2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009



# Coordinated Guidance of Autonomous UAVs via Nominal Belief-State Optimization

Scott A. Miller, Zachary A. Harris, and Edwin K. P. Chong

*Abstract*— **We apply the theory of partially observable Markov decision processes (POMDPs) to the design of guidance algorithms for controlling the motion of unmanned aerial vehicles (UAVs) with on-board sensors for tracking multiple ground targets. While POMDPs are intractable to optimize exactly, principled approximation methods can be devised based on Bellman's principle. We introduce a new approximation method called nominal belief-state optimization (NBO). We show that NBO, combined with other application-specific approximations and techniques within the POMDP framework, produces a practical design that coordinates the UAVs to achieve good long-term mean-squared-error tracking performance in the presence of occlusions and dynamic constraints.**

## I. Introduction

Interest in unmanned aerial vehicles (UAVs) for applications such as surveillance, search, and target tracking has increased in recent years, owing to significant progress in their development and a number of recognized advantages in their use [1], [2]. Of particular interest in ongoing research on this topic is the interplay among signal processing, robotics, and automatic control in the success of UAV systems.

This paper describes a principled framework for designing a planning and coordination algorithm to control a fleet of UAVs for the purpose of tracking ground targets. The algorithm runs on a central fusion node that collects measurements generated by sensors on-board the UAVs, constructs tracks from those measurements, plans the future motion of the UAVs to maximize tracking performance, and sends motion commands back to the UAVs based on the plan.

The focus of this paper is to illustrate a design framework based on the theory of *partially observable Markov decision processes* (POMDPs), and to discuss practical issues related to the use of the framework. With this in mind, the problem scenarios presented here are idealized, and are meant to illustrate qualitative behavior of a guidance system design. Moreover, the particular approximations employed in the design are examples and can certainly be improved. Nevertheless, the intent is to present a design approach that is flexible enough to admit refinements to models, objectives, and approximation methods without damaging the underlying structure of the framework.

S. A. Miller and Z. A. Harris are with Numerica Corporation, 4850 Hahns Peak Dr., Suite 200, Loveland, CO 80538. Emails: scott.miller@numerica.us, zach.harris@numerica.us.

E. K. P. Chong is with the Dept. of ECE, Colorado State University, Fort Collins, CO 80523-1373. Email: edwin.chong@colostate.edu.

This work was supported in part by AFOSR Grant No. FA9550-07-1-0360 and NSF Grant No. ECCS-0700559.

## II. Problem Description

The class of problems we pose in this paper is a rather schematic representation of the UAV guidance problem. Simplifications are assumed for ease of presentation and understanding of the key issues involved in sensor coordination. These simplifications include:

*2-D motion.* The targets are assumed to move in a plane on the ground, while the UAVs are assumed to fly at a constant altitude above the ground.

*Position measurements.* The measurements generated by the sensors are 2-D position measurements with associated covariances describing the position uncertainty. A simplified visual sensor (camera plus image processing) is assumed, which implies that the angular resolution is much better than the range resolution.

*Perfect tracker.* We assume that there are no false alarms and no missed detections, so exactly one measurement is generated for each target visible to the sensor. Also, perfect data association is usually assumed, so the tracker knows which measurement came from which target.

Despite these simplifications, the problem class has a number of important features that influence the design of a good planning algorithm. These include:

*Dynamic constraints.* These appear in the form of constraints on the motion of the UAVs. Specifically, the UAVs fly at a constant speed and have bounded lateral acceleration in the plane, which limits their turning radius. This is a reasonable model of the characteristics of small fixed-wing aircraft. The presence of dynamic constraints implies that the planning algorithm needs to include some form of lookahead for good long-term performance.

*Randomness.* The measurements have random errors, and the models of target motion are random as well.

*Spatially varying measurement error.* The range error of the sensor is an affine function of the distance between the sensor and the target. The bearing error of the sensor is constant, but that translates to a proportional error in Cartesian space as well. This spatially varying error is what makes the sensor placement problem meaningful.

*Occlusions.* There are occlusions in the plane that block the visibility of targets from sensors when they are on opposite sides of an occlusion. The occlusions are generally collections of rectangles in our models, though in the case studies presented they appear more as walls (thin rectangles). Targets are allowed to cross occlusions, and of course the UAVs are allowed to fly over them; their purpose is only to make the observation of targets more challenging.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

*Tracking objectives.* The performance objectives considered here are related to maintaining the best tracks on the targets. Normally, that means minimizing the mean squared error between tracks and targets, but we have also considered the avoidance of track swaps as a performance objective. This differs from most of the guidance literature, where the objective is usually posed as interpolation of way-points.

The UAV guidance problem considered here falls within the class of problems known as *sensor resource management* [3]. In its full generality, sensor resource management encompasses a large body of problems arising from the increasing variety and complexity of sensor systems, including dynamic tasking of sensors, dynamic sensor placement, control of sensing modalities (such as waveforms), communication resource allocation, and task scheduling within a sensor [4]. A number of approaches have been proposed to address the design of algorithms for sensor resource management, which can be broadly divided into two categories: myopic and nonmyopic.

Myopic approaches do not explicitly account for the future effects of sensor resource management decisions (i.e., there is no explicit planning or "lookahead"). One approach within this category is based on fuzzy logic and expert systems [5], which exploits operator knowledge to design a resource manager. Another approach uses information-theoretic measures as a basis for sensor resource management [6], [7], [8]. In this approach, sensor controls are determined based on maximizing a measure of "information."

Nonmyopic approaches to sensor resource management have gained increasing interest because of the need to account for the kinds of requirements described in this paper, which imply that foresight and planning are crucial for good long-term performance. In the context of UAV coordination and control, such approaches include the use of guidance rules [9], [10], [11], [2], oscillator models [12], and information-driven coordination [1], [13]. A more general approach to dealing with nonmyopic resource management involves stochastic dynamic programming formulations of the problem (or, more specifically, POMDPs). Because exact optimal solutions are practically infeasible to compute, recent effort has focused on obtaining *approximate* solutions, and a number of methods have been developed (e.g., see [14], [15], [16]). This paper contributes to the further development of this thrust by introducing a new approximation method, called *nominal belief-state optimization*, and applying it to the UAV guidance problem.

## III. POMDP Specification and Solution

### A. POMDP Formulation of Guidance Problem

To formulate our guidance problem in the POMDP framework, we must specify each of the components of the POMDP as they relate to the guidance system. This subsection is devoted to this specification. For a full treatment of POMDPs and related background, see [17]. For a discussion of POMDPs in sensor management, see [4]. Our specification here follows the terminology in [4].

**States.** In the guidance problem, three subsystems must be accounted for in specifying the state of the system: the sensor(s), the target(s), and the tracker. More precisely, the state at time $k$ is given by $x_k = (s_k, \zeta_k, \xi_k, P_k)$, where $s_k$ represents the sensor state, $\zeta_k$ represents the target state, and $(\xi_k, P_k)$ represents the track state. The sensor state $s_k$ specifies the locations and velocities of the sensors (UAVs) at time $k$. The target state $\zeta_k$ specifies the locations, velocities, and accelerations of the targets at time $k$. Finally, the track state $(\xi_k, P_k)$ represents the state of the tracking algorithm: $\xi_k$ is the posterior mean vector and $P_k$ is the posterior covariance matrix, standard in Kalman filtering algorithms.

**Action.** In our guidance problem, we assume a standard model where each UAV flies at constant speed and its motion is controlled through turning controls that specify lateral instantaneous accelerations. The lateral accelerations can take values in an interval $[-a_{\max}, a_{\max}]$, where $a_{\max}$ represents a maximum limit on the possible lateral acceleration. So the action at time $k$ is given by $a_k \in [-1, 1]^{N_{\text{sens}}}$, where $N_{\text{sens}}$ is the number of UAVs, and the components of the vector $a_k$ specify the normalized lateral acceleration of each UAV.

**State-transition law.** The state-transition law specifies how each component of the state changes from one time step to the next. In general, the transition law takes the form $x_{k+1} \sim p_k(\cdot \mid x_k)$ for some time-varying distribution $p_k$. However, the model for the UAV guidance problem constrains the form of the state transition law. The sensor state evolves according to $s_{k+1} = \psi(s_k, a_k)$, where $\psi$ is the map that defines how the state changes from one time step to the next depending on the acceleration control as described above. The target state evolves according to $\zeta_{k+1} = f(\zeta_k) + v_k$ where $v_k$ represents an i.i.d. random sequence and $f$ represents the target motion model. Most of our simulation results use a nearly constant velocity (NCV) target motion model, except for Section VI-B which uses a nearly constant acceleration (NCA) model. In all cases $f$ is linear, and $v_k$ is normally distributed. We write $v_k \sim \mathcal{N}(0, Q_k)$ to indicate the noise is normal with zero mean and covariance $Q_k$.

Finally, the track state $(\xi_k, P_k)$ evolves according to a tracking algorithm, which is defined by a data association method and the Kalman filter update equations. Since our focus is on UAV guidance and not on practical tracking issues, in most cases a "truth tracker" is used, which always associates a measurement with the track corresponding to the target being detected.

**Observations and observation law.** In general, the observation law takes the form $z_k \sim q_k(\cdot \mid x_k)$ for some time-varying distribution $q_k$. In our guidance problem, since the state has four separate components, it is convenient to express the observation with four corresponding components. The sensor state and track state are assumed to be fully observable. So, for these components of the state, the observations are equal to the underlying state components: $z_k^s = s_k$, $z_k^\xi = \xi_k$, and $z_k^P = P_k$. The target state, however, is not directly observable; instead, what we have are random measurements of the target state that are functions of the

locations of the targets *and* the sensors.

Let $\zeta_k^{\text{pos}}$ and $s_k^{\text{pos}}$ represent the position vectors of the target and sensor, respectively, and let $h(\zeta_k, s_k)$ be a boolean-valued function that is true if the line of sight from $s_k^{\text{pos}}$ to $\zeta_k^{\text{pos}}$ is unobscured by any occlusions. Furthermore, we define a 2D position covariance matrix $R_k(\zeta_k, s_k)$ that reflects a $10\%$ uncertainty in the range from sensor to target, and $0.01\pi$ radian angular uncertainty, where the range is taken to be at least 10 meters. Then the measurement of the target state at time $k$ is given by

$$z_k^{\zeta} = \begin{cases} \zeta_k^{\text{pos}} + w_k & \text{if } h(\zeta_k, s_k) = \text{true}, \\ \emptyset \text{ (no measurement)} & \text{if } h(\zeta_k, s_k) = \text{false}, \end{cases}$$

where $w_k$ represents an i.i.d. sequence of noise values distributed according to the normal distribution $\mathcal{N}(0, R_k(\zeta_k, s_k))$.

**Cost function.** The cost function we use in our guidance problem is the mean squared tracking error, defined by

$$C(x_k, a_k) = \mathop{\mathrm{E}}_{v_k, w_{k+1}} \left[ \|\zeta_{k+1} - \xi_{k+1}\|^2 \mid x_k, a_k \right]. \quad (1)$$

**Belief state.** Although not a part of the POMDP specification, it is convenient at this point to define our notation for the belief state for the guidance problem. The belief state at time $k$ is a *distribution* over states, $b_k(x) = \mathrm{P}_{x_k}\big(x \mid z_0, \ldots, z_k;\ a_0, \ldots, a_{k-1}\big)$. For our guidance problem, the belief state is given by $b_k = (b_k^s, b_k^{\zeta}, b_k^{\xi}, b_k^P)$ where

$$\begin{aligned} b_k^s(s) &= \delta(s - s_k) \\ b_k^{\zeta} &\quad \text{updated with } z_k^{\zeta} \text{ using Bayes theorem} \\ b_k^{\xi}(\xi) &= \delta(\xi - \xi_k) \\ b_k^P(P) &= \delta(P - P_k). \end{aligned}$$

Note that those components of the state that are directly observable have delta functions representing their corresponding belief-state components.

### *B. Optimal Policy*

Given the POMDP formulation of our problem, our goal is to select actions over time to minimize the expected cumulative cost (we take expectation here because the cumulative cost is a random variable, being a function of the random evolution of $x_k$). To be specific, suppose we are interested in the expected cumulative cost over a time horizon of length $H$: $k = 0, 1, \ldots, H-1$. The problem is to minimize the cumulative cost over horizon $H$, given by

$$J_H = \mathrm{E}\left[ \sum_{k=0}^{H-1} C(x_k, a_k) \right]. \quad (2)$$

The goal is to pick the actions so that the objective function is minimized. In general, the action chosen at each time should be allowed to depend on the entire history up to that time (i.e., the action at time $k$ is a random variable that is a function of all observable quantities up to time $k$). However, it turns out that if an optimal choice of such a sequence of actions exists, then there is an optimal choice of actions that depends only on "belief-state feedback." In other words, it suffices for the action at time $k$ to depend only on the belief state at time $k$, as alluded to before.

Let $b_k$ be the belief state at time $k$, updated incrementally using Bayes rule. The objective can be written in terms of belief states:

$$J_H = \mathrm{E}\left[ \sum_{k=0}^{H-1} c(b_k, a_k) \;\middle|\; b_0 \right], \quad c(b, a) = \int C(x, a) b(x)\, dx \quad (3)$$

were $\mathrm{E}\big[\cdot \mid b_o\big]$ represents conditional expectation given $b_0$. Let $\mathcal{B}$ represent the set of possible belief states, and $\mathcal{A}$ the set of possible actions. So what we seek is, at each time $k$, a mapping $\pi_k^* : \mathcal{B} \to \mathcal{A}$ such that if we perform action $a_k = \pi_k^*(b_k)$, then the resulting objective function is minimized. This is the desired optimal policy.

The key result in POMDP theory is Bellman's principle. Let $J_H^*(b_0)$ be the optimal objective function value (over horizon $H$) with $b_0$ as the initial belief state. Then, *Bellman's principle* states that

$$\pi_0^*(b_0) = \operatorname*{argmin}_a \left\{ c(b_0, a) + \mathrm{E}\big[ J_{H-1}^*(b_1) \mid b_0, a \big] \right\}$$

is an optimal policy, where $b_1$ is the random next belief state (with distribution depending on $a$), $\mathrm{E}\big[\cdot \mid b_0, a\big]$ represents conditional expectation (given $b_0$ and action $a$) with respect to the random next state $b_1$, and $J_{H-1}^*(b_1)$ is the optimal cumulative cost over the time horizon $1, \ldots, H$ starting with belief state $b_1$.

Define the *Q-value* of taking action $a$ at state $b_0$ as

$$Q_H(b_0, a) = c(b_0, a) + \mathrm{E}\big[ J_{H-1}^*(b_1) \mid b_0, a \big].$$

Then, Bellman's principle can be rewritten as

$$\pi_0^*(b_0) = \operatorname*{argmin}_a Q_H(b_0, a),$$

i.e., the optimal action at belief state $b_0$ is the one with smallest $Q$-value at that belief state. Thus, Bellman's principle instructs us to minimize a modified cost function ($Q_H$) that includes the term $\mathrm{E}\big[J_{H-1}^*\big]$ indicating the expected future cost of an action; this term is called the *expected cost-to-go* (ECTG). By minimizing the $Q$-value that includes the ECTG, the resulting policy has a *lookahead* property that is a common theme among POMDP solution approaches.

For the optimal action at the next belief state $b_1$, we would similarly define the $Q$-value

$$Q_{H-1}(b_1, a) = c(b_1, a) + \mathrm{E}\big[ J_{H-2}^*(b_2) \mid b_1, a \big],$$

where $b_2$ is the random next belief state and $J_{H-2}^*(b_2)$ is the optimal cumulative cost over the time horizon $2, \ldots, H$ starting with belief state $b_2$. Bellman's principle then states that the optimal action is given by

$$\pi_1^*(b_1) = \operatorname*{argmin}_a Q_{H-1}(b_1, a).$$

A common approach in on-line optimization-based control is to assume that the horizon is long enough that the difference between $Q_H$ and $Q_{H-1}$ is negligible. This has two implications: first, the time-varying optimal policy $\pi_k^*$

may be approximated by a *stationary* policy, denoted $\pi^*$; second, the optimal policy is given by

$$\pi^*(b) = \operatorname*{argmin}_a Q_H(b, a),$$

where now the horizon is fixed at $H$ regardless of the current time $k$. This approach is called *receding horizon control*, and is practically appealing because it provides lookahead capability without the technical difficulty of infinite-horizon control. Moreover, there is usually a practical limit to how far models may be usefully predicted. Henceforth we will assume the horizon length is constant and drop it from our notation.

In summary, we seek a policy $\pi^*(b)$ that, for a given belief state $b$, returns the action $a$ that minimizes $Q(b, a)$, which in the receding-horizon case is

$$Q(b, a) = c(b, a) + \mathrm{E}\left[J^*(b') \mid b, a\right],$$

where $b'$ is the (random) belief state after applying action $a$ at belief state $b$, and $c(b, a)$ is the associated cost. The second term in the $Q$-value is in general difficult to obtain, especially because the belief-state space is large. For this reason, approximation methods are necessary. In the next section, we describe our algorithm for approximating $\operatorname{argmin}_a Q(b, a)$.

## IV. Approximation Method

There are two aspects of a general POMDP that make it intractable to solve exactly. First, it is a stochastic control problem, so the dynamics are properly understood as constraints on *distributions* over the state space, which are infinite-dimensional in the case of a continuous state space as in our tracking application. In practice, solution methods for Markov decision processes employ some parametric representation or nonparametric (i.e., Monte Carlo or "particle") representation of the distribution, to reduce the problem to a finite-dimensional one. Intelligent choices of finite-dimensional approximations are derived from Bellman's principle characterizing the optimal solution. POMDPs, however, have the additional complication that the state space *itself* is infinite-dimensional, since it includes the belief state which is a distribution; hence, the belief state must also be approximated by some finite-dimensional representation. In Section IV-A we present a finite-dimensional approximation to the problem called *nominal belief-state optimization* (NBO), which takes advantage of the particular structure of the tracking objective in our application.

Secondly, in the interest of long-term performance, the objective of a POMDP is often stated over an arbitrarily long or infinite horizon. This difficulty is typically addressed by truncating the horizon to a finite length and adding an approximate ECTG, as discussed in Section IV-B.

Before proceeding to the detailed description of our NBO approach, we first make two simplifying approximations that follow from standard assumptions for tracking problems. The first approximation, which follows from the assumption of a correct tracking model and Gaussian statistics, is that the belief-state component for the target can be expressed as

$$b_k^\zeta = \mathcal{N}(\xi_k, P_k) \tag{4}$$

and can be updated using (extended) Kalman filtering (by the equation above we mean that the distribution represented by $b_k^\zeta$ is Gaussian with mean $\xi_k$ and covariance $P_k$). We adopt this approximation for the remainder of this paper. The second approximation, which follows from the additional assumption of correct data association, is that the cost function can be written as

$$\begin{aligned} &c(b_k, a_k) \\ &= \int \mathop{\mathrm{E}}_{v_k, w_{k+1}} \left[ \left\| \zeta_{k+1} - \xi_{k+1} \right\|^2 \mid s_k, \zeta, \xi_k, a_k \right] b_k^\zeta(\zeta)\, d\zeta \\ &= \operatorname{Tr} P_{k+1}, \end{aligned} \tag{5}$$

where Tr represents trace. We have studied the impact of this approximation in the context of tracking with data association ambiguity [18], but space restrictions preclude further discussion here.

### A. Nominal Belief-State Optimization (NBO)

A number of POMDP approximation methods have been studied in the literature; for a detailed discussion of these methods applied to sensor resource management problems, see [14]. These methods either directly approximate the $Q$-value $Q(b, a)$ or indirectly approximate the $Q$-value by approximating the cost-to-go $J^*(b)$.

The NBO approach may be summarized as

$$J^*(b) \approx \min_{(a_k)_k} \sum_k c(\hat{b}_k, a_k), \tag{6}$$

where $(a_k)_k$ means the ordered list $(a_0, a_1, \ldots)$, and $(\hat{b}_k)_k$ represents a *nominal* sequence of belief states. Thus, NBO resembles both the hindsight and foresight optimization approaches in [14], but with the expectation approximated by one sample. The central motivation behind NBO is computational efficiency. If one cannot afford to simulate multiple samples of the random noise sequences to estimate expectations, and only one realization can be chosen, it is natural to choose the "nominal" sequence (e.g., maximum likelihood or mean). The nominal noise sequence leads to a nominal belief-state sequence $(\hat{b}_k)_k$ as a function of the chosen action sequence $(a_k)_k$. Note that in NBO, the optimization is over a fixed sequence $(a_k)_k$ rather than a noise-dependent sequence or a policy.

There are two points worth emphasizing about the NBO approach. First, the nominal belief-state sequence is not fixed, as (6) might suggest; rather, the underlying random variables are fixed at nominal values and the belief states become deterministic functions of the chosen actions. Second, the expectation implicit in the incremental cost $c(\hat{b}_k, a_k)$ (recall (1) and (3)) need not be approximated by the "nominal" value. In fact, for the mean-squared-error cost we use in the tracking application, the nominal value would be 0. Instead, we use the fact that the expected cost can be evaluated analytically by (5) under the previously stated assumptions of correct tracking model, Gaussian statistics, and correct data association.

Because NBO approximates the belief-state evolution but not the cost evaluation, the method is suitable when the

primary effect of the randomness appears in the cost, not in the state prediction. Thus, NBO should perform well in our tracking application as long as the target motion is reasonably predictable with the tracking model within the chosen planning horizon.

The general procedure for using the NBO approximation may be summarized as follows:

1) Write the state dynamics as functions of zero-mean noise. For example, borrowing from the notation of Section III-A:

$$\begin{aligned} x_{k+1} &= f(x_k, a_k) + v_k, & v_k &\sim \mathcal{N}(0, Q_k) \\ z_k &= g(x_k) + w_k, & w_k &\sim \mathcal{N}(0, R_k)\,. \end{aligned}$$

2) Define *nominal belief-state* sequence $(\hat{b}_1, \ldots, \hat{b}_{H-1})$ based on $b_{k+1} = \Phi(b_k, a_k, v_k, w_{k+1})$:

$$\hat{b}_{k+1} = \Phi(\hat{b}_k, a_k, 0, 0), \qquad \hat{b}_0 = b_0.$$

In the linear Gaussian case, this is the *maximum a posteriori* (MAP) estimate of $b_k$.

3) Replace expectation over random future belief states

$$J_H(b_0) = \mathop{\mathrm{E}}_{b_1,\ldots,b_H}\left[\sum_{k=1}^{H} c(b_k, a_k)\right]$$

with the *sample* given by nominal belief state sequence

$$J_H(b_0) \approx \sum_{k=1}^{H} c(\hat{b}_k, a_k). \tag{7}$$

4) Optimize over action sequence $(a_0, \ldots, a_{H-1})$.

In the specific case of tracking, recall that the belief state $b_k^\zeta$ corresponding to the target state $\zeta_k$ is identified with the track state $(\xi_k, P_k)$ according to (4). Thus, the nominal belief state $\hat{b}_k^\zeta$ evolves according to the nominal track state trajectory $(\hat{\xi}_k, \hat{P}_k)$ given by the (extended) Kalman filter equations with an exactly zero noise sequence. This reduces to

$$\begin{aligned} \hat{b}_k^\zeta &= \mathcal{N}\Big(\hat{\xi}_k, \hat{P}_k\Big) \\ \hat{\xi}_{k+1} &= F_k \hat{\xi}_k \\ \hat{P}_{k+1} &= \Big[(F_k \hat{P}_k F_k^T + Q_k)^{-1} \\ &\qquad + H_{k+1}^T [R_{k+1}(\hat{\xi}_k, s_k)]^{-1} H_{k+1}\Big]^{-1}, \end{aligned}$$

where the (linearized) target motion model is given by

$$\begin{aligned} \zeta_{k+1} &= F_k \zeta_k + v_k, & v_k &\sim \mathcal{N}(0, Q_k) \\ z_k &= H_k \zeta_k + w_k, & w_k &\sim \mathcal{N}(0, R_k(\zeta_k, s_k))\,. \end{aligned}$$

The incremental cost of the nominal belief state is then

$$c(\hat{b}_k, a_k) = \mathrm{Tr}\, \hat{P}_{k+1} = \sum_{i=1}^{N_{\text{targ}}} \mathrm{Tr}\, \hat{P}_{k+1}^i$$

where $N_{\text{targ}}$ is the number of targets.

### B. Finite Horizon

In the guidance problem we are interested in long-term tracking performance. For the sake of exposition, if we idealize this problem as an infinite-horizon POMDP (ignoring the attendant technical complications), Bellman's principle can be stated as

$$J_\infty^*(b_0) = \min_\pi \mathrm{E}\left[\sum_{k=0}^{H-1} c(b_k, \pi(b_k)) + J_\infty^*(b_H)\right] \tag{8}$$

for any $H < \infty$. The term $\mathrm{E}[J_\infty^*(b_H)]$ is the expected cost to go (ECTG) from the end of the horizon $H$. If $H$ represents the practical limit of horizon length, then (8) may be approximated in two ways:

$$J_\infty^*(b_0) \approx \min_\pi \mathrm{E}\left[\sum_{k=0}^{H-1} c(b_k, \pi(b_k))\right] \qquad \text{(truncation)}$$

$$J_\infty^*(b_0) \approx \min_\pi \mathrm{E}\left[\sum_{k=0}^{H-1} c(b_k, \pi(b_k)) + \hat{J}(b_H)\right] \qquad \text{(HECTG)}$$

where $\hat{J}$ represents a heuristic approximation to the ECTG. The first amounts to ignoring the ECTG term, and is often the approach taken in the literature. The second replaces the exact ECTG with a heuristic approximation, typically a gross approximation that is quick to compute. To benefit from the inclusion of a heuristic ECTG (HECTG) term in the cost function for optimization, $\hat{J}$ need only be a better estimate of $J_\infty^*$ than a *constant*. Moreover, the utility of the approximation is in how well it rank actions, not in how well it estimates the ECTG. We develop a HECTG method for our problem in Section V-B.

## V. Single UAV Case

We begin our assessment of a POMDP-based design with the simple case of a single UAV and two targets, where the two targets move along parallel straight-line paths. This is enough to demonstrate the qualitative behavior of the method. It turns out that a straightforward but naive implementation of the POMDP approach leads to performance problems, but these can be overcome by employing an approximate expected cost-to-go (ECTG) term in the objective together with a sufficiently long lookahead horizon.

### A. Scenario Trajectory Plots

First we describe what is depicted in the scenario trajectory plots that appear throughout this section. See, for example, Figures 1 and 2. Each target location at each measurement time is indicated by a small colored dot (blue or red, which is admittedly difficult to see at the scale of the figures). The targets in most scenarios move in straight horizontal lines from left to right at constant speed. The track covariances are indicated by colored ellipses at each measurement time (again, blue or red); these are 1-sigma ellipses corresponding to the position component of the covariances, centered at the mean track position.

The UAV trajectory is plotted as a thin black line, with an arrow periodically. Large X's appear on the tracks that are

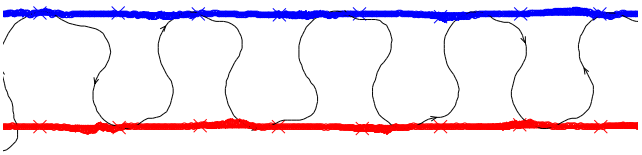

Fig. 1. Myopic policy with no occlusion

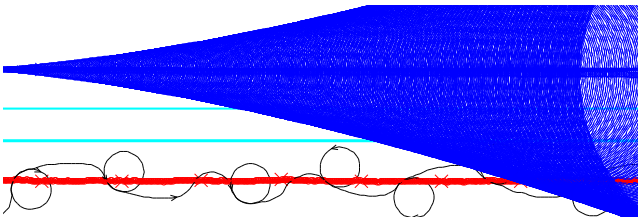

Fig. 2. Myopic policy with occlusions

synchronized with the arrows on the UAV trajectory, to give a sense of relative positions at any time.

Finally, occlusions are indicated by thick light-green lines. When the line of sight from a sensor to a target intersects an occlusion, that target is not visible from that sensor. This is a crude model of buildings or walls that block the visibility of certain areas of the ground from different perspectives. It is not meant to be realistic, but serves to illustrate the effect of occlusions on the UAV guidance algorithm.

### B. Weighted Trace Penalty

Following the NBO procedure, our first design for guiding the UAV optimizes the cost function (7) within a receding horizon approach, issuing only the command $a_0$ and re-optimizing at the next step. In the simplest case, the policy is a myopic one: choose the next action that minimizes the immediate cost at the next step based on current state information. This is equivalent to a receding horizon approach with $H = 1$ and no ECTG term. The behavior of this policy in a scenario with two targets moving at constant velocity along parallel paths is illustrated in Figure 1. For this scenario, the behavior with $H > 1$ (applying NBO) is not qualitatively different. However, it is easy to see that if occlusions were introduced, some lookahead (e.g., longer planning horizon) would be necessary to anticipate the loss of observations. In this case, the simple myopic policy would be suboptimal. This is illustrated in Figure 2, where there are two horizontal walls separating the targets in such a way that the myopic policy does not ever visit the top target (resulting in ever growing track covariance ellipses). So in general we will need to use $H > 1$ and also an approximation to the ECTG term (which we call HECTG), described next.

In our tracking application, a heuristic way to represent the ECTG is in the growth of the covariance of the track on an occluded target while it remains occluded. We estimate this growth by a *weighted trace penalty* (WTP) term, which is a product of the current covariance trace and the *minimum distance to observability* (MDO) for a currently occluded target, a term we define precisely below. With the UAV

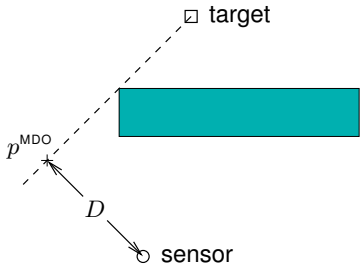


Fig. 3. Minimum distance to observability

moving at a constant speed, this is roughly equivalent to a scaling of the trace by the time it takes to observe the target. When combined with the trace term that is already in the cost function, this amounts to an approximation of the track covariance at the time the target is finally observed. More accurate approximations are certainly possible, but this simple approximation is sufficient to achieve the desired effect.

Specifically, the ECTG term using the WTP has the form

$$\hat{J}(b) = J_{\text{WTP}}(b) := \gamma D(s, \xi^{\mathbf{i}}) \operatorname{Tr} P^{\mathbf{i}}, \tag{9}$$

where $\gamma$ is a positive constant, $\mathbf{i}$ is the index of the worst occluded target, $\mathbf{i} = \operatorname{argmax}_{i \in \mathcal{I}} \operatorname{Tr} P^i$ with $\mathcal{I} = \{i \mid \xi^i \text{ invisible from } s\}$, and $D(s, \xi)$ is the MDO, i.e., the distance from the sensor location given by $s$ to the closest point $p^{\text{MDO}}(s, \xi)$ from which the target location given by $\xi$ is observable. Figure 3 is a simple illustration of the MDO concept. Given a single rectangular occlusion, $p^{\text{MDO}}(s, \xi)$ and $D(s, \xi)$ can be found very easily. Given multiple rectangular occlusions, the exact MDO is cumbersome to compute, so we use a fast approximation instead. For each rectangular occlusion $j$, we compute $p_j^{\text{MDO}}(s, \xi)$ and $D_j(s, \xi)$ as if $j$ were the only occlusion. Then we have $D(s, \xi) \geq \max_j D_j(s, \xi) > 0$ whenever $\xi$ is occluded from $s$, so we use $\max_j D_j(s, \xi)$ as a generally suitable approximation to $D(s, \xi)$.

The reason a worst-case among the occluded targets is selected, rather than including a term for each occluded target, is that this forces the UAV to at least obtain an observation on one target instead of being pulled toward two separate targets and possibly never observing either one. The true ECTG certainly includes costs for all occluded targets. However, given that the ECTG can only be approximated, the quality of the approximation is ultimately judged by whether it leads to the correct ranking of action plans within the horizon, and not by whether it closely models the true ECTG value. We claim that by applying the penalty to only the worst track covariance, the chosen actions are closer to the optimal policy than what would result by applying the penalty to all occluded tracks.

For convenience, let WTP($H$) denote the procedure of optimizing the NBO cost function with horizon length $H$ plus the WTP estimate of the ECTG:

$$\min_{a_0, \ldots, a_{H-1}} \sum_{k=0}^{H-1} c(\hat{b}_k, a_k) + J_{\text{WTP}}(\hat{b}_H). \tag{10}$$

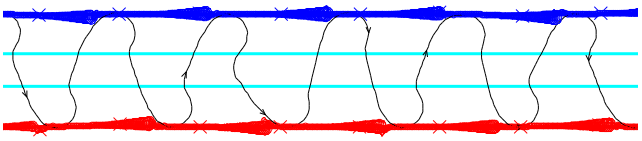

Fig. 4. WTP(4) policy with occlusions

Figure 4 shows the performance of WTP(4), which illustrates the benefits of lookahead and HECTG (overcoming the suboptimal behavior in Figure 2).

## VI. Multiple UAV Case

As it stands, the procedure developed for the single UAV case is ill-suited to the case of multiple UAVs, because the WTP is defined with only a single sensor in mind. We develop an extension of the WTP to multiple sensors here (called MWTP), and apply this extension to a new scenario to demonstrate the coordination of two sensors.

### A. Extension of WTP

A slight modification of the WTP defined in (9) can certainly be used as an ECTG in scenarios with more than one sensor, e.g.,

$$\hat{J}(b) = \gamma \min_j D(s^j, \xi^{\mathbf{i}}) \operatorname{Tr} P^{\mathbf{i}} \tag{11}$$

where $s^j$ is the state of sensor $j$. However, this underutilizes the sensors, because only one sensor can affect the ECTG. One would like the ECTG to guide two sensors toward two separate occluded targets if it makes sense to do so. On the other hand, if one sensor can "cover" two occluded targets efficiently, there is no need to modify the motion of a second sensor. The problem, therefore, is to decide which sensor will receive responsibility for each occluded target.

It is natural to assign the "nearest" sensor to an occluded target, i.e., the one that minimizes the MDO as in (11). However, to account for the effect of previous assignments to that sensor, the MDO should not be measured along a straight line directly from the starting position of the sensor, but rather, along the path the sensor takes while making observations on previously assigned targets. In the spirit of the WTP for a single sensor, it is assumed that if multiple occluded targets are assigned to a sensor, the most uncertain track (the one with the highest covariance trace) is the one that appears in the WTP and governs the motion of the sensor, until the target is actually observed; then, the next most uncertain track appears in the WTP, and so on. So, roughly speaking, the sensor makes observations of occluded targets in order of decreasing uncertainty.

Therefore, a *multiple weighted trace penalty* (MWTP) term is computed according to the following procedure:

1) Find the set of targets occluded from all sensors, and sort in order of decreasing $\operatorname{Tr} P^i$.
2) Set $\hat{J} = 0$, and $D_j = 0$ for each sensor $j$.
3) For each occluded target $i$ (in order):
   a) Find $\mathbf{j} = \operatorname{argmin}_j \left\{ D_j + D(s^j, \xi^i) \right\}$.
   b) If $D_{\mathbf{j}} = 0$ then set $\hat{J} \leftarrow \hat{J} + \gamma D(s^{\mathbf{j}}, \xi^i) \operatorname{Tr} P^i$.
   c) Set $D_{\mathbf{j}} \leftarrow D_{\mathbf{j}} + D(s^{\mathbf{j}}, \xi^i)$ and $s^{\mathbf{j}} \leftarrow p^{\text{MDO}}(s^{\mathbf{j}}, \xi^i)$.

This procedure is an approximation in several respects. First, it ignores the motion of the targets in the interval of time it takes the sensor to move from one $p^{\text{MDO}}$ location to the next. Second, it ignores the dynamic constraints of the UAVs. The total distance is computed by a greedy, suboptimal algorithm. None of these deficiencies is insurmountable, but for the purpose of a quick heuristic ECTG for ranking action plans, this MWTP is sufficient.

### B. Coordinated sensor motion

Figures 5–7 show snapshots of a scenario illustrating the coordination capability of the guidance algorithm using the MWTP from the previous section as an ECTG term. There are three targets (red, blue, and black) and two sensor UAVs (black and green).

Initially, the three targets are divided into two regions by an occlusion, and one sensor covers each region. At this point $H = 1$ is a sufficient horizon. Then the black target heads down and crosses two occlusions to enter the bottom region. In response, the green UAV chases after the downward-bound target, while the black UAV moves to cover both upper regions—the sensors coordinate to maximize coverage of the targets. Figure 6 plots the UAV motion plans at the moment the planner decides to chase the downward-bound target. A large black X marks the spot from which the green sensor expects to first see the black target. Generally, the longer the planning horizon, the earlier the UAVs react to the downward-bound target, and the less time any target remains unseen by a sensor.

Unlike the previous scenarios, this scenario features random target motion as well as random measurement noise. This allows a broader comparison of performance among different planning algorithms. Figure 8 shows a plot of the empirical cumulative distribution function (CDF) of the average tracking performance of six algorithms: $H = 1$ with no ECTG term, MWTP(1), MWTP(3), MWTP(4), MWTP(5), and MWTP(6). The plot shows that use of the approximate ECTG produces substantially better performance. Without it, one of the targets (usually the downward-bound one) is ignored when it becomes occluded. There is no statistically significant difference among the performance curves for MWTP(1) to MWTP(6). The main observation here is simply that in all cases MWTP significantly outperforms the pure myopic policy lacking ECTG.

## References


[1] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, pp. 16–26, Sept. 2006.

[2] R. A. Wise and R. T. Rysdyk, "UAV coordination for autonomous target tracking," in *Proc. AIAA Guidance, Navigation, and Control Conference*, Aug. 2006.

[3] S. Musick, "Defense applications," in *Foundations and Applications of Sensor Management* (A. Hero, D. Castanon, D. Cochran, and K. Kastella, eds.), ch. 11, Springer, 2007.

[4] A. O. Hero, D. Castañon, D. Cochran, and K. Kastella, eds., *Foundations and Applications of Sensor Management*. Springer, 2008.

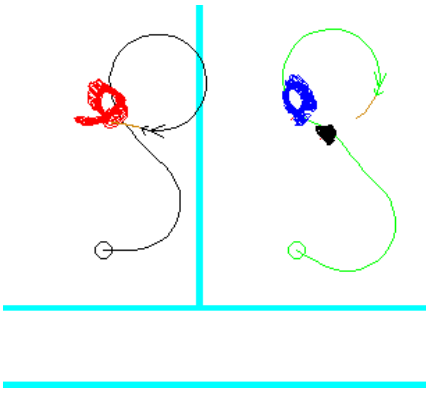

Fig. 5. Beginning of scenario: sensors cover separate regions

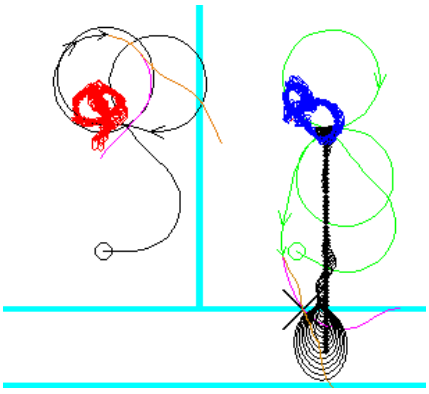

Fig. 6. Transition: sensors coordinate plans to cover all targets as one target moves

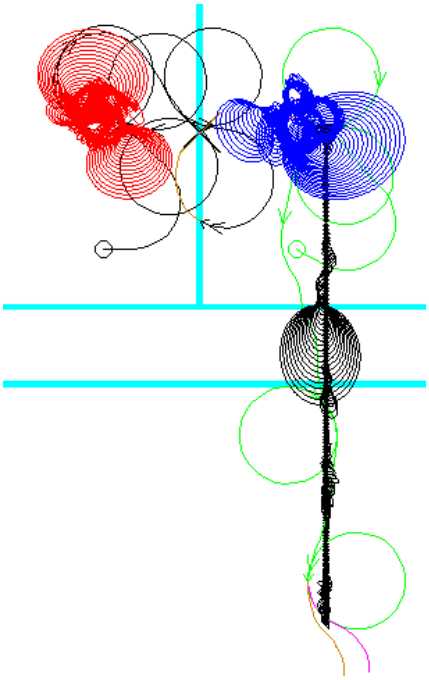

Fig. 7. End of scenario: sensors have coordinated for maximum coverage

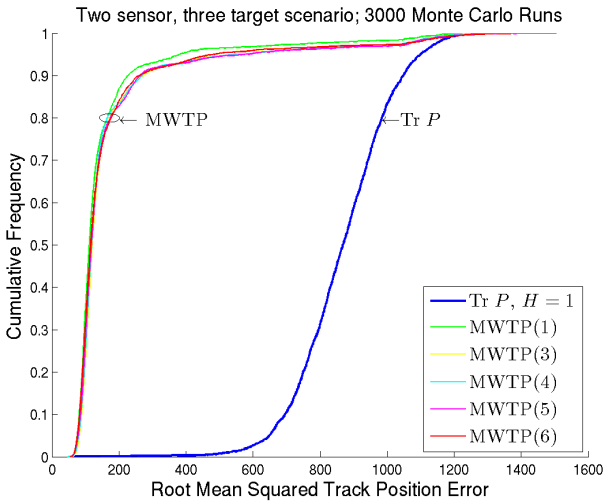


Fig. 8. CDF of tracking performance in multi-sensor scenario


[5] S. Miranda, C. Baker, K. Woodbridge, and H. Griffiths, "Knowledge-based resource management for multifunction radar," *IEEE Signal Processing Magazine*, pp. 66–76, Jan. 2006.

[6] W. W. Schmaedeke and K. D. Kastella, "Sensor management using discrimination gain and interacting multiple model kalman filters," *Proc. SPIE: Signal and Data Processing of Small Targets*, vol. 3373, pp. 390–401, 1998.

[7] B. Grocholsky, H. Durrant-Whyte, and P. Gibbens, "An information-theoretic approach to decentralized control of multiple autonomous flight vehicles," *Proc. SPIE: Sensor Fusion and Decentralized Control in Robotic Systems III*, vol. 4196, pp. 348–359, Oct. 2000.

[8] C. M. Kreucher, A. O. Hero, K. D. Kastella, and M. R. Morelande, "An information based approach to sensor management in large dynamic networks," *Proc. of the IEEE*, vol. 95, no. 5, pp. 978–999, May 2007.

[9] D. J. Klein and K. A. Morgansen, "Controlled collective motion for trajectory tracking," in *Proc. of the American Control Conference*, (Minneapolis, MN), pp. 5269–5275, 2006.

[10] J. Lee, R. Huang, A. Vaughn, X. Xiao, J. K. Hedrick, M. Zennaro, and R. Sengupta, "Strategies of path-planning for a UAV to track a ground vehicle," in *Proc. Autonomous Intelligent Networks and Systems (AINS)*, (Menlo Park, CA), June 2003.

[11] A. Ryan, J. Tisdale, M. Godwin, D. Goatta, D. Nguyen, S. Spry, R. Sengupta, and K. Hedrick, "Decentralized control of unmanned aerial vehicle sensing missions," in *Proc. of the American Control Conference*, (New York), pp. 4672–4677, 2007.

[12] D. J. Klein, C. Matlack, and K. A. Morgansen, "Cooperative target tracking using oscillator models in three dimensions," in *Proc. of the American Control Conference*, (New York), pp. 2569–2575, 2007.

[13] A. Ryan, H. Durrant-Whyte, and K. Hedrick, "Information-theoretic sensor motion control for distributed estimation," in *Proc. ASME International Mechanical Engineering Conference and Exhibition*, (Seattle), 2007.

[14] E. K. P. Chong, C. Kreucher, and A. O. Hero, "Partially observable Markov decision process approximations for adaptive sensing," *Discrete Event Dynamic Systems: Theory and Applications*, to appear.

[15] Y. He and E. K. P. Chong, "Sensor scheduling for target tracking: A monte carlo sampling approach," *Digital Signal Processing*, vol. 16, no. 5, pp. 533–545, Sept. 2006.

[16] Y. Li, L. W. Krakow, E. K. P. Chong, and K. N. Groom, "Approximate stochastic dynamic programming for sensor scheduling to track multiple targets," *Digital Signal Processing*, to appear.

[17] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. I and II. Belmont, MA: Athena Scientific, 1995.

[18] S. A. Miller, Z. A. Harris, and E. K. P. Chong, "A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking," *EURASIP Journal on Applied Signal Processing*, special issue on *Signal Processing Advances in Robots and Autonomy*, 2009.