2020 17th International Conference on Ubiquitous Robots (UR)
June 22-26, 2020. Kyoto, Japan

Cooperative Multi-Robot Control for Monitoring an Expanding Flood Area

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

Abstract— **In this paper, a control strategy is developed for tracking the propagation of an expanding flood zone by using a group of unmanned aerial vehicles (UAVs). The strategy consists of two stages: a caging stage and a covering stage. In the caging stage, a group of UAVs, referring to the boundary drones, are averagely distributed along the boundary of the flood zone, tracking its propagation. In the covering stage, another group of UAVs, referring to the inner drones, are allocated among the interior region of the flood zone, covering the region as much as possible with less overlapping of the UAVs' field of view. Corresponding control algorithms are proposed for the aforementioned types of UAVs to implement the control strategy. The feasibility of the control strategy is verified under simulations.**

I. 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, tracked, and covered, which constitutes the goal of our research.

To tackle the problem, we propose a strategy combined in two stages, a caging stage and a covering stage. The caging stage averagely distributes a group of UAVs along the boundary of the flood zone, tracking its propagation. The covering stage adaptively allocates another group of UAVs among the interior area of the flood zone, covering it as much as possible with little overlapping of each UAV's field of view. Note that in both stages, a formation control problem is raised which requires a flock of UAVs moving cohesively such that a single UAV can communicate with at least one neighbor while keeping a distance from each other for the avoidance of collision.

In the literature, methods for the formation control problem can be classified into the following main types: the position-based control, the displacement-based control, the distance-based control, and the flocking-based control. In the position-based control [3]–[5], each agent knows its absolute position with respect to a global coordinate system. The desired formation of multiple agents is achieved by tracking the position of each agent without any interactions among the agents under ideal conditions. Different from the position-based control, in the displacement-based control [6]–[8], it is assumed that some of the agents cannot sense their absolute positions. Instead, they know the relative positions of, or displacement from, their neighbors with respect to a global coordinate system. By controlling these distances, a desired formation of multiple agents can be achieved.

Note that in our strategy, the utilization of the absolute positions for the UAVs should be avoided due to the assumption that only a few UAVs can get access to the GPS. Therefore, the aforementioned methods are not feasible in our construction of control algorithms. Instead of using the absolute positions, in the distance-based formation control [9]–[12], agents can sense the relative positions of their neighboring agents with respect to their local coordinate systems. They actively control inter-agent distances to achieve their desired formation, which is specified by the desired values for distances between any pair of agents. Compared with the position-based and displacement-based control, the distance-based control requires less global information, and is thus, suitable for the case when limited number of UAVs having access to the GPS.

However, our control strategy requires to adaptively cover 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 method [13] is adopted in our strategy, which automatically adjusts the distance between adjacent UAVs.

The flocking control method 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). As extensions of [13], many control laws [14]–[16] have been proposed in the literature to achieve collective behaviors. In these works, the cohesion and the separation rules have been usually implemented by means of a potential function

[1]Y. Bai, K. Asami and M. Svinin are with the Information Science and Engineering Department, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga 525-8577, Japan. yangbai@fc.ritsumei.ac.jp, is0392hr@ed.ritsumei.ac.jp, svinin@fc.ritsumei.ac.jp

[2]E. Magid is with the Department of Intelligent Robotics, Kazan Federal University, Kremlyovskaya str. 35, Kazan 420008, Russian Federation. magid@it.kfu.ru

978-1-7281-5714-6/20/$31.00 ©2020 IEEE

of inter-agent distances, and the alignment rule has been implemented by means of velocity consensus of agents. Detailed reviews of the formation control techniques can be found in [17]–[19].

Based on the flocking method, we develop control algorithms for both the caging and covering stages such that UAVs can adaptively adjust the distance between each other while tracking the targets. The verification of the control algorithms is conducted by 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 II we state the research problem. In Section III, we develop a control strategy, implement it by corresponding control algorithms, and verify it under simulation in Section IV. Finally, conclusions are drawn in Section V.

II. Formalization of the Research Problem

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 a group of UAVs with only some of them having access to GPS while others can only sense their neighbors within limited distances. Each of the UAV is holonomic and omnidirectional, and its camera can generate high-resolution imagery from a bird's-eye view. It is assumed that the flying height and the field of view for each UAV are limited, so that a single UAV cannot monitor the whole flood region. The research problem is then stated as constructing tracking algorithm for multiple UAVs such that the motion of the complete shape of the flood area (its approximation) can be monitored from the emergency center.

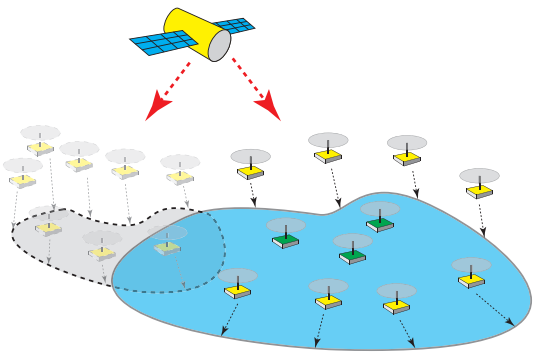

Fig. 1. Statement of the multi-robot tracking problem for expanding flood zone.

The construction of the tracking algorithm is combined in two stages, a caging stage and a covering stage. These two stages are conducted simultaneously. The UAVs are divided into three groups, starting from the same location. Two groups fly along the boundary of the flood zone (it is assumed that there is one unique continuous boundary), undergoing the caging stage, while the other one goes though the interior area, undergoing the covering stage.

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

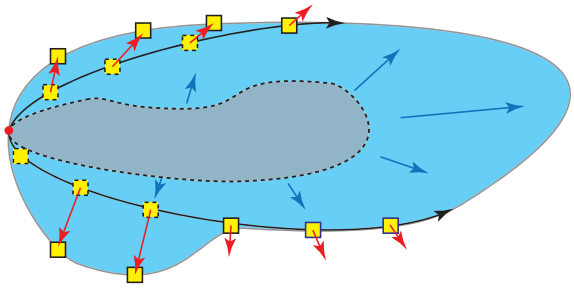

Fig. 2. Illustration of the caging stage.

While the caging stage requires to track the boundary of the propagating flood area, the covering stage aims to allocate the UAVs adaptively above the interior area such that the flood zone can be monitored within the field of view for the UAVs. The flock of UAVs flying above the interior area consists of two types: one has access to GPS, referring to the inner leader, and the other has not, referring to the inner follower. As illustrated in Fig. 3, the red and the green blocks respectively represent the inner leader and the inner followers. After a certain time, the inner followers would drop from the flock and adjust their position automatically. Note that the flood area is expected to be covered as much as possible with less overlapping of the field of view for each UAV.

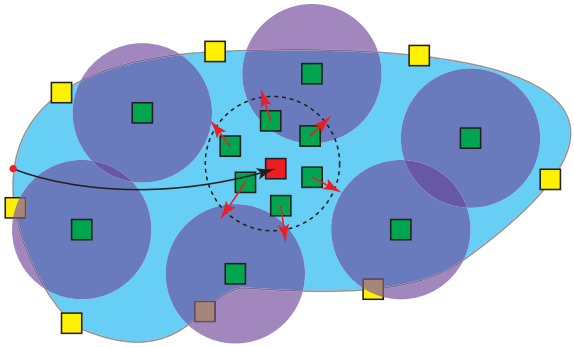

Fig. 3. Illustration of the covering stage.

III. Controller Design Process

In this section, we develop controllers for the three type of UAVs in the cooperative tracking problem. The control strategy described in the previous section for the tracking problem is summarized as follows. All UAVs in the research problem can be classified into three types: the boundary drones (yellow blocks), the inner leaders (red blocks), and the inner followers (green blocks). The boundary drones require

to be averagely distributed along the boundary of the flood zone while tracking the propagation of it. The inner leader is supposed to be located in the central area of the flood zone to cooperate with the boundary drones effectively. The inner followers need to be allocated adaptively among the region between the inner leader and the boundary drones, covering this region as much as possible with less overlapping of field of view for each UAV.

To realize the control strategy, a kinematic model for the UAVs is adopted, described by

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

$$\dot{\boldsymbol{q}}_l = \boldsymbol{u}_l, \tag{2}$$

$$\dot{\boldsymbol{q}}_f^j = \boldsymbol{u}_f^j, \tag{3}$$

where $\boldsymbol{q}_b^i$, $\boldsymbol{q}_l$, $\boldsymbol{q}_f^j \in \mathbb{R}^2$ are respectively the states (horizontal displacements) of a single boundary drone, the inner leader, a single inner follower, and $\boldsymbol{u}_b^i$, $\boldsymbol{u}_l$, $\boldsymbol{u}_f^j \in \mathbb{R}^2$ are the corresponding velocity controllers. Thus the control problem can be stated as constructing the controllers $\boldsymbol{u}_b^i$, $\boldsymbol{u}_l$, $\boldsymbol{u}_f^j$ such that the corresponding $\boldsymbol{q}_b^i$, $\boldsymbol{q}_l$, $\boldsymbol{q}_f^j$ are consistent with the proposed control strategy. By assuming that the numbers of boundary drones and inner followers are respectively N and M, one has $i \in \{1, 2, ..., N\}$ and $j \in \{1, 2, ..., M\}$.

A. Controller design for the boundary drones

The control law $\boldsymbol{u}_b^i$ for the boundary drones 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

$$\boldsymbol{u}_b^i = \boldsymbol{u}_v^i + \boldsymbol{u}_r^i + \boldsymbol{u}_c^i, \tag{4}$$

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 boundary drone 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 boundary drone 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.

1) Boundary tracking for a single UAV: The tracking problem can be stated as designing the velocity controller $\boldsymbol{u}_v^i$ for a boundary drone such that a portion of the edge for the flood zone is always within the UAV's field of view as illustrated in Fig. 4, where the blue area represents the flood zone and the black squared frame for the UAV's field of view. The segmentation problem has been addressed in [20]–[22].

The vision-based controller aims to keep the midpoint 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

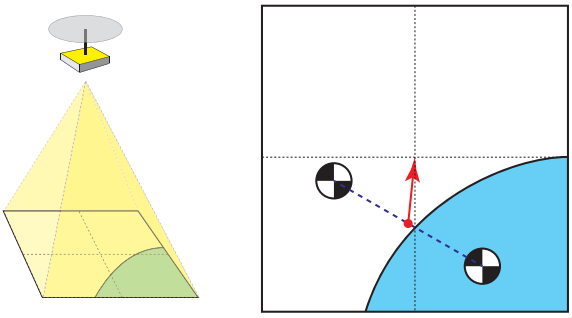

Fig. 4. Tracking problem for a single UAV.

$$\boldsymbol{u}_v^i = -\boldsymbol{K}_b\left(\boldsymbol{q}_b^i - \boldsymbol{p}^i\right) + \dot{\boldsymbol{p}}^i, \tag{5}$$

where $\boldsymbol{K}_b$ is a positive definite matrix, and $\boldsymbol{p}^i$ stands for the absolute position vector (defined in the global coordinate frame) for the midpoint of the centers of mass for both the land the flood regions. The proposed controller steers the error between $\boldsymbol{q}_b^i$ and $\boldsymbol{p}^i$ to zero, implying the UAV tracks the boundary of the flood zone.

Note that this tracking algorithm works only when both the water and the land regions appear in the field of view of a single UAV. When only the land or the water region appears, a UAV would rise such that its field of view is enlarged. It keeps increasing its height until both the land and the flood regions return to its field of view. The proposed vision-based control algorithm is summarized as

Algorithm 1 Vision-based control

Input: $\boldsymbol{u}_v^i$

Output: $\boldsymbol{q}_b^i$

Initialization: $\boldsymbol{u}_v^i = \boldsymbol{q}_b^i = \boldsymbol{0}$, 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 (5)
    Update the state vector $\boldsymbol{q}_b^i$ by solving (1), with the selection of $\boldsymbol{u}_r^i$ as (6), 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

2) Separating control among adjacent UAVs: In the caging stage, 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

$$\boldsymbol{u}_r^i = \sum_{\substack{h=1 \\ h\neq i}}^{N} \beta(\|\boldsymbol{q}_b^i - \boldsymbol{q}_b^h\|)\boldsymbol{v}^i, \tag{6}$$

where β is a 4th order Beta function expressed by

$$\beta = 1 - \frac{35}{d^4}\|\boldsymbol{q}_b^i - \boldsymbol{q}_b^h\|^4 + \frac{84}{d^5}\|\boldsymbol{q}_b^i - \boldsymbol{q}_b^h\|^5 - \frac{70}{d^6}\|\boldsymbol{q}_b^i - \boldsymbol{q}_b^h\|^6 + \frac{20}{d^7}\|\boldsymbol{q}_b^i - \boldsymbol{q}_b^h\|^7, \tag{7}$$

and $\boldsymbol{v}^i$ is a constant vector that weights β. By selecting β as (7), when the distance between UAVs i and h is smaller than a specific value d, β is negative and UAV i moves away from UAV h. When the distance is larger than d, β becomes zero and so is the magnitude of the separating controller $\boldsymbol{u}_r^i$.

By substituting the vision-based controller (5) and the separating controller (6) into (4), one obtains the control law for a boundary drone as

$$\boldsymbol{u}_b^i = -\boldsymbol{K}_b\left(\boldsymbol{q}_b^i - \boldsymbol{p}^i\right) + \dot{\boldsymbol{p}}^i + \sum_{h=1}^{N-1} \beta(\|\boldsymbol{q}_b^i - \boldsymbol{q}_b^h\|)\boldsymbol{v}^i + \boldsymbol{u}_c^i. \tag{8}$$

Under the proposed controller, the boundary drones can track the propagation of the flood zone while at the same time, keeping a safe distance between adjacent agents.

B. Controller design for the inner drones

The covering stage requires a group of UAVs to adaptively cover the flooding area. This problem can be separated into two parts: position control for the inner leader and flocking control for the followers. It is assumed that only the leaders have access to the GPS while the followers can communicate with and know the relative positions of their neighbors. This enables the data for the position and range of the propagating flood zone to be known and sent to the rescue center.

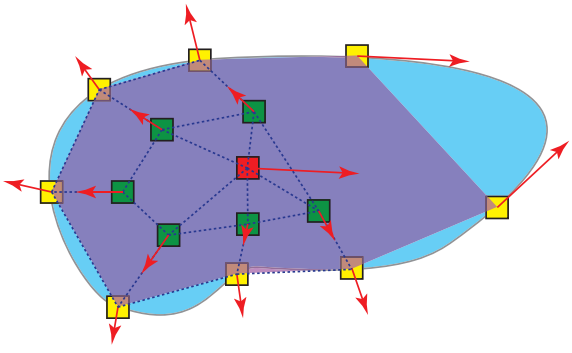

Fig. 5. Control strategy for the covering stage.

1) Position control for the inner leader: The inner leader is predefined and supposed to be located in the central area of the flood zone to cooperate with the boundary drones effectively. Specifically, connecting the boundary drones formulates a polygon and the inner leader is located at the center of mass of the polygon. Therefore, the control law $\boldsymbol{u}_l^i$ for the inner leader is defined as follows.

The center of mass for the polygon formulated by the boundary drones can be calculates as $\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{q}_b^i$. Thus, the controller $\boldsymbol{u}_l$ is constructed as

$$\boldsymbol{u}_l = -\boldsymbol{K}_l\left(\boldsymbol{q}_l - \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{q}_b^i\right) + \frac{1}{N}\sum_{i=1}^{N}\dot{\boldsymbol{q}}_b^i, \tag{9}$$

where $\boldsymbol{K}_l$ is a positive definite matrix, such that the error between $\boldsymbol{q}_l$ and the center of mass for the polygon asymptotically converges to zero.

2) Adaptive flocking control for the inner followers: To adaptively allocate a group of inner followers among the flood region, the design of the controller $\boldsymbol{u}_f^j$ for each inner follower is based on a flocking approach. It relies on relatively simple interactions among individuals, following two basic rules: cohesion (stay close to nearby neighbors) and separation (avoid collisions with nearby neighbors). Consequently, the controller is formulated by two main portions, a cohesive controller between each inner follower and the corresponding boundary drone, and a separating controller among the inner followers themselves.

Note that the flocking controller for the inner followers is functionally similar to that for the boundary drones. The difference is in the vision-based controller in (8). For the inner followers, the vision-based controller is replaced by a position tracking controller for which the tracking targets are the boundary drones. Therefore, the flocking controller for the inner followers takes the same form of (8), as

$$\boldsymbol{u}_f^j = -\boldsymbol{K}_f\left(\boldsymbol{q}_f^j - \boldsymbol{q}_b^i\right) + \dot{\boldsymbol{q}}_b^i + \sum_{\substack{k=1 \\ k\neq j}}^{M} \beta(\|\boldsymbol{q}_f^j - \boldsymbol{q}_f^k\|)\boldsymbol{\omega}^j, \tag{10}$$

where $\boldsymbol{K}_f$ is a positive definite matrix and the constant vector $\boldsymbol{\omega}^j$ is the weight for the beta function. Under the proposed controller (10), the inner followers can track the motion of the corresponding boundary drones while adaptively covering the region between the boundary drones and the inner leader.

Note that compared with the boundary drones, the followers fly around the inner ring of the flood zone. Assuming that all the UAVs have the same communication distance, the number of the inner followers M can be less than that of the boundary drones N to maintain communication between adjacent UAVs. In that case, i and j have no one-to-one correspondence. To implement the controller (10) when $M < N$, one can select, for instance, $i = nj$ where n is a integer larger than 1.

Also note that as illustrated in Fig. 5, there is only one layer of inner followers between the inner leader and the boundary followers. If one layer of inner followers cannot cover the whole flood area due the limited range of the UAV's field of view, more layers, the state of which is defined by $\boldsymbol{q}_{fm}$ where $m = 1, 2, 3, ...$, will be released from the flock of UAVs around the inner leader, tracking the motion of the outer layer followers. The process is repeated until the whole flood area can be covered. The corresponding controller takes the same form of (10) where $\boldsymbol{q}_f$ and $\boldsymbol{q}_b$ are replaced by $\boldsymbol{q}_{fm}$ and $\boldsymbol{q}_{f(m-1)}$ ($\boldsymbol{q}_{f0} = \boldsymbol{q}_f$).

IV. 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 [23], which is a collection of ROS stacks that supply several tools to simulate and interact with the robots, and its extension to multiple quadrotors [24].

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 (Actors), 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), (2), and (3), where the controllers $\boldsymbol{u}_b^i$, $\boldsymbol{u}_l$, and $\boldsymbol{u}_f^i$ are selected as (8), (9), and (10). In the simulation, the flood zone expands from a circle of radius 12m to 16m, and 13 UAVs are utilized for the tracking problem, including 1 inner leader, 4 inner followers, and 8 boundary drones. The simulation lasts for 60 seconds. The numerical values adopted in the simulation are $\boldsymbol{K}_b = \boldsymbol{K}_l = \boldsymbol{K}_f = \mathrm{diag}(1/300, 1/85)$ and the encircling speed of the boundary drones is selected as 1m/s. The maximum distance d in (7) is chosen to be 10m.

The camera image of a boundary drone is shown in Fig. 6. 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 midpoint of the two CMs. According to the proposed vision-based boundary tracking algorithm III-A.1, the black dot follows the motion of the geometric center of the camera image such that the boundary drone tracks the propagation of the flood zone.

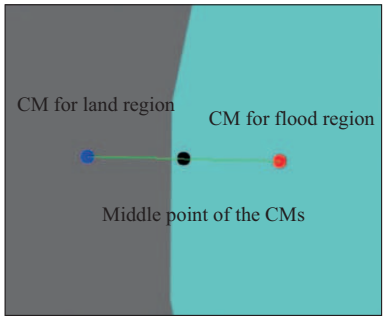

Fig. 6. Camera image of a boundary drone in the caging stage.

It can be seen from Fig. 7 that the boundary drones, illustrated by yellow-colored quadrotors, keep an average distance between adjacent robots, while the inner drones, illustrated by green-colored and red-colored quadrotors adaptively distribute themselves to cover the interior area of the flood zone. The trajectories of all the UAVs are plotted in Fig. 8.

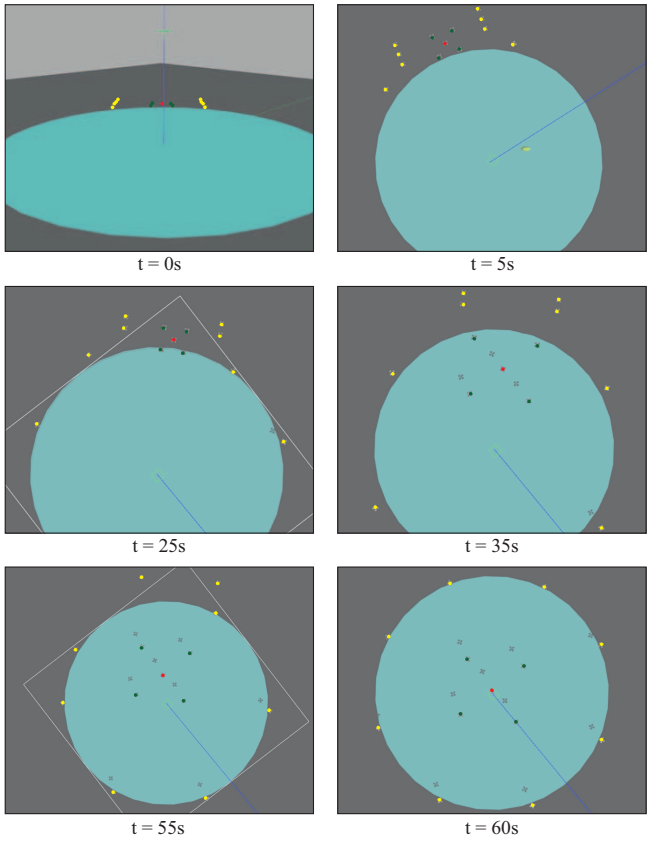

Fig. 7. A group of boundary drones (yellow), inner leader (red), and inner followers (green), cage, track, and cover a circular expanding flood area (the grey UAV-shaped objects are the shadows of the UAVs).

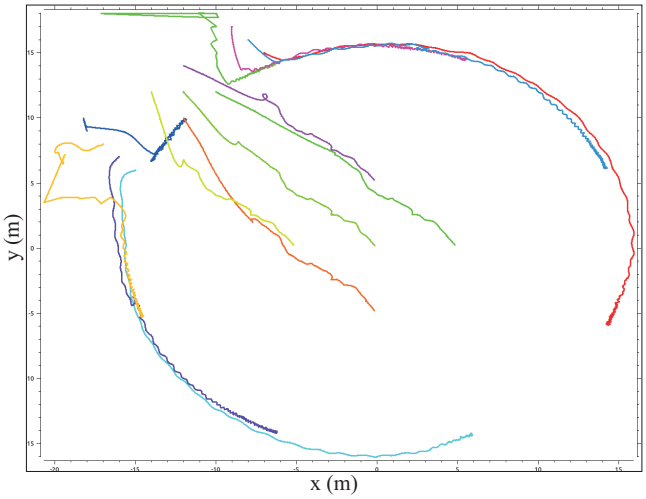

Fig. 8. Traces of UAV 1 (red), UAV 2 (blue), UAV 3 (green), UAV 4 (purple), UAV 5 (yellow), UAV 6 (dark blue), UAV 7 (dark green), UAV 8 (pink), UAV 9 (orange), UAV 10 (light blue), UAV 11 (light green), UAV 12 (dark pink), and UAV 13 (olive).

V. 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 consists of two stages, a caging stage and a covering stage, which take

place simultaneously. The caging stage averagely distributes UAVs along the boundary of the flood zone, tracking its propagation. The covering stage adaptively allocated UAVs among the interior area of the flood zone, covering it as much as possible with little overlapping of field of view for each UAV. Vision and flocking based control algorithms have been proposed to implement the strategy, the validity of which 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 (Actors), 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 [25].

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 complicated when the number of the UAVs increases. To deal with this problem, we plan to develop a higher level user interface for populating Gazebo with multiple robots.

The following issues also need to be addressed in our future work. Firstly, the effect of external disturbance to the UAVs due to strong wind needs to be taken into account. To deal with the disturbance, a robust controller will be developed for the position tracking part in the current control strategy [26]–[28]. Secondly, experiments will 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. Also, to estimate the state of the target UAVs, the Kalman filter will be included in the construction of the flocking control algorithm.

ACKNOWLEDGEMENTS

This research was supported, in part, by the Japan Science and Technology Agency, the JST Strategic International Collaborative Research Program, Project No. 18065977.

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] 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.

[4] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 10–11, pp. 1002–1033, 2007.

[5] 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.

[6] Z. Li, Z. Duan, G. Chen, and L. Huang, "Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint," *IEEE Transactions on Circuits and Systems I*, vol. 57, no. 1, pp. 212–224, 2010.

[7] S. Coogan and M. Arcak, "Scaling the size of a formation using relative position feedback," *Automatica*, vol. 48, no. 10, pp. 2677–2685, 2012.

[8] Y. Kobayashi, T. Endo, and F. Matsuno, "Distributed formation for robotic swarms considering their crossing motion," *Journal of the Franklin Institute*, vol. 355, no. 17, pp. 8698–8722, 2018.

[9] 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.

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

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

[12] ——, "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.

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

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

[15] 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.

[16] D. Sakai, H. Fukushima, 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.

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

[18] R. M. Francos and A. M. Bruckstein, "Search for smart evaders with sweeping agents," *CoRR*, vol. abs/1905.04006, 2019. [Online]. Available: http://arxiv.org/abs/1905.04006

[19] A. Barel, R. Manor, and A. M. Bruckstein, "COME TOGETHER: multi-agent geometric consensus (gathering, rendezvous, clustering, aggregation)," *CoRR*, vol. abs/1902.01455, 2019. [Online]. Available: http://arxiv.org/abs/1902.01455

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] Y. Bai, M. Svinin, and M. Yamamoto, "Backstepping trajectory tracking control for a spherical rolling robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, Oct 2016, pp. 298–303.

[27] 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.

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