Control and Design of Multiple Unmanned Air Vehicles for a Persistent Surveillance Task

Nikhil Nigam[*]
and Ilan Kroo[†]

Stanford University, Stanford, California, 94305, USA

Control of multiple autonomous vehicles is a topic of current research interest for applications such as weather monitoring, geographical surveys, tactical reconnaissance and extra-terrestrial exploration. The task of persistent surveillance is different from a coverage/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, though ideas from these methods can still be used. The dynamics of air vehicles add additional complexity to the autonomous control problem, while the design of a fleet of Unmanned Air Vehicles (UAVs) for desirable system performance involves problems that have not been sufficiently explored. In this research, we investigate techniques for high-level control that are scalable, reliable, efficient, and robust to problem dynamics (new information about the target area, failure of UAVs, etc.). We suggest a modification of the control policy to account for aircraft dynamics. We then describe an architecture for System-of-Systems (SoS) design based on Collaborative Optimization (CO), and present certain results for single UAV design.

Nomenclature

$V_{mission}$	Mission velocity
A_j	Age of j^{th} cell
$\mathcal{V}_j$	Value of j^{th} cell
w_0, w_1	Control policy weights
δ_{ij}	Distance between i^{th} UAV and j^{th} cell
r_{sensor}	Radius of sensor footprint
N_{UAV}	Number of UAVs
u, v, w	Translational velocities (inertial), m/s
L	Lift force, N
D	Drag force, N
T	Thrust force, N
m	Mass of aircraft, Kg
g	Acceleration due to gravity, m/s^2
ϕ	Roll angle, rad
ψ	Yaw angle, rad
γ	Flight path angle, rad
C_L	Coefficient of lift
C_T	Coefficient of thrust
R_{turn}	Minimum radius of turn of UAV, m
n_{max}	Maximum load factor
ρ	Density of air, kg/m^3
S_{ref}	Reference wing area, m^2

[*] PhD Student, Department of Aeronautics and Astronautics, Stanford University, and AIAA Student Member.

[†] Professor, Department of Aeronautics and Astronautics , Stanford University, and AIAA Fellow.

W_{gross}	Maximum takeoff gross weight, N
T_{SLS}	Sea level static thrust, N
AR	Aspect ratio
V_{stall}	Stall velocity, m/s
T_{cruise}	Cruise thrust, N
D_{cruise}	Cruise drag, N
W_{ZFW}	Zero fuel weight, N
α_i	Weighting parameters in SoS architecture (system level)
β_{ij}	Weighting parameters in SoS architecture (subspace level)

I. Introduction

I.A. Background

There has been a growing interest in control and coordination of autonomous vehicles in the fields of Artificial Intelligence (AI) and controls.[1,2] In particular, the task of search/exploration/coverage has received significant attention in the past two decades.[3–5] Dynamic Programming (DP),[6,7] Mixed Integer Linear Programming (MILP),[8] shortest path planning[9] and Spanning Tree Coverage (STC)[10,11] algorithms are popular planning based methods. Traditional AI search techniques like A* and its variants have also been applied to search tasks,[12,13] but do not address the problem of cooperation among multiple vehicles. A vast amount of literature deals with problems involving obstacles and sensing uncertainties.[14] Space decomposition-based methods (such as boustrophedon,[15] morse,[16] and voronoi[17]) have proven effective to deal with such problems.[a] Market-based mechanisms have also been used to divide work among vehicles,[19,20] but have been applied to limited set of problems. Coordination field methods, that include particle swarm optimization, potential functions-based approaches, and digital pheromone mechanisms[12,21,22] are simple and highly scalable, but often suffer from problems of local minima.

Most of these methods for high-level vehicle control described in the literature 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).[23,24] The other class involves approaches that are decentralized and scalable but heuristic.[25,26] Some of the techniques cannot be applied in an online setting and may not be useful for sensor-based coverage. Many techniques either ignore vehicle dynamics or treat them independently of the control scheme. Although some work has focused on problems similar to persistent surveillance,[12,13,27,28] the application of most of these techniques to the problem is not straightforward.

Using Unmanned Air Vehicles (UAVs) has gained considerable attention owing to their relative indifference to terrain and greater range compared to ground vehicles.[2] Bellingham et al.[8,24,29] study the problem of cooperative path planning and trajectory optimization in UAV context. Flint, Polycarpou and Fernandez-Gaucherand[7,23] study UAV cooperation for search. Lawrence, Donahue, Mohseni and Han[21] use large Micro Air Vehicle (MAV) swarms for toxic plume detection. A lot of studies (such as those described in [27,30,31]) have concentrated on military applications. Kovacina, Palmer, Yang and Vaidyanathan[32] claim that the design of control laws for aerial vehicles ties in with the aircraft dynamic constraints. However, most of existing work ignores the coupling between the control laws and aircraft dynamics.

A relatively unexploited, but extremely rich area of research is simultaneous design of the UAVs and their control laws, or System of Systems (SoS) design.[33–35] DeLaurentis, Kang, Lim, Mavris, and Schrage[36] apply SoS engineering to model personal air vehicles, but most literature in this area does not talk about concrete implementation details. Recent work by Underwood and Baldesarra[37] describes a framework for coupling design and mission performance, but requires human-in-the-loop interaction. Frommer and Crossley[38,39] design a fleet of morphing aircraft for a search mission. However, they use a hierarchical architecture for SoS design, which has associated limitations for our problem. Some other Multidisciplinary Design Optimization(MDO) architectures (such as Collaborative Optimization (CO),[40] Modified CO (MCO)[41] or Analytical Target Cascading (ATC)[42]) could perhaps be used for this problem.

[a] Acar and Choset[18] provide a brief survey of existing decomposition-based techniques.

I.B. Our Work

In the present problem, the target space (physical area to be surveyed) is gridded using an approximate cellular decomposition.[b] Each cell has an associated age, that is the time elapsed since it was last observed. The goal of persistent surveillance is to minimize the maximum age over all cells observed over a long period of time. In other words, no area of the target space should be left unobserved for a long time. This task is different from an exploration problem, where the ages of the cells are not important.[c] It also differs from problems of minimizing map uncertainty, where the cumulative uncertainty (as opposed to maximum uncertainty of a cell) is the quantity of interest.

In this work, we first devise a control law for high-level control of a single UAV. We investigate a semi-heuristic approach, which is optimum for a simple case, and extend it to a more complex and realistic sensor-based coverage problem. The approach is compared to a heuristic potential function method, a DP based planning method and a bound on the optimum performance. We then study an approach for multiple-UAV coordination, that is an extension of the reactive policy for a single UAV. We call this the *Multi-agent Reactive Policy* (MRP).

We next study the effect of aircraft dynamics (using a 3-DOF dynamics model) on the mission performance. We assume the aircraft to fly at roughly constant altitude, so the altitude does not affect the sensing. We further assume that the aircraft travels at constant velocity to simplify the dynamics model and study the coupling between the control policy and aircraft dynamics. We then propose a modification to the policy to improve mission performance under dynamic constraints. A minimum-length trajectory tracking controller used for this purpose is also described and implemented.

The other important aspect of the persistent surveillance problem is the design of the UAVs. We develop a simple design analysis code for this purpose, and study the variation in design for different performance targets. We then examine a CO-based architecture for the SoS design problem. We look at how the mission requirements affect the aircraft design for a single UAV. Finally, we draw conclusions about our study and outline future work.

II. Policy for Single UAV

II.A. Policy Structure

We begin by considering a 1-D problem with two cells that need to be visited so as to minimize the maximum of the ages of the cells. The UAV, stationed at distance x from the left cell (see figure 1), is assumed to travel at constant velocity, $V_{mission}$, and A_j denotes the age of the j^{th} cell. Without loss of generality, we assume unit distance between the cells with x and $V_{mission}$ scaled accordingly.

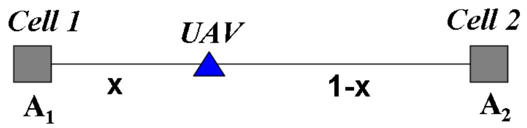


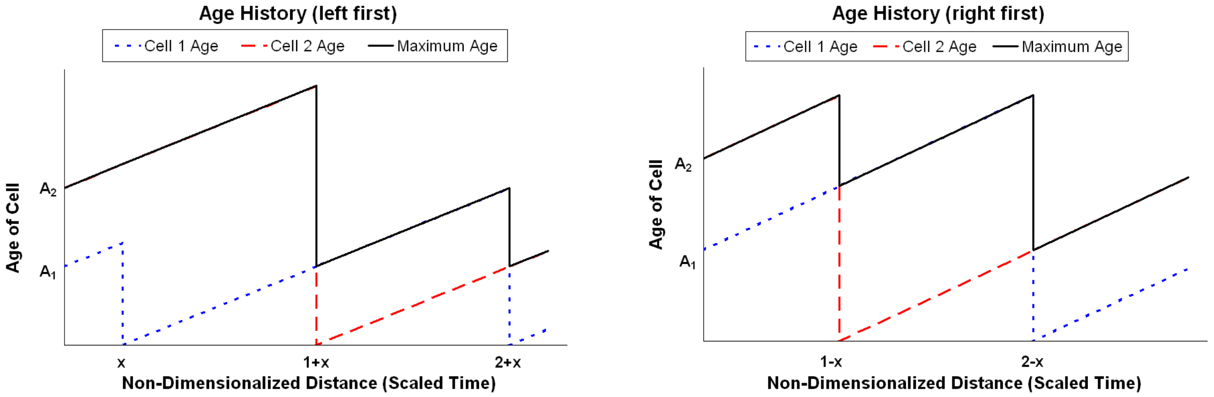
Figure 1. Simple 1-D two-cell problem used to derive structure for control policy.

The UAV can choose to go either left or right.[d] After the UAV has chosen which cell to observe first, the optimum policy is to keep moving back and forth between the cells. Hence a single action defines the optimum policy in this case. Assuming $A_1 \leq A_2$, we can construct a plot of the ages of cells for the case where the UAV chooses to go left or right first (figure 2). This is used to identify the maximum age (over the two cells) as a function of time, and our optimum policy tries to minimize the peak of this maximum age curve.

[b] Approximate cellular decomposition means the sensor footprint equals the cell size,[4] and in our case, the cells exhaustively cover the space.

[c] It is only relevant to know if a cell has been explored or not.

[d] This is the same as choosing to go to cell 1 first or cell 2 first, since any policy which makes a UAV turn back before reaching a cell is necessarily suboptimal.



(a) UAV chooses cell 1

(b) UAV chooses cell 2

Figure 2. Plot of the maximum age curve (i.e. the maximum of the ages of the two cells, as a function of time/scaled distance) when the UAV chooses to go left or right first.

We need to consider the maximum age plot only till time $2/V_{mission}$ since we already know the optimum policy after a cell is reached. Let $A_{max_{left}}$ and $A_{max_{right}}$ denote the peaks of the maximum age curves when the UAV chooses left and right respectively. The decision to choose left or right depends on the ages of the cells and the distance of the UAV from the cells, as illustrated in Eq. (1).

$$\begin{aligned}\text{Choose left} \quad &\Leftrightarrow \quad A_{max_{left}} \leq A_{max_{right}} \\ &\Leftrightarrow \quad \max\left\{\left(A_2 + \frac{1+x}{V_{mission}}\right), \left(\frac{2}{V_{mission}}\right)\right\} \\ &\leq \quad \max\left\{\left(A_2 + \frac{1-x}{V_{mission}}\right), \left(A_1 + \frac{2-x}{V_{mission}}\right), \left(\frac{2}{V_{mission}}\right)\right\} \qquad (1)\end{aligned}$$

These equations (and corresponding ones for $A_1 \geq A_2$) are solved for all possible cases,[e] resulting in Eq. (2).

$$\begin{aligned}&\text{If } A_1 \leq \frac{x}{V_{mission}}, \text{ignore cell 1} \\ &\text{If } A_2 \leq \frac{1-x}{V_{mission}}, \text{ignore cell 2} \\ &\text{If neither is true, choose left} \Leftrightarrow A_1 - \frac{x}{V_{mission}} \geq A_2 - \frac{1-x}{V_{mission}} \qquad (2)\end{aligned}$$

We can in fact make the control policy more concise by defining a value associated with each cell, as in Eq. (3). Here, $\mathcal{V}_j$ is the value of the j^{th} cell, w_0 is a weighting parameter ($w_0 = -1/V_{mission}$ for this two-cell case), and δ_{1j} is the distance between the UAV and j^{th} cell.

$$\mathcal{V}_j = \max\left\{(A_j + w_0\delta_{1j}), 0\right\} \qquad (3)$$

The UAV calculates the value for both cells and goes to the cell with the maximum value. This control policy is in fact optimal for two cells in 2-D as well - that is evident through an analysis analogous to the above.

II.B. Extension to 2-D Multiple-cell Case

The policy structure derived above, can be extended to the more realistic 2-D multiple cell scenario in two ways: one is to combine values of multiple cells to find the direction to go[f] (*Sum of Value* approach), and the other is to go towards the cell with the maximum value (*Target based* approach).

[e] We give only the summary of the results here for brevity.
[f] In 2-D the addition is vectorial.

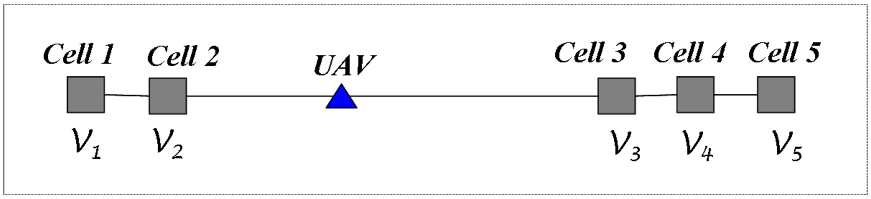


Figure 3. Example 1-D problem with five cells to illustrate the difference between sum of value approach and target based approach.

We consider a case with 5 cells in 1-D, as shown in figure 3, to resolve this issue. Let us assume values of cells such that $\mathcal{V}_1 > \mathcal{V}_2$ and $\mathcal{V}_4 > \mathcal{V}_3 > \mathcal{V}_5$. We further assume that the distances between cells on either side are not significant compared to their distances from the UAV.[g] Under these assumptions, we need to consider only the maximum valued cells on either side in order to decide the direction to go (since these become the critical cells contributing to the maximum age). So we choose to go left iff $\mathcal{V}_1 \geq \mathcal{V}_4$. Even in the case where the cells on either side are not very close, it makes intuitive sense not to combine the values of the cells (since we are concerned only with the maximum age observed).

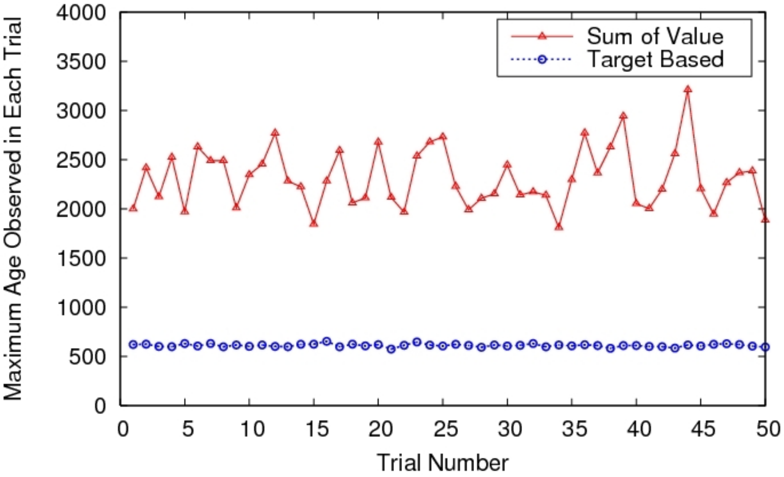


Figure 4. Comparison of sum of value approach to the target based approach for sample problem scenario.

To substantiate our claim, we compare the sum of value approach to the target based approach. The target space is square and assumed to have unit length in both dimensions. The sensor footprint of each UAV is a circle of radius, $r_{sensor} = 0.025$ and the mission velocity is $V_{mission} = 0.03$.[h] Figure 4 shows the maximum age observed over a long time period, for 50 trials (with random initial starting position of the UAV), using the two policies. It is evident that the target based approach works much better than the sum of value approach. So the control law for the UAV involves finding the values of all cells at each time step and moving towards the cell with maximum value. If there are multiple cells with the same maximum value, then instead of choosing randomly between them, the UAV chooses the one with least heading change.

We realize that value of the weight derived for the two-cell problem (w_0 = -1/$V_{mission}$), need not be optimal for the multiple-cell case. So we find the optimum value of the weight using an iterative sampling based optimizer, ISIS. This nongradient, population-based optimizer was developed by one of the authors (see [43] for example). The optimization is offline and the objective function is the actual mission cost (i.e. the maximum age observed over all cells). The optimum weight found is in fact close to our analytical optimum for the simple case, which also indicates that extending the policy as above is reasonable.

II.C. Testing Policy Performance

In this section we compare the policy with certain benchmark techniques to see how well it performs. A random action policy performs very poorly since certain regions in the target space are always left unexplored, so we do not show results of comparison with it.

[g] This ensures that the optimum policy is completely defined by choosing left or right direction.

[h] These quantities are non-dimensionalized w.r.t. target space dimension.

II.C.1. Comparison with a Potential Field Like Approach

We first compare our approach to a heuristic policy similar to work by Tumer and Agogino[26] on a multi-rover exploration problem. In our implementation, we use a linear control policy, which is found to work better than training neural networks online using an evolutionary algorithm. This approach (which is similar to a potential field approach) is then compared to our target based policy on a target space of unit length, with $r_{sensor} = 0.025$, and $V_{mission} = 0.03$. The plots of maximum age over a long time period for 50 trials are shown in figure 5. We observe that the target based approach performs significantly better.

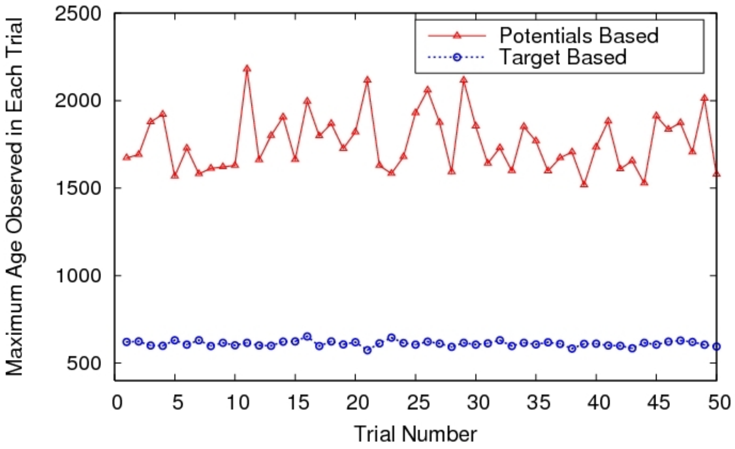


Figure 5. Comparison of a heuristic policy, similar to a potential field method, and our target based approach.

II.C.2. Comparison with a Planning-based Approach

A popular planning algorithm used in literature is the *Dijkstra's shortest path algorithm*.[44] This algorithm is greedy, but finds the optimum for our problem, since it involves a directed graph with nonnegative weights. We modify the algorithm described in [45] to obtain a *longest* path algorithm. The nodes in the graph correspond to the cells in the grid and the weights of the edges are ages of the next cell. The number of nodes in the graph are order $\tilde{O}(20^h)$, where h is the finite time horizon. We implement the planning algorithm for time horizons of up to 3 steps (due to computational limitations) and find that it performs much worse than our reactive policy. This is expected, since the reactive policy tries to convert the time-extended problem into a single step problem, by incorporating a measure of time through distance weighted by velocity. Hence it potentially looks at an infinite time horizon.

II.C.3. Emergence of Search Pattern and Comparison with Optimum

Now consider a special case, where the UAV starts from one corner of the target space. The target based policy then results in a spiral search pattern as shown in figure 6. Basically the UAV spirals in to the center and then returns to the starting location, repeating the pattern. This pattern is not optimal, but it is reasonably close to it, with the additional advantage of being able to react to problem dynamics or failures.

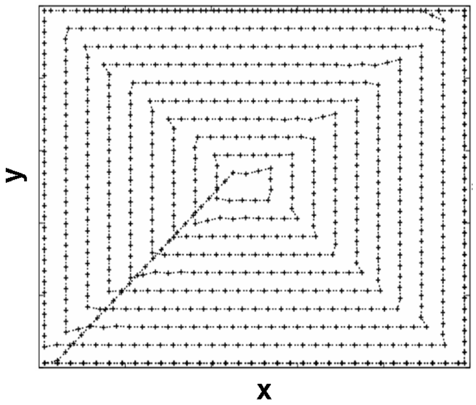


Figure 6. A spiral search pattern emerges from the target based policy under certain conditions.

We further compare the performance of the policy to a lower bound on the optimum.[i] We consider the same target space with $r_{sensor} = 0.02$, and $V_{mission} = 0.04$, for this purpose. Figure 7 shows the maximum age over all cells as a function of number of time steps. The results have been averaged

[i] This equals the number of cells in the domain, and is the best that any policy can do, since the UAV moves one cell-length in one time step.

over 50 trials. We can see that the performance of the target based approach is pretty close to the bound on optimum.

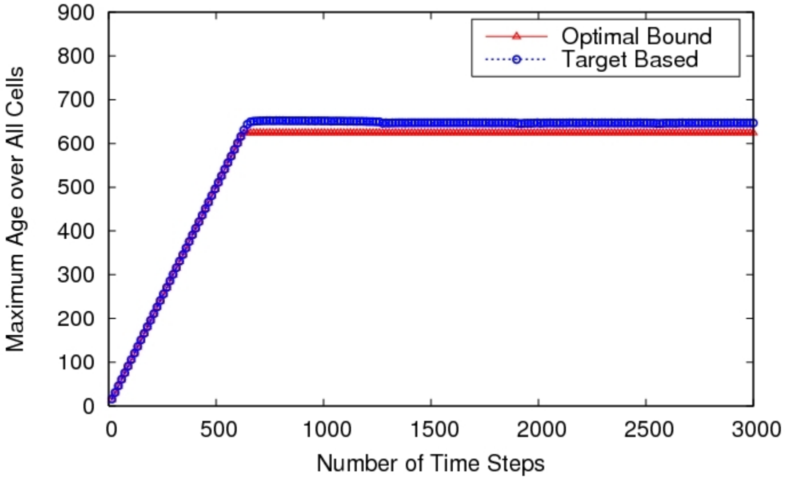


Figure 7. Comparison of target based approach to a bound on the optimum.

III. Policy for Multiple UAVs

A lot of schemes for coordination among multiple vehicles have been proposed in literature. Our focus in this work has been on techniques that are robust, scalable, and simple in concept. We propose such a technique for multi-UAV coordination, that is a simple extension of the reactive policy for a single UAV.

III.A. Multi-agent Reactive Policy

Once again, we look at a simple 1-D, two-cell, two-UAV case to understand how the existing policy can be extended to the multiple UAV case. We consider the case shown in figure 8, for our analysis.[j]

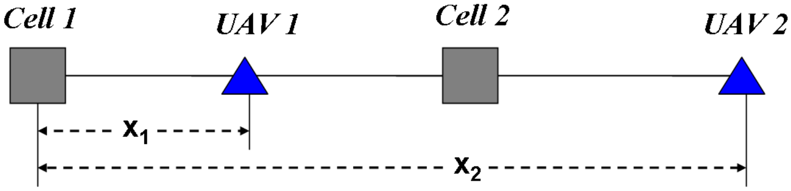


Figure 8. Simple 1-D problem with two cells and two UAVs.

Without loss of generality, we assume unit distance between the cells. It is easy to see that UAV 2 should move to cell 2, so we need to find the control policy for UAV 1. An analysis similar to the single UAV case, results in Eq. (4).

$$\begin{aligned}\text{Choose left} \quad &\Leftrightarrow \quad A_2 + \frac{x_2 - 1}{V_{mission}} \leq A_1 + \frac{2 - x_1}{V_{mission}} \\ &\Leftrightarrow \quad \left(A_2 - \frac{1 - x_1}{V_{mission}} + \frac{x_2 - 1}{V_{mission}}\right) \leq \left(A_1 - \frac{x_1}{V_{mission}} + \frac{x_2}{V_{mission}}\right) + \frac{x_1}{V_{mission}} - \frac{x_2 - 1}{V_{mission}}\end{aligned} \tag{4}$$

This is optimum only for the case $(x_2 - 1) \leq x_1$, but we still use the structure to motivate our policy for the case of multiple UAVs. The value of each cell is now given by Eq. (5), where w_1 is an additional weighting

[j] Note that other possible arrangements of the UAVs are either antisymmetric to this case, or have trivial solutions.

parameter for the distance of the cell to the nearest other UAV.[k]

$$\mathcal{V}_j = \max\left\{\left(A_j + w_0\delta_{ij} + w_1 \min_{k\neq i}\left(\delta_{kj}\right)\right), 0\right\} \tag{5}$$

The control policy weights need to be optimized next. This is analogous to the optimization for the single UAV policy, and we use ISIS for optimization. We allow the control policies for different UAVs to be different, so we need to reoptimize if the group size of UAVs changes. This performs much better than the case of no coordination between UAVs, but before making any claims about the approach, we need to compare it to some other method as well. The method is compared to a space decomposition based approach, that involves allocating subspaces to UAVs for parallel surveillance. We observe that the performance of the MRP improves and gets closer to the latter approach as the number of UAVs increase. In fact, an emergent behavior is observed for MRP - in congested spaces the UAVs tend to spread to different regions in space and create their individual niches, that they survey almost independently of others. The details of the results can be found in [46].

IV. Incorporating Aircraft Dynamics

Work in [32] claims that though devoid of terrain considerations, UAVs have to consider aircraft dynamic constraints in their control policies. Certain existing work has considered constraints imposed by vehicle dynamics,[47,48] but the problem of coupling between dynamic constraints and control policy 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.

IV.A. 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 given by Eqs. (6).[l] Here, u, v, w refer to translational velocities in inertial frame, L, D, T are the lift, drag and thrust respectively, m is the mass of the aircraft, g is the acceleration due to gravity, and ϕ, ψ, γ are the roll, yaw and flight path angles.

$$\begin{aligned}
\dot{u} &= -a_{u1}\left(\frac{D-T}{m}\right) - a_{u2}\left(\frac{L}{m}\right)\\
\dot{v} &= -a_{v1}\left(\frac{D-T}{m}\right) - a_{v2}\left(\frac{L}{m}\right)\\
\dot{w} &= -a_{w1}\left(\frac{D-T}{m}\right) - a_{w2}\left(\frac{L}{m}\right) + g
\end{aligned} \tag{6}$$

where,

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

The control inputs for the dynamical system are: lift coefficient, C_L, thrust coefficient, C_T, and roll angle, ϕ. The control commands are found using *Linear Quadratic Regulator* (LQR) control,[50,51] with the optimization problem solved using DP.[52]

IV.B. Effect of Dynamic Constraints on Mission Performance

[k] This policy makes intuitive sense as well, since a UAV should not go to a cell that is already close to another UAV.

[l] These equations are the same as derived by Sachs,[49] except for the thrust terms, since he considered gliders.

We now quantify the effect of dynamic constraints on performance. We assume that the UAV flies at constant altitude, so the constraining factor is the minimum radius of turn of the UAV, R_{turn}. We also assume that the aircraft has sufficient thrust for a sustained turn. So R_{turn} is governed by maximum lift coefficient, $C_{L_{max}}$, as shown in Eq. (7). Here, n_{max} is the maximum load factor, ρ is the air density at altitude (assumed to be 20000 ft) for our simulations, and S_{ref} is the reference wing area.

$$\begin{aligned} R_{turn} &= \frac{V_{mission}^2}{g\sqrt{n_{max}^2 - 1}} \\ n_{max} &= \frac{\rho V_{mission}^2 S_{ref} C_{L_{max}}}{2mg} \end{aligned} \qquad (7)$$

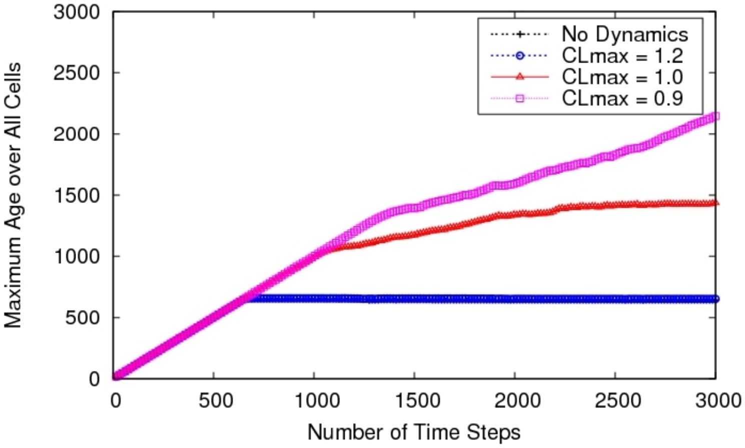


Figure 9. Maximum age as a function of time plotted for the case of a single UAV, with $C_{L_{max}}$ = 0.9, 1.0, 1.2, and compared to the case of no dynamic constraints.

The aircraft we choose for our study is a small 2 m span UAV, with $m = 0.475$ kg, $S_{ref} = 0.33\ m^2$, $C_{T_{max}} = 0.1$, $V_{mission} = 5.37$ m/s, $R_{turn} = 2.96$ m, and flying at an altitude of 200 m. The target space is 134.25 m in each dimension. Figure 9 shows the mission performance (maximum age observed as a function of time) for different values of $C_{L_{max}}$. The performance curve with $C_{L_{max}} = 1.2$ almost coincides with the curve without dynamic constraints, but a reasonable variation in $C_{L_{max}}$ can cause huge mission performance penalties. Note that increasing the value of $V_{mission}$ would have a similar effect on the performance as well. So we infer that we need to consider the coupling between aircraft dynamics and high level control.

IV.C. Dynamics Model with Nonholonomic Constraint

For the purpose of studying the interaction between the control policy and aircraft dynamics, we further simplify the model, by introducing the nonholonomic constraint of constant velocity. Assuming we have sufficient thrust (or can lose slight altitude while turning), the system becomes a single input system, with the side force (directly related to C_L) as our control input. The equations of motion for this system are given by Eqs. (8).

$$\begin{aligned} \dot{x} &= V_{mission}\cos\psi \\ \dot{y} &= V_{mission}\sin\psi \\ \dot{\psi} &= \frac{F_y}{mV_{mission}} \end{aligned} \qquad (8)$$

The minimum radius of turn can be determined from Eq. (9). Note that these equations are equivalent to the 3-DOF system described above, under the given assumptions. So any results we obtain using these are directly applicable to the latter.

$$R_{turn} = \frac{mV_{mission}^2}{F_{y_{max}}} \qquad (9)$$

IV.D. Minimum Length Trajectory Control

For the simplified dynamic model, we do not need to use LQR for control - we can geometrically construct the minimum length trajectories and find the corresponding control inputs. Dubins has proved that the minimum length trajectories between any two points consist of straight line segments and arcs of minimum turn radius.[53] Erzberger and Lee[54] have described the corresponding trajectories. Figure 10 shows how we can construct four possible paths between points A and B. The shortest path is one of the four paths, and can be found analytically.

Recently, Modgalya and Bhat[55] came up with a feedback control algorithm for traversing these paths. But this algorithm does not cater to all the cases we are interested in, so we use our own *minimum distance controller* to traverse the shortest path trajectories.

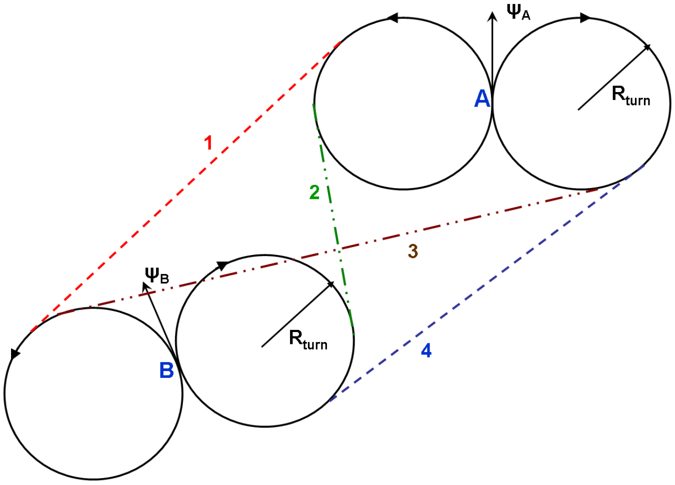


Figure 10. Sample case showing how to find the minimum length trajectory starting from point A (heading, ψ_A), and reaching B (heading, ψ_B). There are four candidate paths, numbered 1 to 4, and the shortest path can be found by calculating the lengths for all of them.

IV.E. Modifying the Control Policy

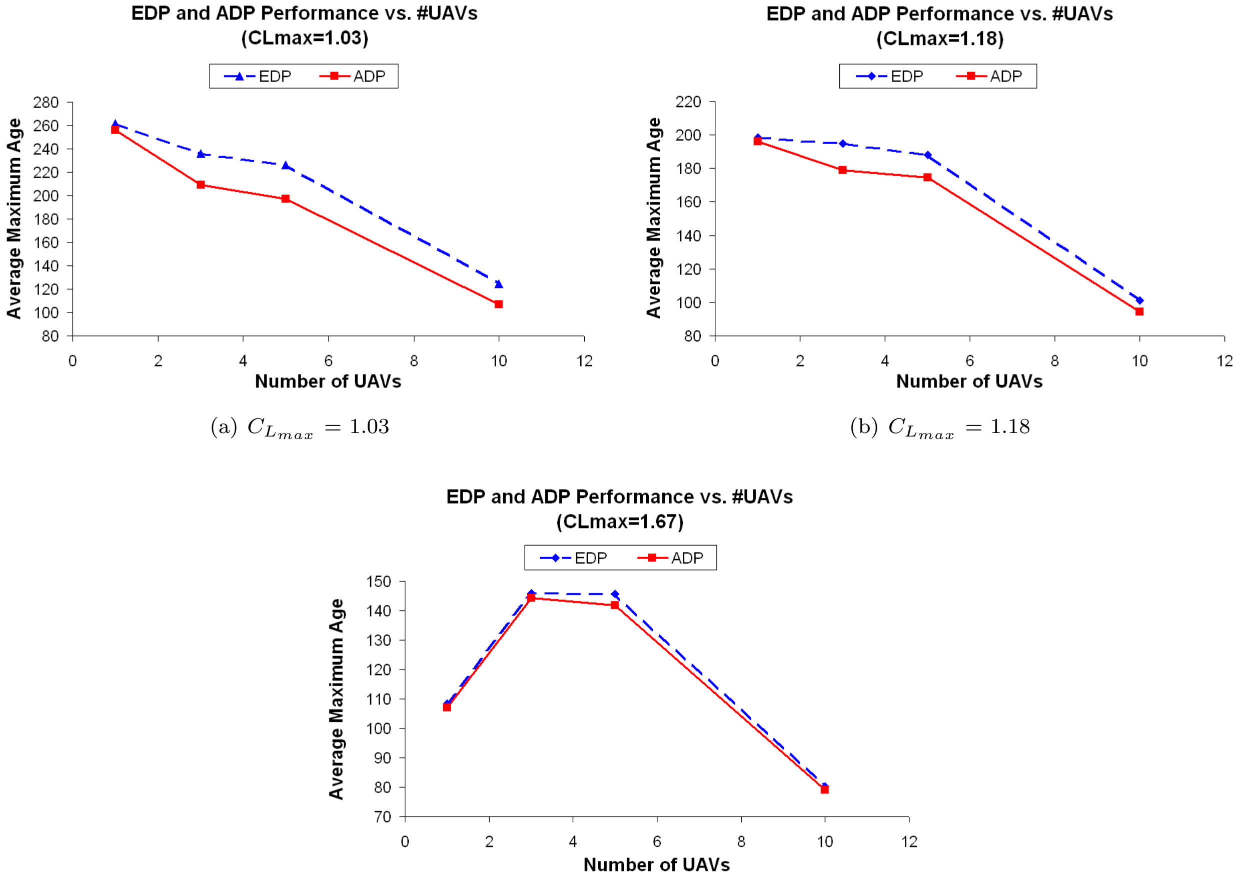
Recall the value functions used in control policies for single and multiple UAVs were given by Eq. (3) and Eq. (5) respectively. We have used Euclidean distances in the policies so far, but under dynamic constraints, it makes more sense to use the actual distances to cells in calculating the values. Since we are using a minimum distance controller, we can use the shortest path distances in our policies. We call the former approach as *Euclidean Distance Policy* (EDP), and our modified control policy as *Actual Distance Policy* (ADP). We can now compare the two to see if we get any performance improvements.

For this purpose, we simulate UAVs with $m = 1$ kg, $S_{ref} = 0.7\ m^2$, $V_{mission} = 5$ m/s, and give control inputs of $F_{y_{max}} = 5, 8, 15$ N. These correspond to $C_{L_{max}} = 1.03, 1.18, 1.67$, and $R_{turn} = 5, 3.125, 1.67$ m respectively. We study scenarios with 1, 3, 5 and 10 UAVs on a target space, 50 m x 50 m in dimension for single UAV and 75 m x 75 m in dimension for multiple UAVs, each with $r_{sensor} = 2.5$ m. We use MRP for coordination between multiple UAVs. Figure 11 compares the maximum ages observed (averaged over 50 trials) as functions of N_{UAV}, for different $C_{L_{max}}$ values. Table 1 gives a summary of the comparison results.

Table 1. Summary of results comparing EDP and ADP for different dynamic constraints

Case	$C_{L_{max}} = 1.03$		$C_{L_{max}} = 1.18$		$C_{L_{max}} = 1.67$	
N_{UAV}	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

We observe that the performance of ADP with respect to EDP improves as the dynamics become more constrained, and as the number of UAVs increase. However, the improvement with N_{UAV} tends to saturate. This is because we observe another interesting *emergent* behavior with MRP. Under dynamic constraints, the UAVs tend to leave unexplored *gaps* in the target space. However, when other UAVs are present, they are able to fill these gaps and hence reduce the degradation in performance.



(a) $C_{L_{max}} = 1.03$

(b) $C_{L_{max}} = 1.18$

(c) $C_{L_{max}} = 1.67$

Figure 11. Comparison of the EDP and ADP (plot of average maximum age as a function of number of UAVs) for several values of $C_{L_{max}}$.

V. System of Systems Design

The next step after deciding on the control policy structure, is to look at the design of the UAVs. This is a SoS problem where the goal is to achieve optimal mission performance for a group of vehicles at a reasonable design cost.[m] This problem encompasses two disciplines: operations and aircraft design. Having devised the control policy structure, the operations problem is to find the values of aircraft performance variables ($V_{mission}$, R_{turn} and range[n]) and control policy weights for minimum mission cost. The mission cost is basically the maximum age observed, for given target space and UAV design, averaged over 10 trials.[o] The design problem is to minimize the cost (the gross takeoff weight in this study) while achieving the desired aircraft performance. In this study we look at the design of a single UAV type to illustrate our SoS design approach, but the design of multiple UAVs can be easily incorporated in the existing framework.

Underwood and Baldesarra[37] have looked at a similar problem, but emphasize on the need for human-inloop interaction. Previous work on SoS design described in [38, 39], involves the design of morphing aircraft for a U.S. Coast Guard problem. In this work, Frommer and Crossley use a hierarchical architecture for integrating operational analysis and sizing modules. But when applied to our problem, the hierarchical approach has associated limitations. In this study we use a design architecture based on Collaborative Optimization (CO) which gives relative independence to each discipline, while ensuring that they converge to a single result.

V.A. SoS Design Architecture

In this section we introduce the architecture we use for the SoS design problem. There are several reasons for avoiding a monolithic optimization technique and separating the mission and design problems. Often the mission and design optimizations use different optimizers and have very different run times. So in that case, it is counter-productive to include the optimization variables from the less expensive problem in the other. In our problem, if we deal with multiple aircraft designs, then it is much more expensive to solve an optimization problem with all the design variables together. Also, decomposed design makes much more sense from a practical point of view, where the organizational structure necessitates dealing with these problems in parallel and with minimal interaction.

CO is a popular architecture for design decomposition that has been applied to many aircraft design problems.[40] In this work we use CO to solve the SoS design problem. We first present the CO-based decomposition architecture specific to our problem, for a single UAV design, in figure 12, and then go on to explain the components in detail. Note that we refer to system level variables without subscripts. The mission subspace local variables do not carry subscripts but the shared variables are subscripted with $S0$ (for instance $V_{mission_{S0}}$)). Similarly, the design subspace shared variables are subscripted with $S1$, while the local variables are not.

V.B. System Level Optimization

In our architecture, the mission performance optimizer and the aircraft design optimizer are the two subspaces and the system level optimizer coordinates them. The objective of the SoS design is a composite function of the subspace objectives: $A_{max} + \alpha W_{gross}$. The mission cost, A_{max} is the maximum age observed over all cells over a long period of time (averaged over 10 trials), and the design cost, W_{gross}, is the maximum takeoff gross weight. α is a weighting parameter deciding the relative importance of the subspace costs. The shared variables relevant to both subspaces are $V_{mission}$, and R_{turn}, which become optimization variables at the system level. Normally, we would include the range of the aircraft as a shared variable as well, but in this study we only require the range to be more than a pre-fixed value. The system level optimizer also includes A_{max} and W_{gross} as variables. These are sent to the subspaces as targets, along with other shared variables. So the subspace optimizers just try to meet these target values, and do not try to minimize the

[m] It is upto the designer to decide what the mission performance and design cost are, and how important they are relative to each other.

[n] We have simply constrained the range to be above a pre-decided target in this paper.

[o] We believe the average mission cost is a more accurate measure of our mission objective, especially if we expect problem dynamics and failures.

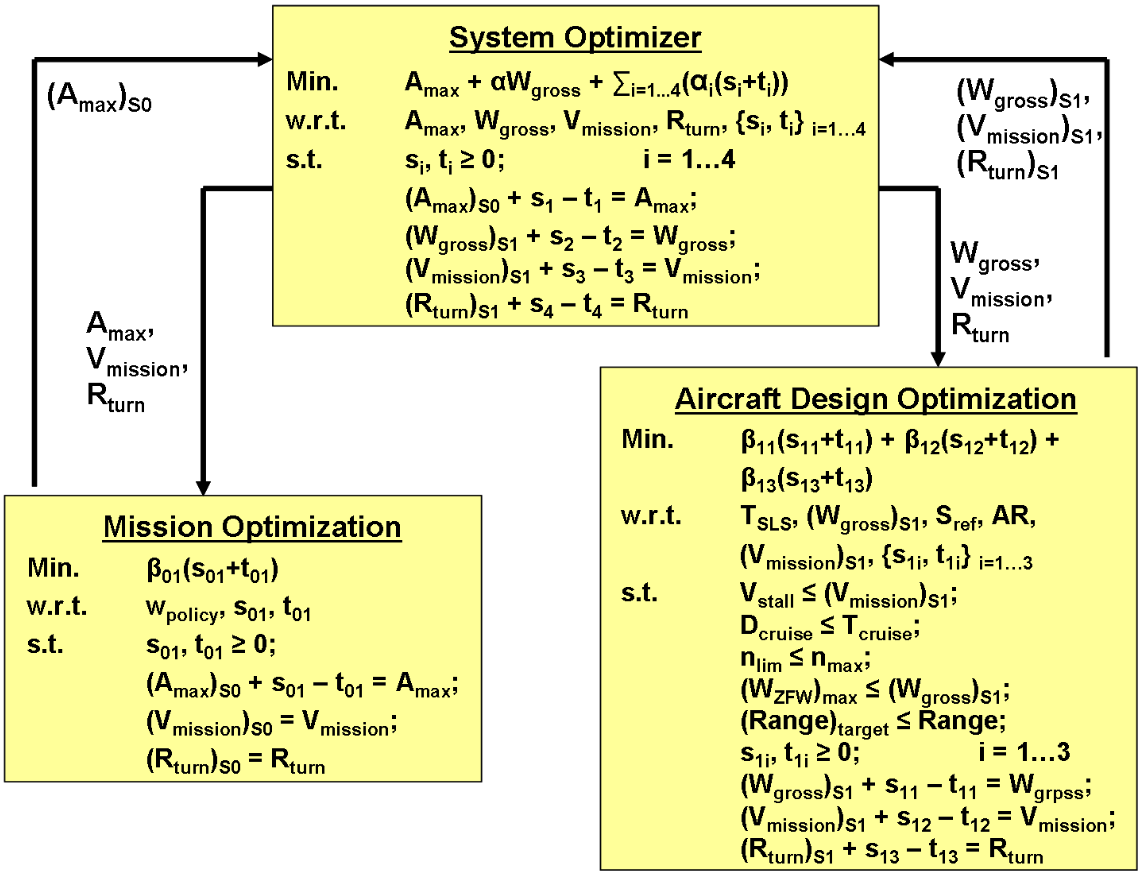

System Optimizer
Min. $A_{max} + \alpha W_{gross} + \sum_{i=1...4}(\alpha_i(s_i+t_i))$
w.r.t. $A_{max}, W_{gross}, V_{mission}, R_{turn}, \{s_i, t_i\}_{i=1...4}$
s.t. $s_i, t_i \geq 0;$ $i = 1...4$
$(A_{max})_{S0} + s_1 - t_1 = A_{max};$
$(W_{gross})_{S1} + s_2 - t_2 = W_{gross};$
$(V_{mission})_{S1} + s_3 - t_3 = V_{mission};$
$(R_{turn})_{S1} + s_4 - t_4 = R_{turn}$
$(A_{max})_{S0}$
$(W_{gross})_{S1}$, $(V_{mission})_{S1}$, $(R_{turn})_{S1}$
A_{max}, $V_{mission}$, R_{turn}
W_{gross}, $V_{mission}$, R_{turn}
Mission Optimization
Min. $\beta_{01}(s_{01}+t_{01})$
w.r.t. $w_{policy}, s_{01}, t_{01}$
s.t. $s_{01}, t_{01} \geq 0;$
$(A_{max})_{S0} + s_{01} - t_{01} = A_{max};$
$(V_{mission})_{S0} = V_{mission};$
$(R_{turn})_{S0} = R_{turn}$
Aircraft Design Optimization
Min. $\beta_{11}(s_{11}+t_{11}) + \beta_{12}(s_{12}+t_{12}) + \beta_{13}(s_{13}+t_{13})$
w.r.t. $T_{SLS}, (W_{gross})_{S1}, S_{ref}, AR, (V_{mission})_{S1}, \{s_{1i}, t_{1i}\}_{i=1...3}$
s.t. $V_{stall} \leq (V_{mission})_{S1};$
$D_{cruise} \leq T_{cruise};$
$n_{lim} \leq n_{max};$
$(W_{ZFW})_{max} \leq (W_{gross})_{S1};$
$(Range)_{target} \leq Range;$
$s_{1i}, t_{1i} \geq 0;$ $i = 1...3$
$(W_{gross})_{S1} + s_{11} - t_{11} = W_{grpss};$
$(V_{mission})_{S1} + s_{12} - t_{12} = V_{mission};$
$(R_{turn})_{S1} + s_{13} - t_{13} = R_{turn}$

Figure 12. The design decomposition architecture used for SoS design using CO.

mission and design costs individually. The system level optimization is carried out using a gradient based optimizer, SNOPT,[56] which uses sequential quadratic programming for optimization.

Also note that we do not use quadratic penalties, as in classical CO, to ensure compatibility between the subspace and system levels. A generic problem with quadratic penalty terms is that they do not ensure compatibility unless the associated weights are infinite. Starting with a very large value of the weight reduces exploration and often makes the jacobean matrrices ill-conditioned. One way to deal with this problem is to use adaptive weighting, where we gradually increase the associated weight. We instead, use linear penalty functions with associated elastic/slack variables,[41] s_i, t_i. These have the advantage of ensuring compatibility for a finite value of associated weight. The addition of elastic variables shifts the discontinuity associated with an L_1 norm to the switching of active constraint sets. The switching of active constraint sets at the subspace level can cause non-smooth gradients at the system level, but it still does not affect the local convergence of the system optimizer. Note that this formulation increases the number of optimization variables and the constraints,[p] but in a gradient based optimization technique, the addition of a few variables and constraints does not incur any significant cost.

V.C. Mission Performance

The goal of the mission performance optimizer in the CO architecture, is to meet the values of the target variables specified by the system level optimizer (i.e. A_{max}, $V_{mission}$, and R_{turn}) as a function of its local

[p] The architecture shows the elastic variables included in the optimization variable set, and the associated constraints. The elastic variables are also added to the system objective, weighted by α_i.

variables (policy weight, w_0[q]), and shared optimization variables ($V_{mission_{S0}}$ and $R_{turn_{S0}}$). However, we know that for our problem, the target values for the velocity and radius of turn can always be met by the mission optimizer. So we need not include them as additional variables. We just specify their values to the mission analysis, and find the optimum value of w_0 (i.e. the value that gets us closest to target A_{max}). s_{00} and t_{00} are elastic variables associated with the compatibility term. There are no constraints in the mission optimization problem, except non-negativity of elastic variables, and one associated with the linear penalty. β_{00} is a weighting parameter for the compatibility term, though it becomes redundant with only one compatibility term in the objective.

V.C.1. Mission Optimization

The mission cost is not a smooth function of the optimization variables. So we can not use a gradient based method directly for mission performance optimization. We could use a non-gradient based method, but that usually turns out to be very expensive if the optimization is done at the subspace level (since the system makes multiple calls to the subspaces). So we use a response surface to represent $A_{max_{S0}}$ as a function of $V_{mission_{S0}}$, $R_{turn_{S0}}$ and w_0. We use Locally Weighted Linear Regression (LWLR)[57] for this purpose, and use SNOPT for mission performance optimization. The optimizer calls the response surface to return values of A_{maxS0}, which it tries to match with the target values.

To speed up the response surface generation as well, we make the mission simulation faster. This is achieved by creating a response surface over the shortest path distances to different points. This response returns the shortest distance (under dynamic constraints) from the UAV to any point in space as a function of the euclidean distance of the point from the UAV and the heading change required to point directly towards the point. This parameterization has the advantage of closer spacing between the data points close to the UAV in actual target space. This in turn means we have greater accuracy of fit where it is required. We use linear interpolation for generating the response surface.[r]

V.D. Aircraft Design Optimization

Before discussing the aircraft design optimization, we briefly look at the aircraft performance analysis.

V.D.1. Aircraft Performance Analysis

For the purposes of design, we assume a simple mission description, as shown in figure 13. We have developed an analysis for aircraft design, based on [58], using a set of five design variables sea level static thrust, T_{SLS}, takeoff gross weight, $W_{gross_{S0}}$, reference wing area, S_{ref}, aspect ratio, AR, and mission velocity, $V_{mission_{S0}}$. We constrain the range to be above 2000 km, which gives a flight time of a few hours to the UAV for the velocities that we have considered.

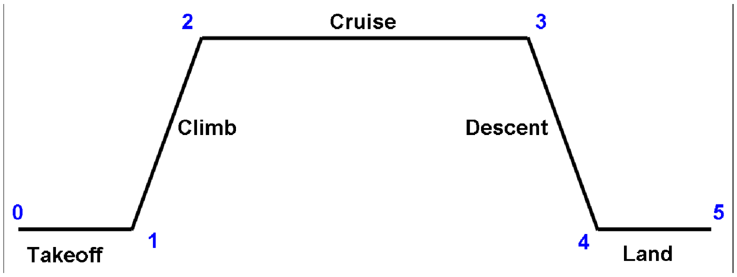


Figure 13. Simple mission description assumed for the purpose of aircraft design.

V.D.2. Optimization Subspace

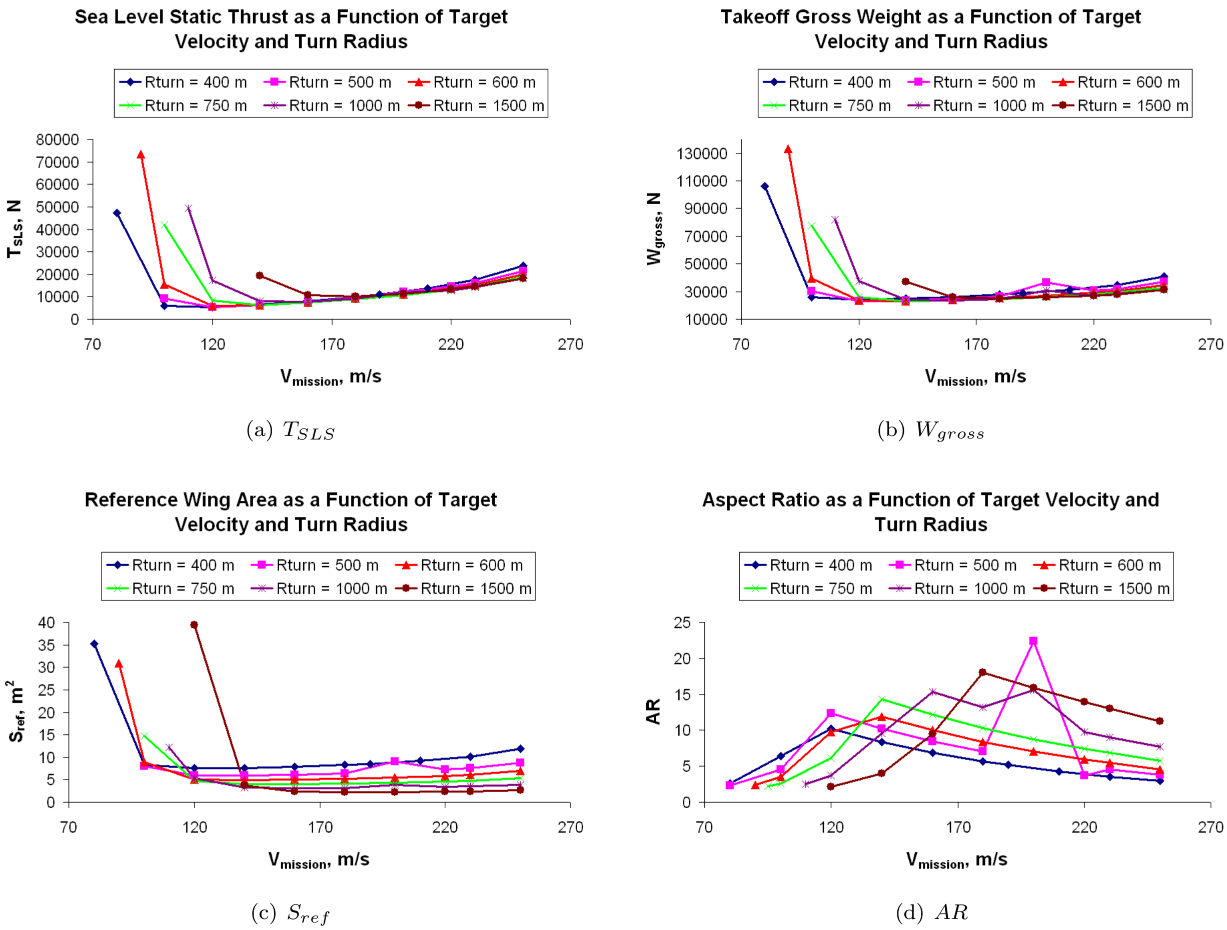
The aircraft design optimizer receives target values for W_{gross}, $V_{mission}$, and R_{turn} from the system level and tries to match these values while satisfying the design constraints. As we can see in figure 12, there are five constraints ensuring the aircraft flies above the stall speed (V_{stall}), there is enough thrust (T_{cruise}) to overcome drag (D_{cruise}) in cruise, $C_{L_{max}}$ is high enough to sustain gust loading, the maximum zero fuel weight ($W_{ZFW_{max}}$) is less than gross takeoff weight, and the range is more than its corresponding constraint value. Linear penalty functions are used to ensure compatibility, and s_{1i}, t_{1i} are the corresponding elastic variables. We again use SNOPT for the optimization process.

[q] Note we introduced this parameter in Eq. (3) for a single UAV case.

[r] Note that we do not require smoothness of the response surface for our purposes here.

V.E. Results

In this section we present some results for studies related to aircraft design. We first studied the variation in the aircraft design parameters for different target values of $V_{mission}$ and R_{turn}. For this purpose, we found the optimum aircraft designs meeting different sets of targets for velocity and radius of turn. The optimization is analogous to the aircraft design subspace shown in figure 12, except that $W_{gross_{S0}}$ is minimized instead of meeting a target value. Figure 14 shows the dependence of the aircraft design variables on $V_{mission}$, for several values of R_{turn}.



(a) T_{SLS}

(b) W_{gross}

(c) S_{ref}

(d) AR

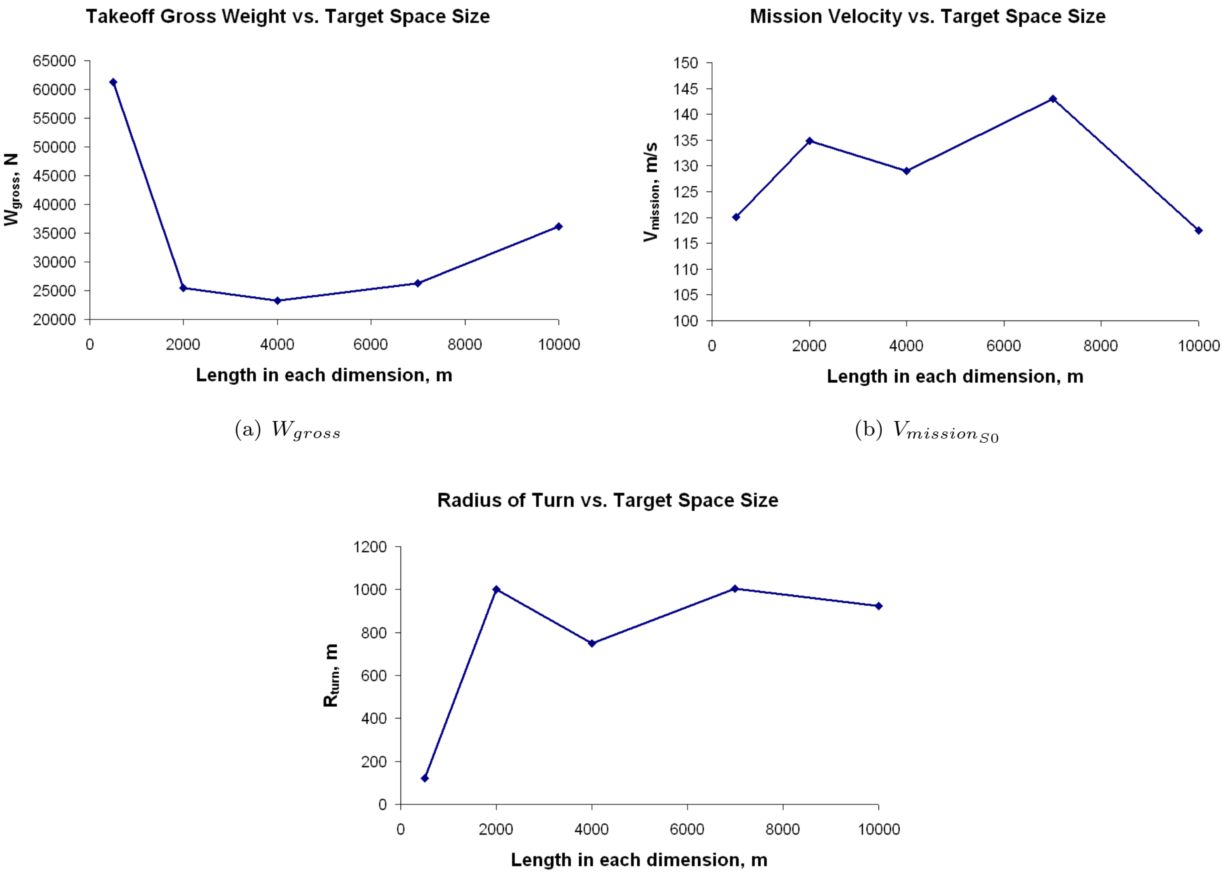
Figure 14. The design variables plotted as a function of target $V_{mission}$ for different target values of R_{turn}. The design is optimized for minimum W_{gross} while meeting the target values.

We observe that T_{SLS}, W_{gross}, and S_{ref} record their minimum values for some intermediate value of target mission velocity and this target value increases with increasing target R_{turn}. AR shows a similar trend, but with a maxima.[s] These plots give an idea of values of shared variables that the aircraft design optimization itself would try to achieve. We also get a sense of the penalty for moving away from those values. Moreover, they show the range of interest for $V_{mission}$ and R_{turn}, since the design cost for values outside this range is very high.

Finally, we study the proposed SoS architecture by optimizing designs for different mission scenarios involving a single UAV. We look at results involving several target space sizes (with proportional change in sensor footprint). The overall objective function is a weighted combination of the mission cost and design cost: $A_{max} + \alpha W_{gross}$, with $\alpha = 0.001$.

[s] There seems to be a spurious peak in the curve with $R_{turn} = 500$ m, which we believe to be an outlier point.

In this problem, we look at square shaped target spaces, while varying the length of each side from 500 m to 10000 m. The sensor footprint (that determines the size of each grid cell as well) is assumed to change in proportion, such that the number of cells remain the same - we fix this number to 20*20, so for a target space 2000 m in length, $r_{sensor} = 50$ m. The results in figure 15 show the dependence of W_{gross}, $V_{mission}$, and R_{turn} on the mission target space size. We observe that the radius of turn increases with increasing space dimension, as expected. The mission velocity, however, does not show a reasonable trend. The reason is that the takeoff gross weight is a strong function of the mission velocity, while the mission cost is not, if the radius of turn is small enough. So the design cost tends to dominate the curve for the velocity. We can observe that high values of W_{gross}, tends to push the velocity to values which result in minimum design cost. We can perform similar design analyses for other mission specifications (say changing the sensor footprint of UAVs), hence performing design for desirable mission performance.



(a) W_{gross}

(b) $V_{mission_{S0}}$

(c) $R_{turn_{S0}}$

Figure 15. The takeoff gross weight, mission velocity and the radius of turn for the optimized design, plotted against the length of each side of the target space.

VI. Conclusion

In this study we have defined an approach to multiple UAV persistent surveillance based on an optimum policy for a particular single-UAV case. Comparison of the policy with selected benchmark methods and a bound on the optimum shows encouraging results. The extension to the multiple UAV case has been done using an approach that can respond to environment dynamics and UAV failures. The approach is heuristic in nature, but highly scalable, robust, and simple to implement.

A 3-DOF simulation is then used to evaluate the effect of UAV dynamics. The effect of aircraft dynamics on performance is evaluated and the control policy is modified to improve performance under aircraft dynamic constraints. We finally look at a SoS design problem where we study the design of UAVs for optimum mission performance. We propose an architecture based on CO, using linear penalty functions for compatibility, for a single design case. We then study the designs as a function of mission specifications.

Future research will include evaluation of environment dynamics and UAV failures, testing the control policies in this more challenging scenario. We will also use the SoS design architecture to study the case of multiple UAVs (both homogeneous and heterogeneous). The inclusion of the range/endurance of the aircraft as a shared variable in the architecture will further enhance the richness of the problem.

Acknowledgments

We would like to thank Prof. Stephen Rock, Prof. Walter Murray, and Brian D. Roth 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] Cao, Y. U., Fukunaga, A. S., and Kahng, A. B., *Cooperative Mobile Robotics: Antecedents and Directions*, Autonomous Robots, Vol. 4, Issue 1, 1997, pp. 7-27.

[2] Parker, L. E., *Current State of the Art in Distributed Mobile Robotics*, Distributed Autonomous Robotic Systems 4, Vol. 4, Oct. 2000, pp. 3-12.

[3] Hougen, D. F., Erickson, M. D., Rybski, P. E., Stoeter, S. A., Gini, M., and Papanikolopoulos, N., *Autonomous Mobile Robots and Distributed Exploratory Missions*, Distributed Autonomous Robotic Systems 4, Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems, 2000, pp. 221-230.

[4] Choset, H., *Coverage for Robotics A Survey of Recent Results*, Annals of Mathematics and Artificial Intelligence, Vol. 31, 2001, pp. 113-126.

[5] Burgard, W., Moors, M., and Schneider, F., *Collaborative Exploration of Unknown Environments with Teams of Mobile Robots*, Lecture Notes in Computer Science, Advances in Plan-based Control of Robotic Agents, Springer Verlag, 2002, pp. 187-215.

[6] Thrun, S. B., *Exploration and Model Building in Mobile Robot Domains*, IEEE International Conference on Neural Networks, Vol. 1, 1993, pp. 175-180.

[7] Flint, M., Fernandez-Gaucherand, E., and Polycarpou, M., *Stochastic Models of a Cooperative Autonomous UAV Search Problem*, Military Operations Research, Vol. 8, No. 4, 2003, pp. 13-33.

[8] Bellingham, J. S., Tillerson, M., Alighanbari, M., and How, J. P., *Cooperative Path Planning for Multiple UAVs in Dynamic and Uncertain Environments*, Proceedings of the 41st IEEE Conference on Decision and Control, Vol. 3, Dec. 2002, pp. 2816-2822.

[9] Sujit, P. B., and Ghose, D., *Search Using Multiple UAVs with Flight Time Constraints*, IEEE Transactions on Aerospace and Electronic Systems, Vol. 40, No.2, April 2004, pp. 491510.

[10] Chang, S. J., and Dan, B. J., *Free Moving Pattern's Online Spanning Tree Coverage Algorithm*, SICE-ICASE International Joint Conference, Oct. 2006, pp. 2935-2938.

[11] Hazon, N., Mieli, F., and Kaminka, G. A., *Towards Robust On-line Multi-robot Coverage*, Proceedings of IEEE International Conference on Robotics and Automation, May 2006, pp. 1710-1715.

[12] Koenig, S., Szymanski, B., and Liu, Y., *Efficient and Inefficient Ant Coverage Methods*, Annals of Mathematics and Artificial Intelligence, Vol. 31, Issue 1-4, 2001, pp. 41-76.

[13] Batalin, M. A., and Sukhatme, G. S., *The Analysis of an Efficient Algorithm for Robot Coverage and Exploration based on Sensor Network Deployment*, Proceedings of the IEEE International Conference on Robotics and Automation, April 2005, pp. 3478-3485.

[14] Burgard, W., Fox, D. Moors, M., Simmons, R., and Thrun, S., *Collaborative Multi-Robot Exploration*, Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, April 2000, pp. 476481.

[15] Choset, H., and Pignon, P., *Coverage Path Planning: The Boustrophedon Decomposition*, International Conference on Field and Service Robotics (FSR'97), Australia, 1997.

[16] Acar, E. U., and Choset, H., *Critical Point Sensing in Unknown Environments*, Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 4, April 2000, pp. 3803-3810.

[17] Kurabayashi, D., Ota, J., Arai, T., and Yoshida, E., *An Algorithm of Dividing a Work Area to Multiple Mobile Robots*, Proceedings of the International Conference on Intelligent Robots and Systems, Vol. 2., Aug. 1995, pp. 286-291.

[18] Acar, E. U., and Choset, H., *Sensor-based Coverage of Unknown Environments: Incremental Construction of Morse Decompositions*, International Journal of Robotics Research, Vol. 21, No. 4, 2002, pp. 345-366.

[19] Min, T. W., and Yin, H. K., *A Decentralized Approach for Cooperative Sweeping by Multiple Mobile Robots*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, Oct. 1998, pp. 380-385.

[20] Berhault, M., Huang, H., Keskinocak, P., Koenig, S., Elmaghraby, W., Griffin, P., and Kleywegt, A., *Robot Exploration with Combinatorial Auctions*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 2, Oct. 2003, pp. 1957-1962.

[21] Lawrence, D. A., Donahue, R. E., Mohseni, K., and Han, R., *Information Energy for Sensor-Reactive UAV Flock Control*, 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit, AIAA20046530, Sep. 2004.

[22] Erignac, C. A., *An Exhaustive Swarming Search Strategy based on Distributed Pheromone Maps*, Infotech@Aerospace Conference and Exhibit, AIAA-2007-2822, May 2007.

[23] Flint, M., Polycarpou, M., and Fernandez-Gaucherand, E., *Cooperative Path Planning for Autonomous Vehicles Using Dynamic Programming*, Proceedings of the 15th Triennial IFAC World Congress, Vol. P, July 2002, pp. 481487.

[24] Richards, A., Bellingham, J. S., Tillerson, M., and How, J., *Coordination and Control of Multiple UAVs*, AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA-2002-4588, Aug. 2002.

[25] Parunak, H. V. D., Making Swarming Happen, Conference on Swarming and Network Enabled Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance, Virginia, 2003.

[26] Tumer, K., and Agogino, A., *Coordinating Multi-Rover Systems: Evaluation Functions for Dynamic and Noisy Environments*, Proceedings of Genetic and Evolutionary Computation Conference, June 2005, pp. 591-598.

[27] Gaudiano, P., Shargel B., and Bonabeau, E., *Control of UAV Swarms: What the Bugs Can Teach Us*, 2nd AIAA Unmanned Unlimited Conference and Workshop and Exhibit, AIAA-2003-6624, Sep. 2003.

[28] Hokayem, P. F., Stipanovic, D., and Spong, M. W., *On Persistent Coverage Control*, Proceedings of the 46th IEEE Conference on Decision and Control, Dec. 2007, pp. 6130-6135.

[29] Bellingham, J., Richards, A., and How, J. P., *Receding Horizon Control of Autonomous Aerial Vehicles*, Proceedings of the American Control Conference, Vol. 5, 2002, pp. 3741-3746.

[30] Schumaker, C., Chandler, P. R., and Rasmussen, S. R., *Task Allocation for Wide Area Search Munitions via Network Flow Optimization*, AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA-2001-4147, Aug. 2001.

[31] Darrah, M., Niland W., and Stolarik, B., *Increasing UAV Task Assignment Performance through Parallelized Genetic Algorithms*, AIAA Infotech@Aerospace Conference and Exhibit, AIAA-2007-2815, May 2007.

[32] Kovacina, M.A., Palmer, D., Yang, G., and Vaidyanathan, R., *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, 2002, pp. 27822788.

[33] Keating, C., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A., Safford, R., Peterson, W. and Rabadi, G., *System of Systems Engineering*, Engineering Management Journal, Vol. 15, No. 3, Sep. 2003, pp. 36-45.

[34] Soban, D. S., and Mavris, D., *The Need for a Military System Effectiveness Framework: The System of Systems Approach*, 1st AIAA Aircraft Technology, Integration, and Operations Forum, AIAA-2001-5226, Oct. 2001.

[35] Keating, C., Sousa-Poza, A., and Mun, N., *Toward a Methodology for System of Systems Engineering*, Proceedings of the American Society of Engineering Management, 2003, pp. 1-8.

[36] Delaurentis, D., Kang, T., Lim, C., Mavris, D., and Schrage, D.,*System-of-Systems Modeling for Personal Air Vehicles*, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA-2002-5620, Sep. 2002.

[37] Underwood, J. E., and Baldesarra, M., *Operations Simulation Framework to Evaluate Vehicle Designs for Planetary Surface Exploration*, AIAA Space Conference and Exposition, AIAA-2007-6254, Sep. 2007.

[38] Frommer, J. B., and Crossley, W. A., *Evaluating Morphing Aircraft in a Fleet Context Using Non-Deterministic Metrics*, 5th AIAA Aviation Technology, Integration, and Operations Conference, AIAA-2005-7400, Sep. 2005.

[39] Frommer, J. B, and Crossley, W. A., *Building Surrogate Models for Capability-Based Evaluation: Comparing Morphing and Fixed Geometry Aircraft in a Fleet Context*, 6th AIAA Aviation Technology, Integration, and Operations Conference, AIAA-2006-7700, Sep. 2006.

[40] Braun, R. D., *Collaborative Optimization: An Architecture for Large-Scale Distributed Design*, PhD Thesis, Stanford University, April 1996.

[41] Miguel, A. D., *Two Decomposition Algorithms for Nonconvex Optimization Problems with Global Variables*, PhD Thesis, Stanford University, June 2001.

[42] Kokkolaras, M., Fellini, R., Kim, H. M., and Papalambros, P. Y., *Analytical Target Cascading in Product Family Design*, Product Platform and Product Family Design, Springer US, 2006, pp. 225-240.

[43] Rajnarayan, D., Kroo, I., and Wolpert, D., *Probability Collectives for Optimization of Computer Simulations*, 48th AIAA/ASME/ASCE/AHS/ASC/ Structures, Structural Dynamics, and Materials Conference, AIAA-2007-1975, April 2007.

[44] Flint, M., Fernandez-Gaucherand, E., and Polycarpou, M., *Cooperative Control for UAVs Searching Risky Environments for Targets*, Proceedings of the 42nd IEEE Conference on Decision and Control, Vol. 4, Dec. 2003, pp. 3567-3572.

[45] Cormen, T. H., Leiserson, C. E., and Rivest, R. L., *Introduction to Algorithms*, Cambridge: MIT Press, 1990.

[46] Nigam, N., and Kroo, I., *Persistent Surveillance Using Multiple Unmanned Air Vehicles*, Proceedings of the IEEE Aerospace Conference, Mar. 2008, pp. 1-14.

[47] Liu, Y., Cruz, J. B., and Sparks, A. G., *Coordinating Networked Uninhabited Air Vehicles for Persistent Area Denial*, 43rd IEEE Conference on Decision and Control, Vol. 3, Dec. 2004, pp. 33513356.

[48] Ousingsawat, J., and Campbell, M. E., *Optimal Cooperative Reconnaissance using Multiple Vehicles*, Journal of Guidance Control and Dynamics, Vol. 30, No. 1, 2007, pp. 122132.

[49] Sachs, G., *Minimum Shear Wind Strength Required for Dynamic Soaring of Albatrosses*, IBIS-London-British Ornithologists Union, Vol. 147, No. 1, Jan. 2005, pp. 1-10.

[50] Stevens, B. L., and Lewis, F. L., *Aircraft Control and Simulation*, New York: John Wiley and Sons Inc., 1992.

[51] Divelbiss, A. W., and Wen, J. T., *Trajectory Tracking Control of a Car-Trailer System*, IEEE Transactions on Control Systems Technology, Vol. 5, No. 3, May 1997.

[52] Bertsekas, D. P., *Dynamic Programming and Optimal Control*, Athena Scientific, Vol. 1, 2nd ed., 2000.

[53] Dubins, L. E., *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, July 1957, pp. 497-516.

[54] Erzberger, H., and Lee, H. Q., *Optimum Horizontal Guidance Techniques for Aircraft*, Journal of Aircraft, Vol. 8, No. 2, Feb. 1971, pp. 95-101.

[55] Modgalya, M., and Bhat, S. P., *Time-Optimal Feedback Guidance in Two Dimensions under Turn-Rate and Terminal Heading Constraints*, National Conference on Control and Dynamical Systems, India, Jan 2005.

[56] Gill, P., Murray, W., and Saunders, M., *SNOPT : An SQP Algorithm for Large-scale Constrained Optimization*, Numerical Analysis Report 97-2, Department of Mathematics, University of California, San Diego, 1997.

[57] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, Springer-Verlag, New York, 2001, pp. 168-175.

[58] Kroo, I., *Aircraft Design: Synthesis and Analysis*, Desktop Aeronautics Inc., 2006.