Proceedings of the SICE Annual Conference 2020
September 23-26, 2020, Chiang Mai, Thailand

Multi-Robot Control for Adaptive Caging and Tracking of a Flood Area

Yang Bai[1†], Mikhail Svinin[1], and Evgeni Magid[2]

[1]Department of Information Science and Engineering, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga 525-8577, Japan.
(E-mail: yangbai@fc.ritsumei.ac.jp; svinin@fc.ritsumei.ac.jp)

[2]Department of Intelligent Robotics, Kazan Federal University, Kremlyovskaya str. 35, Kazan 420008, Russian Federation.
(E-mail: magid@it.kfu.ru)

Abstract: In this paper, a control strategy for tracking the propagation of an expanding flood zone by using a group of unmanned aerial vehicles (UAVs), is developed. The control strategy addresses the following two issues. One is to averagely distribute a group of UAVs along the expanding boundary of the flood zone without collisions among each other. The other is to track the propagation of the flood region by keeping its boundary within the field of vision for each UAV. A combined control algorithm is proposed for dealing with both the flocking-based control and adaptive vision-based tracking problems. The feasibility of the algorithm is verified under simulations in the ROS/Gazebo environment.

Keywords: Rescue systems, disaster robotics.

1. INTRODUCTION

Disaster robotics has drawn an increasing attention in the recent decade [1, 2]. It focuses on the design and control of robotic devices, often heterogeneous groups of robots, in the mitigation, management, recovery and rescue operations in natural (earthquakes, tsunami, hurricanes) or human-made (oil spills, mine waste floods, wildfire, nuclear contamination) catastrophes. Researches in this area aim to facilitates human rescue teams in predicting the expansion of a disaster area, speeding up the process of extracting survivors, and evaluating dangers of construction collapse and environment pollution, while increasing the safety of human rescuers and survivors.

In this paper, we are concerned with an essential problem in the field of disaster robotics: utilizing multiple aerial robots to monitor an expanding flood area. The problem requires to develop a control strategy for the UAVs such that the motion of the complete flood area can be caged and tracked, which constitutes the goal of our research. To tackle the problem, we propose a strategy that addresses the following two issues. One is to averagely distribute a group of UAVs along the expanding boundary of the flood zone without collisions among each other (formation control problem). The other is to track the propagation of the flood region by keeping its boundary within the field of vision for each UAV (vision-based tracking problem).

In the literature, methods for the formation control problem can be classified into the following main types: the position-based control, the distance-based control, and the flocking-based control. In the position-based control [3–6], the desired formation of multiple agents is achieved by tracking the position of each agent without any interactions among the agents under ideal conditions. One drawback of this approach is that absolute positions of the UAVs are required. Instead of using the absolute positions, in the distance-based formation control [7–10], agents can sense the relative positions of their neighboring agents with respect to their local coordinate systems. However, our control strategy requires to adaptively encircle the flood zone with a fixed number of UAVs. As the area of the flood region is uncertain, the desired distances between the UAVs are unknown, and therefore, the distance-based method is not feasible.

Instead, a flocking-based control framework is adopted in our strategy, which automatically adjusts the distance between adjacent UAVs. The flocking control is commonly based on the following three rules: cohesion (stay close to nearby neighbors), separation (avoid collisions with nearby neighbors), and alignment (match velocity with nearby neighbors). In the literature [11–15], the cohesion and the separation rules have been usually implemented by means of a potential function of inter-agent distances, and the alignment rule has been implemented by means of velocity consensus of agents. A detailed review of the formation control techniques can be found in [16].

For the vision-based tracking problem, considering the effect of external disturbance and model uncertainties of the UAVs, we propose a robust adaptive control method based on the function approximation technique (FAT) [17–21]. An FAT-based adaptive control algorithm is designed to eliminate the influence of the time varying uncertainty to the control process. We parameterize the uncertainty term with a set of chosen basis functions weighted by unknown parameters. Then, we define update laws such that the parameters of the weighted basis functions can be automatically determined and the variation between the auxiliary system and the original system can be eliminated.

Note that the conventional FAT-based design has a drawback that the uncertainty estimation is not guaranteed to converge to the actual value which would de-

† Yang Bai is the presenter of this paper.

teriorate the control accuracy and lead to an undesired transient response of the closed loop system. Therefore, in this letter, the design of the update law in the FAT controller follows an immersion and invariance (I&I) approach [22–25] such that not only the state of the control system, but also the error between the uncertainty and its estimation, are steered to zero.

The proposed flocking-based formation control algorithm and the function approximation technique based immersion and invariance (FATII) tracking algorithm are verified under simulations in the ROS/Gazebo programming environment. The simulation framework admits the inclusion of aerial and ground types of mobile robots for testing typical scenarios of monitoring the disaster area.

The rest of the paper is organized as follows. In Section 2 we state the research problem. In Section 3, we develop a control strategy, implement it by corresponding control algorithms, and verify it under simulation in Section 4. Finally, conclusions are drawn in Section 5.

2. FORMALIZATION OF THE CONTROL PROBLEM

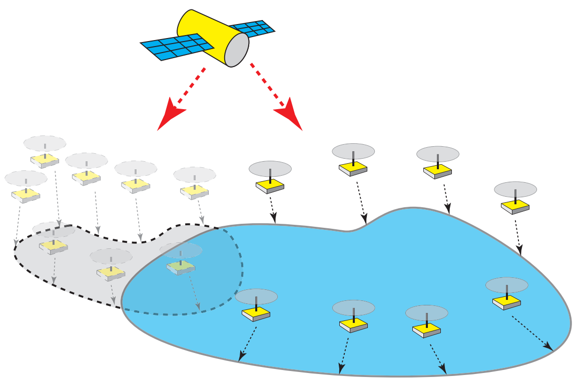

Fig. 1 Statement of the multi-robot tracking problem for expanding flood zone.

This research aims to track the propagation of an expanding flood zone as illustrated in Fig. 1 (the gray and the blue areas respectively represent the flood area before and after the propagation), by using multiple UAVs. It is assumed that each of the UAV's camera can generate high-resolution imagery from a bird's-eye view. The research problem is then stated as constructing tracking algorithm for the UAVs such that the motion of the complete shape of the flood area (its approximation) can be monitored from the emergency center.

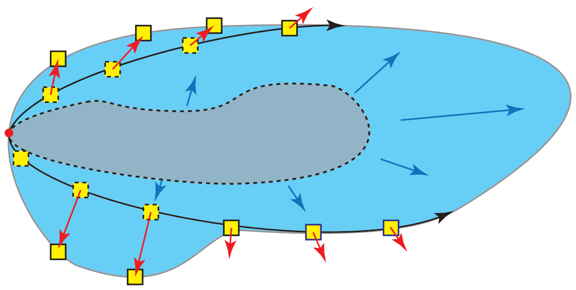

Fig. 2 Illustration of the caging stage.

In the caging stage, two groups of UAVs, referring to the UAVs, move one by one along the edge of the expanding flood zone, as shown in Fig. 2, where the yellow blocks represents the UAVs. They adjust the distances between each other while flying such that eventually they are averagely distributed above the boundary of the flood zone, tracking its propagation.

3. CONTROLLER DESIGN PROCESS

For the design of the control strategy, a kinematic model for the UAVs is adopted, described by

$$\dot{\boldsymbol{q}}^i = \boldsymbol{u}^i, \tag{1}$$

where $\boldsymbol{q}^i$ is the state (horizontal displacements) of a single drone, and $\boldsymbol{u}^i$ is the corresponding velocity controller. Thus the control problem can be stated as constructing the controllers $\boldsymbol{u}^i$ such that the corresponding $\boldsymbol{q}^i$ is consistent with the proposed control strategy. By assuming that the numbers of drones to be N, one has $i \in \{1, 2, ..., N\}$.

The control law $\boldsymbol{u}^i$ for the UAVs needs to realize three basic functions: the first is to move the UAVs along the boundary of the flood zone to complete the caging, the second is to restrict the UAVs on the propagating boundary of the flood zone, and the third is to separate the UAVs by certain distance robustly among each other. Correspondingly, the controller consists of three parts, defined as

$$\boldsymbol{u}^i = \boldsymbol{u}_c^i + \boldsymbol{u}_v^i + \boldsymbol{u}_r^i, \tag{2}$$

where $\boldsymbol{u}_v^i$, $\boldsymbol{u}_r^i$ are respectively the boundary tracking and the separating controllers, and $\boldsymbol{u}_c^i$ is the velocity assigned to each UAV before the encircling of the flood zone completes. The boundary tracking controller $\boldsymbol{u}_v^i$ is constructed by a vision-based approach to achieve real-time autonomous steering of a UAV along the edge of a flood zone. The separating controller $\boldsymbol{u}_r^i$ is designed by a potential field method that keeps adjacent UAVs in certain distance. The constructions of $\boldsymbol{u}_v^i$ and $\boldsymbol{u}_r^i$ are shown specifically as follows.

3.1. Formation control among the UAVs

In the caging problem, adjacent robots are required to keep a specific distance and attempt to separate if the distance is too small. On the other hand, as the robot communication range is limited, the interaction between neighbors will disappear when their distance is larger than the communication range.

Based on these requirements, the separating controller $\boldsymbol{u}_r^i$ is defined as

$$\boldsymbol{u}_r^i = \begin{cases} \sum_{\substack{h=1 \\ h\neq i}}^{N} \Big(1 - \beta(\|\boldsymbol{q}^i - \boldsymbol{q}^h\|)\Big)\boldsymbol{v}^i, & \|\boldsymbol{q}^i - \boldsymbol{q}^h\| \le d \\ \boldsymbol{0}, & \|\boldsymbol{q}^i - \boldsymbol{q}^h\| > d \end{cases} \tag{3}$$

where $\beta(.)$ is a 4th order Beta function expressed by

$$\beta = \frac{35}{d^4}\|\boldsymbol{q}^i - \boldsymbol{q}^h\|^4 - \frac{84}{d^5}\|\boldsymbol{q}^i - \boldsymbol{q}^h\|^5 + \frac{70}{d^6}\|\boldsymbol{q}^i - \boldsymbol{q}^h\|^6 - \frac{20}{d^7}\|\boldsymbol{q}^i - \boldsymbol{q}^h\|^7, \tag{4}$$

and $\boldsymbol{v}^i$ is a constant vector that weights β. By selecting β as (4), when the distance between UAVs i and h is smaller than a specific value d, UAV i moves away from UAV h. When the distance is larger than d, the magnitude of the separating controller $\boldsymbol{u}_r^i$ becomes zero.

3.2. Vision-based boundary tracking for a single UAV

The tracking problem can be stated as designing the velocity controller $\boldsymbol{u}_v^i$ for a UAV such that a portion of the edge for the flood zone is always within the UAV's field of vision as illustrated in Fig. 3, where the blue area represents the flood zone and the black squared frame for the UAV's field of vision. The segmentation problem has been addressed in [26–28].

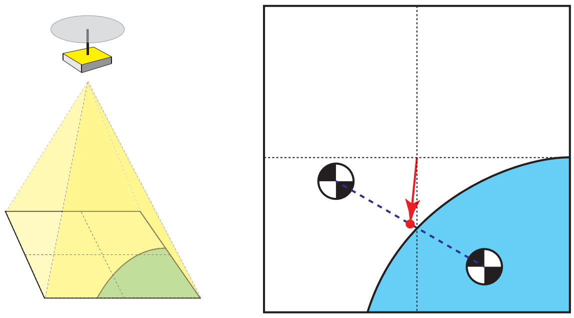

Fig. 3 Tracking problem for a single UAV.

The vision-based controller aims to keep the middle point of the mass centers for both the land and the water area on the geometric center of the filed of vision for the UAV. For this purpose, the vision-based controller $\boldsymbol{u}_v^i$ is constructed as follows.

Define $\boldsymbol{p}^i$ to be the absolute position vector (defined in the global coordinate frame) for the middle point of the centers of mass for both the land the flood regions, and $\boldsymbol{e}^i = \boldsymbol{q}^i - \boldsymbol{p}^i$ to be the error between a single robot and the water propagation. The corresponding error dynamics for the inner leader can be formulated as

$$\dot{\boldsymbol{e}}^i = \boldsymbol{u}_v^i - \dot{\boldsymbol{p}}^i + \boldsymbol{\delta}^i, \tag{5}$$

where $\boldsymbol{\delta}^i$ denotes the system uncertainties such as wind effects. Thus, the tracking control problem is stated as constructing $\boldsymbol{u}_v^i$ such that $\lim\limits_{t\to\infty} \boldsymbol{e}^i = \boldsymbol{0}$, with $\boldsymbol{\delta}^i$ unknown.

To tackle the stated problem, the controller design is under a function approximate technique based framework [17–19]. We utilize the weighted basis functions to approximate $\boldsymbol{\delta}$ in the control system (5) at each time instant as

$$\boldsymbol{\delta}^i(t) = \sum_{k=0}^{K} \boldsymbol{\delta}_k^i \psi_k^i(\boldsymbol{e}^i, t), \tag{6}$$

where $\boldsymbol{\delta}_k^i$ is constant and ψ_k^i consists of $\boldsymbol{e}^i$ and t. Several candidates for the basis function ψ_k^i can be selected to approximate the nonlinear functions. In this paper, we select the Fourier series [17] as the basis functions.

Note that the control problem requires the unknown parameters $\boldsymbol{\delta}_k^i$ in (6) to be identified. For this purpose, these parameters $\boldsymbol{\delta}_k^i$ at each time instant t are estimated by $\hat{\boldsymbol{\delta}}_k^i(t)$. In conventional adaptive control algorithms, $\hat{\boldsymbol{\delta}}_k^i(t)$ is not guaranteed to converge to the actual value. The error between the actual parameters and parameter estimation could deteriorate the control accuracy and lead to an undesired transient response of the control system.

For the convergence of both the system states and the uncertainty estimations, we develop an FATII control method, which is outlined as follows. Define in the extended space $(\boldsymbol{e}^i, \hat{\boldsymbol{\delta}}_n^i)$ the manifold

$$\boldsymbol{M}_k^i = \{(\boldsymbol{e}^i, \hat{\boldsymbol{\delta}}_k^i) \in \boldsymbol{R}^2 \mid \boldsymbol{\delta}_k^i - \hat{\boldsymbol{\delta}}_k^i - \boldsymbol{\xi}_k^i = \boldsymbol{0}\}, \tag{7}$$

where $\boldsymbol{\xi}_k^i$ is a continuous function to be specified [22]. The motivation for this definition is described as follows. By substituting (6) into (5), the dynamics of the system restricted to the manifold $\boldsymbol{M}_k^i$ (provided it is invariant) is described by

$$\dot{\boldsymbol{e}}^i = \boldsymbol{u}_v^i - \dot{\boldsymbol{p}}^i + \sum_{k=0}^{K} (\hat{\boldsymbol{\delta}}_k^i + \boldsymbol{\xi}_k^i)\psi_k^i. \tag{8}$$

Note from (8), the unknown vector $\boldsymbol{\delta}_k^i$ is excluded from the expression of $\dot{\boldsymbol{e}}^i$, which is important in the controller design process since $\boldsymbol{\delta}_k^i$ cannot appear in the control law.

However, (8) is equivalent to (5) only when the system dynamics stay in the manifold $\boldsymbol{M}_k^i$. By defining the off-the-manifold variable

$$\boldsymbol{z}_k^i = \boldsymbol{\delta}_k^i - \hat{\boldsymbol{\delta}}_k^i - \boldsymbol{\xi}_k^i, \tag{9}$$

where $\boldsymbol{z}_k^i \in \boldsymbol{R}^2$ and $\boldsymbol{\xi}_k^i \in \boldsymbol{R}^2$, $\boldsymbol{z}_k^i = \boldsymbol{0}$ implies that the system dynamics stay in the manifold $\boldsymbol{M}_k^i$. With the off-the-manifold variable $\boldsymbol{z}_k^i$, the state equation is transformed to

$$\dot{\boldsymbol{e}}^i = \boldsymbol{u}_v^i - \dot{\boldsymbol{p}}^i + \sum_{k=0}^{K} (\boldsymbol{z}_k^i + \hat{\boldsymbol{\delta}}_k^i + \boldsymbol{\xi}_k^i)\psi_k^i, \tag{10}$$

where $\sum_{k=0}^{K} \boldsymbol{z}_k^i \psi_k$ represents the estimation error of the system uncertainty.

For the construction of $\boldsymbol{u}_v^i$, the Lyapunov candidate function can be formulated as

$$V^i = \frac{1}{2}\left((\boldsymbol{e}^i)^\top \boldsymbol{e}^i + \sum_{k=0}^{K} (\boldsymbol{z}_k^i)^\top \boldsymbol{z}_k^i\right), \tag{11}$$

the derivative of which is calculated as

$$\begin{aligned}\dot{V}^i = {} & (\boldsymbol{e}^i)^\top \Big(\boldsymbol{u}_v^i - \dot{\boldsymbol{p}}^i + \big(\boldsymbol{Z}^i + \hat{\boldsymbol{\delta}}^i + \boldsymbol{\Xi}^i\big)\boldsymbol{\psi}^i\Big) \\ & + \sum_{k=0}^{K} (\boldsymbol{z}_k^i)^\top \Big(- \dot{\hat{\boldsymbol{\delta}}}_k^i - \frac{\partial \boldsymbol{\xi}_k^i}{\partial t} \\ & - \frac{\partial \boldsymbol{\xi}_k^i}{\partial \boldsymbol{e}^i}\Big(\boldsymbol{u}_v^i - \dot{\boldsymbol{p}}^i + \big(\boldsymbol{Z}^i + \hat{\boldsymbol{\delta}}^i + \boldsymbol{\Xi}^i\big)\boldsymbol{\psi}^i\Big)\Big),\end{aligned} \tag{12}$$

where $\boldsymbol{Z}^i = [\boldsymbol{z}_0^i \mid \boldsymbol{z}_1^i \mid ... \mid \boldsymbol{z}_K^i]$, $\hat{\boldsymbol{D}}^i = [\hat{\boldsymbol{d}}_0^i \mid \hat{\boldsymbol{d}}_1^i \mid ... \mid \hat{\boldsymbol{d}}_K^i]$, $\boldsymbol{\Xi}^i = [\boldsymbol{\xi}_0^i \mid \boldsymbol{\xi}_1^i \mid ... \mid \boldsymbol{\xi}_K^i]$, and $\boldsymbol{\psi}^i = [\psi_0^i \quad \psi_1^i \quad ... \quad \psi_K^i]^\top$.

The design of the auxiliary controller $\boldsymbol{u}_l$ requires to render the derivative of the Lyapunov candidate function

(12) negative semi-definite. For this purpose, the following FATII method is proposed. The controller $\boldsymbol{u}_l$ is constructed as

$$\begin{aligned}\boldsymbol{u}_v^i &= -\boldsymbol{K}\left(\boldsymbol{q}^i - \boldsymbol{p}^i\right) + \dot{\boldsymbol{p}}^i \\ &\quad + \sum_{k=0}^{K}\left(\hat{\boldsymbol{\delta}}_k^i + \left(\boldsymbol{q}^i - \boldsymbol{p}^i\right)\right)\psi_k^i, \end{aligned} \tag{13}$$

where

$$\dot{\hat{\boldsymbol{\delta}}}_k^i = -\left(\boldsymbol{q}^i - \boldsymbol{p}^i\right)\dot{\psi}_k^i + \boldsymbol{K}\boldsymbol{\xi}_k^i, \tag{14}$$

and $\boldsymbol{\xi}_k^i = (\boldsymbol{q}^i - \boldsymbol{p}^i)\psi_k^i$, $\boldsymbol{K}$ is a positive definite matrix, such that the error between $\boldsymbol{q}^i$ and the center of mass for the polygon asymptotically converges to zero.

Theorem 1: The closed loop system, formulated by the plant (8) and the controller (13), is asymptotically stable.

Proof: After substituting (8) and (13) into the Lyapunov function candidate (11), one calculates the derivative of the Lyapunov candidate function as

$$\begin{aligned}\dot{V}^i &= -(\boldsymbol{e}^i)^\top\left(\boldsymbol{K}\boldsymbol{e}^i - \sum_{k=0}^{K}\boldsymbol{z}_k^i\psi_k^i\right) \\ &\quad - \left(\sum_{k=0}^{K}\boldsymbol{z}_k^i\psi_k^i\right)^\top\left(\sum_{k=0}^{K}\boldsymbol{z}_k^i\psi_k^i\right) \\ &= -(\boldsymbol{e}^i)^\top\boldsymbol{K}\boldsymbol{e}^i + \frac{1}{4}(\boldsymbol{e}^i)^\top\boldsymbol{e}^i - \frac{1}{4}(\boldsymbol{e}^i)^\top\boldsymbol{e}^i \\ &\quad + (\boldsymbol{e}^i)^\top\left(\sum_{k=0}^{K}\boldsymbol{z}_k^i\psi_k^i\right) \\ &\quad - \left(\sum_{k=0}^{K}\boldsymbol{z}_k^i\psi_k^i\right)^\top\left(\sum_{k=0}^{K}\boldsymbol{z}_k^i\psi_k^i\right) \\ &\leq -\left(\lambda_{\min}(\boldsymbol{K}) - \frac{1}{4}\right)\|\boldsymbol{e}^i\|_2^2 - \left\|\sum_{k=0}^{K}\boldsymbol{z}_k^i\psi_k^i - \frac{1}{2}\boldsymbol{e}^i\right\|_2^2.\end{aligned}$$

The selection of $\lambda_{\min}(\boldsymbol{K}) > \frac{1}{4}$ renders the derivative of the Lyapunov candidate function $\dot{V}^i$ negative semi-definite. ■

By using the Barbalat's lemma, one obtains

$$\lim_{t\to\infty}\boldsymbol{e}^i = \boldsymbol{0},\ \lim_{t\to\infty}\sum_{i=0}^{N}\boldsymbol{z}_k^i\psi_k^i = \boldsymbol{0}, \tag{15}$$

implying that both the state and the uncertainty estimation converge to the desired and the actual uncertainty respectively.

3.3. Combined control algorithm for each UAV

The control strategy for caging and tracking an expanding flood zone can be implemented by the combination the proposed flocking-based formation control and the FATII vision-based tracking algorithms. Note that the tracking algorithm works only when both the water and the land regions appear in the field of vision of a single UAV. When only the land or the water region appears, a UAV would rise such that its field of vision is enlarged. It keeps increasing its height until both the land and the flood regions return to its field of vision.

Thus the combined control algorithm can be summarized as

Algorithm 1

Input: $\boldsymbol{u}^i$
Output: $\boldsymbol{q}^i$

Initialization: $\boldsymbol{u}^i = \boldsymbol{q}^i = \boldsymbol{0}$, obtain initial image from the UAV's camera
Repeat:
if Both the land and the flood regions are in the image **then**
Compute the position vector $\boldsymbol{p}^i$ from the image
Update the controller $\boldsymbol{u}_v^i$ by using (13)
Update the state vector $\boldsymbol{q}^i$ by solving (1), with the selection of $\boldsymbol{u}_r^i$ as (3), and $\boldsymbol{u}_c^i$ as a tangent vector to the boundary of the flood zone
Obtain new image from the UAV's camera
else
Increase the height of the UAV
end if
End

4. CASE STUDY

For the verification of the control strategy, simulations are conducted under the ROS/Gazebo programming environment. Multiple UAVs are modeled with the use of Hector quadrotor package [29], which is a collection of ROS stacks that supply several tools to simulate and interact with the robots, and its extension to multiple quadrotors [30].

Note that as there exists no model in Gazebo for the expandable disaster area, in our simulation, it is implemented with the use of Gazebo animated models (by the Actor plugin), which are limited to modeling of objects of fixed size and shape. Specifically, the flood area is created by a group of cylinders (shown in blue color) hidden under the ground. They rise up successively from the one with the minimum radius to the largest one, to generate the visual effect of a dynamic deformable circular area. The UAVs are expected to cage the expanding circular flood area, track its propagation, and cover its interior region.

The simulation is based on the kinematic model (1) and the controller $\boldsymbol{u}^i$. In the simulation, the flood zone expands from a circle of radius 12m to 16m, and 8 UAVs are utilized for the tracking problem. The simulation lasts for 100 seconds. The numerical values adopted in the simulation are $\boldsymbol{K} = \text{diag}frac1300, \frac{1}{85}$, and the encircling speed of the UAVs is selected as 1m/s. The maximum distance d in the Beta-function (4) is chosen to be 10m.

The camera image of a UAV is shown in Fig. 4. The grey and the blue portions respectively represent the land and the flood region, the blue and red dots are their CMs, and the black dot denotes the middle point of the two

CMs. According to the proposed vision-based boundary tracking algorithm 3.3, the black dot follows the motion of the geometric center of the camera image such that the UAV tracks the propagation of the flood zone.

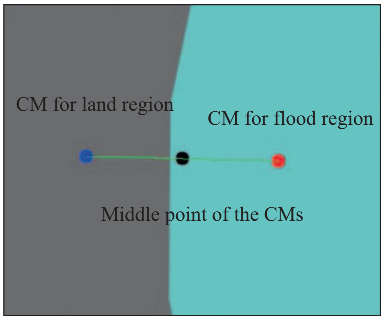

Fig. 4 Camera image of a UAV in the caging stage.

It can be seen from Fig. 5 that the UAVs marked in the yellow color, adaptively distribute themselves to cage and track the expansion of the flood zone.

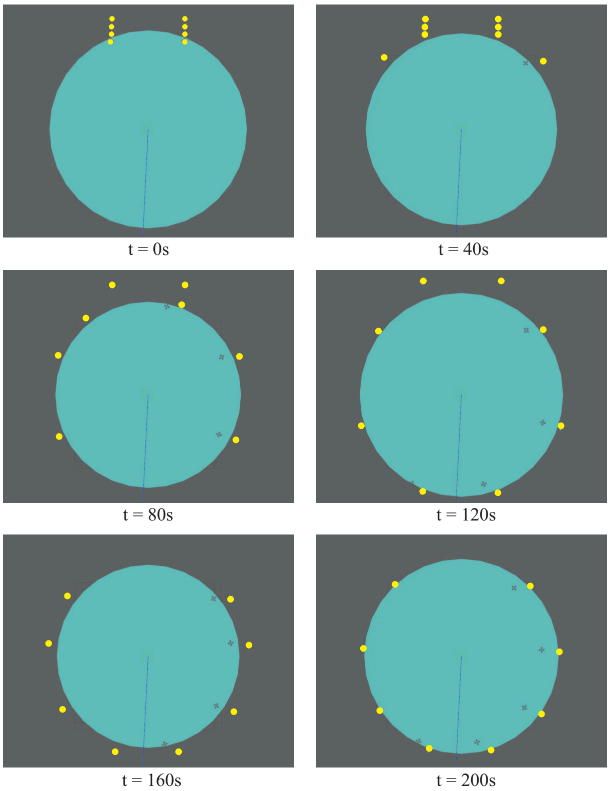

Fig. 5 A group of UAVs (yellow) cage and track a circular expanding flood area (the grey cross-shaped objects are the shadows of the UAVs).

5. CONCLUSIONS

In this paper, a control strategy has been proposed for tracking the propagation of an expanding flood zone by using a group of UAVs. The proposed control strategy has addressed the following two issues. One is to averagely distribute a group of UAVs along the expanding boundary of the flood zone without collisions among each other. The other is to track the propagation of the flood region by keeping its boundary within the field of vision for each UAV. These two problems are tackled by developing corresponding flocking-based control and adaptive vision-based tracking algorithms for the aforementioned problems to implement the control strategy. The validity of the control strategy has be tested under simulations under the ROS/Gazebo environment.

Currently, the motion of the flood is implemented with the use of Gazebo animated models (by the Actor plug-in), which is limited to the modeling of objects of fixed size and shape. The use of Actors for modeling of dynamically changing objects (as implemented in our simulator) is not convenient as it requires the introduction of many dummy objects. In the future work, we plan to integrate Gazebo with the cross-platform game engine Unity where animating a flood area is easier and more realistic [31].

In addition, a higher level programming script for population Gazebo with a swarm of robots would be desirable. Specifically, when the Gazebo program is launched, the indices of the UAVs are assigned randomly. One needs to manually adjust these indices to link each UAV to the corresponding controller. This process becomes inconvenient when the number of the UAVs increases. To deal with this problem, we plan to develop a higher level user interface for populating the Gazebo with multiple robots.

The experiments will also be conducted for the further validation of our approach. Different from the simulations, distinguishing the land and flood regions from the camera image needs to be implemented by a segmentation technique, which will be addressed in the future work.

ACKNOWLEDGEMENTS

This research was supported, in part, by the Japan Science and Technology Agency, the JST Strategic International Collaborative Research Program, Project No. 18065977, and by the Russian Foundation for Basic Research, Project No. 19-58-70002.

REFERENCES

[1] S. Tadokoro, *Disaster Robotics: Results from the ImPACT Tough Robotics Challenge*. Springer, 2019.

[2] R. Murphy, *Disaster Robotics*. The MIT Press, 2014.

[3] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[4] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[5] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *International Journal of Robust and Non-*

linear Control, vol. 17, no. 10–11, pp. 1002–1033, 2007.

[6] W. Dong and J. A. Farrell, "Cooperative control of multiple nonholonomic mobile agents," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1434–1448, 2008.

[7] L. Krick, M. E. Broucke, and B. A. Francis, "Stabilization of infinitesimally rigid formations of multi-robot networks," in *Proc. 47th IEEE Conference on Decision and Control (CDC)*, Cancun, Mexico, Dec. 9–11 2008, pp. 477–482.

[8] M. Cao, C. Yu, and B. D. O. Anderson, "Formation control using range-only measurements," *Automatica*, vol. 47, no. 4, pp. 776–781, 2011.

[9] K. Oh and H. Ahn, "Formation control of mobile agents based on interagent distance dynamics," *Automatica*, vol. 47, no. 10, pp. 2036–2312, 2011.

[10] ——, "Distance-based undirected formations of single and double integrator modeled agents in n-dimensional space," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 12, pp. 1809–1820, 2012.

[11] C. W. Reynolds, "Flocks, herds, and schools: a distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[12] N. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. 40th IEEE Conference on Decision and Control (CDC)*, Orlando, FL, USA, Dec. 4–7 2008, pp. 2968–2973.

[13] V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 692–697, 2003.

[14] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 863–868, 2007.

[15] H. Sakai, D. andFukushima and F. Matsuno, "Flocking for multirobots without distinguishing robots and obstacles," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1019–1027, 2017.

[16] K. Oh, M. Park, and H. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[17] A.-C. Huang and Y.-S. Kuo, "Sliding control of non-linear systems containing time-varying uncertainties with unknown bounds," *International Journal of Control*, vol. 74, no. 3, pp. 252–264, 2001.

[18] A.-C. Huang and Y.-C. Chen, "Adaptive sliding control for single-link flexible-joint robot with mismatched uncertainties," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 5, pp. 770–775, 2004.

[19] H.-Y. Chen and S.-J. Huang, "A new model-free adaptive sliding controller for active suspension system," *International Journal of Systems Science*, vol. 39, no. 1, pp. 57–69, 2008.

[20] Y. Bai, M. Svinin, and M. Yamamoto, "Adaptive trajectory tracking control for the ball-pendulum system with time-varying uncertainties," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 2083–2090.

[21] ——, "Function approximation based control for non-square systems," *SICE Journal of Control, Measurement, and System Integration*, vol. 11, no. 6, pp. 477–485, 2018.

[22] A. Astolfi and R. Ortega, "Immersion and invariance: a new tool for stabilization and adaptive control of nonlinear systems," *IEEE Transactions on Automatic control*, vol. 48, no. 4, pp. 590–606, 2003.

[23] B. Zhao, B. Xian, Y. Zhang, and X. Zhang, "Nonlinear robust sliding mode control of a quadrotor unmanned aerial vehicle based on immersion and invariance method," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 18, pp. 3714–3731, 2015.

[24] J. Á. Acosta, R. Ortega, A. Astolfi, and I. Sarras, "A constructive solution for stabilization via immersion and invariance: The cart and pendulum system," *Automatica*, vol. 44, no. 9, pp. 2352–2357, 2008.

[25] Y. Bai, Y. Wang, M. Svinin, E. Magid, and R. Sun, "Function approximation technique based immersion and invariance control for unknown nonlinear systems," *IEEE Control Systems Letters*, vol. 4, no. 4, pp. 934–939, 2020.

[26] Y. Ma, J. Koscka, and S. Sastry, "Vision guided navigation for a nonholonomic mobile robot," *IEEE Transactions on Automatic Control*, vol. 15, no. 3, pp. 521–536, 1999.

[27] D. Martin, C. Fowlkes, and M. J., "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[28] A. Xu and G. Dudek, "A vision-based boundary following framework for aerial vehicles," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, Oct. 18–22 2010, pp. 81–86.

[29] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk, *Comprehensive simulation of quadrotor UAVs using ROS and Gazebo*. Springer, Berlin, Heidelberg, 2012.

[30] C. Kolon, *Stability of Nonlinear Swarms on Flat and Curved Surfaces*, ser. Trident Scholar project report. U.S. Naval Academy, 2018.

[31] A. Hussein, F. Garcia, and C. Olaverri-Monreal, "Ros and unity based framework for intelligent vehicles control and simulation," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Sep. 2018, pp. 1–6.