A Path Planning Algorithm for Collective Monitoring Using Autonomous Drones

Shafkat Islam, Abolfazl Razi
School of Informatics, Computing and Cyber Systems, Northern Arizona University, Arizona, USA
shafkat_eee_cuet@ieee.org, abolfazl.razi@nau.edu

***Abstract*—This paper presents a novel mission-oriented path planning algorithm for a team of Unmanned Aerial Vehicles (UAVs). In the proposed algorithm, each UAV takes autonomous decisions to find its flight path towards a designated mission area while avoiding collisions to stationary and mobile obstacles. The main distinction with similar algorithms is that the target destination for each UAV is not apriori fixed and the UAVs locate themselves such that they collectively cover a potentially time-varying mission area. One potential application for this algorithm is deploying a team of autonomous drones to collectively cover an evolving forest wildfire and provide virtual reality for fire fighters. We formulated the algorithm based on Reinforcement Learning (RL) with a new method to accommodate continuous state space for adjacent locations. To consider more realistic scenario, we assess the impact of localization errors on the performance of the proposed algorithm. Simulation results show that success probability for this algorithm is about 80% when the observation error variance is as high as 100 (SNR:-6dB).**

***Index Terms*—UAV networks, Reinforcement Learning, Virtual Reality, Wildfire Monitoring.**

I. INTRODUCTION

The use of Unmanned Aerial Vehicles (UAVs) in various applications has been growing in recent years. This growth is mainly due to the technological advances in developing high capability drones with robust control and task management [1], predictive communication protocols [2]–[5], efficient spectrum utilization [6], optimal tracking [7], auto landing [8], [9], path planning and collision avoidance [10], [11] as well as low operation and maintenance costs [12] that make drone technology a suitable solution for many applications including habitat monitoring [13], [14], traffic control [15], remote sensing [16], border patrolling [17], smart agriculture [18], disaster management [19], and providing backhaul connectivity for wireless networks [20], [21] to name a few.

In recent years, incidence of wildfire throughout the USA has been increasing at an alarming rate due to climate change and human factors [22]. The recent California camp fire is accounted for the death of at least 85 people and 296 missing individuals [23]. It destroyed more than 14,000 residences [23], becoming both California's deadliest and most destructive wildfire on record. According to the statistics of National Interagency Fire Center (NIFC) [24], on average 63235 incidents of wildfires per year has been reported within 2008 to 2017 in the US. Various methods have been used for monitoring wildfires. The authors of [25]–[28] proposed UAV based methods for detecting and tracking the wildfires. However, in most UAV-based systems, a single remotely-controlled drone is deployed to perform the designated task, which requires human intervention in a short distance and may seriously endanger the controller's life. Further drawbacks of using single drone for wildfire monitoring include low spatial and temporal resolutions, and limited flight time [29]–[32]. Recently, several efforts have been devoted to deploy a fleet of drones for forest surveillance and fire detection at lower operational costs [26], [27], [33], [34].

This paper aims at developing a path planning algorithm for fire detection when the target field (e.g. the wildfire) is time-varying and the flight area includes stationary and mobile obstacles. In order to accomplish a task like wildfire monitoring, drones have to deal with the dynamic behaviour of the environment through taking autonomous decisions. Use of different model-free learning-based path planning methods can assist achieving higher levels of autonomy in taking decisions. Learning-based methods such as Reinforcement Learning (RL) gain knowledge about the environment through exploring the actions and investigating the rewards to develop optimal policies without the need for prior mathematical models or knowledge about the environment.

In [35], a PID+Q-learning algorithm is used to navigate a UAV to its target destination. In [36], the authors used multi-agent reinforcement learning algorithm for providing coverage of a field. In this case, the UAVs are allowed to learn cooperatively. However, none of these methods consider mobile obstacles and time-varying targets, which is the focus of this paper. In particular, we developed a collision-free path planning algorithm, where the UAVs adjust their actions with the change in environment. In this algorithm, we assumed that the UAVs could not share information with one another.

The rest of this paper is organized as follows. Section II describes the system model and also states the problem formulation. Section III briefly describes the path planning algorithm and section IV presents the simulation results. Finally, section V concludes the paper with offering potential future directions.

II. SYSTEM MODEL

We assume that N drones denoted by $n_1, n_2, \ldots, n_N$ are located in initial positions $\mathbf{p}_1(0), \mathbf{p}_2(0), ..., \mathbf{p}_N(0)$, where $\mathbf{p}_i(t) = \big(x_i(t), y_i(t)\big)$ is the position of drone n_i at time t. We also assume that there is an active fire represented by a continuous region, $D = \{(x, y)\}$. The fire front line of the fire is determined as $\{(x, y)|(x, y) \subset D, N_\epsilon(x, y) \not\subset D\}$, where $N_\epsilon(x, y) = \{(u, v)|\sqrt{(x-u)^2 + (y-v)^2} \leq \epsilon\}$ is the ϵ-neighbourhood of the fire front line. We also assume that there are M point obstacles $o_1, \ldots, o_M$ in the coverage area (e.g. trees or other drones), and the drones are supposed to keep a minimum distance R_o from these obstacles to avoid collision. Likewise, the desired location of drones should be

978-1-7281-1151-3/19/$31.00 ©2019 IEEE

within a safety zone surrounding the fire front line, which is defined by R_s. The goal is to develop a path planning algorithm to guide the drones towards the surrounding area (safety zone) of the fire front line by avoiding stationary and mobile obstacles at lowest possible time. We desire a fully distributed algorithm, where each drone makes independent navigation steps. The drones are not allowed to convene by exchanging information to find their paths. We assume that the drones can estimate the location of obstacles, other drones, and the fire front line on the fly using a proper target tracking system with a Gaussian measurement error $w \sim \mathcal{N}(0, \sigma_n^2)$.

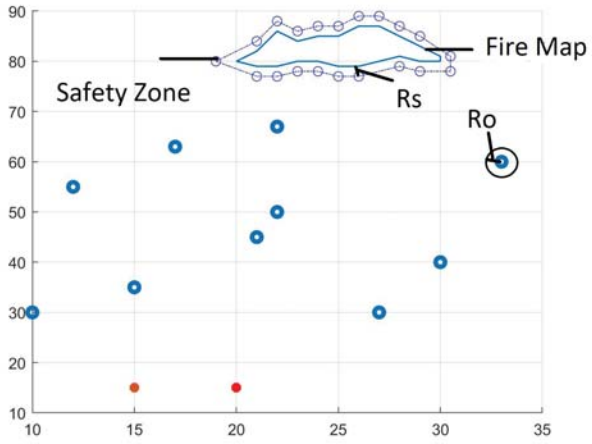

Fig. 1. System Model: The target fire map is shown with blue shape. The red and blue circles, respectively, present the observer drones and the obstacles.

In Fig. 1, blue circles represent the point obstacles, and red dots represent the agents (observer drones). The blue shape represents the fire front line. The position of point obstacles and the shape of the target region can be static or time-varying. We considered both situations in our analysis. For the sake of simplicity, we consider a 2D path planning where drones hover in a given altitude, but the extension to 3D path planning is straightforward.

III. Algorithm

We develop our method based on the RL framework to direct the drones to their destinations. We formulate the algorithm for continuous state space to make it more realistic. We use a new method for solving RL problems in continuous space instead of using conventional function approximation methods. In our method, we use Bellman equation as the action-value function while eliminating the need of exploring Q-value for each state. The underlying assumption of this approach is that the Q-values for adjacent locations are correlated and hence slowly change over time during the exploration phase and we can update the current Q-value based on the accumulated rewards during the recent local exploration. This property makes the algorithm fast and suitable for using in continuous state space problems.

In the algorithm, each UAVs is considered as an independent agent. For simplicity, We consider a 2-D random-walk mobility with constant steps denoted by r, so the actions of drones are restricted to 5 possibilities (no movement, moving east, west, north, and south). This action set is denoted by $A = \{\circ, \rightarrow, \leftarrow, \uparrow, \downarrow\}$. Often, one of the 4 motion steps are selected as the optimal action, due to the definition of the reward function as follows next.

The set of states is defined as $\{S\}$, which represents the location of the agent in a 2-D space considering the location of mobile obstacles and the current fire map. For convenience, we map this set of continuous space into a finite set developed by splitting the region into square tiles with side r. As mentioned earlier, we assume that the adjacent states have similar action-reward mappings, so we keep only one Q-function for the current local region.

A. Location based reward function

The reward function for an agent, when taking action a at state s is defined as

$$s_i(t) : \mathbf{p}_i(t) \xrightarrow{a} s_i(t+1) : \mathbf{p}_i(t+1), \tag{1}$$

$$\begin{aligned} R(s,a) = {} & \alpha_1[f_1(\mathbf{p}_i(t+1)) - f_1(\mathbf{p}_i(t))] \\ & - \alpha_2[f_2(\mathbf{p}_i(t+1)) - f_2(\mathbf{p}_i(t))] \\ & - \alpha_3[f_3(\mathbf{p}_i(t+1)) - f_3(\mathbf{p}_i(t))], \end{aligned} \tag{2}$$

which includes three sub-functions $f_1()$, $f_2()$, $f_3()$ evaluated based on the agent locations. Eq (1) represents the location change due to taking action a at state s, and Eq (2) quantifies the reward. The reward function is designed to offer incentive for flying towards the fire map (by changes incurred to $f_1()$ due to location change), and penalize flying towards the obstacles (by $f_2()$) and other agents (by $f_3()$). The parameters $\alpha_1, \alpha_2, \alpha_3$ are used to balance between different objectives, and we use $\alpha_1 = \alpha_2 = \alpha_3 = 1$.

To be more specific, f_1 is the negative of the distance from the agent's location $\mathbf{p}_i(t)$ (equivalently its state $s_i(t)$ in the RL framework) to the nearest point in the safety zone surrounding the fire front line $D(t)$ at time t. Therefore, we have

$$f_1(\mathbf{p}_i(t)) = - \min_{\mathbf{x}(t) \in D} d(\mathbf{p}_i(t) - \mathbf{x}(t)), \tag{3}$$

where $d(\vec{q}, \vec{r})$ is an arbitrary distance metric between position vectors $d(\vec{q}$ and $\vec{r})$. An intuitive choice is using Euclidean distance defined as $d(\vec{q}, \vec{r}) = |\vec{q} - \vec{r}|_2$.

Similarly, f_2 is a function to enforce collision avoidance with obstacles along the path. It is defined as

$$f_2(\mathbf{p}_i(t)) = \frac{1}{M} \sum_{j=1}^{M} e^{-d(\mathbf{o}_j(t), \mathbf{p}_i(t)) + d_c}, \tag{4}$$

where $\mathbf{o}_j(t)$ represents the location of object o_i at time t. Parameter M denotes the number of visible obstacles from a certain state at time t. The tunable parameter d_c is used to enforce a clearance distance around the obstacles, and we set it to $d_c = R_o$.

Finally, f_3 is a function used to enforce maximal separation among the flying drones to provide a better coverage of the target field. It also decreases the chance of collision among the drones. This function is defined as

$$f_3(\mathbf{p}_i(t)) = \frac{2}{(N-1)\pi} \sum_{j=1, j \neq i}^{N} \operatorname{acot}\big(d(\mathbf{p}_j(t), \mathbf{p}_i(t))/r_c\big), \tag{5}$$

where we penalize the distance with the $N-1$ agents in the system. In this function, we use arccot function to reflect higher sensitivity around zero. The tunanble parameter r_c controls the rate of the penalization and in this paper we arbitrarily set $d_c = R_o$. Also, the coefficient $2/(N-1)\pi$ is used to limit the penalization term into range $[0, 1]$. Here, we consider the current locations of other drones as surrogate for their next locations when evaluating $f_3(\mathbf{p}_i(t+1))$, since information exchange among the drones is not allowed.

We consider two idealistic and realistic scenarios while calculating the reward functions for each state. In case-I, the drones are aware of the current locations of all stationary and mobile obstacles (i.e. a tracking system with unlimited observation range). However, in a more realistic case-II, the drones can locate obstacles only within a specific range. Therefore, only the objects within a specific range around an agent contribute to functions f_2 and f_3. In the simulation results in section IV, we set the value of this specific region equal to d_c, meaning that the obstacles are recognized only when the agent enters their collision range. We take this assumption inspired by the collision-avoidance feature through igniting collision alarms implemented in some commercial drones.

B. Q-learning

We use the well-known Bellman equation to update Q-functions as follows:

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha[R_(s,a) + \gamma \max_{a'}(Q(s',a')], \quad (6)$$

where $Q_t(s,a)$ is the Q-value of taking action a_t at state s_t, α is the learning rate and γ is the discount factor to account for future rewards. In our implementations, we set the discount factor $\gamma = 0$, due to the fact that the agents receive immediate rewards for their actions based on the developed reward functions. In other words, the future impact of actions are already captured through the reward functions. Further, this simplifying approach eliminates the requirement of storing Q-values for distinct states and enables us to approximate all surrounding states by the current local state. Therefore, the updated action-value function is defined as

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha R(s,a) \quad (7)$$

C. Exploitation and exploration phases

The algorithm consists of two alteranating phases of exploration and exploitation. The purpose of the exploration phase (denoted by S1) is to develop and fine tune the action-reward mapping using (7) in order to use it to steer drones' motions in the exploitation phase (S2). In the exploration phase, an agent explores the four possible actions in the current state to update the Q-values. We can use a probabilistic (e.g. random exploration) or a sequential exploration method to cover all states at each exploration phase. In this paper we use the sequential exploration method to cover all actions equally.

In the exploitation phase, each agent takes action based on their experience gained in the exploration phase. They also assume that the property of most adjacent states are similar, and so they take the same action for consecutive states in the exploitation phase.

There is a trade-off between the frequency of exploitation and exploration phase. Using more often exploration phases increases the overhead of the system (in terms of energy consumption and execution time) by taking exploration steps. On the other hand, more frequent exploration phases provide higher agility in recognizing the changes of Q-values due to the dynamic topology of the network and the agent's mobility. In order to implement the transition between the two modes, here we use a Markov model where the initial state probabilities are defined as $P(S1) = 20\%$ and $P(S2) = 80\%$. Let us denote the transition probabilities as $P_{11} = 1 - P_{12} = p_0$ and $P_{22} = 1 - P_{21} = p_1$, and choose p_0 and p_1 such that the resulting steady-state state probabilities are $P(S1) = 20\%$ and $P(S2) = 80\%$. We use the following method to update the transition probabilities. As, we take actions in the exploitation phase, we keep track of the changes in the Q-values as follows:

$$\Delta Q(a) = Q(s_{i+1}, a) - Q(s_i, a) \quad (8)$$

If the change in Q-values is above a predefined positive threshold $T_1 > 0$, the system is stable and we decrease the probability of exploration phase by decreasing p_0 with a constant step (e.g. δp). However, if $\Delta Q(a)$ is negative and below a predefined threshold T_2, it is very likely that the optimal state has been changed due to severe topology changes, so we increase p_0 (or equivalently decrease p_1) to increase the probability of exploration phase. Otherwise, we keep the probabilities unchanged. Here is the summary of this step:

$$\begin{cases} p_0 = \min(0\%, \max(100\%, p_0 - \delta p\%)) & \text{if } \Delta Q(a) \geq T_1 \\ p_0 = \min(0\%, \max(100\%, p_0 + \delta p\%)) & \text{if } \Delta Q(a) \leq T_2 \\ p_0 = p_0 & \text{if } T_1 \leq \Delta Q(a) \leq T_2 \end{cases} \quad (9)$$

Note that we reset to original probabilities after completing a successful exploration phase. This process is repeated until the agents reach the desired position on the neighbourhood of the fire line. A summary of this algorithm is presented below.

IV. Numerical Results

We simulated the algorithm on MATLAB Environment. We considered stationary (Fig. 2-4) and mobile obstacles (Fig. 7,8) obstacles in simulation. In fig. 8 we considered time varying target as well. For stationary obstacles we calculated reward by using case-I and II but for mobile obstacles we calculated rewards by using case-II for reward function. We also calculated the effect of localization errors in simulation. In Figs 2-4, red (UAV-1) and yellow (UAV-2) lines show the paths for two UAVs. The blue 'o' represents the stationary point obstacles and the area encircled by the solid blue line represents the safety zone surrounding the fire front-line. Fire lines can generally be of random shape. The shape of the safety zone would be congruent with the shape of the fire line. The bold red 'x' (Fig. 3 and 4) represents the position of the misinterpreted obstacles due to measurement errors. The UAVs observe rewards based on these misplaced obstacles rather than considering the original position of the obstacles.

Algorithm 1 UAV Path Planning for Collective Monitoring

0: **Initialization:**
0: set the locations for the target, agents, and obstacles
0: set probabilities for exploration and exploitation phases $P(S1) = 20\%$ and $P(S2) = 80\%$ and (p_0 and p_1 accordingly)
0: state=initial position of the UAV
0: **Exploration:**
0: **for** UAV i=1 to N **do**
0: Explore the four possible state, calculate reward functions using (1-5), and update Q-values using (7)
0: Goto exploitation phase with probability $P_{12} = 1 - p_0$.
0: Loop
0: **Exploitation:**
0: **for** UAV i=1 to N **do**
0: Take optimal action a based on Q-function
0: Calculate reward functions using (1-5), and update Q-values using (7)
0: Update ΔQ using (8)
0: Update transition probabilities using (9)
0: Goto exploration phase with probability $P_{21} = 1 - p_1$.
0: Loop

In fig-2, the UAVs avoid the stationary obstacles keeping a minimum clearance distance from the obstacles. The value of this clearance distance can be controlled by the parameter d_c. The UAVs also maintain a constant distance between them. For covering the maximum region in a certain area the UAVs should keep a distance of their coverage capacity between them. In fig-3, Gaussian noise ($\mu = 0$, $\sigma^2 = 2$) is added with the tracking systems of the UAVs. The red 'x' shows the displaced position of the obstacles due to the measurement error. The UAVs calculate the path based on these pseudo obstacles (red 'x') rather than the original ones (represented in blue 'o'). As the measurement error is small, the UAVs could avoid the obstacles instead of the error in distance measurement but the path for each UAV is different from fig. 2.

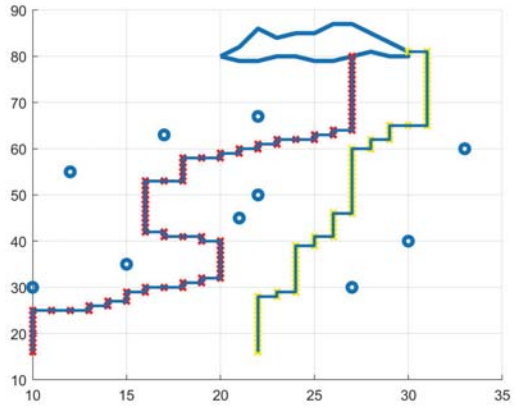

Fig. 2. Simulation of the algorithm without localization errors (case-I for reward).

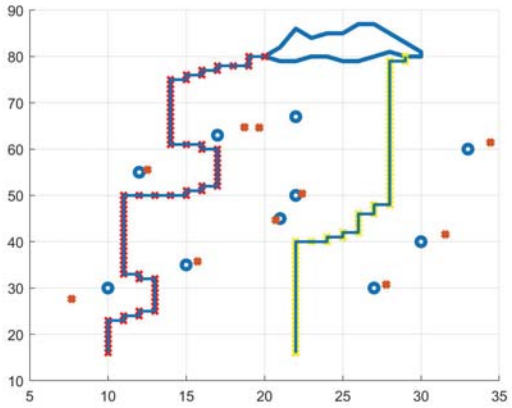

Fig. 3. Simulation of the algorithm with localization error ($\mu = 0$, $\sigma^2 = 2$) in tracking system (case-I for reward).

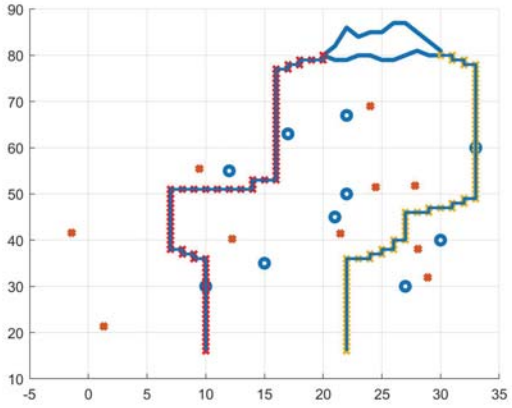

Fig. 4. Simulation of the algorithm with localization error ($\mu = 0$, $\sigma^2 = 50$) in tracking system (case-I for reward).

In Fig-4, we increased the variance of the localization error to 50 while keeping the other parameters same as Fig. 2. As the amount of measurement error is very large compared to the minimum clearance (d_c), the UAV-1 collides with one of the obstacles and UAV-2 also collides with other obstacle. This figure depicts the effect of measurement error more significantly.

Fig-5 and fig-6 show the curve of average number of collisions (out of 100 iterations) between the UAV (UAV-1 and UAV-2) and the obstacles, and also the collision between the UAVs for the different variance level of measurement errors. The variance level is selected from zero to hundred with a step size of two. At each level we iterated the process for 10 times and calculated the average collision number (out of 100). The blue 'o' represents the average number of collision for UAV-1. The red 'x' represents the average collision for UAV-2. The value of error for each UAV is independent of the other UAV at each level. In case-I of the reward function the highest number of collision occurred for UAV-1 is 21 and for UAV-2 is 19. From the Fig. 5 it is also observed that the UAV-1 is more prone to collision than the UAV-2. Though both

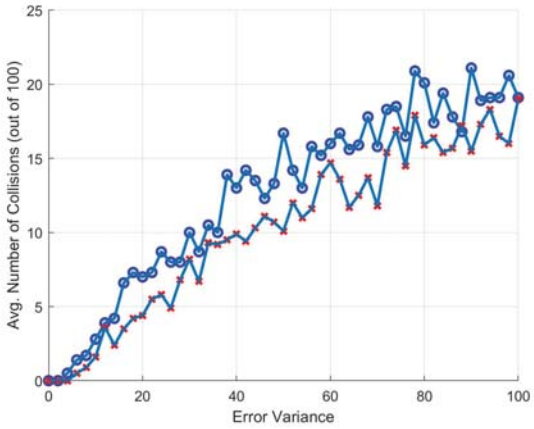

Fig. 5. Variance Vs Avg. number of collision for UAV-1 and UAV-2 (case-I for reward).

number collides at same value in some cases. But the curves show that about 80% of cases each of the UAVs succeeds to reach their destination, without any collision, irrespective of the level of error variance. In Fig. 6 we considered that, from a specific state, the tracking system could track obstacles within a specific range (case-II for reward). In this case the number of collision is slightly higher than the previous case. Here the success rate for the UAVs in reaching the destination, without collision, is about 77% which is lower than the previous one.

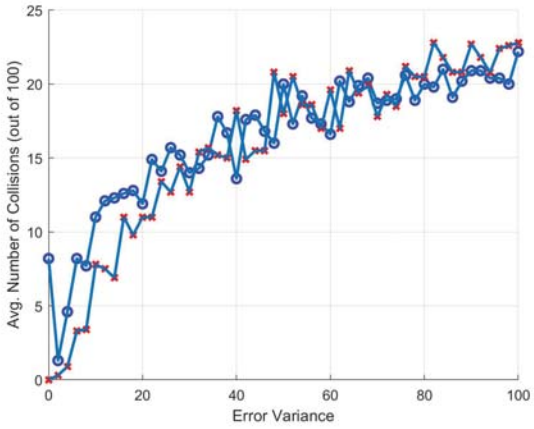

Fig. 6. Variance Vs Avg. number of collision for UAV-1 and UAV-2 (case-II for reward).

In Fig. 7 and Fig. 8, we simulated the algorithm with mobile obstacles and calculated rewards by using the case-II of the reward function. The step size of these mobile obstacles is as equal as the step size of the UAVs. The bold blue 'o' represents the initial positions of the obstacles and the bold green 'o' represents the final positions of the obstacles. The thin 'x' represents the positions of obstacles at different times. At any specific time instance, the mobile obstacles can randomly move any of the four directions or stay in their own positions. From simulation we observed that the UAVs could avoid

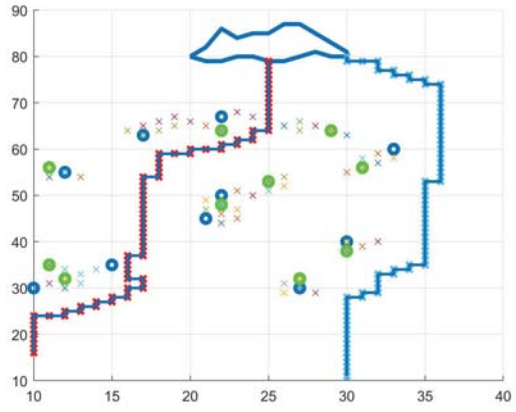

Fig. 7. Simulation of the algorithm with mobile obstacles.

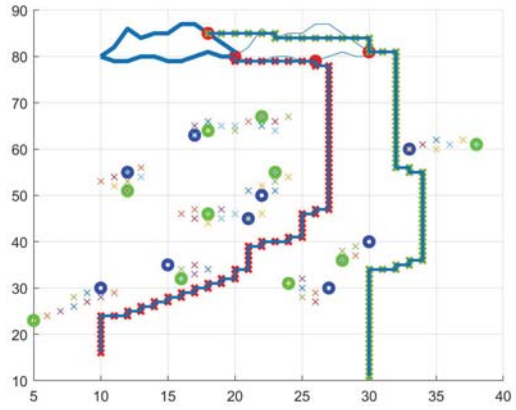

Fig. 8. Simulation of the algorithm with time varying target

collisions with these mobile obstacles at different time steps in the path, and also reach at the desired locations in the target zone. In figure 8, we observed the effect of time-varying target. The thin blue line denotes the initial position of the target region and the bold blue line represents the final position of the target region. The bold red 'o' represents the positions of UAVs in target zone. UAVs calculated reward for a state by using the initial target position as the target. We assumed that after reaching the initial target position, UAVs could detect the new location of the target zone by their tracking system, and also the position of the target zone can only move to the adjacent locations. The simulation results show that UAVs could relocate themselves according to the change in position of the target zone, and also avoid collisions with mobile obstacles at any particular time in the path.

V. Conclusion

A distributed path planning algorithm for a team of UAVs is presented in the paper. Using the algorithm, UAVs can avoid collisions with obstacles (stationary and mobile) and with other UAVs in the path without exchanging information among the UAVs. The algorithm also allows UAVs in finding

the optimal location on the border line of the monitoring region and avoid overlaps in monitoring zone with other UAVs. UAVs can also relocate themselves with the change in position of target region. This algorithm can be applied in wildfire monitoring, disaster management, and other dynamic environments. The algorithm uses a new method for mapping adjacent states into correlated states in continuous state space. In future, our research would focus on predicting the shape of the dynamic wildfire zone with respect to time and synchronizing the speed of the UAVs with the spread rate of wildfire. We would also conduct research on providing virtual reality models for the fire fighters in the system. This inclusions would enrich the system and enable us to utilize its robustness in diverse applications.

REFERENCES

[1] F. Afghah, M. Zaeri-Amirani, A. Razi, J. Chakareski, and E. Bentley, "A coalition formation approach to coordinated task allocation in heterogeneous uav networks," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 5968–5975.

[2] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones," *Ieee vehIcular technology magazIne*, vol. 12, no. 3, pp. 73–82, 2017.

[3] J. Won, S.-H. Seo, and E. Bertino, "A secure communication protocol for drones and smart objects," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. ACM, 2015, pp. 249–260.

[4] A. Rovira-Sugranes and A. Razi, "Predictive routing for dynamic uav networks," in *Wireless for Space and Extreme Environments (WiSEE), 2017 IEEE International Conference on*. IEEE, 2017, pp. 43–47.

[5] M. Khaledi, A. Rovira-Sugranes, F. Afghah, and A. Razi, "On greedy routing in dynamic uav networks," *arXiv preprint arXiv:1806.04587*, 2018.

[6] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, and J. Ashdown, "Distributed cooperative spectrum sharing in uav networks using multi-agent reinforcement learning," *arXiv preprint arXiv:1811.05053*, 2018.

[7] A. Razi, F. Afghah, and J. Chakareski, "Optimal measurement policy for predicting uav network topology," in *Signals, Systems, and Computers, 2017 51st Asilomar Conference on*. IEEE, 2017, pp. 1374–1378.

[8] R. Bapst, R. Ritz, L. Meier, and M. Pollefeys, "Design and implementation of an unmanned tail-sitter," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1885–1890.

[9] M. R. Cacan, E. Scheuermann, M. Ward, M. Costello, and N. Slegers, "Autonomous airdrop systems employing ground wind measurements for improved landing accuracy," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 3060–3070, 2015.

[10] W. K. Bodin, J. Redman, and D. C. Thorson, "Navigating a uav with obstacle avoidance algorithms," Jun. 5 2007, uS Patent 7,228,232.

[11] H. Peng, A. Razi, F. Afghah, and J. Ashdown, "A unified framework for joint mobility prediction and object profiling of drones in uav networks," *Journal of Communications and Networks*, vol. 20, no. 5, pp. 434–442, 2018.

[12] J. Dentler, S. Kannan, M. A. O. Mendez, and H. Voos, "A real-time model predictive position control with collision avoidance for commercial low-cost quadrotors," in *Control Applications (CCA), 2016 IEEE Conference on*. IEEE, 2016, pp. 519–525.

[13] J. Paneque-Gálvez, M. K. McCall, B. M. Napoletano, S. A. Wich, and L. P. Koh, "Small drones for community-based forest monitoring: An assessment of their feasibility and potential in tropical areas," *Forests*, vol. 5, no. 6, pp. 1481–1507, 2014.

[14] J. Zhang, J. Hu, J. Lian, Z. Fan, X. Ouyang, and W. Ye, "Seeing the forest from drones: Testing the potential of lightweight drones as a tool for long-term forest monitoring," *Biological Conservation*, vol. 198, pp. 60–69, 2016.

[15] M. D. Fontaine, P. J. Carlson, and H. G. Hawkins, "Evaluation of traffic control devices for rural high-speed maintenance work zones: Second year activities and final recommendations," Texas Transportation Institute, Texas A & M University System, Tech. Rep., 2000.

[16] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," *ISPRS Journal of photogrammetry and remote sensing*, vol. 92, pp. 79–97, 2014.

[17] C. C. Haddal and J. Gertler, "Homeland security: Unmanned aerial vehicles and border surveillance." LIBRARY OF CONGRESS WASHINGTON DC CONGRESSIONAL RESEARCH SERVICE, 2010.

[18] P. Tripicchio, M. Satler, G. Dabisias, E. Ruffaldi, and C. A. Avizzano, "Towards smart farming and sustainable agriculture with drones," in *Intelligent Environments (IE), 2015 International Conference on*. IEEE, 2015, pp. 140–143.

[19] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, no. 1, pp. 24–32, 2017.

[20] E. Kalantari, M. Z. Shakir, H. Yanikomeroglu, and A. Yongacoglu, "Backhaul-aware robust 3d drone placement in 5g+ wireless networks," in *Communications Workshops (ICC Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 109–114.

[21] S. Naqvi, J. Chakareski, N. Mastronarde, J. Xu, F. Afghah, and A. Razi, "Energy efficiency analysis of uav-assisted mmwave hetnets," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[22] D. V. Spracklen, L. J. Mickley, J. A. Logan, R. C. Hudman, R. Yevich, M. D. Flannigan, and A. L. Westerling, "Impacts of climate change from 2000 to 2050 on wildfire activity and carbonaceous aerosol concentrations in the western united states," *Journal of Geophysical Research: Atmospheres*, vol. 114, no. D20, 2009.

[23] C. R. W. Jr. (2018) The deadliest, most destructive wildfire in californias history has finally been contained. [Online]. Available: https://www.washingtonpost.com/nation/2018/11/25/camp-fire-deadliest-wildfire-californias-history-has-been-contained

[24] "National interagency fire center," https://www.nifc.gov/fireInfo/nfn.htm, accessed: 2018-12-13.

[25] H. X. Pham, H. M. La, D. Feil-Seifer, and M. Deans, "A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 6648–6653.

[26] L. Merino, F. Caballero, J. R. Martínez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple uavs: Application to automatic detection of forest fires," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 165–184, 2006.

[27] H. Cruz, M. Eckert, J. Meneses, and J.-F. Martínez, "Efficient forest fire detection index for application in unmanned aerial systems (uass)," *Sensors*, vol. 16, no. 6, p. 893, 2016.

[28] C. Yuan, Z. Liu, and Y. Zhang, "Fire detection using infrared images for uav-based forest fire surveillance," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 567–572.

[29] J. Moore. (2017) Rise of the fire drones first responders expand role of robots. [Online]. Available: https://www.aopa.org/news-and-media/all-news/2017/april/03/rise-of-fire-drones

[30] J. Larson. (2016) Drones and the fire service. [Online]. Available: http://www.fireengineering.com/articles/print/volume-169/issue-10/features/drones-and-the-fire-service.html

[31] M. Margaritoff. (2017) Drones in firefighting: How, where and when they're used. [Online]. Available: http://www.thedrive.com/aerial/16770/drones-in-firefighting-how-where-and-when-theyre-used

[32] U. T. B. Jansen. (2017) Nyc fire fighters use drone to help battle blaze for first time. [Online]. Available: https://www.usatoday.com/story/news/2017/03/08/drone-firefighters/98848038/

[33] L. Merino, F. Caballero, J. Martinez-de Dios, and A. Ollero, "Cooperative fire detection using unmanned aerial vehicles," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 1884–1889.

[34] L. Merino, F. Caballero, J. R. Martínez-De-Dios, I. Maza, and A. Ollero, "An unmanned aircraft system for automatic forest fire monitoring and measurement," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 533–548, 2012.

[35] H. X. Pham, H. M. La, D. Feil-Seifer, and L. V. Nguyen, "Autonomous uav navigation using reinforcement learning," *arXiv preprint arXiv:1801.05086*, 2018.

[36] H. X. Pham, H. M. La, D. Feil-Seifer, and L. Van Nguyen, "Cooperative and distributed reinforcement learning of drones for field coverage," *arXiv preprint arXiv:1803.07250*, 2018.