Robotica (2018) volume 36, pp. 1701–1727. © Cambridge University Press 2018
doi:10.1017/S0263574718000668

Design and application of an adaptive backstepping sliding mode controller for a six-DOF quadrotor aerial robot

Mohd Ariffanan Mohd Basri*

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

(Accepted July 11, 2018. First published online: August 3, 2018)

SUMMARY

The quadrotor aerial robot is a complex system and its dynamics involve nonlinearity, uncertainty, and coupling. In this paper, an adaptive backstepping sliding mode control (ABSMC) is presented for stabilizing, tracking, and position control of a quadrotor aerial robot subjected to external disturbances. The developed control structure integrates a backstepping and a sliding mode control approach. A sliding surface is introduced in a Lyapunov function of backstepping design in order to further improve robustness of the system. To attenuate a chattering problem, a saturation function is used to replace a discontinuous sign function. Moreover, to avoid a necessity for knowledge of a bound of external disturbance, an online adaptation law is derived. Particle swarm optimization (PSO) algorithm has been adopted to find parameters of the controller. Simulations using a dynamic model of a six degrees of freedom (DOF) quadrotor aerial robot show the effectiveness of the approach in performing stabilization and position control even in the presence of external disturbances.

KEYWORDS: Quadrotor aerial robot, Backstepping control, Adaptive control, Sliding mode control, Particle swarm optimization

1. Introduction

Nowadays, aerial robots are used in different military and civilian applications, such as border patrolling, geological surveying, traffic monitoring, area mapping, search and rescue, and reconnaissance missions. Quadrotors are a special class among different types of aerial robots. In the past few years, research on quadrotor aerial robots has significantly increased. This is due to their advantages such as their capability to show precise movements, hover and vertical take-off and landing (VTOL). Quadrotors belong to the helicopter rotorcraft class, which has nonlinear dynamic behavior and is difficult to stabilize. The main challenges of the aerial robots are mainly due to their unstable nature and complexity of the dynamic model, thus offering a challenge to design a stabilization and position control. Many methods have been proposed to control a quadrotor aerial robot, such as linear quadratic regulator (LQR) control,[1–3] proportional–integral–derivative (PID) control,[4–6] fuzzy logic (FL) control,[7,8] sliding mode control (SMC),[9,10] and backstepping control.[11–14] Among these methods, backstepping control has received great attention because of its systematic and recursive design methodology for nonlinear feedback control.

The backstepping control scheme is a nonlinear control method based on the Lyapunov theorem. The advantage of backstepping control technique lies in its design flexibility, which is mainly because of the recursive use of Lyapunov functions. Unfortunately, the backstepping control system cannot achieve robustness to model uncertainties and external disturbances. Thus, the motivation of this work is to design a new backstepping-based robust control approach for a quadrotor aerial robot with external disturbances. SMC is a type of nonlinear control systems that have proven very robust to model uncertainties and matched disturbances.[15] The SMC has been successfully implemented in

* Corresponding author. E-mail: ariffanan@fke.utm.my

varieties, forms, and numerous real-world applications, such as satellite,[16] rocket,[17] aircraft,[18] batch reactor,[19] flexible spacecraft,[20] robot manipulator,[21] shape memory alloy actuator,[22] and wheeled inverted pendulum systems.[23] Thus, owing to the merits of backstepping control and SMC, in this paper, both control schemes are combined for stabilization and position control of a quadrotor aerial robot. In the design, an SMC scheme is introduced in the final step of backstepping method. To further improve the backstepping sliding mode control (BSMC) design, an adaptation scheme is derived to adapt the design parameters of the controller. Therefore, a prior knowledge on the bound of disturbances is not required. The boundedness of tracking errors and the closed-loop stability of the system are guaranteed, since the control law is derived in the sense of the Lyapunov theorem. Compared with the method presented in refs. [14] and [10], the designed controller has the superiority in robustness and adaptability, respectively. Thus, this approach can be benefited for a wide class of nonlinear systems with the influences of external disturbances.

In order to determine the parameters of the control design, particle swarm optimization (PSO) algorithm has been utilized. PSO is one of the modern heuristic algorithms based on the population optimization algorithm.[24] The method is motivated by the behavior of organisms, such as fish schooling and bird flocking. Generally, PSO has features such as a straightforward algorithm, simple to execute, computationally efficient, and rapid convergence. PSO also has a flexible and well-balanced mechanism to enhance the global and local exploration abilities.[25] The PSO algorithm has been used effectively on a wide range of engineering and computer science problems.[26,27] Thus, due to these advantages, PSO is used to compute the optimal control parameters.

The prominent advantage of the proposed control approach is that the structure of the controller is simple but very effective. Furthermore, no knowledge on the bound of disturbances is needed in advance. The main contribution of this paper is a successful development of a new nonlinear control structure that does not require any prior knowledge on the bound of disturbances based on the backstepping technique augmented with the compensation controller. The effectiveness and merits of the proposed technique are exemplified by conducting several simulation experiments, including altitude and attitude stabilization, automatic take-off and landing, and trajectory tracking missions for a quadrotor aerial robot.

The remainder of this paper is organized as follows. In Section 2, a mathematical model of the quadrotor is presented. In Sections 3, 4, 5, and 6, a backstepping controller, optimization of backstepping controller, backstepping sliding mode controller, and adaptive backstepping sliding mode controller of the quadrotor are described, respectively. The simulation results and their discussions are given in Section 7. Finally, conclusions are made in Section 8.

2. Quadrotor Systems Modeling

2.1. Quadrotor description

The quadrotor aerial robot, shown in Fig. 1, has four rotors to generate the propeller forces $F_{i=1,2,3,4}$. The four rotors can be thought of as two pairs, (1,3)@(front, back) and (2,4)@(left, right). One pair rotates clockwise, while the other rotates counter clockwise in order to balance the torques and produce yaw motion as needed. Yaw motion can be obtained from the difference in the counter torque between each pair of propellers, (1,3) and (2,4). When all four rotors are spinning with the same angular velocity, the net yaw is zero, and a difference in velocities between the two pairs creates either positive or negative yaw motion. To achieve the up (down) motion, the rotor speeds need to be increased (decreased) altogether with the same magnitude. Forward (backward) motion which is related to the pitch angle, θ, can be obtained by decreasing the front (back) rotor thrust and increasing the back (front) rotor thrust. Finally, a sideways motion which is related to the roll angle, ϕ, can be achieved by decreasing the right (left) rotor thrust and increasing the left (right) rotor thrust. Figure 2 shows the various movements of a quadrotor due to changes in rotor speeds.

In order to develop a model of the quadrotor, a few assumptions are established in order to accommodate the controller design. The assumptions are as follows:[28]

Assumption 1: Quadrotor is a rigid body and has symmetric structure.

Assumption 2: Aerodynamic effects can be ignored at low speed.

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

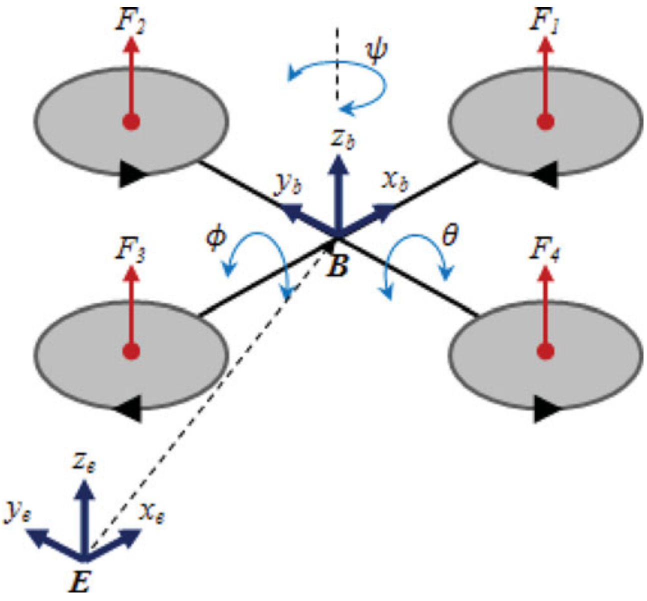


Fig. 1. Quadrotor aerial robot configuration.

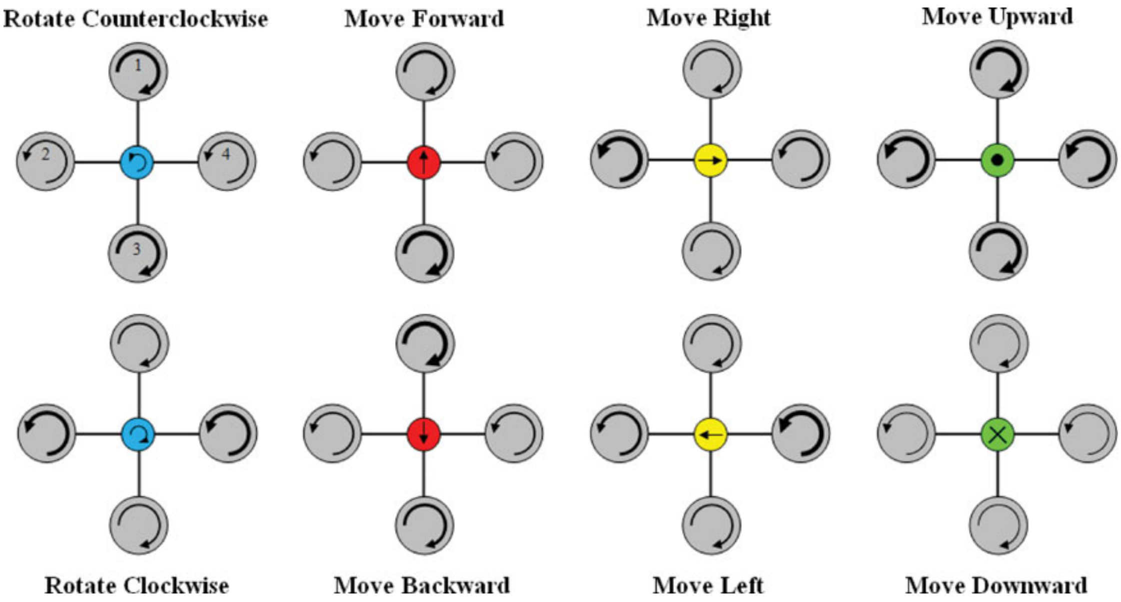


Fig. 2. The movements of a quadrotor: the arrow width is proportional to rotor speeds.

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 coincide.

2.2. Quadrotor kinematic model

Let us consider an earth-fixed frame $E = \{x_e, y_e, z_e\}$ and a body-fixed frame $B = \{x_b, y_b, z_b\}$, as seen in Fig. 1. Let $q = (x, y, z, \phi, \theta, \psi) \in R^6$ be the generalized coordinates for the quadrotor, where (x, y, z) denotes the absolute position of the rotorcraft and (ϕ, θ, ψ) are the three Euler angles (roll, pitch, and yaw) that describe the orientation of the aerial robot. Therefore, the model could be separated in two coordinate subsystems: translational and rotational. They are defined respectively by the following:

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

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

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

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 ref. [29] and defined accordingly by the following equations:

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

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

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 vectors with respect to (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 vectors w.r.t. the earth-fixed frame E and body-fixed frame B, respectively.

2.3. Quadrotor dynamic model

The dynamic model of quadrotor is derived from the Newton–Euler approach, as formulated in ref. [30]. 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:

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

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 equation 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

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

rotor speeds. G_a and τ are defined as

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

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

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

Then, by recalling Eqs. (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 under-actuated 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

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

and Ω_d is defined as

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

3. Backstepping Control System for Quadrotor

The dynamic model (14) with consideration of external disturbance can be represented into a nonlinear dynamic equation described as follows:

$$\ddot{X} = f(X) + g(X)u + \delta \tag{17}$$

where u, X, and δ are, respectively, the input, state, and external disturbance vector given as follows:

$$u = [u_1\ u_2\ u_3\ u_4]^T \tag{18}$$

$$X = [x_1\ x_3\ x_5\ x_7\ x_9\ x_{11}]^T = \left[z\ \phi\ \theta\ \psi\ x\ y\right]^T \tag{19}$$

$$\delta = [\delta_1\ \delta_3\ \delta_5\ \delta_7\ \delta_9\ \delta_{11}]^T \tag{20}$$

From Eqs. (14) and (19), the nonlinear dynamic function $f(X)$ and nonlinear control function $g(X)$ matrices can be written accordingly as

$$f(X) = \begin{pmatrix} -g \\ \dot{\theta}\dot{\psi}a_1 - \dot{\theta}a_2\Omega_d \\ \dot{\phi}\dot{\psi}a_3 + \dot{\phi}a_4\Omega_d \\ \dot{\theta}\dot{\phi}a_5 \\ 0 \\ 0 \end{pmatrix} \quad g(X) = \begin{pmatrix} u_z\frac{1}{m} & 0 & 0 & 0 \\ 0 & b_1 & 0 & 0 \\ 0 & 0 & b_2 & 0 \\ 0 & 0 & 0 & b_3 \\ u_x\frac{1}{m} & 0 & 0 & 0 \\ u_y\frac{1}{m} & 0 & 0 & 0 \end{pmatrix} \tag{21}$$

with the abbreviations $a_1 = (I_{yy} - I_{zz})/I_{xx}$, $a_2 = J_r/I_{xx}$, $a_3 = (I_{zz} - I_{xx})/I_{yy}$, $a_4 = J_r/I_{yy}$, $a_5 = (I_{xx} - I_{yy})/I_{zz}$, $b_1 = l/I_{xx}$, $b_2 = l/I_{yy}$, $b_3 = 1/I_{zz}$, $u_x = (c_\phi s_\theta c_\psi + s_\phi s_\psi)$, $u_y = (c_\phi s_\theta s_\psi - s_\phi c_\psi)$, and $u_z = (c_\phi c_\theta)$.

The control objective is to design a suitable control law so that the state trajectory X of the quadrotor system can track a desired reference trajectory $X_d = [x_{1d}\ x_{3d}\ x_{5d}\ x_{7d}\ x_{9d}\ x_{11d}]^T$ despite the presence of external disturbance. Since the description of the control system design of the quadrotor is similar for each one of the six controllable degrees of freedom (DOF), for simplicity only one DOF is considered.

The design of ideal backstepping control (IBC) is designed sequently as follows:

Step 1: The tracking error is assigned as

$$e_1 = x_{1d} - x_1 \tag{22}$$

where x_{1d} is a desired trajectory.

Differentiating Eq. (22), it is obtained that

$$\dot{e}_1 = \dot{x}_{1d} - \dot{x}_1 \tag{23}$$

The first Lyapunov function is chosen as

$$V_1(e_1) = \frac{1}{2}e_1^2 \tag{24}$$

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

The derivative of V_1 is

$$\dot{V}_1(e_1) = e_1\dot{e}_1 = e_1(\dot{x}_{1d} - \dot{x}_1) \tag{25}$$

$\dot{x}_1$ can be viewed as a virtual control. The desired value of virtual control known as a stabilizing function can be defined as follows:

$$\alpha_1 = \dot{x}_{1d} + k_1 e_1 \tag{26}$$

where k_1 is a positive constant and should be determined by the PSO algorithm.

By substituting the virtual control by its desired value, Eq. (25) then becomes

$$\dot{V}_1(e_1) = -k_1 e_1^2 \leq 0 \tag{27}$$

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

$$e_2 = \dot{x}_1 - \alpha_1 = \dot{x}_1 - \dot{x}_{1d} - k_1 e_1 \tag{28}$$

The derivative of e_2 is expressed as

$$\begin{aligned} \dot{e}_2 &= \ddot{x}_1 - \dot{\alpha}_1 \\ &= f(x_1) + g(x_1)u_1 + \delta_1 - \ddot{x}_{1d} - k_1\dot{e}_1 \end{aligned} \tag{29}$$

The second Lyapunov function is chosen as

$$V_2(e_1, e_2) = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 \tag{30}$$

Finding derivative of Eq. (30) yields

$$\begin{aligned} \dot{V}_2(e_1, e_2) &= e_1\dot{e}_1 + e_2\dot{e}_2 \\ &= e_1(\dot{x}_{1d} - \dot{x}_1) + e_2(\ddot{x}_1 - \dot{\alpha}_1) \\ &= e_1(-e_2 - k_1 e_1) + e_2(f(x_1) + g(x_1)u_1 + \delta_1 - \ddot{x}_{1d} - k_1\dot{e}_1) \\ &= -k_1 e_1^2 + e_2(-e_1 + f(x_1) + g(x_1)u_1 + \delta_1 - \ddot{x}_{1d} - \ddot{x}_d - k_1\dot{e}_1) \end{aligned} \tag{31}$$

Step 3: Assuming that the external disturbance is well known, an IBC can be obtained as

$$u_{IB} = \frac{1}{g(x_1)}(e_1 + k_1\dot{e}_1 + \ddot{x}_{1d} - f(x_1) - \delta_1 - k_2 e_2) \tag{32}$$

where k_2 is a positive constant and should be also determined by the PSO algorithm. The term $k_2 e_2$ is added to stabilize the tracking error e_1.

Substituting Eq. (32) into Eq. (31), the following equation can be obtained:

$$\dot{V}_2(e_1, e_2) = -k_1 e_1^2 - k_2 e_2^2 = -E^T K E \leq 0 \tag{33}$$

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

Therefore, the IBC in Eq. (32) will stabilize the system.

4. Backstepping Control Parameters Optimization

4.1. Overview of PSO

The PSO is a type of swarm intelligence methods and a population-based algorithm that is normally used as an optimization tool.[24] 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

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

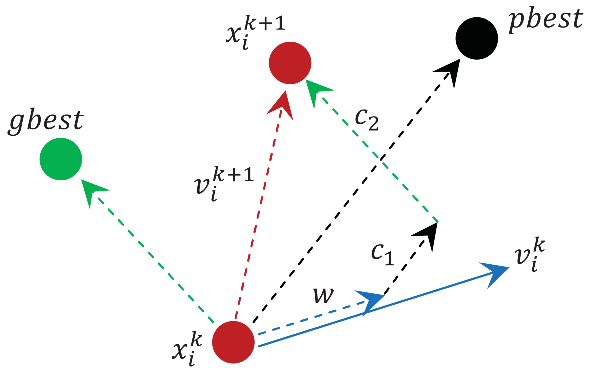


Fig. 3. Depiction of the velocity and position updates in PSO.

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 keep the track of the previous best position (known as *pbest*) and corresponding fitness. The particle with greatest fitness in the population is called *gbest*.

There are three steps involved in the basic PSO algorithm, namely, producing particles' positions and velocities, velocity update, and finally, position update.[31] 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 the following equation:

$$x_i^0 = x_{\min} + \text{rand}\,(x_{\max} - x_{\min}) \tag{34}$$

$$v_i^0 = x_{\min} + \text{rand}\,(x_{\max} - x_{\min}) \tag{35}$$

In Eqs. (34) and (35), the subscript and superscript denote the ith 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:

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot \text{rand} \cdot \left(pbest - x_i^k\right) + c_2 \cdot \text{rand} \cdot \left(gbest - x_i^k\right) \tag{36}$$

Three weight factors, namely, inertia factor, w, self-confidence factor, c_1, and swarm confidence factor, c_2, are incorporated in Eq. (36) to effect the particles direction. The following inertia weight is used:[32]

$$w = w_{\max} - (w_{\max} - w_{\min})\,k/k_{\max} \tag{37}$$

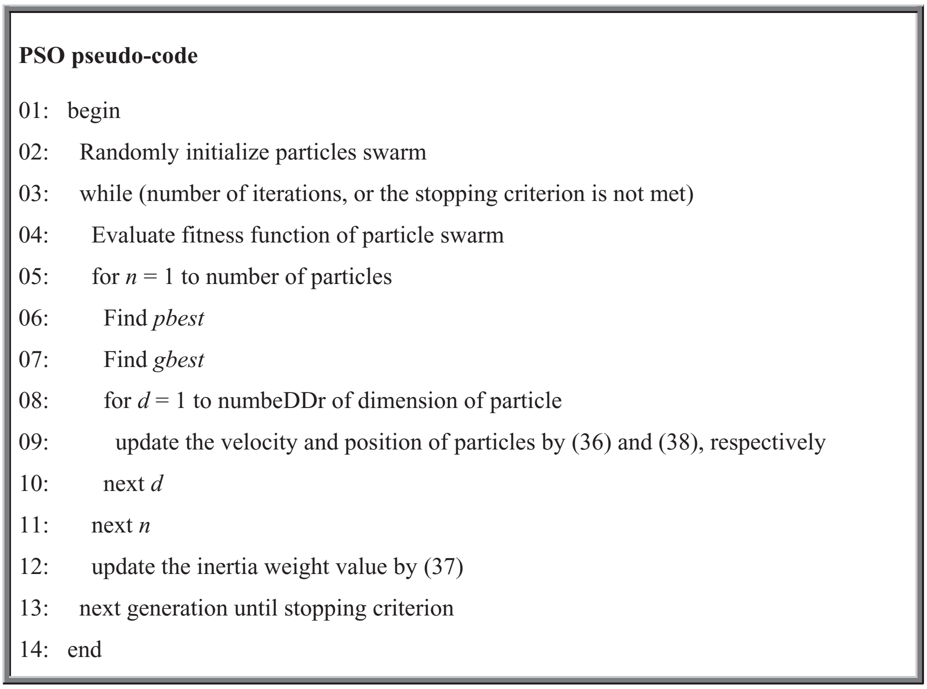
where k and $k_{\max}$ are the current number of iterations and the maximum number of iterations, respectively. $w_{\max}$ and $w_{\min}$ are the maximum and minimum weights, respectively. Appropriate values for $w_{\min}$ and $w_{\max}$ are 0.4 and 0.9, respectively.[33]

Finally, velocity vector is used to update the position of each particle, as shown in the following equation and illustrated in Fig. 3:

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{38}$$

Repeat the three steps of (i) velocity update, (ii) position update, and (iii) fitness calculations until a stopping criterion is reached. The pseudo-code of the PSO algorithm is shown in Fig. 4.

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press



PSO pseudo-code

```
begin
  Randomly initialize particles swarm
  while (number of iterations, or the stopping criterion is not met)
    Evaluate fitness function of particle swarm
    for $n$ = 1 to number of particles
      Find $pbest$
      Find $gbest$
      for $d$ = 1 to numbeDDr of dimension of particle
        update the velocity and position of particles by (36) and (38), respectively
      next $d$
    next $n$
    update the inertia weight value by (37)
  next generation until stopping criterion
end
```

Fig. 4. The pseudo-code of the PSO algorithm.

4.2. Optimal backstepping control system

The dynamic model in Eq. (14) can be divided into two subsystems Π_1 and Π_2, listed as follows:

$$\Pi_1 : \begin{cases} \ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{I_{yy}-I_{zz}}{I_{xx}}\right) - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_d + \frac{l}{I_{xx}}u_2 \\ \ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{I_{zz}-I_{xx}}{I_{yy}}\right) + \frac{J_r}{I_{yy}}\dot{\phi}\Omega_d + \frac{l}{I_{yy}}u_3 \\ \ddot{\psi} = \dot{\theta}\dot{\phi}\left(\frac{I_{xx}-I_{yy}}{I_{zz}}\right) + \frac{1}{I_{zz}}u_4 \end{cases} \tag{39}$$

$$\Pi_2 : \begin{cases} \ddot{x} = \left(c_\phi s_\theta c_\psi + s_\phi s_\psi\right)\frac{1}{m}u_1 \\ \ddot{y} = \left(c_\phi s_\theta s_\psi - s_\phi c_\psi\right)\frac{1}{m}u_1 \\ \ddot{z} = -g + \left(c_\phi c_\theta\right)\frac{1}{m}u_1 \end{cases} \tag{40}$$

Π_1 in Eq. (39) represents the rotation subsystem related with the dynamics of quadrotor roll motion ϕ, pitch motion θ, and yaw motion ψ. Π_2 in Eq. (40) represents the position subsystem related with the dynamics of quadrotor longitude motion x, latitude motion y, and altitude motion z. Hence, the control scheme advocated for the overall system is then logically divided into a rotation controller and a position controller.

In the previous section, a controller (32) has been designed to stabilize one DOF of the overall system. The coefficients k_1 and k_2 are control parameters and need to be positive to satisfy stability criteria. In conventional backstepping method, these parameters are selected by trial and error. It is also possible if the parameters are properly chosen, but it cannot be said that the optimal parameters are selected. To overcome this drawback, the PSO is used off-line for determining the optimal value of the backstepping control parameters. The performance of the controller varies according to adjusted parameters. Since the optimal backstepping control (OBC) aims to improve the control performance yielded by a backstepping controller, it keeps the simple structure of the backstepping controller. As aforementioned, both the rotation and position subsystems are comprised of three DOF. Then, there are in sum six control parameters that need to be selected simultaneously for each subsystem.

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

In the present study, an integral absolute error (IAE) is utilized to judge the performance of the controller. IAE criterion is widely adopted to evaluate the control system dynamic performance.[34] The index IAE is defined as follows:

$$\mathrm{IAE} = \int_0^t |e(t)|\, dt \tag{41}$$

Since the system is comprised of two subsystems, a vector IAE for the rotation subsystem is taken as $\mathrm{IAE}_R = [\mathrm{IAE}_\phi \;\; \mathrm{IAE}_\theta \;\; \mathrm{IAE}_\psi]$, where the subscripts are denoted for roll, pitch, and yaw, respectively. Meanwhile, a vector IAE for the position subsystem is taken as $\mathrm{IAE}_P = [\mathrm{IAE}_x \;\; \mathrm{IAE}_y \;\; \mathrm{IAE}_z]$, where the subscripts are denoted for longitude, latitude, and altitude, respectively.

For the rotation controller, the PSO algorithm is utilized to minimize the fitness function J_R, expressed as

$$J_R = \mathrm{IAE}_R \cdot W \tag{42}$$

and for the position controller, the PSO algorithm is utilized to minimize the fitness function J_P, expressed as

$$J_P = \mathrm{IAE}_P \cdot W \tag{43}$$

where $W = [W_1 \;\; W_2 \;\; W_3]^T$ 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. In this case, equal weights for the three objectives to be met by the each controller are considered as such the minimizations of the error indexes are equally important. 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 Fig. 5.

5. Backstepping Sliding Mode Control System

The optimized IBC (32) effort is unrealizable if an unknown external disturbance is present in the system. Thus, auxiliary control effort is designed to attenuate the effect of the unknown external disturbance. The auxiliary control effort is referred as switching control effort represented by u_{sw}. The switching control effort is designed such that the system state trajectories are forced toward the sliding surface and stay on it. This effort is known as sliding mode.

Procedures to design the BSMC can be described by the following steps:

Step 1: Similar as *Step 1* in the design of IBC.

Step 2: Define a sliding surface in terms of the error such as

$$s = e_2 = \dot{x}_1 - \dot{x}_{1d} - k_1 e_1 \tag{44}$$

Thus, the Lyapunov function (30) can be written as

$$V_2 = \frac{1}{2}e_1^2 + \frac{1}{2}s^2 \tag{45}$$

Differentiating Eq. (45) w.r.t. time leads to

$$\begin{aligned} \dot{V}_2 &= e_1\dot{e}_1 + s\dot{s} \\ &= e_1(-e_2 - k_1 e_1) + s(f(x_1) + g(x_1)u + \delta_1 - \ddot{x}_{1d} - k_1\dot{e}_1) \\ &= -k_1 e_1^2 + s(-e_1 + f(x_1) + g(x_1)u + \delta_1 - \ddot{x}_{1d} - k_1\dot{e}_1) \end{aligned} \tag{46}$$

Step 3: Since the external disturbance is unknown, a backstepping control can be obtained as

$$u_B = \frac{1}{g(x_1)}(e_1 + k_1\dot{e}_1 + \ddot{x}_{1d} - f(x_1) - k_2 s) \tag{47}$$

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

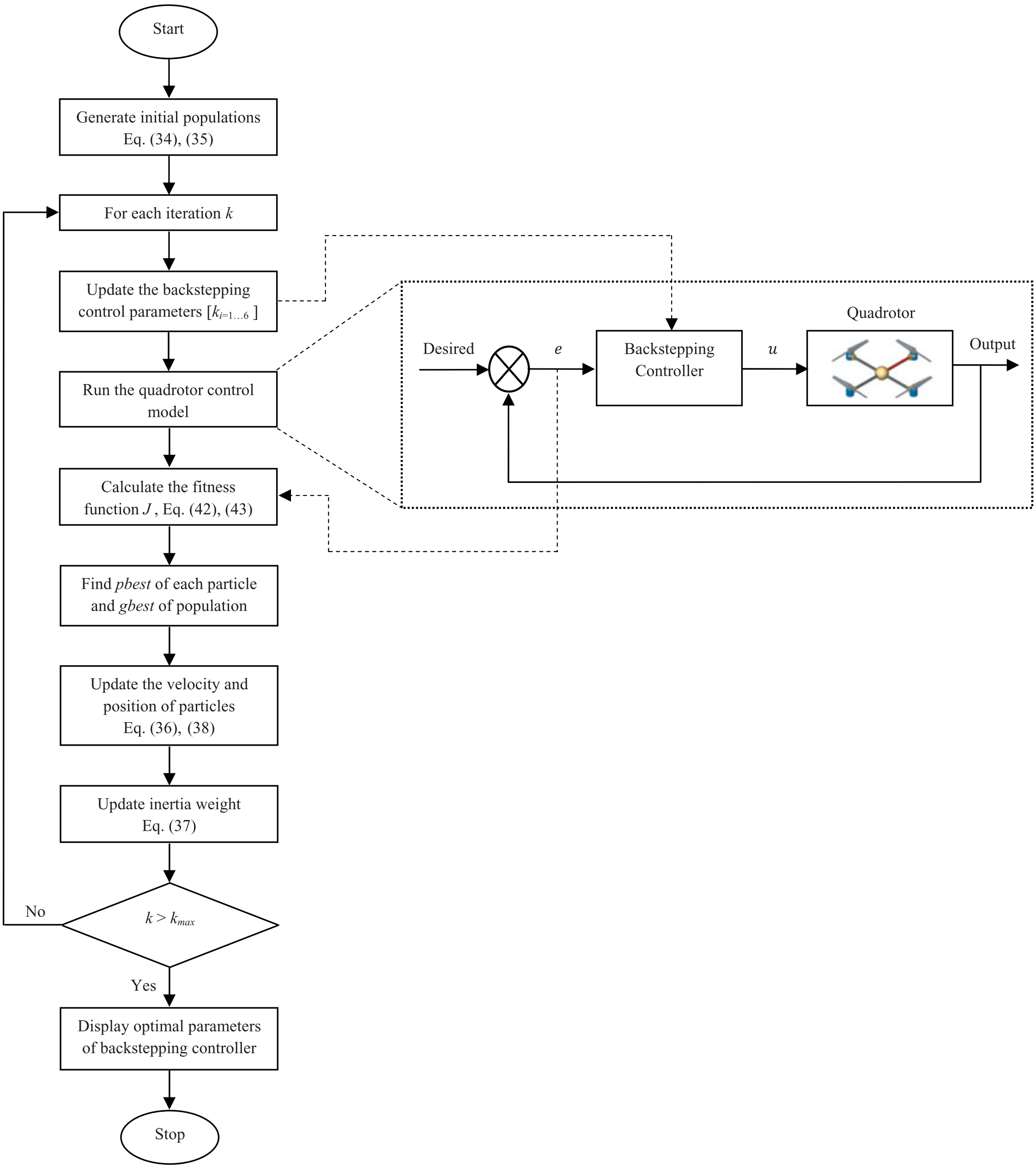

Start
Generate initial populations Eq. (34), (35)
For each iteration k
Update the backstepping control parameters [$k_{i=1...6}$]
Run the quadrotor control model
Calculate the fitness function J, Eq. (42), (43)
Find *pbest* of each particle and *gbest* of population
Update the velocity and position of particles Eq. (36), (38)
Update inertia weight Eq. (37)
$k > k_{max}$
No
Yes
Display optimal parameters of backstepping controller
Stop
Desired
e
Backstepping Controller
u
Quadrotor
Output

Fig. 5. The flowchart of the PSO-based backstepping control parameters optimization.

Step 4: Define the switching control signal such as

$$u_{sw} = \lambda \text{ sign } (s) \tag{48}$$

where λ is a constant determined by design parameter and sign(s) is a sign function:

$$\text{sign } (s) = \begin{cases} 1, & s/\rho > 0 \\ -1, & s/\rho < 0 \end{cases} \tag{49}$$

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

Totally, the robust BSMC law for nonlinear systems with the presence of external disturbance, which guarantees stability and convergence, can be represented as

$$\begin{aligned} u &= u_B + u_{sw} \\ &= \frac{1}{g(x_1)}\left(e_1 + k_1\dot{e}_1 + \ddot{x}_{1d} - f(x_1) - k_2 s\right) + \frac{\varepsilon_1}{g(x_1)}\operatorname{sign}(s) \end{aligned} \tag{50}$$

where ε_1 is design parameter to be determined later.

The utilization of discontinuous switching function will excite undesired phenomenon called chatter. In order to eliminate the chattering problem, a saturation function $\operatorname{sat}(s/\rho)$ is utilized. Thus, replacing $\operatorname{sign}(s)$ by $\operatorname{sat}(s/\rho)$ in Eq. (50) implies

$$u = \frac{1}{g(x_1)}\left(e_1 + k_1\dot{e}_1 + \ddot{x}_{1d} - f(x_1) - k_2 s + \varepsilon_1 \operatorname{sat}(s/\rho)\right) \tag{51}$$

The saturation function $\operatorname{sat}(s/\rho)$ is defined as follows:

$$\operatorname{sat}(s/\rho) = \begin{cases} \operatorname{sign}(s/\rho), & |s/\rho| > 1 \\ s/\rho, & |s/\rho| \le 1 \end{cases} \quad @ \quad \operatorname{sat}(s/\rho) = \begin{cases} 1, & s/\rho > 1 \\ -1, & s/\rho < -1 \\ s/\rho, & |s/\rho| \le 1 \end{cases} \tag{52}$$

where ρ is boundary layer around the sliding surface s and its value is a small positive constant.

The BSMC law for one DOF of quadrotor nonlinear systems with the presence of external disturbance, which guarantees the stability and convergence, can be generally represented as

$$\begin{aligned} u_i &= u_{i(B)} + u_{i(sw)} \\ u_1 &= \frac{1}{g(x_1)}\left(e_1 + k_1\dot{e}_1 + \ddot{x}_{1d} - f(x_1) - k_2 s_1 + \varepsilon_1 \operatorname{sat}(s_1/\rho_1)\right); \quad (\text{for } i = 1) \end{aligned} \tag{53}$$

The control laws for the other states or DOF can be obtained by performing the same steps.

Finally, the control inputs u_1, u_2, u_3, and u_4 for the quadrotor dynamic systems can be defined as follows:

$$\begin{aligned} u_1 &= \frac{m}{c_\phi c_\theta}\left(e_1 + k_1\dot{e}_1 + \ddot{x}_{1d} + g - k_2 s_1 + \varepsilon_1 \operatorname{sat}(s_1/\rho_1)\right) \\ u_2 &= \frac{1}{b_1}\left(e_3 + k_3\dot{e}_3 + \ddot{x}_{3d} - \dot{\theta}\dot{\psi}a_1 + \dot{\theta}a_2\Omega_d - k_4 s_2 + \varepsilon_2 \operatorname{sat}(s_2/\rho_2)\right) \\ u_3 &= \frac{1}{b_2}\left(e_5 + k_5\dot{e}_5 + \ddot{x}_{5d} - \dot{\phi}\dot{\psi}a_3 - \dot{\phi}a_4\Omega_d - k_6 s_3 + \varepsilon_3 \operatorname{sat}(s_3/\rho_3)\right) \\ u_4 &= \frac{1}{b_3}\left(e_7 + k_7\dot{e}_7 + \ddot{x}_{7d} - \dot{\theta}\dot{\phi}a_5 - k_8 s_4 + \varepsilon_4 \operatorname{sat}(s_4/\rho_4)\right) \end{aligned} \tag{54}$$

Theorem: For the nonlinear dynamic equation of quadrotor with external disturbance as represented by Eq. (17), if the control law in Eq. (51) is applied, the system will be asymptotically stable.

Proof: Substituting the BSMC law from Eq. (51), then Eq. (46) becomes

$$\begin{aligned} \dot{V}_2 &= -k_1 e_1^2 - k_2 s^2 + s\left(\varepsilon_1 \operatorname{sat}(s/\rho) + \delta_1\right) \\ &= -k_1 e_1^2 - k_2 s^2 + \varepsilon_1 |s| + \delta_1 s \end{aligned} \tag{55}$$

From the Lyapunov theorem, if $\dot{V}_2$ is negative definite, the system trajectory will be driven and attracted toward the sliding surface, and remain sliding on it until the origin is reached asymptotically. Hence, the design parameter should be chosen in such a way that $\dot{V}_2 < 0$ is always satisfied. Let us assume that δ_1 is bounded with β. So, by choosing $-\varepsilon_1 \ge \beta$, $\dot{V}_2 < 0$ can be guaranteed.

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

6. Adaptive Backstepping Sliding Mode Control System

In practical application, the upper bound of external disturbances, which is required in the conventional BSMC system to guarantee the system stability, is difficult to determine precisely *in priori*. Hence, an adaptation scheme is proposed to adapt the design controller parameters. Consequently, knowledge on the bound of external disturbances is not needed in advance. To derive the adaptation law, the following Lyapunov function is defined:

$$V_3 = V_2 + \frac{1}{2}\gamma\tilde{\varepsilon}_1^2 \tag{56}$$

where $\tilde{\varepsilon}_1 = \hat{\varepsilon}_1 - \varepsilon_1$ and γ is gain adaptation.

Differentiating Eq. (56) w.r.t. time yields

$$\dot{V}_3 = \dot{V}_2 + \gamma\tilde{\varepsilon}_1\left(-\dot{\varepsilon}_1\right) \tag{57}$$

Substituting Eq. (46) into Eq. (57), it is obtained that

$$\dot{V}_3 = -k_1e_1^2 + s\left(-e_1 + f\left(x_1\right) + g\left(x_1\right)u + \delta_1 - \ddot{x}_{1d} - k_1\dot{e}_1\right) - \gamma\tilde{\varepsilon}_1\,\dot{\varepsilon}_1 \tag{58}$$

By utilizing Eq. (51) and substituting Eq. (55) into Eq. (58), it is obtained that

$$\begin{aligned}\dot{V}_3 &= -k_1e_1^2 - k_2s^2 + \varepsilon_1\left|s\right| + \delta_1 s - \gamma\tilde{\varepsilon}_1\,\dot{\varepsilon}_1\\ &= -k_1e_1^2 - k_2s^2 + \left(\hat{\varepsilon}_1 - \tilde{\varepsilon}_1\right)\left|s\right| + \beta\left|s\right| - \gamma\tilde{\varepsilon}_1\,\dot{\varepsilon}_1\\ &= -k_1e_1^2 - k_2s^2 - \tilde{\varepsilon}_1\left|s\right| + \left(\hat{\varepsilon}_1 + \beta\right)\left|s\right| - \gamma\tilde{\varepsilon}_1\,\dot{\varepsilon}_1\\ &= -k_1e_1^2 - k_2s^2 + \left(\hat{\varepsilon}_1 + \beta\right)\left|s\right| - \tilde{\varepsilon}_1\left(\gamma\ \ + \left|s\right|\right)\end{aligned} \tag{59}$$

Note that the value of $\hat{\varepsilon}_1 + \beta$ can remain negative (i.e., $\hat{\varepsilon}_1 + \beta = -\tau$, where $\tau > 0$) by assuming $-\hat{\varepsilon}_1 \geq \beta$. By considering that $\gamma\dot{\varepsilon}_1 + |s| = 0$, thus, the adaptation law is designed as

$$\dot{\varepsilon}_1 = -\frac{1}{\gamma}\left|s\right| \tag{60}$$

Then, it can be verified that

$$\dot{V}_3 = -k_1e_1^2 - k_2s^2 - \tau\left|s\right| \leq 0 \tag{61}$$

The system is negative semi-definite. Therefore, the proposed adaptive BSMC can guarantee the stability of the system even with external disturbance.

The configuration of the proposed control system is depicted in Fig. 6.

7. Simulation Results

In this section, the performance of the proposed approach is evaluated. The corresponding algorithm is implemented in MATLAB/SIMULINK simulation environment. The model parameter values of the quadrotor system are taken from ref. [35] and listed in Table I. Initially, the controller parameter optimization is searched with the quadrotor control model, and then the identified parameter values are transferred to the controller in the quadrotor system developed in MATLAB/SIMULINK for further evaluation.

In this study, the following values are assigned for controller parameters optimization:

1. Dimension of the search space = 6 (i.e., $k_{i=1\ldots6}$ or $k_{i=7\ldots12}$);
2. Population/Swarm size = 15;
3. The number of maximum iteration = 20;
4. The self and swarm confident factors, c_1 and $c_2 = 2$;
5. The inertia weight factor w is set by Eq. (37), where $w_{\max} = 0.9$ and $w_{\min} = 0.4$;
6. The searching ranges for the backstepping parameters are limited to $[0,\ 20]$;

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

Table I. 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 • m^2
I_{yy}	Pitch inertia	4.85×10^{-3}	kg • m^2
I_{zz}	Yaw inertia	8.81×10^{-3}	kg • m^2
B	Thrust factor	2.92×10^{-6}	
D	Drag factor	1.12×10^{-7}	

Table II. The rotation controller parameters and fitness value of each optimal particle.

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

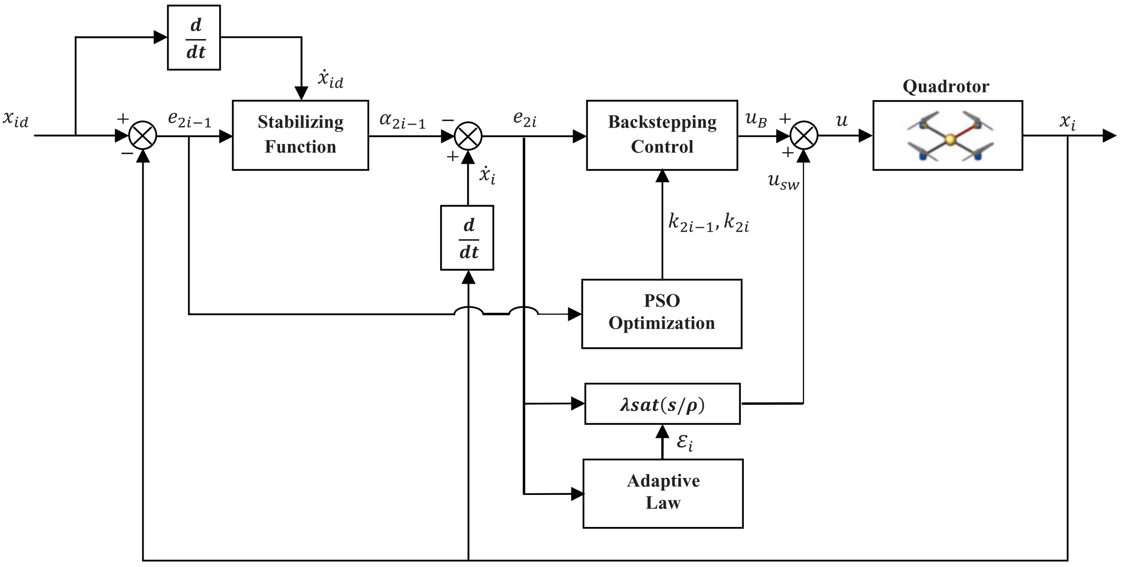


Fig. 6. Block diagram of the robust optimal backstepping control system.

7. The simulation time, t, is equal to 10 s;
8. Optimization process is repeated 20 times.

The finest set of values among the simulation runs is selected as the best optimized controller value. The parameter and fitness values of each particle during the simulation for the rotation and position controller are summarized in Tables II and III, respectively. For the rotation controller, the

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

Table III. The position controller parameters and fitness value of each optimal particle.

Iteration No.	Optimal parameters	Fitness value
1	$k_7 = 14.85,\ k_8 = 13.32$ $k_9 = 14.45,\ k_{10} = 15.20$ $k_{11} = 15.39,\ k_{12} = 16.68$	0.1922
5	$k_7 = 15.45,\ k_8 = 14.74$ $k_9 = 15.84,\ k_{10} = 16.00$ $k_{11} = 15.33,\ k_{12} = 16.00$	0.1741
9	$k_7 = 15.21,\ k_8 = 14.29$ $k_9 = 15.01,\ k_{10} = 14.75$ $k_{11} = 15.42,\ k_{12} = 14.95$	0.1543
20	$k_7 = 15.21,\ k_8 = 14.29$ $k_9 = 15.01,\ k_{10} = 14.75$ $k_{11} = 15.42,\ k_{12} = 14.95$	0.1543

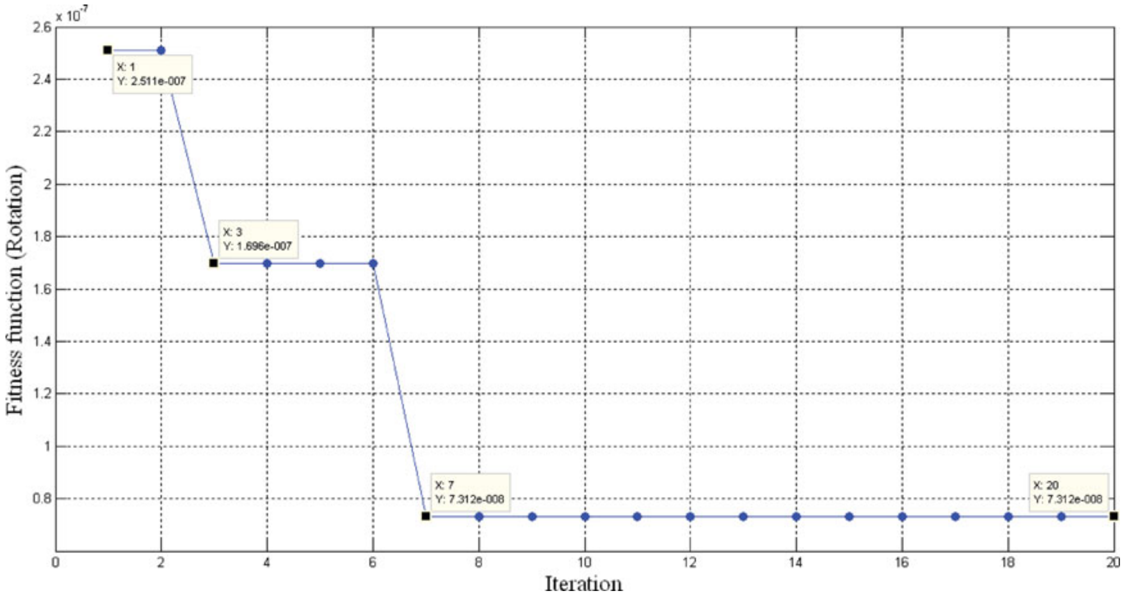


Fig. 7. The convergence of fitness function for rotation controller with number of iterations.

best fitness value is 7.312e − 008 appearing in iteration number 7, and the optimal parameters are $k_1 = 14.64$, $k_2 = 14.14$, $k_3 = 14.38$, $k_4 = 14.21$, $k_5 = 14.61$, and $k_6 = 14.11$. The variation of the fitness function with number of iterations is shown in Fig. 7. Meanwhile, the variations of backstepping control parameters w.r.t. the number of iterations are shown in Fig. 8. For the position controller, the best fitness value is 0.1543 appearing in iteration number 9, and the optimal parameters are $k_7 = 15.21$, $k_8 = 14.29$, $k_9 = 15.01$, $k_{10} = 14.75$, $k_{11} = 15.42$, and $k_{12} = 14.95$. The variation of the fitness function with number of iterations is shown in Fig. 9. Meanwhile, the variations of backstepping control parameters w.r.t. the number of iterations are shown in Fig. 10. As can be seen for both control parameters optimization, 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.

To explore the effectiveness of the proposed ABSMC, three simulation experiments have been performed on the quadrotor. In the first experiment, the designed controller is implemented in a stabilizing problem. In the second, the scheme is used to investigate the altitude tracking performance. Finally, the *x*–*y* position control problem is carried out in order to demonstrate the effectiveness of the designed controller. The nominal case (Case 1) and the external disturbance case (Case 2) are

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

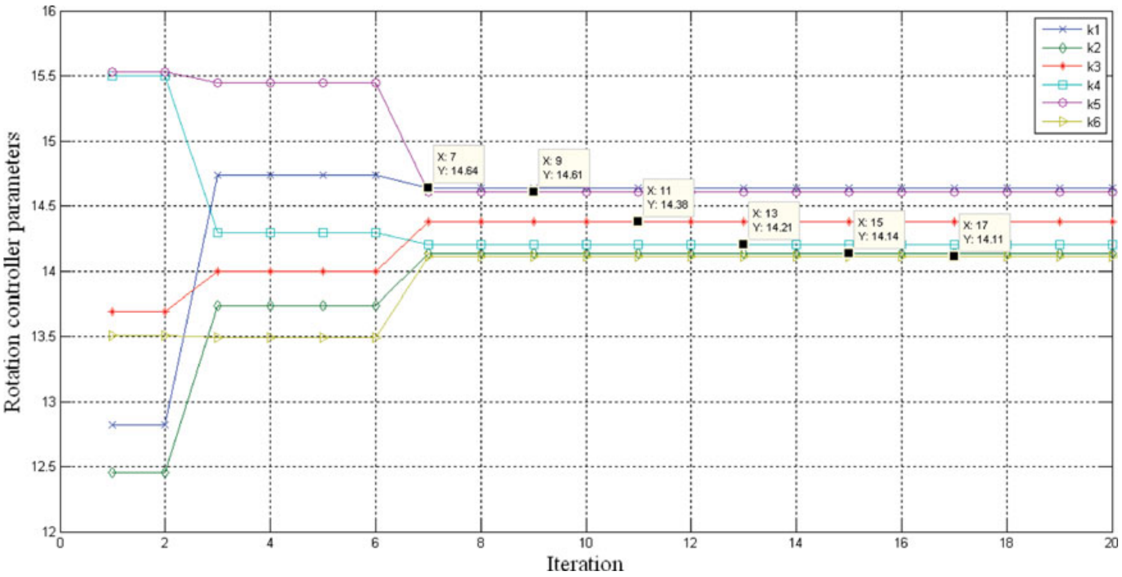


Fig. 8. The variations of rotation controller parameters versus number of iterations.

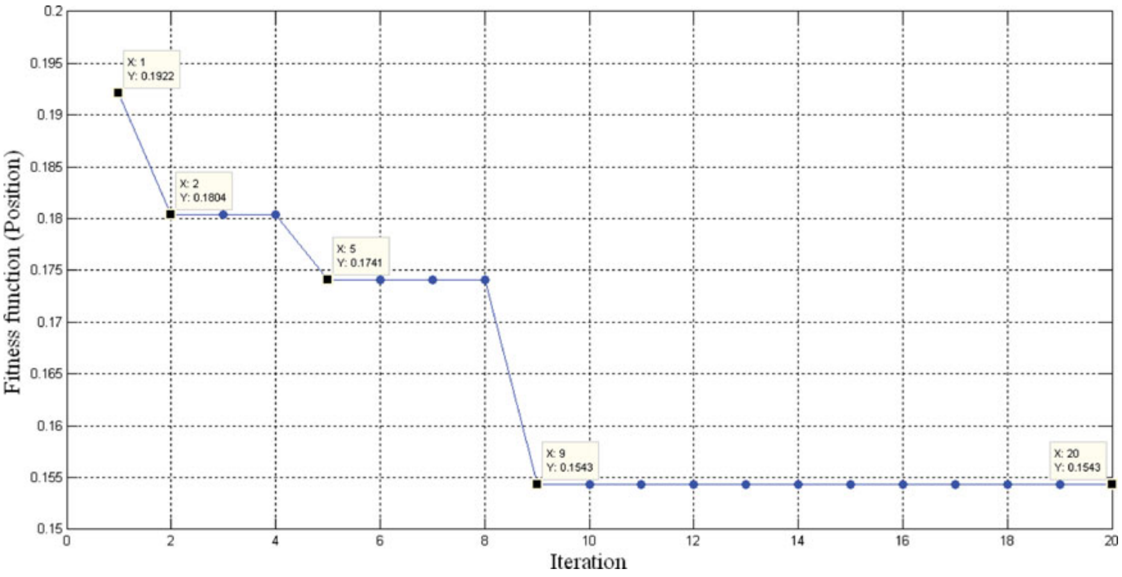


Fig. 9. The convergence of fitness function for position controller with number of iterations.

provided as a test conditions. The external disturbance is generated as forces produced by wind that is modeled using the Dryden wind-gust model.[36] The wind force profile which externally disturbs the quadrotor in the vertical, lateral, and longitudinal direction is shown in Fig. 11.

7.1. Simulation experiment 1: Altitude & attitude stabilization

In this simulation experiment, the control objective is to regulate a quadrotor at a certain desired altitude/attitude, such that the quadrotor can hover at a fixed point. The desired altitude/attitude is given by $x_{id} = [z_d, \phi_d, \theta_d, \psi_d] = [5,\ 0,\ 0,\ 0]^T$. The initial states are given by $z = 5$, $\phi = 0.2$, $\theta = 0.2$, and $\psi = 0.2$. In the simulation, first, the OBC system is considered. The simulation results for stabilizing a quadrotor at Case 1 and Case 2 using the OBC system are depicted in Figs. 12 and 13, respectively. From the simulation results, the OBC system is able to stabilize the quadrotor in hover mode at Case 1. However, the degenerate performance response shown in Fig. 13 is due to the occurrence of external disturbance. The same condition is used to simulate the ABSMC system.

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

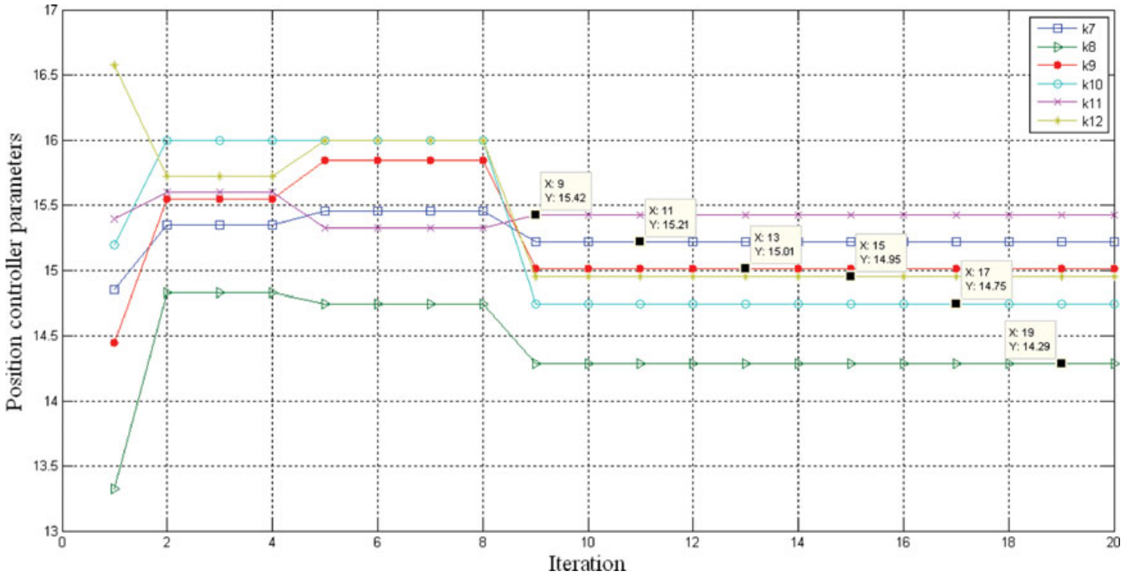


Fig. 10. The variations of position controller parameters versus number of iterations.

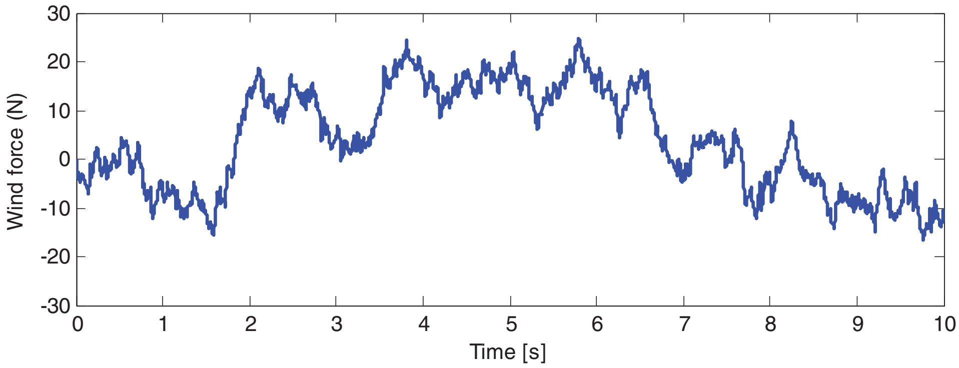


Fig. 11. The external disturbance acting on the quadrotor in the vertical, lateral, and longitudinal direction.

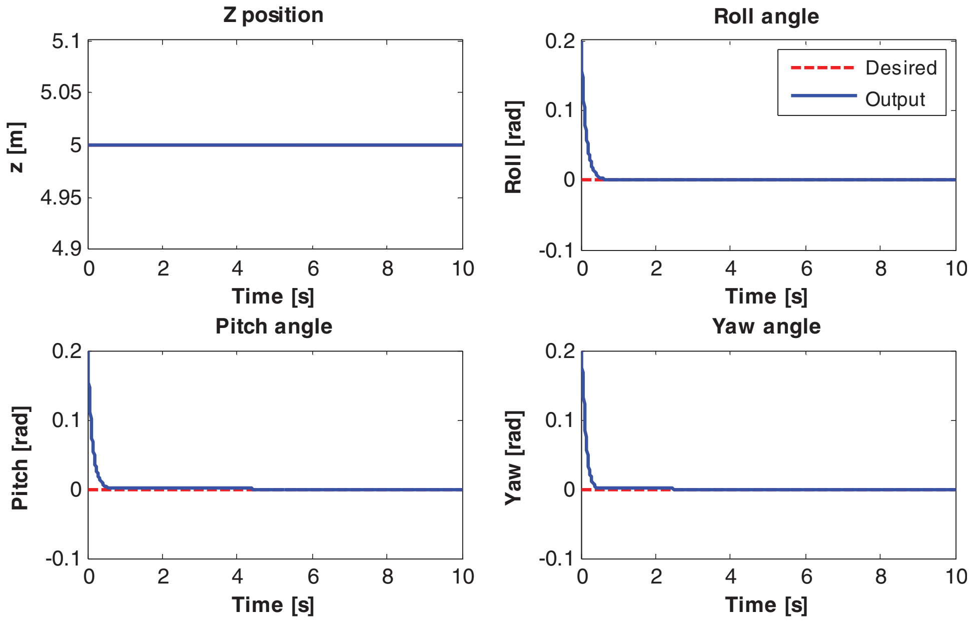


Fig. 12. Altitude/Attitude of the hovering quadrotor using OBC at Case 1.

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

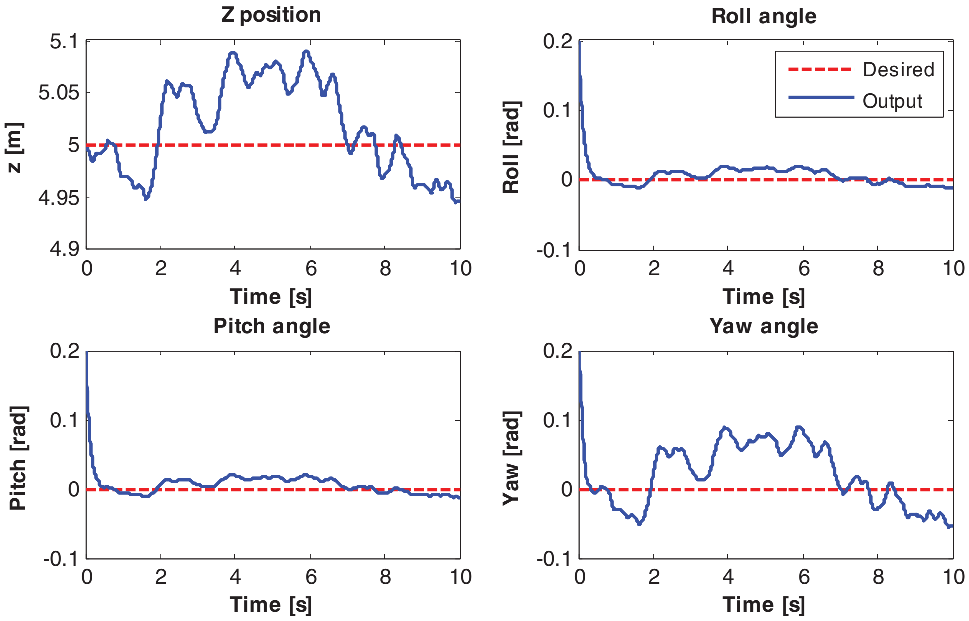


Fig. 13. Altitude/Attitude of the hovering quadrotor using OBC at Case 2.

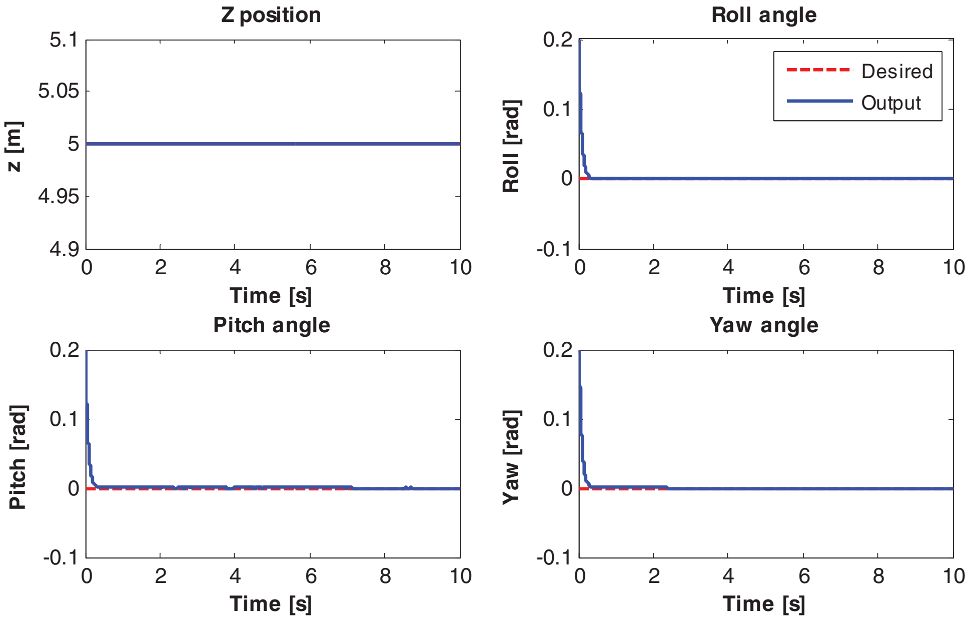


Fig. 14. Altitude/Attitude of the hovering quadrotor using ABSMC at Case 1.

The simulation results for stabilizing a quadrotor at Case 1 and Case 2 are depicted in Figs. 14 and 15, respectively. From the simulation results, the robustness of the proposed ABSMC system in the quadrotor stabilization can be noted. It can be clearly seen that the altitude/attitude of the quadrotor can be maintained at the desired altitude/attitude, that is, the hovering flight is stable even when the external disturbance is exerted, as shown in Fig. 15. The control inputs for ABSMC at Case 1 and 2 are shown in Figs. 16 and 17, respectively. For quantitative comparison between two methods, IAE is used as the criterion. Table IV shows the IAE values of the simulation results using the OBC and ABSMC approach. From the comparison, it can be seen that the performance is improved in all the cases when using the ABSMC as compared to the OBC.

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

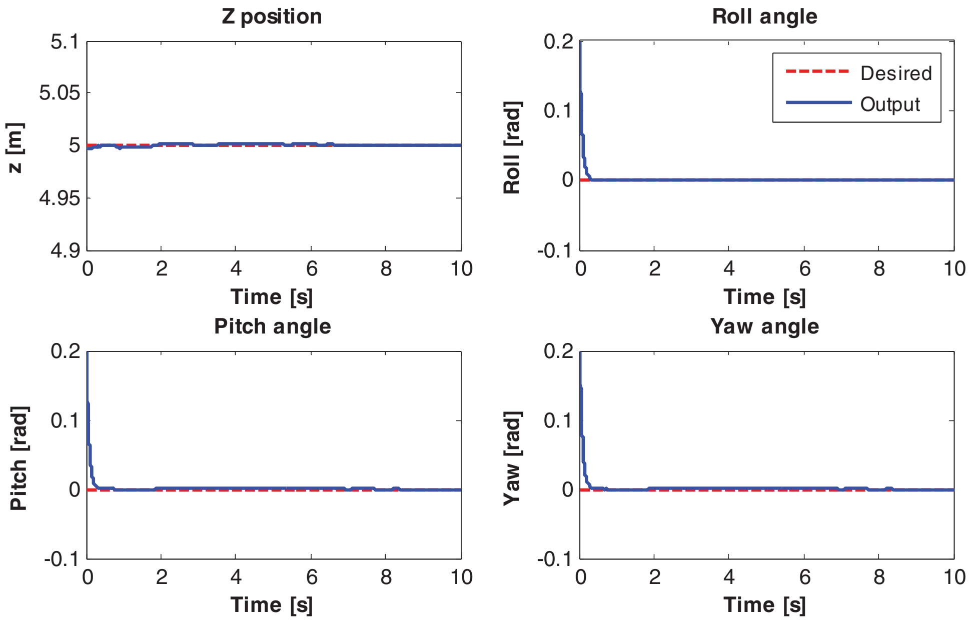


Fig. 15. Altitude/Attitude of the hovering quadrotor using ABSMC at Case 2.

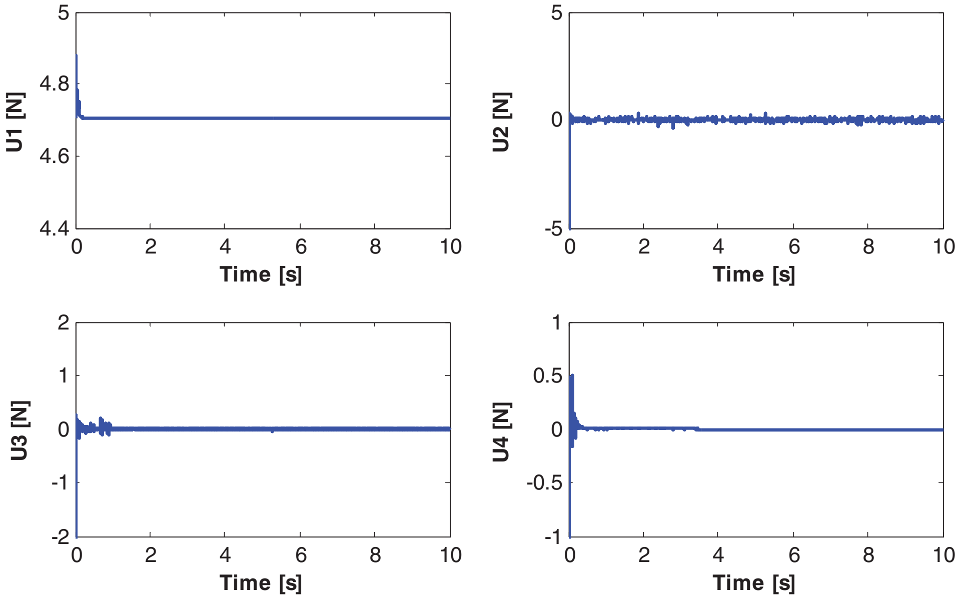


Fig. 16. Control inputs of the hovering quadrotor using ABSMC at Case 1.

7.2. Simulation experiment 2: Automatic take-off, hovering, and landing mission (altitude tracking)

In this simulation experiment, the control objective is to ensure that the quadrotor can perform a smooth automatic take-off, hovering, and landing missions. In this mission, the quadrotor is required to climb up to a position with a 1m height, hover and then land on the ground, which is defined by trapezoidal trajectory. First, the OBC system is simulated. The tracking responses at the nominal condition (Case 1) and external disturbance condition (Case 2) are depicted in Figs. 18 and 19, respectively. From the simulation results, the quadrotor can track the desired trajectory at Case 1 with having a small overshoot by the OBC system, as shown in Fig. 18. However, the degenerate performance response shown in Fig. 19 is due to the occurrence of external disturbance. Under the same simulation condition, the ABSMC system is simulated. The simulation results for tracking a specified trajectory at the nominal

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

Table IV. Quantitative comparison between OBC and ABSMC for stabilization problem.

	IAE							
	Case 1				Case 2			
Control scheme	z	ϕ	θ	ψ	z	ϕ	θ	ψ
OBC	1.207e−4	2.701e−2	2.610e−2	2.655e−2	4.012e−1	1.147e−1	1.136e−1	4.208e−1
ABSMC	5.987e−5	1.595e−2	1.583e−2	1.790e−2	8.738e−3	1.826e−2	1.819e−2	2.318e−2

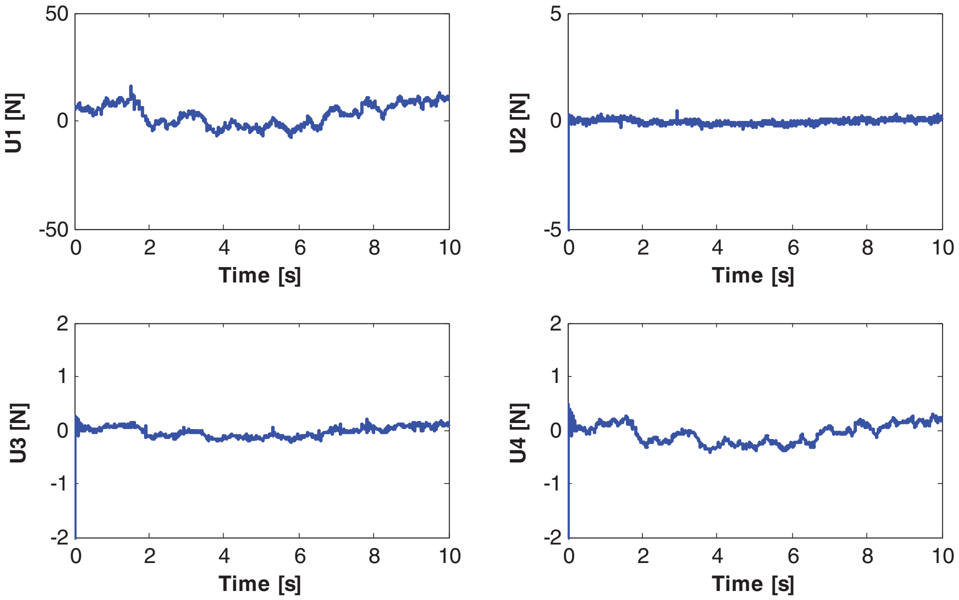


Fig. 17. Control inputs of the hovering quadrotor using ABSMC at Case 2.

and external disturbance condition are shown in Figs. 20 and 21, respectively. From Fig. 20, it can be seen that the quadrotor can smoothly track the trajectory. Meanwhile, the robustness of the proposed control scheme in tracking the desired trajectory can be ascertained, as illustrated in Fig. 21. The control inputs for ABSMC at Case 1 and Case 2 are shown in Figs. 22 and 23, respectively. For a quantitative comparison, IAE of two methods has been reported. According to Table V, it can be observed that the proposed ABSMC gives lower IAE value as compared to OBC.

7.3. Simulation experiment 3: Eight-shape trajectory tracking mission (x–y position tracking)

To further highlight the advantage of the proposed control structure, the OBC and ABSMC for the x–y position tracking are simulated. Considering that the turning curve maneuver is an important practical trajectory maneuver that the quadrotor needs to perform, the control performance of eight-shape trajectory tracking is examined. The desired trajectory is generated using the following command:

$$\begin{cases} x_d = 5\sin\left(\frac{2\pi}{5}t\right), \\ y_d = 5\left[1 - \cos\left(\frac{\pi}{5}t\right)\right] \end{cases} \tag{62}$$

The initial state of the quadrotor is set to be $[x_0, y_0] = [5, 5]$ m. The simulation results of the x–y position tracking for both OBC and ABSMC approach under the occurrence of external disturbance are, respectively, shown in Figs. 24 and 25. As it can be seen, the proposed control scheme can track the desired reference trajectory accurately despite external disturbance. It is obvious that the ABSMC

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

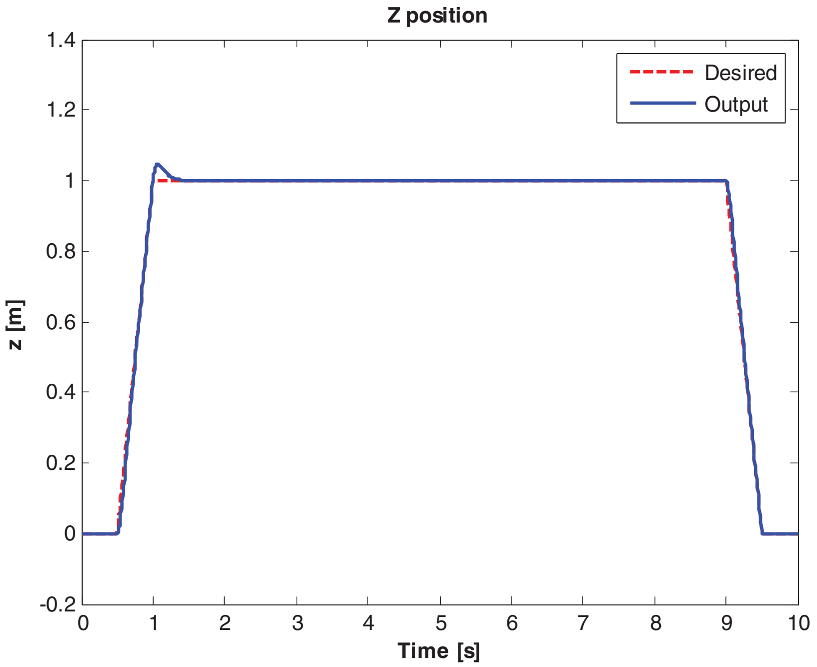


Fig. 18. Altitude tracking response for automatic take-off, hovering, and landing mission using OBC at Case 1.

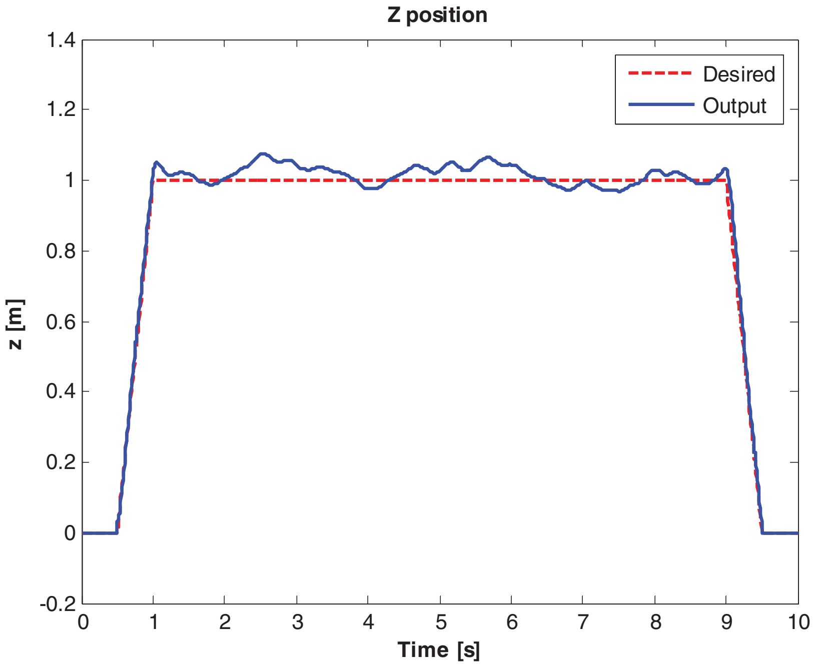


Fig. 19. Altitude tracking response for automatic take-off, hovering, and landing mission using OBC at Case 2.

can give small tracking error and good tracking performance compared to OBC. The control input for ABSMC at Case 2 is shown in Fig. 26. For a quantitative comparison, IAE of two methods has been reported. According to Table VI, it can be observed that the proposed ABSMC gives lower IAE value as compared to OBC. It is evident that the proposed method yields robust and superior control performances.

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

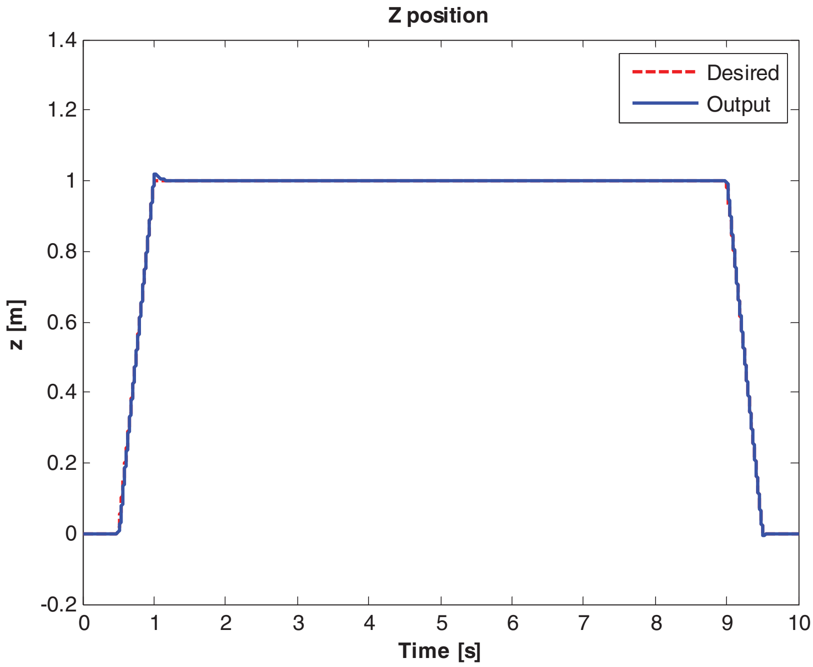


Fig. 20. Altitude tracking response for automatic take-off, hovering, and landing mission using ABSMC at Case 1.

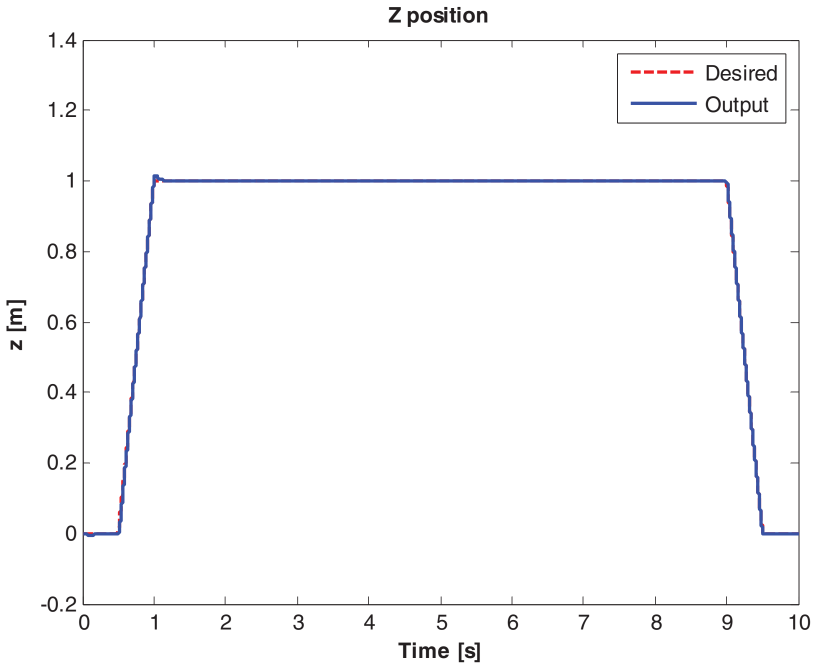


Fig. 21. Altitude tracking response for automatic take-off, hovering, and landing mission using ABSMC at Case 2.

8. Conclusions

In this paper, the application of adaptive BSMC system for maneuvering a quadrotor aerial robot perturbed by external disturbances is successfully demonstrated. First, a mathematical model of the quadrotor is introduced. Then, the proposed control system which combines a backstepping and sliding mode control theory is developed. An adaptation scheme is derived to adapt the control parameters. The PSO algorithm has been utilized to determine the parameters of the controller. The advantage of the controller is that the control strategy does not require any prior knowledge on the bound of

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

Table V. Quantitative comparison between OBC and ABSMC for altitude tracking problem.

	IAE	
Control scheme	Case 1	Case 2
OBC	1. 753e−2	2.556e−1
ABSMC	6.577e−3	1.023e−2

Table VI. Quantitative comparison between OBC and ABSMC for trajectory tracking problem.

Control scheme	IAE (Eight-shape trajectory at Case 2)
OBC	2.792
ABSMC	0.6784

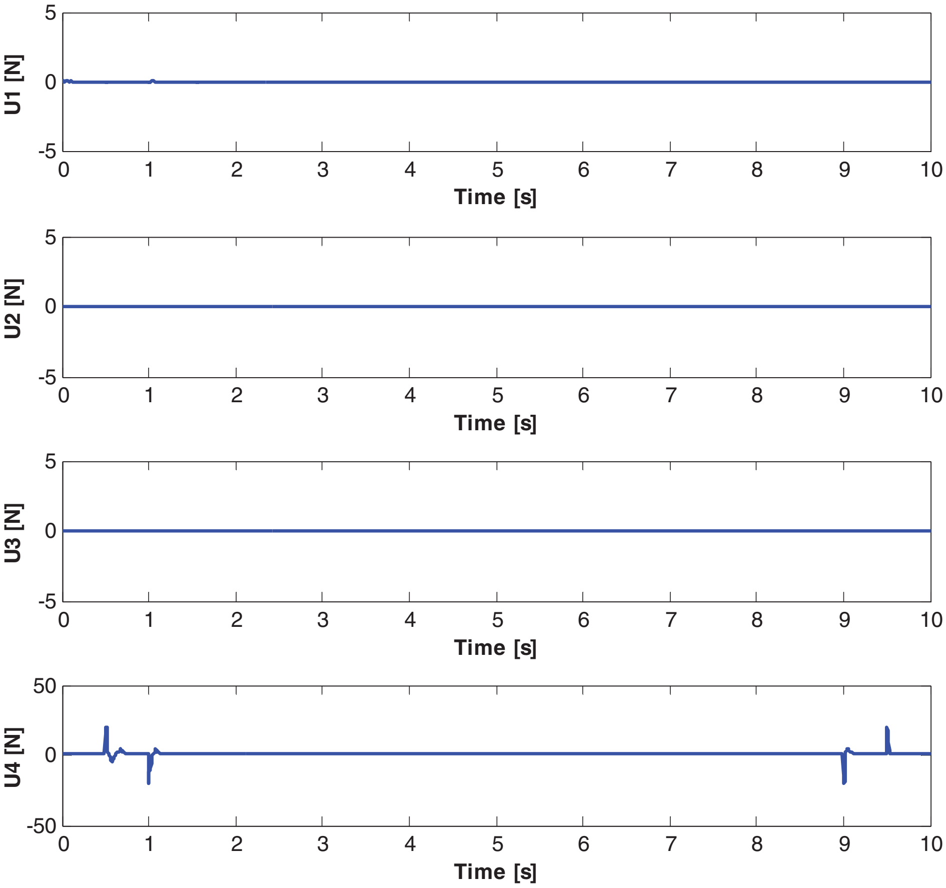


Fig. 22. Control inputs for automatic take-off, hovering, and landing mission using ABSMC at Case 1.

disturbances. Finally, the proposed control scheme is applied for a quadrotor aerial robot to achieve smooth automatic take-off and landing, hovering, and trajectory tracking missions. Simulation results have demonstrated the validity and the effectiveness of the proposed control system. In the future, the possibility of a real-time implementation of the designed controller will be further investigated.

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

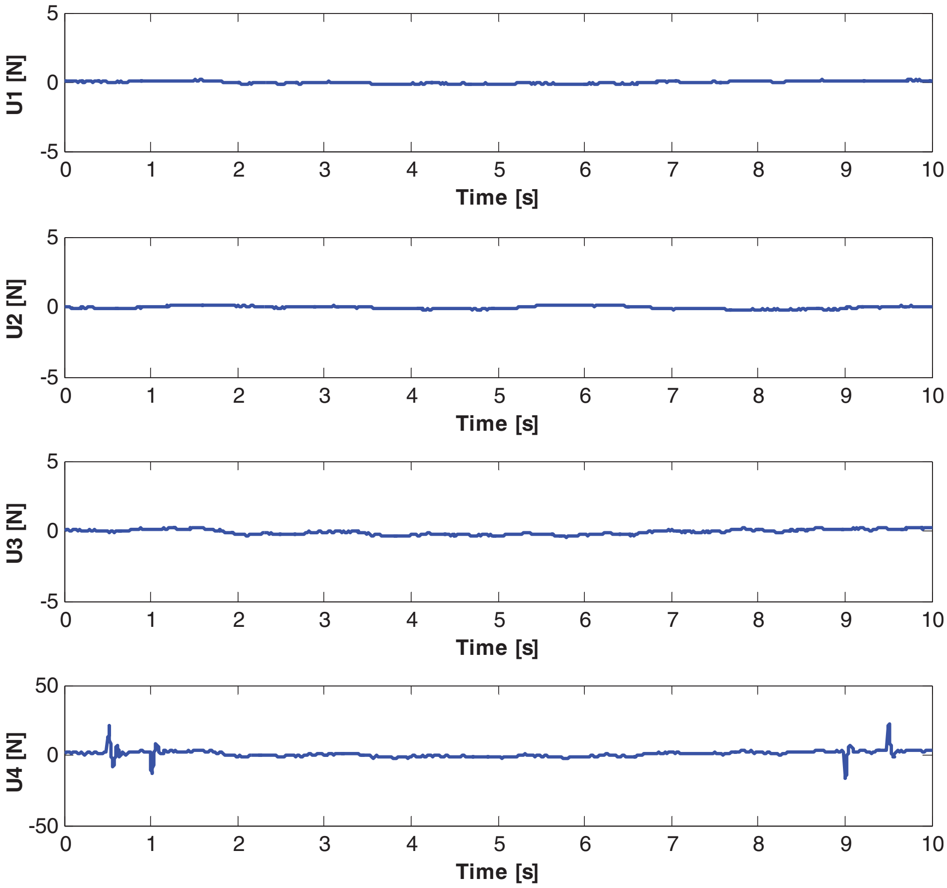


Fig. 23. Control inputs for automatic take-off, hovering, and landing mission using ABSMC at Case 2.

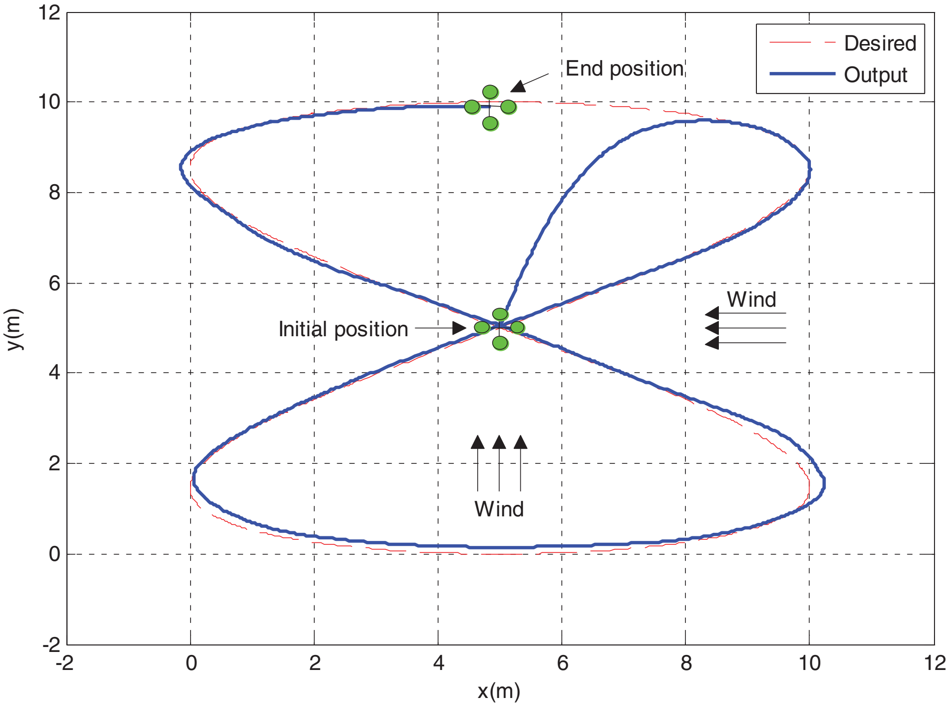


Fig. 24. *x–y* position eight-shape trajectory tracking response using OBC at Case 2.

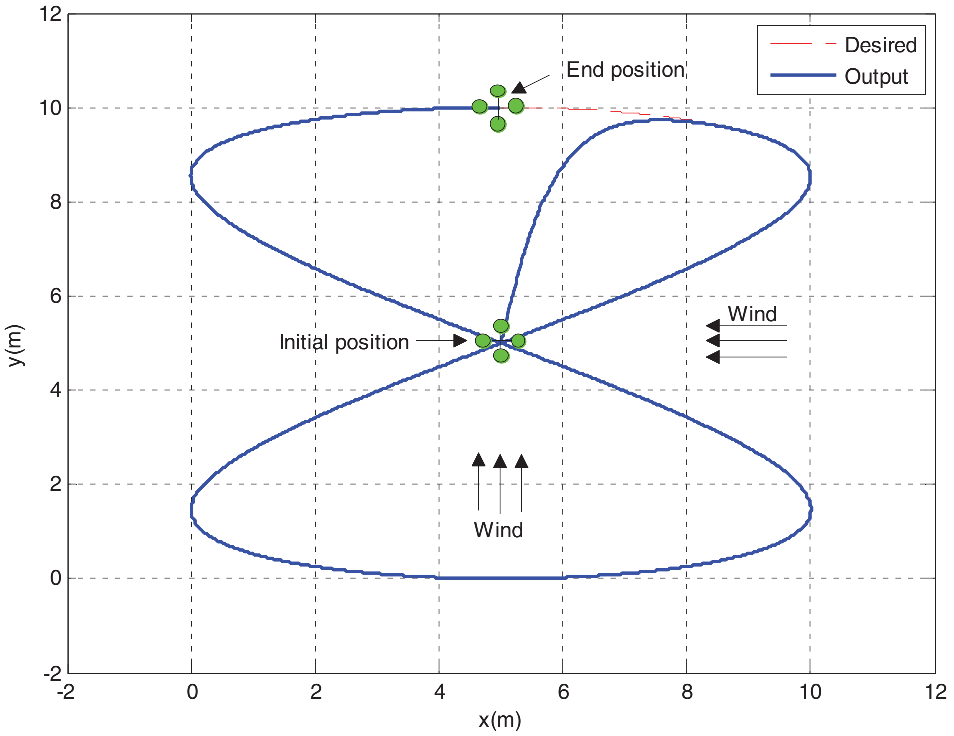


Fig. 25. *x–y* position eight-shape trajectory tracking response using ABSMC at Case 2.

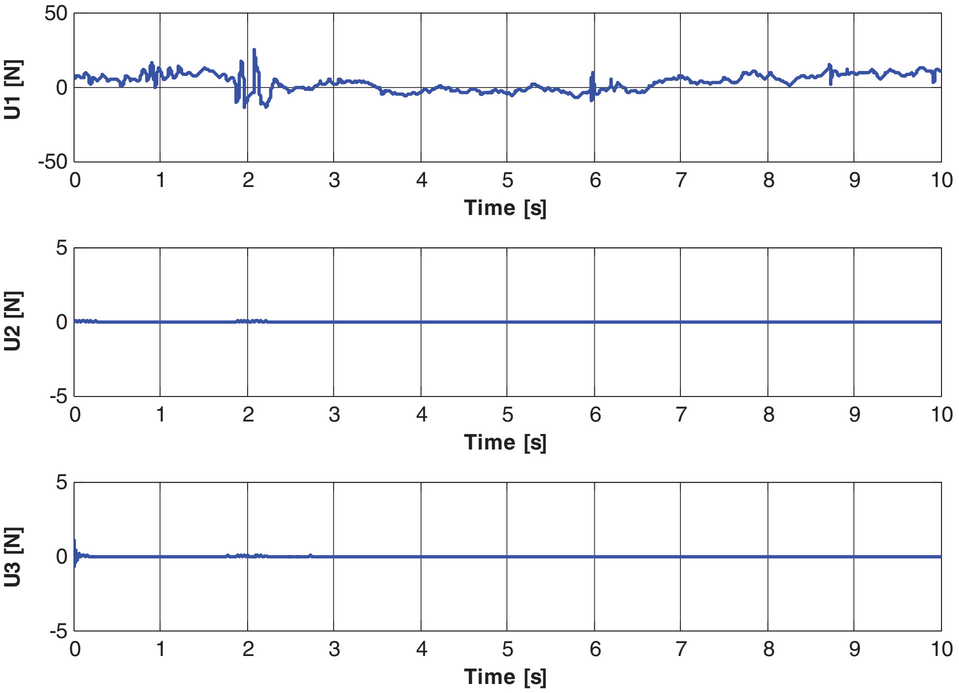


Fig. 26. Control inputs for eight-shape trajectory tracking using ABSMC at Case 2.

Acknowledgments

This work is supported by Universiti Teknologi Malaysia under the Research University Grant (Q.J130000.2523.15H39).

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

References

1. Y. Sun, N. Xian and H. Duan, "Linear-quadratic regulator controller design for quadrotor based on pigeon-inspired optimization," *Aircr. Eng. Aerosp. Technol.* **88**(6), 761–770 (2016).
2. D. Yiqun, F. Jun, Y. Bin, Z. Youmin and A. Jian Liang, "Position and Heading Angle Control of an Unmanned Quadrotor Helicopter Using LQR Method," *Proceedings of the* 34^{th} *Chinese Control Conference (CCC)* Hangzhou, China (2015) pp. 5566–5571.
3. F. Rinaldi, S. Chiesa and F. Quagliotti, "Linear quadratic control for quadrotors UAVs dynamics and formation flight," *J. Intell. Robot. Syst.* **70**(1-4), 203–220 (2013).
4. D. Sabzevari, S. M. Kargar and S. M. A. Zanjani, "Mathematical modeling and designing PID controller for a Quadrotor and optimize its step response by Genetic Algorithm," *Majlesi J. Electr. Eng.* **10**(4), 17–24 (2016).
5. J. C. V. Junior, J. C. De Paula, G. V. Leandro and M. C. Bonfim, "Stability control of a quad-rotor using a PID controller," *Braz. J. Instrum. Control* **1**(1), 15–20 (2013).
6. A. Noordin, M. A. M. Basri, Z. Mohamed and A. F. Z. Abidin, "Modelling and PSO fine-tuned PID control of quadrotor UAV," *Int. J. Adv. Sci. Eng. Inf. Technol.* **7**(4), 1367–1373 (2017).
7. A. Prayitno, V. Indrawati and G. Utomo, "Trajectory tracking of AR. Drone quadrotor using fuzzy logic controller," *TELKOMNIKA* **12**(4), 819–828 (2014).
8. F. Torres, A. Rabhi, D. Lara, G. Romero and C. Pégard, "Fuzzy state feedback for attitude stabilization of quadrotor," *Int. J. Adv. Robot. Syst.* **13**(1), 2 (2016).
9. E.-H. Zheng, J.-J. Xiong and J.-L. Luo, "Second order sliding mode control for a quadrotor UAV," *ISA Trans*. **53**(4), 1350–1356 (2014).
10. B. Sumantri, N. Uchiyama, S. Sano and Y. Kawabata, "Robust tracking control of a quad-rotor helicopter utilizing sliding mode control with a nonlinear sliding surface," *J. Syst. Des. Dyn.* **7**(2), 226–241 (2013).
11. B. Xiao and S. Yin, "A new disturbance attenuation control scheme for quadrotor unmanned aerial vehicles," *IEEE Trans. Ind. Inf.* **13**(6), 2922–2932 (2017).
12. X. Huo, M. Huo and H. R. Karimi, "Attitude stabilization control of a quadrotor UAV by using backstepping approach," *Math. Probl. Eng.* 1–9 (2014).
13. E. C. Suicmez and A. T. Kutay, "Path tracking control of a quadrotor UAV with backstepping method," *J. Aeronaut. Space Technol.* **7**(2), 1–13 (2014).
14. M. A. M. Basri, A. R. Husain and K. A. Danapalasingam, "Backstepping controller with intelligent parameters selection for stabilization of quadrotor helicopter," *J. Eng. Sci. Technol. Rev.* **7**(2), 66–74 (2014).
15. G. Bartolini, L. Fridman, A. Pisano and E. Usai, "Modern sliding mode control theory," *Lecture Notes Control Inf. Sci.* **375**(7), 23–49 (2008).
16. M. A. A. Hallaj and N. Assadian, "Sliding mode control of electromagnetic tethered satellite formation," *Adv. Space Res.* **58**(4), 619–634 (2016).
17. B. Turner, J. Williams, Y. Shtessel and R. Adhami, "Integral Sliding Mode Autopilot for Rocket Stabilization with Unmatched Disturbances," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, USA (2012) p. 4466.
18. D. V. Rao and N. K. Sinha, "A sliding mode controller for aircraft simulated entry into spin," *Aerosp. Sci. Technol.* **28**(1), 154–163 (2013).
19. C. T. Chen, "A sliding mode control strategy for robust temperature trajectory tracking of a batch reactor," *Chem. Eng. Commun.* **201**(1), 1–22 (2014).
20. C. Zhong, Y. Guo, C. Zhou and Q. Chen, "On sliding mode variable structure control for flexible spacecraft and chattering reduction," *Aerosp. Control Appl.* **39**(2), 23–27 (2013).
21. C. Pukdeboon, "Lyapunov optimizing sliding mode control for robot manipulators," *Appl. Math. Sci.* **7**(63), 3123–3139 (2013).
22. F. O. Maria Joseph and T. Podder, "Sliding mode control of a shape memory alloy actuated active flexible needle," *Robotica* **36**(8), 1188–1205 (2018).
23. Z. Q. Guo, J. X. Xu and T. H. Lee, "Design and implementation of a new sliding mode controller on an underactuated wheeled inverted pendulum," *J. Franklin Inst.* **351**(4), 2261–2282 (2014).
24. J. Kennedy, R. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neural Networks*, Perth, WA, Australia (1995) pp. 1942–1948.
25. M. Abido, "Optimal design of power-system stabilizers using particle swarm optimization," *IEEE Trans. Energy Convers.* **17**(3), 406–413 (2002).
26. D. P. Rini, S. M. Shamsuddin and S. S. Yuhaniz, "Particle swarm optimization: Technique, system and challenges," *Int. J. Comput. Appl.* **14**(1), 19–27 (2011).
27. M. Marinaki, Y. Marinakis and G. E. Stavroulakis, "Fuzzy control optimized by a multi-objective particle swarm optimization algorithm for vibration suppression of smart structures," *Struct. Multidiscip. Optim.* **43**(1), 29–42 (2011).
28. Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," *IET Control Theory Appl.* **4**(11), 2343–2355 (2010).
29. R. Olfati-Saber, Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles *Ph.D. Thesis* (Massachusetts Institute of Technology, Cambridge, MA, USA, 2001).
30. S. Bouabdallah, Design and Control of Quadrotors with Application to Autonomous Flying *Ph.D. Thesis* (École Polytechnique Fédérale De Lausanne, Lausanne, EPFL, 2007).

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press

31. R. Hassan, B. Cohanim, O. De Weck and G. Venter, "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," *Proceedings of the* 1*st* *AIAA Multidisciplinary Design Optimization Specialist Conference*, Austin, TX, USA (2005).
32. M. P. Lalitha, V. V. Reddy and V. Usha, "Optimal DG placement for minimum real power loss in radial distribution systems using PSO," *J. Theor. Appl. Inf. Technol.* **13**(2), 107–116 (2010).
33. R. C. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization," *Proceedings of the IEEE Congress on Evolutionary Computation*, La Jolla, CA, USA (2000) pp. 84–88.
34. B. Allaoua, B. Gasbaoui and B. Mebarki, "Setting up PID DC motor speed control alteration parameters using particle swarm optimization strategy," *Leonardo Electron. J. Practices Technol.* **14**, 19–32 (2009).
35. H. Voos, "Nonlinear Control of a Quadrotor Micro-UAV Using Feedback-Linearization," *IEEE International Conference on Mechatronics*, Malaga, Spain (2009) pp. 1–6.
36. S. L. Waslander and C. Wang, "Wind Disturbance Estimation and Rejection for Quadrotor Position Control," *AIAA Infotech@Aerospace Conference and AIAA Unmanned... Unlimited Conference*, Seattle, WA, USA (2009).

https://doi.org/10.1017/S0263574718000668 Published online by Cambridge University Press