text{select}_i,k,j} < b_j, j = 1, \cdots, N \end{array} \right\} \tag{22}$$

And then evolve every selected individuals by executing the procedure *choo* with $k_{\max} = 500$ in the sub-restricted space $S^{\text{fc}}_{r,\text{select}_i,k}$, the optimal individual can be obtained as:

$$\Theta^{\text{fc,opt}}_{\text{select}_i,k} = choo(\Theta^{\text{fc}}_{\text{select}_i,k}, 500), \quad s.t.S^{\text{fc}}_{r,\text{select}_i,k}, i = 1, \cdots, 2n_p/100 \tag{23}$$

At the end of this step, the resulting individuals $\Theta^{\text{fc,opt}}_{\text{select}_i,k}, i = 1, \cdots, 2n_p/100$ together with the remaining unselected individuals in Π^{fc}_k make up a new population as $\Pi^{pc}_k = \{\Theta^{pc}_{1,k}, \Theta^{pc}_{2,k}, \cdots, \Theta^{pc}_{n_p,k}\}$. The above two steps both increase the diversity of the population in the evolution process. With the third step operated, the population will have the possibility to introduce new possible individuals from the whole search space. And the executing of the fourth step is based on the consideration that there may be better individuals in the sub-restricted space which consist of the individuals near the top two percent of individuals. As the individual number for the operation *choo* in the third step is larger, while that for the fourth step is smaller, the iterations $k_{\max}$ is set as 2 and 500 for the third and fourth step respectively.

Step 5 Reproduction. Select the individuals from the population Π^{pc}_k according to their fitness. In this paper, the deterministic sampling which can guarantee that fit individuals will be copied into the mating pool is used for reproduction, and the resulting population in the mating pool is denoted as $\Pi_k^s = \{\Theta_{1,k}^s, \Theta_{2,k}^s, \cdots, \Theta_{n_p,k}^s\}$.

Step 6 Crossover. This link imitates the process of biological evolution that a new individual produced by the recombination of two parent chromosomes. For real-coded CGA, the arithmetic crossover operation is used to preserve the constraint as:

$$\begin{cases} \widetilde{\Theta}_{i,k} = \alpha_{i,j,k}\Theta_{i,k}^s + (1-\alpha_{i,j,k})\Theta_{j,k}^s \\ \qquad i, j = 1, 2, \cdots, n_p \\ \widetilde{\Theta}_{j,k} = \alpha_{i,j,k}\Theta_{j,k}^s + (1-\alpha_{i,j,k})\Theta_{i,k}^s \\ \qquad k = 1, 2, \cdots, G_t \end{cases} \tag{24}$$

where $\alpha_{i,j,k}$ is a random number assigned by a value in the range (0, 1). Before executing the crossover, the individuals in Π_k^s are paired randomly with $n_p/2$ pairs. For each pair, the arithmetic crossover operation is executed only when the crossover probability p_c is larger than a random number in the range (0, 1). The population after crossover is denoted as $\Pi_k^c = \{\Theta_{1,k}^c, \Theta_{2,k}^c, \cdots, \Theta_{n_p,k}^c\}$.

Step 7 Mutation. As an effective operator to increase and retain the population diversity in the processing of real-coded GA, there are several kinds of mutation operation for choice, such as the uniform mutation operation, the non-uniform mutation operation and the Gaussian mutation operation [23, 24]. The non-uniform mutation has the feature of searching the space uniformly at the early stage and very locally at the later stage [23], and the Gaussian mutation performs better searching in a small local area [24]. As described above, the operation of Steps 3 and 4 contains the properties of non-uniform mutation and the Gaussian mutation. In this paper, the uniform mutation operation is employed, and the $(k+1)_{th}$ generation population Π_{k+1} will be obtained after executing this step.

Step 8 Repeat the Steps 2–7 until the number of generations reaches the allowable maximum number G_t.

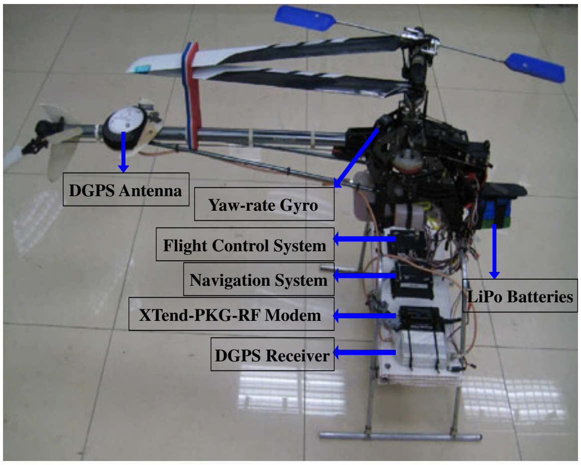


Fig. 4 On board avionics systems mounted on the Raptor-90 RC model helicopter

4 Experiments and Results

The entire experiments were implemented on a Raptor-90 RC model helicopter platform (See Figs. 4 and 5). The helicopter is characterized by a rigid main rotor equipped with a Bell–Hill stabilizer bar and actuated by five high bandwidth digital servos. The avionics used to implement the automatic control is composed by a GPS/INS based navigation system, an embedded flight control system and a data communication system. The navigation system consists of three-axis magnetometers, a static pressure sensor, a DGPS

Fig. 5 Actual flight experiment

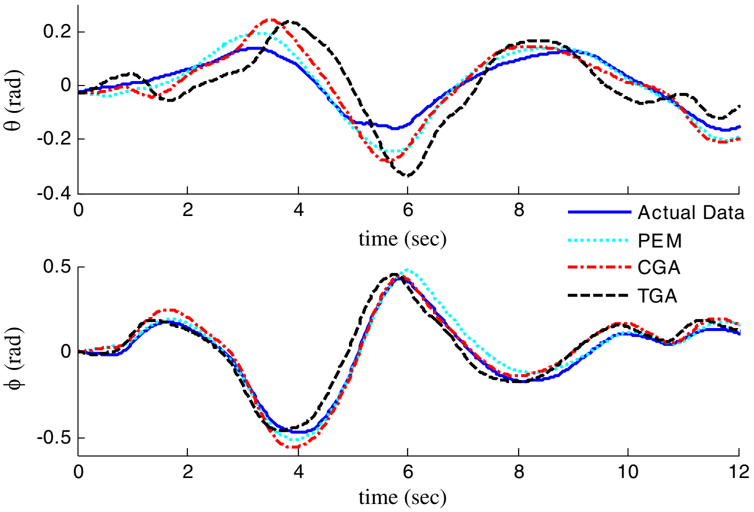


Fig. 6 Comparison of the actual and estimated Euler angles over a validation range. Actual data is solid, estimated data from PEM model, CGA model and TGA model is *dotted*, *dashdotted*, *dashed* respectively

receiver and an IMU unit which integrates three-axis angular rate gyros and accelerometers. An Extended Kalman Filter is used to fuse the values of sensors to get the data of positions, velocities, Euler angles and angular rates. The rotor speed is controlled by a governor. An FPGA-based I/O board was integrated in the flight control system to drive the five onboard servos and to read pilot commands through a standard Futaba transmitter. The avionics systems have an accuracy of 0.2 m RMS for the position, 0.03 m/s RMS for the linear velocity and 0.017 rad RMS for the attitude. During the experiments, all control inputs and vehicle state variables were sampled at 50 Hz and recorded in a 4 GB flash memory on the flight control system.

For identification, firstly, the pilot provided a stabilizing trim command for the helicopter to make it fly in hover. Secondly, small-signal excitations around the trim produced by the pilot

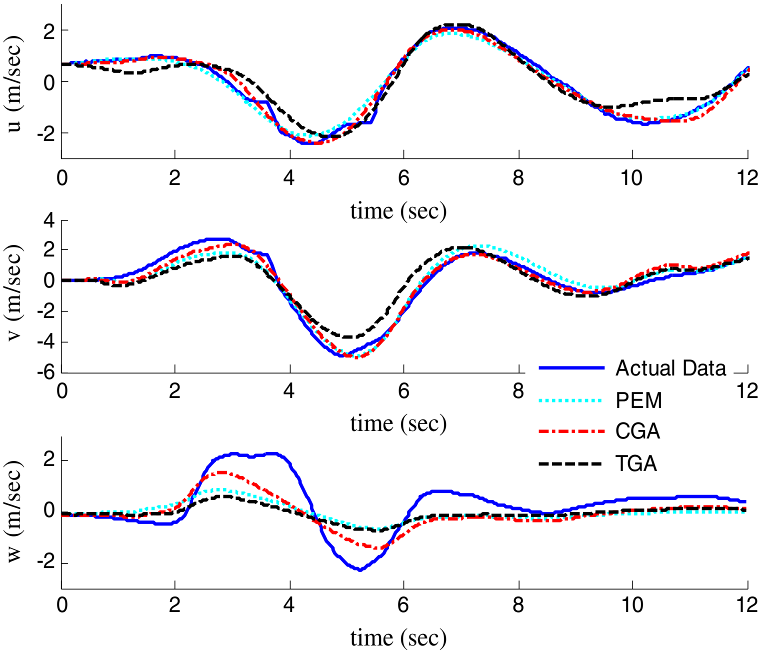


Fig. 7 Comparison of the actual and estimated linear velocities over a validation range. Actual data is solid, estimated data from PEM model, CGA model and TGA model is *dotted*, *dashdotted*, *dashed* respectively

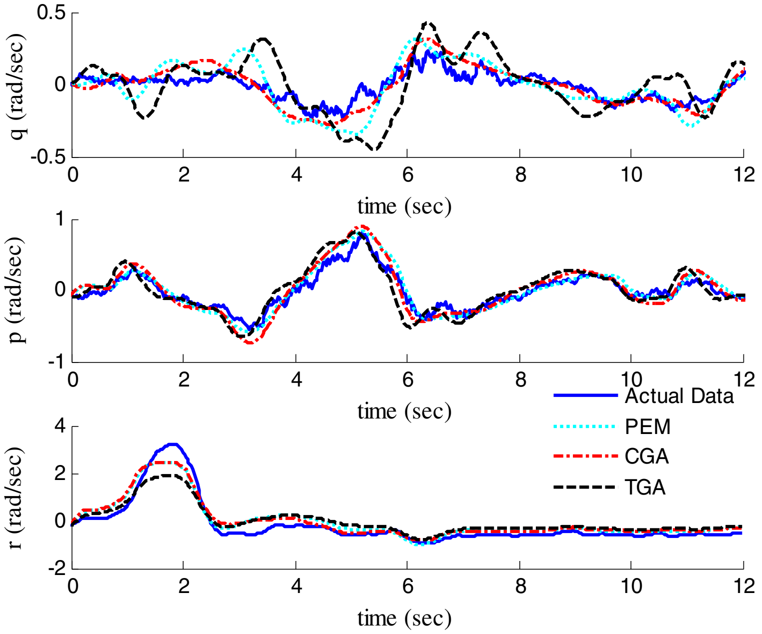


Fig. 8 Comparison of the actual and estimated angular velocities over a validation range. Actual data is solid, estimated data from PEM model, CGA model and TGA model is *dotted*, *dashdotted*, *dashed* respectively

were used to impose on the top of the aileron, elevator, rudder and the collective trim. These control inputs result in the rolling, pitching, yawing and up-and-down motions. The control inputs and the relative raw state variables were sampled and recorded.

Before identification, in order to remove the effects of the trim and the structural vibrations, the experimental data was preprocessed by passing through a ten-point average FIR filter (See Eq. 25, −3 db @2.2 Hz) and executing the trim removal.

$$y_{\text{filter}}(k) = \frac{1}{10}\sum_{i=0}^{9} y(k-i) \tag{25}$$

It is well known that the above FIR filter will result in delay. In order to obtain no phase distortion, the above filter was implemented via the Matlab function 'filtfilt' [25] which processes the input data in both the forward and reverse directions. And the final result is an output sequence with noise attenuation and no phase distortion.

In this paper, for running the genetic algorithm, a population with five hundred individuals is used due to the complexity of the fitness function Eq. 20 with more than twenty parameters to be identified once. The crossover probability and the mutation probability are chosen as 0.9 and 0.1 respectively.

Figures 6, 7 and 8 show the simulation of the identified models over a set of experiment data. The performance comparison of three different identification methods PEM [20], TGA, CGA used to derive the helicopter models is also presented as the three-axis linear velocities, Euler angles and angular velocities illustrated in Figs. 6–8. It shows that these three methods can all get a model that matches the trends of the real experiment data. But it is obviously that the simulation of the model generated by the method CGA approximates the experiment data best.

To verify the precision of the identified model by applying the three different identification

Table 1 The comparison of the performance regarding Eq. 26 by PEM, TGA and CGA

	PEM	TGA	CGA
Forward velocity u	0.8914	0.9559	0.9871
Sideway velocity v	0.9461	0.9363	0.9814
Vertical velocity w	0.7074	0.7937	0.8576
Pitch angle θ	0.9197	0.7315	0.9498
Roll angle ϕ	0.9557	0.9627	0.9877
Pitch angular velocity q	0.8755	0.5867	0.8885
Roll angular velocity p	0.9597	0.9099	0.9689
Yaw angular velocity r	0.9624	0.9584	0.9723

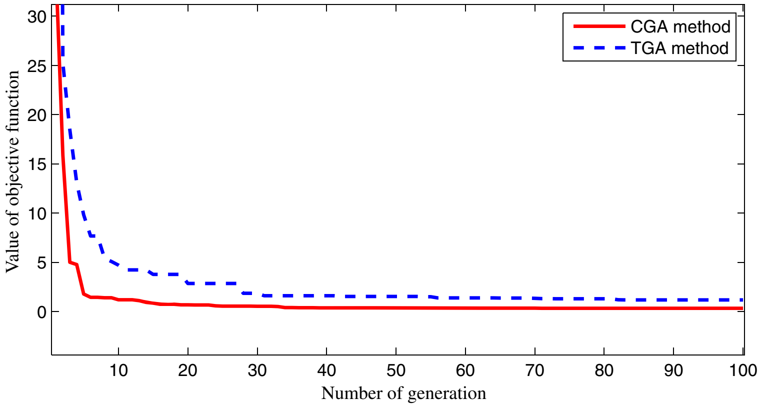


Fig. 9 The evolutionary process regarding Eq. 18 by CGA vs. TGA

methods, the correlation coefficient between the measured ($y_i(t)$) and estimated ($\widehat{y}_i(t)$) data, defined as the normalized cross-covariance function is used to estimate how well the identified model can reproduce the measured data, and the function is defined as:

$$\rho = \frac{\sum_{i=1}^{M}(y_i(t)-\overline{y})(\widehat{y}_i(t)-\overline{\widehat{y}})}{\sqrt{\sum_{i=1}^{M}(y_i(t)-\overline{y})^2\sum_{i=1}^{M}(\widehat{y}_i(t)-\overline{\widehat{y}})^2}} \tag{26}$$

Where $\overline{y} = 1/M\sum_{i=1}^{M} y_i(t)$ and $\overline{\widehat{y}} = 1/M\sum_{i=1}^{M}\widehat{y}_i(t)$. The closer the correlation coefficient is to unity, the better the identified model is. While the coefficient is close to zero, the identified model is poor.

Table 1 compares performance regarding Eq. 26 by PEM, TGA and CGA. The simulated data generated from the model by using the method CGA has the largest correlation coefficient. Since the moment of inertia of the helicopter with respect to the pitching axis is larger than that with respect to the rolling axis, during the actual flight data collection, the amplitudes of Euler angle θ and angular velocity q are smaller than those of Euler angle ϕ and angular velocity p with small-signal excitations provided by the pilot. In this case, the identification result with respect to the pitching axis will be influenced to some extent. However, the simulated data can all track the main trends of the experiment data as shown in Figs. 6–8. Furthermore, as presented in Table 1, the correlation coefficients about the pitching dynamics, i.e., θ, q and u by applying the TGA are 0.7315, 0.5867 and 0.9559 respectively, while those by using the CGA are 0.9498, 0.8885 and 0.9871 respectively. The above indicates that the CGA method can get better identified model with its ability to discover the information contained in the collection data.

Figure 9 shows the evolutionary process regarding Eq. 18 by CGA vs. TGA. It shows that the CGA gets better fitness value than that with

Table 2 The values of the identified parameters

Parameter	Value	Parameter	Value	Parameter	Value
X_u	−0.22682	L_b	81.318	Z_w	−0.2539
Y_v	−0.37992	k_1	0.04521	Z_r	2.0675
M_u	−0.25847	τ_s	0.1438	N_w	−0.2972
M_v	0.2389	X_{lat}	−6.1176	N_r	−7.5162
M_q	0.15592	Y_{lon}	6.82024	N_{rfb}	−30.5372
M_a	−3.7369	M_{lat}	−5.32552	k_r	1.0673
M_b	−35.057	M_{lon}	−7.74152	k_{rfb}	15.0324
L_u	0.95332	L_{lat}	−6.10456	Z_m	−3.2315
L_v	−0.72645	L_{lon}	10.384	Z_t	−6.1224
L_p	−4.8079	A_{lat}	2.64568	N_m	−3.1891
L_a	51.199	B_{lon}	−3.0364	N_t	30.5372

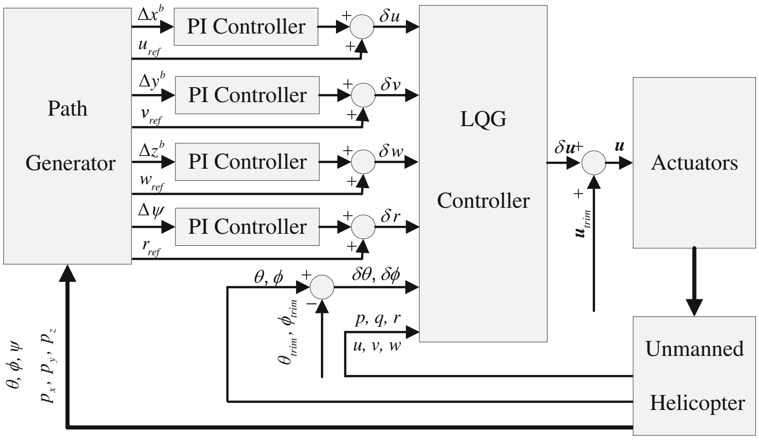


Fig. 10 The control structure of the helicopter

TGA due to the combination of the chaotic optimization operator and GA which enhances the spatial search ability. The chaotic optimization operator inherits the ergodicity and irregularity of the chaotic variable can not only avoid the search being trapped in local optimum, but also make the search have more chance to get the solution in the defined space. All of the above verifies the advantages of the method CGA used for identifying the helicopter model. The identified parameters by applying the method CGA are adopted for controller design, and they are listed in Table 2.

Since there are some unmeasured states (a_s, b_s, r_{rfb}) in the linear models, the LQG technique is useful for controlling this type of model. It is well known that the LQG controller is designed from the combination of linear quadratic regulator and linear quadratic estimator, named as Kalman Filter. According to the separation principle, the optimal controller and optimal estimator can be designed separately based on the linear model. Figure 10 presents the control structure used for implementing autopilot. For tracking the predefined path, the path generator in Fig. 10 generates the reference velocity, the difference of the position and heading. And then four PI controllers (The gains are listed in Table 3) are used to produce the difference of the velocity and the yaw rate. These variables, together with the Euler angles, the angular and linear velocities are then used as the inputs for the LQG controller which will output the control variables to be acted on the actuators. Finally the helicopter is controlled under autopilot.

Figure 11 shows an actual automatic flight of the helicopter with the flight speed equal to 2 m/s, and the predefined path is $A \to B \to C \to D \to E \to F \to G \to H$. The helicopter implemented the automatic take-off from point A to point B, and then tracked the polyline trajectory $B \to C \to D \to E \to F \to G$. When the helicopter arrived at the point G, the helicopter executed the hovering for lasting 20 s. Finally, it descended vertically from the point G.

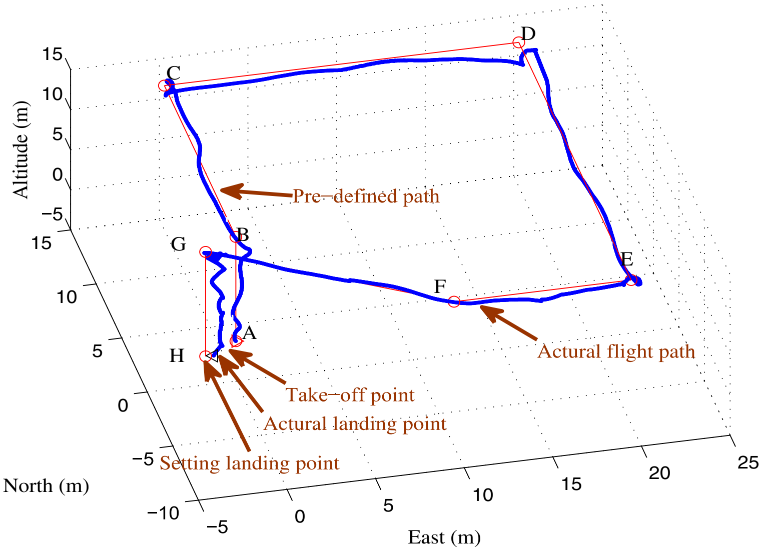
While implementing the automatic take-off from point A, the path generator generated the commands that the reference velocities u_{ref}, v_{ref}, w_{ref} and angular velocity r_{ref} were set as 0 m/s, 0 m/s, −2 m/s and 0 rad/s respectively. Additionally, when the absolute value of the altitude difference (defined as H_{error}) between the helicopter and the point B was larger than one meter, the PI controller for the vertical position control in Fig. 10 did not work with Δz^b set as 0. When H_{error} was less than one meter, the reference velocity w_{ref} was set as 0 m/s, and the PI controller for the vertical position control worked.

For automatic landing from point G, the mechanism how the path generator generated the commands was similar to that for the automatic

Table 3 The gains of PI controllers used for position control

	Δx^b	Δy^b	Δz^b	$\Delta\psi$
P	0.6	0.6	0.1	0.5
I	0.2	0.2	0.1	0.2

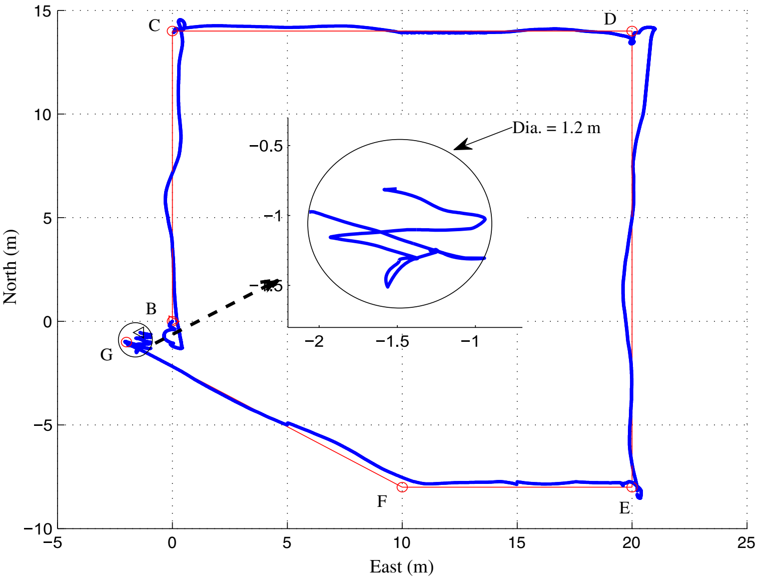
Fig. 11 The real 3D flight test under the control of the LQG regulator



take-off except that the direction of the reference velocity w_{ref} was opposite.

Figure 12 presents the horizontal trajectory of the fight path $B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G$. While the helicopter was switching the waypoint, it can be found that there were some overshoots and oscillations around the points C, D and E. That is because while the helicopter had reached these points, the reference velocities u_{ref} and v_{ref} were changed. In the future work, the normal LQG controller will be modified with feedforward and integral parts added. Also, the path generator will be improved to generate more smooth reference signals for the LQG controller. Nonetheless, the special show in Fig. 12 that the precision during hovering at the point G was within a 1.2 m diameter circle indicates the effectiveness of the controller designed based on the identified model. These further verify the accuracy of the identified model and the effectiveness of the method CGA.

Fig. 12 The *horizontal* trajectory of the real flight test



5 Conclusion and Future Work

Parameter identification of a small unmanned helicopter model in hover or low-speed flight conditions was investigated in this paper. With the flight data collected from actual experiments, a novel identification method CGA which integrates the chaotic optimization operation and TGA was used to identify the parameters in the two decoupled linear models. Furthermore, other two identification methods PEM and TGA were also used to identify the parameters, and the simulation of the identified models generated by using the three methods was compared with the actual experiment data. The comparison indicates that the model generated by applying the method CGA matched the actual flight data better than the other two. Additionally, the evolutionary process of CGA and TGA showed that the CGA can get better fitness value than that from TGA. Furthermore, an LQG controller designed based on the identified model was used to stabilize the helicopter. Actual automatic flight experiment shows that the helicopter regulated by the LQG controller can realize the automatic takeoff, landing, low-speed predefined path tracking and hovering within a 1.2 m diameter circle. The good consistency between experimental data and simulation data demonstrated the accuracy of the identified model and the adequacy of the identification method CGA.

Since the flight speed of the small unmanned helicopter can reach 20 m/s or more, the future work will focus on the modeling and identification of the helicopter in different flight speeds. Since the flight environment is full of disturbance, such as wind, the robust control algorithm and intelligent path tracking algorithm will also be considered to stabilize the inner loop of the helicopter and guide the helicopter to follow the predefined complex path respectively.

Acknowledgements This work owes its existence to the funding from the National High Technology Research and Development Program of China (NO.2011AA040202). The authors are grateful to the researcher Xusheng Lei for his valuable insights throughout suggestions on this project, Qingru Zeng, Chenghao Xue and Chengyin Xie for their assistance during the outdoor experiments from the hot summer to the cold winter.

References

1. Valavanis, K.P.: Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy. Springer, Dordrecht, Netherlands (2007)
2. Mettler, B.: Identification Modeling and Characteristics of Miniature Rotorcraft. Kluwer Academic Pubilshers, Norwell, MA, USA (2003)
3. Mettler, B., Kanade, T., Tischler, M.B.: System Identification Modeling of a Model-Scale Helicopter. Technical report CMU-RI-TR-00-03, Robotics Institute, Carnegie Melon University (2000)
4. Gavrilets, V., Mettler, B., Feron, E.: Nonlinear Model for a Small-Size Acrobatic Helicopter. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada (2001)
5. Tischler, M.B., Remple, R.K.: Aircraft and Rotorcraft System Identification: Engineering Methods with Flight-Test Examples. American Institute of Aeronautics and Astronautics, Virginia, USA (2006)
6. Adiprawita, W., Ahmad, A.S., Sembiring, J.: Automated flight test and system identification for rotary wing small aerial platform using frequency responses analysis. J. Bionic Eng. **4**(4), 237–244 (2007)
7. Song, D.L., Qi, J.T., Dai, L., Han, J.D., Liu, G.J.: Modeling a small-size unmanned helicopter using optimal estimation in the frequency domain. In: 15th International Conference on Mechatronics and Machine Vision in Practice, Auckland, New Zealand, pp. 97–102 (2008)
8. Li, P., Postlethwaite, I., Turner, M.: Subspace-based system identification for helicopter dynamic modeling. In: American Helicopter Society 63rd Annual Forum, Virginia Beach, VA, USA, pp. 2165–2176 (2007)
9. Morris, J.C., Nieuwstadt, M.V., Bendotti, P.: Identification and control of a model helicopter in hover. In: Proceedings of the American Control Conference, Baltimora, Maryland, pp. 1238–1242 (1994)
10. Cai, G.W., Chen, B.M., Peng, K.M., Dong, M.B., Lee, T.H.: Modeling and control of the yaw channel of a UAV helicopter. IEEE Trans. Ind. Electron. **55**(9), 3426–3434 (2008)
11. Schafroth, D., Bermes, C., Bouabdallah, S., Siegwart, R.: Modeling and intification of the muFly micro helicopter. J. Intell. Robot. Syst. **57**(1–4), 27–47 (2010)
12. Kim, S.K., Tilbury, D.M.: Mathematical modeling and experimental identification of a model helicopter. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Boston, MA, pp. 203–213 (1998)
13. Raptis, I.A., Valavanis, K.P., Kandel, A., Moreno, W.A.: System identification and discrete nonlinear control of miniature helicopters using backstepping. J. Intell. Robot. Syst. **55**(2–3), 223–243 (2009)
14. Raptis, I.A., Valavanis, K.P., Kandel, A., Moreno, W.A.: System identification for a miniature helicopter at hover using fuzzy models. J. Intell. Robot. Syst. **56**(3), 345–362 (2009)
15. Chen, H.S., Chen, D.R.: Identification of a model helicopter's yaw dynamics. J. Dyn. Syst. Meas. Control **127**, 140–145 (2005)

16. Bottasso, C.L., Luraghi, F., Maffezzoli, A., Maisano, G.: Parameter estimation of multibody models of unstable systems from experimental data, with application to rotorcraft vehicles. J. Comput. Nonlinear Dyn. **5**, 88–97 (2010)
17. Cruz, R.V., Goes, C.S.: Results of short-period helicopter system identification using output-error and hybrid search-gradient optimization algorithm. Math. Probl. Eng. **2010**, 17 (2010). doi:10.1155/2010/231594
18. Lei, X.S., Du, Y.h.: A linear domain system identification for small unmanned aerial rotorcraft based on adaptive genetic algorithm. J. Bionic Eng. **7**(2), 142–149 (2010)
19. Del, C.J., Valero, J., Barrientos, A.: Identification of a small unmanned helicopter model using genetic algorithms. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3360–3365 (2005)
20. Ljung, L.: System Identification: Theory for the User, 2nd edn. Prentice Hall (1999)
21. Yan, X.F., Chen, D.Z., Hu, S.X.: Chaos-genetic algorithm for optimizing the operating conditions based on RBF-PLS model. Comput. Chem. Eng. **27**, 1393–1404 (2003)
22. Sprott, J.C.: Chaos and Time-Series Analysis. Oxford University Press, Oxford, UK (2003)
23. Zhao, X.C., Gao, X.S., Hu, Z.C.: Evolutionary programming based on non-uniform mutation. Appl. Math. Comput. **192**, 1–11 (2007)
24. Wang, Z.G., Wong, Y.S., Rahman, M.: Development of a parallel optimization method based on genetic simulated annealing algorithm. Parallel Comput. **31**, 839–857 (2005)
25. The MathWorks: Singnal Processing Toolbox, Version 6.6 (2006)