Invited Feature Article

INTERNATIONAL JOURNAL OF
ADVANCED ROBOTIC SYSTEMS

*International Journal of Advanced
Robotic Systems*
September-October 2016: 1–10
© The Author(s) 2016
DOI: 10.1177/1729881416670109
arx.sagepub.com
SAGE

Large-scale stabilization control of input-constrained quadrotor

Jun Jiang, Liying Yang, Feng Gu, Yuqing He, and Jianda Han

Abstract

The quadrotor has been the most popular aircraft in the last decade due to its excellent dynamics and continues to attract ever-increasing research interest. Delivering a quadrotor from a large fixed-wing aircraft is a promising application of quadrotors. In such an application, the quadrotor needs to switch from a highly unstable status, featured as large initial states, to a safe and stable flight status. This is the so-called large-scale stability control problem. In such an extreme scenario, the quadrotor is at risk of actuator saturation. This can cause the controller to update incorrectly and lead the quadrotor to spiral and crash. In this article, to safely control the quadrotor in such scenarios, the control input constraint is analyzed. The key states of a quadrotor dynamic model are selected, and a two-dimensional dynamic model is extracted based on a symmetrical body configuration. A generalized point-wise min-norm nonlinear control method is proposed based on the Lyapunov function, and large-scale stability control is hence achieved. An enhanced point-wise, min-norm control is further provided to improve the attitude control performance, with altitude performance degenerating slightly. Simulation results showed that the proposed control methods can stabilize the input-constrained quadrotor and the enhanced method can improve the performance of the quadrotor in critical states.

Keywords
Quadrotor, control input constraint, large-scale stability, control-Lyapunov function

Date received: 30 June 2016; accepted: 29 August 2016

Topic: Special Issue - Intelligent Flight Control for Unmanned Aerial Vehicles
Topic Editor: Mou Chen

Introduction

Hailed for its marvelous dynamical capabilities, the quadrotor has proved to be a reliable tool in a wide array of applications such as surveillance, transportation, aerial photography, and disaster response. The near-hovering control of the quadrotor has been extensively studied by many researchers from various perspectives.[1–3] However, problems arise in considering the potential application of delivering a quadrotor from a fixed-wing aircraft, as stabilizing a quadrotor with “large initial states” is a challenging issue. The combination of a quadrotor and a fixed-wing aircraft can meet the requirement of a fast response to emergencies in low airspace or on ground by exploiting the advantages of two types of aircraft and significantly enhancing the quadrotor’s working range. The fixed-wing aircraft excels in flight in the context of large timescales and geographical ranges. However, it cannot maneuver in low airspace at low speed. The quadrotor has a small profile and can carry a variety of payloads. More importantly, it has a simple mechanism (as is depicted in Figure 1) and can be designed to be very portable, which makes it the best candidate in the helicopter family. The fixed-wing plane and the quadrotor are two kinds of aircrafts with significant differences in their flight envelopes, which lead to problems in the scenario being considered. The most obvious problem occurs because the fact that a fixed-wing plane needs to deliver the

Shenyang Institute of Automation, Chinese Academy of Sciences (CAS), Shenyang, China

Corresponding author:
Yuqing He, Shenyang Institute of Automation, Chinese Academy of Sciences, 114 Nanta Street, Shenhe District, Shenyang 110016, Liaoning Province, P. R. China.
Email: heyuqing@sia.cn

quadrotor at a certain height at relatively high horizontal speed, and the released quadrotor needs to start up a certain time after it is released to avoid collision with the aircraft. This means that at the very beginning of its control, the quadrotor is already in the so-called large initial states. In such an extreme situation, where the initial velocity of the quadrotor is large and its initial altitude is high, it is difficult for a quadrotor with normal linear controllers to fulfill the task of stabilizing altitude without crashing. This is nontrivial when the quadrotor is restricted to actuation limitation. With a simpler transmission mechanism, the quadrotor is more conducive to the incorporation and implementation of advanced control algorithms to deal with potential problems.

Although no quadrotor control algorithm available at present can be directly adopted to the above aims, by reviewing past work, crucial aspects of the challenge can be surmised. Early studies on quadrotors mostly focused on dynamic modeling and basic control while hovering. Some research groups investigated the inner loop stability of the quadrotor in the presence of disturbance[4–8] but did not consider the actuator limitation. With the increasing number of applications of quadrotors, modeling and control aimed at mere equilibrium do not satisfy the need for more difficult maneuvering scenarios, such as acrobat flying, which is drawing increasing attention. In Mellinger et al.' study,[9] researchers at the General Robotics, Automation, Sensing and Perception (GRASP) laboratory reported their development of an iterative pattern learning method for quadrotors to realize the challenging maneuver of passing through several slots placed at large angles. In each iteration, the controller parameters were updated to generate better control. A similar maneuver was implemented in the study by Mellinger and Kumar[10] using a geometric method. In contrast to Mellinger's method, Lupashin[11] parameterized the main control values in a flip maneuver and used a learning method to obtain a better control parameter in each loop. As a result, flips, double flips, and triple flips were realized with great precision. The learning algorithm was based on a trial-and-error method, was limited to a known scenario, and therefore is inapplicable to arbitrary situations. To overcome this disadvantage, researchers at the GRASP laboratory[10] developed an algorithm that allows real-time generation of optimal trajectories through a sequence of three-dimensional positions and yaw angles. Constraints on velocity, acceleration, and the inputs of the quadrotor were considered in planning the trajectory to render it traceable. Similarly, Hehn et al.[12] used the bang–bang control algorithm to develop a temporally optimal state-to-state control algorithm. A two-dimensional (2-D) first principles model was deduced to simplify the problem, and the time-optimal trajectory was calculated using a search method in 2-D space. Only a few studies discuss the large initial states problem. Wang and Su[13] designed a switch control algorithm for problems involving a large initial attitude with external disturbances and internal uncertainties in consideration. Similar studies were carried out by Faessler et al.[14] and Sanchez et al.,[15] where controllers successfully stabilized the quadrotor starting from a large initial attitude. However, the outer loop states were not considered. The abovementioned studies are helpful in at least two ways: First, the researchers confirmed the capacity of the quadrotor for aggressive maneuvering. Second, they provided a reliable model of the quadrotor. However, past work does not directly address the application considered in this study. The learning strategies of abovementioned literature can attain high precision in terms of position and altitude control and resist disturbance. However, it should be trained before use, and the application scenarios need to be identical to the training situation. The geometric and optimal methods have other disadvantages as well. For instance, the algorithms used in these methods involve massive amounts of calculation, which is beyond the capability of most controllers for quadrotors.

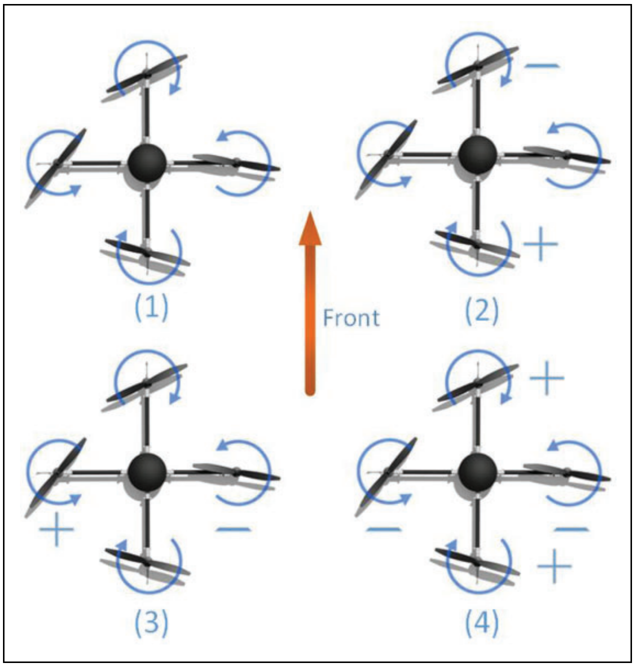

Figure 1. Configuration of the quadrotor.

In our proposed application, the quadrotor is supposed to be dropped from a fixed-wing plane and finally operate at its equilibrium states (namely, zero states). As is well known, the fixed-wing plane has a minimum speed owing to its flying mechanism. Thus, there is a large gap between its large initial states and normal operating states (which is normally the equilibrium state). This article aims at designing a stable control algorithm to safely switch the quadrotor from large initial states to a normal operating state. For this stability control problem, the control input constraint is a key factor that affects the stability of the system. This is because in order to converge to equilibrium from the large initial states, the quadrotor is supposed to generate a large control input. An improper control algorithm incorrectly updates and hence fails to stabilize the quadrotor. This is further explained in Quadrotor actuation constraints. The limitation in the source of actuation needed to attain these two goals leaves us with the question: What is the appropriate choice of control to exercise within the control constraint to satisfy both aims? The chosen strategy needs to be simple in terms of the calculation required so that it can be implemented in a quadrotor controller. As constraints incur nonlinear characteristics, which invariably imply large amounts of calculation and no analyze solution, the ideal method for this scenario would operate offline in order to deal with such complex constraints as the learning strategy. It would also need to use a simple strategy to stabilize the system, which is acceptable for normal quadrotors on board chips.

He and Han[16] proposed a control-Lyapunov function for robust control as well as a generalized point-wise min-norm control that can handle inputs and outputs in nonlinear systems with constraints. This article adopts this algorithm and improves it to fit quadrotor dynamics and its actuation constraint. The control-Lyapunov function is a method to design controllers for nonlinear systems using the Lyapunov function. Compared with the robust controller design of the traditional Lyapunov function, it omits the "guess-trial" process. It is also known as the direct Lyapunov function robust control method. By introducing the control-Lyapunov function, the constraints on the controller can be analyzed offline, in contrast to the original design. By dealing with the problem offline, massive calculations online are avoided, and only simple control calculations are conducted to generate a guiding control output. This method features various combinations of the control algorithm providing diverse control targets such as stabilization and tracking.

The remainder of this article is organized as follows: In section "Quadrotor actuation constraint and dynamics," the actuation constraint of the quadrotor is discussed and the dynamic model of the quadrotor in two key dimensions is introduced. Section "Controller design" introduces the method of generalized point-wise min-norm control and describes its adaptation to quadrotor dynamics. An augmented method is also provided here. In section "Simulation," we describe two simulations to test our methods. The first simulation highlighted the problem facing with a linear controller and showed how the proposed controller can address the control constraint barrier. The second showed that the proposed augmented algorithm betters off in attitude control. We discuss potential avenues for future work in section "Conclusion."

Quadrotor actuation constraint and dynamics

As it is an important factor in the stabilization problem, we discuss the form of the actuation constraint. A simplified dynamic of the quadrotor with two degrees of freedoms is established, and usage conventions for the entire article are provided in this section.

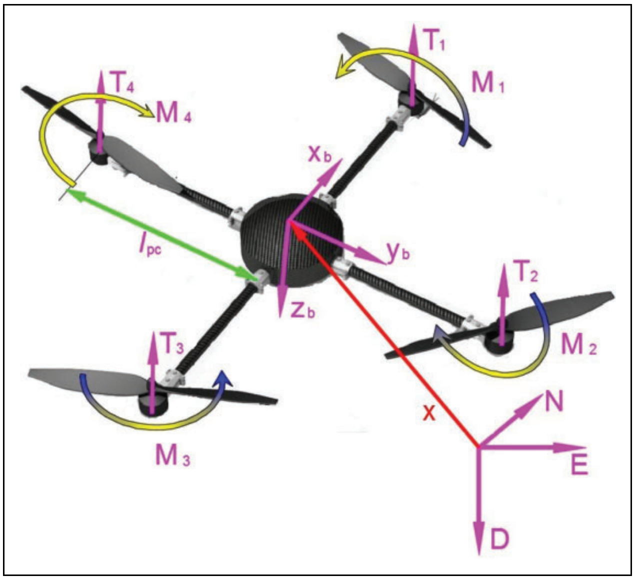

Figure 2. Convention of the coordinates.

Quadrotor actuation constraints

A convention should be confirmed before the dynamics are analyzed. As shown in Figure 2, the north-east-down frame is chosen as the reference coordinate fixed on the ground at point o_e from where the quadrotor is assumed to take off. A body frame $o_b x_b y_b z_b$ is attached to the quadrotor with origin coincident with the center of gravity of the quadrotor. For the sake of simplicity of coupling in actuators, the arrangement of the frame is chosen in the "+" manner. Unit vector $o_b x_b$ points to the head of the quadrotor, $o_b y_b$ points to the right of the quadrotor, and $o_b z_b$ points downward.

The constraint on the quadrotor is an important factor in this article. The generalized forces acting on the quadrotor are the original forces and torques generated by the propellers. However, in the controller design, the control input is to be in accordance with the freedoms established in the dynamic equation. Hence, the map from the original forces and torques to the control outputs should be examined.

For a single actuator, the limitation on the rate of rotation is Ω_{lmt}. Thus, the actuator works in the following zone: $0 \leq \Omega_i \leq \Omega_{\text{lmt}}$. The relationship between the generalized forces and the propellers' rates is depicted by equation (1). By applying the map from the ration rates to the generalized forces, the actuation zone is obtained, as shown in Figure 3. For now, we analyze the T-M_1 space, considering that the quadrotor is highly symmetric and this space is representative. T_{total} stands for the total thrust of the two actuators and $M_{\max}$ is the maximum torque. Figure 3 shows the conversion from the actuation constraint zone

$$\begin{bmatrix} T \\ M_1 \end{bmatrix} = \begin{bmatrix} b & b \\ -l_{pc}b & l_{pc}b \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_3^2 \end{bmatrix} \tag{1}$$

In this article, we use simplified quadrotor dynamics. There are three reasons for the choice of this model. First,

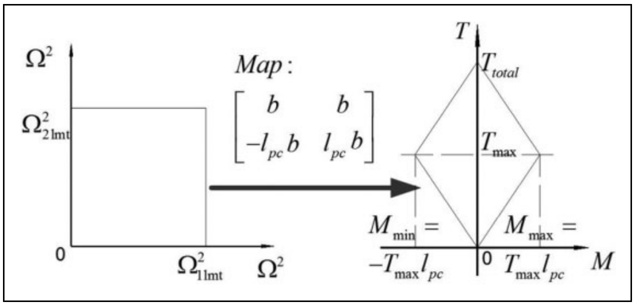

Figure 3. Actuation constraint of a 2-D quadrotor model.
2-D: two-dimensional.

the quadrotor is highly symmetrical in terms of configuration and dynamics. In our application, the trajectory of the dropped quadrotor is mainly in the *xoz* plane. By quickly adjusting the quadrotor's attitude (this is simple because the attitude loop is in full actuation and the timescale is much smaller than in the problem discussed here), the dynamics is representative in a vertical plane; we call it the *xoz* plane. Second, the dropping zone is supposed to be flat and vast, which is common. Hence, of all states, horizontal freedom is far less important than that along the other two axes, namely the vertical axis and the pitch. Third, the dynamics of the actuator are neglected because actuator dynamics involve small time delays and are assumed to be instantaneous in the problem at hand

$$\begin{pmatrix} \ddot{z} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} g - T\cos\theta \\ M/I_{yy} \end{pmatrix} + \begin{pmatrix} \Delta_1 \\ \Delta_2 \end{pmatrix} \tag{2}$$

where z is the vertical position, θ is the pitch angle, g stands for gravitational acceleration, Δ_1 is the disturbance in vertical freedom, and Δ_2 is the disturbance in pitch freedom. T stands for thrust, M for torque, and I_{yy} is the inertia about the x-axis.

The actuator constraint of the quadrotor also explains a special phenomenon: when a large vertical error is imposed on the quadrotor, the controller generates a large collective force. If the force is greater than T_{total}, the actual force input is T_{total}. As can be seen from the constraint, torque is zero; this causes the quadrotor to spiral and crash.

Controller design

In this section, we provide two algorithms. First, a nonlinear controller based on generalized point-wise min-norm control is designed. The main objective of this controller is to coordinate the inner and outer outputs of the quadrotor controller, so that the full states of the quadrotor are stabilized through actuation limitation. Second, to improve the performance of the quadrotor at high altitude, an augmented point-wise min-norm control is designed. The proposed algorithm functions better in situations where attitude control is more critical than vertical control.

Generalized point-wise min-norm control

The proposed algorithm is based on the concept of the control-Lyapunov function.[16] It can be stated as follows:

For an autonomous dynamical system

$$\dot{x} = f(x, u)$$

where $x\epsilon R$ stands for the states of the system, $u\epsilon R$ is the control input for the system; we want to feedback stabilize it to $x = 0$ in some domain $D\epsilon R$.

A control-Lyapunov function is a continuously differentiable, positive-definite function V: $D \rightarrow R$, and such that: $\forall x \neq 0$, $\exists u$

$$\dot{V}(x, u) = \nabla \cdot f(x, u) < 0$$

Furthermore, the Lyapunov function $V(x)$ is a local robust control-Lyapunov function for system , such that

$$\inf_{u\in U} \sup_{\omega\in W(x)} \begin{bmatrix} \dfrac{\partial V}{\partial x} f(x,\omega) + \dfrac{\partial V}{\partial x} g(x,\omega)u \\ + \dfrac{1}{2\gamma^2} \dfrac{\partial V}{\partial x} q(x,\omega) q^T(x,\omega) \dfrac{\partial V^T}{\partial x} \\ + \dfrac{1}{2} h(x,\omega) h^T(x,\omega) \end{bmatrix} < 0$$

Consider the nonlinear system of the form

$$\begin{cases} \dot{x} = f(x,\omega) + g(x,\omega)u + q(x,\omega)\Delta \\ y = h(x,\omega) \\ u \in U(x) \subset R^m \\ \omega \in W(x) \subset R^p \end{cases} \tag{3}$$

where x stands for the states of the system, $y\epsilon R^t$ is the output of the system, $u\epsilon R^t$ is the control input for the system, $u\epsilon R^p$ is an uncertain parameter of the system, $\Delta\epsilon R^r$ is the external disturbance acting on the system, and f, g, q, and h are smooth functions with corresponding dimensions. U and $W(x)$ are the control set and the disturbance set, respectively. If $V(x)$ is a local robust control-Lyapunov function, and $U(x)$ and $W(x)$ satisfy the following conditions:

1. Uncertain constraint $W(x)$ is continuous set-valued mapping, $x\epsilon D_{c1}/D_{c2}$, and $\left(D_c \cong \{x|V(x) \leq c\}\right)$ is a nonempty compact set.
2. $U(x)$ is a lower, semi-continuous, convex set-valued enclosed mapping.
3. $\xi(x)$ is a continuous function of quadrotor state x.

The following conclusion can be drawn:

There exists a positive continuous function $a(x)$ such that equation holds

$$\inf_{u\in U} \sup_{\omega\in W(x)} \begin{bmatrix} \dfrac{\partial V}{\partial x} f(x,\omega) + \dfrac{\partial V}{\partial x} g(x,\omega)u + \\ \dfrac{1}{2\gamma^2} \dfrac{\partial V}{\partial x} q(x,\omega) q^T(x,\omega) \dfrac{\partial V^T}{\partial x} + \dfrac{1}{2} h(x,\omega) h^T(x,\omega) \end{bmatrix} < -a(x) \tag{4}$$

The following controller (generalized point-wise min-norm control of affine nonlinear controller)

$$u(x) = \underset{u \in K_V(x)}{argmin} \|u - \xi(x)\| \tag{5}$$

$$s.t. \sup_{\omega \in W(x)} \left[\begin{array}{l} \frac{\partial V}{\partial x} f(x,\omega) + \frac{\partial V}{\partial x} g(x,\omega)u + \\ \frac{1}{2\gamma^2} \frac{\partial V}{\partial x} q(x,\omega) q^T(x,\omega) \frac{\partial V^T}{\partial x} + \frac{1}{2} h(x,\omega) h^T(x,\omega) \end{array} \right] \leq -a(x) \tag{6}$$

makes the system from disturbance to output L_2 stable with gain less than γ.

ξ is a continuous function of quadrotor state x, and $u \epsilon K_V$ satisfies the following conditions: $K_V(x) \triangleq L_V(x) \cap U(x) = \{u \in U(x) | h(x,u) \leq -a(x)\}$.

Where $L_V(x) \triangleq \{u \in R^t | \hbar(x,u) \leq -a(x)\}$

$$\hbar_1(x,\bar{\omega}) = \frac{\partial V}{\partial x} f(x,\bar{\omega}) + \frac{1}{2\gamma^2} \frac{\partial V}{\partial x} q(x,\bar{\omega}) q^T(x,\bar{\omega}) \frac{\partial V^T}{\partial x} + \frac{1}{2} h^T(x,\bar{\omega}) h(x,\bar{\omega})$$

$$\hbar_2(x,\bar{\omega}) = \frac{\partial V}{\partial x} g(x,\bar{\omega})$$

The generalized point-wise min-norm control is a robust control method aimed at nonlinear systems with control constraints. It can deal with bounded uncertainty and is more flexible in design for various performance requirements.

Quadrotor controller based on generalized point-wise min-norm control

The design of the generalized point-wise min-norm controller expressed by equation for the quadrotor can be separated into three main parts. First, $K_V(x)$ should be solved given the form of restraint on the quadrotor. Second, the guiding function $\xi(x)$ is built. Third, to complete the design, the solution of equation should be calculated.

1. $K_V(x)$ is defined in section "Generalized point-wise min-norm control."

$$K_V(x) = L_V(x) \cap U(x) \tag{7}$$

K_v is shown in Figure 3 and redrawn in Figure 4, which is enclosed by four line segments—l_1, l_2, l_3, and l_4—with point group P_c: P_{c1}, P_{c2}, P_{c3}, and P_{c4} as the end points of the line segments, respectively. $U(x)$ is defined by the actuation constraint, which is analyzed in section "Quadrotor Actuation Constraint and Dynamics." As stated in section "Generalized point-wise min-norm control," $L_v(x)$ is the actuation zone that guarantees the stability of the system without considering the input constraint. It is calculated based on the Lyapunov function and eventually depends on the states of the system. The Lyapunov function is a scalar function of the states of the system; in our set, it is the sum of the squares of states of the equivalent state equation. To calculate $L_v(x)$, we need to obtain the control-Lyapunov function. This is done through the feedback linearization method proposed in the study by Jiang et al.[17]

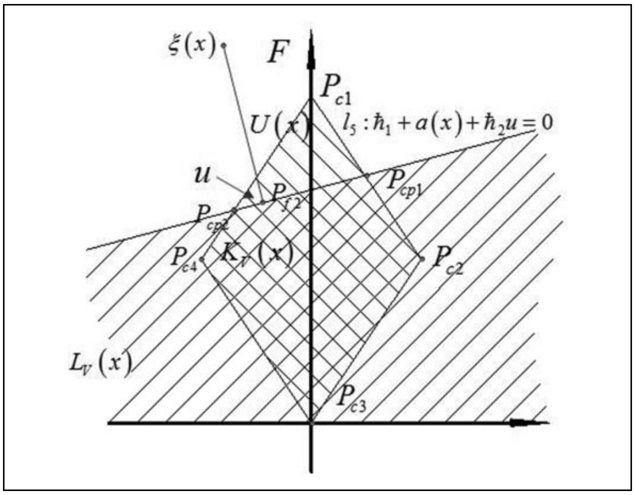

Figure 4. Control input.

2. Based on the linearized system, we design a linear controller $\xi(x) = Ke$ as the guiding function, where $e = x_d - x, x = [x_1, x_2, x_3, x_4], x_d = [x_{1d}, x_{2d}, x_{3d}, x_{4d}]$.

This strategy is used because the feedback linearization technology converts the nonlinear system into a virtual linear system, and the input of the virtual linear system is noted as vector v with components $v_1$and v_2, the virtual collective force input and the virtual torque input in the virtual linear system, respectively. The linear controller design technique can thus be implemented to calculate v. When the linear virtual control inputs are calculated, the real control input u is composed of actual collective force T_{total} and torque M_1 and can be solved by considering the relationship between u and v: $T_{\text{total}} = (g - v_1)m/\cos x_3$; $M_1 = v_2$. $v_1$and v_2 are virtual control inputs from the previous step.

3. We present the calculation of the analyzed control input solution for two situations depending on the relative position of the guiding function $\xi(x)$ and the polygon, as is shown in Figure 4.

In the first situation, where $\xi(x) \epsilon K_V(x)$, control input u can be calculated by equation: $u = \xi(x)$.

Note that $\beta_{n,l}$ is the angle between vectors n and l. Through simple geometrical analysis, the control input solution is obtained as follows.

For the sake of concise representation, a complete deduction is not given in this article. The details can be found in the work done by Jiang et al.[17]

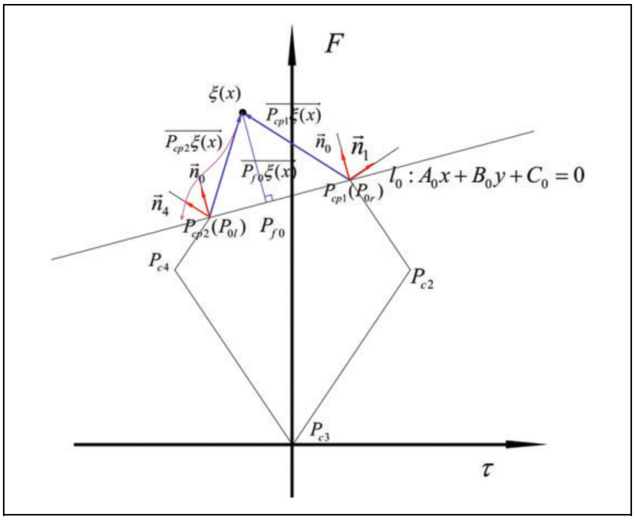

Figure 5. Calculation of the actual control input.

$$u(x) = \begin{cases} \xi(x) \quad \xi(x) \in K_V(x) \\ P_{il} \quad \beta_{\vec{n}_i,\vec{\xi}_{il}} + \beta_{\vec{n}_{4-i},\vec{\xi}_{il}} = \beta_{\vec{n}_{4-i},\vec{n}_i} \wedge (\vec{n}_i \cdot \vec{n}_{4-i}) \cdot \vec{\xi}_{il} > 0 \\ P_{fi} \quad \beta_{\vec{n}_i,\vec{\xi}_{il}} + \beta_{\vec{l}_i,\vec{\xi}_{il}} = \beta_{\vec{l}_i,\vec{n}_i} \wedge \beta_{\vec{n}_i,\vec{\xi}_{ir}} + \beta_{\vec{l}_i,\vec{\xi}_{ir}} = \beta_{-\vec{l}_i,\vec{n}_i} \\ P_{ir} \quad \beta_{\vec{n}_i,\vec{\xi}_{il}} + \beta_{\vec{n}_{i+1},\vec{\xi}_{il}} = \beta_{\vec{n}_{i+1},\vec{n}_i} \wedge (\vec{n}_i \cdot \vec{n}_{4-i}) \cdot \vec{\xi}_{ir} > 0 \end{cases} \tag{8}$$

where i in the subscripts represent the number of line sections, l for the left, r stands for the right, and f represents the perpendicular foot of ξ. For example, P_{il} represents the left end point of line section i. The subscript i is arranged clockwise along the polygon's contour.

Augmented quadrotor controller based on generalized point-wise min-norm

In the control of a quadrotor, the inner loop control (namely the attitude control) should be given sufficient consideration. This is because attitude is critical to the safety of the quadrotor. It would be plausible to make the best of the torque ability of the quadrotor within the limit of the constraint. In the situation shown in Figure 5, line 0 intersects the convex polygon and leaves a relatively tight horizontal constraint for torque generation. In this subsection, we describe an augmented algorithm that involves deliberately lowering line 0 (the new line section is marked with 0′). The horizontal constraint is therefore enlarged, with stabilization ensured. To maximize the horizontal constraint, line 0′ is supposed to pass through points P_{c4} and P_{c2}, which makes it a horizontal line section. For simpler expression of the analytical results, the original line sections 2 and 3 are renamed 1 and 2 here.

There are five positions for line 0 relative to the polygon. The augmented algorithm differs from the original only in the situation shown in Figure 5: namely the line intersects polygon sides 1 and 4. In the other situations, the analytical solution of the control input u of the augmented algorithm remains the same as the original one.

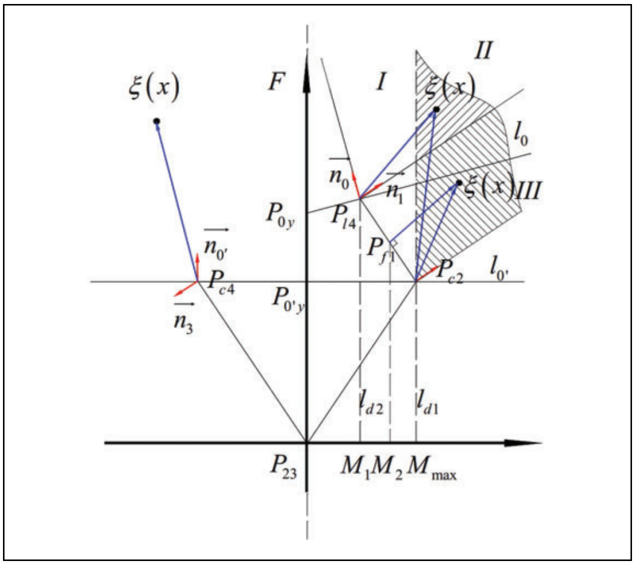

Figure 6. The analyzed resolution of the augmented algorithm, and a comparison of the two algorithms.

Table 1. Characteristics of the quadrotor.

Item	m (kg)	I_{yy} (kg m^2)	Ω_{lmt} (r/min)
Value	0.7	0.06	7100
Item	l_{pc} (m)	c_{thr}	c_{torq}
Value	0.2	1.356e-5	3.391e-7

For the sake of concise representation, Figure 6 is divided into two parts by a vertical broken line (that overlaps with the T-axis). The left part is to represent the analyzed resolution of the augmented algorithm, and the right part shows a comparison between the original algorithm and the augmented algorithm in terms of the calculated torque.

As in the original algorithm, the choice of actuation input depends on the relative position of the guiding function $\xi(x)$ and the polygon. The left part of Figure 6 is representative and can be used to illustrate the calculation. Note the coordinates of the guiding function in plane T-M (ξx, ξy). The other notations for line sections remain the same as in section "Augmented quadrotor controller based on generalized point-wise min-norm," with only letter i standing for i'. P_{c2} and P_{c4} are noted as P_{c1} and P_{c2} in this part. The left part of the figure is divided into three subparts by norm vector n'_0 and norm vector n'_3

$$u(x) = \begin{cases} \xi(x) \quad \xi(x) \in K_{Va}(x) \\ P_{il} \quad \beta_{\vec{n}_i,\vec{\xi}_{il}} + \beta_{\vec{n}_{2-i},\vec{\xi}_{il}} = \beta_{\vec{n}_{2-i},\vec{n}_i} \wedge (\vec{n}_i \cdot \vec{n}_{4-i}) \cdot \vec{\xi}_{il} > 0 \\ P_{fi} \quad \beta_{\vec{n}_i,\vec{\xi}_{il}} + \beta_{\vec{l}_i,\vec{\xi}_{il}} = \beta_{\vec{l}_i,\vec{n}_i} \wedge \beta_{\vec{n}_i,\vec{\xi}_{ir}} + \beta_{\vec{l}_i,\vec{\xi}_{ir}} = \beta_{-\vec{l}_i,\vec{n}_i} \\ P_{ir} \quad \beta_{\vec{n}_i,\vec{\xi}_{il}} + \beta_{\vec{n}_{i+1},\vec{\xi}_{il}} = \beta_{\vec{n}_{i+1},\vec{n}_i} \wedge (\vec{n}_i \cdot \vec{n}_{2-i}) \cdot \vec{\xi}_{ir} > 0 \end{cases}$$

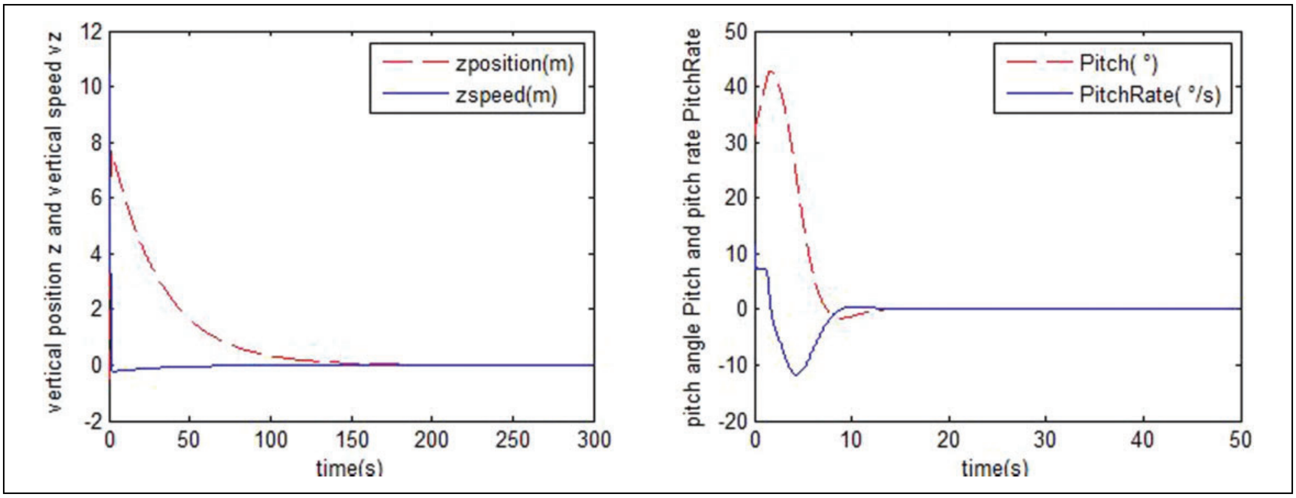

Figure 7. Vertical stabilization of position and velocity.

The right part shows the advantage of the augmented algorithm over the original one in terms of torque control. As can be easily deduced, the augmented algorithm is superior to the original when the guiding function is in two situations—namely, when the guiding function is in area II and in area III, marked by a slash and a backslash, respectively. When the guiding function is in area II, the result of the original algorithm is P_{l4} and that of the augmented algorithm is P_{c2}. The horizontal coordinates of the two points are $(A_0C_1B_0 - A_1C_0B_0)/(A_0^2B_1 - A_0A_1B_0) - C_0/A_0$, and $M_{\max}$, respectively (A_i, B_i, C_i, and D_i stand for the coefficients in the equation $A_ix + B_iy + C_i = 0$ of line i). Hence, the difference between the two algorithms in the torque input in this situation is $M_{\max} - (A_0C_1B_0 - A_1C_0B_0)/(A_0^2B_1 - A_0A_1B_0) - C_0/A_0$. When the guiding function is in area III, the result of the original algorithm is P_{f1} and that of the augmented algorithm is $M_{\max}$. The horizontal coordinates of the two points are $({B_4}^2\xi_x - A_4B_4\xi_y - A_4C_4)/({A_4}^2 + {B_4}^2)$ and $M_{\max}$, respectively. Thus, the difference between the two algorithms in this situation is $M_{\max} - ({B_4}^2\xi_x - A_4B_4\xi_y - A_4C_4)/({A_4}^2 + {B_4}^2)$. The results of the original algorithm for the situation in Figure 6 areM_1 and M_2, respectively, and are much smaller than the maximum value $M_{\max}$. This implies that the potential actuation ability of the system is weakened. The proposed augmented algorithm instead chooses the maximum value of the torque. In Figure 6, the two $\xi(x)$s are close to each other, while their outcomes are significantly different. This algorithm is especially suitable for initial states with large altitude errors, and vertical convergence is not as important, as the algorithm weakens the vertical input constraint.

Proof of convergence of the augmented algorithm: For the sake of concise representation, we name the polygon enclosed by P_{0y}, $P_{0'y}$, P_{l4}, and P_{c2} as K_{vb}, and name that enclosed by P_{c2}, $P_{0'y}$, and P_{23} as K_{va}. We name the input of the augmented algorithm $u_a(x)$, and it is obverse $u_a(x)\epsilon K_{va}$. The generalized point-wise min-norm control tells that as long as the control input u is within K_v, the derivative of the Lyapunov function V is smaller than 0. In our set, the Lyapunov function is chosen as the sum of the squares of the states of the equivalent state equation. Hence, if the Lyapunov function is negative along the states' trajectory, the convergence is guaranteed. Therefore, if we can prove $u_a(x)\epsilon K_v$, the convergence of the states can be proven. From Figure. 6, it is easy to draw the conclusion that $K_v = K_{va} \cup K_{vb}$; then, $K_{va} \subset K_v$; hence, if $u_a(x)\epsilon K_{va}$, $u_a(x)\epsilon K_v$. Thus, the convergence of the states and the stabilization of the system are guaranteed.

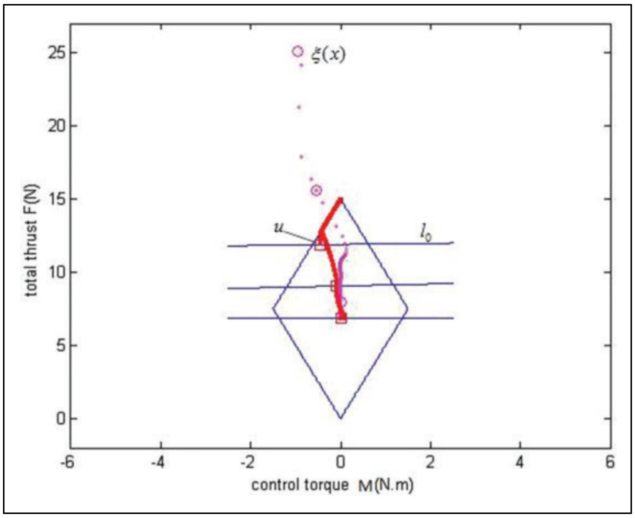

Figure 8. Actuation constraint and control.

Simulation

We ran our simulations on a Dell laptop with a 32-bit CPU with 4 GB of RAM. The code was written in MATLAB installed on the Windows 7 operating system.

In the two simulations, the model properties were chosen as reasonable values according to common sense. The other properties are given in Table 1.

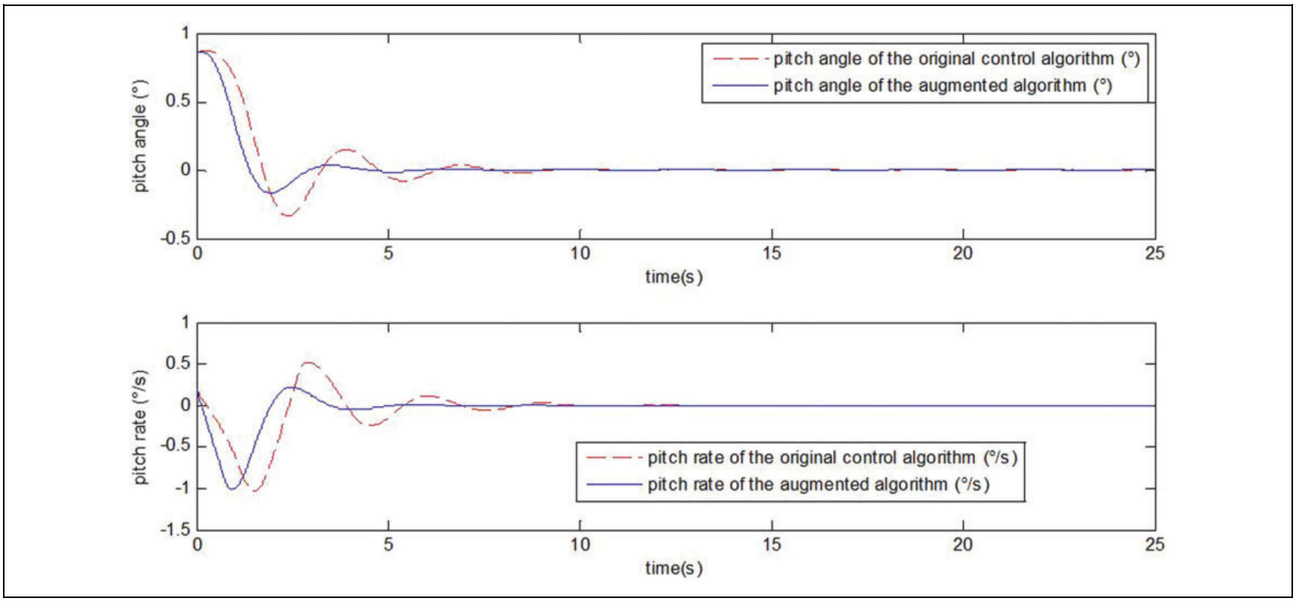

Figure 9. Comparison of inner loop control outcomes of the two algorithms.

Stabilization using generalized point-wise min-norm control

The generalized point-wise min-norm controller was used to stabilize the quadrotor in this situation. The Lyapunov function $V(x)$ was calculated online, and the feedback coefficient matrix was set as $K = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}$.

Through trial and error, matrix P satisfying $AP + PA = -Q$ was obtained

$$P = \begin{bmatrix} 8 & 2 & 0 & 0 \\ 2 & 4 & 0 & 0 \\ 0 & 0 & 8 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix}, Q = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 12 \end{bmatrix}$$

where A is the state matrix in the linearized state equation, and Q is a positive-definite matrix.

The stable zone $K_V(x)$ was calculated online based on given states x. This ensured the stability of the system. With guiding function $\xi(x)$, the preferred control output was confirmed. In contrast to simulation A, the section line cut off the head of the rhombus when its counterpart in the simulation in A was stuck there.

In Figure 7, the simulation was set up with large initial states. All the states are stabilized. Both vertical position and vertical velocity converged to zero. The maximum error in the position was approximately 8 m; hence, the controller mildly stabilized it. Velocity converged to zero in the way as position did. The inner loop stabilized considerably more quickly than the outer one. Instead of over 2 min, the inner loop stabilized within 20 s.

Figure 8 shows the evolution of the controller output in the *T-M* plane. The constraint on the actuator was confined to the rhombus in blue. The group of lines intersecting the rhombus were resolved for $L_V(\mathrm{x})$. The guiding control is drawn in the pink circle and the actual control is shown in red line. The control method can stabilize the quadrotor, when the actuators are in saturation. This controller is much better than a linear controller, which fails to stabilize the quadrotor in this case. In Figure 8, only three lines (which accord with l_5 in Figure 4) are shown, for a clearer image.

Comparison between original and augmented algorithm

Figure 9 shows the attitude control results of the two algorithms. To stress the difference in attitude control between the two algorithms, the initial errors were deliberately exaggerated in comparison with the first simulation to provide a larger guiding torque control input. Because of attitude control, the convergence time was shortened from 10 s to 5 s. This proved the efficiency of the augmented algorithm for attitude control, as it did not weaken the torque control constraint.

Figure 10 shows the vertical freedom control results for the two algorithms. Vertical control performance was slightly weakened. In our configuration, vertical error at 100 s was 1 m greater than in the original control algorithm. The velocity convergence time increased from 40 s to 60 s. This was acceptable as attitude control was greatly enhanced.

To summarize, the proposed methods first calculated the stable control force zone, through the guiding function, chose a preferred control input in this zone. This ensured stabilization in the first place by simultaneously considering guiding control. The simulation of the augmented algorithm showed its advantage in attitude control and provides a better choice than the original one when attitude performance is preferred.

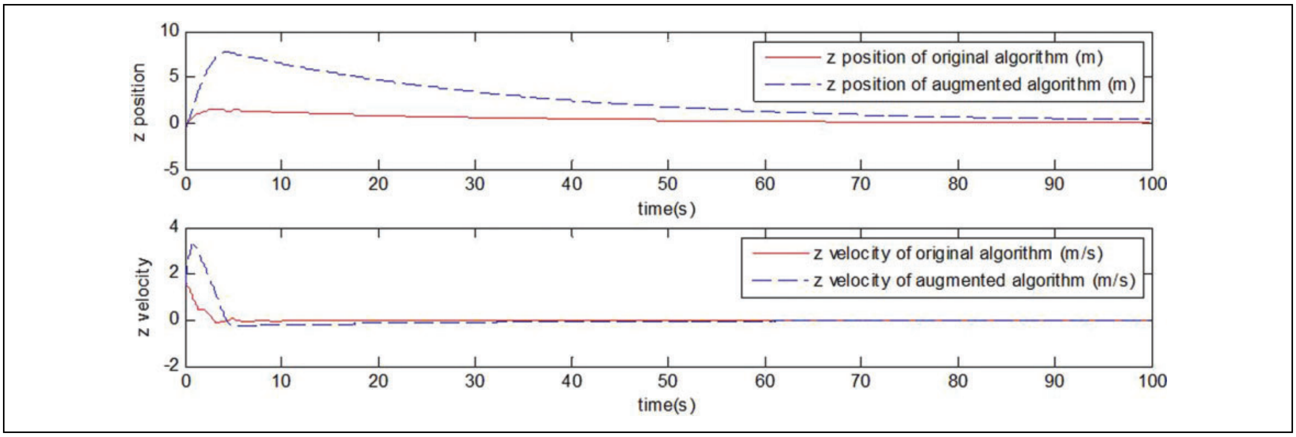

Figure 10. Comparison of the position and velocity control results of the two algorithms.

Conclusion

A simple linear controller fails to stabilize quadrotors due to the saturation of the actuators with large initial states. To cope with this problem, in this article, we first studied the form of actuation of the quadrotor. By adapting the generalized point-wise min-norm control of an affine nonlinear controller to the actuation constraint of the quadrotor, a new controller was implemented on the simplified quadrotor model with vertical and pitch freedoms. Our simulations showed that the nonlinear controller can satisfactorily stabilize a quadrotor with large initial states. This study is the first to consider this situation and used a simple 2-D model. In future work, we intend to create a full-state controller on the premise that the form of actuation of the quadrotor in higher dimensions is convex.

Author's Note

Jun Jiang is also affiliated to University of Chinese Academy of Sciences, China.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: This work was supported by National Natural Science Foundation of China (Grant Nos. 61503369 and 61433016).

References

1. Bouabdallah S. *Design and control of quadrotors with application to autonomous flying*. PhD Thesis, EPFL, Lausanne, 2007.
2. Huang H, Hoffmann GM, Waslander SL, et al. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In: *IEEE international conference on robotics and automation*, Kobe, Japan, 12–17 May 2009, pp. 3277–3282. IEEE. DOI: 10.1109/ROBOT.2009.5152561.
3. Powers C, Mellinger D, Kushleyev A, et al. Influence of aerodynamics and proximity effects in quadrotor flight. In: Desai JP, Dudek G, and Khatib O, et al. (eds) *Experimental robotics. Springer tracts in advanced robotics. Vol. 88*. New York: Springer International Publishing, 2013, pp. 289–302. DOI: 10.1007/978-3-319-00065-7_21.
4. Fernando T, Chandiramani J, Taeyoung L, et al. Robust adaptive geometric tracking controls on SO(3) with an application to the attitude dynamics of a quadrotor UAV. In: *IEEE conference on decision and control and European control conference*, Orlando, FL, USA, 12–15 December 2011.
5. Liu H, Li D, Xi J, et al. Robust attitude controller design for miniature quadrotors. *Int J Robust Nonlin* 2015; 26(4): 681–696. DOI: 10.1002/rnc.3332.
6. Satici AC, Poonawala H, and Spong MW. Robust optimal control of quadrotor UAVs. *IEEE Access* 2013; 1: 79–93. DOI: 10.1109/ACCESS.2013.2260794.
7. Waslander SL, Hoffmann GM, Jung Soon J, et al. Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning. In: *IEEE/RSJ international conference on intelligent robots and systems*, Alberta, Canada, 2–6 August 2005. DOI: 10.1109/IROS.2005.1545025.
8. Hao L, Xiafu W, and Yisheng Z. Quaternion-based robust attitude control for uncertain robotic quadrotors. *IEEE T Ind Inform* 2015; 11(2): 406–415. DOI: 10.1109/TII.2015.2397878.
9. Mellinger D, Michael N, and Kumar V. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *Int J Robot Res* 2012; 31(5): 664–674. DOI: 10.1177/0278364911434236.
10. Mellinger D and Kumar V. Minimum snap trajectory generation and control for quadrotors. In: *IEEE international conference on robotics and automation*, Shanghai, China, 9–13 May 2011, pp. 664–674. IEEE. DOI: 10.1109/ICRA.2011.5980409.
11. Lupashin S, Schollig A, Sherback M, et al. A simple learning strategy for high-speed quadrocopter multi-flips. In: *IEEE international conference on robotics and automation*,

Anchorage, Alaska, USA, 3–8 May 2010, pp. 1642–1648. DOI: 10.1109/ROBOT.2010.5509452.

12. Hehn M, Ritz R, and D'Andrea R. Performance benchmarking of quadrotor systems using time-optimal control. *Auton Robot* 2012; 33(1–2): 69–88. DOI: 10.1007/s10514-012-9282-3.
13. Wang L and Su J Switching control of attitude tracking on a quadrotor UAV for large-angle rotational maneuvers. In: *IEEE international conference on robotics and automation*, Hong Kong, China, May 31–June 7 2014, pp. 1050–4729. DOI: 10.1109/ICRA.2014.6907277.
14. Faessler M, Fontana F, Forster C, et al. Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor. In: *IEEE International Conference on Robotics and Automation*, Seattle, Washington, 26–30 May 2015, pp. 1722–1729. DOI: 10.1109/ICRA.2015.7139420.
15. Sanchez A, Parra-Vega V, Garcia O, et al. Time-parametrization control of quadrotors with a robust quaternion-based sliding mode controller for aggressive maneuvering. In: *Control Conference*, Zürich, Switzerland, 17–19 July 2013, pp. 3876–3881.
16. He Y and Han J. A robust generalized pointwise min-norm controller for input constrained nonlinear systems. *Biopolymers* 2010; 30(7): 79–89.
17. Jiang J, Yan L, He Y, et al. Large scale stabilization of multicopter flying robot with input constraint. 2016.