2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

Robust output feedback finite-horizon optimal control for landing unmanned quadrotors on a slope

Hao Wang, Geng Lu, Zongying Shi, and Yisheng Zhong†

Abstract— **The ability to land on a slope is a meaningful capability of a quadrotor. In slope landing, the controller applied to the quadrotor should precisely control the bottom plane parallel to the slope when touching down, as well as attenuate the disturbances from nonlinearities, ground effects and other interference. In this paper, an output feedback finite-horizon optimal control is used to accurately adjust the attitude of the quadrotor to make the bottom plane track a desired trajectory, and a robust compensator (RC) technique is applied to attenuate the disturbances. The effectiveness of the proposed method is shown by experiments using a quadrotor constrained by a universal joint.**

I. INTRODUCTION

Autonomous quadrotors are widely used in all kinds of civilian tasks, including agricultural irrigation, geological exploration and maritime operations. As the quadrotor needs to be retrieved in most of the tasks, it is crucial for fully autonomous quadrotors to have the ability to land on various areas. In most cases, the landing area can be regarded as a slope, so the general circumstance is that the unmanned quadrotors are expected to land on a slope.

Landing unmanned aerial vehicles on a slope has been challenging work due to the dangerous touchdown moment. Various methods have been applied, but it still remains an open-ended topic. Methods for solving this problem can be divided into two classes: landing by using extra auxiliary mechanisms and landing by specially designed control strategies.

One of the most representative studies using extra auxiliary mechanisms is [1]. The authors used a biomimetic structure for quadrotors to land on a hazardous terrain. The landing gear of the quadrotor can automatically adapt to the rough surface, making it well suited for slope landing. In [2], the researchers designed a landing foot for the quadrotor and a landing platform with four cones as a passive centering system. This design can dampen the vibration caused by a hard landing, and the cones can still provide a suitable platform for the quadrotor to land when the platform is inclined. In [3], a serial robotic manipulator was used to capture the quadrotor in flight, so as to avoid the dangerous touchdown phase. A mobile self-leveling landing platform was designed in [4]. The landing platform can automatically adjust its attitude to provide the quadrotor with a flat plane. The dedicatedly designed devices in those methods need to be controlled separately, and some of them may add weight to the quadrotor, which will restrict its application.

Using specially designed control strategies is a more efficient way to land quadrotors on a slope, with no external devices needed. However, the quadrotor needs to perform some aggressive maneuvers to safely touch the slope, making this type of strategy more difficult and dangerous. Experiments of such aggressive maneuvers were conducted in [5]. The authors used geometric control for trajectory tracking, and the quadrotors were able to fly through a hole or dock on a wall. Such aggressive maneuvers can be easily modified for slope landing tasks. However, this strategy relies greatly on an external positioning system, making it unsuitable to be used in outdoor environments. In [6], the authors built a visual system with one camera and four lasers mounted on a quadrotor to estimate the pose of the slope, so as to remove the reliance on external positioning systems, but only preliminary tests of the positioning capability of the designed system were carried out.

A finite state machine (FSM)-based strategy was presented in [7] for the quadrotor to land on a sloped incline. The quadrotor first touches the slope with only one side, and then slides along the slope to make the other side settled. In the research by [8], an Parrot AR.Drone quadrotor was able to land on a moving inclined platform using marker methods for pose estimation. The authors used a constrained discrete-time nonlinear model predictive control (MPC) to solve the landing problem. This method focuses on the high dynamic of the moving platform, and the trajectory is calculated online.

In this paper, A robust output feedback finite-horizon optimal control method is used to precisely control the quadrotor under terminal constraints in which the landing gear of the quadrotor should be parallel to the slope. Finite-horizon optimal control is widely used in persuing, landing and other control problems that take terminal constraints into account. One can refer to [9] and [10] for more information. Furthermore, as finite-horizon optimal control requires an accurate model, a robust compensator (RC) is adopted to eliminate disturbances. RC techniques have been used in controlling quadrotors ([11], [12], [13]), but it has not been employed in the output feedback finite-horizon control problem, to the best of our knowledge. The algorithms are tested using a universal joint–constrained quadrotor.

The remainder of this paper is organized as follows. Section II describes the quadrotor model. The robust output feedback finite-horizon optimal controller is presented in Section III, which entails the design of the control and the

*This work was supported by the National Natural Science Foundation of China under Grants 61203071 and 61210012.

The authors are with Department of Automation, Tsinghua University, Beijing, 100084, P. R. China.

† Corresponding author, e-mail: zys-dau@mail.tsinghua.edu.cn

generation of the trajectory. The experiments are detailed in Section IV. The conclusion and future work considerations are discussed in Section V.

II. THE QUADROTOR MODEL

The dynamic model of the quadrotor can be obtained via a Lagrange approach, and a simplified model of the quadrotor's attitude and height is formulated into the following equations ([14]):

$$\begin{aligned} m\ddot{z} &= b(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2)\cos\phi\cos\theta - mg + \xi_1,\\ I_x\ddot{\theta} &= \dot{\phi}\dot{\psi}(I_y-I_z) + lb(\omega_4^2-\omega_2^2) + \xi_2,\\ I_y\ddot{\phi} &= \dot{\theta}\dot{\psi}(I_z-I_x) + lb(\omega_1^2-\omega_3^2) + \xi_3,\\ I_z\ddot{\psi} &= \dot{\phi}\dot{\theta}(I_x-I_y) + d(\omega_1^2+\omega_3^2-\omega_2^2-\omega_4^2) + \xi_4, \end{aligned} \tag{1}$$

where z, θ, ϕ and ψ are the height, roll, pitch and yaw, respectively; ω_i the angular velocity of rotor i; ξ_i the disturbances; l the displacement of the rotors with respect to the center of mass of the quadrotor; b the thrust coefficient and d the drag coefficient.

In this paper, $\cdot(t)$ is omitted for time-domain variables, unless there is a need to emphasise that the variable is time-varying; $\cdot(s)$ is used to denote the corresponding frequency-domain variable.

Define the control inputs $u_i (i = 1, 2, 3, 4)$ as

$$\begin{aligned} u_1 &= b(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2)\cos\phi\cos\theta - mg\\ u_2 &= b(\omega_4^2-\omega_2^2),\\ u_3 &= b(\omega_1^2-\omega_3^2),\\ u_4 &= d(\omega_1^2+\omega_3^2-\omega_2^2-\omega_4^2), \end{aligned} \tag{2}$$

Choose the state variables as $x = (z, \dot{z}, \theta, \dot{\theta}, \phi, \dot{\phi}, \psi, \dot{\psi})$, and the quadrotor's model can be formulated into the following state-space model:

$$\begin{aligned} \dot{x} &= Ax + B(u+\delta),\\ y &= Cx, \end{aligned} \tag{3}$$

where

$$A = \begin{bmatrix} 0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0 \end{bmatrix} \quad B = \begin{bmatrix} 0&0&0&0\\ b_1^N&0&0&0\\ 0&0&0&0\\ 0&b_2^N&0&0\\ 0&0&0&0\\ 0&0&b_3^N&0\\ 0&0&0&0\\ 0&0&0&b_4^N \end{bmatrix}$$

$$C = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&0 \end{bmatrix}, \tag{4}$$

and

$$\delta = \frac{1}{l}\begin{bmatrix} \xi_1 + \frac{b_1-b_1^N}{b_1^N}u_1\\ \xi_2 + \dot{\phi}\dot{\psi}(I_y-I_z) + \frac{b_2-b_2^N}{b_2^N}u_2\\ \xi_3 + \dot{\theta}\dot{\psi}(I_z-I_x) + \frac{b_3-b_3^N}{b_3^N}u_3\\ \xi_4 + \dot{\phi}\dot{\theta}(I_x-I_y) + \frac{b_4-b_4^N}{b_4^N}u_4 \end{bmatrix}, \tag{5}$$
$$b_1^N = \frac{1}{m}, b_2^N = \frac{l}{I_x}, b_3^N = \frac{l}{I_y}, b_4^N = \frac{1}{I_z}.$$

In the above, b_i^N is the nominal value of b_i. Assume that $\|b_i - b_i^N\| \le b_i^N$.

Supposing the quadrotor is hovering at h m above the slope at time t_0,

$$y(t_0) = \begin{bmatrix} h & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{6}$$

The objective is to make $y(t_f)$ equal to a desired value y_d, given by

$$y_d = \begin{bmatrix} 0 & 0 & \theta_d & 0 & \phi_d & 0 & \psi_d & 0 \end{bmatrix}, \tag{7}$$

where θ_d, ϕ_d and ψ_d are the inclination angles of the slope relative to the quadrotor. Therefore, the trajectory is generated using a modified sigmoid function defined by

$$\hat{y}(t) = \frac{\gamma y_d}{1+e^{-\alpha(t-(t_f+t_0)/2+\beta)}},\ t \in [t_0, t_f], \tag{8}$$

where α and β are curvature parameters, and γ is used to make sure $\hat{y}(t_f) = y_d$, namely $\gamma = 1 + e^{-\alpha(t_f-t_0)/2+\beta)}$.

III. CONTROLLER DESIGN

In this part, the following generalized model is considered:

$$\begin{aligned} \dot{x} &= Ax + B(u+\delta),\\ y &= Cx, \end{aligned} \tag{9}$$

where x is the n-dimensional state variable, u is the p-dimensional control input, y is the m-dimensional output, and δ is the p-dimensional equivalent disturbance. $\{A, B\}$ is controllable, $\{A, C\}$ is observable, and C is full row rank.

Assumption 1: δ fulfills the following inequalities:

$$\|\delta\| \le \zeta_x\|x\|_2 + \zeta_{x2}\|x\|_2^2 + \zeta_u\|u\|_2 + \zeta_\xi\|\xi\|_2, \tag{10}$$

where ξ is the disturbance ($\xi = [\xi_1\ \xi_2\ \xi_3\ \xi_4]^T$), and $\zeta_i (i = x, x2, u, \xi)$ are positive constants. $\zeta_u \le 1$.

A. Finite-horizon optimal control

For this trajectory tracking problem, a finite-horizon optimal control is applied as follows.

Considering the nominal model of (9),

$$\begin{aligned} \dot{x} &= Ax + Bu_0,\\ y &= Cx, \end{aligned} \tag{11}$$

and the performance index of the optimal control problem is

$$J = \frac{1}{2}y_e^T(t_f)S_1y_e(t_f) + \frac{1}{2}\int_{t_0}^{t_f}[y_e^TQ_1y_e + u^TRu]dt, \tag{12}$$

where $y_e = y - \hat{y}$, S_1, Q_1 are semi-positive definite matrices, and R is a positive definite matrix.

Let $Q = C^T Q_1 C$ and $S = C^T S_1 C$, and according to the optimal control theory, the control law is

$$u_0 = -R^{-1}B^T[P(t)\bar{x} - \beta(t)], \tag{13}$$

where $P(t)$ and $\beta(t)$ satisfy the following equations,

$$\begin{aligned}
&\dot{P}(t) = -P(t)A - A^T P(t) + P(t)BR^{-1}B^T P(t) - Q,\\
&P(t_f) = S,\\
&\dot{\beta}(t) = -[A - BR^{-1}B^T P(t)]^T \beta(t) + Q\hat{x},\\
&\beta(t_f) = -S\hat{x}(t_f),
\end{aligned} \tag{14}$$

where $\hat{x} = L\hat{y}$ and $L = C^T(CC^T)^{-1}$. $\bar{x}$ can be obtained by a state observer,

$$\begin{aligned}
&\dot{\bar{x}} = A\bar{x} + Bu_0 - H(\bar{y} - y),\\
&\bar{y} = C\bar{x}, \quad \bar{x}(t_0) = Ly(t_0),
\end{aligned} \tag{15}$$

where H is chosen such that $A - HC$ is a Hurwitz matrix.

B. Robust compensator design

In this part, an RC is designed to eliminate the effects of $\delta(t)$. According to the system model (9),

$$\begin{aligned}
y(s) &= C(sI - A)^{-1}[B(u(s) + \delta(s)) + x(t_0)]\\
&= D^{-1}(s)[N(s)(u(s) + \delta(s)) + \hat{N}(s)x(t_0)],
\end{aligned} \tag{16}$$

where $D^{-1}(s)N(s) = C(sI - A)^{-1}B$ and $D^{-1}(s)\hat{N}(s) = C(sI - A)^{-1}$ are the matrix-fraction descriptions (MFDs). $D(s)$ and $N(s)$ are real-valued, right-coprime and row-reduced polynominal matrices, and δ can be expressed as

$$\delta(s) = N^{-1}(s)D(s)y(s) - u(s) - N^{-1}(s)\hat{N}(s)x(t_0). \tag{17}$$

Assumption 2: The nominal system is minimum phase, and the column relative degrees in the transfer function matrix are $\{d_1, d_2, ..., d_p\}$.

Consider the control input u as

$$u = u_0 + v, \tag{18}$$

and $v(t)$ is used to eliminate equivalent disturbance δ,

$$v(s) = -F(s)\delta(s), \tag{19}$$

where $F(s)$ is a low-pass filter called the robust filter, and it has the form

$$F(s) = \text{diag}\left[(\tfrac{f_1}{s+f_1})^{d_1} \quad ... \quad (\tfrac{f_p}{s+f_p})^{d_p}\right]. \tag{20}$$

As d_is are the column relative degrees in the transfer function matrix, $F(s)$ is used for avoiding differential. The f_is are parameters need to be tuned.

As $N(s)$ is nonsingular real matrix, There exist real matrices $\{H_D(s), R_D(s)\}$ and $\{H_{\hat{N}}(s), R_{\hat{N}}(s)\}$, such that

$$\begin{aligned}
D(s) &= N(s)H_D(s) + R_D(s),\\
\hat{N}(s) &= N(s)H_{\hat{N}}(s) + R_{\hat{N}}(s).
\end{aligned} \tag{21}$$

The RC can be expressed as

$$v(s) = -F(s)[H_D(s)y(s) - H_{\hat{N}}(s)x(t_0) - u(s) + \eta(s)], \tag{22}$$

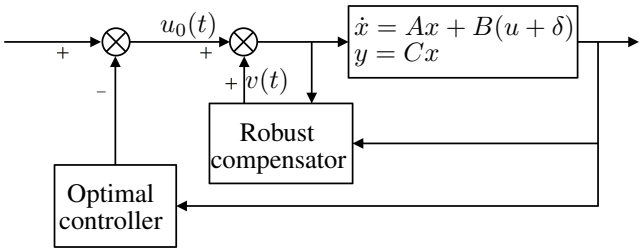

Fig. 1. The schematic diagram of the control system

where

$$\eta(s) = N^{-1}(s)R_D(s)y(s) - N^{-1}(s)R_{\hat{N}}(s)x(t_0). \tag{23}$$

The schematic diagram of the control system is described in Figure 1. The implementation of the RC defined by (22) and (23) is derived as follows.

Define two auxiliary matrices

$$\begin{aligned}
\Phi_N(s) &= \text{diag}\{s^{q_i}, i = 1, ..., m\},\\
S_N(s) &= \text{block diag}\{\begin{bmatrix} s^{q_i - 1} & ... & s & 1 \end{bmatrix}\},
\end{aligned}$$

where q_i is the maximum degree in row i of $N(s)$. The $N(s)$ can be expressed as

$$N(s) = \Phi_N(s)N_h + S_N(s)N_l,$$

where N_h and N_l are constant matrices. Then $R_D(s)$ and $R_{\hat{N}}(s)$ can be expressed as

$$\begin{aligned}
R_D(s) &= S_N(s)R_{Dl},\\
R_{\hat{N}}(s) &= S_N(s)R_{\hat{N}l},
\end{aligned}$$

where R_{Dl} and $R_{\hat{N}l}$ are constant matrices, and $\eta(s)$ becomes

$$\eta(s) = (\Phi_N(s)N_h + S_N(s)N_l)^{-1}S_N(s)(R_{Dl}y(s) - R_{\hat{N}l}x(t_0)). \tag{24}$$

Assume the state-space implementation of $\Phi_N^{-1}(s)S_N(s)$ is (A_η, B_η, C_η), then η can be obtained through the following system

$$\begin{aligned}
\dot{x}_\eta &= (A_\eta - B_\eta N_l N_h^{-1} C_\eta)x_\eta + B_\eta R_{Dl} y,\\
\eta &= N_h^{-1} C_\eta x_\eta, \quad x_\eta(t_0) = -B_\eta R_{\hat{N}l} x(t_0).
\end{aligned} \tag{25}$$

Define

$$\begin{aligned}
N_I &= \text{diag}\{f_i^{d_i}, i = 1, ..., p\},\\
D_I(s) &= \text{diag}\{(s + f_i)^{d_i}, i = 1, ..., p\},
\end{aligned} \tag{26}$$

$$\begin{aligned}
S_I(s) &= \text{block diag}\{S_d^i(s), i = 1, ..., p\},\\
S_d^i(s) &= \begin{bmatrix} (s + f_i)^{d_i - 1} & ... & s + f_i & 1 \end{bmatrix},
\end{aligned} \tag{27}$$

then $F(s)$, $H_D(s)$ and $H_{\hat{N}}(s)$ can be expressed as

$$\begin{aligned}
F(s) &= N_I D_I^{-1}(s),\\
H_D(s) &= D_I(s)H_{Dh} + S_I(s)H_{Dl},\\
H_{\hat{N}}(s) &= S_I(s)H_{\hat{N}l},
\end{aligned} \tag{28}$$

where H_{Dh}, H_{Dl} and $H_{\hat{N}l}$ are real-valued matrices. Then

$$\begin{aligned}
v(t) &= -N_I H_{Dh} y(t) - N_I \zeta(t),\\
\zeta(s) &= D_I^{-1}(s)S_I(s)[H_{Dl}y(s) - H_0(u(s) - \eta(s))\\
&\quad - H_{\hat{N}l}x(t_0)],
\end{aligned} \tag{29}$$

and

$$H_0 = \text{block diag}\{\begin{bmatrix} 0_{(d_i-1)\times 1} \\ 1 \end{bmatrix}, i = 1, ..., p\}.$$

Assume that the state-space implementation of $D_I^{-1}(s)S_I(s)$ is $(A_\zeta, B_\zeta, C_\zeta)$, then ζ can be obtained through the following system

$$\begin{aligned} \dot{x}_\zeta &= A_\zeta x_\zeta + B_\zeta H_{Dl} y - B_\zeta H_0(u-\eta), \\ \zeta &= C_\zeta x_\zeta, \quad x_\zeta(t_0) = -B_\zeta H_{\hat{N}l} x(t_0). \end{aligned} \tag{30}$$

Then the implementation of the RC can be obtained by the equations (29), (30) and (25).

Substituting (13) into (3) and (15), one can get

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{\bar{x}} \end{bmatrix} &= [A_c - E(t)] \begin{bmatrix} x \\ \bar{x} \end{bmatrix} \\ &\quad + \begin{bmatrix} B \\ B \end{bmatrix} R^{-1} B^T \beta(t) + B_c[v(t) + \delta(t)], \\ y &= C_c \begin{bmatrix} x \\ \bar{x} \end{bmatrix}, \end{aligned} \tag{31}$$

where

$$\begin{aligned} A_c &= \begin{bmatrix} A & 0 \\ HC & A - HC \end{bmatrix}, E(t) = \begin{bmatrix} 0 & BR^{-1}B^T P(t) \\ 0 & BR^{-1}B^T P(t) \end{bmatrix}, \\ B_c &= \begin{bmatrix} B \\ 0 \end{bmatrix}, C_c = \begin{bmatrix} C & 0 \end{bmatrix}. \end{aligned} \tag{32}$$

The nominal system of (31) is

$$\begin{aligned} \begin{bmatrix} \dot{x}_N \\ \dot{\bar{x}}_N \end{bmatrix} &= [A_c - E(t)] \begin{bmatrix} x_N \\ \bar{x}_N \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} R^{-1} B^T \beta(t), \\ y_N &= C_c \begin{bmatrix} x_N \\ \bar{x}_N \end{bmatrix}. \end{aligned} \tag{33}$$

The robustness of the proposed RC is shown in the following theorem.

Theorem 1: Under Assumption 1, $\forall x(t_0) = x_0$, $\forall \epsilon > 0$, $\exists t_f^* > 0, f^* > 0$, such that when t_f is designed to be $t_f > t_f^*$, and $f \geq f^*$,

$$\|y(t_f) - y_N(t_f)\|_2 \leq \epsilon. \tag{34}$$

Moreover, if $x(t_0) = x_N(t_0)$,

$$\|y(t) - y_N(t)\|_2 \leq \epsilon, \forall t \in [t_0, t_f]. \tag{35}$$

IV. Simulation and Experimental Results

In this section, the effectiveness of the finite-horizon optimal controller is shown through a simulation experiment; then the methods are tested on a universal joint–constrained quadrotor. In the experiments, the quadrotor is expected to track a desired trajectory, as shown in Figure 2. The trajectory only involves height and roll, for the quadrotor can always adjust its yaw angle to the marker to make sure that the inclination angle of pitch is zero.

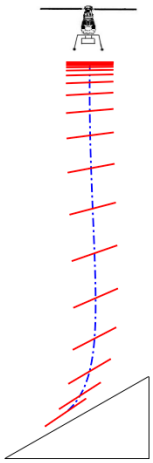

Fig. 2. The desired trajectory in the landing process.

A. Simulation results

The quadrotor model used in the simulation is built based on [15]. The nominal parameters and the initial conditions of the quadrotor for simulation are

$$\begin{aligned} x(t_0) &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\ \hat{y}(t_0) &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \hat{y}(t_f) = \begin{bmatrix} 0 & 10^\circ & 0 & 0 \end{bmatrix}, \\ m &= 1.6\,\text{kg}, l = 0.28\,\text{m}, I_x = 0.0387\,\text{Ns}^2/\text{rad}, \\ I_y &= 0.0351\,\text{Ns}^2/\text{rad}, I_z = 0.0563\,\text{Ns}^2/\text{rad}, \\ b &= 4.109 \times 10^{-5}\,\text{N/rad}^2, d = 6.0327 \times 10^{-7}\,\text{N/rad}^2. \end{aligned}$$

Also, white noises are added to simulate the terms $\xi_i, i = 1...4$.

In the simulation, the quadrotor is initially at 1 m above the slope, and tracks a desired trajectory to land on the slope. The comparison of the results and the desired trajectory are shown in Figures 3-5. From the results it can be seen that when RC is applied, the tracking errors as well as the errors at the touchdown moment are much smaller than when only the finite-horizon optimal controller is applied. Moreover, the quadrotor is only about 4 cm away from the origin in the x-direction at the touchdown moment, which is fairly acceptable.

B. Experimental results

In this part, the experimental results of the slope landing are presented. The quadrotor is constrained by a universal joint for safety reasons, as shown in Figure 6, and roll angle tracking results are given to show that the presented method is adoptable in a real scenario.

The roll angle tracking results are shown in Figure 7. The root-mean-square errors (RMSEs) of using the finite-horizon optimal controller with and without an RC are 0.0127 rad and 0.0461rad, respectively; the tracking errors at the touchdown moment are 0.0026 rad and 0.0918 rad, respectively.

V. Conclusions

In this paper, the finite-horizon optimal control with an RC was implemented to drive the quadrotor to a desired

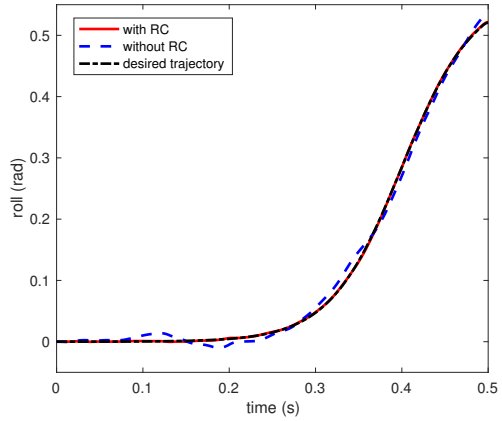

Fig. 3. Tracking results of roll angle in simulation.

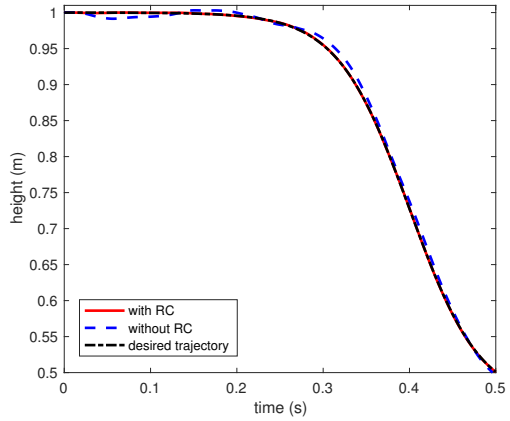

Fig. 4. Tracking results of height in simulation.

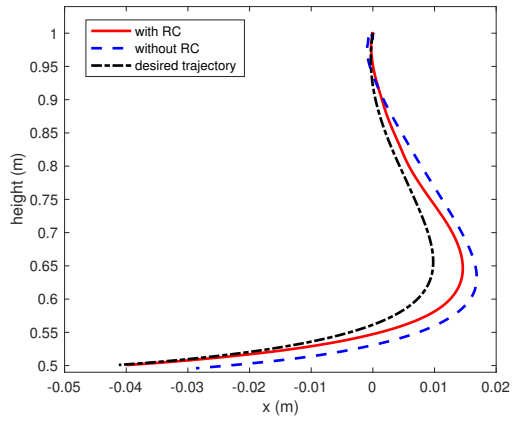

Fig. 5. Trajectories in the landing process in simulation.

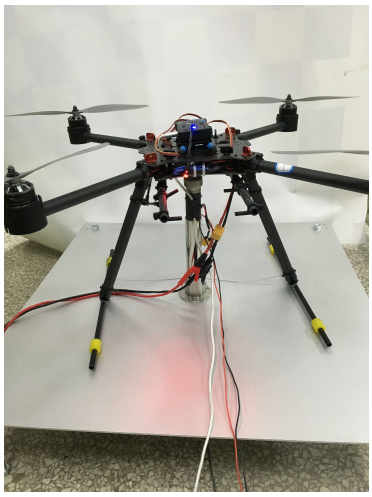

Fig. 6. The universal joint–constrained quadrotor used in the experiment.

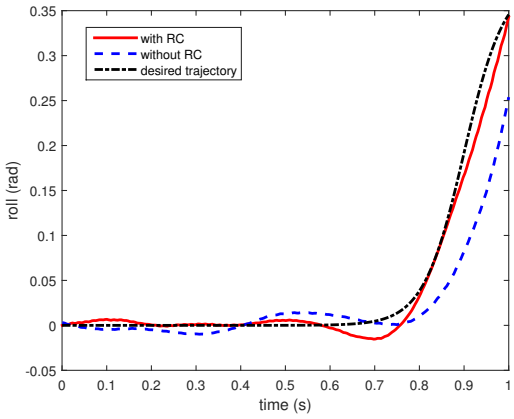

Fig. 7. Tracking results of roll angle.

state for landing on a slope at the specified time. Simulation results and experimental results were carried out to verify the effectiveness of the proposed methods.

In future work, real slope landing experiments may be conducted, and the translational displacements may be taken into consideration for landing on a small slope area.

REFERENCES

[1] C. Luo, X. Li, and Q. Dai, "Biology's drones: new and improved." *Science (New York, NY)*, vol. 344, no. 6190, pp. 1351–1351, 2014.

[2] F. Cocchioni, E. Frontoni, G. Ippoliti, S. Longhi, A. Mancini, and P. Zingaretti, "Visual based landing for an unmanned quadrotor," *Journal of Intelligent & Robotic Systems*, pp. 1–18, 2015.

[3] M. Maier, A. Oeschger, and K. Kondak, "Robot-assisted landing of vtol uavs: Design and comparison of coupled and decoupling linear state-space control approaches," *Robotics and Automation Letters, IEEE*, vol. 1, no. 1, pp. 114–121, 2016.

[4] S. A. Conyers, N. I. Vitzilaios, M. J. Rutherford, and K. P. Valavanis, "A mobile self-leveling landing platform for vtol uavs," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 815–822.

[5] D. Mellinger, N. Michael, and V. Kumar, *Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors*, 2014.

[6] J. Dougherty, D. Lee, and T. Lee, "Laser-based guidance of a quadrotor uav for precise landing on an inclined surface," pp. 1210–1215, 2014.

[7] D. Cabecinhas, R. Cunha, and C. Silvestre, "A robust landing and sliding maneuver controller for a quadrotor vehicle on a sloped incline," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 523–528.
[8] P. Vlantis, P. Marantos, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Quadrotor landing on an inclined platform of a moving ground vehicle," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2202–2207.
[9] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific,, 2000.
[10] M. C. Tsai and D. W. Gu, "Robust and optimal control," *Advances in Industrial Control*, vol. volume 33, no. 97, pp. 2095–2095(1), 2014.
[11] H. Liu, Y. Bai, G. Lu, Z. Shi, and Y. Zhong, "Robust tracking control of a quadrotor helicopter," *Journal of Intelligent & Robotic Systems*, vol. 75, no. 3-4, pp. 595–608, 2014.
[12] Y. Bai, H. Liu, Z. Shi, and Y. Zhong, "Robust control of quadrotor unmanned air vehicles," in *Control Conference (CCC), 2012 31st Chinese*. IEEE, 2012, pp. 4462–4467.
[13] H. Liu, D. Li, J. Xi, and Y. Zhong, "Robust attitude controller design for miniature quadrotors," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 4, pp. 681–696, 2016.
[14] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *Ieee/rsj International Conference on Intelligent Robots and Systems*, 2007, pp. 153–158.
[15] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," *MSc Theses*, 2008.