Proceedings of IMECE 2007
ASME 2007 International Mechanical Engineering Congress and Exhibition
November 11-15, 2007, Seattle, USA

IMECE2007-43728

PATH PLANNING FOR COOPERATIVE SENSING USING UNMANNED VEHICLES

John Tisdale *
Vehicle Dynamics Lab
Department of Mechanical Engineering
University of California
Berkeley, California 94720

Hugh Durrant-Whyte
ARC Centre of Excellence
for Autonomous Systems,
The University of Sydney
Sydney, NSW
Australia

J. Karl Hedrick
Vehicle Dynamics Lab
Department of Mechanical Engineering
University of California
Berkeley, California 94720

ABSTRACT

This work explores online path-planning for unmanned vehicles performing cooperative sensing. Much existing work employs receding-horizon optimization, where an objective function is repeatedly optimized over some short lookahead length. The use of receding horizon optimization often results in ad-hoc methods for dealing with the problem of myopic lookahead, where no value is visible in an agent's planning horizon. This work examines the use of an algorithm for receding-horizon optimization that explicitly accounts for myopia by allowing for a variable lookahead length. Cooperation is maintained by ensuring that all agents plan to the same horizon, potentially with different strategies. This algorithm is used to develop trajectories for a team of unmanned vehicles searching for a target using a probabilistic framework. Simulation results are presented and discussed.

INTRODUCTION

One promising area of unmanned systems research is in the area of distributed sensing. Unmanned aerial vehicles (UAVs) are an attractive solution for sensing applications, due to their relative cost, compared to manned vehicles, and due to the fact that they can be inserted into dangerous or hostile environments. The development of autonomous search strategies, that take into account target and environment models, are vital to the deployment of UAVs in this role.

Online control strategies for teams of mobile sensor platforms often fall into one of two categories: discrete task allocation and receding-horizon optimization. Task allocation schemes assume that the mission can be split into a finite set of tasks, to be divided up among a set of agents, as in [1]. Tasks are allocated cooperatively, but performed individually; vehicles plan trajectories ignorant of other agents. Cooperation to perform a specific task often requires ad-hoc methods, or requires a single task to somehow be split, as in [2]. Task allocation is effective for developing cooperation among a team, assuming other algorithms are available for local vehicle control. The algorithms considered in this paper are developed to account for problems that are inherently continuous, and do not lend themselves to discretization into tasks.

In a receding-horizon trajectory generation scheme, an optimization problem is solved at each time step, to maximize an objective function for some portion of the future trajectory. The first control in the sequence is then applied to the vehicle, and the optimization is repeated. In the case of a continuous system, the system is discretized by holding the control variable constant over short time intervals. Often, a gradient based optimization routine is used to solve this optimization. Alternately, dynamic programming approaches have been used, as in [3].

Optimization of the trajectory for the entire mission duration is computational infeasible. To make this problem tractable, the optimization procedure is broken up into a series of optimizations, over a short horizon, as in [3–5]. Intuitively, sequential trajectory optimization over a short time horizon often leads to

*Address all correspondence to this author.

good performance over the mission horizon, however, no formal guarantees can be made at this point.

In the context of motion planning for sensing, receding horizon algorithms have problems when the reward function is of little value within the optimization horizon. In this sense, the optimization routine is too myopic with respect to the objective function. This problem generally occurs when the agent is far from areas of value, with respect to the planning horizon. Previous work has examined various ad-hoc methods of perturbing the objective function to deal with this myopia. The work considered here takes an approach that allows the optimization horizon to vary online. Other frameworks for trajectory development, such as the POMDP method and the cell-based scheme detailed in [6], are thought to be too computationally expensive for online motion planning.

The paper is organized as follows; first, some basic results for cooperative sensing problems are presented. A solution strategy for the general cooperative optimization problem is then introduced. An example problem is then presented, in which a team of UAVs cooperatively search for a lost target. The solution strategy for this problem, which relies on probabilistic target modeling and Bayesian filtering, was first examined in [4] and will be used to illustrate various concepts in this paper. The cooperative sensing algorithm is then applied to the example search problem. Example trajectories for the search problem are then developed. Lastly, conclusions, and potential future work are discussed.

PROBLEM STATEMENT

This work is concerned with developing trajectories for teams of unmanned vehicles performing sensing missions. The cooperative planning problem may be stated as,

$$\mathbf{U}^* = \arg\max_{U} J(\mathbf{U}) \tag{1}$$

where $\mathbf{U} = \{U^1_{1:k}, \ldots, U^m_{1:k}\}$ are the control inputs of the m team-members over n time steps, subject to some constraints, and $\mathbf{J}(\mathbf{U})$ is a reward function that captures the information collected by the team. For most cooperative sensing problems, computing the optimal control sequence $\mathbf{U}^*$ is intractable for any significant problem length. To make the problem tractable, a sequential approach is used, where optimal solutions are computed for a short time horizon, N, at a fixed interval.

$$\mathbf{U}^*_{k:N+k} = \arg\max_{U_{k:N+k}} \mathbf{J}(\mathbf{U}_{k:N+k} | \mathbf{U}_{1:k-1}) \tag{2}$$

In practice, this approach is useful in developing effective sensing trajectories. While the choice of lookahead horizon, N, is limited by the available computational resources, it has a great effect on the value of the trajectories. Often, a single-step lookahead is used, where N is set to one, as in [7]. The use of a short lookahead horizon often produces myopic strategies, which fail to take into account valuable parts of the state space. The problem of myopia is often countered by optimizing an alternate objective function, which better accounts for 'global' goals. For instance, the authors in [4] add a term to $\mathbf{J}$ to steer the UAV away from areas of the search space of low value. In [3], an 'interference' term is added to the objective function to account for sensor overlap between UAVs.

This work examines a procedure for sequential optimization for cooperative sensing, but considers the lookahead horizon, N, as a parameter that can be varied online, to avoid problems of a myopic optimization. By varying N online, a control algorithm can adapt to the geometry of the objective function. To limit computational requirements, the optimization routine is also allowed to vary with N, to allow the use of computationally cheaper routines, as N grows large. An algorithm is presented that maintains team cooperation, despite different planning algorithms among team members.

COOPERATIVE SENSING

This section is concerned with developing some basic results regarding cooperative planning. We examine the performance of a team of mobile sensors trying to maximize a set function $J(\mathbf{Z})$, where $\mathbf{Z}$ is the aggregate of the team's observations. In this case, Z is a random variable, rather than actual observation data. The reward function $J(\mathbf{Z})$ can be thought of as a metric for the value of a set of measurements. The use of mobile sensors makes this problem especially difficult in that the choice of Z_k constrains possible choices of Z_{k+1}, as the platform moves to take sensor measurements. Lastly, it is important to note that Z refers to the outcome of some sensing action; in the general case, it will be required to distribute likelihood functions $p(Z|X)$ where X represents the random variable of interest.

Definition 1. *The impact of measurement Z^i is defined as $J(Z^i|Z^{1:i-1,i+1:m}) = J(Z^{1:m}) - J(Z^{1:i-1,i+1:m})$. Specifically, the impact of Z^i is the reward or value lost by removing measurement Z^i from measurement pool.*

Note that any value function, for a team of m mobile sensors taking k measurements apiece, can be written as $J(\mathbf{Z}) = \sum_{i=1}^{m} J_i(Z^i_{1:k}) + J_c(Z^1_{1:k}, Z^2_{1:k}, \ldots, Z^m_{1:k})$, where J_i is term local to each vehicle and J_c is term representing the coupling between vehicles. In this work, we shall assume that all agents have the same value function, and that $J_i(Z^i) = J(Z^i)$

We will consider reward functions J that are both submodular and non-decreasing, such as mutual information (MI). Mutual information is a metric for evaluating the value of a measurement sequence, prior to taking the measurements.

Definition 2 (Submodular). *A set function f is submodular if* $f(C \cup A) - f(A) \geq f(C \cup B) - f(B)\, \forall A \subseteq B$.

Submodularity captures the notion that the more measurements one adds to a pool, the less valuable an individual measurement becomes.

$$J(Z^1, Z^2) - J(Z^2) \geq J(Z^1, Z^2, Z^3) - J(Z^2, Z^3)$$

One consequence of submodularity is that $J(Z^2) \geq J(Z^2|Z^1)$.

Definition 3 (Nondecreasing). *A set function f is nondecreasing if* $f(B) \geq f(A)\, \forall A \subseteq B$.

The non-decreasing property implies that the addition of a measurement always increases your reward, $J(Z^1, Z^2) \geq J(Z^1)$; taking an observation of a target can never increase your uncertainty.

It is worth noting that the coupling term in a submodular, nondecreasing reward function is always negative.

$$\begin{aligned}
J_c(Z^{1:m}) &= J(Z^{1:m}) - J(Z^1) + J(Z^2) + \ldots + J(Z^m) \\
&= J(Z^1) + \ldots + J(Z^m|Z^{1:m-1}) - J(Z^1) \ldots - J(Z^m) && (3) \\
&\leq J(Z^1) + \ldots + J(Z^m) - J(Z^1) - \ldots - J(Z^m) && (4) \\
&= 0 && (5)
\end{aligned}$$

Eqn. (3) is an application of the chain rule, from Definition 1; while Eqn. (4) is due to sub-modularity. Also, the magnitude of the coupling term, in a submodular, nondecreasing reward function, is bounded by the sum of all but the largest J_i. Let $J(Z^1) \geq J(Z^i)\, \forall i$. Then $J_c \geq -\sum_{i=2}^{m} J(Z^i)$ This implies that the coupling could potentially negate all but the single best sensor measurement, i.e. $J \geq \max_i J(Z^i)$.

$$\begin{aligned}
J_c(Z^{1:m}) &= J(Z^{1:m}) - \sum_{i=1}^{m} J(Z^i) \\
&= J(Z^1) + J(Z^2|Z^1) \ldots + J(Z^m|Z^{1:m-1}) - \sum_{i=1}^{M} J(Z^i) \\
&= J(Z^2|Z^1) - J(Z^2) + \ldots + J(Z^m|Z^{1:m-1}) - J(Z^m) \\
&\geq -\sum_{i=2}^{m} J(Z^i) && (6)
\end{aligned}$$

Ignorant Strategies

Many multi-agent sensing problems, such as those in [4] use an ignorant, or coordinated approach, where sensor information is shared, but no information is shared regarding future actions. This strategy aims to induce cooperation through the previously collected information. While this strategy is effective for certain problems, there is no guarantee that an ignorant approach will improve performance over the use of a single sensor.

The Value of an Ignorant Policy Let $J^o = J(\bar{Z}^1, \ldots, \bar{Z}^m)$ be defined as the value of a policy where $\bar{Z}^k$ are selected according to some optimal (cooperative) policy, i.e. $J(\bar{Z}^1, \ldots, \bar{Z}^m) \geq J(Z^1, \ldots, Z^m)\, \forall Z^i$. Let $J^i = J(\tilde{Z}^1, \ldots, \tilde{Z}^m)$ be the value of an ignorant policy, where each $\tilde{Z}_k$ is selected according to $\tilde{Z}^k = \text{argmax}\, (J(Z^k))$. Then $J^o \leq mJ^i$.

$$\begin{aligned}
J(\bar{Z}^{1:m}) &= J(\bar{Z}^1) + J(\bar{Z}^2|\bar{Z}^1) + \ldots + J(\bar{Z}^m|\bar{Z}^1, \ldots, \bar{Z}^{m-1}) \\
&\leq J(\bar{Z}^1) + J(\bar{Z}^2) + \ldots + J(\bar{Z}^m) \\
&\leq m \max_k J(\bar{Z}^k) \\
&\leq m \max_{Z^k} J(Z^k) \\
&\leq m J^i && (7)
\end{aligned}$$

This yields the obvious result that an ignorant policy of m measurements will have have a value, at worst, $\frac{1}{m}$ of the optimal. This results implies that an approach, where only sensor measurements are shared, could lead to a policy where the efforts of all but one agent are wasted.

While there are no performance guarantees for an ignorant policy, there are guarantees for other policies. A greedy policy $\hat{Z}$ is defined by

$$\hat{Z}^k = \text{argmax}\, J(Z^k|\hat{Z}^1, \ldots \hat{Z}^{k-1}) \qquad (8)$$

A greedy policy is one where measurements are selected sequentially, to optimize the cost function, based on all previously selected measurements, i.e. $\hat{Z}^1 = \text{argmax}\, J(Z^1)$ and $\hat{Z}^2 = \text{argmax}\, J(Z^2|\hat{Z}^1)$. A lower bound on a greedy policy may be established using results from the previous section. This result was due to [8], and is repeated here, for a pair of measurements, only for completeness. Let $\bar{Z}^1, \bar{Z}^2$ be an optimal set of measurements.

$$
\begin{aligned}
J(\bar{Z}^1,\bar{Z}^2) &\le J(\bar{Z}^1,\bar{Z}^2,\hat{Z}^1,\hat{Z}^2) \\
&= J(\hat{Z}^1)+J(\hat{Z}^2|\hat{Z}^1)+J(\bar{Z}^1|\hat{Z}^2,\hat{Z}^1)+J(\bar{Z}^2|\bar{Z}^1,\hat{Z}^2,\hat{Z}^1) \\
&\le J(\hat{Z}^1)+J(\hat{Z}^2|\hat{Z}^1)+J(\bar{Z}^1)+J(\bar{Z}^2|\bar{Z}^1) \\
&\le 2J(\hat{Z}^1)+2J(\hat{Z}^2|\hat{Z}^1) \\
&= 2J(\hat{Z}^1,\hat{Z}^2) \qquad (9)
\end{aligned}
$$

This bound indicates that a greedy policy develops plans that are, at worst, half the value of the optimal. A greedy policy is a good tradeoff between an ignorant policy, where no expected information is exchanged, and a fully cooperative policy, where some negotiation takes place to find an optimal policy. When applied to a group of m cooperating agents, planning to take n sensor measurements, it is important to note that the bound only applies when the greedy policy is applied agent-wise. Specifically, each agent must plan its n sensor measurements optimally. The policy is greedy in the sense that the i^{th} agent must only plan with regard to the plans of $i-1$ agents. This result motivates the use of an agent-wise greedy algorithm cooperative planning.

Dealing with myopia When using a receding-horizon optimization scheme for cooperative sensing, it is important to account for myopia, to prevent degenerate trajectories in which an agent's measurements have no impact on the value function. To account for myopia in a receding-horizon optimization routine, a solution was proposed in [4] to allow an agent to switch to an algorithm with a longer planning horizon, when the agent is in an area of low sensing value. For a single agent, this seems to be an appropriate solution; however, this approach limits the ability to cooperate with agents using a shorter planning horizons. Specifically, by switching to a longer planning horizon, an agent loses its ability to evaluate the coupling term J_c.

Consider the case of a pair of cooperating agents, each performing a receding horizon optimization routine, with different horizon. Agent 2 forms a plan $\hat{Z}^2_{1:2} = \text{argmax}\, J(Z^2_{1:2}|Z^1_1)$, based on the available information regarding agent 1. In the worst case, the bound of Eqn. (6) is tight, and the efforts of agent 2 are wasted, as there is no guarantee that $J(\hat{Z}^2_{1:2}|Z^1_{1:2}) > 0$, since potentially, $J_1(Z^1_{1:2}) + J_c(\mathbf{Z}) = 0$.

In the worst case, the value of the partially cooperative plan is equal to the value of an ignorant plan, indicating that planning cooperatively to different horizons is not guaranteed to be yield better results than an ignorant policy. This result motivates the development of algorithms that explicitly account for different time horizons, and allow all team members to calculate the coupling term of the reward function, J_c.

For this example, possibly a better strategy is to allow agent 1 to run two sequential iterations of the 1 step algorithm, and then communicate that information to agent 2. Agent 2 may then accurately evaluate the coupling term, and form a plan with some value. In the case of 2 agents, this plan will be, at best, twice the value of previous strategy.

Algorithm for Cooperative Control In this section, an algorithm for receding horizon cooperative control is presented; myopia is dealt with by allowing for a variable optimization horizon. It is assumed that there exists a long term planning strategy that can efficiently develop trajectories by sacrificing optimality. Cooperation, however, is never sacrificed in that each team member can evaluate the coupling, J_c.

At this point, it is worth distinguishing between planning horizon, and planning scope. Define planning horizon, k^i_h as the number of look-ahead steps for the i^{th} agent's optimization algorithm, while planning scope, k_s, is a team parameter, and represents the maximum k_h of any member of the team. X^i_s is the vehicle position.

Each agent may locally increase their planning horizon until a plan is developed that has some value. Planning is done in an agent-wise greedy fashion; the algorithm is also greedy in the sense that agents with shorter planning horizons plan first. In the case where each agent can form a non-degenerate plan using the initial planning horizon k_o, the algorithm develops a normal greedy plan, as in Eqn. (8).

Algorithm 1 Planning Algorithm for i^{th} agent of m total

```
k^i = 0 (Let k^i be the length of the current plan)
k_s = k^i + k_o (k_o is initial min. planning horizon)
while k^j < k_s ∀ j do
    if k^i ≤ k^j and ( i < j ∀ k^i = k^j ) (Greedy policy) then
        (Plan if it is the i^th agent's turn)
        k^i_h = k^i + k_o
        while J(Z̄^i_{1:k^i_h} < threshold) do
            Z̄^i_{1:k^i_h} ≈ argmax J(Z^i_{k^i:k^i_h} | Z^j; X^i_s)
            k^i_h = k^i_h + 1
        end while
        k^i = k^i_h
        Transmit k^i_h, Z̄^i_{k:k^i}
        Update vehicle position X^i_s
    else
        Receive k^j_h, Z̄^j_{k_p:k^j}
    end if
    k_s = max_j k^j_h (Calculate planning scope)
end while
Execute plan to take measurements Z̄^i_{1:k_s}
```

It is assumed that the planning algorithm, line 7 in Algorithm 1, can vary with the planning horizon. To ensure computational tractability, it is assumed that efficient plans may be developed for large time horizons, at the expense of optimality. For agents in areas of value, where a short horizon produces a plan with some significant impact, a plan is developed by iterating the short-horizon planning algorithm, using the final condition of the last iteration as the initial condition of the next.

Ultimately, this algorithm represents a decentralized optimization of some objective function, where each agent makes approximations to sacrifice optimality for the sake of computational efficiency. These approximations, however, are specific to each agent, and motivated by the geometry of the objective function surrounding each agent. When in agent is in an area of value, optimality is sacrificed by planning greedily, in time. The mechanism for cooperation is useful in that not all vehicles have to be performing the same type of planning for cooperation to occur. All receding horizon schemes sacrifice some sort of optimality for the sake of computational tractability; this algorithm makes these approximations explicit, and allows them to vary online, to adapt to the local geometry of the objective function.

Local Planning Strategies Local planning strategies are inherently problem dependent, and will need to be adapted for each type of sensing task. For short term planning, gradient based strategies have proved effective for sensing problems, as in [7] and [9]. To ensure that Algorithm 1 is computationally tractable, it is assumed that, given a long planning horizon, there is an efficient method to calculate a suboptimal plan. One possible method is to search for the most significant possible measurement within the planning horizon, and then plan a path backwards to the vehicle initial position. Another option is to run a gradient based algorithm, but with a more coarse discretization of the control parameter. Regardless of the algorithm used to generate a given plan, the same objective function is used to calculate the impact, and the expected measurements are distributed among team members.

APPLICATION: SEARCH FOR MOBILE TARGET

Algorithm 1 was applied to the search problem detailed in [4]; a team of unmanned vehicles search for a target. The target is static, and a prior probability distribution of the target's coordinates (x,y) is known.

A kinematic model of a UAV with a constrained turn rate and a constant velocity was used in these simulations. The position of the vehicle is modeled by

$$\dot{x} = V cos(\psi) \tag{10}$$
$$\dot{y} = V sin(\psi) \tag{11}$$
$$\dot{\psi} = u \tag{12}$$
$$|u| \leq u_{max} \tag{13}$$

Equations (10)-(13) represents the dynamics of the UAV; specifically, the UAV moves in the horizontal plane with constant velocity, V, at a heading angle of ψ. The control, u, is $\dot{\psi}$.

It was assumed that the sensor takes measurements at some fixed rate, T_{sensor}. For the sake of gradient based path planning, the dynamics were discretised by holding the control constant over $T_{sensor} = \Delta T$. This implies that the UAV can make discrete decisions at each time step, but that the rate of heading change, $\dot{\psi}$ remains constant over a given time step.

BAYESIAN FILTER

Likelihood functions, $p(z|X)$, are distributed among the team for the sake of developing a common target estimate. A likelihood function is a function on X, for a fixed value of z. Recursive Bayesian estimation is then employed to maintain the target estimate. The data fusion algorithm consists of two parts, a prediction step, Eqn. (14), in which the target PDF evolves forward in time, and an update step, Eq. 15, in which sensor updates are applied to the PDF.

$$\begin{aligned} p(X_{k+1}) &= \int p(X_{k+1}, X_k) dX_k \\ &= \int p(X_{k+1}|X_k) p(X_k) dX_k \end{aligned} \tag{14}$$

$$p(X|Z_{1:k}) = \frac{1}{K} p(X) \prod_{i=1}^{k} p(Z_i|X) \tag{15}$$

These equations represent a general recursive Bayesian estimation procedure, such as a Kalman filter. In this work, p(X) was stored on a grid, as the update equation is inherently non-linear. In this example, the target was assumed static, $p(X_{k+1}) = p(X_k)$.

In the case of searching for a target, a sensor model is used that gives some likelihood of detection, given the target location. In this case, $Z_k = D$ indicates that the target was detected at time k. False detections are not modeled.

$$\begin{aligned} p(Z_k = D|X_k; X_s) &= \frac{P_o D_o^2}{D_o^2 + d^2} \\ d &= ||X_s - X|| \end{aligned} \tag{16}$$

d is the distance between the sensor, at position X_s, and the target. Lastly, it is assumed that the sensor can take samples at 1 Hz, and that the probability of detecting a target outside the footprint is zero; for this work, a circular footprint, of radius 120 meters, was assumed.

Control Strategy

This section presents the objective function used in [4] and an instantiation of Algorithm 1 that produces trajectories for a co-operating team. Additionally, a planning algorithm that is computationally tractable for long horizons, is presented.

Objective Function: Cumulative Probability of Detection Given $p(X)$, and a series of future observations $Z_{1:k}$, the probability of detecting the target in each of those observations may be calculated. This gives rise to the cumulative probability of detection (POD), or the probability that the target will be detected over some series of observations. Define Q as the probability of not detecting a target, for sensor readings 1 to k for m vehicles.

$$\begin{aligned} Q(Z_{1:k}^{1:m}) &= \int p(X|Z_{1:k}^{1:m} = \bar{D})dX \\ &= \int p(X)\prod_{i=1}^{m}\prod_{j=1}^{k} p(Z_j^i = \bar{D}|X;X^s)dX \end{aligned} \tag{17}$$

Then, the cumulative probability of detection (POD) at time k is equal to $1-Q$; this was used as an objective function in [9].

As this objective function satisfies the submodularity and non-decreasing properties, the analysis regarding cooperation from previous sections applies. It is worth noting that $p(Z = \bar{D}|X;X^s) \leq 1$. Thus, the addition of more measurements simply decreases Q_k, which increases the POD, satisfying the non-decreasing property.

To establish submodularity, let A, B and C represent measurement pools, or collection of observations where it is assumed that $Z = \bar{D}$. Then, POD is submodular if

$$\begin{aligned} POD(C\cup A) - POD(A) &\geq POD(C\cup B) - POD(B)\,\forall A \subseteq B. \\ -Q(C\cup A) + Q(A) &\geq -Q(C\cup B) + Q(B) \end{aligned} \tag{18}$$

To prove that Eqn. (18) is satisfied, assume that $C\cap B = \emptyset$ and $C \cap A = \emptyset$. The result still holds if this is not the case, but this assumption makes the proof more readable. Also, let $L(C) = \prod_{i\in C} p(Z^i = \bar{D}|X)$. Then, L represents a product of likelihood functions for a collection of measurements. It is worth noting that $L(C\cup A) = L(C)L(A)$. Also L is a non-increasing function; adding more measurements to a likelihood function can only decrease L. Rearranging terms to the left side of the equation yields

$$\begin{aligned} -Q(C\cup A) + Q(A) + Q(C\cup B) - Q(B) &\geq 0 \\ \int p(X)(-L(C\cup A) + L(A) + L(C\cup B) - L(B))dX &\geq 0 \\ \int p(X)(-L(C)L(A) + L(A) + L(C)L(B) - L(B))dX &\geq 0 \\ \int p(X)(L(A)(1-L(C)) + -L(B)(1-L(C)))dX &\geq 0 \\ \int p(X)(L(A)-L(B))(1-L(C))dX &\geq 0 \end{aligned} \tag{19}$$

The proof assumes $A \subseteq B$, which implies $(L(A)-L(B)) \geq 0$, due to the non-increasing nature of L. Also, $(1-L(C)) \geq 0$, because $L(C) \leq 1$. This fact implies that all the terms within the integral are non-negative, which completes the proof.

Trajectory Generation

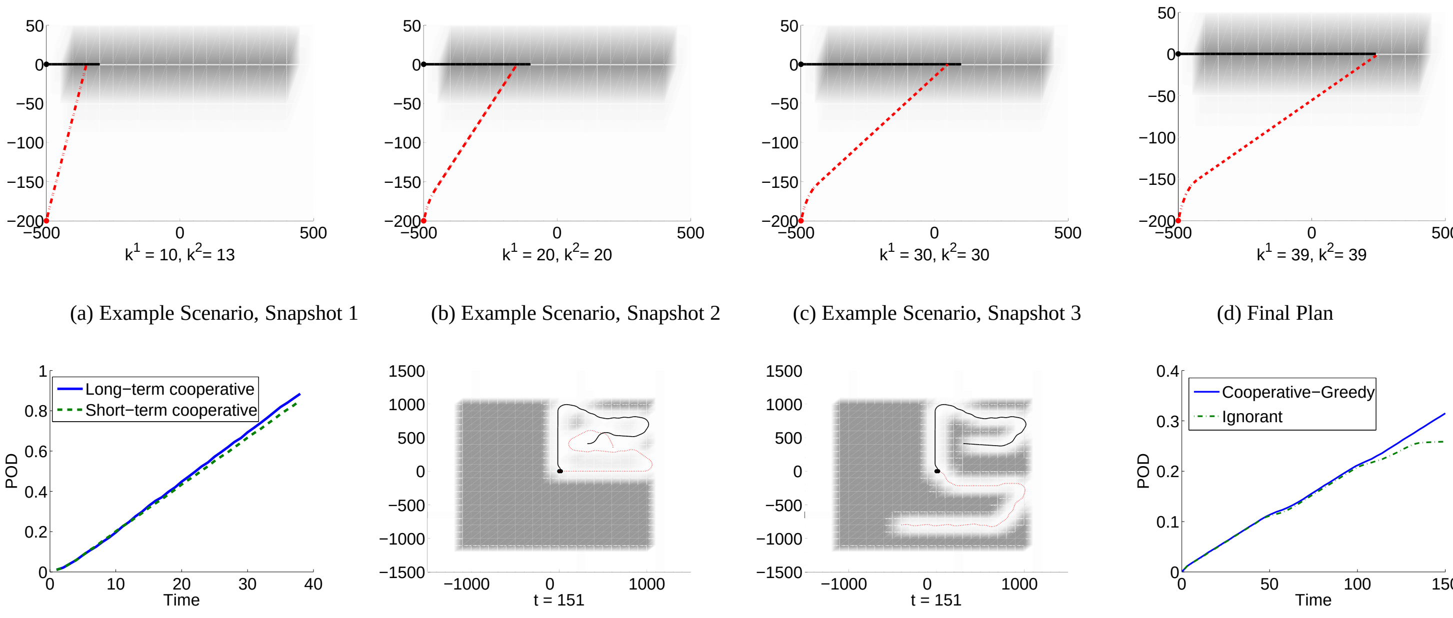
To develop trajectories, using the cost functions specified in the previous section, Algorithm 1 was implemented. The threshold value was set to %5 of the value of the best unconstrained sensor placement. A gradient-based optimization was run to determine the threshold value at each time step.

For short term planning, a gradient-based solver was used, using a 5-step planning horizon and a control parameterization of 1 second. To avoid local maxima, a series of initial conditions were used to seed the optimization at each step. For planning horizons greater than 5 steps, a long term planning strategy was employed, rather than using a gradient-based algorithm. Referred to as an *increasing-horizon planner*, this algorithm attempts to form a plan, of the shortest possible length, that has a value greater than some threshold ε. This algorithm takes the place of lines 7-10 in Algorithm 1, when the planning horizon is greater than 5 steps.

For a threshold ε, plans will be within $(k_h-1)\varepsilon$ of the optimal path. For certain problems, this planner can be run very quickly for long time horizons. For the search problem detailed here, the planner can be run efficiently when the modes of the

Algorithm 2 Increasing Horizon Planner

while $J(\bar{Z}_{k:k_h}) < threshold$ **do**
  $\bar{Z}_{k_h} = \text{argmax}\, J(Z_{k_h})$
  $k_h = k_h + 1$
  Calculate shortest path, X^s to take measurement $\bar{Z}_{k_h}$
  Find measurement sequence $\bar{Z}_{k:k_h-1}$ for path X^s
end while



(a) Example Scenario, Snapshot 1
(b) Example Scenario, Snapshot 2
(c) Example Scenario, Snapshot 3
(d) Final Plan
(e) Example Scenario: POD
(f) Ignorant Trajectories
(g) Cooperative Trajectories
(h) Probability of detection

Figure 1. FIGS. (a)-(d) ILLUSTRATE SNAPSHOTS OF ALGORITHM 1 FOR AN EXAMPLE PROBLEM. FIG. (e) REPRESENTS THE PROBABILITY OF DETECTION FOR THE EXAMPLE PROBLEM, FOR DIFFERENT COOPERATIVE STRATEGIES. FIGS. (f), (g) AND (h) INDICATE THE DIFFERENCES BETWEEN COOPERATIVE AND IGNORANT STRATEGIES. GREY INDICATES AREAS OF HIGH PROBABILITY MASS. DISTANCE UNITS ARE IN METERS; TIME UNITS IN SECONDS.

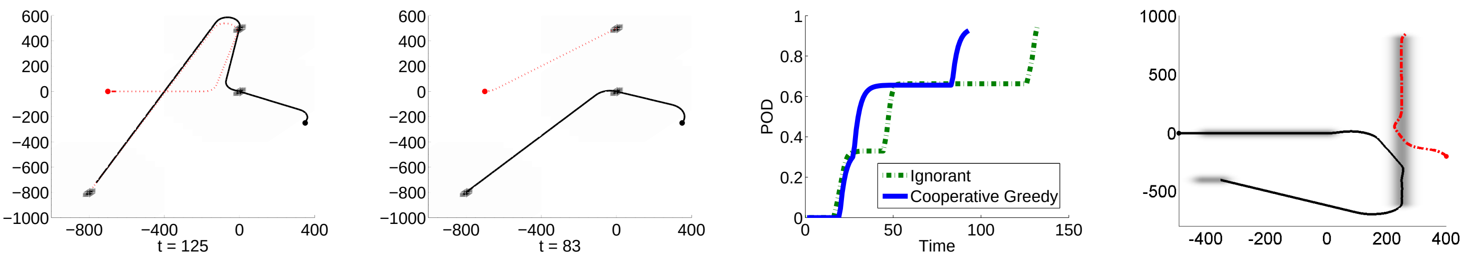
PDF, within the reachable set, may be evaluated quickly, to seed the unconstrained optimization. When the probability distribution is stored as a grid, evaluation of the modes is computationally efficient. For other representations, such as a particle filter, a different approach might be necessary. Also, the shortest path must easily be calculated; for a UAV with a constrained turn-rate, Dubin's paths are the optimal trajectories between any two points, and are easily calculated, as in [10].

SIMULATION RESULTS

Figures 1 (a)-(d) illustrate a single run of Algorithm 1 for a pair of UAVs searching for a target, whose PDF is indicated by the gray areas. UAV 1, at top left of the plot, is in an area of local value; the planning horizon is equal to 1 for the entire planning process. UAV 2 is positioned far enough away from the target PDF that no significant sensor measurements are possible within the initial planning horizon. Snapshot's of each vehicles plan are taken at specific instances in the algorithm. After successive iterations of Algorithm 1, the vehicles converge on a plan 39 seconds long. UAV 2 plans for the entire 39 seconds in a single step; UAV 1 plans for the 39 seconds by iterating the gradient descent algorithm. The trajectories of this variable horizon long-term cooperative strategy were compared to a strategy that only allows for cooperation over short time horizons; the probability of detection over 39 seconds, for each strategy, is shown in Fig. 2(e). The variable-horizon long-cooperative strategy clearly outperforms the short-term cooperative strategy. Using a fixed-horizon planning strategy, this type of plan could not be developed, as the value function is zero, within the initial lookahead horizon of the second UAV.

Figures 1(f) and 1(g) illustrate ignorant and cooperative-greedy trajectories for a pair of UAVs searching for a target with a uniform prior PDF. Qualitatively, the cooperative trajectories appear better than the ignorant, as the UAVs attempt to avoid sensor overlap; a comparison of the probability of detection values, Fig. 2(h), indicate this as well, as the cooperative trajectories better maximize POD.

Figures 2(a) and 2(b) indicate the trajectories and prior probability distributions for a pair of UAVs searching for a target; in this case, the target prior distribution is a sum of three widely separated Gaussian distributions. Figure 2(a) indicates trajectories for an algorithm in which short term cooperation is maintained, but long term plans are not shared among UAVs, as in [4]. This strategy results in an ineffective strategy; similar performance could be developed using a single vehicle, which is in line with the bound in Eqn. (7). While short term cooperation is maintained, for this scenario, long term cooperation is vital to



(a) Short-term cooperation only (b) Long-term cooperation (c) Probability of detection (d) An example two UAV scenario

Figure 2. FIGS. (a) AND (b) ILLUSTRATE TRAJECTORIES FOR A SCENARIO WHERE LONG TERM COOPERATIVE PLANNING IS GREATLY BENEFICIAL. FIG. (c) SHOWS THE POD FOR THOSE TRAJECTORIES. FIG. (d) PRESENTS A TARGET PRIOR WHICH REQUIRES BOTH SHORT AND LONG TERM PLANNING STRATEGIES.

develop an effective trajectory. Fig. 2(b) shows the UAV trajectories, for the same target prior distribution, developed using Algorithm 1. Both short- and long-term planning strategies are employed, but long term cooperation is maintained by planning to a common scope. Fig. 2(c) shows the performance metric for each of these strategies; clearly, planning to common scope more rapidly increases POD. This example scenario is interesting in that the problem might be aptly represented as a collection of tasks; the success of Algorithm 1 in this scenario indicates that separate task allocation schemes might be unnecessary for cooperative sensing problems.

Fig. 2(d) shows trajectories for a pair of UAVs searching for a target over the indicated prior distribution; due to the nature of the prior, both long- and short-term planning strategies are used to develop an effective trajectory.

CONCLUSION

Rather than treating receding horizon optimization as a necessary evil, the work presented here specifically accounts for its deficiencies. The need for different planning horizons, when cooperating to perform collaborative sensing, is demonstrated; furthermore, the potential loss of performance due to unsynchronized planning scope is shown. The algorithm presented successfully maintains cooperation among a team, despite the need for different planning horizons. Clearly, the success of such a receding horizon algorithm depends on the ability to develop computationally efficient algorithms, that sacrifice optimality, when the planning horizon grows large. These conditions hold for the example problem developed; for different representations of the underlying probability distribution, or different motion models of the vehicle, this condition may not hold. It is important to recognize that despite changes in the local planning scheme as the horizon increases, the mechanism for cooperation remains the same, as does the objective function.

REFERENCES

[1] Alighanbari, M., and How, J., 2006. “Robust decentralized task assignment for cooperative uavs”. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit.

[2] Ryan, A., Tisdale, J., Godwin, M., Coatta, D., Spry, S., Sengupta, R., and Hedrick, J., 2007. “A distributed control system for unmanned aerial vehicles executing collaborative sensing missions using multi-step plans”. In American Controls Conference.

[3] Flint, M., Polycarpou, M., and Fernández-Gaucherand, E., 2002. “Cooperative path for autonomous vehicles using dynamic programming”. In Proceedings of the 15th IFAC World Congress on Automatic Control.

[4] Wong, E., Bourgault, F., and Furukawa, T., 2005. “Multi-vehicle bayesian search for multiple lost targets”. In ICRA, IEEE.

[5] Mathews, G., Durrant-Whyte, H., Bourgault, F., and Furukawa, T., 2007. “Decentralised asynchronous optimisation and team decision making”. In IEEE Conference on Decision and Control.

[6] Singh, A., Krause, A., Guestrin, C., Kaiser, W., and Batalin., M., 2007. “Efficient planning of informative paths for multiple robots”. In International Joint Conference on Artificial Intelligence.

[7] Grocholsky, B., Bayraktar, S., Kumar, V., Taylor, C., and Pappas, G., 2004. “Synergies in feature localization by air-ground robot teams”. In 9th International Symposium on Experimental Robotics.

[8] Williams, J., 2007. “Information theoretic sensor management”. Doctoral thesis, Massachussetts Institute of Technology.

[9] Bourgault, F., Furukawa, T., and Durrant-Whyte, H., 2003. “Optimal search for a lost target in a bayesian world”. In Proc. Int. Conf. on Field and Service Robotics.

[10] Dubins, L. E., 1976. “On curves of minimal length with a

constraint on average curvature and with prescribed initial and terminal positions and tangents". *American Journal of Mathematics*, **79**, pp. 497–516.