

Cooperative UAV Trajectory Planning for Plume Wrapping of a Spherical Dome

Baisakhi Chatterjee

*Computer Science Department
North Carolina State University
bchatt@ncsu.edu*

Rudra Dutta

*Computer Science Department
North Carolina State University
rdutta@ncsu.edu*

Abstract—Unmanned Aerial Vehicles (UAV) are surging in popularity as a result of the versatility of their usage. Research has focused on a variety of applications by UAV groups including surveillance, target tracking, network coverage and mapping of various areas or incidents. Autonomous UAVs can be utilized to perform tasks which may be potentially dangerous for human intervention. Earlier work has shown that a group of UAVs can successfully circumscribe a plume of hazardous contaminants in the air and map the boundary of such a spill. In this paper, we show that this idea can be extended to plumes which contain elements heavier than air and, as such, fall to the ground such that they approximately form a spherical dome. Mapping such a boundary is challenging because UAVs need to follow the plume as close to the ground as possible without crashing. Our results prove that a UAV platoon can successfully communicate and coordinate amongst themselves in order to surround such a leak.

Index Terms—unmanned aerial vehicle, UAV, drone, autonomous agents, aerial communication, plume wrapping

I. INTRODUCTION

In recent times, we have seen a growing interest in the applications of Unmanned Aerial Vehicles. Both civil and commercial agencies are looking to make use of the vast varieties of missions that are possible due to the low risk of autonomous flights. UAVs are small, highly maneuverable and ready for rapid deployment, making them ideal agents for performing tasks which are either too remote or too dangerous for human intervention. More and more research is now focusing on the the benefits of multi-UAV swarms. Search and rescue [1], traffic surveillance [2] and target monitoring [3] are a few of the operations that have seen the positive effects of utilizing UAV platoons.

In this work, we have focused on the problem of Plume Wrapping by an autonomous UAV platoon. Plume Wrapping is the problem of mapping the boundary of some pollutant or contaminant by a group of autonomous agents. Prior work in this area have considered statistical approximations [4], [5], use of single UAVs [6], [7] and multi-UAV groups [8], [9]. More recently, Chatterjee and Dutta [10] have proposed a technique for mapping hazardous leaks in the air by having a group of agents surround a spherical plume based on the Fibonacci spiral method of placing n points on the surface

The material presented in this paper is based upon work supported by the National Science Foundation PAWR Program under Grant No.: CNS-1939334, and its supplement for studying NRDZs.

of a sphere. While it is reasonable to assume that the starting phase of a plume with pollutants lighter than air will assume a roughly spherical shape [11], this technique poses a challenge when the contaminants are heavier than air and drop to the ground. In these cases, a roughly dome-shaped shroud can be seen, as noted in the Apex, NC, Chemical Fire of 2006 [12] where an explosion at the chemical site generated a dense plume of unknown contaminants that moved close to the ground. Emergency responders were unable to begin attacking the fire immediately since they could not accurately gauge the composition and nature of the pollutants. A group of UAVs could have proved invaluable in this effort. Another source for plumes which stay relatively close to the ground during their formation are Bubble Plumes [13]. Bubble Plumes occur as a result of gaseous bubbles escaping from a source and radial dispersion causes them to release liquid at various levels. Finally, if we expand the definition of ‘plume’ to refer to any entity that produces detectable levels of emission, UAVs can also be used to detect network strength emitted by base stations [14]. The challenge in mapping such a spread is that, to gain as much information as possible, UAVs need to fly close to the ground while maintaining a safe distance from it. Additionally, the nature of incidents like the Apex Chemical Fire require a fast response. Thus, it is imperative that UAVs not only respond quickly, but are also able to estimate the shape of the plume in real-time while maintaining safety.

In this paper, we have extended [10] so that the UAVs can now map the emissions in a dome shape with the ground serving as the base. As can be seen in Fig. 1, each plume roughly follows the hemispherical pattern which validates the choice of our shape. We show that a group of UAVs is successful in mapping the boundary of the plume while ensuring that no UAV crashes into the ground. Once this predefined boundary region is reached, the UAVs can individually adjust their radial position such that a more precise mapping is achieved. Since individual movement of UAVs is a trivial problem to solve, we focus on coordinated trajectory planning to form the dome shaped bound.

II. ORIGINAL PLUME WRAPPING ALGORITHM

In [10], the authors designed an algorithm where a group of UAVs could autonomously map a spherical plume by periodically sampling the plume concentration and sharing

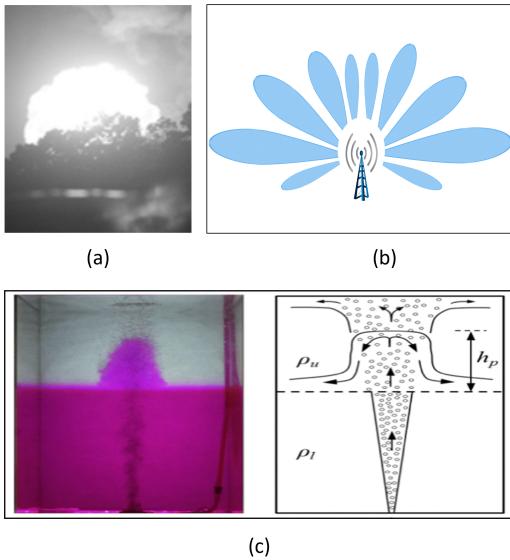


Fig. 1. (a) Initial Explosion and Plume Cloud for Chemical Fire in Apex, NC in 2006 (Reproduced from [12]). (b) Propagation of Radiation from Cell Tower. (c) Bubble Plume (reproduced from [15]).

this information with every other member of the group. The authors envisioned this problem as that of placing n points on the surface of a sphere, accomplished by using a 3D projection of the Fibonacci spiral. Initially, a single UAV, called ‘anchor’ UAV, starts its flight in an arbitrary direction till it detects a boundary region depicted by ‘low’ gaseous concentration of predetermined strength. The UAV then broadcasts this information to every other agent and the other UAVs calculate their own target position around the anchor UAV such that the combined positions resembled a sphere. Each UAV occupies a unique position based on its own RID (Remote ID).

The next phase of this process is the *Stop-Click* phase where each UAV decides their next target point based on the data observed by every member of the group. In the *Stop* portion, each UAV sends out a message with its current position and concentration, and then waits to receive updates from other members of the group. Once the UAVs receive the relevant updates, each agent individually executes the algorithm and chooses to either ‘Expand’ or ‘Rotate’. ‘Expansion’ refers to the UAV calculating its own position on a slightly larger Fibonacci sphere such that the radius is increased but the anchor point does not move. The UAV chooses ‘Expansion’ if all other members detect that they are inside the plume (i.e. not at the boundary or outside the plume region). If at least one UAV detects that it is outside, the agents choose ‘Rotation’. For ‘Rotation’ the UAVs move such that the imaginary sphere is pivoted along the anchor point towards the plume center. Since every UAV makes this decision on the same information received, the group as a whole makes the same choice, thereby, preserving synchronization. Once the decision is made, the UAV begins its flight and this is now called the *Click* turn. The *Stop* turn begins again when the UAV reaches its destination

and this phase continues iteratively till all UAVs reach the boundary region.

III. HEMISPHERICAL PLUME BOUNDARY

Our main contribution in this research has been to modify the Plume Wrapping algorithm such that the UAV group can wrap around plumes which remain close to the ground and can be approximately generalized to a dome shape. In order to model the plume, we initially chose a hemispherical spread of pollutants such that density decreases uniformly from the center to the boundary. The boundary region is modelled as an hemispherical ring where the minimum and maximum density can be controlled based on mission characteristics. When extending the original algorithm, we chose to preserve the core approach of placing n points using a Fibonacci spiral. A Fibonacci spiral is a fast, distributed approach to quickly locating target points by UAVs. Since this algorithm requires only RID of each UAV, it makes it an obvious choice for decentralized processing.

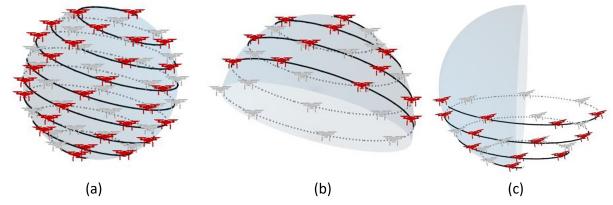


Fig. 2. (a) Original Fibonacci Spherical Spiral. (b) The correct orientation for wrapping a hemisphere. (c) Incorrect orientation for wrapping a hemisphere

The first challenge we faced was in limiting the spiral to a hemisphere instead of a sphere. The Fibonacci method is a 3D projection of the 2D spiral and thus, wraps around a plume in only a specific angle. This was not concerning in the case of a sphere since, for a perfect sphere, the orientation is irrelevant. However, while wrapping around a hemisphere, the spiral must be oriented such that the anchor point is at the top. If this does not happen, the placement of n points on a hemisphere will not result in an accurate mapping. On the contrary, this may even lead to up to half of the UAVs crashing into the ground. Thus, for this approach to work, it is imperative that the anchor UAV is placed as close to the top of the plume as possible. Fig. 2 shows the original Fibonacci spiral for a sphere and both the correct and incorrect wrappings for a hemisphere.

A. Boundary Search

To find the appropriate anchor point, the first UAV needs to be anchored at a boundary point which is also close the center top of the plume. From a logistical perspective, it might not always be possible for the first UAV to be placed in such a manner before launch, that it accurately reaches the best anchor position. Hence, the UAV must search for the highest point in the boundary region before fixing its position. To do so, we formulated a hill climbing approach. The first UAV, when launched, moves in an arbitrary direction towards the

plume till it reaches the boundary. When this UAV detects the boundary, it then changes its direction such that it samples concentrations all around it by moving left, right, forward and backwards respectively. At each point, if the UAV senses that the concentration is higher, this indicates that the UAV is moving inwards. At this point, the UAV begins moving upward till boundary is once again detected. Once the UAV no longer detects higher concentrations when moving on all four sides, it determines that the anchor position has been reached. The anchor UAV then broadcasts a *Boundary Reached* message. The details of this algorithm can be found in Algorithm 1.

Algorithm 1 : Boundary Search

Procedure: *boundarySearch (boundaryRegDens, curPosition)*

```

1: curDensity  $\leftarrow$  getDensity (curPosition)
2: if curDensity  $\in$  boundaryRegDens then
3:   DirectionList  $\leftarrow$  (Left, Right, Forward, Backward)
4:   for curDirection  $\in$  DirectionList do
5:     Move in curDirection
6:     Record newPosition
7:     newDensity  $\leftarrow$  getDensity (newPosition)
8:     if newDensity  $>$  curDensity then
9:       DirectionList  $\leftarrow$  (Upward)
10:      curPosition  $\leftarrow$  (newPosition)
11:      break
12:    else
13:      Move to curPosition
14: return curPosition

```

This algorithm is triggered each time the first UAV reaches a new position while the boundary has not been detected. Once the tentative boundary is found, the UAV moves in four directions to determine if any of these have higher concentration than the one currently observed. If it does, the tentative boundary is discarded and the UAV moves upward. If no higher concentration is found, then the UAV fixes the current boundary position as the anchor point. A 2D visual representation of this search can also be seen in Fig. 3. When the anchor point is fixed, the anchor UAV broadcasts a *Boundary Reached* message to all other UAVs.

B. Fibonacci Hemisphere Formation

When the *Boundary Reached* message is received by each UAV, the agents must then find the target position they need to occupy such that a unit hemisphere is formed. To do so, we modify the original Fibonacci sphere algorithm such that the UAVs only occupy points along the specified orientation, as shown in Fig. 1 (a).

Our proposal is fairly straightforward. Instead of plotting n points on the surface of the sphere, we plot $2n$ points. Each UAV is aware of its own RID and the number of UAVs in the group. When forming the initial sphere, the UAVs calculate the target position for every member in the group centered at the origin. The distance between the anchor UAV's current position and the hypothetical position at the unit sphere is calculated. The calculated point of each UAV is then translated parallel to the line joining the anchor UAV's current and hypothetical positions to find the target position of each UAV.

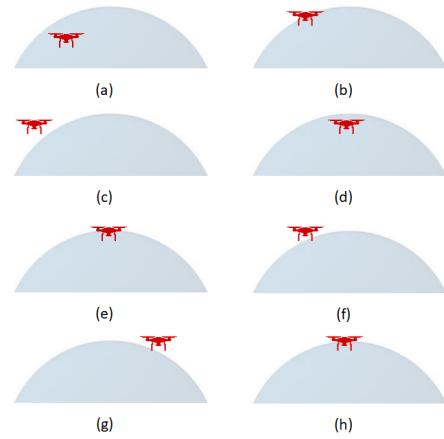


Fig. 3. A 2D representation of hill climbing by the anchor UAV with only left and right operations. The picture shows (a) Arbitrary start position (b) UAV reaches boundary (c) UAV samples concentration to the left, which is lower than current concentration. (d) UAV samples concentration to the right, which is higher than current concentration. (e) UAV moves up. Boundary reached again. (f) UAV samples concentration to the left, which is lower than current concentration (g) UAV samples concentration to the right which is lower than current concentration (h) UAV returns to the boundary point and accepts it.

If any UAV's target position is lower than the base (i.e., z-coordinate of the center) with some minimum safety distance, this mapping is discarded. Instead the UAV picks the next point and checks whether it satisfies the z-coordinate condition. This is done till all n UAVs are placed. The algorithm for Initial Sphere Formation can be found in Algorithm 2. This algorithm is executed by all the UAVs. Lines 1 and 17 – 18 are the parts we proposed to make sure the UAVs form a hemisphere about the anchor.

Once all the UAVs have been assigned a coordinate, they store the *effectiveRID* which was used to calculate their Fibonacci point. In the *Stop-Click* phase each UAV uses its own *effectiveRID* to calculate its target position when rotating or expanding. This allows the UAVs to maintain their relative positions in the hemisphere, while also maintaining the fast, distributed deployment property employed by the original Fibonacci spiral algorithm. However, while testing our data, we noticed that in some cases, it so happens that when the UAVs are near the boundary, one particular UAV will have a target position too close to the ground for safety. This is because the algorithm assigns positions in a greedy manner. This can lead to one UAV being slightly lower than the others in the original placement. So when the hemisphere expands, the lone UAV cannot find a safe place to move. In such a scenario, the UAV aborts the mission and safely exits the group. The positioning of 30 UAVs at different stages throughout the simulation can be seen in Fig. 4.

IV. SPHERICAL DOME PLUME BOUNDARY

Our algorithm showed that a group of UAVs could successfully surround a plume which was centered at the ground and formed a hemispherical dome. This led us to consider ways to relax the requirements such that a more gradual spread of

Algorithm 2 : Initial Sphere Formation

Procedure: *initSphere* (N , *anchorId*, *anchorPos*, *initRad*, *id*, *safetyRad*)

- 1: $points \leftarrow N * 2$
- 2: $\phi \leftarrow \pi * (3 - \sqrt{5})$
- 3: $Y_{anchor} \leftarrow (1 - \frac{anchorId}{points} * 2) * initRad$
- 4: $radius_{anchor} \leftarrow \sqrt{initRad^2 - Y_{anchor}^2}$
- 5: $\Theta_{anchor} \leftarrow \phi * anchorId$
- 6: $X_{anchor} \leftarrow \cos \Theta_{anchor} * radius_{anchor}$
- 7: $Z_{anchor} \leftarrow \sin \Theta_{anchor} * radius_{anchor}$
- 8: $anchorPosAtOrigin \leftarrow (X_{anchor}, Y_{anchor}, Z_{anchor})$
- 9: $center \leftarrow anchorPos - anchorPosAtOrigin$
- 10: $count \leftarrow 0$
- 11: **for** $i \in points$ **do**
- 12: $Y_i \leftarrow (1 - \frac{i}{points} * 2) * initRad$
- 13: $radius_i \leftarrow \sqrt{initRad^2 - Y_i^2}$
- 14: $\Theta_i \leftarrow \phi * anchorId$
- 15: $X_i \leftarrow \cos \Theta_i * radius_i$
- 16: $Z_i \leftarrow \sin \Theta_i * radius_i$
- 17: **if** $Z_i \geq (anchorPos.Z - initRad + safetyRad)$ **then**
- 18: $posAtOrigin_i \leftarrow (X_{anchor}, Y_{anchor}, Z_{anchor})$
- 19: $pos_i \leftarrow center + posAtOrigin_i$
- 20: $posArray.append(pos_i)$
- 21: $count \leftarrow count + 1$
- 22: **if** $count = id$ **then**
- 23: Store i as *effectiveRID*
- 24: $count \leftarrow count + 1$
- 25: **return** *posArray*

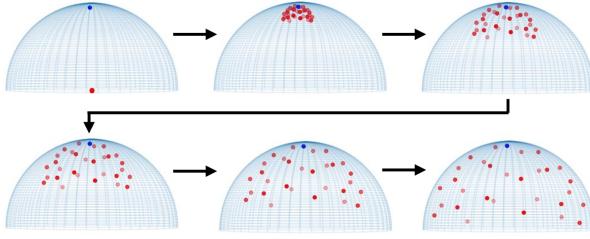


Fig. 4. Stages of mapping a hemispherical plume using 30 UAVs. The blue point represents the anchor position.

the plume, or more accurately, a spherical cap, could also be mapped. A spherical cap is the section of a sphere cut off by a plane, as shown in Fig. 5. The height h refers to the height of the dome from the plane to the apex of the cap. In essence, a hemisphere is a special version of a spherical cap where $h = r$. The algorithm described in the previous section was limited to this special case since we have explicitly placed n points on a sphere formed by $2n$ points, essentially mapping ‘half’ a sphere. To map a spherical cap, we would need to sample more points on the spiral and then choose n points such that the UAVs do not hit the ground. In essence, we would, ‘tighten’ the spiral. While it would be possible to supply the parameter for tightening the spiral based on human

observation, this would not suffice for our goal since our aim is to have UAVs autonomously decide on this value based on their observations.

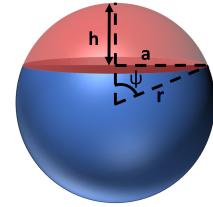


Fig. 5. Spherical Cap with height h , radius r and angle ψ .

To do this, we first modify Algorithm 2 such that Y and Z coordinates are exchanged. We have fixed the Z -axis as the altitude and thus, it is essential that the spiral maps downward from the boundary position. The original version of this algorithm used a spiral which wrapped the hypothetical sphere from left to right, abandoning those points which went below the safety altitude. By changing this orientation, it is easy to select the first n points out of the total number of points generated on the spiral.

The next change is in selecting the number of points to be generated. For a hemisphere, we sample $2n$ points which allows us to restrict the UAVs to the upper half of the spiral. However, for a dome of arbitrary height, it is not immediately evident how many points must be sampled so that the spiral is ‘tight’ enough. Instead, our algorithms allow the UAVs to detect the size of this dome in real time. To do this, we relax the requirement for number of points initially and instead, fall back to the original sphere formation algorithm with n points. The UAV group forms an initial unit sphere by selecting the i^{th} point to occupy on the spiral where i is indicated by its RID and then, Rotates and Expands as required. When any UAV reaches an altitude which is lower than the safety limit, it ‘tightens the spiral’, i.e., updates the number of sampled points to $q * n$, where n is the number of UAVs and

$$q = \begin{cases} 1, & \text{initially} \\ q + 1, & \text{if } q > 0 \end{cases} \quad (1)$$

The UAVs keep occupying the first n points generated as a result of selecting points based on their RID. As the value of q increases, the space between each rung of the spiral reduces, forcing the UAV group to occupy a smaller space. When a UAV updates its q -value, it broadcasts a message with the updated P^{th} position and the new q -value. Other UAVs then update their own q -values. If any UAV has already reached its own P^{th} position with the previous q -value, the UAV then recalculates its P^{th} position and moves to it. This allows the UAVs to maintain synchronization. Fig. 6 shows the spiral for different values of q . At each instance, the spiral wrapping is stopped when every UAV reaches the boundary.

V. SIMULATION & RESULTS

To test the efficacy of our algorithm, we have designed a multi-agent discrete event simulation. The simulation has

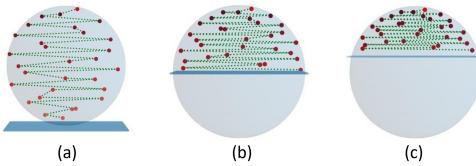


Fig. 6. Placement of 30 UAVs for (a) $q = 1$, (b) $q = 2$ and (c) $q = 3$. The blue rectangle indicates the ground. The portion of the sphere below the ground is hypothetical, included for comparative reference. The green lines connect each successive UAV showing a rough approximation of the spiral.

been written in Java and executed on a Ubuntu 20.04.1 LTS platform. In our design, we have not explicitly modeled message loss. However, to show the practicalities of imperfect communication, we have, instead, simulated message loss through bounded message delay based on an exponential distribution. Messages are delayed by up to $150ms$ which accounts for transmission delay, propagation delay, as well as, delay for retransmission needed when messages are not received. As we have mentioned previously, if some UAV does not receive updates from every other member of the group, it stops and waits for the rest of the agents to catch up. Message Delay accounts for this partial loss of synchronization which is eventually corrected by the group.

We have executed our algorithm for 10, 20 and 30 UAVs for dome shaped plumes with radius $400m$ and altitude $800m$, $400m$, $300m$ and $150m$. For each scenario, we have compared the time taken in the *Stop-Click Phase*. To account for the randomization due to message delay, we have executed each scenario 30 times and then calculated the confidence interval for each scenario with a 95% confidence. The graph can be seen in Fig. 7. In each case, the time taken to surround the hemisphere is the greatest. However, it is interesting to note that for 10 and 20 UAVs, the time taken to surround the $300m$ dome is smaller than taken to surround the $150m$ dome, while it is the opposite in the case of 30 UAVs. This is because for 10 and 20 UAVs, the total number of rounds needed for all the UAVs to reach the boundary (78 and 76 respectively) was significantly larger for the $150m$ dome than the number of rounds needed by 30 UAVs (64). As the size of the UAV group grows, the spiral fits better to the defined boundary region. However, for each case, the total execution time for the plume is still quite fast, taking only about 50s more than the time to wrap the spherical plume ($h = 2 * r$), lending credence to our claim of a rapid deployment.

Next we take a closer look at the time taken for each operation per round. In Fig. 8, we have plotted the time taken by 30 UAVs per round to wrap a $150m$ dome. Time per rounds is the summation of *Stop Time* and *Click Time*. *Stop Time* is defined as the time that the UAV group waits for positional updates from other UAVs in round P and is calculated as the duration of time starting from when the last UAV from the group reaches its P^{th} position to the time when the last UAV receives $n - 1$ P messages. Similarly, *Click Time* refers to the time spent by the UAV in flight. *Click Time* for round P is

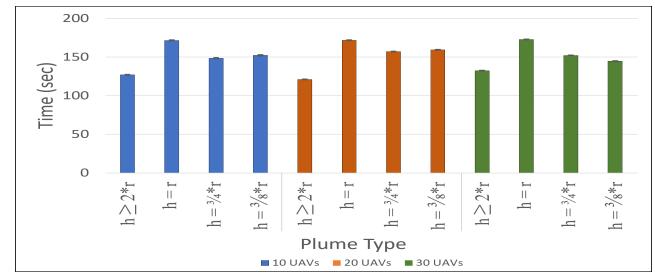


Fig. 7. Time taken by UAVs during *Stop-Click Phase* for different heights. In each case radius $r = 400m$. Height h refers to the distance from the intersection of the ground to the apex of the plume.

calculated as the time when the UAV starts moving to reach its P^{th} target position, to the time that the UAV actually reaches this position. At each round, three possible operations occur – *Expansion*, *Rotation* or *Spiral Tightening*. In the given graph, triangular markers refer to *Tightening* operations while rectangular markers are for *Rotation*. Circular markers, consequently, are for *Expansion*. While *Expansion* and *Rotation* are mutually exclusive operations, *Spiral Tightening* occurs in conjunction with one of the other two. However, since the effect on time due to either *Expansion* or *Rotation* is not significant during a *Tightening* operation, they have not been explicitly shown.

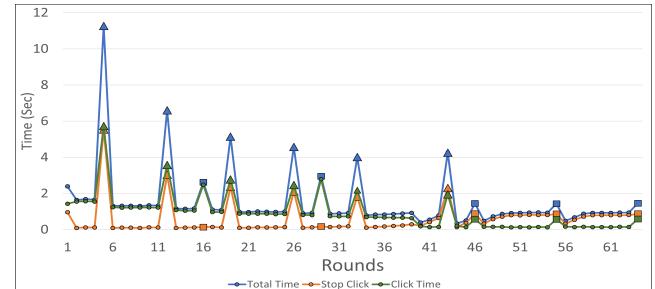


Fig. 8. Time taken by 30 UAVs per round during *Stop-Click Phase*.

As can be seen in the figure, multiple *Tightening* operations occurred while wrapping the plume. It is interesting to note that the time taken for each such operation reduced as the rounds progressed. This is expected, since the maximum distance being travelled decreases as q -value increases. The maximum distance travelled when the spiral reduces from a sphere to a hemisphere is equal to approximately 1.4 times the radius of the UAV group at that round. When reducing further to a smaller dome, this distance is reduced for each successive tightening of the spiral first by $\frac{1}{4}^{th}$, then by half, $\frac{2}{3}^{rd}$ and so on, as seen in Fig. 9. Interestingly, the maximum distance travelled is the same, irrespective of the number of UAVs in the group. This is because the last point occupied by the group is always the same. However, this property is not completely maintained in practice since subsequent *Tightening* operations do not happen on the same radius. The increase in radius also leads to a subsequent increase in movement time, and consequently, waiting time, which leads to a gentler curve.

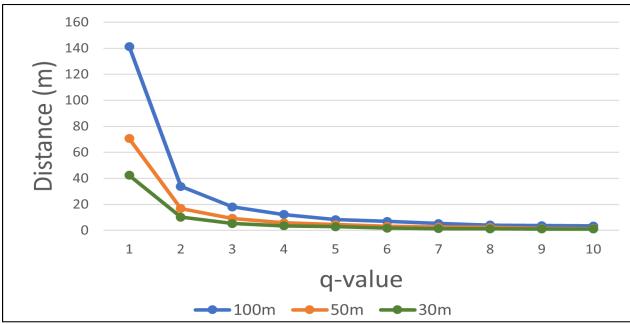


Fig. 9. Maximum distance moved as q -value increases. Blue, Orange and Green lines indicate radius of 100m, 50m and 30m respectively.

Our results also show a few smaller peaks for *Rotation*. However, unlike *Tightening*, we do not see an uptick in *Stop Time*. This is because, when *Tightening* occurs, the UAVs must wait for another round of updates from the UAV group with the updated q -value. No such wait time is required for *Rotation*. An interesting observation, though, is that when the *Click Time* is lower, the *Stop Time* increases. As can be seen from the Fig. 8, time taken in round 40 is considerably lower than in previous rounds. This is because, when the UAVs are all near the boundary, the distance moved per round is reduced to 20% of the previous value which subsequently reduces movement time. This action leads us to notice that, in subsequent rounds, *Stop Time* gradually increases. Our understanding of this phenomenon is that, for larger movement times, a part of the waiting time was offset due to UAVs buffering messages while in flight. This benefit is lost towards the end of the execution as the *Click Time* reduces and the resultant total time is only slightly less than the time obtained for rounds prior to the reduction in distance moved.

VI. CONCLUSION & FUTURE WORK

In this paper, we have designed an algorithm for a multi-UAV group which allows them to coordinate and cooperate with each other in order to complete a task. The task that we have chosen is an extension of the generic Plume Wrapping scenario presented by Chatterjee and Dutta, which we have modified to be able to safely map plumes that stay close to the ground. We have described in detail the modifications we have made to the original algorithm and shown exactly how the various rounds would look. We have also presented evidence validating our claims for a fast, successful execution, and have further shown that safety is preserved when UAVs fly too close to the ground. While we have not considered an explicit message loss model, our delay model is successful in showing how an imperfect communication channel would affect execution times. Our algorithm is robust and is able to complete the mission successfully. We have also shown that the algorithm is scalable and is suited for rapid deployment in a highly changeable crisis scenario.

There are several different directions we wish to explore in the future. While we have considered a specific modification of

the Plume Wrapping algorithm, such that a particular type of mission can be handled, our algorithm does not map exactly to the shape of the plume. In future, we would like to focus on a variety of plume shapes and propose a much more generic approach to this task. We would also like to consider dynamically changing plume boundaries. To track a dynamic plume using the current algorithm, the UAV group would need to restart the algorithm, which is inefficient. Finally, we would also like to delve into a more realistic communication model in a future publication. This would include explicitly modeling message loss with the relevant retransmission messages needed. We would also like to consider situations where plume density may create a highly unreliable channel through the plume, but allow for communication with immediate neighbours. In such cases, multi-hopping of messages will be critical.

REFERENCES

- [1] J. Cho, J. Sung, J. Yoon, and H. Lee, “Towards persistent surveillance and reconnaissance using a connected swarm of multiple uavs,” *IEEE Access*, vol. 8, pp. 157906–157917, 2020.
- [2] X. Liu, Z.-R. Peng, L.-Y. Zhang, and Q. Chen, “Real-time and coordinated UAV path planning for road traffic surveillance: A penalty-based boundary intersection approach,” *International Journal of Control, Automation and Systems*, vol. 20, pp. 2655–2668, July 2022.
- [3] T. Z. Muslimov and R. A. Munasypov, “Multi-UAV cooperative target tracking via consensus-based guidance vector fields and fuzzy MRAC,” *Aircraft Engineering and Aerospace Technology*, vol. 93, pp. 1204–1212, Aug. 2021.
- [4] C. H. Bosanquet and J. L. Pearson, “The spread of smoke and gases from chimneys,” *Transactions of the Faraday Society*, vol. 32, p. 1249, 1936.
- [5] O. G. Sutton, “The theoretical distribution of airborne pollution from factory chimneys,” *Quarterly Journal of the Royal Meteorological Society*, vol. 73, pp. 426–436, July 1947.
- [6] M. Awadalla, T.-F. Lu, Z. F. Tian, B. Dally, and Z. Liu, “3d framework combining CFD and MATLAB techniques for plume source localization research,” *Building and Environment*, vol. 70, pp. 10–19, Dec. 2013.
- [7] J. Sánchez-Sosa, J. Castillo-Mixcóatl, G. Beltrán-Pérez, and S. Muñoz-Aguirre, “An application of the gaussian plume model to localization of an indoor gas source with a mobile robot,” *Sensors*, vol. 18, p. 4375, Dec. 2018.
- [8] P. P. Menon and D. Ghose, “Simultaneous source localization and boundary mapping for contaminants,” in *2012 American Control Conference (ACC)*, pp. 4174–4179, 2012.
- [9] A. Arputha Babu John and R. Dutta, “Cooperative trajectory planning in an intercommunicating group of uavs for convex plume wrapping,” in *2017 IEEE 38th Sarnoff Symposium*, pp. 1–6, 2017.
- [10] B. Chatterjee and R. Dutta, “Studying the effect of network latency on an adaptive coordinated path planning algorithm for UAV platoons,” in *Proceedings of the 2022 workshop on Eighth Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, ACM, June 2022.
- [11] N. Arnold, J. Gruber, and J. Heitz, “Spherical expansion of the vapor plume into ambient gas: an analytical model,” *Applied Physics A*, vol. 69, pp. S87–S93, Dec. 1999.
- [12] FEMA, “Chemical fire in Apex, North Carolina,” USFA-TR 163, US Fire Administration, Apr. 2008. Homeland Security Digital Library.
- [13] A. Sigurðardóttir, J. Barnard, D. Bullimore, A. McCormick, J. Cartwright, and S. Cardoso, “Radial spreading of turbulent bubble plumes,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 378, p. 20190513, Aug. 2020.
- [14] C. Temaneh-Nyah and E. Victor, “Rf radiation exposure levels from the valombola base station, in the faculty of engineering and it vicinity, ongwediva, namibia,” in *2015 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, pp. 27–31, 2015.
- [15] X. Xu, X. Dong, L. Gao, F. Liu, W. Wei, and Z. Liu, “Mixing of a confined two-layer stratified liquid by a bubble plume,” *Heat and Mass Transfer*, vol. 56, pp. 175–182, July 2019.