LETTER

A Data-Driven Gain Tuning Method for Automatic Hovering Control of Multicopters via Just-in-Time Modeling

Tatsuya KAI[†a)], ***Member and*** **Ryouhei KAKURAI**[††], ***Nonmember***

SUMMARY This study develops a new automatic hovering control method based on just-in-time modeling for a multicopter. Especially, the main aim is to compute gains of a feedback control law such that the multicopter hovers at a desired height and at a desired time without overshoot/undershoot. First, a database that contains various hovering data is constructed, and then the proposed method computes gains for a query input from the database. From simulation results, it turns out that the multicopter achieves control purposes, and hence the new method is effective.

key words: *multicopters, automatic hovering control, just-in-time modeling, data-driven control, gain tuning*

1. Introduction

Multicopters have been well known as the name "drones," and utilized as not only hobbies but also industrial devices in the various fields whole entire world [1], [2]. They are expected to play active roles peculiarly in aerial photography, commodity distribution, agriculture, inspection work and so on [3]. Moreover, with the improvement of automatic driving technologies for automobiles, automatic operation techniques for multicopters have been eagerly researched. However, the number of accidents and troubles that are caused by multicopters is increasing recently, and regulations on multicopters by laws are being tightened. Therefore, safe and high accuracy control of multicopters are strongly needed. Researches on flight control of multicopters are mainly based on model-based control, which is a controller synthesis method by using mathematical models of multicopters. On the other hand, few studies on data-driven control for multicopters have been done so far, but it is expected that such new approaches provide high accuracy control of multicopters by blending of big data technologies.

This study shall give a new data-driven approach to automatic hovering control, and we especially focus on "just-in-time modeling," which is one of the data-driven control methods. First, in Sect. 2, the settings on the multicopter and its automatic hovering control problem are given. Next, Sect. 3 proposes a gain tuning method of automatic hovering control for the multicopter based on just-in-time modeling. Then, numerical simulations are performed in order to confirm effectiveness of the proposed method in Sect. 4.

Manuscript received July 29, 2022.
Manuscript revised August 16, 2022.
Manuscript publicized August 29, 2022.

[†]The author is with the Faculty of Advanced Engineering, Tokyo University of Science, Tokyo, 125-8585 Japan.
[††]The author is with TOTO LTD., Kitakyushu-shi, 802-8601 Japan.
a) E-mail: kai@rs.tus.ac.jp

DOI: 10.1587/transfun.2022EAL2064

2. Problem Settings

This section gives the problem settings of this study. First, a model of a multicopter as a controlled object is explained. We deal with a multicopter with 4 rotors (so-called "quadcopter") as shown in Fig. 1 [1], [2], [4], and assume that there is no disturbance for the multicopter. Let us denote the center of gravity for the multicopter by (x, y, z), and its attitude angle by (ϕ, θ, ψ), which is represented by roll, pitch, and yaw angles. For the physical parameters, m is the mass of the multicopter, l is the distance between the center gravity and each rotor, I_x, I_y, I_z are the inertia moments for the multicopter, about x, y, z axes, respectively, J is the inertia moments for the rotor, Ω is the sum of rotational velocities for all the rotors, g is the acceleration of gravity. In addition, u_i $(i = 1, 2, 3, 4)$ are the control inputs which are redefined from rotational velocities of the rotors [4].

Then, the equation of motion for translational movement is given by

$$\begin{cases} \ddot{x} = \dfrac{1}{m}(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\,u_1, \\ \ddot{y} = \dfrac{1}{m}(\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi)\,u_1, \\ \ddot{z} = \dfrac{1}{m}\cos\phi\cos\theta\,u_1 - g, \end{cases} \tag{1}$$

and the one of motion for rotational movement is given by

$$\begin{cases} \ddot{\phi} = \left(\dfrac{I_y - I_z}{I_x}\right)\dot{\theta}\,\dot{\psi} - \dfrac{J}{I_x}\Omega\,\dot{\theta} + \dfrac{l}{I_x}u_2, \\ \ddot{\theta} = \left(\dfrac{I_z - I_x}{I_y}\right)\dot{\phi}\,\dot{\psi} + \dfrac{J}{I_y}\Omega\,\dot{\phi} + \dfrac{l}{I_y}u_3, \\ \ddot{\psi} = \left(\dfrac{I_x - I_y}{I_z}\right)\dot{\phi}\,\dot{\theta} + \dfrac{1}{I_z}u_4. \end{cases} \tag{2}$$

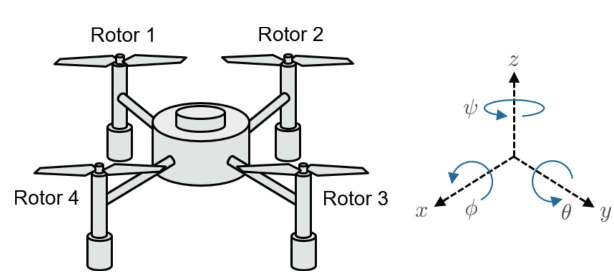

Fig. 1 A multicopter model.

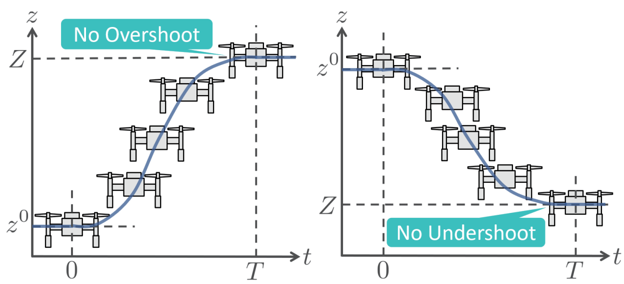

Fig. 2 Automatic hovering control without overshoot/undershoot.

Hence, the set of (1) and (2) represents the equation of motion for the multicopter. In this study, we shall consider the next problem on automatic hovering control for the multicopter (see also Fig. 2).

Problem 1: For the multicopter (1), (2) hovering at the height $z = z^0\,(t = 0)$, design the control inputs $u_i\,(i = 1, \cdots, 4)$ such that the multicopter transfers to the desired height $z = Z$ at the desired time $t = T$ with keeping stable attitude and without overshoot or undershoot.

Note that Problem 1 requires not only convergence of the height for the multicopter but also the stabilization time.

3. Hovering Control for Multicopter via JIT Modeling

This section will propose an automatic hovering control method as a solution to Problem 1. Let us consider the next control law [4]:

$$\begin{cases} u_1 = \dfrac{m}{\cos\phi\cos\theta}\{g - k_1(z - Z) + k_2\dot{z}\}, \\ u_2 = -\dfrac{I_x}{l}(\phi - \Phi) - k_3\dot{\phi}, \\ u_3 = -\dfrac{I_y}{l}(\theta - \Theta) - k_4\dot{\theta}, \\ u_4 = -I_z(\psi - \Psi) - k_5\dot{\psi}, \end{cases} \tag{3}$$

where Z, Φ, Θ, Ψ are desired value of z, ϕ, θ, ψ, respectively, and $k_i > 0\,(i = 1, \cdots, 5)$ are gains. It is known that for the closed-loop system which is derived by substituting (3) into (1), (2), $\lim_{t\to\infty}(z(t), \phi(t), \theta(t), \psi(t)) = (Z, \Phi, \Theta, \Psi)$ holds. That is, the multicopter can hover at the desired height Z by (3). However, it is quite difficult to tune the five gains $k_i\,(i = 1, \cdots, 5)$ in order to avoid unstable attitude of the multicopter and overshoot/undershoot. Thus, this study provides a data-driven method to tune the gains, and especially we focus on “just-in-time (JIT) modeling,” which is one of the data-driven control methods [5], [6]. If the desired attitude of the multicopter is equal to the initial one in (3), it is known that the attitude of the multicopter keeps stable for small values of k_3, k_4, k_5 [4]. Hence, we consider a tuning method for the gains k_1, k_2 based on JIT modeling.

First, let us derive a construction method of a database for JIT modeling. For a given height Z, perform simulations for the multicopter (1), (2) with the control inputs (3), and obtain gains k_1, k_2 such that there is no overshoot or undershoot, and the condition:

$$|z(t) - Z| < \varepsilon, \ \forall t \geq T \tag{4}$$

holds for a given value $\varepsilon > 0$ and a time T. Then, a data (Z, T, k_1, k_2) is retracted into the database. Repeating the above computation method for various data of Z, we construct a database whose inputs are (Z_i, T_i) and outputs are $(k_{1,i}, k_{2,i})$ (N: the number of data in the database, data are represented as $(Z_i, T_i, k_{1,i}, k_{2,i})\ (i = 1, \cdots, N)$). Next, we consider a calculation method of gains for a given input data via JIT modeling. For a query data (Z^*, T^*), calculate a distance between it and each input data in the database as

$$d_i = \sqrt{(Z_i - Z^*)^2 + (T_i - T^*)^2}, \tag{5}$$

and add the distance d_i values to the database as $(Z_i, T_i, k_{1,i}, k_{2,i}, d_i)\ (i = 1, \cdots, N)$. Then, sort all the data in the database in the ascending order of the distance, and the sorted data is represented by $(\tilde{Z}_i, \tilde{T}_i, \tilde{k}_{1,i}, \tilde{k}_{2,i}, \tilde{d}_i)\ (i = 1, \cdots, N)$ (the symbol $\tilde{}$ stands for “sorted data”). After that, we extract $M\ (< N)$ data from the database in ascending order with respect to (5). From the extracted data $(\tilde{Z}_i, \tilde{T}_i, \tilde{k}_{1,i}, \tilde{k}_{2,i}, \tilde{d}_i)\ (i = 1, \cdots, M)$, the predicted output (k_1^*, k_2^*) for the query input (Z^*, T^*) is calculated by

$$k_1^* = \frac{\sum_{i=1}^{M} \dfrac{\tilde{k}_{1,i}}{\tilde{d}_i}}{\sum_{i=1}^{M} \dfrac{1}{\tilde{d}_i}}, \quad k_2^* = \frac{\sum_{i=1}^{M} \dfrac{\tilde{k}_{2,i}}{\tilde{d}_i}}{\sum_{i=1}^{M} \dfrac{1}{\tilde{d}_i}}, \tag{6}$$

where an inverse number of the distance $1/\tilde{d}_i$ is utilized as a weights in weighted mean, and hence it puts higher values for data near the query input. Finally, the predicted output k_1^*, k_2^* is applied to the multicopter (1), (2), and we can expect that it transfers from the initial height z^0 to the desired height Z^*.

4. Simulations

This section shows numerical simulations in order to check validity of the proposed method shown in Sect. 3. The physical parameters of the multicopter as set as $m = 0.4\,\mathrm{kg}$, $l = 0.248\,\mathrm{m}$, $I_x = I_y = 0.01467\,\mathrm{kgm}^2$, $I_z = 0.02331\,\mathrm{kgm}^2$, $J = 0.0001757\,\mathrm{kgm}^2$, $g = 9.807\,\mathrm{m/s}^2$.

In construction of a database, we set $z^0 = 0$ without loss of generality, and the ranges and intervals for Z, k_1, k_2 are set as shown in Table 1. The initial and final attitude of the multicopter are set as the same values: $(\phi, \theta, \psi) = (0, 0, 0)$, and hence k_3, k_4, k_5 are fixed as small values: $k_3 = k_4 = k_5 = 0.001$. As a result, a database that contains only data satisfying no overshoot/undershoot and (4) is obtained with $N = 1547$, where $\varepsilon = 0.01$. By using the database, we perform numerical simulations. It is assumed that the multicopter first hovers at the initial height for 10 s.

Consider four simulation settings as shown in Table 2, and the number of neighbor data in JIT modeling is set as $M = 100$. Figs. 3–6 illustrate the time histories of the height

Table 1 The ranges and intervals for construction of a database.

Data	Minimum	Maximum	Intervals
Z	−5 m	5 m	0.5 m
k_1	0.02	0.4	0.001
k_2	0.2	1.3	0.001

Table 2 The simulation settings and the obtained gains.

No.	Type	z^0	Z^*	T^*	k_1^*	k_2^*
I	Up	3.85 m	5.82 m	14.35 s	0.1943	0.828
II	Up	3.85 m	8.69 m	20.24 s	0.3098	0.5693
III	Down	9.52 m	5.82 m	14.35 s	0.1934	0.8082
IV	Down	9.52 m	8.59 m	20.24 s	0.00977	0.5693

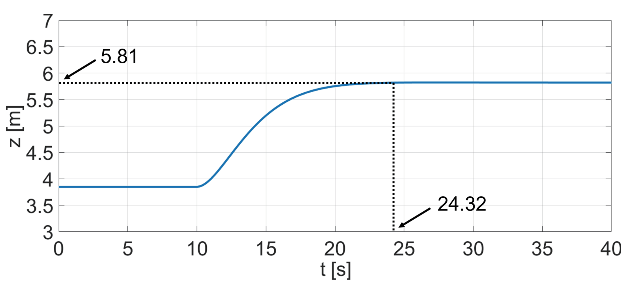

Fig. 3 The time history for the height of the multicopter (Simulation I).

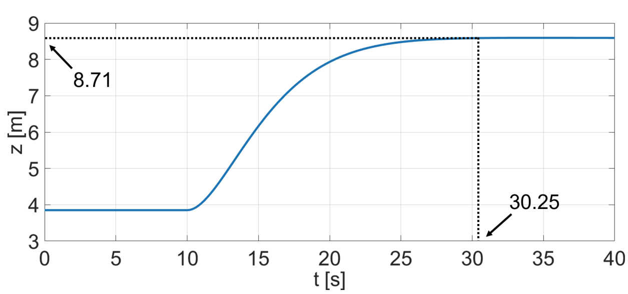

Fig. 4 The time history for the height of the multicopter (Simulation II).

of the multicopter as simulation results. From Figs. 3 and 4, it turns out that in Simulation I and II, the multicopter goes up to the desired height Z^* at the desired time T^* with small error and without overshoot. In addition, from Figs. 5 and 6, we can see that the multicopter goes down to the desired height Z^* at the desired time T^* with small error and without undershoot in Simulation III and IV. In all the simulations, it is confirmed that the attitude of the multicopter is always stable. Moreover, the computation times of the control gains by JIT modeling is quite small, thus real time hovering control for the multicopter is realized. Consequently, from the simulation results we can see that the proposed method is effective.

5. Conclusions

In this letter, a new automatic hovering control method based

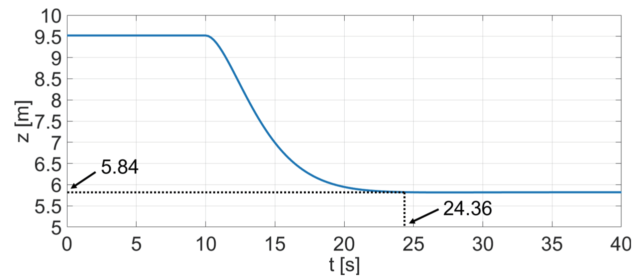

Fig. 5 The time history for the height of the multicopter (Simulation III).

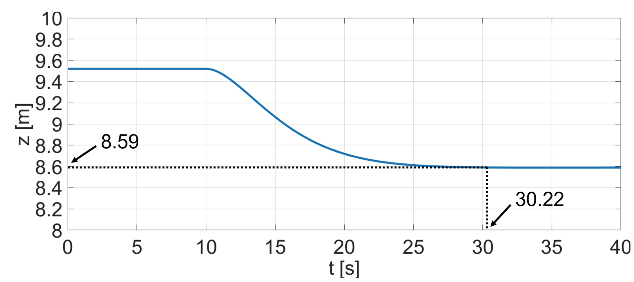

Fig. 6 The time history for the height of the multicopter (Simulation IV).

on JIT modeling has been developed. Especially, the method computes the gains of the controller such that the multicopter can hover at the desired heights at the desired time without overshoot/undershoot by using a database that contain various hovering data. Some simulations show that the multicopter can achieve given control purposes, and hence the new method has effectiveness. Future themes for multicopters are as follows: experimental verification using real multicopter, 3D automatic path following control, and recovery control to stable attitudes from disturbances. This work was partly supported by JSPS KAKENHI Grant Numbers 19K04460.

References

[1] Q. Quan, Introduction to Multicopter Design and Control, Springer, 2017. DOI: 10.1007/978-981-10-3382-7

[2] M. Labbadi, Y. Boukal, and M. Cherkaoui, Advanced Robust Nonlinear Control Approaches for Quadrotor Unmanned Aerial Vehicle, Springer, 2021. DOI: 10.1007/978-3-030-81014-6

[3] R. Krishnamurthi, A. Nayyar, and A.E. Hassanien, Development and Future of Internet of Drones (IoD), Springer, 2021. DOI: 10.1007/978-3-030-63339-4

[4] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," Proc. International Conf. on Robotics and Automation, vol.5, pp.4393–4398, 2004. DOI: 10.1109/ROBOT.2004.1302409

[5] A. Stenman, F. Gustafsson, and L. Ljung, "Just in time models for dynamical systems," Proc. 35th IEEE Conf. on Decision and Control, pp.1115–1120, 1996. DOI: 10.1109/CDC.1996.574658

[6] Q. Zheng and H. Kimura, "Just-in-time modeling for function prediction and its applications," Asian J. Control, vol.3, no.1, pp.35–44, 2001. DOI: 10.1111/j.1934-6093.2001.tb00040.x