Space Efficient Airspace Geofence Volume Sizing

Christopher Barkey*
Department of Mechanical Engineering
University of Michigan, Ann Arbor, MI, 48105

Joseph Kim†
Robotics Department
University of Michigan, Ann Arbor, MI, 48105

Ella Atkins‡
Department of Aerospace and Ocean Engineering
Virginia Tech, Blacksburg, VA, 24060

Introduction

Drone geofences are used to keep autonomous drones from flying in areas where they are not allowed to be. These areas include airports, prisons, and sports stadiums. These geofences are three dimensional volumes in space that allow the autonomous drone's navigation system to know when it is about to veer off from its designated area. An example of this is shown in Figure 3 below. The red geofence around the building represents a keep out geofence that the drone is not allowed to enter. The yellow volume represents a keep in geofence for the drone's current flight path.

Fig. 1 Visual example of keep in and keep out geofences[1].

This paper will present methodologies to construct space-efficient airspace geofence volumes around Unmanned

*Undergraduate Student, Department of Mechanical Engineering, University of Michigan
†PhD Candidate, Robotics Department, University of Michigan, Student Member
‡Fred D. Durham Professor and Head, Department of Aerospace and Ocean Engineering, Fellow

Aircraft Systems (UAS) for two specific cases: longitudinal climbing/descending flight paths, and cooperatively controlled swarms for which a provable containment boundary can be defined. Airspace geofencing defines polygon or polyhedron boundaries that partition the airspace into available fly zones (keep-in boundaries) and no-fly zones (keep-out boundaries) to assure aircraft separation and obstacle/terrain avoidance. Geofencing is a key enabler for safe Unmanned Aircraft System (UAS) Traffic Management (UTM). In densely populated low-altitude airspace, UTM must safely and efficiently manage the airspace geofence volumes around different UAS missions. Particularly, UAS operations often include complex flight paths with several climb/descent phases for missions such as package delivery and search and rescue. Constructing spatially efficient geofences around climb/descent paths becomes increasingly important in densely populated airspace to maximize usable airspace for other UAS. For the case of swarm flight/containment control, a single geofence volume can be used to wrap the entire team for air traffic control treatment as a "flight-of-n" vehicles, assuming the controller and connected network are robust. In both cases of climb/descent and swarm flight/containment control, the geofencing problem is to construct spatially efficient airspace volumes wrapping the UAS or swarm throughout its flight trajectory. This paper will extend our previous work [2] in three-dimensional climb/descent geofence by generating parallelepiped airspace geofence volumes with variable ceilings and floors. This paper's parallelepiped geofencing for climb/descent trajectories complements previous work defining efficient airspace geofence volumes for optimal cruise trajectories [1]. This paper extends previous work in single-vehicle geofencing to multi-agent teams following containment control by wrapping this team with a three-dimensional convex hull [3]. Algorithms, case studies, and benchmark comparisons of geofence volume sizings will be presented in the full paper. This research can be used by people creating Unmanned Aircraft System (UAS) Traffic Management (UTM) to better understand the trade off between computational complexity and space efficiency when looking at these types of drone geofences. These systems basically act as the air traffic controller but for unmanned drones. As the number of drones in the sky increase and as we begin to trust them with more autonomous tasks, it becomes very important to ensure that these traffic management systems are using the airspace in an efficient way and that they can handle the amount of drones that will be in the air.

A. Parallelepiped Geofence

A common type of geofence is the climb and descent geofence. These geofences are used to keep in drones as they gain or lose altitude as part of their overall flight plan. As a drone moves through an urban environment, it is possible that it will have to gain or lose altitude many times in order to complete its trip.

Our previous work [2] constructed constant ceiling and floor multiple-staircase geofence (MSG) volumes for UAS climb/descent flight phases. The MSG design creates multiple stairs or "blocks" of geofences that progressively activate and deactivate to assure the UAS will always stay inside at least one MSG volume during its climb/descent. MSG was shown to reduce total airspace volume reserved compared to a single constant floor/ceiling geofence volume enclosing the entire climb/descent. This paper constructs parallelepiped geofence volumes to further reduce geofence airspace

volumes reserved during climb/descent trajectories. Although geofence definition and usage complexities are increased with parallelepiped construction per [4], this paper will show significant airspace volume savings with the parallelepiped. Figure 2 illustrates MSG and parallelepiped volume designs. The parallelepiped geofence is constructed to maintain a minimum safety buffer around the UAS at every point of its climb. Initially, a box representing that safety buffer is drawn around the drone as shown on the right half of Figure 1. Another box is then constructed at the end of the drone's climb/descent. These boxes are then connected by their outside edges to ensure the drone will always be within the safety buffer. Parallelepiped climb/descent geofence construction is formally described in Algorithm 1.

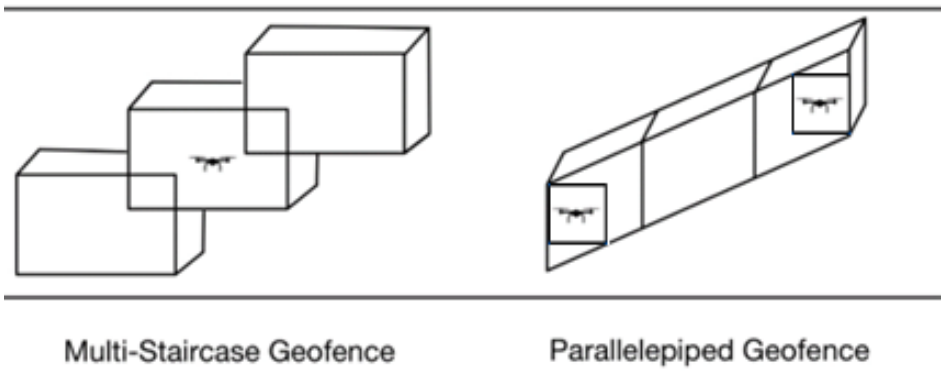

Fig. 2 Multiple staircase geofence (left) and parallelepiped geofence (right) for a steady climbing flight example.

Figure 3 shows preliminary results of a geofence volume sizing comparison between parallelepiped and multiple-staircase geofences in climb/descent. Number of geofence blocks or partitions, flight path angle, and geofence safety buffer size were collectively used to generate geofence volumes. Percentages of volume saved with parallelepiped geofence designs are reported. The safety buffer size is varied from 10m to 100m, and the number of blocks ranged from 5 to 50. For each combination of safety buffer and number of blocks, we generated two geofences, a multi-staircase and parallelepiped, and calculated their volume. As the number of blocks and safety distance decrease, the parallelepiped geofence becomes more spatially efficient compared to the multiple-staircase geofence (shown in yellow). As the number of blocks increases, the MSG approximates a smooth climb boundary resulting in similar volume sizing to the parallelepiped. When safety buffer distance increases, however, parallelepiped geofence efficiency is reduced due to the large amount of extra space created to accommodate large safety buffer height and length. Figure 3 below shows these patterns. In terms of what this means for creating geofences, the parallelepiped geofence is always either more space efficient than the multi-staircase or it uses the same amount of space as the multi-staircase geofence. There is no scenario in which the multi-staircase geofence uses less space than the parallelepiped geofence.

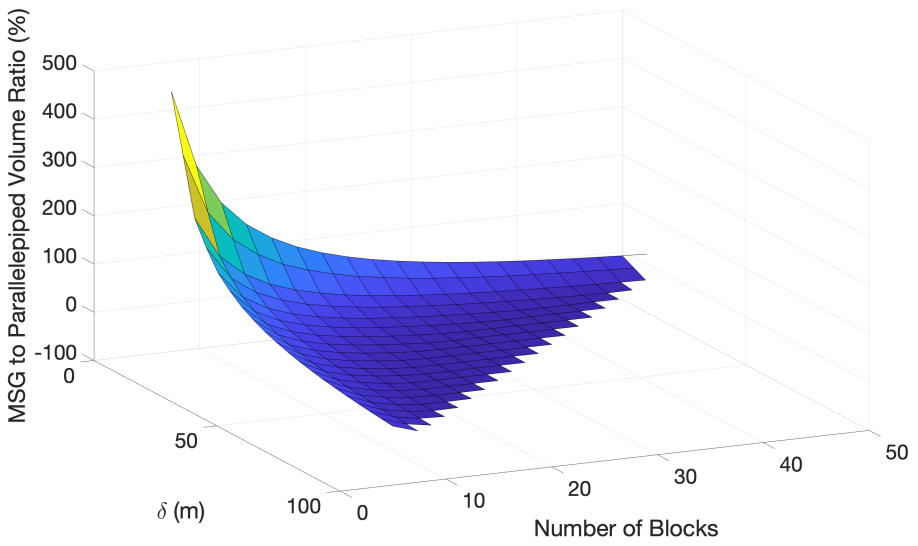

Fig. 3 Comparison of MSG and parallelepiped geofence volumes as a function of number of geofence blocks and safety buffer distances for a 1000m climb with 45° flight path angle.

Algorithm 1 below describes the method of generating the parallelepiped geofence which was described in simpler terms above. The basic concept of this parallelepiped is to generate a volume where at every point along the flight path you can draw a user inputted safety buffer around the drone and no part of it leaves the volume. The most space efficient version of this would use curved surfaces, but that is too computationally expensive to be useful for now. The way the parallelepiped was generated for this project was to take the start and end point of the flight path and calculate two squares that faced each other with that start and end point in the middle of the two squares. You then expand those squares into boxes that surround the start and end points. You then connect those boxes by their outside corners to create the parallelepiped geofence.

Algorithm 1 Parallelepiped Geofence (PG) Construction

Inputs: Departure Point $\mathcal{R}_{start}$, Velocity $\mathcal{V}$, Climb Time $\mathcal{T}$, Number of Geofence Blocks N_{geo}, Safety Distance δ_{sd}

Outputs:3-D Geofence Sets for Climb / Descent $\mathcal{G}$

Algorithm:

$\mathcal{G} \leftarrow$ empty N_{geo} by 2 cell
$\mathcal{R}_{bs} \leftarrow \mathcal{R}_{start}$
$\mathcal{T}_{block} \leftarrow \mathcal{T}_{climb}/N_{geo}$ ◁ calculate time to pass through one geofence
$\mathcal{S} \leftarrow \sqrt{V(1)^2 + V(2)^2 + V(3)^2}$ ◁ calculate the speed
$\gamma \leftarrow atan2(V(3), \sqrt{V(1)^2 + V(2)^2})$ ◁ the angle between the starting and ending points
$T_{sd} \leftarrow \delta_{sd}/\mathcal{S}/cos(\gamma)$

for $i \in N_{geo}$ **do**
    $\mathcal{R}_{be} \leftarrow \mathcal{R}_{bs} + \mathcal{V} * T_{block}$ ◁ calculate the point at the end of the individual block
    $\mathcal{T}_{bs} \leftarrow \mathcal{T}_{block} * i - T_{block} - \delta_{sb}/\mathcal{S}$ ◁ calculate the time at which the block should activate
    $\mathcal{T}_{be} \leftarrow \mathcal{T}_{block} * i + \delta_{sb}/\mathcal{S}$ ◁ calculate the time at which the block should activate
    $\theta \leftarrow atan2(\mathcal{R}_{be}(2) - \mathcal{R}_{bs}(2), \mathcal{R}_{be}(1) - \mathcal{R}_{bs}(1))$ ◁ the angle between the starting and ending points on the xy plane
    $Angle_{rise} \leftarrow atan2(\mathcal{R}_{be}(3) - \mathcal{R}_{bs}(3), \sqrt{(\mathcal{R}_{be}(1) - \mathcal{R}_{bs}(1))^2 + (\mathcal{R}_{be}(1) - \mathcal{R}_{bs}(1))^2})$ ◁ the rise angle
    $z_{sb} \leftarrow \delta_{sd}/sin(\pi/2 - Angle_{rise})$ ◁ safety buffer in the z direction
    $[x_1, x_2] \leftarrow \mathcal{R}_{bs}(1) + \delta_{sd} \times cos(\theta \pm \pi/2)$
    $[x_3, x_4] \leftarrow \mathcal{R}_{be}(1) + \delta_{sd} \times cos(\theta \pm \pi/2)$
    $[y_1, y_2] \leftarrow \mathcal{R}_{bs}(2) + \delta_{sd} \times cos(\theta \pm \pi/2)$
    $[y_3, y_4] \leftarrow \mathcal{R}_{be}(2) + \delta_{sd} \times cos(\theta \pm \pi/2)$
    $[z_1, z_2] \leftarrow \mathcal{R}_{bs}(3) \pm z_{sb}$
    $[z_3, z_4] \leftarrow \mathcal{R}_{be}(3) \pm z_{sb}$
    $\lambda_1 \leftarrow perms([x_1, x_2], [y_1, y_2], [z_1, z_2])$
    $\lambda_2 \leftarrow perms([x_3, x_4], [y_3, y_4], [z_3, z_4])$

    if $Angle_{rise} > 45 degrees$ **then**
        $z_1 \leftarrow \mathcal{R}_{bs}(3)$
        $z_2 \leftarrow \mathcal{R}_{be}(3)$
        $\lambda_1 \leftarrow perms([x_1, x_2], [y_1, y_2], [z_1])$
        $\lambda_2 \leftarrow perms([x_3, x_4], [y_3, y_4], [z2])$
    end if

    $[v, k] \leftarrow$ ConvexHull3D($[\lambda_1; \lambda_2]$) ◁ 3D convex hull algorithm
    $[v, e] \leftarrow RemoveEdges(v, k)$ ◁ removed unnecessary edges from convex hull volume
    $\mathcal{G}(i) \leftarrow [v, e]$ ◁ add the vertices and edges of a single block to the overall data structure
    $\mathcal{R}_{bs} \leftarrow \mathcal{R}_{be}$ ◁ the end of one block is the beginning of the next
end for
return $\mathcal{G}$

Figure 4 shows a comparison of parallelepiped and multiple-staircase geofences over a range of flight path angles γ and safety buffer sizes δ. For this comparison, the flight path angle changes from 0° to 90°, and the safety buffer size varies from 10m to 40m. Results show that the parallelepiped geofence is most efficient relative to the MSG at a flight path angle of 45° and at a small safety buffer distance. This occurs due to MSG space inefficiency at 45°. Small

safety buffers make these spatial inefficiencies more pronounced. At 0° and 90°, MSG and parallelepiped volumes are identical as the geofences became rectangular prisms to wrap horizontal and vertical flight paths. There is no point where the multi-staircase is more space efficient than the parallelepiped. The closer the angle is to 45°, the greater efficiency you obtain with the parallelepiped geofence

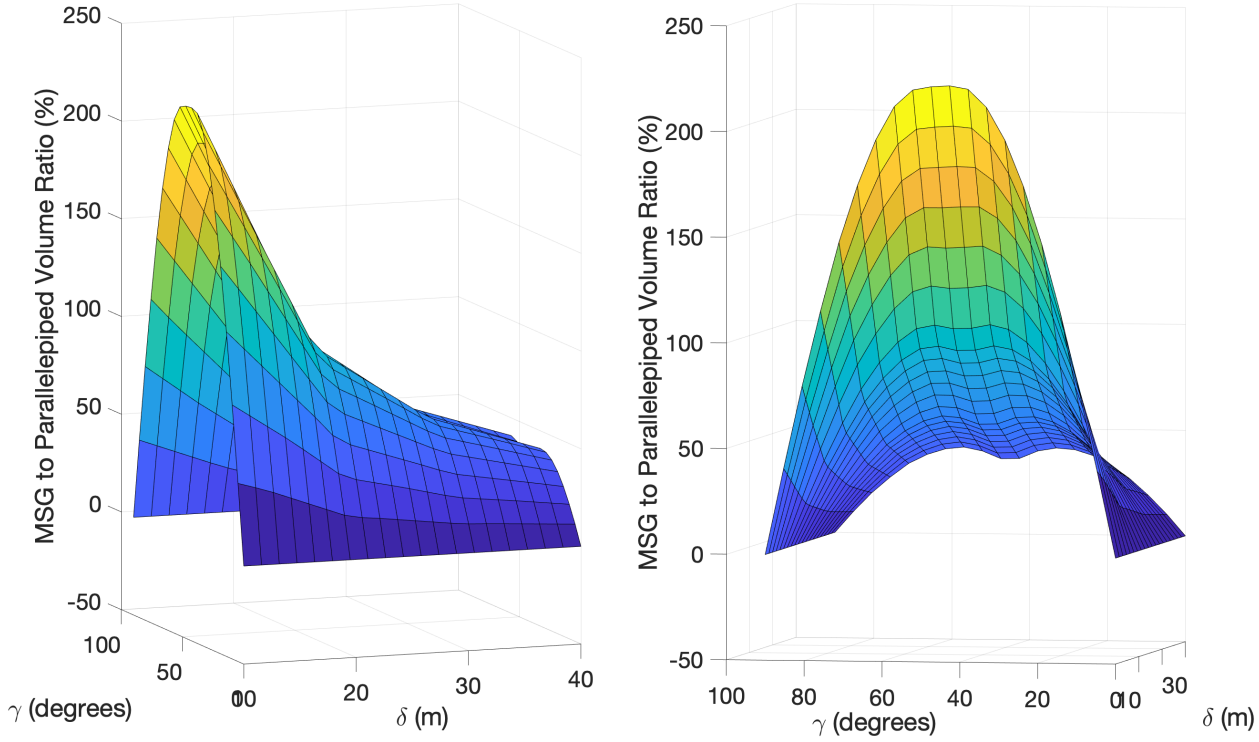

Fig. 4 Comparison of MSG and parallelepiped geofence volumes as a function of flight path angle and safety buffer size for a 1000m climb distance with 10 geofence blocks.

Overall, the parallelepiped geofence is more spatially efficient than the MSG in terms of volume sizing. In its worst case, the parallelepiped results in the same amount of volume required for the MSG at γ of 0° and 90°. MSG spatial efficiency is comparable to parallelepiped efficiency outside of these extremes only for cases in which UAS safety buffer distances are large relative to climb/descent path length. However, the MSG can utilize existing geofence definitions per [4] and is computationally more efficient in generating the volume.

B. Convex Hull Geofence Volume Definition for Swarm Flight with Containment Control

One of the interesting problems with drone geofencing is determining how to create a geofence for several drones moving in formation or in a swarm. There are several ways to do this each with varying levels of computational complexity and space efficiency. A common way to define a swarm or formation is through lead drones. These drones define the outer geometry within which the swarm or formation operates. Going off of this, these multi drone geofences

define their volumes from a user inputted set of lead drones. These are shown with a blue square in Figure 5 below. This research used a regular tetrahedron as the main formation. A regular tetrahedron is a shape with four points equidistant from each other. This was done to allow for easier comparison between input variables.

A containment geofence for a cooperative UAS team can be generated by first creating minimum separation boxes around each UAS in the swarm, and then generating a three-dimensional convex hull containment geofence as shown in Figure 5. A bounding box geofence is generated to compare the volume saved using a more complex but tight convex hull containment geofence. The bounding box (shown in red) was constructed by taking the largest value in the x, y, and z axes to generate a box. Analogous to the parallelepiped geofence, the convex hull geofence volume has more in-plane geometric complexity and also may not have a constant floor and ceiling. However, the volume saved from the containment geofence can be significant. The containment geofence generation algorithm is shown in Algorithm 2.

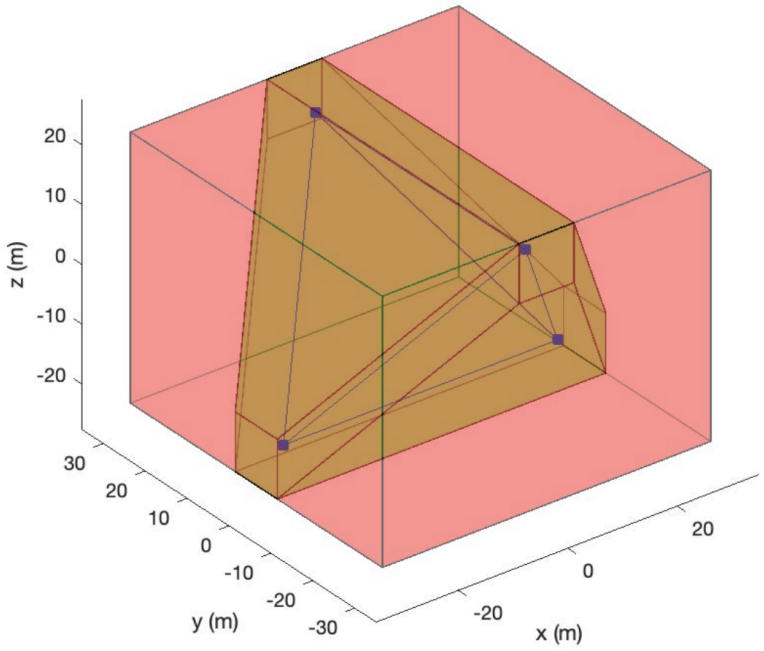

Fig. 5 Containment geofence case study for a four UAS team. A bounding box geofence volume is shown in red, and a convex hull containment geofence is shown in green. Blue squares illustrate the UAS positions.

Algorithm 2 below describes in more detail how to generate a convex hull geofence around several drones. One important thing to note is the addition of a $RemoveEdges$ function at the end of the algorithm. When using a 3D convex hull algorithm, it generally creates surfaces using a matrix of three points to form many triangles. This $RemoveEdges$ function creates solid surfaces from those triangles which lie on the same three dimensional plane. After the construction of these geofences, they look much like the green shape in Figure 6 above.

Algorithm 2 Convex Hull Geofence Construction

Inputs: Drone Positions $\mathcal{D}_{pos}$, Safety Buffer δ_{sb}

Outputs: Vertices $v[]$, Edges $e[]$

Algorithm:

$Vertices_{all} \leftarrow$ empty N x 3 matrix

for $i \in length(D_{pos})$ **do**

  $[x_1, y_1, z_1] \leftarrow D_{pos}(i,:) + \delta_{sb}$ ◁ safety buffer around drone position D_{pos}

  $[x_2, y_2, z_2] \leftarrow D_{pos}(i,:) - \delta_{sb}$

  $Box \leftarrow perms([x_1, x_2], [y_1, y_2], [z_1, z_2])$ ◁ permutation of x,y,z

  $Vertices_{all} \leftarrow$ all vertices in Box added to the larger matrix

end for

$[v, k] \leftarrow$ ConvexHull3D($Vertices_{all}$) ◁ 3D convex hull algorithm

$[v, e] \leftarrow RemoveEdges(v, k)$ ◁ removed unnecessary edges from convex hull volume

Figure 6 shows a volume sizing comparison between containment geofence and bounding box geofence to support a coordinated flight of four UAS. The swarm flight formation in this example case defines a regular tetrahedron shape. A large bounding box geofence and a convex hull geofence were generated from this regular tetrahedron with different input variables. Results show that the containment geofence volume is significantly more efficient compared to the bounding box method given relatively small safety buffer distances and relatively large separations between the four UAS. As the distances between the UAS in a formation increase, the bounding box geofence generates a volume much greater than the necessary volume to contain the coordinated UAS team. Note that this case study is specifically for the regular tetrahedron drone formation shown in Figure 5. Further work needs to be done to more generally compare the airspace volume and computational efficiencies for cooperative UAS team geofencing.

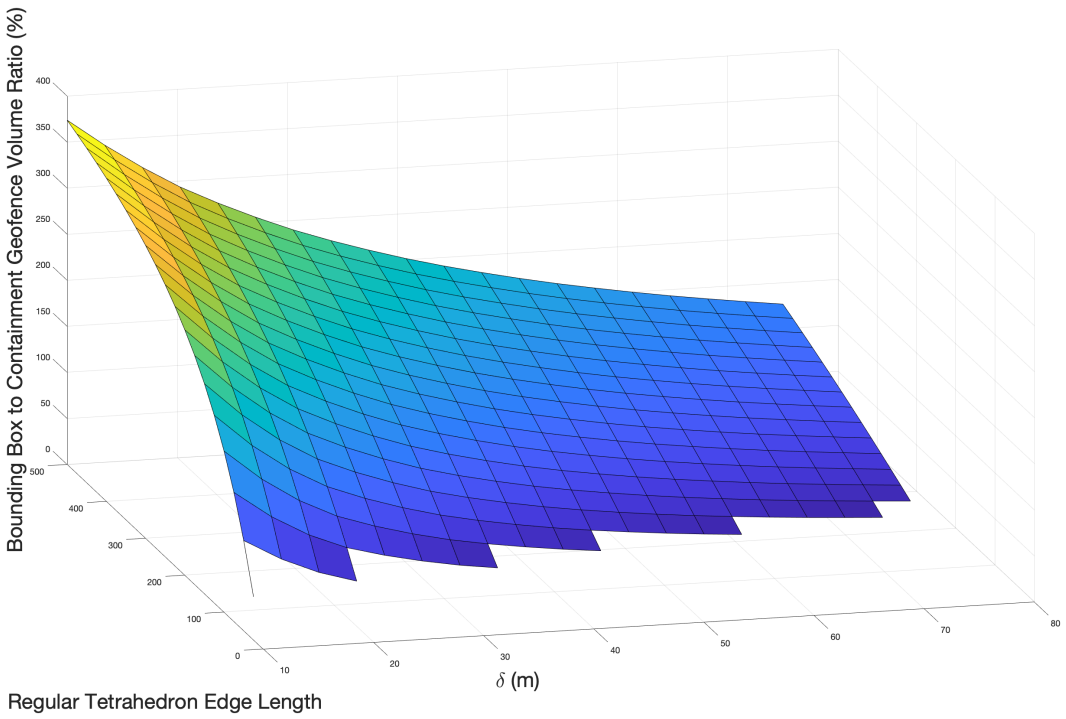

Fig. 6 Comparison of containment and bounding box cooperative UAS team geofencing designs as a function of distances between UAS and safety buffer sizing.

Conclusion

Space-efficient UAS geofencing will minimize the airspace volume reserved in densely populated airspace to maximize airspace availability for other UAS missions. This paper has presented parallelepiped climb/descent and convex hull swarm containment geofencing designs along with a summary of results indicating space efficiency gains as a function of flight path angle, safety buffer, number of geofence blocks, and trajectory length. A regular tetrahedron convex hull was examined as a swarm containment geofence case study. For climb/descent geofencing, we compared a parallelepiped airspace volume with varying ceiling and floor to the multiple-staircase geofence of our previous work[2]. The result showed that the parallelepiped is more spatially efficient than the multiple-staircase geofence in most scenarios. The magnitude of this efficiency is determined by the several inputs for the system, but the parallelepiped geofence consistently used less space than the multiple-staircase geofence. For swarm flight/containment control geofencing, we analyzed a three-dimensional convex hull for enclosing the airspace around a fleet of multi-agent systems. The result showed that our method was more spatially efficient than using simpler geometries to generate the geofence volume around the swarm/formation flight. Much like the parallelepiped geofence, the magnitude of the efficiency of the three-dimensional convex hull geofence compared to the geofence of simpler geometry was determined by several factors including spacing between the multi-agent systems and their safety buffer distances. However, the three-dimensional convex hull geofence consistently used less space than the geofences with simpler geometries. Overall, several of the

original research questions were answered. The parallelepiped geofence was more space efficient than the multi-staircase geofence and the convex hull geofence was more space efficient than the bounding box geofence. New infromation is know known about the scenarios in which these different geofences are more or less efficient. Additional work will need to be done to determine the true comparison between computational complexity and space efficiency. This work provides a basis for that.

Acknowledgments

I would like to thank my honors capstone advisor, Professor Atkins, for the opportunity to pursue this research as well as the knowledge and advice to lead it to fruition. I would also like to thank PhD candidate Joseph Kim for all wonderful help he gave me throughout this research as well as his constant upbeat attitude. In addition, thank you Mrs. Armstrong-Ceron for her years of help and advice in the honors program.

Finally, I would like to thank Collins Aerospace whose grant made this research possible.

References

[1] Kim, J., Liberko, N., and Atkins, E., "Airspace Geofencing Volume Sizing with an Advanced Air Mobility Vehicle Performance Model," *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, IEEE, 2022, pp. 1–8.

[2] Kim, J. T., Mathur, A., Liberko, N., and Atkins, E., "Volumization and Inverse Volumization for Low-Altitude Airspace Geofencing," *AIAA AVIATION 2021 FORUM*, 2021, p. 2383.

[3] Preparata, F. P., and Hong, S. J., "Convex hulls of finite sets of points in two and three dimensions," *Communications of the ACM*, Vol. 20, No. 2, 1977, pp. 87–93.

[4] Stevens, M., and Atkins, E., "Geofence definition and deconfliction for UAS traffic management," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 9, 2020, pp. 5880–5889.