Jestr Journal of Engineering Science and Technology Review 8 (3) (2015) 39-45

JOURNAL OF
Engineering Science and Technology Review

Research Article

www.jestr.org

Nonlinear Control of an Autonomous Quadrotor Unmanned Aerial Vehicle using Backstepping Controller Optimized by Particle Swarm Optimization

Mohd Ariffanan Mohd Basri*, Abdul Rashid Husain and Kumeresan A. Danapalasingam

Department of Control and Mechatronics Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.

Received 13 January 2015; Accepted 25 September 2015

Abstract

Quadrotor unmanned aerial vehicle (UAV) is an unstable nonlinear control system. Therefore, the development of a high performance controller for such a multi-input and multi-output (MIMO) system is important. The backstepping controller (BC) has been successfully applied to control a variety of nonlinear systems. Conventionally, control parameters of a BC are usually chosen arbitrarily. The problems in this method are the adjustment is time demanding and a designer can never tell exactly what are the optimal control parameters should be selected. In this paper, the contribution is focused on an optimal control design for stabilization and trajectory tracking of a quadrotor UAV. Firstly, a dynamic model of the aerial vehicle is mathematically formulated. Then, an optimal backstepping controller (OBC) is proposed. The particle swarm optimization (PSO) algorithm is used to compute control parameters of the OBC. Finally, simulation results of a highly nonlinear quadrotor system are presented to demonstrate the effectiveness of the proposed control method. From the simulation results it is observed that the OBC tuned by PSO provides a high control performance of an autonomous quadrotor UAV.

Keywords: Quadrotor; Nonlinear control; Backstepping control; Particle swarm optimization.

1 Introduction

The quadrotor system is a widely researched control problem. Many classic and modern control techniques have been utilized to stabilize the quadrotor system. In most research works, dynamic properties of the quadrotor are neglected in order to simplify equations of the system [1-6]. For an example, Hamel et al. present a simplified model of the X-4 Flyer in [1]. Both Pound et al. [2] and McKerrow et al. [3] use this model to for quadrotor control. However the simplicity of the model imply that sufficient quadrotor dynamics are not represented for effective control. Castillo et al. [4] apply a linear quadratic regulator (LQR) on a quadrotor platform. In the research, the roll and pitch angles of the quadrotor oscillate considerably, and the helicopter is not able to perform a good hovering motion. Even though the oscillation is reduced over a number of trials, an effective autonomous hover is not produced by the quadrotor. In [5] a proportional, integral and derivative (PID) controller is considered to stabilize a quadrotor helicopter. However, the model of the vehicle is modified in order to simplify controller design.

The backstepping control is a nonlinear control strategy based on the Lyapunov theorem. The backstepping control design techniques have received a great attention because of its systematic and iterative design procedure for nonlinear closed-loop control systems [6-9]. The backstepping approach provides a design instrument for adjustment of nonlinearities. The cancellation of useful nonlinearity problem as in the feedback linearization method also can be avoided. Compared with other methods, backstepping has the advantage of design flexibility through recursive utilization of Lyapunov functions. The key idea of the backstepping design is a recursive selection of some appropriate state variables as virtual inputs for lower dimension subsystems of the overall system. Subsequently, Lyapunov functions are designed for each stable virtual controller [10]. Therefore, the stability of a control system can be guaranteed through the designed control law. Hence, the backstepping controller is used in this study for the quadrotor system.

The backstepping technique has been used to solve the stabilization and trajectory tracking problems of quadrotor helicopter [11-15]. Although the backstepping method can meet the desired robustness of the system, an accurate selection of controller parameters is not easy. Normally the backstepping controller parameters are chosen variously. If the parameters are selected improperly, it can lead to inappropriate responses. Thus, it is vital to choose proper parameters to acquire a good response. Even if a good output response is obtained, there is no formal way to ascertain the optimality of a controller parameter selection.

One of the most widely applied metaheuristic optimization methods is particle swarm optimization (PSO). PSO is a computational method that is based on population optimization algorithm. The method is motivated by the behavior of organisms, such as fish schooling and bird

* E-mail address: ariffanan@fke.utm.my
ISSN: 1791-2377 © 2015 Kavala Institute of Technology. All rights reserved.

flocking [16]. Generally, PSO has features such as a straightforward algorithm, simple to execute, computationally efficient and rapid convergence. Unlike the other metaheuristic techniques, PSO has a flexible and well-balanced mechanism to enhance the global and local exploration abilities [17]. PSO has been extensively applied in off-line tuning of controller parameters, computer science and engineering [18-20]. Thus, due to these advantages, in this work PSO is used to compute the optimal backstepping controller parameter for a quadrotor system. The main contribution of this paper is the design of a backstepping control strategy using PSO algorithm to control a quadrotor UAV.

2 Quadrotor Systems Modeling

In order to develop the model of the quadrotor, reasonable assumptions are established in order to accommodate the controller design. The assumptions are as follows [21]:

Assumption 1: Quadrotor is a rigid body and has symmetric structure.
Assumption 2: Aerodynamic effects can be ignored at low speed.
Assumption 3: The rotor dynamics are relatively fast and thus can be neglected.
Assumption 4: The quadrotor's center of mass and body-fixed frame origin coincides.

2.1 Quadrotor Kinematic Model

Let consider earth fixed frame $\mathrm{E} = \{\mathrm{x_e}, \mathrm{y_e}, \mathrm{z_e}\}$and body fixed frame $B = \{x_b, y_b, z_b\}\mathrm{B} = \{\mathrm{x_b}, \mathrm{y_b}, \mathrm{z_b}\}$, as seen in Figure 1. Let $q = (x, y, z, \phi, \theta, \psi) \in R^6$ be the generalized coordinates for the quadrotor, where (x, y, z) denote the absolute position of the rotorcraft and (ϕ, θ, ψ) are the attitude angles (roll, pitch and yaw) that describe the vehicle orientation. Thus, the model could be defined respectively in translational and rotational subsystems by (1) and (2):

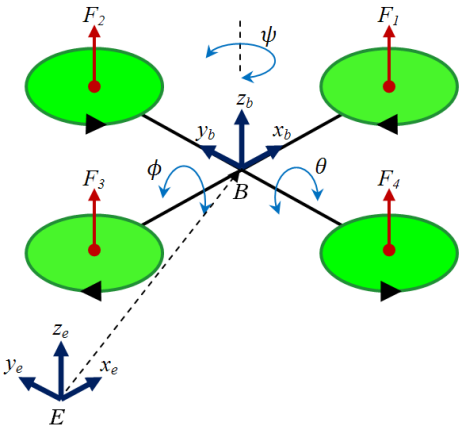


Fig. 1 Quadrotor UAV configuration

$$\xi = (x, y, z) \in R^3 \tag{1}$$

$$\eta = (\phi, \theta, \psi) \in R^3 \tag{2}$$

The kinematic equations of the translational and rotational movements are obtained by means of the rotation R and transfer T matrices respectively. The expression of the rotation R and transfer T matrices can be found in [22] and defined accordingly by (3) and (4):

$$R = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \tag{3}$$

$$T = \begin{pmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{pmatrix} \tag{4}$$

where $s(\cdot)$, $c(\cdot)$ and $t(\cdot)$ are abbreviations for $sin(\cdot)$, $cos(\cdot)$ and $tan(\cdot)$, respectively.

The translational kinematic can be written as:

$$\dot{\xi} = RV \tag{5}$$

where $\dot{\xi}$ and V are respectively the linear velocity vector w.r.t. the earth fixed frame E and body fixed frame B.

The rotational kinematics can be defined as follows:

$$\dot{\eta} = T\omega \tag{6}$$

where $\dot{\eta}$ and ω are the angular velocity vector w.r.t. the earth fixed frame E and body fixed frame B, respectively.

2.2 Quadrotor Dynamic Model

The dynamic model of quadrotor is derived from Newton-Euler approach. It can be useful to express the translational dynamic equations w.r.t. the earth fixed frame E and rotational dynamic equations w.r.t. the body fixed frame B. Therefore, the translational dynamic equations of quadrotor can be written as follows:

$$m\ddot{\xi} = -mge_z + u_T R e_z \tag{7}$$

where m denotes the quadrotor mass, g the gravity acceleration, $e_z = (0,0,1)^T$ the unit vector expressed in the frame E and u_T the total thrust produced by the four rotors.

$$u_T = \sum_{i=1}^{4} F_i = b\sum_{i=1}^{4} \Omega_i^2 \tag{8}$$

where F_i and Ω_i denote respectively, the thrust force and speed of the rotor i and b is the thrust factor.

The rotational dynamic equations of quadrotor can be written as follows:

$$I\dot{\omega} = -\omega \times I\omega - G_a + \tau \tag{9}$$

where I is the inertia matrix, $-\omega \times I\omega$ and G_a are the gyroscopic effect due to rigid body rotation and propeller orientation change respectively, while τ is the control torque obtained by varying the rotor speeds. G_a and τ are defined as:

$$G_a = \sum_{i=1}^{4} J_r(\omega \times e_z)(-1)^{i+1}\Omega_i \tag{10}$$

$$\tau = \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{pmatrix} \tag{11}$$

where J_r is the rotor inertia, l represent the distance from the rotors to the centre of mass and d is the drag factor.

Then, by recalling (7) and (9), the dynamic model of the quadrotor in terms of position (x, y, z) and rotation (ϕ, θ, ψ) is written as:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} + \frac{1}{m} \begin{pmatrix} c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\phi s_\theta s_\psi - s_\phi c_\psi \\ c_\phi c_\theta \end{pmatrix} u_T \tag{12}$$

$$\begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \dot{\theta}\dot{\psi}\left(\frac{I_{yy}-I_{zz}}{I_{xx}}\right) \\ \dot{\phi}\dot{\psi}\left(\frac{I_{zz}-I_{xx}}{I_{yy}}\right) \\ \dot{\theta}\dot{\phi}\left(\frac{I_{xx}-I_{yy}}{I_{zz}}\right) \end{pmatrix} - \begin{pmatrix} \frac{J_r}{I_{xx}}\dot{\theta}\Omega_d \\ -\frac{J_r}{I_{yy}}\dot{\phi}\Omega_d \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{1}{I_{xx}}\tau_\phi \\ \frac{1}{I_{yy}}\tau_\theta \\ \frac{1}{I_{zz}}\tau_\psi \end{pmatrix} \tag{13}$$

Consequently, quadrotor is an underactuated system with six outputs $(x, y, z, \phi, \theta, \psi)$ and four control inputs ($u_T, \tau_\phi, \tau_\theta, \tau_\psi$).

Finally, the quadrotor dynamic model can be written in the following form:

$$\begin{aligned} \ddot{x} &= \left(c_\phi s_\theta c_\psi + s_\phi s_\psi\right)\frac{1}{m}u_1 \\ \ddot{y} &= \left(c_\phi s_\theta s_\psi - s_\phi c_\psi\right)\frac{1}{m}u_1 \\ \ddot{z} &= -g + \left(c_\phi c_\theta\right)\frac{1}{m}u_1 \\ \ddot{\phi} &= \dot{\theta}\dot{\psi}\left(\frac{I_{yy}-I_{zz}}{I_{xx}}\right) - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_d + \frac{l}{I_{xx}}u_2 \\ \ddot{\theta} &= \dot{\phi}\dot{\psi}\left(\frac{I_{zz}-I_{xx}}{I_{yy}}\right) + \frac{J_r}{I_{yy}}\dot{\phi}\Omega_d + \frac{l}{I_{yy}}u_3 \\ \ddot{\psi} &= \dot{\theta}\dot{\phi}\left(\frac{I_{xx}-I_{yy}}{I_{zz}}\right) + \frac{1}{I_{zz}}u_4 \end{aligned} \tag{14}$$

with a renaming of the control inputs as:

$$\begin{aligned} u_1 &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ u_2 &= b(\Omega_4^2 - \Omega_2^2) \\ u_3 &= b(\Omega_3^2 - \Omega_1^2) \\ u_4 &= d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{aligned} \tag{15}$$

and the definition of disturbance:

$$\Omega_d = \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3 \tag{16}$$

3 Control System for Quadrotor

In this paper, only the z-directional linear motion (altitude) and angular motion (three attitude angles, roll, pitch and yaw) are chosen as four controllable degrees of freedom (DOF). For the design of the controller, the following state variables are defined:

$$x = \left[z\ \dot{z}\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\right]^T = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8]^T \tag{17}$$

Generally, the altitude and the rotational dynamics of quadrotor can be decomposed into four nonlinear subsystems:

Altitude subsystem:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1(x) + g_1(x)u_1 \end{aligned} \tag{18}$$

where

$$\begin{aligned} f_1(x) &= -g \\ g_1(x) &= c_\phi c_\theta \left(\frac{1}{m}\right) \end{aligned}$$

Roll subsystem:

$$\begin{aligned} \dot{x}_3 &= x_4 \\ \dot{x}_4 &= f_2(x) + g_2(x)u_2 \end{aligned} \tag{19}$$

where

$$f_2(x) = \dot{\theta}\dot{\psi}\left(\frac{I_{yy}-I_{zz}}{I_{xx}}\right) - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_d$$

$$g_2(x) = \frac{l}{I_{xx}}$$

Pitch subsystem:

$$\begin{aligned} \dot{x}_5 &= x_6 \\ \dot{x}_6 &= f_3(x) + g_3(x)u_3 \end{aligned} \tag{20}$$

where

$$f_3(x) = \dot{\phi}\dot{\psi}\left(\frac{I_{zz}-I_{xx}}{I_{yy}}\right) + \frac{J_r}{I_{yy}}\dot{\phi}\Omega_d$$

$$g_3(x) = \frac{l}{I_{yy}}$$

Yaw subsystem:

$$\begin{aligned} \dot{x}_7 &= x_8 \\ \dot{x}_8 &= f_4(x) + g_4(x)u_4 \end{aligned} \tag{21}$$

where

$$f_4(x) = \dot{\theta}\dot{\phi}\left(\frac{I_{xx}-I_{yy}}{I_{zz}}\right)$$

$$g_4(x) = \frac{1}{I_{zz}}$$

Thus each subsystem can be expressed into a single-input nonlinear system as the following form:

$$x^{(n)} = f(x) + g(x)u, \quad n = 2 \tag{22}$$

where u is the input; $f(x)$and $g(x)$ are the nonlinear function.

3.1 Backstepping Control System

A suitable control law for the system (22) need to be designed so that the desired control objective can be achieved. Since the description of the control system design is similar for each subsystem, for simplicity only one subsystem of the four DOF quadrotor systems is considered. The backstepping control is designed sequently as follows:

Step 1: The tracking error is assigned as:

$$e_1 = x_d - x \tag{23}$$

where x_d is a desired trajectory.

Differentiating Eq. (23), it is obtained that:

$$\dot{e}_1 = \dot{x}_d - \dot{x} \tag{24}$$

The first Lyapunov function is selected as:

$$V_1(e_1) = \frac{1}{2}e_1^2 \quad (25)$$

The derivative of V_1 is:

$$\dot{V}_1(e_1) = e_1\dot{e}_1 = e_1(\dot{x}_d - \dot{x}) \quad (26)$$

$\dot{x}$ can be viewed as a virtual control. The desired value of virtual control known as a stabilizing function can be defined as follows:

$$\alpha = \dot{x}_d + k_1 e_1 \quad (27)$$

where k_1 is a positive constant.

By substituting the virtual control by its desired value, Eq. (26) then becomes:

$$\dot{V}_1(e_1) = -k_1 e_1^2 \leq 0 \quad (28)$$

Step 2: The deviation of the virtual control from its desired value can be defined as:

$$e_2 = \alpha - \dot{x} = \dot{x}_d + k_1 e_1 - \dot{x} \quad (29)$$

The derivative of e_2 is expressed as:

$$\dot{e}_2 = \dot{\alpha} - \ddot{x} = k_1\dot{e}_1 + \ddot{x}_d - f(x) - g(x)u \quad (30)$$

The second Lyapunov function is chosen as:

$$V_2(e_1, e_2) = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 \quad (31)$$

Finding derivative of (31), yields:

$$\dot{V}_2(e_1, e_2) = e_1\dot{e}_1 + e_2\dot{e}_2 = e_1(\dot{x}_d - \dot{x}) + e_2(\dot{\alpha} - \ddot{x}) = e_1(e_2 - k_1 e_1) + e_2(k_1\dot{e}_1 + \ddot{x}_d - f(x) - g(x)u) = -k_1 e_1^2 + e_2(e_1 + k_1\dot{e}_1 + \ddot{x}_d - f(x) - g(x)u) \quad (32)$$

Step 3: For satisfying $\dot{V}_2(e_1, e_2) \leq 0$, the control input u is selected as:

$$u = \frac{1}{g(x)}(e_1 + k_1\dot{e}_1 + \ddot{x}_d - f(x) + k_2 e_2) \quad (33)$$

where k_2 is a positive constant. The term $k_2 e_2$ is added to stabilize the tracking error e_1.

Substituting (33) into (32), the following equation can be obtained:

$$\dot{V}_2(e_1, e_2) = -k_1 e_1^2 - k_2 e_2^2 = -E^T K E \leq 0 \quad (34)$$

where $E = [e_1\ e_2]^T$ and $K = diag(k_1, k_2)$. Since $\dot{V}_2(e_1, e_2) \leq 0$, $\dot{V}_2(e_1, e_2)$ is negative semi-definite.

Therefore, the control law in (33) will asymptotically stabilize the system.

3.2 Overview of Particle Swarm Optimization

The PSO is a type of swarm intelligence methods and a population based algorithm that normally used as optimization tool. Each individual (particle) of the population is a candidate solution. In PSO each particle navigates around the search (solution) space by updating their velocity according to its own and also the other particles searching experience. Each particle attempts to imitate the successful peers attributes to improve themselves. Further, each particle has a memory to keeps track the previous best position (known as *pbest*) and corresponding fitness. The particles with greatest fitness in the population is called *gbest*.

There are three steps involve in the basic PSO algorithm, namely, generating particles' positions and velocities, velocity update, and finally, position update [23]. First, by using the design upper, x_{max} and lower, x_{min} bound values, the initial positions, x_i^k, and velocities, v_i^k, of particles are randomly generated, as expressed in Eqs. (34) and (35):

$$x_i^0 = x_{min} + rand(x_{max} - x_{min}) \quad (35)$$

$$v_i^0 = x_{min} + rand(x_{max} - x_{min}) \quad (36)$$

In Eqs. (34) and (35), the subscript and superscript denoting the *i*th particle at iteration k, respectively, while *rand* is a uniformly distributed random variable that can take any value between 0 and 1.

The second step is to update the velocities of all particles according to the following expressions:

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot rand \cdot (pbest - x_i^k) + c_2 \cdot rand \cdot (gbest - x_i^k) \quad (37)$$

Three weight factors, namely, inertia factor, w, self confidence factor, c_1, and swarm confidence factor, c_2, are incorporated in Eq. (37) to effect the particles direction. The following inertia weight is used [24]:

$$w = w_{max} - (w_{max} - w_{min})\, k/k_{max} \quad (38)$$

where k and k_{max} are the current number of iterations and the maximum number of iterations, respectively. w_{max} and w_{min} are the maximum and minimum weights respectively.

Lastly, velocity vector is used to update the position of each particle as Eq. (39) and illustrated in Figure 2.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (39)$$

Repeat the three steps of (i) velocity update, (ii) position update, and (iii) fitness calculations until a stopping criterion is reached.

3.3 PSO-Based Optimal Backstepping Control System

In the previous section a controller (33) has been designed to stabilize each subsystem. The coefficients k_1, k_2 are control parameters and need to be positive to satisfy stability criteria. Conventionally, these parameters are selected by

trial and error. To overcome this drawback, this paper adopts the PSO for selecting the optimal value of the backstepping control parameters. The performance of the controller varies according to adjusted parameters. The structure of the proposed OBC for one subsystem is as shown in Figure 3. As aforementioned, the quadrotor system is comprised of four subsystems. Then there are in sum eight control parameters that need to be selected simultaneously so that each subsystem is asymptotically stable.

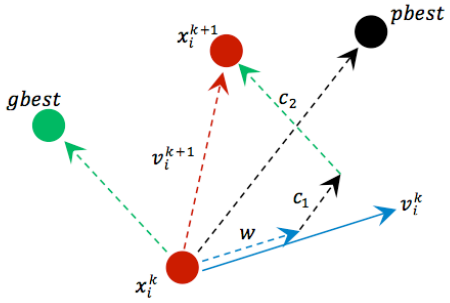


Fig. 2 Depiction of the velocity and position updates in PSO

In the present study, an integral absolute error (*IAE*) is utilized to judge the performance of the controller. *IAE* criterion is broadly used to assess the performance of the control framework [25]. The index *IAE* is defined such as:

$$IAE = \int_0^t |e(t)|\, dt \tag{40}$$

Since the system is comprised of four subsystems, hence a vector integral absolute error for the entire system is taken as $IAE_T = [IAE_A\ IAE_R\ IAE_P\ IAE_Y]$, where the subscripts are denoted for altitude, roll, pitch and yaw subsystem, respectively. The purpose of PSO algorithm is to minimize the fitness function J, expressed as:

$$J = IAE_T \cdot W \tag{41}$$

Where $W = [W_1\ W_2\ W_3\ W_4]^T$ is weighting vector used to set the priority of the multiple objective performance index (MOPI) parameters and the value of "W" varies from 0 to 1. Since the minimizations of the indexes are equally important, hence weights for the four objectives to be obtained are considered equal. For calculating the fitness function, the quadrotor system model is simulated for the time period, t. In order to enhance the transient response and steady-state errors, the fitness function has to be minimized. The PSO based approach to find out the optimal set of backstepping controller parameters is shown in Figure 4.

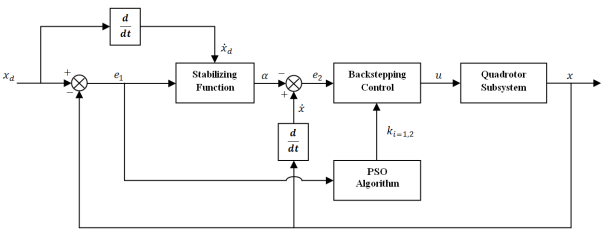


Fig. 3 Structure of the optimal backstepping controller for one subsystem

4 Simulation Results

In this section, the performance of the proposed approach is evaluated. The algorithm is executed in MATLAB/SIMULINK simulation environment. The model parameter values of the quadrotor system are adopted from [26] and listed in Table 1. Initially, the quadrotor control model is utilized to search the controller parameter optimization, and later the identified parameter values are used by the quadrotor control system developed in MATLAB/SIMULINK for further evaluation.

The following values are assigned in this work:

i. Dimension of the search space = 8 (i.e., $k_{i=1\ldots8}$);
ii. Population/swarm size = 15;
iii. The number of maximum iteration = 20;
iv. The self and swarm confident factor, c_1 and c_2 = 2;
v. The inertia weight factor w is set by (38), where $w_{max} = 0.9$ and $w_{min} = 0.4$;
vi. The searching ranges for the backstepping parameters are limited to $[0, 20]$;
vii. The simulation time, t is equal to 10s;
viii. Optimization process is repeated for 20 times;

Table 1 Parameters of the quadrotor

Parameter	Description	Value	Units
g	Gravity	9.81	m/s^2
m	Mass	0.5	kg
l	Distance	0.2	m
I_{xx}	Roll inertia	4.85×10^{-3}	$kg \cdot m^2$
I_{yy}	Pitch inertia	4.85×10^{-3}	$kg \cdot m^2$
I_{zz}	Yaw inertia	8.81×10^{-3}	$kg \cdot m^2$
b	Thrust factor	2.92×10^{-6}	
d	Drag factor	1.12×10^{-7}	

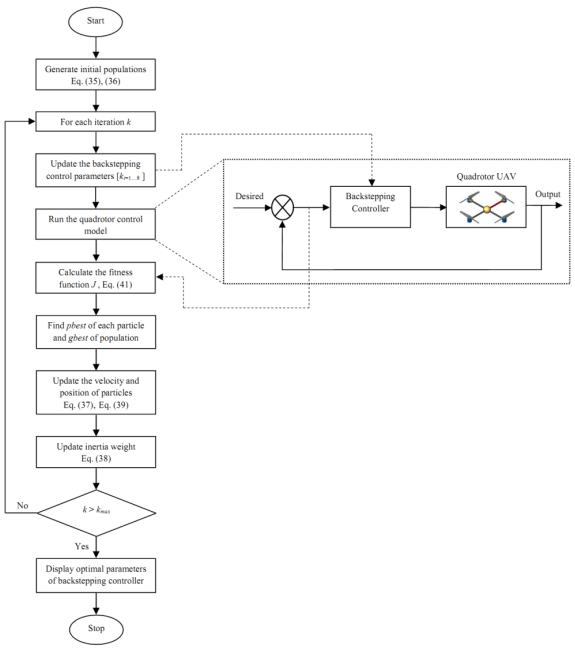


Fig. 4 The flowchart of the PSO-based optimal backstepping control system

The best optimized controller value is chosen from the finest set of values among the simulation runs. The parameter and fitness values of each particle during the simulation are summarized in Table 2. The best fitness value is $7.312e-008$ appeared in iteration number 7, and the optimal parameters are $k_1 = 14.28$, $k_2 = 14.52$, $k_3 = 14.64$, $k_4 = 14.14$, $k_5 = 14.38$, $k_6 = 14.21$, $k_7 = 14.61$ and $k_8 = 14.11$. The variation of the fitness function with number of iterations is shown in Figure 5. Meanwhile, the variations of backstepping control parameters with respect to the number of iterations are shown in Figure 6. As can be seen, through about 20 iterations, the PSO method can prompt convergence and obtain good fitness value. These results show that the PSO approach can search optimal backstepping controller parameters quickly and efficiently.

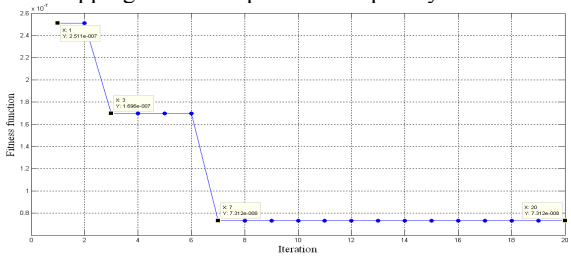


Fig. 5 The convergence of fitness function with number of iterations

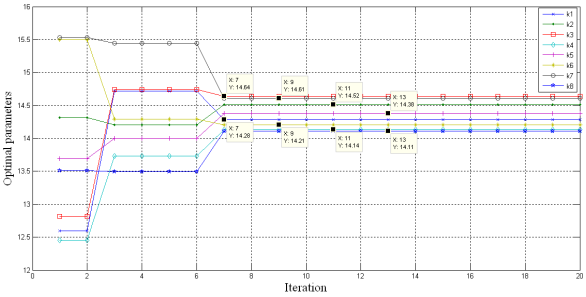


Fig. 6 The variations of optimal parameters versus number of iterations

Table 2 The parameters and fitness value of each optimal particle

Iteration No.	Optimal parameters	Fitness value
1	$k_1 = 12.60, k_2 = 14.31$ $k_3 = 12.82, k_4 = 12.46$ $k_5 = 13.69, k_6 = 15.50$ $k_7 = 15.53, k_8 = 13.51$	$2.511e-007$
3	$k_1 = 14.72, k_2 = 14.21$ $k_3 = 14.74, k_4 = 13.74$ $k_5 = 14.00, k_6 = 14.29$ $k_7 = 15.44, k_8 = 13.49$	$1.696e-007$
7	$k_1 = 14.28, k_2 = 14.52$ $k_3 = 14.64, k_4 = 14.14$ $k_5 = 14.38, k_6 = 14.21$ $k_7 = 14.61, k_8 = 14.11$	$7.312e-008$
20	$k_1 = 14.28, k_2 = 14.52$ $k_3 = 14.64, k_4 = 14.14$ $k_5 = 14.38, k_6 = 14.21$ $k_7 = 14.61, k_8 = 14.11$	$7.312e-008$

To explore the effectiveness of the proposed optimal backstepping controller, two simulation experiments have been performed on the quadrotor UAV. The first experiment provides the simulation results of the proposed controller for a stabilizing problem. While the second experiment investigates the performance of the scheme for a tracking problem.

4.1 Simulation experiment 1: stabilizing problem

The control objectives are to reach and maintain quadrotor at a certain desired altitude/attitude, such that the helicopter can hover at a fixed point. The desired altitude/attitude is given by $x_{id} = [z_d, \phi_d, \theta_d, \psi_d] = [20, 0, 0, 0]^T$. The initial states are given by $z = 0, \phi = 0.2, \theta = 0.2$ and $\psi = 0.2$. Simulation results show the control design is able to stabilize the helicopter in hover mode. Under the proposed OBC, it can be observed that the altitude/attitude of the quadrotor can be maintained at the desired altitude/attitude, that is, the hovering flight is stable as shown in Figure 7. Also from this figure, it can be noted that the attitude states converge to zero set-point for a given initial condition rapidly as the system starts to achieve asymptotic stability of the quadrotor system.

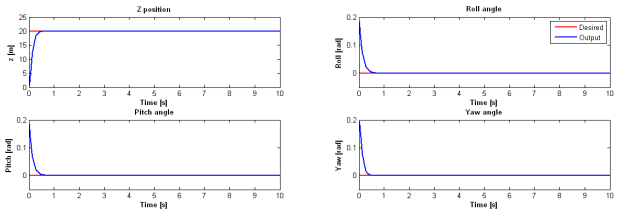


Fig. 7 Altitude/attitude of the hovering quadrotor using OBC

4.2 Simulation experiment 2: tracking problem

To further highlight the advantage of the proposed control structure the simulation results of the OBC for altitude tracking due to periodic trapezoidal function is depicted in Figure 8. As it can be seen, the system can track the desired reference trajectory quickly. Also it is obviously that the OBC can give small tracking error and good tracking performance.

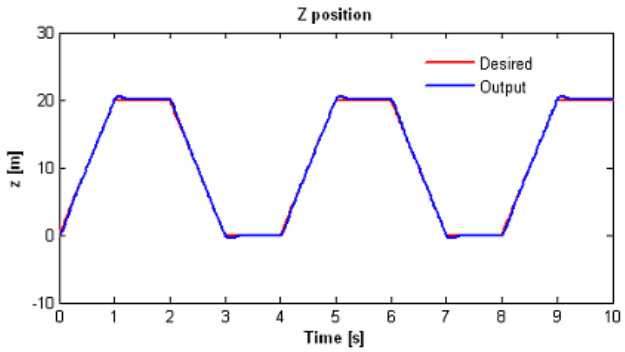


Fig. 8 Altitude tracking response due to periodic trapezoidal function using OBC

As aforementioned, the improper selection of the backstepping control parameters leads to inappropriate responses of the system. As can be seen from Figure 9 and 10, it is evident that the poorly defined of backstepping control parameters will degrade the performance response of the system. From Figure 9, some oscillation in the transient response can be observed. The settling time is also significantly longer than that achieved by using OBC. At the

same time, the ability of the system to track the reference trajectory is also affected as shown in Figure 10.

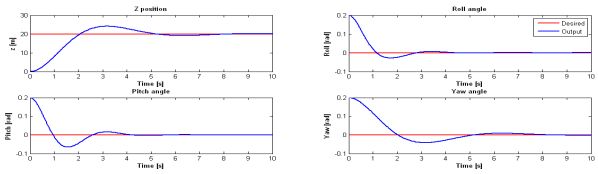
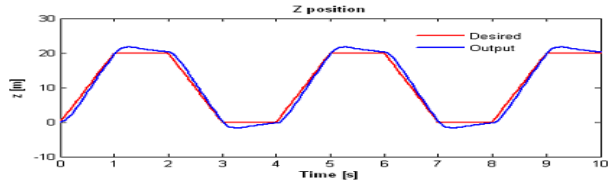


Fig. 9 Altitude/attitude of the hovering quadrotor using BC with improper parameters



.

Fig. 10 Altitude tracking response due to periodic trapezoidal function using BC with improper parameters

5 Conclusions

In this paper, the application of an optimal backstepping controller for the trajectory tracking and stabilization of quadrotor UAV is successfully demonstrated. First, the mathematical model of the quadrotor is introduced. Then, the proposed optimal backstepping controller which can automatically select the controller parameters by using PSO algorithm is developed. The backstepping control design is derived based on Lyapunov function, so that the stability of the system can be guaranteed. Finally, the proposed control scheme is applied to autonomous quadrotor UAV. Simulation results show that high performance response can be achieved by using the proposed control system

References

1. Hamel T., Mahony R., Lozano R., Ostrowski J.: Dynamic Modelling and Configuration and Stabilization for an X4-Flyer, International Federation of Automatic Control Symposium, 2002; 1(2): 3.
2. Pounds P., Mahony R., Hynes P., Roberts J.: Design of a four-rotor aerial robot, Proc. 2002 Australasian Conference on Robotic and Automation, Auckland, 2002; 29.
3. McKerrow P.: Modelling the Draganflyer four-rotor helicopter, IEEE International Conference on Robotics and Automation ICRA '04, vol. 4, 2004; 3596-3601.
4. Castillo P., Lozano R., Dzul A.: Stabilization of a mini rotorcraft with four rotors, IEEE Control Systems Magazine 2005; 25(6): 45-55.
5. Salih A.L., Moghavvemi M., Mohamed H.A.F., Gaeid K.S.: Modelling and PID controller design for a quadrotor unmanned air vehicle, IEEE International Conference on Automation Quality and Testing Robotics (AQTR), 2010; 1-5.
6. De Queiroz M.S., Dawson D.M.: Nonlinear control of active magnetic bearings: a backstepping approach, IEEE Transactions on Control Systems Technology 1996; 4(5): 545-552.
7. Kim K.S., Kim Y.: Robust backstepping control for slew maneuver using nonlinear tracking function, IEEE Transactions on Control Systems Technology 2003; 11(6): 822-829.
8. Karimi A., Feliachi A.: Decentralized adaptive backstepping control of electric power systems, Electric Power Systems Research 2008; 78(3): 484-493.
9. Wu ZJ, Xie XJ, Shi P, Xia Y. Backstepping controller design for a class of stochastic nonlinear systems with Markovian switching. Automatica 2009; 45(4): 997-1004.
10. Krstic M., Kanellakopoulos I, Kokotovic P. Nonlinear and adaptive control design. New York 1995.
11. Madani T., Benallegue A.: Backstepping Control for a Quadrotor Helicopter, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006; 3255-3260.
12. Bouadi H., Bouchoucha M., Tadjine M.: Modelling and Stabilizing Control Laws Design Based on Backstepping for an UAV Type-Quadrotor, Proceedings of the IFAC Symposium on IAV, Toulouse, France, 2007
13. De Vries E., Subbarao K.: Backstepping based nested multi-loop control laws for a quadrotor, 11th International Conference on Control Automation Robotics & Vision (ICARCV), 2010; 1911-1916.
14. Raffo G.V., Ortega M.G., Rubio F.R.: Backstepping/nonlinear H∞ control for path tracking of a quadrotor unmanned aerial vehicle, American Control Conference, 2008, 2008; 3356-3361.
15. Regula G., Lantos B.: Backstepping based control design with state estimation and path tracking to an indoor quadrotor helicopter, Electrical Engineering and Computer Science, 2011; 53(3-4): 151-161.
16. Kennedy J., Eberhart R.: Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, 1995; 1942-1948.
17. Abido M.: Optimal design of power-system stabilizers using particle swarm optimization, IEEE Transactions on Energy Conversion 2002; 17(3): 406-413.
18. Rini D.P., Shamsuddin S.M., Yuhaniz S.S.: Particle swarm optimization: Technique, system and challenges, International Journal of Computer Applications 2011; 14(1):19–27.
19. Marinaki M., Marinakis Y., Stavroulakis G.E.: Fuzzy control optimized by a multi-objective particle swarm optimization algorithm for vibration suppression of smart structures, Structural and Multidisciplinary Optimization 2011; 43(1): 29-42.
20. Sinha S, Patel R, Prasad R. Application of GA and PSO tuned fuzzy controller for AGC of three area thermal-thermal-hydro power system. International Journal of Computer Theory and Engineering 2010; 2(2): 1793-8201.
21. Zuo Z.: Trajectory tracking control design with command-filtered compensation for a quadrotor, IET Control Theory & Applications 2010; 4(11): 2343-2355.
22. Olfati-Saber R.: Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles, Ph.D. thesis, Massachusetts Institute of Technology, 2000.
23. Hassan R., Cohanim B., De Weck O., Venter G.: A comparison of particle swarm optimization and the genetic algorithm, Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference, 2005.
24. Lalitha M.P., Reddy V.V., Usha V.: Optimal DG placement for minimum real power loss in radial distribution systems using PSO, Journal of Theoretical and Applied Information Technology 2010; 13(2): 107-116.
25. Allaoua B., Gasbaoui B., Mebarki B.: Setting up PID DC motor speed control alteration parameters using particle swarm optimization strategy, Leonardo Electronic Journal of Practices and Technologies 2009; 14: 19-32.
26. Voos H.: Nonlinear control of a quadrotor micro-UAV using feedback-linearization, IEEE International Conference on Mechatronics (ICM 2009), 2009; 1-6.