

---

## Control of a quadrotor group based on maximum hands-off distributed control

---

Kimiko Motonaka\* and Takuya Watanabe

Department of Electrical Electronic and Information Engineering,  
Kansai University,  
Osaka, Japan  
Fax: +81-6-6368-1424  
Email: motonaka@kansai-u.ac.jp  
Email: k287771@kansai-u.ac.jp  
\*Corresponding author

Yuhwan Kwon

Graduate School of Science and Technology,  
Nara Institute of Science and Technology,  
Nara, Japan  
Email: y-kwon@is.naist.jp

Masaaki Nagahara

Institute of Environmental Science and Technology,  
The University of Kitakyushu,  
Fukuoka, Japan  
Email: nagahara@kitakyu-u.ac.jp

Seiji Miyoshi

Department of Electrical Electronic and Information Engineering,  
Kansai University,  
Osaka, Japan  
Email: miyoshi@kansai-u.ac.jp

**Abstract:** This paper presents the implementation of maximum hands-off distributed control in a quadrotor group. We assume that individual quadrotors can only communicate with neighbouring quadrotors to obtain state information and do not know the state of the whole group. The maximum hands-off distributed control is an extension of the consensus control using sparse optimal control. It converges the state of a group using only local information while maximising the time when the control input is zero by optimising the consensus control input through sparse optimisation. This minimises the time required for control and reduces energy consumption. We applied this control to a four-quadrotor group using MATLAB and CoppeliaSim simulations. The results confirm that the four quadrotors converged to the same state and the control inputs became sparse. Experiments using four small Tello EDU quadrotors further confirmed that they could reach the same altitude using the maximum hands-off distributed controller.

**Keywords:** multi-agent system; consensus control; sparse optimal control; maximum hands-off distributed control; unmanned aerial vehicle; distributed control.

**Reference** to this paper should be made as follows: Motonaka, K., Watanabe, T., Kwon, Y., Nagahara, M. and Miyoshi, S. (2021) 'Control of a quadrotor group based on maximum hands-off distributed control', *Int. J. Mechatronics and Automation*, Vol. 8, No. 4, pp.200–207.

**Biographical notes:** Kimiko Motonaka received her BE from the Department of System Engineering in 2010, and ME and DE from the Department of Natural Science and Technology in 2012 and 2014 of Okayama University, respectively. She is currently an Assistant Professor at the Department of Electrical and Electronic Engineering at the Kansai University. Her research interests include motion planning and group control for robots. He is a member of IEEE.

Takuya Watanabe received his BE in Electrical, Electronic and Information Engineering from the Faculty of Engineering Science, Kansai University in 2020. He is currently working at the All Nippon Airways Co., Ltd..

Yuhwan Kwon received his BE in Electrical, Electronic and Information Engineering from the Faculty of Engineering Science, Kansai University in 2018, and ME from the Graduate School of Science and Engineering in 2020. He is currently a PhD student at the Graduate School of Science and Technology, Nara Institute of Science and Technology.

Masaaki Nagahara received his Bachelor in Engineering from the Kobe University in 1998, and Master and Doctoral in Informatics from the Kyoto University in 2000 and 2003, respectively. He was a recipient of George S. Axelby Outstanding Paper Award in 2018 and Transition to Practice Award in 2012 from IEEE Control Systems Society. He was also awarded SICE Control Division Research Award (Kimura Award) in 2020, SICE Best Book Authors Award in 2021 and 2016, SICE Best Paper Award in 2012, and IEICE Communications Society Best Tutorial Paper Award in 2014. He is a senior member of IEEE.

Seiji Miyoshi received his BEng and MEng from the Kyoto University, Japan, in 1986 and 1988, respectively, and his PhD from the Kanazawa University, Japan, in 1998. He joined the Faculty of Engineering Science of Kansai University in 2008, where he is currently a Professor. His research interests include statistical-mechanical informatics, image processing, signal processing, and learning theory. He is a senior member of IEICE and IEEE, and a member of JPS and JNNS. He received the 2011 Papers of Editors' Choice from JPSJ, the Excellent Paper Award in ICONIP2016, and the 2019 Best Paper Award from IEICE.

This paper is a revised and expanded version of a paper entitled 'Application of maximum hands-off distributed control to a quadrotor group' presented at IEEE International Conference on Mechatronics and Automation, Takamatsu, Japan, 8–11 August 2021.

## 1 Introduction

The number of situations that require quadrotors as a group is increasing. The principal applications of multiple quadrotors include support in disaster situations, providing infrastructure, and patrols (Chen et al., 2020; Shakhathreh et al., 2016; Cheng et al., 2019; Trotta et al., 2020). The use of a group of quadrotors has the following merits:

- 1 even if the capabilities of an individual quadrotor are low, many quadrotors can be used to cover other capabilities and achieve the same task
- 2 even if some quadrotors fail, other quadrotors can continue to perform a task
- 3 a group of quadrotors can efficiently perform a shared task in parallel.

Sometimes, a central controller is used to control the quadrotor group. In this case, the central controller collects the state information, calculates the control inputs, and sends them to all the quadrotors in the group. However, if the number of quadrotors in a group increases, the communication and computational costs of the central controller also increase proportionally. Thus, it is desirable for each quadrotor to calculate the control input using only the information obtained from their mounted sensors. Groups of living things, such as ants or birds, can make decisions for themselves by collecting information using their sensory organs and constructing a group without a central controller (Hinchey et al., 2007; Slowik and Kwasnicka, 2018).

Methods of controlling a group constructed by agents without a central controller (multi-agent systems) have been widely researched in the robotics and communication fields

as distributed control frameworks (Bandyopadhyay et al., 2017; Oh et al., 2015; Ren et al., 2005; Stone and Veloso, 2020). Here, we focus on consensus control, which is a type of distributed control (Olifati-Saber and Murray, 2004; Olifati-Saber et al., 2007; Qin et al., 2017; Ren et al., 2007). In the context of distributed control, 'reach a consensus' is a situation in which the states of all agents in a group have converged to the same state. The control for reaching a consensus is called 'consensus control'. Ikeda et al. (2019) proposed a maximum hands-off distributed control, which is an extended version of consensus control for a multi-agent system. The concept of maximum hands-off control, which is also called sparse optimal control or dynamical sparse modelling (Ikeda and Nagahara, 2021; Nagahara et al., 2015; Nagahara, 2020), is introduced into the consensus control method for maximum hands-off distributed control. The maximum hands-off distributed control calculates the consensus control input (the control input for reaching a consensus) and optimises it with respect to time using the  $L^0$  norm to maximise the time when the control input is zero. Thus, the time needed for control is minimised, which reduces the energy required for control compared with standard consensus control. Generally, it is not easy to optimise the  $L^0$  norm, especially when the controlled system is nonlinear. However, if the control input consists of only 0 or  $\pm 1$ , the  $L^0$ -optimised control input becomes the same as the  $L^1$ -optimised control input, which can be solved easily.

In this study, we apply maximum hands-off distributed control to a quadrotor group. Specifically, we consider a controlled system of four quadrotors, use maximum hands-off distributed control to control their altitudes and horizontal directions, and confirm that this quadrotor group can reach a consensus. Moreover, we confirmed

that the control inputs were sparse in the time direction. Each quadrotor can communicate only with neighbouring quadrotors to obtain state information. The usefulness of the maximum hands-off distributed control was confirmed by simulations using MATLAB and CoppeliaSim and an actual experiment using Tello EDUs, which are small quadrotors, as shown in Figure 1. The contribution of this study is the application of maximum hands-off distributed control (which has not yet been applied to a specific controlled object) in a quadrotor group and the evaluation of the behaviour of quadrotors.

**Figure 1** Tello EDU quadrotors (see online version for colours)



The remain of this paper is organised as follows. Section 2 describes the consensus control and its extension to the maximum hands-off distributed control. Section 3 presents the MATLAB simulation results. Here, the model of a quadrotor is simplified, and its dynamics are not considered. Section 4 presents the simulation results obtained using CoppeliaSim. In these simulations, the dynamics of the quadrotor are considered in detail. Section 5 describes the experiment using the Tello EDU quadrotors. Finally, Section 6 concludes the study.

## 2 Maximum hands-off distributed control

In maximum hands-off control, the definition of sparsity is expanded, and the control input with sparsity in the time dimension is calculated by taking the  $L^0$  norm of the control input. First, the consensus control input for each quadrotor to reach a consensus was determined. Next, the maximum hands-off distributed control input was calculated by optimising the consensus control input using the  $L^0$  norm over the sampling period. In the following subsections, the details of the consensus control and maximum hands-off distributed control are described.

### 2.1 Consensus control of a multi-agent system

A multi-agent system is composed of multiple agents that can make decisions and move autonomously. In this study, we assumed that the agents are quadrotors.

Consensus control is a basic control method that multi-agent systems can use to converge the states of all

agents (e.g., position, orientation, and speed) to the same state using distributed control. In this study, we assumed that  $N$  agents exist. The  $i^{\text{th}}$  agent determines the control input  $\mathbf{u}_i = [u_{1,i}, \dots, u_{D,i}]^T$  for itself using its state  $\mathbf{x}_i = [x_{1,i}, \dots, x_{D,i}]^T$  and state  $\mathbf{x}_j$  of the  $j^{\text{th}}$  neighbouring agent  $j \in \mathcal{N}(i)$ . Here,  $D$  denotes the dimension of the state,  $\mathcal{N}(i)$  is the set of neighbours of the  $i^{\text{th}}$  agent, and the  $i^{\text{th}}$  agent can obtain only the information of the  $j^{\text{th}}$  agent ( $j \in \mathcal{N}(i)$ ). Then, the dynamics of the  $i^{\text{th}}$  agent are expressed as

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t), \quad \mathbf{x}_i(0) = \mathbf{x}_{0i}, \quad (1)$$

where  $\mathbf{x}_{0i}$  is the initial state. The states of the agents were observed at every sampling period,  $T > 0$ . Each agent determines the consensus control input for the time interval  $t \in [kT, (k+1)T)$  using its state  $\mathbf{x}_i(t)$  at time  $t = kT$  ( $k = 0, 1, 2, \dots$ ). The consensus control input  $\mathbf{u}_i(t)$  was calculated as follows:

$$\mathbf{u}_i(t) = -\epsilon \sum_{j \in \mathcal{N}_i} (\mathbf{x}_i(t) - \mathbf{x}_j(t)). \quad (2)$$

Note that  $\epsilon$  is the constant gain used to adjust the size of the control input.

### 2.2 Extension to maximum hands-off distributed control

Now, we optimise the derived consensus control input  $\mathbf{u}_i(t)$  for the  $i^{\text{th}}$  agent using the maximum hands-off control (Ikeda et al., 2019; Nagahara et al., 2015). We set the discretisation rate to  $\Delta t$  and optimise the  $T/\Delta t$  discrete control inputs over duration  $kT$  to  $(k+1)T$  to obtain the optimal control input sequence  $\mathbf{U}_i^*(kT) = (\mathbf{u}_i^*(kT), \mathbf{u}_i^*(kT + \Delta t), \dots, \mathbf{u}_i^*((k+1)T))$ . Note that  $k = 0, 1, 2, \dots$ , and  $T/\Delta t$  is assumed to be a natural number. Here, we assume that each element  $u_{d,i}^*(t) \square d = 1, \dots, D$  of  $\mathbf{u}_i^*(t)$  has a maximum value constraint at  $t = kT, kT + \Delta t, kT + 2\Delta t, \dots, (k+1)T$  as follows:

$$|u_{d,i}^*(t)| \leq 1. \quad (3)$$

We then define the  $L^0$  norm of the function  $f(t)$  over the time span  $[kT, (k+1)T)$  as

$$\|f(\cdot + kT)\|_0 \triangleq \int_{kT}^{(k+1)T} |f(t)|^0 dt. \quad (4)$$

Minimising the  $L^0$  norm of the control input  $u_{d,i}$  is equivalent to maximising the time when the control inputs are zero during  $[kT, (k+1)T)$ . Therefore, the maximum hands-off distributed control input is calculated by minimising  $\|u_{d,i}(\cdot + kT)\|_0$ , which is the  $L^0$  norm of each element of the consensus control input  $\mathbf{u}_i(t)$  calculated using equation (2). Typically, the  $L^0$ -norm minimisation problem is difficult to solve, particularly when the system is non-convex. However, if the control inputs consist of only the values 0 or  $\pm 1$ , the  $L^0$ -optimal control input becomes the same as the  $L^1$  optimal control input and can be calculated easily (Ikeda et al., 2019; Nagahara et al.,

2015). The algorithm for maximum hands-off distributed control is shown in Algorithm 1.

**Algorithm 1** Maximum hands-off distributed control for agent  $i$

Given a sampling period  $T$  and an initial state  $\mathbf{x}_i(0)$   
**for**  $k = 0, 1, 2, \dots$  **do**  
 1. Observe  $\mathbf{x}_i(kT)$  and  $\mathbf{x}_j(kT)$ ,  $j \in \mathcal{N}_i$ .  
 2. Calculate consensus control input  $\mathbf{u}_i(t)$   
 3. Calculate the maximum hands-off distributed control input

$$\mathbf{U}_i^*(kT) = \arg \min \|u_{d,i}(\cdot + kT)\|_0$$

4. Apply  $\mathbf{U}_i(kT)$  to the agent  $i$ .  
**end for**

Source: Nagahara et al. (2015)

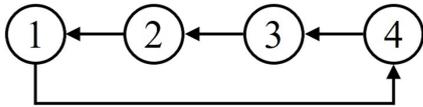
### 3 MATLAB simulations

In this section, we confirm that these quadrotors can reach a consensus with respect to the altitude and horizontal direction in simulations using MATLAB. Here, we considered a group of four quadrotors. The quadrotors had the communication constraints shown in Figure 2 for each quadrotor  $i$ , where  $i \in \{1, 2, 3, 4\}$  and the arrows show the communication direction. Each quadrotor sent its position information to the next quadrotor, as indicated by the arrows. In this simulation, we assumed that the dynamical model of the quadrotors is given by equation (1). The states of the quadrotors were updated using the following formula:

$$\mathbf{x}_i(kT + (n+1)\Delta t) = \mathbf{x}_i(kT + n\Delta t) + \mathbf{u}_i^*(kT + n\Delta t)\Delta t.$$

Here,  $i = 1, 2, \dots, N$ ,  $k = 0, 1, 2, \dots$ , and  $n = 0, 1, \dots, T/\Delta t - 1$ .

**Figure 2** Direction of agent communication

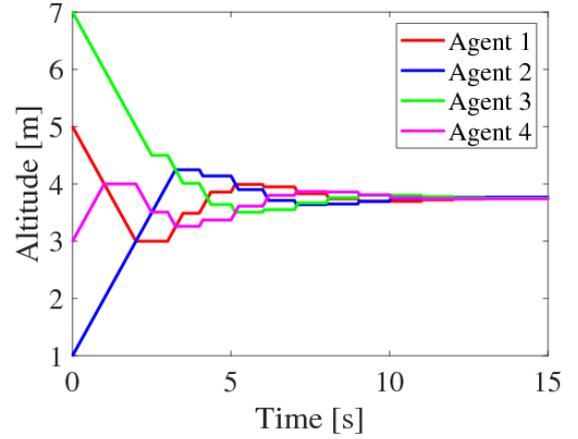


#### 3.1 Altitude control

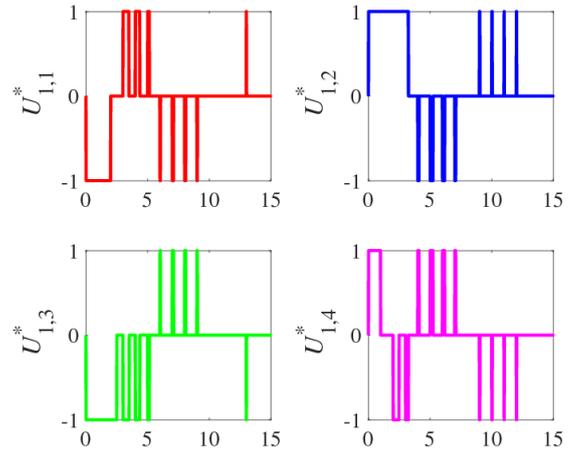
This simulation demonstrated that the altitudes of the four quadrotors can be converged using maximum hands-off distributed control. For this state, dimension  $D = 1$  because only the altitude was controlled in this simulation. The initial altitudes of the four quadrotors were  $\mathbf{x}(0) = [5, 1, 7, 3]^T$  [m], and the gain for adjusting the size of the control input was set to  $\epsilon = 0.5$ . The discretisation rate was  $\Delta t = 0.02$  [s] and the sampling period was  $T = 1$  [s]. Thus, each quadrotor received the position information from the connected quadrotor every 1 [s], calculated  $T/\Delta t = 50$  control inputs as a sequence, and moved during the next

1 [s] using the obtained inputs. By repeating this procedure, the quadrotors reached a consensus.

**Figure 3** Time response of each quadrotor's altitude simulated on MATLAB (see online version for colours)



**Figure 4** Control input for each quadrotor corresponding to the result shown in Figure 3 (see online version for colours)

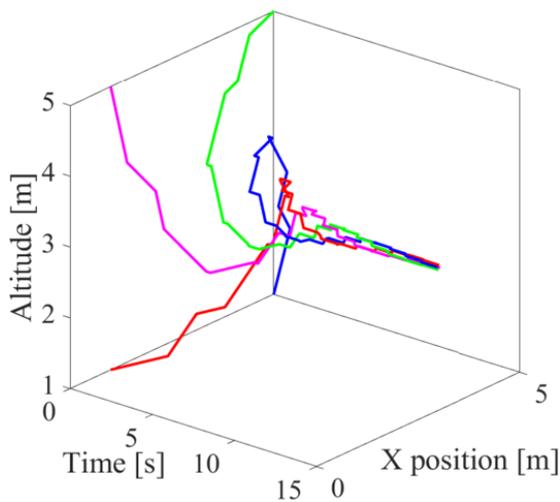


The results are shown in Figures 3 and 4. Figure 3 shows the time responses of the altitudes of the quadrotors, which are differentiated by colour. As shown in the figure, the quadrotors started at different altitudes, which became more similar over time and finally converged to a single value. For example, quadrotor 1, shown by the red line, moved following quadrotor 2, as shown by the blue line, because quadrotor 1 referred only to the state of quadrotor 2, as shown in the communication graph. Here, the sampling period was set to 1 [s]. Therefore, each quadrotor updated the state of the other quadrotor every 1 [s] and then updated the consensus control input. Each quadrotor stayed at the current position while waiting for information from the other quadrotor. Thus, the trajectory graph has an angular shape. Figure 4 shows the time response of the control inputs for each quadrotor, where the colours match those of the quadrotors in Figure 3. The control inputs comprised the value  $\pm 1$  for the minimum necessary time and 0 for other times. Thus, it was confirmed that the control inputs were sparse in the time dimension.

### 3.2 Altitude and horizontal position control

This simulation demonstrates that the altitude and horizontal ( $X$ ) positions of the four quadrotors can be converged. Each quadrotor moved at different depth ( $Y$ ) positions. Therefore, the quadrotors could not collide with each other even if they reached the same altitude and  $X$  position. For this state, dimension  $D = 2$  because only the altitude and  $X$  positions were controlled. For the four quadrotors, the initial states of the horizontal positions were  $\mathbf{x}_{1,:}(0) = [1, 5, 5, 1]^T$  [m] and those of the altitudes were  $\mathbf{x}_{2,:}(0) = [1, 1, 5, 5]^T$  [m]. The gain for adjusting the size of the control input was set to  $\epsilon = 0.5$ , the discretisation rate was  $\Delta t = 0.02$  [s], and the sampling period was  $T = 1$  [s]. The altitude and horizontal directions were separately controlled.

**Figure 5** Time response of the altitude and  $X$  positions of each quadrotor in simulation by MATLAB (see online version for colours)



**Figure 6** Horizontal directional control inputs for each quadrotor corresponding to Figure 5 (see online version for colours)

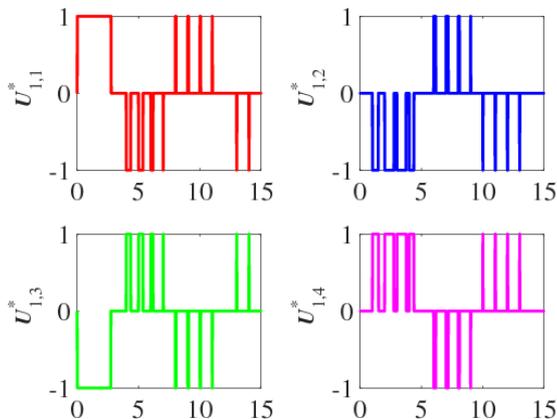
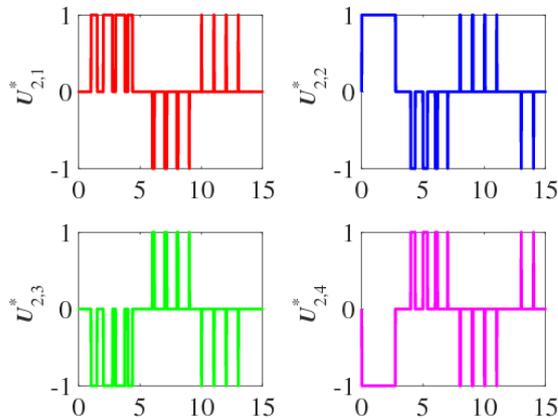


Figure 5 shows the time response of the altitude and horizontal positions of each quadrotor. As shown in the figures, each quadrotor started at a different position and finally reached the same position as the other quadrotors. Figures 6 and 7 show the time response of the horizontal

and altitude directional control inputs for each quadrotor, respectively. From the experiments in Section 3.1, it was confirmed that the control inputs became sparse in the time dimension.

**Figure 7** Altitude directional control inputs for each quadrotor corresponding to Figure 5 (see online version for colours)



## 4 CoppeliaSim simulations

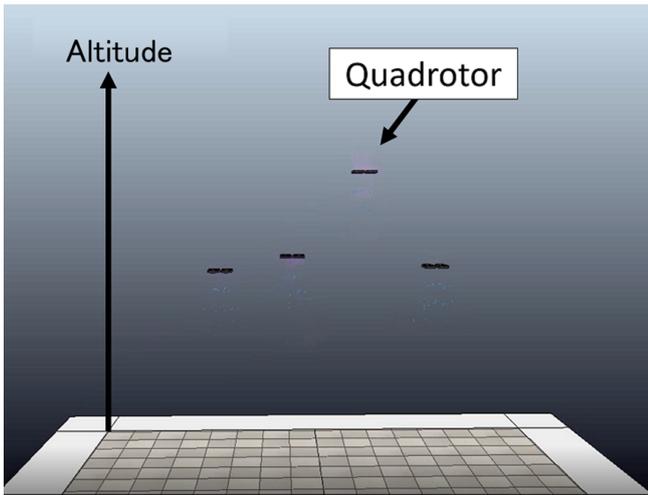
In the MATLAB simulation, a simplified model was used to calculate the quadrotor movement. In this section, we describe the use of maximum hands-off distributed control in four quadrotors using CoppeliaSim (which is a tool for robotic dynamics simulation) and evaluate the performance of the controller with a more realistic quadrotor model. Here, we used the standard quadrotor model provided by CoppeliaSim. This model is equipped with a simple PD position controller that can be easily controlled by specifying the position of the corresponding target ball.

First, we connected MATLAB and CoppeliaSim for mutual communication using the MATLAB application programming interface function. Second, the information of the quadrotors in CoppeliaSim was sent to MATLAB, and the control inputs were calculated using MATLAB. Finally, the calculated control inputs were returned to the quadrotors in CoppeliaSim. The control inputs for the four quadrotors were separately calculated on one machine to simulate a distributed control situation. The control input was given as the target position. Figure 8 shows the four flying quadrotors in the simulation environment of CoppeliaSim.

### 4.1 Altitude control

The initial states of the four quadrotors were set to  $\mathbf{x}(0) = [5, 1, 7, 3]^T$  [m], and the gain for adjusting the size of the control input was set to  $\epsilon = 0.5$ . The discretisation rate was  $\Delta t = 0.05$  [s] and the sampling period was  $T = 1$  [s]. Thus, each quadrotor received the position information from the connected quadrotor every 1 [s], calculated  $T/\Delta t = 20$  control inputs as a sequence, and moved during the next 1 [s] using the control inputs. By repeating this procedure, the quadrotors reached a consensus.

**Figure 8** Simulation environment in CoppeliaSim (see online version for colours)



**Figure 9** Time response of the altitude of each quadrotor in the dynamical simulation using CoppeliaSim (see online version for colours)

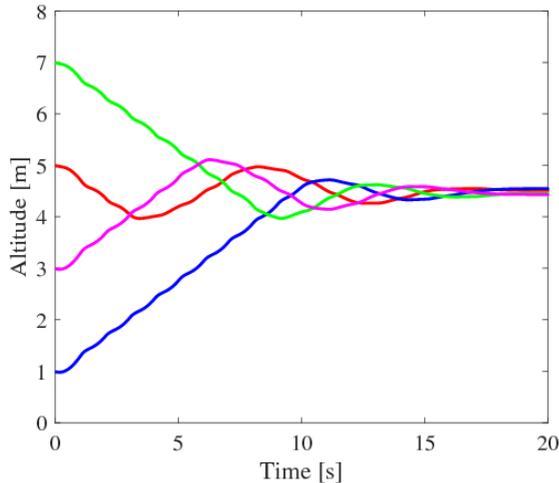


Figure 9 shows the time response of the altitude of the quadrotors when the maximum hands-off distributed control was used to control the altitude. The state information was received from the quadrotors in CoppeliaSim, and the individual trajectories are shown in different colours. The trajectories fluctuated slightly from the MATLAB simulation results owing to the inertia of the quadrotors in CoppeliaSim; nevertheless the quadrotors eventually reached a consensus.

#### 4.2 Altitude and horizontal position control

The initial states of the horizontal positions of the four quadrotors were  $x_{1,:}(0) = [1, 5, 5, 1]^T$  [m] and those of the altitudes were  $x_{2,:}(0) = [1, 1, 5, 5]^T$  [m]. Here, the depth directional positions of each quadrotor were fixed as  $x_{3,:}(0) = [-2, 0, 2, 4]^T$  [m]. The gain for adjusting the size of the control input was set to  $\epsilon = 0.5$ , the discretisation rate was  $\Delta t = 0.02$  [s], and sampling period was  $T = 1$  [s]. The altitude and horizontal directions were separately controlled.

**Figure 10** Time response of the trajectories of each quadrotor in the CoppeliaSim simulation (see online version for colours)

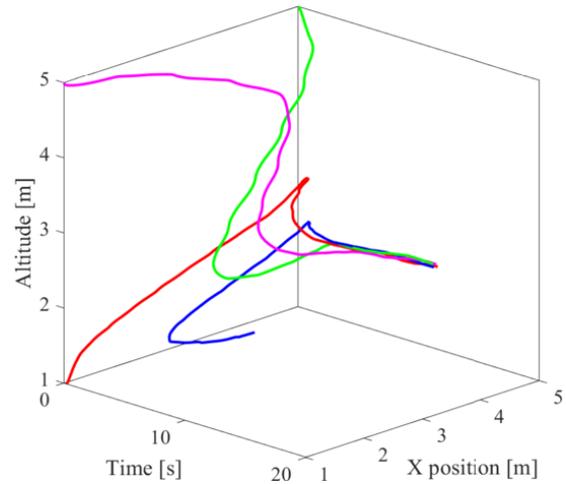


Figure 10 shows the time response of the trajectories of each quadrotor in CoppeliaSim. Figure 11 shows the same results shown in Figure 10, but the 3D graph is divided into 2D graphs for altitude and horizontal position. As shown in the figures, the  $X$  position oscillated more than the altitude. This is because a quadrotor easily slips when it performs lateral motion. However, the quadrotors finally reached a consensus.

## 5 Real experiments

In this section, we present the results of a real experiment using four Tello EDUs developed by Shenzhen Ryze Technology Co., Ltd., as shown in Figure 1. The experimental environment is illustrated in Figure 12. The size of a Tello EDU is  $98$  [mm]  $\times$   $92.5$  [mm]  $\times$   $41$  [mm], and its weight, including the battery, is  $87$  [g]. A Tello EDU is equipped with a camera and sensors, and it can localise its position and determine its state. In addition, a Tello EDU can be connected to Wi-Fi and the simultaneous flight of several quadrotors can be realised through a router. These systems were implemented in Python software. Conventional consensus control does not require a central controller and assumes that the agents communicate directly with each other; however, a Tello EDU cannot communicate with other Tello EDUs directly. Therefore, in this experiment, we used a router and a laptop PC to calculate the control inputs and send them to the quadrotors, thus artificially limiting the communication, as shown in Figure 2.

In this simulation, we evaluated the behaviour of the quadrotors when maximum hands-off consensus control was applied to their altitudes. The initial altitudes of the four quadrotors were  $x(0) = [0.5, 2.0, 1.7, 0.7]^T$  [m], and the gain for adjusting the size of the control input was set to  $\epsilon = 0.8$ . The discretisation rate was  $\Delta t = 0.02$  [s] and the sampling period was  $T = 1$  [s].

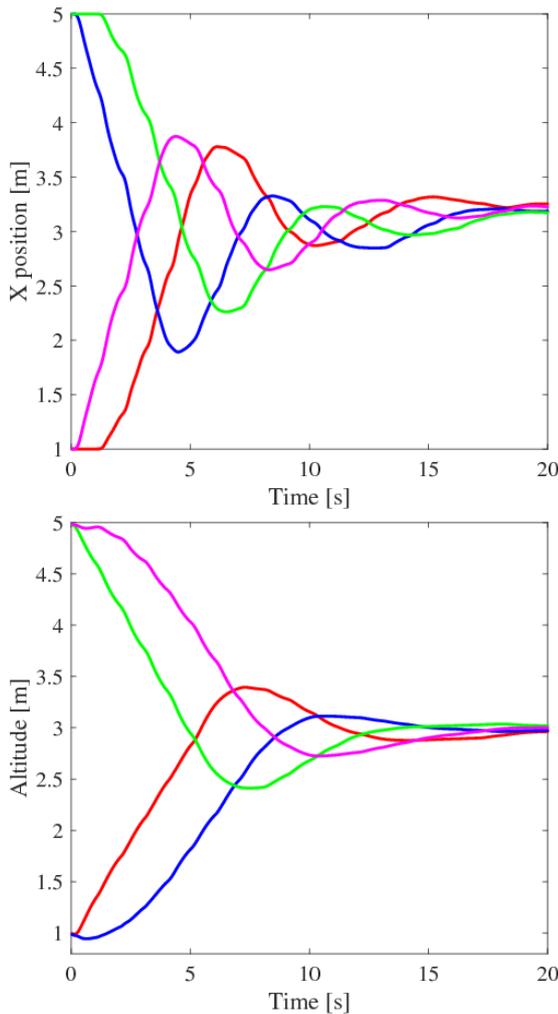
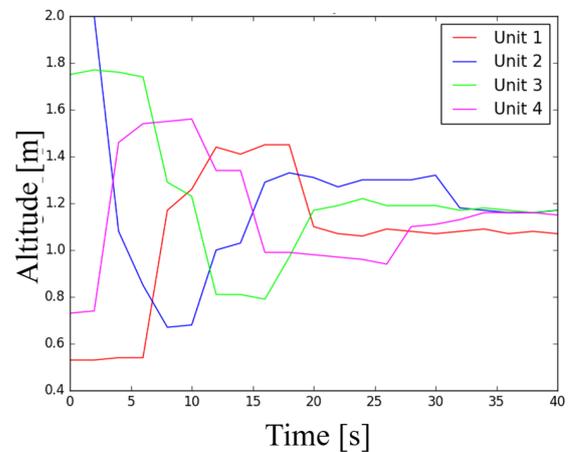
**Figure 11** Time response of altitude and  $X$  positions according to Figure 10 (see online version for colours)

Figure 13 shows the time response of the altitude of each quadrotor in the actual experiments. The altitude was measured using a sensor mounted on each Tello EDU. The trajectories of the quadrotors are shown in different colours.

In the actual experiments, the altitudes of the quadrotors eventually converged to almost the same altitude as that observed in the simulation. Here, the time response was slower than the simulation because the quadrotors sometimes failed to send or receive the state information and needed to wait for communication with other quadrotors to be completed. Sometimes the communication time delay also occurred, but it did not affect the convergence in this experiment. In addition, the trajectory of quadrotor 1 ( $i = 1$ , unit 1; indicated by the red line) converged to an altitude of approximately 0.1 m, which was far from the converged altitude of the other quadrotors. We speculate that quadrotor 1 could not reach the converged altitude either because there was blow-back from the wall or because a Tello EDU cannot perform fine movements. However, it can be considered that the quadrotors reached a consensus as a whole. From the above, we confirm that the maximum hands-off distributed control can converge the states of the quadrotors in an actual experiment.

**Figure 12** Actual experimental environment (see online version for colours)**Figure 13** Time response of the altitudes of each Tello EDU in the actual experiment (see online version for colours)

## 6 Conclusions

In this study, we implemented the maximum hands-off distributed control for a quadrotor group, in which individual quadrotors can only communicate with the specified quadrotor to obtain the state information. We considered a group of four quadrotors and applied maximum hands-off distributed control in simulation environments based on MATLAB and CoppeliaSim. The results confirm that the four quadrotors finally converge to the same state, and the control inputs become sparse. Specifically, the time at which the control input is zero is maximised to reduce the energy needed for the control. In addition, we experimented using a group of four Tello EDU quadrotors and confirmed that the four quadrotors converged to the same altitude using the maximum hands-off distributed controller.

In future work, we will consider dynamic changes in the communication graph. We considered a static communication graph; however, it is effective to change the communication graph dynamically to ensure that each quadrotor receives information from the nearest quadrotor instead of the specified quadrotor. This is also effective for the case in which a quadrotor in the group is broken.

## Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Numbers 18K13782, 18K11483, 20K04494, JP19H02301, JP20H02172, JP20K21008, and 21K14132 and Kansai University Fund for the Promotion and Enhancement of Education and Research, 2018, ‘Development of system analysis and design methods integrating model and data premised on IoT technology’.

## References

- Bandyopadhyay, S., Chung, S. and Hadaegh, F.Y. (2017) ‘Probabilistic and distributed control of a large-scale swarm of autonomous agents’, *IEEE Transactions on Robotics*, Vol. 33, No. 5, pp.1103–1123.
- Chen, E., Chen, J., Mohamed, A.W., Wang, B., Wang, Z. and Chen, Y. (2020) ‘Swarm intelligence application to UAV aided IoT data acquisition deployment optimization’, *IEEE Access*, Vol. 8, pp.175660–175668.
- Cheng, L., Zhong, L., Tian, S. and Xing, J. (2019) ‘Task assignment algorithm for road patrol by multiple UAVs with multiple bases and rechargeable endurance’, *IEEE Access*, Vol. 7, pp.144381–144397.
- Hinchey, M.G., Sterritt, R. and Rouff, C. (2007) ‘Swarms and swarm intelligence’, *Computer*, Vol. 40, No. 4, pp.111–113.
- Ikeda, T. and Nagahara, M. (2021) ‘Maximum hands-off control with time-space sparsity’, *IEEE Control Systems Letters*, Vol. 5, No. 4, pp.1213–1218.
- Ikeda, T., Nagahara, M. and Kashima, K. (2019) ‘Maximum hands-off distributed control for consensus of multi-agent systems with sampled-data state observation’, *IEEE Transactions on Control of Network Systems*, Vol. 6, No. 2, pp.852–862.
- Nagahara, M. (2020) *Sparsity Methods for Systems and Control*, Now Publishers, Boston-Delft.
- Nagahara, M., Quevedo, D.E. and Nešić, D. (2015) ‘Maximum hands-off control: a paradigm of control effort minimization’, *IEEE Transactions on Automatic Control*, Vol. 61, No. 3, pp.735–747.
- Oh, K.K., Park, M.C. and Ahn, H.S. (2015) ‘A survey of multi-agent formation control’, *Automatica*, Vol. 53, pp.424–440.
- Olfati-Saber, R. and Murray, R.M. (2004) ‘Consensus problems in networks of agents with switching topology and time-delays’, *IEEE Transactions on Automatic Control*, Vol. 49, No. 9, pp.1520–1533.
- Olfati-Saber, R., Fax, J.A. and Murray, R.M. (2007) ‘Consensus and cooperation in networked multi-agent systems’, in *Proceedings of the IEEE*, Vol. 95, No. 1, pp.215–233.
- Qin, J., Ma, Q., Shi, Y. and Wang, L. (2017) ‘Recent advances in consensus of multi-agent systems: a brief survey’, *IEEE Transactions on Industrial Electronics*, Vol. 64, No. 6, pp.4972–4983.
- Ren, W., Beard, R.W. and Atkins, E.M. (2005) ‘A survey of consensus problems in multi-agent coordination’, in *Proc. of the 2005 American control Conf.*, pp.1859–1864.
- Ren, W., Beard, R.W. and Atkins, E.M. (2007) ‘Information consensus in multivehicle cooperative control’, *IEEE Control Systems Magazine*, Vol. 27, No. 2, pp.71–82.
- Shakhathreh, H., Khreishah, A., Chakareski, J., Salameh, H.B. and Khalil, I. (2016) ‘On the continuous coverage problem for a swarm of UAVs’, in *Proc. of IEEE 37th Sarnoff Symp.*
- Slowik, A. and Kwasnicka, H. (2018) ‘Nature inspired methods and their industry applications—swarm intelligence algorithms’, *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 3, pp.1004–1015.
- Stone, P. and Veloso, M. (2020) ‘Multiagent system: a survey from a machine learning perspective’, *Autonomous Robots*, Vol. 8, pp.345–383.
- Trotta, A., Montecchiari, L., Felice, M.D. and Bononi, L. (2020) ‘A GPS-free flocking model for aerial mesh deployments in disaster-recovery scenarios’, *IEEE Access*, Vol. 8, pp.91558–91573.