Cooperative Surveillance in GPS-denied Environments with deployed Sensor Networks

Rohith Boyinine[1], Dipin Nair[2]

***Abstract*—In this paper, we solve the surveillance problem of a given hostile area, in GPS-denied scenario by deploying multiple vehicles. Vehicles estimate their states collectively by sharing sensor information with each other, known as Cooperative Localization in literature. For the system to be observable, a path to two landmarks is required. Since deploying landmarks is not a pliable solution all the time, we deploy another team of vehicles that act as a sensor network by maintaining connectivity to landmarks outside the environment, to maintain localization accuracy. Since the overall optimization problem is NP-hard, we divide the problem into three sub-problems: (a) Maximizing surveillance area using Particle Swarm Optimization, (b) Solving Travelling Salesman Problem for efficient vehicle routing and (c) developing a control strategy for the sensor network to maintain graph connectivity. We present simulation results using MATLAB/Simulink**

***Index Terms*—Swarm Robotics, Evolutionary Algorithms, Particle Swarm Optimization, Ant Colony Optimization, Genetic Algorithm, Sensor Network, Graph connectivity**

I. Introduction

Usage of multiple Unmanned Vehicles (UVs) over a single UV is gaining more precedence as it makes the system more robust and efficient [1]. Having multiple vehicles in a mission can be helpful in scenarios like border patrol [2], surveillance, exploration, disaster relief management etc., because of parallel search and computation power available to the system. Localization is the most fundamental aspect of autonomous navigation of such UV systems. Several localization strategies are dependent on the Global Positioning System (GPS). However it is known that GPS can work efficiently only when the vehicles is in direct line-of-sight with atleast four satellites and hence it be unreliable in urban environments, canyons, poor weather conditions [3]. GPS signals can also be jammed [3] or spoofed [4] in hostile environments. Hence, sensors like cameras [5], [6], LiDAR [7], radar, infrared [8], ultra-sonic sensors [9] are used extensively to aid autonomous navigation. UVs use these sensors to obtain information from the environment like range/bearing information from known locations or "landmarks" for decision making and path-planning.

It has been shown that each vehicle should have access to atleast two known landmarks [10] for navigating precisely in an environment without GPS. Sharma *et. al* in [11] extended this work and showed that when multiple vehicles share sensor information with each other for collective state estimation, only two landmarks are sufficient for the entire group of vehicles. This process of collective state estimation is known as Cooperative Localization (CL) in literature. Although CL offers several advantages, it has it's own set of challenges pertaining to sensing and communication limitations.

Maintaining connectivity of the network consistently with landmarks can prove to be very challenging. Several studies tried to address this issue. Dutta *et. al* in [12] used dynamic graph connectivity to address the problem of maintaining connectivity and sharing information. Sharma *et. al* in [13] used greedy optimization technique to maintain graph connectivity and improve observability of the system. Chakraborty *et. al* in [14] studied the effect of switching sensing topologies on localization accuracy, and proved that if connectivity to two landmarks in maintained over a time duration, instead of all the time, the system is observable. However, most missions involve several other tasks along with maintaining connectivity. Hence, sub-tasks and path-planning must also be taken into account.

When surveilling hostile unknown environments and in harsh weather conditions, it is difficult to find known landmarks in the environment and deploying new landmarks is not feasible all the time. Deploying a team of vehicles as a sensor network to maintain connectivity from known landmarks and surveilling vehicles can be a feasible approach and is more robust to landmark installations at all locations. This approach can also be useful for designing urban air navigation frameworks.

In this work, we solve the search area optimization problem described in [15] where, Particle Swarm Optimization (PSO) is used to place the waypoints optimally in the search area, so that search area is maximized and overlap area is reduced. After the optimal locations of waypoints in the environment are found, the waypoints are distributed to the group of surveillance team randomly. Once each vehicle receives its set of targets to be visited, Ant-Colony Optimization (ACO) is used to find the optimal sequence to visit each target. We use Genetic Algorithm (GA) coupled with receding horizon approach, for path-planning of the vehicles in the sensor network. (See Fig. 1). These evolutionary algorithms are proved to be more robust and scalable in finding global optimal solutions compared to other techniques, justifying their use for our problem. Therefore our objectives for this project can be outlined as follows -

- Use PSO to determine the optimal location of waypoints in the environment.
- Allocate these waypoints randomly to all the vehicles, and solve the Travelling Salesman Problem (TSP) to find

[1] Rohith Boyinine is a PhD student in the Department of Aerospace Engineering and Engineering Mechanics at the University of Cincinnati. email: boyinirh@mail.uc.edu

[2] Dipin Nair is an MS student in the Department of Mechanical and Materials Engineering at the University of Cincinnati. email: nairdk@mail.uc.edu

optimal route for each vehicle

- Use Genetic Algorithm to determine optimal paths for the vehicles in the sensor network to maintain graph connectivity,

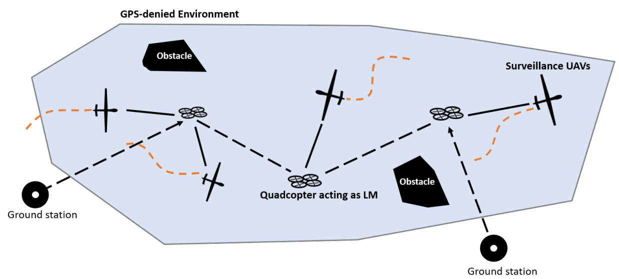

Fig. 1: Surveillance area

The rest of the paper is organized as follows - Section II describes the system architecture and problem framework. Section III describes the Search Area Optimization problem, Vehicle routing problem and Sensor network control problem methodologies in detail. Simulation results are presented in Section IV and we conclude the paper by providing some insights and directions for future work in Section V.

II. System Architecture

Consider m vehicles ($i = 0, 1, 2 \ldots, m-1$) surveilling a hostile environment. These vehicles are aided by another team of n vehicles, that form a sensor network and help in maintaining connectivity with known locations or landmarks as shown in the figure.

We assume that all the vehicles fly in a horizontal plane. Hence unicycle based motion model can be used to describe equations of motion of the system. Let $[x_i, y_i]$ be the inertial position vector and ψ_i be the absolute heading direction of Vehicle-i. Let lm_{x_i} and lm_{y_i} be the X and Y positions of landmarks. Let forward speed of the vehicle be v_i and ω_i be the turn-rate with respect to the down-axis. The equations of motion are given by eq. (1).

$$\begin{bmatrix} \dot{p}_{n_i} \\ \dot{p}_{e_i} \\ \dot{\psi}_i \end{bmatrix} = \begin{bmatrix} v_i \cos \psi_i \\ v_i \sin \psi_i \\ \omega_i \end{bmatrix} \tag{1}$$

The vehicles are equipped with IMU that gives speed and turn-rate of the vehicle, along with range sensors that can measure distance from other vehicles and landmarks when in sensing range. The range measurements between vehicles and landmarks is given in eq. (2) and eq. (3) respectively.

$$\rho_{v_{ij}} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2}$$

$$\rho_{lm_{ij}} = \sqrt{(x_i - lm_{x_j})^2 + (y_i - lm_{y_j})^2} \tag{3}$$

Vehicles collectively estimate their states using Extended Kalman Filter (EKF) by sharing sensor information with each other. The detailed system architecture is shown in Fig.

III. Methodology

In this section, we describe the details of implementation of solution techniques and algorithms. We describe the search area optimization using PSO in Section III-A , TSP using ACO in Section III-B, and Sensor Graph Connectivity using GA in Section III-C.

A. Search Area Optimization

From Section II, we have m Unmanned Aerial Vehicles (UAVs) surveilling an environment with an area A_T. We assume that each UAV is equipped with a downward facing camera, that can cover a circular area of radius r on the surface of earth. Hence, when flying over a waypoint j, the area covered by the vehicle i is given by

$$A_{ij} = \pi r^2 \tag{4}$$

Therefore, when flying through a sequence of waypoints WP_i allocated to vehicle i, total area covered by the UAV can be approximated as

$$A_i = \bigcup_{j \in WP_i} A_{ij} \tag{5}$$

and the total area covered by all the UAVs is given as

$$A_C = \bigcup_{i=1}^{m} A_i \tag{6}$$

Therefore, area covered is dependent on location of the waypoints. No. of waypoints required for covering the entire area is given by

$$n_{WP} = ceil(\frac{A_T}{A_{ij}}) \tag{7}$$

The percentage of the total area covered A_C with respect to total coverage area A_T should be maximized by the distribution of waypoints. Therefore, the cost of coverage is given by

$$CF_c = 1 - \frac{A_C}{A_T} \tag{8}$$

If the waypoints are placed poorly, it can result in large overlap areas between waypoints in some areas and no coverage at all in other areas. Hence, we need to minimize overlap area percentage. The overlap area A_O is given by

$$A_O = n_{WP} A_{ij} - A_C \tag{9}$$

and the cost for penalizing overlap area is given by

$$CF_o = \frac{A_O}{n_{WP} A_{ij}} \tag{10}$$

In PSO, since each particle tends to move to explore the area, it is possible that some waypoints fall outside the feasible area. Hence, we introduce another term that penalizes infeasible points. The cost function for penalizing infeasible points is defined as

$$CF_f = \frac{n_{wp_{if}}}{n_{WP}} + n_{oc} \tag{11}$$

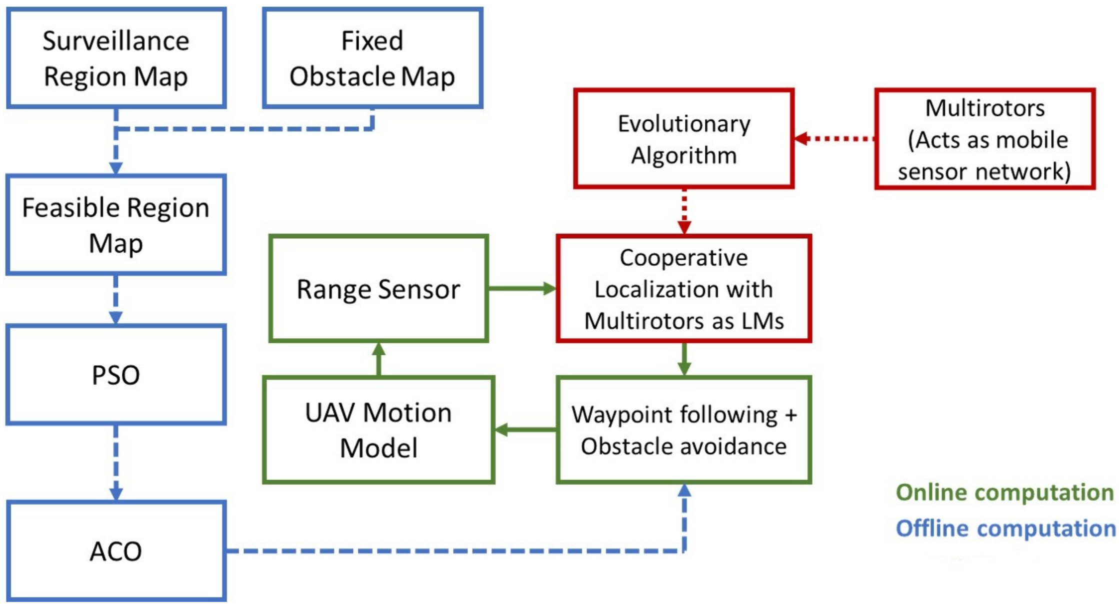

Fig. 2: Detailed framework showing data flow of UAV surveillance. Blue section shows the computation which is done offline and green shows which is done in realtime.

where $n_{wp_{if}}$ are the number of infeasible waypoints and n_{oc} is the number of optimization criteria. n_{oc} adds weight to this term when waypoints are infeasible, increasing the penalty. n_{oc} for this project is 2 when number of $n_{wp_{if}}$ is more than 0. Otherwise $n_{oc} = 0$.

Therefore, the overall cost function becomes

$$CF = \lambda_1 CF_f + \lambda_2 CF_c + \lambda_3 CF_o \tag{12}$$

where λ_i are the weights for each individual cost of the multi-objective cost function.

The PSO algorithm is outlined in 1, where ω is the inertial coefficent, c_1 and c_2 are cognitive and social acceleration coefficients, r_1 and r_2 are random numbers between 0 and 1. Inertial coefficient creates a balance between global and local exploration and ideally should decrease as time increases. Through extensive simulations, we notice that using constriction coefficients defined by Clark and Kennedy [16] gives stable and quicker convergence.

B. Waypoint Allocation and Routing

After determining the optimal locations for the waypoints, we allocate the waypoints to each of the vehicle randomly such that all the waypoints are covered. We then solve for the optimal sequence of the waypoints to be visited by each vehicle so that the tour length is reduced, saving time for another loop. This problem is a Travelling Salesman Problem and its complexity increases with increasing number of waypoints, increasing complexity for traditional optimization techniques. In such cases, Genetic Algorithms (GA) and Ant Colony Optimization (ACO) techniques have proven to be effective in finding a near optimal solution in a given time. It has been shown in literature that ACo is proved to be ore eficient than the basic GA. Hence, we use ACO to solve this problem.

Algorithm 1 Particle Swarm Optimization

Initialize N number of n_{WP} dimensional particles with position in the feasible region and velocity.
set high global cost
for each particle j **do**
    Set high local best cost
    Calculate the cost for each particle from eq. (12).
    Update local best cost for each particle.
end for
Update Global Cost for the whole system.
for i in 1 - max iterations **do**
    for m in 1 - N **do**
        Compute velocity for each particle

$$v_k^m = \omega v_k^m(i-1) + c_1 * r_1(p_k^m - x_k^m) \\ + c_2 * r_2(p_k - x_k^m)$$

        Update Particle's position

$$x_k^m(i+1) = x_k^m(i) + v_k^m$$

        Evaluate Cost from eq. (12)
        Update local best cost if a best one is found
    end for
    if global best cost is found **then**
        Update Global Best Cost
    end if
end for

ACO exploits the foraging behavior of ants in finding shortest path to food resources by communicating by depositing pheromones through the environment as they move. Although all the ants start randomly, they tend to converge onto a path with most pheromone concentration, which is usually the shortest path between food resource and their inhabitant.

We use the strategy in finding the optimal route, for the TSP problem. The Ant Colony Algorithm is described below

Algorithm 2 Ant Colony Optimization

Set of n_{WP} waypoints are given.
Assign each ant randomly to all waypoints
for each particle (i, j) **do**
  $\tau_{i,j}(0) = \tau_0$
end for
for k = 1 to m **do**
  Place ant k on a randomly chosen waypoint
end for
Let T^+ be the shortest tour found from the beginning and L^+ its length.
/* Main loop */
for t = 1 to t_{max} **do**
  for k = 1 to m **do**
    Build tour $T^k(t)$ by applying n - 1 times the following step :
    Choose the next waypoint j with probability
    $p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta]}{\sum_{l \epsilon j_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}(t)]^\beta}$
    where i is the current waypoint
  end for
  for k = 1 to m **do**
    Compute the length $L^k(t)$ of the tour $T^k(t)$ produced by and k
  end for
  if an improved tour is found **then**
    update T^+ and L^+
  end if
end for

C. Controller Design for Sensor Network

In this section, we describe the control approach used for vehicles in the sensor network. Vehicles in the sensor network follow a receding horizon approach or Model Predictive Control approach, where the vehicles plan their path for T time steps ahead as shown in Fig. 3. The approach is outlined in [17].

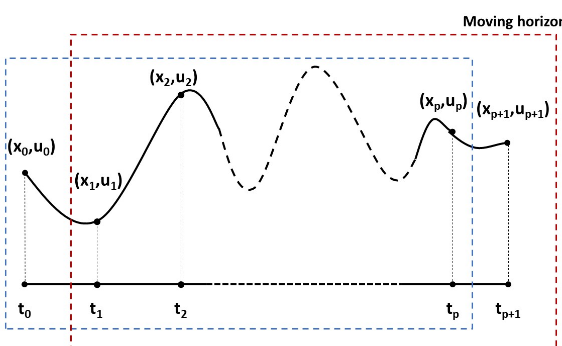

Fig. 3: Trajectory optimization with Moving Horizon approach

Each vehicle predicts the states of other vehicles based on sensor information available at current time-step and plans its path using the Genetic Algorithm (GA) such that graph connectivity is maintained. All the vehicles in the sensor network move at a constant speed. The turn-rates are determined using GA. Each genome of the solution contains turn-rates for all the vehicles for the next T time steps. About 50% solutions undergo single point crossover and the remaining 50% undergo two point crossovers and the resulting child genomes are appended to the original solution set. Randomly selected 50% of the solutions undergo multiple mutations and the resulting solutions are appended as well. About 45% of Best solutions are selected based on elitist selection, other 45% are selected based on tournament selection and remaining are selected randomly. Th best solution is selected after a maximum number of iterations are reached. The algorithm is outlined in 3.

Algorithm 3 Genetic Algorithm

for each control loop **do**
  Predict States of Surveillance Vehicles
  for each vehicle j in sensor network **do**
    initialize m sets of ω_j for T time-steps ahead
  end for
  for i in max iterations **do**
    for each vehicle j in sensor network **do**
      Perform crossover operations on existing solutions.
      Append new solutions
      Perform mutation operations
      Append new solutions
      for each solution **do**
        Propagate states of vehicle j
      end for
    end for
    Calculate the cost for the network from eq. (12).
    for for each vehicle j **do**
      Select 45% of m based on elite selection
      Select 45% of m by tournament selection between two randomly chosen solutions
      Select remaining 10% of m randomly
    end for
  end for
  Select best solution
  for each vehicle j **do**
    update velocity and turn rate as first value of best solution
  end for
end for

We assume that the vehicles and landmarks together for an undirected graph with vehicles and landmarks acting as nodes. An edge from node i and node j is said to be connected if node i and node j are in sensing range of each other. We use eigenvalues of Graph Laplacian matrix as a metric to measure graph connectivity. Graph Laplacian L is calculated from Degree Matrix D and Adjacency Matrix A as follows -

$$L = D - A \tag{13}$$

Since second smallest eigenvalue measures the connectivity of the graph [18]., we use this as cost to the Genetic algorithm.

IV. Results

This section discusses the results of our simulation which was performed in MATLAB and simulink, and its efficiency.

A. Search Area Optimization

Three different surveillance region have been used in this project to identify the efficiency of coverage. It is found that lagrangian multipliers plays a role on the cost of each particle during optimization.Table I shows the properties of each region/maps and table II shows the Lagrangian values which has been used for each cases on the all three maps. PSO simulations were carried out without parallel processing and Max iteration set to 500 because of limitation of computational power. As area of region increases, number of waypoints are also increased thus each simulation requires more number of iteration and run time. It is clear that as $\lambda_{\text{env}}, \lambda_{\text{coverage}}$ and λ_{overlap} plays important role in distribution of WPs to maximize area coverage. λ_{env} should always have higher value than $\lambda_{\text{coverage}}$ and λ_{overlap} combined so as to penalize particles going away from the region of surveillance. Assigning higher λ to overlap cost will penalize higher degree of overlap which will results in more coverage. Whereas assigning higher λ value to coverage cost will penalize lower percentage of coverage which will results in less overlap. From table III, it is evident that it is recommended to give $\lambda_{\text{coverage}}$ and λ_{overlap} , twice the value given to λ_{env} for maximizing coverage.

We focus on the region/map 1 for the remaining results with area of 7315 m^2. 6 UAV has been assigned to surveil the area with each one having 250 m^2 of individual circular ground surface area. Total of 30 WPs are generated allowing overlap between UAVs. Fig 4 shows the disribution of WPs in orange blue circle and orange cirle indicate the area covered when a UAV visit the respective WP.

Map	Scaling Factor	Surveillance Area(m^2)	Total WPs	Run Time
1	1	7315	30	1 Hr 9 Mins
2	2	14200	57	3 Hr 40 Mins
3	3	21987	88	6 Hr 52 Mins

TABLE I: Maps

Cases	λ_{env}	$\lambda_{coverage}$	$\lambda_{overlap}$
1	100	80	0
2	50	100	100
3	20	200	200

TABLE II: Lagrangian multipliers

Cases	Map1		Map2		Map3	
	Cov.%	Ov.%	Cov.%	Ov.%	Cov.%	Ov.%
1	92.54	10.80	88.88	12.91	83.76	19.45
2	92.54	10.80	89.17	12.53	85.00	17.71
3	92.37	10.80	89.04	12.70	83.76	19.45

TABLE III: Coverage and Overlap Percentage

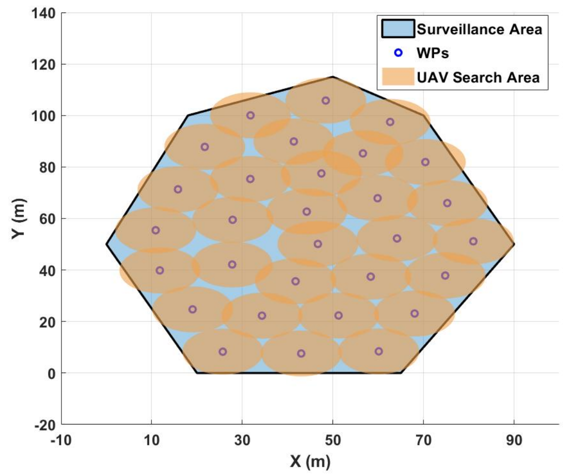

Fig. 4: Waypoint distribution Map 1

B. Waypoint Allocation and Routing

Once the waypoints has been obtained by PSO, it has been randomly assigned to each UAV equally. If total number of waypoints are not divisible by number of vehicles, then the remaining WPs are assigned randomly to any UAVs. Sets of WPS for UAV were provided as input to Ant Colony Optimization to get the sequence of waypoint at which each vehicle has to travel for the effective surveillance. Fig 7 shows the the path in which each UAV is visiting the assigned WPs in an optimal path.

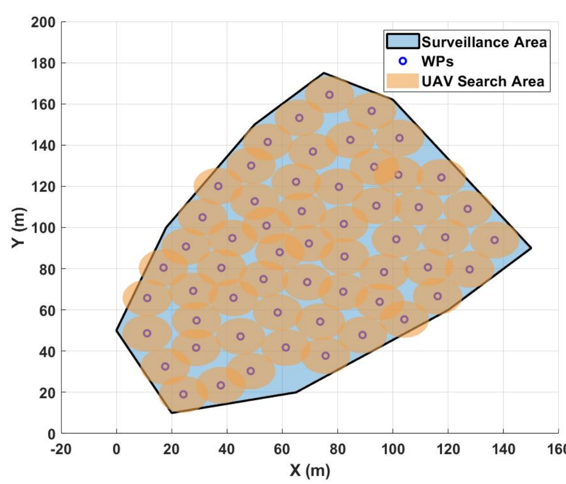

Fig. 5: Waypoint distribution Map 2

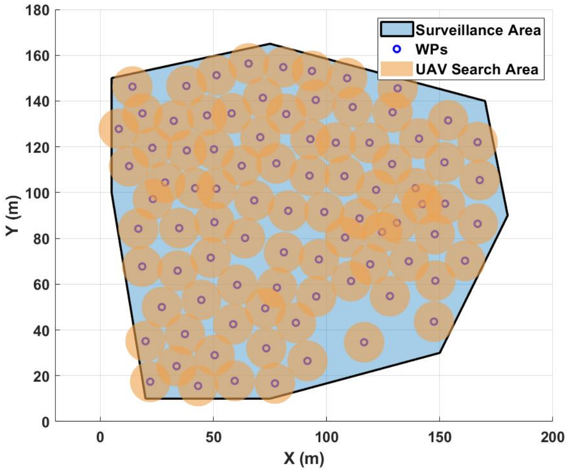

Fig. 6: Waypoint distribution Map 3

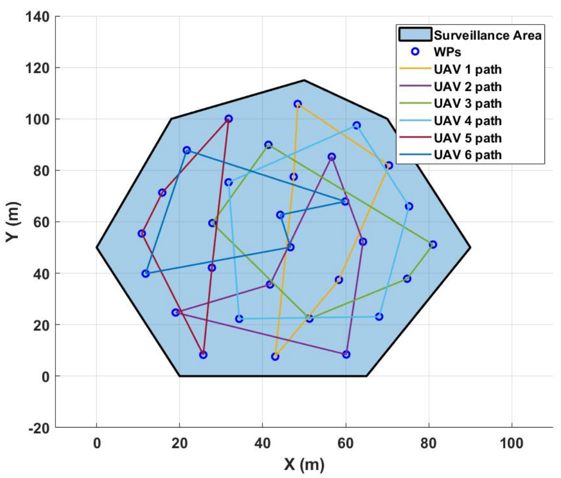

Fig. 7: Sequence of Waypoints for each UAV, Map1

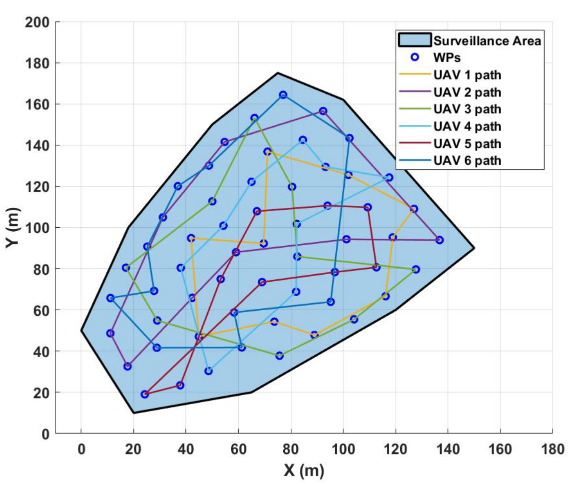

Fig. 8: Sequence of Waypoints for each UAV, Map2

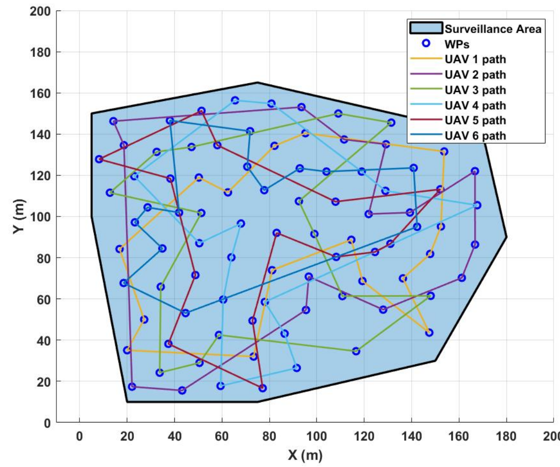

Fig. 9: Sequence of Waypoints for each UAV, Map3

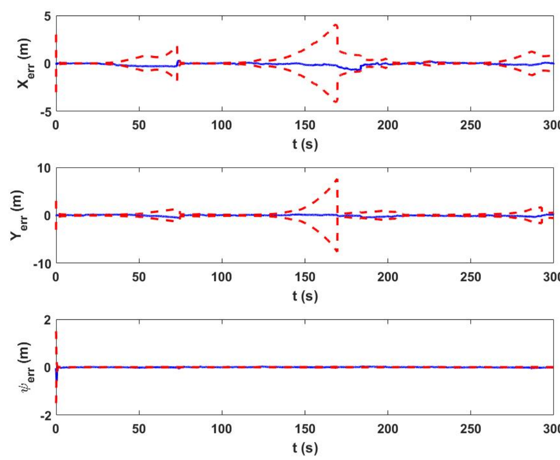

Fig. 10: Localization error plot for Surveillance network

C. *Controller Design for Sensor Network*

After extensive simulations using Genetic Algorithms and traditional optimization techniques using MATLAB's fmincon, we notice that both the solvers fail in converging or finding an optimal solution. Upon investigation, it is observed the second least eigenvalue of graph laplacian doesn't vary in the local neighborhood of the initial solution, rendering the optimization techniques obsolete because there is no way of discriminating in the neighborhood of the initial solution. Hence, eigenvalue of graph laplacian cannot be used as cost function for this problem. Metrics like observability gramian might be required to derive an appropriate cost function for maintaining observability of the system, which is hard to obtain and hence is not derived because of time constraints.

We present some intuitive trajectories for the vehicles in the sensor network and observe the variation of uncertainty with the second lowest eigenvalue of graph laplacian. However, it should be noted that pinpointing the exact connectivity of each node of the network based on one value alone cannot be determined. We just make a case here purely based on known

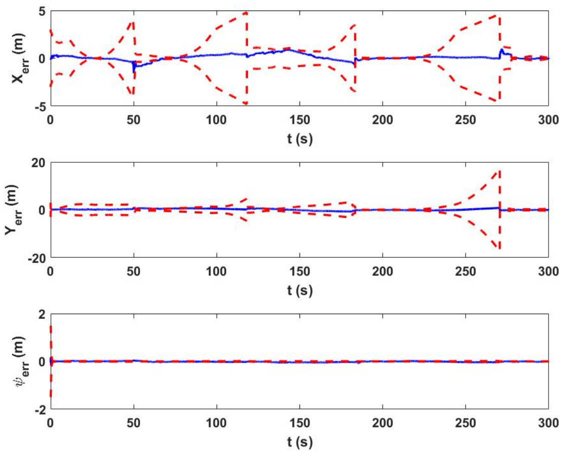

Fig. 11: Localization error plot for Sensor network

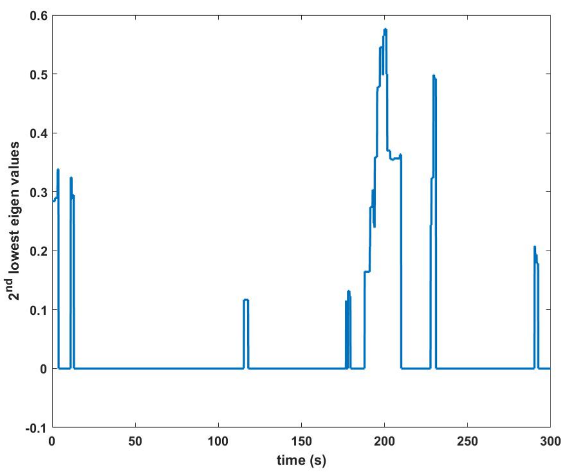

Fig. 12: Second lowest eigen values of graph laplacian

theory and observation alone.

We plot error-covariance plot of one vehicle from surveillance team and one vehicle from sensing team in Fig. 10 and Fig. 11. We also plot the second lowest eigenvalue as the network evolves in Fig. 12. We notice here that covariance increases in both scenarios until 200^{th} second and drops down after 200s. Increased connectivity of the graph after 200^{th} can be attributed as one of the reason for this behavior.

V. Conclusion

In this paper, we solve the problem of maximizing coverage area using Particle Swarm Optimization and show that it handles well in finding an optimal solution. If parallel computing can be exploited as described in [15], the efficiency of the algorithm increases as well. We solved the vehicle routing problem using Ant Colony Optimization. We also describe the importance of having a dedicated mobile sensor network to help in localization. We also observed that eigenvalue properties of graph laplacian matrix cannot be used alone for optimizing graph connectivity. Observability based properties should be used to derive an appropriate cost function for solving this problem. The mobile sensor network itself poses several new problems to solve in terms of number of vehicles required, optimal routing based on area and application, communication challenges and demand based allocation of resources, which can be useful for several applications pertaining to urban air traffic problem.

References

[1] K. M. Brink, "Multi-agent relative pose estimation: approaches and applications," in *Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation 2018*, vol. 10651. International Society for Optics and Photonics, 2018, p. 106510D.

[2] A. R. Girard, A. S. Howell, and J. K. Hedrick, "Border patrol and surveillance missions using multiple unmanned air vehicles," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 1. IEEE, 2004, pp. 620–625.

[3] P. D. Groves, "Shadow matching: A new gnss positioning technique for urban canyons," *Journal of Navigation*, vol. 64, no. 3, pp. 417–430, 2011.

[4] J. S. Warner and R. G. Johnston, "Gps spoofing countermeasures," *Homeland Security Journal*, vol. 25, no. 2, pp. 19–27, 2003.

[5] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *The international Journal of robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[6] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 237–260, 2007.

[7] D. Fontanelli, L. Ricciato, and S. Soatto, "A fast ransac-based registration algorithm for accurate localization in unknown environments using lidar measurements," in *2007 IEEE International Conference on Automation Science and Engineering*. IEEE, 2007, pp. 597–602.

[8] S. Lee and J.-B. Song, "Mobile robot localization using infrared light reflecting landmarks," in *2007 International Conference on Control, Automation and Systems*. IEEE, 2007, pp. 674–677.

[9] M. Drumheller, "Mobile robot localization using sonar," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 325–332, 1987.

[10] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE transactions on robotics and automation*, vol. 13, no. 2, pp. 251–263, 1997.

[11] R. Sharma, R. W. Beard, C. N. Taylor, and S. Quebe, "Graph-based observability analysis of bearing-only cooperative localization," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 522–529, 2011.

[12] R. Dutta, L. Sun, M. Kothari, R. Sharma, and D. Pack, "A cooperative formation control strategy maintaining connectivity of a multi-agent system," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1189–1194.

[13] R. Sharma, "Observability based control for cooperative localization," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014, pp. 134–139.

[14] A. Chakraborty, S. Misra, R. Sharma, and C. N. Taylor, "Observability conditions for switching sensing topology for cooperative localization," *Unmanned Systems*, vol. 5, no. 03, pp. 141–157, 2017.

[15] N. Mishra, A. Chakraborty, R. Sharma, and K. Brink, "Cooperative relative pose estimation to aid landing of an unmanned aerial vehicle on a moving platform," in *2019 Fifth Indian Control Conference (ICC)*. IEEE, 2019, pp. 277–282.

[16] S. Y. Lim, M. Montakhab, and H. Nouri, "A constriction factor based particle swarm optimization for economic dispatch," 2009.

[17] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*. Birkhäuser, 2012, vol. 26.

[18] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," in *Proceedings of the 2005, American Control Conference, 2005*. IEEE, 2005, pp. 99–103.