Persistent Surveillance Using Multiple Unmanned Air Vehicles

Nikhil Nigam and Ilan Kroo
Department of Aeronautics and Astronautics
Stanford University, CA 94305
650-723-3317
nnigam@stanford.edu
kroo@stanford.edu

Abstract—Search and exploration using multiple autonomous sensing platforms has been extensively studied in the fields of controls and artificial intelligence. The task of persistent surveillance is different from a coverage or exploration problem, in that the target area needs to be continuously searched, minimizing the time between visitations to the same region. This difference does not allow a straightforward application of most exploration techniques to the problem, although ideas from these methods can still be used. In this research we investigate techniques that are scalable, reliable, efficient, and robust to problem dynamics. These are tested in a multiple unmanned air vehicle (UAV) simulation environment, developed for this program.

A semi-heuristic control policy for a single UAV is extended to the case of multiple UAVs using two methods. One is an extension of a reactive policy for a single UAV and the other involves allocation of sub-regions to individual UAVs for parallel exploration. An optimal assignment procedure (based on auction algorithms) has also been developed for this purpose. A comparison is made between the two approaches and a simplified optimal result. The reactive policy is found to exhibit an interesting emergent behavior as the number of UAVs becomes large. The control policy derived for a single UAV is modified to account for actual aircraft dynamics (a 3 degree-of-freedom nonlinear dynamics simulation is used for this purpose) and improvements in performance are observed. Finally, we draw conclusions about the utility and efficiency of these techniques.[1 2]

TABLE OF CONTENTS

1———————————————
[1] 1-4244-1488-1/08/$25.00 ©2008 IEEE
[2] IEEEAC paper #1318, Version 7, Updated December 15, 2007

1. INTRODUCTION

Surveillance of a target space using aerial vehicles is a topic of current research interest for applications such as weather monitoring, geographical surveys, and perhaps extra-terrestrial exploration. These missions often require long endurance and unmanned aerial vehicles are a natural choice for the sensing platforms. The use of multiple UAVs in this context is motivated by concerns for reliability and efficiency of exploration. Extensive research in the controls and AI communities has been geared toward cooperative multiple robot problems [1] [2], and in particular, autonomous search and exploration tasks [3]. The use of aerial vehicles brings additional concerns pertaining to dynamics of the vehicles [4], but many of the same ideas apply. Flint et al. [5] [6] have applied deterministic Dynamic Programming (DP) for path planning for each vehicle, with improvements suggested by Yang [7]. A K-shortest path algorithm has been used by Sujit et al. [8] for maximizing explored area under endurance constraints, while [9] uses mixed integer linear programming for task assignment and trajectory planning. Work in [10] focuses on a combination of centralized and decentralized approaches to address communication and computational constraints. Much effort has been associated with uncertainties in sensing and motion in presence of obstacles for robots exploring an office environment [11]. Traditional search strategies such as A* have also been applied for search tasks [12], but do not address the problem of cooperation between vehicles. Latimer et al. [13] have used boustrophedon decomposition to divide target space into cells, which are then searched using either a single UAV or multiple UAVs in formation. Other work using decomposition based methods has addressed the problems of static [14] and dynamic coverage [15]. Coordination field methods [16], [17] that incorporate particle swarm optimization, digital pheromone mechanisms, and potential function based approaches, are highly scalable, using reactive control policies. Caselli et al. [18] use a probabilistic path planning approach and suggest ways to escape from local minima associated with potential fields. Among other scalable approaches, work by Tumer et al. [19] [20] is notable. In that work neural networks are used to represent control policies that are learnt online using an evolutionary algorithm. Recent work at Boeing [21] uses a combination of four basic behaviors to define the overall reactive policy of each UAV to solve an exploration problem.

Most of the methods described above can be classified into two categories: One class includes approaches with a formal derivation or proof of optimality but not scalable to a very large number of vehicles (tens or hundreds) because of embedded planners. The other class involves approaches that are decentralized and scalable but heuristic. Some of the techniques cannot be applied in an online setting and may not be useful for sensor-based coverage. Many strategies either ignore vehicle dynamics or treat it independently of the control scheme. Finally, application of many existing methods to a persistent surveillance problem is not straightforward.

In the present problem, the target space (physical area to be searched) uses an approximate cellular decomposition[3] and each cell has an associated age, determined from the time elapsed since it was last observed. The system level goal is to minimize the maximum age over all cells that are observed over a long period of time. That is, no area of the target space should be left unexplored for a long period of time. This problem differs from an exploration task in that the space has to be continually explored, and the age of the cell becomes important[4]. It also differs from problems of minimizing map uncertainty[5] generally encountered in literature [22], wherein the cumulative uncertainty is what matters and not the maximum uncertainty of a cell. We also require that the control policies followed by UAVs be responsive to changes in the environment and able to deal with cases in which all areas are not of equal importance (though we do not deal specifically with these aspects in this paper).

In this study, we start by deriving an optimum policy for a simple case with a single UAV in 1-D. We then extend this policy to a more realistic scenario for a single UAV and analyze its performance. The method is compared to a heuristic approach, similar to potential function methods, and to a DP based planning approach. The single UAV policy is then extended to a multiple-UAV case resulting in a multi-agent reactive policy (MRP).

Another second approach to multiple UAV coordination is also studied. The target space is optimally divided into subspaces that are allocated to individual UAVs and are then searched independently by each UAV. This space decomposition (SD) method involves finding the optimum partitioning of the space. A real-encoded genetic algorithm is used for this purpose. After the decomposition, UAVs are allocated to partitions using an optimal assignment procedure based on auction algorithms [23] [24]. A multi-UAV simulation environment has been developed to compare the MRP and SD methods to a bound on the optimal performance. The MRP is found to exhibit an emergent behavior as the number of UAVs becomes large. The effect of aircraft dynamics (using a 3-DOF model) on system performance is also studied. The altitude of the aircraft is assumed to have no effect on sensing, so only the turn radius and velocity of aircraft affect the performance. We further introduce a non-holonomic constraint by assuming that the aircraft travels (and turns) at constant velocity. This allows us to simplify the dynamics model, which is then used for a single UAV to analyze the coupling between the control policy and dynamics. A minimum length trajectory-tracking controller is also described and implemented. We then look at a way to improve UAV performance with dynamic constraints.

[3] Approximate cellular decomposition refers to the fact that the sensor footprint equals the cell dimensions and in our case, the cells exhaustively cover the target space too.

[4] As opposed to a binary variable indicating whether a cell has been explored or not.

[5] The uncertainty reduces when a cell is explored and increases over time, analogous to the age parameter in our implementation.

2. POLICY FOR A SINGLE UAV

An optimum policy for a single UAV in 1-D with only two cells is analyzed as a starting point. Extension to a 2-D, multiple cell case leads to a search pattern that emerges from the simple reactive control policy. This policy is compared with some benchmark techniques.

Optimum Policy Structure

This exercise attempts to find a structure for a reactive control policy that would work well for this problem. We consider a case of two cells that need to be visited so as to minimize the maximum of the ages of the cells observed. The problem is 1-D, and the UAV, stationed at distance x from the left cell (see Fig. 1), is assumed to travel at constant velocity, V_{survey}. Without loss of generality, we can assume that the distance between the cells is 1 unit and x and V are scaled accordingly. Let T_j denote the age of the j[th] cell.

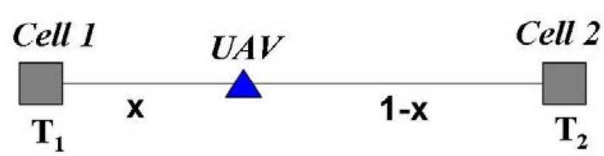

Figure 1 – Simplified two-cell problem

There are two possible actions for the UAV - go left or go right – which is the same as choosing to go to cell 1 first or cell 2 first[6]. After the UAV has chosen which cell to observe first, the optimum policy is to keep moving back and forth between the cells, so that a single action defines the optimum policy in this very simple case.

[6] It is easy to see that any other strategy where the UAV initially starts going to a cell and turns back mid-way will necessarily be sub-optimal.

Assuming $T_1 \leq T_2$, we can construct a plot of the ages of cells for the case where the UAV chooses to go left (Fig. 2) or right (Fig. 3) first. Such a plot can be used to identify the maximum age (over the two cells) as a function of time. Our optimum policy tries to minimize the peak of this maximum age curve.

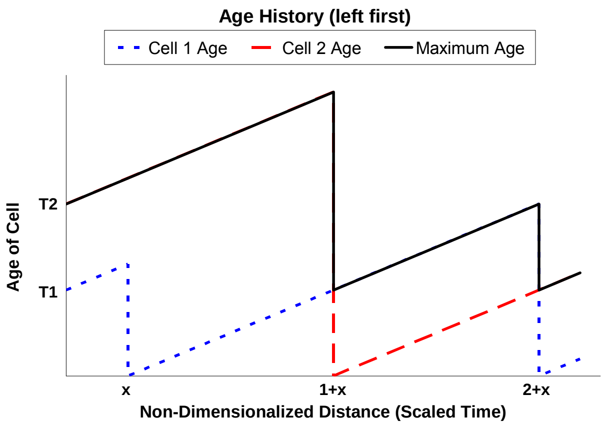

Figure 2 – Ages of cells as a function of time when UAV goes to cell 1 first

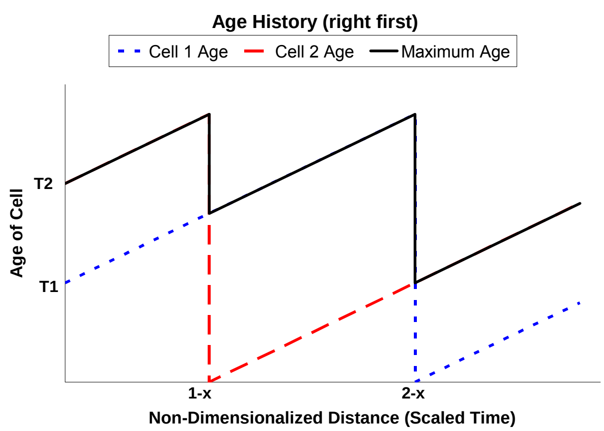

Figure 3 – Ages of cells as a function of time when UAV goes to cell 2 first

As pointed out earlier, we already know the optimum policy after the first cell has been visited, so we need to consider the maximum age plot only until a finite time ($2/V_{survey}$ in our case)[7]. Let M_{left} and M_{right} denote the peaks of the maximum age curves when the UAV chooses left and right respectively. Hence we have:

$$\text{Choose left} \Leftrightarrow M_{left} \leq M_{right}$$
$$\Leftrightarrow \max\left\{\left(T_2 + \frac{1+x}{V_{survey}}\right), \frac{2}{V_{survey}}\right\}$$
$$\leq \max\left\{\left(T_2 + \frac{1-x}{V_{survey}}\right), \left(T_1 + \frac{2-x}{V_{survey}}\right), \frac{2}{V_{survey}}\right\}$$

These equations are dealt with on a case by case basis and the same analysis is done for $T_1 > T_2$. The results can be summarized as:

$$\text{If } T_1 \leq \frac{x}{V_{survey}}, \text{ignore cell 1}$$
$$\text{If } T_2 \leq \frac{1-x}{V_{survey}}, \text{ignore cell 2}$$
$$\text{If none of the above is true, then,}$$
$$\text{choose left} \Leftrightarrow T_1 - \frac{x}{V_{survey}} \geq T_2 - \frac{1-x}{V_{survey}}$$

More succinctly, we can define a value for each cell to be a linear combination of the age of the cell and the distance of the UAV from the cell (threshold at zero):

$$V_j = \max\{(T_j + w_0\delta_{ij}), 0\}$$

where, V_j is value of j^{th} cell, T_j is the age of j^{th} cell, w_0 is a weight ($= -1/V_{survey}$ for the two-cell case), and δ_{ij} is the distance between i^{th} UAV and j^{th} cell. The optimum policy is then to go to the cell with maximum value.

It is easy to see that this is optimal for a 2-D, two cell case also, where instead of choosing left and right, the UAV chooses to go towards one cell or the other. The analysis for that case is completely analogous to the above analysis.

Extension to 2-D Multiple Cells Case

In this section we use the policy structure obtained from the above analysis and extend it to a 2-D case with multiple cells. We have two options – either combine values of multiple cells to find the direction to go in[8] (*Sum of Value* approach), or go towards the cell with the maximum value (*Target based* approach). To resolve this issue, we consider a case with 5 cells in 1-D, as shown in Fig. 4.

3

[7] In case the initial ages of cells are such that the peak of the curve lies outside this time range, it does not matter what action we choose.

3

[8] In the 2-D case, we can use vector addition (with value of cell divided by distance as magnitude of vector and direction along vector from UAV position to cell).

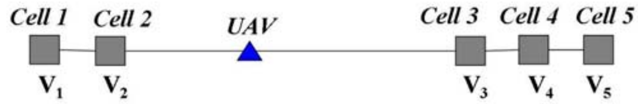

Figure 4 – Example 1-D scenario with 5 cells

Suppose that $V_1 > V_2$ and $V_4 > V_3$, $V_3 > V_5$, where V_j is the value of the j[th] cell. Now, if the cells on either side are approximately[9] at the same locations, then we need to consider only the values V_1 and V_4 to decide on optimum policy, (ie. if $V_1 > V_4$, we need to go left irrespective of other cell values and vice versa). Even for the case when the cells are not at the same location, it is intuitive to see that we should not combine the values of cells, since given our objective where only the maximum age recorded matters, the cell with the maximum value is the critical cell[10]. Hence, for the 2-D multiple cell case also, we choose the cell with the maximum value as our *target* cell and take maximum step towards that cell. It should be noted that due to the nature of the exploration task, it does not help to move slow, unless there are sensor issues, which we will ignore for now.

To validate our reasoning, both approaches are compared. The target space is assumed to be square-shaped and normalized to have a unit length in each dimension. The sensor footprint of each UAV is circular with radius, r_{sensor} = 0.025, and the velocity of the UAV is V_{survey} = 0.03[11]. Fig. 5 shows the maximum ages observed over a long time for 50 different trials (UAV starts from a random location in the target space in each trial), using the two policies. As we can see from the results, the target-based approach works much better.

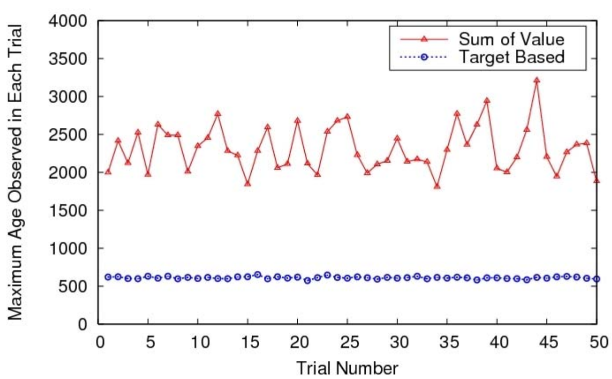

Figure 5 – Comparison of Target based and Sum of Value approach

If more than one cell has the same value, our policy treats the cells equally and there is no reason to go to one cell as opposed to the other. However, instead of choosing the cell randomly, we choose the cell which results in least heading change. Though this does not affect the performance of the system much, it will be helpful in practical implementation, since an actual aircraft wants to minimize the number of turns it has to make[12].

The weight parameter is $w_0 = -1/V_{survey}$, as for the two cell case. However, this value might not be optimal for the multiple cell scenario. So we first find the optimum value of w_0 using an Iterative Sampling (ISIS) based optimizer (developed by one of the authors, see for example [25]). Put simply, ISIS maintains a population of sample points, and the population is re-sampled randomly from a region around the best point, while expanding or reducing the size of sample space based on performance. The optimum weight thus found is approximately equal to the analytical optimum. This also indicates that extending the policy as above is reasonable.

Testing the Policy

In this section we try to gauge how good our target based policy is compared to some benchmark methods. We do not show comparison to a random action policy (UAV moves in a random direction at each time step), since the performance with random actions is much worse (as typically some area in the target space is left unexplored for a very long time).

Comparison with Potential Field Like Approach—We first compare our policy to a heuristic policy similar to the approach used by Tumer et al. for a rover exploration problem [19]. In their work, a rover observes the environment through sensors in four quadrants. The sensors return the sum of values of *points of interest* weighted by inverse squared distances of the cells in each quadrant. These become the inputs for the policy and the output is the direction the UAV wants to go (obtained through a neural network learned using an evolutionary algorithm).

In our implementation, the centers of cells become the points of interest, and their ages are the values. The form of the policy is linear (which is found to work better than training the neural network) – ie., the outputs are a linear combination of the inputs and no online learning is used. This policy (which is quite similar to a potential field approach) is then compared with our target based approach. The target space is of unit length in each dimension, r_{sensor} = 0.025 and V_{survey} = 0.03. The results are shown in Fig. 6, which plots the maximum age over all cells observed in each trial, for a total of 50 trials. We can see that our target based approach performs much better.

4

9 This assumption is made to ensure that the optimum policy can still be defined completely by choosing either left or right at the first time step.

10 This is not a formal proof, but an intuitive observation.

11 These values are non-dimensionalized with respect to target space dimension.

4

12 A more elaborate discussion on how to account for vehicle dynamics will follow in a later section.

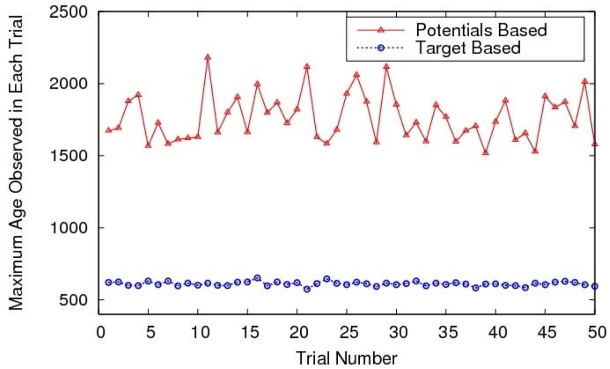

Figure 6 – Comparison of Target based and Potential function based approach

Comparison with Planning Based Approach—The target based approach is reactive in nature, so it is useful to think whether we can do any better using planning. A straightforward implementation of planning would involve the UAV choosing a sequence of steps as opposed to a single step. However, by virtue of the structure of our policy, if a UAV takes a step towards a cell with maximum value, then the value of the cell only increases relative to other cells as long as the cell is not reached. Hence, it does not help to plan for future steps using the policy as is. An alternative explanation is that the target based approach tries to convert the time-extended problem to a single step problem by incorporating the time element through the weighted distance. Hence ideally, it should not require any sort of planning.

However, there exist different implementations of planning techniques in literature that can be tried. One popular method is the *Dijkstra's Shortest Path Algorithm (SPA)* [26]. Though the algorithm is a greedy algorithm, it still finds the optimum path for the given problem[13]. The Dijkstra's algorithm described in [27] is modified to get a *longest* path algorithm. In the graph, the nodes correspond to cells in the grid, so all the cells that the UAV can go to from a given cell/node, are the possible next states of that cell (~20 for our problem). The age of the cell becomes the weight of the edge connecting the nodes. Some might argue that we should use the maximum age over all cells observed after taking a path, as the length of the path and try to find the shortest path. This is true if we have an infinite planning horizon, but for finite time horizon with short look ahead, this does not work. The number of nodes in the graph is of the order ~$O(20^{\boldsymbol{h}})$, where $\boldsymbol{h}$ is the time horizon.

The planning algorithm is implemented for time horizons of 1, 2 and 3 steps due to computational limitations. The results are however much worse than the reactive policy. This is understandable, since as pointed out above, the reactive policy tries to convert the time-extended problem to a single step problem and potentially looks at an infinite time horizon. So the finite time horizon based planning does not work as well. Hence, at least for the single UAV case, the reactive policy seems to be better than a planning based approach.

Emergence of a Search Pattern—In this section we briefly look at the search pattern that emerges from following the reactive policy in an ideal scenario.

Suppose that the ages of the cells are all zero to start with, and the UAV starts from one corner of the target space. The target based approach then results in a spiral search pattern as shown in Fig. 7. The UAV starts from a corner, spirals in to the center and then returns to the starting position, repeating the pattern. Note that it is possible to find an optimal pattern for the UAV to travel in, such that it does not visit any cell twice in each run through the space. Though the optimal policy is certainly a desirable trait, we can tolerate slight deviations from it. Especially in case of problem dynamics and UAV failures, this distinction would not be relevant. Moreover, the target based approach can react to changes in the environment and would in general perform better than a pre-fixed policy.

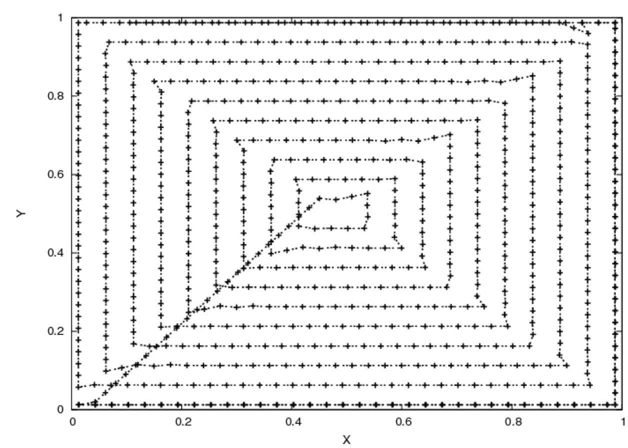

Figure 7 – Path followed by a single UAV under ideal initial conditions

Even when the UAV starts from random locations in the target space, it does not perform much worse. We consider a same problem with $\boldsymbol{r}_{sensor} = 0.02$, $\boldsymbol{V}_{survey} = 0.04$[14]. An optimal policy (if it exists) can search the space in 625 time steps. Fig. 8 compares the target based approach with this lower bound on the optimum. The plot shows the maximum age over all cells as a function of number of time steps. The results are averaged over 50 trials (different starting position for UAV in each case). As we can observe, the performance of our approach is close to the lower bound on optimal performance for a single UAV.

5

[13] The graph is a directed graph with non-negative weights.

5

[14] The UAV travels from one cell center to the next one.

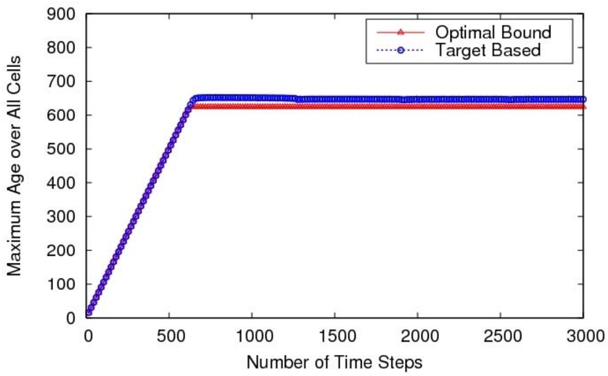

Figure 8 – Comparison of Target based policy performance to a lower optimal bound

3. Policy for Multiple UAVs

A lot of schemes have been proposed for coordination among multiple agents. However, we have investigated techniques which are robust, scalable, and simple in concept. Two techniques have been developed for this purpose. The first one, the MRP, is a simple extension of the policy used for a single UAV. The second approach, SD, involves optimal partitioning of the target space and subsequent allocation of sub-spaces to UAVs for independent surveillance.

Multi-agent Reactive Policy

To understand how the existing policy might be extended to a multiple UAV case, we start with a simple case of 2 UAVs and 2 cells in a 1-D space, as shown in Fig. 9. Without loss of generality, the distance between the cells can be assumed to be 1 unit.

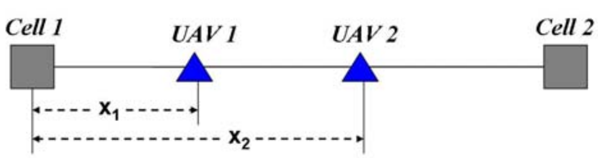

Figure 9 – Example 1-D scenario with 2 cells and 2 UAV's

It is clear that the best policy in this case is for the UAV nearest to a cell to move to that cell and stay there.

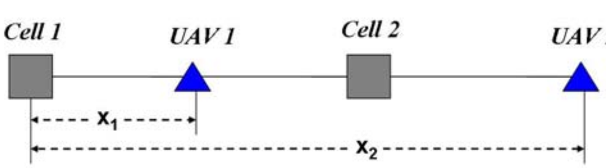

Figure 10 – Variant of 1-D scenario with 2 cells and 2 UAV's

Now, consider another case as shown in Fig. 10. In this case, assuming that UAV-2 moves to the 2[nd] cell, we need to find decision policy for UAV-1. Using an analysis analogous to the single UAV case, we get:

$$\text{Choose left} \Leftrightarrow \left(T_2 + \frac{x_2 - 1}{V_{survey}} \right) \leq \left(T_1 + \frac{2 - x_1}{V_{survey}} \right)$$

$$\Leftrightarrow \left(T_2 - \frac{1 - x_1}{V_{survey}} + \frac{x_2 - 1}{V_{survey}} \right)$$

$$\leq \left(T_1 - \frac{x_1}{V_{survey}} + \frac{x_2}{V_{survey}} \right) + \frac{x_1}{V_{survey}} - \frac{x_2 - 1}{V_{survey}}$$

This is optimum for the case when $(x_2 - 1) \geq x_1$. However, we draw motivation from this result and choose a policy where values of cells are defined by:

$$V_j = \max \{(T_j + w_0 \delta_{ij} + w_1 \min_{k \neq i}(\delta_{kj})), 0\}$$

where, V_j is the value of j[th] cell, T_j is the age of j[th] cell, δ_{ij} is the distance between the i[th] UAV and j[th] cell, w_0 and w_1 are weight parameters (= $-1/V_{survey}$ for the simple case described above). This becomes the policy structure for the MRP. To use this policy, the UAVs need to know positions of all UAVs at all time steps. The weight parameters however need to be optimized.

Offline Optimization for Weights in MRP—The weights of the policy can be found using offline optimization. The objective function for the optimizer is the actual system objective. The value of the objective for the same parameters can vary depending on the starting positions of the UAVs, so the objective function is noisy. Therefore, we use a population based method, ISIS for tuning the weights, which does not rely on function gradients. The weights for different policies are allowed to be different, resulting in different policies for each UAV. The optimization thus needs to be conducted for different number of UAVs too.

This approach performs much better compared to the case of no coordination (except for the implicit coordination due to sharing of a common map between UAVs). However, to make any claims about the approach, it has to be compared with other methods. This method is purely reactive in nature, so it is useful to look at an approach at the opposite end of the spectrum – we describe such an approach next.

Space Decomposition Approach

This approach involves decomposing the target space into ***m*** partitions (***m*** is the number of UAVs) and assigning one partition to each UAV. If all the UAVs are homogeneous, the environment is static, and the UAVs can start from any place, then we just need to equi-partition the space. However, we have not placed such constraints on our problem, so there is a need to find an *optimum* partitioning of target space. The optimization would also need to be carried on online if re-partitioning is required.

The problem of optimal partitioning of the space is, in general, a hard optimization problem. Kurabayashi et al [28] use a decomposition technique to divide work among multiple robots, but it suffers from scalability issues. Polygon area decomposition techniques used in [29] require a priori knowledge of the domain. Voronoi partitions have been studied for a special class of coverage problems [14], but it is not clear how to extend their work to the present problem. We are continuing work on this problem of optimal partitioning.

In this paper we present a conceptually simple way to deal with this problem. We decided to restrict the paper to rectangular partitions only. It is believed that a completely general partitioning of the space would not offer as much of an improvement in performance, as would be the penalty for increased computational cost. The optimization is done using a real-encoded GA, PCGA (developed by one of the authors, see for example [25]). The objective function for the optimization is the system level mission objective (maximum age recorded over all cells over a long period of time). It might seem that the use of this approach entails a lot of communication between UAVs and a centralized architecture. However, given enough computational resources on each platform, this procedure can be implemented independently by each UAV, and with proper synchronization, all UAVs will arrive at the same result.

Re-parameterization of Space for SD—The optimization problem involving only rectangular partitions is still computationally expensive – if we choose the coordinates of corners of rectangles as design variables. It seems that choosing a different sort of parameterization which results only in feasible (covering the complete target space) partitions might help. There are various ways to do this, but we use a parameterization that makes most sense in terms of making the optimization more efficient.

We divide the space using horizontal or vertical lines. In general, the number of partitions is not a fixed function of the number of lines. So we consider a *recursive partitioning* approach where we recursively divide the space till we get the required number of partitions. The complete domain is first divided into two parts using a line and half the number of UAVs are assigned to each domain[15]. Next, in each respective sub-domain, the same procedure can be repeated. Fig. 11 shows a case where three lines are used to obtain four partitions.

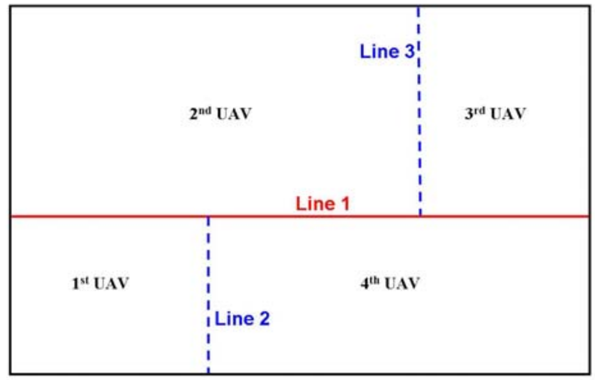

Figure 11 – Illustration of recursive partitioning

We claim that this can cover the space of all feasible rectangular partitions of the space[16]. The task of the optimizer is to find out the optimum positions of the lines (ordered sequentially) and also decide whether they are horizontal or vertical. The fact that the lines can be either vertical or horizontal creates a discontinuity in the objective function. However, for our problem in particular, it can be argued that this is not a huge issue, and in fact the use of a real-encoded GA (as opposed to a binary GA) is also fine. The number of design variables is now $\boldsymbol{m}$–1, as opposed to 4$\boldsymbol{m}$ earlier. When compared with the latter, we observe that the re-parameterization indeed saves a lot of computational effort in terms of the optimization cost.

Optimal UAV Assignment—After finding the optimum partitioning, the UAVs need to be assigned optimally to the partitions. Ideally, this process should be coupled with the optimization, but that would make the optimization intractable even for a modest number of UAVs. So within the optimization, we use a certain heuristic for assignment, and then do an optimal allocation after the optimization has been completed.

We use *auction algorithms* for this purpose, which are optimal for assignment and related problems [23] [24]. However, they address the problem of maximizing (minimizing) the sum of values (costs) associated with the assignments. In our problem, we want to find an assignment that minimizes the maximum age recorded in any sub-space. So this is a *minimax* (sometimes referred to as *Bottleneck*) assignment problem. Garfinkel suggests a *Threshold algorithm* [30] for this problem, which repeatedly solves an assignment problem using the *Hungarian method*. In his implementation, the assignment problem in each iteration, is solved using a *Ford Fulkerson algorithm* [27]. We, however, replace the Ford Fulkerson method by an auction algorithm. The reader is encouraged to look at the references for details regarding these algorithms.

7

[15] In case of odd number of UAVs, the left (lower) domain gets the extra UAV, without loss of generality.

7

[16] There are a few pathological partitions which might not be achieved by virtue of the fact that we assign larger number of UAVs to the left/lower partition, but we choose to ignore those here.

When this procedure is implemented, we do not always end up with the optimal allocation. The problem is identified to be in the way the threshold value, V_{thresh}, is initialized and updated. In our implementation, the initial value of V_{thresh} is chosen to be the minimum value in the entire cost matrix. And its value is updated in each iteration by choosing the smallest value greater than current value in the cost matrix. While this makes the procedure slower, it does result in optimal allocation and that is our major concern since the cost of running this algorithm is insignificant compared to the overall problem.

Comparison of MRP and SD Approach

We now compare the MRP (weights optimized using ISIS) and SD (optimization done using PCGA) approaches. The problem studied for this purpose uses a target space of unit length in each dimension with r_{sensor} = 0.02 and V_{survey} = 0.04. The comparisons are made for cases of 3, 5 and 10 UAVs. Figs. 12-14 show the comparisons by plotting the maximum age observed over all cells as a function of number of time steps (averaged over 50 trials). The variations in the results for different trials are not significant. Also plotted is a lower bound on the maximum ages observed by following an optimum policy. The exact optimum policy for each case is not known, and is difficult to compute. But at best a UAV can sense one cell at each time step without going over cells twice[17].

Table 1 summarizes the results. It shows the average maximum ages observed using the two policies and compares it to the lower bound on the optimum. The MRP* and SD* values are normalized with respect to the lower bound.

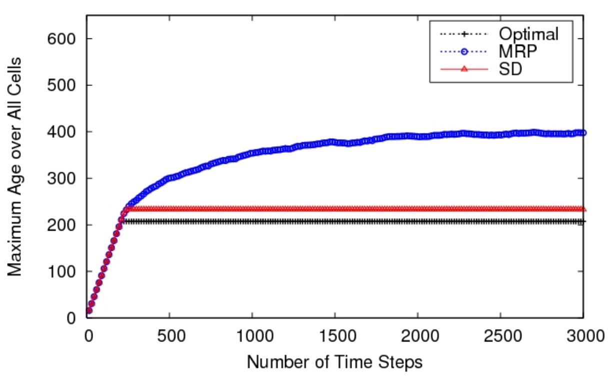

Figure 12 – Maximum age vs. time for 3 UAV cases for MRP, SD and optimal policies averaged over 50 trials

[17] This might not be achievable in all scenarios, but a lower bound on the optimum can be defined to be the total number of cells in the domain divided by the number of UAVs

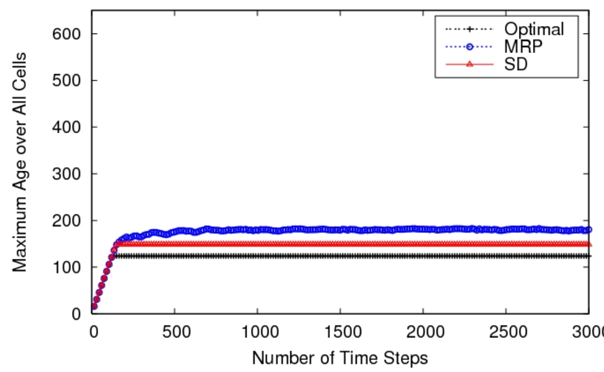

Figure 13 – Maximum age vs. time for 5 UAV cases for MRP, SD and optimal policies averaged over 50 trials

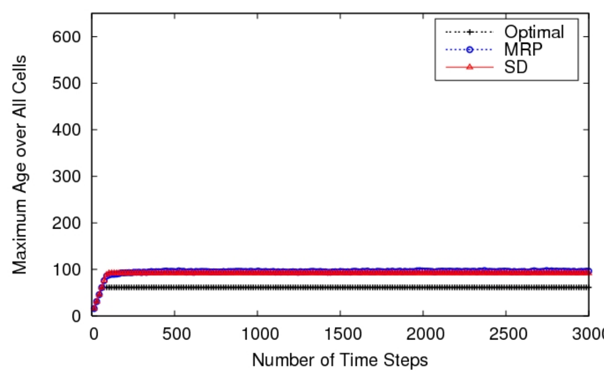

Figure 14 – Maximum age vs. time for 10 UAV cases for MRP, SD and optimal policies averaged over 50 trials

Table 1. Summary of results comparing MRP, SD and lower bound on optimum

No. of UAVs	MRP	SD	Lower Bound	MRP*	SD*
3	423.8	233.3	208.3	2.03	1.12
5	208.0	148.5	125.0	1.66	1.12
10	110.3	92.0	62.5	1.76	1.47

We observe that the performance of the MRP improves and gets closer to the SD approach as the target space becomes more congested. In the former case, an *emergent behavior* is observed, where the UAVs automatically spread to different regions in space and carve out their individual niches, which they survey almost independently of others.

The performances of the both methods with respect to the lower bound on optimum, however, become worse. There can be several reasons for this behavior. In the more congested space, the lower bound on the optimum is more difficult to achieve for arbitrary starting locations of UAVs. Also, as we pointed out earlier, the policy for a single UAV was only near-optimal. The percentage deviation from optimality is however expected to increase as the value of the optimum reduces. At some point, even the restriction to rectangular partitions starts affecting the performance.

A few particular cases (initial conditions) where the optimum is known to be achievable are, however, tried. If the weights of the reactive policy are tuned for those particular cases, and if the optimization in space decomposition is given enough function evaluations, we are able to achieve the optimum in both cases. What we can manage in a practical setting would depend on the actual problem specification.

4. AIRCRAFT DYNAMICS

Kovacina et al [4] have pointed out the significance of including aircraft dynamics in the control policy design, and some work in literature has considered constraints imposed by vehicle dynamics [31] [32]. However, it is not clear how much the dynamics actually affect the performance. Also, the issue of considering aircraft dynamics while designing control policies has not been sufficiently addressed. In this section we first study the effect of aircraft dynamics on the performance of the UAV. We have put together a 3-DOF aircraft dynamics simulation for this purpose, which is briefly described. Next, we impose a non-holonomic constraint that the UAV travels at constant velocity[18]. This results in a simplified dynamics model, which is used to analyze the dynamics–control policy coupling, and the performance benefits obtained by changing the control policy to account for the dynamic constraints. A minimum length trajectory (also minimum time under assumption of constant velocity) tracking controller has also been implemented and is briefly described.

3-DOF Dynamics Simulation

A 3-DOF simulation ignoring the turn rates and moments is found to be suitable for our application. In inertial coordinates the equations of motion are:

$$\dot{u} = -a_{u1}\frac{D-T}{m} - a_{u2}\frac{L}{m}$$
$$\dot{v} = -a_{v1}\frac{D-T}{m} - a_{v2}\frac{L}{m}$$
$$\dot{w} = -a_{w1}\frac{D-T}{m} - a_{w2}\frac{L}{m} + g$$

where,

$$a_{u1} = \cos\gamma\cos\psi$$
$$a_{u2} = \cos\phi\sin\gamma\cos\psi + \sin\phi\sin\psi$$
$$a_{v1} = \cos\gamma\sin\psi$$
$$a_{v2} = \cos\phi\sin\gamma\sin\psi - \sin\phi\cos\psi$$
$$a_{w1} = -\sin\gamma$$
$$a_{w2} = \cos\phi\cos\gamma$$

In the above, ***u, v, w*** are translational velocities in inertial frame, ***L, D, T*** are the lift, drag and thrust forces respectively, ***m*** is the mass of aircraft, ***g*** is the acceleration due to gravity, and ***φ, ψ, γ*** are the roll, heading, and flight path angles respectively. These are the same equations as derived by Sachs [33], except for the thrust term.

Implementation—To implement the dynamics, we define a 6-D state vector, $\boldsymbol{s = [u\ v\ w\ x\ y\ h]^T}$. There are 3 control commands - lift coefficient, $\boldsymbol{C_L}$, thrust coefficient, $\boldsymbol{C_T}$, and bank angle, ***φ***. The integration of the non-linear state equations is done using a 4th order *Runge Kutta* scheme. For controlling the UAV, we use a Linear Quadratic Regulator (LQR) based system [34]. The LQR problem formulation is similar to that in [35] and the resulting optimization problem is solved using DP [36]. The control policy returns a target cell location at each time step. However, this target cell can be arbitrarily far away (hence not reachable in unit time), so we choose the target point for the LQR controller to be distance $\boldsymbol{V_{survey}}$ away from present location, in the direction of original target cell.

Effect on Performance

We assume that we fly at constant velocity at constant altitude. In this case, the constraining factor for the performance, becomes the turn radius of the aircraft. We further assume that we have sufficient thrust to maintain a coordinated turn[19], for simplicity. So the turn radius is then constrained by the maximum value of lift coefficient, $\boldsymbol{C_{Lmax}}$. The following equations are applicable to the scenario described:

9———

[18] This assumption has been made for simplicity and will be relaxed in future work.

9———

[19] Actually this ties in to the UAV design, which is another interesting direction of research currently under investigation.

$$R_{turn} = \frac{V_{survey}^2}{g\sqrt{n_{\max}^2 - 1}}$$

$$\dot{\psi}_{\max} = \frac{g\sqrt{n_{\max}^2 - 1}}{V_{survey}}$$

$$n_{\max} = \frac{\rho V_{survey}^2 S_{ref} C_{L_{\max}}}{2mg}$$

where, $\boldsymbol{R_{turn}}$ is the minimum turn radius, $\dot{\psi}_{\max}$ is maximum turn rate, $\boldsymbol{n_{max}}$ is the maximum load factor, $\boldsymbol{\rho}$ is the density, $\boldsymbol{S_{ref}}$ is the reference wing area, $\boldsymbol{m}$ is the mass of the aircraft.

We choose a small 2 meter span UAV for our study. The parameters for the UAV are – $\boldsymbol{m}$ = 0.475kg, $\boldsymbol{S_{ref}}$ = 0.33m^2, $\boldsymbol{C_{Tmax}}$ = 0.1, $\boldsymbol{V_{survey}}$ = 5.37m/s. The UAV flies at an altitude of 200m. This results in $\boldsymbol{R_{turn}}$ = 2.96m, $\dot{\psi}_{\max} = 1.82$ rad/s. The target space is 134.25m in each dimension, and $\boldsymbol{r_{sensor}}$ = 2.69m. Using these parameters, the dynamics do not seem to affect the performance of the UAV. However, if we reduce the $\boldsymbol{C_{Lmax}}$ value, then we start observing an effect on performance. Fig. 15 shows the performance for a single UAV mission, with $\boldsymbol{C_{Lmax}}$ = 1.2, 1.0, 0.9, compared to the case with no dynamic constraints.

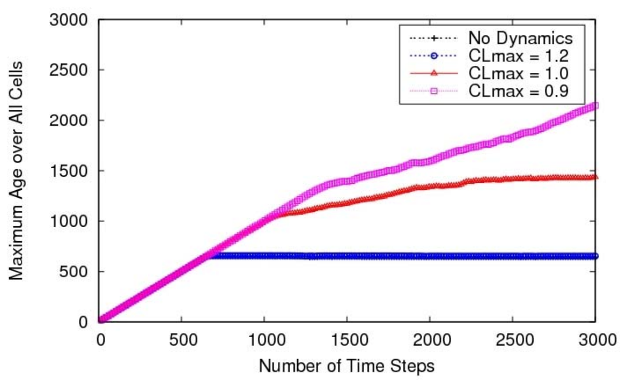

Figure 15 – Effect of C_{Lmax} on mission performance

We observe that reasonable changes to the $\boldsymbol{C_{Lmax}}$ value have a considerable effect on performance. Note that, we could similarly increase $\boldsymbol{V_{survey}}$, or some aircraft design parameter and observe similar effects. Hence, we conclude that aircraft dynamics have a significant effect on the mission performance. Whether we can change the control policy to better accommodate the aircraft dynamics or not, is the subject matter of a later section.

Simplified Dynamics Model

Assuming sufficient thrust, the constraining value for turning flight, is that of $\boldsymbol{C_{Lmax}}$, which governs the maximum side force. For the case of sustained turns, the $\boldsymbol{C_L}$ value would determine the bank angle and turn rate. The relation of turn rate to $\boldsymbol{C_L}$ was shown in the previous section, and the bank angle can be found as:

$$\varphi = \tan^{-1}\left(\frac{V_{survey}\dot{\psi}}{g}\right)$$

C_T is fixed by drag coefficient. Hence the system is essentially a single input system now. We can thus simplify the equations of motion:

$$\dot{x} = V_{survey} \cos\psi$$

$$\dot{y} = V_{survey} \sin\psi$$

$$\dot{\psi} = \frac{F_y}{mV_{survey}}$$

where, $\boldsymbol{F_y}$ is the side force – this becomes our input[20]. In this case, the minimum turn radius is defined by:

$$R_{turn} = \frac{mV_{survey}^2}{F_{y_{\max}}}$$

Note that this system is equivalent to the 3-DOF simulation described above under the given assumptions. Hence any results that we obtain using this are directly applicable to the former.

Minimum Length Trajectory Control

Given the simplified dynamics system, it is fairly easy to construct minimum length paths to target points geometrically. Thus, we design a controller (*minimum distance controller*) to travel those paths instead of the LQR control used previously.

Dubins [37] proved that the minimum length trajectory from present position and heading to a target position and heading, comprises only of straight line and maximum curvature paths (arcs corresponding to minimum turn radius). Erzberger and Lee [38] gave the specific trajectories describing these paths. Fig. 16 shows an example where the UAV has to travel from point A to point B with the initial and desired headings shown. The circles shown correspond to minimum turn radius and are tangent to the heading direction. The optimum trajectory in this case (where distance between points > 4$\boldsymbol{R_{turn}}$) would be comprised of two arcs along two of the circles and a straight path in between, along one of the tangents. The candidate paths for the optimum are numbered 1-4 in the figure. To find the minimum length trajectory, the lengths of the 4

10

[20] This is the same as choosing C_L as input, since the side force is determined by it - $F_y = \sqrt{\left((1/2)\rho V_{survey}^2 S_{ref} C_L\right)^2 - (mg)^2}$

paths are computed and the shortest one is selected. A similar analysis can be done for other cases also.

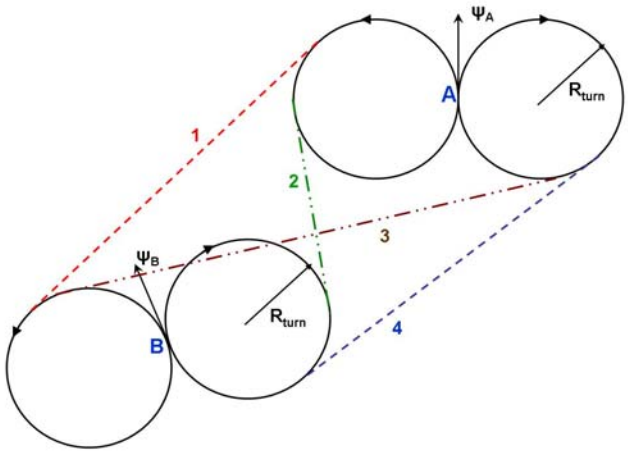

Figure 16 – Illustration showing candidate optimal paths from point A to B

Erzberger and Lee also mention the case where the final heading is not important and we just need to find the shortest path to a target point, but do not give the exact trajectories for this case. However, we can modify the trajectories for the former case by assuming a zero radius of turn at the target point[21], and obtain minimum length trajectories for the latter. Modgalya and Bhat [39] also give a feedback control law for traversing the optimum trajectory. We, however, find the optimum path and then devise our own control algorithm (minimum distance controller) to traverse it. We refrain from giving the tedious details of the implementation here.

Modifying Control Policy

In this section we look at how to modify the control policy so that we get performance improvements when the UAV is constrained by dynamics.

Euclidean vs. Actual Distance—Recall that our reactive policy for a single UAV uses a weighted combination of the age of a cell and the distance of the UAV from the cell. So far, we have used the euclidean distance in this policy. However, what the policy suggests is using the actual distance to the cell (under dynamic constraints). In our implementation, we follow the shortest path to the cell, so this distance becomes the minimum distance to the cell (which can be calculated as mentioned above, by calculating the optimal trajectories). We call the former, the Euclidean Distance Policy (EDP), and the latter is called the Actual Distance Policy (ADP).

Table 2 shows the comparison of EDP and ADP in terms of average maximum ages observed. The comparison is done for 1, 3, 5, and 10 UAVs and averaged over 50 trials. In case of multiple UAVs, we use the MRP here. The UAV parameters are – $\boldsymbol{m}$ = 1kg, $\boldsymbol{S_{ref}}$ = 0.7m^2, $\boldsymbol{C_{Tmax}}$ = 0.2, $\boldsymbol{V_{survey}}$ = 5m/s. Three values of maximum side force are used, $\boldsymbol{F_{y\ max}}$ = 5, 8, 15N. These correspond to maximum lift coefficients of C_{Lmax} = 1.03, 1.18, 1.67 and turn radii of $\boldsymbol{R_{turn}}$ = 5, 3.125, 1.67m respectively. The cell length is 5m, and the target space size is 50*50m^2 for the single UAV case, and 75*75m^2 for more UAVs. The sensor footprint equals the cell size in all cases.

Table 2. Summary of results (average maximum ages recorded) comparing EDP and ADP

	C_{Lmax} = 1.03		C_{Lmax} = 1.18		C_{Lmax} = 1.67	
#UAVs	**EDP**	**ADP**	**EDP**	**ADP**	**EDP**	**ADP**
1	261.7	255.8	198.5	196.1	108.4	107.1
3	235.9	209.2	194.7	179.0	145.9	144.4
5	226.3	196.9	188.2	174.7	145.7	141.8
10	124.6	106.6	101.3	94.5	80.3	79.0

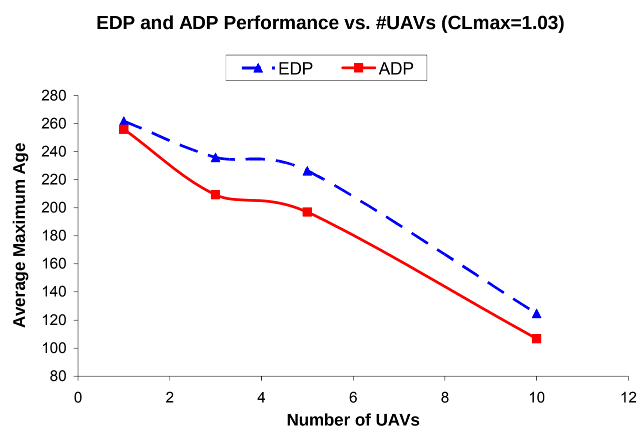

Figure 17 – Average maximum ages using EDP and ADP as function of number of UAVs for C_{Lmax}=1.03

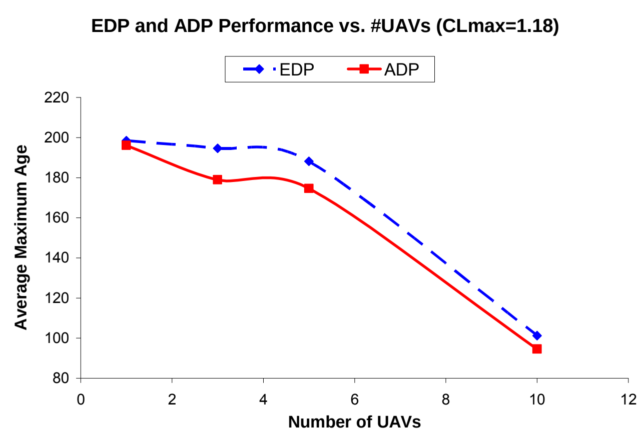

Figure 18 – Average maximum ages using EDP and ADP as function of number of UAVs for C_{Lmax}=1.18

11

[21] This is the same as saying that the UAV can instantly change heading at the target point, or in other words, the heading does not matter.

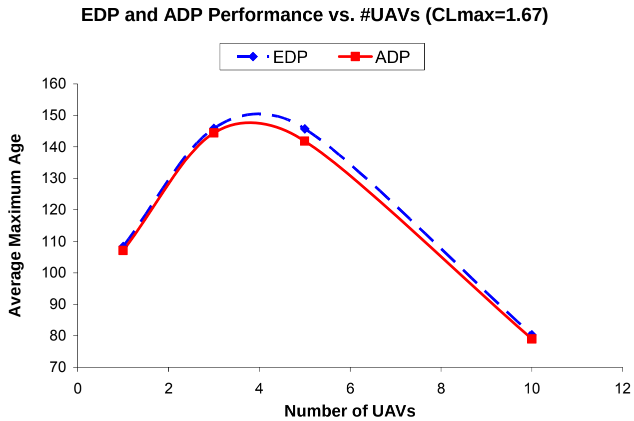

Figure 19 – Average maximum ages using EDP and ADP as function of number of UAVs for C_{Lmax}=1.67

The results shown in Figs. 17-19 plot the average maximum ages observed using EDP and ADP as a function of the number of UAVs for each of value of C_{Lmax}. We can see that ADP performs better than EDP when the dynamics are a constraining factor. The performance benefits are greater for more number of UAVs, but tend to saturate. Also, if dynamic constraints are not stringent, the benefits might actually reduce for greater number of UAVs.

This can be explained as follows. Due to limits on the side force, the turning radius is constrained, and thus a UAV tends to leave "gaps" while searching the environment. In case of multiple UAVs, the other UAVs tend to fill the gaps and hence improve the performance. This is another form of *emergence* that we observe in the MRP.

Heading Considerations—So far we have disregarded the target heading (heading at the target location) of the UAV altogether. This leads us to the question – is there a heading that is more desirable than others when the aircraft reaches the target? The precise answer to this question entails an intractable planning problem, so we used a simplified analysis to account for the heading. Some performance benefits were obtained using this method, but the basic results did not change, so we will not present the details here.

5. CONCLUSIONS

In this study we have defined an approach to multiple UAV persistent surveillance based on an optimum policy for a single-UAV case. The policy is compared with selected benchmark methods and is found to work well. The extension to the multiple UAV case has been done using two approaches. The emphasis has been on techniques that would work with environment dynamics and UAV failures and are simple in concept. The approaches include a purely reactive technique (MRP) and one that explicitly divides the work between UAVs (SD). The latter method seems to be close to the optimal case, but may suffer from computational complexity when the number of UAVs becomes very large. The optimal assignment algorithm developed for the allocation problem has also shown good results. The MRP is more heuristic in nature, but highly scalable, robust, and simple to implement. It shows an emergent behavior as the search space becomes more congested – replacing the SD method for a large number of UAVs with modest resources. We have also developed a multiple UAV simulation environment to study these policies.

A 3-DOF simulation was used to evaluate the effect of UAV dynamics. The effect of dynamic constraints on performance was evaluated and the control policy was modified to better accommodate UAV dynamics.

Future research will include evaluation of environment dynamics and UAV failures, comparing the two policies (MRP and SD) in this more challenging scenario.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. Stephen Rock and Prof. Yoav Shoham for their discussions and feedback regarding certain aspects of this paper. We also acknowledge the Boeing Company for funding this research work and Dr. Stefan Bieniawski and Dr. John Vian for their many suggestions for interesting problems and approaches.

REFERENCES

[1] Lynne E. Parker, "Current State of the Art in Distributed Autonomous Mobile Robotics," Distributed Autonomous Robotic Systems 4, Vol. 4, 3–12, October 2000.

[2] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng, "Cooperative Mobile Robotics: Antecedents and Directions," Autonomous Robots, Vol. 4, Issue 1, 7–27, 1997.

[3] Howie Choset, "Coverage for Robotics – A survey of Recent Results," Annals of Mathematics and Artificial Intelligence, Vol. 31, 113–126, 2001.

[4] Michael A. Kovacina, Daniel Palmer, Guang Yang, and Ravi Vaidyanathan, "Multi-agent Control Algorithms for Chemical Cloud Detection and Mapping using Unmanned Air Vehicles," IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3, 2782–2788, 2002.

[5] Matthew Flint, Marios Polycarpou, and Emmanuel Fernandez-Gaucherand, “Cooperative Path Planning for Autonomous Vehicles Using Dynamic Programming,” Proceedings of the 15th Triennial IFAC World Congress, Vol. P, 481–487, July 2002.

[6] Matthew Flint, Emmanuel Fernandez-Gaucherand, and Marios Polycarpou, “Stochastic Models of a Cooperative Autonomous UAV Search Problem,” Military Operations Research, Vol. 8, No. 4, 13–33, 2003.

[7] Yanli Yang, Marios Polycarpou, and Ali A. Minai, “Opportunistically Cooperative Neural Learning in Mobile Agents,” Proceedings of the IEEE/INNS International Joint Conference on Neural Networks, IJCNN, 2638-2643, 2002.

[8] P.B. Sujit, and D. Ghose, “Search Using Multiple UAVs with Flight Time Constraints,” IEEE Transactions on Aerospace and Electronic Systems, Vol. 40, No.2, 491–510, April 2004.

[9] Arthur Richards, John S. Bellingham, Michael Tillerson, and Jonathan How, “Coordination and Control of Multiple UAVs," AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA-2002-4588, August 2002.

[10] Christopher M. Clark, Stephen M. Rock, and Jean-Claude Latombe, “Motion Planning for Multiple Mobile Robot Systems using Dynamic Networks,” IEEE International Conference on Robotics and Automation, Vol. 3, 4222–4227, September 2003.

[11] Wolfram Burgard, Dieter Fox, Mark Moors, Reid Simmons, and Sebastian Thrun, “Collaborative Multi-Robot Exploration,” Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, 476–481, April 2000.

[12] K. Madhava Krishna, Henry Hexmoor, Subbarao Pasupuleti, and James Llinas, “Parametric Control of Multiple Unmanned Air Vehicles over an Unknown Hostile Territory,” Proceedings of International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 117–121, April 2005.

[13] De Witt Latimer IV, Siddhartha Srinivasa, Vincent Lee-Shue, Samuel Sonne, Howie Choset, and Aaron Hurst, “Towards Sensor Based Coverage with Robot Teams,” Proceedings of IEEE International Conference on Robotics and Automation, ICRA’02, Vol.1, 961–967, May 2002.

[14] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo, “Coverage Control for Mobile Sensing Networks,” Proceedings of IEEE International Conference on Robotics and Automation, Vol. 2, 1327–1332, 2002.

[15] Islam I. Hussein, and Dusan M. Stipanovic, “Effective Coverage Control for Mobile Sensor Networks With Guaranteed Collision Avoidance,” IEEE Transactions on Control Systems Technology, Vol. 15, Issue 4, 642–657, July 2007.

[16] H. Van Dyke Parunak, “Making Swarming Happen,” Conference on Swarming and Network Enabled Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance, Virginia, 2003.

[17] Dale A. Lawrence, Ryan E. Donahue, Kamran Mohseni, and Richard Han, “Information Energy for Sensor-Reactive UAV Flock Control,” 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit, AIAA–2004–6530, September 2004.

[18] Stefano Caselli, Monica Reggiani, and Roberto Sbravati, “Parallel Path Planning with Multiple Evasion Strategies,” IEEE International Conference on Robotics and Automation, ICRA’02, Vol. 1, 260–266, 2002.

[19] Kagan Tumer, and Adrian Agogino, “Coordinating Multi-Rover Systems: Evaluation Functions for Dynamic and Noisy Environments,” Proceedings of 2005 Genetic and Evolutionary Computation Conference, 591–598 , June 2005.

[20] Kagan Tumer, Adrian Agogino, and David H. Wolpert, “Learning Sequences of Actions in Collectives of Autonomous Agents,” Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems, 378–385, July 2002.

[21] Charles A. Erignac, “An Exhaustive Swarming Search Strategy based on Distributed Pheromone Maps,” Infotech@Aerospace Conference and Exhibit, AIAA-2007-2822, May 2007.

[22] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, Probabilistic Robotics, Cambridge: MIT Press, 569–601, 2005.

[23] Dimitri P. Bertsekas, “Auction Algorithms,” Encyclopedia of Optimization, Kluwer, 2001.

[24] Dimitri P. Bertsekas, “Auction Algorithms for Network Flow Problems: A Tutorial Introduction,” Computational Optimization and Applications, Vol.1, 7–66, 1992.

[25] Dev Rajnarayan, Ilan Kroo, and David Wolpert, "Probability Collectives for Optimization of Computer Simulations," 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA-2007-1975, April 2007.

[26] Matthew Flint, Emmanuel Fernandez-Gaucherand, and Marios Polycarpou, "Cooperative Control for UAVs Searching Risky Environments for Targets," Proceedings of the 42nd IEEE Conference on Decision and Control, Vol. 4, 3567–3572, December 2003.

[27] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, Introduction to Algorithms, Cambridge: MIT Press, 527–532, 1990.

[28] Daisuke Kurabayashi, Jun Ota, Tamio Arai, and Eiichi Yoshida, "An Algorithm of Dividing a Work Area to Multiple Mobile Robots," Proceedings of the International Conference on Intelligent Robots and Systems, Vol. 2., 286–291, August 1995.

[29] Ivan Maza, and Anibal Ollero, "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms," 7th International Symposium on Distributed Autonomous Robotics Systems, Toulouse, June 2004

[30] Robert S. Garfinkel, "An Improved Algorithm for Bottleneck Assignment Problem," Operations Research, Vol.19, No.7, 1747–1751, Nov-Dec 1971.

[31] Yong Liu, Jose B. Cruz, and Andrew G. Sparks, "Coordinating Networked Uninhabited Air Vehicles for Persistent Area Denial," 43rd IEEE Conference on Decision and Control, Vol. 3, 3351–3356, December 2004.

[32] Jarurat Ousingsawat, and Mark E. Campbell, "Optimal Cooperative Reconnaissance using Multiple Vehicles," Journal of Guidance Control and Dynamics, Vol. 30, No. 1, 122–132, 2007.

[33] Gottfried Sachs, "Minimum shear wind strength required for dynamic soaring of albatrosses," IBIS–London–British Ornithologists Union, Vol. 147, No. 1, 1–10, January 2005.

[34] Brian L. Stevens, and Frank L. Lewis, Aircraft Control and Simulation, New York: John Wiley and Sons Inc., 1992.

[35] Adam W. Divelbiss, and John T. Wen, "Trajectory Tracking Control of a Car-Trailer System," IEEE Transactions on Control Systems Technology, Vol. 5, No. 3, 269–278, May 1997.

[36] Stanford EE-363 Course Website http://www.stanford.edu/class/ee363

[37] L. E. Dubins, "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," American Journal of Mathematics, Vol. 79, No. 3, 497-516, July 1957.

[38] Heinz Erzberger, and Homer Q. Lee, "Optimum Horizontal Guidance Techniques for Aircraft," Journal of Aircraft, Vol. 8, No. 2, 95-101, February 1971.

[39] Mrityunjay Modgalya, and Sanjay P. Bhat, "Time-Optimal Feedback Guidance in Two Dimensions under Turn-Rate and Terminal Heading Constraints," National Conference on Control and Dynamical Systems, NCCDS'05, India, Jan 2005.

BIOGRAPHY

Nikhil Nigam *is a PhD student at Stanford University in the department of Aeronautics and Astronautics. He received his Bachelors at Indian Institute of Technology (IIT) - Bombay in Aerospace engineering and his Masters at Stanford University in Aeronautics and Astronautics. His primary interest is in optimization for aircraft design. His research has focused on aircraft control and design issues with emphasis on multi-agent systems and artificial intelligence.*

Ilan Kroo *is a Professor of Aeronautics and Astronautics at Stanford University. He received his B.S. degree in Physics from Stanford in 1978, then continued studies at Stanford in Aeronautics, leading to a Ph.D. degree in 1983. He worked in the Advanced Aerodynamic Concepts Branch at NASA's Ames Research Center for four years before returning to Stanford as a member of the Aero/Astro faculty. Prof. Kroo's research in aerodynamics and multidisciplinary design optimization includes the study of innovative airplane concepts for reduced environmental impact and efficient supersonic flight.*