



# Controlling a Quadrotor Carrying a Cable-Suspended Load to Pass Through a Window

Minhuan Guo<sup>1</sup> · Dongbing Gu<sup>2</sup> · Wenzhong Zha<sup>3</sup> · Xinhua Zhu<sup>4</sup> · Yan Su<sup>4</sup>

Received: 7 October 2018 / Accepted: 8 May 2019 / Published online: 23 May 2019  
© Springer Nature B.V. 2019

## Abstract

In this paper, we design an optimal control system for a quadrotor to carry a cable-suspended load flying through a window. As the window is narrower than the length of the cable, it is very challenging to design a practical control system to pass through it. Our solution includes a system identification component, a trajectory generation component, and a trajectory tracking control component. The exact dynamic model that usually derived from the first principles is assumed to be unavailable. Instead, a model identification approach is adopted, which relies on a simple but effective low order equivalent system (LOES) to describe the core dynamical characteristics of the system. After being excited by some specifically designed manoeuvres, the unknown parameters in the LOES are obtained by using a frequency based least square estimation algorithm. Based on the estimated LOES, a numerical optimization algorithm is then utilized for aggressive trajectory generation when relevant constraints are given. The generated trajectory can lead to the quadrotor and load system passing through a narrow window with a cascade PD trajectory tracking controller. Finally, a practical flight test based on an Astec Hummingbird quadrotor is demonstrated and the result validates the proposed approach.

**Keywords** Micro aerial vehicles · Model identification · Optimal control · Trajectory generation

## 1 Introduction

Nowadays, unmanned aerial vehicles (UAVs) are gaining more and more popularity in many applications, such as last-mile deliveries [1], wireless communications [2], disaster relief operations [3] and acrobat demonstration [4]. Quadrotors have also been applied to payload manipulation. For example, Quentin Lindsey et al. [5] utilized quadrotor teams to construct cubic structures. Justin Thomas et al. [6] designed and equipped a quadrotor with an actuated appendage enabling grasping at high speed. Robin Ritz et al. [7] introduced a method for carrying a flexible payload with multiple flying vehicles.

As discussed in [8], a load held by a gripper can undesirably increase the inertia of the system. Instead, attaching the load to a quadrotor via a cable suspension

could retain more of the vehicle's agility. Many papers have been published in terms of stabilization of the load or minimization of the load swing while traversing trajectories. Cruz et al. [9] addressed the problem of lifting a cable-suspended load from the ground by a quadrotor, where the mass of the load is unknown. Alothman et al. [10] proposed a linear quadratic regulator (LQR) for lifting and transporting the load. A fixed-gain nonlinear proportional-derivative controller was presented in [11] to transport the load to a desired position while aligning the links along the vertical direction from an arbitrary initial position. Frust et al. [12] presented a motion planning method for generating trajectories with minimal residual oscillations and completed the multi-waypoint flight in the cluttered environment. Palunko et al. [13] used a high-level planner to provide desired waypoints and utilized a dynamic programming approach to generate the swing-free trajectory to keep the minimum load swing.

The problem has also been described as a multi-phase model. Koushil Sreenath et al. [14] established the quadrotor-load system to be a differentially-flat hybrid system and developed a nonlinear geometric control which exhibits almost-global properties in each simulation case. Furthermore, the single vehicle system was extended to

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s10846-019-01038-6>) contains supplementary material, which is available to authorized users.

✉ Minhuan Guo  
guominhuan@jssvc.edu.cn

multi-vehicles in [15], which addresses the problem of cooperative transportation of a cable-suspended load by multiple quadcopters.

With regard to trajectory generation, De Crousaz et al. [16] addressed planning and control problems for the quadrotor with payload system using an iterative LQG (iLQG) algorithm. Another popular approach is to plan trajectories directly in the flat space using differential flatness. Koushil Sreenath et al. [14] studied how the quadrotor trajectory evolves as the frequency of the load trajectory varies. Sarah Tang et al. [17] addressed navigating a similar system through obstacle-filled environments by formulating the hybrid system as a Mixed Integer Quadratic Programming (MIQP) problem and then searching the optimal coefficients for the polynomial basis in the flat space. Similarly in [8], choosing a trigonometric basis in the flat space, they designed an aggressive trajectory using Quadratic Programming (QP).

In the researches mentioned above, the system's dynamic model are derived from the First Principle without considering the unknowns in practical systems. It is very challenging to match the theoretical models with practical systems. An engineering solution to this is to identify the systems via collecting experiment data.

Basically, the system identification for UAVs is a procedure by which a mathematical description of UAV dynamic behaviour is extracted from experiment data. The work in [18] detailed the basic issues of system identification for manned aircraft, including dynamic model structure, estimation theories as well as engineering practices. The work in [19] described the detailed identification approach mainly from the perspective of frequency-domain. The work in [20] presented a survey and categorization of current methods and applications for small low-cost UAV, for example, helicopter, fixed-wing, multirotor, flapping-wing, and lighter-than-air. The work in [21, 22] both used state space equations as the system model and the unknown parameters were estimated through fitting the model to the frequency response extracted from flight data. The work in [23–26] chose neural networks as the dynamic model and presented the corresponding identification method. The work in [27] summarized the experiences using CIFER tool for the modeling of UAV flight dynamics. The work in [28] utilized a time domain system identification software (SIDPAC) developed by NASA Langley Research Center to estimate a linear mathematical model for a micro quadrotor. Apart from those off-line approaches, some papers have been published about real-time system identification methods [29, 30]. The work in [31, 32] used a least square method in the frequency domain to estimate the unknown aerodynamic coefficients for manned vehicles. These publications are mainly aimed at fixed-wing planes. Potentially, they could be used for quadrotors when well-designed excitation inputs are designed.

Other related research on the control aspect of quadrotor and payload carrying systems includes the work of [33] and [34], where an indirect optimal control method was applied to generate the time-optimal trajectory for the quadrotor, and the work [35] where a novel method for learning optimal control solutions and generalizing them in real-time for a quadrotor-type vehicle with flat dynamics was presented.

In this paper, we will address a full solution to the practical control problem of a quadrotor carrying a cable-suspended load flying through a narrow window. Different from the similar work in [8, 17], it is assumed that the exact non-linear dynamic model is unavailable. Thus, we will present an easy but effective low order equivalent system (LOES) to describe the system's core dynamical characteristics and a corresponding estimation algorithm. It is also assumed that the best trajectory for the quadrotor and payload passing through a narrow window is not available. Thus we will present an optimal trajectory generation algorithm, which can generate an aggressive trajectory to pass through a narrow window based on the identified results. Finally we will adopt a cascade PD controller for the system to track the generated trajectory. In summary, our main contributions in this paper are listed as follows.

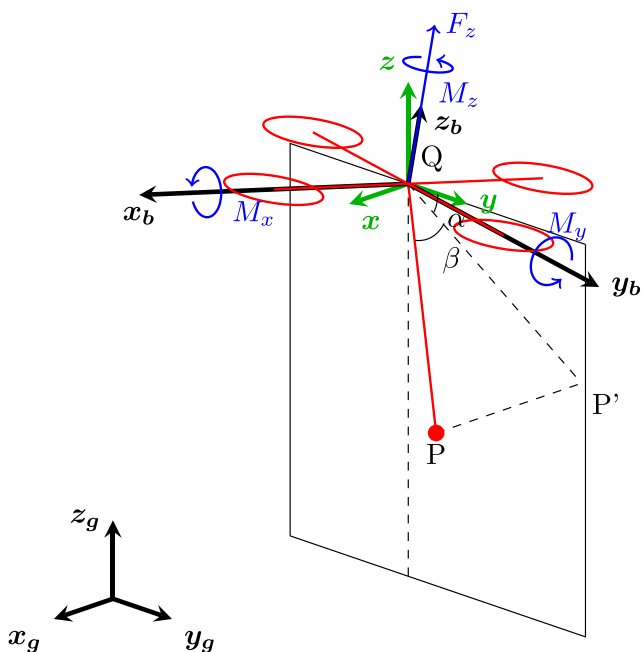
- 1) We describe the core dynamical characteristics with a low order equivalent system rather than the full complicated non-linear model. Then we design a series of specific manoeuvres to fully excite the model and implement an estimation algorithm for the unknown parameters.
- 2) Based on the above identification results, we apply a numerical optimization algorithm to generate an aggressive trajectory which meets the relevant constraints and is able to allow the system passing through a narrow window.
- 3) A practical test is successfully demonstrated in our lab environment. A swing behavior of the quadrotor with a cable-suspended load is observed.

In the following, we will give an introduction to the model identification in Section 3. Section 4 will address the trajectory optimization problem. In Section 5, we will demonstrate the aggressive manoeuvre by using an Astec Hummingbird quadrotor in an indoor environment. In Section 6, we will give a summary conclusion.

## 2 Preliminaries

### 2.1 3D Dynamic Model

The dynamic system, the quadrotor with a cable-suspended load, is presented in Fig. 1, including the inertial frame,



**Fig. 1** Single quadrotor (Q) carrying a cable-suspended payload (P), including ground frame  $x_g y_g z_g$ , body frame  $x_b y_b z_b$ , intermediate frame  $x y z$ , control input  $\{F_x, M_x, M_y, M_z\}$ , swing angles  $\alpha$  and  $\beta$ . Specifically, frame  $x y z$  is obtained after translating frame  $x_g y_g z_g$  to the center of quadrotor

intermediate frame, body-fixed frame, thrust and torques with respect to the body frame and swing angles of the cable with respect to the intermediate frame.

The dynamics of this system should be modeled as a hybrid one, which includes two subsystems [17]. The first one is referred to as the “quadrotor-with-load” subsystem, where the cable-connection is taut. The second one is the “quadrotor” subsystem, where the load has detached from the cable and undergoes a projectile manoeuvre. In order to simplify the problem, some reasonable assumptions are made as follows:

- 1) The quadrotor is considered as a geometric symmetric rigid body.
- 2) The payload is considered as a point mass attached on an inextensible massless cable and the cable is attached at the center of the quadrotor.
- 3) The mass of the payload is much smaller than quadrotor, which means its motion has little impacts on the quadrotor.
- 4) Only “quadrotor-with-load” subsystem is considered and the trajectory is optimized, while “quadrotor” subsystem is very straight forward and its details are neglected.
- 5) The locations of both the quadrotor and the payload are provided by an indoor localisation system.

### 2.2 2D Planar Model for Window Passing

In the task of passing the narrow window, the quadrotor and load are both considered only moving in a vertical plane. Thus the model in Fig. 1 could be further simplified from 3-dimensional Cartesian coordinate into a 2-dimensional coordinate, which is shown in Fig. 2.

The positions of quadrotor and payload in the planar plane are given as

$$\begin{aligned} \xi_Q &= y_Q e_y + z_Q e_z \\ \xi_P &= (y_Q - y_e) e_y + (z_Q - z_e) e_z \end{aligned} \tag{1}$$

where,  $\{e_y, e_z\}$  is the basis unit vector in  $y_g z_g$  plane,  $z_e = \sqrt{L^2 - y_e^2}$ .

Thus, the Lagrangian of the payload is

$$\begin{aligned} \mathcal{T} &= \frac{1}{2} m_P \dot{\xi}_P \cdot \dot{\xi}_P \\ \mathcal{U} &= m_P g \xi_P \cdot e_z \\ \mathcal{L} &= \mathcal{T} - \mathcal{U} \\ &= \frac{1}{2} m_P \dot{\xi}_P \cdot \dot{\xi}_P - m_P g \xi_P \cdot e_z \end{aligned} \tag{2}$$

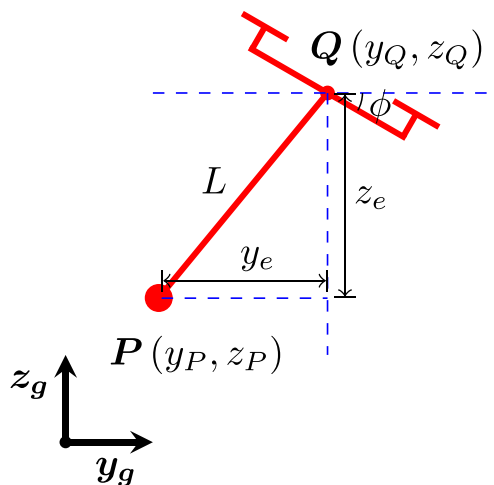
Then, the Euler-Lagrange equation is

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{y}_e} \right) - \frac{\partial \mathcal{L}}{\partial y_e} = 0 \tag{3}$$

where,  $y_e$  is chosen as the generalized coordinate and its generalized force is equal to 0.

Furthermore, we can get

$$\Rightarrow M \ddot{y}_e + f(y_e, \dot{y}_e, \ddot{\xi}_Q) = 0 \tag{4}$$



**Fig. 2** Simplified 2-dimensional ( $y_g z_g$  plane) model, including position of quadrotor ( $y_Q, z_Q$ ), position of payload with respect to quadrotor ( $y_e, z_e$ ), position of payload in  $y_g z_g$  frame ( $y_P, z_P$ ), roll angle of quadrotor  $\phi$ , where,  $y_e = y_P - y_Q, z_e = z_Q - z_P$

where,

$$M = \frac{L^2}{L^2 - y_e^2}$$

$$f(y_e, \dot{y}_e, \ddot{\xi}_Q) = \ddot{y}_Q + y_e \left( \frac{L^2 \dot{y}_e^2}{(L^2 - y_e^2)^2} + \frac{g + \ddot{z}_Q}{\sqrt{L^2 - y_e^2}} \right) \quad (5)$$

After taking Eq. 5 into Eq. 4, we get

$$-M\ddot{y}_e = \ddot{y}_Q + \frac{y_e}{\sqrt{L^2 - y_e^2}} \ddot{z}_Q + \left( \frac{y_e L^2 \dot{y}_e^2}{(L^2 - y_e^2)^2} + \frac{y_e g}{\sqrt{L^2 - y_e^2}} \right)$$

$$\Rightarrow \ddot{y}_e = f_y(y_e)\ddot{y}_Q + f_z(y_e)\ddot{z}_Q + f_n(y_e, \dot{y}_e) \quad (6)$$

As a result, the non-linear 2D model is given in Eq. 7, where  $[\ddot{y}_Q, \ddot{z}_Q]^T$  is the control input and  $[y_Q, dy_Q, z_Q, dz_Q, y_e, dy_e]^T$  is the control state. ( $dy_Q$  denotes the first derivation of  $y_Q$ , similarly hereafter)

$$\begin{aligned} \dot{y}_Q &= dy_Q \\ d\dot{y}_Q &= \ddot{y}_Q \\ \dot{z}_Q &= dz_Q \\ d\dot{z}_Q &= \ddot{z}_Q \\ \dot{y}_e &= dy_e \\ d\dot{y}_e &= f_y(y_e)\ddot{y}_Q + f_z(y_e)\ddot{z}_Q + f_n(y_e, \dot{y}_e) \end{aligned} \quad (7)$$

This non-linear dynamic model is built based on the first principle where some practical limitations, such as air resistance, geometric errors, etc, are not considered. This actually makes the model quite different from the real system. For example, if given a pulse signal of  $\ddot{y}_Q$  to this system, the simulated output of  $y_e$  will be a continuous oscillation without any damp, which is not real in fact due to the existence of the air drag. Also some vehicle parameters are actually difficult to obtain, such as aerodynamic coefficients of the propellers. This makes the system difficult to control. Finally, the desired trajectory in this task is generated by an optimization algorithm based on the dynamic model. Thus, the difference between the model and the practical system has a significant impact on the control performance.

In order to minimise the impact caused by the first principle model, we have to identify the dynamic model via experimental data. To do so, a so-called low order equivalent system (LOES), a linear transfer function, is adopted here to represent the dynamic model and is to be identified next.

As the desired altitude of the quadrotor is fixed, the effect of its acceleration  $\ddot{z}_Q$  is much smaller compared with  $\ddot{y}_Q$  and can be regarded as a disturbance. The model is further simplified as a single input and single output one. Linearizing the equation using Taylor series and then taking Laplace transform, we can get the low order equivalent

system as follows, where all the high-order terms are removed.

$$\frac{y_e(s)}{\ddot{y}_Q(s)} = \frac{K}{s^2 + 2\omega_n \zeta s + \omega_n^2} \quad (8)$$

where,  $K$ ,  $\omega_n$ , and  $\zeta$  are the unknown parameters to determine.  $y_e(s)$  and  $\ddot{y}_Q(s)$  are the input and output data respectively.

Thus, the non-linear dynamic model (Eq. 7) is simplified by following linearized form.

$$\frac{d}{dt} \begin{bmatrix} y_Q \\ \dot{y}_Q \\ y_e \\ \dot{y}_e \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_n^2 & -2\omega_n \zeta \end{bmatrix} \begin{bmatrix} y_Q \\ \dot{y}_Q \\ y_e \\ \dot{y}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ K \end{bmatrix} \ddot{y}_Q \quad (9)$$

where, the control input  $u$  is ( $\ddot{y}_Q$ ) and the control state  $x$  is  $(y_Q, \dot{y}_Q, y_e, \dot{y}_e)^T$ .

### 2.3 Model Identification

Once given the transfer function (Eq. 8) as the model to be identified, the system identification has now been transformed into the estimation of unknown parameters based on experimental data. A frequency based recursive least square technique is introduced as follows.

With Discrete Time Fourier Transform (DTFT) seen in Eq. 10, the discrete samplings  $\{y_e[i], \ddot{y}_Q[i]\}$  can be transformed into a continuous functions of  $j\omega$ , namely,  $\{\tilde{y}_e(\omega), \tilde{\ddot{y}}_Q(\omega)\}$ .

$$\tilde{x}(j\omega) = \sum_{i=0}^{N-1} x[i]e^{-j\omega i} \quad (10)$$

where  $N$  is the total number of sampled data.

As a result, the transfer function (Eq. 8) is transformed from Laplace domain  $s$  to frequency domain  $\omega$  seen in Eq. 11, where the variables are denoted with tilde.

$$(j\omega)^2 \tilde{y}_e(\omega) = \left[ (j\omega)\tilde{y}_e(\omega), \tilde{y}_e(\omega), \tilde{\ddot{y}}_Q(\omega) \right] \begin{bmatrix} -2\omega_n \zeta \\ -\omega_n^2 \\ K \end{bmatrix} \quad (11)$$

However, in order to implement a least square estimation, Eq. 11 should be discretized on a series of given frequencies, which starts from  $\omega_1$  (minimum) to  $\omega_m$  (maximum) with a fixed frequency resolution  $\Delta\omega$ . Usually,  $\omega_1$ ,  $\omega_m$  and  $\Delta\omega$  are determined based on the sampling information, e.g. sampling frequency, total sampling number and frequency of interest.

After discretization, the system can be arranged as a standard form for the least square estimation seen in Eq. 12, where  $\tilde{\mathbf{Y}}$  and  $\tilde{\mathbf{X}}$  are appropriate vector or matrix in terms of the discrete frequencies, namely,  $\{\omega_1, \omega_2, \dots, \omega_m\}$ .

$$\begin{aligned} \tilde{\mathbf{Y}} &= \tilde{\mathbf{X}}\theta + \varepsilon \\ \tilde{\mathbf{Y}} &= \begin{bmatrix} (j\omega_1)^2 \tilde{y}_e(\omega_1) \\ (j\omega_2)^2 \tilde{y}_e(\omega_2) \\ \vdots \\ (j\omega_m)^2 \tilde{y}_e(\omega_m) \end{bmatrix}, \theta = \begin{bmatrix} -2\omega_n \zeta \\ -\omega_n^2 \\ K \end{bmatrix} \\ \tilde{\mathbf{X}} &= \begin{bmatrix} (j\omega_1) \tilde{y}_e(\omega_1) & \tilde{y}_e(\omega_1) & \tilde{y}_Q(\omega_1) \\ (j\omega_2) \tilde{y}_e(\omega_2) & \tilde{y}_e(\omega_2) & \tilde{y}_Q(\omega_2) \\ \vdots & \vdots & \vdots \\ (j\omega_m) \tilde{y}_e(\omega_m) & \tilde{y}_e(\omega_m) & \tilde{y}_Q(\omega_m) \end{bmatrix} \end{aligned} \tag{12}$$

where,  $\theta$  and  $\varepsilon$  denote the unknown parameters and disturbances, respectively.

In practice, DTFT is not suitable for real time calculation, as it requires to save all the samples beforehand. Thus, a recursive DFT is utilized here, which can update the frequency spectrum recursively. Detailed derivation of Eq. 14 is available in the author’s previous publication [36].

The DFT and recursive DFT are given in Eqs. 13 and 14 respectively, where  $W_N = e^{-j\frac{2\pi}{N}}$  and  $0 \leq k \leq N - 1$ .

$$\tilde{x}[k] = \sum_{i=0}^{N-1} x[i] \cdot W_N^{ki} \tag{13}$$

$$\Rightarrow \begin{cases} \tilde{x}_k[n] = x[n] + W_N^{-k} \cdot \tilde{x}_k[n - 1] \\ \tilde{x}[k] = \tilde{x}_k[n]_{n=N} \end{cases} \tag{14}$$

Applying Eq. 14, the frequency spectrum of  $\tilde{y}_e[k]$  and  $\tilde{y}_Q[k]$  is updated and therefore  $\tilde{\mathbf{Y}}$  and  $\tilde{\mathbf{X}}$  can be restated in the following form.

$$\begin{aligned} \tilde{\mathbf{Y}} &= \begin{bmatrix} (j\Delta\omega)^2 \tilde{y}_e[1] \\ (2j\Delta\omega)^2 \tilde{y}_e[2] \\ \vdots \\ (mj\Delta\omega)^2 \tilde{y}_e[m] \end{bmatrix} \\ \tilde{\mathbf{X}} &= \begin{bmatrix} (j\Delta\omega) \tilde{y}[1] & \tilde{y}[1] & (j\Delta\omega) \tilde{u}[1] & \tilde{u}[1] \\ (2j\Delta\omega) \tilde{y}[2] & \tilde{y}[2] & (2j\Delta\omega) \tilde{u}[2] & \tilde{u}[2] \\ \vdots & \vdots & \vdots & \vdots \\ (mj\Delta\omega) \tilde{y}[m] & \tilde{y}[m] & (mj\Delta\omega) \tilde{u}[m] & \tilde{u}[m] \end{bmatrix} \end{aligned}$$

Moreover, once given the sampling frequency  $F_s(Hz)$ , the frequency resolution in terms of Hertz is defined as

$\Delta f = F_s/N$  and thus the corresponding angular frequency is defined as  $\Delta\omega = \Delta f (2\pi/F_s)$ .

Thus the frequency of interest  $[\omega_1, \omega_m]$  is determined as follows.

$$\begin{aligned} \omega_1 &= \Delta\omega \\ \omega_m &= m\Delta\omega \end{aligned} \tag{15}$$

where,  $m = \lfloor f_{max}/\Delta f \rfloor$  (rounding down the quotient) and  $f_{max}$  is a user-defined maximum value which is much smaller than  $F_s$ .

When the recursive updating of the spectrum of  $\tilde{\mathbf{Y}}$  and  $\tilde{\mathbf{X}}$  is done, a batch least square estimation is implemented to estimate  $\theta$ , as seen in Eq. 16.

$$J = \frac{1}{2} (\tilde{\mathbf{Y}} - \tilde{\mathbf{X}}\theta)^\dagger (\tilde{\mathbf{Y}} - \tilde{\mathbf{X}}\theta) \tag{16}$$

$$\Theta = \left[ \text{Re}(\tilde{\mathbf{X}}^\dagger \tilde{\mathbf{X}}) \right]^{-1} \text{Re}(\tilde{\mathbf{X}}^\dagger \tilde{\mathbf{Y}}) \tag{17}$$

where,  $\dagger$  denotes conjugate transpose.

In this way, the unknown parameters in Eq. 9 are determined. The main benefit of frequency based estimation is that the information on high frequency band could be removed which mainly is uncertainty and noise.

### 2.4 Controller Design

Before we start to generate the trajectory, it is necessary to introduce the tracking controller that will be utilized in our practical experiment. Briefly speaking, a cascade controller is adopted here. Firstly, a PD controller is running on the PC tracking the reference trajectory and calculating the desired thrust and attitudes. Meanwhile, there is another PD controller on the quadrotor receiving and tracking those out-loop commands. In this task, the system is assumed to move mainly in  $y_g$  direction, the reference trajectory is then considered to be  $\{y_Q^r, \dot{y}_Q^r, \ddot{y}_Q^r\}$ . Thus, the controller on PC can be summarized in following equations.

$$\ddot{y}_Q^{des} = \ddot{y}_Q^r + K_{p,y} (y_Q^r - y_Q) + K_{d,y} (\dot{y}_Q^r - \dot{y}_Q) \tag{18}$$

The desired trajectories in the other two directions are all set to be zero.

$$\begin{aligned} \ddot{x}_Q^{des} &= 0 + K_{p,x} (0 - x_Q) + K_{d,x} (0 - \dot{x}_Q) \\ \ddot{z}_Q^{des} &= 0 + K_{p,z} (0 - z_Q) + K_{d,z} (0 - \dot{z}_Q) \end{aligned} \tag{19}$$



The desired accelerations are transformed into the desired thrust and attitudes which will be sent to quadrotor afterwards.

$$\begin{aligned}
 F_z^{des} &= m(g + \ddot{z}_Q^{des}) \\
 \phi^{des} &= K_{p,\phi} \left( \ddot{x}_Q^{des} \sin \psi_r - \ddot{y}_Q^{des} \cos \psi_r \right) / g \\
 \theta^{des} &= K_{p,\theta} \left( \ddot{x}_Q^{des} \cos \psi_r + \ddot{y}_Q^{des} \sin \psi_r \right) / g \\
 \psi^{des} &= \psi_r
 \end{aligned} \tag{20}$$

where,  $\{F_z^{des}, \phi^{des}, \theta^{des}, \psi^{des}\}$  are the control inputs of inner-loop controller. The parameters  $K_{p,\phi}$  and  $K_{p,\theta}$  are scale factors. They are all determined by the API functions of Astec Hummingbird quadrotor [37].

In the next section, we will show how to generate the reference trajectory  $\{y_Q^r, \dot{y}_Q^r, \ddot{y}_Q^r\}$ .

### 3 Aggressive Trajectory Generation

#### 3.1 Trajectory Optimization Problem

A trajectory optimization problem is to find a feasible trajectory  $\{t_0, t_F, u^*(t), x^*(t)\}$  for a dynamic system that satisfies a set of constraints while minimizing a cost function. A general framework for a trajectory optimization problem is illustrated in Eq. 21 [38].

$$\begin{aligned}
 \min_{t_0, t_F, x(t), u(t)} \quad & J = J_B(t_0, t_F, x(t_0), x(t_F)) \\
 & + \int_{t_0}^{t_F} J_P(\tau, x(\tau), u(\tau)) d\tau \\
 \text{s.t.} \quad & \dot{x}(t) = f(t, x(t), u(t)) \\
 & C_P(t, x(t), u(t)) \leq 0 \\
 & C_B(t_0, t_F, x(t_0), x(t_F)) \leq 0 \\
 & x^- \leq x(t) \leq x^+ \\
 & u^- \leq u(t) \leq u^+ \\
 & t_0^- \leq t_0 \leq t_0^+ \\
 & t_0 < t_F \\
 & t_F^- \leq t_F \leq t_F^+ \\
 & x_0^- \leq x(t_0) \leq x_0^+ \\
 & x_F^- \leq x(t_F) \leq x_F^+
 \end{aligned} \tag{21}$$

The minimum target,  $J$ , is the user-defined objective function in terms of boundary ( $J_B$ ) and whole trajectory ( $J_P$ ).

The group of functions  $\{f, C_P, C_B\}$  denotes the dynamic model, the trajectory constraint function and the boundary constraint function, respectively. All the functions  $\{J_B, J_P, f, C_P, C_B\}$  are assumed to be smooth from  $t_0$  to  $t_F$ .

Apart from that, the constant low bounds for the trajectory are given in  $\{x^-, u^-, t_0^-, t_F^-, x_0^-, u_0^-, x_F^-, u_F^-\}$  and similarly  $\{x^+, u^+, t_0^+, t_F^+, x_0^+, u_0^+, x_F^+, u_F^+\}$  are the upper bounds. Specifically, if a low bound  $-$  is equal to its upper bound  $+$ , the inequality constraint will then become an equality.

#### 3.2 Cost Functions and Boundary Constraints

Like the minimum snap trajectory [39], we would like to set  $J_B$  to be 0 and only introduce the minimum acceleration as the objective function seen in Eq. 22.

$$\begin{aligned}
 J_B(t_0, t_F, x(t_0), x(t_F)) &= 0 \\
 J_P(\tau, x(\tau), u(\tau)) &= \ddot{y}_Q^2(\tau)
 \end{aligned} \tag{22}$$

As for the constraints, the dynamic model is determined by the identified result in Eq. 9. In this task, there are no complicated constraint functions in terms of  $C_P(t, x(t), u(t))$  or  $C_B(t_0, t_F, x(t_0), x(t_F))$ .

Theoretically, the initial position ( $\{x_0^-, x_0^+\}$ ) for the system could be chosen at any appropriate places. Without loss of generality, we will choose three different places as the starting points and analyze their effects. As for the constraints for terminal states ( $\{x_F^-, x_F^+\}$ ), however, they have to be carefully chosen. Specifically, the quadrotor will swing up the load to the highest point with some velocity. In this way, the quadrotor and load can both fly through the window. The exact values are presented in Section slowromancapiv@-B.

Meanwhile, the whole trajectory should be limited in an appropriate range, namely, the boundary constraints  $\{x^-, u^-, x^+, u^+\}$ .

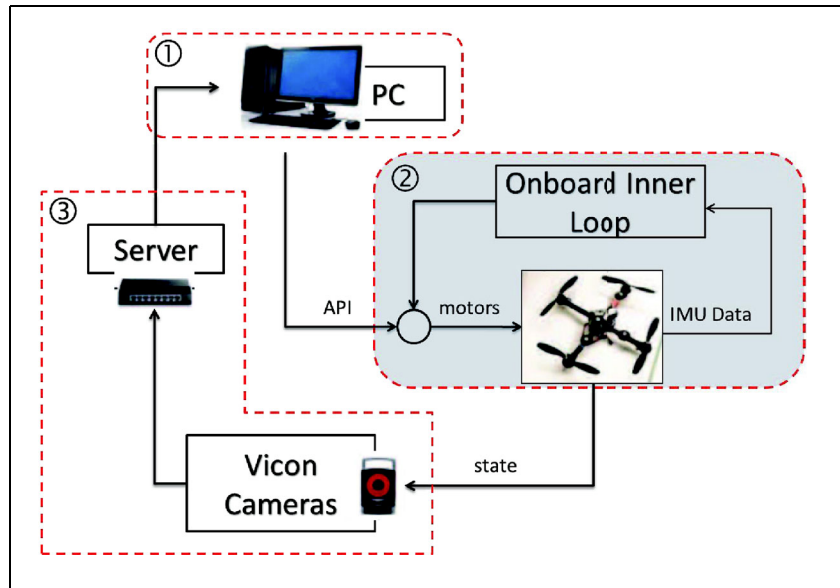
#### 3.3 Choosing $t_F$

Without loss of generality, the starting time of the desired manoeuvre could be set as 0, which means  $t_0^- = t_0^+ = 0$ . Now we need to determine the last two parameters  $t_F^-$  and  $t_F^+$  before we start to solve the trajectory optimization problem.

Generally, if  $t_F$  is too short the desired trajectory would become too aggressive or even infeasible because the quadrotor might not be able to generate such a big acceleration. If  $t_F$  is too long, the generated trajectory will experience a long beginning period to adjust the load status by moving it back and forth for too many times.

Actually,  $t_F$  can be set as a free variable and determined by the algorithm automatically. However, this will result in a more complicated cost function and accordingly lead to a greater computational burden on the computer. So as far as current work is concerned, we would like to choose the  $t_F^-$  and  $t_F^+$  by the trial and error method.

**Fig. 3** Diagram of the hardware platform, including a desktop computer running with the control algorithm, quadrotor-payload system with onboard inner loop stabilization system, motion capture system (VICON)



### 3.4 Solving the Optimization Problem

Generally, there are three types of algorithms for solving optimal control problems: dynamic programming, indirect methods and direct methods. In this paper, we utilize a so-called transcription method to solve this problem by converting the continuous problem into a non-linear programming problem. Specifically, an orthogonal collocation method is utilized here by using orthogonal polynomials to approximate the state and control functions. Once in this form, the problem can be passed to a commercial solver, such as SNOPT, IPOPT, or FMINCON. Because of space constraints, more details of calculation can be found in [38, 40].

## 4 Practical Experiment

In order to validate the proposed approach, a practical experiment is presented here.

The hardware platform consists of three main parts seen in Fig. 3. The first part is the desktop computer with a real-time algorithm running on it. It receives the position information from the Vicon Server, calculates the desired thrust and attitude and finally sends them to the quadrotor every 0.02s via the Zigbee module. The second part is the quadrotor (Astec HummingBird) as well as its cable-suspended load and both of them are labeled with appropriate reflective markers. The last part is the motion capture system (VICON) which updates the information of the attitude and position of the quadrotor and the load at the frequency of 100Hz. The fast and high-precision 3D position data makes it possible for the HummingBird to track an aggressive trajectory.

The optimization problem is solved offline using Matlab on 2.0GHz Intel Core i7 laptop and the runtime for solving the optimization problem is around 100s. In real time, as a low order equivalent system is utilized instead of a complicated nonlinear one, the computation load for the real time part is acceptable.

### 4.1 System Identification

The model identification is a preparation step for the experiment. In this stage we choose some specific references  $\{y_Q^r, \dot{y}_Q^r, \ddot{y}_Q^r\}$  to excite the system. A so-called chirp signal is designed as the position reference  $y_Q^r$  for quadrotor which lasts for  $t_{max}$  long and its frequency linearly increases from 0 up to  $f_{max}$  (Hz) seen in Eq. 23. In this way, the system could be sufficiently excited in the frequency band, namely,  $[0, f_{max}]$ .

$$y_Q^r = A \sin \left( \int_0^t \lambda \tau d\tau \right) = A \sin \left( \frac{1}{2} \lambda t^2 \right)$$

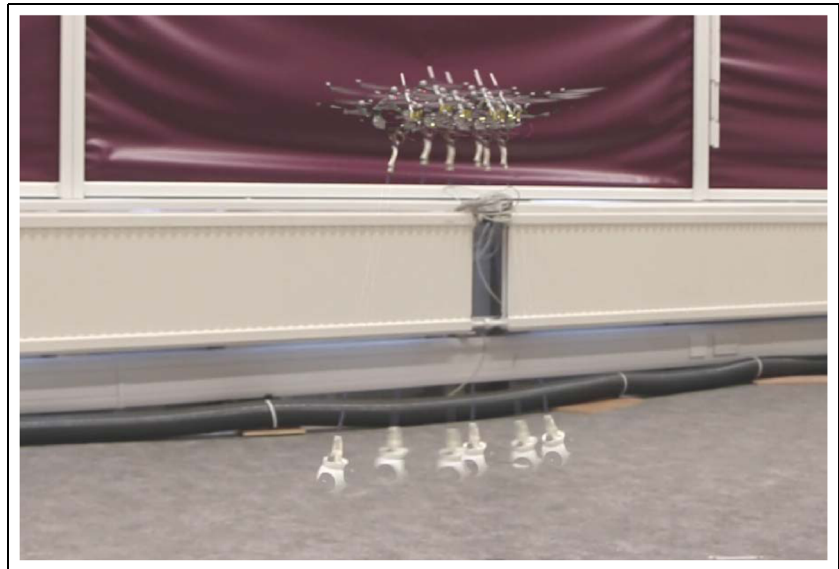
$$\lambda = \frac{2\pi f_{max}}{t_{max}} \tag{23}$$

where,  $0 \leq t \leq t_{max}$ ,  $A = -0.2m$ ,  $f_{max} = 0.7Hz$ ,  $t_{max} = 20s$ ,  $\lambda = 0.2199(rad/s^2)$ .

Based on the PD controller in Eq. 18, the quadrotor is controlled to follow  $\{y_Q^r, \dot{y}_Q^r, \ddot{y}_Q^r\}$  and it will fly back and forth in  $y_g$  direction with a faster and faster velocity as well as a bigger and bigger swing angle. The time lapse snapshot is shown in Fig. 4 and the relevant results are plotted in Fig. 5.

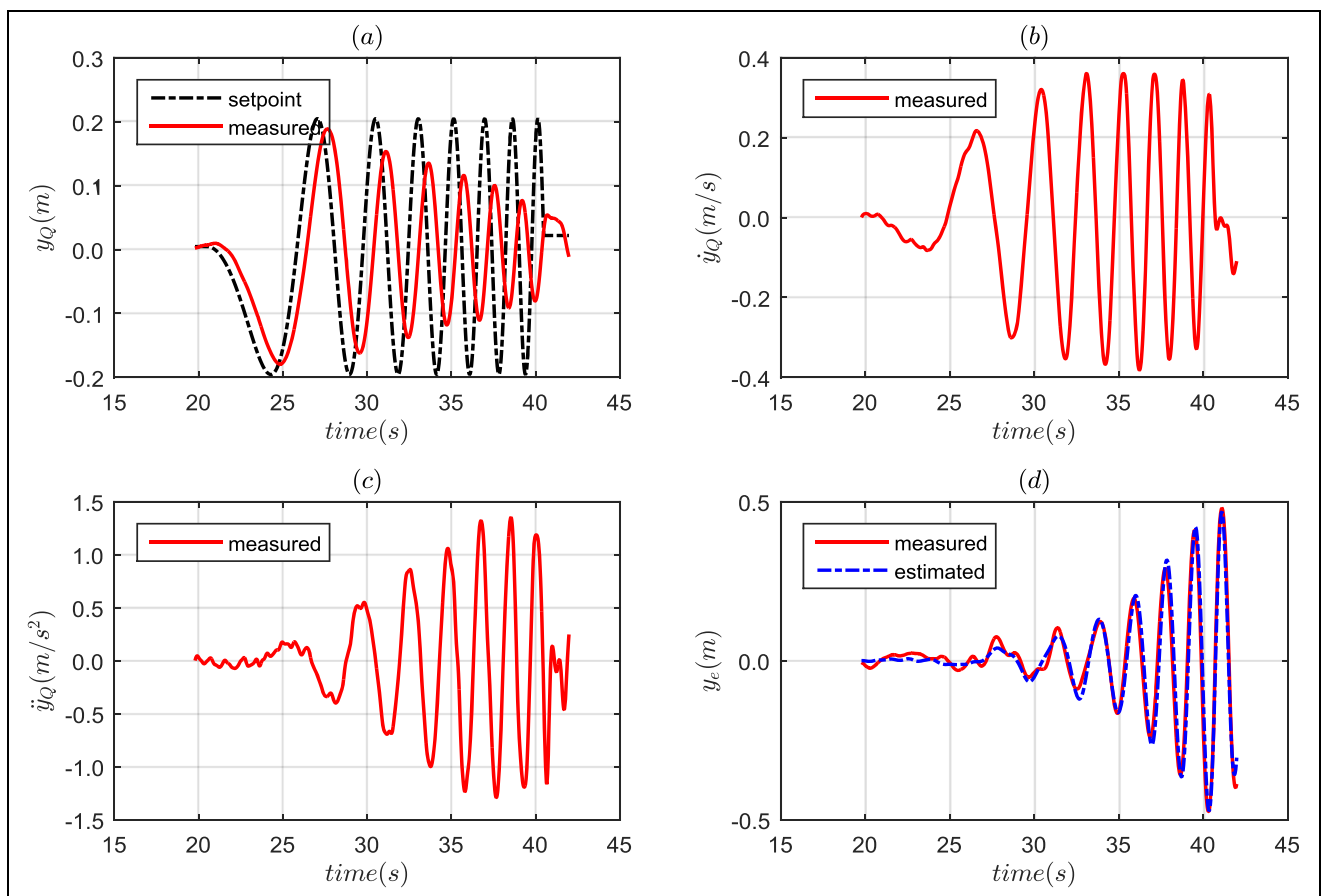
As seen in Fig. 5a, the measured position (red solid line) of the quadrotor has an obvious decline in the magnitude while the desired trajectory (black dashed line) becomes more aggressive. It matches well with the frequency

**Fig. 4** Time-lapse snapshot of the manoeuvre for model identification



characteristic of a practical system that the magnitude gain decreases while the frequency increases. In (b) and (c),

the measurements of velocity and acceleration of quadrotor are obtained by calculating the first and second derivation

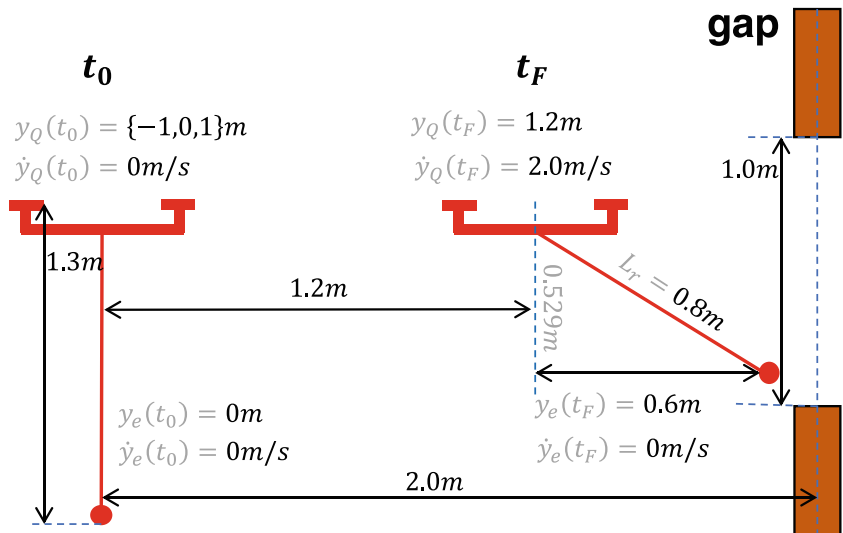


**Fig. 5** Practical test for model identification. **a** black dashed line: a user-defined trajectory for quadrotor to track which is a chirp signal, red line: practical position of quadrotor  $y_Q$  measured by VICON. **b** red solid line: velocity of quadrotor  $\dot{y}_Q$  obtained by calculating the first

derivative of  $y_Q$ . **c** red solid line: acceleration of quadrotor  $\ddot{y}_Q$  obtained by calculating the second derivative of  $y_Q$ . **d** red solid line:  $y_e$ , position of payload with respect to quadrotor. blue dashed line: estimated  $y_e$  based on the identified low order equivalent system

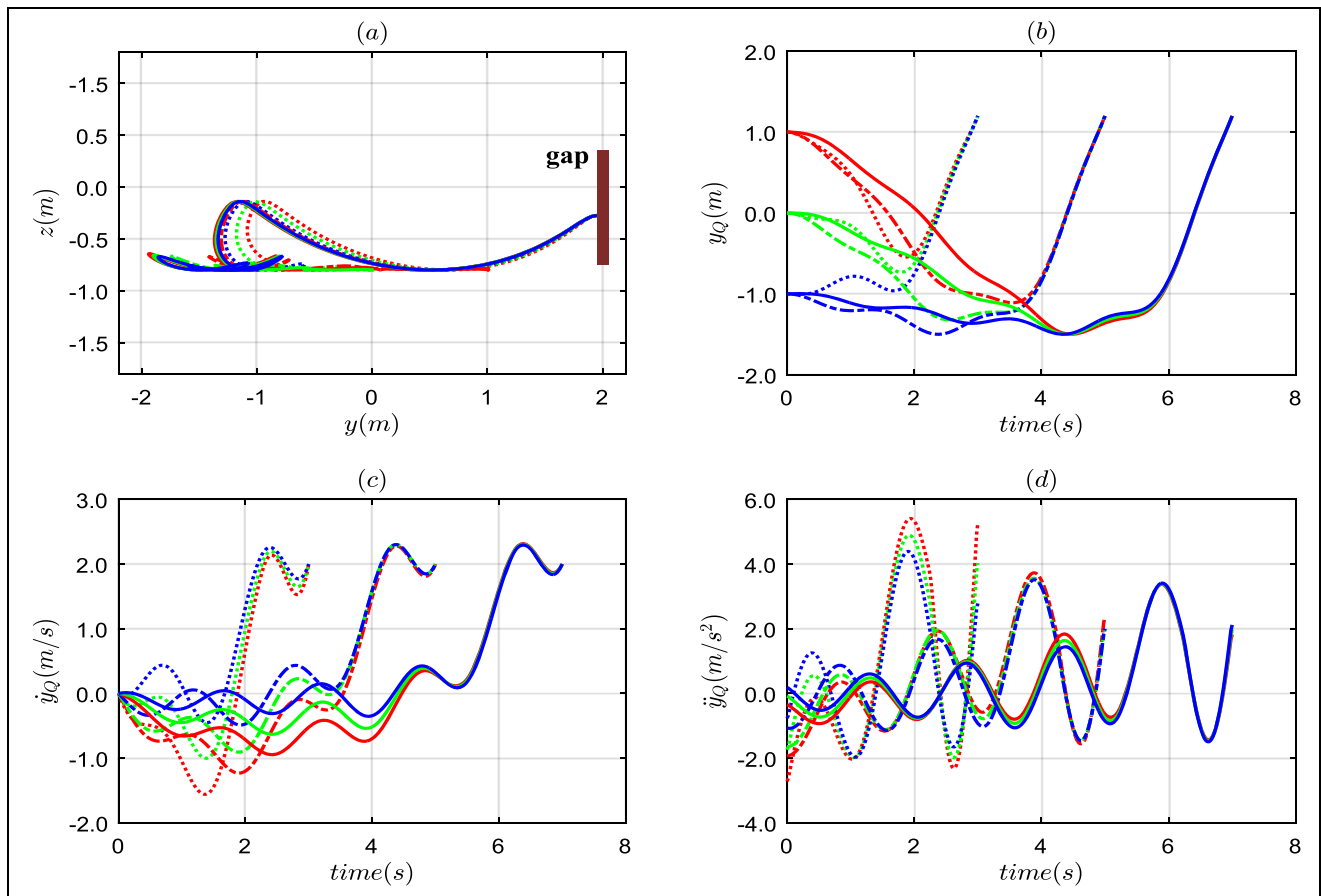


**Fig. 6** Boundary constraints at  $t_0$  and  $t_F$  for trajectory tracking phase



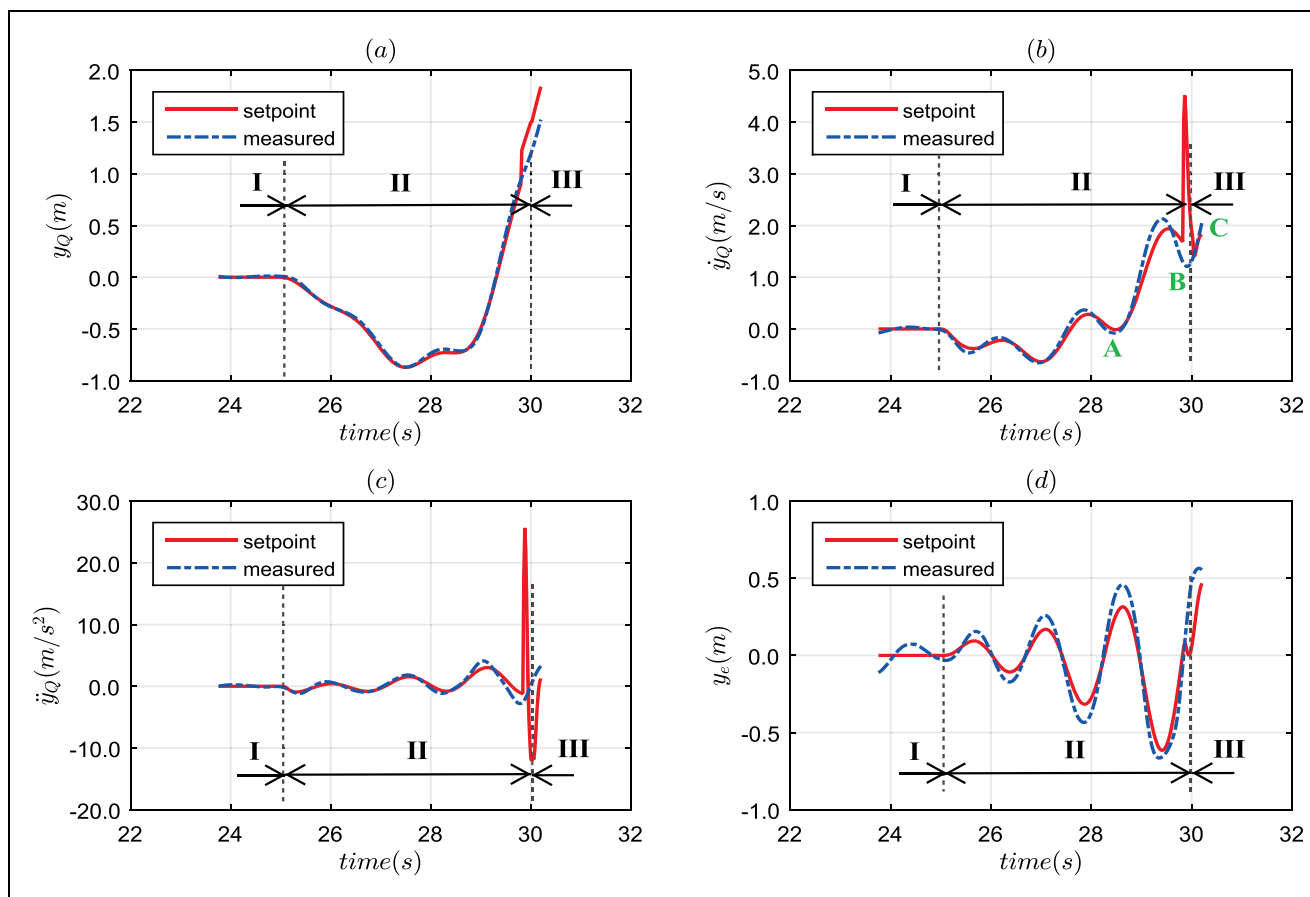
of  $y_Q$ , respectively. In (d), the measurement of  $y_e$  is the difference between the positions of quadrotor and load.

Then, choosing  $\ddot{y}_Q$  and  $y_e$  as the input and output sequences respectively, the frequency based recursive least

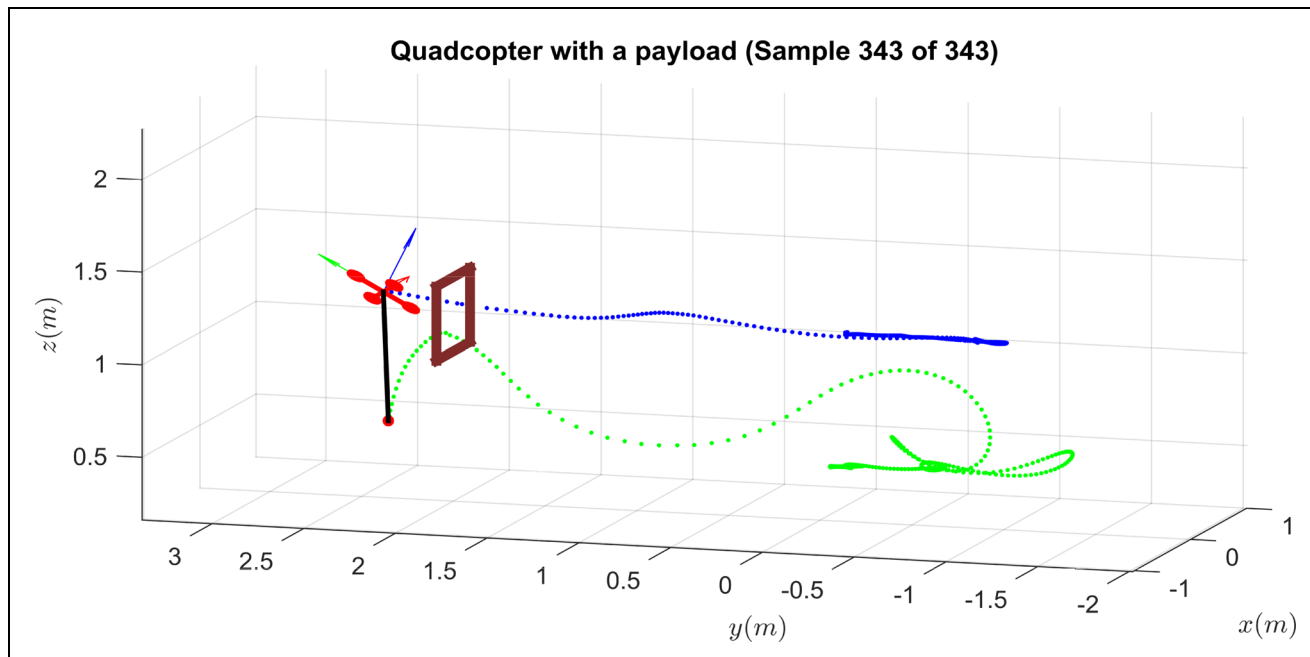


**Fig. 7** Numerical simulated trajectories under different conditions, namely, different starting points: 1.0m(red), 0.0m(green), -1.0m(blue), different  $t_F$ : 3s(dotted), 5s(dashed), 7s(solid). **a** trajectory of load

in  $y_g z_g$  plane. **b** position of quadrotor ( $y_Q$  vs. time). **c** velocity of quadrotor ( $\dot{y}_Q$  vs. time). **d** acceleration of quadrotor ( $\ddot{y}_Q$  vs. time)



**Fig. 8** Desired (red solid) and practical (blue dashed) trajectory. **a** position of quadrotor  $y_Q$ . **b** velocity of quadrotor  $\dot{y}_Q$ . **c** acceleration of quadrotor  $\ddot{y}_Q$ . **d** position of payload with respect to quadrotor  $y_e$



**Fig. 9** 3D animation of the flight based on VICON data. blue dotted line: trajectory of the quadrotor. green dotted line: trajectory of the load. The animation video is available on <https://www.youtube.com/watch?v=0xjwftBTvWM>

**Fig. 10** Time-lapse snapshots of the flying manoeuvre. The video is available on <https://www.youtube.com/watch?v=.OLmyNKiCg4>



square estimation technique is applied and the identified parameters are acquired in Eq. 24.

$$\begin{aligned}
 K &= -1.3477 \\
 \omega &= 4.1268, \zeta = 0.0634
 \end{aligned}
 \tag{24}$$

Thus, the state space equation in Eq. 9 is now available. In order to check its effectiveness, the estimation of  $y_e$  based on the identified model is also plotted in Fig. 5d (blue dashed line) and it is very close to the measured one.

### 4.2 Trajectory Generation

As seen in Fig. 6, the window locates in  $y = 2.0m$  and the height of the gap is  $1.0m$ . The total height of “quadrotor and load” system is larger than that, which means it cannot pass the window without swinging. In order to solve this problem, the “passing window” strategy consists of three phases. In the first phase, the quadrotor takes off with the cable-suspended load, then moves to a starting position and

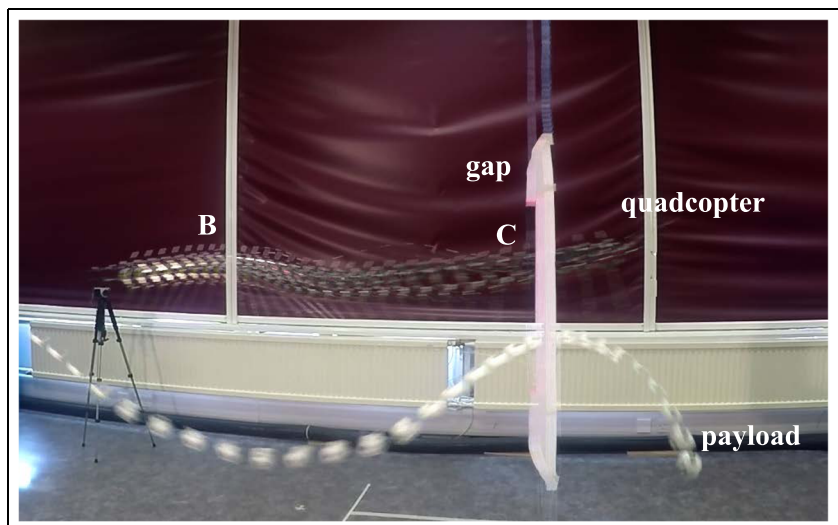
maintains hovering. The second phase is also illustrated in Fig. 6 where the quadrotor starts to track a pre-defined trajectory from  $t_0$  to  $t_F$  so that the load will be swung up to an appropriate height and be able to pass the window. In the last phase, the quadrotor switches to tracking the load and then the whole system will fly through the gap. Among all three phases, generating a feasible trajectory for the second phase is the most important and challenging problem. With the identified model Eqs. 9, 24, the trajectory generation problem is solved under the general framework of Eq. 21.

At initial moment in Fig. 6, the system maintains hovering, so  $\dot{y}_Q(t_0)$ ,  $y_e(t_0)$  and  $\dot{y}_e(t_0)$  are all equal to 0. In order to analyse the impact of flight distance,  $y_Q(t_0)$  is given three different values ( $-1m, 0m, 1m$ ). As a result, the constraints at  $t_0$  are as follows.

$$x_0^- = x_0^+ = (\{-1m, 0m, 1m\}, 0m/s, 0m, 0m/s)
 \tag{25}$$

The constraint of the states at  $t_F$  is set to be the same for trajectories with different starting positions. Specifically,

**Fig. 11** Time-lapse snapshots of the trajectory through the window



$y_Q(t_F)$  and  $y_e(t_F)$  are given to make sure that the quadrotor can swing the load up to an appropriate height, which is also the highest position where  $\dot{y}_e(t_F)$  is equal to  $0m/s$ . At that moment, there will be enough space for quadrotor and load to pass through the gap and the velocity  $\dot{y}_Q(t_F)$  is set as  $0.6m/s$ .

$$x_F^- = x_F^+ = (1.2m, 2.0m/s, 0.6m, 0m/s) \quad (26)$$

For the sake of safety, the trajectory should respect the following boundary constraints in terms of distance and velocity from  $t_0$  to  $t_F$ .

$$\begin{aligned} x^- &= (-2.0m, -4.0m/s, -0.7m, -4m/s) \\ x^+ &= (2.0m, 4.0m/s, 0.7m, 4m/s) \end{aligned} \quad (27)$$

Considering the trajectory's feasibility, the limitation of the control inputs are considered as follows.

$$u^- = -6m/s^2, \quad u^+ = 6m/s^2 \quad (28)$$

Without loss of generality, starting time  $t_0$  is equal to  $0s$ . In order to analyze the impact of different flying time, the terminal time  $t_F$  is given as three values.

$$t_F^- = t_F^+ = \{3s, 5s, 7s\} \quad (29)$$

Based on different sets of boundary constraints from Eqs. 25–29, several trajectories are generated by the optimization algorithm and are illustrated in Fig. 7.

As seen in Fig. 7a and b, all the trajectories start from moving left to nearly the same position and then moving right again, which is viewed as the initial adjustment. After that, all the trajectories nearly have the same segment where the quadrotor suddenly moves right and starts to swing up the load, which is determined by the constraints in Eq. 26. Meanwhile, in Fig. 7c and d, it can be seen that when the simulation time is shorter the generated acceleration will be bigger which means the trajectory will become more aggressive. On the contrary, when the simulation time is longer it will generate a trajectory with smaller acceleration but more oscillations.

### 4.3 Tracking Result

The complete flying process consists of three stages (Fig. 8). The first stage is to position the system. The second stage is the tracking stage where the generated trajectory is tracked. The third stage is the final one where the quadrotor and the payload is separated and the tension of the cable becomes zero. The detailed behavior is described as follows:

In stage *slowromancapi@* (0s–25s), the quadrotor takes off with the load, moves to the initial position, keeps hovering and waits for the command to step into the next stage.

In stage *slowromancapii@* (25s–30s), the quadrotor begins to track the generated trajectory  $\{\ddot{y}_Q^r, \dot{y}_Q^r, y_Q^r\}$ .

Firstly, it starts to move back and forth in order to gradually increase the swing angle of the load. At the same time, it adjusts its distance to the window so that it is feasible to swing up for passing. After that, the quadrotor starts the “final shot”. It speeds up from position A in order to increase the kinematic energy of the load and then decreases the speed until position B waiting for the load to swing up to an ideal height and pass the center of the window, position C. In this stage the tension of the cable keeps non-zero. At the end of stage *slowromancapii@*, the load is in position C and the quadrotor is in position B. In order to keep passing through the window, the load's position in  $y$  direction is automatically given as the set-point for the quadrotor, which is stage *slowromancapiii@* (after 30s). This is the reason why there are some specific pulse signals at the very first beginning.

In stage *slowromancapiii@* (after 30s), the quadrotor starts to immediately speed up from position B and fly through position C just following the load. Meanwhile, the load undergoes a projectile motion from position C as the cable tension becomes zero. The system then goes back to hovering situation when the quadrotor and load pass through the gap. The result shows the quadrotor is able to carry the load to track the desired trajectory.

However, considering the time delay from wireless communication, there are some discrepancies between the practical test and simulation results. In Fig. 8a, the set-point for stage III is set for several sampling periods earlier which begins at around 29.6s. Meanwhile, the measurements of stage II might end later than 30s, where  $y_e$  at this moment is around 0.6m and  $\dot{y}_Q$  is around 2m/s.

The trajectory that calculated under the condition (0m, 5s) is then chosen as the set-points  $(y_Q^r, \dot{y}_Q^r, \ddot{y}_Q^r)$  for the quadrotor to track in real experiment. By using the PD controller in Eq. 18, the quadrotor is able to track the generated optimal trajectory. The trajectories of both quadrotor and load are measured with the VICON system and thus a 3D animation is shown in Fig. 9. Additionally two time-lapse snapshots of the tracking result are shown in Figs. 10 and 11. The comparison between the desired trajectory and the measured one is shown in in Fig. 8.

## 5 Conclusions

In this paper, we have presented a solution to the problem - a quadrotor to carry a payload flying through a narrow window with the assumption that its complicated dynamic model derived from the First Principle is unavailable. Our solution includes three parts, namely the system identification, the trajectory generation and the trajectory tracking controller. The system is assumed to conduct the window passing behaviour in the vertical plane and

then a 2D planar model of this system is developed. To take the practical environmental factors into consideration, the system model is identified through experimental data. The identified model is then used for the trajectory generation and the tracking controller. The identified system is simplified as a low order equivalent system (LOES) with a single input and single output. And a least square estimation in frequency domain is employed to estimate the system parameters. The trajectory generation is turned into an optimization problem with appropriate constraints. The generated trajectory is aggressive, which means the quadrotor has to swing the load to acquire sufficient kinematic energy to pass the window. Finally, a practical demonstration is implemented which validates the proposed approach. In the future, we would like to choose a more complicated model for system identification to generate the corresponding trajectory. The corresponding trajectory optimization problem would be imposed with more constraints.

**Acknowledgments** The authors would like to thank Yaser Alothman, Robin Dowling and Ian Dukas at the University of Essex for their technical support and M. P. Kelly from Cornell University for sharing his Matlab library of OptimTraj.

## References

- Mathew, N., Smith, S.L., Waslander, S.L.: Planning paths for package delivery in heterogeneous multirobot teams. *IEEE Trans. Autom. Sci. Eng.* **12**(4), 1298–1308 (2015)
- Zhan, P., Yu, K., Swindlehurst, A.L.: Wireless relay communications with unmanned aerial vehicles: Performance and optimization. *IEEE Trans. Aerosp. Electron. Syst.* **47**(3), 2068–2085 (2011)
- DeBusk, W.M.: Unmanned aerial vehicle systems for disaster relief: Tornado alley. In: *Infotech@ Aerospace Conferences* (2010)
- Zhang, C., Hu, H., Gu, D., Wang, J.: Cascaded control for balancing an inverted pendulum on a flying quadrotor. *Robotica* **35**(6), 1263–1279 (2017)
- Lindsey, Q., Mellinger, D., Kumar, V.: Construction of Cubic Structures with Quadrotor Teams. In: *Proc. Robotics: Science & Systems VII* (2011)
- Thomas, J., Polin, J., Sreenath, K., Kumar, V.: Avian-inspired grasping for quadrotor micro UAVs. In: *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, pp. V06AT07A014–V06AT07A014 (2013)
- Ritz, R., D’Andrea, R.: Carrying a flexible payload with multiple flying vehicles. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3465–3471. IEEE (2013)
- Tang, S., Sreenath, K., Kumar, V.: Aggressive maneuvering of a quadrotor with a cable-suspended payload. In: *Robotics: Science and Systems, Workshop on Women in Robotics*. Citeseer (2014)
- Cruz, P., Fierro, R.: Autonomous lift of a cable-suspended load by an unmanned aerial robot. In: *2014 IEEE Conference on Control Applications (CCA)*. pp. 802–807. IEEE (2014)
- Alothman, Y., Jasim, W., Gu, D.: Quad-rotor lifting-transporting cable-suspended payloads control. In: *2015 21st International Conference on Automation and Computing (ICAC)*, pp. 1–6. IEEE (2015)
- Dai, S., Lee, T., Bernstein, D.S.: Adaptive control of a quadrotor UAV transporting a cable-suspended load with unknown mass. In: *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, pp. 6149–6154. IEEE (2014)
- Faust, A., Palunko, I., Cruz, P., Fierro, R., Tapia, L.: Learning swing-free trajectories for UAVs with a suspended load. In: *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4902–4909. IEEE (2013)
- Palunko, I., Fierro, R., Cruz, P.: Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. In: *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2691–2697. IEEE (2012)
- Sreenath, K., Lee, T., Kumar, V.: Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load. In: *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, pp. 2269–2274. IEEE (2013)
- Sreenath, K., Kumar, V.: Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. *rm 1(r2), r3* (2013)
- De Crousaz, C., Farshidian, F., Buchli, J.: Aggressive optimal control for agile flight with a slung load. In: *IROS 2014 Workshop on Machine Learning in Planning and Control of Robot Motion*. Citeseer (2014)
- Tang, S., Kumar, V.: Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2216–2222. IEEE (2015)
- Klein, V., Morelli, E.A.: *Aircraft system identification: theory and practice*. American Institute of Aeronautics and Astronautics Reston, Va USA (2006)
- Tischler, M.B., Remple, R.K.: *Aircraft and rotorcraft system identification*. AIAA education series (2006)
- Hoffer, N.V., Coopmans, C., Jensen, A.M., Chen, Y.: A survey and categorization of small low-cost unmanned aerial vehicle system identification. *J. Intell. Robot. Syst.* **74**(1-2), 129 (2014)
- Park, S.: Modeling with vortex lattice method and frequency sweep flight test for a fixed-wing UAV. *Control. Eng. Pract.* **21**(12), 1767–1775 (2013)
- Dorobantu, A., Murch, A., Mettler, B., Balas, G.: System identification for small, low-cost, fixed-wing unmanned aircraft. *J. Aircr.* **50**(4), 1117–1130 (2013)
- Ghosh, A., Raisinghani, S.: Frequency-domain estimation of parameters from flight data using neural networks. *J. Guid. Control. Dyn.* **24**(3), 525–530 (2001)
- Shamsudin, S.S., Chen, X.: Identification of an unmanned helicopter system using optimised neural network structure. *Int. J. Model. Identif. Control.* **17**(3), 223–241 (2012)
- Puttige, V.R., Anavatti, S.G.: Real-time system identification of unmanned aerial vehicles: a multi-network approach. *JCP* **3**(7), 31–38 (2008)
- Chen, S., Billings, S.: Neural networks for nonlinear dynamic system modelling and identification. *Int. J. Control.* **56**(2), 319–346 (1992)
- Theodore, C.R., Tischler, M.B., Colbourne, J.D.: Rapid frequency-domain modeling methods for unmanned aerial vehicle flight control applications. *J. Aircr.* **41**(4), 735–743 (2004)
- Gremillion, G., Humbert, J.: System identification of a quadrotor micro air vehicle. In: *AIAA Atmospheric Flight Mechanics Conference*, pp. 7644 (2010)



29. Holzel, M.S., Morelli, E.A.: Real-time frequency response estimation from flight data. *J. Guid. Control. Dyn.* **35**(5), 1406–1417 (2012)
30. Grauer, J., Morelli, E.: Method for real-time frequency response and uncertainty estimation. *J. Guid. Control. Dyn.* **37**(1), 336–344 (2013)
31. Morelli, E.A., Smith, M.S.: Real-time dynamic modeling: data information requirements and flight-test results. *J. Aircr.* **46**(6), 1894–1905 (2009)
32. Morelli, E.A.: Real-time parameter estimation in the frequency domain. *J. Guid. Control. Dyn.* **23**(5), 812–818 (2000)
33. Hehn, M., D’Andrea, R.: Quadcopter trajectory generation and control. *IFAC Proceedings Volumes* **44**(1), 1485–1491 (2011)
34. Hehn, M., Ritz, R., D’Andrea, R.: Performance benchmarking of quadrotor systems using time-optimal control. *Auton. Robot.* **33**(1-2), 69–88 (2012)
35. Tomić, T., Maier, M., Haddadin, S.: Learning quadrotor maneuvers from optimal control and generalizing in real-time. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp 1747–1754, IEEE (2014)
36. Guo, M., Su, Y., Gu, D.: System identification of the quadrotor with inner loop stabilisation system. *Int. J. Model. Identif. Control.* **28**(3), 245–255 (2017)
37. Ascending technologies. [Online]. Available: <http://www.ascotec.de>
38. Kelly, M.P.: Transcription methods for trajectory optimization (2015)
39. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 2520–2525, IEEE (2011)
40. Kelly, M.P.: An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, pp. 2520–2525 (2016)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Minhuan Guo** received the B.E. degree, M.E. and Ph.D. in mechanical engineering from Nanjing University of Science and Technology, Nanjing, China, in 2012, 2014 and 2018 respectively. In 2016, he was a visiting Ph.D student at the School of Computer Science and Electronic Engineering, University of Essex, U.K. In 2019, he joined the Suzhou Vocational University and his research interests include robotics and automation technology.

**Dongbing Gu** received the B.Sc. and M.Sc. degrees in control engineering from the Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in robotics from the University of Essex, Colchester, U.K.

He was an Academic Visiting Scholar at the Department of Engineering Science, University of Oxford, Oxford, U.K., from 1996 to 1997. In 2000, he joined the University of Essex as a Lecturer, where he is a Professor with the School of Computer Science and Electronic Engineering. His current research interests include robotics, multi-agent systems, cooperative control, model predictive control, visual SLAM, wireless sensor networks, and machine learning.

**Wenzhong Zha** received the B.Sc. degree in mathematics and Ph.D. degree in automation from Beijing Institute of Technology, Beijing, China in 2008 and 2016. Then he joined the Information Science Academy of China Electronics Technology Group Corporation. His current research interests include differential games, dynamic games, multiagent systems, multiobjective optimization and decision, and incomplete information processing.

**Xinhua Zhu** received the B.Sc. and M.Sc. degrees in control engineering from the Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in Instrument Science and Technology from Southeast University, Nanjing, China. He served in Southeast University as an associate professor from 1996 to 2005. Since 2005, he has been a professor in School of Mechanical Engineering, Nanjing University of Science and Technology. His research interests include MEMS sensors, inertial navigation systems, and infrared sensors.

**Yan Su** received the B.E., M.E. and Ph.D. degrees in Instrumentation Science and Technology from Southeast University in 1989, 1996 and 2001, respectively. During 2001 and 2005, he served in Southeast University as an associate professor. Since 2005, he has been a professor in School of Mechanical Engineering, Nanjing University of Science and Technology. His research interests include MEMS sensors, biosensors, inertial navigation systems, and infrared sensors.

## Affiliations

Minhuan Guo<sup>1</sup>  · Dongbing Gu<sup>2</sup> · Wenzhong Zha<sup>3</sup> · Xinhua Zhu<sup>4</sup> · Yan Su<sup>4</sup>

Dongbing Gu  
dgu@essex.ac.uk

Wenzhong Zha  
zhawenzhong@126.com

Xinhua Zhu  
zhuxinhua@njust.edu.cn

Yan Su  
suyan@njust.edu.cn

- <sup>1</sup> School of Electronics Information Engineering, Suzhou Vocational University, Suzhou, 215104, China
- <sup>2</sup> School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, UK
- <sup>3</sup> Information Science Academy of China Electronics Technology Group Corporation, Chaoyang, China
- <sup>4</sup> School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China