Multi-Agent Systems (MAS) related data analytics in the Hybrid Aerial Underwater Robotic System (HAUCS)

Srijita Mukherjee[a], Bing Ouyang[b], Kamesh Namuduri[c], and Paul S. Wills[d]

[a,c]Department of Electrical Engineering, University of North Texas, Denton, TX, 76203, USA
[b,d]Harbor Branch Oceanographic Institute, Florida Atlantic University, Fort Pierce, FL, 34946, USA

ABSTRACT

This paper investigates Multi-Agent Systems (MAS) related data analytics. MAS is used for modeling complex, decentralized, and real-world tasks such as package delivery by Unmanned Aircraft Systems (UAS), environmental monitoring, precision agriculture, security, disaster management, UAS Traffic Management (UTM) among others. Fish farming is one such area, where the deployment of UAS platforms could drastically improve the current labor-intensive and resource-constraint operations. This research addresses the design of mission control and path planning strategies for UASs deployed on the fish farm. The proposed strategy enables periodic monitoring of mission-critical parameters. A control strategy is designed to address the tracking control of the UAS under wind conditions.

Keywords: HAUCS AUP, Coverage Path Planning, Antipodal points, Shamos Algorithm

1. INTRODUCTION

In the modern era, autonomy plays an important role in all industries. Autonomous vehicles provide scope for real-time applications to be performed without human interference. Multi-Agent System (MAS) is a term used to define multiple autonomous agents working in coordination. MAS is used for modeling complex, decentralized, and real-world tasks such as package delivery by Unmanned Aircraft Systems (UAS), environmental monitoring, precision agriculture, security, disaster management, and UAS Traffic Management (UTM). One such area, where the deployment of UAS platforms could drastically improve the current labor-intensive and resource-constraint operations is aquaculture farms. In this paper, a path planning and path following strategy for the HAUCS platform is developed. HAUCS stands for Hybrid Aerial/Underwater RobotiC System and consists of an unmanned robotic vehicle along-with submerged underwater sensors. This robotic platform has been designed to travel on the water surface and fly in the air[1] for monitoring environmental metrics. A hybrid control strategy is proposed that consists of 1) centralized path allocation to HAUCS autonomous unmanned platforms (AUP) and 2) autonomous trajectory tracking of AUPs under wind disturbances. The optimal coverage path planning is based on the Shamos Algorithm which requires the approximation of the fish farm as a convex polygon. This enables

1. Computation of all antipodal points on the surface of polygon
2. Finding the optimal path for each antipodal pair
3. Selecting the path with the least number of flight lines

The proposed strategy enables periodic monitoring of mission-critical parameters in back and forth pattern (BFP). Sharing of these critical parameters among the agents allows one to estimate the state of the system and to predict potential scenarios of failures. Optimizing the area of coverage on the fish farm by each of the UASs in the presence of dynamic events will be discussed in future works.

Further author information: (Send correspondence to B. Ouyang)
S. Mukherjee: E-mail: srijitamukherjee@my.unt.edu, Telephone: 1 940 390 0620
B. Ouyang: E-mail: bouyang@ fau.edu, Telephone: 1 214 683 7888
K. Namuduri: E-mail: kamesh.namuduri@unt.edu, Telephone: 1 972 639 6340
P. Wills: E-mail: pwills2@fau.edu

2. BACKGROUND

Path planning is a computational problem that deals with finding way-points for the object to move from source to destination. The navigation of robotic vehicles provides great advantages while operating in extreme scenarios. They are also expected to operate in environments filled with obstacles like a crowded building, or target recognition, etc. Therefore, robotic motion planning can be classified as path planning in a static and dynamic environment. It is easier to navigate a robot in a static environment. Graph-based representation is one of the methods to provide a safe path plan. The dynamic environment is filled with obstacles, other humans, and neighboring vehicles and requires intelligent planning techniques. Some of them are 'Ant colony optimization', 'Voronoi diagram ', 'Artificial Potential fields ', and 'Fuzzy logic methods '.[2]

An important factor in the path planning of vehicles is collision avoidance. Artificial Potential Functions (APF) are a set of algorithms that are frequently used in the navigation of autonomous agents,[3].[4] The APF algorithms use the gradient of the potential function to navigate the robot. The concept of artificial potential functions is adopted from potential across opposite electric charges. The push/pull force between elements of equal/opposite charges serves as the basis of APF. The idea is that the robot moves towards a lower energy configuration. The attractive potential leads the robot to the goal, while the repulsive potential helps the robot to avoid obstacles. Applications of APF can be found in path planning[3] and task allocation.[4] APF algorithms have high scalability and low bandwidth use.[5]

The Traveling Salesman Problem (TSP) is another path planning algorithm that generates an optimal path for an AUP to travel across some target locations with a minimum cost of travel. It is also known as a task allocation algorithm. It is an NP-hard problem in combinatorial optimization.[1] If the TSP problem is applied for routing vehicles with limited energy and limited weight carrying capacity, then it is characterized as a Vehicle Routing Problem (VRP). Applications of TSP/VRP algorithms are in package delivery and data collection.[6] Bio-inspired algorithms like Ant Colony Optimization are proposed in[7] for farmland monitoring.

In this paper, we present an optimal path planning approach for the coverage of aquaculture farms. We also implement a motion control strategy for trajectory stabilization under wind conditions. The control strategy is based on sliding mode control (SMC) and adaptive control methods that deal with uncertain wind conditions.

2.1 Coverage Path Planning

The path planning algorithms that can be used for Hybrid Aerial Underwater Robotic System (HAUCS) AUP should be able to organize agents spatially and make decisions cooperatively.[5] Specifically, they include coverage algorithms, task allocation protocols, and motion planning algorithms. Coverage Path Planning (CPP) algorithms determine the optimal route of an AUP to cover a desired area or space while avoiding obstacles. The algorithms also consider environmental conditions such as weather changes. The agents are instructed to make continuous and sequential movements without overlapping their paths. CPP is generally used in mapping, crop monitoring, and land assessment.[6] Approaches to CPP are categorized as randomized and complete. A randomized approach does not consider the geographical information of the farm and, therefore, takes a long time to cover the whole area. On the contrary, in a complete approach, the coverage area is decomposed into cellular regions, and an optimal path is traversed to cover all the cells. Cellular decomposition and optimal path searching are few other techniques of CPP[8].[9]

A baseline algorithm based on a hybrid control system has been developed to control and regulate the HAUCS framework. The architecture of the hybrid controller is shown in Fig. 1. This includes a central server, operators, and individual AUPs assigned for each segment of the farm.

The goal of HAUCS mission control is to provide path plans and maintain the traffic of deployed AUPs within the context of UAS traffic management. The central server in the control architecture provides information about weather and distress calls from ponds to the operators. The server also tracks and monitors the movement of AUPs in the deployed region to make sure that their paths do not intersect. Both the central server and operators can exchange information. Each server is connected to multiple operators. The operators perform flight/path planning procedures and send instructions to the AUPs. The AUPs communicate with each other and operate in a decentralized manner. The HAUCS AUPs are assumed to be homogeneous. Each HAUCS AUP will be assigned to a home station and operate in a known environment with unpredicted variables (e.g., weather, moving obstacles, and pond conditions).

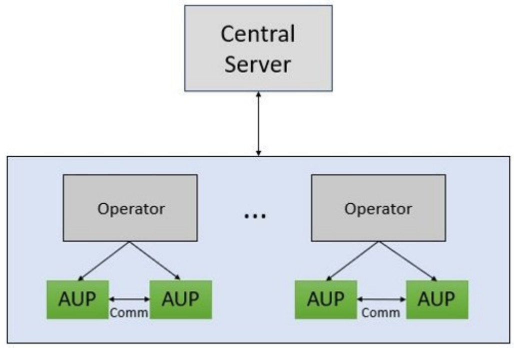

Figure 1. Hybrid Control Scheme

3. PROBLEM DEFINITION

In this section, we develop an optimal path planning strategy for covering the target area of fish farms. The fish farm is assumed to be a convex polygon, $P = (V, E)$ with the ends of the farm as vertices $V = 1, ...n$ and the sides as edges, $E = (1, 2), ..., (n, 1)$. For complete coverage of the farm, a back and forth pattern (BFP) of the flight plan is desired. Here, the AUP would take off at x_s, cover the assigned area, and land at x_e. The final flight path includes take-off and landing locations along with the way-points given by, $W = (x_s, x_0, ..., x_m, x_e)$. The goal is to select the optimal path, which has the least number of back and forth flight lines. The first step in the strategy is to find the ideal direction of the BFP from the take-off to the landing location. The path corresponding to the minimum width of the polygon is the ideal direction for the coverage. In this context, a concept from computational geometry is found to be useful. The usage of antipodal points in deciding the best path is key to the optimal coverage of the polygon. The way-points generated by the path planner require the computation of all pairs of antipodal points on the surface of the polygon.

```
Data: V,x_s,x_e
Result: W
A ← AntipodalPair(V)
c ← ∞
foreach (i, j) ∈ A do
    path ← Bestpath(V,i,j)
    if cost(x_s, path, x_e) < c then
        W ← (x_s, path, x_e)
        c ← cost(W)
    end
end
```

Algorithm 1: Optimal Path Planning Algorithm

The algorithm then finds the best path for each antipodal pair. The best path is the one with the least back and forth flight lines, corresponding to the minimum width of the polygon. Algorithm 1 selects the path having the lowest cost, with a computational complexity of O(n).

3.1 Antipodal Points

The concept of antipodal points was first proposed in Shamos's Thesis.[10] Antipodal points are those vertices of a convex polygon that admit parallel lines of support such that the whole polygon lies between the two infinite parallel lines. Such vertices of a convex polygon are called antipodal pairs as shown in Fig. 2. The parallel lines of support can either pass through a vertex or lie on an edge of the polygon. Here, lines C and D represent the parallel lines of support passing through the antipodal pair (p_1, p_4). Other pairs of antipodal points shown in

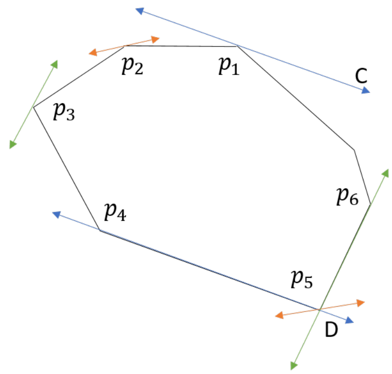

Figure 2. Antipodal Points on Convex Polygon

the figure are (p_2, p_5), $(p_{3,6})$ and, (p_1, p_5). Algorithm 2 describes the Shamos algorithm to compute all the sets of antipodal points for a given convex polygon.

```
Data: V
Result: A
/* Find initial antipodal pair by locating vertex opposite p_1 */
i← 1;j← 2
while (angle(i,j) < π) do
|   j ← j + 1
|   current ← i
|   A ← (i,j)
end
/* Loop on j until whole polygon is scanned */
while j≠ n do
|   if (angle(current,i+1) < angle(current,j+1)) then
|   |   j ← j + 1
|   |   current ← j
|   else
|   |   i ← i + 1
|   |   current ← i
|   end
|   A ← (i,j)
end
/* On parallel edges */
if (angle(current,i+1) = angle(current,j+1)) then
|   A ← (i+1,j)
|   A ← (i,j+1)
|   A ← (i+1,j+1)
|   if current = i then
|   |   j ← j + 1
|   else
|   |   i ← i + 1
|   end
end
```

Algorithm 2: Shamos Algorithm

3.2 Optimal Coverage Algorithm

The antipodal pair computation gives us a set of antipodal points (i, j) for the convex polygon. The goal of the optimal coverage strategy is to find a path that traverses the minimum width of the polygon. The minimum width is traced by considering the pair of antipodal points with the shortest distance between them.[11] A block diagram of the solution approach for the whole process is shown in Fig. 3

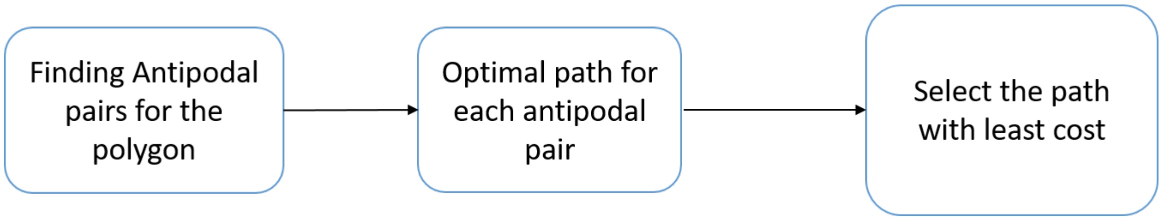

Figure 3. Sequence of Operations

The parallel lines of support are rotated on the surface of the polygon to find the best path for a given antipodal pair, as described in algorithm 3. It resembles a caliper measuring the edges of the polygon as it rotates, hence known as 'rotating caliper algorithm', as shown in Fig. 4. For instance, let the base edges be defined as $(b_1, b_1 + 1)$ and $(b_2, b_2 + 1)$.

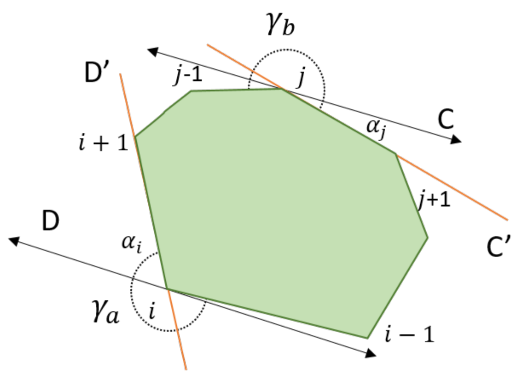

Figure 4. Rotating Caliper Illustration on Convex polygon

Firstly, the lines of support (or the caliper) are rotated in the clockwise direction to find the base edge $(b_1, b_1 + 1)$. Let lines C and D be the parallel lines of support passing through antipodal pair (i, j). Both the lines C and D are rotated until they touch the next adjacent vertex $(i+1)$ and $(j+1)$ respectively. The line that first touches the adjacent vertex, is assigned as the base edge of the polygon. This can be found by comparing the angles between lines of support and the adjacent vertices, α_i, and α_j. The minimum of these two angles reveals the first base edge. This gives the first path from the base edge $(b_1, b_1 + 1)$ to the antipodal pair i.

Secondly, the lines of support are rotated in the counterclockwise direction to reveal the second base edge of the polygon $(b_2, b_2 + 1)$. To measure the angles, a line $B1$ lying on the first base $(b_1, b_1 + 1)$ is considered. The angle of rotation of the lines of support from previous vertices $(i - 1)$ and $(j - 1)$ is compared. The lowest of the angles γ_i and γ_j gives us the second base edge. Here a path is formed from the second base $(b_2, b_2 + 1)$ to the opposing vertex, j.

In this case, path 1 constitutes first base, $(b_1, b_1 + 1)$ and vertex i. Path 2 starts from second base $(b_2, b_2 + 1)$

until vertex j. The path with the smallest distance is selected as the best path for a given antipodal pair.

Data: V,(i, j)
Result: path
if *angle*$(i, j) <$ *angle*(j, i) **then**
  $b_1 \leftarrow j$
  $a_1 \leftarrow i$
else
  $b_1 \leftarrow i$
  $a_1 \leftarrow j$
end
$\beta \leftarrow$angle(b_1, a_1) -π
$\gamma_{b_1} \leftarrow$angle(b_1-1,b_1)
$\gamma_a \leftarrow$angle(a_1-1,a_1) - β
if $\gamma_{b_1} < \gamma_a$ **then**
  $b_2 \leftarrow (b_1 - 1)$
  $a_2 \leftarrow a_1$
else
  $b_2 \leftarrow (a_1 - 1)$
  $a_2 \leftarrow b_1$
end
if *dist*$(b_1, a_1) <$ *dist*(b_2, a_2) **then**
  $W \leftarrow$getpath$(b_1, b_1 + 1)$
else
  $W \leftarrow$getpath$(b_2 + 1, b_2)$
end

Algorithm 3: Optimal Algorithm for Best Path

Finally, after computing the best paths for all sets of antipodal points, $A = (i_1, j_1), (i_2, j_2), ..., (i_n, j_n)$, the path with the lowest cost is selected for the optimal coverage of a convex polygon. The back and forth pattern (BFP) starts from the base edge and ends at the opposing antipodal point. An example of area coverage is shown in Fig. 5. The BFP flight lines are parallel to the lines of support of the convex polygon. The computational complexity of the given algorithm is $O(n)$.

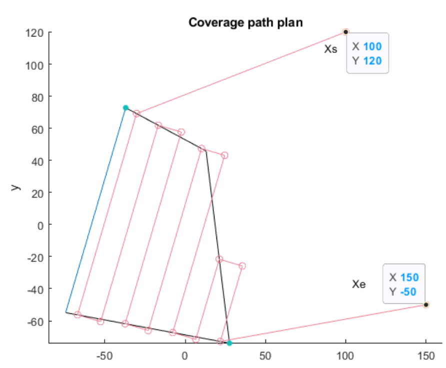

Figure 5. Generated BFP flight path

4. AUP CONTROL STRATEGY

In this section, the trajectory control of AUP in the presence of wind disturbances is presented.[12] Initially, we assume that a central operator computes the flight plan for the AUPs offline. The flight path generated by the algorithm is then assigned to the AUPs ready to cover the fish farms. The path is represented as a series of way-points, $W = (x_s, x_0, ..., x_m, x_e)$, including the starting and ending location. We assume the shape of the farm as a convex polygon as shown in Fig. 6. The whole area is split into three sub-polygons and manually assigned to three AUPs for coverage. The way-points of the three AUPs covering the polygons in a back and forth pattern (BFP) is shown in different colors. The blue, red and, green colors display the path traversed by AUP 1, 2, and 3 respectively.

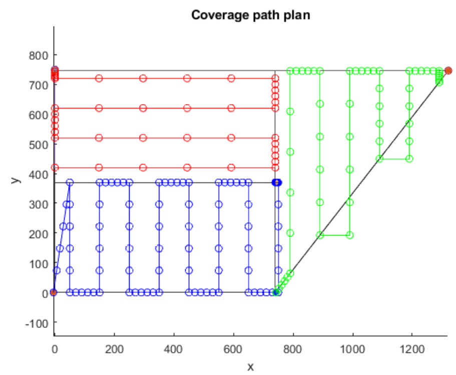

Figure 6. Coverage of Fish Farm

4.1 Kinematic Model of AUP

The kinematics of the AUP is modeled as

$$\dot{p}_i = v_i \tag{1}$$

where, the p_i is the position, and $v_i = (\dot{x}, \dot{y})$ is the translational velocity of each AUP. In the presence of wind, the disturbed kinematic model is given as

$$\dot{p}_i = v_i + W_i \tag{2}$$

where, $W_i = w_0 + \Delta w(t)$, w_0 is the constant slow time-varying wind in the region, and $\Delta w(t)$ is the time-varying wind gusts. In practical scenarios, the weather in the fish farm locations may not always be available. This requires the use of adaptive control, which deals with estimated wind velocities, $\tilde{W}_0 = \hat{W}_0 - W_0$. Similarly, the wind gusts over the fish farms can also create some nonlinear uncertainties. To control the motion of AUP disturbed by wind gusts, a sliding mode control (SMC) method is used. A sliding mode control drives the system states to a sliding surface,[13] where it keeps the system states in the neighborhood of the sliding surface. Normally, the error in the system states is taken as the sliding surface. A general model for a sliding surface is given by:

$$\sigma = (\frac{d}{dt} + c)^k e \tag{3}$$

where, $\sigma(x) : R^n \rightarrow R$ is a scalar function of the system state,[13] c is a positive constant and, $k = n - 1$ is the relative degree between output and input variables. Substituting $k = 0$ in equation (3) gives the position error, $\tilde{e} = e - e(t)^d$ which is taken as the sliding surface. The sliding surface, X is denoted as X1 and X2 in x and y-direction.

$$\begin{pmatrix} X1 \\ X2 \end{pmatrix} = \begin{pmatrix} \tilde{e}_x \\ \tilde{e}_y \end{pmatrix} \tag{4}$$

The control input applied to each AUP, i, for the trajectory control is given by

$$v_i = -[K_1 X + K_2 Sgn(X) + \hat{W}_0 - \dot{p}^d] \tag{5}$$

Here, K_1 and K_2 are positive-definite diagonal matrices, Sgn is the sign or signum function and, $\hat{W}_0$ is the estimated wind vector. The Lyapunov function for the system is given as:

$$V = \frac{1}{2}(X^T X + \tilde{W}_0^T A^{-1} \tilde{W}_0) \tag{6}$$

Here, A is a positive-definite gain matrix. On substituting the control input, the derivative of Lyapunov function obtained is negative semi-definite.[12] Since the path followed by AUP is time-independent, the closed-loop control system is autonomous. Therefore, the state vector of the system is asymptotically stable and the AUP converges with the desired trajectory within finite time.

5. RESULTS AND DISCUSSIONS

In this paper, we have presented the trajectory control of an AUP during optimal coverage of an aquaculture farm. The optimal path plan for the coverage of the farm has been implemented based on Shamos Algorithm. The usage of the rotating caliper algorithm to yield pairs of antipodal points allows computation of the most cost-effective path with a complexity of O(n). Here, the reference path following and trajectory stabilization of AUP is done autonomously.

Simulation of trajectory correction in the presence of wind is performed in MATLAB as shown in Fig. 7. For better visualization, we have inserted 4 observation way-points between every two main way-points. The main way-points are numbered at (1,6,11,16,21,26,31,36,...), while the observation points are located in-between (2 to 5, 7 to 10, 12 to 15, 17 to 20,..).It should be noted from Fig. 7, that, despite the drift due to wind at main way-point 5, the AUP tracks back to the next main way-point. The observation points between main way-points 5 and 7 provide a measure of convergence to desired path plan.

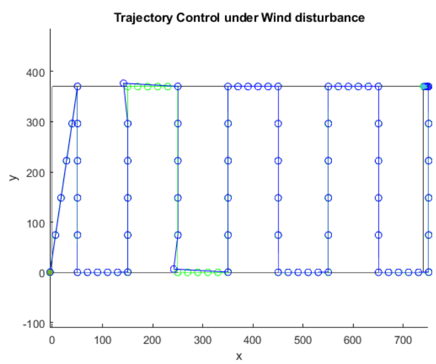

Figure 7. Trajectory Control under Wind conditions

The path re-routing by AUP when wind drift is added to main-waypoints 4 and 10, is shown in Fig. 8.

6. CONCLUSION

This paper presents a motion control model of HAUCS platforms for covering the aquaculture farms. Optimal path computation using the Shamos algorithm provides a back and forth pattern (BFP) of flight lines. Simulation of the trajectories of autonomous HAUCS platforms is implemented using MATLAB. The model is also tested under 2D wind conditions, and the simulation results confirm the trajectory stabilization. In the future, the later stages of the path planning will be proposed:

1. Area assignment of AUPs to cover the fish farm
2. Coverage planning of AUPs in a cooperative fashion

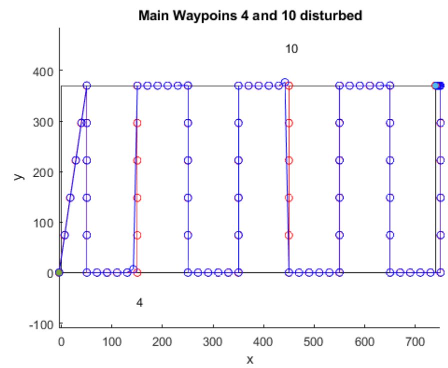

Figure 8. Trajectory tracking at main way-points 4 and 10

3. Collision Avoidance in dynamic environment

To cope with severe weather, a control method for the third type of waypoint – protective waypoints will be presented.[1] For example, upon the detection of potential for strong winds, the control center can update the waypoints to protective waypoints to allow the HAUCS AUPs to take evasive action such as operating as USVs. The control center can restore the waypoint status at a later time when conditions return to normal.

7. ACKNOWLEDGMENTS

This work was supported in part by the US Department of Agriculture, National Institute of Food and Agriculture through the National Robotics Initiative 2.0 (NRI-2.0), under Grant 2019-67022-29204

REFERENCES

[1] Ouyang, B., Wills, P. S., Tang, Y., Hallstrom, J. O., Su, T. C., Namuduri, K., Mukherjee, S., Rodriguez-Labra, J. I., Li, Y., and Den Ouden, C. J., "Initial development of the hybrid aerial underwater robotic system (haucs): Internet of things (iot) for aquaculture farms," *IEEE Internet of Things Journal* , 1–1 (2021).

[2] Mahajan, B. and Marbate, P., "Literature review on path planning in dynamic environment," (2013).

[3] Koditschek, D. E. and Rimon, E., "Robot navigation functions on manifolds with boundary," *Advances in Applied Mathematics* **11**(4), 412–442 (1990).

[4] Weigel, T., Gutmann, J. ., Dietl, M., Kleiner, A., and Nebel, B., "Cs freiburg: coordinating robots for successful soccer playing," *IEEE Transactions on Robotics and Automation* **18**(5), 685–699 (2002).

[5] Rossi, F., Bandyopadhyay, S., Wolf, M., and Pavone, M., "Review of multi-agent algorithms for collective behavior: a structural taxonomy," *IFAC-PapersOnLine* **51**(12), 112–117 (2018). IFAC Workshop on Networked Autonomous Air Space Systems NAASS 2018.

[6] Xie, J., Garcia Carrillo, L. R., and Jin, L., "Path planning for uav to cover multiple separated convex polygonal regions," *IEEE Access* **8**, 51770–51785 (2020).

[7] Yang, J., Wang, X., Li, Z., Yang, P., Luo, X., Zhang, K., Zhang, S., and Chen, L., "Path planning of unmanned aerial vehicles for farmland information monitoring based on wsn," in [*2016 12th World Congress on Intelligent Control and Automation (WCICA)*], 2834–2838 (2016).

[8] Galceran, E. and Carreras, M., "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems* **61**(12), 1258–1276 (2013).

[9] Choset, H., "Coverage for robotics - a survey of recent results," *Ann. Math. Artif. Intell.* **31**, 113–126 (10 2001).

[10] Shamos, M. l., *Computational Geometry*, PhD dissertation, Yale University (1978).

[11] Vasquez Gomez, J. I., Melchor, M. M., and Herrera Lozada, J. C., "Optimal coverage path planning based on the rotating calipers algorithm," in [*2017 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*], 140–144 (2017).

[12] Escareno, J., Salazar, S., Romero, H., and Lozano, R., "Trajectory control of a quadrotor subject to 2d wind disturbances," *Journal of Intelligent Robotic Systems* **70** (04 2012).

[13] DeCarlo, R. and Zak, S., "A quick introduction to sliding mode control and its applications 1," (2008).