J Intell Robot Syst
DOI 10.1007/s10846-014-0161-3

Solving the Boundary Value Problem of an Under-Actuated Quadrotor with Subspace Stabilization Approach

Wei Dong · Guo-Ying Gu · Xiangyang Zhu · Han Ding

Received: 3 May 2014 / Accepted: 14 November 2014

Abstract In this paper, a control strategy based on the optimal control and subspace stabilization approach is developed to solve the two-point boundary value problem of a highly under-actuated quadrotor. To facilitate the development, the dynamic model of the quadrotor is firstly presented. Then the boundary value problem is mathematically formulated based on the optimal control theory. According to the problem formulation and utilizing the subspace stabilization approach, the control strategy is proposed to suppress the state-trajectory tracking errors and manipulate the quadrotor from a known initial state to the desired final state in a finite time horizon. As there exist input delays in real-time flights, the Smith predictor is designed to enhance the performance of the developed control strategy. Finally, an indoor experimental platform of the quadrotor is built and real-time experiments of the ball-batting is conducted with a coefficient of restitution of approximate 0.7 and a racket with diameter of 0.13 m. The experimental results show that the quadrotor can well establish the desired final state and bat the ball towards its target location (the deviation of position is less than 0.15 m), which verify the feasibility of the proposed control strategy.

Keywords Under-actuated · Quadrotor · Subspace stabilization · Smith predictor

1 Introduction

During the last decade, the quadrotors have attracted increasingly attentions in the robotics community [1]. These kind of aerial robots can be potentially implemented in both military and civil environment, thus outlines grand opportunities and challenges for researches and applications. A number of researchers have chosen them as their experimental platforms and tried to manipulate them to fulfill various missions [2, 3]. Through their activities, the design [4, 5], flight control [6–9] and path planning [10, 11] are extensively investigated.

In view of the state-of-the-art, it is hopeful that in the near future the quadrotors can undertake a diverse range of challenging missions imposed by the real world which might be dangerous or space limited for human beings, such as building exploration, bridge inspection and agricultural care [1, 12–14]. To achieve

W. Dong · G.-Y. Gu (✉) X. Zhu · H. Ding
State Key Laboratory of Mechanical System and Vibration,
Institute of Robotics, School of Mechanical Engineering,
Shanghai Jiao Tong University, 800 Dongchuan Road,
Shanghai, China
e-mail: guguoying@sjtu.edu.cn

W. Dong
e-mail: chengquess@sjtu.edu.cn

X. Zhu
e-mail: mexyzhu@sjtu.edu.cn

H. Ding
e-mail: hding@sjtu.edu.cn

Published online: 25 November 2014

this objective, the quadrotors need to possess more agile and aggressive maneuvers that can cope with the changing environment in the complex real-time scenarios [1]. In this sense, all of the concerned states of the quadrotors should be well controlled in their specific missions. However, such control objective is hindered by the fact that the quadrotors are highly under-actuated systems with strong nonlinearities and coupling characteristics [15]. As pointed out in [16, 17], for the under-actuated systems, as the dimension of the output space is higher than that of the input space, it is impractical to utilize the smooth static feedback to stabilize the quadrotors into an arbitrary state by regarding the desired state as an equilibrium state [18]. To tackle this problem, several groups have started to develop innovative control strategies based on specific missions. For example, in [19], the quadrotors were controlled to fly through narrow, vertical gaps and perch on inverted surfaces. In such missions, nine dimensional state variables (including position, velocity, and attitude) were required to be manipulated to the desired final values within a finite time horizon. To solve this problem, the authors in [19] firstly built an ideal state-trajectory based on the nominal model of the quadrotor, and the corresponding control sequence was initialized. To well track the built state-trajectory, this control sequence was refined by the iteratively learning through successive experimental trials. Similarly, quadrotors from the Federal Institute of Technology Zurich were manipulated to juggle tennis balls [20], where eight dimensional state variables were required to reach their desired values within approximate 1 second. The ideal state-trajectory for this mission was prescribed based on the linearized model and iterative learning was adopted to enhance the performance.

It can be seen that from the literatures, control strategies are commonly developed in open-loop forms by adopting both model-based state-trajectory generation and iterative learning [19–21]. In this way, to guarantee the controllers' performance, a sufficient number of successive experimental trials associating with the iterative learning should be conducted. To get rid of the reliance on the iterative learning process and establish control strategies of closed-loop forms, this work is thereby inspired to formulate the problem as a two-point boundary value problem [16, 17], and then solve this problem based on the subspace stabilization approach and linear quadratic regulator (LQR). In this control scheme, a subset of the output space is stabilized at the first step. The states of this stabilized subspace are then chosen as a new set of virtual control inputs for its complementary subspace. Subsequently, the LQR is introduced to suppress the state-trajectory tracking errors in the whole output space. Considering the fact that there exist input delays due to the transport delays in communications and the dynamics of the quadrotors, the Smith predictor is designed to enhance the performance of the developed control strategy. To verify the effectiveness of the development, the task of ball-batting, which is similar to [20], is performed in the experimental tests. With lower coefficient of restitution (COR) [22] and a smaller racket than that adopted in [20], the real-time experiments show satisfactory results. This demonstrates that the developed control strategy can effectively solve the two-point boundary value problems of the under-actuated quadrotors.

The distinctive features of this paper are as follows. Firstly, a control strategy based on subspace stabilization approach and LQR is developed to solve the two-point boundary value problems of the under-actuated quadrotors. Secondly, the Smith predictor is introduced to alleviate effects of the input delays in the real-time flight control. Finally, the effectiveness of the developed control strategy is verified through the real-time experiments of ball-batting.

The remainder of this paper is organized as follows. Section 2 presents the dynamic model of the quadrotor. The control strategy is developed in Section 3, and its application to the ball-batting is demonstrated in Section 4. Then the experimental results are shown in Section 5, and Section 6 concludes this work.

2 The Quadrotor Model and Problem Formulation

To facilitate the controller development, the dynamic model of the quadrotor is presented in this section. Based on this model, the problem considered by this work is formulated accordingly.

2.1 Quadrotor Dynamics

The coordinates and free body diagram of the quadrotor are shown in Fig. 1. The quadrotor is actuated by four rotors on the endpoints of an X-shaped frame.

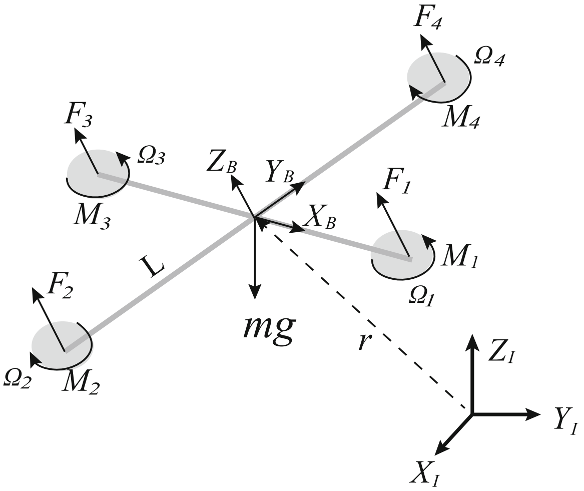


Fig. 1 A free body diagram of the quadrotor. $X_I - Y_I - Z_I$ is the inertial coordinates and $X_B - Y_B - Z_B$ is the body fixed coordinates

With this configuration, the acceleration along its normal direction (Z_B) is produced by the collective thrust of these four rotors. In order to balance the yawing torque, rotors attached on the Y_B axis rotate in clockwise direction, and the rotors attached on the X_B axis rotate in counterclockwise direction. In such a case, the yawing torque is produced according to the difference of collective torques between these two axes. Similarly, differences of thrusts between rotors on the X_B axis and the Y_B axis produce the pitching torque and the rolling torque. Therefore, four control inputs can be defined as

$$\begin{array}{ll} U_1 = F_1 + F_2 + F_3 + F_4, & U_2 = (F_4 - F_2)L, \\ U_4 = M_1 - M_2 + M_3 - M_4, & U_3 = (F_3 - F_1)L. \end{array} \tag{1}$$

where L is the length from the rotor to the center of the mass of the quadrotor, and F_i and M_i are the thrust and torque generated by rotor i $(i \in \{1, 2, 3, 4\})$.

In view of Eq. 1, equations governing dynamics of the quadrotor with respect to the inertial coordinates are generally expressed as [23, 24]

$$\begin{cases} \ddot{x} = \frac{U_1}{m}(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\ \ddot{y} = \frac{U_1}{m}(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \\ \ddot{z} = \frac{U_1}{m}\cos\phi\cos\theta - g \\ \ddot{\phi} = \frac{U_2}{I_{xx}} + \dot{\theta}\dot{\psi}(\frac{I_{yy}-I_{zz}}{I_{xx}}) - \frac{J_R}{I_{xx}}\dot{\theta}\Omega_R \\ \ddot{\theta} = \frac{U_3}{I_{yy}} + \dot{\phi}\dot{\psi}(\frac{I_{zz}-I_{xx}}{I_{yy}}) - \frac{J_R}{I_{yy}}\dot{\phi}\Omega_R \\ \ddot{\psi} = \frac{U_4}{I_{zz}} + \dot{\phi}\dot{\theta}(\frac{I_{xx}-I_{yy}}{I_{zz}}) \end{cases} \tag{2}$$

where x, y, and z are the position of the center of mass in the inertial coordinates; ϕ, θ, and ψ are the attitude; m, I_{xx}, I_{yy}, and I_{zz} are the mass and rotary inertia of the quadrotor, respectively; J_R and Ω_R are the rotary inertia and angular velocity of the propeller blades; and g is the gravity constant.

Linearizing (2) in the near hovering state ($\phi \approx 0,\ \theta \approx 0,\ \psi \approx 0$), one can obtain [24]

$$\begin{array}{lll} \ddot{x} = g\theta, & \ddot{y} = -g\phi, & \ddot{z} = \frac{1}{m}U_1 - g, \\ \ddot{\phi} = \frac{U_2}{I_{xx}}, & \ddot{\theta} = \frac{U_3}{I_{yy}}, & \ddot{\psi} = \frac{U_4}{I_{zz}}. \end{array} \tag{3}$$

A control vector can thus be redefined as

$$u = [U_1 - mg,\ U_2,\ U_3,\ U_4]^T \tag{4}$$

The control input space is then denoted as $\boldsymbol{u} = \{u\}$.

2.2 Problem Formulation

To formulate the problem mathematically, two spaces of the quadrotor are firstly defined as follows.

The first space is the configuration space. Similarly to other rigid bodies, this space for the quadrotor is defined as [25]

$$\boldsymbol{q} = \{q | q = [x,\ y,\ z,\ \phi,\ \theta,\ \psi]^T\} \tag{5}$$

The second space is the state space, which is described by

$$\boldsymbol{\xi} = \boldsymbol{q} \times \dot{\boldsymbol{q}} \times \ddot{\boldsymbol{q}} \tag{6}$$

The element ξ in the space $\boldsymbol{\xi}$ has a dimension of 18, which is much higher than that of the control input defined in Eq. 4. In the real-time flights, the quadrotors are commonly required to be controlled in its output space, denoted as $\boldsymbol{\zeta}$, which is a subspace of $\boldsymbol{\xi}$.

For the two-point boundary value problem, the goal is to manipulate the quadrotor from a known initial state $\zeta_0 \in \boldsymbol{\zeta}$ to the desired final state $\zeta_T^* \in \boldsymbol{\zeta}$ within a finite time horizon $t \in [0, T]$ through a feasible state-space trajectory $\zeta_d(t)$ which is consistent with the dynamics of the quadrotor. Therefore, the problem can be equivalently solved when this $\zeta_d(t)$ is well tracked. In this way, denoting the tracking error with respect to the desired trajectory as $\bar{\zeta}(t)$, the problem considered by this work can be stated as the follows

$$\min \mathcal{J}(\bar{\zeta}, u) = \int_0^T (\frac{1}{2}\bar{\zeta}(t)Q(t)\bar{\zeta}(t) + \frac{1}{2}u(t)R(t)u(t))\mathrm{d}t \tag{7}$$

subject to the dynamics given in Eq. 3 and the boundary value conditions: $\bar{\zeta}(T) = 0$.

3 Controller Development

In this section, a control strategy is developed based on the subspace stabilization approach and LQR. The subspace stabilization approach is a “divide-and-conquer” approach for nonlinear systems [26, 27]. This approach constructs the control input by two terms as follows

$$u = u_{ss} + v \tag{8}$$

To design the controller with the form of Eq. 8, a fully-actuated subspace of $\boldsymbol{\zeta}$, denoted as $\boldsymbol{w}$, is chosen to be stabilized by the control input u_{ss} at the first step. The dynamic model of the complementary subspace, denoted as $\boldsymbol{w}^{\perp}$, of the stabilized subspace is then obtained by substituting u_{ss} into Eq. 2, and the compensating term v can be thereby designed to stabilize the overall system.

To facilitate the development, the dynamic equations of the quadrotor are rewritten in the following form

$$w^{(k)} = H_1(\xi) + G_1(\xi, u) + d_w(\xi, t) \tag{9a}$$
$$w^{\perp(k)} = H_2(\xi) + G_2(\xi, u) + d_{w^{\perp}}(\xi, t) \tag{9b}$$

where $k \in \{0,\ 1,\ 2\}$ stands for the k^{th} order derivative, $H_1(\cdot)$, $G_1(\cdot)$, $H_2(\cdot)$ and $G_2(\cdot)$ are modelable nonlinear functions, and $d_w(\cdot)$ and $d_{w^{\perp}}(\cdot)$ are unmodeled nonlinearities and disturbances.

The following two assumptions are assumed to hold.

Assumption 1 *The desired trajectory ζ_d in the output space can be obtained by proper design. Its projection in the subspace w and $w^{\perp}$ are denoted as w_d and $w_d^{\perp}$, respectively.*

Assumption 2 Equation 9 *can be well estimated by the equations with the following forms*

$$w^{(k)} = H_1(\xi) + \hat{G}_1(\xi)u \tag{10a}$$
$$w^{\perp(k)} = H_2(\xi) + \hat{G}_2(\xi, u)\hat{v} \tag{10b}$$

where $\hat{G}_1(\xi)$ and and $\hat{G}_2(\xi, u)$ are nonzero functions, and $\hat{v}$ is a virtual control input such that $\hat{v} \in \hat{\boldsymbol{v}} = \hat{\boldsymbol{v}}_{\boldsymbol{w}} \oplus \hat{\boldsymbol{v}}_{\boldsymbol{u}}$ with $\hat{\boldsymbol{v}}_{\boldsymbol{w}} \leq \boldsymbol{w}$ and $\hat{\boldsymbol{v}}_{\boldsymbol{u}} \leq \boldsymbol{u}$.

Remark 1 With the developed dynamic model of Eq. 3 and given boundary value conditions, it is obviously possible to design a feasible trajectory ζ_d by taking the inverse kinematics and dynamics of the quadrotor. This implies Assumption 1 is reasonable. For Assumption 2, as the dynamics of the quadrotor can be estimated as a linear Eq. 3, one can obtain its state-space representation, which is exactly the same form of Eq. 10.

In view of Eq. 10a, to stabilize the subspace $\boldsymbol{w}$, a simplest control input u_{ss} can be designed as follows

$$\begin{cases} u_{ss} = \hat{G}_1^{-1}(\xi)(\alpha_k - H_1(\xi)) \\ \alpha_i = k_{pi}(\alpha_{i-1} - w^{(i-1)}) \qquad 1 \leq i \leq k \end{cases} \tag{11}$$

where k_{pi} is the control gain, and α_i is a virtual control input with $\alpha_0 = w_d$.

Remark 2 Obviously, any other conventional controller is also qualified to stabilize this fully-actuated subsystem. It can be seen that such controllers actually belong to a mapping $\lambda : w_d \mapsto u_{ss}$. If the dynamic model is accurate, the developed controller will closely track the desired trajectory ζ_d according to the forward dynamics and kinematics. However, as there exist lumped disturbances, an additional controller is required to stabilize the whole output space in a closed-loop form.

Substituting u_{ss} into Eq. 10b and taking variations with respect to $w^{\perp(k)}$ and $\hat{v}$, one can obtain the model for $w^{\perp(k)}$

$$\delta w^{\perp(k)} = A\delta w^{\perp(k)} + B\delta\hat{v} \tag{12}$$

where $A = J_{w^{\perp(k)}}(H_2(\xi) + \hat{G}_2(\xi,\ u_{ss})\hat{v})$, and $B = J_{\hat{v}}(H_2(\xi) + \hat{G}_2(\xi,\ u_{ss})\hat{v})$. $J_{w^{\perp(k)}}$ and $J_{\hat{v}}$ denote the Jacobians with respect to $w^{\perp(k)}$ and $\hat{v}$.

To track the desired trajectory, both $\delta w^{\perp}$ and $\delta\hat{v}$ should be suppressed to zero, and this can be expressed as the following optimal problem.

$$\begin{aligned} \min\ \mathcal{J}^* = & \tfrac{1}{2}\delta w_T^{\perp(k)} Q_T^* \delta w_T^{\perp(k)} \\ & + \int_0^T (\tfrac{1}{2}\delta w_t^{\perp(k)} Q_t^* \delta w_t^{\perp(k)} + \tfrac{1}{2}\delta\hat{v}_t R_t^* \delta\hat{v}_t)\mathrm{d}t \end{aligned} \tag{13}$$

To solve the control problem of Eq. 13, the LQR is adopted and the input for Eq. 13 is chosen as [28]

$$\delta\hat{v} = -R_t^{*-1} B^T P(t) \delta w^{\perp(k)} \tag{14}$$

where $P(t)$ is found by solving the continuous time Riccati differential equation

$$A^T P(t) + P(t) A - P(t) B R^{-1} B^T P(t) + Q_t^* = -\dot{P}(t) \tag{15}$$

with the boundary condition $P(T) = Q_T^*$

Therefore, the second term v in Eq. 8 can be obtained by

$$v = \lambda(\delta\hat{v}_w) + \delta\hat{v}_u \tag{16}$$

where $\hat{v}_w \in \hat{\boldsymbol{v}}_{\boldsymbol{w}}$ and $\hat{v}_u \in \hat{\boldsymbol{v}}_{\boldsymbol{u}}$.

4 Application to the Ball-Batting

In this section, the developed control strategy is applied to the task of ball-batting. As shown in Fig. 2, this application aims to bat a ball coming from an arbitrary direction to a desired location. In order to achieve this goal, the quadrotor has to establish the desired position, velocity and attitude at the batting point [20]. Therefore, the output space of the quadrotor in this application is selected as

$$\boldsymbol{\zeta} = \{\zeta | \zeta = [q_1,\ \dot{q}_1,\ q_2,\ \dot{q}_2,\ q_3,\ \dot{q}_3,\ q_4,\ q_5]^T\} \tag{17}$$

According to Eqs. 2 and 4, the control inputs for $\boldsymbol{\zeta}$ is $u_b = [U_1 - mg,\ U_2,\ U_3]$. This means the dimension of w must be less than or equal to three. Considering the fact that q_4 and q_5 possess different dynamic behaviors compared to q_1, q_2 and q_3 [24], the full-actuated subspace $\boldsymbol{w}$ is chosen as $\boldsymbol{w} = \{w | w = [q_4,\ q_5]^T\}$. Then $\boldsymbol{w}^{\perp}$ is determined as $\boldsymbol{w}^{\perp} = \{w^{\perp} | w^{\perp} = [q_1,\ \dot{q}_1,\ q_2,\ \dot{q}_2,\ q_3,\ \dot{q}_3]^T\}$.

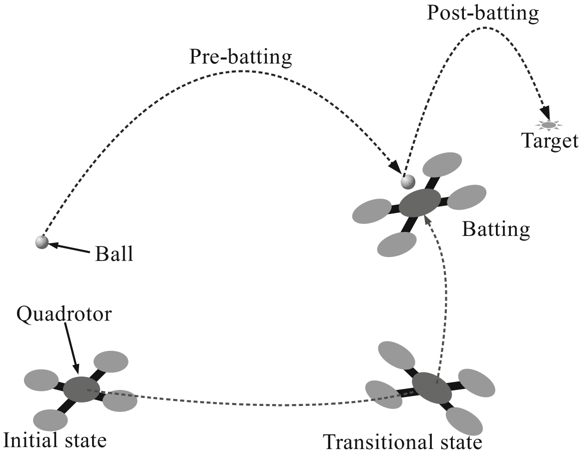


Fig. 2 An illustration for the ball batting

4.1 Trajectory Design

According to the development in Section 3, it is necessary to design a referenced trajectory $\zeta_d(t)$ $(t \in [0,\ T])$ prior to the controller design. In this work, $\zeta_d(t)$ is constructed with two segments in the time domain $t \in [0,\ t_1)$ and $t \in [t_1, T]$.

Denoting the state of the quadrotor in the batting point as $\zeta(T) = \zeta_T^*$, the trajectory $\zeta_d(t)$ in $t \in [t_1, T]$ can be designed to meet this boundary condition in the following way.

Firstly, the desired trajectory in w is selected as

$$\theta(t) = \theta_d,\ \phi(t) = \phi_d \quad (t \in [t_1, T]) \tag{18}$$

where θ_d and ϕ_d are the desired attitude to bat the ball.

Subsequently, adopting a constant collective thrust $U_1 = mg + U_{10}$ and taking integration of Eq. 18 with respect to Eq. 3, the desired trajectory of the quadrotor in the output space $\boldsymbol{\zeta}$ is $(t \in [t_1, T])$

$$\zeta_d(t) = \begin{bmatrix} \theta_d(t) \\ \phi_d(t) \\ \dot{x}_d(t) \\ \dot{y}_d(t) \\ \dot{z}_d(t) \\ x_d(t) \\ y_d(t) \\ z_d(t) \end{bmatrix} = \begin{bmatrix} \theta_d \\ \phi_d \\ \frac{U_{10}}{m}\theta_d(t - t_1) \\ -\frac{U_{10}}{m}\phi_d(t - t_1) \\ \frac{U_{10}}{m}(t - t_1) \\ \frac{U_{10}}{2m}\theta_d(t - t_1)^2 + x_d^*(t_1) \\ -\frac{U_{10}}{2m}\phi_d(t - t_1)^2 + y_d^*(t_1) \\ \frac{U_{10}}{2m}(t - t_1)^2 + z_d^*(t_1) \end{bmatrix} \tag{19}$$

with $\zeta_d(T) = \zeta_T^*$, which will be calculated in Section 4.4, and $x_d^*(t_1)$, $y_d^*(t_1)$ and $z_d^*(t_1)$ determines the desired position of the quadrotor at $t = t_1$.

The trajectory $\zeta_d(t)$ in $t \in [0,\ t_1)$ is then defined with two set-points: the initial position, which is the current position of the quadrotor, and the desired final position $r_d^* = [x_d^*(t_1),\ y_d^*(t_1),\ z_d^*(t_1)]^T$. With this design, the quadrotor only need to be controlled to reach the desired position by a position controller in $t \in [0,\ t_1)$. Thereafter, the developed under-actuated control strategy takes over the control tasks and manipulate the quadrotor to establish the desired final state ζ_T^*.

4.2 Controller Design

According to Eqs. 3 and 9a, the dynamic model of the subspace $\boldsymbol{w}$ is estimated by

$$w^{(2)} = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 \\ 0 & \frac{1}{I_{yy}} \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \end{bmatrix} \tag{20}$$

It can be seen (20) is a linear time-invariant system. A proportional-derivative controller is developed for this system

$$u_{wi} = k_d(\alpha - \dot{w}_i),\ \alpha = k_p(w_{di} - w_i) \tag{21}$$

where $i \in \{1, 2\}$, k_p and k_d are the controller gains, and w_{di} is the desired value for w_i.

Assuming $\ddot{z} = 0$, the control input to stabilize the subspace w is thereby constructed as

$$u_{ss} = [0,\ u_{w1},\ u_{w_2}]^T \tag{22}$$

Subsequently, denoting $\hat{v} = [U_1 - mg,\ w_1,\ w_2]^T$, based on Eqs. 3 and 17, the dynamics of the complementary subspace $\boldsymbol{w}^{\perp}$ can be expressed in a discrete form

$$w^{\perp}_{(k+1)} = \begin{bmatrix} w^{\perp}_{1(k)} + w^{\perp}_{2(k)}\delta t \\ w^{\perp}_{2(k)} + g\hat{v}_{3(k)}\delta t \\ w^{\perp}_{3(k)} + w^{\perp}_{4(k)}\delta t \\ w^{\perp}_{4(k)} - g\hat{v}_{2(k)}\delta t \\ w^{\perp}_{5(k)} + w^{\perp}_{6(k)}\delta t \\ w^{\perp}_{6(k)} + \frac{\hat{v}_{1(k)}}{m}\delta t \end{bmatrix} = f(w^{\perp}_{(k)}, \hat{v}_{(k)}) \tag{23}$$

Taking variations of Eq. 23 with respect to $w^{\perp}$ and $\hat{v}$, one can obtain

$$\delta w^{\perp}_{(k+1)} = A_{(k)}\delta w^{\perp}_{(k)} + B_{(k)}\delta\hat{v}_{(k)} \tag{24}$$

where $A_{(k)} = J_{w^{\perp}} f(w^{\perp}_{(k)}, \hat{v}_{(k)})$, and $B_{(k)} = J_{\hat{v}} f(w^{\perp}_{(k)}, \hat{v}_{(k)})$. $J_{w^{\perp}} f$ and $J_{\hat{v}} f$ denote the Jacobians of $f(\cdot)$ with respect to $w^{\perp}$ and $\hat{v}$. In this study, $A^{(k)}$ and $B^{(k)}$ are constant matrices as follows

$$A = \begin{bmatrix} 1 & \delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 0 & \frac{g\delta t^2}{2} & 0 \\ 0 & g\delta t & 0 \\ 0 & 0 & -\frac{g\delta t^2}{2} \\ 0 & 0 & -g\delta t \\ \frac{\delta t^2}{2m} & 0 & 0 \\ \frac{\delta t}{m} & 0 & 0 \end{bmatrix} \tag{25}$$

Thus, an optimal control problem is stated as

$$\min \mathcal{J}^* = \sum_{k=1}^{N}(\delta w_k^{\perp} Q \delta w_k^{\perp} + \delta\hat{v}_k R \delta\hat{v}_k) \tag{26}$$

In view of Eq. 26, a LQR-based controller is constructed as

$$\delta\hat{v} = -K_k \delta w^{\perp} \tag{27}$$

where

$$K_k = (R + B^T P_k B)^{-1} B^T P_k A \tag{28}$$

P_k can be estimated iteratively backwards in time by

$$P_{k-1} = Q + A^T(P_k - P_k B(R + B^T P_k B)^{-1} B^T P_k)A \tag{29}$$

with initial condition $P_N = Q$.

Consequently, the control input for the whole system can be constructed based on Eqs. 22 and 27 as follows

$$u^*_{b(k)} = u_{ss(k)} + \lambda(\delta\hat{v}_{w(k)}) + \delta\hat{v}_{u(k)} \tag{30}$$

4.3 Eliminate the Input Delay by Smith Predictor

In the real-time flights, there exist input delays caused by the transport delays in the communications and the dynamics of the quadrotors [23]. To eliminate this effects, the Smith predictor is adopted in a similar manner as in [29, 30]. As indicated in Fig. 3, the Smith Predictor uses a nominal model G_p to predict the delay-free response y_p of the actual process P. The control input for the developed controller C is then calculated according to the difference between the desired value y_r and the predicted y_p. Meanwhile, to eliminate the effects of the transport delays, modeling errors and external disturbances d, the Smith predictor further compares the actual response y with the

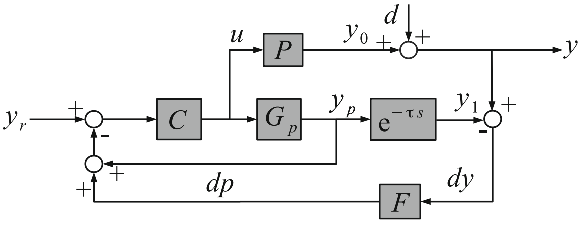


Fig. 3 Smith predictor

Algorithm 1 Estimate the state of the ball

```
for $i = 1$ to $N$ do
    Obtain the point set $\mathcal{P}_i$ which contains the coordinates of all of the recognized points [33].
    Construct a domain $\mathcal{S}_i = \Gamma(\mathcal{P}_i)$ where $\Gamma(\mathcal{P}_i) = \{\gamma | \rho_l \le ||\gamma - p_{ij}|| \le \rho_h, p_{ij} \in \mathcal{P}_i\}$. $\mathcal{S}_i$ represents the range of movement of $\mathcal{P}_i$ in view of Eq. 31.
    Find the free flying point set $\mathcal{A}_i = \{a | a \in (\mathcal{S}_{i-1} \bigcap \mathcal{S}_i \bigcap \mathcal{P}_i)\}$
    Add $\mathcal{A}_i$ as initial points of new trajectories in the trajectory set $\mathcal{T}$, and update $\mathcal{S}_i$ by $\mathcal{S}_i = \mathcal{S}_i - \Gamma(\mathcal{A}_i)$
    for $l = 1$ to count($\mathcal{T}$) do
        $k$=count($\mathcal{T}_l$), where $\mathcal{T}_l$ is the $l^{th}$ trajectory and $k$ is the number of samples of of this trajectory
        if $\mathcal{T}_{lk} \in \mathcal{S}_{ij} \subset \mathcal{S}_i$ then
            $k = k + 1$
            $\mathcal{T}_{lk}.Position = \Gamma^{-1}(\mathcal{S}_{ij}),\ \mathcal{T}_l.MissingCount = 0$
            Run Kalman filter, update the state, including the velocity, deviation, etc., of $\mathcal{T}_{lk}$
        else
            $\mathcal{T}_l.MissingCount = \mathcal{T}_l.MissingCount + 1$
            if $\mathcal{T}_l.MissingCount > 5$ then
                Remove $\mathcal{T}_l$
            end if
        end if
        Estimate the state $\mathcal{T}_{lf}$ in the batting point and the batting time $tm_l$ based on Eq. 31.
    end for
    Output $\{\mathcal{T}_{lf}\}$ and $\{tm_l\}$
end for
```

prediction $y_1 = y_p e^{-\tau s}$, where $e^{-\tau s}$ is the estimated transport delay of the actual process. In this way, the difference dy can be fed back through a low-pass filter F to compensate for the lumped disturbances and transport delays.

4.4 Estimate the State of the Free Flying Ball

To determine the terminal boundary condition for the quadrotor in the ball-batting, the state of the ball must be well estimated [20]. For this reason, an algorithm is developed to capture the motion of the ball in the real-time tests.

Denoting the position of the ball in the inertial coordinates as r_b, the dynamics of a free flying ball can be described as [31, 32]

$$\begin{cases} \dot{r}_b = v_b \\ \dot{v}_b = -gZ_I - \frac{1}{2m_b} C_d \rho S \circ |v_b| \circ v_b \end{cases} \tag{31}$$

where C_d is the drag coefficient, ρ is the density of the air, S is the referenced area, and the symbol $\circ$ denotes the element-wise product.

With the dynamics described by Eq. 31, Algorithm 1 is designed to capture the motion of the ball and predict the state at the batting point. Combining this prediction with the ball's target location after the batting, the post-batting state of the ball can be calculated. For the detailed calculation, the interested readers are referred to [20].

Denoting the pre- and post-batting state of the ball as r_b^- and r_b^+, the terminal state of the quadrotor is determined as

$$\begin{cases} [x_d(T), y_d(T), z_d(T)]^T = r_b^- \\ [\dot{x}_d(T), \dot{y}_d(T), \dot{z}_d(T)]^T = \frac{1}{1+\beta}(\beta \dot{r}_b^- + \dot{r}_b^+)^T n \\ [\theta_d(T), \phi_d(T)]^T = [\tan^{-1}(\frac{n_1}{n_3}), \sin^{-1}(n_2)]^T \end{cases} \tag{32}$$

where β is the COR, and $n = \frac{\dot{r}_b^- - \dot{r}_b^+}{||\dot{r}_b^- - \dot{r}_b^+||}$.

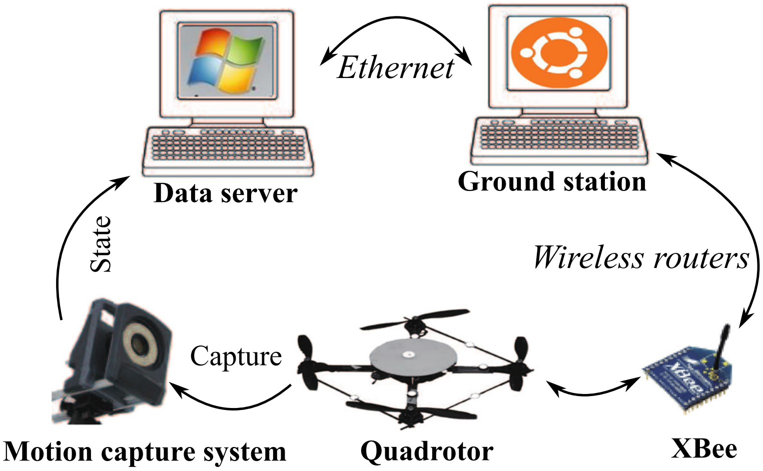
5 Experiments

Real-time experiments are carried out in this section to demonstrate the effectiveness of the developed control strategy.

5.1 Experimental Setup

The experiments are conducted in an indoor test bed, as shown in Fig. 4, where a commercial quadrotor platform, namely the Hummingbird quadrotor, is adopted. This quadrotor communicates with a ground control station via a couple of XBee wireless routers at a frequency of 50 Hz. To meet the requirements of real-time flight control and improve the performance, the control station runs on a Linux (Ubuntu 13.10) operating system and is constructed in the Simulink environment. This control station estimates the state of the quadrotor and the bouncing ball based on the data fetched from the data sever, which runs on

Fig. 4 The quadrotor test bed



the Windows Server 2008, of the Vicon motion capture system. The estimated state is then adopted as the feedback for the developed control strategy. The motion capture system runs at a frequency of 200 Hz, and the quadrotor is captured by kinematic fitting of the reflective markers which are attached on the quadrotor.

The bouncing ball and the racket are shown in Fig. 5. The ball is a 20 mm bouncing ball covered by a thin layer of reflective powder, whereas the racket is printed by a 3D printer and covered by a high-elastic sponge with thickness of 2.2 mm. The radius of the racket is 65 mm.

5.2 Identification and Parameter Tunning

In order to correctly estimate the state of the quadrotor in the batting point, the COR between the ball and racket is identified firstly. This is accomplished by conducting a series of random batting with the adopted racket and bouncing ball in the test bed. The results are shown in Fig. 6. Although the racket face is solid, there still exits a sweet spot. This results in a highest COR, roughly 0.7, at the center of the racket, and the COR decreases outwards. This means the post-batting velocity of the ball decreases with the tracking error of the position. In addition, compared to the racket used in [20], the COR is smaller in this work. This requires the quadrotor to establish a higher final velocity such that the ball can reach its desired post-batting velocity.

With the estimated COR, the trial and error method is adopted for parameter tunning. In this way, the control gains for the controller stabilizing the subspace w are selected as

$$k_p = 10, k_d = 12 \tag{33}$$

The weighting matrix Q and R for the controller stabilizing subspace $w^\perp$ are designed as diagonal

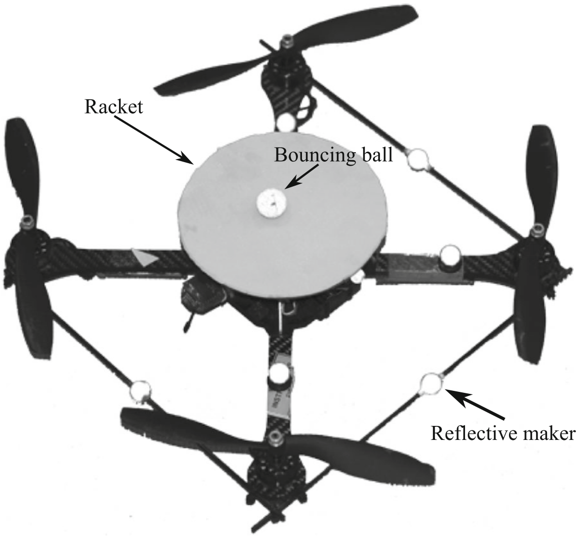


Fig. 5 The quadrotor and the bouncing ball

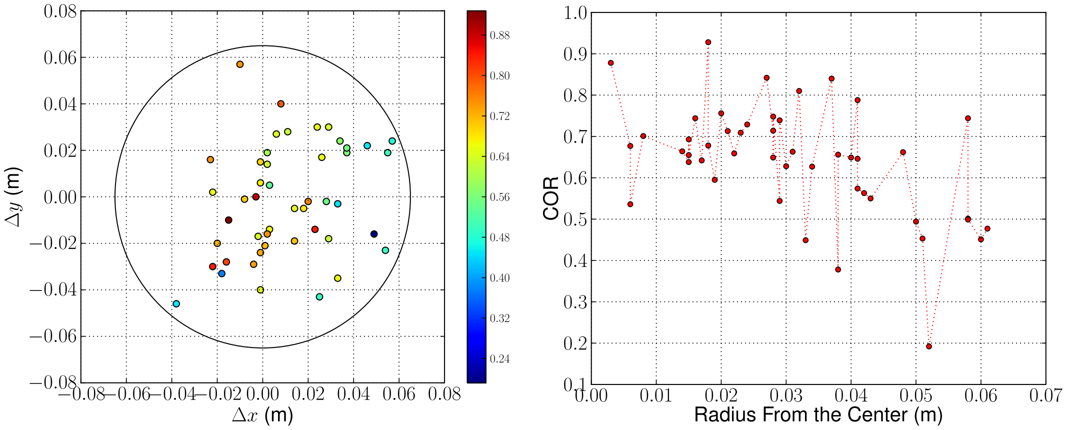


Fig. 6 The estimated coefficient of restitution of the selected bouncing ball and racket

matrices, and their diagonal entries are selected as

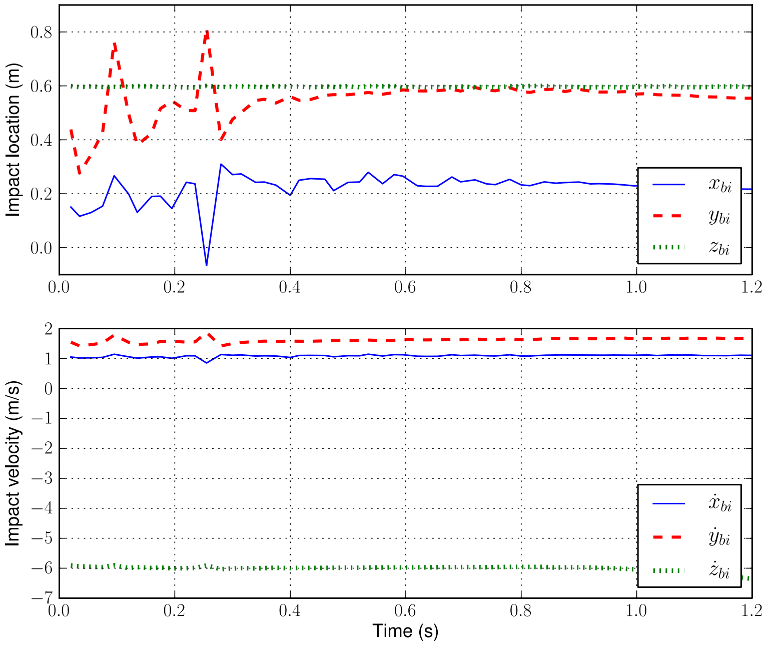
$$Q_{11} = 50,\ Q_{22} = 1,\ Q_{33} = 50,\ Q_{44} = 1,\ Q_{55} = 10,\ Q_{66} = 1,\quad R_{11} = 1,\ R_{22} = 1,\quad R_{33} = 1 \tag{34}$$

5.3 Experimental Results

In this experiment, the quadrotor is initially hovering at the location of (0, 0, 0.4). The bouncing ball is then thrown into the test bed from the location of (−1.5, −1.5, 1), and the developed control strategy manipulates the quadrotor to bat the ball at the horizontal plane $z = 0.6$ and direct it to the origin of the same horizontal plane. After this batting, the quadrotor also returns to the center of the test bed.

In this process, Algorithm 1 predicts the state of the ball at the batting point at the first step, and the results are shown in Fig. 7. It can be seen that at the very beginning the predicted value is not stable. This is because the noises introduced in the measurement cannot be effectively filtered out with the limited number of samples at the beginning. Nevertheless, after

Fig. 7 The state of the ball in the batting point predicted by the developed Algorithm 1



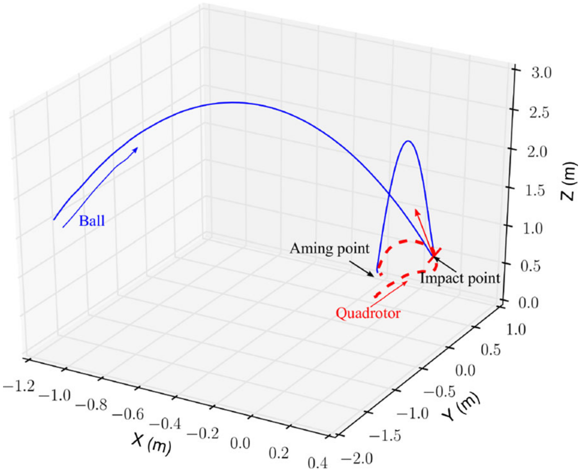


Fig. 8 The 3D plot from the result of the ball batting

approximate 0.4 s, this prediction converges to a stable value. The standard deviation of the predicted batting position is approximate 0.05 m.

With the predicted state of the ball, the control strategy is able to estimate the desired final state of the quadrotor and manipulate the quadrotor to bat the ball. Figure 8 shows the results from the real-time experiments, the ball is batted by the quadrotor at (0.20, 0.52, 0.72) and returns to the position (0.05, -0.03, 0.72). Figure 9 plots the history of position and Fig. 10 shows the history of speed and attitude. In Fig. 10, the markers "+" indicate the desired values for the corresponding state variables at the batting point. The state tracking errors and the response time that indicate the performance of the developed control strategy are listed in Table 1. It can be seen that in the whole process, the response of the developed control strategy is pretty fast. After the bouncing ball is thrown out, the quadrotor completes the path planning and starts the real-time trajectory tracking with $t_a < 0.1$ s, and the quadrotor finally bats the ball at $t_r = 1.18$ s. Thereafter, it takes $t_b < 0.1$ s to start the returning process, and the quadrotor finally reaches the center of the test bed within $t_h = 1.15$ s. At the batting point, all of the eight states are well brought to their desired values with only small deviations, i.e. the final state tracking errors $\Delta[\cdot]_q$ in Table 1. The imperfect aspect is that there are still perceptible deviations occurring in the altitude control. This might be because the required vertical velocity is higher than the translational velocity and the vertical motion itself owns fast dynamics according to Eq. 3. Nevertheless, such deviation does not significantly affect the final results. This can be seen from the fact that the ball is finally batted to the desired location of the same horizontal plane with only small aiming errors $\Delta[\cdot]_a$ as shown in Table 1, Figs. 8 and 9, where the aiming errors represent the achieved minimum distances of the ball towards its desired target location. The real-time scenario of this

Fig. 9 The history of position

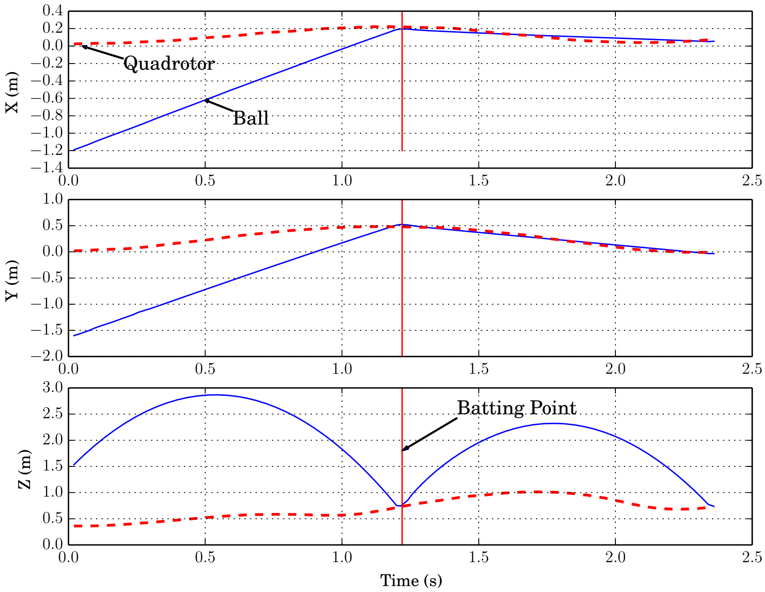
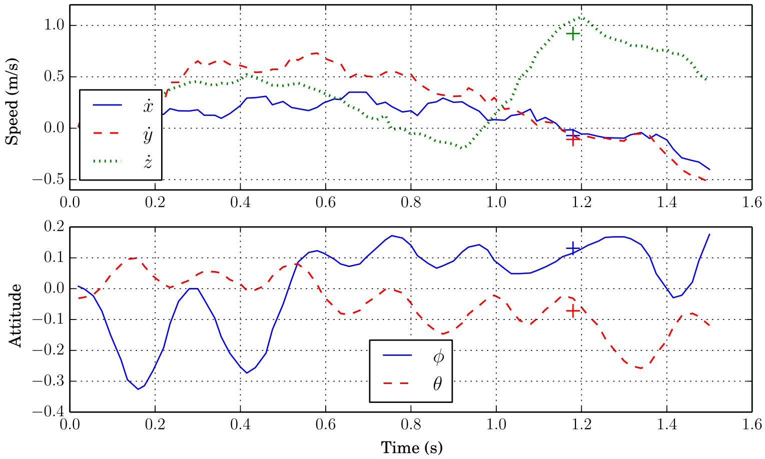


Fig. 10 The history of speed and attitude



experiments is indicated in Fig. 11, where a series of images showing the batting process are combined into a single picture.

To further demonstrate the effectiveness of the developed control strategy, four additional tests batting the ball to different target locations are carried out. As shown in Fig. 12, with the bouncing ball thrown out from nearly a same initial location, the quadrotor is commanded to bat the bouncing ball to four different target locations (-1.0,0,0.6), (-0.5,0,0.6), (0.5,0,0.6), and (1,0,0.6), which are abbreviated in the discussions as $x^* = -1.0$, $x^* = -0.5$, $x^* = 0.5$, and $x^* = 1.0$, respectively. The aiming errors of these four tests are shown in Table 2, where the maximum error $\Delta x_a = 0.15$ m occurs in the test $x^* = 1.0$ m. It should be noted that according to [20], when there is an attitude deviation of 5° in the final state of the quadrotor at the batting point, the aiming error of the ball will be as large as 1.4 m. Therefore, the small aiming errors of these four tests in Fig. 12 imply that the final state of the quadrotor is well established. Besides, the response rates of the quadrotor in all of these tests are nearly the same, which thereby demonstrate the real-time performance of the developed control strategy.

Table 1 The performance of the proposed control strategy corresponding to Fig. 9

	Item	Value	Item	Value
Final	Δx_q (m)	0.02	$\Delta \dot{x}_q$ (m/s)	0.03
state	Δy_q (m)	0.04	$\Delta \dot{y}_q$ (m/s)	0.04
tracking	Δz_q (m)	0.12	$\Delta \dot{z}_q$ (m/s)	0.16
errors	$\Delta \theta_q$	0.02	$\Delta \phi_q$	0.01
Aiming	Δx_a (m)	0.05	Δy_a (m/s)	-0.03
errors	Δz_a (m)	0.12		
Time	t_r (s)	1.18	t_a (s)	<0.1
	t_h (s)	1.15	t_b (s)	<0.1

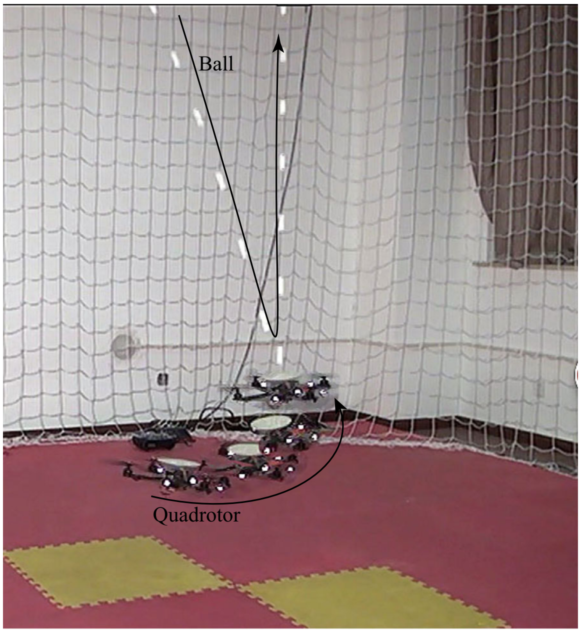


Fig. 11 The composite image of the quadrotor batting the bouncing ball in the real-time experiments

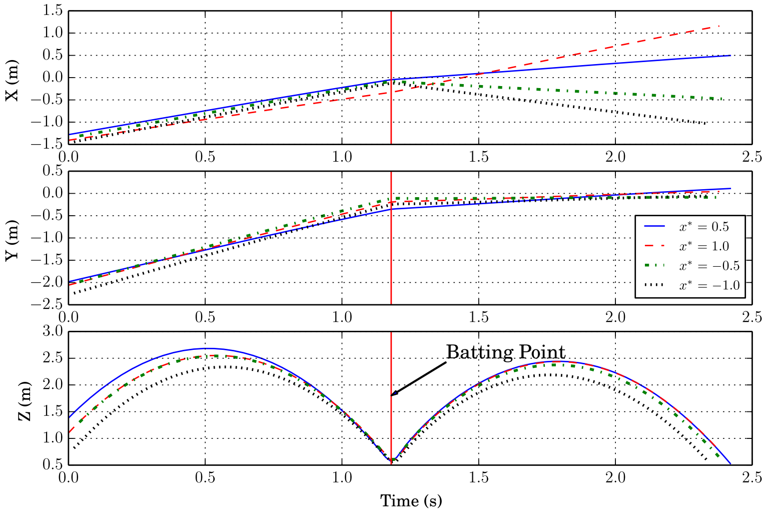


Fig. 12 The experimental results of the bouncing ball towards different target locations

Table 2 The aiming errors of the four tests shown in Fig. 12

Target x^* (m)	-1.0	-0.5	0.5	1.0
Δx_a (m)	-0.02	0.03	-0.02	0.15
Δy_a (m)	-0.06	-0.08	0.09	0.04
Δz_a (m)	-0.04	-0.02	-0.06	0.14

It can be seen that all of the results prove that the developed control strategy can well manipulate the quadrotor from a known initial state to a desired final state within a limited time horizon, and in this way, the bouncing ball can be batted to the target location with relatively high accuracy. All of these characteristics demonstrate the effectiveness of this development.

6 Conclusion

To solve the two-point boundary value problem of a highly under-actuated quadrotor, this work has developed a control strategy based on the LQR and subspace stabilization approach. This strategy is constructed in a closed-loop form and requires no further iterative learning along with successive experimental trials, thus shows the advantages in view of the state-of-the-art of the quadrotors. To verify the effectiveness of this development, real-time experiments are carried out and the task of batting a ball towards a target location is performed. Experimental results show that the quadrotor can well establish the desired final state within a limited time horizon, and thereby, be capable of batting the ball towards its target location with satisfactory accuracy (with a maximum deviation of 0.15 m in position).

Based on these convincing results, the development of this work is believed to be able to well extend the capabilities of the quadrotors in real-time flights. In addition, it can be seen that the developed control strategy is possible to be applied to some conventional control tasks such as the flight control and waypoint navigation, since these control tasks can also be reasonably formulated as two-point boundary value problems with the desired state-trajectory designed in the fully-actuated output space.

Acknowledgments This work was funded by the Open Foundation of the State Key Laboratory of Fluid Power Transmission and Control.

References

1. Kumar, V., Michael, N.: Opportunities and challenges with autonomous micro aerial vehicles. Int. J. Robot. Res. **31**(11), 1279–1291 (2012)

2. Abdolhosseini, M., Zhang, Y., Rabbath, C.A.: An efficient model predictive control scheme for an unmanned quadrotor helicopter. J. Intel. Robotic Syst. **70**(1-4), 27–38 (2013)
3. Lim, H., Park, J., Lee, D., Kim, H.: Build your own quadrotor: Open-source projects on unmanned aerial vehicles. IEEE Robotics Autom. Mag. **19**(3), 33–45 (2012)
4. Bouabdallah, S.: Design and control of quadrotors with application to autonomous flying. Ph.D. thesis, Lausanne Polytechnic University (2007)
5. Pounds, P., Mahony, R.: Design principles of large quadrotors for practical applications. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 3265–3270 (2009)
6. Bouabdallah, S., Noth, A., Siegwart, R.: Pid vs lq control techniques applied to an indoor micro quadrotor. In: Proceedings of IEEE international conference on intelligent robots and systems, vol. 3, pp. 2451–2456 (2004)
7. Dong, W., Gu, G.Y., Zhu, X., Ding, H.: High-performance trajectory tracking control of a quadrotor with disturbance observer. Sensors and Actuators A Phys. **211**, 67–77 (2014). doi:10.1016/j.sna.2014.03.011
8. Kendoul, F.: Nonlinear hierarchical flight controller for unmanned rotorcraft: design, stability, and experiments. J. Guid. Control. Dyn. **32**(6), 1954–1958 (2009)
9. Tayebi, A., McGilvray, S.: Attitude stabilization of a vtol quadrotor aircraft. IEEE Trans. Control Syst. Technol. **14**(3), 562–571 (2006)
10. Chamseddine, A., Zhang, Y., Rabbath, C.A., Theilliol, D.: Trajectory planning and replanning strategies applied to a quadrotor unmanned aerial vehicle. J. Guid. Control Dyn **35**(5), 1667–1671 (2012)
11. He, R., Bachrach, A., Achtelik, M., Geramifard, A., Gurdan, D., Prentice, S., Stumpf, J., Roy, N.: On the design and use of a micro air vehicle to track and avoid adversaries. Int. J. Robot. Res. **29**(5), 529–546 (2010)
12. Mahony, R., Kumar, V., Corke, P.: Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. IEEE Robot Autom. Mag. **19**(3), 20–32 (2012)
13. Orsag, M., Korpela, C., Oh, P.: Modeling and control of mm-uav: Mobile manipulating unmanned aerial vehicle. J. Int. Robot. Syst. **69**(1-4), 227–240 (2013)
14. Metni, N., Hamel, T.: A uav for bridge inspection: Visual servoing control law with orientation limits. Automation in construction **17**(1), 3–10 (2007)
15. Fang, Z., Gao, W.: Adaptive backstepping control of an indoor micro-quadrotor. Res. J. Appl. Scince Eng. Technol. **4**(21), 4216–4226 (2012)
16. Oriolo, G., Nakamura, Y.: Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators. In: Proceedings of the 30th IEEE Conference on Decision and Control, pp. 2398–2403 (1991)
17. Rui, C., Reyhanoglu, M., Kolmanovsky, I., Cho, S., McClamroch, N.: Nonsmooth stabilization of an underactuated unstable two degrees of freedom mechanical system. In: Proceedings of the 36th IEEE conference on decision and control, vol. 4, pp. 3998–4003. IEEE (1997)
18. Brockett, R.W., et al.: Asymptotic stability and feedback stabilization. Differential geometric control theory, pp. 181–191 (1983)
19. Mellinger, D., Michael, N., Kumar, V.: Trajectory generation and control for precise aggressive maneuvers with quadrotors. Int. J. Robot. Res. **31**(5), 664–674 (2012)
20. Muller, M., Lupashin, S., D'Andrea, R.: Quadrocopter ball juggling. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems, pp. 5113–5120 (2011)
21. Mueller, M.W., Hehn, M., D'Andrea, R.: A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasibility verification. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems, pp. 3480–3486 (2013)
22. Wikipedia: Coefficient of restitution http://en.wikiped-ia.org/wiki/Coefficient_of_restitution. Accessed 21 (2014)
23. Bouabdallah, S., Murrieri, P., Siegwart, R.: Design and control of an indoor micro quadrotor. In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 5, pp. 4393–4398 (2004)
24. Dydek, Z.T., Annaswamy, A.M., Lavretsky, E.: Adaptive control of quadrotor uavs: A design trade study with flight evaluations. IEEE Trans. Control Syst. Technol. **21**(4), 1400–1406 (2013)
25. Wikipedia: Configuration space. http://en.wikipedia.org/wiki/Configu-ration_space. Accessed 22 (2014)
26. Márton, L., Hodel, A.S., Lantos, B., Hung, J.Y.: Underactuated robot control: comparing lqr, subspace stabilization, and combined error metric approaches **55**(10), 3724–3730 (2008)
27. Simmons, A.T., Hung, J.Y., Hodel, A.S.: A hybrid improvement to traditional nonlinear control. In: Proceedings of IEEE international symposium on industrial electronics, vol. 1, pp. 49–56. IEEE (2005)
28. Athans, M.: The role and use of the stochastic linear-quadratic-gaussian problem in control system design. IEEE Trans. Autom. Control **16**(6), 529–552 (1971)
29. Ingimundarson, A., Hägglund, T.: Robust tuning procedures of dead-time compensating controllers. Control. Eng. Pract. **9**(11), 1195–1208 (2001)
30. MathWorks: Control of processes with long dead time: The smith predictor. http://www.mathworks.cn/cn/help/control/examples/control-of-processes-with-long-dead-time-the-smith-predictor.html. Accessed 26 February (2014)
31. Nonomura, J., Nakashima, A., Hayakawa, Y.: Analysis of effects of rebounds and aerodynamics for trajectory of table tennis ball. In: Proceedings of SICE Annual Conference, pp. 1567–1572 (2010)
32. Yang, Z.J., Hara, S., Kanae, S., Wada, K., Su, C.Y.: An adaptive robust nonlinear motion controller combined with disturbance observer. IEEE Trans. Control Syst. Technol. **18**(2), 454–462 (2010)
33. Vicon: Vicon mx systems. http://www.vicon.com/products/viconmx.html. Accessed 10 October (2012)