2017 IEEE Conference on Control Technology and Applications (CCTA)
August 27-30, 2017. Kohala Coast, Hawai'i, USA

Robust UAV Path Planning using POMDP with Limited FOV Sensor

Christopher M. Eaton[1], Edwin K.P. Chong[2] and Anthony A. Maciejewski[3]

Abstract— **Significant development in path planning algorithms for unmanned aerial vehicles (UAVs) has been performed using numerous different methods. One such method, Partially Observable Markov Decision Processes (POMDP), has been used effectively for tracking fixed and moving targets. One limitation of those efforts has been the assumption that the UAVs could always see the targets, with a few unique exceptions, e.g., building obscuration. In reality, there will be times when a vehicle will not be able to observe a target due to constraints such as turn requirements or tracking multiple targets that are not within a single field of view (FOV). The POMDP formulation proposed in this paper is robust enough to handle those missed observations. Monte Carlo runs of 1000 iterations per configuration are run to provide statistical confidence in the performance of the algorithm. UAV altitude and sensor configuration are varied to show robustness across multiple configurations. A sensor with a limited FOV is assumed and changes in fixed look angle are evaluated. Changes in altitude provide results equivalent to changes in sensor window or focal length. Results show that the POMDP algorithm is capable of tracking single and multiple moving targets successfully with limited FOV sensors across a range of conditions.**

I. INTRODUCTION

The growth in use of unmanned systems has resulted in an explosion in control methods. Unmanned aerial vehicles (UAVs) provide the opportunity to perform many of the dull, dirty, and dangerous missions that are ill-suited for manned systems. In order to effectively and efficiently perform these missions, robust algorithms to control these UAVs must be developed. Significant work is being performed in numerous control areas for UAVs [1]. One method of path planning control that has shown significant promise is the use of decentralized partially observable Markov decision processes (POMDP) with a nominal belief-state optimization (NBO) approximation [2].

We design a UAV control method that utilizes POMDPs with NBO approximation utilizing a fixed field of view (FOV) sensor that has limited observation windows. We build upon our previous efforts [2], [3], [4], that have focused on the ability of the POMDP to provide sufficient path planning for a UAV tracking of ground based fixed or moving targets. During initial development, it was assumed that the UAV(s) could always see the targets, giving rise to a restrictive model for the partially observable portion of the POMDP. We incorporate a fixed FOV sensor to determine the performance of the POMDP with limited observations. We vary the fixed look angles and the altitude to determine the performance of the method while tracking one or two targets. The performance is evaluated using the tracking errors and percentage of observations across the configurations.

Previous work [2] has shown the ability of POMDP with NBO to perform effective path planning for tracking fixed and moving targets. Additionally, collision avoidance and wind compensation schemes were also shown to be effective using the same POMDP algorithms [3]. The POMDP with NBO approach uses a receding horizon method that is computationally efficient and provides effective responses to changes in target dynamics or environmental changes. The contribution of this paper is to show that the POMDP with NBO approach provides robust path planning and target tracking with *limited observations* due to the fixed FOV of the sensor. In this paper, we focus on non-evasive moving targets with limited observations.

II. PROBLEM SPECIFICATION

The ground targets move in 2-D. A simplified UAV motion model is used, utilizing forward acceleration and bank angle for control application with constant altitude. The UAVs have fixed FOV sensors mounted underneath the vehicle that provide limited observation measurement. These measurements are corrupted by spatially varying random errors, dependent upon UAV and target locations. The UAV has a limited speed range that can be varied by controlling the forward acceleration. The limited bank angle of the UAV can be adjusted to change the heading, but also impacts the sensor FOV. The UAV altitude can be set for a given scenario, but will maintain a constant altitude during that entire scenario. Changing the UAV altitude is effectively the same as changing the sensor size or FOV. The objective is to minimize the mean-squared error between the tracks and targets.

III. POMDP AND NBO APPROXIMATION

A POMDP is a discrete time controlled dynamical process that is useful for modeling resource control problems. A POMDP can be interpreted as a controlled version of a hidden Markov reward process. The NBO approximation is a method that has been shown to be computationally efficient for guidance optimization.

A. POMDP Ingredients

States: The POMDP states represent time evolving system features. Three subsystems are defined: the sensors, the

[1]Christopher M. Eaton is a Ph.D. student at Colorado State University, Fort Collins, CO 80523 eatonc@rams.colostate.edu

[2]Edwin K. P. Chong is with the Electrical & Computer Engineering Department, Colorado State University, Fort Collins, CO 80523 Edwin.Chong@colostate.edu

[3]Anthony A. Maciejewski is the Chair of the Electrical & Computer Engineering Department, Colorado State University, Fort Collins, CO 80523 aam@colostate.edu

Supported in part by NSF grant CCF-1422658 and by the CSU Information Science and Technology Center (ISTeC).

targets, and the tracker. At time k, the state is given by $x_k = (s_k, \chi_k, \xi_k, \mathbf{P}_k)$, where s_k is the sensor state, χ_k is the target state, and $(\xi_k, \mathbf{P}_k)$ is the state of the tracker. The sensor state provides the velocities and position of the UAV, and the target state provides the velocities, positions, and accelerations of the targets. The tracker state is a standard Kalman filter [5], [6], with ξ_k the posterior mean vector and $\mathbf{P}_k$ the posterior covariance matrix.

Actions: The actions are controls available to the system. The actions for this problem are the forward acceleration and the bank angle of the UAV. Specifically, the action at time k is given by $u_k = (a_k, \phi_k)$, with a_k and ϕ_k containing the forward acceleration and bank angle, respectively, for the UAV.

Observations and Observation Law: The states of a POMDP are not fully observable; only a random observation of the underlying state is available at any given time. Let χ_k^{pos} be the position vectors of a target and s_k^{pos} be the position vectors of a sensor/UAV. The observation of the target's position can be expressed by

$$z_k^{\chi} = \begin{cases} \chi_k^{pos} + w_k, & \text{if target is visible} \\ \text{no measurement}, & \text{otherwise} \end{cases} \quad (1)$$

where w_k is a random measurement error whose distribution is dependent on the UAV (s_k^{pos}) and the target (χ_k^{pos}) locations. Sensor and tracker states are assumed to be fully observable.

State-Transition Law: The state-transition law defines the next-state distribution given a current state and action pair. It is convenient to separately define the state-transition law for each of the three predefined subsystems. The sensor state progresses by $s_{k+1} = \psi(s_k, u_k)$, where ψ is a mapping function (defined later). The target state evolves according to $\chi_{k+1} = f(\chi_k) + \nu_k$,where ν_k is an independent and identically distributed (IID) random noise sequence and f represents the target motion model (also defined later). The tracker state evolves according to the Kalman filter equations with joint probabilistic data association (JPDA) [5], [7]. When target observations are not available, only the prediction step in the Kalman filter is performed, the update equation is not evaluated.

Cost Function: The cost function defines the cost of taking an action in a given state. Our cost function considers the mean-squared error between the tracks and the targets: $C(x_k, u_k) = E_{\nu_k, w_{k+1}}[\|\chi_{k+1} - \xi_{k+1}\|^2 | x_k, u_k]$

Belief State: The belief state is the posterior distribution of the underlying state, which is incrementally updated via 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 $b_k^s = \delta(s - s_k)$, $b_k^{\xi} = \delta(\xi - \xi_k)$, $b_k^{\mathbf{P}} = \delta(\mathbf{P} - \mathbf{P}_k)$ (because the sensor and tracker states are fully observable), and b_k^{χ} is the target state posterior distribution.

B. Optimal Policy

The objective of the optimization policy, given the POMDP formulation, is to choose actions over a time horizon $H, k = 0, 1, ..., H-1$, that minimize the expected cumulative cost. The expected cumulative cost over the time horizon H can be written as $J_H = E[\sum_{k=0}^{H-1} C(x_k, u_k)]$. The action chosen at time k should depend on the history of all observable quantities until time $k-1$. If an optimal choice of actions exists, then there exists an optimal action sequence that depends only on "belief-state feedback" [8]. Therefore, the objective function can be written in terms of the belief states as follows: $J_H = E[\sum_{k=0}^{H-1} c(b_k, u_k)|b_0]$, where $c(b_k, u_k) = \int C(x, u_k) b_k(x) dx$.

According to Bellman's celebrated principle of optimality [9], the optimal objective function value J_H^* given the current belief state b_0 can be written as follows: $J_H^*(b_0) = \min_u \{c(b_0, u) + E[J_{H-1}^*(b_1)|b_0, u]\}$, where b_1 is the random next belief state, J_{H-1}^* is the optimal cumulative cost over the horizon $H-1, k = 1, 2, ..., H-1$, and $E[\cdot|b_0, u]$ is the conditional expectation given the current belief state b_0 and an action u taken at time $k = 0$. The Q-value of taking an action u given the current belief state b_0 is defined by $Q_H(b_0, u) = c(b_0, u) + E[J_{H-1}^*(b_1)|b_0, u]$. The optimal policy (from Bellman's principle) at time $k = 0$ can be written as $\pi_0^*(b_0) = \text{argmin}_u Q_H(b_0, u)$. The optimal policy at time k is $\pi_k^*(b_k) = \text{argmin}_u Q_{H-k}(b_k, u)$.

In practice, the second term in the Q-function is difficult to obtain exactly. Numerous methods have been studied [4], [10], [11], [12], [13], [14], [15], [16] to approximate the Q-values. We use the NBO approximation method, which was introduced in [4] along with other guidance problem approximations and techniques.

C. NBO Approximation

Although there are a number of approximation methods available to solve POMDPs, we chose the NBO method because it is less computationally expensive compared to other POMDP approximation methods like Q-learning, policy rollout, hindsight optimization, and foresight optimization [8]. In practice, a UAV guidance algorithm needs to be implementable in real-time, requiring a method that is not computationally prohibitive. In our previous work [2], we showed that the NBO algorithm provided an acceptable computation quality.

Assume there are N_{targs} targets. We represent the target state as $\chi_k = (\chi_k^1, \chi_k^2, ..., \chi_k^{N_{\text{targs}}})$, where χ_k^i represents the ith target. The track-state is $\xi_k = (\xi_k^1, \xi_k^2, ..., \xi_k^{N_{\text{targs}}})$ and $\mathbf{P}_k = (\mathbf{P}_k^1, ..., \mathbf{P}_k^{N_{\text{targs}}})$, where $(\xi_k^i, \mathbf{P}_k^i)$ is the track-state corresponding to the ith target. We use a linearized target motion model with zero-mean noise to model the target-state dynamics, as given below $(\forall i)$

$$\chi_{k+1}^i = \mathbf{F}_k \chi_k^i + \nu_k^i, \qquad \nu_k^i \sim N(0, \mathbf{Q}_k) \quad (2)$$

with the observations as follows: $z_k^{\chi^i} = \mathbf{H}_k \chi_k^i + w_k^i$ if the target is visible, and no measurement otherwise, where $w_k^i \sim N(0, \mathbf{R}_k(\chi_k^i, s_k))$, $\mathbf{F}_k$ is the target motion model (same for all targets), and $\mathbf{H}_k$ is the observation model (1) (same for every target) according to which only the position of a target is observed. The state of the ith target (χ_k^i)

includes its 2-D position coordinates (x_k, y_k), its velocities (v_k^x, v_k^y) and accelerations (a_k^x, a_k^y) in x- and y-directions, i.e., $\chi_k^i = [x_k, y_k, v_k^x, v_k^y, a_k^x, a_k^y]^T$. Therefore, the observation model is $\mathbf{H}_k = [\mathbf{I}_{2x2}, \mathbf{0}_{4x4}]$. We adopt a constant velocity (CV) model [5], [6] for target dynamics in (2), which defines $\mathbf{F}_k$. The belief state corresponding to the ith target, based upon assumed Gaussian distributions, can be expressed as $b_k^{\chi^i}(\chi) = N(\chi - \xi_k^i, \mathbf{P}_k^i)$, where ξ_k^i and $\mathbf{P}_k^i$ are the track-states of the ith target, which evolve according to the JPDA algorithm [5], [7].

The NBO method approximates the objective function as $J_H(b_0) \approx \sum_{k=0}^{H-1} c(\hat{b}_k, u_k)$, where $\hat{b}_1, \hat{b}_2, ..., \hat{b}_{H-1}$ is a nominal belief-state sequence and the optimization is over the action sequence $u_0, u_1, ..., u_{H-1}$. The nominal belief-state sequence for the ith target can be identified with the nominal tracks $(\hat{\xi}_k^i, \hat{\mathbf{P}}_k^i)$, which are obtained from the Kalman filter equations [31, 32] with exactly zero-noise sequence as follows: $\hat{b}_k^{x^i}(\chi) = N(\chi - \hat{\xi}_k^i, \hat{\mathbf{P}}_k^i)$, $\hat{\xi}_{k+1}^i = \mathbf{F}_k \hat{\xi}_k^i$, and

$$\hat{\mathbf{P}}_{k+1}^i = \begin{cases} [[\hat{\mathbf{P}}_{k+1|k}^i]^{-1} + \mathbf{S}_{k+1}^i]^{-1} & \text{if measurement available} \\ \hat{\mathbf{P}}_{k+1|k}^i & \text{otherwise} \end{cases} \tag{3}$$

where $\hat{\mathbf{P}}_{k+1|k}^i = \mathbf{F}_k \hat{\mathbf{P}}_k^i \mathbf{F}_k^T + \mathbf{Q}_k$, $\mathbf{S}_{k+1}^i = \mathbf{H}_{k+1}^T [\mathbf{R}_{k+1}(\hat{\xi}_{k+1}^i, s_{k+1})]^{-1} \mathbf{H}_{k+1}$ and $s_{k+1} = \psi(s_k, u_k)$ (ψ is defined in the next subsection). In (3), the nominal error covariance matrix $\hat{\mathbf{P}}_{k+1}^i$ is dependent upon the availability of future observations. Because the availability of these observations is uncertain, we can guess by assuming the location of the target at time $k+1$ as $\hat{\xi}_{k+1}^{i,pos}$ (component of nominal track-state corresponding to the ith target at time $k+1$) and checking its line of sight from the sensor location, i.e., s_{k+1}^{pos}. The cost function, i.e., the mean-squared error between the tracks and the targets, can be written as $c(\hat{b}_k, u_k) = \sum_{i=1}^{N_{\text{targs}}} \text{Tr}\hat{\mathbf{P}}_{k+1}^i$. The goal is to find an action sequence $(u_0, u_1, ..., u_{H-1})$ that minimizes the cumulative cost function (truncated horizon [4]) $J_H(b_0) = \sum_{k=0}^{H-1} \sum_{i=1}^{N_{\text{targs}}} \text{Tr}\hat{\mathbf{P}}_{k+1}^i$, where $\hat{\mathbf{P}}_{k+1}^i$ represents the nominal error covariance matrix of the ith target at time $k+1$. Here, we adopt a "receding horizon control" approach, where we optimize the action sequence for $\mathbf{H}$ time-steps from the current time-step but implement only the action corresponding to the current time-step followed by an optimization of the action sequence for $\mathbf{H}$ time-steps in the next time-step.

Our approach is related to model predictive control (MPC), as argued by the authors of [17]. According to [17], the MPC method is a type of rollout algorithm (an approximation method to solve Markov decision processes (MDPs) and POMDPs) with a particular base policy, where the stability property of MPC is a special case of the cost improvement property of rollout algorithms that employ a sequentially improving base policy. In other words, MPC can also be viewed as an approach to solve a POMDP.

The measurement error, i.e., w_k in (1), has a normal distribution $N(0, \mathbf{R}_k(\chi_k, s_k))$, where $\mathbf{R}_k$ reflects $p\%$ range uncertainty and q-rad angular uncertainty. If r_k is the distance between the target and the sensor at time k, then the standard deviations corresponding to the range ($\sigma_{\text{range}}(k)$) and the angle ($\sigma_{\text{angle}}(k)$) are $\sigma_{\text{range}}(k) = (p/100)r_k$ and $\sigma_{\text{angle}}(k) = qr_k$. The information matrix depends on the inverse of the measurement covariance matrix, which depends on the distance between the sensor and the target. Therefore, the information matrix blows up when the UAV is exactly on top of the target (i.e., when $r_k = 0$ the sensor's location overlaps with the target's location in our 2-D environment). To address this problem, we define the effective distance (r_{eff}) between the sensor and the target as follows: $r_{\text{eff}}(k) = \sqrt{r_k^2 + b^2}$, where r_k is the actual distance between the target and the sensor and b is some non-zero real value. Therefore, the standard deviations of the range and the angle are given by $\sigma_{\text{range}}(k) = (p/100)r_{\text{eff}}(k)$ and $\sigma_{\text{angle}}(k) = qr_{\text{eff}}(k)$. If θ_k is the angle between the target and the sensor at time k, then $\mathbf{R}_k$ is calculated as follows:

$$\mathbf{R}_k = M_k \begin{bmatrix} \sigma_{\text{range}}^2(k) & 0 \\ 0 & \sigma_{\text{angle}}^2(k) \end{bmatrix} M_k^T \quad \text{where}$$

$$M_k = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix}.$$

The eigenvalues of the matrix $\mathbf{R}_k$ are therefore $\sigma_{\text{range}}^2(k)$ and $\sigma_{\text{angle}}^2(k)$.

IV. UAV KINEMATICS AND FIXED FOV

Simplified UAV kinematics are used for calculating the UAV motion. With utilization of a fixed FOV sensor, the FOV can be calculated in camera axes. Using Euler angle transformations, it can be determined if the target is within the sensor FOV.

A. UAV Kinematics

In this subsection 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 $s_k^i = (p_k^i, q_k^i, V_k^i, \theta_k^i)$, 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, i.e., $u_k^i = (a_k^i, \phi_k^i)$. 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 [36] are as follows. The speed is updated according to $V_{k+1}^i = [V_k^i + a_k^i T]_{V_{\text{min}}}^{V_{\text{max}}}$, where $[v]_{V_{\text{min}}}^{V_{\text{max}}} = \max\{V_{\text{min}}, \min(V_{\text{max}}, v)\}$, where V_{min} and V_{max} are the minimum and the maximum limits on the speed of the UAVs. The heading angle is updated according to $\theta_{k+1}^i = \theta_k^i + (gT\tan(\phi_k^i)/V_k^i)$, where g is the acceleration due to gravity and T is the length of the time-step. The position coordinate are updated according to $p_{k+1}^i = p_k^i + V_k^i T\cos(\theta_k^i)$ and $q_{k+1}^i = q_k^i + V_k^i T\sin(\theta_k^i)$.

B. Fixed FOV Calculation

Given a sensor width (x_{sens}) and height (y_{sens}), along with the focal length (f) of an installed sensor, a field of view can be calculated for a given altitude (alt). The angular field of view for the sensor width is $\text{FOV}_w = 2\tan^{-1}(x_{\text{sens}}/2f)$ and for the sensor height is $\text{FOV}_h = 2\tan^{-1}(y_{\text{sens}}/2f)$. Given a height above the ground, the edges of the field of view can be calculated in camera axes. It is assumed that the camera is installed concurrent to the body axis, for ease of calculation. The top and bottom edges of the FOV are $x_{FOV} = \pm(\text{alt})\tan(0.5\text{FOV}_h)$ and the left and right edges of the FOV are $y_{FOV} = \pm(\text{alt})\tan(0.5\text{FOV}_w)$, where alt is the altitude of the sensor.

In order to determine if an observation is made, the position of the target must be translated and rotated into the UAV body axis. A vector η is calculated between the UAV position and the target position, $\eta = (p_{\text{UAV}} - p_{\text{target}}, q_{\text{UAV}} - q_{\text{target}}, \text{alt}_{\text{UAV}} - \text{alt}_{\text{target}})$. Given the UAV Euler angles, pitch (θ), roll (ψ), and heading (ϕ) a rotation of η can be made between the world axis and body axis. The value of η in the body axis can now be compared to the FOV for the sensor. If the position of the target, η, is within the field of view, an observation is made and the POMDP will calculate normally with an updated observation, as specified in (1). However, if η is not in the FOV, there will be no observation and no measurement will be passed to the POMDP.

V. TARGET TRACKING RESULTS

Monte Carlo runs were performed on several configurations to evaluate the effectiveness of our method to track the target. The two primary conditions were whether the UAV was tracking one or two targets. Three sensor angle configurations were evaluated: downward, 15 degrees forward, and 15 degrees right. Six altitudes were evaluated for each of the six configurations: 200, 500, 750, 1000, 1500, and 2000 meters. A range uncertainty of 10 percent and an angular uncertainty of 1 percent were used for this evaluation.

The sensor simulated was modeled after several commercially available sensors [18]. The sensor was assumed to have a focal plane width of 5 mm, height of 3 mm, and a lens focal length of 5.3 mm. This configuration provides a field of view of 50.5 degree wide and 31.6 degrees tall. Varying the altitude provides a different size of field of view. Table I provides the total height and width of the sensor field of view at each of the altitudes, based upon a straight lookdown angle. As the angle of the sensor is rotated from straight down, the actual ground plane field of view will change. It was assumed that all targets could be tracked at all altitudes.

TABLE I
SENSOR FIELD OF VIEW

Altitude (m)	FOV Height (m)	FOV Width (m)
200	113	188
500	283	471
750	424	707
1000	566	943
1500	849	1415
2000	1132	1886

The targets moved at a constant speed of 10 m/s at a constant heading. The UAV was allowed to vary speed between 16 and 25 m/s and a maximum bank angle of 30 degrees. In order to provide a consistent analysis baseline, altitude was only considered for variation of sensor field of view; otherwise the problem was treated as a 2-D problem. Tracking errors were not considered as a function of altitude, only of x-y distance variations to enable FOV comparison. Previous efforts have shown that the average location error was 2-3 meters for similar scenarios [3].

Overall, the algorithm showed a robust ability to track moving targets even with a limited FOV that decreased the observations and increased the tracking error. However, for general tracking of vehicles, the method shows the ability to adequately track single and multiple targets for observation purposes. The algorithm also showed computational efficiency. The POMDP calculation for a random run averaged 0.46 seconds on an Intel Core I7-6700, 4 GHz, 64-bit processor with 24 GB of Ram.

A. Single Target Tracking

Single target tracking with limited field of view sensors showed effective tracking of a single moving target. Figure 1 shows an example run of a UAV tracking the moving target. The figure shows when the target was observed (green circles), when it was not observed (red x) and what the predicted location of the vehicle was by the POMDP algorithm (black dot). The line with asterisks shows the general flight path of the UAV with the location recorded at 2 second intervals.

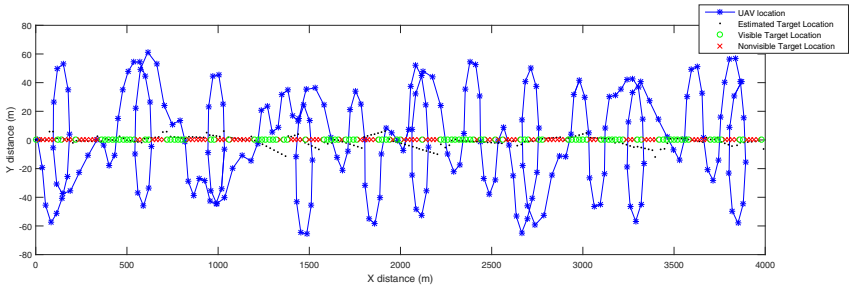

Fig. 1. Tracking Example: 1 Target, 1 Sensor, 200 m, 15 deg fwd sensor

The quality of the tracking can normally be related to the quantity of observations. The greater the observation percentage, the lower the location tracking error will be. This trend can be seen in Figure 2 for downward and side facing sensor configurations. However, it can be noted that the location error increased with increasing observations for the forward facing sensor configuration. The forward facing sensor does not show a decreasing error due to the geometric conditions that result in observations. The decreased tracking accuracy, even under high quality observations, can be related to the geometry of the observations. The observations may be made but the change in observation may not significantly improve the tracking error, resulting in larger errors than other configurations.

In general, the increase in altitude results in an increase in the number of observations and a reduction in the location error. This characteristic relationship is expected, because as the altitude increases, the size of the observation window increases, as shown in Table I. However, as can be seen, even with a relatively large decrease in the percent of observations seen at lower altitudes, there is not a significant increase in the tracking error. As expected, the POMDP algorithm was able to provide sufficient estimates of target location during non-observation periods such that there was not a dramatic increase in tracking errors.

Overall, for single target tracking, the downward sensor shows the best overall performance, followed by the right looking sensor. This result is what is expected when you consider that the downward and right facing sensor configurations provide more observations at lower altitudes and provide better geometry for observation updates. Ultimately, the goal of having quality observations for the moving target is met for all configurations. If having the lowest error is a primary consideration, then the downward sensor configuration is likely the best choice for single target tracking.

B. Two Target Tracking

Tracking two targets is well suited for the POMDP algorithm. The increased potential of missed observations due to the fixed FOV and separated targets can be addressed by the POMDP predictions. Figure 3 provides an example of a nominal two target tracking run with a 15 degree forward sensor angle. The plot is derived similarly to Figure 1, except there are two targets being tracked, as indicated by two separate lines, as opposed to one.

Because two targets are being tracked, the errors and observations can be presented separately or averaged. For comparisons in this paper, the average errors for each target were averaged together instead of reported separately. Similarly, the percentage of observations were averaged. Figure 4 shows the cumulative distribution function (CDF) for each altitude, for a downward looking sensor. A CDF provides the representation of how often (frequency) within the data set the average observation was at or less than the observation percent at the point. As can be seen, the CDF for each target is well behaved, as are the averages of the two targets at a given altitude. This behavior supports using the averages

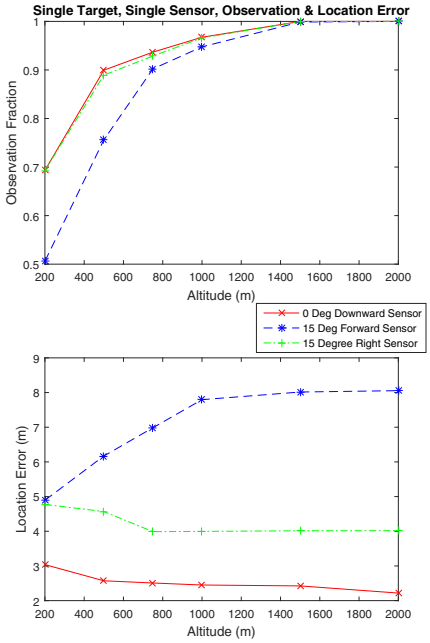

Fig. 2. Average Location Error and Observation Fraction: 1 Target, 1 Sensor, 1000 Runs

of the percent observation and the cumulative target error of both targets.

As was the case in single target tracking, the ability of the POMDP to track two targets shows promise. For all three sensor configurations, as shown in Figure 5, the increase in altitude results in a decrease in average tracking error. This trend is consistent with single target tracking and correlates with the expectation that increased altitude should provide increased observations. However, as can be see in Figure 5, the average percent observation actually decreases as the altitude increases for the downward looking sensor. The algorithm, in this case, attempts to cross over each target to reduce the tracking error for the target, then transverses to the other target. Crossing over and passing, along with the turns, would result in less time in the sensor FOV, especially for a directly downward looking sensor

Overall, for two target tracking, the downward sensor shows the best overall performance, followed by the forward looking sensor. This result is what is expected when you consider that the downward and forward facing sensor configurations provide more simultaneous observations for the two target configuration. If having the lowest error is a primary consideration, then the downward sensor configuration is likely the best choice for single target tracking.

VI. CONCLUSIONS

Overall, the performance of the POMDP algorithm is similar to previous results. The sensor model used here, specifically the use of a limited FOV sensor, is a more realistic use case for small UAV applications that do not have gimbaled sensor capabilities. The ability to track both single

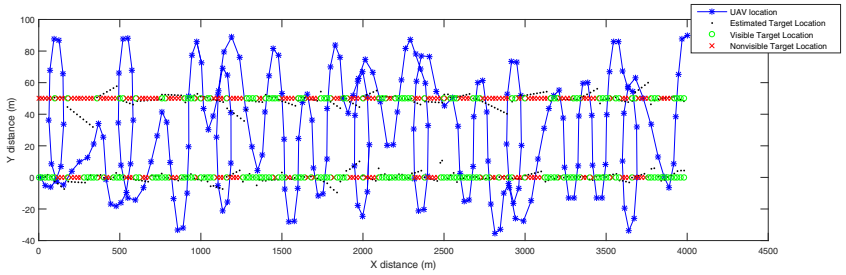

Fig. 3. Tracking example: 2 Targets, 1 Sensor, 200 m, 15 deg fwd sensor

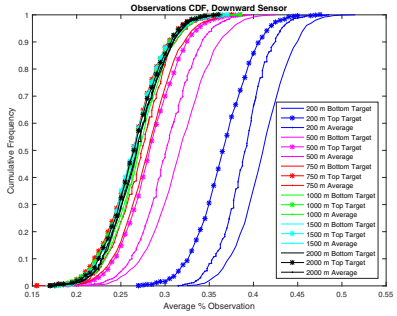

Fig. 4. Tracking performance at various altitudes for each target

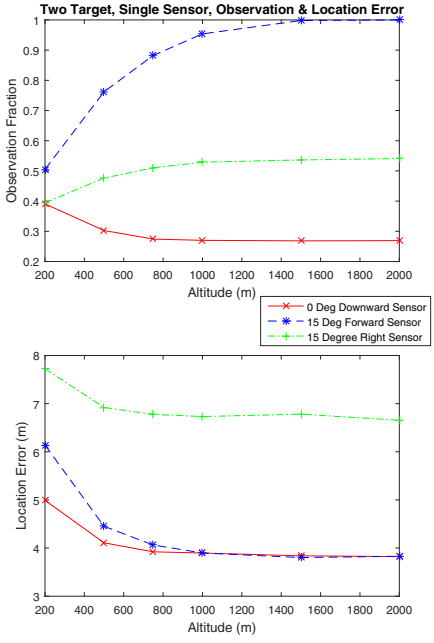

Fig. 5. Average Location Error and Observation Fraction: 2 Target, 1 Sensor, 1000 Runs

or dual moving targets can be performed well with different sensor configurations. The POMDP can address the increased number of missed observations under certain circumstances and still provide adequate predictions of the target location(s) until observations can be made again. While some sensor configurations may be preferred for certain tracking cases, all configurations showed acceptable performance to be able to maintain a track on the target(s).

REFERENCES

[1] C. M. Eaton *et al.*, "Multiple-Scenario Unmanned Aerial System Control: A Systems Engineering Approach and Review of Existing Control Methods," *Aerospace*, vol. 3, no. 1, pp. 1–26, Jan. 2016.

[2] S. Ragi and E. K. P. Chong, "UAV Guidance Algorithms Via Partially Observable Markov Decision Processes," in *Handbook of Unmanned Aerial Vehicles*, K.P. Valavanis, G.J. Vachtsevanos, Eds., Netherlands: Springer, 2015, pp. 1775–1810.

[3] S. Ragi and E. K. P. Chong,"UAV path planning in a dynamic environment via partially observable Markov decision process," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2397–2412, Oct. 2013.

[4] S. A. Miller *et al.*, "A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking," *EURASIP J. Adv. Signal Process.*, vol. 2009, 2009.

[5] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[6] Y. Bar-Shalom *et al.*, *Estimation with Applications to Tracking and Navigation*. Hoboken, NJ: Wiley-Interscience, 2001.

[7] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. Waltham, MA: Academic Press, 1988.

[8] E. K. P. Chong *et al.*, "Partially observable Markov decision process approximations for adaptive sensing," *Discrete Event Dynamic Systems*, vol. 19, 2009, pp. 377–422.

[9] R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.

[10] C. Kreucher *et al.*, "Efficient methods of non-myopic sensor management for multitarget tracking." in *Proc. 43rd IEEE Conf. Decision and Control*, Paradise Island, Bahamas, Dec. 2004, pp. 722–727.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[13] D. P. Bertsekas and D. A. Castanon, "Rollout algorithms for stochastic scheduling problems." *J. Heuristics*, vol. 5, no. 1, Apr. 1999, pp. 89–108.

[14] E. K. P. Chong *et al.*, "A framework for simulation-based network control via hindsight optimization," *Proc. 39th IEEE Conf. Decision and Control*, Sydney, Australia, Dec. 2000, pp. 1433–1438.

[15] G. Wu *et al.*, "Burst-level congestion control using hindsight optimization," *IEEE Trans. Autom. Control*, vol. 47, no. 6, 2002, pp. 979–991.

[16] D.P. Bertsekas, *Dynamic Programming and Optimal Control (vol. 2)*, Belmont, MA: Athena Scientific, 2007.

[17] D.P. Bertsekas, "Dynamic programming and suboptimal control: A survey from ADP to MPC," *European J. on Control: Fundamental Issues on Control*, vol. 11, pp. 4–5, 2005.

[18] Sony FCBEV7500 Camera Block, https://pro.sony.com/bbsc/ssr/cat-camerasindustrial/cat-ciblockcameras/product-FCBEV7500/ accessed Mar. 2017.