2012 American Control Conference
Fairmont Queen Elizabeth, Montréal, Canada
June 27-June 29, 2012

Dynamic UAV Path Planning for Multitarget Tracking

Shankarachary Ragi, *Student Member, IEEE* and Edwin K. P. Chong, *Fellow, IEEE*

Abstract— **We design a path-planning algorithm to guide unmanned aerial vehicles (UAVs) for tracking multiple ground targets based on the theory of partially observable Markov decision processes (POMDPs). We demonstrate the power and flexibility of the POMDP framework by showing that a variety of features of interest are easy to incorporate into the framework by plugging in the appropriate models. Specifically, in this paper we show how to incorporate the following features by appropriately formulating the POMDP action space, transition law, and objective function: 1) control UAVs with both forward acceleration and bank angle subject to constraints; 2) account for the effect of wind disturbance on UAVs; and 3) mitigate track swaps.**

I. Introduction

Unmanned aerial vehicles (UAVs) are becoming increasingly important in both civilian and military applications. There has been a growing interest in the design of guidance algorithms [1]–[3] for autonomous UAVs because of a wide range of applications: surveillance, tracking, reconnaissance, ocean exploration, and forest fire monitoring. With this motivation, we design a path-planning algorithm to guide autonomous UAVs for tracking multiple ground targets based on the theory of *partially observable Markov decision processes* (POMDPs) [4], [5]. The algorithm collects measurements from the sensors (mounted on the UAVs), constructs the tracks, and computes the control commands for the UAVs.

In practice, POMDP problems are intractable to solve exactly, so we use an approximation method called the *nominal belief-state optimization* (NBO) [4] (explained later). We build upon the work in [4] by: controlling UAVs with both forward acceleration and bank angle subject to constraints, accounting for wind disturbance on UAVs, and mitigating track swaps. We address these issues as explained below:

- In our study, we use variable-speed UAVs in contrast to [4], which focuses on fixed-speed UAVs. We control the speed and heading direction of a UAV by controlling the forward acceleration and bank angle. These control commands are subject to certain constraints. We incorporate these controls into the framework by appropriately formulating the POMDP action space and state-transition law. We use the kinematic equations presented in [3] to define the state-transition law.
- Wind speeds are often comparable to UAV cruise speeds. As a result, the trajectory of a UAV gets deviated from its desired course (due to wind) leading to tracking performance degradation. We address this problem by appropriately formulating the POMDP state-transition law. More precisely, we incorporate the speed and direction of wind into the UAV kinematic model. Although there are many approaches in the literature for wind compensation (e.g., [6]–[9]), our work shows how to account for wind in the context of our POMDP framework.
- A *track swap* is an undesirable event that causes the swapping of target identities. The likelihood of a track swap depends on the location of the UAVs and the tracker state. Therefore, track swaps can be mitigated by controlling the UAVs appropriately. We accomplish this by incorporating a metric into the objective function. Although this enhancement was introduced in [4], our work builds on it by considering multiple candidate metrics and providing extensive experimental results not available before.

The contribution of this paper is to show how a variety of features of interest in UAV guidance for target tracking can be incorporated naturally into the POMDP framework by plugging in the appropriate components into the framework.

II. Problem Specification

Our guidance problem is specified as follows:

- *2-D Motion*: The targets are assumed to be moving in a plane on the ground. We use a simplified UAV motion model, where the altitude of a UAV is assumed to be constant. The position coordinates of each UAV, i.e., (x, y) are varied by applying UAV controls: forward acceleration and bank angle.
- *Noisy Observations*: The UAVs are mounted with sensors that generate the position measurements of the targets. These measurements have random errors that are spatially varying, i.e., the measurement error covariance of a target depends on the locations of sensors (UAVs) and the target.
- *UAV Controls*: A UAV is controlled by: forward acceleration (controls the speed) and bank angle (controls the heading angle). The values of these control variables are restricted to lie within certain minimum and maximum limits.
- *Tracking Objectives*: The objective is to minimize the mean-squared error between the tracks and the targets. In Section V, we modify the tracking objective to avoid track swaps.

This work was supported in part by Northrop Grumman Corporation through the Rocky Mountain Aerospace Technology Incubator (RMATI) program and by AFOSR contract FA9550-09-1-0518.

The authors are with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373, USA. Emails: {shankar.ragi, edwin.chong}@colostate.edu

978-1-4577-1094-0/12/$26.00 ©2012 AACC

III. POMDP Formulation and the NBO Approximation Method

A POMDP is a controlled dynamical process in discrete time useful for modeling resource control problems [10] (e.g., our guidance problem). In the following sub-section, we define the POMDP ingredients in terms of our guidance problem.

A. POMDP Ingredients

- *States*. The POMDP states represent those features in the system that possibly evolve over time. In our guidance problem, we define three sub-systems (same as in [4]): the sensors, the targets, and the tracker. Therefore, the state at time k is given by $x_k = (s_k, \chi_k, \xi_k, \mathbf{P}_k)$, where s_k represents the sensor state, χ_k represents the target state, and $(\xi_k, \mathbf{P}_k)$ represents the tracker state. The sensor and the target states include the locations and velocities of the UAVs and the targets respectively. The tracker state is a standard in the Kalman filter, where ξ_k is the posterior mean vector and $\mathbf{P}_k$ is the posterior covariance matrix.
- *Actions*. The actions are the controllable aspects of the system. In our problem, the actions include the forward accelerations and the bank angles of all the UAVs. More precisely, the action at time k is given by $u_k = [a_k\,; \phi_k]$, where a_k and ϕ_k are the vectors containing the forward accelerations and the bank angles respectively for all the UAVs. This is in contrast to [4], which focuses on fixed-speed UAVs.
- *State-Transition Law*. The state-transition law specifies the next-state distribution given an action taken at a current state. Since we defined three sub-systems, it is convenient to define the state-transition law for each sub-system separately. The sensor state evolves according to $s_{k+1} = \psi(s_k, u_k)$, where ψ is a mapping function (defined later). The target state evolves according to $\chi_{k+1} = f(\chi_k) + v_k$, where v_k represents an i.i.d. random sequence and f represents the target motion model. Finally, the tracker state evolves according to the Kalman filter equations with a data association technique called *joint probabilistic data association* (JPDA) [11], [12].
- *Observations and Observation Law*. The POMDP states are not fully observable; only a random observation of the underlying state is available at any given time. Let χ_k^{pos} and s_k^{pos} be the position vectors of a target and a sensor/UAV respectively. Then the observation of the target's position is given by:
$$z_k^{\chi} = \chi_k^{\text{pos}} + w_k, \qquad (1)$$
where w_k represents a random measurement error whose distribution depends on the locations of the UAV (s_k^{pos}) and the target (χ_k^{pos}). The sensor and the tracker states are assumed to be fully observable.
- *Cost Function*. The cost function specifies the cost of taking an action in a given state. We use the mean-squared error between the tracks and the targets as the cost function (same as in [4]):
$$C(x_k, u_k) = \mathrm{E}_{v_k, w_{k+1}}\left[||\chi_{k+1} - \xi_{k+1}||^2 \,\middle|\, x_k, u_k \right]. \qquad (2)$$
- *Belief State*. The belief state is the posterior distribution of the underlying state, which is updated incrementally using Bayes rule given the observations. The belief state at time k is given by: $b_k = (b_k^s, b_k^{\chi}, b_k^{\xi}, b_k^{\mathbf{P}})$, where $b_k^s = \delta(s - s_k)$, $b_k^{\xi} = \delta(\xi - \xi_k)$, $b_k^{\mathbf{P}} = \delta(\mathbf{P} - \mathbf{P}_k)$ (since the sensor and the tracker states are fully observable), and b_k^{χ} is the posterior distribution of the target state.

B. Optimal Policy

Given the POMDP formulation, our objective is to choose actions over a time horizon $H : k = 0, 1, \ldots, H-1$, such that the expected cumulative cost is minimized. The expected cumulative cost over the time horizon H can be written as:

$$J_H = \mathrm{E}\left[\sum_{k=0}^{H-1} C(x_k, u_k)\right]. \qquad (3)$$

The action chosen at time k should be allowed to depend on the history of all observable quantities till time $k-1$. If an optimal choice of such actions exist, then there exists an optimal sequence of actions that depends only on "belief-state feedback" [5]. Therefore, the objective function can be written in terms of the belief states as follows:

$$J_H = \mathrm{E}\left[\sum_{k=0}^{H-1} c(b_k, u_k) \,\middle|\, b_0\right], \qquad (4)$$

where $c(b_k, u_k) = \int C(x, u_k) b_k(x)\,\mathrm{d}x$.

According to the celebrated Bellman's principle of optimality [13], the optimal objective function value J_H^* given the current belief state b_0 can be written as follows:

$$J_H^*(b_0) = \min_u \left\{ c(b_0, u) + \mathrm{E}\left[J_{H-1}^*(b_1) \,\middle|\, b_0, u \right] \right\}, \qquad (5)$$

where b_1 is the random next belief state, J_{H-1}^* is the optimal cumulative cost over the horizon $H-1 : k = 1, 2, \ldots, H-1$, and $\mathrm{E}[.|b_0, u]$ is the conditional expectation given the current belief state b_0 and an action u taken at time $k = 0$. Let us define the Q-value of taking an action u given the current belief state b_0:

$$Q_H(b_0, u) = c(b_0, u) + \mathrm{E}\left[J_{H-1}^*(b_1) \,\middle|\, b_0, u \right]. \qquad (6)$$

The optimal policy (from Bellman's principle) at time $k = 0$ can be written as:

$$\pi_0^*(b_0) = \arg\min_u Q_H(b_0, u). \qquad (7)$$

The optimal policy at time k is given by:

$$\pi_k^*(b_k) = \arg\min_u Q_{H-k}(b_k, u). \qquad (8)$$

In practice, the second term in the Q-function (6) is hard to obtain exactly. A number of methods have been studied in the literature [5] to approximate the Q-values. We use one such approximation method called the *nominal belief-state optimization* (NBO), which was introduced in [4] along with other approximations and techniques specific to our guidance problem.

C. NBO Approximation Method

We use a linearized target motion model with zero-mean noise, as given below:

$$\chi_{k+1} = \mathbf{F}_k \chi_k + v_k,\ v_k \sim \mathcal{N}(0, \mathbf{Q}_k), \tag{9}$$

and the observations are given by:

$$z_k^{\chi} = \mathbf{H}_k \chi_k + w_k,\ w_k \sim \mathcal{N}(0, \mathbf{R}_k(\chi_k, s_k)), \tag{10}$$

where $\mathbf{F}_k$ is the target motion model and $\mathbf{H}_k$ is the observation model. Since we assumed Gaussian distributions, the target belief state can be expressed (or approximated) as: $b_k^{\chi}(\chi) = \mathcal{N}(\chi - \xi_k, \mathbf{P}_k)$, where ξ_k and $\mathbf{P}_k$ evolve according to the JPDA algorithm [11], [12].

According to the NBO method, the objective function can be approximated as follows:

$$J_H(b_0) \approx \sum_{k=0}^{H-1} c(\hat{b}_k, u_k), \tag{11}$$

where $\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_{H-1}$ is a *nominal* belief-state sequence and the optimization is over an action sequence $u_0, u_1, \ldots, u_{H-1}$. The nominal target belief-state sequence can be identified with the nominal tracks $\left(\hat{\xi}_k, \hat{\mathbf{P}}_k\right)$, which are obtained from the Kalman filter equations with exactly zero-noise (*nominal* noise) sequence as follows:

$$\begin{aligned}
&\hat{b}_k^{\chi}(\chi) = \mathcal{N}\left(\chi - \hat{\xi}_k, \hat{\mathbf{P}}_k\right), \\
&\hat{\xi}_{k+1} = \mathbf{F}_k \hat{\xi}_k, \\
&\hat{\mathbf{P}}_{k+1} = \left[\hat{\mathbf{P}}_{k+1|k}^{-1} + \mathbf{H}_{k+1}^{\mathrm{T}} \left[\mathbf{R}_{k+1}\left(\hat{\xi}_{k+1}, s_{k+1}\right)\right]^{-1} \mathbf{H}_{k+1}\right]^{-1},
\end{aligned} \tag{12}$$

where $\hat{\mathbf{P}}_{k+1|k} = \mathbf{F}_k \hat{\mathbf{P}}_k \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$ and $s_{k+1} = \psi(s_k, u_k)$ (ψ is defined in the next sub-section). The cost function, i.e., the mean-squared error between the tracks and the targets can be written as:

$$c(\hat{b}_k, u_k) = \operatorname{Tr} \hat{\mathbf{P}}_{k+1}. \tag{13}$$

Therefore, the objective is to find an action sequence $(u_0, u_1, \ldots, u_{H-1})$ that minimizes the cumulative cost function (truncated horizon [4]):

$$J_H(b_0) = \sum_{k=0}^{H-1} \operatorname{Tr} \hat{\mathbf{P}}_{k+1} = \sum_{k=0}^{H-1} \left(\sum_{i=1}^{N_{\text{targs}}} \operatorname{Tr} \hat{\mathbf{P}}_{k+1}^{i} \right), \tag{14}$$

where N_{targs} represents the number of targets. When there are multiple UAVs, the nominal covariance matrix for ith target (based on data fusion techniques) at time $k+1$ is expressed as follows:

$$\hat{\mathbf{P}}_{k+1}^{i} = \left[\sum_{j=1}^{N_{\text{sens}}} \left(\hat{\mathbf{P}}_{k+1}^{i,j} \right)^{-1} \right]^{-1}, \tag{15}$$

where N_{sens} represents the number of sensors/UAVs and $\hat{\mathbf{P}}_{k+1}^{i,j}$ is the nominal covariance matrix of the ith target computed at the jth sensor. See [4] for a detailed description of the NBO method.

D. UAV Kinematics

In this sub-section, we define the mapping function ψ introduced in Section III to describe the evolution of the sensor (UAV) state given an action, i.e., $s_{k+1} = \psi(s_k, u_k)$. The state of the ith UAV at time k is given by: $s_k^i = \left(p_k^i, q_k^i, V_k^i, \theta_k^i\right)$, where (p_k^i, q_k^i) represents the position coordinates, V_k^i represents the speed, and θ_k^i represents the heading angle. Let a_k^i be the forward acceleration (control variable) and ϕ_k^i be the bank angle (control variable) of the UAV. The mapping function ψ can be specified as a collection of simple kinematic equations that govern the UAV motion. The kinematic equations of the UAV motion [3] are as follows:

1) speed update:

$$V_{k+1}^i = \left[V_k^i + a_k^i T\right]_{V_{\min}}^{V_{\max}}, \tag{16}$$

where $[v]_{V_{\min}}^{V_{\max}} = \max\{V_{\min}, \min(V_{\max}, v)\}$,

2) heading angle update:

$$\theta_{k+1}^i = \theta_k^i + (gT \tan(\phi_k^i)/V_k^i), \tag{17}$$

3) location update:

$$\begin{aligned}
p_{k+1}^i &= p_k^i + V_k^i T \cos(\theta_k^i), \\
q_{k+1}^i &= q_k^i + V_k^i T \sin(\theta_k^i),
\end{aligned} \tag{18}$$

where $V_{\min}$ and $V_{\max}$ are the minimum and the maximum limits on the speed of the UAVs, g is the acceleration due to gravity, and T is the length of the time-step.

E. Empirical Study of NBO

We implement the NBO method in MATLAB, where the optimization problem discussed in Subsection III-C is solved using the command *fmincon* (gradient-based search algorithm). The measurement error i.e., w_k in (1) is distributed according to the normal distribution $\mathcal{N}(0, \mathbf{R}_k(\chi_k, s_k))$, where $\mathbf{R}_k$ reflects 10% range uncertainty and 0.01π radian angular uncertainty. In all our simulations, we set the time horizon i.e., H in (14) to 6.

We simulate a few scenarios to demonstrate the coordinative behavior among the UAVs and the ability of the algorithm to track multiple targets with a single UAV. These scenarios are similar to the scenarios in [4]. The scenarios in [4] were simulated with fixed-speed UAVs, whereas we simulate our scenarios with variable-speed UAVs. In our simulations, the trajectory of a target is represented by a sequence of red dots and the trajectory of a UAV is represented by a curve joining the arrows (size of an arrow is proportional to the instantaneous speed of the UAV) that point toward the heading direction of the UAV.

1) Coordinated UAV Motion: We simulate a scenario with three UAVs and two targets as shown in Figure 1, which depicts the scenario at the end of the simulation. Both targets start at the bottom, and as the simulation progresses, one target moves towards the north-east and the other target moves towards the north-west. In all our simulations, the targets move at a constant speed. The UAVs start at the bottom and move according to the kinematic equations in

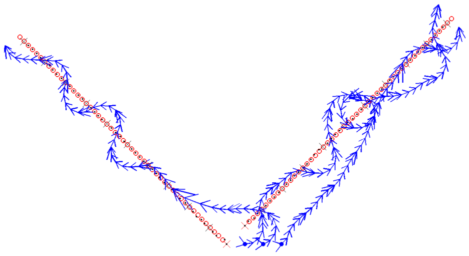


Fig. 1. Three UAVs (blue) tracking two targets (red).

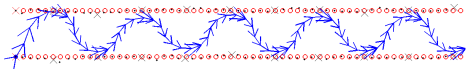


Fig. 2. One UAV (blue) tracking two targets (red).

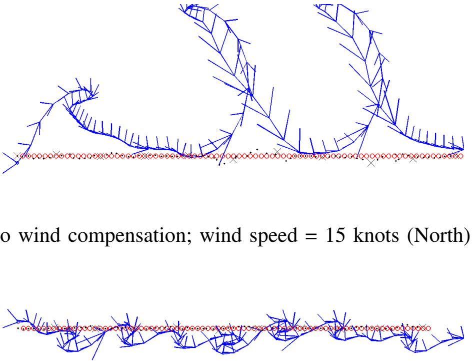
Subsection III-D, with controls obtained from the command *fmincon*, which minimize the cumulative cost function (14). The UAVs coordinate to maximize the coverage of the targets as shown in Figure 1.

2) Weaving Between Targets: We simulate a scenario with one UAV and two targets as shown in Figure 2, which depicts the scenario at the end of the simulation. Both targets start from the left and move towards the right with constant speed. The UAV weaves between the targets as shown in Figure 2, so that the average speed of the UAV towards the right is close to the speed of the targets. Also, the tracks get refined by looking at the targets from different angles, which is achieved by weaving between the targets. This scenario demonstrates the ability of the planning algorithm to maximize the coverage of multiple targets with a single UAV.

IV. Wind Compensation

If unaccounted for, wind drags the UAVs from their planned paths, which results in tracking performance degradation. In this section, we present a wind compensation method in the context of our POMDP framework. More precisely, we incorporate the speed and direction of wind into the framework to nullify the effect of wind on UAVs.

The NBO method requires a nominal belief-state sequence of future states of the targets and the sensors (UAVs) given the actions. Since our planning algorithm has a *lookahead* property, it is important to have a UAV motion model that correctly predicts the trajectory of each UAV. We simulate a scenario with one UAV and one target in the presence of wind as shown in Figure 3(a). We use a simple wind model, i.e., wind blowing in the north direction at a constant speed. Figure 3(a) shows a plot at the end of the simulation where wind was unaccounted for, which resulted in drifting of the UAV from its desired course. As a result, the tracking performance (measured in average target-location error) gets deteriorated. To nullify the effect of wind, we incorporate the speed and direction of wind into the state-transition law



(a) No wind compensation; wind speed = 15 knots (North).

(b) Wind compensation; wind speed = 15 knots (North).

Fig. 3. UAV (blue) tracking a target (red) in the presence of wind.

corresponding to the sensor sub-system, i.e., the kinematics of the UAV motion. We assume that the information on wind (i.e., the speed and direction of wind) is available. The modified location update equations are given below:

$$\begin{aligned} p_{k+1}^i &= p_k^i + V_k^i T \cos(\theta_k^i) + w_x T, \\ q_{k+1}^i &= q_k^i + V_k^i T \sin(\theta_k^i) + w_y T, \end{aligned} \tag{19}$$

where w_x and w_y are the average wind speeds in x and y directions. The speed and bank angle update equations remain same as in (16) and (17). This wind compensation method is not limited to constant wind scenarios (as in our example); we can incorporate any wind model into the framework.

Figure 3(b) shows the simulation of the same scenario as in Figure 3(a), but with modified UAV kinematic model (19). With wind compensation, the UAV trajectory stays close to that of the target's, which improves the tracking performance. We run this simulation (with and without wind compensation) for 2000 runs and calculate the average target-location error at the end of every run. The advantage of having wind compensation can be seen in Figure 4, where the cumulative frequency of average target-location errors is shown for various wind speeds. From Figure 4, it is evident that the tracking performance degrades with increasing wind speed when wind is not compensated, and with wind compensation the tracking performance remains good even with increasing wind speed.

V. Track Swap Avoidance

A track swap is a switch in the association between the tracks and the targets. The identities of the targets are interpreted through the association between the tracks and the targets. Therefore, a track swap switches the identities of the targets, which is undesirable. The likelihood of a track swap depends on the tracker state, which in-turn depends on the location of the UAVs relative to the targets. Therefore, we can mitigate track swaps by appropriately controlling the UAVs. This is achieved by incorporating a term into the objective function that represents the risk of a track swap. Although this enhancement was introduced in [4], our work builds on

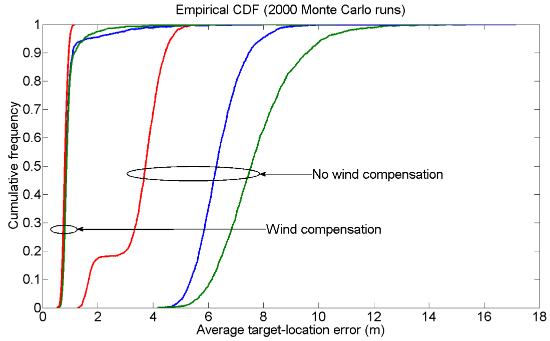


Fig. 4. Tracking performance under various wind speeds: 10 knots (red), 15 knots (blue), 20 knots (green).

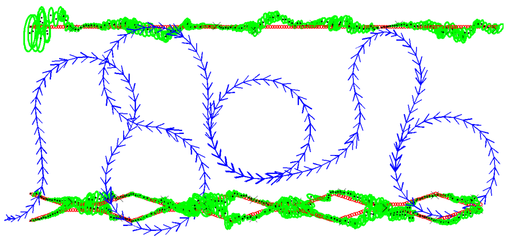


Fig. 5. UAV tracking three targets; UAV (blue), targets (red), error concentration ellipses of target's location (green).

it by incorporating multiple candidate metrics and comparing their performance for variable-speed UAVs via Monte Carlo simulations.

A. Problem Description

Suppose that a UAV is tracking three targets as shown in Figure 5. In this scenario, the bottom two targets come in close proximity to each other periodically and the topmost target remains far from the bottom two targets. Our planning algorithm maximizes the coverage of the targets by guiding the UAV to weave between the topmost target and the bottom targets (as shown in Figure 5), which is achieved by minimizing the overall trace objective. The likelihood of a track swap is high when the measurement sources are ambiguous. In the scenario of Figure 5, when the UAV is far from the bottom targets, i.e., when the UAV is close to the topmost target, the likelihood of a track swap (corresponding to the bottom targets) is high because the chance that the sources of the measurements from the bottom two targets becoming ambiguous is high. In the following sub-section, we present an enhancement [4] to mitigate track swaps.

B. Enhancement for Mitigating Track Swaps

The similarity between the target state distributions is a good predictor for a track swap because the likelihood of the measurement sources becoming ambiguous is high when the similarity between the target state distributions is high. The target state distributions depend on the tracker states and the locations of the UAVs over time. Therefore, we can control the UAVs appropriately such that the target state distributions are less similar. The similarity between the probability distributions can be measured as the inverse of a statistical distance between the distributions. To minimize the likelihood of a track swap, we incorporate a term that is inversely proportional to a statistical distance (between the target state distributions) into the objective function. The new objective function is written as

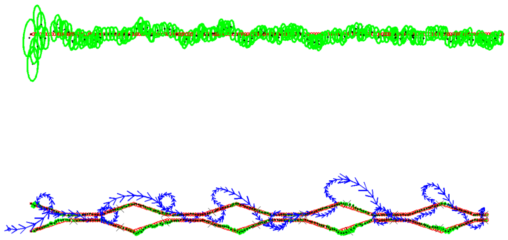


Fig. 6. UAV (blue) tracking three targets (red) while mitigating track swaps.

$$J_H = \sum_k \left(\sum_{i=1}^{N_{\text{targs}}} \operatorname{Tr} \mathbf{P}_{k+1}^i + \gamma(1/D_{k+1}) \right), \qquad (20)$$

where γ is a scaling factor, N_{targs} represents the number of targets, and

$$D_{k+1} = \min_{p \neq q;\, p,q \in \mathcal{S}} D(\chi_{k+1}^p || \chi_{k+1}^q),$$

where $\mathcal{S} = \{1, 2, \ldots, N_{\text{targs}}\}$ and $D(\chi_{k+1}^p || \chi_{k+1}^q)$ is a statistical distance between the distributions of the pth target and the qth target.

There are several statistical distances defined in the literature: KL-divergence, Bhattacharya distance, Hellinger distance, and worst-case chi-square distance (see [4] for a description of the worst-case chi-square distance). We simulate a scenario with three targets and one UAV (similar to the scenario in Figure 5) with the new objective function (20) as shown in Figure 6, where the UAV stays close to the bottom two targets in contrast to the behavior in Figure 5. This reduces the similarity between the probability distributions of the bottom targets, which in-turn reduces the chance of a track swap.

KL-divergence, Bhattacharya distance, and Hellinger distance are average-case measures of how often the state values from the distributions fall in a small neighborhood of each other, whereas worst-case chi-square distance is a worst-case measure. The authors of [4] explain that a worst-case measure is more suited to track swap prediction than an average-case measure. In the next sub-section, we provide an empirical study to show that the worst-case chi-square distance is an appropriate metric in predicting track swaps, corroborating the claim in [4].

C. Empirical Study

We simulate 2000 runs of the scenario where a UAV tracks three targets (as described in Subsection V-A) with

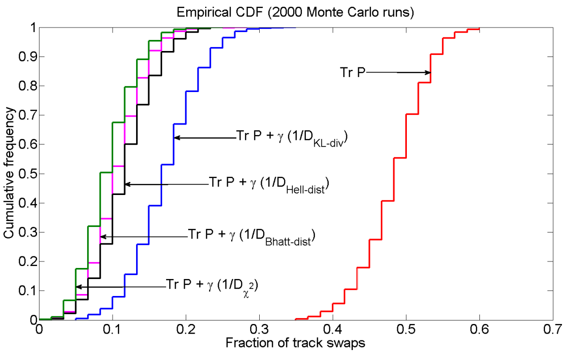


Fig. 7. Performance comparison for various statistical distances.

and without the enhancement in the objective function. In every run, we calculate the fraction of track swaps—a metric to evaluate the tracking performance—according to the following method.

Fraction of Track Swaps: Let N_{TS} represent the number of track swaps, defined as follows. We check for a track swap only at instances when the bottom targets are the farthest apart. These instances occur periodically as can be seen in Figure 5 and Figure 6. At each of these instances, we evaluate the associations between the tracks and the targets. Whenever the association at a particular instance differs from the association in the previous instance, we increment N_{TS} by one. At the end of the simulation, we evaluate the fraction of track swaps as follows:

$$\text{fraction of track swaps} = \frac{N_{\text{TS}}}{N}, \qquad (21)$$

where N represents the total number of instances we evaluate the track associations.

In our simulations, we use the following notation to represent the statistical distances: 1) KL-divergence: $D_{\text{KL-div}}$, 2) Hellinger distance: $D_{\text{Hell-dist}}$, 3) Bhattacharya distance: $D_{\text{Bhatt-dist}}$, and 4) worst-case chi-square distance: D_{χ^2}. We plot the cumulative frequency of the fraction of track swaps for various statistical distances in Figure 7. From Figure 7, it is evident that the enhancement to the objective function improved the performance (in terms of the fraction of track swaps) significantly for all candidate statistical distances. It is also evident that D_{χ^2} is the best among other statistical distances in mitigating track swaps. This shows that the worst-case chi-square distance is an appropriate metric for predicting track swaps—corroborating the claim in [4].

VI. Conclusions

We designed a path-planning algorithm to guide UAVs for multitarget tracking based on the theory of POMDPs. Our main contribution in this paper is summarized as follows: 1) In our study, we used variable-speed UAVs (in contrast to [4], which focused on fixed-speed UAVs), which are controlled by forward acceleration and bank angle. We incorporated these UAV controls into the framework by appropriately formulating the POMDP action space and state-transition law. 2) We incorporated the speed and direction of wind into the UAV kinematic model to nullify the effect of wind on UAVs. The results from the Monte Carlo simulations show that the tracking performance remained good even with increasing wind speed when wind was accounted for, whereas the tracking performance deteriorated significantly when wind was unaccounted for. 3) To reduce the likelihood of a track swap, we included a term that is proportional to the inverse of the statistical distance (between the target state distributions) in the objective function. Our simulation of the scenario with one UAV and three targets shows that the modified objective function successfully enabled the UAV to track the targets while reducing the similarity between the target state distributions, which effectively reduced the chance of a track swap. We compared the performance of various statistical distances, and our results support the argument that D_{χ^2} is an appropriate metric in predicting track swaps. We successfully demonstrated the power and flexibility of the POMDP framework by showing that it is easy to incorporate various features of interest into the framework.

References

[1] Y. J. Zhao and Y. C. Qi, “Minimum fuel powered dynamic soaring of unmanned aerial vehicles utilizing wind gradients,” *Optim. Control Appl. Meth.*, vol. 25, pp. 211–233, 2010.

[2] J. R. Azinheira and A. Moutinho, “Hover control of an UAV with backstepping design including input saturations,” *IEEE Trans. Control Syst. Technol.*, vol. 16, pp. 517–526, 2008.

[3] B. R. Geiger, J. F. Horn, A. M. DeLullo, and L. N. Long, “Optimal path planning of UAVs using direct collocation with nonlinear programming,” in *Proc. of AIAA GNC Conference*, Aug. 2006.

[4] 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 Advances in Signal Processing*, vol. 2009, 2009.

[5] E. K. P. Chong, C. Kreucher, and A. O. Hero, “Partially observable markov decision process approximations for adaptive sensing,” *Discrete Event Dyn. Syst.*, vol. 19, pp. 377–422, 2009.

[6] A. L. Jennings, R. Ordonez, and N. Ceccarelli, “Dynamic programming applied to UAV way point path planning in wind,” in *Proc. IEEE Multi-conference on Systems and Control*, San Antonio, Texas, USA, Sep. 2008, pp. 215–220.

[7] T. G. McGee, S. Spry, and J. K. Hedrick, “Optimal path planning in a constant wind with a bounded turning rate,” in *Proc. of the AIAA Guidance, Navigation and Control Conference and Exhibit*, 2005.

[8] N. Ceccarelli, J. J. Enright, E. Frazzoli, S. J. Rasmussen, and C. J. Schumacher, “Micro UAV path planning for reconnaissance in wind,” in *Proc. of American Control Conference*, New York City, USA, Jul. 2007, pp. 5310–5315.

[9] T. G. McGee and J. K. Hedrick, “Path planning and control for multiple point surveillance by an unmanned aircraft in wind,” in *Proc. of American Control Conference*, Minneapolis, USA, Jun. 2006, pp. 4261–4266.

[10] E. K. P. Chong, C. Kreucher, , and A. O. H. III, “POMDP approximation methods based on heuristics and simulation,” in *Foundations and Applications of Sensor Management*, A. O. H. III, D. Castanon, D. Cochran, and K. Kastella, Eds. New York, NY, USA: Springer, 2008, ch. 8, pp. 95–120.

[11] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. London: Academic Press Inc., 1988.

[12] S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. Boston, USA: Artech House, 1999.

[13] R. Bellman, *Dynamic Programming*. Princeton, N. J.: Princeton Univ. Press, 1957.